

최대의 이익을 위한 최대의 선택 !

LS ELECTRIC에서는 저희 제품을 선택하시는 분들께 최대의 이익을 드리기 위하여
항상 최선의 노력을 다하고 있습니다.

프로그래머블 로직 컨트롤러

모션 제어 모듈 명령어집

XGT Series

사용설명서

XGF-M32E



안전을 위한 주의사항


- 사용 전에 안전을 위한 주의사항을 반드시 읽고 정확하게 사용하여 주십시오.
- 사용설명서를 읽고 난 뒤에는 제품을 사용하는 사람이 항상 볼 수 있는 곳에 잘 보관하십시오.


LS ELECTRIC

제품을 사용하기 전에...


제품을 안전하고 효율적으로 사용하기 위하여 이 사용설명서의 내용을 끝까지 잘 읽으신 후에 사용해 주십시오.


- ▶ 안전을 위한 주의 사항은 제품을 안전하고 올바르게 사용하여 사고나 위험을 미리 막기 위한 것이므로 반드시 지켜 주시기 바랍니다.
- ▶ 주의사항은 ‘경고’ 와 ‘주의’ 의 2가지로 구분되어 있으며, 각각의 의미는 다음과 같습니다.

 **경고** 지시 사항을 위반하였을 때, 심각한 상해나 사망이 발생할 가능성이 있는 경우

 **주의** 지시 사항을 위반하였을 때, 경미한 상해나 제품 손상이 발생할 가능성이 있는 경우

- ▶ 제품과 사용설명서에 표시된 그림 기호의 의미는 다음과 같습니다.

 는 위험을 끼칠 우려가 있는 사항과 조작에 대하여 주의를 환기시키기 위한 기호입니다. 이 기호가 있는 부분은 위험 발생을 피하기 위하여 주의 깊게 읽고 지시에 따라야 합니다.

 는 특정 조건 하에서 감전의 가능성이 있으므로 주의를 나타내는 기호입니다.

- ▶ 사용설명서를 읽고 난 뒤에는 제품을 사용하는 사람이 항상 볼 수 있는 곳에 보관해 주십시오.

설계 시 주의 사항

주의

- ▶ 아날로그 입출력 신호 또는 펄스 입출력선은 고압선이나 동력선과는 최소 100mm 이상 떨어뜨려 노이즈나 자기장 변화에 의한 영향을 받지 않게 설계하여 주십시오.

노이즈에 의한 오동작의 원인이 됩니다.

- ▶ 설치 환경에 진동이 많은 경우에는 PLC에 직접 진동이 인가되지 않도록 조치하여 주십시오.

감전/화재 또는 오동작의 원인이 됩니다.

- ▶ 설치 환경에 금속성 분진이 있는 곳은 제품에 금속성 분진이 유입되지 않도록 조치하여 주십시오.

감전/화재 또는 오동작의 원인이 됩니다.

설치 시 주의 사항

주의

- ▶ PLC는 사용설명서 또는 데이터시트의 일반 규격에 명기된 환경 조건에서 사용해 주십시오.

감전/화재 또는 제품 오동작 및 열화의 원인이 됩니다.

- ▶ 모듈을 장착하기 전에 PLC의 전원이 꺼져 있는지 반드시 확인해 주십시오.

감전, 또는 제품 손상의 원인이 됩니다.

- ▶ PLC의 각 모듈이 정확하게 고정되었는지 반드시 확인해 주십시오.

제품이 느슨하거나 부정확하게 장착되면 오동작, 고장, 또는 낙하의 원인이 됩니다.

배선 시 주의 사항

경고

- ▶ 배선 작업을 시작하기 전에 PLC의 전원 및 외부 전원이 꺼져 있는지 반드시 확인하여 주십시오.

감전 또는 제품 손상의 원인이 됩니다.

주의

- ▶ 각 제품의 정격 전압 및 단자 배열을 확인한 후 정확하게 배선하여 주십시오.

정격과 다른 전원을 접속하거나, 배선을 잘못하면 화재 또는 고장의 원인이 됩니다.

- ▶ 배선시 단자의 나사는 규정 토크로 단단하게 조여 주십시오.

단자의 나사 조임이 느슨하면 단락 또는 오동작의 원인이 됩니다.

- ▶ FG 단자의 접지는 PLC전용 3종 접지를 반드시 사용해 주십시오.

접지하지 않은 경우, 오동작의 원인이 될 수 있습니다.

- ▶ 배선 작업 중 모듈 내로 배선 찌꺼기 등의 이물질이 들어가지 않도록 하여 주십시오.

화재, 제품 손상, 또는 오동작의 원인이 됩니다.

시운전, 보수 시 주의사항

경고

- ▶ 전원이 인가된 상태에서 단자대를 만지지 마십시오.
감전 또는 오동작의 원인이 됩니다..
- ▶ 청소를 하거나, 단자를 조일 때에는 PLC 및 모든 외부 전원을 Off시킨 상태에서 실시하여 주십시오.
감전 또는 오동작의 원인이 됩니다.

주의

- ▶ 모듈의 케이스로부터 PCB를 분리하거나 제품을 개조하지 마십시오.
고장, 오동작, 제품의 손상 및 화재의 원인이 됩니다.
- ▶ 모듈의 장착 또는 분리는 PLC 및 모든 외부 전원을 Off시킨 상태에서 실시하여 주십시오.
감전 또는 오동작의 원인이 됩니다.
- ▶ 무전기 또는 휴대전화는 PLC로부터 30cm 이상 떨어뜨려 사용하여 주십시오.
오동작의 원인이 됩니다.

폐기 시 주의사항

주의

- ▶ 제품을 폐기할 경우, 산업 폐기물로 처리하여 주십시오.
유독 물질의 발생 위험이 있습니다.

Version	일자	주요 변경 내용	수정 Page
V1.3	2021.02	1. 신규 플래그 추가 2. 오타 수정	부록2 5장, 7장, 10장

LS ELECTRIC PLC를 구입하여 주셔서 감사 드립니다.

제품을 사용하시기 이전에 올바른 사용을 위하여 구입하신 제품의 기능과 성능, 설치, 프로그램 방법 등에 대해서 본 사용설명서의 내용을 숙지하여 주시고 최종 사용자와 유지 보수 책임자에게 본 사용설명서가 잘 전달될 수 있도록 하여 주시기 바랍니다.

다음의 사용설명서는 본 제품과 관련된 사용설명서입니다.

필요한 경우, 아래의 사용설명서의 내용을 보시고 주문하여 주시기 바랍니다.

또한, 당사 홈페이지 <http://www.lselectric.co.kr/> 에 접속하여 PDF 파일로 다운로드 받으실 수 있습니다.

관련된 사용설명서 목록

사용설명서 명칭	사용설명서 내용	사용설명서 번호
XGK/XGB 명령어집	XGK, XGB CPU 모듈을 장착한 PLC 시스템에서 사용하는 명령어의 사용 방법 설명 및 프로그래밍하기 위한 사용설명서입니다.	10310000509
XGI/XGR 명령어집	XGI, XGR CPU 모듈을 장착한 PLC 시스템에서 사용하는 명령어의 사용 방법 설명 및 프로그래밍하기 위한 사용설명서입니다.	10310000739
XGK-CPUH/CPUA /CPUS/CPUE/CPUU 사용설명서	XGK CPU 모듈, 전원 모듈, 베이스, 입출력 모듈, 증설 케이블의 각 규격 및 시스템 구성, EMC 규격 대응 등에 대해서 설명한 XGK-CPUH/CPUA/CPUS/CPUE/CPUU 사용설명서입니다.	10310000507
XGI-CPUU/CPUH/CPUS 사용설명서	XGI CPU 모듈, 전원 모듈, 베이스, 입출력 모듈, 증설 케이블의 각 규격 및 시스템 구성, EMC 규격 대응 등에 대해서 설명한 XGI-CPUU/CPUH/CPUS 사용설명서입니다.	10310000738
XGR-CPUH 사용설명서	XGR CPU 모듈, 증설 드라이브 모듈, 전원 모듈, 베이스, 입출력 모듈, 증설 케이블의 각 규격 및 시스템 구성, EMC 규격 대응 등에 대해서 설명한 XGR-CPUH 사용설명서입니다.	10310000919
XG5000 사용설명서 (XGK, XGB용)	XGK, XGB CPU 모듈을 사용하여 프로그래밍, 인쇄, 모니터링, 디버깅과 같은 온라인 기능을 설명한 XG5000 소프트웨어 사용설명서입니다.	10310000511
XG5000 사용설명서 (XGI, XGR용)	XGI, XGR CPU 모듈을 사용하여 프로그래밍, 인쇄, 모니터링, 디버깅과 같은 온라인 기능을 설명한 XG5000 소프트웨어 사용설명서입니다.	10310000746
XG-PM 사용설명서	XGT PLC 시리즈의 CPU에 대응하는 위치 결정모듈을 운영할 수 있는 소프트웨어인 XG-PM의 기능 및 조작 방법에 대해 설명한 사용설명서입니다.	10310000986

◎ 목차 ◎

제 1 장 개요	1-2
1.1 IEC 61131-3 언어의 특징	1-1
1.2 언어의 종류	1-1
제 2 장 소프트웨어 구조	2-1 ~ 2-2
2.1 개요	2-1
2.2 프로젝트(PROJECT)	2-1
2.3 글로벌/직접변수	2-1
2.4 파라미터 및 위치/캠 데이터	2-1
2.5 주 태스크, 주기 태스크, 초기화 태스크	2-1
2.6 사용자 데이터 타입	2-2
제 3 장 공통 요소	3-1 ~ 3-12
3.1 표현	3-1
3.1.1 식별자	3-1
3.1.2 데이터의 표현	3-1
3.2 데이터 타입	3-3
3.2.1 기본 데이터 타입	3-3
3.2.2 데이터 타입 계층도	3-4
3.2.3 초기값	3-4
3.2.4 데이터 타입별 구조	3-4
3.3 변수	3-7
3.3.1 변수의 표현	3-7
3.3.2 변수의 선언	3-8
3.3.3 예약 변수	3-9
3.3.4 예약어	3-10
3.4 명령어 선정	3-11
3.4.1 내부적으로 결정되는 명령어	3-11

제 4 장 LD (LADDER DIAGRAM) 4-1 ~ 4-8

4.1 개요 4-1
 4.2 모션 4-1
 4.3 연결선 4-2
 4.4 접점 4-2
 4.5 코일 4-3
 4.6 평선과 평선 블록의 호출 4-4

제 5 장 평선과 평선 블록 5-1 ~ 5-14

5.1 기본 평선 5-1
 5.1.1 타입 변환 평선 5-1
 5.1.2 수치 연산 평선 5-6
 5.1.3 각도 변환 평선 5-6
 5.1.4 비트열 평선 5-7
 5.1.5 선택 평선 5-7
 5.1.6 데이터 교환 평선 5-7
 5.1.7 비교 평선 5-7
 5.1.8 날짜 시각 평선 5-8
 5.1.9 데이터 처리 평선 5-8
 5.2 기본 평선 블록 5-9
 5.2.1 바이스테이블 평선 블록 5-9
 5.2.2 에지 검출 평선 블록 5-9
 5.2.3 카운터 5-9
 5.2.4 타이머 5-9
 5.3 확장 평선 5-10
 5.4 모션 제어 평선 블록 5-10

제 6 장 기본 평선 6-1 ~ 6-122

제 7 장 응용 평선 7-1 ~ 7-38

제 8 장 기본 평선 블록.....	8-1 ~ 8-22
제 9 장 응용 평선 블록.....	9-1 ~ 9-22
제 10 장 모션 평선 블록.....	10-1 ~ 10-178
제 11 장 확장 평선.....	11-1 ~ 11-6
제 12 장 ST (STRUCTURED TEXT)	12-1 ~ 12-26
12.1 개요	12-1
12.2 비 실행문(설명문)	12-1
12.3 표현식	12-2
12.3.1 + 연산자	12-3
12.3.2 - 연산자	12-3
12.3.3 * 연산자	12-4
12.3.4 / 연산자	12-4
12.3.5 MOD 연산자	12-5
12.3.6 ** 연산자	12-6
12.3.7 AND, & 연산자	12-6
12.3.8 OR 연산자	12-7
12.3.9 XOR 연산자	12-7
12.3.10 = 연산자	12-8
12.3.11 <> 연산자	12-8
12.3.12 > 연산자	12-9
12.3.13 < 연산자	12-10
12.3.14 >= 연산자	12-10
12.3.15 <= 연산자	12-11
12.3.16 NOT 연산자	12-11
12.3.17 - 연산자	12-12
12.4 명령문	12-12
12.4.1 할당문	12-12

12.4.2 선택문 12-12

12.4.3 반복문 12-13

12.4.4 IF문 12-15

12.4.5 CASE문 12-16

12.4.6 FOR문 12-17

12.4.7 WHILE문 12-18

12.4.8 REPEAT문 12-18

12.4.9 EXIT문 12-19

12.5 평선 및 평선 블록 12-20

12.5.1 사용 방법 12-20

12.5.2 사용예 12-24

부록 1 수치체계 및 데이터 구조 부1-1 ~ 부1-6

부1.1 수치(데이터)의 표현 부1-1

부1.2 정수 표현 부1-6

부1.3 음수의 표현 부1-6

부록 2 플래그 일람(XGI) 부2-1 ~ 부2-8

부2.1 시스템 플래그 부2-1

부2.2 모션 플래그 부2-3

부2.3 입출력 플래그 부2-7

부2.4 모션 좌표계 플래그 부2-8

환경 방침과 보증 내용

제1장 개요

이 설명서는 모션 제어 모듈에서 지원하는 언어에 대한 설명서입니다.

모션 제어 모듈은 IEC (International Electrotechnical Commission - 국제전기표준회의)에서 국제 표준으로 발표한 언어를 기본으로 합니다.

1.1. IEC 61131-3 언어의 특징

IEC 언어에서 새로 도입한 가장 중요한 특징들은 다음과 같습니다.

- ▷ 다양하고 강력한 데이터 타입을 지원합니다.
- ▷ 평선, 평선 블록, 프로그램 같은 프로그램 구성 요소가 도입되어 상향식, 또는 하향식 설계가 가능하며, PLC 프로그램을 구조적으로 작성할 수 있습니다.
- ▷ 사용자가 작성한 프로그램을 다른 환경에서 사용할 수 있어 소프트웨어의 재사용을 가능하게 합니다.
- ▷ 다양한 언어를 지원하므로 사용자는 적용환경에 최적의 언어를 선택하여 사용할 수 있습니다.

1.2. 언어의 종류

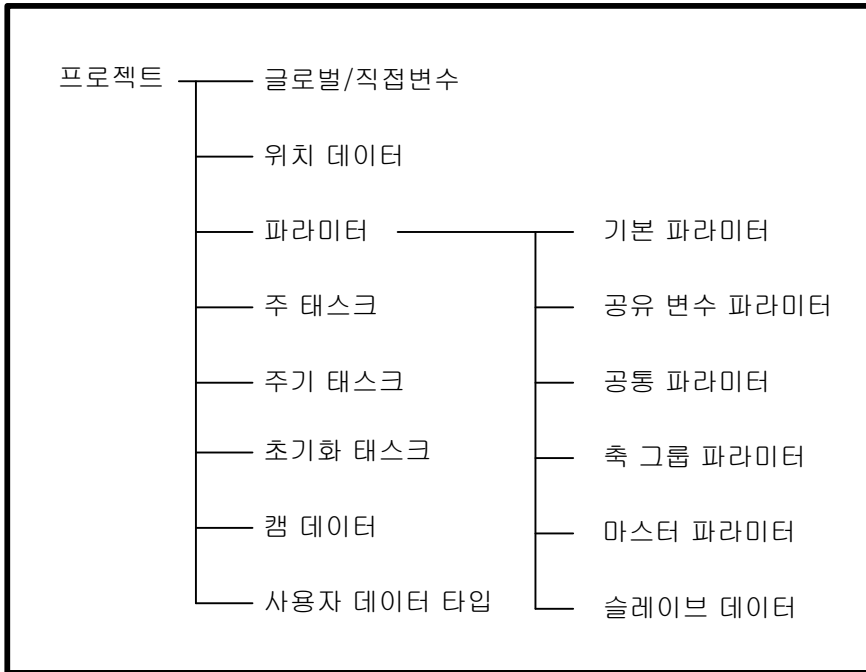
IEC에서 표준화된 PLC용 언어는 두 개의 도식 언어와 두 개의 문자식 언어, 그리고 SFC로 이루어져 있습니다.(XGF-M32E 모션 제어 모듈은 LD, ST 언어를 지원합니다.)

- ▷ 도식 언어
 - a) LD (Ladder Diagram): 릴레이 로직 표현 방식의 언어
 - b) FBD (Function Block Diagram): 블록화한 기능을 서로 연결하여 프로그램을 표현하는 언어
- ▷ 문자식 언어
 - a) IL (Instruction List): 어셈블리 언어 형태의 언어
 - b) ST (Structured Text): 파스칼 형식의 고 수준 언어
- ▷ SFC (Sequential Function Chart)

제2장 소프트웨어 구조

2.1. 개요

모션 제어 모듈의 프로그램을 작성하기 전에 소프트웨어 측면에서 전체적인 모션 제어 시스템을 구성합니다. 모션 제어 모듈에서는 PLC 시스템 전체를 하나의 프로젝트로 정의합니다. 프로젝트 안에는 하나의 모션 제어 시스템에 필요한 모든 것이 계층적으로 정의되어 있습니다.



2.2. 프로젝트(Project)

- ▶ 모션 제어 모듈의 프로그램을 작성하기 위해서는 우선 프로젝트를 구성하여야 합니다. 하나의 프로젝트를 구성한다는 것은 하나의 모션 제어 시스템에 필요한 모든 구성 요소를 작성한다는 의미입니다. 즉, 가장 기본적인 주 태스크 뿐 만 아니라 주기 태스크, 기본 파라미터, 위치/캠 데이터 등을 작성합니다.

2.3. 글로벌/직접 변수

- ▶ 글로벌 변수, 직접 변수 설명문, 플래그 부분의 탭으로 보여주며 사용자가 필요한 정보를 작성 또는 사용하는 부분입니다.

2.4. 파라미터 및 위치/ 캠 데이터

- ▶ 파라미터 및 위치/캠 데이터는 모션 제어 시스템을 구동하는데 필요한 각종 파라미터 및 데이터를 나타내며 해당 파라미터 및 데이터에 대한 자세한 내용은 모션 모듈 사용설명서를 참고하시기 바랍니다.

2.5. 주 태스크, 주기 태스크, 초기화 태스크

- ▶ 주 태스크는 제어주기 및 통신주기 마다 실행되는 명령을 처리하는 태스크입니다.
- ▶ 주기 태스크는 제어주기의 배수로 설정된 주기마다 실행되는 명령을 처리하는 태스크입니다.
- ▶ 초기화 태스크는 모듈의 운전 모드가 스톱에서 런으로 변경될 때 실행하는 태스크입니다.
- ▶ 각 태스크에 대한 자세한 내용은 모션 모듈 사용설명서(4.3 모션 제어 태스크)를 참고하시기 바랍니다.

2.6. 사용자 데이터 타입

- ▷ 프로그램에서 사용할 데이터 타입을 사용자가 직접 설정할 수 있습니다. 변수, 타입, 설명문 등을 작성하여 등록할 수 있습니다. 등록된 사용자 데이터 타입은 기본적인 데이터 타입과 동일하게 프로그램에서 사용 가능합니다.

제3장 공통 요소

모션 제어 모듈의 프로그램 구성 요소(프로그램, 평선, 평선 블록)는 LD, ST 언어 등으로 작성할 수 있습니다. 이 언어들은 공통적으로 사용하는 문법 요소들을 가지고 있습니다.

3.1. 표현

3.1.1. 식별자(Identifiers)

- ▷ 영문자나 밑줄 문자(_)로 시작하는 모든 문자, 숫자, 밑줄 문자의 조합이 식별자가 될 수 있습니다.
- ▷ 식별자는 변수의 이름으로 쓰입니다.
- ▷ 식별자는 빈 칸(Space)을 포함하지 않아야 합니다.
- ▷ 식별자는 보통 변수 또는 인스턴스 이름인 경우에는 한글, 영문, 한자 모두 제한이 없습니다.
- ▷ 영문자의 경우, 대·소문자를 구별하지 않고 모두 대문자로 인식합니다.

종 류	사 용 예
대문자와 숫자	IW210, IW215Z, QX75, IDENT
대문자와 숫자, 밑줄 글자	LIM_SW_2, LIMSW5, ABCD, AB_CD
밑줄 글자로 시작하는 대문자와 숫자	_MAIN, _12V7, _ABCD

3.1.2. 데이터의 표현

모션 제어 모듈에서 데이터로 사용하는 것은 숫자(Numeric Literals)와 문자열(Character String), 시간 문자(Time Literals) 등입니다.

종 류	사 용 예
정 수	-12, 0, 123_456, +986
실 수	-12.0, 0.0, 0.456, 3.14159_26
승수부를 갖는 실수	-1.34E-12, 1.0E+6, 1.234E6
2 진수	2#1111_1111, 2#11100000
8 진수	8#377(십진수 255) 8#340(십진수 224)
16 진수	16#FF(십진수 255) 16#E0(십진수 224)
BOOL 데이터	0, 1, TRUE, FALSE

1) 숫자(Numeric Literals)

- ▷ 숫자에는 정수(Integer Literals)와 실수(Real Literals)가 있습니다.
- ▷ 연속되지 않은 밑줄 글자(_)가 숫자 사이에 올 수 있으며 그 의미는 무시됩니다.
- ▷ 십진수는 일반적인 십진 표현법을 따르고 소수점이 있으면 실수로 구별됩니다.
- ▷ 승수(Exponent) 표현 시 +, -의 부호가 올 수 있습니다. 승수부를 구분하는 문자 'E'는 대소문자를 구분하지 않습니다.
- ▷ 승수부가 있는 실수의 사용시 다음은 가능하지 않습니다.

예) 12E-5 (×) 12.0E-5 (○)

- ▷ 정수에는 십진수 이외에 2,8,16 진수가 올 수 있으며, 숫자의 앞부분에 진수 #을 사용하여 구분합니다. 아무것도 불

이지 않으면 십진수로 간주합니다.

- ▷ 16 진수 표현 시 0 - 9, A - F 를 쓰며 소문자 a - f 도 쓸 수 있습니다.
- ▷ 16 진수 표현 시에는 부호(+, -)가 올 수 없습니다.
- ▷ BOOL 데이터(Boolean Data)는 정수 0 과 1 로도 표현할 수 있습니다.

2) 문자열(Character String)

- ▷ 작은 따옴표(')로 둘러싸인 모든 문자가 문자열에 해당됩니다.
- ▷ 그 길이는 문자열 상수일 때에는 31 자 이내이며, 초기화에 사용할 때 역시 31 자로 제한합니다.

예) 'CONVEYER'

3) 시간 문자(Time Literals)

- ▷ 시간 문자는 제어 사건(Control Event)의 경과 시간(Elapsed Time)을 재거나 조절하기 위한 경과 시간(Duration) 데이터와, 제어 사건의 시작점과 끝점의 시각을 표시하기 위한 날짜와 시각(Time Of Day And Date) 데이터로 구분됩니다.

(1) 경과 시간(Duration)

- ▷ 경과 시간 데이터는 예약어 'T#' 또는 't#'으로 시작합니다.
- ▷ 일(d), 시(h), 분(m), 초(s), ms 의 순으로 써야 하고 어느 단위에서 시작되어도 상관없으며, 최소 단위인 ms 까지 꼭 쓰지 않아도 되나 중간 단위를 생략할 수는 없습니다.
- ▷ 밑줄 글자(_)는 사용하지 않습니다.
- ▷ 최대 단위에서의 오버플로(Overflow)는 허용되며, 최소 단위에서의 소수점 이하 표현도 ms 이외에는 가능합니다. 단 최대는 T#49d17h2m47s295ms 을 초과할 수 없습니다. (즉 ms 단위로 32 비트)
- ▷ 소수점 이하 자릿수의 제한은 현재 초(s)단위에서의 3 자리까지입니다.
- ▷ ms 단위에서는 소수점이 올 수 없습니다.
- ▷ 단위를 나타내는 문자로는 대·소문자 어느 경우나 다 가능합니다.

내 용	사 용 예
경과 시간(Under line 없음)	T#14ms, T#14.7s, T#14.7m, T#14.7h t#14.7d, t#25h15m, t#5d14h12m18s356ms

(2) 날짜와 시각(Time Of Day And Date)

- ▷ 날짜와 시각의 표현 방법에는 날짜, 시각, 날짜와 시각의 3가지가 있으며 다음과 같습니다.

내 용	접두 예약어
날짜 접두어	D#
시각 접두어	TOD#
날짜 시각 접두어	DT#

- ▷ 날짜의 시작점은 1984년 1월 1일을 기점으로 합니다.
- ▷ 시각과 날짜 시각의 표현에는 엄격한 자릿수의 제한이 있으며, 초를 나타낼 경우 ms 단위는 소수점 이하 세 자리까지 가능합니다. (1ms 단위)
- ▷ 시각과 날짜 시각의 표현 시에는 모든 단위에서 오버플로(Overflow)가 허용되지 않습니다.

내 용	사 용 예
날짜	D#1984-06-25 d#1984-06-25
시각	TOD#15:36:55.36 tod#15:36:55.369
날짜 시각	DT#1984-06-25-15:36:55.36 dt#1984-06-25-15:36:55.369

3.2. 데이터 타입

데이터는 그 데이터의 고유 성질을 나타내는 데이터 타입을 가지고 있습니다.

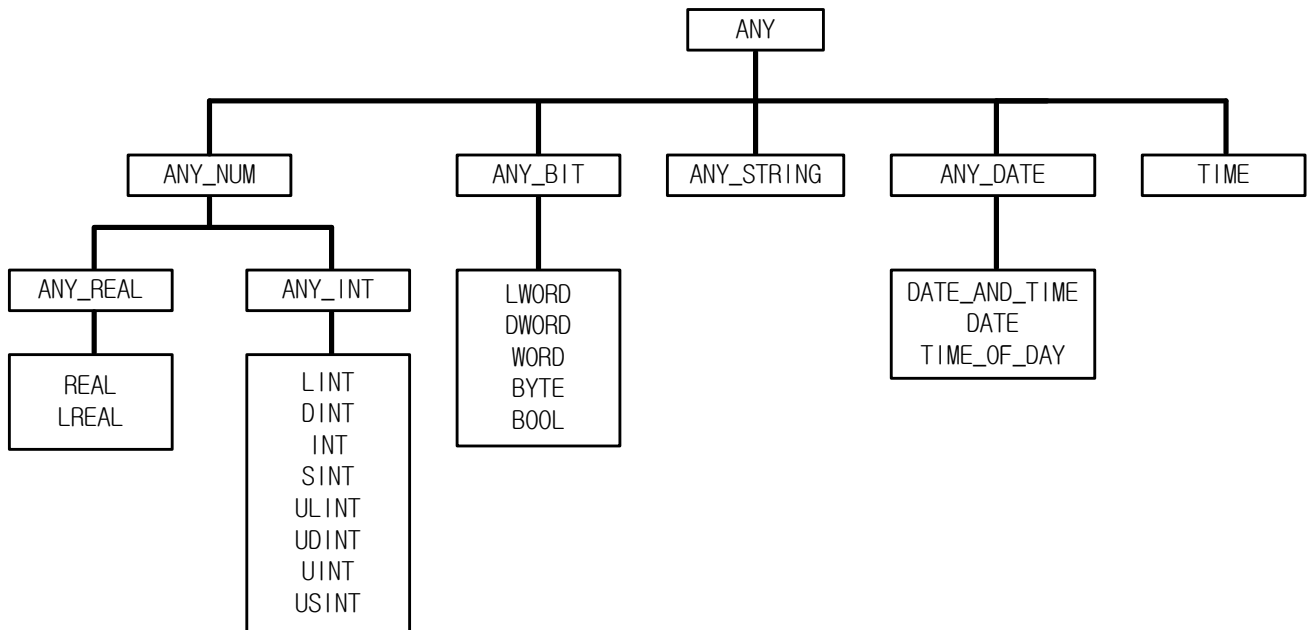
3.2.1. 기본 데이터 타입

모션 제어 모듈에서는 다음의 기본 데이터 타입을 지원합니다.

번호	예 약 어	데이터 타입	크기 (비트)	범 위
1	SINT	Short Integer	8	-128 ~ 127
2	INT	Integer	16	-32,768 ~ 32,767
3	DINT	Double Integer	32	-2,147,483,648 ~ 2,147,483,647
4	LINT	Long Integer	64	$-2^{63} \sim 2^{63}-1$
5	USINT	Unsigned Short Integer	8	0 ~ 255
6	UINT	Unsigned Integer	16	0 ~ 65,535
7	UDINT	Unsigned Double Integer	32	0 ~ 4,294,967,295
8	ULINT	Unsigned Long Integer	64	$0 \sim 2^{64}-1$
9	REAL	Real Numbers	32	-3.402823466e+038 ~ -1.175494351e-038 or 0 or 1.175494351e-038 ~ 3.402823466e+038
10	LREAL	Long Real Numbers	64	-1.7976931348623157e+308 ~ -2.2250738585072014e-308 or 0 or 2.2250738585072014e-308 ~ 1.7976931348623157e+308
11	TIME	Duration	32	T#0S ~ T#49D17H2M47S295MS
12	DATE	Date	16	D#1984-01-01 ~ D#2163-6-6
13	TIME_OF_DAY	Time Of Day	32	TOD#00:00:00 ~ TOD#23:59:59.999
14	DATE_AND_TIME	Date And Time Of Day	64	DT#1984-01-01-00:00:00 ~ DT#2163-12-31-23:59:59.999
15	STRING	Character String	32*8	-
16	BOOL	Boolean	1	0, 1
17	BYTE	Bit String Of Length 8	8	16#0 ~ 16#FF
18	WORD	Bit String Of Length 16	16	16#0 ~ 16#FFFF
19	DWORD	Bit String Of Length 32	32	16#0 ~ 16#FFFFFFFF
20	LWORD	Bit String Of Length 64	64	16#0 ~ 16#FFFFFFFFFFFFFFFF

3.2.2. 데이터 타입 계층도

모션 제어 모듈에서 사용되는 데이터 타입은 다음과 같습니다.



- ▶ 앞으로 데이터 타입을 표현할 때, ANY_NUM 으로 나타내면 다음 계층도와 같이 LREAL, REAL, LINT, DINT, INT, SINT, ULINT, UDINT, UINT, USINT 를 모두 포함합니다.
- ▶ 예를 들어 타입이 ANY_BIT 로 표현되면 LWORD, DWORD, WORD, BYTE, BOOL 중 하나를 사용할 수 있습니다.

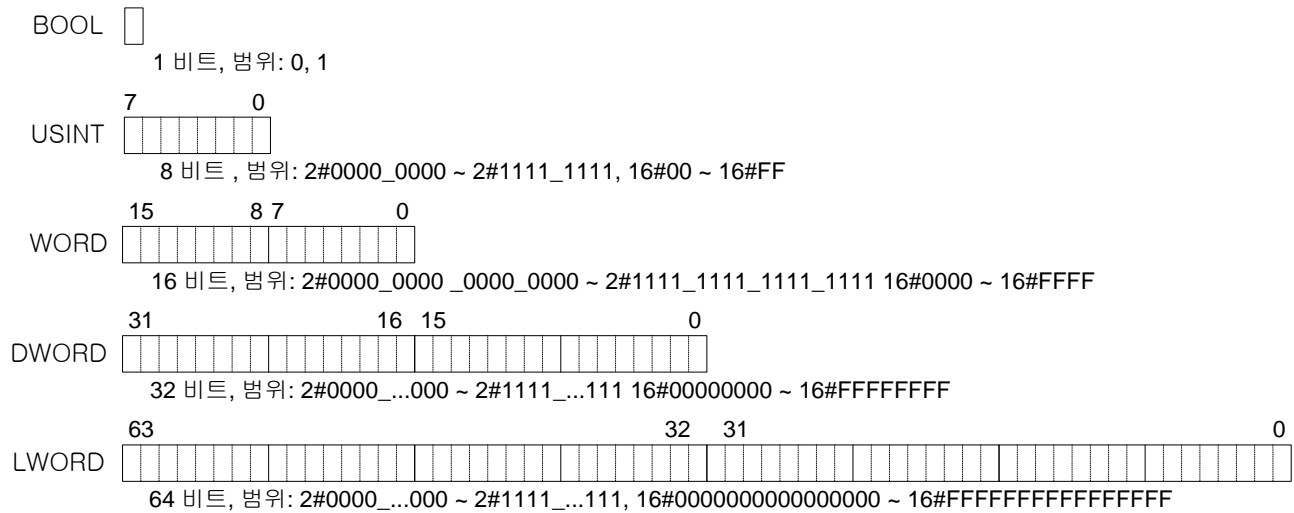
3.2.3. 초기값

데이터의 초기값을 지정하지 않으면 자동적으로 아래와 같이 지정됩니다.

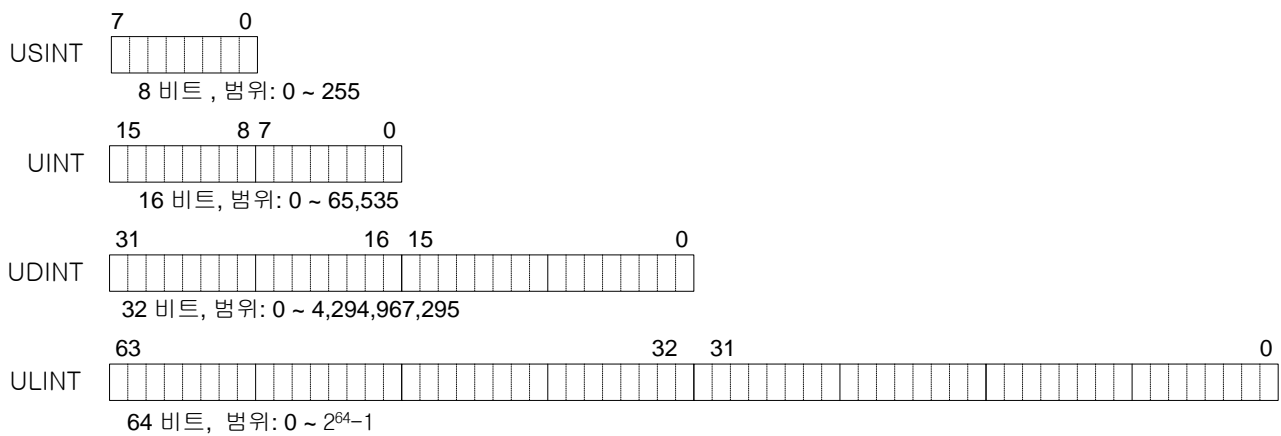
데이터 타입	초기값
SINT, INT, DINT, LINT	0
USINT, UINT, UDINT, ULINT	0
BOOL, BYTE, WORD, DWORD, LWORD	0
REAL, LREAL	0.0
TIME	T#0s
DATE	D#1984-01-01
TIME_OF_DAY	TOD#00:00:00
DATE_AND_TIME	DT#1984-01-01-00:00:00
STRING	'' (empty string)

3.2.4. 데이터 타입별 구조

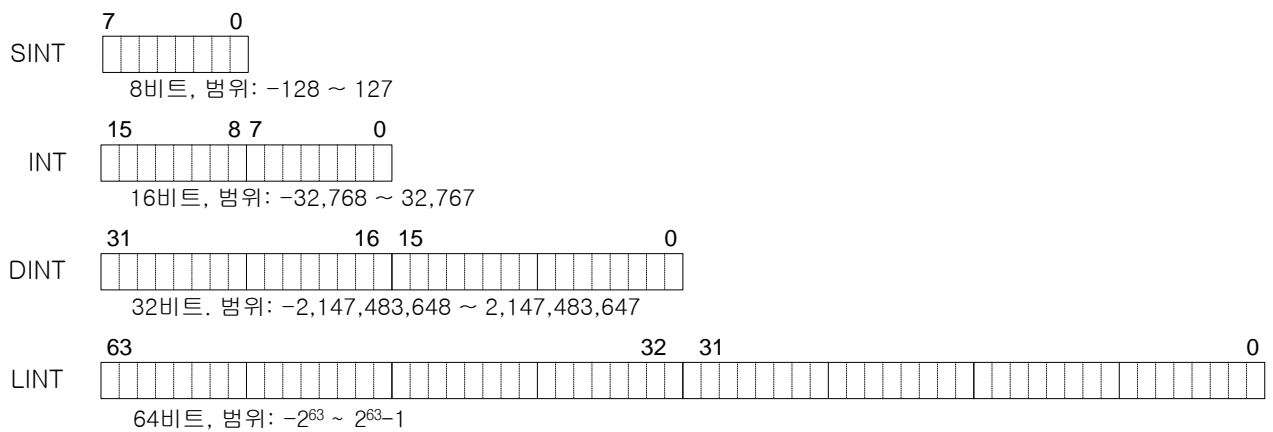
Bit String



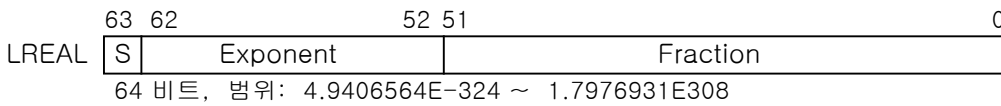
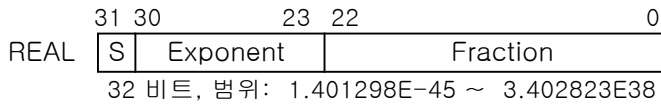
Unsigned Integer



Integer (음수는 2' Complement 표현)

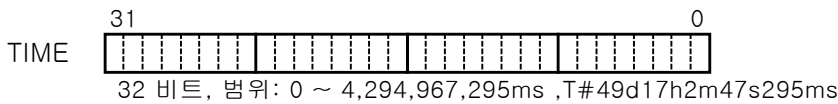


Real (IEEE Standard 754-1984 기준)

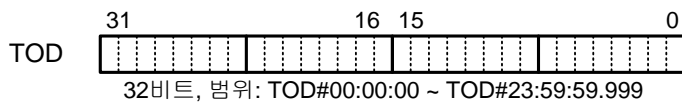
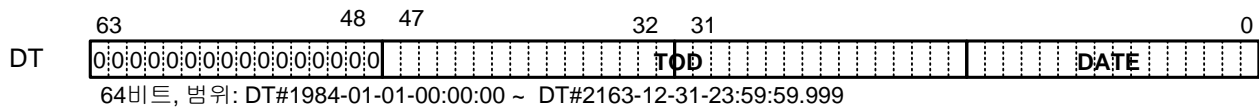


- S: 부호표시 (0 일 때 양수, 1 일 때 음수)
- Exponent: 2의 승수부 (2^{e-127} : $e=b_{30}b_{29}...b_{23}$, $e=b_{62}b_{61}...b_{52}$)
- Fraction: 소수점 이하 값 (Fraction: $f=b_{22}b_{21}...b_0$, $f=b_{51}b_{52}...b_0$)

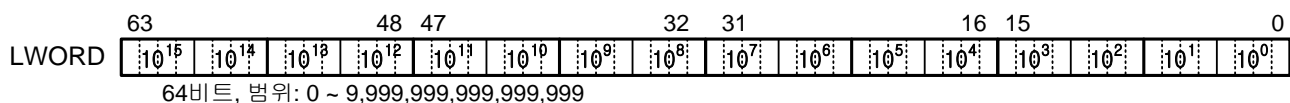
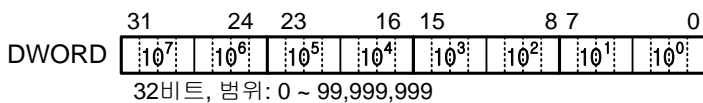
Time



Date



#BCD



3.3. 변수

변수란 프로그램 안에서 사용하는 데이터로서 값을 가지고 있습니다. 변수는 PLC 의 입력이나 출력, 내부 메모리 등과 같이 변할 수 있는 대상을 가리킵니다.

3.3.1. 변수의 표현

- ▷ 변수의 표현에는 2 가지가 있습니다.
 식별자에 의한 변수: 식별자에 의해 변수에 이름을 부여하는 것
 직접변수: PLC 의 입 · 출력 또는 기억 장소에 대하여 직접적으로 표현하는 것
- ▷ 식별자에 의한 변수는 다른 변수들과 구별하기 위하여 그 이름의 변수가 선언된 프로그램 안에서 유일해야 합니다.
- ▷ 직접 변수의 표현은 퍼센트 문자(%)를 시작으로 위치를 나타내는 접두어와 데이터의 크기를 나타내는 접두어 그리고 마침표로 분리되는 하나 이상의 부호 없는 정수의 순으로 표현할 수 있습니다. 그 접두어들은 다음에 나타나 있습니다.

위치 접두어

번호	접두어	의 미
1	I	입력 위치(Input Location)
2	Q	출력 위치(Output Location)
3	G	내부 메모리 중 G 영역 위치(Memory Location)
4	D	내부 메모리 중 D 영역 위치(Memory Location)
5	M	내부 메모리 중 M 영역 위치(Memory Location)
6	P	내부 메모리 중 P 영역 위치(Memory Location)

크기 접두어

번호	접두어	의 미
1	X	1 비트의 크기
2	None	1 비트의 크기
3	B	1 바이트(8 비트)의 크기
4	W	1 워드(16 비트)의 크기
5	D	1 더블 워드(32 비트)의 크기
6	L	1 롱 워드(64 비트)의 크기

표현 형식

%[위치 접두어][크기 접두어] n1.n2.n3

번호	I, Q	G, D, M, P
n1	0: 내장 입출력 1~36: 축 번호	[크기 접두어]에 따른 n1 번째 데이터 (0 부터 시작)
n2	[크기 접두어]에 따른 n2 번째 데이터	n1 번째 데이터상의 n2 번째 비트 (0 부터 시작) : 생략 가능

예

%QX0.1	내장 입출력의 1번 출력(1 비트)
%IW2.1	2번 축의 1번 입력(16 비트)
%MD48	48의 위치에 있는 더블 워드 단위의 메모리
%MW40.3	40의 위치에 있는 워드 단위의 메모리 중 3번 비트 (내부 메모리는 베이스, 슬롯 등의 개념이 없음)

- ▷ 접두어로는 소문자가 올 수 없습니다.
- ▷ 크기 접두어를 붙이지 않으면 그 변수는 1비트로 처리합니다.
- ▷ 직접변수는 선언하지 않고 사용할 수 있습니다.

3.3.2. 변수의 선언

- ▷ 프로그램 구성 요소(즉 프로그램, 평선, 평선 블록)는 그 구성 요소에서 사용할 변수를 선언할 수 있는 선언 부분을 가지고 있습니다.
- ▷ 프로그램 구성 요소에서 변수를 사용하기 위해서는 우선 사용할 변수를 선언해야 합니다.
- ▷ 변수의 선언에서 설정해야 할 사항은 다음과 같습니다.

1) 변수 종류 : 변수를 어떻게 선언할 것인가를 설정합니다.

변수종류	내 용
VAR	읽고 쓸 수 있는 일반적인 변수
VAR_CONSTANT	읽기만 할 수 있는 변수
VAR_EXTERNAL	VAR_GLOBAL 로 선언된 변수를 사용하기 위한 선언
VAR_EXTERNAL_CONSTANT	VAR_GLOBAL 로 선언된 변수를 사용하기 위한 선언, 읽기만 할 수 있는 변수

2) 데이터 타입 : 변수의 데이터 타입을 지정합니다.

3) 메모리 할당 : 변수가 차지할 메모리를 할당합니다.

자동 — 컴파일러가 변수의 위치를 자동으로 지정(자동 배치 변수).

사용자 정의(AT) — 사용자가 직접표현 변수를 사용하여 강제로 위치를 지정(직접변수).

참고

자동 배치 변수는 그 실제 위치가 고정되어 있지 않습니다. 예를 들어 VAL1 이란 변수를 BOOL 데이터타입으로 선언하였다면 그 변수가 내부 데이터 영역의 어느 위치에 있는지 고정되어 있지 않다는 것입니다. 그 위치는 프로그램을 다 작성한 후 컴파일러와 링커에 의해 정해집니다. 만약 프로그램을 수정한 후에 다시 컴파일 하였다면 그 위치가 변할 수 있습니다. 자동 배치 변수의 장점은 사용자가 내부 변수로 사용하는 것들의 위치에 신경 쓰지 않아도 된다는 것입니다. 다른 이름으로 선언한 변수들은 결코 데이터 메모리에 중복되어 위치하지 않기 때문입니다. 직접변수는 변수의 위치가 정해지기 때문에 %I 와 %Q 를 제외하고는 될 수 있으면 사용하지 않는 것이 좋습니다. 직접변수는 자동 배치 변수가 아니므로 사용자가 잘못 사용할 경우, 중복될 수 있습니다.

- ▷ 초기값(Initial Value) 지정 : 변수의 초기값을 지정합니다. 지정하지 않으면 3.2.3. 항의 초기값으로 지정됩니다.

참고

VAR_EXTERNAL 의 선언 시에는 초기값을 줄 수 없습니다.

변수 선언 시 G 영역, D 영역, M 영역을 할당한 변수에는 초기값을 입력할 수 없습니다.

- ▷ PLC 의 전원이 끊긴 후에도 데이터의 값을 유지할 필요가 있는 변수는 정전 유지(Retention)의 기능을 사용해야 합니다. 이 기능은 리테인 체크 박스를 설정하여 사용할 수 있으며 다음의 규칙을 따릅니다.
 - 1) 정전 유지 변수는 시스템의 웜 리스타트 시 그 값이 유지됩니다.
 - 2) 시스템의 콜드 리스타트 시에는 사용자가 정의한 초기값이나 기본 초기값으로 초기화됩니다.
- ▷ 리테인으로 선언되지 않은 변수는 콜드 리스타트나 웜 리스타트 어느 경우에도 사용자가 정의한 초기값이나 기본 초기값으로 초기화됩니다.
- ▷ 변수는 기본 데이터 타입을 인자로 갖는 어레이로 선언하여 사용할 수 있습니다. 어레이 변수로 선언할 때에는 인자로 사용할 데이터의 타입과 어레이의 크기를 설정하여야 합니다. 단, 기본 데이터 타입 중에 STRING 데이터 타입은 인자로 설정할 수 없습니다.
- ▷ 변수 선언의 유효 영역(Scope), 즉 변수를 사용할 수 있는 영역은 그 변수가 선언된 프로그램 구성 요소에 한합니다. 따라서 다른 프로그램 구성 요소에서 선언된 변수는 사용할 수 없습니다. 글로벌 변수로 선언된 변수는 이와 달리 모든 곳에서 VAR_EXTERNAL 선언에 의해 변수 접근이 가능합니다.

변수의 선언 예

변수 이름	변수형	타입	초기값	메모리 할당
I_VAL	VAR	INT	1234	자동
BIPOLAR	VAR_RETAIN	REAL	-	자동
LIMIT_SW	VAR	BOOL	-	%MX10
GLO_SW	VAR_EXTERNAL	DWORD	-	자동
READ_BUF	VAR	ARRAY OF INT[10]	-	자동

3.3.3. 예약 변수

- ▷ 예약 변수는 시스템에서 미리 선언한 변수들로서 플래그로 사용됩니다. 사용자가 이 변수 이름으로 변수 선언을 할 수는 없습니다.
- ▷ 이 예약 변수를 사용할 때에는 변수 선언 없이 사용합니다.
- ▷ 자세한 사항은 부록 2 '플래그 일람' 을 참조하시기 바랍니다.

3.3.4. 예약어

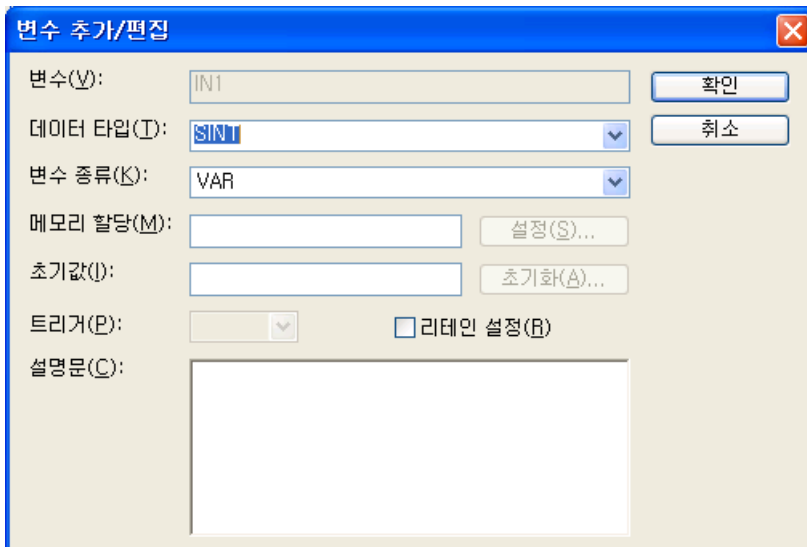
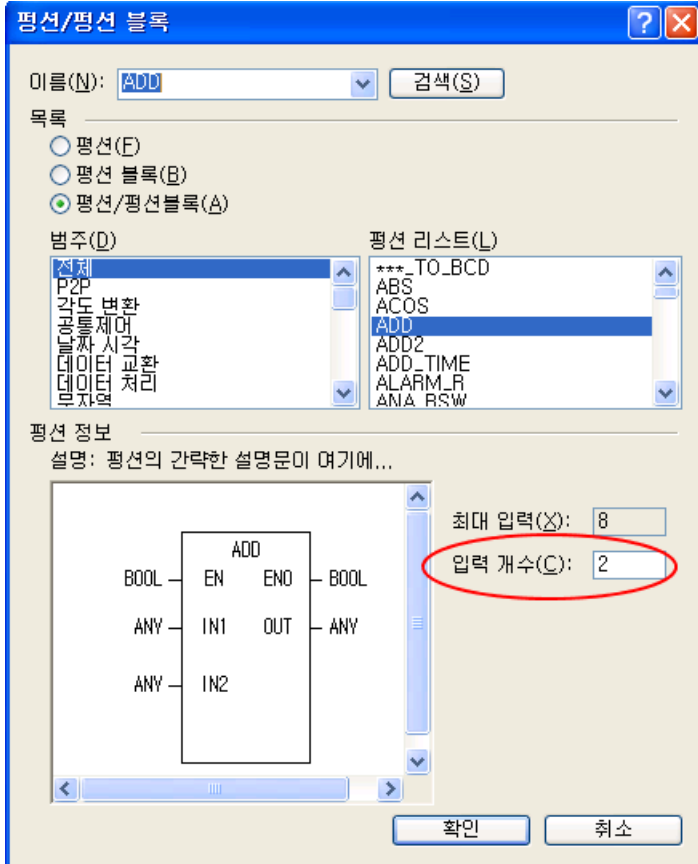
예약어는 시스템에서 사용하기 위해 미리 정의한 단어입니다. 따라서 식별자로 이 예약어를 사용할 수는 없습니다.

예 약 어
ACTION ... END_ACTION
ARRAY ... OF
AT
CASE ... OF ... ELSE ... END_CASE
CONFIGURATION ... END_CONFIGURATION
데이터 타입 이름
DATE#, D#DATE_AND_TIME#, DT#
EXIT
FOR ... TO ... BY ... DO ... END_FOR
FUNCTION ... END_FUNCTION
FUNCTION_BLOCK ... END_FUNCTION_BLOCK
평선 블록의 이름들
IF ... THEN ... ELSIF ... ELSE ... END_IF
OK
연산자 (IL 언어)
연산자 (ST 언어)
PROGRAM
PROGRAM ... END_PROGRAM
REPEAT ... UNTIL ... END_REPEAT
RESOURCE ... END_RESOURCE
RETAIN
RETURN
STEP ... END_STEP
STRUCTURE ... END_STRUCTURE
T#
TASK ... WITH
TIME_OF_DAY#, TOD#
TRANSITION ... FROM... TO ... END_TRANSITION
TYPE ... END_TYPE
VAR ... END_VAR
VAR_INPUT ... END_VAR
VAR_OUTPUT ... END_VAR
VAR_IN_OUT ... END_VAR
VAR_EXTERNAL ... END_VAR
VAR_ACCESS ... END_VAR
VAR_GLOBAL ... END_VAR
WHILE ... DO ... END_WHILE
WITH

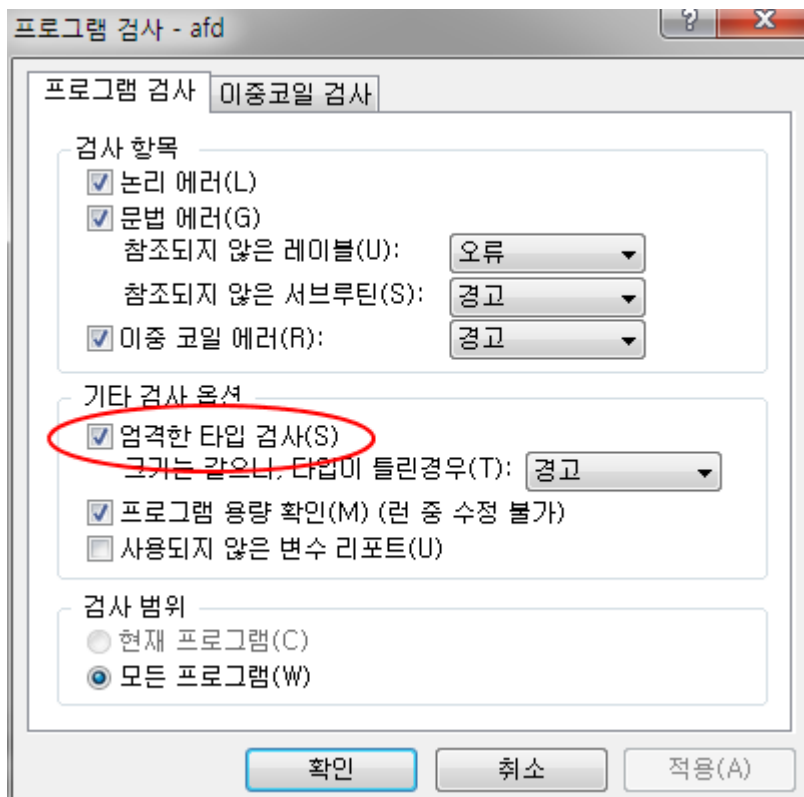
3.4. 명령어 선정

3.4.1. 내부적으로 결정되는 명령어

- ▶ 펄스의 경우 하나의 이름을 갖지만 여러 종류의 변수타입을 입력할 수 있는 명령어는 사용되는 변수에 따라서 내부적으로 여러 명령어로 구분됩니다. 예를 들어 ADD 의 경우, 설정한 입력개수 및 입력/출력 변수 타입에 따라서 여러 종류로 구분되어 처리됩니다. 아래와 같이 선택한 경우에는 래더 프로그램 상에서 보이는 명령어는 ADD 이지만, 내부적으로 ADD2_SINT 명령어를 수행하게 되는 것입니다.



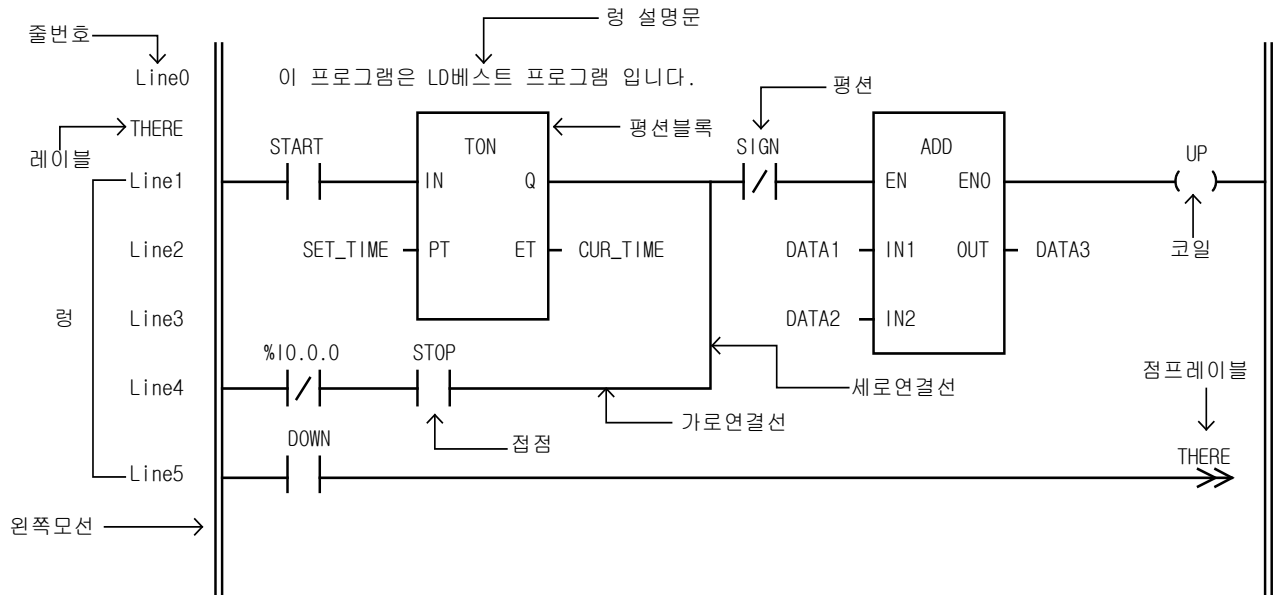
- ▷ 내부적으로 사용되는 명령어의 선정은 사용자가 선택한 변수 타입에 따라 XG-PM 에서 자동 선정하게 됩니다. 예를 들어 ADD 명령 중 입력개수 2 개를 선택하고, 입력과 출력변수를 모두 DINT 로 선택하면 위에서 설명한 것처럼 ADD2_DINT 선정됩니다.
- ▷ IEC 에서는 같은 타입끼리의 연산만을 허용하고 있으나, XG-PM 에서는 검사옵션에 엄격한 검사 옵션을 두어 변수의 크기(BYTE, WORD, DWORD, LWORD)만 같으면 연산을 허용하는 옵션이 있습니다.



제4장 LD (Ladder Diagram)

4.1. 개요

- ▶ LD 프로그램은 릴레이 로직 다이어그램에서 많이 사용하는 코일이나 접점 등의 그래픽 기호를 통하여 PLC 의 프로그램을 표현하는 것입니다.
- ▶ 형태




4.2. 모션

- ▶ LD 그래픽 구성도의 왼쪽 끝과 오른쪽 끝에는 전원선 개념의 모션이 세로로 양쪽에 놓여 있게 됩니다.

No	기호	이름	설명
1		왼쪽 모션	언제나 BOOL 1의 값을 가지고 있습니다.
2		오른쪽 모션	값은 정해지지 않습니다



4.3. 연결선

- ▷ 왼쪽 모선의 BOOL 1 값은 작성한 도면에 따라 오른쪽으로 전달됩니다. 그 전달되는 값을 가진 선을 전원 흐름선 또는 연결선이라고 하며, 접점이나 코일에 연결되어 있는 선입니다. 전원 흐름선은 언제나 BOOL 값을 가지고 있으며, 한 링(Rung)에서 하나만 존재합니다. 여기서 링이란 LD의 처음부터 밑으로 내려가는 선이 없는 줄까지를 말합니다.
- ▷ LD의 각 요소를 연결하는 연결선에는 가로 연결선과 세로 연결선이 있습니다.

No.	기호	이름	설명
1		가로연결선	왼쪽의 값을 오른쪽으로 전달
2		세로연결선	왼쪽에 있는 가로 연결선들의 논리합

4.4. 접점

- ▷ 접점은 왼쪽에 있는 가로연결선의 상태와 현 접점과 연관된 BOOL 입력, 출력, 또는 메모리 변수 간의 논리곱(Boolean AND)을 한 값을 오른쪽에 위치한 가로 연결선에 전달합니다. 접점과 관련된 변수 값 자체는 변화시키지 않습니다. 표준 접점 기호는 다음 표와 같습니다.

정적 접점			
No	기호	이름	설명
1		평상시 열린 접점 (Normally Open Contact)	BOOL 변수("***" 로 표시된 것)의 상태가 On일 때에는 왼쪽의 연결선 상태는 오른쪽의 연결선으로 복사됩니다. 그렇지 않을 경우에는 오른쪽의 연결선 상태가 Off입니다.
2		평상시 닫힌 접점 (Normally Closed Contact)	BOOL 변수("***" 로 표시된 것)의 상태가 Off일 때에는 왼쪽의 연결선 상태는 오른쪽의 연결선으로 복사됩니다. 그렇지 않을 경우에는 오른쪽의 연결선 상태가 Off입니다.

상태 변환 검출 접점			
No	기호	이름	설명
3	*** — P —	양 변환 검출 접점 (Positive Transition-Sensing Contact)	BOOL 변수("***" 로 표시된 것)의 값이 전 스캔에서 Off 였던 것이 현재 스캔에서 On으로 되고, 왼쪽 연결선 상태가 On 되어 있는 경우에 한해서 오른쪽의 연결선 상태는 현재 스캔 동안에 On 이 됩니다.
4	*** — N —	음 변환 검출 접점 (Negative Transition-Sensing Contact)	BOOL 변수("***" 로 표시된 것)의 값이 전 스캔에서 On 이었던 것이 현재 스캔에서 Off 되고 왼쪽 연결선 상태가 On 되어 있는 경우에 한해서 오른쪽의 연결선 상태는 현재 스캔 동안에 On 이 됩니다.

4.5. 코일

- ▶ 코일은 왼쪽의 연결선의 상태 또는 상태 변환에 대한 처리 결과를 연관된 BOOL 변수에 저장시킵니다. 표준 코일 기호는 다음 표와 같습니다.
- ▶ 코일은 LD의 가장 오른쪽에만 올 수 있습니다. 즉 코일의 우측에는 언제나 오른쪽 모션만 있습니다.

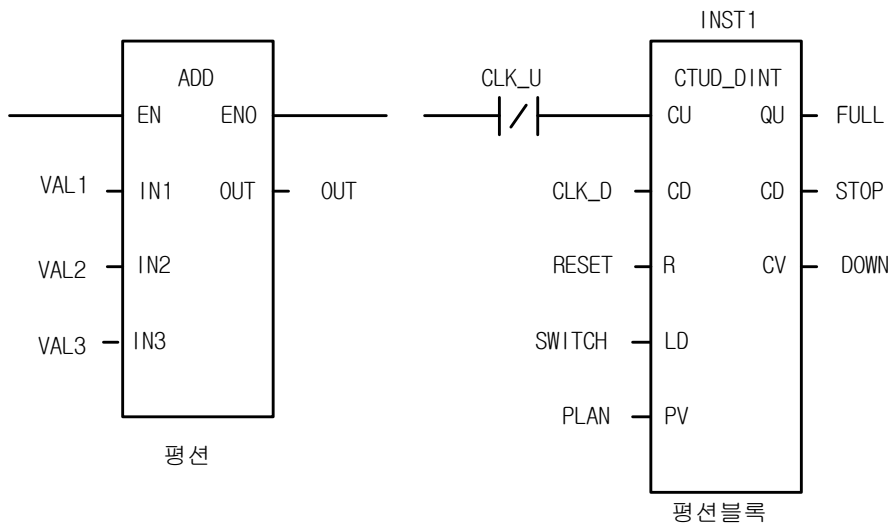
임시 코일(Momentary Coils)			
No.	기호	이름	설명
1	*** —()—	코일(Coil)	왼쪽에 있는 연결선의 상태를 관련된 BOOL 변수("***" 로 표시된 것)에 넣습니다.
2	*** —(/)—	역 코일(Negated Coil)	왼쪽에 있는 연결선 상태의 역(Negated)값을 관련된 BOOL 변수("***" 로 표시된 것)에 넣습니다. 즉, 왼쪽 연결선 상태 Off 이면 관련된 변수를 On 시키고, 왼쪽 연결선 상태가 On 이면 관련된 변수를 Off 시킵니다.
래치 코일(Latched Coils)			
No.	기호	이름	설명
3	*** —(S)—	Set(Latch) Coil	왼쪽의 연결선 상태가 On 이 되었을 때에는 관련된 BOOL 변수("***" 로 표시된 것)는 On 이 되고 Reset 코일에 의해 Off 되기 전까지는 On 되어 있는 상태로 유지됩니다.
4	*** —(R)—	Reset(Unlatch) Coil	왼쪽의 연결선 상태가 On 이 되었을 때에는 관련된 BOOL 변수("***" 로 표시된 것)는 Off 되고 Set 코일에 의해 On 되기 전까지는 Off 되어 있는 상태로 유지됩니다.

상태 변환 검출 코일(Transition-Sensing Coils)			
No.	기호	이름	설명
5	*** —(P)—	양 변환 검출 코일 (Positive Transition-Sensing Coil)	왼쪽 연결선 상태가 바로 전 스캔에서 Off 였던 것이 현재 스캔에서 On 이 되어 있는 경우에 관련된 BOOL 변수("***" 로 표시된 것)의 값은 현재 스캔 동안만 On 이 됩니다.
6	*** —(N)—	음 변환 검출 코일 (Negative Transition-Sensing Coil)	왼쪽 연결선 상태가 바로 전 스캔에서 On 이었던 것이 현재 스캔에서 Off 되어 있는 경우에 관련된 BOOL 변수("***" 로 표시된 것)의 값은 현재 스캔 동안만 On 이 됩니다.

4.6. 평선과 평선 블록의 호출

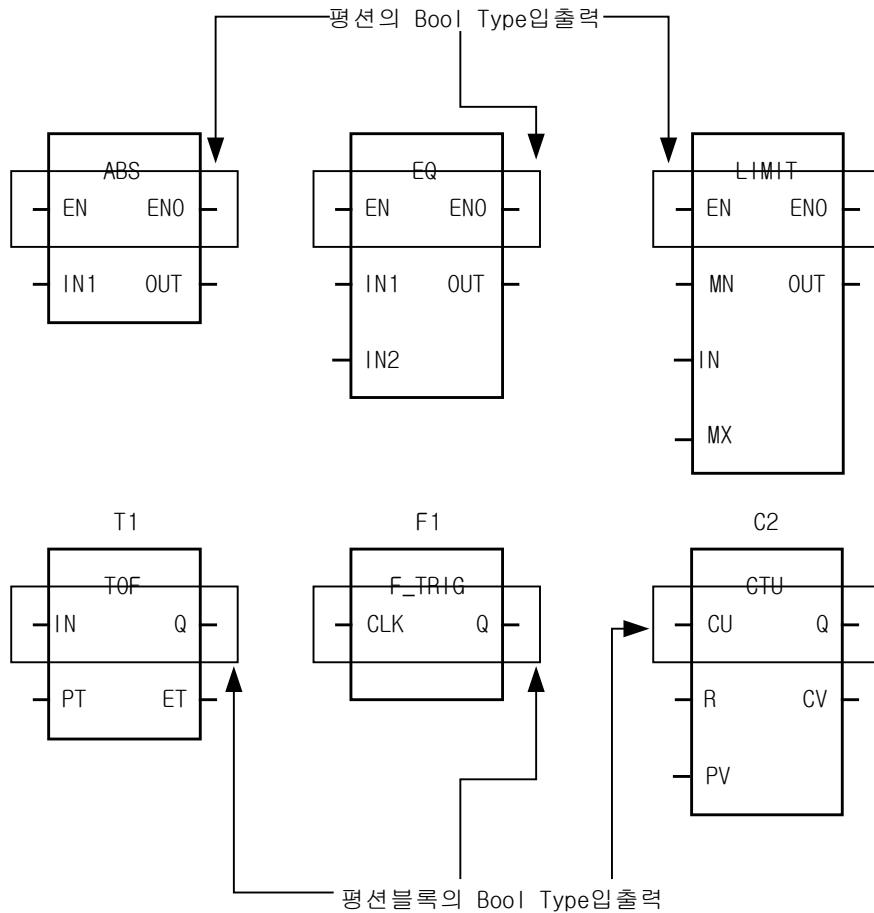
- ▶ 평선과 평선 블록에 대한 실제적인 입출력 연결은 입출력 표시가 있는 블록 외부에 적절한 데이터 또는 변수를 기입함으로써 이루어집니다.

예



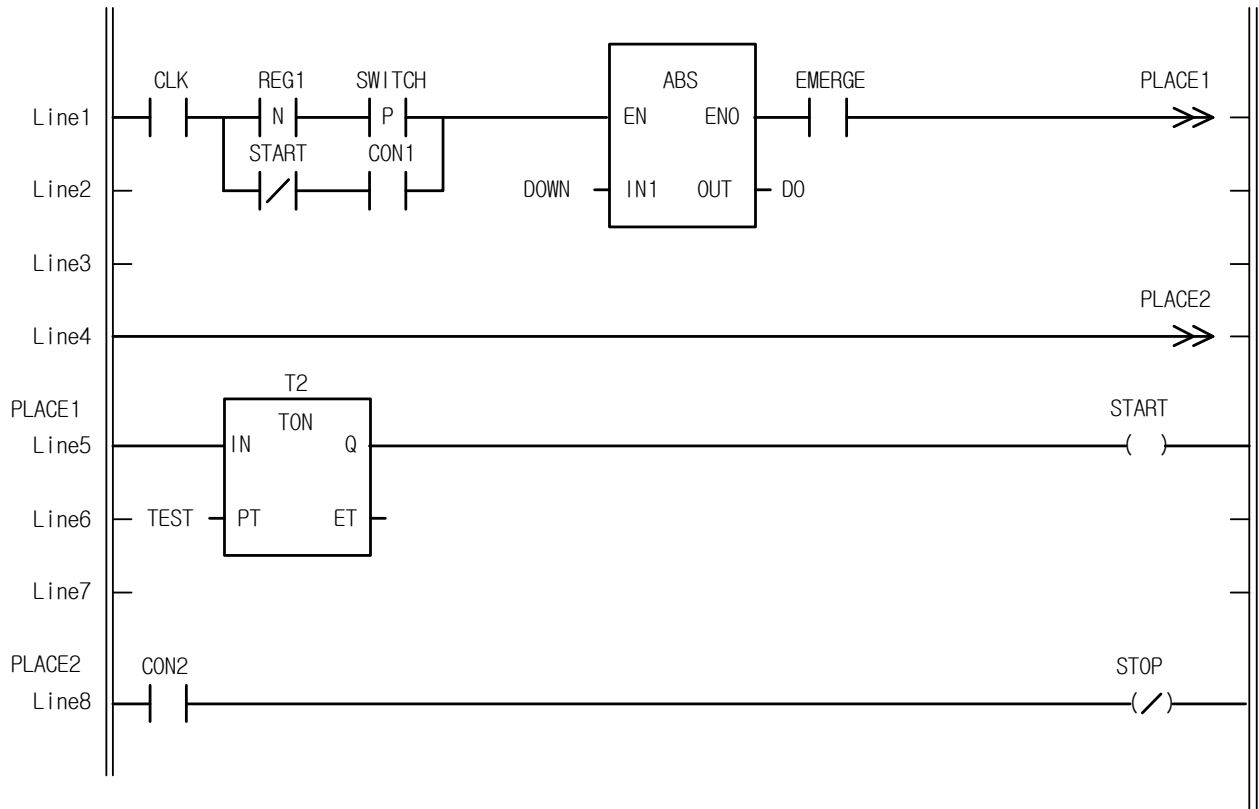
- ▶ 평선이나 평선 블록 내부로의 전원 흐름(Power Flow)을 허용하기 위해서는 적어도 한 개의 BOOL 타입 입력과 BOOL 타입 출력이 각 평선이나 평선 블록마다 존재해야 합니다. 평선에서는 EN 과 ENO 가 BOOL 타입 입출력이며, 평선 블록에서는 첫 번째 입력과 첫 번째 출력의 데이터 타입이 BOOL 입니다.

예



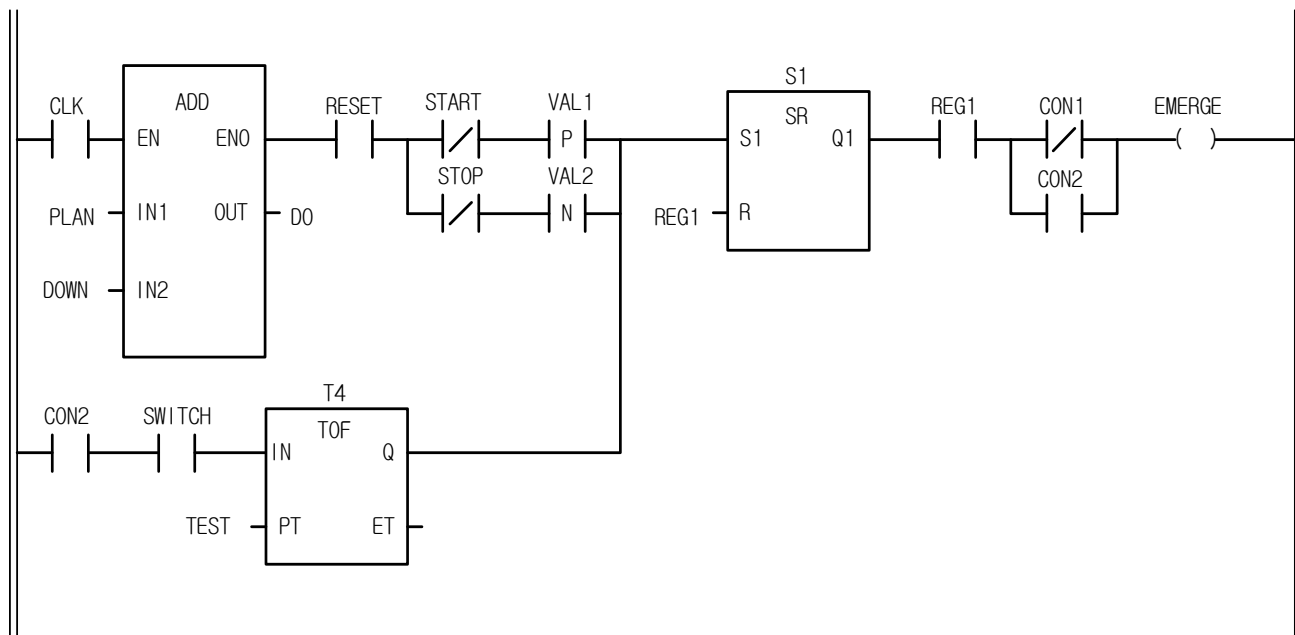
- ▶ LD 에서의 평선은 EN 이라는 입력과 ENO 라는 출력이 붙습니다. 두 개 다 BOOL 데이터 타입을 가지고 있으며, EN 입력 값이 BOOL 1 이면 그 평선을 수행하고, BOOL 0 이면 그 평선을 수행하지 않습니다. ENO 출력은 보통 EN 값이 그대로 나오지만 그 평선의 수행 시 에러가 발생하면 EN 값이 BOOL 1 이라도 ENO 값은 BOOL 0 이 나옵니다. 평선의 EN 은 언제나 전원 흐름선이 되어야 하지만 ENO 는 꼭 전원 흐름선이 될 필요는 없습니다. 하지만 ENO 가 아닌 평선 출력에 전원 흐름선을 연결할 때에는 그 출력의 데이터 타입이 반드시 BOOL 이어야 합니다. 또한 ENO 가 아닌 평선 출력에 전원 흐름선을 연결할 때에는 ENO 출력에는 아무것도 연결하면 안됩니다. 평선의 모든 입력은 평선의 왼쪽에 그 값을 기입함으로써 지정되는데 빠짐없이 지정하여야 합니다. 평선의 출력 값은 평선의 오른쪽에 지정한 변수에 보관됩니다.
- ▶ LD 평선 블록의 입력도 평선과 같은 방법으로 지정합니다. 평선 블록의 출력은 그 인스턴스 안에 저장되어 있으므로 변수를 지정하지 않더라도 상관없습니다. 평선 블록에는 EN, ENO 입출력이 없으므로 평선 블록을 만나면 항상 수행합니다. 따라서 어떤 로직 결과에 따라 평선 블록의 수행 여부를 결정하기 위해서는 점프(→>>)를 사용하여야 합니다. 평선 블록에 전원 흐름선을 연결할 때는 역시 데이터 타입이 BOOL 인 입출력에 연결하여야 합니다.

예



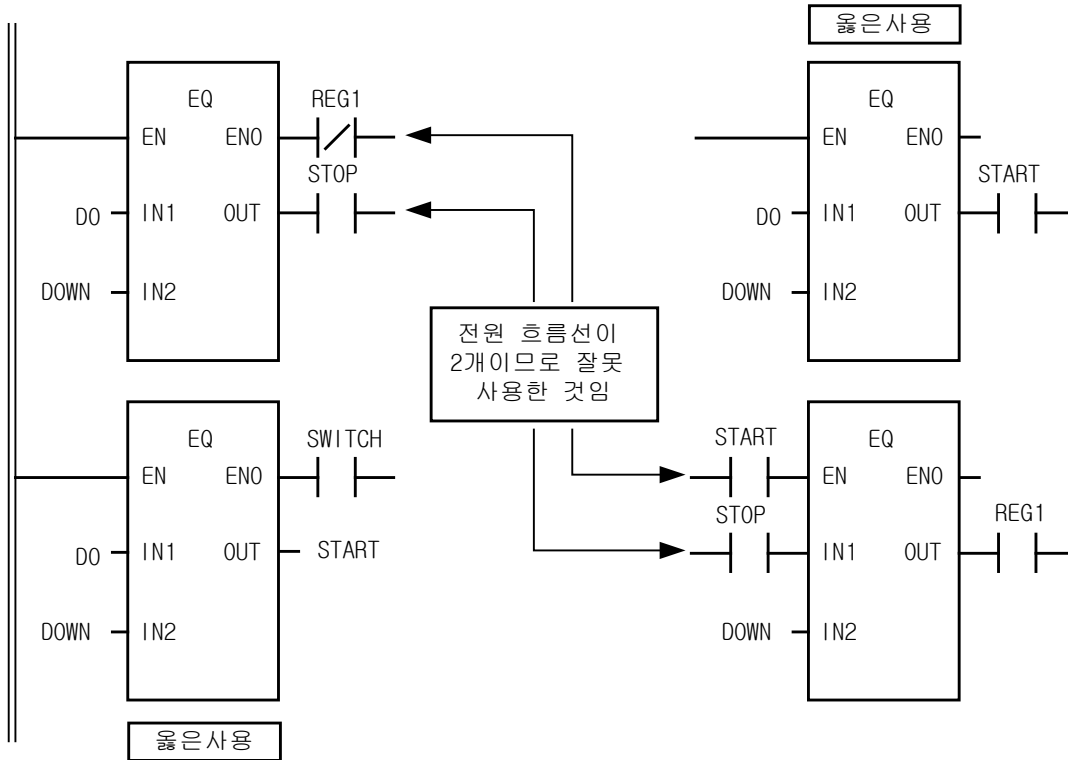
- ▶ LD 에서 평선과 평선 블록은 어느 곳이라도 올 수 있습니다. 평선과 평선 블록의 출력에 전원 흐름선을 연결하고 거기에 접점 등을 연결하여 로직 연산을 계속할 수도 있습니다.

예



▷ 하나의 평선이나 평선 블록에 연결할 수 있는 전원 흐름선은 단 하나입니다.

예



제5장 평선과 평선 블록

평선과 평선 블록에 대한 목록 요약입니다. 각각의 평선과 평선 블록에 대해서는 6장, 7장 기본/응용 평선과 8장, 9장 기본/응용 평선 블록을 참고 하시기 바라며, 모션 제어 평선 블록에 대한 상세 설명은 모션 제어 모듈 사용설명서 6장 명령어 편을 참고하시기 바랍니다.

5.1. 기본 평선

5.1.1. 타입 변환 평선

각각의 입력 데이터 타입을 출력 데이터 타입으로 변환합니다.

평선 그룹	평선 이름	입력 데이터 타입	출력 데이터 타입	비 고
REAL_TO_***	REAL_TO_SINT	REAL	SINT	-
	REAL_TO_INT	REAL	INT	-
	REAL_TO_DINT	REAL	DINT	-
	REAL_TO_LINT	REAL	LINT	-
	REAL_TO_USINT	REAL	USINT	-
	REAL_TO_UINT	REAL	UINT	-
	REAL_TO_UDINT	REAL	UDINT	-
	REAL_TO_ULINT	REAL	ULINT	-
	REAL_TO_DWORD	REAL	DWORD	-
	REAL_TO_LREAL	REAL	LREAL	-
LREAL_TO_***	LREAL_TO_SINT	LREAL	SINT	-
	LREAL_TO_INT	LREAL	INT	-
	LREAL_TO_DINT	LREAL	DINT	-
	LREAL_TO_LINT	LREAL	LINT	-
	LREAL_TO_USINT	LREAL	USINT	-
	LREAL_TO_UINT	LREAL	UINT	-
	LREAL_TO_UDINT	LREAL	UDINT	-
	LREAL_TO_ULINT	LREAL	ULINT	-
	LREAL_TO_LWORD	LREAL	LWORD	-
	LREAL_TO_REAL	LREAL	REAL	-
SINT_TO_***	SINT_TO_INT	SINT	INT	-
	SINT_TO_DINT	SINT	DINT	-
	SINT_TO_LINT	SINT	LINT	-
	SINT_TO_USINT	SINT	USINT	-
	SINT_TO_UINT	SINT	UINT	-
	SINT_TO_UDINT	SINT	UDINT	-
	SINT_TO_ULINT	SINT	ULINT	-
	SINT_TO_BOOL	SINT	BOOL	-
	SINT_TO_BYTE	SINT	BYTE	-
	SINT_TO_WORD	SINT	WORD	-
	SINT_TO_DWORD	SINT	DWORD	-
	SINT_TO_LWORD	SINT	LWORD	-
	SINT_TO_REAL	SINT	REAL	-
	SINT_TO_LREAL	SINT	LREAL	-

평선 그룹	평선 이름	입력 데이터 타입	출력 데이터 타입	비 고
INT_TO_***	INT_TO_SINT	INT	SINT	-
	INT_TO_DINT	INT	DINT	-
	INT_TO_LINT	INT	LINT	-
	INT_TO_USINT	INT	USINT	-
	INT_TO_UINT	INT	UINT	-
	INT_TO_UDINT	INT	UDINT	-
	INT_TO_ULINT	INT	ULINT	-
	INT_TO_BOOL	INT	BOOL	-
	INT_TO_BYTE	INT	BYTE	-
	INT_TO_WORD	INT	WORD	-
	INT_TO_DWORD	INT	DWORD	-
	INT_TO_LWORD	INT	LWORD	-
	INT_TO_REAL	INT	REAL	-
	INT_TO_LREAL	INT	LREAL	-
DINT_TO_***	DINT_TO_SINT	DINT	SINT	-
	DINT_TO_INT	DINT	INT	-
	DINT_TO_LINT	DINT	LINT	-
	DINT_TO_USINT	DINT	USINT	-
	DINT_TO_UINT	DINT	UINT	-
	DINT_TO_UDINT	DINT	UDINT	-
	DINT_TO_ULINT	DINT	ULINT	-
	DINT_TO_BOOL	DINT	BOOL	-
	DINT_TO_BYTE	DINT	BYTE	-
	DINT_TO_WORD	DINT	WORD	-
	DINT_TO_DWORD	DINT	DWORD	-
	DINT_TO_LWORD	DINT	LWORD	-
	DINT_TO_REAL	DINT	REAL	-
	DINT_TO_LREAL	DINT	LREAL	-
LINT_TO_***	LINT_TO_SINT	LINT	SINT	-
	LINT_TO_INT	LINT	INT	-
	LINT_TO_DINT	LINT	DINT	-
	LINT_TO_USINT	LINT	USINT	-
	LINT_TO_UINT	LINT	UINT	-
	LINT_TO_UDINT	LINT	UDINT	-
	LINT_TO_ULINT	LINT	ULINT	-
	LINT_TO_BOOL	LINT	BOOL	-
	LINT_TO_BYTE	LINT	BYTE	-
	LINT_TO_WORD	LINT	WORD	-
	LINT_TO_DWORD	LINT	DWORD	-
	LINT_TO_LWORD	LINT	LWORD	-
	LINT_TO_REAL	LINT	REAL	-
	LINT_TO_LREAL	LINT	LREAL	-
USINT_TO_***	USINT_TO_SINT	USINT	SINT	-
	USINT_TO_INT	USINT	INT	-
	USINT_TO_DINT	USINT	DINT	-
	USINT_TO_LINT	USINT	LINT	-

평선 그룹	평선 이름	입력 데이터 타입	출력 데이터 타입	비 고
	USINT_TO_UINT	USINT	UINT	-
	USINT_TO_UDINT	USINT	UDINT	-
	USINT_TO_ULINT	USINT	ULINT	-
	USINT_TO_BOOL	USINT	BOOL	-
	USINT_TO_BYTE	USINT	BYTE	-
	USINT_TO_WORD	USINT	WORD	-
	USINT_TO_DWORD	USINT	DWORD	-
	USINT_TO_LWORD	USINT	LWORD	-
	USINT_TO_REAL	USINT	REAL	-
	USINT_TO_LREAL	USINT	LREAL	-
UINT_TO_***	UINT_TO_SINT	UINT	SINT	-
	UINT_TO_INT	UINT	INT	-
	UINT_TO_DINT	UINT	DINT	-
	UINT_TO_LINT	UINT	LINT	-
	UINT_TO_USINT	UINT	USINT	-
	UINT_TO_UDINT	UINT	UDINT	-
	UINT_TO_ULINT	UINT	ULINT	-
	UINT_TO_BOOL	UINT	BOOL	-
	UINT_TO_BYTE	UINT	BYTE	-
	UINT_TO_WORD	UINT	WORD	-
	UINT_TO_DWORD	UINT	DWORD	-
	UINT_TO_LWORD	UINT	LWORD	-
	UINT_TO_REAL	UINT	REAL	-
	UINT_TO_LREAL	UINT	LREAL	-
	UINT_TO_DATE	UINT	DATE	-
UDINT_TO_***	UDINT_TO_SINT	UDINT	SINT	-
	UDINT_TO_INT	UDINT	INT	-
	UDINT_TO_DINT	UDINT	DINT	-
	UDINT_TO_LINT	UDINT	LINT	-
	UDINT_TO_USINT	UDINT	USINT	-
	UDINT_TO_UINT	UDINT	UINT	-
	UDINT_TO_ULINT	UDINT	ULINT	-
	UDINT_TO_BOOL	UDINT	BOOL	-
	UDINT_TO_BYTE	UDINT	BYTE	-
	UDINT_TO_WORD	UDINT	WORD	-
	UDINT_TO_DWORD	UDINT	DWORD	-
	UDINT_TO_LWORD	UDINT	LWORD	-
	UDINT_TO_REAL	UDINT	REAL	-
	UDINT_TO_LREAL	UDINT	LREAL	-
	UDINT_TO_TOD	UDINT	TOD	-
UDINT_TO_TIME	UDINT	TIME	-	
ULINT_TO_***	ULINT_TO_SINT	ULINT	SINT	-
	ULINT_TO_INT	ULINT	INT	-
	ULINT_TO_DINT	ULINT	DINT	-
	ULINT_TO_LINT	ULINT	LINT	-
	ULINT_TO_USINT	ULINT	USINT	-

평선 그룹	평선 이름	입력 데이터 타입	출력 데이터 타입	비 고
	UL INT_TO_UINT	UL INT	U INT	-
	UL INT_TO_UD INT	UL INT	UD INT	-
	UL INT_TO_BOOL	UL INT	BOOL	-
	UL INT_TO_BYTE	UL INT	BYTE	-
	UL INT_TO_WORD	UL INT	WORD	-
	UL INT_TO_DWORD	UL INT	DWORD	-
	UL INT_TO_LWORD	UL INT	LWORD	-
	UL INT_TO_REAL	UL INT	REAL	-
	UL INT_TO_LREAL	UL INT	LREAL	-
BOOL_TO_***	BOOL_TO_SINT	BOOL	S INT	-
	BOOL_TO_INT	BOOL	I NT	-
	BOOL_TO_DINT	BOOL	D INT	-
	BOOL_TO_LINT	BOOL	L INT	-
	BOOL_TO_USINT	BOOL	U SINT	-
	BOOL_TO_UINT	BOOL	U INT	-
	BOOL_TO_UDINT	BOOL	UD INT	-
	BOOL_TO_ULINT	BOOL	UL INT	-
	BOOL_TO_BYTE	BOOL	BYTE	-
	BOOL_TO_WORD	BOOL	WORD	-
	BOOL_TO_DWORD	BOOL	DWORD	-
	BOOL_TO_LWORD	BOOL	LWORD	-
BYTE_TO_***	BYTE_TO_SINT	BYTE	S INT	-
	BYTE_TO_INT	BYTE	I NT	-
	BYTE_TO_DINT	BYTE	D INT	-
	BYTE_TO_LINT	BYTE	L INT	-
	BYTE_TO_USINT	BYTE	U SINT	-
	BYTE_TO_UINT	BYTE	U INT	-
	BYTE_TO_UDINT	BYTE	UD INT	-
	BYTE_TO_ULINT	BYTE	UL INT	-
	BYTE_TO_BOOL	BYTE	BOOL	-
	BYTE_TO_WORD	BYTE	WORD	-
	BYTE_TO_DWORD	BYTE	DWORD	-
	BYTE_TO_LWORD	BYTE	LWORD	-
WORD_TO_***	WORD_TO_SINT	WORD	S INT	-
	WORD_TO_INT	WORD	I NT	-
	WORD_TO_DINT	WORD	D INT	-
	WORD_TO_LINT	WORD	L INT	-
	WORD_TO_USINT	WORD	U SINT	-
	WORD_TO_UINT	WORD	U INT	-
	WORD_TO_UDINT	WORD	UD INT	-
	WORD_TO_ULINT	WORD	UL INT	-
	WORD_TO_BOOL	WORD	BOOL	-
	WORD_TO_BYTE	WORD	BYTE	-
	WORD_TO_DWORD	WORD	DWORD	-
	WORD_TO_LWORD	WORD	LWORD	-
	WORD_TO_DATE	WORD	DATE	-

평선 그룹	평선 이름	입력 데이터 타입	출력 데이터 타입	비 고
DWORD_TO_***	DWORD_TO_SINT	DWORD	SINT	-
	DWORD_TO_INT	DWORD	INT	-
	DWORD_TO_DINT	DWORD	DINT	-
	DWORD_TO_LINT	DWORD	LINT	-
	DWORD_TO_USINT	DWORD	USINT	-
	DWORD_TO_UINT	DWORD	UINT	-
	DWORD_TO_UDINT	DWORD	UDINT	-
	DWORD_TO_ULINT	DWORD	ULINT	-
	DWORD_TO_BOOL	DWORD	BOOL	-
	DWORD_TO_BYTE	DWORD	BYTE	-
	DWORD_TO_WORD	DWORD	WORD	-
	DWORD_TO_LWORD	DWORD	LWORD	-
	DWORD_TO_REAL	DWORD	REAL	-
	DWORD_TO_TIME	DWORD	TIME	-
	DWORD_TO_TOD	DWORD	TOD	-
LWORD_TO_***	LWORD_TO_SINT	LWORD	SINT	-
	LWORD_TO_INT	LWORD	INT	-
	LWORD_TO_DINT	LWORD	DINT	-
	LWORD_TO_LINT	LWORD	LINT	-
	LWORD_TO_USINT	LWORD	USINT	-
	LWORD_TO_UINT	LWORD	UINT	-
	LWORD_TO_UDINT	LWORD	UDINT	-
	LWORD_TO_ULINT	LWORD	ULINT	-
	LWORD_TO_BOOL	LWORD	BOOL	-
	LWORD_TO_BYTE	LWORD	BYTE	-
	LWORD_TO_WORD	LWORD	WORD	-
	LWORD_TO_DWORD	LWORD	DWORD	-
	LWORD_TO_LREAL	LWORD	LREAL	-
	LWORD_TO_DT	LWORD	DT	-
TIME_TO_***	TIME_TO_UDINT	TIME	UDINT	-
	TIME_TO_DWORD	TIME	DWORD	-
DATE_TO_***	DATE_TO_UINT	DATE	UINT	-
	DATE_TO_WORD	DATE	WORD	-
TOD_TO_***	TOD_TO_UDINT	TOD	UDINT	-
	TOD_TO_DWORD	TOD	DWORD	-
DT_TO_***	DT_TO_LWORD	DT	LWORD	-
	DT_TO_DATE	DT	DATE	-
	DT_TO_TOD	DT	TOD	-

5.1.2. 수치 연산 평선

1) 하나의 입력을 갖는 수치 연산 평선

No.	평선 이름	기 능	비 고
일반 평선			
1	ABS	절대값 연산(Absolute Value)	-
2	SQRT	제곱근 연산(Square Root)	-
로그 평선			
3	LN	자연 대수 연산(Natural Logarithm)	-
4	LOG	상용 대수 연산(Common Logarithm Base To 10)	-
5	EXP	자연 지수 연산(Natural Exponential)	-
삼각 평선			
6	SIN	사인 값 연산(Sine)	-
7	COS	코사인 값 연산(Cosine)	-
8	TAN	탄젠트 값 연산(Tangent)	-
9	ASIN	아크 사인 값 연산(Arc Sine)	-
10	ACOS	아크 코사인 값 연산(Arc Cosine)	-
11	ATAN	아크 탄젠트 값 연산(Arc Tangent)	-

2) 기본 수치 연산 평선

No.	평선 이름	기 능	비 고
입력 개수를 확장할 수 있는 연산 평선(단, n은 8까지 가능)			
1	ADD	더하기 ($OUT \leq IN1 + IN2 + \dots + INn$)	-
2	MUL	곱하기 ($OUT \leq IN1 * IN2 * \dots * INn$)	-
입력 개수가 일정한 연산 평선			
3	SUB	빼기($OUT \leq IN1 - IN2$)	-
4	DIV	나누기($OUT \leq IN1 / IN2$)	-
5	MOD	나머지 구하기($OUT \leq IN1 \text{ Modulo } IN2$)	-
6	EXPT	지수 연산($OUT \leq IN1^{IN2}$)	-
7	MOVE	데이터 복사($OUT \leq IN$)	-

5.1.3. 각도 변환 평선

No.	평선 이름	기 능	비 고
1	DEG	각도 변환	-
2	RAD	Radian 값을 각도로 변환	-

5.1.4. 비트열 평선

1) 비트 이동 평선

No.	평선 이름	기 능	비 고
1	SHL	입력을 N 비트 왼쪽으로 이동(오른쪽은 0으로 채움)	-
2	SHR	입력을 N 비트 오른쪽으로 이동(왼쪽은 0으로 채움)	-
3	ROL	입력을 N 비트 왼쪽으로 회전	-
4	ROR	입력을 N 비트 오른쪽으로 회전	-

2) 비트 연산 평선

No.	평선 이름	기 능 (단, n은 8까지 가능)	비 고
1	AND	논리곱($OUT \leftarrow IN1 \text{ AND } IN2 \text{ AND } \dots \text{ AND } INn$)	-
2	OR	논리합($OUT \leftarrow IN1 \text{ OR } IN2 \text{ OR } \dots \text{ OR } INn$)	-
3	XOR	배타적 논리합($OUT \leftarrow IN1 \text{ XOR } IN2 \text{ XOR } \dots \text{ XOR } INn$)	-
4	NOT	논리 반전($OUT \leftarrow \text{NOT } IN1$)	-
5	XNR	배타적 논리곱($OUT \leftarrow IN1 \text{ XNR } IN2 \text{ XNR } \dots \text{ XNR } INn$)	-

5.1.5. 선택 평선

No.	평선 이름	기 능 (단, n은 8까지 가능)	비 고
1	MAX	입력 $IN1, \dots, INn$ 중에 최대값 출력	-
2	MIN	입력 $IN1, \dots, INn$ 중에 최소값 출력	-
3	LIMIT	상, 하한 제한 값 출력	-
4	MUX	입력 $IN0, \dots, INn$ 중 K 번째 입력을 출력	-

5.1.6. 데이터 교환 평선

No.	평선 이름	기 능	비 고
1	SWAP	데이터의 상위 하위 바꾸기	-
2	XCHG	2 개의 입력변수 값을 서로 교환	-

5.1.7. 비교 평선

No.	평선 이름	기 능 (단, n은 8까지 가능)	비 고
1	GT	'크다' 비교 $OUT \leftarrow (IN1 > IN2) \& (IN2 > IN3) \& \dots \& (INn-1 > INn)$	-
2	GE	'크거나 작다' 비교 $OUT \leftarrow (IN1 \geq IN2) \& (IN2 \geq IN3) \& \dots \& (INn-1 \geq INn)$	-
3	EQ	'같다' 비교 $OUT \leftarrow (IN1 = IN2) \& (IN2 = IN3) \& \dots \& (INn-1 = INn)$	-

No.	평선 이름	기 능 (단, n은 8까지 가능)	비 고
4	LE	'작거나 같다' 비교 $OUT \leq (IN1 \leq IN2) \& (IN2 \leq IN3) \& \dots \& (IN_{n-1} \leq IN_n)$	-
5	LT	'작다' 비교 $OUT \leq (IN1 < IN2) \& (IN2 < IN3) \& \dots \& (IN_{n-1} < IN_n)$	-
6	NE	'같지 않다' 비교 $OUT \leq (IN1 > IN2) \& (IN2 > IN3) \& \dots \& (IN_{n-1} > IN_n)$	-
7	DZA	불감대 A $Out = Input (Input \geq Zone)$, $Out = 0 (Input < Zone)$	-
8	DZB	불감대 B $Out = Input - Zone (Input \geq Zone , Input < 0)$ $Out = Input + Zone (Input \geq Zone , Input \geq 0)$ $Out = 0 (Input < Zone)$	-

5.1.8. 날짜 시각 평선

No.	평선 이름	기 능	비 고
1	ADD_TIME	시간, 시각 및 날짜 시각에 시간 더하기	-
2	SUB_TIME	시간, 시각 및 날짜 시각에 시간 빼기	-
	SUB_DATE	날짜에서 날짜를 빼서 시간 산출하기	-
	SUB_TOD	시각에서 시각을 빼서 시간 산출하기	-
	SUB_DT	날짜 시각에서 날짜 시각을 빼서 시간 산출하기	-
	CONCAT_TIME	날짜와 시각을 붙여서 날짜 시각 만들기	-

5.1.9. 데이터 처리 평선

No.	평선 이름	기 능	비 고
1	BIT_BYTE	8개의 Bit들을 BYTE로 모음	-
2	BYTE_BIT	BYTE를 8개의 Bit로 나눔	-
3	BYTE_WORD	2개의 BYTE들을 WORD로 모음	-
4	WORD_BYTE	WORD를 2개의 BYTE로 나눔	-
5	WORD_DWORD	2개의 WORD들을 DWORD로 모음	-
6	DWORD_WORD	DWORD를 2개의 WORD로 나눔	-
7	DWORD_LWORD	2개의 DWORD들을 LWORD로 모음	-
8	LWORD_DWORD	LWORD를 2개의 DWORD로 나눔	-
9	DEC	데이터를 하나 감소	-

10	DECO	지정된 비트 위치를 0n	-
11	ENCO	0n된 비트 위치를 숫자로 출력	-
12	INC	IN 데이터를 하나 증가	-
13	MEMCPY	메모리 복사(OUT1, OUT2... <= IN1, IN2...)	-
14	MEMSET	메모리 설정(OUT1 = OUT2 = ... <= IN1)	-

5.2. 기본 평선 블록

5.2.1. 바이스테이블 평선 블록

No.	평선 블록 이름	기 능	비 고
1	SR	세트 우선 쌍안정 출력	-
2	RS	리셋 우선 쌍안정 출력	-

5.2.2. 에지 검출 평선 블록

No.	평선 블록 이름	기 능	비 고
1	R_TRIG	상승 에지 검출(Rising Edge Detector)	-
2	F_TRIG	하강 에지 검출(Falling Edge Detector)	-
3	FF	입력조건 상승 시 출력 반전	-

5.2.3. 카운터

No.	평선 블록 이름	기 능	비 고
1	CTU_***	가산 카운터(Up Counter) INT, DINT, LINT, UINT, UDINT, ULINT	-
2	CTD_***	감산 카운터(Down Counter) INT, DINT, LINT, UINT, UDINT, ULINT	-
3	CTUD_***	가감산 카운터(Up Down Counter) INT, DINT, LINT, UINT, UDINT, ULINT	-
4	CTR	링 카운터(Ring Counter)	-

5.2.4. 타이머

No.	평선 블록 이름	기 능	비 고
1	TP	펄스 타이머(Pulse Timer)	-
2	TON	On 딜레이 타이머(On-Delay Timer)	-

No.	평선 블록 이름	기 능	비 고
3	TOF	Off 딜레이 타이머(Off-Delay Timer)	-
4	TMR	적산 타이머(Integrating Timer)	-
5	TP_RST	펄스 타이머의 출력 Off가 가능한 노스테인블 타이머	-
6	TRTG	리트리거블 타이머(Retriggerable Timer)	-
7	TOF_RST	동작 중 출력 Off가 가능한 Off 딜레이 타이머(Off-Delay Timer)	-
8	TON_UINT	정수 설정 On 딜레이 타이머(On-Delay Timer)	-
9	TOF_UINT	정수 설정 Off 딜레이 타이머(Off-Delay Timer)	-
10	TP_UINT	정수 설정 펄스 타이머(Pulse Timer)	-
11	TMR_UINT	정수 설정 적산 타이머(Integrating Timer)	-
12	TMR_FLK	점멸 기능 타이머	-
13	TRTG_UINT	정수 설정 리트리거블 타이머	-

5.3. 확장 평선

No.	평선 이름	기 능	비 고
1	FOR	FOR ~ NEXT 구간을 n 번 실행	-
2	NEXT		-
3	BREAK	FOR ~ NEXT 구간을 빠져 나옴	-
4	CALL	SBRT 루틴 호출	-
5	SBRT	CALL 에 의해 호출될 루틴 지정	-
6	RET	RETURN	-
7	JMP	LABLE 위치로 점프	-
8	INIT_DONE	초기화 태스크 종료	-
9	END	프로그램의 종료	-

5.4. 모션 제어 평선 블록

No.	이름	상세설명	동작조건
단축 모션 명령어			
1	MC_Power	서보 온/오프	Level
2	MC_Home	원점복귀	Edge
3	MC_Stop	즉시정지	Edge
4	MC_Halt	정지	Edge
5	MC_MoveAbsolute	절대위치 운전	Edge
6	MC_MoveRelative	상대위치 운전	Edge

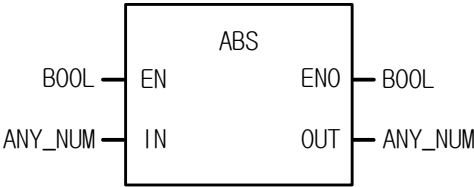
No.	이름	상세설명	동작조건
7	MC_MoveAdditive	추가위치 운전	Edge
8	MC_MoveVelocity	지정속도 운전	Edge
9	MC_MoveContinuousAbsolute	절대위치 운전 후 지정속도 운전	Edge
10	MC_MoveContinuousRelative	상대위치 운전 후 지정속도 운전	Edge
11	MC_TorqueControl	토크제어 운전	Edge
12	MC_SetPosition	현재위치 설정	Edge
13	MC_SetOverride	속도/가속도 오버라이드	Level
14	MC_ReadParameter	파라미터 읽기	Level
15	MC_WriteParameter	파라미터 쓰기	Edge
16	MC_Reset	축 에러 리셋	Edge
17	MC_TouchProbe	터치 프로브	Edge
18	MC_AbortTrigger	트리거 해제	Edge
19	MC_MoveSuperImposed	SuperImposed 운전	Edge
20	MC_HaltSuperImposed	SuperImposed 운전 정지	Edge
다축 모션 명령어			
21	MC_CamIn	캠 운전	Edge
22	MC_CamOut	캠 운전 중지	Edge
23	MC_GearIn	전자기어 운전	Edge
24	MC_GearOut	전자기어 해제	Edge
25	MC_GearInPos	위치지정 전자기어 운전	Edge
26	MC_Phasing	위상 보정	Edge
27	MC_GearInEx	확장 기어 운전	Edge
28	MC_GearInPosEx	확장 위치지정 기어운전	Edge
그룹 모션 명령어			
29	MC_AddAxisToGroup	그룹 축 추가	Edge
30	MC_RemoveAxisFromGroup	그룹 축 제외	Edge
31	MC_UngroupAllAxes	그룹 내 모든 축 제외	Edge
32	MC_GroupEnable	그룹 활성화	Edge
33	MC_GroupDisable	그룹 비활성화	Edge
34	MC_GroupHome	그룹 원점복귀 운전	Edge
35	MC_GroupSetPosition	그룹 현재위치 설정	Edge
36	MC_GroupStop	그룹 즉시정지	Edge
37	MC_GroupHalt	그룹 정지	Edge
38	MC_GroupReset	그룹 에러 리셋	Edge
39	MC_MoveLinearAbsolute	절대위치 직선보간 운전	Edge
40	MC_MoveLinearRelative	상대위치 직선보간 운전	Edge

No.	이름	상세설명	동작조건
41	MC_MoveCircularAbsolute	절대위치 원호보간 운전	Edge
42	MC_MoveCircularRelative	상대위치 원호보간 운전	Edge
LS 명령어			
43	LS_Connect	통신연결	Edge
44	LS_Disconnect	통신연결 해제	Edge
45	LS_ReadServoParameter	서보파라미터 읽기	Edge
46	LS_WriteServoParameter	서보파라미터 쓰기	Edge
47	LS_SaveServoParameter	서보 파라미터 저장	Edge
48	LS_EncoderPreset	엔코더 현재위치 설정	Edge
49	LS_Jog	조그운전	Level
50	LS_ReadCamData	캠 데이터 읽기	Edge
51	LS_ReadCamData2	캠 데이터 읽기 2	Edge
52	LS_WriteCamData	캠 데이터 쓰기	Edge
53	LS_ReadEsc	ESC 읽기	Edge
54	LS_WriteEsc	ESC 쓰기	Edge
55	LS_CamSkip	캠 스킵	Edge
56	LS_VarCamIn	변수 캠 운전	Edge
57	LS_VarGearIn	변수 기어 운전	Edge
58	LS_VarGearInPos	변수 위치지정 기어 운전	Edge
59	LS_ReadCamTableSlavePos	캠테이블 슬레이브 위치 읽기	Edge
60	LS_SyncMoveVelocity	속도제어 운전 (csv 모드)	Edge
61	LS_ReadCamTableMasterPos	캠 테이블 마스터 위치 읽기	Edge
62	LS_OnOffCam	온오프캠 운전	Edge
63	LS_RotaryKnifeCamGen	로터리커터 캠 프로파일 생성	Edge
64	LS_CrossSealCamGen	크로스실러 캠 프로파일 생성	Edge
65	LS_OnOffCamEx	확장 온오프캠 운전	Edge
66	LS_VarGearInEx	확장 변수 기어 운전	Edge
67	LS_VarGearInPosEx	확장 변수 위치지정 기어운전	Edge
좌표계 전용 명령어			
68	MC_SetKinTransform	기구정보 설정	Edge
69	MC_SetCartesianTransform	PCS 설정	Edge
70	LS_SetWorkSpaceTransform	워크 스페이스 설정	Edge
71	LS_MoveLinearAbsolute	좌표계 절대위치 직선보간운전	Edge
72	LS_MoveLinearRelative	좌표계 상대위치 직선보간운전	Edge
73	LS_MoveLinearTimeAbsolute	좌표계 절대위치 시간직선보간운전	Edge
74	LS_MoveLinearTimeRelative	좌표계 상대위치 시간직선보간운전	Edge

No.	이름	상세설명	동작조건
75	MC_MoveCircularAbsolute2D	좌표계 절대위치 원호보간 운전	Edge
76	MC_MoveCircularRelative2D	좌표계 상대위치 원호보간 운전	Edge
77	MC_TrackConveyorBelt	컨베이어벨트 동기 설정	Edge
78	MC_TrackRotaryTable	로터리 테이블 동기 설정	Edge
79	LS_RobotJog	좌표계 조그 운전	Level
상태 읽기 명령어			
80	MC_ReadActualPosition	현재 위치 읽기	Level
81	MC_ReadActualTorque	현재 토크 읽기	Level
82	MC_ReadActualVelocity	현재 속도 읽기	Level
83	MC_ReadCommandedPosition	명령 위치 읽기	Level
84	MC_ReadCommandedTorque	명령 토크 읽기	Level
85	MC_ReadCommandedVelocity	명령 속도 읽기	Level
86	MC_ReadMotionInfo	모션 정보 읽기	Level
기타 명령어			
87	PID	PID 연산	Level
88	LINAC	가감속기 명령	Level
89	SLINAC	S-커브 가감속기 명령	Level

제6장 기본 평선

1. 각각의 기본 평선에 대한 설명입니다.

ABS		적용 기종	발생플래그																	
절대값 연산		XGF-M32E	_ERR, _LER																	
평 선		설 명																		
		<p>입력 EN : 1 일 때 평선 실행 IN : 절대값 연산의 입력 값</p> <p>출력 ENO : 1 을 출력 OUT : 절대값</p> <p>IN, OUT 은 같은 데이터 타입이어야 함.</p>																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	IN						○	○	○	○	○	○	○	○	○	○				
	OUT						○	○	○	○	○	○	○	○	○	○				

■ 기능

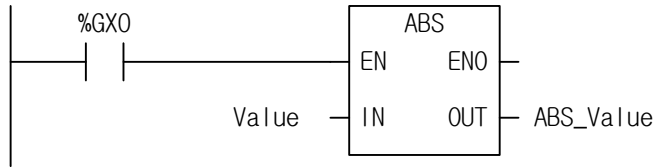
1. IN의 절대값을 OUT으로 출력시킵니다.
OUT = |IN|
2. X의 절대값 |X|는
 - A. X ≥ 0 이면 |X| = X 이고,
 - B. X < 0 이면 |X| = -X 입니다.

■ 플래그

플래그	설명
_ERR	IN의 값이 (-)최소값일 때 _ERR, _LER 플래그가 셋(Set)됩니다. 예) 데이터 타입이 SINT 일 때 IN의 값이 -128 이면 에러

■ 프로그램 예

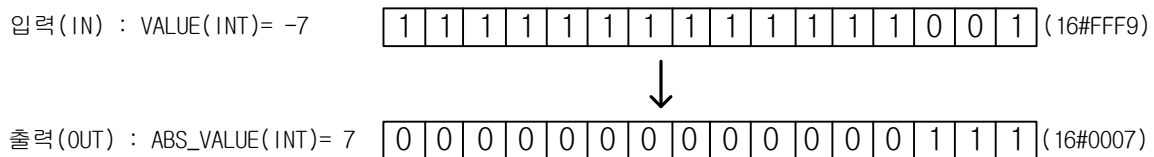
1. LD

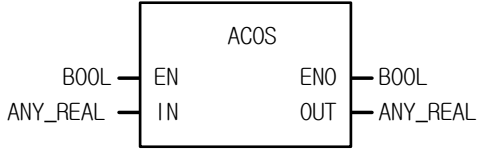


2. ST

```
ABS_Value := ABS(EN:=%GX0, IN:=Value);
```

- (1) 실행조건(%GX0)이 On 하면 평선 ABS가 실행됩니다.
- (2) VALUE = -7 이면, ABS_VALUE = $|-7| = 7$ 이 됩니다.
VALUE = 200 이면, ABS_VALUE = $|200| = 200$ 이 됩니다.
- (3) INT 형의 음수 표시는 2의 Complement 표현 (3.2.4. 데이터 타입별 구조 참조)



ACOS		적용 기종	발생플래그																																																												
Arc Cosine 연산		XGF-M32E	_ERR, _LER																																																												
평 선		설 명																																																													
		<p>입력 EN : 1일 때 평선 실행 IN : Arc Cosine 연산의 입력 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : Arc Cosine 연산의 각도 결과 값(Radian)</p> <p>IN, OUT은 같은 데이터 타입이어야 함</p>																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN														○	○					OUT														○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN														○	○																																																
OUT														○	○																																																

■ 기능

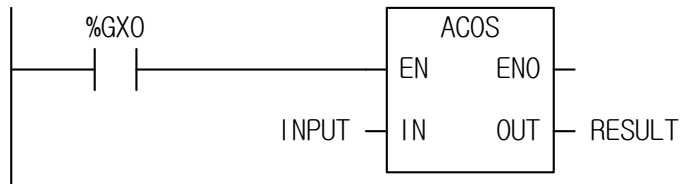
1. IN의 Arc Cosine 값을 구해 OUT으로 출력시킵니다. 출력 값은 0에서 π 사이의 값이 됩니다.
OUT = ACOS (IN)

■ 플래그

플래그	설명
_ERR	입력 값이 범위가 -1.0과 1.0 사이에 있지 않을 경우 _ERR, _LER 플래그가 셋(SET)됩니다.

■ 프로그램 예

1. LD



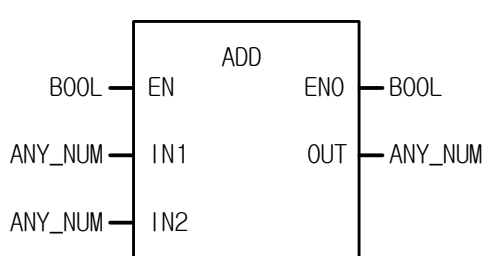
2. ST

```
RESULT := ACOS(EN:=%GX0, IN:=INPUT);
```

(1) 실행조건(%IX0.1.3)이 On 하면 평선 Arc Cosine 연산 평선 ACOS 가 실행됩니다.

(2) INPUT 으로 선언된 입력 변수의 값이 0.8660... ($\sqrt{3}/2$) 일 때 출력 변수로 선언된 RESULT 은 0.5235...

($\pi/6$ rad = 30°) 입니다.

ADD		적용 기종	발생플래그																																																																																
더하기		XGF-M32E	_ERR, _LER																																																																																
평 선		설 명																																																																																	
		<p>입력 EN : 1일 때 평선 실행 IN1 : 더할 값 IN2 : 더할 값 입력은 8 개까지 확장 가능</p> <p>출력 ENO : 에러 없이 실행되면 1 을 출력 OUT : 더한 결과 값</p> <p>IN1, IN2, ..., OUT 에 연결되는 변수는 모두 같은 데이터 타입이어야 함.</p>																																																																																	
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>LINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>IN2</td> <td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>OUT</td> <td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	IN1						○	○	○	○	○	○	○	○	○	○					IN2						○	○	○	○	○	○	○	○	○	○					OUT						○	○	○	○	○	○	○	○	○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT																																																																
IN1						○	○	○	○	○	○	○	○	○	○																																																																				
IN2						○	○	○	○	○	○	○	○	○	○																																																																				
OUT						○	○	○	○	○	○	○	○	○	○																																																																				

■ 기능

1. IN1, IN2, ..., INn (n은 입력 개수, 최대 8 개)를 더해서 OUT으로 출력시킵니다.
 $OUT = IN1 + IN2 + \dots + INn$

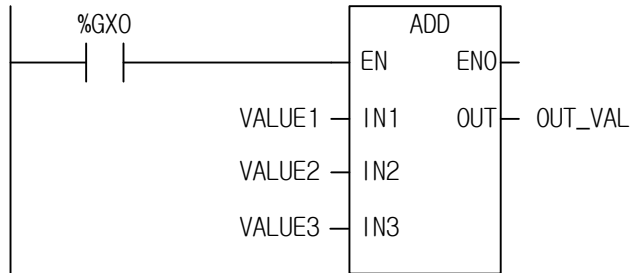
■ 플래그

플래그	설명
_ERR	출력 값이 해당 데이터 타입의 범위를 벗어날 경우 _ERR, _LER 플래그가 셋(Set)됩니다

☆ REAL, LREAL 타입 연산에서는 IN1 에서 IN8 로 순차적으로 연산을 하기 때문에 중간 연산과정에서 이미 REAL, LREAL 에 최대값이나 최소값을 넘게 되면 _ERR, _LER 플래그가 셋(Set)되고 결과값은 무한대 값 또는 비정상적인 값 (1.#INF000000000000e+000, 1.#SNAN000000000000e+000, 1.#QNAN000000000000e+000)으로 표시됩니다.

■ 프로그램 예

1. LD

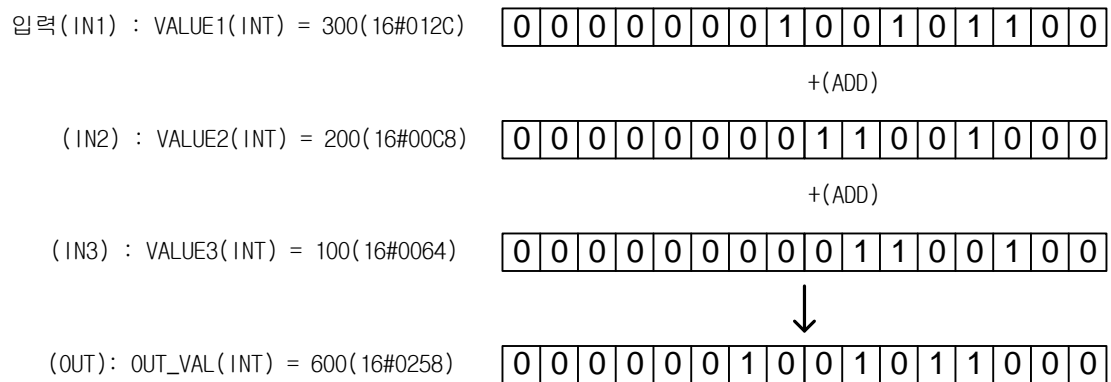


2. ST

```
OUT_VAL := ADD(EN:=%GX0, IN1:= VALUE1, IN2:= VALUE2, IN3:= VALUE3);
```

(1) 실행조건(%GX0)이 On 하면 더하기 평선 ADD 가 실행됩니다.

(2) 입력변수 값이 VALUE1 = 300, VALUE2 = 200, VALUE3 = 100 이면, 출력변수로 설정한 OUT_VAL = 300 + 200 + 100 = 600 이 됩니다.



ADD_TIME		적용 기종	발생플래그																																																												
시간 더하기		XGF-M32E	_ERR, _LER																																																												
평 선		설 명																																																													
		<p>입력 EN : 1일 때 평선 실행 IN1 : 기준시각 또는 시간 IN2 : 더할 시간</p> <p>출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 더한 결과 시각 또는 시간</p> <p>OUT의 타입은 입력 IN1의 타입을 따름. 즉 IN1의 타입이 TIME_OF_DAY 이면, 출력 OUT의 타입도 TIME_OF_DAY 임.</p>																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td>○</td> <td>○</td> </tr> <tr> <td>OUT</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td>○</td> <td>○</td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN1																○		○	○	OUT																○		○	○		
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN1																○		○	○																																												
OUT																○		○	○																																												

■ 기능

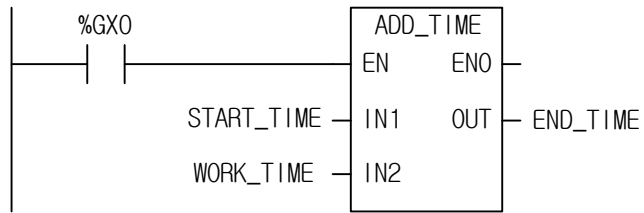
1. IN1이 TIME일 경우에는 시간과 시간을 더해서 합한 시간을 출력시킵니다.
2. IN1이 TIME_OF_DAY일 경우에는 기준시각에 시간을 더해서 하루 중의 시각을 출력시킵니다.
3. IN1이 DATE_AND_TIME일 경우에는 기준이 되는 날짜와 시각에 시간을 더해서 날짜와 시각을 출력시킵니다.

■ 플래그

플래그	설 명
_ERR	출력 값이 해당 데이터 타입의 범위를 벗어날 경우, _ERR, _LER 플래그가 셋(Set)됩니다. 시간과 시간을 더한 결과가 TIME 데이터 타입의 범위 T#49D17H2M47S295MS를 넘거나 시각(TOD)과 시간을 더한 결과가 24시를 넘을 경우, 또는 날짜와 시각(DT)과 시간을 더한 결과가 2163년을 넘을 경우 에러가 됩니다.

■ 프로그램 예

1. LD



2. ST

```
END_TIME := ADD_TIME(EN:= %GX0, IN1:= START_TIME, IN2:= WORK_TIME);
```

- (1) 실행조건(%GX0)이 On 하면 시간 더하기 평선 ADD_TIME 이 실행됩니다.
 (2) 작업을 시작한 START_TIME 이 TOD#08:30:00 이고 작업한 시간 WORK_TIME 이 T#2H10M20S500MS 이면 작업이 종료된 시각 END_TIME 에는 TOD#10:40:20.5 가 출력됩니다.

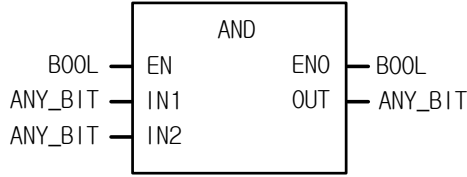
입력 (IN1) : START_TIME(TOD) = TOD#08:30:00

+ (ADD_TIME)

(IN2) : WORK_TIME(TIME) = T#2H10M20S500MS



출력 (OUT) : END_TIME(TOD) = TOD#10:40:20.5

AND		적용 기종	발생플래그																	
논리곱		XGF-M32E	-																	
평 선		설 명																		
		<p>입력 N : 1일 때 평선 실행 IN1 : AND 될 값 IN2 : AND 될 값 입력이 8 개까지 확장 가능</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : AND 된 값</p> <p>IN1, IN2, OUT은 모두 같은 타입이어야 함.</p>																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	IN1	○	○	○	○	○														
	IN2	○	○	○	○	○														
	OUT	○	○	○	○	○														

■ 기능

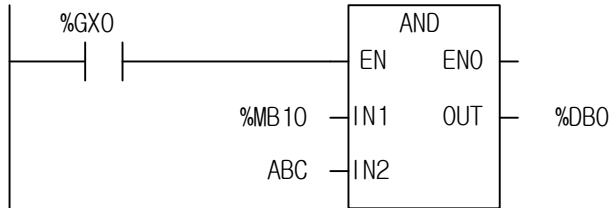
1. IN1을 IN2와 비트별로 AND 해서 OUT으로 출력시킵니다.

```

IN1  1111 ..... 0000
      &
IN2  1010 ..... 1010
OUT  1010 ..... 0000
    
```

■ 프로그램 예

1. LD



2. ST

ST 언어는 AND 는 지원하지 않습니다.

AND2_BYTE 인 경우

```
%DBO := AND2_BYTE(EN:=%GX0, IN1:= %MB10, IN2:= ABC);
```

- (1) 실행조건(%GX0)이 On 하면 평선 AND 가 실행됩니다.
- (2) IN1= %MB10 과 IN2 = ABC 값을 AND 시킨 결과가 OUT = %DBO 에 출력됩니다.

ASIN		적용 기종	발생플래그																	
Arc Sine 연산		XGF-M32E	_ERR, _LER																	
평 선		설 명																		
		<p>입력 EN : 1일 때 평선 실행 IN : Arc Sine 연산의 입력 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : Arc Sine 연산의 결과 값 (Radian)</p> <p>IN, OUT 은 같은 데이터 타입이어야 함</p>																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	IN														○	○				
	OUT														○	○				

■ 기능

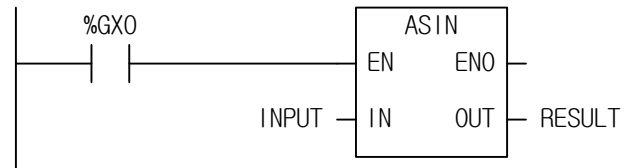
1. IN의 Arc Sine 값을 구해 OUT으로 출력시킵니다. 출력 값은 $-\pi/2$ 에서 $\pi/2$ 사이의 값이 됩니다.
OUT = ASIN (IN)

■ 에러

플래그	설명
_ERR	입력 값이 범위가 -1.0 과 1.0 사이에 있지 않을 경우 _ERR, _LER 플래그가 셋(SET)됩니다.

■ 프로그램 예

1. LD



2. ST

```
RESULT := ASIN(EN:=%GX0, IN1:= INPUT);
```

- (1) 실행조건(%GX0)이 On 하면 평선 ASIN 이 실행됩니다.
- (2) INPUT으로 선언된 입력 변수의 값이 0.8660... ($\sqrt{3}/2$) 일 때 출력 변수로 선언된 RESULT 은 1.0471... ($\pi/3$ rad = 60°) 입니다.

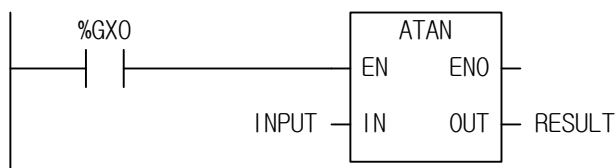
ATAN		적용 기종	발생플래그																	
Arc Tangent 연산		XGF-M32E	-																	
평 선		설 명																		
		<p>입력 EN : 1일 때 평선 실행 IN : Arc Tangent 연산의 입력 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : Arc Tangent 연산의 결과 값(Radian)</p> <p>IN, OUT은 같은 데이터 타입이어야 함</p>																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	IN														○	○				
	OUT														○	○				

■ 기능

1. IN의 Arc Tangent 값을 구해 OUT으로 출력시킵니다. 출력 값은 $-\pi/2$ 에서 $\pi/2$ 사이의 값이 됩니다.
OUT = ATAN (IN)

■ 프로그램 예

1. LD



2. ST

```
RESULT := ATAN(EN:=%GX0, IN1:= INPUT);
```

- (1) 실행조건(%GX0)이 On하면 평선 Arc Tangent 연산 평선 ATAN이 실행됩니다.
- (2) INPUT으로 선언된 입력 변수의 값이 1.0일 때 출력 변수로 선언된 RESULT은 0.7853... ($\pi/4$ rad = 45°) 입니다.

BOOL_TO_***		적용 기종	발생플래그																		
BOOL 타입 변환		XGF-M32E	-																		
평 선		설 명																			
<pre> graph LR subgraph BOOL_TO_*** EN[EN] IN[IN] ENO[ENO] OUT[OUT] end EN --- ENO IN --- OUT </pre>		<p>입력 EN : 1 일 때 평선 실행 IN : 타입 변환할 비트(1 비트)</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터</p>																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	OUT		○	○	○	○	○	○	○	○	○	○	○	○							

*ANY_BIT: ANY_BIT 타입 중 BOOL 제외

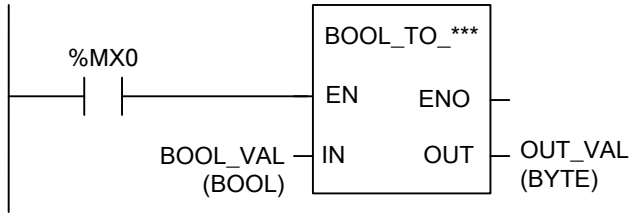
■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

평 선	출력 타입	동작 설명
BOOL_TO_SINT	SINT	BOOL 입력의 값이 2#0 면 정수값 '0' 을, 2#1 이면 정수값 '1' 을 출력 데이터 타입에 맞추어 출력합니다.
BOOL_TO_INT	INT	
BOOL_TO_DINT	DINT	
BOOL_TO_LINT	LINT	
BOOL_TO_USINT	USINT	
BOOL_TO_UINT	UINT	
BOOL_TO_UDINT	UDINT	
BOOL_TO_ULINT	ULINT	
BOOL_TO_BYTE	BYTE	BOOL 을 상위 비트들을 0 으로 채운 출력 데이터 타입 타입으로 변환합니다.
BOOL_TO_WORD	WORD	
BOOL_TO_DWORD	DWORD	
BOOL_TO_LWORD	LWORD	

■ 프로그램 예

1. LD



2. ST

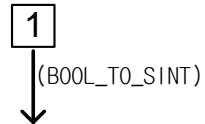
ST 언어는 BOOL_TO_***는 지원하지 않습니다.

BOOL_TO_BYTE 인 경우

```
OUT_VAL := BOOL_TO_BYTE(EN:=%MX0, IN:= BOOL_VAL);
```

- (1) 실행조건(%MX0)이 On 하면 평선 BOOL_TO_***이 실행됩니다.
- (2) 입력 변수로 선언된 BOOL_VAL(BOOL 타입) = 2#1 이면, 출력 변수로 선언된 OUT_VAL(BYTE 타입) = 2#0000_0001이 출력 됩니다.

입력(IN) : BOOL_VAL(BOOL) = 2#1



출력(OUT): OUT_VAL(BYTE) = 16#1

0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---

BYTE_TO_***		적용 기종										발생플래그										
BYTE 타입 변환		XGF-M32E										-										
평 선		설 명																				
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 비트(1 비트) 출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터																				
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		
	OUT	○		○	○	○	○	○	○	○	○	○	○	○								

*ANY_BIT : ANY_BIT 타입 중 BOOL 제외

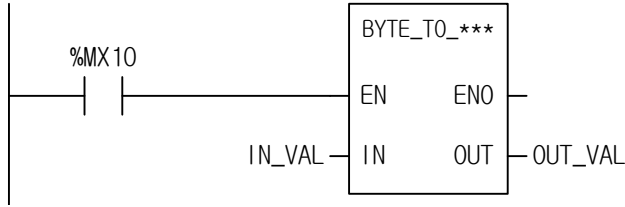
■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

평선	출력 타입	동작 설명
BYTE _TO_SINT	SINT	내부 비트 배열의 변환 없이 SINT 타입으로 변환합니다.
BYTE _TO_INT	INT	상위비트를 0으로 채워 INT 타입으로 변환합니다.
BYTE _TO_DINT	DINT	상위비트를 0으로 채워 DINT 타입으로 변환합니다.
BYTE _TO_LINT	LINT	상위비트를 0으로 채워 LINT 타입으로 변환합니다.
BYTE _TO_USINT	USINT	내부 비트 배열의 변환 없이 SINT 타입으로 변환합니다.
BYTE _TO_UINT	UINT	상위비트를 0으로 채워 UINT 타입으로 변환합니다.
BYTE _TO_UDINT	UDINT	상위비트를 0으로 채워 UDINT 타입으로 변환합니다.
BYTE _TO_ULINT	ULINT	상위비트를 0으로 채워 ULINT 타입으로 변환합니다.
BYTE _TO_BOOL	BOOL	하위 1비트를 취해 BOOL 타입으로 변환합니다.
BYTE _TO_WORD	WORD	상위비트를 0으로 채워 WORD 타입으로 변환합니다.
BYTE _TO_DWORD	DWORD	상위비트를 0으로 채워 DWORD 타입으로 변환합니다.
BYTE _TO_LWORD	LWORD	상위비트를 0으로 채워 LWORD 타입으로 변환합니다.

■ 프로그램 예

1. LD



2. ST

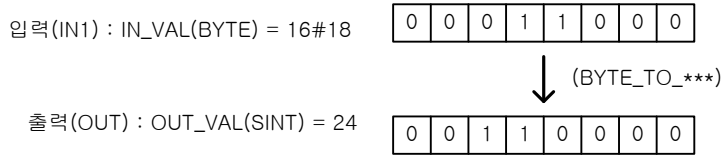
ST 언어는 BYTE_TO_***는 지원하지 않습니다.

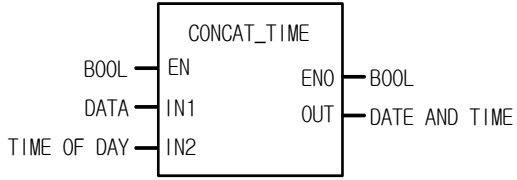
BYTE_TO_SINT 인 경우

OUT_VAL := BYTE_TO_SINT(EN:=%MX10, IN:= IN_VAL);

(1) 실행조건(%MX10)이 On 하면 평선 BYTE_TO_***이 실행됩니다.

(2) IN_VAL(BYTE 타입) = 2#0001_1000 이면, OUT_VAL(SINT 타입) = 24(2#0011_0000)가 됩니다



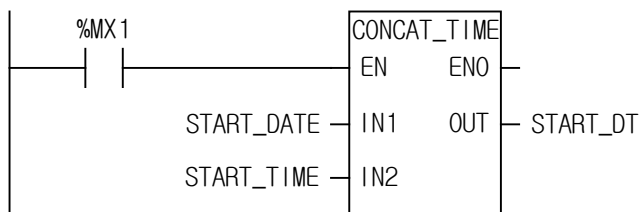
CONCAT_TIME	적용 기종	발생플래그
날짜와 시각 연결하기	XGF-M32E	-
평 선	설 명	
	<p>입력 EN : 1일 때 평선 실행 IN1 : 날짜 데이터 입력 IN2 : 시각 데이터 입력</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 날짜와 시각 출력</p>	

■ 기능

1. IN1(날짜)과 IN2(시각)를 붙여서 날짜와 시각(OUT)으로 출력합니다.

■ 프로그램 예

1. LD



2. ST

START_DT := CONCAT_TIME(EN:=%MX1, IN1:= START_DATE, IN2:= START_TIME);

- (1) 실행조건(%MX1)이 On 하면 평선 CONCAT_TIME 이 실행됩니다.
- (2) 운전시작 날짜 데이터 변수 START_DATE = D#1995-12-06 이고 운전시작 시각 START_TIME = TOD#08:30:00 이면 날짜와 시각이 합쳐진 START_DT 에는 DT#1995-12-06-08:30:00 이 출력됩니다.

입력(IN1) : START_DATE(DATE) = D#1995-12-06

(CONCAT_TIME)

입력(IN2) : START_TIME(TOD) = TOD#08:30:00



출력(OUT) : START_DT(DT) = DT#1995-12-06-08:30:00

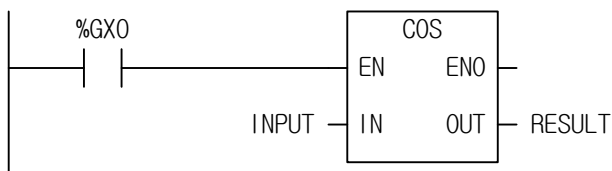
COS		적용 기종	발생플래그																		
Cosin 연산		XGF-M32E	-																		
평 선		설 명																			
		<p>입력 EN : 1일 때 평선 실행 IN : Cosine 연산의 각도 입력값(Radian)</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : Cosine 연산결과 값</p> <p>IN, OUT 은 같은 데이터 타입이어야 함</p>																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	IN														○	○					
	OUT														○	○					

■ 기능

1. IN의 Cosine 값을 구해 OUT으로 출력시킵니다.
OUT = COS (IN)

■ 프로그램 예

1. LD



2. ST

```
RESULT := COS(EN:=%GX0, IN:= INPUT);
```

- (1) 실행조건(%GX0)이 On 하면 평선 COS 이 실행됩니다.
- (2) INPUT 으로 선언된 입력 변수의 값이 0.5235 ($\pi/6$ rad = 30°)일때 출력 변수로 선언된 RESULT 은 0.8660 ($\sqrt{3}/2$) 입니다.

$$\text{COS} (\pi/6) = \sqrt{3}/2 = 0.866$$

입력(IN) : INPUT(REAL) =	0.5235
	↓ (COS)
출력(OUT) : RESULT(REAL) =	8.66074800E-01

DATE_TO_***		적용 기종										발생플래그									
DATE 타입변환		XGF-M32E										-									
평 선		설 명																			
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 날짜 데이터 출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	IN			○								○									

■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

평선	출력 타입	동작 설명
DATE_TO_UINT	UINT	DATE 를 UINT 타입으로 변환합니다.
DATE_TO_WORD	WORD	DATE 를 WORD 타입으로 변환합니다.

DINT_TO_***		적용 기종										발생플래그									
DINT 타입 변환		XGF-M32E										_ERR, _LER									
평선		설명																			
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 Double Integer 값 출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 타입 변환된 데이터																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	OUT	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			○	○	

*ANY: ANY 타입 중 DINT, TIME, DATE 제외

■ 기능

1. IN을 타입 변환해서 OUT으로 출력시킵니다.

평선	출력타입	동작 설명
DINT_TO_SINT	SINT	입력이 -128 ~ 127 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
DINT_TO_INT	INT	입력이 -32,768~32,767 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
DINT_TO_LINT	LINT	LINT 타입으로 정상 변환합니다.
DINT_TO_USINT	USINT	입력이 0~255 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
DINT_TO_UINT	UINT	입력이 0~65,535 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
DINT_TO_UDINT	UDINT	입력이 0~2,147,483,647 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
DINT_TO_ULINT	ULINT	입력이 0~2,147,483,647 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
DINT_TO_BOOL	BOOL	하위 1 비트를 취해 BOOL 타입으로 변환합니다.
DINT_TO_BYTE	BYTE	하위 8 비트를 취해 BYTE 타입으로 변환합니다.
DINT_TO_WORD	WORD	하위 16 비트를 취해 WORD 타입으로 변환합니다.

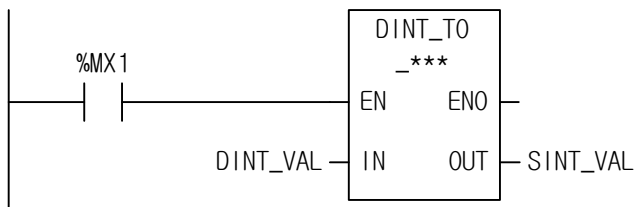
평선	출력타입	동작 설명
DINT_TO_DWORD	DWORD	내부 비트 배열의 변화 없이 DWORD 타입으로 변환합니다.
DINT_TO_LWORD	LWORD	상위 비트를 0으로 채운 LWORD 타입으로 변환합니다.
DINT_TO_REAL	REAL	DINT 를 REAL 타입으로 변환합니다. 변환 중 정밀도에 따른 오차가 발생할 수 있습니다.
DINT_TO_LREAL	LREAL	DINT 를 LREAL 타입으로 변환합니다. 변환 중 정밀도에 따른 오차가 발생할 수 있습니다.

■ 플래그

플래그	설명
_ERR	변환 에러 발생시 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 출력 타입의 비트 수 만큼 하위 비트를 취해 내부 비트 배열의 변화 없이 출력 시킵니다

■ 프로그램 예

1. LD



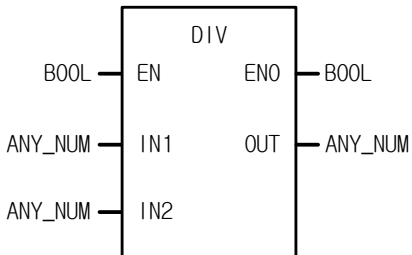
2. ST

ST 언어는 DINT_TO_***는 지원하지 않습니다.

DINT_TO_SINT 인 경우

```
SINT_VAL := DINT_TO_SINT(EN:= %MX1, IN:= DINT_VAL);
```

- (1) 실행조건(%MX1)이 On 하면 데이터 타입 변환 평선 DINT_TO_***가 실행됩니다.
- (2) INI = DINT_VAL(DINT 타입) = -77 이면, SINT_VAL(SINT 타입) = -77 이 됩니다.

DIV		적용 기종	발생플래그																	
나누기		XCF-M32E	_ERR, _LER																	
평 선		설 명																		
		<p>입력 EN : 1일 때 평선 실행 IN1 : 나누어질 값(피제수) IN2 : 나눌 값(제수)</p> <p>출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 나눈 결과 값(몫)</p> <p>IN1, IN2, OUT 에 연결되는 변수는 모두 같은 데이터 타입이어야 함.</p>																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	IN1						○	○	○	○	○	○	○	○	○	○				
	IN2						○	○	○	○	○	○	○	○	○	○				
	OUT						○	○	○	○	○	○	○	○	○	○				

■ 기능

1. IN1을 IN2로 나눠서 그 몫 중에서 소수점 이하를 버린 값을 OUT으로 출력시킵니다.

$$OUT = IN1/IN2$$

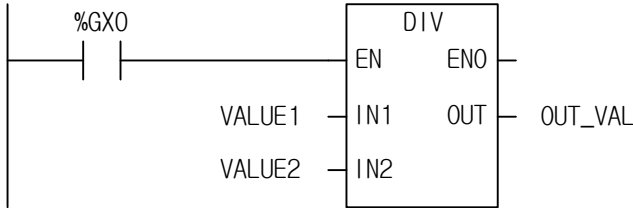
IN1	IN2	OUT	비 고
7	2	3	소수점 이하 버림
7	-2	-3	
-7	2	-3	
-7	-2	3	
7	0	×	에러

■ 플래그

플래그	설명
_ERR	나눌 값(제수)이 '0' 인 경우, 나눈 결과값이 각 타입의 최대값을 넘을 경우 _ERR, _LER 플래그가 셋(Set)됩니다.

■ 프로그램 예

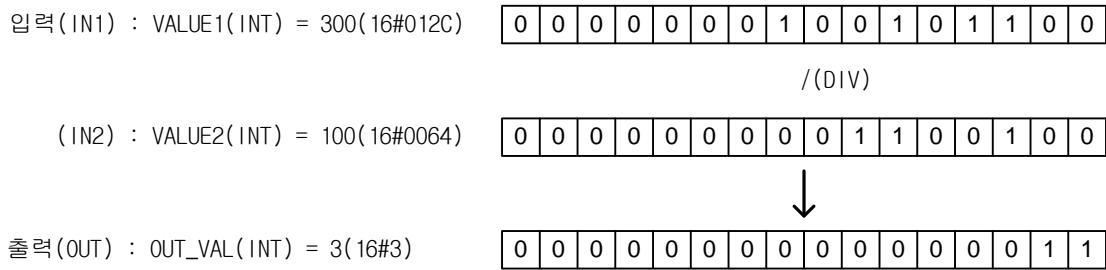
1. LD



2. ST

```
OUT_VAL := DIV(EN:= %GX0, IN1:= VALUE1, IN2:= VALUE2);
```

- (1) 실행조건(%GX0)이 On 하면 나누기 평선 DIV 가 실행됩니다.
- (2) 입력 변수 VALUE1 = 300, VALUE2 = 100 이면, 출력 변수로 선언된 OUT_VAL = 300/100 = 3 이 출력됩니다.



DT_TO_***		적용 기종										발생플래그								
DT 타입변환		XGF-M32E										-								
평 선		설 명																		
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 날짜와 시각 데이터 출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	OUT					○												○	○	

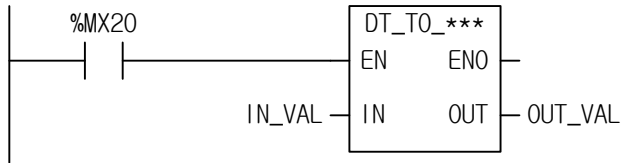
■ 기능

1. IN을 타입 변환해서 OUT으로 출력시킵니다.

평선	출력타입	동작 설명
DT_TO_LWORD	LWORD	DT를 LWORD 타입으로 변환합니다. (내부 데이터의 변환이 없어 역 변환이 가능합니다.)
DT_TO_DATE	DATE	DT를 DATE 타입으로 변환합니다.
DT_TO_TOD	TOD	DT를 TOD 타입으로 변환합니다.

■ 프로그램 예

1. LD



2. ST

ST 언어는 DT_TO_***는 지원하지 않습니다.

DT_TO_DATE 인 경우

```
OUT_VAL := DT_TO_DATE(EN:= %MX20, IN1:= IN_VAL);
```

- (1) 실행조건(%MX20)이 On 하면 DT 타입 변환 평선 DT_TO_***가 실행됩니다.
- (2) 입력 변수로 선언된 IN_VAL(DT 타입) = DT#1995-12-01-12:00:00 이면, 출력 변수로 선언된 OUT_VAL (DATE 타입) = D#1995-12-01 이 됩니다.

입력(IN) : IN_VAL(DT) = DT#1995-12-01-12:00:00

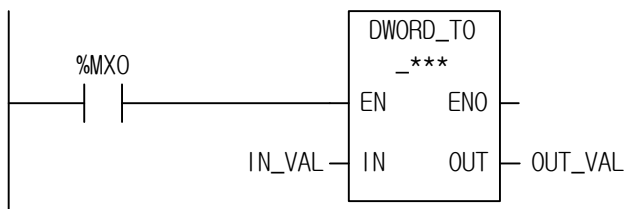
↓ (DT_TO_DATE)

출력(OUT) : OUT_VAL(DATE) = D#1995-12-01

평션	출력타입	동작 설명
DWORD_TO_TIME	TIME	내부 비트 배열의 변화 없이 TIME 타입으로 변환합니다.
DWORD_TO_TOD	TOD	내부 비트 배열의 변화 없이 TOD 타입으로 변환합니다. 단, TOD 범위 (TOD#23:59:59.999)를 벗어난 값 입력시 _ERR, _LER 플래그를 셋(SET) 하고 TOD 범위 내에서 순환하여 변환합니다.

■ 프로그램 예

1. LD



2. ST

ST 언어는 DWORD_TO_***는 지원하지 않습니다.

DWORD_TO_TOD 인 경우

```
OUT_VAL := DWORD_TO_TOD(EN:= %MX0, IN1:= IN_VAL);
```

- (1) 실행조건(%MX0)이 0n 하면 타입 변환 평션 DWORD_TO_TOD 가 실행됩니다.
- (2) IN_VAL(DWORD 타입) = 16#3E8(1000)이면, OUT_VAL(TOD 타입) = TOD#1S 가 됩니다.
- (3) TIME, TOD 는 10 진 값을 MS 단위로 환산해 계산합니다. 즉, 1000 은 1000MS = 1S 로 계산합니다.
(3.2.4. 데이터 타입별 구조 참조)

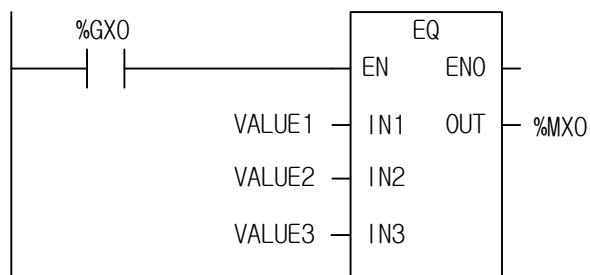
EQ		적용 기종	발생플래그																		
'같다' 비교		XGF-M32E	-																		
평선		설명																			
		<p>입력 EN : 1일 때 평선 실행 IN1 : 비교할 값 IN2 : 비교할 값 입력은 8 개까지 확장 가능 IN1, IN2, ...는 모두 같은 타입이어야 함.</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 비교 결과값</p>																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	IN1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	IN2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

■ 기능

1. IN1=IN2=IN3...=INn(n은 입력개수)이면 OUT 으로 1이 출력됩니다.
2. 다른 경우에는 OUT 으로 0이 출력됩니다.

■ 프로그램 예

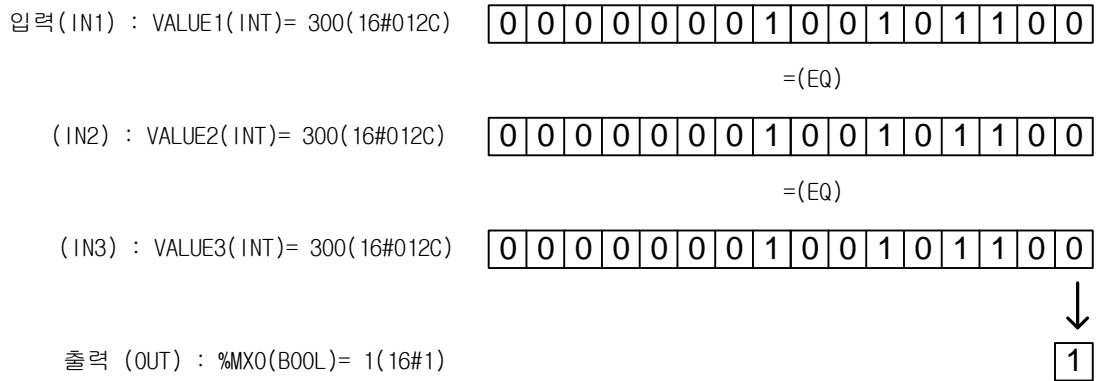
1. LD



2. ST

`%MX0 := EQ(EN:= %GX0, IN1:= VALUE1, IN2:= VALUE2, IN3:= VALUE3);`

- (1) 실행조건(%GX0)이 On 하면 비교 평선 'EQ' 가 실행됩니다.
- (2) VALUE1 = 300, VALUE2 = 300, VALUE3 = 300 이면, 비교결과 VALUE1 = VALUE2 = VALUE3 이므로 출력 값 %MX0 = 1 이 됩니다.



EXP		적용 기종														발생플래그								
EXP 연산		XGF-M32E														_ERR, _LER								
평 선		설 명																						
		<p>입력 EN : 1일 때 평선 실행 IN : 지수연산의 입력 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 지수연산 결과 값</p> <p>IN, OUT은 같은 데이터 타입이어야 함</p>																						
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT				
	IN														○	○								
	OUT														○	○								

■ 기능

1. IN의 지수연산 값을 구해 OUT으로 출력시킵니다.

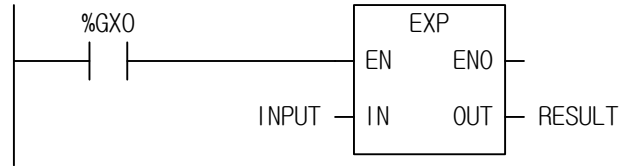
$$OUT = e^{IN}$$

■ 에러

플래그	설명
_ERR	출력 값이 해당 타입의 범위를 벗어 날 경우 _ERR, _LER 플래그가 셋(SET)됩니다.

■ 프로그램 예

1. LD



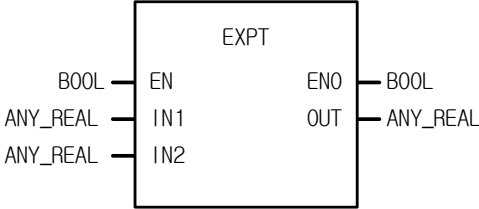
2. ST

```
RESULT := EXP(EN:= %GX0, IN1:= INPUT);
```

- (1) 실행조건(%GX0)이 On 하면 평선 EXP 가 실행됩니다.
- (2) INPUT 으로 선언된 입력 변수의 값이 2.0 일 때 출력 변수로 선언된 RESULT 은 7.3890... 입니다.

$$RESULT = e^{INPUT}$$

$$INPUT = 2.0, RESULT = 7.3890...$$

EXPT		적용 기종	발생플래그																																																																																
지수 연산		XGF-M32E	_ERR, _LER																																																																																
평 선		설 명																																																																																	
		<p>입력 EN : 1일 때 평선 실행 IN1 : 진수 IN2 : 지수</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 연산결과 값</p> <p>IN1, OUT 은 같은 데이터 타입이어야 함</p>																																																																																	
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>IN2</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>OUT</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN1														○	○					IN2														○	○					OUT														○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																																																
IN1														○	○																																																																				
IN2														○	○																																																																				
OUT														○	○																																																																				

■ 기능

1. IN1 에 IN2 를 지수승하여 OUT 으로 출력시킵니다.

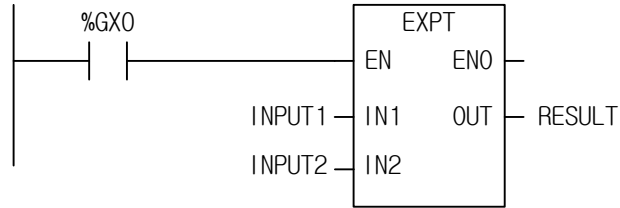
$$OUT = IN1^{IN2}$$

■ 예러

플래그	설명
_ERR	출력 값이 해당 타입의 범위를 벗어 날 경우 _ERR, _LER 플래그가 셋(SET)됩니다.

■ 프로그램 예

1. LD

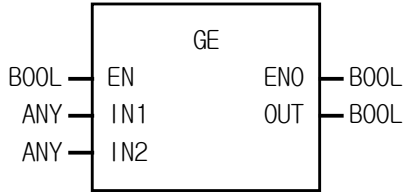


2. ST

RESULT := EXPT(EN:= %GX0, IN1:= INPUT1, IN2:= INPUT2);

- (1) 실행조건(%GX0)이 On 하면 평선 EXPT 가 실행됩니다.
- (2) INPUT 으로 선언된 입력 변수의 값 INPUT1 = 1.5, INPUT2 = 3 이면 출력 변수로 선언된 RESULT 는 3.375 가 됩니다.

$$3.375 = 1.5^3$$

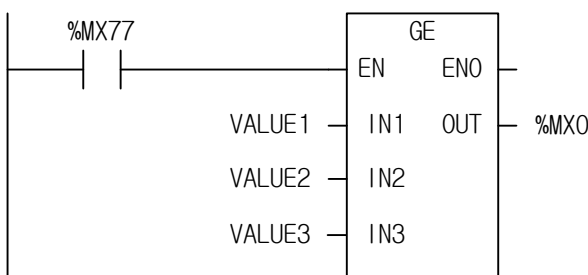
GE		적용 기종	발생플래그																																																															
'크거나 같다' 비교		XGF-M32E	-																																																															
평 선		설 명																																																																
		<p>입력 EN : 1일 때 평선 실행 IN1 : 비교할 값 IN2 : 비교할 값 입력은 8 개까지 확장 가능 IN1, IN2, ...는 모두 같은 타입이어야 함.</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 비교 결과값</p>																																																																
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> <th></th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> <tr> <td>IN2</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																															
IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															
IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															

■ 기능

1. $IN1 \geq IN2 \geq IN3 \dots \geq INn$ (n 은 입력개수)이면 OUT 으로 1 이 출력됩니다.
2. 다른 경우에는 OUT 으로 0 이 출력됩니다.

■ 프로그램 예

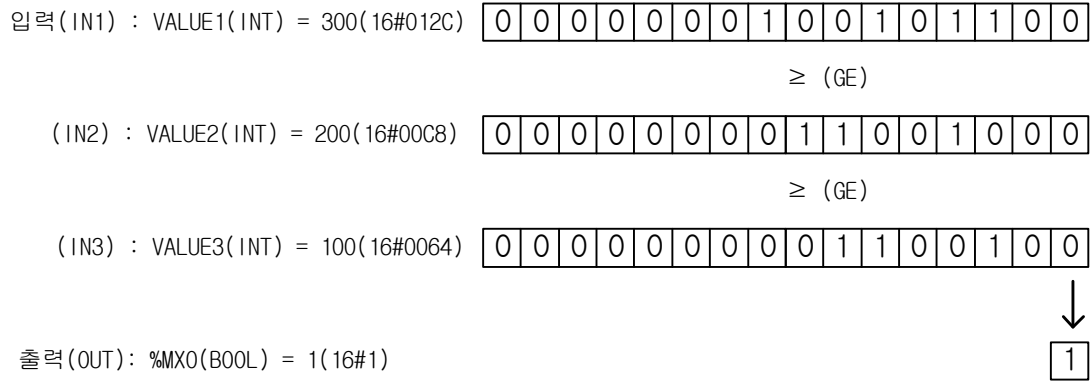
1. LD



2. ST

```
%MX0 := GE(EN:= %MX77, IN1:= VALUE1, IN2:= VALUE2, IN3:= VALUE3);
```

- (1) 실행 조건(%MX77)이 On 하면 GE(비교:크거나 같다) 평선이 실행됩니다.
- (2) 입력변수로 선언된 VALUE1=300, VALUE2=200, VALUE3=100 이면, 비교 결과 VALUE1 ≥ VALUE2 ≥ VALUE3 이므로, 출력 결과 값 %MX0=1 이 됩니다.



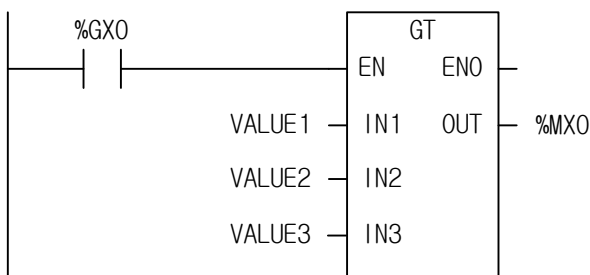
GT		적용 기종	발생플래그																																																															
'크다' 비교		XGF-M32E	-																																																															
평 선		설 명																																																																
		<p>입력 EN : 1일 때 평선 실행 IN1 : 비교될 값 IN2 : 비교할 값 입력은 8 개까지 확장 가능 IN1, IN2, ...는 모두 같은 타입이어야 함.</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 비교 결과 값</p>																																																																
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> <th></th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> <tr> <td>IN2</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																															
IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															
IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															

■ 기능

1. IN1>IN2>IN3...>INn(n은 입력 개수)이면 OUT 으로 1 이 출력됩니다.
2. 다른 경우에는 OUT 으로 0 이 출력됩니다.

■ 프로그램 예

1. LD



2. ST

`%MX0 := GT(EN:= %GX0, IN1:= VALUE1, IN2:= VALUE2, IN3:= VALUE3);`

(1) 실행조건(%GX0)이 On 하면 GT(비교: 크다) 평선이 실행됩니다.

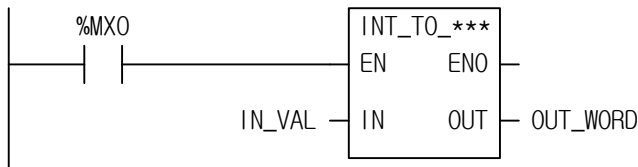
(2) 입력변수로 선언된 VALUE1 = 300, VALUE2 = 200, VALUE3 = 100 이면, 비교결과 VALUE1 > VALUE2 > VALUE3
이므로 출력결과값 %MX0 = 1 이 됩니다.

■ 플래그

플래그	설명
_ERR	변환에러 발생시 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 출력 타입의 비트 수 만큼 하위 비트를 취해 내부 비트 배열의 변환 없이 출력 시킵니다

■ 프로그램 예

1. LD



2. ST

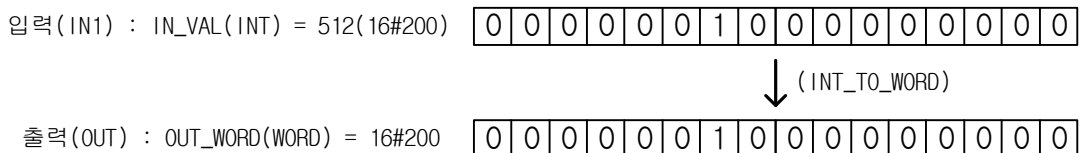
ST 언어는 INT_TO_***는 지원하지 않습니다.

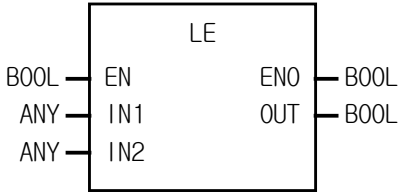
INT_TO_WORD 인 경우

OUT_WORD := INT_TO_WORD(EN:= %MX0, IN1:= IN_VAL);

(1) 입력조건(%MX0)이 0n 하면 INT_TO_*** 평선이 실행됩니다.

(2) 입력 변수로 선언된 IN_VAL(INT 타입) = 512(16#200)이면, 출력 변수로 선언된 OUT_WORD(WORD 타입) =16#200 이 됩니다.



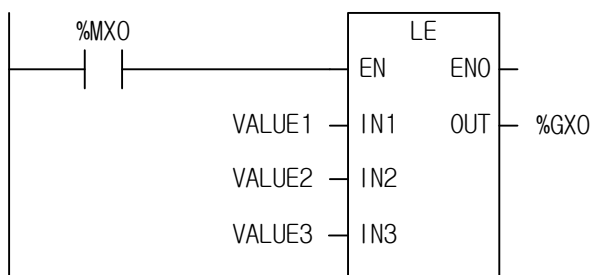
LE		적용 기종	발생플래그																																																															
'작거나 같다' 비교		XGF-M32E	-																																																															
평 선		설 명																																																																
		<p>입력 EN : 1 일 때 평선 실행 IN1 : 비교할 값 IN2 : 비교할 값 입력은 8 개까지 확장 가능 IN1, IN2, ...는 모두 같은 타입이어야 함.</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 비교 결과</p>																																																																
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> <th></th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> <tr> <td>IN2</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																															
IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															
IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															

■ 기능

1. $IN1 \leq IN2 \leq IN3 \dots \leq INn$ (n은 입력 개수)이면 OUT 으로 1 이 출력됩니다.
2. 다른 경우에는 OUT 으로 0 이 출력됩니다.

■ 프로그램 예

1. LD



2. ST

`%GX0 := LE(EN:= %MX0, IN1:= VALUE1, IN2:= VALUE2, IN3:= VALUE3);`

- (1) 실행조건(%MX0)이 On 하면 LE(비교: 작거나 같다) 평선이 실행됩니다.
- (2) 입력 변수로 선언된 VALUE1 = 100, VALUE2 = 200, VALUE3 = 300 이면, 비교결과 $VALUE1 \leq VALUE2 \leq VALUE3$ 이므로, 출력 결과값 %GX0 = 1 이 됩니다.

입력(IN1) : VALUE1(INT) = 100(16#0064)

0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(IN2) : VALUE2(INT) = 200(16#00C8)

0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(IN3) : VALUE3(INT) = 300(16#012C)

0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↓

출력(OUT): %GX0(BOOL) = 1(16#1)

1

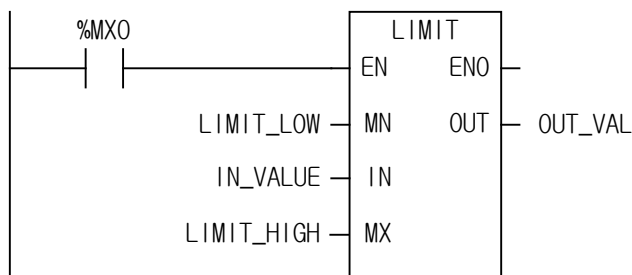
LIMIT		적용 기종	발생플래그																			
상하한 제한		XGF-M32E	-																			
평 선		설 명																				
		<p>입력 EN : 1일 때 평선 실행 MN : 최소값 IN : 제한될 값 MX : 최대값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 범위 안에 든 값</p> <p>MN, IN, MX, OUT 은 데이터 타입이 모두 같아야 함</p>																				
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		
	MN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	IN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	MX	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	OUT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

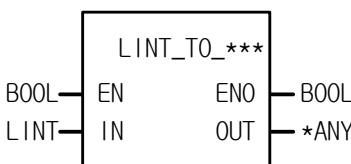
■ 기능

1. 입력 IN 값이 MN 과 MX 사이에 있으면, OUT 으로 IN 이 출력됩니다. 즉, $MN \leq IN \leq MX$ 이면 $OUT = IN$
2. 입력 IN 값이 MN 보다 작으면, OUT 으로 MN 이 출력됩니다. 즉, $IN < MN$ 이면 $OUT = MN$
3. 입력 IN 값이 MX 보다 크면, OUT 으로 MX 가 출력됩니다. 즉, $IN > MX$ 이면 $OUT = MX$

■ 프로그램 예

1. LD



LINT_TO_***		적용 기종										발생플래그									
LINT 타입변환		XGF-M32E										_ERR, _LER									
평선		설명																			
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 Long Integer 값										출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 타입 변환된 데이터									
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	OUT	○	○	○	○	○	○	○	○		○	○	○	○	○	○					

*ANY: ANY 타입 중 LINT, TIME, DATE, TOD, DT 제외

■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

평선	출력타입	동작 설명
LINT_TO_SINT	SINT	입력이 -128 ~ 127 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
LINT_TO_INT	INT	입력이 -32,768~32,767 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
LINT_TO_DINT	DINT	입력이 $-2^{31} \sim 2^{31}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
LINT_TO_USINT	USINT	입력이 0~ 255 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
LINT_TO_UINT	UINT	입력이 0~ 65,535 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
LINT_TO_UDINT	UDINT	입력이 0 ~ $2^{32}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
LINT_TO_ULINT	ULINT	입력이 0 ~ $2^{63}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
LINT_TO_BOOL	BOOL	하위 1 비트를 취해 BOOL 타입으로 변환합니다.
LINT_TO_BYTE	BYTE	하위 8 비트를 취해 BYTE 타입으로 변환합니다.
LINT_TO_WORD	WORD	하위 16 비트를 취해 WORD 타입으로 변환합니다.
LINT_TO_DWORD	DWORD	하위 32 비트를 취해 DWORD 타입으로 변환합니다.
LINT_TO_LWORD	LWORD	내부 비트 배열의 변화 없이 LWORD 타입으로 변환합니다.
LINT_TO_REAL	REAL	LINT 를 REAL 타입으로 변환합니다.

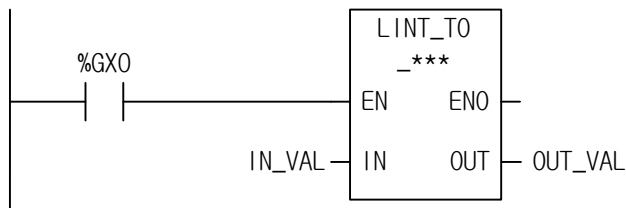
평선	출력타입	동작 설명
		변환 중 정밀도에 따른 오차가 발생할 수 있습니다.
LINT_TO_LREAL	LREAL	LINT 를 LREAL 타입으로 변환합니다. 변환 중 정밀도에 따른 오차가 발생할 수 있습니다.

■ 플래그

플래그	설명
_ERR	변환에러 발생시 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 출력 타입의 비트 수만큼 하위 비트를 취해 내부 비트 배열의 변환 없이 출력 시킵니다.

■ 프로그램 예

1. LD



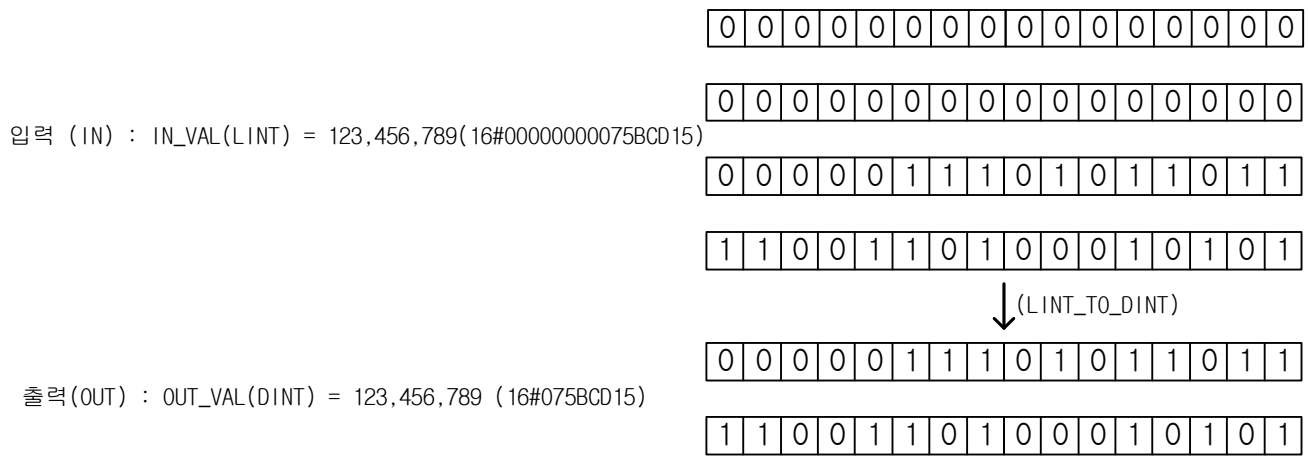
2. ST

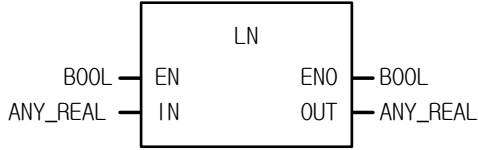
ST 언어는 LINT_TO_***는 지원하지 않습니다.

LINT_TO_DINT 인 경우

```
OUT_VAL := LINT_TO_DINT(EN:= %GX0, IN:= IN_VAL);
```

- (1) 입력조건 (%GX0)이 On 하면 LINT_TO_*** 평선이 실행됩니다.
- (2) 입력변수로 선언된 IN_VAL(LINT 타입) = 123,456,789 이면, OUT_VAL(DINT 타입) = 123,456,789 가 됩니다.



LN		적용 기종	발생플래그																																																												
LN 연산		XGF-M32E	_ERR, _LER																																																												
평 선		설 명																																																													
		<p>입력 EN : 1 일 때 평선 실행 IN : 자연대수 연산의 입력 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 자연대수 값</p> <p>IN, OUT 은 같은 데이터 타입이어야 함</p>																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN														○	○					OUT														○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN														○	○																																																
OUT														○	○																																																

■ 기능

1. IN의 자연대수 값을 구해 OUT으로 출력시킵니다.

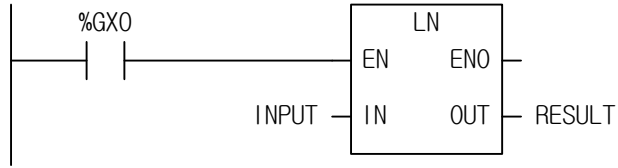
$$OUT = \ln(IN)$$

■ 에러

플래그	설명
_ERR	입력 값이 0 또는 음수일 때 _ERR, _LER 플래그가 셋(SET)됩니다.

■ 프로그램 예

1. LD



2. ST

```
RESULT := LN(EN:= %GX0, IN1:= INPUT);
```

- (1) 실행조건(%GX0)이 On 하면 LN 평선이 실행됩니다.
- (2) INPUT으로 선언된 입력 변수의 값이 2.0 일 때 출력 변수로 선언된 RESULT은 0.6931... 입니다.
 $\ln(2.0) = 0.6931\dots$

LOG		적용 기종	발생플래그																																																												
LOG 연산		XGF-M32E	_ERR, _LER																																																												
평 선		설 명																																																													
		<p>입력 EN : 1 일 때 평선 실행 IN : 상용대수 연산의 입력 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 상용대수 연산결과 값</p> <p>IN, OUT 은 같은 데이터 타입이어야 함</p>																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN														○	○					OUT														○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN														○	○																																																
OUT														○	○																																																

■ 기능

1. IN의 상용대수 값을 구해 OUT으로 출력시킵니다.

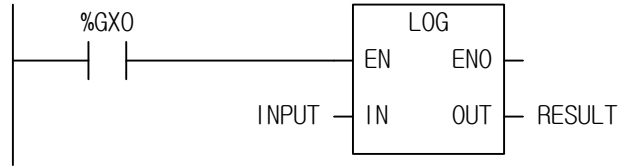
$$OUT = \log_{10} (IN) = \log (IN)$$

■ 에러

플래그	설명
_ERR	입력 값이 0 또는 음수일 때 _ERR, _LER 플래그가 셋(SET)됩니다.

■ 프로그램 예

1. LD



2. ST

```
RESULT := LOG(EN:= %GX0, IN:= INPUT);
```

- (1) 실행조건(%GX0)이 On 하면 LOG 평선이 실행됩니다.
- (2) INPUT 으로 선언된 입력 변수의 값이 2.0 일 때 출력 변수로 선언된 RESULT 은 0.3010... 입니다.
 $\text{Log}_{10}(2.0) = 0.3010\dots$

LREAL_TO_***		적용 기종	발생플래그
LREAL 타입변환		XGF-M32E	_ERR, _LER
평 선		설 명	
		입력 EN : 1 일 때 평선 실행 IN : 타입 변환할 LREAL 값 출력 ENO : 에러 없이 실행되면 1 을 출력 OUT : 타입 변환된 데이터	
ANY 타입 변수설명	변수명 OUT	BOOL BYTE WORD DWORD LWORD SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL TIME DATE TOD DT	

■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

평선	출력타입	동작 설명
LREAL_TO_SINT	SINT	입력의 정수 부분이 -128 ~ 127 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
LREAL_TO_INT	INT	입력의 정수 부분이 -32,768 ~ 32,767 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
LREAL_TO_DINT	DINT	입력의 정수 부분이 $-2^{31} \sim 2^{31}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
LREAL_TO_LINT	LINT	입력의 정수 부분이 $-2^{63} \sim 2^{63}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
LREAL_TO_USINT	USINT	입력의 정수 부분이 0 ~ 255 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
LREAL_TO_UINT	UINT	입력의 정수 부분이 0 ~ 65,535 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
LREAL_TO_UDINT	UDINT	입력의 정수 부분이 0 ~ $2^{32}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
LREAL_TO_ULINT	ULINT	입력의 정수 부분이 0 ~ $2^{64}-1$ 일 경우 정상 변환되나, 그 외 값은 에

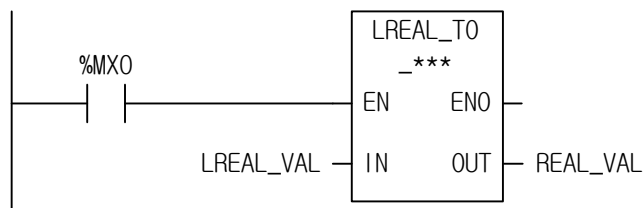
평선	출력타입	동작 설명
		러가 발생합니다. (소수점 이하는 반올림)
LREAL_TO_LWORD	LWORD	내부 비트 배열의 변화 없이 LWORD 타입으로 변환합니다.
LREAL_TO_REAL	REAL	LREAL 을 REAL 타입으로 정상 변환합니다. 변환 중 정밀도에 따른 오차가 발생할 수 있습니다.

■ 플래그

플래그	설명
_ERR	입력 값이 출력 타입에 저장할 수 있는 값보다 커서 오버 플로(Overflow) 발생시 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 결과값에 0을 출력합니다.

■ 프로그램 예

1. LD



2. ST

ST 언어는 LREAL_TO_***는 지원하지 않습니다.

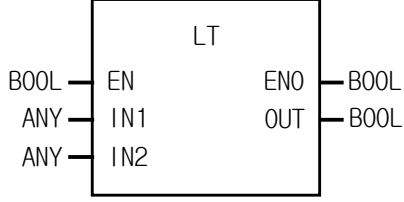
LREAL_TO_REAL 인 경우

```
REAL_VAL := LREAL_TO_REAL(EN:= %MX0, IN:= LREAL_VAL);
```

- (1) 입력조건(%MX0)이 On 하면, LREAL_TO_*** 평선이 실행됩니다.
- (2) 입력변수로 선언된 LREAL_VAL(LREAL 타입) = -1.34E-12 이면,
출력변수로 선언된 REAL_VAL(REAL 타입) = -1.34E-12 가 됩니다.

입력(IN) : LREAL_VAL (LREAL) = -1.34E-12
↓
(LREAL_TO_REAL)

출력(OUT) : REAL_VAL (REAL) = -1.34E-12

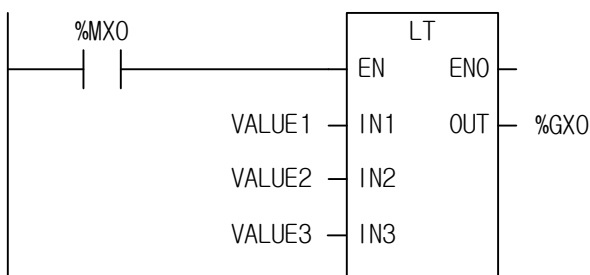
LT		적용 기종	발생플래그																																																															
'작다' 비교		XGF-M32E	-																																																															
평 선		설 명																																																																
		<p>입력 EN : 1 일 때 평선 실행 IN1 : 비교할 값 IN2 : 비교할 값 입력은 8 개까지 확장 가능 IN1, IN2, ...는 모두 같은 타입이어야 함.</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 비교 결과값</p>																																																																
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> <th></th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> <tr> <td>IN2</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																															
IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															
IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															

■ 기능

1. $IN1 < IN2 < IN3 \dots < INn$ (n 은 입력개수)이면 OUT 으로 1 이 출력됩니다.
2. 다른 경우에는 OUT 으로 0 이 출력됩니다.

■ 프로그램 예

1. LD



2. ST

```
%GX0 := LT(EN:= %MX0, IN1:= VALUE1, IN2:= VALUE2, IN3:= VALUE3);
```

- (1) 실행조건(%MX0)이 On 하면 LT(비교: 작다) 평선이 실행됩니다.
- (2) 입력변수로 선언된 VALUE1 = 100, VALUE2 = 200, VALUE3 = 300 이면, 비교결과 VALUE1 < VALUE2 < VALUE3 이므로 출력 결과값 %GX0 = 1 이 됩니다.

입력(IN1) : VALUE1(INT) = 100(16#0064)

0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<(LT)

(IN2) : VALUE2(INT) = 200(16#00C8)

0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<(LT)

(IN3) : VALUE3(INT) = 300(16#012C)

0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↓

출력(OUT) : %GX0(BOOL) = 1(16#0001)

1

LWORD_TO_***		적용 기종	발생플래그																		
LWORD 타입변환		XGF-M32E	-																		
평 선		설 명																			
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 비트 열(64 비트)	출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	OUT	○	○	○	○		○	○	○	○	○	○	○	○		○					○

*ANY: ANY 타입 중 LWORD, REAL, TIME, DATE, TOD 제외

■ 기능

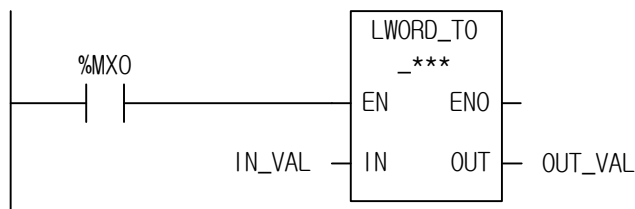
1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

평선	출력타입	동작 설명
LWORD_TO_SINT	SINT	하위 8 비트를 취하여 SINT 타입으로 변환합니다.
LWORD_TO_INT	INT	하위 16 비트를 취하여 INT 타입으로 변환합니다.
LWORD_TO_DINT	DINT	하위 32 비트를 취하여 DINT 타입으로 변환합니다.
LWORD_TO_LINT	LINT	내부 비트 배열의 변화 없이 LINT 타입으로 변환합니다.
LWORD_TO_USINT	USINT	하위 8 비트를 취하여 USINT 타입으로 변환합니다.
LWORD_TO_UINT	UINT	하위 16 비트를 취하여 UINT 타입으로 변환합니다.
LWORD_TO_UDINT	UDINT	하위 32 비트를 취하여 UDINT 타입으로 변환합니다.
LWORD_TO_ULINT	ULINT	내부 비트 배열의 변화 없이 ULINT 타입으로 변환합니다.
LWORD_TO_BOOL	BOOL	하위 1 비트를 취하여 BOOL 타입으로 변환합니다.
LWORD_TO_BYTE	BYTE	하위 8 비트를 취하여 BYTE 타입으로 변환합니다.
LWORD_TO_WORD	WORD	하위 16 비트를 취하여 WORD 타입으로 변환합니다.
LWORD_TO_DWORD	DWORD	하위 32 비트를 취하여 DWORD 타입으로 변환합니다.

평선	출력타입	동작 설명
LWORD_TO_LREAL	LREAL	LWORD 를 LREAL 타입으로 변환합니다.
LWORD_TO_DT	DT	내부 비트 배열의 변화 없이 DT 타입으로 변환합니다. 단 DT 범위 (DT#2163-12-31-23:59:59:999)를 벗어난 값 입력시 _ERR, _LER 플래그를 SET 하고 DT 범위 내에서 순환하여 변환합니다.

■ 프로그램 예

1. LD



2. ST

ST 언어는 LWORD_TO_***는 지원하지 않습니다.

LWORD_TO_LINT 인 경우

```
OUT_VAL := LWORD_TO_LINT(EN:= %MX0, IN:= IN_VAL);
```

- (1) 입력조건(%MX0)이 On 하면 LWORD_TO_*** 평선이 실행됩니다.
- (2) 입력변수로 선언된 IN_VAL(LWORD 타입) = 16#FFFFFF_FFFF_FFFF_FFFF 면, 출력변수로 선언된 OUT_VAL(LINT 타입) = -1(16#FFFFFF_FFFF_FFFF_FFFF)이 됩니다.

입력(IN) : IN_VAL(LWORD) = 16#FFFFFFFFFFFFFFFF
↓ (LWORD_TO_LINT)
출력(OUT) : OUT_VAL(LINT) = -1

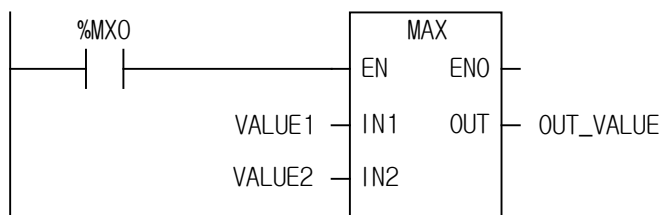
MAX		적용 기종	발생플래그																																																																																								
최대값		XGF-M32E	-																																																																																								
평 선		설 명																																																																																									
		<p>입력 EN : 1일 때 평선 실행 IN1 : 비교될 값 IN2 : 비교될 값 입력 8 개까지 확장 가능</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 입력 값 중 최대값</p> <p>IN1, IN2, ..., OUT 은 모두 같은 타입이어야 함.</p>																																																																																									
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>IN2</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>OUT</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT			IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	OUT	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																																																								
IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																						
IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																						
OUT	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																						

■ 기능

1. 입력 IN1, IN2, ..., INn(n은 입력 개수)중에서 최대값을 OUT 으로 출력시킵니다.

■ 프로그램 예

1. LD

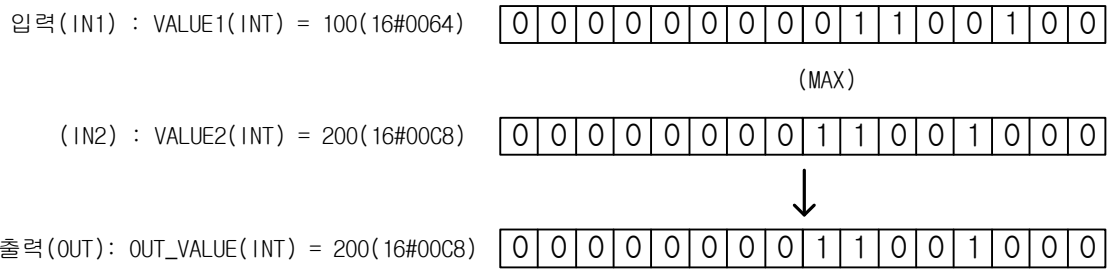


2. ST

```
OUT_VALUE := MAX(EN:= %MX0, IN1:= VALUE1, IN2:= VALUE2);
```

- (1) 실행조건(%MX0)이 On 하면 MAX(최대값) 평선이 실행됩니다.
- (2) 입력변수로 선언된 VALUE1 = 100, VALUE2 = 200 을 비교결과 최대값이 200 이므로 출력변수로 선언된

OUT_VALUE = 200 으로 출력됩니다.



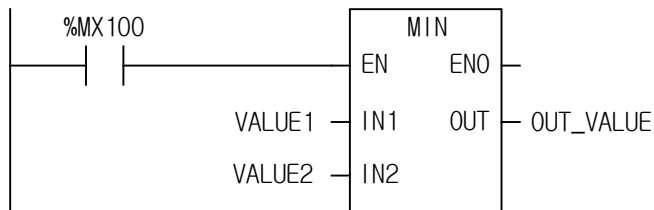
MIN		적용 기종	발생플래그																		
최소값		XGF-M32E	-																		
평 선		설 명																			
		<p>입력 EN : 1일 때 평선 실행 IN1 : 비교될 값 IN2 : 비교될 값 입력 8 개까지 확장 가능</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 입력 값 중 최소값</p> <p>IN1, IN2, ..., OUT 은 모두 같은 타입이어야 함.</p>																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	IN1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	IN2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	OUT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

■ 기능

1. 입력 IN1, IN2, ..., INn(n은 입력 개수) 중에서 최소값을 OUT 으로 출력시킵니다.

■ 프로그램 예

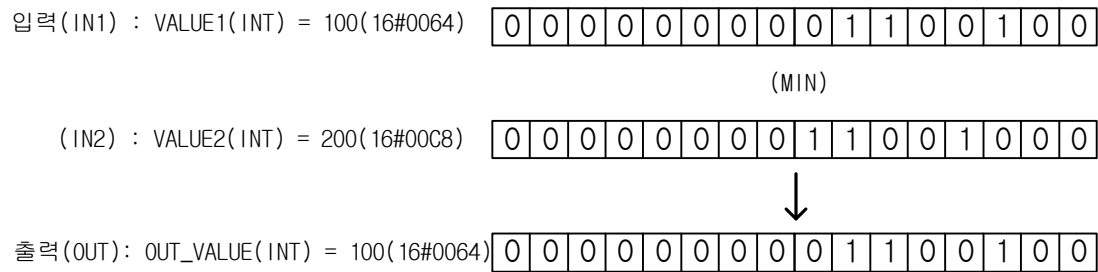
1. LD

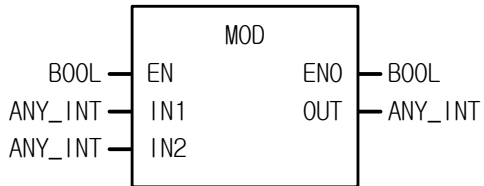


2. ST

```
OUT_VALUE := MIN(EN:= %MX100, IN1:= VALUE1, IN2:= VALUE2);
```

- (1) 실행조건(%MX100)이 0n 하면 MIN(최소값) 평선이 실행됩니다.
 (2) 입력변수로 선언된 VALUE1 = 100, VALUE2 = 200 을 비교결과 최소값이 100 이므로 출력변수로 선언된 OUT_VALUE = 100 이 출력됩니다.



MOD		적용 기종	발생플래그																				
나머지 구하기		XGF-M32E	-																				
평 선		설 명																					
		<p>입력 EN : 1일 때 평선 실행 IN1 : 나누어 질 값(피제수) IN2 : 나눌 값(제수)</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 나눈 결과값(나머지)</p> <p>IN1, IN2, OUT 에 연결되는 변수는 모두 같은 데이터 타입이어야 함.</p>																					
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT			
	IN1						○	○	○	○	○	○	○	○									
	IN2						○	○	○	○	○	○	○	○									
	OUT						○	○	○	○	○	○	○	○									

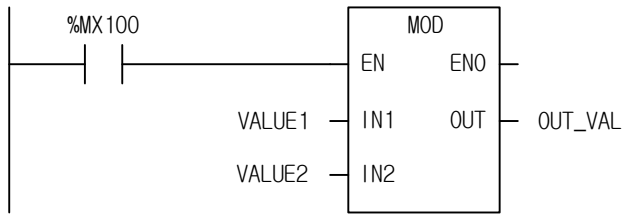
■ 기능

1. IN1 을 IN2 로 나눠서 그 나머지를 OUT 으로 출력시킵니다.
 $OUT = IN1 - (IN1/IN2) \times IN2$ (단, $IN2 = 0$ 이면 $OUT = 0$)

IN1	IN2	OUT
7	2	1
7	-2	1
-7	2	-1
-7	-2	-1
7	0	0

■ 프로그램 예

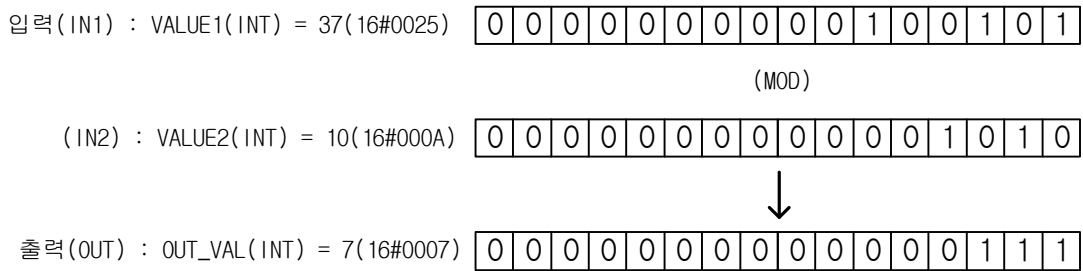
1. LD



2. ST

```
OUT_VAL := MOD(EN:= %MX100, IN1:= VALUE1, IN2:= VALUE2);
```

- (1) 실행조건(%MX100)이 0n 하면 MOD(나머지 구하기) 평선이 실행합니다.
- (2) 입력변수 중 나누어질 값 VALUE1 = 37 이고, 나눌 값 VALUE2 = 10 이면, 출력변수로 선언된 OUT_VAL 의 값은 37 을 10 으로 나눈 나머지 7 이 됩니다.



MOVE		적용 기종	발생플래그
데이터 복사		XGF-M32E	-
평 선		설 명	
		<p>입력 EN : 1 일 때 평선 실행 IN : MOVE 할 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : MOVE 된 값</p> <p>IN, OUT 에 연결되는 변수는 같은 데이터 타입이어야 함</p>	
ANY 타입 변수설명	변수명	BOOL	BYTE
		WORD	DWORD
		LWORD	SINT
		INT	DINT
		LINT	USINT
		UINT	UDINT
		ULINT	REAL
		LREAL	TIME
		DATE	TOD
		DT	

	IN	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	OUT	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

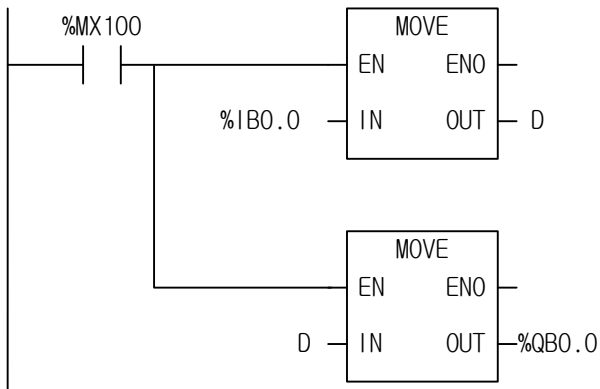
■ 기능

1. IN의 값을 OUT으로 이동합니다.

■ 프로그램 예

입력 %IX0.0~%IX0.7의 8점의 입력상태를 변수 D로 전송한 후, 전송된 데이터를 출력 %QX0.0~%QX0.7의 8점으로 출력시키는 프로그램

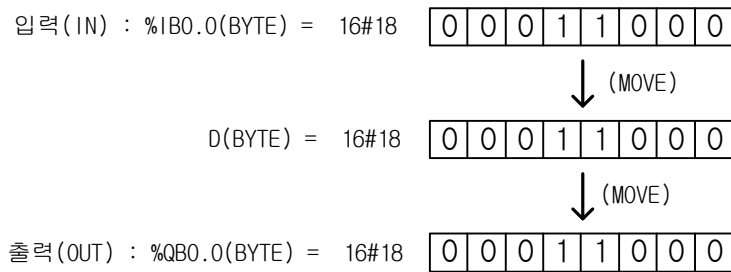
1. LD

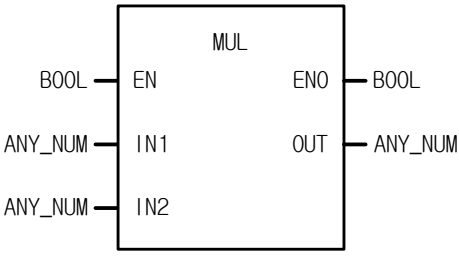


2. ST

```
D := MOVE(EN:= %MX100, IN:= %IB0.0);
%QB0.0 := MOVE(EN:= %MX100, IN:= D);
```

- (1) 실행조건(%MX100)가 On 되면 MOVE(데이터 복사) 평선이 실행됩니다.
 (2) 첫번째 MOVE 평선에 의해 입력모듈의 8 점 입력 데이터가 변수 D 영역으로 옮겨지고 두번째 MOVE 평선에 의해 변수 D 에 저장된 입력모듈의 상태가 출력모듈로 출력됩니다.



MUL		적용 기종	발생플래그																																																																																
곱하기		XGF-M32E	_ERR, _LER																																																																																
평 선		설 명																																																																																	
		<p>입력 EN : 1일 때 평선 실행 IN : 곱해질 값 (피승수) IN2 : 곱할 값 (승수) 입력은 8개까지 확장 가능</p> <p>출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 곱한 결과 값</p> <p>IN1, IN2, ..., OUT 에 연결되는 변수는 모두 같은 데이터 타입이어야 함</p>																																																																																	
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>IN2</td> <td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>OUT</td> <td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN1						○	○	○	○	○	○	○	○	○	○					IN2						○	○	○	○	○	○	○	○	○	○					OUT						○	○	○	○	○	○	○	○	○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																																																
IN1						○	○	○	○	○	○	○	○	○	○																																																																				
IN2						○	○	○	○	○	○	○	○	○	○																																																																				
OUT						○	○	○	○	○	○	○	○	○	○																																																																				

■ 기능

1. IN1, IN2, ..., INn (n은 입력 개수)를 곱해서 OUT으로 출력시킵니다.
 $OUT = IN1 \times IN2 \times \dots \times INn$

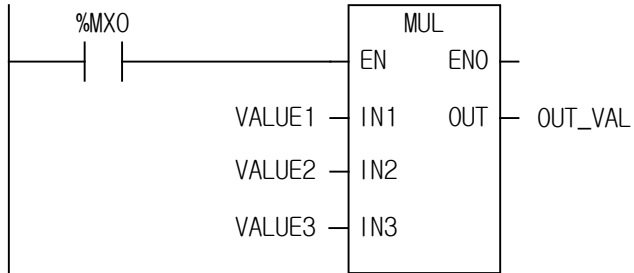
■ 플래그

플래그	설명
_ERR	출력 값이 해당 데이터 타입의 범위를 벗어날 경우 _ERR, _LER 플래그가 셋(Set)됩니다.

☆ REAL, LREAL 타입 연산에서는 IN1 에서 IN8 로 순차적으로 연산을 하기 때문에 중간 연산과정에서 이미 REAL, LREAL 에 최대값이나 최소값을 넘게 되면 _ERR, _LER 플래그가 셋(Set)되고 결과값은 무한대 값 또는 비정상적인 값 (1.#INF000000000000e+000, 1.#SNAN000000000000e+000, 1.#QNAN000000000000e+000)으로 표시됩니다.

■ 프로그램 예

1. LD



2. ST

```
OUT_VAL := MUL(EN:= %MX0, IN1:= VALUE1, IN2:= VALUE2, IN3:= VALUE3);
```

(1) 실행조건(%MX0)이 On 하면 MUL(곱하기) 평선이 On 합니다.

(2) MUL 평선의 입력변수로 선언된 VALUE1 = 30, VALUE2 = 20, VALUE3 = 10 이면, 출력변수로 선언된 OUT_VAL = 30 × 20 × 10 = 6,000 이 됩니다.

입력(IN1) : VALUE1(INT) = 30(16#001E)	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0
	+(MUL)
(IN2) : VALUE2(INT) = 20(16#0014)	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0
	+(MUL)
(IN3) : VALUE3(INT) = 10(16#000A)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
	↓
출력(OUT) : OUT_VAL(INT) = 6,000(16#1770)	0 0 0 1 0 1 1 1 0 1 1 1 0 0 0 0 0 0

MUX		적용 기종	발생플래그																		
여러 개 중 선택		XGF-M32E	_ERR, _LER																		
평 선		설 명																			
		<p>입력 EN : 1일 때 평선 실행 K : 선택 IN0 : 선택될 값 IN1 : 선택될 값 입력은 8 개까지 확장 가능(IN0, IN1, ..., IN7)</p> <p>출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 선택된 값 IN0, IN1, ..., OUT 은 모두 같은 타입이어야 함</p>																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	IN0	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	OUT	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

■ 기능

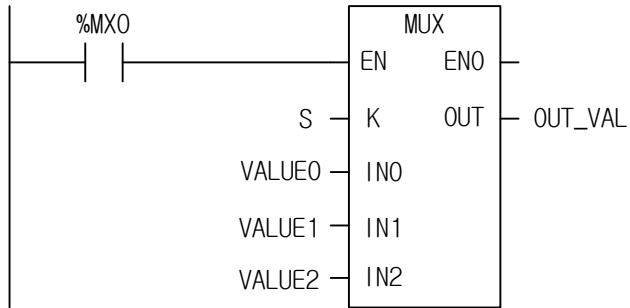
1. K값으로 여러 입력(IN0, IN1, ..., INn) 중 하나를 선택하여 출력시킵니다.
2. K = 0 이면 IN0 이, K = 1 이면 IN1 이, K = n 이면 INn 이 OUT 으로 출력됩니다.

■ 플래그

플래그	설명
_ERR	K 의 값이 입력 변수 INn 의 개수보다 크거나 같은 경우에 OUT 으로는 IN0 값이 출력되고, _ERR, _LER 플래그가 셋(Set)됩니다. K 값이 음수일 때 _ERR, _LER 플래그가 셋(SET)됩니다.

■ 프로그램 예

1. LD



2. ST

```
OUT_VAL := MUX(EN:= %MX0, K:= S, IN0:= VALUE0, IN1:= VALUE1, IN2:= VALUE2);
```

- (1) 실행조건(%MX0)이 On 하면 MUX(여러 개를 선택) 평선이 실행됩니다.
 (2) 입력변수로 설정된 VALUE0, 1, 2 중 선택변수 S의 값에 따라 선택되어 출력변수로 선언된 OUT 으로 출력시킵니다.

```
입력 (K) : S(INT) = 2
           (IN0) : VALUE0(WORD) = 16#0011
           (IN1) : VALUE1(WORD) = 16#0022
           (IN2) : VALUE2(WORD) = 16#0033
                               ↓ (MUX)
출력 (OUT) : OUT_VAL(WORD) = 16#0033
```

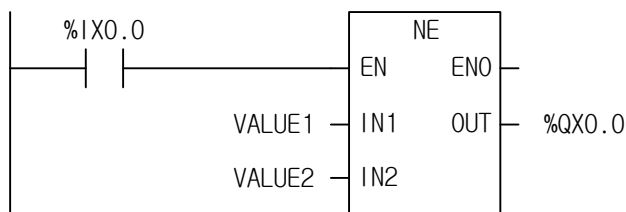
NE		적용 기종	발생플래그																																																															
'같지 않다' 비교		XGF-M32E	-																																																															
평 선		설 명																																																																
		<p>입력 EN : 실행 허용 IN1 : 비교될 값 IN2 : 비교될 값 IN1, IN2 는 같은 타입이어야 함.</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 비교 결과값</p>																																																																
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> <th></th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> <tr> <td>IN2</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																															
IN1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															
IN2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																															

■ 기능

1. IN1 이 IN2 와 같지 않으면 OUT 으로 1 이 출력됩니다.
2. 같으면 OUT 으로 0 이 출력됩니다.

■ 프로그램 예

1. LD

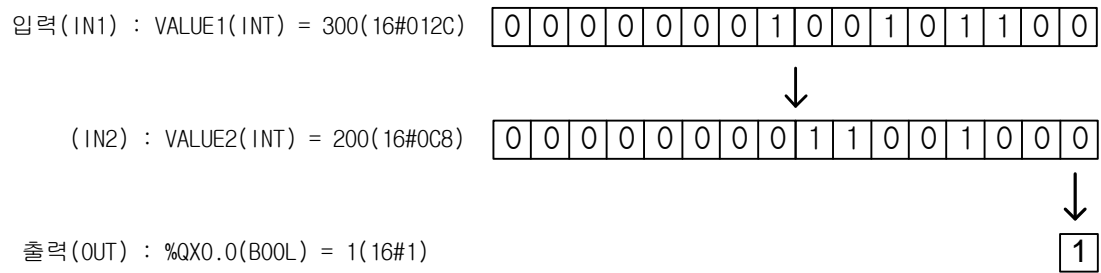


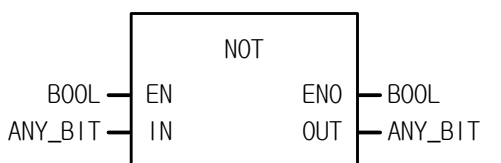
2. ST

```
%QX0.0 := NE(EN:= %IX0.0, IN1:= VALUE1, IN2:= VALUE2);
```

- (1) 실행조건(%IX0.0)이 On 하면 NE(비교: 같지 않다) 평선이 실행됩니다.
- (2) 입력변수로 선언된 VALUE1 = 300, VALUE2 = 200 이면, 비교결과 VALUE1 과 VALUE2 가 다르므로 출력결과값

%QX0.1 = 1 이 됩니다



NOT		적용 기종	발생플래그																																																												
논리 반전		XGF-M32E	-																																																												
평 선		설 명																																																													
		<p>입력 EN : 1일 때 평선 실행 IN : NOT 될 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : NOT 된 값</p> <p>IN, OUT은 모두 같은 타입이어야 함.</p>																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN	○	○	○	○	○															OUT	○	○	○	○	○																
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN	○	○	○	○	○																																																										
OUT	○	○	○	○	○																																																										

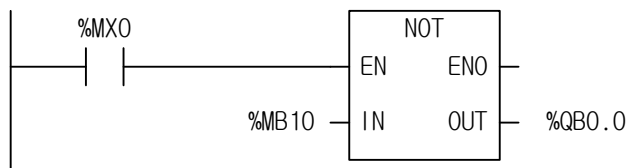
■ 기능

1. IN을 비트별로 NOT(반전)해서 OUT으로 출력시킵니다.

```
IN  1100 ..... 1010
OUT 0011 ..... 0101
```

■ 프로그램 예

1. LD



2. ST

```
%QB0.0 := NOT_BYTE(EN:= %MX0, IN1:=MB10);
```

- (1) 실행조건(%MX0)이 On 하면 NOT(논리반전) 평선이 실행됩니다.
- (2) NOT 평선이 실행되면 입력변수로 선언된 %MB10의 데이터 값이 반전되어 출력변수로 선언된 %QB0.0에 출력됩니다.

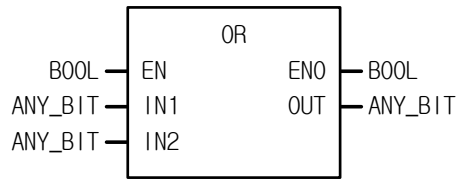
입력 (IN) : %MB10(BYTE) = 16#CC

1	1	0	0	1	1	0	0
---	---	---	---	---	---	---	---

↓ (NOT)

출력 (OUT) : %QB0.0(BYTE) = 16#33

0	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

OR		적용 기종	발생플래그																																																															
논리합		XGF-M32E	-																																																															
평 선		설 명																																																																
		<p>입력 EN : 1일 때 평선 실행 IN1 : OR 될 값 IN2 : OR 될 값 입력 8 개까지 확장 가능</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : OR 된 값</p> <p>IN1, IN2, OUT 은 모두 같은 타입이어야 함.</p>																																																																
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> <th></th> </tr> </thead> <tbody> <tr> <td>IN</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		IN	○	○	○	○	○																OUT	○	○	○	○	○																	
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																															
IN	○	○	○	○	○																																																													
OUT	○	○	○	○	○																																																													

■ 기능

1. IN1 을 IN2 와 비트별로 OR 해서 OUT 으로 출력시킵니다.

IN1 1111 0000

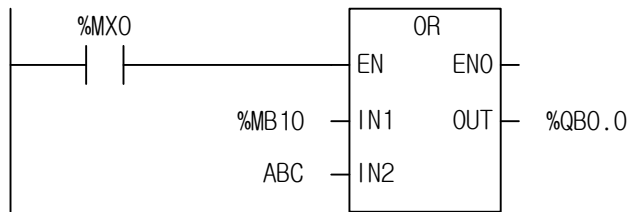
OR

IN2 1010 1010

OUT 1111 1010

■ 프로그램 예

1. LD

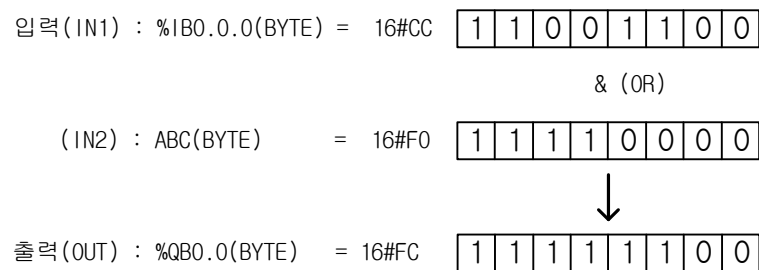


2. ST

```
%QB0.0 := OR2_BYTE(EN:=%MX0, IN1:=%MB10, IN2:=ABC);
```

(1) 실행조건(%MX0)이 On 하면 OR 평선이 실행됩니다.

(2) %MB10 = 2#1100_1100 을 ABC = 2#1111_0000 와 OR 시킨 결과가 %QB0.0 = 2#1111_1100 로 출력됩니다.



REAL_TO_***		적용 기종										발생플래그								
REAL 타입 변환		XGF-M32E										_ERR, _LER								
평 선		설 명																		
		<p>입력 EN : 1 일 때 평선 실행 IN : 타입 변환할 REAL 값</p> <p>출력 ENO : 에러 없이 실행되면 1 을 출력 OUT : 타입 변환된 데이터</p>																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	OUT				○		○	○	○	○	○	○	○	○	○					

■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

평선	출력 타입	동작 설명
REAL_TO_SINT	SINT	입력의 정수 부분이 -128 ~ 127 일 경우 정상 변환되나, 그 외 값은 에러가 발생 합니다. (소수점 이하는 반올림)
REAL_TO_INT	INT	입력의 정수 부분이 -32768 ~ 32767 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
REAL_TO_DINT	DINT	입력의 정수 부분이 $-2^{31} \sim 2^{31}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
REAL_TO_LINT	LINT	입력의 정수 부분이 $-2^{63} \sim 2^{63}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
REAL_TO_USINT	USINT	입력의 정수 부분이 0 ~ 255 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
REAL_TO_UINT	UINT	입력의 정수 부분이 0 ~ 65,535 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
REAL_TO_UDINT	UDINT	입력의 정수 부분이 0 ~ $2^{32}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)
REAL_TO_ULINT	ULINT	입력의 정수 부분이 0 ~ $2^{64}-1$ 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다. (소수점 이하는 반올림)

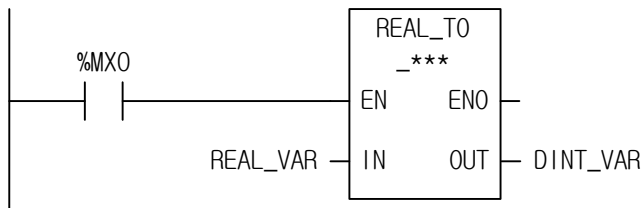
평선	출력 타입	동작 설명
REAL_TO_DWORD	DWORD	내부 비트 배열의 변화 없이 DWORD 타입으로 변환합니다.
REAL_TO_LREAL	LREAL	REAL 을 LREAL 타입으로 정상 변환합니다.

■ 플래그

플래그	설명
_ERR	입력값이 출력타입에 저장할 수 있는 값보다 커서 오버 플로(Overflow)가 발생하면 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 결과값에 0 을 출력합니다.

■ 프로그램 예

1. LD



2. ST

ST 언어는 REAL_TO_***은 지원하지 않습니다.

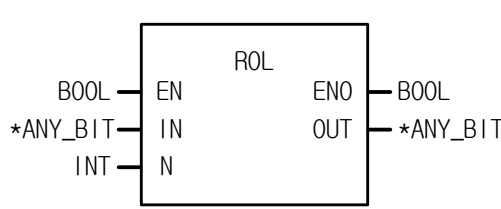
REAL_TO_DINT 인 경우

```
DINT_VAR := REAL_TO_DINT(EN:=%MX0, IN:=REAL_VAR);
```

(1) 실행조건(%MX0)이 On 하면 REAL_TO_*** 평선이 실행됩니다.

(2) REAL_VAL(REAL 타입) = 1.234E4 이면, DINT_VAL(DINT 타입) = 12,340 이 됩니다.

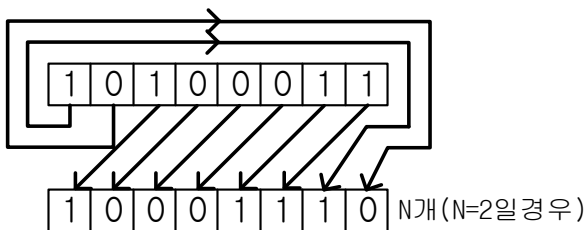
입력(IN) : REAL_VAL(REAL) = 1.234E4
 ↓ (REAL_TO_DINT)
 출력(OUT) : DINT_VAL(DINT) = 12,340

ROL		적용 기종	발생플래그																		
왼쪽으로 회전 (Rotate Left)		XGF-M32E	-																		
평 선		설 명																			
		입력 EN : 1일 때 평선 실행 IN : 회전될 값 N : 회전할 비트 수 출력 ENO : EN 값이 그대로 출력 OUT : 회전된 값																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	IN		○	○	○	○															
	OUT		○	○	○	○															

*ANY_BIT: ANY_BIT 중 BOOL 제외

■ 기능

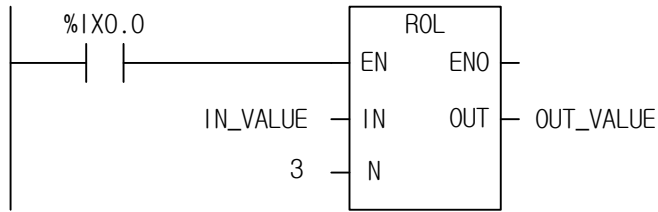
1. 입력 IN을 N 비트 수만큼 왼쪽으로 회전시킵니다.



■ 프로그램 예

입력 %IX0.0 이 On 하면 입력 데이터(2#1100_1100_1100_1100:16#CCCC)의 값을 좌로 3 비트 만큼 회전시키는 프로그램

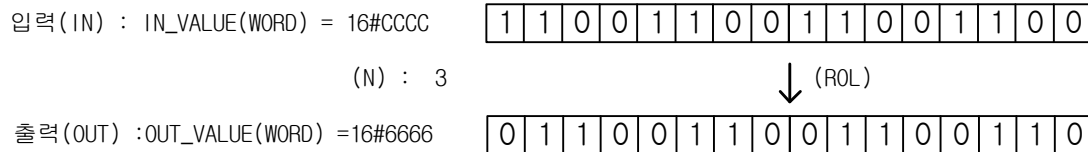
1. LD

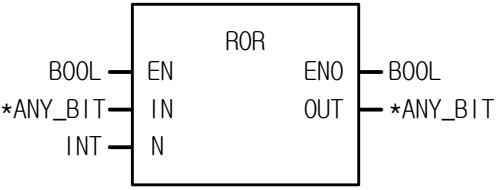


2. ST

```
OUT_VALUE := ROL(EN := %IX0.0, IN:=IN_VALUE, N:=3);
```

- (1) 회전할 데이터 값을 입력한 변수 IN_VALUE 로 설정한다.
- (2) 좌회전할 비트 수 3 을 회전할 비트 수 지정 입력(N)에 쓴다.
- (3) 회전된 데이터 값을 출력할 출력변수를 OUT_VALUE 로 설정한다.
- (4) 실행조건 %IX0.0 이 0n 하면 ROL(왼쪽으로 회전) 평선이 실행되어 입력변수로 설정된 데이터 비트를 좌로 3 비트 회전하여 출력변수로 선언된 OUT_VALVE 값에 출력된다.

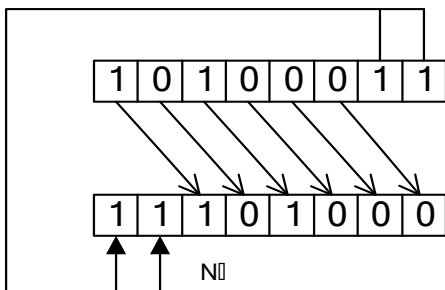


ROR		적용 기종	발생플래그																																																												
오른쪽으로 회전 (Rotate Right)		XGF-M32E	-																																																												
평 선		설 명																																																													
		입력 EN : 1 일 때 평선 실행 IN : 회전될 값 N : 회전할 비트 수 출력 ENO : EN 값이 그대로 출력 OUT : 회전된 값																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN		○	○	○	○															OUT		○	○	○	○																
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN		○	○	○	○																																																										
OUT		○	○	○	○																																																										

*ANY_BIT: ANY 타입 중 BOOL 제외

■ 기능

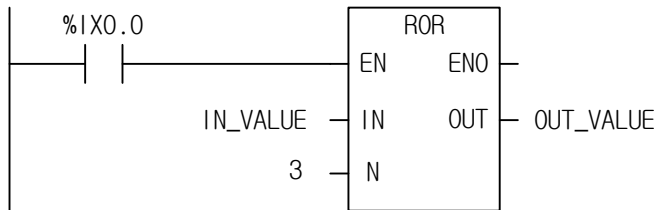
1. 입력 IN 을 N 비트 수만큼 오른쪽으로 회전시킵니다.



■ 프로그램 예

입력 %IX0.0 이 On 하면 입력 데이터 값(2#1110_0011_0011_0001: 16#E331)을 우로 3 비트 만큼 회전시키는 프로그램

1. LD



2. ST

```
OUT_VALUE := ROR(EN:=%IX0.0, IN:=IN_VALUE, N:=3);
```

- (1) 회전할 데이터 값을 입력한 변수를 IN_VALUE 로 설정한다.
- (2) 우회전할 비트 수 3 을 회전할 비트 수 지정 입력(N)에 설정한다.
- (3) 실행조건 %IX0.0 이 On 하면 ROR(오른쪽으로 회전) 평선이 실행되어 입력변수로 설정된 데이터 비트가 우로 3 비트 만큼 회전되어 출력변수로 선언된 OUT_VALUE 값에 출력된다.

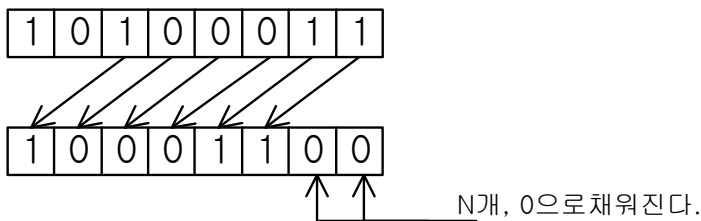
입력(IN): IN_VALUE(WORD) = 16#E331	1	1	1	0	0	0	1	1	0	0	1	1	0	0	0	1
(N) : 3	↓ (ROR)															
출력(OUT): OUT_VALUE(WORD) = 16#3C66	0	0	1	1	1	1	0	0	0	1	1	0	0	1	1	0

SHL		적용 기종	발생플래그																																																												
왼쪽으로 이동 (Shift Left)		XGF-M32E	-																																																												
평 선		설 명																																																													
<pre> graph LR subgraph SHL EN[EN] IN[IN] N[N] ENO[ENO] OUT[OUT] end EN --- ENO IN --- OUT N --- OUT </pre>		입력 EN : 1일 때 평선 실행 IN : 이동될 비트 열 N : 이동할 비트 수 출력 ENO : EN 값이 그대로 출력 OUT : 이동된 값																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN		○	○	○	○															OUT		○	○	○	○																
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN		○	○	○	○																																																										
OUT		○	○	○	○																																																										

*ANY_BIT: ANY_BIT 중 BOOL 제외

■ 기능

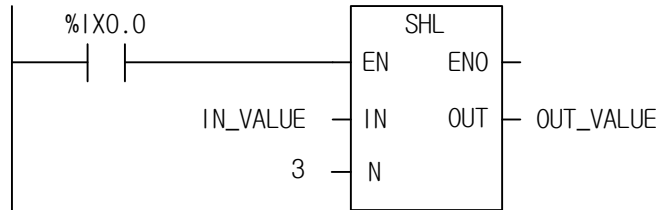
1. 입력 IN을 N 비트 수만큼 왼쪽으로 이동합니다.
2. 입력 IN의 맨 오른쪽에 있는 N개 비트는 0으로 채워집니다.



■ 프로그램 예

입력 %IX0.0 이 On 하면 입력 데이터 값(2#1100_1100_1100_1100: 16#CCCC)을 좌로 3 비트 만큼 이동시키는 프로그램

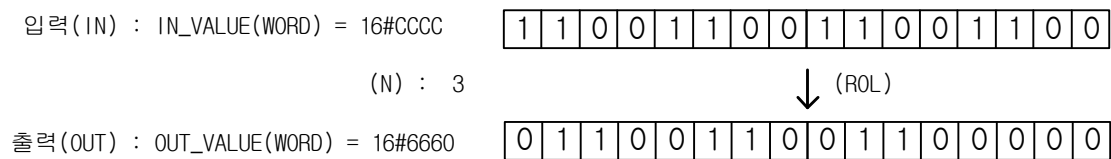
1. LD



2. ST

```
OUT_VALLUE := SHL(EN:=%IX0.0, IN:=IN_VALUE, N:=3);
```

- (1) 이동할 데이터 값을 입력할 변수를 IN_VALUE(2#1100_1100_1100_1100: 16#CCCC)로 설정한다.
- (2) 좌로 이동한 비트 수 3 을 지정 입력(N)에 쓴다. (변수 지정 후 쓰기로 가능)
- (3) 실행조건(%IX0.0.0)이 On 하면 SHL(왼쪽으로 이동) 평선이 실행되어 입력변수로 설정된 데이터 비트가 좌로 3 비트 이동하여, 출력변수로 선언된 OUT_VALUE 에 출력됩니다.

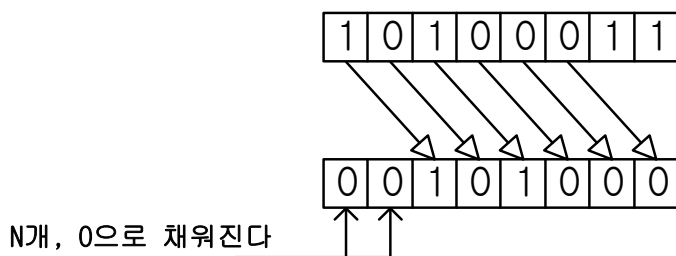


SHR		적용 기종	발생플래그																																																												
오른쪽으로 이동 (Shift Right)		XGF-M32E	-																																																												
평 선		설 명																																																													
<pre> graph LR EN[EN] --- SHR[SHR] IN[*ANY_BIT] --- SHR N[N] --- SHR SHR --- ENO[ENO] SHR --- OUT[*ANY_BIT] </pre>		<p>입력 EN : 1일 때 평선 실행</p> <p>IN : 이동될 비트 열</p> <p>N : 이동할 비트 수</p> <p>출력 ENO : EN 값이 그대로 출력</p> <p>OUT : 이동된 값</p>																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN		○	○	○	○															OUT		○	○	○	○																
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN		○	○	○	○																																																										
OUT		○	○	○	○																																																										

*ANY_BIT: ANY_BIT 중 BOOL 제외

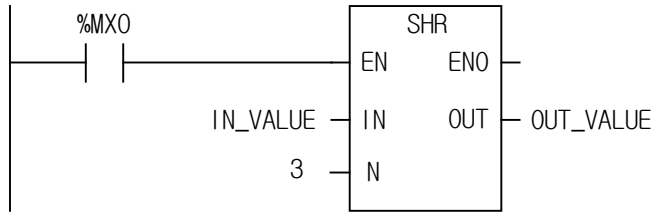
■ 기능

1. 입력 IN 을 N 비트 수만큼 오른쪽으로 이동합니다.
2. 입력 IN 의 맨 왼쪽에 있는 N 개 비트는 0으로 채워집니다.



■ 프로그램 예

1. LD



2. ST

```
OUT_VALUE := SHR(EN:=%MX0, IN:=IN_VALUE, N:=3);
```

- (1) 실행조건(%MX0)이 0n 하면 SHR(오른쪽으로 이동) 평선이 실행됩니다.
- (2) 입력변수로 설정된 데이터 비트가 우로 3 비트 이동하여, 출력변수로 선언된 OUT_VALUE 에 출력됩니다.

입력(IN) : IN_VALUE(WORD) = 16#E331	1	1	1	0	0	0	1	1	0	0	1	1	0	0	0	1
(N) : 3	↓ (ROR)															
출력(OUT) : OUT_VALUE(WORD) = 16#1C66	0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	0

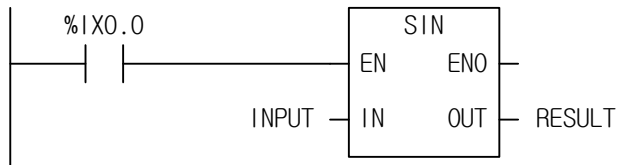
SIN		적용 기종	발생플래그																	
Sine 연산		XGF-M32E	-																	
평 선		설 명																		
		입력 EN : 1일 때 평선 실행 IN : Sine 연산의 각도 입력 값(Radian)																		
		출력 ENO : EN 값이 그대로 출력 OUT : Sine 연산결과 값 IN, OUT은 같은 데이터 타입이어야 함.																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	IN														○	○				
	OUT														○	○				

■ 기능

1. IN의 Sine 값을 구해 OUT으로 출력시킵니다.
 $OUT = SIN (IN)$

■ 프로그램 예

1. LD



2. ST

```
RESULT := SIN(EN:=IX0.0, IN:=INPUT);
```

(1) 실행조건(%IX0.0)이 On 하면 SIN(Sine 연산) 평선이 실행됩니다.

(2) INPUT 으로 선언된 입력변수의 값이 1.0471 ($\pi/3$ rad = 60°) 일 때 출력변수로 선언된 RESULT 는 0.8660

($\sqrt{3}/2$)이 출력됩니다. $\text{SIN}(\pi/3) = \sqrt{3}/2 = 0.8660$

입력 (IN) : INPUT (REAL) = 1.0471

↓ (SIN)

출력 (OUT) : RESULT (REAL) = 8.65976572E-01

SINT_TO_***		적용 기종	발생플래그																																								
SINT 타입 변환		XGF-M32E	_ERR, _LER																																								
평 선		설 명																																									
		입력 EN : 1 일 때 평선 실행 IN : 타입 변환할 Short Integer 값 출력 ENO : 에러 없이 실행되면 1 을 출력 OUT : 타입 변환된 데이터																																									
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>OUT</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	OUT	○	○	○	○	○		○	○	○	○	○	○	○	○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																								
OUT	○	○	○	○	○		○	○	○	○	○	○	○	○	○																												

*ANY: ANY 타입 중 SINT, TIME, DATE, TOD, DT 제외

■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

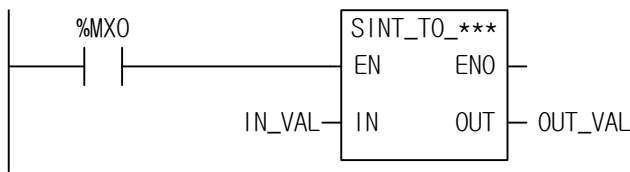
평 선	출력 타입	동작 설명
SINT_TO_INT	INT	INT 타입으로 정상 변환합니다.
SINT_TO_DINT	DINT	DINT 타입으로 정상 변환합니다.
SINT_TO_LINT	LINT	LINT 타입으로 정상 변환합니다.
SINT_TO_USINT	USINT	입력이 0 ~ 127 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
SINT_TO_UINT	UINT	입력이 0 ~ 127 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
SINT_TO_UDINT	UDINT	입력이 0 ~ 127 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
SINT_TO_ULINT	ULINT	입력이 0 ~ 127 일 경우 정상 변화되나, 그 외 값은 에러가 발생합니다.
SINT_TO_BOOL	BOOL	하위 1 비트를 취해 BOOL 타입으로 변환합니다.
SINT_TO_BYTE	BYTE	내부 비트 배열의 변화 없이 BYTE 타입으로 변환합니다.
SINT_TO_WORD	WORD	상위 비트들을 0 으로 채운 WORD 타입으로 변환합니다.
SINT_TO_DWORD	DWORD	상위 비트들을 0 으로 채운 DWORD 타입으로 변환합니다.
SINT_TO_LWORD	LWORD	상위 비트들을 0 으로 채운 LWORD 타입으로 변환합니다.
SINT_TO_REAL	REAL	SINT 를 REAL 타입으로 정상 변환합니다.
SINT_TO_LREAL	LREAL	SINT 를 LREAL 타입으로 정상 변환합니다.

■ 플래그

플래그	설명
_ERR	변환에러 발생시 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 출력 타입의 비트 수만큼 하위 비트를 취해 내부 비트 배열의 변환 없이 출력 시킵니다.

■ 프로그램 예

1. LD



2. ST

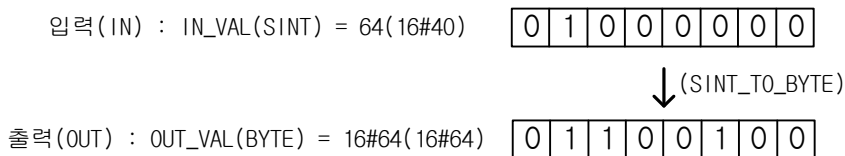
ST 언어는 SINT_TO_***는 지원하지 않습니다.

SINT_TO_BYTE 인 경우

OUT_VAL := SINT_TO_BYTE(EN:=%MX0, IN:=IN_VAL);

(1) 입력조건(%MX0)이 On 하면 SINT_TO_*** 펄스가 실행됩니다.

(2) 입력변수로 선언된 IN_VAL(SINT 타입)=64(2#0100_0000)이면, OUT_VAL(BYTE 타입)=16#40(2#0100_0000)가 됩니다.



SQRT		적용 기종	발생플래그																																																												
제공근 연산		XGF-M32E	_ERR, _LER																																																												
평 선		설 명																																																													
		<p>입력 EN : 1일 때 평선 실행 IN : 제공근 연산의 입력 값</p> <p>출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 제공근 값</p> <p>IN, OUT은 같은 데이터 타입이어야 함.</p>																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN														○	○					OUT														○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN														○	○																																																
OUT														○	○																																																

■ 기능

1. IN의 제공근 값을 구해 OUT으로 출력시킵니다.

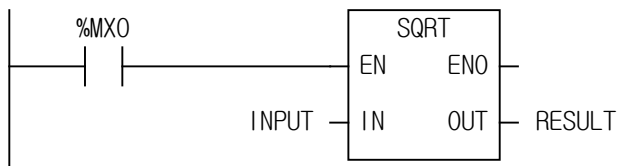
$$OUT = \sqrt{IN}$$

■ 플래그

플래그	설명
_ERR	IN의 값이 음수일 때 _ERR, _LER 플래그가 셋(Set)됩니다.

■ 프로그램 예

1. LD



2. ST

```
RESULT := SQRT(EN:=%MX0, IN:=INPUT);
```

- (1) 실행조건(%MX0)이 On 하면 SQRT(제곱근 연산) 평선이 실행됩니다.
- (2) INPUT 으로 선언된 입력변수의 값이 9.0 일 때 출력변수로 선언된 RESULT 는 3.0 이 출력됩니다.

$$\sqrt{9.0} = 3.0$$

입력(IN) : INPUT(REAL) = 9.0

↓ (SQRT)

출력(OUT) : RESULT(REAL) = 3.0

SUB		적용 기종	발생플래그																				
빼기		XGF-M32E	_ERR, _LER																				
평 선		설 명																					
		<p>입력 EN : 1일 때 평선 실행 IN1 : 피감수 IN2 : 감수</p> <p>출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 뺀 결과 값</p> <p>IN1, IN2, OUT에 연결되는 변수는 모두 같은 데이터 타입이어야 함.</p>																					
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT			
	IN1						○	○	○	○	○	○	○	○	○	○							
	IN2						○	○	○	○	○	○	○	○	○	○							
	OUT						○	○	○	○	○	○	○	○	○	○							

■ 기능

1. IN1에서 IN2를 빼서 OUT으로 출력시킵니다.
 $OUT = IN1 - IN2$

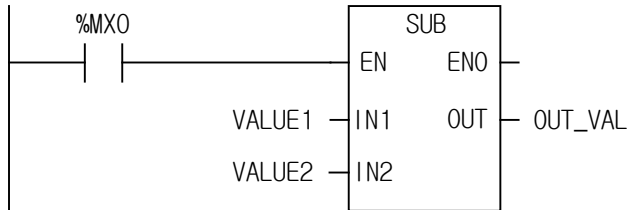
■ 플래그

플래그	설명
_ERR	출력 값이 해당 데이터 타입의 범위를 벗어날 경우 _ERR, _LER 플래그가 셋(Set)됩니다.

☆ REAL, LREAL 타입 연산에서는 IN1에서 IN8로 순차적으로 연산을 하기 때문에 중간 연산과정에서 이미 REAL, LREAL의 최대값이나 최소값을 넘게 되면 _ERR, _LER 플래그가 셋(Set)되고 결과값은 무한대 값 또는 비정상적인 값 (1.#INF000000000000e+000, 1.#SNAN000000000000e+000, 1.#QNAN000000000000e+000)으로 표시됩니다.

■ 프로그램 예

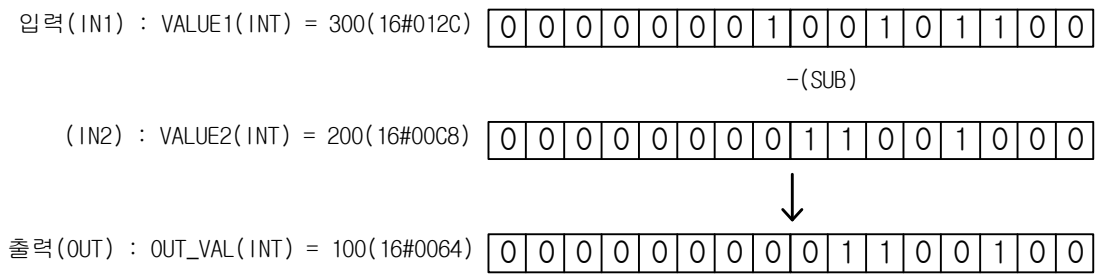
1. LD



2. ST

```
OUT_VAL := SUB(EN:=%MX0, IN1:=VALUE1, IN2:=VALUE2);
```

- (1) 실행조건(%MX0)이 0n 하면 SUB(빼기) 평선이 실행됩니다.
- (2) 입력변수 값이 VALUE1 = 300, VALUE2 = 200 이면, 출력변수로 설정한 OUT_VAL 는 연산을 실시한 결과 (300-200=100)가 출력됩니다.



SUB_DATE	적용 기종	발생플래그
날짜 빼기	XGF-M32E	_ERR, _LER
평 선	설 명	
	<p>입력 EN : 1일 때 평선 실행 IN1 : 기준 날짜 IN2 : 뺄 날짜</p> <p>출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 두 날짜간의 차이를 시간으로 출력</p>	

■ 기능

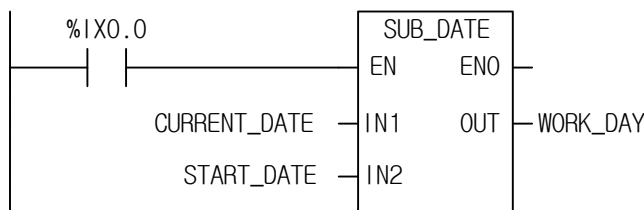
1. IN1(기준 날짜)에서 IN2(특정 날짜)를 빼서 날짜 차이를 OUT 으로 출력시킵니다.

■ 플래그

플래그	설명
_ERR	출력 값이 TIME 데이터 타입의 범위를 벗어날 경우, _ERR, _LER 플래그가 셋(Set)됩니다. 날짜 차이가 TIME 데이터 타입의 범위 T#49D17H2M47S295MS 를 넘거나 날짜 연산의 결과가 음수가 되면 에러가 됩니다.

■ 프로그램 예

1. LD



SUB_DT	적용 기종	발생플래그
시간과 날짜 빼기	XGF-M32E	_ERR, _LER
평 선	설 명	
	입력 EN : 1일 때 평선 실행 IN1 : 기준 날짜와 시각 IN2 : 뺄 날짜와 시각 출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 뺄 결과 시간	

■ 기능

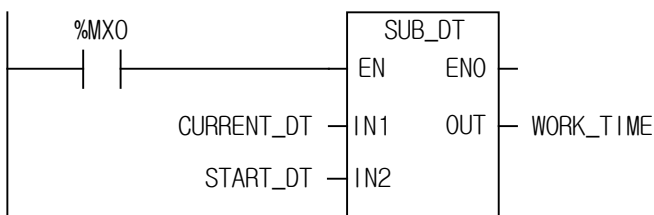
1. IN1(기준 날짜와 시각)에서 IN2(특정 날짜와 시각)를 빼서 시간 차이를 OUT 으로 출력시킵니다.

■ 플래그

플래그	설명
_ERR	출력 값이 TIME 데이터 타입의 범위를 벗어날 경우, _ERR, _LER 플래그가 셋(Set)됩니다. 날짜와 시각 빼기 연산의 결과가 음수가 되면 에러가 됩니다.

■ 프로그램 예

1. LD

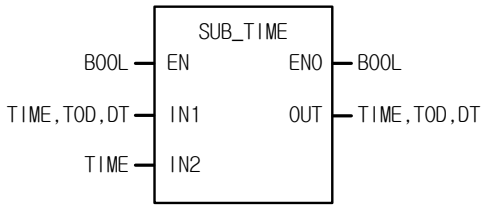


2. ST

```
WORK_TIME := SUB_DT(EN:=%MX0, IN1:=CURRNET_DT, IN2:=START_DT);
```

- (1) 실행조건(%MX0)이 On 하면 SUB_DT(시간과 날짜빼기) 평선이 실행됩니다.
 (2) 입력변수로 선언한 현재의 날짜와 시각 CURRENT_DT 가 DT#1995-12-15-14:30:00 이고 조업이 재개된 날짜와 시각 START_DT 가 DT#1995-12-13-12:00:00 이면, 출력변수로 선언된 연속 조업한 시간 WORK_TIME 에는 T#2D2H30M 가 출력됩니다.

```
입력 (IN1) : CURRENT_DT(DT) = DT#1995-12-15-14:30:00
                                     (SUB_DATE)
      (IN2) : START_DT(DT) = DT#1995-12-13-12:00:00
                                     ↓
출력 (OUT) : WORK_TIME(TIME) = T#2D2H30M
```

SUB_TIME		적용 기종	발생플래그																																																												
시간과 빼기		XGF-M32E	_ERR, _LER																																																												
평 선		설 명																																																													
		<p>입력 EN : 1 일 때 평선 실행 IN1 : 기준 시각 또는 시간 IN2 : 뺄 시간</p> <p>출력 ENO : 에러 없이 실행되면 1을 출력 OUT : 뺀 결과 시각 또는 시간</p> <p>OUT의 타입은 입력 IN1의 타입을 따름. 즉 IN1의 타입이 TIME이면, 출력 OUT의 타입도 TIME임.</p>																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td>○</td> <td>○</td> </tr> <tr> <td>OUT</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td></td> <td>○</td> <td>○</td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN1																○		○	○	OUT																○		○	○		
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN1																○		○	○																																												
OUT																○		○	○																																												

■ 기능

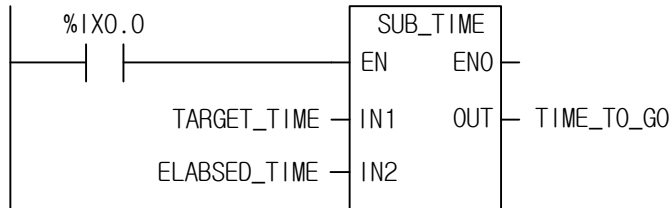
1. IN1이 TIME일 경우에는 시간에서 시간을 빼서 차이 시간을 출력시킵니다.
2. IN1이 TIME_OF_DAY일 경우에는 기준시각에서 시간을 빼서 하루 중의 시각을 출력시킵니다.
3. IN1이 DATE_AND_TIME일 경우에는 기준이 되는 날짜와 시각에서 시간을 빼서 날짜와 시각을 출력시킵니다.

■ 플래그

플래그	설명
_ERR	출력 값이 해당 데이터 타입의 범위를 벗어날 경우, _ERR, _LER 플래그가 셋(Set)됩니다. 시간에서 시간을 뺀 결과가 음수가 되거나 시각(TOD)에서 시간을 뺀 결과가 음수가 되면 에러가 됩니다.

■ 프로그램 예

1. LD



2. ST

```
TIME_TO_GO := SUB_TIME(EN:=%IX0.0, IN1:=TARGET_TIME, IN2:=ELAPSED_TIME);
```

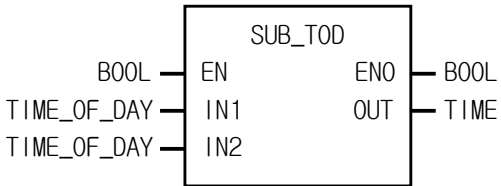
- (1) 실행조건(%IX0.0)이 On 하면 SUB_TIME(시간배기) 평선이 실행됩니다.
- (2) 입력변수로 선언된 전체 작업시간 TARGET_TIME 이 T#2H30M 이고, 경과된 시간 ELAPSED_TIME 이 T#1H10M30S300MS 면 출력변수로 선언된 남아 있는 작업시간 TIME_TO_GO 에는 T#1H19M29S700MS 가 출력됩니다.

입력 (IN1) : TARGET_TIME(TIME) = T#2H30M
(SUB_DATE)

(IN2) : ELAPSED_TIME(TIME)= T#1H10M30S300MS



출력 (OUT) : TIME_TO_GO(TIME) = T#1H19M29S700MS

SUB_TOD	적용 기종	발생플래그
시각에서 시각빼기	XGF-M32E	_ERR, _LER
평 선	설 명	
	<p>입력 EN : 1일 때 평선 실행 IN1 : 기준 시각 IN2 : 뺄 시각</p> <p>출력 ENO : 에러 없이 실행되면 1 출력 OUT : 뺀 결과 시간</p>	

■ 기능

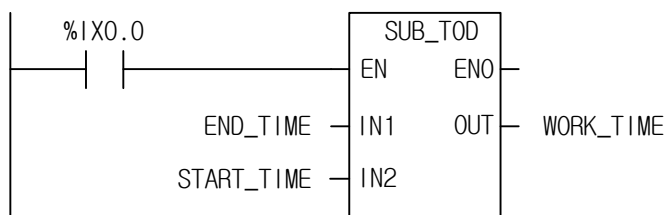
1. IN1(기준 시각)에서 IN2(특정 시각)를 빼서 시간 차이를 OUT 으로 출력시킵니다.

■ 플래그

플래그	설명
_ERR	시각에서 시각을 뺀 결과가 음수가 되면 에러가 됩니다.

■ 프로그램 예

1. LD



2. ST

```
WORK_TIME := SUB_TOD(EN:=%IX0.0, IN1:=END_TIME, IN2:=START_TIME);
```

- (1) 실행조건(%IX0.0)이 0n 하면 SUB_TOD(시각에서 시각빼기) 평선이 실행됩니다.
 (2) 입력변수로 선언된 END_TIME 이 TOD#14:20:30.500 이고 작업을 시작한 START_TIME 이 TOD#12:00:00 이면, 출력변수로 선언된 작업에 소요된 시간 WORK_TIME 에는 T#2H20M30S500MS 가 출력됩니다.

입력(IN1) : END_TIME(TOD) = TOD#14:20:30.500
 (SUB_TOD)

(IN2) : START_TIME(TOD) = TOD#12:00:00



출력(OUT) : WORK_TIME(TIME) = T#2H20M30S500MS

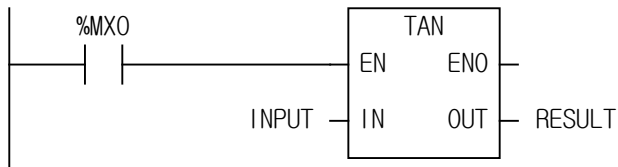
TAN		적용 기종	발생플래그																																																												
Tangent 연산		XGF-M32E	-																																																												
평 선		설 명																																																													
		<p>입력 EN : 1일 때 평선 실행 IN : Tangent 각도입력 값(Radian)</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : Tangent 연산결과 값</p> <p>IN, OUT 은 같은 데이터 타입이어야 함.</p>																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN														○	○					OUT														○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN														○	○																																																
OUT														○	○																																																

■ 기능

1. IN의 Tangent 값을 구해 OUT으로 출력시킵니다.
OUT = TAN(IN)

■ 프로그램 예

1. LD



2. ST

```
RESULT := TAN(EN:=%MX0, IN:=INPUT);
```

(1) 실행조건(%MX0)이 0n 하면 TAN(Tangent 연산) 평선이 실행합니다.

(2) INPUT 으로 선언된 입력변수의 값이 0.7853 ($\pi/4$ rad = 45°)일 때 출력 변수로 선언된 RESULT 는 1.0000 입니다.

$TAN(\pi/4) = 1$

입력(IN) : INPUT(REAL) = 0.7853

↓ (TAN)

출력(OUT) : RESULT(REAL) = 9.99803722E-01

TIME_TO_***		적용 기종										발생플래그										
TIME 타입변환		XGF-M32E										-										
평 선		설 명																				
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 시간 데이터 출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터																				
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		
	OUT				○								○									

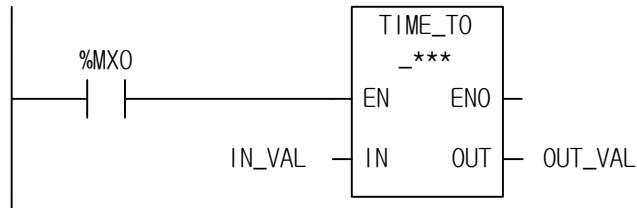
■ 기능

1. IN을 타입 변환해서 OUT으로 출력시킵니다.

평 선	출력 타입	동작 설명
TIME_TO_UDINT	UDINT	TIME 를 UDINT 타입으로 변환합니다. 데이터(내부 비트 배열 상태)의 변화 없이 데이터 타입만 변환합니다.
TIME_TO_DWORD	DWORD	TIME 를 DWORD 타입으로 변환합니다. 데이터(내부 비트 배열 상태)의 변화 없이 데이터 타입만 변환합니다.

■ 프로그램 예

1. LD



2. ST

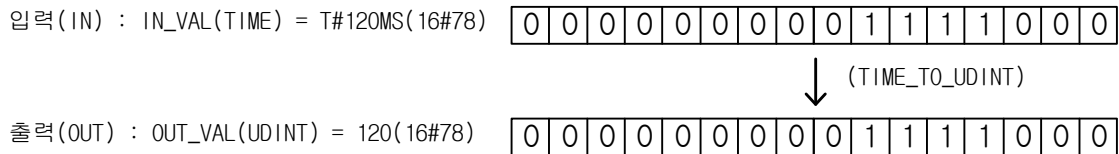
ST 언어는 TIME_TO_***는 지원하지 않습니다.

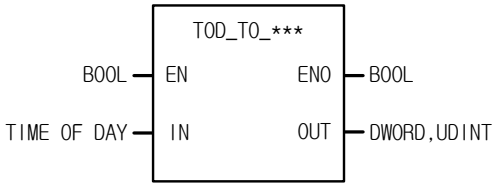
TIME_TO_UDINT 인 경우

```
OUT_VAL := TIME_TO_UDINT(EN:=%MX0, IN:=IN_VAL);
```

(1) 입력조건(%MX0)이 On 하면 TIME_TO_*** 평선이 실행됩니다.

(2) 입력변수로 선언된 IN_VAL(TIME 타입) = T#120MS 이면, 출력변수로 선언된 OUT_VAL(UDINT 타입) = 120 이 됩니다.



TOD_TO_***		적용 기종										발생플래그										
TOD 타입변환		XGF-M32E										-										
평 선		설 명																				
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 하루 중 시각 데이터 출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터																				
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		
	OUT				○								○									

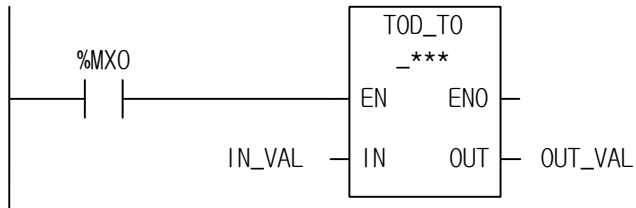
■ 기능

1. IN을 타입 변환해서 OUT으로 출력시킵니다.

평 선	출력 타입	동작 설명
TOD_TO_UDINT	UDINT	TOD를 UDINT 타입으로 변환합니다. 데이터(내부 비트 배열 상태)의 변화 없이 데이터 타입만 변환합니다.
TOD_TO_DWORD	DWORD	TOD를 DWORD 타입으로 변환합니다. 데이터(내부 비트 배열 상태)의 변화 없이 데이터 타입만 변환합니다.

■ 프로그램 예

1. LD



2. ST

ST 언어는 TIME_TO_***는 지원하지 않습니다.

TIME_TO_UDINT 인 경우

```
OUT_VAL := TOD_TO_ (EN:=%MX0, IN:=IN_VAL);
```

(1) 입력조건(%MX0)이 On 하면 TOD_TO_*** 평선이 실행됩니다.

(2) 입력변수로 선언된 IN_VAL(TOD 타입) = TOD#12:00:00 이면, 출력변수로 선언된 OUT_VAL(타입) = 'TOD#12:00:00' 이 됩니다.

입력(IN) : IN_VAL(TOD) = TOD#12:00:00
 ↓ (TOD_TO_STRING)

출력(OUT) : OUT_VAL(STRING) = 'TOD#12:00:00'

UDINT_TO_***		적용 기종										발생플래그								
UDINT 타입 변환		XGF-M32E										_ERR, _LER								
평 선		설 명																		
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 Unsigned Double Integer값 출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	OUT	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

*ANY: ANY 타입 중 UDINT, DATE, DT 제외

■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

평 선	출력 타입	동작 설명
UDINT_TO_SINT	SINT	입력이 0~127 일 경우 정상 변환되나 그 외 값은 에러가 발생합니다.
UDINT_TO_INT	INT	입력이 0~32,767 일 경우 정상 변환되나 그 외 값은 에러가 발생합니다.
UDINT_TO_DINT	DINT	입력이 0~2,147,483,647 일 경우 정상 변환되나 그 외 값은 에러가 발생합니다.
UDINT_TO_LINT	LINT	UDINT 를 LINT 타입으로 정상 변환합니다.
UDINT_TO_USINT	USINT	입력이 0~255 일 경우 정상 변환되나 그 외 값은 에러가 발생합니다.
UDINT_TO_UINT	UINT	입력이 0~65,535 일 경우 정상 변환되나 그 외 값은 에러가 발생합니다.
UDINT_TO_ULINT	ULINT	UDINT 를 ULINT 타입으로 정상 변환합니다.
UDINT_TO_BOOL	BOOL	하위 1 비트를 취해 BOOL 타입으로 변환합니다.
UDINT_TO_BYTE	BYTE	하위 8 비트를 취해 BYTE 타입으로 변환합니다.
UDINT_TO_WORD	WORD	하위 16 비트를 취해 WORD 타입으로 변환됩니다.
UDINT_TO_DWORD	DWORD	내부 비트 배열의 변환 없이 DWORD 타입으로 변환합니다.
UDINT_TO_LWORD	LWORD	상위 비트를 0으로 채운 LWORD 타입으로 변환합니다.
UDINT_TO_REAL	REAL	UDINT 를 REAL 타입으로 변환합니다. 변환 중 정밀도에 따른 오차가 발생할 수 있습니다.

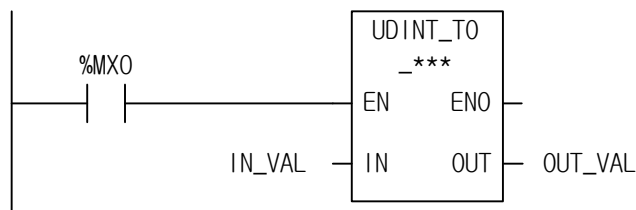
평 선	출력 타입	동작 설명
UDINT_TO_LREAL	LREAL	UDINT 를 LREAL 타입으로 변환합니다. 변환 중 정밀도에 따른 오차가 발생할 수 있습니다.
UDINT_TO_TOD	TOD	내부 비트 배열의 변환 없이 TOD 타입으로 변환합니다. 단, TOD범위 (TOD#23:59:59.999)를 벗어난 값 입력시 에러를 발생하고 TOD 범위 내에서 순환하여 변환합니다.
UDINT_TO_TIME	TIME	내부 비트 배열의 변환 없이 TIME 타입으로 변환합니다.

■ 플래그

플래그	설명
_ERR	변환에러 발생시 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 출력 타입의 비트 수만큼 하위 비트를 취해 내부 비트 배열의 변환 없이 출력 시킵니다.

■ 프로그램 예

1. LD



2. ST

ST 언어는 UDINT_TO_***는 지원하지 않습니다.

UDINT_TO_TIME 인 경우

```
OUT_VAL := UDINT_TO_TIME(EN:=%MX0, IN:=IN_VAL);
```

(1) 입력조건(%MX0)이 On 하면 UDINT_TO_*** 평선이 실행됩니다.

(2) 입력변수로 선언된 IN_VAL(UDINT 타입) = 123 이면, 출력변수로 선언된 OUT_VAL(TIME 타입) = T#123MS 가 됩니다.

입력(IN) : IN_VAL(UDINT) = 123



출력(OUT) : OUT_VAL(TIME) = T#123MS

UINT_TO_***		적용 기종										발생플래그								
UINT 타입 변환		XGF-M32E										_ERR, _LER								
평 선		설 명																		
		<p>입력 EN : 1일 때 평선 실행 IN : 타입 변환할 Unsigned Integer 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터</p>																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	OUT	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

*ANY: ANY 타입 중 UINT, TIME, TOD, DT 제외

■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

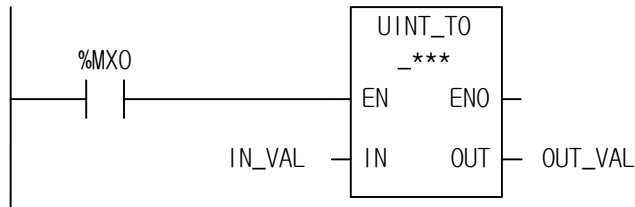
평 선	출력타입	동작 설명
UINT_TO_SINT	SINT	입력이 0~127 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
UINT_TO_INT	INT	입력이 0~32,767 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
UINT_TO_DINT	DINT	UINT 를 UDINT 타입으로 정상 변환합니다.
UINT_TO_LINT	LINT	UINT 를 ULINT 타입으로 정상 변환합니다.
UINT_TO_USINT	USINT	입력이 0~255 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
UINT_TO_UDINT	UDINT	UINT 를 UDINT 타입으로 정상 변환합니다.
UINT_TO_ULINT	ULINT	UINT 를 ULINT 타입으로 정상 변환합니다.
UINT_TO_BOOL	BOOL	하위 1 비트를 취해 BOOL 타입으로 변환합니다.
UINT_TO_BYTE	BYTE	하위 8 비트를 취해 BYTE 타입으로 변환합니다.
UINT_TO_WORD	WORD	내부 비트 배열의 변환 없이 WORD 타입으로 변환합니다.
UINT_TO_DWORD	DWORD	상위 비트를 0으로 채운 DWORD 타입으로 변환합니다.
UINT_TO_LWORD	LWORD	상위 비트를 0으로 채운 LWORD 타입으로 변환합니다.
UINT_TO_REAL	REAL	UINT 를 REAL 타입으로 변환합니다.
UINT_TO_LREAL	LREAL	UINT 를 LREAL 타입으로 변환합니다.
UINT_TO_DATE	DATE	내부 비트 배열의 변환 없이 DATE 타입으로 변환합니다.

■ 플래그

플래그	설명
_ERR	변환에러 발생시 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 출력 타입의 비트 수만큼 하위 비트를 취해 내부 비트 배열의 변환 없이 출력시킵니다.

■ 프로그램 예

1. LD



2. ST

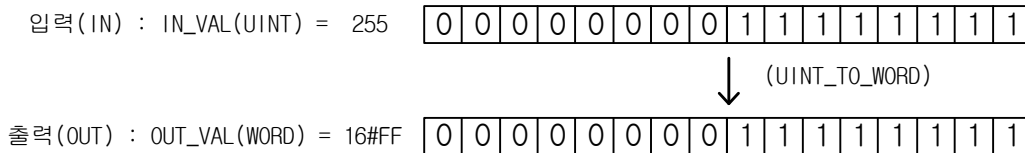
ST 언어는 UINT_TO_***는 지원하지 않습니다.

UINT_TO_WORD 인 경우

OUT_VAL := UINT_TO_WORD(EN:=%MX0, IN:=IN_VAL);

(1) 입력조건(%MX0)이 On 하면 UINT_TO_*** 평선이 실행됩니다.

(2) 입력변수로 선언된 IN_VAL(UINT 타입) = 255(2#0000_0000_1111_1111)이면, 출력변수로 선언된 OUT_VAL(WORD 타입) = 2#0000_0000_1111_1111 이 됩니다.



ULINT_TO_***		적용 기종										발생플래그									
ULINT 타입 변환		XGF-M32E										_EPR, _LER									
평 선		설 명																			
		<p>입력 EN : 1 일 때 평선 실행 IN : 타입 변환할 Unsigned Long Integer 값</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터</p>																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	OUT	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	

*ANY: ANY 타입 중 UINT, TIME, TOD, DT 제외

■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

평 선	출력 타입	동작 설명
ULINT_TO_SINT	SINT	입력이 0~127 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
ULINT_TO_INT	INT	입력이 0~32,767 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
ULINT_TO_DINT	DINT	입력이 0~2 ³¹ -1 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
ULINT_TO_LINT	LINT	입력이 0~2 ⁶³ -1 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
ULINT_TO_USINT	USINT	입력이 0~255 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
ULINT_TO_UINT	UINT	입력이 0~65,535 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
ULINT_TO_UDINT	UDINT	입력이 0~2 ³² -1 일 경우 정상 변환되나, 그 외 값은 에러가 발생합니다.
ULINT_TO_BOOL	BOOL	하위 1 비트를 취해 BOOL 타입으로 변환합니다.
ULINT_TO_BYTE	BYTE	하위 8 비트를 취해 BYTE 타입으로 변환합니다.
ULINT_TO_WORD	WORD	하위 16 비트를 취해 WORD 타입으로 변환합니다.
ULINT_TO_DWORD	DWORD	하위 32 비트를 취해 DWORD 타입으로 변환합니다.
ULINT_TO_LWORD	LWORD	내부 비트 배열의 변환 없이 LWORD 타입으로 변환합니다.
ULINT_TO_REAL	REAL	ULINT 를 REAL 타입으로 변환합니다. 변환 중 정밀도에 따른 오차가 발생할 수 있습니다.

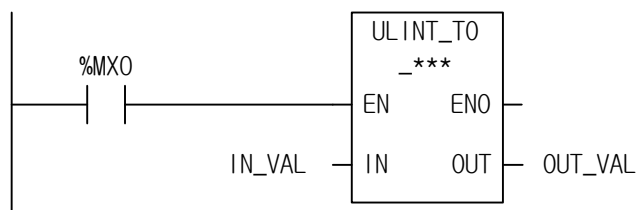
평 선	출력 타입	동작 설명
ULINT_TO_LREAL	LREAL	ULINT 를 LREAL 타입으로 변환합니다. 변환 중 정밀도에 따른 오차가 발생할 수 있습니다.

■ 플래그

플래그	설명
_ERR	변환에러 발생시 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 출력 타입의 비트 수만큼 하위 비트를 취해 내부 비트 배열의 변환 없이 출력 시킵니다.

■ 프로그램 예

1. LD



2. ST

ST 언어는 ULINT_TO_***는 지원하지 않습니다.

ULINT_TO_LINT 인 경우

OUT_VAL := ULINT_TO_LINT(EN:=%MX0, IN:=IN_VAL);

(1) 입력조건(%MX0)이 On 하면 ULINT_TO_*** 평선이 실행됩니다.

(2) 입력변수로 선언된 IN_VAL(ULINT 타입) = 123,567,899 이면, 출력변수로 선언된 OUT_VAL(LINT 타입) = 123,567,899 가 됩니다.

입력(IN) : IN_VAL(ULINT) = 123,567,899

↓ (ULINT_TO_LINT)

출력(OUT) : OUT_VAL(LINT) = 123,567,899

USINT_TO_***		적용 기종	발생플래그
USINT 타입 변환		XGF-M32E	_ERR, _LER
평 선		설 명	
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 Unsigned Short Integer 값 출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터	
ANY 타입 변수설명	변수명 BOOL BYTE WORD DWORD LWORD SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL TIME DATE TOD DT		
	OUT ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○		

*ANY : ANY 타입 중 USINT, TIME, DATE, TOD, DT 제외

■ 기능

1. IN 을 타입 변환해서 OUT 으로 출력시킵니다.

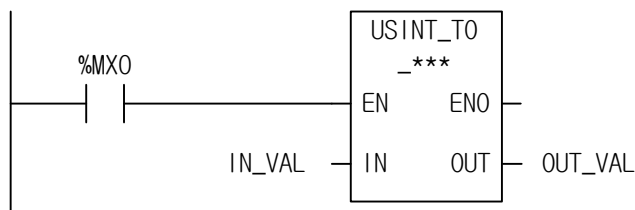
평 선	출력 타입	동작 설명
USINT_TO_SINT	SINT	입력이 0~127 일 경우 정상 변환되나 그 외 값은 에러가 발생합니다.
USINT_TO_INT	INT	입력을 INT 타입으로 정상 변환합니다.
USINT_TO_DINT	DINT	입력을 DINT 타입으로 정상 변환합니다.
USINT_TO_LINT	LINT	입력을 LINT 타입으로 정상 변환합니다.
USINT_TO_UINT	UINT	입력을 UINT 타입으로 정상 변환합니다.
USINT_TO_UDINT	UDINT	입력을 UDINT 타입으로 정상 변환합니다.
USINT_TO_ULINT	ULINT	입력을 ULINT 타입으로 정상 변환합니다.
USINT_TO_BOOL	BOOL	하위 1비트를 취해 BOOL 타입으로 변환합니다.
USINT_TO_BYTE	BYTE	내부 비트 배열의 변환 없이 BYTE 타입으로 변환합니다.
USINT_TO_WORD	WORD	상위 비트를 0으로 채운 WORD 타입으로 변환합니다.
USINT_TO_DWORD	DWORD	상위 비트를 0으로 채운 DWORD 타입으로 변환합니다.
USINT_TO_LWORD	LWORD	상위 비트를 0으로 채운 LWORD 타입으로 변환합니다.
USINT_TO_REAL	REAL	USINT 를 REAL 타입으로 변환합니다.
USINT_TO_LREAL	LREAL	USINT 를 LREAL 타입으로 변환합니다.

■ 플래그

플래그	설명
_ERR	변환에러 발생시 _ERR, _LER 플래그가 셋(Set)됩니다. 에러 발생시 출력 타입의 비트 수만큼 하위 비트를 취해 내부 비트 배열의 변환 없이 출력 시킵니다.

■ 프로그램 예

1. LD



2. ST

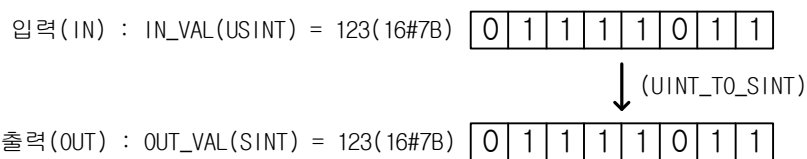
ST 언어는 USINT_TO_***는 지원하지 않습니다.

USINT_TO_SINT 인 경우

OUT_VAL := USINT_TO_SINT(EN:=%MX0, IN:=IN_VAL);

(1) 입력조건(%MX0)이 On 하면 USINT_TO_*** 평선이 실행됩니다.

(2) 입력변수로 선언된 IN_VAL(USINT 타입) = 123 이면, 출력변수로 선언된 OUT_VAL(SINT 타입) = 123 이 됩니다.



WORD_TO_***		적용 기종	발생플래그																	
WORD 타입변환		XGF-M32E	-																	
평 선		설 명																		
		입력 EN : 1일 때 평선 실행 IN : 타입 변환할 비트 열(16 비트) 출력 ENO : EN 값이 그대로 출력 OUT : 타입 변환된 데이터																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	OUT	○	○		○	○	○	○	○	○	○	○	○	○				○		

*ANY: ANY 타입 중 WORD, REAL, LREAL, TIME, TOD, DT 제외

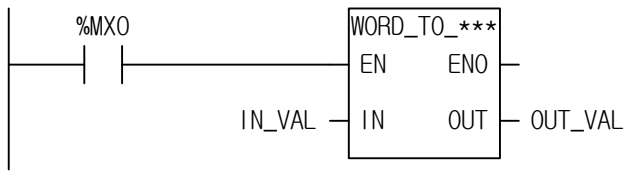
■ 기능

1. IN을 타입 변환해서 OUT으로 출력시킵니다.

평 선	출력 타입	동작 설명
WORD_TO_SINT	SINT	하위 8비트를 취해 SINT 타입으로 변환합니다.
WORD_TO_INT	INT	내부 비트 배열의 변환 없이 INT 타입으로 변환합니다.
WORD_TO_DINT	DINT	상위 비트를 0으로 채운 DINT 타입으로 변환합니다.
WORD_TO_LINT	LINT	상위 비트를 0으로 채운 LINT 타입으로 변환합니다.
WORD_TO_USINT	USINT	하위 8비트를 취해 SINT 타입으로 변환합니다.
WORD_TO_UINT	UINT	내부 비트 배열의 변환 없이 INT 타입으로 변환합니다.
WORD_TO_UDINT	UDINT	상위 비트를 0으로 채운 DINT 타입으로 변환합니다.
WORD_TO_ULINT	ULINT	상위 비트를 0으로 채운 LINT 타입으로 변환합니다.
WORD_TO_BOOL	BOOL	하위 1비트를 취해 BOOL 타입으로 변환합니다.
WORD_TO_BYTE	BYTE	하위 8비트를 취해 SINT 타입으로 변환합니다.
WORD_TO_DWORD	DWORD	상위 비트를 0으로 채운 DWORD 타입으로 변환합니다.
WORD_TO_LWORD	LWORD	상위 비트를 0으로 채운 LWORD 타입으로 변환합니다.
WORD_TO_DATE	DATE	내부 비트 배열의 변환 없이 DATE 타입으로 변환합니다.

■ 프로그램 예

1. LD



2. ST

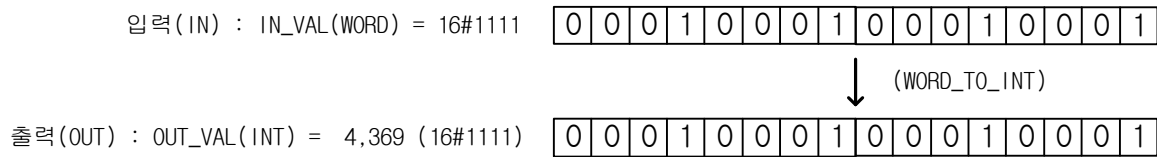
ST 언어는 WORD_TO_***는 지원하지 않습니다.

WORD_TO_INT 인 경우

```
OUT_VAL := WORD_TO_INT(EN:=%MX0, IN:=IN_VAL);
```

(1) 입력조건(%MX0)이 On 하면 WORD_TO_*** 평선이 실행됩니다.

(2) 입력변수로 선언된 IN_VAL(WORD 타입) = 2#0001_0001_0001_0001 이면, 출력변수로 선언된 OUT_VAL(INT 타입) = 4,096 + 256 + 16 + 1 = 4,369 가 됩니다.



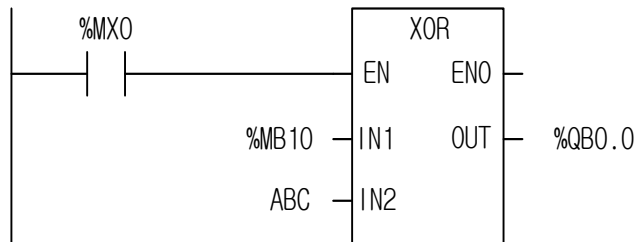
XOR		적용 기종	발생플래그																		
배타적 논리합		XGF-M32E	-																		
평션		설명																			
<pre> graph LR subgraph XOR EN[EN] IN1[IN1] IN2[IN2] ENO[ENO] OUT[OUT] end EN --- ENO IN1 --- OUT IN2 --- OUT style EN fill:none,stroke:none style ENO fill:none,stroke:none style IN1 fill:none,stroke:none style IN2 fill:none,stroke:none style OUT fill:none,stroke:none </pre>		<p>입력 EN : 1일 때 평션 실행 IN1 : XOR 될 값 IN2 : XOR 될 값 입력 8 개까지 확장 가능</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : XOR 된 값</p> <p>IN1, IN2, OUT 은 모두 같은 타입이어야 함.</p>																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	IN	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>															
	OUT	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>															

■ 기능

- IN1 을 IN2 와 비트별로 XOR 해서 OUT 으로 출력시킵니다.
 IN1 1111 0000
 XOR
 IN2 1010 1010
 OUT 0101 1010

■ 프로그램 예

1. LD



2. ST

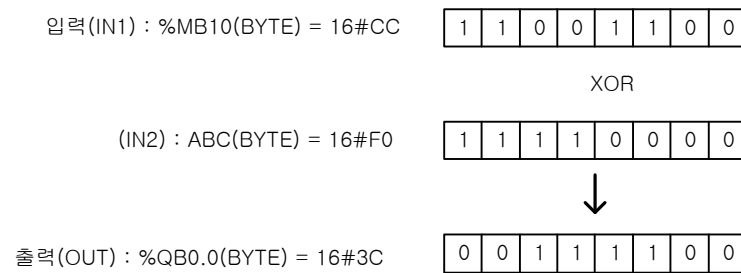
ST 언어는 XOR 는 지원하지 않습니다.

XOR2_BYTE 인 경우

```
%QB0.0 := XOR2_BYTE(EN:=%MX0, IN1:=%MB10, IN2:=ABC);
```

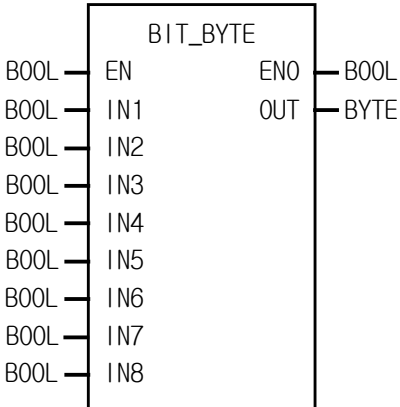
(1) 실행조건(%MX0)이 On 하면 XOR(배타적 논리합) 평선이 실행됩니다.

(2) 입력변수 %MB10 = 1100_1100, ABC = 1111_0000 이면, 두 값을 XOR 시킨 결과가 출력변수 %QB0.0 = 0011_1100 로 출력됩니다.



제7장 응용 평선

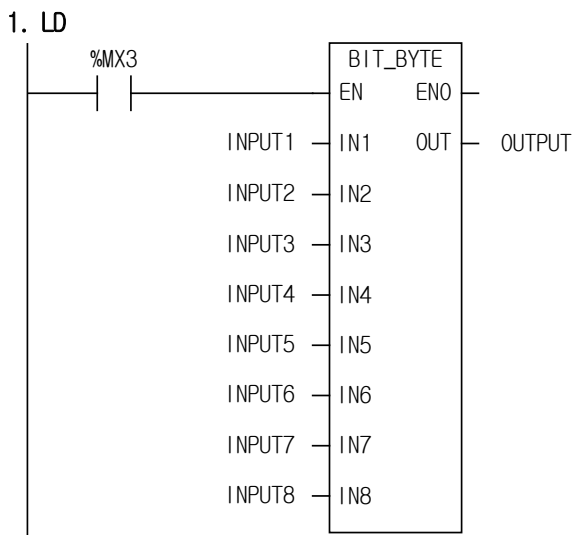
6 장에서 설명한 기본 평선과는 다른 응용 평선에 대하여 설명합니다.

BIT_BYTE	적용 기종	발생플래그
8 개 BIT 들을 BYTE 로 묶음	XGF-M32E	_ERR, _LER
평 선		설 명
		<p>입력 EN : 1 일 때 평선 실행 IN1 ~ IN8 : Bit 입력</p> <p>출력 ENO : EN 값을 그대로 출력 OUT : Byte 출력</p>

■ 기능

1. 8 개의 비트를 하나의 바이트로 조합합니다.
IN8: 최상의 비트(MSB), IN1: 최하의 비트(LSB)

■ 프로그램 예



2. ST

```
OUTPUT := BIT_BYTE(EN:=%MX3, IN1:=INPUT1, IN2:=INPUT2, IN3:=INPUT3, IN4:=INPUT4, IN5:=INPUT5, IN6:=INPUT6,  
IN7:=INPUT7, IN8:=INPUT8);
```

- (1) 실행조건(%MX3)이 On 되면 BIT_BYTE 평선이 실행됩니다.
- (2) 8개의 입력이 IN1~ IN8까지 {0,1,1,0,1,1,0,0}이면 출력변수로 선언된 OUTPUT(BYTE타입) = 2#0110_1100 입니다.

BSUM		적용 기종										발생플래그										
0n 된 비트 개수를 숫자로 출력		XGF-M32E										-										
평 선		설 명																				
		입력 EN : 1 일 때 평선 실행 IN : 0n 비트를 검색할 입력 데이터 출력 ENO : EN 값이 그대로 출력 OUT : 0n 된 비트 개수를 합한 결과 데이터																				
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT		
	IN		○	○	○	○																

*ANY_BIT: ANY_BIT 타입 중 BOOL 제외

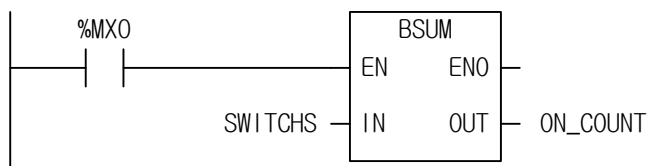
■ 기능

1. EN 이 1 이면, IN 의 비트 스트링 데이터 중, 1 로 되어있는 비트의 숫자를 세어서 OUT 으로 출력합니다.
2. 입력에는 BYTE, WORD, DWORD, LWORD 타입의 데이터가 접속 가능합니다.

FUNCTION	IN 변수 타입	동작 설명
BSUM	BYTE	입력 데이터에 맞추어 4 가지 타입 중 하나를 사용합니다.
BSUM	WORD	
BSUM	DWORD	
BSUM	LWORD	

■ 프로그램 예

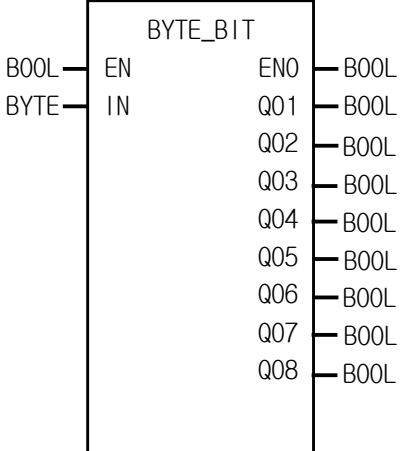
1. LD



2. ST

```
ON_COUNT := BSUM(EN:=%MX0, IN:=SWITCHS);
```

- (1) 실행조건(%MX0)이 On 되면 BSUM 평션이 실행됩니다.
- (2) 입력변수로 선언된 SWITCHS(WORD 타입) = 2#0000_0100_0010_1000 이라면, 출력변수(On_COUNT)는 On 되어 있는 비트의 개수를 출력합니다. 즉, '3' 을 출력하여 ON_COUNT(INT 타입)에 정수값 '3' 이 저장됩니다.

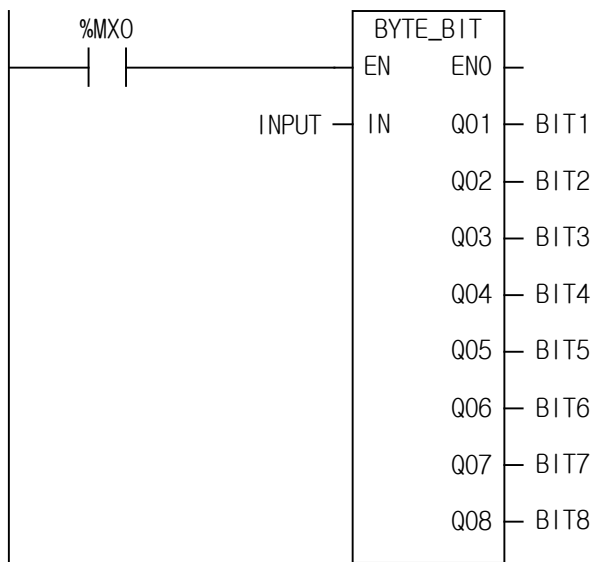
BYTE_BIT	적용 기종	발생플래그
BYTE 를 8 개 BIT 들로 나눔	XGF-M32E	-
평 선		설 명
		<p>입력 EN : 1 일 때 평선 실행 IN : Byte 입력</p> <p>출력 ENO : EN 값을 그대로 출력 Q01~8 : Bit 출력</p>

■ 기능

1. 1 개의 바이트를 출력 변수로 선언된 8 개의 비트(Q01~Q02)로 분산합니다.
2. Q08: 최상위 비트(MSB), Q01: 최하위 비트(LSB)

■ 프로그램 예

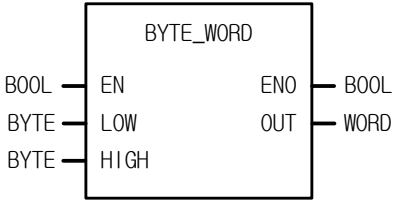
1. LD



2. ST

```
BYTE_BIT(EN:=%MX0, IN:= INPUT, Q01=> BIT1, Q02=> BIT2, Q03=> BIT3, Q04=> BIT4, Q05=> BIT5, Q06=> BIT6, Q07=> BIT7,  
Q08=> BIT8);
```

- (1) 실행조건(%MX0)가 On되면 BYTE_BIT 평션이 실행 됩니다.
- (2) 입력으로 선언된 INPUT = 16#AC = 2#1010_1100 이면 출력 변수로 선언된 Q01 ~ Q08까지 Q01부터 순서대로 2#{0,0,1,1,0,1,0,1}이 저장됩니다.

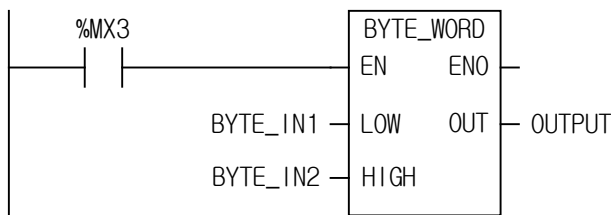
BYTE_WORD	적용 기종	발생플래그
2 개의 BYTE 를 WORD 로 모음	XGF-M32E	-
평 선	설 명	
	입력 EN : 1 일 때 평 선 실행 LOW : 하위 BYTE 입력 HIGH : 상위 BYTE 입력	출력 ENO : EN 값을 그대로 출력 OUT : WORD 출력

■ 기능

- 2 개의 바이트를 하나의 워드로 조합합니다.
 LOW: 하위 바이트 입력, HIGH: 상위 바이트 입력

■ 프로그램 예

1. LD



2. ST

```
OUTPUT := BYTE_WORD(EN:=%MX3, LOW:=BYTE_IN1, HIGH:=BYTE_IN2);
```

- 실행조건(%MX3)이 On 되면 BYTE_WORD 평선이 실행됩니다.
- 입력변수로 선언된 BYTE_IN1 = 16#56 이고 BYTE_IN2 = 16#AD 이면 출력변수로 선언된 OUTPUT = 16#AD56 입니다.

DEC		적용 기종	발생플래그																																																												
데이터를 하나 감소		XGF-M32E	-																																																												
평 선		설 명																																																													
<pre> graph LR subgraph DEC EN[EN] --- ENO[ENO] IN[IN] --- OUT[OUT] end ENO --- ENO_OUT[ENO] OUT --- OUT_ANY[*ANY_BIT] style EN fill:none,stroke:none style ENO fill:none,stroke:none style IN fill:none,stroke:none style OUT fill:none,stroke:none style ENO_OUT fill:none,stroke:none style OUT_ANY fill:none,stroke:none </pre>		입력 EN : 1 일 때 평선 실행 IN : 감소시킬 입력 데이터 출력 ENO : EN 값이 그대로 출력 OUT : 감소시킨 결과 데이터																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN		○	○	○	○															OUT		○	○	○	○																
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN		○	○	○	○																																																										
OUT		○	○	○	○																																																										

*ANY_BIT: ANY_BIT 타입 중 BOOL 제외

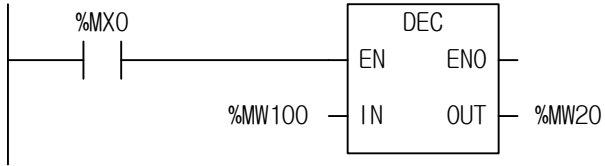
■ 기능

1. EN 이 1 이면, IN 의 비트 스트링 데이터를 1 만큼 감소시켜서 OUT 으로 출력합니다.
2. 언더플로어가 발생해도 에러는 발생하지 않으며, 결과는 16#0000 인 경우에 16#FFFF 가 됩니다.
3. 입력에는 BYTE, WORD, DWORD, LWORD 타입의 데이터가 접속 가능합니다.

FUNCTION	IN/OUT 변수 타입	동작 설명
DEC	BYTE	입출력 데이터에 맞추어 4 가지 타입 중 하나를 사용합니다.
DEC	WORD	
DEC	DWORD	
DEC	LWORD	

■ 프로그램 예

1. LD



2. ST

```
%MW20 := DEC(EN:=%MX0, IN:=%MW100);
```

- (1) 실행조건(%MX0)이 0n 되면 DEC 평선이 실행됩니다.
- (2) 입력 변수로 선언된 %MW100 = 16#0007(2#0000_0000_0000_0111) 이라면, 연산이 실행된 후에는 %MW20 = 16#0006(2#0000_0000_0000_0110)이 됩니다.

DECO		적용 기종																발생플래그				
지정된 비트 위치를 On		XGF-M32E																-				
평 선		설 명																				
		입력 EN : 1일 때 평선 실행 IN : Decoding 할 입력 데이터 출력 ENO : 에러 없이 실행되면 1을 출력 OUT : Decoding 한 결과 데이터																				
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		
	IN		○	○	○	○																
	OUT		○	○	○	○																

*ANY_BIT: ANY_BIT 타입 중 BOOL 제외

■ 기능

1. EN 이 1 이면, IN 의 값 즉 비트 위치지정 데이터에 따라서 출력의 비트 스트링 데이터 중 지정된 위치의 비트만 1로 하여 출력합니다.
2. 출력에는 BYTE, WORD, DWORD, LWORD 타입의 데이터가 접속 가능합니다.

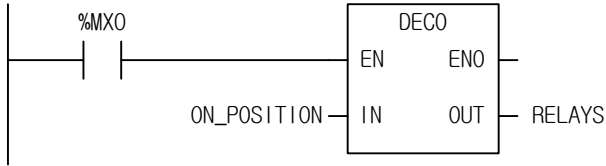
FUNCTION	OUT 변수 타입	동작 설명
DECO	BYTE	출력 데이터에 맞추어 4 가지 타입 중 하나를 사용합니다.
DECO	WORD	
DECO	DWORD	
DECO	LWORD	

■ 플래그

플래그	설명
_ERR	입력데이터가 음수이거나, 비트 위치지정 데이터가 출력타입의 비트한계를 넘으면(DECO 의 WORD 타입인 경우 16 이상), OUT 은 0 이 되고 _ERR, _LER 플래그가 셋(SET)됩니다.

■ 프로그램 예

1. LD



2. ST

```
RELAYS := DECO(EN:=%MX0, IN:=ON_POSITION);
```

- (1) 실행조건(%MX0)이 0n 되면 DECO 평선이 실행됩니다.
- (2) 입력변수로 선언된 ON_POSITION(INT 타입) = 5 라면, 출력의 5 번 비트만 0n 되므로, RELAYS(WORD 타입) = 2#0000_0000_0010_0000 이 됩니다.

DEG		적용 기종										발생플래그									
Radian 값을 각도로 변환		XGF-M32E										-									
평 선		설 명																			
		입력 EN : 1 일 때 평선 실행 IN : 라디안(Radian)값 입력 출력 ENO : EN 값을 그대로 출력 OUT : 각도 출력																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	IN														○	○					
	OUT														○	○					

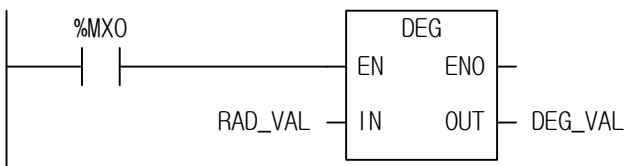
■ 기능

1. 라디안(Radian) 값을 입력 받아서 각도(Degree)로 출력 합니다.

평선	입력 타입	출력 타입	동작 설명
DEG	REAL	REAL	라디안(Radian)값을 출력 데이터 타입에 해당하는 각도 값으로 변환 합니다.
DEG	LREAL	LREAL	

■ 프로그램 예

1. LD



2. ST

```
DEG_VAL := DEG(EN:=%MX0, IN:=RAD_VAL);
```

- (1) 실행조건(%MX0)이 On 되면 DEG 평선이 실행됩니다.
- (2) 평선의 입력변수로 선언된 RAD_VAL = 1.0일 경우 평선의 출력변수 DEG_VAL = 5.7295779513078550E+001가 됩니다.

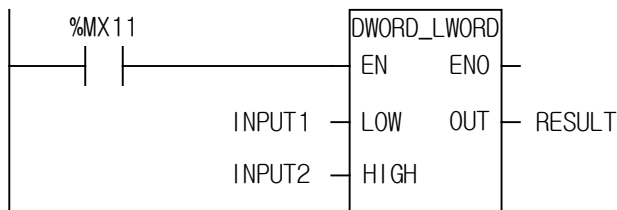
DWORD_LWORD	적용 기종	발생플래그
2 개의 DWORD 를 LWORD 로 모음	XGF-M32E	-
평 선	설 명	
	<p>입력 EN : 1일 때 평선 실행 LOW : 하위 DWORD 입력 HIGH : 상위 DWORD 입력</p> <p>출력 ENO : EN 값을 그대로 출력 OUT : LWORD 출력</p>	

■ 기능

- 2 개의 DWORD 를 하나의 LWORD 로 조합합니다.
 LOW: 하위 더블워드 입력, HIGH: 상위 더블워드 입력

■ 프로그램 예

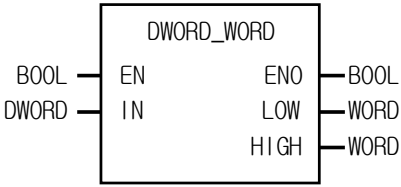
1. LD



2. ST

```
RESULT := DWORD_LWORD(EN:=%MX11, LOW:=INPUT1, HIGH:=INPUT2);
```

- 실행조건(%MX11)이 On 되면 DWORD_LWORD 평선이 실행됩니다.
- 입력변수로 선언된 INPUT1=16#1A2A_3A4A이고, INPUT2=16#8C7C_6C5C 일 때 출력변수로 선언된 RESULT = 16#8C7C_6C5C_1A2A_3A4A가 출력됩니다.

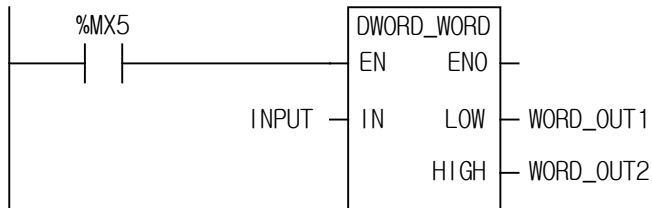
DWORD_WORD	적용 기종	발생플래그
DWORD 를 2 개의 WORD 로 나눔	XGF-M32E	-
평 선	설 명	
	입력 EN : 1일 때 평선 실행 IN : DWORD 입력 출력 ENO : EN 값을 그대로 출력 LOW : 하위 WORD 출력 HIGH : 상위 WORD 출력	

■ 기능

1. 하나의 DWORD 를 2 개의 WORD 로 분산합니다.
 LOW: 하위 워드 출력, HIGH: 상위 워드 출력

■ 프로그램 예

1. LD



2. ST

```
DWORD_WORD(EN:=%MX5, IN:=INPUT, LOW=>WORD_OUT1, HIGH=>WORD_OUT2);
```

- (1) 실행조건(%MX5)이 On 되면 DWORD_WORD 평선이 실행됩니다.
- (2) 입력변수로 선언된 INPUT=16#1122_AABB 일 때 입출력 변수로 선언된 WORD_OUT1 = 16#AABB, WORD_OUT2 = 16#11220이 저장됩니다.

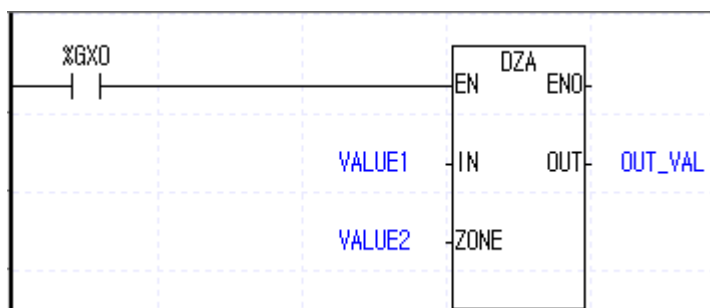
DZA		적용 기종	발생플래그																																																																																
불감대 A		XGF-M32E	-																																																																																
평 선		설 명																																																																																	
		<p>입력 EN : 1 일 때 평선 실행 IN : 불감대 연산을 수행할 입력 값 ZONE : 불감대 영역</p> <p>출력 ENO : 에러 없이 실행되면 1 을 출력 OUT : 불감대 연산 출력 값</p> <p>* Out = Input (Input >= Zone) * Out = 0 (Input < Zone) * IN, OUT 은 모두 같은 타입이어야 함</p>																																																																																	
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>ZONE</td> <td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> <tr> <td>OUT</td> <td></td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN						○	○	○	○	○	○	○	○	○	○					ZONE						○	○	○	○	○	○	○	○	○	○					OUT						○	○	○	○	○	○	○	○	○	○						
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																																																
IN						○	○	○	○	○	○	○	○	○	○																																																																				
ZONE						○	○	○	○	○	○	○	○	○	○																																																																				
OUT						○	○	○	○	○	○	○	○	○	○																																																																				

■ 기능

1. IN의 절대값을 ZONE과 비교하여 Zone의 절대값보다 작은 경우 0을 OUT으로 출력하고 Zone의 절대값보다 큰 경우 IN값을 OUT으로 출력 시킵니다

■ 프로그램 예

1. LD



2. ST

```
OUT_VAL := DZA(EN:=%GX0, IN:=VALUE1, Zone:VALUE2);
```

- (1) 실행조건(%GX0)이 On 하면 불감대 평선 DZA가 실행 됩니다.
- (2) 입력변수 IN=10 Zone=5 이면 출력 변수로 OUT_VAL = 10 이 출력 됩니다.

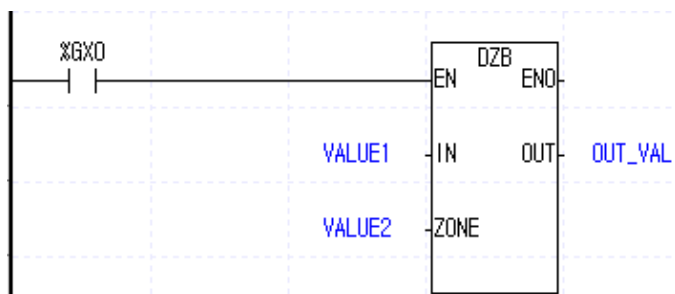
DZB		적용 기종													발생플래그					
불감대 B		XGF-M32E													-					
평 선		설 명																		
		<p>입력 EN : 1 일 때 평선 실행 IN : 불감대 연산을 수행할 입력 값 ZONE : 불감대 영역</p> <p>출력 ENO : 에러 없이 실행되면 1 을 출력 OUT : 불감대 연산 출력 값</p> <p>* Out = Input - Zone (Input >= Zone , Input < 0) * Out = Input + Zone (Input >= Zone , Input >= 0) * Out = 0 (Input < Zone) * IN, OUT 은 모두 같은 타입이어야 함</p>																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	IN						○	○	○	○	○	○	○	○	○	○				
	ZONE						○	○	○	○	○	○	○	○	○	○				
	OUT						○	○	○	○	○	○	○	○	○	○				

■ 기능

1. IN의 절대값을 Zone과 비교하여 ZONE의 절대값보다 작은 경우 0을 OUT으로 출력하고 ZONE의 절대값보다 큰 경우 IN값과 ZONE값의 합을 OUT으로 출력 시킵니다.

■ 프로그램 예

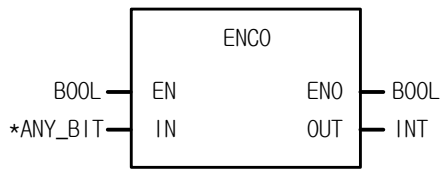
1. LD



2. ST

```
OUT_VAL := DZB(EN:=%GX0, IN:=VALUE1, Zone:VALUE2);
```

- (1) 실행조건(%GX0)이 On 하면 불감대 평선 DZB가 실행 됩니다.
- (2) 입력변수 IN=10, Zone=5 이면 출력 변수로 OUT_VAL = 15 이 출력 됩니다.

ENCO		적용 기종										발생플래그										
On 된 비트 위치를 숫자로 출력		XGF-M32E										_ERR, _LER										
평 선		설 명																				
		입력 EN : 1 일 때 평선 실행 IN : Encoding 할 입력 데이터 출력 ENO : 에러 없이 실행되면 1 을 출력 OUT : Encoding 한 결과 데이터																				
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT		
	IN		○	○	○	○																
	OUT							○														

*ANY_BIT: ANY_BIT 타입 중 BOOL 제외

■ 기능

1. EN 이 1 이면, IN 의 비트 스트링 데이터 중, 1 로 되어있는 비트 중 최상위 비트의 위치를 OUT 으로 출력합니다.
2. 입력에는 B(BYTE), W(WORD), D(DWORD), L(LWORD) 타입의 데이터가 접속 가능합니다.

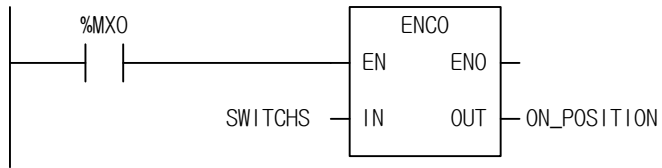
FUNCTION	IN 변수 타입	동작 설명
ENCO	BYTE	각 입력 변수 타입에 따라 원하는 ENCO 평선의 타입을 사용합니다.
ENCO	WORD	
ENCO	DWORD	
ENCO	LWORD	

■ 플래그

플래그	설명
_ERR	입력데이터 중 하나의 비트도 1 이 되어있지 않은 경우는 OUT 은 -1 이 되고, _ERR, _LER 플래그가 셋(Set)됩니다.

■ 프로그램 예

1. LD



2. ST

```
ON_POSITION := ENCO(EN:=%MX0, IN:=SWITCHS);
```

(1) 실행조건(%MX0)이 On 되면 ENCO 평선이 실행됩니다.

(2) SWITCHS(WORD 타입) = 2#0000_1000_0000_0010 이라면, On 되어 있는 2 비트의 위치, 즉 '11' 과 '1' 중 상위 위치인 '11' 을 출력하여 ON_POSITION(INT 타입)에 정수값 '11' 이 저장됩니다.

INC		적용 기종	발생플래그																																																												
데이터를 하나 증가		XGF-M32E	-																																																												
평 선		설 명																																																													
		입력 EN : 1 일 때 평선 실행 IN : 증가시킬 입력 데이터 출력 ENO : EN 값이 그대로 출력 OUT : 증가시킨 결과 데이터																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN		○	○	○	○															OUT		○	○	○	○																
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN		○	○	○	○																																																										
OUT		○	○	○	○																																																										

*ANY_BIT: ANY_BIT 타입 중 BOOL 제외

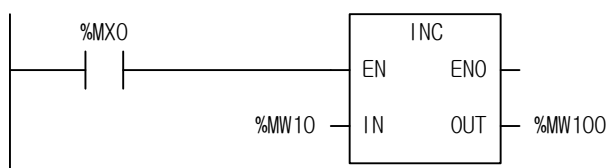
■ 기능

1. EN 이 1 이면, IN 의 비트스트링 데이터를 1 만큼 증가시켜서 OUT 으로 출력합니다.
2. 오버플로우가 발생해도 에러는 발생하지 않으며, 결과는 16#FFFF 인 경우에 16#0000 이 됩니다.
3. 입력에는 BYTE, WORD, DWORD, LWORD 타입의 데이터가 접속 가능합니다.

FUNCTION	IN/OUT 변수 타입	동작 설명
INC	BYTE	입출력 데이터에 맞추어 4 가지 평선 중 하나를 사용합니다.
INC	WORD	
INC	DWORD	
INC	LWORD	

■ 프로그램 예

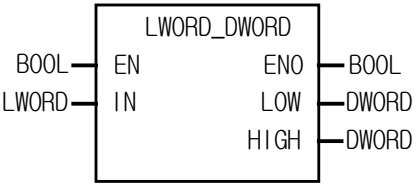
1. LD



2. ST

```
%MW100 := INC(EN:=%MX0, IN:=%MW10);
```

- (1) 실행조건(%MX0)이 0n 되면 INC 평선이 실행됩니다.
- (2) 입력변수 %MW10 = 16#0007(2#0000_0000_0000_0111) 이라면, 연산이 실행된 후에는 %MW100 = 16#0008(2#0000_0000_0000_1000)이 됩니다.

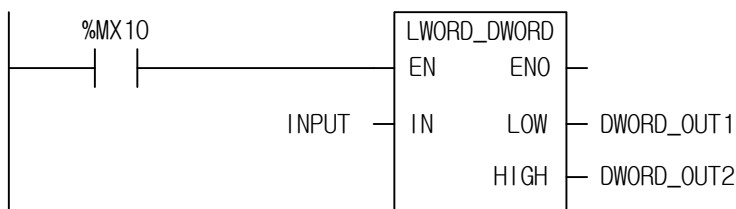
LWORD_DWORD	적용 기종	발생플래그
LWORD 를 2 개의 DWORD 로 나눔	XGF-M32E	-
평 선	설 명	
	<p>입력 EN : 1일 때 평선 실행 IN : LWORD 입력</p> <p>출력 ENO : EN 값을 그대로 출력 LOW : 하위 DWORD 출력 HIGH : 상위 DWORD 출력</p>	

■ 기능

- 하나의 LWORD 를 2 개의 DWORD 로 분산합니다.
LOW: 하위 더블워드 출력, HIGH: 상위 더블워드 출력

■ 프로그램 예

1. LD



2. ST

```
LWORD_DWORD(EN:=%MX10, IN:=INPUT, LOW=>DWORD_OUT1, HIGH=>DWORD_OUT2);
```

- 실행조건(%MX10)이 On 되면, LWORD_DWORD 평선이 실행됩니다.
- 입력변수로 선언된 INPUT=16#AAAA_BBBB_CCCC_DDDD 일 때, 출력 변수로 선언된 DWORD_OUT1=16#CCCC_DDDD, DWORD_OUT2=16#AAAA_BBBB 가 출력됩니다.

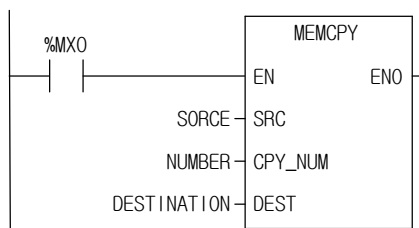
MEMCPY		적용 기종	발생플래그																																																												
메모리 복사		XGF-M32E	-																																																												
평 선		설 명																																																													
<pre> graph LR subgraph MEMCPY EN[EN] SRC[SRC] CPY_NUM[CPY_NUM] DEST[DEST] ENO[ENO] end EN --- MEMCPY SRC --- MEMCPY CPY_NUM --- MEMCPY DEST --- MEMCPY MEMCPY --- ENO </pre>		입력 EN : 1일 때 평선 실행 SRC : 복사할 값의 시작 영역 CPY_NUM : 복사할 값의 개수 DEST : 복사할 값을 쓸 시작 영역 출력 ENO : EN 값이 그대로 출력																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>SRC</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> </tr> <tr> <td>DEST</td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> <td><input type="radio"/></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	SRC	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	DEST	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
SRC	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>																																												
DEST	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>																																												

■ 기능

1. 이 명령은 해당 영역에 메모리 복사를 수행하도록 하는 명령입니다.
2. EN 에 ‘1’ 이 입력되면 SRC 에 설정된 메모리로부터 CPY_NUM 에 설정된 개수만큼 DEST 에 설정된 메모리 영역으로 복사합니다.

■ 프로그램 예

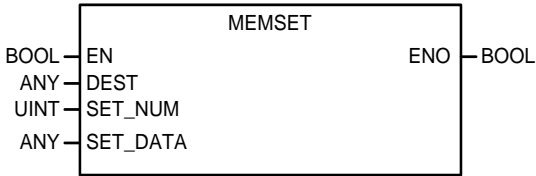
1. LD



2. ST

```
MEMCPY(SRC:=SOURCE, CPY_NUM:=NUM, DEST:=DESTINATION);
```

(1) SOURCE 로 지정된 메모리 영역으로부터 NUM 의 개수 만큼 연속된 메모리를 DESTINATION 으로 설정된 메모리 영역에 복사합니다.

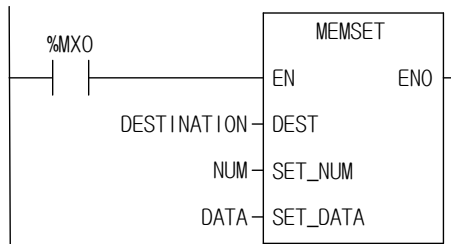
MEMSET		적용 기종	발생플래그
메모리 설정		XGF-M32E	-
평 선		설 명	
		입력 EN : 1일 때 평선 실행 DEST : 설정할 값을 쓸 시작 영역 CPY_NUM : 설정할 값의 개수 SET_DATA : 메모리에 설정할 값	출력 ENO : EN 값이 그대로 출력
ANY 타입 변수설명	변수명 BOOL BYTE WORD DWORD LWORD SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL TIME DATE TOD DT	SRC ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	DEST ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

■ 기능

1. 이 명령은 해당 영역에 메모리 설정을 수행하도록 하는 명령입니다.
2. EN 에 '1' 이 입력되면 DEST 에 설정된 메모리 영역에 CPY_NUM 에 설정된 개수만큼 SET_DATA 에 설정된 값을 씁니다.

■ 프로그램 예

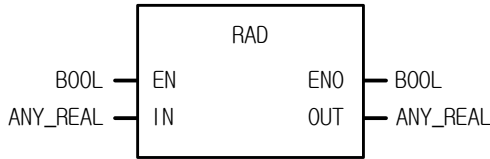
1. LD



2. ST

```
MEMSET(DEST:=DESTINATION, SET_NUM:=NUM, SET_DATA:=DATA);
```

(1) DESTINATION 에 설정된 메모리 영역에 NUM 에 설정된 개수 만큼 DATA 의 값을 씁니다.

RAD		적용 기종	발생플래그
각도(DEG)를 Radian 값으로 변환		XGF-M32E	-
평 선		설 명	
		입력 EN : 1일 때 평선 실행 IN : 각도 입력 출력 ENO : EN 값을 그대로 출력 OUT : 라디안 값 출력	
ANY 타입 변수설명	변수명 IN OUT	BOOL BYTE WORD DWORD LWORD SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL TIME DATE TOD DT	○ ○ ○ ○

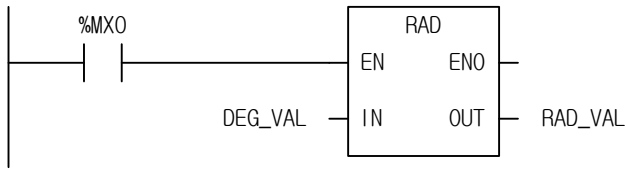
■ 기능

1. 각도의 단위를 도(°)에서 라디안(Radian) 값으로 출력 합니다.
2. 각도가 360°를 넘어서더라도 정상적으로 변환시켜 줍니다.
 (EX: 입력이 370°이면 출력은 360°를 뺀 10°에 해당하는 라디안 값을 출력합니다.)

평선	입력 타입	출력 타입	동작 설명
RAD	REAL	REAL	각도의 단위를 도(°)에서 출력 데이터 타입에 해당하는 라디안 값으로 변환합니다.
RAD	LREAL	LREAL	

■ 프로그램 예

1. LD



2. ST

```
RAD_VAL := RAD(EN:=%MX0, IN:= DEG_VAL);
```

- (1) 실행조건(%MX0)이 On 하면, RAD_REAL 평선이 실행됩니다.
- (2) 평선의 입력 변수로 선언된 DEG_VAL = 127(°)일 경우, 평선의 출력변수 값은 RAD_VAL = 2.21656823 이 됩니다.

SWAP		적용 기종	발생플래그																																																												
데이터의 상위 하위 바꾸기		XGF-M32E	-																																																												
평 선		설 명																																																													
		입력 EN : 1일 때 평선 실행 IN : 입력 출력 ENO : EN 값을 그대로 출력 OUT : Swap 된 값																																																													
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> </tr> </thead> <tbody> <tr> <td>IN</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>OUT</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	IN		○	○	○	○															OUT		○	○	○	○																
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT																																												
IN		○	○	○	○																																																										
OUT		○	○	○	○																																																										

*ANY_BIT: ANY_BIT 타입 중 BOOL 제외

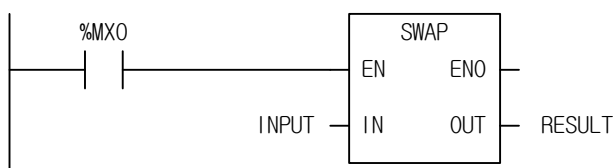
■ 기능

1. 입력된 변수를 2개의 크기로 구분하여 상위와 하위를 서로 교환합니다.

평선	입력 타입	동작 설명
SWAP	BYTE	BYTE의 상하위 니블(Nibble)을 서로 교환하여 출력합니다.
SWAP	WORD	WORD의 상하위 BYTE를 서로 교환하여 출력합니다.
SWAP	DWORD	DWORD의 상하위 WORD를 서로 교환하여 출력합니다.
SWAP	LWORD	LWORD의 상하위 DWORD를 서로 교환하여 출력합니다.

■ 프로그램 예

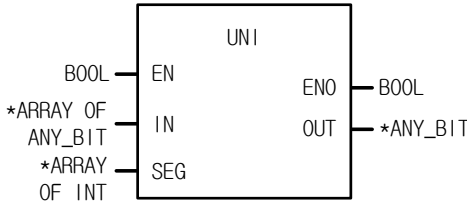
1. LD



2. ST

```
RESULT := SWAP(EN:=%MX0, IN:=INPUT);
```

- (1) 실행조건(%MX0)이 On 되면, SWAP 평선이 실행됩니다.
- (2) 평선의 입력 변수 INPUT(BYTE 타입) = 16#5F 일 경우, 평선의 출력 변수 RESULT(BYTE 타입) = 16#F5 가 됩니다.

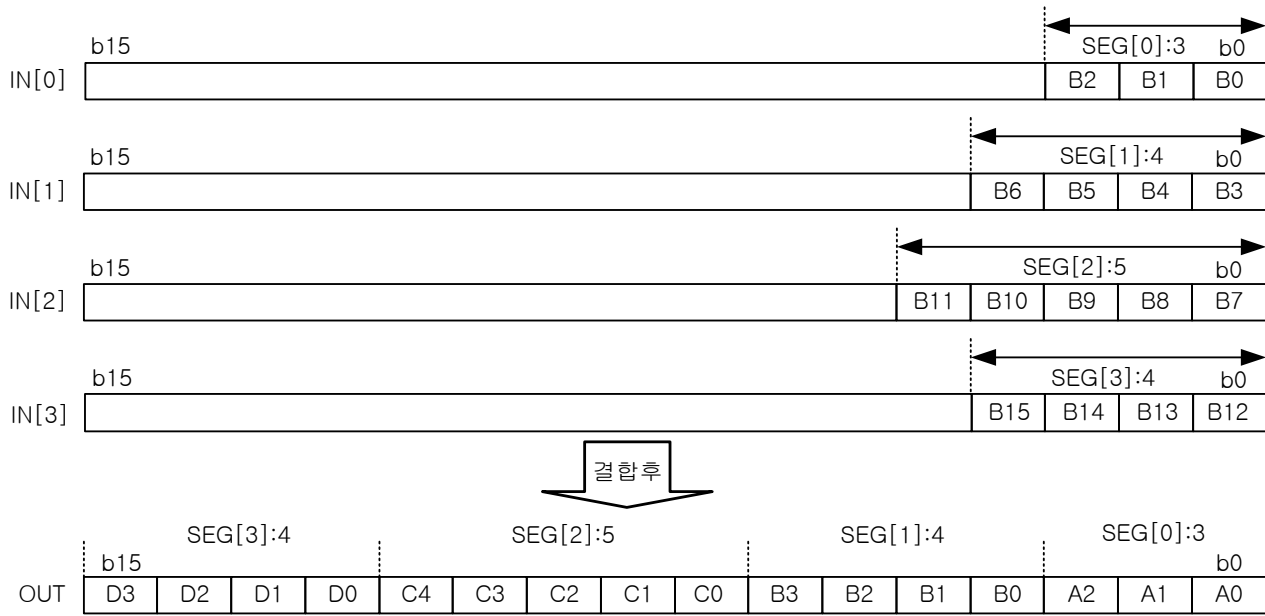
UNI		적용 기종	발생플래그																		
데이터 결합(union)		XGF-M32E	_ERR, _LER																		
평 선		설 명																			
		입력 EN : 1 일 때 평선 실행 IN : 입력 데이터 어레이 SEG : 데이터 결합 비트수 지정 어레이 출력 ENO : 에러 없이 실행되면 1 을 출력 OUT : 결합된 데이터 출력																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	IN		○	○	○	○															
	OUT		○	○	○	○															

*ANY_BIT: ANY_BIT 타입 중 BOOL 제외

■ 기능

1. 입력 데이터 어레이를 SEG 어레이에서 지정한 비트수 별로 하위 비트부터 지정된 비트 수만큼 결합하여 출력합니다.

평선	입력 타입	출력 타입	동작 설명
UNI	BYTE	BYTE	입력 어레이를 SEG가 지정하는 비트의 개수만큼씩 자른 후, 입력 어레이 타입과 동일한 하나의 값으로 묶어 출력합니다.
UNI	WORD	WORD	
UNI	DWORD	DWORD	
UNI	LWORD	LWORD	



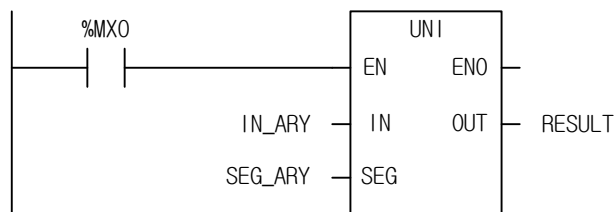
SEG로 지정된 값들의 합이 입력 변수 타입의 비트수를 초과할 경우 `_ERR/_LER` 플래그가 셋(Set)됩니다.

■ 플래그

플래그	설명
<code>_ERR</code>	SEG 로 지정된 값들의 합이 입력 변수 타입의 비트수와 일치하지 않는 경우 <code>_ERR</code> , <code>_LER</code> 플래그가 셋(SET)됩니다. IN 과 SEG 어레이의 개수가 서로 다를 경우, 0 이 출력되고 <code>_ERR</code> , <code>_LER</code> 플래그가 셋(SET)됩니다.

■ 프로그램 예

1. LD



2. ST

RESULT := UNI(EN:=%MX0, IN:=IN_ARY, SEG:=SEG_ARY);

- (1) 실행조건(%MX0)이 0n 되면, UNI 평선이 실행됩니다.
- (2) 입력 변수로 선언된 IN_ARY 와 SEG_ARY 의 값이 다음과 같을 경우

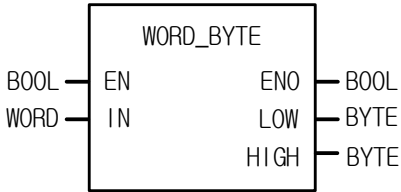
IN_ARY[0]	A3B5	SEG_ARY[0]	3
IN_ARY[1]	B4C6	SEG_ARY[1]	4
IN_ARY[2]	C5D7	SEG_ARY[2]	7
IN_ARY[3]	D6E8	SEG_ARY[3]	2

평선이 실행된 후에 출력변수의 RESULT 의 값은 2#0010_10111_0110_0101 = 16#2BB5 로 출력됩니다.

IN_ARY[0]	2#1010 0011 1011 0101	SEG_ARY[0]	3
IN_ARY[1]	2#1011 0100 1100 0110	SEG_ARY[1]	4
IN_ARY[2]	2#1100 0101 1101 0111	SEG_ARY[2]	7
IN_ARY[3]	2#1101 0110 1110 1000	SEG_ARY[3]	2



RESULT : 2#00 1010111 0110 101

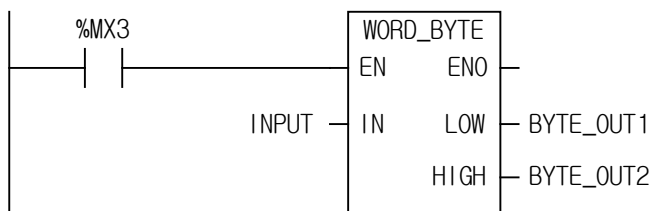
WORD_BYTE	적용 기종	발생플래그
WORD를 2개의 BYTE로 나눔	XGF-M32E	-
평 선	설 명	
	<p>입력 EN : 1일 때 평선 실행 IN : WORD 입력</p> <p>출력 ENO : EN 값을 그대로 출력 LOW : 하위 BYTE 출력 HIGH : 상위 BYTE 출력</p>	

■ 기능

- 하나의 워드를 2개의 바이트로 분산합니다.
LOW: 하위 바이트 출력, HIGH: 상위 바이트 출력

■ 프로그램 예

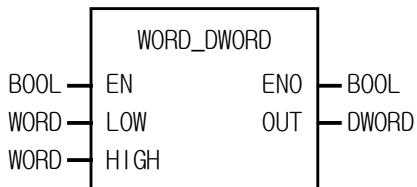
1. LD



2. ST

```
WORD_BYTE(EN:=%MX3, IN:=INPUT, LOW=>BYTE_OUT1, HIGH=>BYTE_OUT2);
```

- 실행조건(%MX3)이 On 되면 WORD_BYTE 평선이 실행됩니다.
- 입력변수로 선언된 INPUT값이 16#ABCD일 때 출력 변수로 선언된 변수 BYTE_OUT1 = 16#CD, BYTE_OUT2 = 16#AB 값이 저장됩니다.

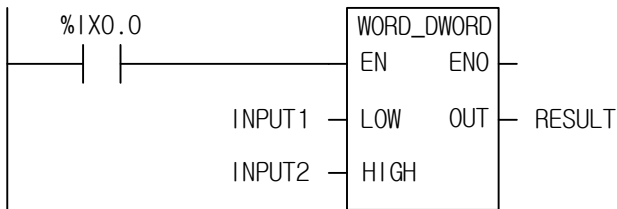
WORD_DWORD	적용 기종	발생플래그
2개의 WORD를 DWORD로 모음	XGF-M32E	-
평 선	설 명	
	<p>입력 EN : 1일 때 평선 실행 LOW : 하위 WORD 입력 HIGH : 상위 WORD 입력</p> <p>출력 ENO : EN 값을 그대로 출력 OUT : DWORD 출력</p>	

■ 기능

- 2개의 WORD를 하나의 DWORD로 조합합니다.
 LOW: 하위 워드 입력, HIGH: 상위 워드 입력

■ 프로그램 예

1. LD



2. ST

```
RESULT := WORD_DWORD(EN:=%IX0.0, LOW:=INPUT1, HIGH:=INPUT2);
```

- 실행조건(%IX0.0)이 On 되면 WORD_DWORD 평선이 실행됩니다.
- 입력변수로 선언된 INPUT1 = 16#1020이고 INPUT2 = 16#A0B0 일 때 출력변수로 선언된 RESULT = 16#A0B0_1020가 됩니다.

XCHG		적용 기종										발생플래그									
2 개의 입력변수 값을 서로 교환		XGF-M32E										-									
평 선		설 명																			
<pre> graph LR subgraph XCHG EN[EN] --- SRC1[SRC1] SRC1 --- SRC2[SRC2] SRC2 --- ENO[ENO] SRC1 --- SRC1_OUT[SRC1] SRC2 --- SRC2_OUT[SRC2] end EN --- ENO SRC1 --- SRC1_OUT SRC2 --- SRC2_OUT </pre>		입력 EN : 1일 때 평선 실행 출력 ENO : EN 값을 그대로 출력 입출력 SRC1 : 입출력 1 SRC2 : 입출력 2																			
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	
	SRC1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
	SRC2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	

■ 기능

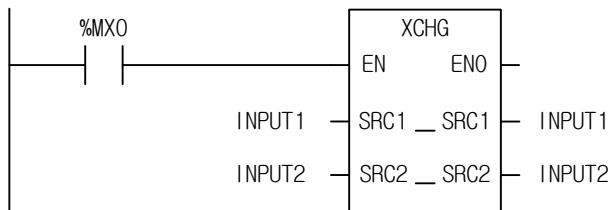
1. 입력 1 과 입력 2 의 데이터를 교환합니다.

평 선	입출력타입	동작 설명
XCHG	BOOL	두 개의 BOOL 입력 값을 서로 교환합니다.
XCHG	BYTE	두 개의 BYTE 입력 값을 서로 교환합니다.
XCHG	WORD	두 개의 WORD 입력 값을 서로 교환합니다.
XCHG	DWORD	두 개의 DWORD 입력 값을 서로 교환합니다.
XCHG	LWORD	두 개의 LWORD 입력 값을 서로 교환합니다.
XCHG	SINT	두 개의 SINT 입력 값을 서로 교환합니다.
XCHG	INT	두 개의 INT 입력 값을 서로 교환합니다.
XCHG	DINT	두 개의 DINT 입력 값을 서로 교환합니다.
XCHG	LINT	두 개의 LINT 입력 값을 서로 교환합니다.
XCHG	USINT	두 개의 USINT 입력 값을 서로 교환합니다.
XCHG	UINT	두 개의 UINT 입력 값을 서로 교환합니다.
XCHG	UDINT	두 개의 UDINT 입력 값을 서로 교환합니다.
XCHG	ULINT	두 개의 ULINT 입력 값을 서로 교환합니다.
XCHG	REAL	두 개의 REAL 입력 값을 서로 교환합니다.

평 선	입출력타입	동작 설명
XCHG	LREAL	두 개의 LREAL 입력 값을 서로 교환합니다.
XCHG	TIME	두 개의 TIME 입력 값을 서로 교환합니다.
XCHG	DATE	두 개의 DATE 입력 값을 서로 교환합니다.
XCHG	TOD	두 개의 TOD 입력 값을 서로 교환합니다.
XCHG	DT	두 개의 DT 입력 값을 서로 교환합니다.

■ 프로그램 예

1. LD



2. ST

XCHG(EN:=%MX0, SRC1:=INPUT1, SRC2:=INPUT2);

- (1) 실행조건(%MX0)이 On 되면 XCHG 평선이 실행됩니다.
- (2) 평선의 입력변수 INPUT1 = 0 이고, INPUT2 = 1이면 평선이 실행된 후 2개의 입력값은 서로 바뀌어 출력됩니다. 따라서 평선이 실행된 후 INPUT1 = 1이고, INPUT2 = 0이 됩니다.

XNR		적용 기종	발생플래그																	
배타적 논리곱		XGF-M32E	-																	
평 선		설 명																		
<pre> graph LR subgraph XNR EN[EN] IN1[IN1] IN2[IN2] ENO[ENO] OUT[OUT] end EN --- ENO IN1 --- OUT IN2 --- OUT </pre>		<p>입력 EN : 1일 때 평선 실행 IN1 : XNR 될 값 IN2 : XNR 될 값 입력 8 개까지 확장 가능</p> <p>출력 ENO : EN 값이 그대로 출력 OUT : XNR 된 값</p> <p>IN1, IN2, OUT 은 모두 같은 타입이어야 함.</p>																		
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT
	IN1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>														
	IN2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>														
	OUT	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>														

■ 기능

1. IN1 을 IN2 와 비트별로 XNR 해서 OUT 으로 출력시킵니다.

IN1 1111 0000

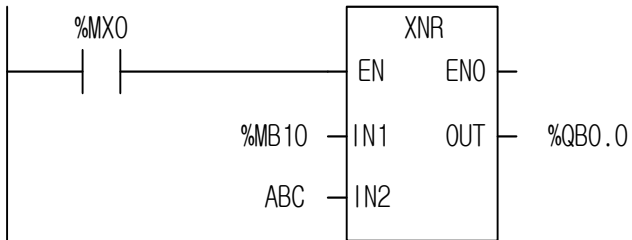
XNR

IN2 1010 1010

OUT 1010 0101

■ 프로그램 예

1. LD



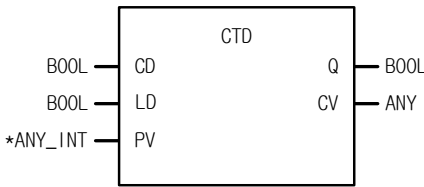
2. ST

```
%QB0.0 := XNR(EN:=%MX0, IN1:=%MB10, IN2:=ABC);
```

- (1) 실행조건(%MX0)이 On 되면 XNR(배타적 논리곱) 평선이 실행됩니다.
- (2) %MB10 = 16#F0 = 2#1111_0000 이고, ABC(BYTE 타입) = 16#AA = 2#1010_1010 이면 XNR 평선이 실행된 후의 출력값은 %QB0.0.0 = 16#A5 = 2#1010_0101 이 됩니다.

제 8 장 기본 평선 블록

1. 기본 평선 블록에 대한 설명입니다.

CTD		적용 기종	발생플래그																			
감산 카운터 (평선 블록)		XGF-M32E	-																			
평선 블록		설 명																				
		입력 CD : 다운_카운트(Down_Count) 펄스입력 LD : 설정값 입력(Load) PV : 설정값(Preset Value)																				
		출력 Q : 카운트_다운(Count_Down) 출력 CV : 현재값(Current Value)																				
ANY 타입 변수설명	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
	PV							○	○	○		○	○	○								
	CV							○	○	○		○	○	○								

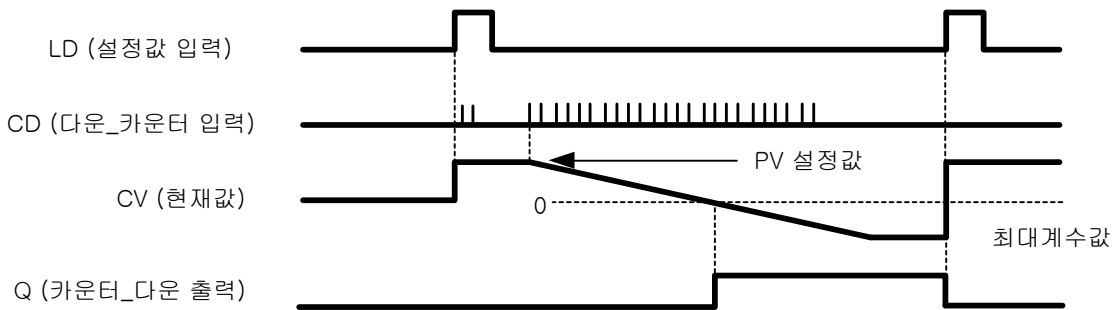
*ANY_INT: ANY_INT 타입 중 SINT, USINT 제외.

■ 기능

1. 감산 카운터 평선 블록 CTD는 다운 카운터 펄스입력 CD가 0에서 1이 되면, 현재값 CV가 이전값보다 1만큼 감소하는 카운터입니다.
2. 단, CV는 PV의 최소보다 클 때만 감소하고, 최소값이 되면 더이상 감소하지 않습니다.
3. 설정값 입력 LD가 1이 되면 현재값 CV에는 설정값 PV값이 로드 됩니다. (CV = PV)
4. 출력 Q는 CV가 0이하일 때만 1이 됩니다.

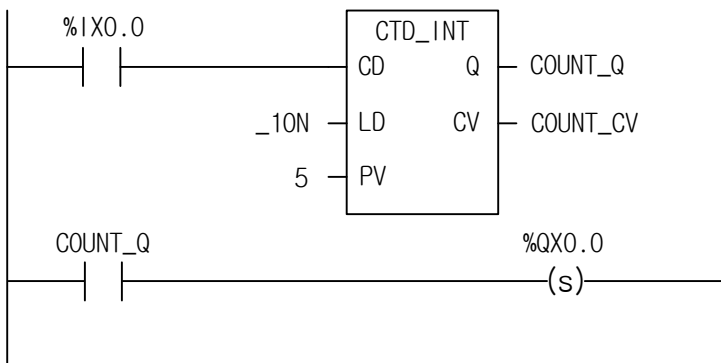
평선 블록	PV	동작 설명
CTD_INT	INT	INT(설정값)의 최소값(-32,768)만큼 감소합니다.
CTD_DINT	DINT	DINT(설정값)의 최소값(-2,147,483,648)만큼 감소합니다.
CTD_LINT	LINT	LINT(설정값)의 최소값(-9,223,372,036,854,775,808)만큼 감소합니다.
CTD_UINT	UINT	UINT(설정값)의 최소값(0)만큼 감소합니다.
CTD_UDINT	UDINT	UDINT(설정값)의 최소값(0)만큼 감소합니다.
CTD_ULINT	ULINT	ULINT(설정값)의 최소값(0)만큼 감소합니다.

■ 타임차트



■ 프로그램 예

1. LD

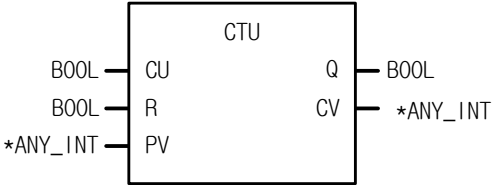


2. ST

```
INST_CTD_INT(CD:=%IX0.0, LD:=_10N, PV:=5, Q=>COUNT_Q, CV=>COUNT_CV);
%QX0.0 := COUNT_Q;
```

입력점점 %IX0.0 에 5 번의 펄스가 들어오면, 출력점점 %QX0.0 을 Set 시키는 프로그램

- (1) CTD 평선 블록의 이름을 등록합니다. (COUNT_D)
- (2) CD 에 펄스입력이 들어올 점점 %IX0.0 를 입력합니다.
- (3) PV 를 CV 로 로드시킬 사용자 플래그 _10N(첫스캔 On)을 입력합니다.
- (4) PV 값은 INT 범위(-32,768~32,767) 내에서 5 를 입력합니다.
- (5) CV 에는 임의의 출력변수(COUNT_CV)를 설정하여 입력합니다.
- (6) Q 에는 임의의 출력변수(COUNT_Q)를 설정 입력합니다.
- (7) 프로그램의 작성이 완료되면, 컴파일을 실행하고, PLC 로 쓰기를 합니다.
- (8) 쓰기를 완료하면 모드전환(Stop → Run) 시킵니다.
- (9) 프로그램이 Run 되면 PV 값 5 가 CV(Count_CV)로 로드됩니다.
- (10) 입력펄스가 입력점점 %IX0.0 에 들어오면 현재값 CV(COUNT_CV)는 1 만큼 줄어듭니다.
- (11) 입력점점에 5 번의 펄스가 들어오면 현재값 CV 는 0 가 되고, Q(COUNT_Q)는 1 이 됩니다.
- (12) Q(COUNT_Q)가 1 이 되면 출력점점 %QX0.0 이 Set 됩니다.

CTU		적용 기종	발생플래그																																																															
가산 카운터 (평선 블록)		XGF-M32E	-																																																															
평선 블록		설명																																																																
		<p>입력 CU : 업_카운트(Up_Count) 펄스입력 R : 리셋 입력(Reset) PV : 설정값 (Preset Value)</p> <p>출력 Q : 업_카운트(Up_Count) 출력 CV : 현재값(Current Value)</p>																																																																
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> <th>STRING</th> </tr> </thead> <tbody> <tr> <td>PV</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>CV</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	PV							○	○	○		○	○	○								CV							○	○	○		○	○	○									
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING																																														
PV							○	○	○		○	○	○																																																					
CV							○	○	○		○	○	○																																																					

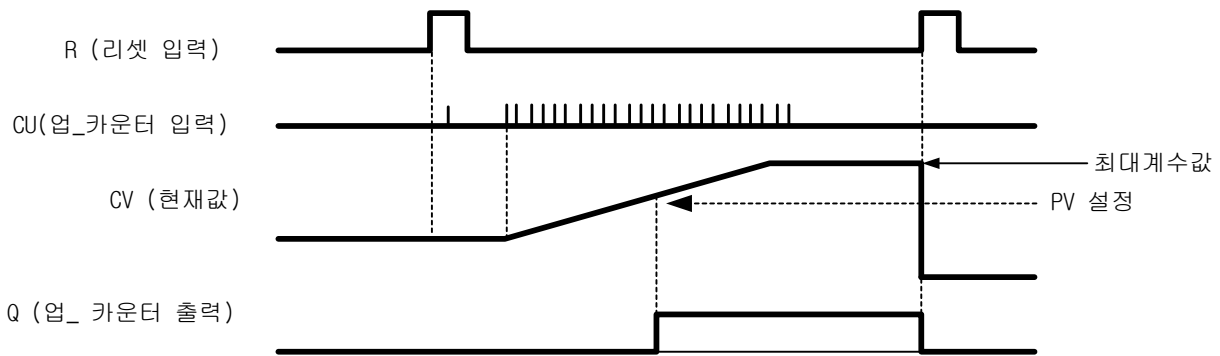
*ANY_INT: ANY_INT 타입 중 SINT, USINT 제외.

■ 기능

1. 가산 카운터 평선 블록 CTU 는 업 카운터 펄스입력 CU 가 0 에서 1 이 되면 현재값 CV 가 이전값 보다 1 만큼 증가하는 카운터입니다.
2. 단, CV 가 PV 의 최대값 미만일 때만 증가하고, 최대값이 되면 더이상 증가하지 않습니다.
3. 리셋 입력 R 이 1 이 되면 현재값 CV 는 0 으로 클리어(Clear)됩니다.
4. 출력 Q 는 CV 가 PV 이상이 될 때만 1 이 됩니다.
5. PV 값은 CTU 평선 블록을 수행 시 설정값을 새롭게 가져와 연산합니다.

평선 블록	PV	동작 설명
CTU_INT	INT	INT(설정값)의 최대값(32767)만큼 증가합니다.
CTU_DINT	DINT	DINT(설정값)의 최대값(2147483647)만큼 증가합니다.
CTU_LINT	LINT	LINT(설정값)의 최대값(9223372036854775807)만큼 증가합니다.
CTU_UINT	UINT	UINT(설정값)의 최대값(0)만큼 증가합니다.
CTU_UDINT	UDINT	UDINT(설정값)의 최대값(0)만큼 증가합니다.
CTU_ULINT	ULINT	ULINT(설정값)의 최대값(0)만큼 증가합니다.

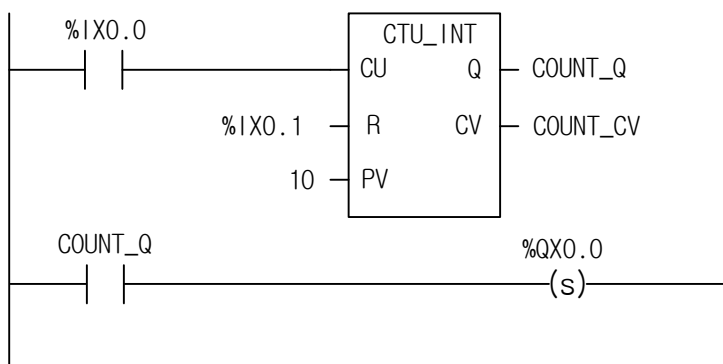
■ 타임차트



■ 프로그램 예

1. 입력점점 %IX0.0 에 10 번의 펄스가 들어오면, 출력점점 %QX0.0 을 Set 시키는 프로그램

1. LD

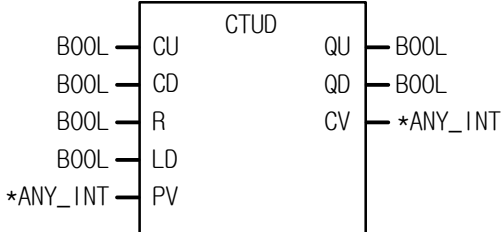


2. ST

```
INST_CTU_INT(CU:=%IX0.0, R:=%IX0.1, PV:=10, Q=>COUNT_Q, CV=>COUNT_CV);
%QX0.0 := COUNT_Q;
```

- (1) CTU 평선 블록의 이름을 등록합니다. (COUNT_U)
- (2) CU 에 펄스 입력이 들어올 입력 점점 %IX0.0 를 입력합니다.
- (3) PV 에 10 을 입력합니다.
- (4) CV 를 초기화시키는 R 에는 임의의 입력 점점을 설정합니다.(%IX0.1)
- (5) CV 에는 임의의 변수(COUNT_CV)를 설정하여 입력합니다.
- (6) Q 에는 임의의 출력변수(COUNT_Q)를 설정 입력합니다.
- (7) 프로그램의 작성이 완료되면, 컴파일을 실행하고, PLC 로 쓰기를 합니다.
- (8) 쓰기를 완료하면 모드전환(Stop →Run) 시킵니다.
- (9) 입력펄스가 입력점점 %IX0.1.15 에 들어오면 현재값 CV(COUNT_CV)는 1 만큼 증가합니다.

- (10) 입력접점에 10 번의 펄스가 들어오면 현재값 CV 는 10 이 되고, 설정값과 같게 되므로 출력 Q(COUNT_Q)가 1 이 됩니다.
- (11) Q(COUNT_Q)가 1 이 되면 출력접점 %QX0.0 이 셋(Set)됩니다.

CTUD		적용 기종	발생플래그																																																															
가감산 카운터 (평선 블록)		XGF-M32E	-																																																															
평선 블록		설 명																																																																
		입력 CU : 업_카운트(Up_Count) 펄스입력 CD : 다운_카운트(Down_Count) 펄스입력 R : 리셋 입력(Reset) LD : 설정값 입력(Load) PV : 설정값(Preset Value) 출력 QU : 카운트_업(Count_UP) 출력 QD : 카운트_다운(Count_Down) 출력 CV : 현재값(Current Value)																																																																
ANY 타입 변수설명	<table border="1"> <thead> <tr> <th>변수명</th> <th>BOOL</th> <th>BYTE</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>SINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>USINT</th> <th>UINT</th> <th>UDINT</th> <th>ULINT</th> <th>REAL</th> <th>LREAL</th> <th>TIME</th> <th>DATE</th> <th>TOD</th> <th>DT</th> <th>STRING</th> </tr> </thead> <tbody> <tr> <td>PV</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>CV</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	PV							○	○	○		○	○	○								CV							○	○	○		○	○	○									
변수명	BOOL	BYTE	WORD	DWORD	LWORD	SINT	INT	DINT	LINT	USINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING																																														
PV							○	○	○		○	○	○																																																					
CV							○	○	○		○	○	○																																																					

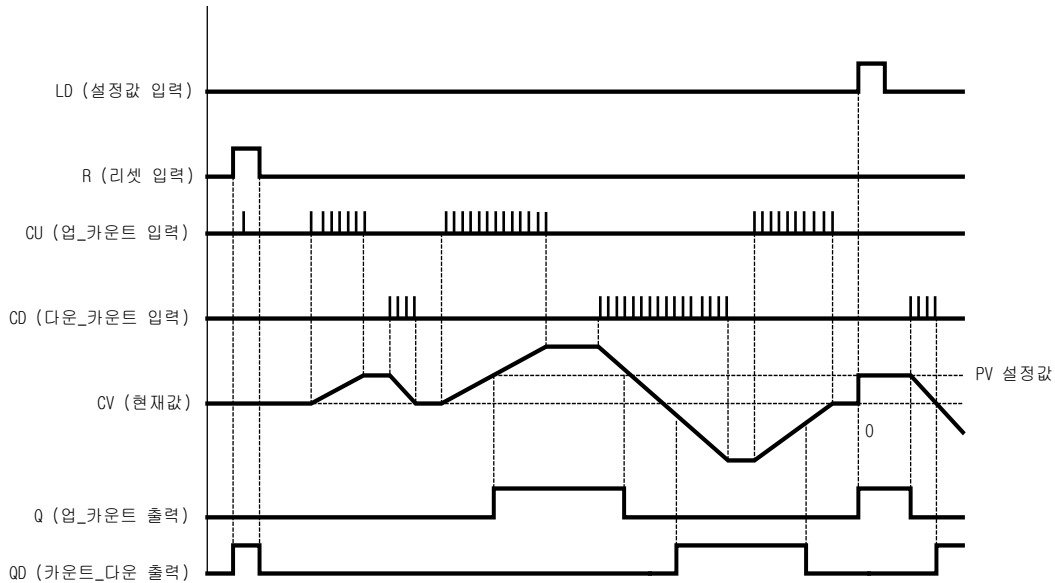
*ANY_INT: ANY_INT 타입 중 SINT, USINT 제외.

■ 기능

1. 가감산 카운터 평선 블록 CTUD 는 업카운터 펄스 입력 CU 가 0 에서 1 이 되면 현재값 CV 가 이전값보다 1 만큼 증가 하고, 다운 카운터 펄스입력 CD 가 0 에서 1 이 되면 현재값 CV 가 이전값보다 1 만큼 감소하는 카운터 입니다.
2. 단, CV 가 PV 의 최소값과 최대값 사이의 값을 가지며 최대, 최소값에 이르면 각각 더이상 증가, 감소하지 않습니다.
3. 설정값 입력 LD 가 1 이 되면 현재값 CV 에는 설정값 PV 값이 로드됩니다. (CV = PV)
4. 설정값 입력 R 이 1 이 되면 현재값 CV 는 0 으로 클리어(Clear) 됩니다. (CV = 0)
5. 출력 QU 는 CV 가 PV 보다 크면 1 이 되고, QD 는 CV 가 0 이하일 때 1 이 됩니다.
6. 각 입력신호에 대해서 R > LD > CU > CD 순으로 동작을 수행하며, 신호의 중복 발생시 우선 순위가 높은 동작 하나만 수행합니다.

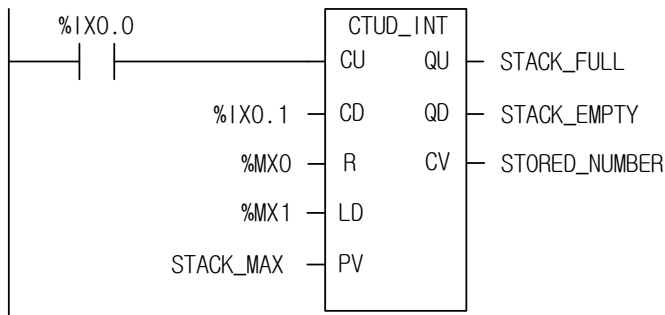
평선 블록	PV	동작 설명
CTUD_INT	INT	INT(설정값)의 -32768 ~ 32767 만큼 증가, 감소합니다.
CTUD_DINT	DINT	DINT(설정값)의 0 ~ 2 ³¹ -1 만큼 증가, 감소합니다.
CTUD_LINT	LINT	LINT(설정값)의 0 ~ 2 ⁶³ -1 만큼 증가, 감소합니다.
CTUD_UINT	UINT	UINT(설정값)의 0 ~ 65535 만큼 증가, 감소합니다.
CTUD_UDINT	UDINT	UDINT(설정값)의 0 ~ 2 ³² -1 만큼 증가, 감소합니다.
CTUD_ULINT	ULINT	ULINT(설정값)의 0 ~ 2 ⁶³ -1 만큼 증가, 감소합니다.

■ 타임차트



■ 프로그램 예

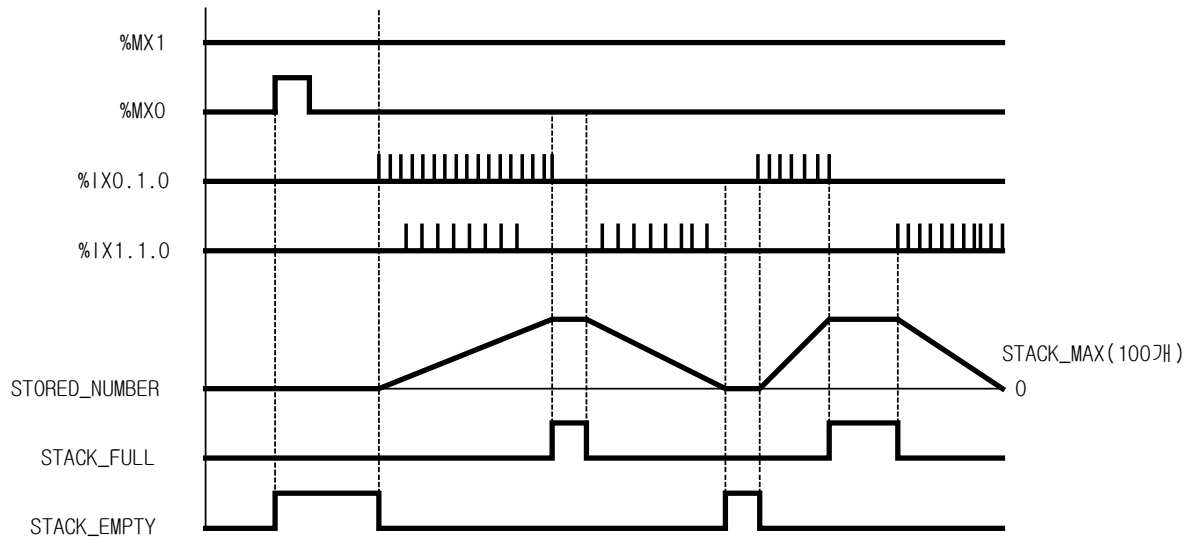
1. LD



2. ST

```
INST_CTUD_INT(CU:=%IX0.0, CD:=%IX0.1, R:=%MX0, LD:=%MX1, PV:=STACK_MAX, QU=>STACK_FULL, QD=>STACK_EMPTY, CV=>STORED_NUMBER);
```

공정라인에서 중간 적재 완충부에 임시 저장될 수 있는 용량 STACK_MAX 의 값이 100 입니다. 적재부에 자재가 투입될 때마다 IN 신호가 1 이 되고, 적재부에서 자재가 빠져 나갈 때마다 OUT 신호가 1 이 되도록 설정되어 있습니다. 운전을 시작하여 선 공정에서 완충부에 투입되는 속도가 다음 공정을 위해 빠져나가는 속도보다 빠를 때, 중간 적재부에 100 개가 쌓이면 카운터의 상한 출력 QU 가 1 이 되어 STACK_FULL 에 1 이 출력됩니다. 반대로 중간 적재부에 남아 있는 것이 없으면 하한 출력 QD 가 1 이 되어 STACK_EMPTY 에 1 이 출력됩니다. STORED_NUMBER 에는 현재 적재부에 남아 있는 자재의 숫자가 표시됩니다.

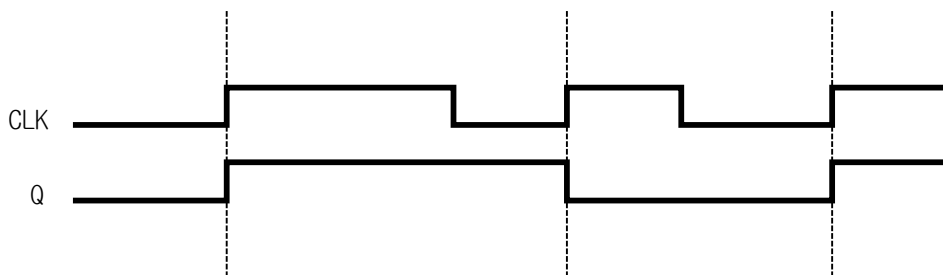


FF	적용 기종	발생플래그
출력 비트 반전	XGF-M32E	-
평선 블록	설 명	
	입력 CLK : 입력신호 출력 Q : 상승에 지시 출력반전	

■ 기능

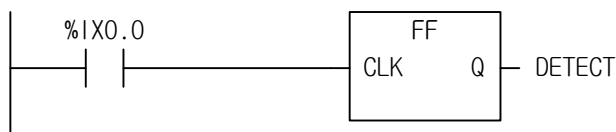
1. FF 는 CLK 에 연결된 입력의 상태가 0 에서 1 로 변할 때 출력 Q 를 반전 시킵니다.

■ 타임차트



■ 프로그램 예

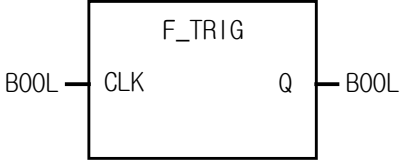
1. LD



2. ST

```
INST_FF(CLK:=%IX0.0, Q=>DETECT);
```

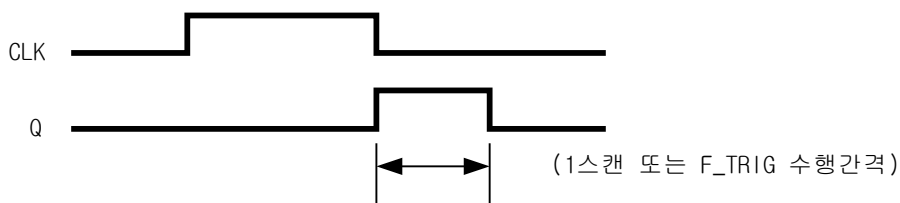
- (1) 입력변수 %IX0.0 의 상태를 감시하여, 입력변수가 0 에서 1 으로 변할 때마다 출력변수 DETECT 의 출력을 반전시킵니다.

F_TRIG	적용 기종	발생플래그
하강 에지 검출	XGF-M32E	-
평선 블록	설 명	
	<p>입력 CLK : 입력신호</p> <p>출력 Q : 하강 에지 검출결과</p>	

■ 기능

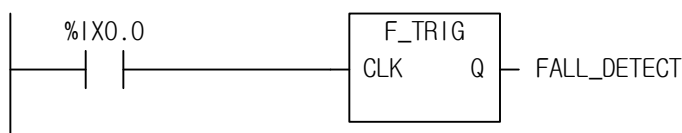
1. F_TRIG는 CLK에 연결된 입력의 상태가 1에서 0으로 변할 때 출력 Q를 1로 만들고 다음 수행에서 0으로 만듭니다. 그 외는, 출력Q가 항상 0이 됩니다.

■ 타임차트



■ 프로그램 예

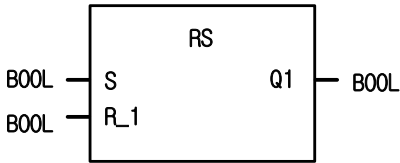
1. LD



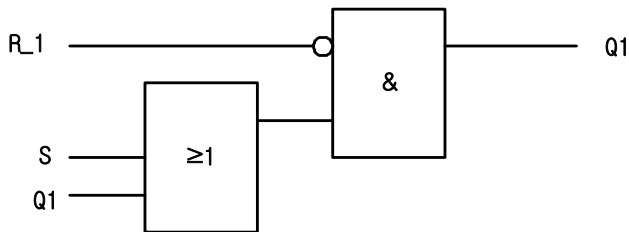
2. ST

```
INST_F_TRIG(CLK:=%IX0.0, Q=>FALL_DETECT);
```

- (1) 입력변수 %IX0.0의 상태를 감시하여 1에서 0으로 변할 때에 출력변수 FALL_DETECT에 1을 출력하고, 다음에 평선 블록 수행시 FALL_DETECT에 0을 출력합니다.

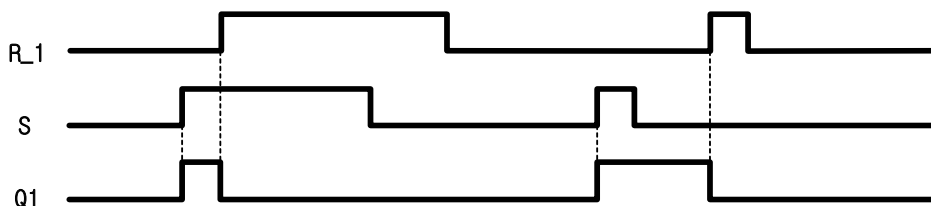
RS	적용 기종	발생플래그
Reset 우선 Bistable	XGF-M32E	-
평선 블록	설 명	
 <p> BOOL — S BOOL — R_1 Q1 — BOOL </p>	<p> 입력 R_1 : Reset 조건 S : Set 조건 </p> <p> 출력 Q1 : 연산결과 </p>	

■ 기능



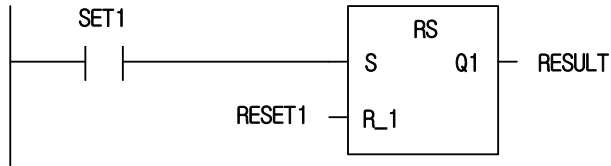
R_1 이 1 이면, S 와 관계없이 출력 Q1 은 항상 0 이 됩니다. 출력 Q1 은 이전 상태를 유지하며, R_1 이 0 이고 S 가 1 일 때 1 이 됩니다. Q1 의 초기상태는 0 입니다.

■ 타임차트



■ 프로그램 예

1. LD



2. ST

```
INST_RS(S:=SET1, R_1:=RESET1, Q=>RESULT);
```

RESET1 을 Reset 조건으로, SET1 을 Set 조건으로 하여 연산 결과를 RESULT 에 출력합니다.

동작 조건은 위의 타임 차트에서 R_1 을 RESET1, S 를 SET1 으로, Q1 을 RESULT 로 대체하여 보십시오.

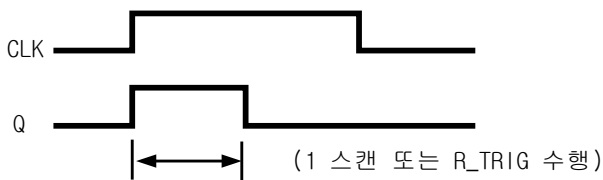
- (1) 입력 변수로 선언된 SET1 이 On 되면 출력 변수 RESULT 는 1 이 됩니다.
- (2) 입력 변수로 선언된 RESET1 이 On 되면 RESULT 로 선언된 출력 변수 값은 0 이 됩니다.
- (3) 입력 변수로 선언된 SET1 과 RESET1 이 On 되면 출력 변수 RESULT 는 0 가 됩니다.

R_TRIG	적용 기종	발생플래그
상승에지 검출 (평선 블록)	XGF-M32E	-
평선 블록	설 명	
	입력 CLK : 입력신호 출력 Q : 상승 에지 검출결과	

■ 기능

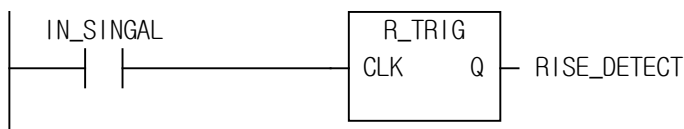
1. R_TRIG 는 CLK 에 연결된 입력의 상태가 0 에서 1 로 변할 때 출력 Q 를 1 로 만들고 R_TRIG 재 실행시 0 으로 만듭니다. 그 이외의 경우, 출력 Q 는 항상 0 이 됩니다.

■ 타임차트



■ 프로그램 예

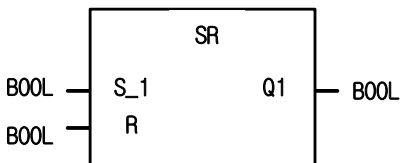
1. LD



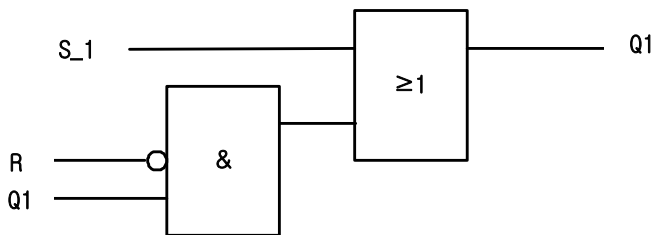
2. ST

```
INST_R_TRIG(CLK:=IN_SIGNAL, Q=>RISE_DETECT);
```

입력변수로 선언된 IN_SIGNAL 의 상태를 감시하여 0 에서 1 로 변할 때, RISE_DETECT 에 1 을 출력하고 R_TRIG 평선 블록 수행 시 RISE_DETECT 에 0 을 출력합니다.

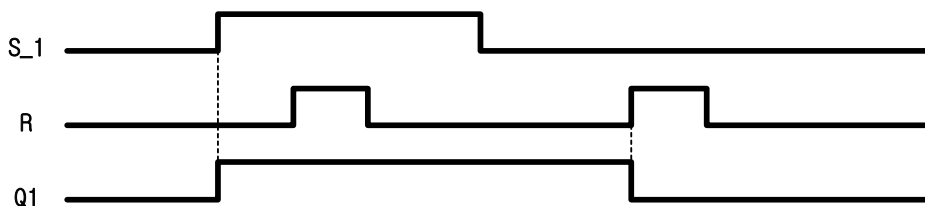
SR	적용 기종	발생플래그
Set 우선 Bistable	XGF-M32E	-
평선 블록	설 명	
	<p>입력 S_1 : Set 조건 R : Reset 조건</p> <p>출력 Q1 : 연산결과</p>	

■ 기능



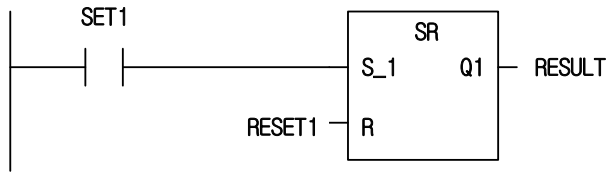
1. S_1이 1이 되면, R과 관계없이 출력 Q1은 항상 1이 됩니다.
2. 출력 Q1은 이전 상태를 유지하며, S_1이 0이고 R이 1일 때 0이 됩니다.
3. Q1의 초기상태는 0입니다.

■ 타임차트



■ 프로그램 예

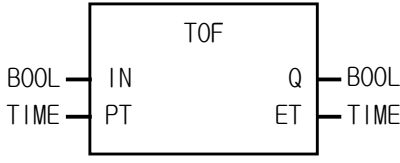
1. LD



2. ST

```
INST_SR(S_1:=SET1, R:=RESET1, Q=>RESULT);
```

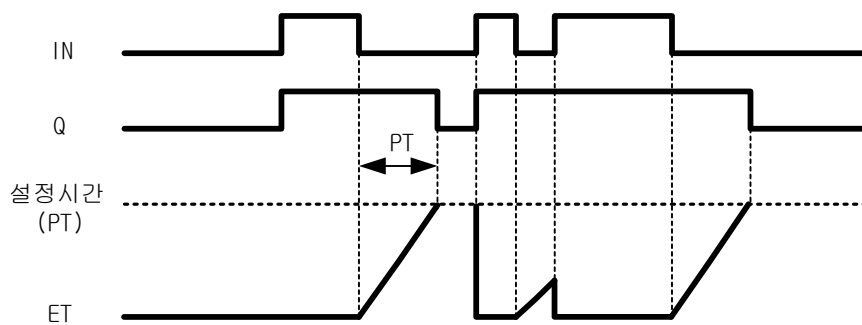
- (1) 입력변수로 선언된 SET1 이 On 되면, 출력변수로 선언된 RESULT 값이 1 이 됩니다.
- (2) 출력변수로 선언된 RESULT 값이 0 이 되도록 하려면, 입력변수로 선언된 SET1 이 Off 되고, RESET 이 On 되어야 합니다.

TOF	적용 기종	발생플래그
Off 딜레이 타이머	XGF-M32E	-
평선 블록	설 명	
	입력 IN : 타이머 기동 조건 PT : 설정 시간(Preset Time)	출력 Q : 타이머 출력 ET : 경과 시간(Elapsed Time)

■ 기능

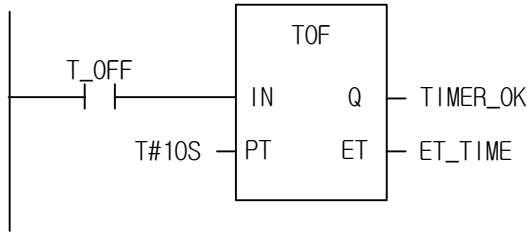
1. IN이 1이 되면, Q가 1이 되고, IN이 0이 된 후부터 PT에 의해서 지정된 설정시간이 경과한 후 Q가 0이 됩니다.
2. IN이 0이 된 후 경과시간이 ET로 출력됩니다.
3. 만일 경과시간 ET가 설정시간에 도달하기 전에 IN이 1이 되면, 경과시간은 다시 0으로 됩니다.

■ 타임차트



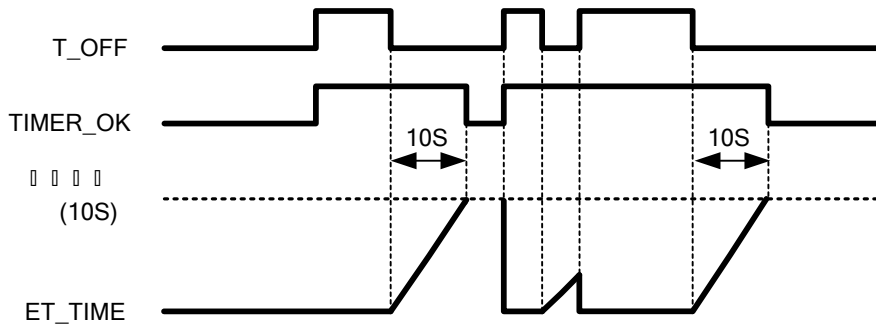
■ 프로그램 예

1. LD

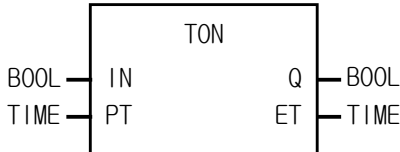


2. ST

```
INST_TOF(IN:=T_OFF, PT:=T#10S, Q=>TIMER_OK, ET=>ET_TIME);
```



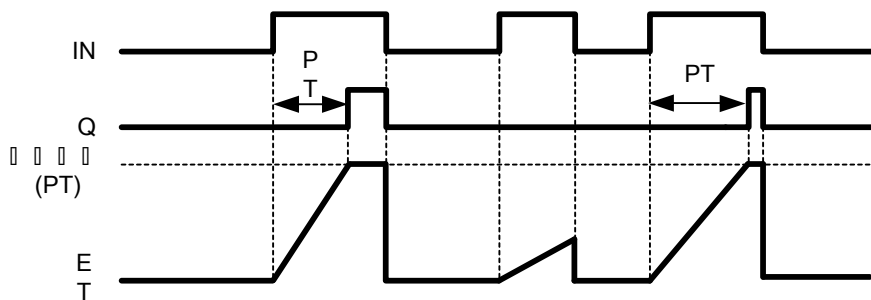
- (1) 입력변수로 설정된 T_OFF 가 1 이 되면, 출력변수 TIMER_OK 에 1 이 출력되고 T_OFF 가 0 이 된 후 10 초 후에 TIMER_OK 가 0 이 됩니다.
- (2) T_OFF 가 0 이 된 후 10 초 이내에 다시 1 이 되면 타이머는 다시 초기상태가 됩니다.
- (3) 타이머의 시간 측정값은 ET_TIME 로 출력됩니다.

TON	적용 기종	발생플래그
On 딜레이 타이머	XGF-M32E	-
평선 블록	설 명	
	입력 IN : 타이머 기동 조건 PT : 설정 시간 (Preset Time)	출력 Q : 타이머 출력 ET : 경과 시간(Elapsed Time)

■ 기능

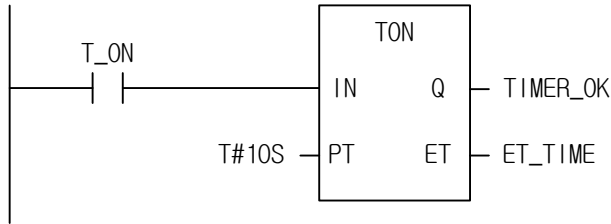
1. IN이 1이 된 후 경과 시간이 ET로 출력됩니다.
2. 만일 경과시간 ET가 설정 시간에 도달하기 전에 IN이 0이 되면, 경과 시간은 0으로 됩니다.
3. Q가 1이 된 후 IN이 0이 되면, Q는 0이 됩니다.

■ 타임차트



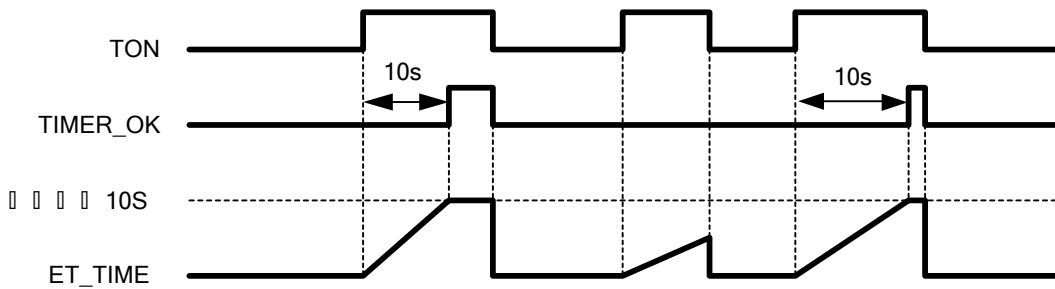
■ 프로그램 예

1. LD

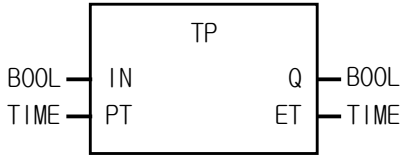


2. ST

```
INST_TON(IN:=T_ON, PT:=T#10S, Q=>TIMER_OK, ET=>ET_TIME);
```



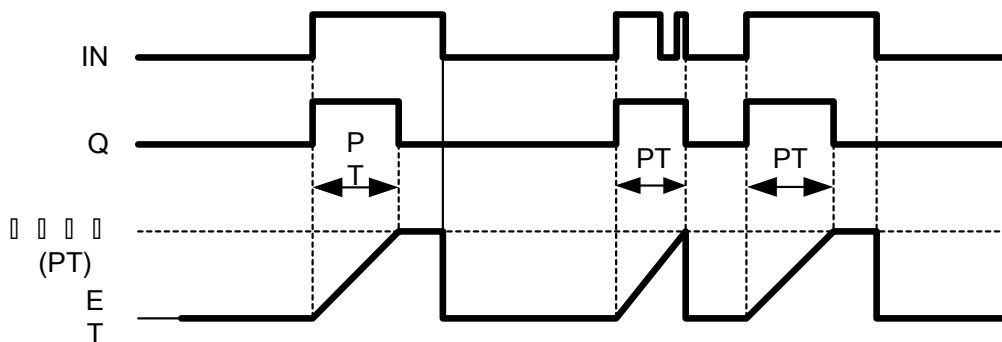
- (1) 입력변수 T_ON이 1이 된 후 10초가 경과한 후 출력 변수 TIMER_OK가 1이 됩니다.
- (2) 입력변수 T_ON이 1이 된 후 경과 시간이 출력 변수 ET_TIME로 출력됩니다.
- (3) 만일 경과 시간 ET_TIME이 설정 시간 10초에 도달하기 전에 T_ON이 0이 되면, 경과 시간 ET_TIME은 0으로 됩니다.
- (4) TIMER_OK가 1이 된 후 T_ON이 0이 되면, TIMER_OK는 0이 되고 경과 시간 ET_TIME도 0으로 됩니다.

TP	적용 기종	발생플래그
펄스 타이머 (평선 블록)	XGF-M32E	-
평선 블록	설 명	
	<p>입력 IN : 타이머 기동조건 PT : 설정 시간 (Preset Time)</p> <p>출력 Q : 타이머 출력 ET : 경과 시간 (Elapsed Time)</p>	

■ 기능

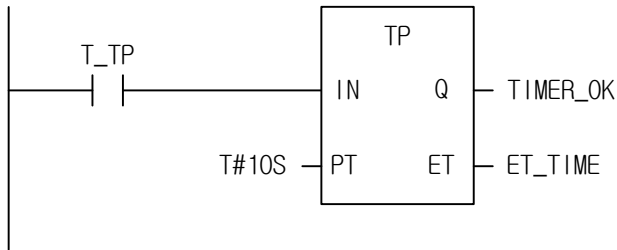
1. IN 이 1 이 되면 PT 에 의해서 지정된 설정 시간 동안만 Q 가 1 이 되고, ET 가 PT 에 도달하면 자동으로 0 이 됩니다.
2. 경과시간 ET 는 IN 이 1 이 되었을 때부터 증가하며 PT 에 이르면 값을 유지하다가 IN 이 0 이 될 때 0 의 값이 됩니다.
3. ET 가 증가할 동안은 IN이 0 이 되거나 재차 1 이 되어도 영향이 없습니다.

■ 타임차트



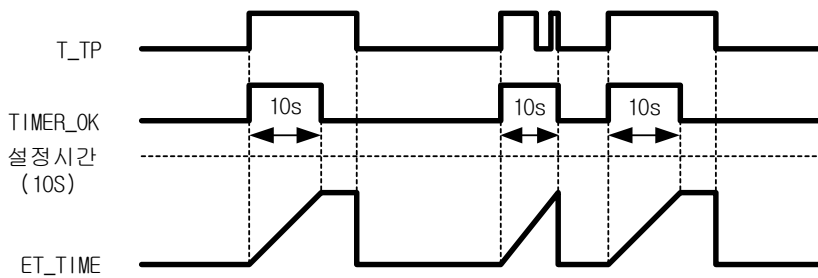
■ 프로그램 예

1. LD



2. ST

```
INST_TP(IN:=T_TP, PT:=T#10S, Q=>TIMER_OK, ET=>ET_TIME);
```



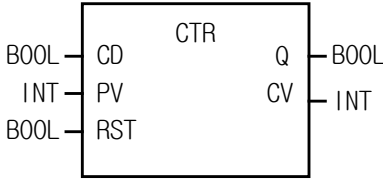
- (1) 입력변수 T_TP 가 0 에서 1 이 된 후 10 초 동안 TIMER_OK 는 1 이 됩니다. 일단 타이머가 가동된 후에는 10 초 동안 T_TP 신호의 변화는 무시됩니다.
- (2) ET_TIME 값은 증가 후 T#10S 에서 멈춥니다. 그리고 T_TP 가 0 이 될 때 0 으로 됩니다.

☆ 참고

TP 평선 블록은 접점이 On된 후 Off되어도 해당 동작이 마무리 될 때까지 평선 블록은 동작합니다. 배열 인덱스를 이용한 변수를 사용하는 경우 배열 인덱스 에러는 접점이 On이 되었을 때만 발생합니다. 그렇기 때문에 TP 평선 블록에서는 평선 블록이 동작하더라도 접점이 Off되어 있다면 배열 인덱스 에러가 발생하지 않습니다.

제 9 장 응용 평선 블록

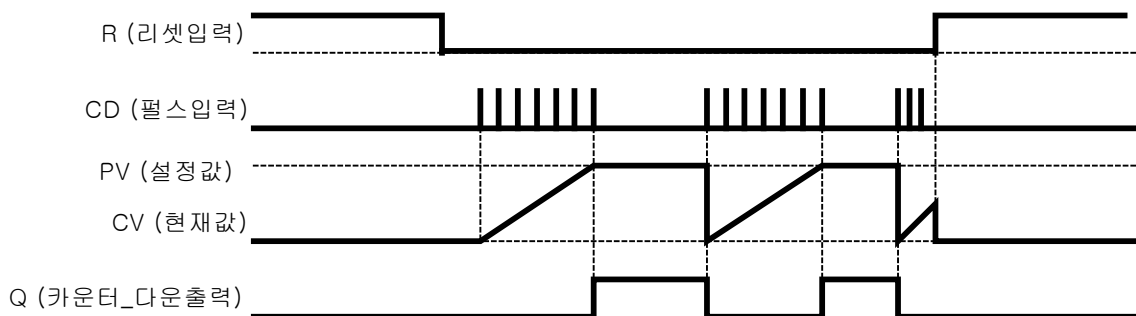
8 장에서 설명한 기본 평선 블록과는 다른 응용 평선 블록에 대하여 설명합니다.

CTR	적용 기종	발생플래그
Ring 카운터	XGF-M32E	-
평선 블록	설 명	
	입력 CD : 링 카운트(Ring Count) 펄스 입력 PV : 설정값(Preset Value) RST : 리셋 입력(Reset)	출력 Q : 링 카운트(Ring Count) 출력 CV : 현재값(Current Value)

■ 기능

1. 링 카운터 CTR 평선 블록은 펄스 입력 CD 가 0 에서 1 이 될 때마다 현재값 CV 를 +1 하고 현재값 CV 가 설정값 PV 에 도달한 후 0 이 다시 0 에서 1 로 되면 현재값은 1 로 됩니다.
2. 현재값이 설정값에 도달하면 출력(Q)은 1 이 됩니다.
3. 현재치가 설정치 미만이거나 Reset 조건이 1 이 되면 출력(Q)은 0 이 됩니다

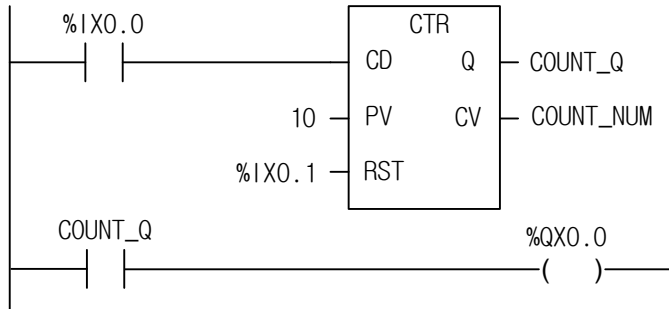
■ 타임 차트



■ 프로그램 예

입력 접점 %IX0.0 에 10 번의 펄스가 들어올 때마다, 출력접점 %QX0.0 을 On 시키는 프로그램

1. LD



2. ST

```
INST_CTR(CD:=%IX0.0, PV:=10, RST:=%IX0.1, Q=>COUNT_Q, CV=>COUNT_NUM);
%QX0.0 := COUNT_Q;
```

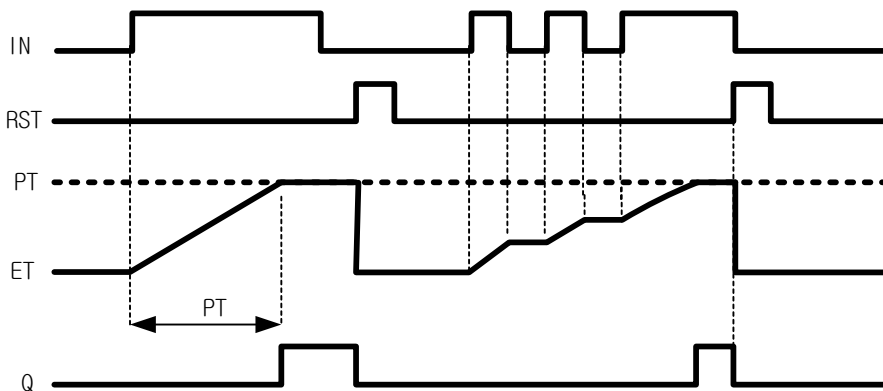
- (1) CTR 평선 블록의 이름을 등록합니다. (INS_CTR)
- (2) CD 에 펄스 입력이 들어올 입력 접점 %IX0.0 을 입력합니다.
- (3) PV 에 10 을 입력합니다.
- (4) CV 를 초기화 시키는 RST 에는 임의의 입력 접점을 설정합니다. (%IX0.1)
- (5) CV 에는 임의의 변수(COUNT_NUM)를 설정하여 입력합니다.
- (6) Q 에는 임의의 출력변수(COUNT_Q)를 설정 입력합니다.
- (7) 프로그램의 작성이 완료되면, 컴파일을 실행하고, PLC 로 쓰기를 합니다.
- (8) 쓰기를 완료하면 모드전환(Stop → Run)시킵니다.
- (9) 입력펄스가 입력접점 %IX0.0 에 들어오면 현재값 CV(COUNT_NUM)은 1 만큼 증가합니다.
- (10) 입력접점에 10 번의 펄스가 들어오면 현재값 CV 는 10 이 되고, 설정값과 같게 되므로 출력 Q(COUNT_Q)가 1 이 됩니다.
- (11) Q(COUNT_Q)가 1 이 되면 출력 접점 %QX0.0 은 On 됩니다.
- (12) 다시 한번의 입력 펄스가 입력 접점 %IX0.0 에 들어오면 Q(COUNT_Q)는 0 이 되고 출력 접점인 %QX0.0 은 Off 됩니다.

TMR	적용 기종	발생플래그
적산 타이머	XGF-M32E	-
평선 블록	설 명	
<p>The diagram shows a rectangular block labeled 'TMR'. On the left side, there are three input terminals: 'IN' (labeled 'BOOL'), 'PT' (labeled 'TIME'), and 'RST' (labeled 'BOOL'). On the right side, there are two output terminals: 'Q' (labeled 'BOOL') and 'ET' (labeled 'TIME').</p>	<p>입력 IN : 타이머 기동 조건 PT : 설정 시간(Preset Time) RST : 리셋 입력(Reset)</p> <p>출력 Q : 타이머 출력 ET : 경과 시간(Elapsed Time)</p>	

■ 기능

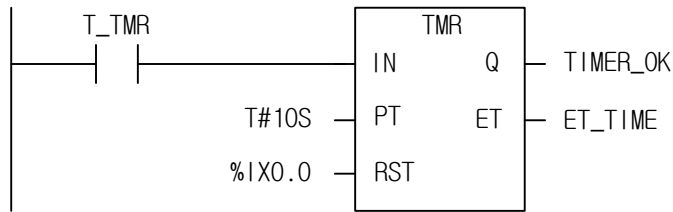
1. TMR 평선 블록은 IN이 1이 된 후 경과 시간이 ET로 출력됩니다.
2. 경과 시간 ET가 설정시간에 도달하기 전에 IN이 0이 되어도 현재의 경과 시간을 유지하다가 IN이 다시 1이 되면 경과 시간을 다시 증가시킵니다.
3. 경과 시간이 설정 시간에 도달하면 Q가 1이 됩니다.
4. Reset 입력 조건이 성립되면 Q는 0이 되며 경과 시간도 0이 됩니다.

■ 타임 차트



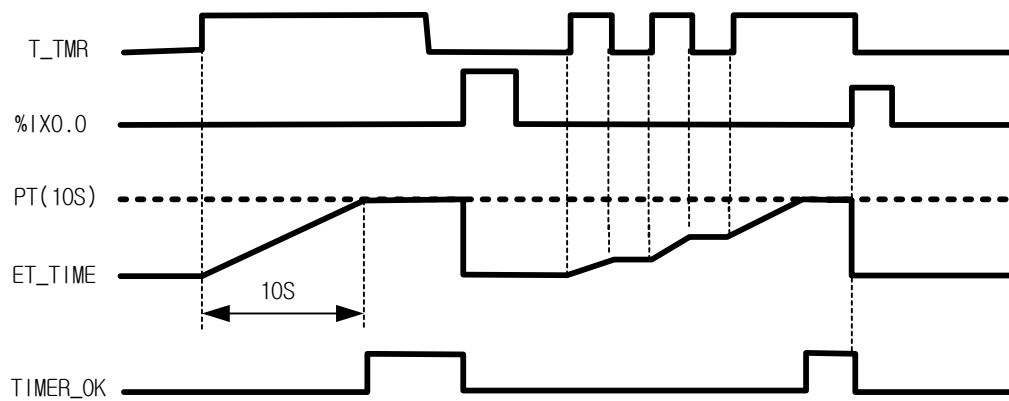
■ 프로그램 예

1. LD



2. ST

```
INST_TMR(IN:=T_TMR, PT:=T#10S, RST:=%IX0.0, Q=>TIMER_OK, ET=>ET_TIME);
```



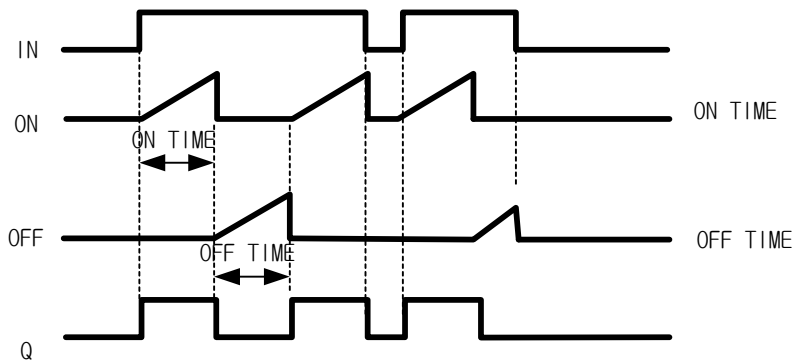
- (1) 입력변수 T_TMR 이 1 이 된 후 10 초가 경과하면 출력 변수 TIMER_OK 가 1 이 됩니다.
- (2) 입력변수 T_TMR 이 1 이 된 후 경과 시간이 출력변수 ET_TIME 으로 출력됩니다.
- (3) 경과 시간 ET_TIME 이 설정시간 10 초에 도달하기 전에 T_TMR 이 0 이 되더라도 현재의 경과 시간을 유지합니다.
- (4) 입력 변수 T_TMR 이 다시 1 이 되면 정지 이전의 경과 시간부터 다시 시작합니다.
- (5) 입력 접점 %IX0.0 가 1 이 되면 경과 시간 ET_TIME 및 출력 변수 TIMER_OK 모두 0 으로 클리어(Clear)됩니다.

TMR_FLK	적용 기종	발생플래그
점멸기능 타이머	XGF-M32E	-
평선 블록	설명	
	입력 IN : 타이머 기동 조건 ON : On 타이머 설정 시간 OFF : Off 타이머 설정 시간 RST : 리셋 입력 출력 Q : 타이머 출력 ET : 경과 시간(Elapsed Time)	

■ 기능

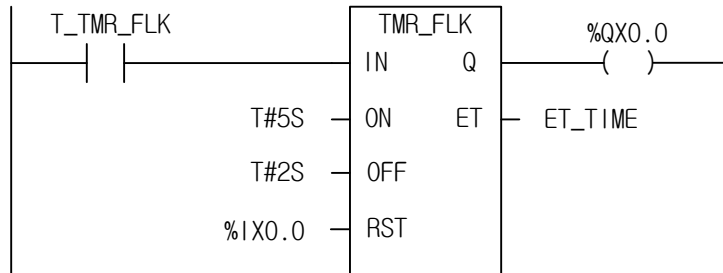
1. TMR_FLK 평선 블록은 IN이 1이 되는 순간 Q는 1이 되고, ON에서 지정된 시간만큼 Q는 1을 유지합니다.
2. ON에서 지정된 시간이 경과하면 OFF에서 지정된 시간만큼 Q는 0이 됩니다.
3. IN이 0이 되면 On 또는 Off 동작을 수행을 중지하고, IN이 0인 동안 중지된 시간을 유지하다가 IN이 다시 1이 되면 정지된 시간부터 다시 타이머가 동작합니다.
4. IN이 0인 동안 출력 Q는 0이 됩니다.
5. ON이 0이면 출력 Q는 항상 0이 됩니다.

■ 타임 차트



■ 프로그램 예

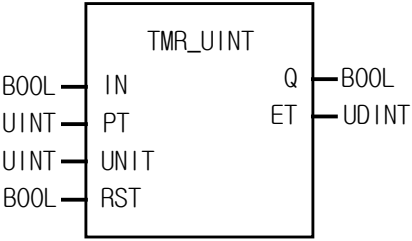
1. LD



2. ST

```
INST_TMR_FLK(IN:=T_TMR_FLK, ON:=T#5S, OFF:=T#2S, RST:=%IX0.0, Q=>%QX0.0, ET=>ET_TIME);
```

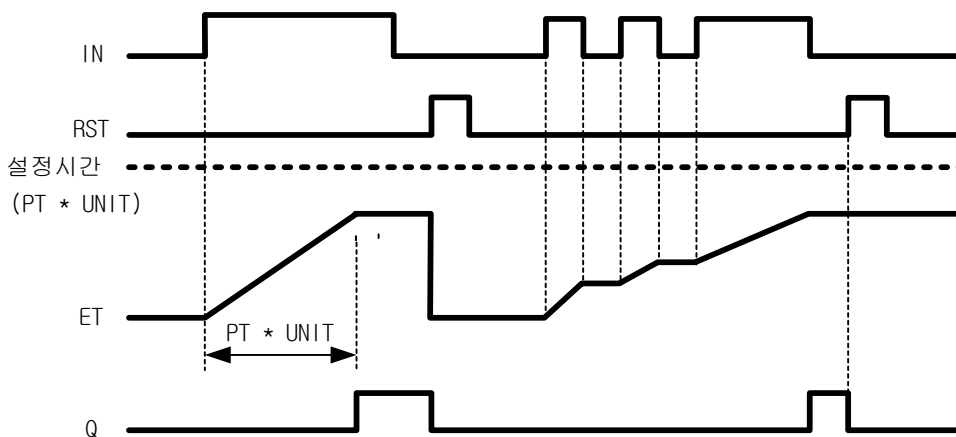
- (1) 입력변수 T_TMR_FLK 가 0에서 1이 되면, TMR_FLK 평선 블록이 동작을 시작합니다.
- (2) 입력변수 T_TMR_FLK 가 1이 된 후 ON에서 지정된 5초만큼 출력 접점 %QX0.0는 1이 됩니다.
- (3) 입력변수 T_TMR_FLK 가 1이 된 후 ON에서 지정된 시간이 경과하면 OFF에서 지정된 2초만큼 출력 접점 %QX0.0는 0이 됩니다.
- (4) 입력변수 T_TMR_FLK 가 1인 동안 출력 Q가 1인 동안의 경과 시간과 Q가 0인 동안의 경과 시간이 번갈아 ET_TIME으로 출력됩니다.
- (5) 입력변수 T_TMR_FLK 가 0이 되면 동작 중인 시간을 유지하고 출력접점 %QX0.0는 0이 되며, T_TMR_FLK가 다시 1이 되면 정지되었던 시간부터 다시 동작합니다.
- (6) 입력 접점 %IX0.0가 1이 되면 경과 시간 ET_TIME 및 출력접점 %QX0.0 모두 0으로 클리어(Clear)됩니다.

TMR_UINT	적용 기종	발생플래그
정수 설정 적산 타이머	XGF-M32E	-
평선 블록	설 명	
	<p>입력</p> <ul style="list-style-type: none"> IN : 타이머 기동 조건 PT : 설정 시간(Preset Time) UNIT : 설정 시간의 시간 단위(Unit) RST : 리셋 입력 <p>출력</p> <ul style="list-style-type: none"> Q : 타이머 출력 ET : 경과 시간(Elapsed Time) 	

■ 기능

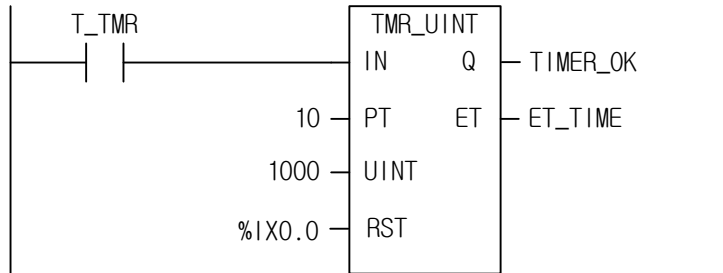
1. TMR_UINT 평선 블록은 IN이 1이 된 후 경과 시간이 ET로 출력됩니다.
2. 경과 시간 ET가 설정 시간에 도달하기 전에 IN이 0이 되어도 현재의 경과 시간을 유지하다가 IN이 다시 1이 되면 경과 시간이 다시 증가됩니다.
3. 경과 시간이 설정 시간에 도달하면 Q가 1이 됩니다.
4. Reset 입력 조건이 성립되면 Q는 0이 되고 경과 시간도 0이 됩니다.
5. 설정시간은 $PT * UNIT[ms]$ 입니다.

■ 타임 차트



■ 프로그램 예

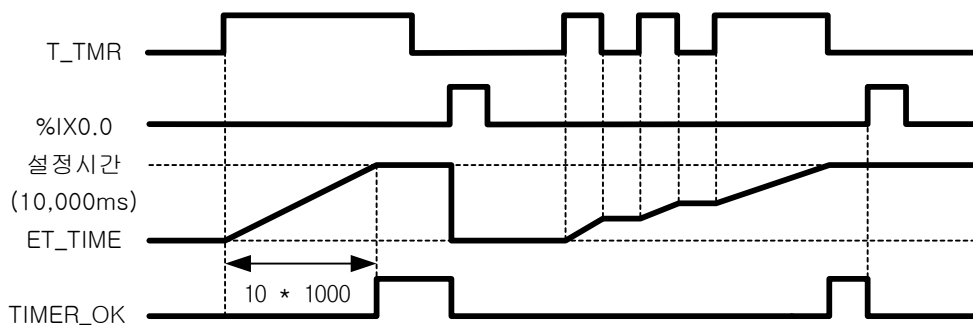
1. LD

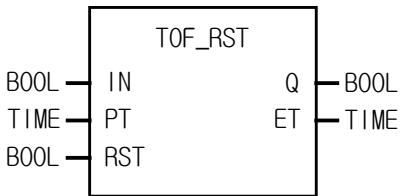


2. ST

```
INST_TMR_UINT(IN:=T_TMR, PT:=10, UNIT:=1000, RST:=%IX0.0, Q=>TIMER_OK, ET=>ET_TIME);
```

- (1) 설정 시간은 $PT * UNIT[ms] = 10 * 1000[ms] = 10[s]$ 가 됩니다.
- (2) 입력변수 T_TMR 이 1 이 된 후 10 초가 경과하면 출력 변수 TIMER_OK 가 1 이 됩니다.
- (3) 입력 변수 T_TMR 이 1 이 된 후 경과 시간이 출력변수 ET_TIME 으로 출력됩니다.
- (4) 경과 시간 ET_TIME 이 설정시간 10 초에 도달하기 전에 T_TMR 이 0 이 되더라도 현재의 경과 시간을 유지합니다.
- (5) 입력 변수 T_TMR 이 다시 1 이 되면 멈추어 섰던 이전의 경과 시간부터 다시 시작합니다.
- (6) 입력 접점 %IX0.0 가 1 이 되면 경과 시간 ET_TIME 및 출력 변수 TIMER_OK 모두 0 으로 클리어(Clear)됩니다.

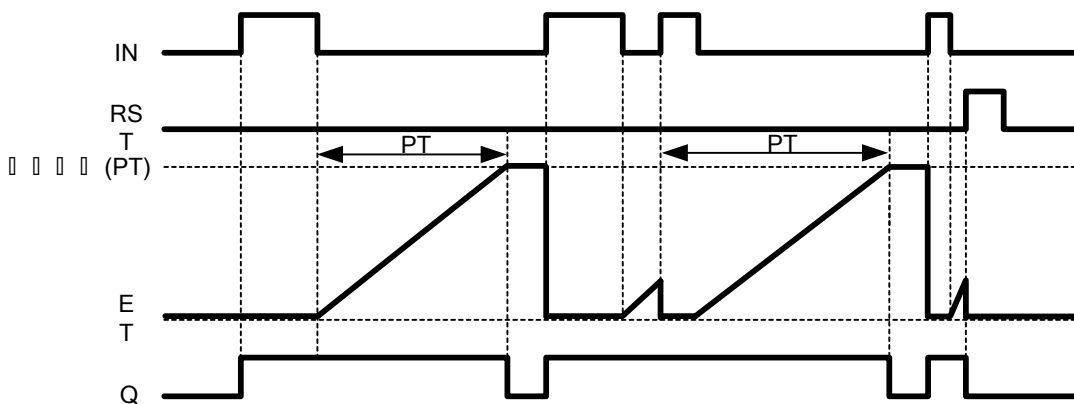


TOF_RST	적용 기종	발생플래그
동작 중 출력 Off 가 가능한 딜레이 타이머	XGF-M32E	-
평선 블록	설 명	
	<p>입력 IN : 타이머 기동 조건 PT : 설정 시간(Preset Time) RST : 리셋 입력(Reset)</p> <p>출력 Q : 타이머 출력 ET : 경과 시간(Elapsed Time)</p>	

■ 기능

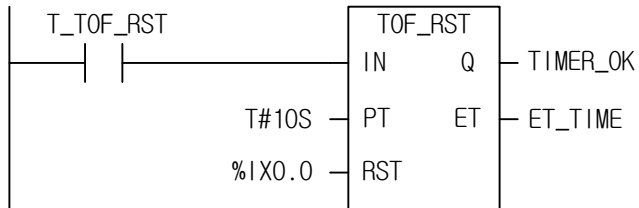
1. TOF_RST 평선 블록은 기동 조건 IN이 1이 되는 순간 Q는 1이 되고, IN이 0이 된 후부터 PT에 의하여 지정된 설정 시간이 경과한 후 Q가 0이 됩니다.
2. IN이 0이 된 후 경과 시간이 ET로 출력됩니다.
3. 만일 경과시간 ET가 설정 시간에 도달하기 전에 IN이 1이 되면, 경과 시간은 다시 0으로 됩니다.
4. Reset 입력 조건이 성립하면 타이머 출력 Q는 0이 되고 경과 시간도 0이 됩니다.

■ 타임 차트



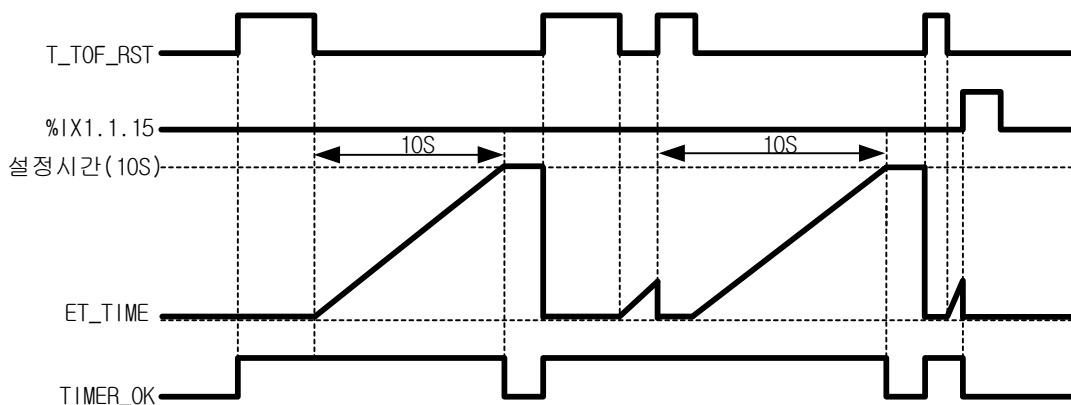
■ 프로그램 예

1. LD



2. ST

```
INST_TOF_RST(IN:=T_TOF_RST, PT:=T#10S, RST:=%IX0.0, Q=>TIMER_OK, ET=>ET_TIME);
```



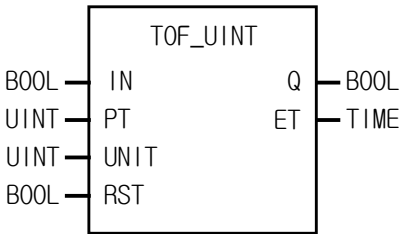
- (1) 입력변수로 설정된 T_TOF_RST 가 1 이 되면, 출력변수 TIMER_OK 에 1 이 출력되고 T_TOF_RST 가 0 이 된 후 10s 후에 TIMER_OK 가 0 이 됩니다.
- (2) T_TOF_RST 가 0 이 된 후 10초 이내에 다시 1 이 되면 타이머는 다시 초기 상태가 됩니다.
- (3) 타이머의 시간 측정값은 ET_TIME 로 출력됩니다.
- (4) 입력접점 %IX0.0 가 1 이 되면 TIMER_OK 와 ET_TIME 모두 0 으로 클리어(Clear)됩니다.

☆ 참고

TOF_RST 평선 블록은 접점이 On 된 후 Off 되어도 해당 동작이 마무리 될 때까지 평선 블록은 동작합니다.

배열인덱스를 이용한 변수를 사용했을 경우 배열인덱스 에러는 접점이 On 이 되었을 때만 발생합니다.

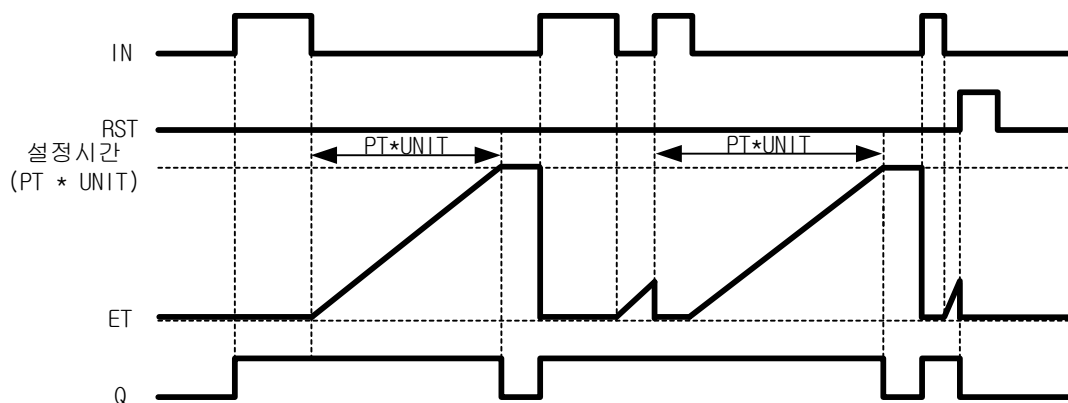
따라서 TOF_RST 평선 블록에서는 평선 블록이 동작하더라도 접점이 Off 되어 있다면 배열인덱스 에러가 발생하지 않습니다.

TOF_UINT	적용 기종	발생플래그
정수 설정 Off 타이머	XGF-M32E	-
평선 블록	설 명	
	<p>입력</p> <p>IN : 타이머 기동 조건 PT : 설정 시간(Preset Time) UNIT : 설정 시간의 시간 단위(Unit) RST : 리셋 입력</p> <p>출력</p> <p>Q : 타이머 출력 ET : 경과 시간(Elapsed Time)</p>	

■ 기능

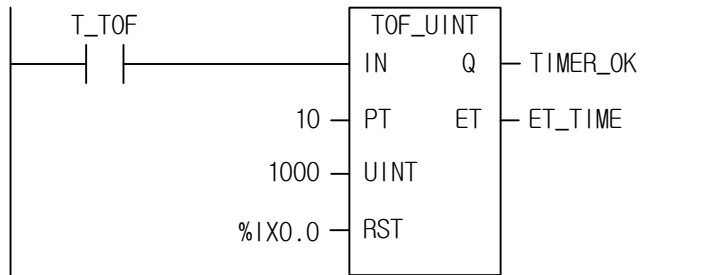
1. TOF_UINT 평선 블록은 기동 조건 IN이 1이 되는 순간 Q는 1이 되고, IN이 0이 된 후부터 PT에 의하여 지정된 설정 시간이 경과한 후 Q가 0이 됩니다.
2. IN이 0이 된 후 경과 시간이 ET로 출력됩니다.
3. 만일 경과시간 ET가 설정시간에 도달하기 전에 IN이 1이 되면, 경과 시간은 다시 0으로 됩니다.
4. Reset 입력 조건이 성립하면 타이머 출력 Q는 0이 되고 경과 시간도 0이 됩니다.
5. 설정시간은 $PT * UNIT[ms]$ 입니다.

■ 타임 차트



■ 프로그램 예

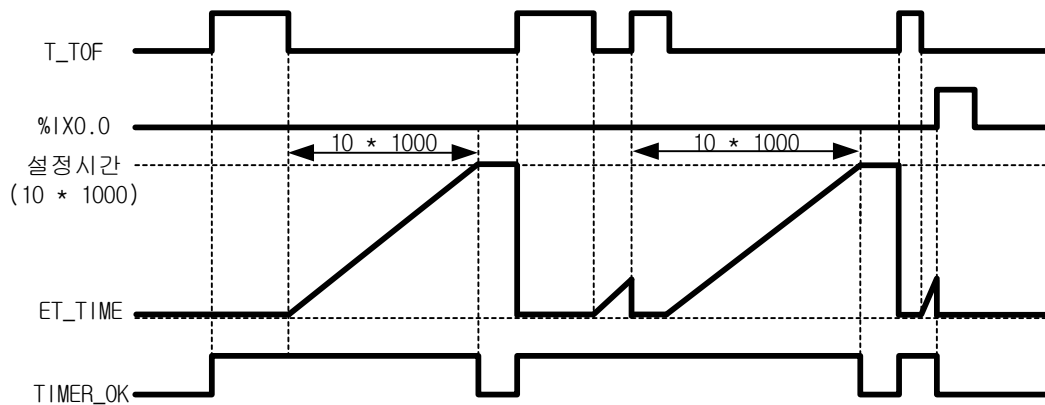
1. LD



2. ST

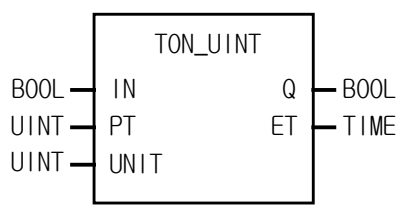
```
INST_TOF_UINT(IN:=T_TOF, PT:=10, UNIT:=1000, RST:=%IX0.0, Q=>TIMER_OK, ET=>ET_TIME);
```

- (1) 설정 시간은 $PT * UNIT[ms] = 10 * 1000[ms] = 10[s]$ 가 됩니다.
- (2) 입력변수로 설정된 T_TOF 가 1 이 되면, 출력변수 TIMER_OK 에 1 이 출력되고 T_TOF 가 0 이 된 후 10 초 후에 TIMER_OK 가 0 이 됩니다.
- (3) T_TOF 가 0 이 된 후 10 초 이내에 다시 1 이 되면 타이머는 다시 초기 상태가 됩니다.
- (4) 타이머의 시간 측정값은 ET_TIME 로 출력됩니다.
- (5) 입력접점 %IX0.0 가 1 이 되면 TIMER_OK 와 ET_TIME 모두 0 으로 클리어(Clear)됩니다.



☆ 참고

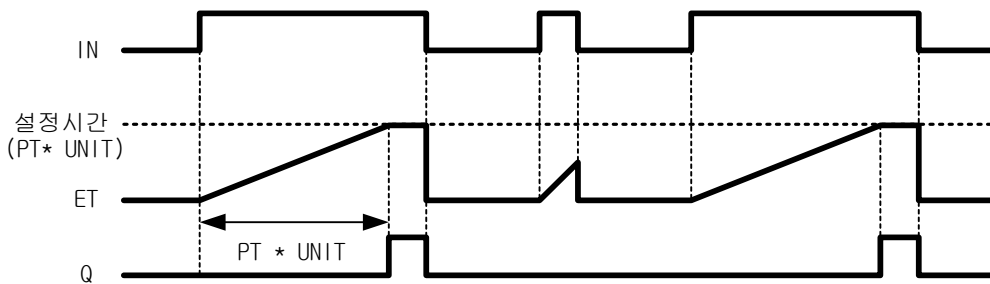
TOF_UINT 평선 블록은 접점이 On 된 후 Off 되어도 해당 동작이 마무리 될 때까지 평선 블록은 동작합니다. 배열인덱스를 이용한 변수를 사용했을 경우 배열인덱스 에러는 접점이 On 이 되었을 때만 발생합니다. 때문에 TOF_UINT 평선 블록에서는 평선 블록이 동작하더라도 접점이 Off 되어 있다면 배열인덱스 에러가 발생하지 않습니다.

TON_UINT	적용 기종	발생플래그
정수 설정 On 타이머	XGF-M32E	-
평선 블록	설 명	
	입력 IN : 타이머 기동 조건 PT : 설정 시간(Preset Time) UNIT : 설정 시간의 시간 단위(Unit)	출력 Q : 타이머 출력 ET : 경과 시간(Elapsed Time)

■ 기능

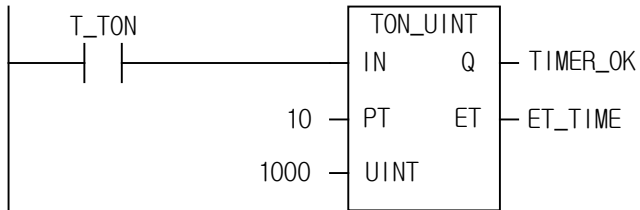
1. TON_UINT 평선 블록은 IN이 1이 된 후 경과 시간이 ET로 출력됩니다.
2. 만일 경과시간 ET가 설정 시간에 도달하기 전에 IN이 0이 되면, 경과 시간 ET는 0으로 됩니다.
3. Q가 1이 된 후 IN이 0이 되면, Q는 0이 됩니다.
4. 설정 시간은 $PT * UNIT[ms]$ 입니다.

■ 타임 차트



■ 프로그램 예

1. LD

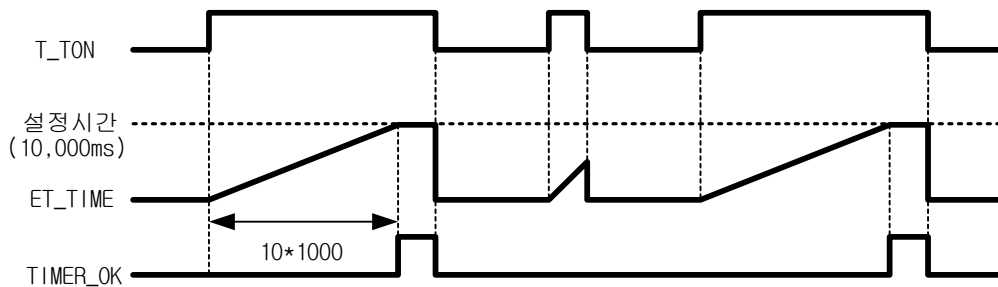


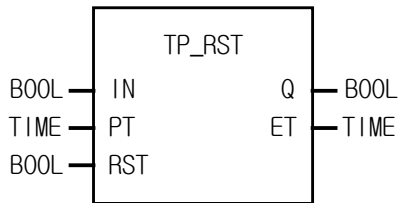
2. ST

```
INST_TON_UINT(IN:=T_TON, PT:=10, UNIT:=1000, Q=>TIMER_OK, ET=>ET_TIME);
```

- (1) 설정 시간은 $PT * UNIT[ms] = 10 * 1000[ms] = 10[s]$ 가 됩니다.
- (2) 입력변수 T_TON 이 0n 된 후, 10 초가 경과한 후에 출력 변수 TIMER_OK 가 1 이 됩니다.
- (3) 입력변수 T_TON 이 0n 된 후, 경과 시간이 출력 변수 ET_TIME 로 출력됩니다.
- (4) 만일 경과 시간 ET_TIME 이 설정 시간 10 초에 도달하기 전에 T_TON 이 0 이 되면, 경과 시간 ET_TIME 은 0 으로 됩니다.
- (5) TIMER_OK 가 1 이 된 후 T_TON 이 0 이 되면, TIMER_OK 는 0 이 되고 경과 시간 ET_TIME 도 0 이 됩니다.

RST

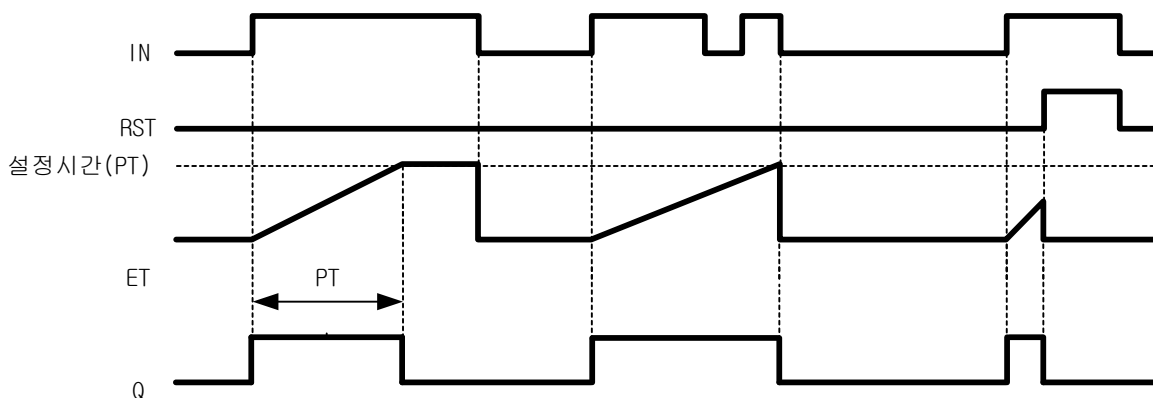


TP_RST	적용 기종	발생플래그
점점 출력 Off 가 가능한 펄스 타이머	XGF-M32E	-
평선 블록	설 명	
	<p>입력 IN : 타이머 기동 조건 PT : 설정 시간(Preset Time) RST : 리셋 입력(Reset)</p> <p>출력 Q : 타이머 출력 ET : 경과 시간(Elapsed Time)</p>	

■ 기능

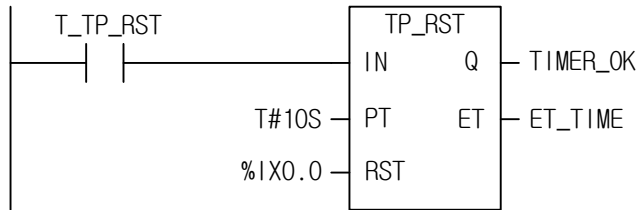
1. TP_RST 평선 블록은 IN이 1이 되는 순간 Q는 1이 되고, 경과 시간이 설정 시간에 도달하면 타이머 출력 Q는 0이 됩니다.
2. 경과 시간 ET는 IN이 1이 되었을 때부터 증가하며 PT에 이르면 값을 유지하다가 IN이 0이 될 때 0으로 클리어 (clear) 됩니다.
3. 타이머 출력 Q가 1인 동안(펄스 출력 중)에는 타이머 기동조건 IN이 1, 0 변화를 하여도 무시합니다.
4. Reset 입력 조건이 성립하면 펄스 출력 중에도 타이머 출력 Q는 0이 되고 경과 시간도 0이 됩니다.

■ 타임 차트



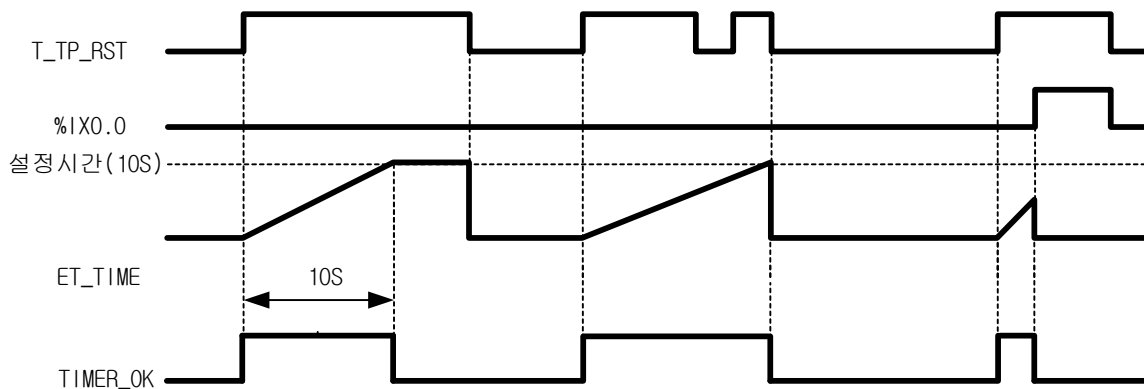
■ 프로그램 예

1. LD



2. ST

```
INST_TP_RST(IN:=T_TP_RST, PT:=T#10S, RST:=%IX0.0, Q=>TIMER_OK, ET=>ET_TIME);
```



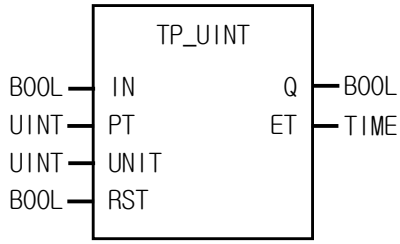
- (1) 입력변수 T_TP_RST 가 1 이 되면 출력변수 TIMER_OK 가 1 이 되고 10 초가 경과하면 출력변수 TIMER_OK 는 0 이 됩니다. 일단 타이머가 가동된 후에는 10 초 동안 T_TP_RST 신호는 무시됩니다.
- (2) ET_TIME 값은 증가 후 10S에서 멈춥니다. 그리고 T_TP_RST 가 0 이 될 때 0으로 됩니다.
- (3) 입력 접점 %IX0.0가 1이 되면 TIMER_OK와 ET_TIME 모두 0으로 클리어(Clear)됩니다.

☆ 참고

TP_RST 평선 블록은 접점이 On 된 후 Off 되어도 해당 동작이 마무리 될 때까지 평선 블록은 동작합니다.

배열인덱스를 이용한 변수를 사용했을 경우 배열인덱스 에러는 접점이 On 이 되었을 때만 발생합니다.

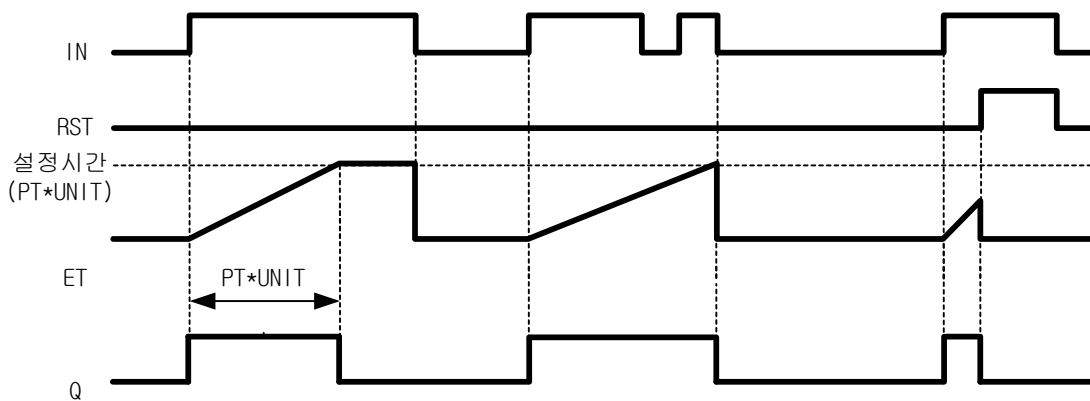
때문에 TP_RST 평선 블록에서는 평선 블록이 동작하더라도 접점이 Off 되어 있다면 배열인덱스 에러가 발생하지 않습니다.

TP_UINT	적용 기종	발생플래그
정수 설정 펄스 타이머	XGF-M32E	-
평선 블록	설 명	
	<p>입력</p> <ul style="list-style-type: none"> IN : 타이머 기동 조건 PT : 설정 시간(Preset Time) UNIT : 설정 시간의 시간 단위(Unit) RST : 리셋 입력 <p>출력</p> <ul style="list-style-type: none"> Q : 타이머 출력 ET : 경과 시간(Elapsed Time) 	

■ 기능

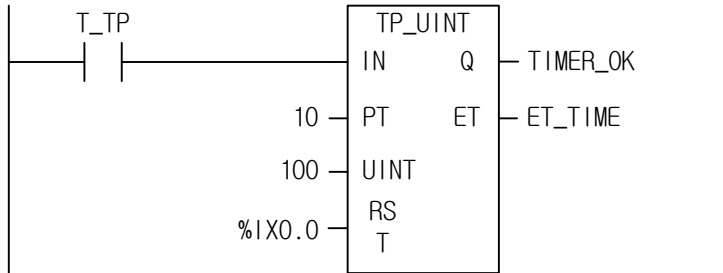
1. TP_UINT 평선 블록은 IN이 1이 되는 순간 Q는 1이 되고, 경과 시간이 설정 시간에 도달하면 타이머 출력 Q는 0이 됩니다.
2. 경과 시간 ET 는 IN 이 1 이 되었을 때부터 증가하며 PT 에 이르면 값을 유지하다가 IN 이 0 이 될 때 0 으로 클리어 (Clear) 됩니다.
3. 타이머 출력 Q가 1인 동안(펄스 출력 중)에는 타이머 기동 조건 IN이 1,0 변화를 하여도 무시합니다.
4. Reset 입력 조건이 성립하면 펄스 출력 중에도 타이머 출력 Q는 0이 되고 경과 시간도 0이 됩니다.
5. 설정시간은 $PT * UNIT[ms]$ 입니다.

■ 타임 차트



■ 프로그램 예

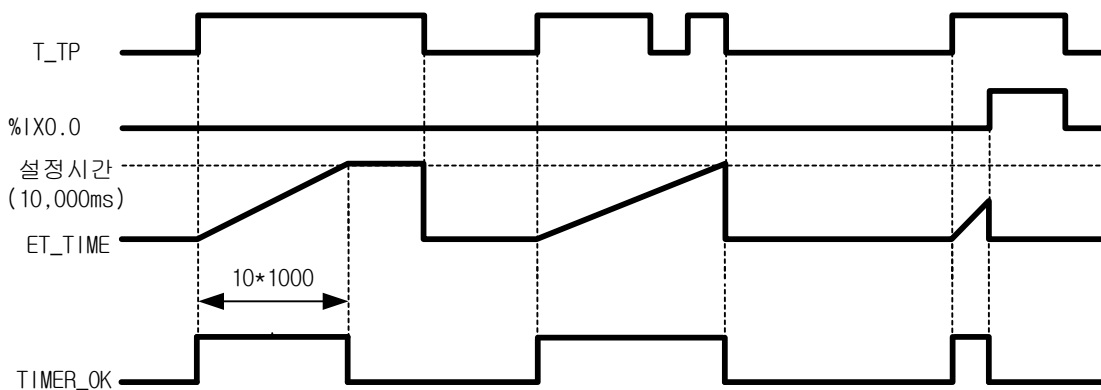
1. LD



2. ST

```
INST_TP_UINT(IN:=T_TP, PT:=10, UNIT:=100, RST:=%IX0.0, Q=>TIMER_OK, ET=>ET_TIME);
```

- (1) 설정 시간은 $PT * UNIT[ms] = 10 * 100[ms] = 1[s]$ 가 됩니다.
- (2) 입력변수 T_TP 가 1 이 되면 출력변수 TIMER_OK 가 1 이 되고 1 초가 경과하면 출력변수 TIMER_OK 는 0 이 됩니다. 일단 타이머가 가동된 후에는 1 초 동안 T_TP 신호는 무시됩니다.
- (3) ET_TIME 값은 증가 후 1,000 에서 멈춥니다. 그리고 T_TP 가 0 이 될 때 0 으로 됩니다.
- (4) 입력 접점 %IX0.0 가 1 이 되면 TIMER_OK 와 ET_TIME 모두 0 으로 클리어(Clear) 됩니다.



☆ 참고

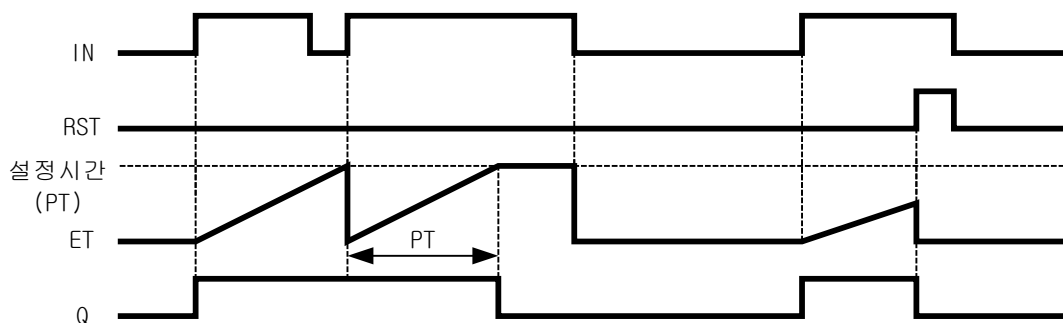
TP_UINT 평선 블록은 접점이 On 된 후 Off 되어도 해당 동작이 마무리 될 때까지 평선 블록은 동작합니다. 배열인덱스를 이용한 변수를 사용했을 경우 배열인덱스 에러는 접점이 On 이 되었을 때만 발생합니다. 때문에 TP_UINT 평선 블록에서는 평선 블록이 동작하더라도 접점이 off 되어 있다면 배열인덱스 에러가 발생하지 않습니다.

TRTG	적용 기종	발생플래그
리트리거블 타이머	XGF-M32E	-
평선 블록	설 명	
	<p>입력 IN : 타이머 기동 조건 PT : 설정 시간(Preset Time) RST : 리셋 입력(Reset)</p> <p>출력 Q : 타이머 출력 ET : 경과 시간(Elapsed Time)</p>	

■ 기능

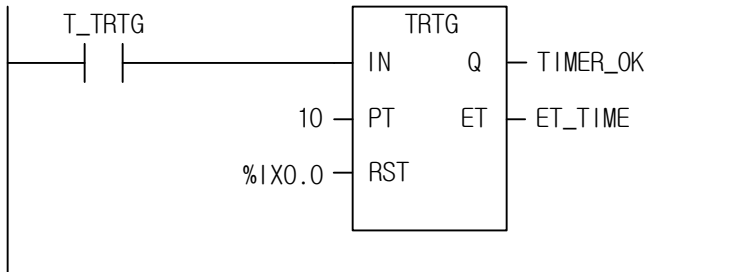
1. TRTG 평선 블록은 기동 조건 IN이 1이 되는 순간 Q는 1이 되고, 경과 시간이 설정 시간에 도달하면 타이머 출력 Q는 0이 됩니다.
2. 타이머 경과 시간이 설정 시간이 되기 전에 IN이 또 다시 0에서 1로 되면 경과 시간은 0으로 재설정 되고 다시 증가하여 설정 시간 PT에 도달하면 Q는 0이 됩니다.
3. Reset 입력 조건이 성립하면 타이머 출력 Q는 0이 되고 경과 시간도 0이 됩니다.

■ 타임 차트



■ 프로그램 예

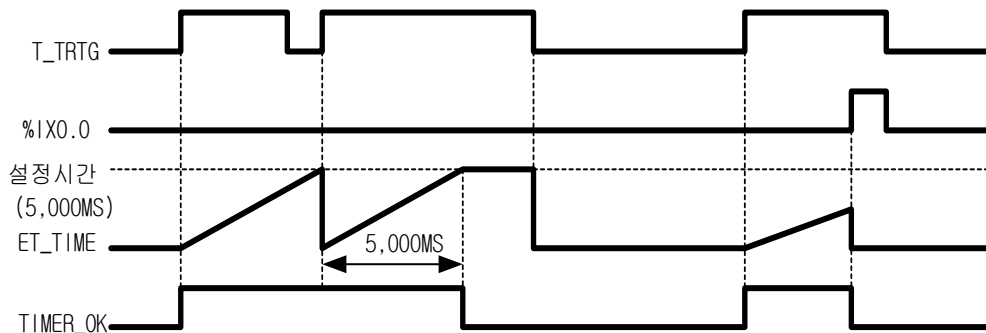
1. LD



2. ST

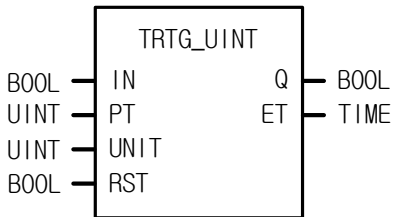
```
INST_TRTG(IN:=T_TRTG, PT:=10, RST:=%IX0.0, Q=>TIMER_OK, ET=>ET_TIME);
```

- (1) 입력변수 T_TRTG 가 0 에서 1 이 된 후 10 초 동안 TIMER_OK 는 1 이 됩니다. 타이머가 가동된 후 T_TRTG 가 다시 0 에서 1 이 되면 ET_TIME 은 0 부터 다시 시작됩니다.
- (2) T_TRTG 가 1 에서 0 이 되어도 10 초 동안 TIMER_OK 는 1 이 됩니다.
- (3) ET_TIME 은 증가 후 T#10S 에서 멈춥니다. 그리고 T_TRTG 가 0 이 될 때 0 으로 됩니다.
- (4) 입력접점 %IX0.0 가 1 이 되면 TIMER_OK 와 ET_TIME 모두 0 으로 클리어(Clear) 됩니다.



☆ 참고

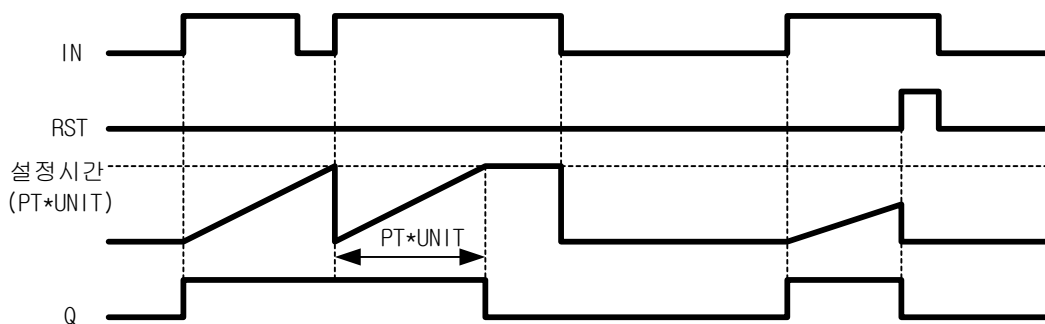
TRTG 평선 블록은 접점이 On 된 후 Off 되어도 해당 동작이 마무리 될 때까지 평선 블록은 동작합니다. 배열인덱스를 이용한 변수를 사용했을 경우 배열인덱스 에러는 접점이 On 이 되었을 때만 발생합니다. 때문에 TRTG 평선 블록에서는 평선 블록이 동작하더라도 접점이 Off 되어 있다면 배열인덱스 에러가 발생하지 않습니다

TRTG_UINT	적용 기종	발생플래그
정수 설정 리트리거블 타이머	XGF-M32E	-
평선 블록	설 명	
	입력 IN : 타이머 기동 조건 PT : 설정 시간(Preset Time) UNIT : 설정 시간의 시간 단위(Unit) RST : 리셋 입력	출력 Q : 타이머 출력 ET : 경과 시간(Elapsed Time)

■ 기능

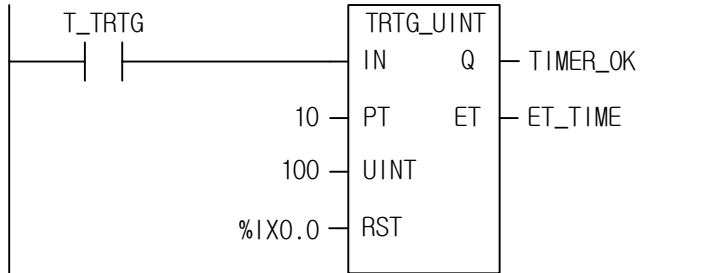
1. TRTG_UINT 평선 블록은 기동 조건 IN이 1이 되는 순간 Q는 1이 되고, 경과 시간이 설정 시간에 도달하면 타이머 출력 Q는 0이 됩니다.
2. 타이머 경과 시간이 설정 시간이 되기 전에 IN이 또 다시 0에서 1로 되면 경과 시간은 0으로 재설정 되고 다시 증가하여 설정 시간 PT에 도달하면 Q는 0이 됩니다.
3. Reset 입력 조건이 성립하면 타이머 출력 Q는 0이 되고 경과 시간도 0이 됩니다.
4. 설정 시간은 $PT * UNIT[ms]$ 입니다.

■ 타임 차트



■ 프로그램 예

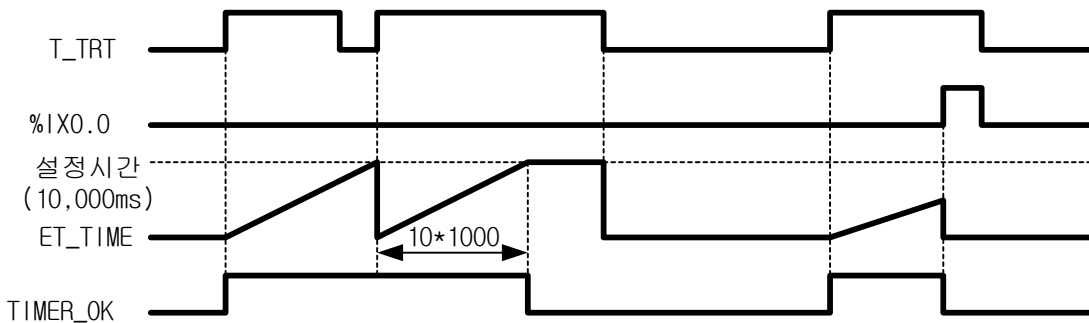
1. LD



2. ST

```
INST_TRTG_UINT(IN:=T_TRTG, PT:=10, UNIT:=100, RST:=%IX0.0, Q=>TIMER_OK, ET=>ET_TIME);
```

- (1) 설정 시간은 $PT * UNIT[ms] = 10 * 1000[ms] = 10[s]$ 가 됩니다.
- (2) 입력변수 T_TRTG 가 0 에서 1 이 된 후 10 초 동안 TIMER_OK 는 1 이 됩니다. 타이머가 가동된 후 T_TRTG 가 다시 0 에서 1 이 되면 ET_TIME 은 0 부터 다시 시작됩니다.
- (3) T_TRTG 가 1 에서 0 이 되어도 10 초 동안 TIMER_OK 는 1 이 됩니다.
- (4) ET_TIME 은 증가 후 10,000 에서 멈춥니다. 그리고 T_TRTG 가 0 이 될 때 0 으로 됩니다.
- (5) 입력접점 %IX0.0 이 1 이 되면 TIMER_OK 와 ET_TIME 모두 0 으로 클리어(Clear) 됩니다.



☆ 참고

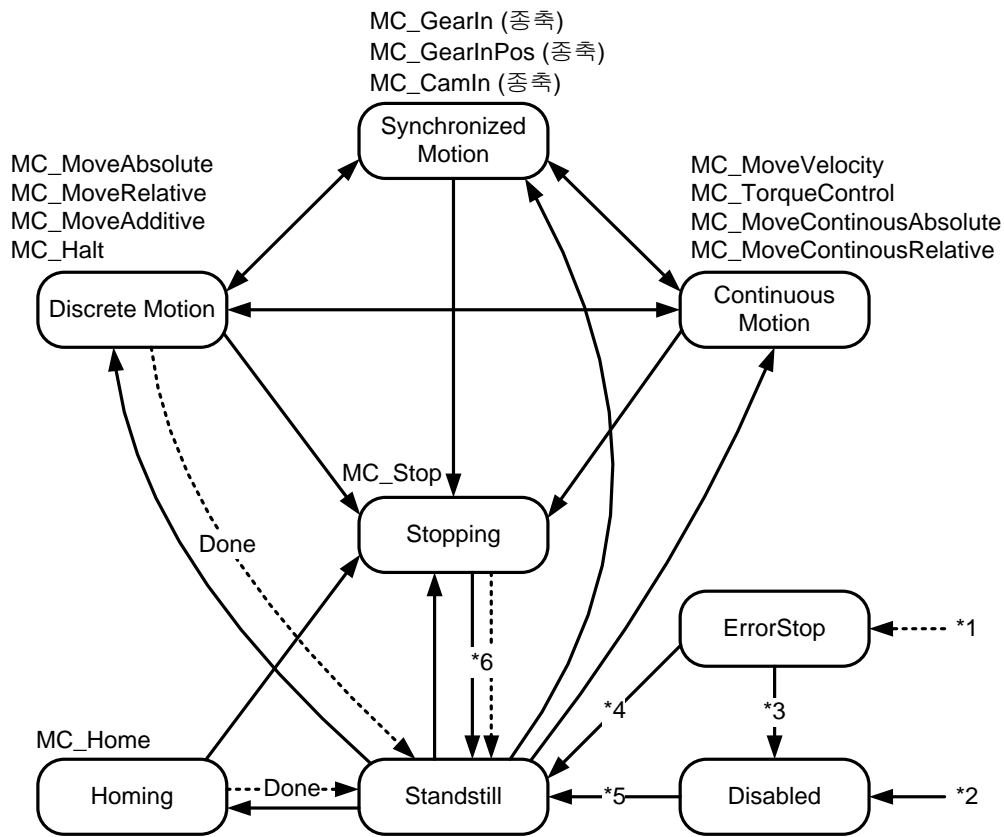
TRTG_UINT 평선 블록은 접점이 On 된 후 Off 되어도 해당 동작이 마무리 될 때까지 평선 블록은 동작합니다. 배열인덱스를 이용한 변수를 사용했을 경우 배열인덱스 에러는 접점이 On 이 되었을 때만 발생합니다. 때문에 TRTG_UINT 평선 블록에서는 평선 블록이 동작하더라도 접점이 Off 되어 있다면 배열인덱스 에러가 발생하지 않습니다.

제 10 장 모션 평선 블록

10.1. 모션 평선 블록의 공통 사항

10.1.1. 축 상태

모션 제어 모듈의 각 축은 상황과 명령에 따라 그에 해당하는 상태로 변화됩니다. 각 상태의 변화 구조는 아래 그림에 나타나 있습니다.



*1(에러 정지 상태): 현재 축이 어느 상태에 있든지, 축 에러가 발생하는 경우

*2(비활성화 상태): 축 에러가 발생하지 않았을 때, MC_Power.Enable 입력이 오프(Off) 되었을 경우

*3(에러 정지 -> 비활성화): MC_Power.Status 출력이 오프(Off) 상태일 때, MC_Reset 명령이 내려진 경우

*4(에러 정지 -> 운전 대기): MC_Power.Status 출력이 온(on), MC_Power.Enable 입력이 온(on) 상태일 때, MC_Reset 명령이 내려진 경우

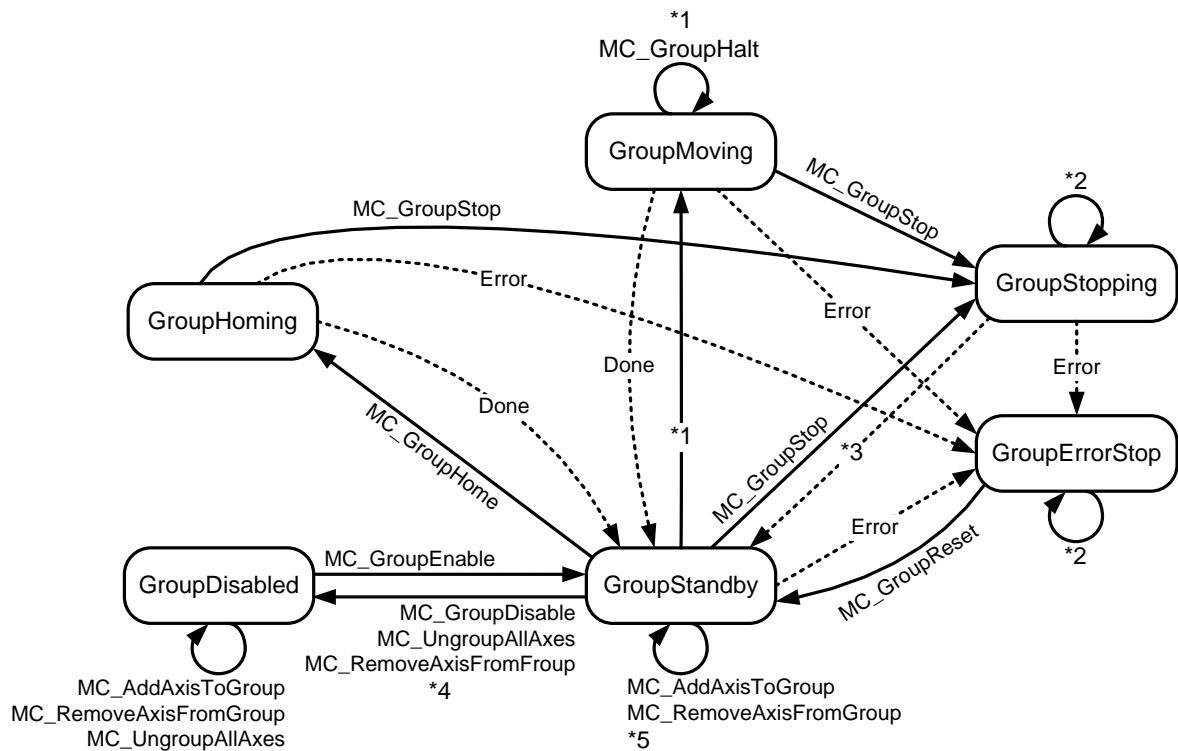
*5(비활성화 -> 운전 대기): MC_Power.Status 출력이 온(On) 상태일 때, MC_Power.Enable 입력을 온(On)시켰을 경우

*6(정지 중 -> 운전 대기): MC_Stop.Done 출력이 온(On) 상태일 때, MC_Stop.Execute 입력을 오프(Off)시켰을 경우

축 상태	설명
비활성화 (Disabled)	<p>비활성화(Disabled) 상태는 단일 축에 아무런 명령을 내리지 않았고 에러가 발생하지 않았을 때의 상태를 나타냅니다.</p> <p>처음 동작할 때 모션 제어 모듈이 에러가 없는 경우 각 축은 Disable 상태로 시작합니다.</p> <p>이후 서보 온/오프(MC_Power) 평션 블록의 Enable 입력을 온(On)하여 서보 온 상태가 되면 축 상태는 운전대기(Standstill) 상태로 변화합니다.</p> <p>에러정지(ErrorStop) 상태가 아닐 때 서보 온/오프(MC_Power) 평션 블록의 Enable 입력을 오프(Off) 시키면 축은 비활성화(Disabled) 상태가 됩니다.</p> <p>현재 수행중인 모션 평션 블록이 있을 경우 명령은 중단됩니다. (해당 모션 평션 블록의 CommandAborted 출력이 온(On)됩니다)</p>
에러정지 (ErrorStop)	<p>현재 축이 어느 상태에 있던지, 축 에러가 발생하면 에러정지(ErrorStop) 상태로 변경되며 해당 축은 감속정지 합니다.</p> <p>에러가 발생한 상태에서는 서보 온/오프(MC_Power) 명령을 실행해도 에러정지(ErrorStop) 상태를 유지합니다..</p> <p>에러정지(ErrorStop) 상태인 모션 축은 정지상태를 유지하며, 에러 리셋을 제외한 어떠한 명령도 수행하지 않습니다.</p>
운전대기 (StandStill)	<p>축의 전원이 활성화 되어있고, 축에 에러가 없으며 아무런 명령이 내려지지 않은 상태일 때 축의 상태는 운전대기(StandStill) 상태입니다.</p>
원점복귀중 (Homing)	<p>원점복귀중(Homing) 상태는 축이 원점복귀 운전 중인 상태를 나타냅니다.</p>
정지 (Stopping)	<p>비상정지(MC_Stop) 평션 블록을 수행하면 해당 축의 상태는 '정지중(Stopping)'상태로 바뀝니다. 해당 축이 '정지중(Stopping)'상태일 때는 정지가 완료되기 전까지(Done 출력이 활성화 되기 전 까지) 해당 축에 다른 모션 명령을 내릴 수 없습니다. Done 출력이 온(On)되고 Execute 입력이 온(On)되면 '운전대기(Standstill)'상태로 전환됩니다.</p>
연속운전 (Continuous Motion)	<p>현재 축이 운전 정지 상황이 될 때까지 계속 운전하는 상태입니다.</p>
개별운전 Discrete Motion	<p>목표위치가 있는 단축 운전 상태입니다.</p>
동기운전 Synchronized Motion	<p>동기운전(Synchronized Motion)은 축이 동기 운전 중인 상태를 나타냅니다.</p>

10.1.2. 그룹 상태

모션 제어 모듈의 각 그룹은 상황과 명령에 따라 그에 해당하는 상태로 변화됩니다. 각 상태의 변화 구조는 아래 그림에 나타나 있습니다

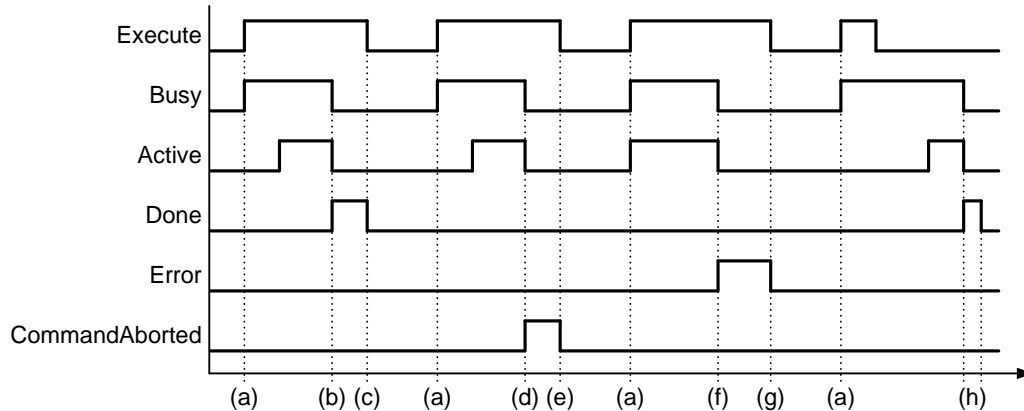


- *1(그룹 운전중): 일반 그룹 동작 모션 평선 블록을 수행했을 경우
- *2(그룹 정지중, 그룹 에러정지): 그룹 정지중(GroupStopping) 또는 그룹 에러정지(GroupErrorStop) 상태일 때 다른 모션 평선 블록을 수행했을 경우 해당 모션 평선 블록은 수행되지 않으며, 그룹 에러정지(GroupErrorStop) 상태일 때 그룹 에러리셋(MC_GroupReset) 평선 블록을 수행하면 해당 그룹의 상태는 그룹 대기(GroupStandby)로 전환됨
- *3(그룹 정지중 -> 그룹 대기): MC_GroupStop.DONE 출력이 온(On)되고, MC_SroupStop.EXECUTE 입력이 오프(Off) 상태일 때
- *4(그룹 대기 -> 그룹 비활성화): 그룹에서 축 제외 명령(MC_RemoveAxisFromGroup, MC_UnGroupAllAxes)을 수행한 결과 그룹에 속한 축이 하나도 없을 경우
- *5(그룹 대기): 그룹에서 축 추가 또는 제외 명령(MC_AddAxisToGroup, MC_RemoveAxisFromGroup)을 수행하여 그룹에 축이 하나 이상 속해있을 경우
- *6(그룹 비활성화): MC_GroupDisable 또는 MC_UnGroupAllDisable 평선 블록을 수행하면 그룹이 어떤 상태에 있든지 해당 그룹은 그룹 비활성화(GroupDisabled) 상태로 전환됨

10.1.3. 기본 입출력 변수

■ Edge 동작 모션 평선 블록

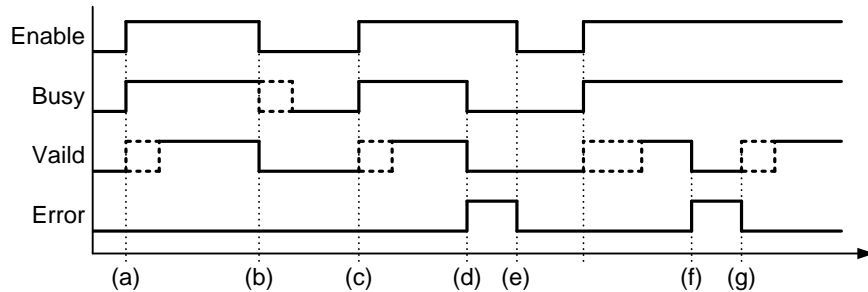
Edge 동작 모션 평선 블록에서의 기본 입출력 파라미터의 관계는 아래와 같습니다.



변수	설명
Execute	Edge 실행 평선 블록에서 해당 평선 블록을 실행시키는 입력입니다. 상승 Edge 에서(그림 a 상태) 평선 블록이 실행됩니다.
Busy	해당 모션 평선 블록이 현재 실행 중(=완료되지 않음)임을 나타내는 출력이며, 모션 평선 블록의 출력이 변화할 수 있음을 나타냅니다. Busy 출력은 Execute 입력의 상승 Edge 에서 온(On)되며(그림 a 상태) Done 출력이 온(On)되거나(그림 b 상태), CommandAborted 출력이 온(On)될 때(그림 d 상태), 또는 Error 출력이 온(On) 될 때(그림 f 상태) 오프(Off)됩니다.
Active	해당 모션 평선 블록이 실제로 축을 제어하고 있음을 나타냅니다. 한 축에 대해 여러 모션 평선 블록을 실행시켰을 때(한 모션 평선 블록만 제어 중이고 나머지 모션 평선 블록은 대기 중(Buffered)일 경우), 제어 중인 하나의 모션 평선 블록만 Active 출력이 온(On)되며 대기 중(Buffered)인 모션 평선 블록은 Busy 출력만 온(On)됩니다.
Done	해당 모션 평선 블록의 운전이 성공적으로 완료되었음을 나타내는 출력입니다. Done 출력이 온(On) 되면 Busy, Active 출력은 오프(Off)됩니다(그림 d 상태). Done 출력은 Execute 입력이 오프(Off)될 때 오프(Off) 되며(그림 e 상태), Done 출력이 온(On) 될 때 Execute 출력이 오프(Off) 상태였다면, 한 스캔 동안만 온(On)됩니다(그림 h 상태).
Error	모션 평선 블록 실행 중 에러가 발생했음을 나타내는 출력입니다. Error 출력은 Execute 입력이 오프(Off)될 때 오프(Off)됩니다(그림 f 상태). Error 출력이 온(On) 될 당시에 Execute 출력이 오프(Off) 상태였다면 한 스캔 동안만 온(On)됩니다(그림 h 상태).
ErrorID	모션 평선 블록 실행 중 에러가 발생했을 경우 해당 에러에 대한 에러코드를 출력합니다. 에러 코드 출력 및 소거 시점은 Error 출력과 동일합니다.
CommandAborted	해당 모션 평선 블록이 다른 모션 평선 블록에 의해 중단되었음을 나타냅니다. CommandAborted 출력은 Execute 입력이 오프(Off)될 때 같이 오프(Off) 됩니다(그림 g 상태). Done 출력이 온(On) 될 때 Execute 출력이 오프(Off) 상태였다면 한 스캔 동안만 온(On)됩니다.

※ Edge 동작(Execute 입력) 모션 평선 블록에서 Execute 입력이 온(On)되면 축의 상태에 따라 Busy, Done, Error, CommandAborted 출력 중 한 출력이 온(On)됩니다. Busy, Done, Error, CommandAborted 출력은 한번에 한 출력만 온(On) 가능하며, 빛 중 한 출력이 온(On)되면 다른 세 출력들은 오프(Off)됩니다.

■ 레벨 동작 모션 평선 블록



변수	설명
Enable	레벨 동작 모션 평선 블록을 실행시키는 입력입니다. 상승 Edge 에서(그림 a 상태) 모션 평선 블록을 실행시키고 하강 Edge(그림 b 상태)에서 정지시킵니다.
Busy	해당 모션 평선 블록이 현재 실행 중(=완료되지 않음)임을 나타내는 출력이며, 모션 평선 블록의 출력이 변화할 수 있음을 나타냅니다. Busy 출력은 Enable 입력의 상승 Edge 에서 온(On)되며(그림 b 상태) 모션 평선 블록이 동작하는 동안 온(On) 상태를 유지합니다.
Valid	해당 모션 평선 블록이 성공적으로 수행되어 출력 및 동작이 유효함을 나타내는 출력입니다. Valid 출력은 Enable 입력이 오프(Off)될 때 같이 오프(Off) 됩니다(그림 b 상태).
Error	모션 평선 블록 실행 중 에러가 발생했음을 나타내는 출력입니다. 모션 평선 블록 동작 중 자동적으로 복구할 수 없는 에러가 발생하면 Error 출력이 온(On)되고 Busy 출력과 Valid 출력을 오프(Off)시키며(그림 d 상태), 모션 평선 블록은 동작이 중지됩니다. Error 출력은 Enable 입력이 오프(Off)될 때 같이 오프(Off) 됩니다(그림 e 상태). 평선 블록 동작 중 자동적으로 복구 가능한 에러가 발생하면 Error 출력이 온(On)되고 Valid 입력만이 오프(Off) 됩니다(그림 f 상태). 해당 모션 평선 블록의 에러가 복구되면 Error 출력은 오프(Off)되고, 동작을 재개합니다(그림 g 상태).
ErrorID	모션 평선 블록 실행 중 에러가 발생했을 경우 해당 에러에 대한 에러코드를 출력합니다. 에러 코드 출력 및 소거 시점은 Error 출력과 동일합니다.

※ Valid 와 Error 출력은 동시에 온(On)되지 않습니다.

■ Axis 입력

각 모션 평선 블록은 해당 명령의 대상이 될 축을 Axis 입력으로 지정 가능합니다. 모션 제어 모듈은 실제 축 1~32 축, 가상축 37~40 축을 제어 가능하며 모션 평선 블록에 따라 엔코더 41~42 를 주축으로 사용할 수도 있습니다. 따라서 Axis 입력에는 모션 평선 블록에 따라 1~32, 37~40, 41~42 의 값 입력이 가능합니다. 각 모션 평선 블록에서 설정할 수 있는 범위를 벗어났을 경우 “에러 0x0006” 이 발생합니다.

10.1.4. BufferMode 입력

어떤 축에 모션 평선 블록을 실행할 때 축이 이미 다른 모션 평선 블록을 실행중인 경우 기존 모션 평선 블록을 취소시키고 명령을 수행할지, 기존 명령이 완료되는 것을 기다릴지 여부를 지정할 수 있는 입력입니다. 0~5 까지의 수를 지정 가능하며 범위를 벗어났을 경우 축 명령의 경우 “에러 0x101A” , 축그룹 명령의 경우 “에러 0x201A”가 발생합니다. BufferMode 에 설정 가능한 값은 아래와 같습니다.

번호	버퍼 모드	설명
0	mcAborting	명령을 즉시 실행시킵니다. 기존 운전 중인 명령은 중단됩니다.
1	mcBuffered	기존 운전 중인 명령이 완료된 이후에 명령을 실행시킵니다.
2	mcBlendingLow	기존 명령과 내릴 명령의 속도를 비교하여 속도가 낮은 쪽으로 조합시키는 조합운전을 합니다.
3	mcBlendingPrevious	기존 명령의 속도로 조합시키는 조합운전을 합니다.
4	mcBlendingNext	내릴 명령의 속도로 조합시키는 조합운전을 합니다.
5	mcBlendingHigh	기존 명령과 내릴 명령의 속도를 비교하여 속도가 높은 쪽으로 조합시키는 조합운전을 합니다.

10.1.5. 모션 평선 블록 실행 중 파라미터 변경

모션 평선 블록이 실행 중인 시점에서 해당 명령의 파라미터를 변경할 수 있으며, 자세한 동작은 아래와 같습니다.

- Edge 동작 모션 평선 블록을 ContinuousUpdate입력이 오프(Off)인 상태에서 실행(Execute입력을 온(On))했을 경우, 해당 모션 평선 블록은 Execute입력이 온(On)될 당시(상승 Edge)의 파라미터가 적용되어 동작합니다. 이 경우 모션 평선 블록의 실행 중간에 파라미터 입력 값을 변경시켜도 동작에 영향을 주지 않습니다. 해당 모션 평선 블록이 동작하는 중에 파라미터를 변경하려면 파라미터를 변경 후 Execute 입력을 다시 온(On)시켜주면 됩니다.
- Edge 동작 모션 평선 블록을 ContinuousUpdate입력이 온(On)인 상태에서 실행(Execute입력을 온(On)) 시키면 최초에는 Execute입력이 온(On) 될 당시(상승 Edge)의 파라미터가 적용됩니다. ContinuousUpdate 입력이 온(On) 되어 있는 동안 파라미터를 변경하면 해당 모션 평선 블록은 파라미터가 바뀔 때마다 이를 반영하여 동작합니다. 단, 해당 모션 평선 블록의 동작이 완료되거나 멈춘 이후(Busy 출력이 오프(Off))에 파라미터를 변경하면 더 이상 반영되지 않습니다.(ContinuousUpdate 를 이용한 파라미터 변경 동작은 완료되거나 중단된 모션 평선 블록을 재실행시키지 않습니다. 즉, ContinuousUpdate 동작은 현재 실행되고 있는 모션 평선 블록에만 적용됩니다.)
- 레벨 동작 모션 평선 블록의 경우, Enable입력이 온(On)될 당시(상승 Edge)의 파라미터들이 적용되어 동작하고, Enable입력이 온(On) 된 동안 연속적으로 파라미터 변경이 가능합니다.

10.1.6. 그룹운전 경로 변경 설정

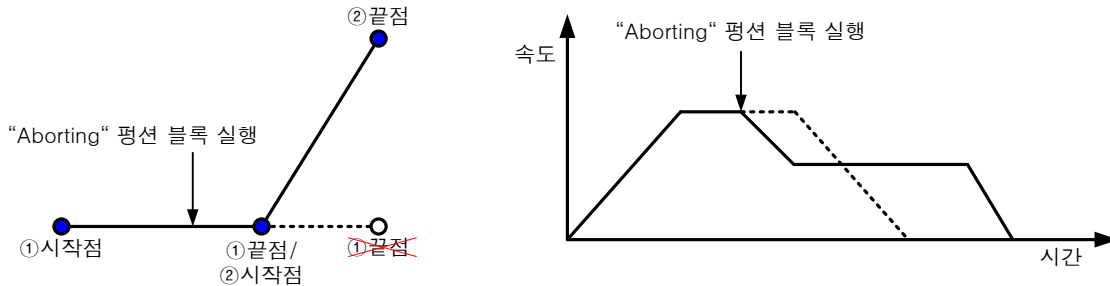
현재 모션 제어 모듈의 축 그룹이 명령을 수행 중일 때, 해당 축 그룹에 다른 명령을 내릴 수 있습니다. 이 때, 다음 명령이 이를 궤적이 기존 명령이 그리고 있던 궤적과 어떤 식으로 연결될지를 지정할 수 있습니다. 연결 궤적의 파라미터는 TransitionParameter 입력에 지정합니다.

번호	TRANSITION 모드	설명
0	TMNone	연결 궤적을 생성하지 않습니다.
3	TMCornerDistance	연결 궤적의 반지름을 지정하여 지정한 반지름의 원을 그리는 연결 궤적을 생성합니다.

■ TransitionMode “TMNone”

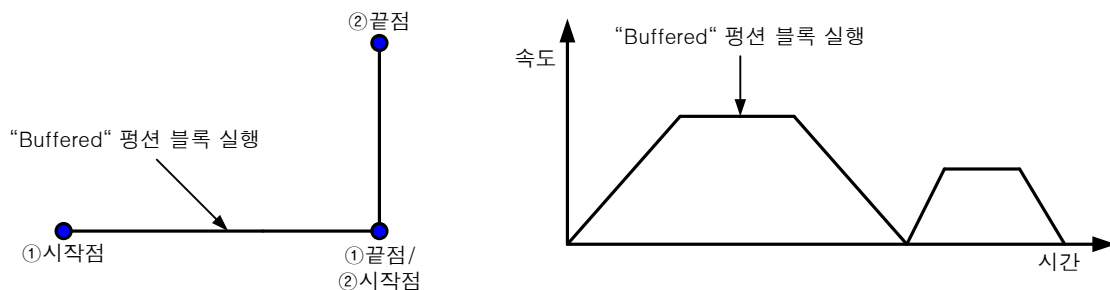
연결 궤적을 생성하지 않습니다. 모션 평선 블록의 BufferMode 입력이 “Aborting” 이거나 “Buffered” 일 경우엔 TransitionMode 입력은 “TMNone” 만 가능합니다.

아래 그림은 모션 평선 블록의 BufferMode 를 ‘Aborting’으로 설정하여 실행하였을 경우를 보여줍니다. 왼쪽 그림을 보면 ①번 모션 평선 블록의 실행 중에 ②번 모션 평선 블록이 ‘Aborting’으로 실행되었습니다. ①번 모션 평선 블록이 ‘①끝점’에 도달하지 못한 상태에서 ‘①끝점/②시작점’에서 강제 종료되었습니다. 오른쪽 그림을 보면 ‘Aborting’ 평선 블록이 실행된 순간 감속정지를 하고 다음 모션 평선 블록을 실행합니다.



<BufferMode 를 “Aborting” 으로 지정했을 때>

아래 그림은 모션 평선 블록의 BufferMode 를 ‘Buffered’로 설정하여 실행했을 경우를 보여줍니다. 왼쪽 그림을 보면 ①번 모션 평선 블록이 실행 중에 ②번 모션 평선 블록이 ‘Buffered’로 실행되었습니다. ①번 모션 평선 블록이 목표위치에 도달한 이후에 ②번 모션 평선 블록을 수행합니다. 오른쪽 그림을 보면 ‘Buffered’ 평선 블록이 실행되면 원래의 목표위치까지 도달한 이후에 다음 모션 평선 블록을 실행합니다.



<BufferMode 를 “Buffered” 으로 지정했을 때>

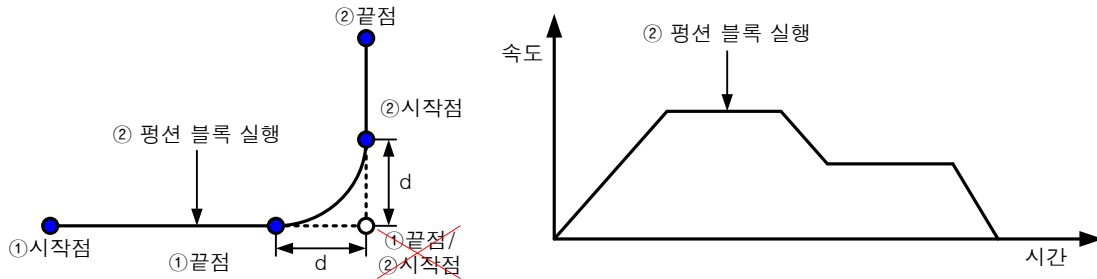
■ TransitionMode “TMCornerDistance”

연결 궤적의 반지름을 지정하여 지정한 반지름의 원을 그리는 연결 궤적을 생성합니다. BufferMode 가 “BlendingXXXX” 일 때만 이 모드가 동작하며, BufferMode 가 “Aborting” 또는 “Buffered” 일 경우엔 “TMNone” 으로 동작합니다.

연결 궤적을 그릴 때의 경로의 최대 속도는 BufferMode 에 지정된 바를 따르며 이 때, 반지름의 길이는 TransitionParameter 에 지정한 값을 따릅니다.

아래 그림은 두 개의 직선보간 명령에 반지름 원을 그리는 연결 궤적이 생성된 것을 보여줍니다. 왼쪽 그림을 보면 ①번 모션 평선 블록 실행 중에 ②번 모션 평선 블록이 “TMcornerDistance”로 설정하여 실행되었습니다. ①번 모션 평선 블록의 원래 목표위치는 ①끝점/②시작점 이었으나 반지름 ‘d’ 만큼 앞 지점(①끝점)에서 직선 운동을 정지하고 원호 운동을 시작합니다. 원호 운동은 ①끝점에서 시작하여 ②시작점 에서 완료되며 ②시작점부터 ②번 모션 평선 블록을 수행합니다.

오른쪽 그림은 두 평선 블록의 중간에 속도가 정지하지 않고 연속적으로 이어지는 것을 보여줍니다.



<BufferMode 를 “BlendingLow” 으로, TransitionMode 를 “TMCornerDistance” 로 지정했을 때>

10.1.7. 모션 평선 블록 공통 에러

모션 평선 블록의 ErrorID 변수에 발생하는 에러는 다음과 같습니다.

STAT	내용	상세 설명
0x0000	정상	모션 평선 블록이 정상적으로 실행된 경우 ErrorID 에 “0”이 출력됩니다.
0x0005	현재 모션 모듈이 해당 모션 평선 블록을 지원하지 않습니다.	해당 모션 평선 블록은 현재 모듈의 버전에서는 수행되지 않습니다. 해당 모션 평선 블록이 수행될 수 있는 버전을 확인하시기 바랍니다..
0x0006	모션 평선 블록의 축 번호 (Axis 입력)가 허용 범위를 초과하였습니다.	모션 평선 블록 별로 할당할 수 있는 축을 확인하고, 축 번호를 1~32, 37~40 의 영역으로 설정하여 주십시오.
0x0007	모션 평선 블록의 축 그룹 번호 (AxisGroup 입력)가 허용 범위를 초과하였습니다.	축 그룹 번호를 1~16 사이의 값으로 설정하여 주십시오.
0x0012	모션 평선 블록 실행 도중 모션 평선 블록 내부 실행 오류가 발생하였습니다.	XG-PM 및 XGF-M32E 의 버전을 확인하여 주십시오.
0x0013	모션 평선 블록 실행 도중 모션 응답 오류가 발생하였습니다.	XG-PM 및 XGF-M32E 의 버전을 확인하여 주십시오.
0x0020 : 0x0FFF	모션 제어 모듈의 공통 에러 입니다. 자세한 내용은 ‘부록 1 장 에러 정보 및 조치 방법’을 참조하여 주십시오.	
0x1000 : 0x1FFF	모션 제어 모듈의 축 제어 관련 발생 에러 입니다. 자세한 내용은 ‘부록 1 장 에러 정보 및 조치 방법’을 참조하여 주십시오.	
0x2000 : 0x2FFF	모션 제어 모듈의 축그룹 제어 관련 발생 에러 입니다. 자세한 내용은 ‘부록 1 장 에러 정보 및 조치 방법’을 참조하여 주십시오.	

10.2. 모션 평선 블록

MC_Power		적용 기종
서보 온/오프		XGF-M32E
모션 평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_Power Enable[Enable] Axis[Axis] Status[Status] Axis2[Axis] Valid[Valid] Error[Error] ErrorID[ErrorID] end Enable --- Status Axis --- Axis2 Axis2 --- Valid Axis2 --- Error Axis2 --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 축의 서보 모터가 서보 온 됩니다
출력		
BOOL	Status	해당 축의 전원 인가상태를 나타냅니다.
BOOL	Valid	모션 평선 블록 출력의 유효성을 나타냅니다. (여기선 Status 출력과 동일합니다.)
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 서보 온/오프 지령을 내리는 모션 평선 블록입니다.
- (2) Enable 입력 On 일 때 해당 축에 서보 온 지령이, Off 일 때 서보 오프 지령이 내려지게 됩니다.
- (3) 축이 '비활성화(Disable)'상태일 때 서보 온 지령이 수행되면 축 상태는 '운전대기(StandStill)'이 되고, 서보 온에 실패하면 '에러정지(ErrorStop)' 상태가 됩니다.

MC_Home		적용 기종
원점 복귀 기능		XGF-M32E
모션 평선 블록 형태		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 원점복귀 운전을 시작합니다.
LREAL	Position	기준신호가 감지되었을 때의 축 절대위치를 지정합니다. (명령 위치 값을 펄스로 환산했을 때 -2147483648 ~ 2147483647 범위에서 설정)
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 해당 축을 제어중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 다른 명령에 의해 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 원점복귀 지령을 내리는 모션 평선 블록입니다.
- (2) 원점 복귀 방식은 사전에 해당 축의 운전 파라미터에 지정된 대로 동작합니다.
- (3) Position 입력에는 기준 신호(Reference Signal)이 감지되었을 때, 또는 원점복귀가 완료되었을 때 축의 절대 위치를 지정합니다.

- (4) 이 모션 평선 블록이 수행 중일 때 축은 '원점복귀중(Homing)' 상태이며, 명령이 완료되면 '운전대기(Standstill)' 상태로 전환됩니다
- (5) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.
Position 입력만 업데이트 가능합니다.

MC_Stop		적용 기종
즉시 정지		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_Stop Execute[Execute] Axis[Axis] Deceleration[Deceleration] Jerk[Jerk] Done[Done] Busy[Busy] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done Axis --- Axis Deceleration --- Deceleration Jerk --- Jerk Done --- Done Busy --- Busy CommandAborted --- CommandAborted Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 즉시 정지 지령을 내립니다.
LREAL	Deceleration	정지 시의 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	감속도 변화율을 지정합니다. [u/s ³]
출력		
BOOL	Done	해당 축의 속도가 0 에 도달했음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중단되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 비상 정지 지령을 내리는 모션 평선 블록입니다.
- (2) 즉시 정지(MC_Stop) 모션 평선 블록을 수행하면 해당 축에 기존에 수행하고 있던 모션 평선 블록은 중단되며, 축 상태는 '정지중(Stopping)'상태로 바뀝니다. 해당 축이 '정지중(Stopping)'상태일 때에는 정지가 완료되기 전 까지(Done 출력이 활성화 되기 전까지) 해당 축에 다른 모션 평선 블록을 실행 시킬 수 없습니다.
- (3) CommandAborted 출력은 현재 모션 평선 블록이 실행 중 중단되었음을 나타냅니다. 즉시 정지(MC_Stop) 모션 평선 블록 실행 중에는 다른 모션 평선 블록이 즉시 정지(MC_Stop) 모션 평선 블록을 중단시킬 수 없으므로, 일반적으로 서보의 전원이 차단되거나 서보 오프 지령이 실행되었을 경우에 CommandAborted 출력이 온(On)됩니다.

- (4) Execute 입력이 온(On) 되어 있거나, 축의 속도가 0 이 아니라면 축은 '정지중(Stopping)'상태이며, Done 출력이 온(On) 되고 Execute 입력이 오프(Off) 되면 '운전대기중(Standstill)'상태로 전환됩니다.
- (5) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.
Deceleration, Jerk 입력만 업데이트 가능합니다

MC_Halt		적용 기종
정지		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_Halt Execute[Execute] Axis[Axis] Deceleration[Deceleration] Jerk[Jerk] BufferMode[BufferMode] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done Axis --- Axis Deceleration --- Deceleration Jerk --- Jerk BufferMode --- BufferMode Done --- Done Busy --- Busy Active --- Active CommandAborted --- CommandAborted Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 정지 지령을 내립니다.
LREAL	Deceleration	정지 시의 감속도를 지정합니다 [u/s ²]
LREAL	Jerk	가감속도 변화율을 지정합니다 [u/s ³]
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
출력		
BOOL	Done	해당 축의 속도가 0 에 도달했음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 정지 지령을 내리는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록이 수행 중일 때 축은 '개별운전(DiscreteMotion)'상태이며, 해당 축의 속도가 0 이 되면 'Done'출력이 온(On)되고 '운전대기(Standstill)'상태로 바뀝니다.

- (3) MC_Stop 명령과는 다르게 BufferMode 설정이 가능하며, 다른 모션 평선 블록에 의해서 정지 명령(MC_Halt)이 중단 될 수 있습니다.
- (4) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.
Deceleration, Jerk 입력만 업데이트 가능합니다.

MC_MoveAbsolute		적용 기종
절대위치 운전		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph TD subgraph MC_MoveAbsolute Execute[Execute] Axis[Axis] ContinuousUpdate[ContinuousUpdate] Position[Position] Velocity[Velocity] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] Direction[Direction] BufferMode[BufferMode] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done Axis --- Axis ContinuousUpdate --- Busy Position --- Active Velocity --- CommandAborted Acceleration --- Error Deceleration --- ErrorID Jerk --- ErrorID Direction --- ErrorID BufferMode --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 절대위치 운전 지령을 내립니다.
BOOL	ContinuousUpdate	입력 값의 운전 중 파라미터 변경 방식을 지정합니다. (10.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)
LREAL	Position	목표 위치를 지정합니다.
LREAL	Velocity	최대 속도를 지정합니다. [u/s]
LREAL	Acceleration	가속도를 지정합니다. [u/s ²]
LREAL	Deceleration	감속도를 지정합니다. [u/s ²]
LREAL	Jerk	가감속도 변화율을 지정합니다 [u/s ³]
UINT	Direction	운전 방향을 지정합니다. (0 ~ 4 : 0: 지정없음, 1-정방향, 2-최단거리, 3-역방향, 4-현재방향)
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
출력		
BOOL	Done	지정한 위치에 도달 완료 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.

BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 절대위치 운전 지령을 내리는 모션 평선 블록입니다.
- (2) **Direction** 입력에는 무한길이 반복 운전 시 축의 운전 방향을 설정하며, 무한길이 반복 운전이 금지로 설정된 경우에는 **Direction** 입력은 무시됩니다. **Direction** 입력이 최단거리(=2)일 경우, 해당 축이 무한길이 반복 운전을 한다면 자동적으로 최단거리로 갈 수 있는 방향을 선택하여 운전합니다. 0~4(0-지정없음, 1-정방향, 2-최단거리, 3-역방향, 4-현재방향)범위로 지정 가능하며 범위 초과시 “에러 0x1017” 이 발생합니다.
- (3) 현재 모션 평선 블록 이후에 대기중인 모션 평선 블록이 없는 상태에서 목표 위치 도달 후 속도가 0이 되면 운전은 완료되고 **Done**출력이 온(On)됩니다
- (4) 이 모션 평선 블록이 실행 중일 때 축은 ‘개별운동(DiscreteMotion)’상태이며, 운전이 완료되면 ‘운전대기(Standstill)’상태로 전환됩니다.
- (5) 명령이 종료되기 전에 평선 블록을 재 실행(Execute입력을 On) 시키면 변경된 파라미터가 적용됩니다. Position, Velocity, Acceleration, Deceleration, Jerk, Direction입력만 업데이트 가능합니다.

MC_MoveRelative		적용 기종																																								
상대위치 운전		XGF-M32E																																								
평선 블록 형태																																										
<div style="text-align: center;"> <table border="1"> <thead> <tr> <th colspan="4">MC_MoveRelative</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>Done</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td>Axis</td> <td>UINT</td> </tr> <tr> <td>BOOL</td> <td>ContinuousUpdate</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Distance</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </tbody> </table> </div>			MC_MoveRelative				BOOL	Execute	Done	BOOL	UINT	Axis	Axis	UINT	BOOL	ContinuousUpdate	Busy	BOOL	LREAL	Distance	Active	BOOL	LREAL	Velocity	CommandAborted	BOOL	LREAL	Acceleration	Error	BOOL	LREAL	Deceleration	ErrorID	WORD	LREAL	Jerk			UINT	BufferMode		
MC_MoveRelative																																										
BOOL	Execute	Done	BOOL																																							
UINT	Axis	Axis	UINT																																							
BOOL	ContinuousUpdate	Busy	BOOL																																							
LREAL	Distance	Active	BOOL																																							
LREAL	Velocity	CommandAborted	BOOL																																							
LREAL	Acceleration	Error	BOOL																																							
LREAL	Deceleration	ErrorID	WORD																																							
LREAL	Jerk																																									
UINT	BufferMode																																									
입력-출력																																										
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)																																								
입력																																										
BOOL	Execute	상승 Edge 에서 해당 축에 상대위치 운전 지령을 내립니다.																																								
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (10.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)																																								
LREAL	Distance	목표거리를 지정합니다.																																								
LREAL	Velocity	최대속도를 지정합니다. [u/s]																																								
LREAL	Acceleration	가속도를 지정합니다. [u/s ²]																																								
LREAL	Deceleration	감속도를 지정합니다. [u/s ²]																																								
LREAL	Jerk	가감속도 변화율을 지정합니다 [u/s ³]																																								
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)																																								
출력																																										
BOOL	Done	지정한 거리 도달 완료 여부를 나타냅니다.																																								
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.																																								
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다																																								
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.																																								
BOOL	Error	에러 발생 여부를 나타냅니다.																																								

WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
------	---------	--------------------------------

- (1) 이 모션 평선 블록은 해당 축에 상대위치 운전 지령을 내리는 모션 평선 블록입니다.
- (2) 상대위치 운전(MC_MoveRelative)은 현재위치를 기준으로 Distance 입력에 지정된 목표거리만큼 이동하는 모션 평선 블록입니다.
- (3) 이동 방향은 Distance 입력에 지정된 목표거리의 부호에 따라 결정되며 이동 방향이 양(+ 혹은 부호없음)일 경우 정방향, 음(-)일 경우는 역방향으로 이동하게 됩니다.
- (4) 현재 모션 평선 블록 이후에 대기 중인 모션 평선 블록이 없는 상태에서 목표거리 이동 후 속도가 0 이 되면 운전은 완료되고 Done 출력이 온(On) 됩니다.
- (5) 이 모션 평선 블록이 실행 중일 때 축은 “개별운동(DiscreteMotion)” 상태이며, 운전이 완료되면 “운전대기 (StandStill)” 상태로 전환됩니다.
- (6) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.
Distance, Velocity, Acceleration, Deceleration, Jerk 입력만 업데이트 가능합니다.

MC_MoveAdditive		적용 기종																														
추가 위치 운전		XGF-M32E																														
평선 블록 형태																																
<div style="text-align: center;"> <table border="1" style="margin: auto;"> <thead> <tr> <th colspan="3">MC_MoveAdditive</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>Done</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td>Axis</td> </tr> <tr> <td>BOOL</td> <td>ContinuousUpdate</td> <td>Busy</td> </tr> <tr> <td>LREAL</td> <td>Distance</td> <td>Active</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>CommandAborted</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td>Error</td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td>ErrorID</td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> </tr> </tbody> </table> </div>			MC_MoveAdditive			BOOL	Execute	Done	UINT	Axis	Axis	BOOL	ContinuousUpdate	Busy	LREAL	Distance	Active	LREAL	Velocity	CommandAborted	LREAL	Acceleration	Error	LREAL	Deceleration	ErrorID	LREAL	Jerk		UINT	BufferMode	
MC_MoveAdditive																																
BOOL	Execute	Done																														
UINT	Axis	Axis																														
BOOL	ContinuousUpdate	Busy																														
LREAL	Distance	Active																														
LREAL	Velocity	CommandAborted																														
LREAL	Acceleration	Error																														
LREAL	Deceleration	ErrorID																														
LREAL	Jerk																															
UINT	BufferMode																															
입력-출력																																
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)																														
입력																																
BOOL	Execute	상승 Edge 에서 해당 축에 추가위치 운전 지령을 내립니다.																														
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (10.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)																														
LREAL	Distance	중점 거리를 지정합니다.																														
LREAL	Velocity	최대 속도를 지정합니다. [u/s]																														
LREAL	Acceleration	가속도를 지정합니다. [u/s ²]																														
LREAL	Deceleration	감속도를 지정합니다. [u/s ²]																														
LREAL	Jerk	가감속도 변화율을 지정합니다 [u/s ³]																														
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)																														
출력																																
BOOL	Done	지정한 거리 도달 완료 여부를 나타냅니다.																														
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.																														
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다																														
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.																														
BOOL	Error	에러 발생 여부를 나타냅니다.																														

WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
------	---------	--------------------------------

- (1) 이 모션 평선 블록은 해당 추가위치 운전 지령을 내리는 모션 평선 블록입니다.
- (2) 추가 위치 운전(MC_MoveAdditive)은 '개별운전(DiscreteMotion)'상태로 수행되었던 가장 최근의 모션 평선 블록 또는 현재 수행되고 있는 모션 평선 블록의 최종 목표 위치를 기준으로 Distance입력에 지정된 위치만큼 추가로 이동시키는 모션 평선 블록입니다. 현재 축이 '연속운전(ContinuousMotion)'상태로 모션 평선 블록을 수행 중이라면 추가 위치 운전(MC_MoveAdditive)이 수행될 때의 위치를 기준으로 운전을 수행하게 됩니다.
- (3) 이동 방향은 Distance입력에 지정된 목표거리의 부호에 따라 결정되며 이동 방향이 양(+ 또는 부호 없음)일 때는 정방향, 음(-)일 때는 역방향으로 이동하게 됩니다.
- (4) 현재 모션 평선 블록 이후에 대기중인 모션 평선 블록이 없는 상태에서 목표 위치에 도달하면 'Done'출력이 온(On)됩니다.
- (5) 이 모션 평선 블록이 실행 중일 때 축은 '개별운전(DiscreteMotion)'상태이며, 운전이 완료되면 '운전대기(Standstill)'상태로 전환됩니다.
- (6) 명령이 종료되기 전에 평선 블록을 재 실행(Execute입력을 On) 시키면 변경된 파라미터가 적용됩니다.

Distance, Velocity, Acceleration, Deceleration, Jerk 입력만 업데이트 가능합니다.

MC_MoveVelocity		적용 기종																																								
지정 속도 운전		XGF-M32E																																								
평선 블록 형태																																										
<div style="text-align: center;"> <table border="1" style="margin: auto;"> <thead> <tr> <th colspan="4">MC_MoveVelocity</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>Execute</td> <td>InVelocity</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td>Axis</td> <td>UINT</td> </tr> <tr> <td>BOOL</td> <td>ContinuousUpdate</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>UINT</td> <td>Direction</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </tbody> </table> </div>			MC_MoveVelocity				BOOL	Execute	InVelocity	BOOL	UINT	Axis	Axis	UINT	BOOL	ContinuousUpdate	Busy	BOOL	LREAL	Velocity	Active	BOOL	LREAL	Acceleration	CommandAborted	BOOL	LREAL	Deceleration	Error	BOOL	LREAL	Jerk	ErrorID	WORD	UINT	Direction			UINT	BufferMode		
MC_MoveVelocity																																										
BOOL	Execute	InVelocity	BOOL																																							
UINT	Axis	Axis	UINT																																							
BOOL	ContinuousUpdate	Busy	BOOL																																							
LREAL	Velocity	Active	BOOL																																							
LREAL	Acceleration	CommandAborted	BOOL																																							
LREAL	Deceleration	Error	BOOL																																							
LREAL	Jerk	ErrorID	WORD																																							
UINT	Direction																																									
UINT	BufferMode																																									
입력-출력																																										
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)																																								
입력																																										
BOOL	Execute	상승 Edge 에서 해당 축에 지정 속도 운전 지령을 내립니다.																																								
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (10.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)																																								
LREAL	Velocity	운전 속도를 지정합니다. [u/s]																																								
LREAL	Acceleration	가속도를 지정합니다. [u/s ²]																																								
LREAL	Deceleration	감속도를 지정합니다. [u/s ²]																																								
LREAL	Jerk	가/감속도 변화율을 지정합니다 [u/s ³]																																								
UINT	Direction	운전 방향을 지정합니다. (1 ~ 3 : 1-정방향, 2-역방향, 3-현재방향)																																								
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)																																								
출력																																										
BOOL	InVelocity	지정 속도 도달 여부를 나타냅니다																																								
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.																																								
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다																																								
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.																																								
BOOL	Error	에러 발생 여부를 나타냅니다.																																								

WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
------	---------	--------------------------------

- (1) 이 모션 평선 블록은 해당 축에 지정 속도 운전 지령을 내리는 모션 평선 블록입니다.
- (2) 지정 속도 운전을 정지하기 위해서는, 정지 명령을 내리거나 다른 모션 평선 블록을 수행함으로써 가능합니다.
- (3) **Velocity** 입력에는 운전할 속도를 지정합니다. 운전 속도 값의 부호가 양(+ 또는 부호 없음)일 때는 정방향, 음(-)일 때는 역방향으로 이동하게 됩니다.
- (4) **Direction** 입력에는 운전할 방향을 지정합니다. 단, **Velocity** 입력으로 지정된 속도값의 부호에 따라 운전방향에 영향을 받습니다. 예를 들어 **Velocity** 값에 음수값을 지정하고, **Direction** 입력에 역방향을 지정했다면 최종적으로 해당 축은 정방향 운전을 하게 됩니다.
- (5) 출력 **InVelocity** 는 해당 축이 지정한 속도에 도달했을 때 온(On)되며, 지정속도 운전이 중단되면 오프(Off)됩니다.
- (6) 이 모션 평선 블록이 실행 중일 때 축은 '연속운동(ContinuousMotion)' 상태입니다.
- (7) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.

Velocity, Acceleration, Deceleration, Jerk, Direction 입력만 업데이트 가능합니다.

MC_MoveContinuousAbsolute		적용 기종																																				
절대위치 운전 후 지정속도 운전		XGF-M32E																																				
평선 블록 형태																																						
<div style="text-align: center;"> <table border="1"> <tr> <th colspan="3">MC_MoveContinuousAbsolute</th> </tr> <tr> <td>BOOL</td> <td>Execute</td> <td>InEndVelocity</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td>Axis</td> </tr> <tr> <td>BOOL</td> <td>ContinuousUpdate</td> <td>Busy</td> </tr> <tr> <td>LREAL</td> <td>Position</td> <td>Active</td> </tr> <tr> <td>LREAL</td> <td>EndVelocity</td> <td>CommandAborted</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>Error</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td>ErrorID</td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> </tr> <tr> <td>UINT</td> <td>Direction</td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> </tr> </table> </div>			MC_MoveContinuousAbsolute			BOOL	Execute	InEndVelocity	UINT	Axis	Axis	BOOL	ContinuousUpdate	Busy	LREAL	Position	Active	LREAL	EndVelocity	CommandAborted	LREAL	Velocity	Error	LREAL	Acceleration	ErrorID	LREAL	Deceleration		LREAL	Jerk		UINT	Direction		UINT	BufferMode	
MC_MoveContinuousAbsolute																																						
BOOL	Execute	InEndVelocity																																				
UINT	Axis	Axis																																				
BOOL	ContinuousUpdate	Busy																																				
LREAL	Position	Active																																				
LREAL	EndVelocity	CommandAborted																																				
LREAL	Velocity	Error																																				
LREAL	Acceleration	ErrorID																																				
LREAL	Deceleration																																					
LREAL	Jerk																																					
UINT	Direction																																					
UINT	BufferMode																																					
입력-출력																																						
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)																																				
입력																																						
BOOL	Execute	상승 Edge 에서 해당 축에 절대위치 운전 후 지정속도 운전 지령을 내립니다.																																				
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (10.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)																																				
LREAL	Position	목표위치를 지정합니다.																																				
LREAL	EndVelocity	목표위치 도달 후에 운전할 속도를 지정합니다. [u/s]																																				
LREAL	Velocity	목표위치에 도달하기 위한 최대 속도를 지정합니다. [u/s]																																				
LREAL	Acceleration	가속도를 지정합니다. [u/s ²]																																				
LREAL	Deceleration	감속도를 지정합니다. [u/s ²]																																				
LREAL	Jerk	가/감속도 변화율을 지정합니다 [u/s ³]																																				
UINT	Direction	운전 방향을 지정합니다. (0 ~ 4 : 0-지정없음, 1-정방향, 2-최단거리, 3-역방향, 4-현재방향)																																				
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)																																				
출력																																						
BOOL	InEndVelocity	목표위치에 도달 후 지정 속도로 운전중임을 나타냅니다																																				

BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 절대위치 운전 후 지정속도 운전 지령을 내리는 모션 평선 블록입니다.
- (2) 절대위치 운전 후 지정속도 운전(MC_MoveContinuousAbsolute)을 수행하게 되면 해당 축은 Position 에 지정된 위치로 이동한 후 대기중인 모션 평선 블록이 없는 상태라면 EndVelocity 에 지정된 속도로 운전합니다.
- (3) 속도 운전을 정지하기 위해서는 정지 명령을 내리거나 다른 모션 평선 블록을 동작시켜야 합니다.
- (4) Direction 입력에는 무한길이 반복 운전 시 축의 운전 방향을 설정하며, 무한길이 반복 운전이 금지로 설정된 경우에는 Direction 입력은 무시됩니다. Direction 입력이 최단거리(=2)일 경우, 해당 축이 무한길이 반복 운전을 한다면 최단 거리로 갈 수 있는 방향을 선택하여 운전합니다. 0~4(0-지정없음, 1-정방향, 2-최단거리, 3-역방향, 4-현재방향)범위로 설정할 수 있으며 범위 외의 값을 설정하고 모션 평선 블록을 실행하면 Error 가 On 되고 ErrorID 에 "0x1017" 이 발생합니다.
- (5) 출력 InEndVelocity 는 해당 축이 지정한 위치에 도달 후 속도운전을 시작하여 지정속도에 도달했을 때 온(On)되며, 지정속도 운전이 중단되면 오프(Off) 됩니다.
- (6) 이 명령이 실행 중일 때 축 상태는 '연속운전(ContinuousMotion)'입니다.
- (7) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다. Position, EndVelocity, Velocity, Acceleration, Deceleration, Jerk, Direction 입력만 업데이트 가능합니다. (단, InEndVelocity=On 경우, EndVelocity 입력만 반영됩니다.)

MC_MoveContinuousRelative		적용 기종																																								
상대위치 운전 후 지정속도 운전		XGF-M32E																																								
평선 블록 형태																																										
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center; margin: 0;">MC_MoveContinuousRelative</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL Execute</td> <td style="width: 40%;"></td> <td style="width: 30%;">InEndVelocity</td> <td>BOOL</td> </tr> <tr> <td>UINT Axis</td> <td>-----</td> <td>Axis</td> <td>UINT</td> </tr> <tr> <td>BOOL ContinuousUpdate</td> <td></td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL Distance</td> <td></td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL EndVelocity</td> <td></td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL Velocity</td> <td></td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL Acceleration</td> <td></td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL Deceleration</td> <td></td> <td></td> <td></td> </tr> <tr> <td>LREAL Jerk</td> <td></td> <td></td> <td></td> </tr> <tr> <td>UINT BufferMode</td> <td></td> <td></td> <td></td> </tr> </table> </div> </div>			BOOL Execute		InEndVelocity	BOOL	UINT Axis	-----	Axis	UINT	BOOL ContinuousUpdate		Busy	BOOL	LREAL Distance		Active	BOOL	LREAL EndVelocity		CommandAborted	BOOL	LREAL Velocity		Error	BOOL	LREAL Acceleration		ErrorID	WORD	LREAL Deceleration				LREAL Jerk				UINT BufferMode			
BOOL Execute		InEndVelocity	BOOL																																							
UINT Axis	-----	Axis	UINT																																							
BOOL ContinuousUpdate		Busy	BOOL																																							
LREAL Distance		Active	BOOL																																							
LREAL EndVelocity		CommandAborted	BOOL																																							
LREAL Velocity		Error	BOOL																																							
LREAL Acceleration		ErrorID	WORD																																							
LREAL Deceleration																																										
LREAL Jerk																																										
UINT BufferMode																																										
입력-출력																																										
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)																																								
입력																																										
BOOL	Execute	상승 Edge 에서 해당 축에 상대위치 운전 후 지정속도 운전 지령을 내립니다.																																								
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (10.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)																																								
LREAL	Distance	목표거리를 지정합니다.																																								
LREAL	EndVelocity	목표거리 이동 후에 운전할 속도를 지정합니다. [u/s]																																								
LREAL	Velocity	목표거리를 이동하기 위한 최대 속도를 지정합니다. [u/s]																																								
LREAL	Acceleration	가속도를 지정합니다. [u/s ²]																																								
LREAL	Deceleration	감속도를 지정합니다. [u/s ²]																																								
LREAL	Jerk	가/감속도 변화율을 지정합니다 [u/s ³]																																								
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)																																								
출력																																										
BOOL	InEndVelocity	목표거리만큼 이동 후 지정 속도로 운전중임을 나타냅니다																																								
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.																																								
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다																																								
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.																																								

BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 상대위치 운전 후 지정속도 운전 지령을 내리는 모션 평선 블록입니다.
- (2) 절대위치 운전 후 지정속도 운전(MC_MoveContinuousRelative)을 수행하게 되면 해당 축은 Distance 에 지정된 거리만큼 이동한 후 대기중인 모션 평선 블록이 없는 상태라면 EndVelocity 에 지정된 속도로 운전합니다.
- (3) 지정속도 운전을 정지하기 위해서는 정지 명령을 내리거나 다른 모션 평선 블록을 동작시켜야 합니다.
- (4) 출력 InEndVelocity 는 해당 축이 지정한 거리만큼 이동 후 속도운전을 시작하여 지정속도에 도달했을 때 온(On)되며, 지정속도 운전 중단되면 오프(Off) 됩니다.
- (5) 이 모션 평선 블록이 실행 중일 때 축 상태는 '연속운동(ContinuousMotion)'입니다.
- (6) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.
Distance, EndVelocity, Velocity, Acceleration, Deceleration, Jerk 입력만 업데이트 가능합니다. (단, InEndVelocity=On 경우, EndVelocity 입력만 반영됩니다.)

MC_TorqueControl		적용 기종
토크 제어		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_TorqueControl direction TB Execute[Execute] Axis[Axis] ContinuousUpdate[ContinuousUpdate] Torque[Torque] TorqueRamp[TorqueRamp] Velocity[Velocity] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] Direction[Direction] BufferMode[BufferMode] InTorque[InTorque] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- InTorque Axis --- Axis ContinuousUpdate --- Busy Torque --- Active TorqueRamp --- CommandAborted Velocity --- Error Acceleration --- ErrorID Deceleration --- ErrorID Jerk --- ErrorID Direction --- ErrorID BufferMode --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 토크제어 지령을 내립니다.
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (10.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)
LREAL	Torque	목표 토크를 지정합니다. [u]
LREAL	TorqueRamp	토크 상승 기울기를 지정합니다. [u/s]
LREAL	Velocity	미사용
LREAL	Acceleration	미사용
LREAL	Deceleration	미사용
LREAL	Jerk	미사용
UINT	Direction	운전 방향을 지정합니다. (1 ~ 2: 1-정방향, 2-역방향)
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
출력		
BOOL	InTorque	입력한 토크 값과 현재 운전중인 토크 값이 같음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다

BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 토크제어 지령을 내리는 모션 평선 블록입니다.
- (2) 토크제어(MC_Torque)를 수행하게 되면 해당 축은 Torque 입력에 지정한 토크 값을 유지하는 제어를 수행합니다.
- (3) 지정속도 운전을 정지하기 위해서는 정지 명령을 내리거나 다른 모션 평선 블록을 동작시켜야 합니다.
- (4) TorqueRamp 입력에는 목표 토크 값까지 도달하는 기울기를 지정합니다.
- (5) Velocity 입력에는 토크제어 운전시의 최대 속도를 지정하며 음수 값은 허용되지 않습니다. 회전 방향은 토크와 해당 축에 걸리는 부하의 크기에 따라 결정됩니다.
- (6) Direction 입력에는 운전할 방향을 지정합니다. 범위 이외의 값을 설정하고 모션 평선 블록을 실행하면 Error 가 On 되고 ErrorID 에 “0x1017” 이 발생합니다.
- (7) 출력 InTorque 는 해당 축이 지정한 토크에 도달했을 때 온(On)되며, 토크제어 운전이 중단되면 오프(Off) 됩니다.
- (8) 이 모션 평선 블록이 실행 중일 때 축 상태는 ‘연속운동(ContinuousMotion)’입니다.
- (9) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다. Torque, TorqueRamp, Direction 입력만 업데이트 가능합니다.

MC_SetPosition		적용 기종
현재위치 설정		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_SetPosition Execute[Execute] --- Done[Done] Axis[Axis] --- Axis[Axis] Position[Position] --- Busy[Busy] Relative[Relative] --- Error[Error] ExcutionMode[ExcutionMode] --- ErrorID[ErrorID] end Execute --- Axis Axis --- Position Position --- Relative Relative --- ExcutionMode Done --- Axis Axis --- Busy Busy --- Error Error --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축의 현재 위치를 지정합니다.
LREAL	Position	위치를 지정합니다.
BOOL	Relative	0 - 위치 값 = 절대위치, 1 - 위치 값 = 상대위치
UINT	ExcutionMode	0 - 위치 값 즉시 적용, 1 - Buffermode 의 'Buffered'와 동일한 시점에 적용
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축의 현재위치를 설정하는 모션 평선 블록입니다.
- (2) Position 입력에는 위치를 지정합니다. 모션 평선 블록을 실행했을 때 Relative 입력이 오프(Off) 상태라면 해당 축의 위치는 Position 입력 값으로 대체되며, Relative 입력이 온(On) 상태라면 해당 축의 현재 위치에 Position 입력 값이 더해집니다.
- (3) ExcutionMode 입력은 설정 시점을 지정합니다. 0 일 때는 모션 평선 블록 실행 즉시 설정되며, 1 일 경우에는 순차운전 설정의 'Buffered'와 동일한 시점에 설정됩니다. 설정 불가능한 값을 설정했을 경우 "에러 0x101B" 가 발생합니다.
 0 (mcImmediatly): 평선 블록 실행 즉시(Execute 입력의 상승 Edge) 파라미터 값을 변경합니다. 해당 축이 구동 중이라면 동작에 영향이 있을 수 있습니다.
 1 (mcQueued): Buffermode 의 'Buffered'와 같은 시점에 변경됩니다. (10.1.4 BufferMode 입력 참조)
- (4) ExecutionMode 를 1 로 하여 명령이 실행된 후 실행 대기 중일 때 BufferMode 0(Aborting)으로 다른 명령이 실행되어

중단된 경우는 Error 는 1, ErrorId 는 0 으로 출력됩니다.

MC_SetOverride		적용 기종
속도/가속도 오버라이드		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_SetOverride Execute[Execute] Axis[Axis] VelFactor[VelFactor] AccFactor[AccFactor] JerkFactor[JerkFactor] Enabled[Enabled] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Enabled Axis --- Axis VelFactor --- Busy AccFactor --- Error JerkFactor --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 축에 오버라이드 운전을 수행합니다
LREAL	VelFactor	속도의 오버라이드 비율을 지정합니다
LREAL	AccFactor	가속도/감속도의 오버라이드 비율을 지정합니다
LREAL	JerkFactor	가속도 변화율의 오버라이드 비율을 지정합니다
출력		
BOOL	Enabled	오버라이드 비율이 성공적으로 적용되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축의 속도, 가속도, 가속도 변화율을 오버라이드 하는 모션 평선 블록입니다.
- (2) Enable 입력이 온(On) 되어 있는 동안 해당 축에 적용되는 오버라이드 비율을 지정 및 변경할 수 있습니다. Enable 입력이 오프(Off)되면, 오프(Off) 직전의 오버라이드 비율을 유지합니다.
- (3) VelFactor 입력에는 속도 오버라이드 비율을 지정합니다. 지정값이 0.0 일 경우 해당 축은 정지하게 되지만 '운전대기(StandStill)'상태로 변하지 않습니다.
- (4) AccFactor 입력에는 가속도/감속도, JerkFactor 입력에는 저크(가속도 변화율)의 오버라이드 비율을 각각 지정합니다.
- (5) 각 Facotr 에 음수는 입력할 수 없으며, 입력했을 경우 "에러 0x10C1" 이 발생합니다.
- (6) 각 오버라이드 비율의 기본값은 1.0 이고, 현재 실행되고 있는 평선 블록의 명령 속도 100%를 의미합니다.
- (7) 오버라이드 동작은 해당 축의 종축에는 영향을 미치지 않습니다.

MC_ReadParameter		적용 기종
파라미터 읽기		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_ReadParameter Enable[Enable] Axis[Axis] ParameterNumber[ParameterNumber] Vaild[Vaild] Busy[Busy] Error[Error] ErrorID[ErrorID] Value[Value] end Enable --- Vaild Axis --- Vaild ParameterNumber --- Vaild Enable --- Busy Axis --- Busy ParameterNumber --- Busy Enable --- Error Axis --- Error ParameterNumber --- Error Enable --- ErrorID Axis --- ErrorID ParameterNumber --- ErrorID Enable --- Value Axis --- Value ParameterNumber --- Value </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 파라미터가 출력됩니다.
INT	ParameterNumber	읽을 파라미터의 번호를 지정합니다. (0 ~ 28, 100 ~ 116)
출력		
BOOL	Vaild	현재 모션 평선 블록의 출력이 유효한지 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
LREAL	Value	파라미터 값을 출력합니다.

- (1) 이 명령은 해당 축의 파라미터를 출력하는 모션 평선 블록입니다.
- (2) Enable 입력이 온(On) 되어 있는 동안 Value 에 해당 파라미터의 값이 연속적으로 출력됩니다.
- (3) ParameterNumber 입력에는 읽을 파라미터의 번호를 지정합니다.

(4) 파라미터의 번호는 아래와 같습니다.

번호	파라미터	항목	비고	
0	기본 파라미터	단위	0:pulse,1:mm,2:inch,3:degree	
1		1 회전 당 펄스 수	1 ~ 4,294,967,295 [pulse]	
2		1 회전 당 이송 거리	0.000000001 ~ 4,294,967,295 [단위]	
3		속도명령단위	0:단위/시간, 1:rpm	
4		속도 제한치	배장형 실수(LREAL) 양수 ^{*주1)} [단위/s, rpm] (단위, 1 회전 당 펄스 수, 1 회전 당 이송거리, 속도명령 단위에 따라 설정범위가 변경됨)	
5		급정지 시 감속도	0 또는 배장형 실수(LREAL) 양수 ^{*주1)} [단위/s ²]	
6		엔코더 선택	0:인크리멘탈 엔코더,1:엡솔루트 엔코더	
7		모터측 기어비	1 ~ 65,535	
8		기계측 기어비	1 ~ 65,535	
9		역회전 동작 모드	0:금지, 1:허용	
26		속도 제어 운전 모드	0: CSP (Cyclic Sync. Position) 1: CSV (Cyclic Sync. Velocity)	
28		위치 제어 범위 확장	0: 사용안함, 1: 사용	
10		확장 파라미터	소프트 상한	LREAL ^{*주1)} [단위]
11			소프트 하한	LREAL ^{*주1)} [단위]
12	무한길이 반복위치		배장형 실수(LREAL) 양수 ^{*주1)} [단위]	
13	무한길이 반복		0:금지, 1:허용	
14	명령 인포지션폭		0 또는 배장형 실수(LREAL) 양수 ^{*주1)} [단위]	
15	추종오차 초과값		0 또는 배장형 실수(LREAL) 양수 ^{*주1)} [단위]	
16	현재위치 표시 보정량		0 또는 배장형 실수(LREAL) 양수 ^{*주1)}	
17	현재속도 필터 시정수		0 ~ 100	
18	에러 리셋 모니터링 시간		1 ~ 1000 [ms]	
19	속도제어 중 소프트웨어 상/하한		0:검출안함, 1:검출	
20	추종 오차 에러 레벨		0:경고, 1:알람	
21	조그 고속 속도		배장형 실수(LREAL) 양수 ^{*주1)} (조그 저속 속도 ~ 속도제한치) [단위/s]	
22	조그 저속 속도		배장형 실수(LREAL) 양수 ^{*주1)} (< 조그 고속 속도) [단위/s]	
23	조그 가속도		0 또는 배장형 실수(LREAL) 양수 ^{*주1)} [단위/s ²]	
24	조그 감속도		0 또는 배장형 실수(LREAL) 양수 ^{*주1)} [단위/s ²]	
25	조그 저크		0 또는 배장형 실수(LREAL) 양수 ^{*주1)} [단위/s ³]	

27		동기 운전 시 주축 동기 방법	0: 주축 운전의 양방향에 동기 1: 주축 운전의 정방향에 동기 2: 주축 운전의 역방향에 동기
100	공통 파라미터	엔코더 1 단위	0: pulse, 1: mm, 2: inch, 3:degree
101		엔코더 1 1회전당 펄스 수	1 ~ 4294967295
102		엔코더 1 1회전당 이송거리	0.000000001 ~ 4294967295
103		엔코더 1 펄스 입력	0: CW/CCW 1 체배, 1: PULSE/DIR 1 체배 2: PULSE/DIR 2 체배, 3: PHASE A/B 1 체배 4: PHASE A/B 2 체배, 5: PHASE A/B 4 체배
104		엔코더 2 단위	0: pulse, 1: mm, 2: inch, 3:degree
105		엔코더 2 1회전당 펄스 수	1 ~ 4294967295
106		엔코더 2 1회전당 이송거리	0.000000001 ~ 4294967295
107		엔코더 2 펄스 입력	0: CW/CCW 1 체배, 1: PULSE/DIR 1 체배 2: PULSE/DIR 2 체배, 3: PHASE A/B 1 체배 4: PHASE A/B 2 체배, 5: PHASE A/B 4 체배
108		엔코더 1 최대값	배장형 실수(LREAL)
109		엔코더 1 최소값	
110		엔코더 2 최대값	
111		엔코더 2 최소값	
112		오버라이드	0: 비율 지정, 1: 단위값 지정
113		엔코더 1 위치필터 시정수	0~1000 ms
114		엔코더 2 위치필터 시정수	0~1000 ms
115		엔코더 1 속도 단위	0: 단위/sec, 1: rpm, 2: 단위/min
116	엔코더 2 속도 단위	0: 단위/sec, 1: rpm, 2: 단위/min	

*주 1) LREAL 범위: 2.2250738585072e-308 ~ 1.79769313486232e+308

배장형 실수(LREAL) 양수 범위: $0 < x \leq 1.79769313486232e+308$

MC_WriteParameter		적용 기종
파라미터 쓰기		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_WriteParameter direction TB Execute[Execute] Axis[Axis] ParameterNumber[ParameterNumber] Value[Value] ExecutionMode[ExecutionMode] Vaild[Vaild] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Vaild Axis --- Axis ParameterNumber --- Busy Value --- Error ExecutionMode --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	입력의 상승 Edge 에서 해당 파라미터가 쓰기 지령을 내립니다.
INT	ParameterNumber	쓸 파라미터의 번호를 지정합니다. (0 ~ 28, 100 ~ 116)
LREAL	Value	쓸 파라미터 값을 지정합니다.
UINT	ExecutionMode	파라미터가 쓰여질 시점을 지정합니다.
출력		
BOOL	Vaild	파라미터가 성공적으로 쓰여졌는지 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축의 운전 파라미터 또는 공통 파라미터에 지정한 값을 쓰는 모션 평선 블록입니다.
- (2) Execute 입력의 상승 Edge 에서 파라미터가 쓰여지게 됩니다.
- (3) ParameterNumber 입력에는 쓸 파라미터의 번호를 지정합니다. 설정 불가능한 값을 설정했을 경우 “에러 0x10F0” 이 발생합니다.
- (4) Value 입력에는 파라미터에 쓸 값을 지정합니다.
- (5) ExecutionMode 에는 파라미터가 쓰여질 시점을 지정하며 아래와 같은 값을 설정할 수 있습니다. 설정 불가능한 값을 설정했을 경우 “에러 0x101B” 가 발생합니다.
 - 0 (mcImmediatly): 평선 블록 실행 즉시(Execute 입력의 상승 Edge) 파라미터 값을 변경합니다. 해당 축이 구동 중이라면 동작에 영향이 있을 수 있습니다.
 - 1 (mcQueued): Buffermode 의 'Buffered'와 같은 시점에 변경됩니다. (10.1.4 BufferMode 입력 참조)

(6) ExecutionMode 를 1 로 하여 명령이 실행된 후 실행 대기 중일 때 BufferMode 0(Aborting)으로 다른 명령이 실행되어 중단된 경우는 Error 는 1, ErrorId 는 0 으로 출력됩니다.

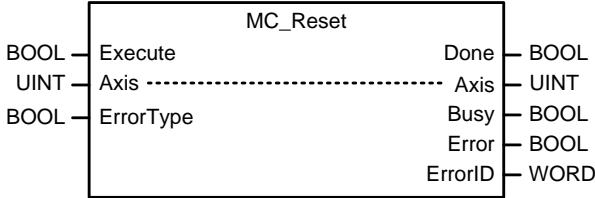
(7) 파라미터의 번호는 아래와 같습니다.

번호	파라미터	항목	비고
0	기본 파라미터	단위	0:pulse,1:mm,2:inch,3:degree
1		1 회전 당 펄스 수	1 ~ 4,294,967,295 [pulse]
2		1 회전 당 이송 거리	0.000000001 ~ 4,294,967,295 [단위]
3		속도명령단위	0:단위/시간, 1:rpm
4		속도 제한치	배장형 실수(LREAL) 양수 *주1) [단위/s, rpm] (단위, 1 회전 당 펄스 수, 1 회전 당 이송거리, 속도명령 단위에 따라 설정범위가 변경됨)
5		급정지 시 감속도	0 또는 배장형 실수(LREAL) 양수 *주1) [단위/s ²]
6		엔코더 선택	0:인크리멘탈 엔코더,1:앱솔루트 엔코더
7		모터측 기어비	1 ~ 65,535
8		기계측 기어비	1 ~ 65,535
9		역회전 동작 모드	0:금지, 1:허용
26		속도 제어 운전 모드	0: CSP (Cyclic Sync. Position) 1: CSV (Cyclic Sync. Velocity)
28		위치 제어 범위 확장	0: 사용안함, 1: 사용
10		확장 파라미터	소프트 상한
11	소프트 하한		LREAL *주1) [단위]
12	무한길이 반복위치		배장형 실수(LREAL) 양수 *주1) [단위]
13	무한길이 반복		0:금지, 1:허용
14	명령 인포지션폭		0 또는 배장형 실수(LREAL) 양수 *주1) [단위]
15	추종오차 초과값		0 또는 배장형 실수(LREAL) 양수 *주1) [단위]
16	현재위치 표시 보정량		0 또는 배장형 실수(LREAL) 양수 *주1)
17	현재속도 필터 시정수		0 ~ 100
18	에러 리셋 모니터링 시간		1 ~ 1000 [ms]
19	속도제어 중 소프트웨어 상/하한		0:검출안함, 1:검출
20	추종 오차 에러 레벨		0:경고, 1:알람
21	조그 고속 속도		배장형 실수(LREAL) 양수 *주1) (조그 저속 속도 ~ 속도제한치) [단위/s]
22	조그 저속 속도		배장형 실수(LREAL) 양수 *주1) (< 조그 고속 속도) [단위/s]
23	조그 가속도		0 또는 배장형 실수(LREAL) 양수 *주1) [단위/s ²]
24	조그 감속도		0 또는 배장형 실수(LREAL) 양수 *주1) [단위/s ²]
25	조그 저크	0 또는 배장형 실수(LREAL) 양수 *주1) [단위/s ³]	

27		동기 운전 시 주축 동기 방법	0: 주축 운전의 양방향에 동기 1: 주축 운전의 정방향에 동기 2: 주축 운전의 역방향에 동기
100	공통 파라미터	엔코더 1 단위	0: pulse, 1: mm, 2: inch, 3:degree
101		엔코더 1 1회전당 펄스 수	1 ~ 4294967295
102		엔코더 1 1회전당 이송거리	0.000000001 ~ 4294967295
103		엔코더 1 펄스 입력	0: CW/CCW 1 체배, 1: PULSE/DIR 1 체배 2: PULSE/DIR 2 체배, 3: PHASE A/B 1 체배 4: PHASE A/B 2 체배, 5: PHASE A/B 4 체배
104		엔코더 2 단위	0: pulse, 1: mm, 2: inch, 3:degree
105		엔코더 2 1회전당 펄스 수	1 ~ 4294967295
106		엔코더 2 1회전당 이송거리	0.000000001 ~ 4294967295
107		엔코더 2 펄스 입력	0: CW/CCW 1 체배, 1: PULSE/DIR 1 체배 2: PULSE/DIR 2 체배, 3: PHASE A/B 1 체배 4: PHASE A/B 2 체배, 5: PHASE A/B 4 체배
108		엔코더 1 최대값	배장형 실수(LREAL)
109		엔코더 1 최소값	
110		엔코더 2 최대값	
111		엔코더 2 최소값	
112		오버라이드	0: 비율 지정, 1: 단위값 지정
113		엔코더 1 위치필터 시정수	0~1000 ms
114		엔코더 2 위치필터 시정수	0~1000 ms
115		엔코더 1 속도 단위	0: 단위/sec, 1: rpm, 2: 단위/min
116	엔코더 2 속도 단위	0: 단위/sec, 1: rpm, 2: 단위/min	

*주 1) LREAL 범위: 2.2250738585072e-308 ~ 1.79769313486232e+308

배장형 실수(LREAL) 양수 범위: $0 < x \leq 1.79769313486232e+308$

MC_Reset		적용 기종
축 에러 리셋		XGF-M32E
평선 블록 형태		
		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	입력의 상승 Edge 에서 축 에러를 리셋합니다.
BOOL	ErrorType	리셋할 에러의 종류 (0: 축 에러, 1: 공통 에러)
출력		
BOOL	Done	축 에러가 성공적으로 리셋되었는지 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

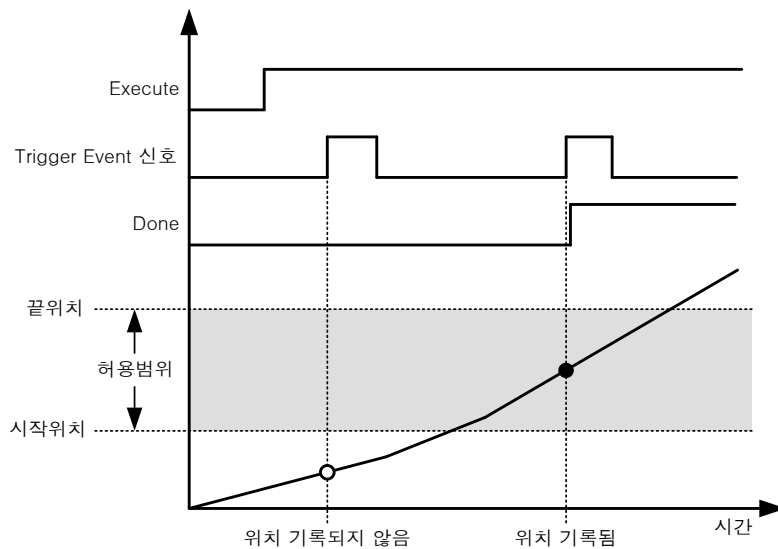
- (1) 이 모션 평선 블록은 해당 축의 에러를 리셋하는 모션 평선 블록입니다. 해당 축이 '에러정지(ErrorStop)'상태일 때 ErrorType 을 '0'으로 설정하고 모션 평선 블록을 수행하면 축의 에러는 모두 리셋되고 축 상태는 '운전대기(StandStill)' 또는 '비활성화(Disabled)'상태로 전환됩니다.
- (2) ErrorType 을 '1'로 설정하고 모션 평선 블록을 수행하면 해당 모듈에 발생한 공통 에러가 리셋됩니다.
- (3) Execute 입력의 상승 Edge 에서 모션 평선 블록이 실행됩니다.

MC_TouchProbe		적용 기종
터치 프루브		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph TD subgraph MC_TouchProbe Execute[Execute] Axis[Axis] TriggerInput[TriggerInput] WindowOnly[WindowOnly] FirstPosition[FirstPosition] LastPosition[LastPosition] Done[Done] Busy[Busy] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] RecordedPosition[RecordedPosition] end Execute --- Done Axis --- Axis TriggerInput --- TriggerInput WindowOnly --- Busy FirstPosition --- CommandAborted LastPosition --- Error ErrorID --- ErrorID RecordedPosition --- RecordedPosition </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축)
UINT	TriggerInput	트리거로 사용될 신호를 지정합니다. (0: 터치 프로브 1, 1: 터치 프로브 2)
입력		
BOOL	Execute	입력의 상승 Edge 에서 터치 프로브 기능이 시작됩니다.
BOOL	WindowOnly	윈도우 모드를 활성화 시킵니다.
LREAL	FirstPosition	윈도우 모드에서 허용영역 시작위치를 지정합니다.
LREAL	LastPosition	윈도우 모드에서 허용영역 끝 위치를 지정합니다.
출력		
BOOL	Done	트리거 신호가 성공적으로 기록되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
REAL	RecordedPosition	트리거가 발생한 축 위치를 출력합니다.

- (1) 이 모션 평선 블록은 트리거 이벤트가 발생한 시점의 축 위치를 기록하는 ‘터치 프로브’ 기능을 수행하는 모션 평선 블록입니다.
- (2) Execute 입력의 상승 Edge 에서 터치 프로브 기능이 시작됩니다.
- (3) TriggerInput 에는 트리거로 사용될 신호를 지정합니다. 설정 불가능한 값을 설정했을 경우 에러 “0x10E1” 이 발생합니다.
- (4) 터치 프로브 명령이 트리거 조건을 만족하면 Done 출력은 On 되고, RecordedPosition 출력에 트리거 시점의 위치

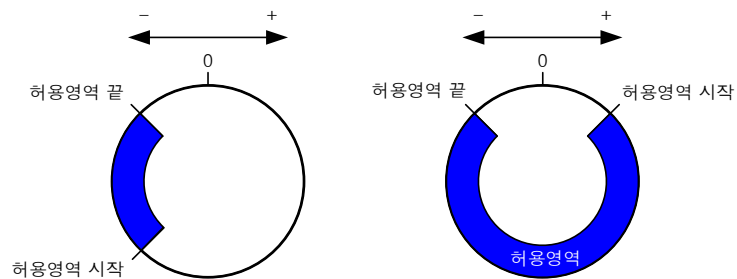
값이 저장됩니다. RecordedPosition 출력은 Execute 입력이 On 또는 Off 로 변경되거나 에러가 발생하는 경우에도 초기화되지 않고 이전에 저장된 값을 유지합니다. 터치 프로브 명령이 재실행되어 정상적으로 트리거 조건을 만족할 때 RecordedPosition 출력이 갱신됩니다.

- (5) 윈도우 모드를 활성화시키면 축의 트리거 신호를 받아들이는 허용 영역을 설정시킬 수 있습니다. 윈도우 모드가 활성화 되어 있을 때의 각 신호들의 동작 타이밍은 아래와 같습니다.

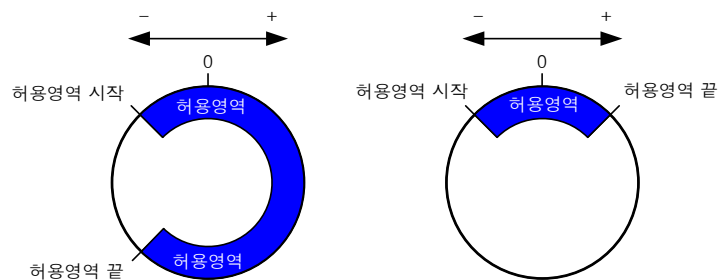


< 터치 프루브 기능이 윈도우 모드일 때, 동작 타이밍 >

- 허용영역 시작위치 < 허용영역 끝위치인 경우



- 허용영역 시작위치 > 허용영역 끝위치인 경우



MC_AbortTrigger		적용 기종
트리거 해제		XGF-M32E
평선 블록 형태		
<pre> graph LR subgraph MC_AbortTrigger direction TB Execute[Execute] Axis[Axis] TriggerInput[TriggerInput] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done Axis --- AxisOut[Axis] TriggerInput --- TriggerInputOut[TriggerInput] Busy --- BusyOut[Busy] Error --- ErrorOut[Error] ErrorID --- ErrorIDOut[ErrorID] </pre>		
입력-출력		
UINT	Axis	지령을 내릴 축을 설정합니다. (1~32: 실축)
UINT	TriggerInput	해제할 트리거 신호를 지정합니다. (0: 터치 프로브 1, 1: 터치 프로브 2)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 대기중인 트리거가 해제됩니다.
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 대기 중인 트리거를 해제합니다.
- (2) TriggerInput 에는 해제할 트리거 신호를 지정합니다. 설정 불가능한 값을 설정했을 경우 에러 “0x10E1” 이 발생합니다.

MC_MoveSuper Imposed		적용 기종
Super Imposed 운전		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph TD subgraph MC_MoveSuperImposed Execute[Execute] Axis[Axis] ContinuousUpdate[ContinuousUpdate] Distance[Distance] VelocityDiff[VelocityDiff] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] CoveredDistance[CoveredDistance] end Execute --- Done Axis --- Axis ContinuousUpdate --- Busy Distance --- Active VelocityDiff --- CommandAborted Acceleration --- Error Deceleration --- ErrorID Jerk --- CoveredDistance </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1 ~ 32 : 1 축 ~ 32 축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 SuperImposed 운전 지령을 내립니다.
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (6.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)
LREAL	Distance	추가 목표거리를 지정합니다. [u]
LREAL	VelocityDiff	추가 운전속도를 지정합니다. [u/s]
LREAL	Acceleration	추가 가속도를 지정합니다. [u/s ²]
LREAL	Deceleration	추가 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	추가 가/감속도 변화율을 지정합니다 [u/s ³]
출력		
BOOL	Done	지정한 거리 도달 완료 여부를 나타냅니다.
BOOL	Busy	평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 평선 블록이 다른 명령에 의해 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	평선 블록 실행 중 발생한 에러번호를 출력합니다.
LREAL	CoveredDistance	SuperImposed 명령 후 SuperImposed 운전으로 운전한 거리를 나타냅니다.

- (1) 이 모션 평선 블록은 해당 축에 **SuperImposed** 운전 지령을 내리는 명령입니다.
- (2) **SuperImposed** 운전 명령은 명령 수행 시점의 현재 위치를 기준으로 **Distance** 입력에 지정된 목표거리만큼 추가로 이동하는 명령입니다.
- (3) 이동 방향은 **Distance** 입력에 지정된 목표거리의 부호에 따라 결정되며 이동 방향이 양(+ 또는 부호 없음)일 때는 정방향, 음(-)일 때는 역방향으로 이동하게 됩니다.
- (4) 명령 이후에 목표 거리 이동 후 속도가 0 이 되면 명령은 완료되고 **Done** 출력이 **On** 됩니다.

MC_HaltSuperImposed		적용 기종
SuperImposed 운전 정지		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_HaltSuperImposed Execute[Execute] Axis[Axis] Deceleration[Deceleration] Jerk[Jerk] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done Axis --- Axis Deceleration --- Busy Jerk --- Active CommandAborted --- CommandAborted Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1 ~ 32 : 1 축 ~ 32 축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 SuperImposed 운전 정지 지령을 내립니다.
LREAL	Deceleration	정지 시의 감속도를 지정합니다 [u/s ²]
LREAL	Jerk	가/감속도 변화율을 지정합니다 [u/s ³]
출력		
BOOL	Done	해당 축의 속도가 0 에 도달했음을 나타냅니다.
BOOL	Busy	평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 평선 블록이 축을 제어하는 중임을 나타냅니다.
BOOL	CommandAborted	현재 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 SuperImposed 운전 정지 지령을 내리는 명령입니다.
- (2) SuperImposed 운전 정지 명령은 명령 수행 시점에서 주어진 가속도 및 저크로 감속 정지 하는 명령입니다.
- (3) 명령 이후에 속도가 0 이 되면 명령은 완료되고 Done 출력이 On 됩니다.

MC_ReadMotionInfo		적용 기종
모션 정보 읽기		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_ReadMotionInfo Enable[Enable] Axis[Axis] ParameterNumber[ParameterNumber] Vaild[Vaild] Busy[Busy] Error[Error] ErrorID[ErrorID] Value32[Value32] Value64[Value64] end Enable --- Vaild Axis --- Busy ParameterNumber --- Error ParameterNumber --- ErrorID ParameterNumber --- Value32 ParameterNumber --- Value64 </pre> </div>		
입력-출력		
UINT	Axis	명령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 파라미터가 출력됩니다.
INT	ParameterNumber	읽을 파라미터의 번호를 지정합니다.
출력		
BOOL	Vaild	현재 모션 평선 블록의 출력이 유효한지 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
REAL	Value32	파라미터 값을 실수로 출력합니다.
LREAL	Value64	파라미터 값을 배장형 실수로 출력합니다.

- (1) 이 명령은 해당 축의 모션 정보를 출력하는 모션 평선 블록입니다.
- (2) Enable 입력이 온(On) 되어 있는 동안 Value32 과 Value64 에 해당 파라미터의 값이 연속적으로 출력됩니다.
- (3) ParameterNumber 입력에는 읽을 파라미터의 번호를 지정합니다. 축 파라미터의 번호는 아래와 같습니다.

번호	항목	비고	OS
0	현재 위치	단위	V1.30
1	명령 위치	단위	V1.30
2	현재 속도	단위/sec	V1.30
3	명령 속도	단위/sec	V1.30

번호	항목	비고	OS
4	현재 토크	정격 토크의 %	V1.30
5	명령 토크	정격 토크의 %	V1.30

MC_ReadActualPosition		적용 기종
현재 위치 읽기		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_ReadActualPosition Enable[Enable] Axis[Axis] Vaild[Vaild] Axis2[Axis] Busy[Busy] Error[Error] ErrorID[ErrorID] Value32[Value32] Value64[Value64] end Enable --- Vaild Axis --- Axis2 Axis --- Busy Axis --- Error Axis --- ErrorID Axis --- Value32 Axis --- Value64 </pre> </div>		
입력-출력		
UINT	Axis	명령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 파라미터가 출력됩니다.
출력		
BOOL	Vaild	현재 모션 평선 블록의 출력이 유효한지 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
REAL	Value32	현재 위치 값을 실수로 출력합니다.
LREAL	Value64	현재 위치 값을 배장형 실수로 출력합니다.

- (1) 이 명령은 해당 축의 현재 위치를 출력하는 모션 평선 블록입니다.
- (2) Enable 입력이 온(On) 되어 있는 동안 Value32 과 Value64 에 해당 파라미터의 값이 연속적으로 출력됩니다.

MC_ReadActualVelocity		적용 기종
현재 속도 읽기		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_ReadActualVelocity Enable[Enable] Axis[Axis] Vaild[Vaild] Busy[Busy] Error[Error] ErrorID[ErrorID] Value32[Value32] Value64[Value64] end Enable --- Vaild Axis --- Vaild Axis --- Busy Axis --- Error Axis --- ErrorID Axis --- Value32 Axis --- Value64 </pre> </div>		
입력-출력		
UINT	Axis	명령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 파라미터가 출력됩니다.
출력		
BOOL	Vaild	현재 모션 평선 블록의 출력이 유효한지 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
REAL	Value32	현재 속도 값을 실수로 출력합니다.
LREAL	Value64	현재 속도 값을 배장형 실수로 출력합니다.

- (1) 이 명령은 해당 축의 현재 속도를 출력하는 모션 평선 블록입니다.
- (2) Enable 입력이 온(On) 되어 있는 동안 Value32 과 Value64 에 해당 파라미터의 값이 연속적으로 출력됩니다.

MC_ReadActualTorque		적용 기종
현재 토크 읽기		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_ReadActualTorque direction TB Enable[Enable] Axis[Axis] Vaild[Vaild] Busy[Busy] Error[Error] ErrorID[ErrorID] Value32[Value32] Value64[Value64] end Enable --- MC_ReadActualTorque Axis --- MC_ReadActualTorque MC_ReadActualTorque --- Vaild MC_ReadActualTorque --- Busy MC_ReadActualTorque --- Error MC_ReadActualTorque --- ErrorID MC_ReadActualTorque --- Value32 MC_ReadActualTorque --- Value64 </pre> </div>		
입력-출력		
UINT	Axis	명령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 파라미터가 출력됩니다.
출력		
BOOL	Vaild	현재 모션 평선 블록의 출력이 유효한지 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
REAL	Value32	현재 토크 값을 실수로 출력합니다.
LREAL	Value64	현재 토크 값을 배장형 실수로 출력합니다.

- (1) 이 명령은 해당 축의 현재 토크를 출력하는 모션 평선 블록입니다.
- (2) Enable 입력이 온(On) 되어 있는 동안 Value32 과 Value64 에 해당 파라미터의 값이 연속적으로 출력됩니다.

MC_ReadCommandedPosition		적용 기종
명령 위치 읽기		XGF-M32E
펄스 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_ReadCommandedPosition Enable[Enable] Axis1[Axis] Vaild[Vaild] Axis2[Axis] Busy[Busy] Error[Error] ErrorID[ErrorID] Value32[Value32] Value64[Value64] end Enable --- MC_ReadCommandedPosition Axis1 --- MC_ReadCommandedPosition Vaild --- MC_ReadCommandedPosition Axis2 --- MC_ReadCommandedPosition Busy --- MC_ReadCommandedPosition Error --- MC_ReadCommandedPosition ErrorID --- MC_ReadCommandedPosition Value32 --- MC_ReadCommandedPosition Value64 --- MC_ReadCommandedPosition </pre> </div>		
입력-출력		
UINT	Axis	명령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 파라미터가 출력됩니다.
출력		
BOOL	Vaild	현재 모션 펄스 블록의 출력이 유효한지 나타냅니다.
BOOL	Busy	모션 펄스 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 펄스 블록 실행 중 발생한 에러번호를 출력합니다.
REAL	Value32	명령 위치 값을 실수로 출력합니다.
LREAL	Value64	명령 위치 값을 배장형 실수로 출력합니다.

- (1) 이 명령은 해당 축의 명령 위치를 출력하는 모션 펄스 블록입니다.
- (2) Enable 입력이 온(On) 되어 있는 동안 Value32 과 Value64 에 해당 파라미터의 값이 연속적으로 출력됩니다.

MC_ReadCommandedVelocity		적용 기종
명령 속도 읽기		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_ReadCommandedVelocity Enable[Enable] Axis[Axis] Vaild[Vaild] Busy[Busy] Error[Error] ErrorID[ErrorID] Value32[Value32] Value64[Value64] end Enable --- Vaild Axis --- Value32 Axis --- Value64 </pre> </div>		
입력-출력		
UINT	Axis	명령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 파라미터가 출력됩니다.
출력		
BOOL	Vaild	현재 모션 평선 블록의 출력이 유효한지 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
REAL	Value32	명령 속도 값을 실수로 출력합니다.
LREAL	Value64	명령 속도 값을 배장형 실수로 출력합니다.

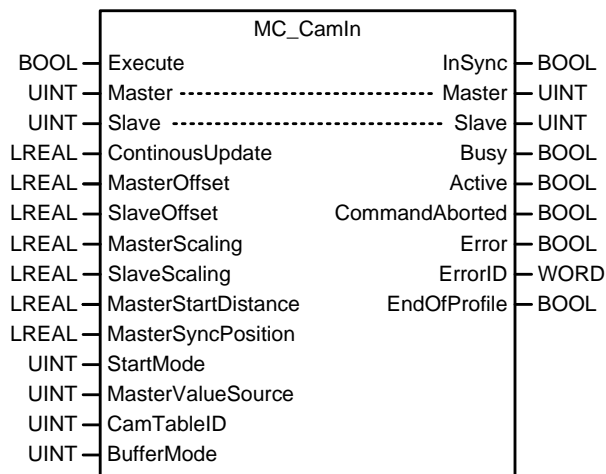
- (1) 이 명령은 해당 축의 명령 속도를 출력하는 모션 평선 블록입니다.
- (2) Enable 입력이 온(On) 되어 있는 동안 Value32 과 Value64 에 해당 파라미터의 값이 연속적으로 출력됩니다.

MC_ReadCommandedTorque		적용 기종
명령 토크 읽기		XGF-M32E
펄스 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_ReadCommandedTorque Enable[Enable] Axis[Axis] Vaild[Vaild] Axis2[Axis] Busy[Busy] Error[Error] ErrorID[ErrorID] Value32[Value32] Value64[Value64] end Enable --- Vaild Axis --- Axis2 Axis --- Busy Axis --- Error Axis --- ErrorID Axis --- Value32 Axis --- Value64 </pre> </div>		
입력-출력		
UINT	Axis	명령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 파라미터가 출력됩니다.
출력		
BOOL	Vaild	현재 모션 펄스 블록의 출력이 유효한지 나타냅니다.
BOOL	Busy	모션 펄스 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 펄스 블록 실행 중 발생한 에러번호를 출력합니다.
REAL	Value32	명령 토크 값을 실수로 출력합니다.
LREAL	Value64	명령 토크 값을 배장형 실수로 출력합니다.

- (1) 이 명령은 해당 축의 명령 토크를 출력하는 모션 펄스 블록입니다.
- (2) Enable 입력이 온(On) 되어 있는 동안 Value32 과 Value64 에 해당 파라미터의 값이 연속적으로 출력됩니다.

MC_CamIn	적용 기종
캠 운전	XGF-M32E

평선 블록 형태



입력-출력

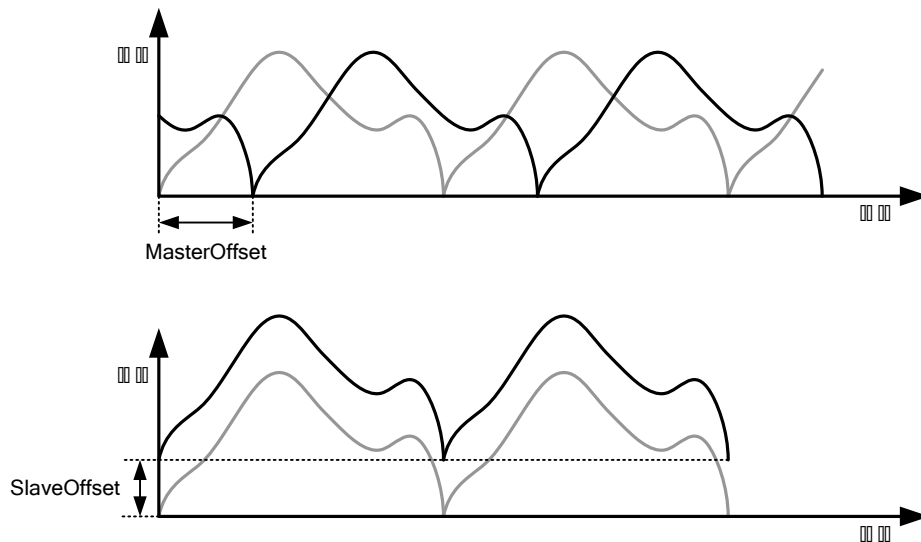
UINT	Master	주축을 설정합니다. (1~32: 실축, 37~40: 가상축, 41~42: 엔코더)
UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축)

입력

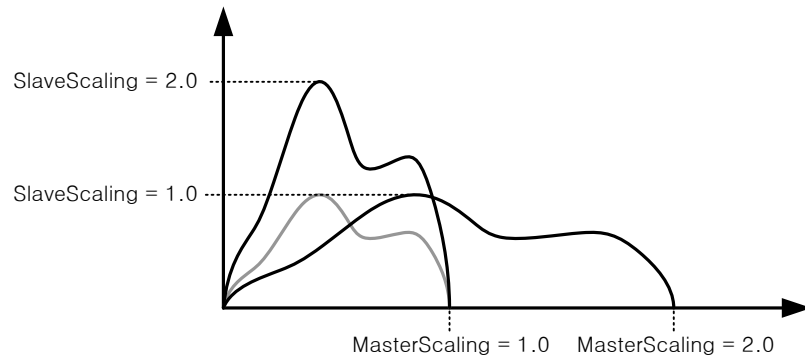
BOOL	Execute	상승 Edge 에서 해당 축에 캠 운전 지령을 내립니다.
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (10.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)
LREAL	MasterOffset	주축의 오프셋 값을 설정합니다.
LREAL	SlaveOffset	종축 캠 테이블의 오프셋 값을 설정합니다.
LREAL	MasterScaling	주축의 배율을 지정합니다.
LREAL	SlaveScaling	종축 캠 테이블의 배율을 지정합니다.
LREAL	MasterStartDistance	종축의 캠 운전이 시작될 주축의 위치를 지정합니다.
LREAL	MasterSyncPosition	캠 운전이 시작할 때, 캠 테이블에서의 시작점을 지정합니다.
UINT	StartMode	캠 운전 모드를 설정합니다. 0: 캠 테이블이 절대값으로 적용됨 (mcAbsolute) 1: 캠 테이블이 명령 시작 위치를 기준으로 상대값으로 적용됨 (mcRelative)
UINT	MasterValueSource	캠 운전할 주축의 소스를 선택합니다.

		0: 주축의 목표 값에 동기화합니다. 1: 주축의 현재 값에 동기화합니다.
UINT	CamTableID	운전할 캠 테이블을 지정합니다.
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
출력		
BOOL	InSync	캠 동작이 정상적으로 이루어지고 있음을 나타냅니다. (종축이 캠 테이블을 추종 중임을 나타냅니다)
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
BOOL	EndOfProfile	현재 운전중인 캠의 끝에 도달했음을 나타냅니다

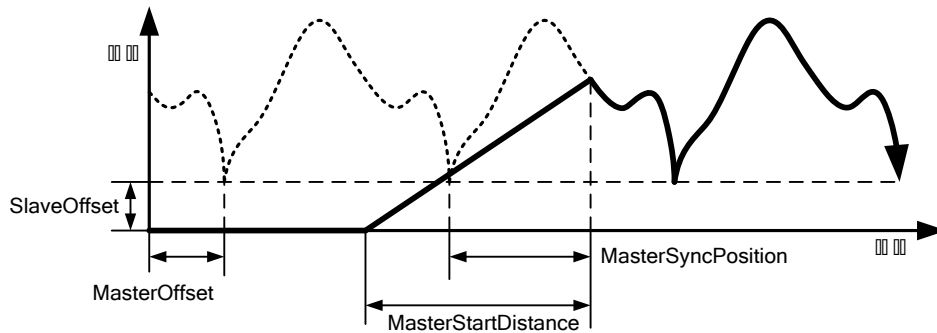
- (1) 이 모션 평선 블록은 주축에 따라 종축 캠 운전을 시키는 모션 평선 블록입니다.
- (2) 주축이 정지상태가 아니더라도 종축에 캠 운전을 내릴 수 있습니다.
- (3) 캠 운전을 정지하기 위해서는 종축에 **MC_CamOut** 명령을 내리거나 다른 모션 평선 블록을 동작시켜야 합니다.
- (4) 이 모션 평선 블록이 실행 중일 때 축은 '동기운전(Synchronized Motion)' 상태입니다.
- (5) **MasterOffset** 및 **SlaveOffset** 에는 적용할 캠 테이블의 오프셋을 설정합니다. **MasterOffset** 은 주축 시작점과의 오프셋을, **SlaveOffset** 은 종축 시작점과의 오프셋을 설정합니다. 아래 그림을 참고하시기 바랍니다.



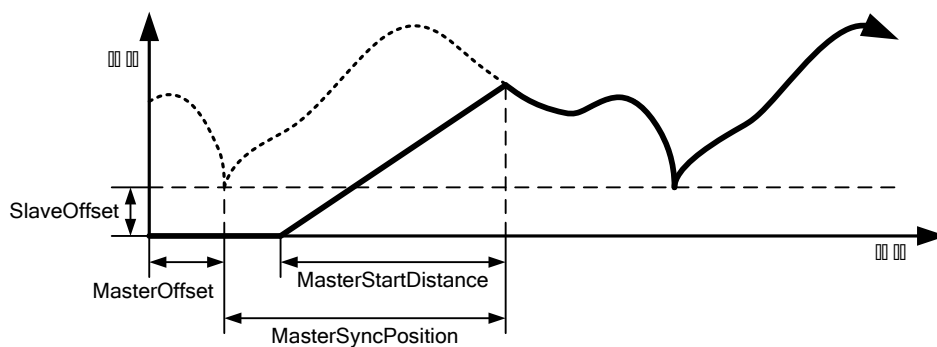
- (6) **MasterScaling** 및 **SlaveScaling** 에는 적용될 캠 데이터의 배율을 설정합니다. **MasterScaling** 은 주축 데이터의 배율을, **SlaveScaling** 에는 종축 데이터의 배율을 설정합니다. 아래 그림을 참고하시기 바랍니다.



(7) MasterSyncPosition 입력은 실제 캠 운전이 동기화 완료될 테이블 내의 주축의 위치를 지정하며, MasterStartDistance 입력은 동기화가 시작될 주축의 상대 위치를 지정합니다.



MasterScaling 이 1.0 일 때



MasterScaling 이 2.0 일 때

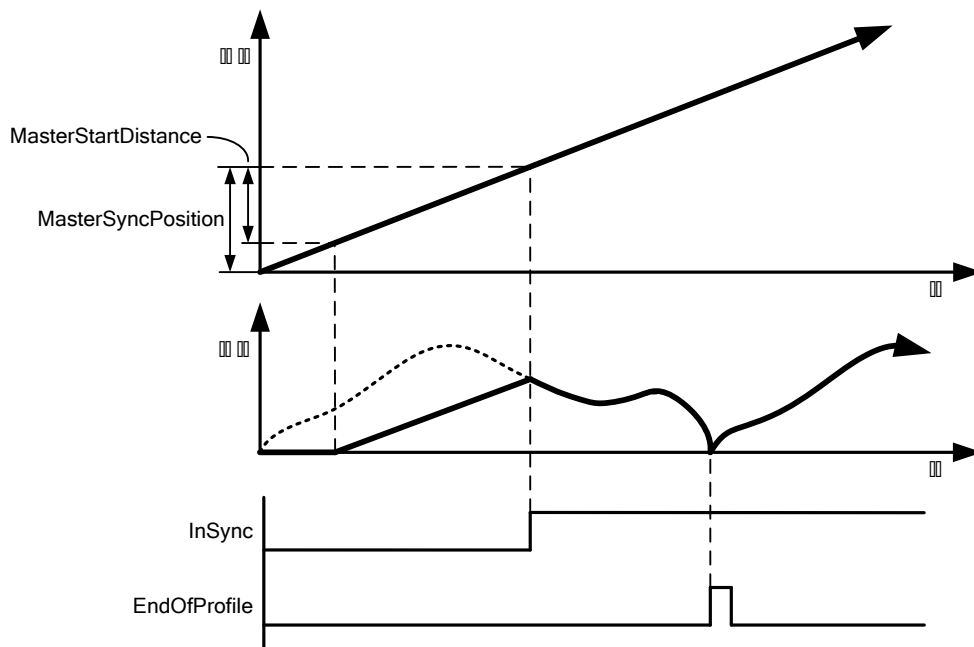
MasterSyncPosition 위치는 캠 테이블 내의 위치를 기준으로 한 위치이며, 명령 실행시의 MasterOffset 및 MasterScale 파라미터를 감안하여 실제 동기화 위치가 결정됩니다.

MasterSyncPosition 이 실제 적용된 위치를 기준으로 MasterStartDistance 입력 값만큼 떨어진 위치에서부터 종축이 동기화 위치로 이동을 시작합니다. 이동을 시작하기 전이라면 종축은 해당 위치에 정지상태로 대기하며, 만약 명령 시작 시 종축이

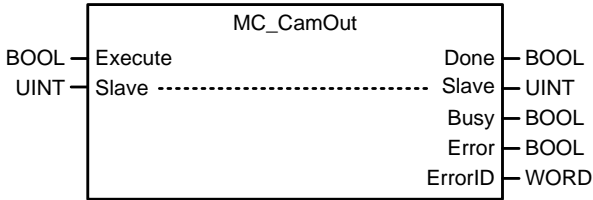
이미 동기화 위치로 이동해야 하는 구간에 위치해 있으면 MasterStartDistance 범위를 벗어날 때까지 동기화 시작점의 위치를 한 테이블 길이만큼씩 미룹니다.

MasterSyncPosition은 캠 테이블 내를 기준으로 한 값이기 때문에 MasterScaling과 SlaveScaling에 따라 실제 동기화 위치는 달라질 수 있으나, MasterOffset, MasterStartDistance 값은 영향을 받지 않습니다.

- (8) 캠 운전이 정상적으로 실행되기 시작하면 InSync 출력이 0n 되고, 한 개의 캠 테이블 운전이 완료될 때 마다 EndOfProfile 출력이 1스캔 0n 됩니다.



- (9) StartMode 에는 캠 종축의 설정합니다. 설정 불가능한 값을 설정했을 경우 에러가 발생합니다.
- (10) MasterValueSource 는 동기할 주축의 소스를 선택합니다. 0 으로 설정하면 모션 제어 모듈에서 연산된 주축의 명령 위치를, 1 로 설정하면 주축 서보드라이브에서 통신을 통해서 받은 현재위치를 기준으로 종축이 캠 운전을 합니다.
- (11) CamTableID 는 캠 운전이 적용될 캠 테이블의 번호를 설정합니다. 설정범위는 1~32 이며 입력 값이 설정 범위를 벗어나면 모션 평선 블록에서 에러 “ 0x1115 ” 이 발생합니다.
- (12) 이 모션 평선 블록이 실행 중일 때 해당 축은 “동기운전(SynchronizedMotion)” 상태입니다.

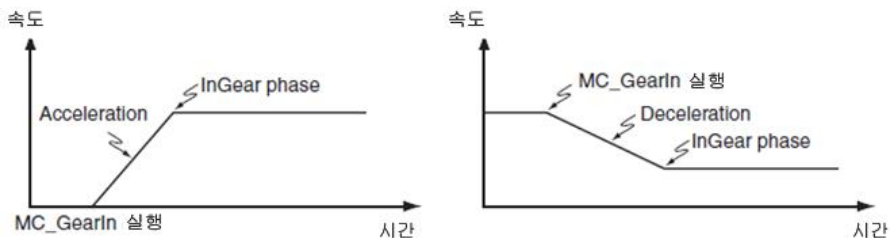
MC_CamOut		적용 기종
캠 운전 해제		XGF-M32E
평선 블록 형태		
		
입력-출력		
UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 캠 운전 정지 지령을 내립니다.
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 종축에 실행 중인 캠 운전을 즉시 해제합니다.
- (2) 캠 운전이 실행 중인 종축에 BufferMode 가 Aborting 인 모션 평선 블록이 실행되면 자동적으로 캠 운전이 해제된 후, 해당 모션 평선 블록을 수행하게 됩니다. 대신 캠 운전 해제(MC_CamOut) 모션 평선 블록을 수행하면 해당 축은 캠 운전이 해제된 시점의 속도를 유지하는 운전을 합니다. 종축을 완전히 멈추려면 정지(MC_Halt) 또는 즉시정지(MC_Stop) 모션 평선 블록을 사용하면 됩니다.
- (3) 캠 운전 해제(MC_CamOut) 모션 평선 블록을 실행하는 순간 MC_CamIn 평선 블록의 InSync 출력과 동기운전(Synchronized) 상태 플래그(_AXxx_Synchronized)는 Off 됩니다.

MC_Gear In		적용 기종
기어 운전		XGF-M32E
펄스 블록 형태		
<div style="text-align: center;"> </div>		
입력-출력		
UINT	Master	주축을 설정합니다. (1~32: 실축, 37~40: 가상축, 41~42: 엔코더)
UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 기어 운전 지령을 내립니다.
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (10.1.5. 모션 펄스 블록 실행 중 파라미터 변경 참조)
INT	RatioNumerator	기어비의 분자 값을 지정합니다(-32768 ~ 32767)
UINT	RatioDenominator	기어비의 분모 값을 지정합니다(0 ~ 65535)
UINT	MasterValueSource	동기화될 주축의 데이터를 선택합니다. 0: 주축의 명령위치에 동기화합니다. 1: 주축의 현재위치에 동기화합니다.
LREAL	Acceleration	기어운전 동기 시작시의 가속도를 지정합니다. [u/s ²]
LREAL	Deceleration	기어운전 동기 시작시의 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	가/감속도 변화율을 지정합니다 [u/s ³]
UINT	BufferMode	모션 펄스 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
출력		
BOOL	InGear	기어비를 적용하여 기어 운전 중임을 나타냅니다.

BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 설정한 기어비에 따라서 주축과 종축을 속도 동기 시키는 운전입니다.
- (2) 기어 운전을 해제하기 위해서는 해당 축에 기어 운전 해제(MC_GearOut) 명령을 내리거나 다른 모션 평선 블록을 동작시켜야 합니다.
- (3) RatioNumerator, RatioDenominator 는 종축에 적용될 기어비의 분자 및 분모 값을 각각 설정합니다. 분자값을 음수로 설정하면 종축의 회전 방향은 주축의 반대가 됩니다.
- (4) MasterValueSource 는 동기의 기준이 되는 주축의 데이터를 선택합니다. 0 으로 설정하면 모션 제어 모듈의 주축의 명령위치를, 1 로 설정하면 현재위치를 기준으로 동기 운전을 합니다. 이 외의 값을 설정했을 때는 모션 평선 블록의 Error 가 온(On)되고 ErrorID 에 “0x1114” 이 발생합니다.
- (5) 이 모션 평선 블록이 실행되면 종축은 해당 기어비에 맞는 속도로 가속/감속하여 주축과 동기되며, 동기가 완료되었을 때 InGear 출력이 온(On)됩니다.
- (6) 이 모션 평선 블록이 실행 중일 때 종축은 ‘동기운전(SynchronizedMotion)’상태입니다.



- (7) ContinuousUpdate 입력을 On 시키면 변경된 파라미터가 적용됩니다.
RatioNumerator, RatioDenominator, Acceleration, Deceleration 입력만 업데이트 가능합니다.(단, InGear=On 인 경우에는 RatioNumerator, RatioDenominator 만 업데이트 됩니다.)

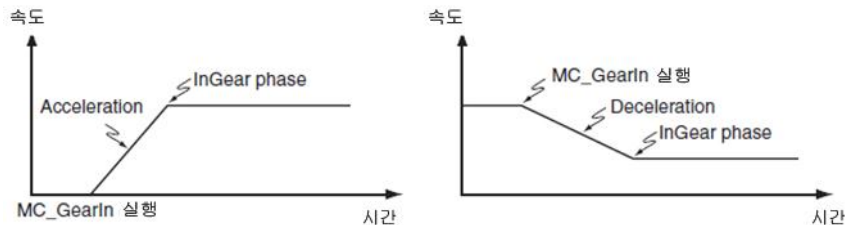
MC_GearOut		적용 기종
기어운전 해제		XGF-M32E
평선 블록 형태		
입력-출력		
UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 기어 운전 해제 지령을 내립니다.
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

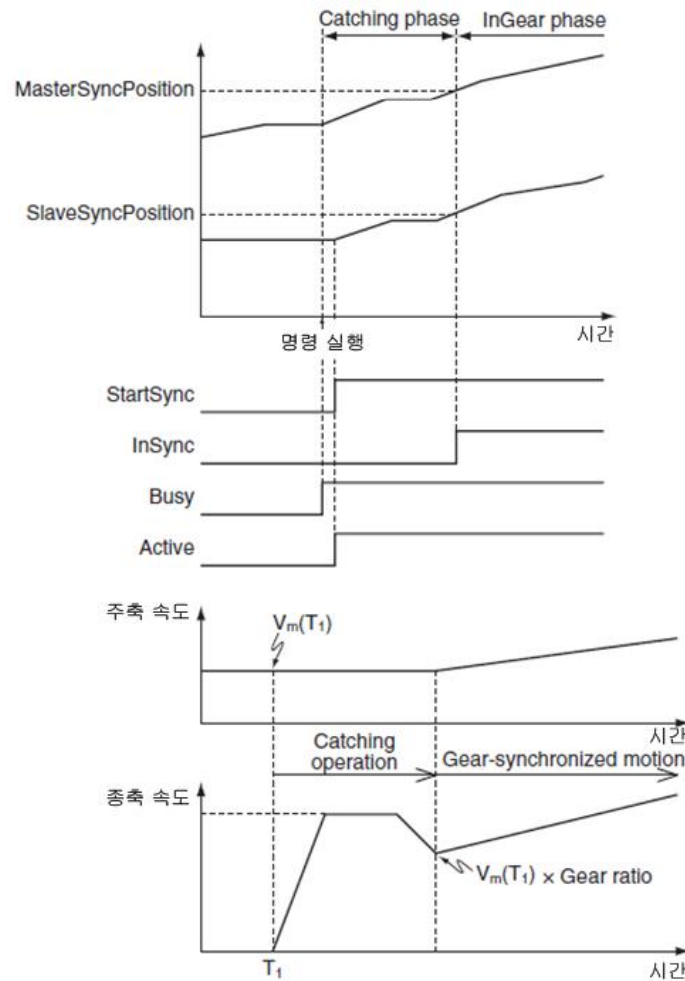
- (1) 이 모션 평선 블록은 종축에 실행 중인 기어 운전을 즉시 해제합니다.
- (2) 기어 운전이 실행 중인 종축에 **Buffermode** 가 **Aborting** 인 모션 평선 블록이 실행되면 자동적으로 기어 운전이 해제된 후, 해당 모션 평선 블록을 수행하게 됩니다. 만약 기어 운전 해제(**MC_GearOut**) 모션 평선 블록만을 수행한다면 해당 축은 기어 운전이 해제된 시점의 속도를 유지하는 운전을 합니다. 종축을 완전히 멈추려면 정지(**MC_Halt**) 또는 즉시정지(**MC_Stop**) 모션 평선 블록을 사용하면 됩니다.

MC_Gear InPos		적용 기종
위치치정 기어운전		XGF-M32E
평선 블록 형태		
입력-출력		
UINT	Master	주축을 설정합니다. (1~32: 실축, 37~40: 가상축, 41~42: 엔코더)
UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 기어 운전 지령을 내립니다.
INT	RatioNumerator	기어비의 분자 값을 지정합니다(-32768~32767)
UINT	RatioDenominator	기어비의 분모 값을 지정합니다(0~65535)
UINT	MasterValueSource	동기화될 주축 값의 기준을 선택합니다. 0(mcSetValue): 주축의 목표위치에 동기화합니다. 1(mcActualValue): 주축의 현재위치에 동기화합니다.
LREAL	MasterSyncPosition	기어운전이 시작될 주축의 위치를 지정합니다.
LREAL	SlaveSyncPosition	기어운전이 시작될 종축의 위치를 지정합니다.
LREAL	SyncMode	미사용
LREAL	MasterStartDistance	동기화를 시작할 주축의 거리를 지정합니다.
LREAL	Velocity	동기화 시작시의 종축의 최대 속도를 지정합니다. [u/s]
LREAL	Acceleration	동기화 시작시의 종축의 최대 가속도를 지정합니다. [u/s ²]

LREAL	Deceleration	동기화 시작시의 종축의 최대 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	동기화 시작시의 종축의 최대 가/감속도 변화율을 지정합니다 [u/s ³]
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
출력		
BOOL	InSync	지정한 기어비가 적용되어 기어운전이 정상 수행중임을 나타냅니다.
BOOL	StartSync	동기화 시작중임을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 설정된 위치에서 주축과 종축의 특정 위치에서 설정한 기어 비에 따라 주축과 종축을 속도 동기 시키는 운전입니다.
- (2) 기어 운전을 정지하기 위해서는 종축에 기어 운전 해제(MC_GearOut) 명령을 내리거나 다른 모션 평선 블록을 동작시켜야 합니다.
- (3) RatioNumerator, RatioDenominator는 종축에 적용될 기어비의 분자 분모 값을 각각 지정합니다. 분자값을 음수로 설정하면 종축의 회전방향은 주축의 역방향이 됩니다.
- (4) MasterValueSource는 동기화할 주축의 소스를 선택합니다. 0(mcSetValue)으로 설정하면 현재 모션 제어 주기에서의 주축의 목표 위치를, 1(mcActualValue)로 설정하면 현재 모션 제어 주기에서 피드백 받은 주축의 현재 위치를 소스로 동기화 합니다. 이외의 값을 설정했을 경우 “에러 0x10D1” 이 발생합니다.
- (5) MasterSyncPosition입력과 SlaveSyncPosition입력에는 기어운전이 동기화 완료될 주축과 종축의 위치를 각각 입력합니다. MasterStartDistance입력에는 종축이 동기화를 시작할 거리를 입력하며, MasterSyncPosition입력에 설정한 위치에서 MasterStartDistance입력에 설정한 거리만큼 떨어진 위치부터 종축이 동기화를 시작합니다.
- (6) 동기화가 시작되면 StartSync 출력이 온(On)됩니다. 동기화가 완료되어 기어운전이 시작되면 StartSync 출력은 오프(Off)되고, InSync 출력이 온(On)됩니다.
- (7) 이 모션 평선 블록이 실행 중일 때 종축은 ‘동기운전(SynchronizedMotion)’ 상태입니다.





(8) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.

RatioNumerator, RatioDenominator, MasterSyncPosition, SlaveSyncPosition, MasterStartDistance, Velocity, Acceleration, Deceleration 입력만 업데이트 가능합니다. (단, InGear=On 인 경우에는 RatioNumerator, RatioDenominator 만 업데이트 가능합니다.)

MC_Phasing		적용 기종
위상 보정		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <p>The diagram shows a central box labeled 'MC_Phasing'. On the left side, there are inputs: BOOL Execute, UINT Master, UINT Slave, LREAL PhaseShift, LREAL Velocity, LREAL Acceleration, LREAL Deceleration, and LREAL Jerk. On the right side, there are outputs: BOOL Done, UINT Master, UINT Slave, BOOL Busy, BOOL Active, BOOL CommandAborted, BOOL Error, WORD ErrorID, and LREAL CoveredPhaseShift.</p> </div>		
입력-출력		
UINT	Master	주축을 설정합니다. (1~32: 실축, 37~40: 가상축, 41~42: 엔코더)
UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 위상 보정 명령을 내립니다.
LREAL	PhaseShift	주축의 위상 보정량을 지정합니다.
LREAL	Velocity	위상 보정 속도를 지정합니다. [u/s]
LREAL	Acceleration	가속도를 지정합니다. [u/s ²]
LREAL	Deceleration	감속도를 지정합니다. [u/s ²]
LREAL	Jerk	가/감속도 변화율을 지정합니다 [u/s ³]
출력		
BOOL	Done	지정한 위상 보정 거리 도달 완료 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

LREAL	CoveredPhaseShift	위상 보정이 실행되는 동안 반영된 위상 보정량을 연속적으로 출력합니다.
-------	-------------------	---

- (1) 이 모션 평선 블록은 동기 제어 운전 중인 축의 위상 보정을 실행합니다. 동기 제어 운전 중인 종축이 참조하는 주축의 위치에 대해서 위상 보정을 실행하여 종축은 보정된 주축의 위치에 대해 동기 제어 운전됩니다.
- (2) 위상 보정 명령이 실행되면 **PhaseShift** 에 설정된 위상 보정량을 **Velocity / Acceleration / Deceleration / Jerk** 설정으로 운전 하면서 주축의 현재 위치를 위상 보정합니다.
- (3) 위상 보정이 실행되더라도 실제 주축의 명령 위치 및 현재 위치가 변화하는 것은 아니며, 종축이 동기 제어 운전 시 참조하는 주축 위치 값에 대해서 위상 보정이 실행됩니다. 즉, 주축은 종축에 의해 위상 보정 제어가 실행되고 있음을 알지 못합니다.
- (4) 명령이 종료되기 전에 평선 블록을 재 실행(**Execute** 입력을 **On**) 시키면 그 지점에서 위상 보정량 만큼 다시 위상 보정을 실행한다. 즉, 위상 보정량은 실행 지점을 기준으로 상대 값으로 동작합니다.
- (5) 위상 보정 명령 실행 후 위상 보정량(**PhaseShift**)에 도달하면 **Done** 출력이 **On**됩니다.

MC_AddAxisToGroup		적용 기종
그룹 축 추가		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_AddAxisToGroup direction TB Execute[Execute] AxesGroup[AxesGroup] Axis[Axis] IdentInGroup[IdentInGroup] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup Axis --- Axis IdentInGroup --- Busy Busy --- Error Error --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	해당 축을 추가할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
UINT	Axis	해당 그룹에 추가할 축을 설정합니다. (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 그룹 축 추가 지령을 내립니다.
UINT	IdentInGroup	해당 그룹에서 사용될 해당 축의 ID 를 설정합니다. (1 ~ 4)
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 Axis 지정된 축을 AxesGroup 입력에 지정된 축 그룹에 추가합니다.
- (2) IdentInGroup 으로 지정되는 축 그룹 내 ID 는 각각의 축이 고유한 값을 가져야 합니다(각각의 축의 ID 가 모두 달라야 합니다). 각 축 그룹에는 최대 4 개의 축을 포함할 수 있으며, 축 ID 는 1~4 의 범위로 지정 가능합니다. 지정한 축 번호가 범위를 벗어나면 “에러 0x0006” 이 발생하고 축 그룹 내 번호가 중복되는 경우 “에러 0x2051” 이 발생합니다.
- (3) XG-PM 내 축 그룹 파라미터 설정에서도 동일하게 축 그룹 설정이 가능합니다.

MC_RemoveAxisFromGroup		적용 기종
그룹 축 제외		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_RemoveAxisFromGroup Execute[Execute] AxesGroup[AxesGroup] IdentInGroup[IdentInGroup] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup IdentInGroup --- IdentInGroup Done --- Done Busy --- Busy Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	해당 축을 제외할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 그룹 축 제외 지령을 내립니다.
UINT	IdentInGroup	해당 그룹에서 제외할 해당 그룹 내 축 번호를 설정합니다.
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 AxesGroup 입력에 지정된 축 그룹에서 IdentInGroup 로 지정된 축을 제외합니다.
- (2) 축 그룹이 그룹 비활성화(GroupDisabled), 그룹 대기(GroupStandBy), 그룹 에러정지(GroupErrorStop) 상태가 아닐 때 그룹 축 제외를 수행하면 현재 축 그룹의 상태에 따라 “에러 0x2003 또는 0x2004 또는 0x2005” 가 발생하며 축은 제외되지 않습니다. 즉, 축 그룹이 완전히 정지하지 않은 상태에서는 축을 제외할 수 없습니다.
- (3) 지정된 축 그룹에서 구성 축이 모두 제외되어 그룹에 속한 축이 없을 경우 해당 그룹은 그룹 비활성화 (GroupDisabled) 상태로 전환 됩니다.

MC_UngroupAllAxes		적용 기종
그룹 내 모든 축 제외		XGF-M32E
평선 블록 형태		
<pre> graph LR subgraph MC_UngroupAllAxes direction TB Execute[Execute] AxesGroup[AxesGroup] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup Busy --- Busy Error --- Error ErrorID --- ErrorID </pre>		
입력-출력		
UINT	AxesGroup	모든 축을 제외할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 그룹 내 모든 축 제외 지령을 내립니다.
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 AxesGroup 입력에 지정된 축 그룹에 속한 축을 모두 제외합니다.
- (2) 축 그룹이 그룹 비활성화(GroupDisabled), 그룹 대기(GroupStandBy), 그룹 에러정지(GroupErrorStop) 상태가 아닐 때 이 모션 평선 블록을 실행하면 “에러 0x2003 또는 0x2004 또는 0x2005” 가 발생하며 축은 제외되지 않습니다. 즉, 축 그룹이 완전히 정지하지 않은 상태에서는 축을 제외할 수 없습니다.
- (3) 그룹에 속한 축이 성공적으로 제외되면 해당 그룹은 그룹 비활성화(GroupDisabled) 상태로 전환 됩니다.

MC_GroupEnable		적용 기종
그룹 활성화		XGF-M32E
평선 블록 형태		
<pre> graph LR subgraph MC_GroupEnable Execute[Execute] AxesGroup[AxesGroup] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup Busy --- Busy Error --- Error ErrorID --- ErrorID </pre>		
입력-출력		
UINT	AxesGroup	활성화할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 그룹 활성화 지령을 내립니다.
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 AxesGroup 입력에 지정된 축 그룹을 활성화 시키는 모션 평선 블록입니다.
- (2) 그룹 비활성화(GroupDisable) 상태인 축 그룹에 이 명령을 내리면 해당 축 그룹은 그룹 대기(GroupStandby) 상태로 전환됩니다.
- (3) 이 모션 평선 블록은 해당 그룹의 각 축의 전원 상태에 영향을 미치지 않습니다.

MC_GroupDisable		적용 기종
그룹 비활성화		XGF-M32E
평선 블록 형태		
<pre> graph LR subgraph MC_GroupDisable Execute[Execute] AxesGroup[AxesGroup] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup Busy --- Busy Error --- Error ErrorID --- ErrorID </pre>		
입력-출력		
UINT	AxesGroup	비활성화할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 그룹 비활성화 지령을 내립니다.
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 AxesGroup 입력에 지정된 축 그룹을 비활성화 시키는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록을 실행한 축 그룹은 그룹 비활성화(GroupDisabled) 상태로 전환됩니다.
- (3) 이 모션 평선 블록은 해당 그룹의 각 축의 전원 상태에 영향을 미치지 않습니다.

MC_GroupHome		적용 기종
그룹 원점복귀 운전		XGF-M32E
평선 블록 형태		
입력-출력		
UINT	AxesGroup	원점복귀할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 그룹 원점복귀 지령을 내립니다.
LREAL[]	Position	기준신호가 감지되었을 때의 축 별 절대위치를 지정합니다.
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 해당 축 그룹을 제어중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 AxesGroup 입력에 지정된 축 그룹에 원점복귀 지령을 내리는 모션 평선 블록입니다.
- (2) 원점 복귀 방식은 사전에 해당 축의 서보 파라미터에 지정된 대로 동작합니다.
- (3) Position 입력에는 기준 신호(Reference Signal)이 감지되었을 때, 또는 원점복귀가 완료되었을 때 설정될 절대 위치를 배열로 지정합니다. 배열 내의 값들과 그룹 내 축은 [①, ②, ③, ④]의 순서로 대응됩니다. (①~④ 는 축 그룹 내의 축 ID)
- (4) 이 모션 평선 블록이 수행 중일 때 축 그룹은 '그룹 원점복귀중(Group Homing)'상태이며, 모션 평선 블록이 완료되면

‘그룹 대기(GroupStandby)’상태로 전환됩니다.

(5) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.

Position 입력만 업데이트 가능합니다.

MC_GroupSetPosition		적용 기종
그룹 현재위치 설정		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_GroupSetPosition Execute[Execute] AxesGroup[AxesGroup] Position[Position] Relative[Relative] ExecutionMode[ExecutionMode] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup Position --- Position Relative --- Error ExecutionMode --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	현재위치 설정을 할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 그룹 현재위치 설정 지령을 내립니다.
LREAL[]	Position	위치를 지정합니다.
BOOL	Relative	0 - 위치 값 = 절대위치, 1 - 위치 값 = 상대위치
UINT	ExecuteMode	0 - 위치 값 즉시 적용, 1 - Buffermode 의 'Buffered'와 동일한 시점에 적용
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 해당 축 그룹을 제어중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축 그룹의 현재위치를 설정합니다.
- (2) Position 입력에는 그룹 내 각 축의 위치를 배열로 지정합니다. 이 모션 평선 블록을 실행했을 때 Relative 입력이 오프(Off) 상태라면 해당 축의 위치는 Position 입력 값으로 대체되며, Relative 입력이 온(On) 상태라면 해당 축의 현재 위치에 Position 입력 값이 더해집니다. 배열 내의 값들과 그룹 내 축은 [①, ②, ③, ④]의 순서로 대응됩니다. (①~④ 는 축 그룹 내의 축 ID)

(3) ExecutionMode 입력은 설정 시점을 지정합니다. 0 일 때는 명령 실행 즉시 설정되며, 1 일 경우에는 순차운전 설정의 'Buffered'와 동일한 시점에 설정됩니다. 설정 불가능한 값을 설정했을 경우 "에러 0x201B" 가 발생합니다.

0 (mcImmediately): 모션 평선 블록 실행 즉시(Execute 입력의 상승 Edge) 파라미터 값을 변경합니다. 해당 축이 운전 중이라면 동작에 영향이 있을 수 있습니다.

1 (mcQueued): Buffermode 의 'Buffered'와 같은 시점에 변경됩니다. (10.1.4 BufferMode 입력 참조)

MC_GroupStop		적용 기종
그룹 즉시정지		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_GroupStop Execute[Execute] AxesGroup[AxesGroup] Deceleration[Deceleration] Jerk[Jerk] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup Deceleration --- Busy Jerk --- Active Done --- Done Busy --- Busy Active --- Active CommandAborted --- CommandAborted Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	즉시 정지를 할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 그룹 즉시 정지 지령을 내립니다.
LREAL	Deceleration	정지 시의 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	감속도 변화율을 지정합니다. [u/s ³]
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 해당 축 그룹을 제어중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축 그룹에 비상정지 지령을 내리는 모션 평선 블록입니다
- (2) 완전히 정지하기 전까지 해당 축 그룹은 추종하고 있던 경로선상을 이동하게 됩니다.
- (3) 그룹 즉시정지(MC_GroupStop) 모션 평선 블록을 수행하면 해당 축 그룹이 기존에 수행하고 있던 모션 평선 블록은 중단되며, 축은 '그룹 정지중(GroupStopping)'상태로 바뀝니다. 해당 축 그룹이 '그룹 정지중(GroupStopping)' 상태일 때에는 정지가 완료되기 전까지(Done 출력이 활성화 되기 전까지) 해당 축에 다른 모션 평선 블록을 내릴 수 없습니다.
- (4) CommandAborted 출력은 현재 모션 평선 블록이 실행 중 중단되었음을 나타냅니다. 그룹 즉시정지(MC_GroupStop)

명령 실행 중에는 다른 모션 평선 블록이 그룹 즉시정지(MC_GroupStop) 명령을 중단시킬 수 없으므로, 일반적으로 서버의 전원이 차단되거나 서버 오프 지령이 실행되었을 경우 또는 서버 연결이 끊어졌을 때 CommandAborted 출력이 활성화됩니다.

- (5) Execute 입력이 온(On) 되어 있거나, 축의 속도가 0 이 아니라면 축은 '그룹 정지중(GroupStopping)' 상태이며, Done 출력이 온(On) 되고 Execute 입력이 오프(Off) 되면 '그룹 대기(GroupStandBy)'상태로 전환됩니다.
- (6) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다. Deceleration, Jerk 입력만 업데이트 가능합니다.

MC_GroupHalt		적용 기종
그룹 정지		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_GroupHalt Execute[Execute] AxesGroup[AxesGroup] Deceleration[Deceleration] Jerk[Jerk] BufferMode[BufferMode] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup Deceleration --- Busy Jerk --- Active BufferMode --- CommandAborted Done --- Done Busy --- Busy Active --- Active CommandAborted --- CommandAborted Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	정지 할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 그룹 정지 지령을 내립니다.
LREAL	Deceleration	정지 시의 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	감속도 변화율을 지정합니다. [u/s ³]
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 해당 축 그룹을 제어중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 정지 지령을 내리는 모션 평선 블록입니다.
- (2) 완전히 정지하기 전까지 해당 축 그룹은 추종하고 있던 경로선상을 이동하게 됩니다.
- (3) 이 모션 평선 블록이 수행 중일 때 축은 '그룹 운전중(GroupMoving)'상태이며, 축 그룹이 완전히 정지하면 'Done'출력이 온(On)되고 그룹 상태는 '그룹 대기(GroupStandBy)'상태로 바뀝니다.
- (4) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.

Deceleration, Jerk 입력만 업데이트 가능합니다.

MC_GroupReset		적용 기종
그룹 에러 리셋		XGF-M32E
평선 블록 형태		
<pre> graph LR subgraph MC_GroupReset direction LR Execute[Execute] --- AxesGroup[AxesGroup] Done[Done] --- Busy[Busy] Error[Error] --- ErrorID[ErrorID] end Execute --- AxesGroup AxesGroup --- Done Done --- Busy Busy --- Error Error --- ErrorID </pre>		
입력-출력		
UINT	AxesGroup	에러 리셋을 할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 그룹 그룹 에러 리셋 지령을 내립니다.
출력		
BOOL	Done	모션 평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축 그룹의 에러를 리셋하는 모션 평선 블록입니다. 해당 축이 ‘그룹 에러정지(GroupErrorStop)’상태일 때 모션 평선 블록을 수행하면 현재 해당 축에 발생한 에러가 리셋되고 축 그룹은 ‘그룹 대기(GroupStandBy)’ 상태로 전환됩니다.
- (2) 이 모션 평선 블록을 실행하면 그룹 내의 각 축에 발생한 에러도 모두 리셋합니다. (각 축에 축 에러리셋 (MC_Reset) 명령을 수행한 효과와 동일합니다.)

MC_MoveLinearAbsolute		적용 기종
절대위치 직선보간 운전		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_MoveLinearAbsolute Execute[Execute] AxesGroup[AxesGroup] Position[Position] Velocity[Velocity] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] BufferMode[BufferMode] TransitionMode[TransitionMode] TransitionParameter[TransitionParameter] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup Position --- Position Velocity --- Velocity Acceleration --- Acceleration Deceleration --- Deceleration Jerk --- Jerk BufferMode --- BufferMode TransitionMode --- TransitionMode TransitionParameter --- TransitionParameter Done --- Done Busy --- Busy Active --- Active CommandAborted --- CommandAborted Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	절대위치 직선보간 운전을 할 그룹을 설정합니다. (1 ~ 16: 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 그룹에 절대위치 직선보간 운전 지령을 내립니다.
LREAL[]	Position	축 별 목표 위치를 지정합니다.
LREAL	Velocity	경로의 최대 속도를 지정합니다. [u/s]
LREAL	Acceleration	최대 가속도를 지정합니다. [u/s ²]
LREAL	Deceleration	최대 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	최대 가/감속도 변화율을 지정합니다 [u/s ³]
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)
UINT	TransitionMode	그룹 운전의 경로변경 모드를 지정합니다. (6.1.6. 그룹운전 경로 변경 설정 : TransitionMode 참조)
LREAL	TransitionParameter	그룹 운전의 경로변경 설정의 파라미터를 지정합니다. (6.1.6. 그룹운전 경로 변경 설정 : TransitionMode 참조)
출력		
BOOL	Done	지정한 위치에 도달 완료 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 그룹을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.

BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 **AxesGroup** 입력에 지정한 축 그룹에 절대위치 직선보간 지령을 시키는 모션 평선 블록입니다.
(2) 이 모션 평선 블록이 실행되면 각 축의 현재위치부터 목표위치까지 직선의 궤적으로 보간 제어를 실행하며 이동 방향은 각 축의 시작위치와 목표위치에 의해 결정됩니다.

시작위치 < 목표위치: 정방향 운전

시작위치 > 목표위치: 역방향 운전

- (3) **Position** 입력에는 그룹 내 각 축의 목표 위치를 행렬로 지정합니다. 배열 내 값들과 그룹 내 축은 [①, ②, ③, ④]의 순서로 대응됩니다. (①~④ 는 축 그룹 내의 축 ID)
(4) **Velocity, Acceleration, Deceleration, Jerk** 입력에는 보간 경로의 속도, 가속도, 감속도, 가/감속 변화율을 각각 지정합니다.

- (5) **Velocity** 는 축그룹의 보간 속도를 설정하는 것으로 각 축의 합성된 속도를 나타냅니다.

각 구성 축의 운전 속도는 다음과 같이 계산됩니다.

보간 속도 (F) = Velocity에 설정한 목표속도

$$\text{보간 이동량 (S)} = \sqrt{S_1^2 + S_2^2 + S_3^2 + S_4^2}$$

$$\text{구성축1 속도 (V}_1\text{)} = \text{보간속도 (F)} \times \frac{\text{구성축1 이동량 (S}_1\text{)}}{\text{보간 이동량 (S)}}$$

$$\text{구성축2 속도 (V}_2\text{)} = \text{보간속도 (F)} \times \frac{\text{구성축2 이동량 (S}_2\text{)}}{\text{보간 이동량 (S)}}$$

$$\text{구성축3 속도 (V}_3\text{)} = \text{보간속도 (F)} \times \frac{\text{구성축3 이동량 (S}_3\text{)}}{\text{보간 이동량 (S)}}$$

$$\text{구성축4 속도 (V}_4\text{)} = \text{보간속도 (F)} \times \frac{\text{구성축4 이동량 (S}_4\text{)}}{\text{보간 이동량 (S)}}$$

- (6) 자세한 내용은 모션 제어 모듈 메뉴얼의 직선 보간 제어 부분을 참고하시기 바랍니다.
(7) 명령이 종료되기 전에 평선 블록을 재 실행(**Execute** 입력을 **On**) 시키면 변경된 파라미터가 적용됩니다.
Velocity, Acceleration, Deceleration, Jerk, Position 입력만 업데이트 가능합니다.

MC_MoveLinearRelative		적용 기종																																	
상대위치 직선보간 운전		XGF-M32E																																	
평선 블록 형태																																			
<div style="text-align: center;"> <table border="1"> <tr> <th colspan="3">MC_MoveLinearRelative</th> </tr> <tr> <td>BOOL</td> <td>Execute</td> <td>Done</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>----- AxesGroup</td> </tr> <tr> <td>LREAL[]</td> <td>Distance</td> <td>Busy</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>Active</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td>CommandAborted</td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td>Error</td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td>ErrorID</td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> </tr> <tr> <td>UINT</td> <td>TransitionMode</td> <td></td> </tr> <tr> <td>LREAL</td> <td>TransitionParameter</td> <td></td> </tr> </table> </div>			MC_MoveLinearRelative			BOOL	Execute	Done	UINT	AxesGroup	----- AxesGroup	LREAL[]	Distance	Busy	LREAL	Velocity	Active	LREAL	Acceleration	CommandAborted	LREAL	Deceleration	Error	LREAL	Jerk	ErrorID	UINT	BufferMode		UINT	TransitionMode		LREAL	TransitionParameter	
MC_MoveLinearRelative																																			
BOOL	Execute	Done																																	
UINT	AxesGroup	----- AxesGroup																																	
LREAL[]	Distance	Busy																																	
LREAL	Velocity	Active																																	
LREAL	Acceleration	CommandAborted																																	
LREAL	Deceleration	Error																																	
LREAL	Jerk	ErrorID																																	
UINT	BufferMode																																		
UINT	TransitionMode																																		
LREAL	TransitionParameter																																		
입력-출력																																			
UINT	AxesGroup	상대위치 직선보간 운전을 할 그룹을 설정합니다. (1 ~ 16: 1 그룹 ~ 16 그룹)																																	
입력																																			
BOOL	Execute	상승 Edge 에서 해당 그룹에 상대위치 직선보간 운전 지령을 내립니다.																																	
LREAL[]	Distance	축 별 목표 거리를 지정합니다.																																	
LREAL	Velocity	경로의 최대 속도를 지정합니다. [u/s]																																	
LREAL	Acceleration	최대 가속도를 지정합니다. [u/s ²]																																	
LREAL	Deceleration	최대 감속도를 지정합니다. [u/s ²]																																	
LREAL	Jerk	최대 가/감속도 변화율을 지정합니다 [u/s ³]																																	
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)																																	
UINT	TransitionMode	그룹 운전의 경로변경 모드를 지정합니다. (6.1.6. 그룹운전 경로 변경 설정 : TransitionMode 참조)																																	
LREAL	TransitionParameter	그룹 운전의 경로변경 설정의 파라미터를 지정합니다. (6.1.6. 그룹운전 경로 변경 설정 : TransitionMode 참조)																																	
출력																																			
BOOL	Done	지정한 위치에 도달 완료 여부를 나타냅니다.																																	
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.																																	
BOOL	Active	현재 모션 평선 블록이 그룹을 제어하는 중인지 여부를 나타냅니다																																	
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.																																	

BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 **AxesGroup** 입력으로 지정한 축 그룹에 상대위치 직선보간 지령을 내리는 모션 평선 블록입니다.
(2) 이 모션 평선 블록이 실행되면 각 축의 현재위치에서 목표거리만큼 직선의 궤적으로 보간 제어를 실행하며, 이동 방향은 각 축의 목표거리의 부호에 따라 결정됩니다.

목표거리 > 0: 정방향 운전

목표거리 < 0: 역방향 운전

- (3) **Distance** 입력에는 그룹 내 각 축의 목표거리를 배열로 지정합니다. 입력한 배열과 그룹 내 축은 지정한 축 ID 의 순서로 대응됩니다 [**ID1** 목표거리, **ID2** 목표거리, ...].
(4) **Velocity**, **Acceleration**, **Deceleration**, **Jerk** 입력에는 보간 경로의 속도, 가속도, 감속도, 가감속도 변화율을 각각 설정합니다.

- (5) **Velocity** 는 축그룹의 보간 속도를 설정하는 것으로 각 축의 합성된 속도를 나타냅니다.

각 구성 축의 운전 속도는 다음과 같이 계산됩니다.

보간 속도 (F) = Velocity에 설정한 목표속도

$$\text{보간 이동량 (S)} = \sqrt{S_1^2 + S_2^2 + S_3^2 + S_4^2}$$

$$\text{구성축1 속도 (V}_1\text{)} = \text{보간속도 (F)} \times \frac{\text{구성축1 이동량 (S}_1\text{)}}{\text{보간 이동량 (S)}}$$

$$\text{구성축2 속도 (V}_2\text{)} = \text{보간속도 (F)} \times \frac{\text{구성축2 이동량 (S}_2\text{)}}{\text{보간 이동량 (S)}}$$

$$\text{구성축3 속도 (V}_3\text{)} = \text{보간속도 (F)} \times \frac{\text{구성축3 이동량 (S}_3\text{)}}{\text{보간 이동량 (S)}}$$

$$\text{구성축4 속도 (V}_4\text{)} = \text{보간속도 (F)} \times \frac{\text{구성축4 이동량 (S}_4\text{)}}{\text{보간 이동량 (S)}}$$

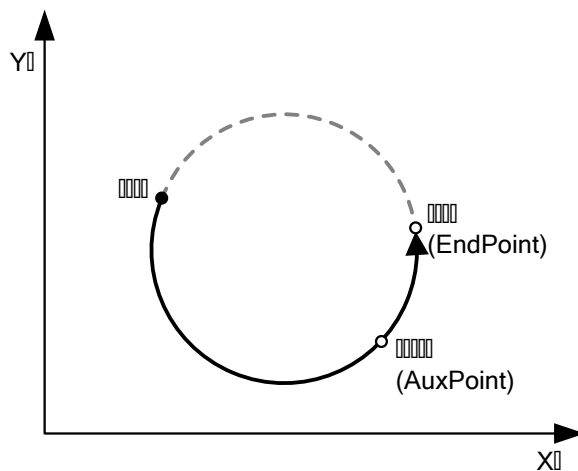
- (6) 자세한 내용은 모션 제어 모듈 메뉴얼의 직선 보간 제어 부분을 참고하시기 바랍니다.
(7) 명령이 종료되기 전에 평선 블록을 재 실행(**Execute** 입력을 **On**) 시키면 변경된 파라미터가 적용됩니다.
Velocity, **Acceleration**, **Deceleration**, **Jerk**, **Position** 입력만 업데이트 가능합니다.

MC_MoveCircularAbsolute		적용 기종																																																				
절대위치 원호보간 운전		XGF-M32E																																																				
평선 블록 형태																																																						
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">MC_MoveCircularAbsolute</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">Done</td> <td style="width: 10%;">BOOL</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>----- AxesGroup</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>CircMode</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>AuxPoint</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>EndPoint</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>PathChoice</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>TransitionMode</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>TransitionParameter</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	Done	BOOL	UINT	AxesGroup	----- AxesGroup	UINT	UINT	CircMode	Busy	BOOL	LREAL[]	AuxPoint	Active	BOOL	LREAL[]	EndPoint	CommandAborted	BOOL	UINT	PathChoice	Error	BOOL	LREAL	Velocity	ErrorID	WORD	LREAL	Acceleration			LREAL	Deceleration			LREAL	Jerk			UINT	BufferMode			UINT	TransitionMode			LREAL	TransitionParameter		
BOOL	Execute	Done	BOOL																																																			
UINT	AxesGroup	----- AxesGroup	UINT																																																			
UINT	CircMode	Busy	BOOL																																																			
LREAL[]	AuxPoint	Active	BOOL																																																			
LREAL[]	EndPoint	CommandAborted	BOOL																																																			
UINT	PathChoice	Error	BOOL																																																			
LREAL	Velocity	ErrorID	WORD																																																			
LREAL	Acceleration																																																					
LREAL	Deceleration																																																					
LREAL	Jerk																																																					
UINT	BufferMode																																																					
UINT	TransitionMode																																																					
LREAL	TransitionParameter																																																					
입력-출력																																																						
UINT	AxesGroup	절대위치 원호보간 운전을 할 그룹을 설정합니다. (1 ~ 16: 1 그룹 ~ 16 그룹)																																																				
입력																																																						
BOOL	Execute	상승 Edge 에서 해당 그룹에 절대위치 원호보간 운전 지령을 내립니다.																																																				
UINT	CirMode	원호보간 방법 설정 [0: 중간점, 1: 중심점, 2: 반지름]																																																				
LREAL[]	AuxPoint	원호보간 방법에 따른 보조점 위치를 절대좌표로 지정합니다.																																																				
LREAL[]	EndPoint	원호 끝 위치를 절대좌표로 지정합니다.																																																				
BOOL	PathChoice	원호 경로 선택 0: 시계방향, 1: 반시계 방향																																																				
LREAL	Velocity	경로의 최대 속도를 지정합니다. [u/s]																																																				
LREAL	Acceleration	최대 가속도를 지정합니다. [u/s ²]																																																				
LREAL	Deceleration	최대 감속도를 지정합니다. [u/s ²]																																																				
LREAL	Jerk	최대 가/감속도 변화율을 지정합니다 [u/s ³]																																																				
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)																																																				
UINT	TransitionMode	미사용																																																				
LREAL	TransitionParameter	미사용																																																				

출력		
BOOL	Done	지정한 위치에 도달 완료 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 그룹을 제어하는 중인지 여부를 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

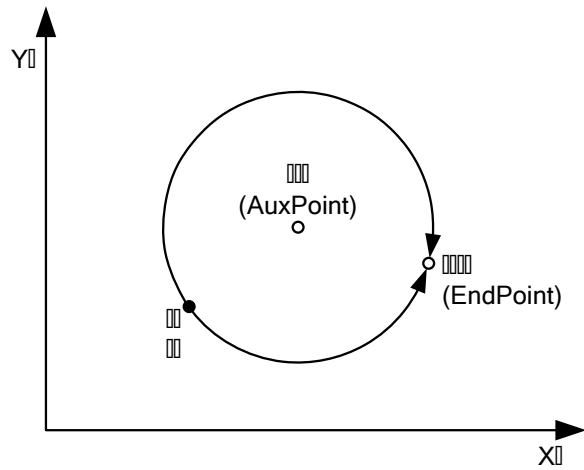
- (1) 이 모션 평선 블록은 **AxesGroup** 입력으로 지정한 축 그룹에 절대위치 원호보간 명령을 내리는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록이 시작되면 각 축은 입력된 보조점을 참고로 하는 원호궤적 보간 제어를 실행하며 이동 방향은 **PathChoice** 입력에 의해 결정됩니다. **PathChoice** 입력을 0으로 설정하면 시계방향, 1로 설정하면 반시계방향으로 원호 보간 운전을 하게 됩니다.
- (3) **AuxPoint**, **EndPoint** 입력에는 각 축 원호보간시 참고할 보조점의 절대 위치를 배열로 지정합니다. 입력한 배열과 그룹 내 축은 지정한 축 ID [**ID1**, **ID2**, **ID3**, ...]의 순서로 대응됩니다. (모션 제어 모듈에서 원호보간 운전 명령을 내릴 그룹을 구성하는 축 수는 3개이므로 **Position** 입력에는 **LEAL** 형 3개 크기의 배열이 입력되어야 합니다.)
- (4) **Velocity**, **Acceleration**, **Deceleration**, **Jerk** 입력에는 보간 경로의 속도, 가속도, 감속도, 가속도 변화율을 각각 지정합니다.
- (5) **CircMode** 입력에는 원호보간 방식을 설정합니다. **CircMode**에 지정한 값에 다른 원호보간 방식은 아래와 같습니다.
 - 중간점 지정 방식 원호보간(**BORDER**, **CircMode** = 0)

시작 위치에서 운전을 시작하고 지정된 중간 점 위치를 거쳐 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 시작 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 중간점 위치에는 **AuxPoint**에 입력된 좌표가, 목표 위치에는 **EndPoint**에 입력된 좌표가 절대값으로 대응됩니다.

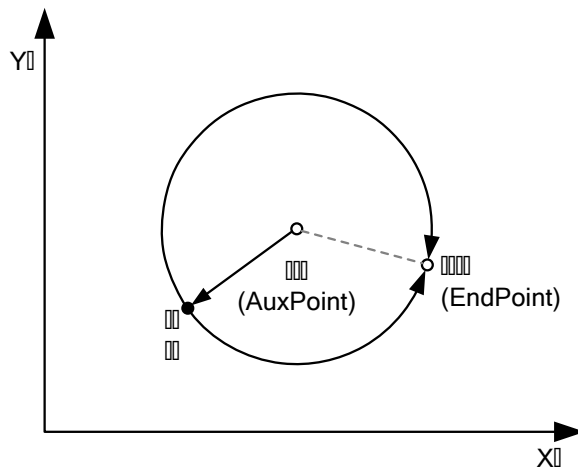


- 중심점 지정 방식을 이용한 원호 보간

현재 위치에서 운전을 시작하고 지정된 중심점 위치까지의 거리를 반지름으로 하는 원호 궤적을 따라 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 중심점 위치에는 **AuxPoint**에 입력된 좌표가, 목표점에는 **EndPoint**에 입력된 좌표가 절대값으로 대응됩니다.



- 반지름 지정 방식을 이용한 원호 보간 제어
 현재 위치에서 운전을 시작하고 현재 위치에서 반지름에 지정된 값을 반지름으로 하는 원호의 궤적을 따라 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 반지름에는 AuxPoint의 X 축에 입력된 값이, 목표점에는 EndPoint에 입력된 좌표가 절대값으로 대응됩니다

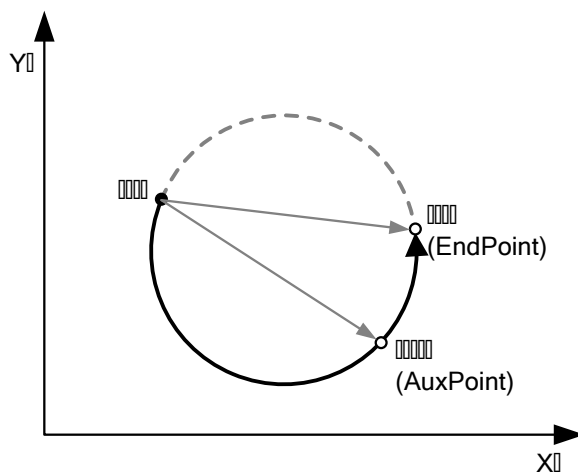


- (6) 자세한 내용은 모션 제어 모듈 메뉴얼의 원호 보간 제어 부분을 참고하시기 바랍니다.
- (7) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다. Velocity, Acceleration, Deceleration, Jerk, AuxPoint, Endpoint 입력만 업데이트 가능합니다.

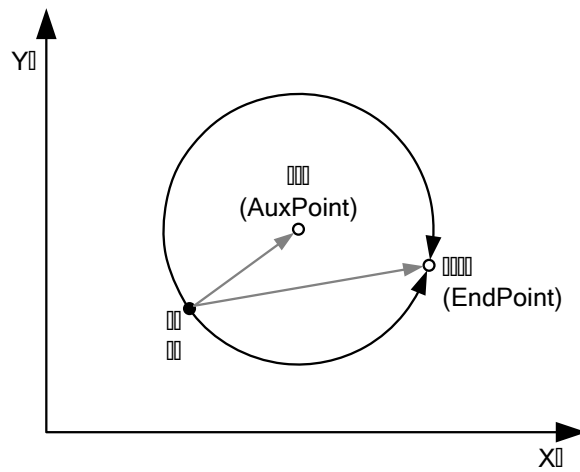
MC_MoveCircularRelative		적용 기종																																																				
상대위치 원호보간 운전		XGF-M32E																																																				
평선 블록 형태																																																						
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">MC_MoveCircularRelative</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">Done</td> <td style="width: 10%;">BOOL</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>----- AxesGroup</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>CircMode</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>AuxPoint</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>EndPoint</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>USINT</td> <td>PathChoice</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>TransitionMode</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>TransitionParameter</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	Done	BOOL	UINT	AxesGroup	----- AxesGroup	UINT	UINT	CircMode	Busy	BOOL	LREAL[]	AuxPoint	Active	BOOL	LREAL[]	EndPoint	CommandAborted	BOOL	USINT	PathChoice	Error	BOOL	LREAL	Velocity	ErrorID	WORD	LREAL	Acceleration			LREAL	Deceleration			LREAL	Jerk			UINT	BufferMode			UINT	TransitionMode			LREAL	TransitionParameter		
BOOL	Execute	Done	BOOL																																																			
UINT	AxesGroup	----- AxesGroup	UINT																																																			
UINT	CircMode	Busy	BOOL																																																			
LREAL[]	AuxPoint	Active	BOOL																																																			
LREAL[]	EndPoint	CommandAborted	BOOL																																																			
USINT	PathChoice	Error	BOOL																																																			
LREAL	Velocity	ErrorID	WORD																																																			
LREAL	Acceleration																																																					
LREAL	Deceleration																																																					
LREAL	Jerk																																																					
UINT	BufferMode																																																					
UINT	TransitionMode																																																					
LREAL	TransitionParameter																																																					
입력-출력																																																						
UINT	AxesGroup	상대위치 원호보간 운전을 할 그룹을 설정합니다. (1 ~ 16: 1 그룹 ~ 16 그룹)																																																				
입력																																																						
BOOL	Execute	상승 Edge 에서 해당 그룹에 상대위치 원호보간 운전 지령을 내립니다.																																																				
UINT	CircMode	원호보간 방법 설정 [0: 중간점, 1: 중심점, 2: 반지름]																																																				
LREAL[]	AuxPoint	원호보간 방법에 따른 보조점 위치를 시작점을 기준으로 한 상대좌표로 지정합니다.																																																				
LREAL[]	EndPoint	원호 끝 위치를 시작점을 기준으로 한 상대좌표로 지정합니다.																																																				
BOOL	PathChoice	원호 경로 선택 0: 시계방향, 1: 반시계 방향																																																				
LREAL	Velocity	경로의 최대 속도를 지정합니다. [u/s]																																																				
LREAL	Acceleration	최대 가속도를 지정합니다. [u/s ²]																																																				
LREAL	Deceleration	최대 감속도를 지정합니다. [u/s ²]																																																				
LREAL	Jerk	최대 가/감속도 변화율을 지정합니다 [u/s ³]																																																				
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (10.1.4 BufferMode 입력 참조)																																																				
UINT	TransitionMode	미사용																																																				

LREAL	TransitionParameter	미사용
출력		
BOOL	Done	지정한 위치에 도달 완료 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 그룹을 제어하는 중인지 여부를 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 **AxesGroup** 입력으로 지정한 축 그룹에 상대위치 원호보간 명령을 내리는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록이 시작되면 각 축은 입력된 보조점을 참고로 하는 원호궤적 보간 제어를 실행하며 이동 방향은 **PathChoice** 입력에 의해 결정됩니다. **PathChoice** 입력을 0 으로 설정하면 시계방향, 1 로 설정하면 반시계방향으로 원호 보간 운전을 하게 됩니다.
- (3) **AuxPoint**, **EndPoint** 입력에는 각 축 원호보간시 참고할 보조점의 상대 위치를 배열로 지정합니다. 입력한 배열과 그룹 내 축은 지정한 축 ID [**ID1**, **ID2**, **ID3**, ...]의 순서로 대응됩니다. (모션 제어 모듈에서 원호보간 운전 명령을 내릴 그룹을 구성하는 축 수는 3 개이므로 **Position** 입력에는 **LEAL** 형 3 개 크기의 배열이 입력되어야 합니다.)
- (4) **Velocity**, **Acceleration**, **Deceleration**, **Jerk** 입력에는 보간 경로의 속도, 가속도, 감속도, 가속도 변화율을 각각 지정합니다.
- (5) **CircMode** 입력에는 원호보간 방식을 설정합니다. **CircMode** 에 지정한 값에 다른 원호보간 방식은 아래와 같습니다.
 - 중간점 지정 방식 원호보간(**BORDER**, **CircMode** = 0)
 현재 위치에서 운전을 시작하고 지정된 중간 점 위치를 거쳐 목표 위치 까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 중간점 위치에는 **AuxPoint** 에 입력된 좌표가, 목표점 에는 **EndPoint** 에 입력된 좌표가 상대값으로 대응됩니다.

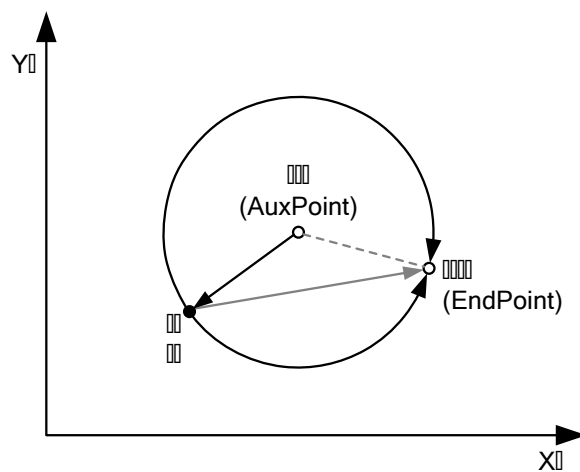


- 중심점 지정 방식을 이용한 원호 보간
 현재 위치에서 운전을 시작하고 지정된 중심점 위치까지의 거리를 반지름으로 하는 원호 궤적을 따라 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 중심점 위치에는 **AuxPoint** 에 입력된 좌표가, 목표점에는 **EndPoint** 에 입력된 좌표가 상대값으로 대응됩니다.

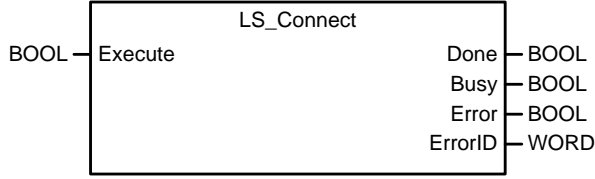


- 반지름 지정 방식을 이용한 원호 보간 제어

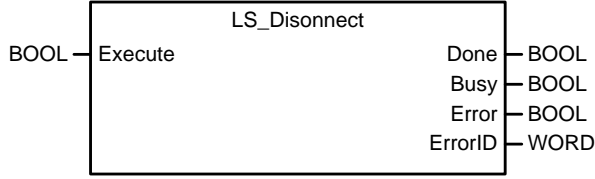
현재 위치에서 운전을 시작하고 현재 위치에서 반지름에 지정된 값을 반지름으로 하는 원호의 궤적을 따라 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 반지름에는 AuxPoint의 X 축에 입력된 값이, 목표점에는 EndPoint에 입력된 좌표가 상대값으로 대응됩니다



(6) 자세한 내용은 모션 제어 모듈 메뉴얼의 원호 보간 제어 부분을 참고하시기 바랍니다.

LS_Connect		적용 기종
통신 연결		XGF-M32E
펄스 블록 형태		
 <pre> graph LR subgraph LS_Connect Execute[Execute] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done Execute --- Busy Execute --- Error Execute --- ErrorID </pre>		
입력		
BOOL	Execute	상승 Edge 에서 해당 모듈에 통신 연결 지령을 내립니다.
출력		
BOOL	Done	통신연결 완료 여부를 나타냅니다.
BOOL	Busy	모션 펄스 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 펄스 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 펄스 블록은 모듈에 서보 드라이브나 외부 입출력 기기와 통신 연결 지령을 내리는 모션 펄스 블록입니다.
- (2) 슬레이브 기기들이 정상적으로 연결되면 **Done** 이 온(On)되고 **Busy** 가 오프(Off)됩니다.
- (3) 통신 연결 중 에러가 발생하면 **Error** 가 온(On)되고 **ErrorID** 에 원인에 따라 에러번호가 출력됩니다.

LS_Disconnect		적용 기종
통신 연결 끊기		XGF-M32E
평선 블록 형태		
		
입력		
BOOL	Execute	상승 Edge 에서 해당 모듈에 통신 연결 끊기 지령을 내립니다.
출력		
BOOL	Done	통신연결 끊기 완료 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 모듈에 서보드라이브나 외부 입출력 기기들 과의 통신 연결을 끊으라는 지령을 내립니다.
- (2) 통신 슬레이브와의 연결이 해제되면 **Done** 이 온(On)되고 **Busy** 가 오프(Off)됩니다.
- (3) 통신 연결 끊기 실행 중 에러가 발생하면 **Error** 가 온(On)되고 에러 상황에 따라 **ErrorID** 에 에러 번호가 출력됩니다.

LS_ReadServoParameter		적용 기종
서보 파라미터 읽기		XGF-M32E
펄스 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_ReadServoParameter Execute[Execute] Axis[Axis] Index[Index] SubIndex[SubIndex] Length[Length] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] Value[Value] end Execute --- Done Axis --- Axis Index --- Busy SubIndex --- Error Length --- ErrorID Value --- Value </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축을 설정합니다. (1~32: 실축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 서보파라미터 읽기 지령을 내립니다.
UINT	Index	읽을 서보파라미터 Object 의 Index 를 설정합니다. (0x2000~0x9FFF)
UINT	SubIndex	읽을 서보파라미터 Object 의 SubIndex 를 설정합니다. (0 ~ 255)
UINT	Length	읽을 서보파라미터 Object 의 길이를 Byte 단위로 설정합니다. (1 ~ 4)
출력		
BOOL	Done	서보파라미터를 성공적으로 읽었음을 나타냅니다.
BOOL	Busy	모션 펄스 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 펄스 블록 실행 중 발생한 에러번호를 출력합니다.
DINT	Value	읽어온 서보파라미터 값을 출력합니다.

- (1) 이 모션 펄스 블록은 해당 축 서보드라이브의 파라미터 (CoE Object) 값을 읽는 모션 펄스 블록이며, Axis 입력으로 지정된 축의 Index, SubIndex 로 지정된 위치의 서보파라미터 값을 Length 크기만큼 읽어 Value 출력에 표시합니다.
- (2) Value 출력은 모션 펄스 블록이 실행되면 0 으로 소거되었다가 실행이 완료되면 (Done 출력 온(On)) 읽어온 값으로 출력됩니다.
- (3) Index 입력은 아래와 같이 설정할 수 있습니다. 설정값 이외의 값으로 설정했을 경우 “에러 0x1F12” 가 발생합니다.

설정값	내용
16#0000 ~ 16#0FFF	Data Type Description
16#1000 ~ 16#1FFF	Communication objects
16#2000 ~ 16#5FFF	Manufacturer Specific Profile Area
16#6000 ~ 16#9FFF	Standardized Device Profile Area

- (4) SubIndex 에는 0~255 범위의 값을 입력할 수 있으며, 범위 이외의 값을 설정했을 경우 “에러 0x1F12” 가 발생합니다.
- (5) Length 에는 1~4 범위의 값을 입력할 수 있으며, 이는 1~4 Byte 를 의미합니다. 범위 이외의 값을 설정했을 경우 “에러 0x1F12” 가 발생합니다.

LS_WriteServoParameter		적용 기종
서보 파라미터 쓰기		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_WriteServoParameter Execute[Execute] Axis[Axis] Index[Index] SubIndex[SubIndex] Length[Length] Value[Value] ExecutionMode[ExecutionMode] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done Axis --- Axis Index --- Index SubIndex --- SubIndex Length --- Length Value --- Value ExecutionMode --- ExecutionMode Done --- Done Busy --- Busy Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축을 설정합니다. (1~32: 실축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 서보파라미터 쓰기 지령을 내립니다.
UINT	Index	쓸 서보파라미터 Object 의 Index 를 설정합니다. (0x2000~0x9FFF)
UINT	SubIndex	쓸 서보파라미터 Object 의 SubIndex 를 설정합니다. (0 ~ 255)
UINT	Length	쓸 서보파라미터 Object 의 길이를 Byte 단위로 설정합니다. (1 ~ 4)
DINT	Value	서보 파라미터에 쓸 값을 설정합니다.
UINT	ExecuteMode	서보파라미터 쓰기를 수행할 시점을 지정합니다. 0: 즉시 실행 1: BufferMode 의 'Buffered'와 동일한 시점에 적용
출력		
BOOL	Done	서보파라미터를 성공적으로 읽었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축 서보드라이브의 파라미터 (CoE Object) 값을 쓰는 모션 평선 블록이며, Axis 입력으로 지정된 축의 Index, SubIndex 로 지정된 위치의 서보파라미터에 Length 크기만큼 Value 에 입력된 값을 씁니다.
- (2) Index 입력은 아래와 같이 설정할 수 있습니다. 설정값 이외의 값으로 설정했을 경우 “에러 0x1F12” 가 발생합니다.

설정값	내용
16#0000 ~ 16#0FFF	Data Type Description
16#1000 ~ 16#1FFF	Communication objects
16#2000 ~ 16#5FFF	Manufacturer Specific Profile Area
16#6000 ~ 16#9FFF	Standardized Device Profile Area

- (3) SubIndex 에는 0~255 범위의 값을 입력할 수 있으며, 범위 이외의 값을 설정했을 경우 “에러 0x1F12” 가 발생합니다.
- (4) Length 에는 1~4 범위의 값을 입력할 수 있으며, 이는 1~4 Byte 를 의미합니다. 범위 이외의 값을 설정했을 경우 “에러 0x1F12” 가 발생합니다.
- (5) ExecutionMode 에는 파라미터가 쓰여질 시점을 설정하며 아래와 같은 값을 설정할 수 있습니다. 설정 불가능한 값을 설정했을 경우 “에러 0x101B” 가 발생합니다.
- 0 (mcImmediately): 모션 펄스 블록 실행 즉시(Execute 입력의 상승 Edge) 파라미터 값을 변경합니다. 해당 축이 구동 중이라면 동작에 영향이 있을 수 있습니다.
 - 1 (mcQueued): BufferMode 의 ‘Buffered’ 와 같은 시점에 변경됩니다. (10.1.4. BufferMode 입력 참조)
- (6) ExecutionMode 를 1 로 하여 명령이 실행된 후 실행 대기 중일 때 BufferMode 0(Aborting)으로 다른 명령이 실행되어 중단된 경우는 Error 는 1, ErrorId 는 0 으로 출력됩니다.

LS_EncoderPreset		적용 기종
엔코더 현재위치 설정		XGF-M32E
펄스 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_EncoderPreset Execute[Execute] Encoder[Encoder] Position[Position] Relative[Relative] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done Encoder --- Busy Position --- Error Relative --- ErrorID </pre> </div>		
입력		
BOOL	Execute	상승 Edge 에서 해당 엔코더의 위치를 지정합니다.
UINT	Encoder	위치를 설정할 엔코더를 설정합니다. (1~2: 엔코더 1~엔코더 2)
LREAL	Position	설정할 위치를 지정합니다. [u]
BOOL	Relative	0 – 절대 좌표 위치 1 – 상대 좌표 위치
출력		
BOOL	Done	모션 펄스 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 펄스 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 펄스 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 펄스 블록은 해당 엔코더의 현재위치를 설정합니다.
- (2) Position 입력에는 위치를 지정합니다. 모션 펄스 명령을 실행했을 때 Relative 입력이 오프(Off) 상태라면 해당 축의 위치는 Position 입력 값으로 대체되며, Relative 입력이 온(On) 상태라면 해당 축의 현재 위치에 Position 입력 값이 더해집니다.

LS_Jog		적용 기종
조그 운전		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_Jog direction LR subgraph Inputs Enable[Enable] Axis[Axis] Direction[Direction] LowHigh[Low/High] end subgraph Outputs Enabled[Enabled] Busy[Busy] Error[Error] ErrorID[ErrorID] end Enable --- Enabled Axis --- Axis Direction --- Busy LowHigh --- Error ErrorID --- ErrorID end </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축을 설정합니다. (1~32: 실축)
입력		
BOOL	Enable	입력이 온(On) 되어 있는 동안 해당 축에 조그운전 지령을 내립니다.
BOOL	Direction	조그 운전시 회전 방향을 설정합니다. (0: 정방향, 1: 역방향)
BOOL	Low/High	조그 운전시 조그 속도를 설정합니다. (0: 조그 저속 운전, 1: 조그 고속 운전)
출력		
BOOL	Enabled	해당 축이 조그 운전 중임을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 축에 조그 운전을 시키는 모션 평선 블록입니다.
- (2) 조그 운전은 테스트를 위한 수동 운전 기능으로 시스템의 동작, 배선상태 검사 및 티칭을 위한 위치 어드레스 확인용으로 사용되며, 속도를 고속과 저속으로 구분하여 사용할 수 있습니다.
- (3) **Enable** 입력이 온(On)인 상태(조그 운전 중인 상태)에서 **Low/High** 에 설정된 값을 변경하면 조그 운전을 멈추지 않은 상태에서 속도 변경이 이루어지며, **JOG_DIR** 에 설정된 값을 변경하면 감속정지 한 후 방향을 바꾸어 조그 운전을 계속합니다.

LS_ReadCamData		적용 기종																																													
캠 데이터 읽기		XGF-M32E																																													
평선 블록 형태																																															
<div style="text-align: center; border: 1px solid black; padding: 10px;"> <p>LS_ReadCamData</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td>Enable</td> <td style="width: 30%;"></td> <td>Done</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td style="text-align: center;">-----</td> <td>Axis</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>CamTable ID</td> <td></td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterPoint</td> <td></td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>SlavePoint</td> <td></td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td></td> <td></td> <td></td> <td>StartSlope</td> <td>LREAL</td> </tr> <tr> <td></td> <td></td> <td></td> <td>EndSlope</td> <td>LREAL</td> </tr> <tr> <td></td> <td></td> <td></td> <td>CamPointNum</td> <td>UINT</td> </tr> <tr> <td></td> <td></td> <td></td> <td>CamCurveSel</td> <td>Array [4] of DWORD</td> </tr> </table> </div>			BOOL	Enable		Done	BOOL	UINT	Axis	-----	Axis	UINT	UINT	CamTable ID		Busy	BOOL	LREAL	MasterPoint		Error	BOOL	LREAL	SlavePoint		ErrorID	WORD				StartSlope	LREAL				EndSlope	LREAL				CamPointNum	UINT				CamCurveSel	Array [4] of DWORD
BOOL	Enable		Done	BOOL																																											
UINT	Axis	-----	Axis	UINT																																											
UINT	CamTable ID		Busy	BOOL																																											
LREAL	MasterPoint		Error	BOOL																																											
LREAL	SlavePoint		ErrorID	WORD																																											
			StartSlope	LREAL																																											
			EndSlope	LREAL																																											
			CamPointNum	UINT																																											
			CamCurveSel	Array [4] of DWORD																																											
입력-출력																																															
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)																																													
입력																																															
BOOL	Enable	입력이 활성화된 동안 해당 캠 데이터를 읽습니다.																																													
UINT	CamTableID	읽을 캠 테이블의 번호를 지정합니다. (1~32)																																													
LREAL	MasterPoint	설정된 디바이스를 선두번지로 하는 영역에 해당 캠 테이블의 MasterPoint 값을 출력합니다.																																													
LREAL	SlavePoint	설정된 디바이스를 선두번지로 하는 영역에 해당 캠 테이블의 SlavePoint 값을 출력합니다.																																													
출력																																															
BOOL	Valid	평선 블록 출력의 유효 여부를 나타냅니다.																																													
BOOL	Busy	평선 블록 수행이 완료되지 않았음을 나타냅니다.																																													
BOOL	Error	에러 발생 여부를 나타냅니다.																																													
WORD	ErrorID	평선 블록 실행 중 발생한 에러번호를 출력합니다.																																													
LREAL	StartSlope	해당 캠 테이블의 StartSlope 값을 출력합니다.																																													
LREAL	EndSlope	해당 캠 테이블의 EndSlope 값을 출력합니다.																																													
UINT	CamPointNum	해당 캠 테이블의 캠 데이터 개수 값을 출력합니다.																																													
DWORD[]	CamCurveSel	해당 캠 테이블의 캠 데이터 중 Bit 별로 각 point 의 캠 커브 형식을 출력합니다. (0: Linear, 1: Cubic)																																													

(1) 이 모션 평선 블록은 해당 캠 테이블의 데이터를 출력하는 평선 블록입니다.

- (2) **Enable** 입력이 활성화 되어 있는 동안 해당 캠 테이블의 데이터 값이 연속적으로 출력됩니다.
- (3) **MasterPoint** 와 **SlavePoint** 에는 해당 캠프로파일에서 읽은 데이터 중 '주축위치' 값과 '종축위치' 값을 저장할 변수들의 첫번째 번지를 설정합니다. 예를 들어 읽은 데이터 중 '주축위치' 값을 %ML100~%ML199 디바이스에 저장하고, '종축위치' 값을 %ML300~399 디바이스에 저장하려면 평선블록의 **MasterPoint** 에는 %ML100 을 **SlavePoint** 에는 %ML300 을 설정하여야 합니다.
- (4) **CamCurveSel** 에는 각 Bit 별로 해당 point 의 curve 종류가 출력됩니다. (Point32 까지만 사용 가능)

CamCurveSel[0]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 8	Point 7	Point 6	Point 5	Point 4	Point 3	Point 2	Point 1
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 16	Point 15	Point 14	Point 13	Point 12	Point 11	Point 10	Point 9
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 24	Point 23	Point 22	Point 21	Point 20	Point 19	Point 18	Point 17
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 32	Point 31	Point 30	Point 29	Point 28	Point 27	Point 26	Point 25

LS_ReadCamData2		적용 기종
캠 데이터 읽기 2		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_ReadCamData2 Enable[Enable] Axis[Axis] CamTableID[CamTable ID] MasterPoint[MasterPoint] SlavePoint[SlavePoint] CamCurveSel[CamCurveSel] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] StartSlope[StartSlope] EndSlope[EndSlope] CamPointNum[CamPointNum] end Enable --- LS_ReadCamData2 Axis --- LS_ReadCamData2 CamTableID --- LS_ReadCamData2 MasterPoint --- LS_ReadCamData2 SlavePoint --- LS_ReadCamData2 CamCurveSel --- LS_ReadCamData2 LS_ReadCamData2 --- Done LS_ReadCamData2 --- Busy LS_ReadCamData2 --- Error LS_ReadCamData2 --- ErrorID LS_ReadCamData2 --- StartSlope LS_ReadCamData2 --- EndSlope LS_ReadCamData2 --- CamPointNum </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Enable	입력이 활성화된 동안 해당 캠 데이터를 읽습니다.
UINT	CamTableID	읽을 캠 테이블의 번호를 지정합니다. (1~32)
LREAL	MasterPoint	설정된 디바이스를 선두번지로 하는 영역에 해당 캠 테이블의 MasterPoint 값을 출력합니다.
LREAL	SlavePoint	설정된 디바이스를 선두번지로 하는 영역에 해당 캠 테이블의 SlavePoint 값을 출력합니다.
DWORD	CamCurveSel	설정된 디바이스를 선두번지로 하는 영역에 해당 캠 테이블의 캠 데이터 중 Bit 별로 각 point 의 캠 커브 형식을 출력합니다.
출력		
BOOL	Vaid	평선 블록 출력의 유효 여부를 나타냅니다.
BOOL	Busy	평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	평선 블록 실행 중 발생한 에러번호를 출력합니다.
LREAL	StartSlope	해당 캠 테이블의 StartSlope 값을 출력합니다.
LREAL	EndSlope	해당 캠 테이블의 EndSlope 값을 출력합니다.
UINT	CamPointNum	해당 캠 테이블의 캠 데이터 개수 값을 출력합니다.

- (1) 이 모션 평선 블록은 해당 캠 테이블의 데이터를 출력하는 평선 블록입니다. LS_ReadCamData 평선 블록의 CamCurveSel 기능을 보완하기 위해 추가 되었습니다. CamCurveSel 의 출력을 33 Point 이상 사용하고자 하는 경우 LS_ReadCamData2 를 사용해야 합니다.
- (2) Enable 입력이 활성화 되어 있는 동안 해당 캠 테이블의 데이터 값이 연속적으로 출력됩니다.
- (3) MasterPoint 와 SlavePoint, CamCurveSel 에는 해당 캠프로파일에서 읽은 데이터 중 '주축위치' 값과 '종축위치' 값, '캠 커브 형식'을 저장할 변수들의 첫번째 번지를 설정합니다. 예를 들어 읽은 데이터 중 '주축위치' 값을 %ML100~%ML199 디바이스에 저장하고, '종축위치' 값을 %ML300~399 디바이스에 저장하고, '캠 커브 형식'을 %MD1000~%MD1003 디바이스에 저장하려면 평선블록의 MasterPoint 에는 %ML100 을 SlavePoint 에는 %ML300 을 CamCurveSel 에는 %MD1000 을 설정하여야 합니다.
- (4) CamCurveSel 에 설정한 내부 메모리 선두 번지부터 각 Bit 별로 해당 point 의 curve 종류가 출력됩니다.

CamCurveSel 에 설정된 내부 메모리 번지 DWORD 디바이스	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 8	Point 7	Point 6	Point 5	Point 4	Point 3	Point 2	Point 1
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 16	Point 15	Point 14	Point 13	Point 12	Point 11	Point 10	Point 9
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 24	Point 23	Point 22	Point 21	Point 20	Point 19	Point 18	Point 17
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
CamCurveSel 에 설정된 내부 메모리 번지 +1 DWORD 디바이스	Point 32	Point 31	Point 30	Point 29	Point 28	Point 27	Point 26	Point 25
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 40	Point 39	Point 38	Point 37	Point 36	Point 35	Point 34	Point 33
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 48	Point 47	Point 46	Point 45	Point 44	Point 43	Point 42	Point 41
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 56	Point 55	Point 54	Point 53	Point 52	Point 51	Point 50	Point 49
CamCurveSel 에 설정된 내부 메모리 번지 +2 DWORD 디바이스	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 64	Point 63	Point 62	Point 61	Point 60	Point 59	Point 58	Point 57
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 72	Point 71	Point 70	Point 69	Point 68	Point 67	Point 66	Point 65
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 80	Point 79	Point 78	Point 77	Point 76	Point 75	Point 74	Point 73
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
CamCurveSel 에 설정된 내부 메모리 번지 +3 DWORD 디바이스	Point 88	Point 87	Point 86	Point 85	Point 84	Point 83	Point 82	Point 81
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Point 96	Point 95	Point 94	Point 93	Point 92	Point 91	Point 90	Point 89
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	사용하지 않음				Point 100	Point 99	Point 98	Point 97
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	사용하지 않음							
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	
사용하지 않음								
Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
사용하지 않음								

LS_WriteCamData		적용 기종
캠 데이터 쓰기		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_WriteCamData Execute[Execute] Axis[Axis] CamTableID[CamTable ID] StartSlope[StartSlope] EndSlope[EndSlope] CamPointNum[CamPointNum] CamCurveSel[CamCurveSel] MasterPoint[MasterPoint] SlavePoint[SlavePoint] ExecutionMode[ExecutionMode] Done[Done] AxisOut[Axis] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done Axis --- AxisOut CamTableID --- Busy StartSlope --- Error EndSlope --- ErrorID CamPointNum --- Done CamCurveSel --- Done MasterPoint --- Done SlavePoint --- Done ExecutionMode --- Done </pre> </div>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	입력의 상승 Edge 에 캠 데이터 쓰기 지령을 내립니다.
UINT	CamTableID	쓸 캠 테이블의 ID 를 지정합니다. (1~32)
LREAL	StartSlope	쓸 캠 데이터의 Start Slope 값을 설정합니다.
LREAL	EndSlope	쓸 캠 데이터의 End Slope 값을 설정합니다.
UINT	CamPointNum	쓸 캠 데이터의 개수를 설정합니다.
DWORD[]	CamCurveSel	쓸 캠 데이터 중 Bit 별로 각 point 의 캠 커브 형식을 지정합니다. (0: Linear, 1: Cubic)
LREAL	MasterPoint	쓸 캠 데이터 중 Master Point 값이 저장된 디바이스의 선두 번지를 설정합니다.
LREAL	SlavePoint	쓸 캠 데이터 중 Slave Point 값이 저장된 디바이스의 선두 번지를 설정합니다.
UINT	ExecutionMode	캠 데이터가 쓰여질 시점을 지정합니다. 0 - 즉시 적용, 1 - Buffermode 의 'Buffered'와 동일한 시점에 적용
출력		
BOOL	Done	캠데이터를 성공적으로 썼음을 나타냅니다.
BOOL	Busy	평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 해당 캠 Table 의 데이터 값을 쓰는 명령이며, CamTableID 입력으로 지정된 캠 테이블의 데이터 중 StartSlope, EndSlope 에 설정된 값 및 CamPointNum 에 설정된 개수 만큼 MasterPoint 와 SlavePoint 에 설정된 디바이스의 값을 MasterPoint 와 SlavePoint 값으로 씁니다.
- (2) CamTableID 입력은 1~32 값을 설정할 수 있습니다. 설정값 이외의 값을 설정했을 경우 “에러 16#000B” 가 발생합니다.
- (3) CamPointNum 에는 1~100 범위의 값을 입력할 수 있으며, 범위 이외의 값을 설정했을 경우 “에러 16#000B” 가 발생합니다.
- (4) CamCurveSel[4]에는 각 Bit 별로 해당 point 의 curve 종류를 설정할 수 있다.

CamCurveSel[0]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 8	Point 7	Point 6	Point 5	Point 4	Point 3	Point 2	Point 1
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 16	Point 15	Point 14	Point 13	Point 12	Point 11	Point 10	Point 9
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 24	Point 23	Point 22	Point 21	Point 20	Point 19	Point 18	Point 17
	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
CamCurveSel[1]	Point 32	Point 31	Point 30	Point 29	Point 28	Point 27	Point 26	Point 25
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 40	Point 39	Point 38	Point 37	Point 36	Point 35	Point 34	Point 33
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 48	Point 47	Point 46	Point 45	Point 44	Point 43	Point 42	Point 41
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 56	Point 55	Point 54	Point 53	Point 52	Point 51	Point 50	Point 49
Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
CamCurveSel[2]	Point 64	Point 63	Point 62	Point 61	Point 60	Point 59	Point 58	Point 57
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Point 72	Point 71	Point 70	Point 69	Point 68	Point 67	Point 66	Point 65
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Point 80	Point 79	Point 78	Point 77	Point 76	Point 75	Point 74	Point 73
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Point 88	Point 87	Point 86	Point 85	Point 84	Point 83	Point 82	Point 81
Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
CamCurveSel[3]	Point 96	Point 95	Point 94	Point 93	Point 92	Point 91	Point 90	Point 89
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	사용하지 않음				Point 100	Point 99	Point 98	Point 97
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	사용하지 않음							
	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	사용하지 않음							
Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	
사용하지 않음								

- (5) ExecutionMode 입력은 설정 시점을 지정합니다. 0 일 때는 명령 실행 즉시 설정되며, 1 일 경우에는 순차운전 설정의 'Buffered' 와 동일한 시점에 설정됩니다. 설정 불가능한 값을 설정했을 경우 “에러 16#000B” 가 발생합니다.
 - 0(mcImmediatly) : 평선 블록 실행 즉시(Execute 입력의 상승 Edge) 파라미터 값을 변경합니다. 해당 축이 구동 중이라면 동작에 영향이 있을 수 있습니다.
 - 1(mcQueued) : Buffermode 의 'Buffered' 와 같은 시점에 변경됩니다. (6.1.4. 명령어 버퍼링 : BufferMode 참조)
- (6) ExecutionMode 를 1 로 하여 명령이 실행된 후 실행 대기 중일 때 BufferMode 0(Aborting)으로 다른 명령이 실행되어 중단된 경우는 Error 는 1, ErrorId 는 0 으로 출력됩니다.
- (7) MasterPoint 와 SlavePoint 에는 해당 캠 테이블에 쓸 데이터 중 Main Ax. Pos 값과 Sub Ax. Pos 값의 변수들의 첫번째 번지를 설정합니다. 예를 들어 쓸 데이터 중 Main Ax. Pos 값이 MainAxPos[100]이라는 어레이 변수에 설정되어 있고, Sub Ax. Pos 값이 SubAxPos[100]이라는 어레이 변수에 설정이 되어 있다면 Function Block 의 MasterPoint 에는 MainAxPos[0]을 SlavePoint 에는 SubAxPos[0]를 설정하여야 합니다.

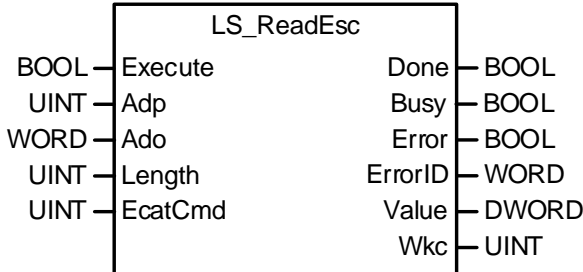
LS_SaveServoParameter		적용 기종
서보 파라미터 저장		XGF-M32E
평선 블록 형태		
<pre> graph LR subgraph LS_SaveServoParameter Execute[Execute] Axis[Axis] ExecutionMode[ExecutionMode] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] end Execute --- Done Axis --- Axis ExecutionMode --- Busy ExecutionMode --- Error ExecutionMode --- ErrorID </pre>		
입력-출력		
UINT	Axis	지령을 내릴 축 지정 (1 ~ 32 : 1 축 ~ 32 축)
입력		
BOOL	Execute	입력의 상승 Edge 에 서보 파라미터 저장 지령을 내립니다.
UINT	ExecutionMode	서보 파라미터가 저장 될 시점을 지정합니다. 0 - 서보 저장 즉시 적용, 1 - Buffermode 의 'Buffered'와 동일한 시점에 적용
출력		
BOOL	Done	서보 파라미터가 성공적으로 저장되었음을 나타냅니다.
BOOL	Busy	평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 지정된 축의 서보 드라이브의 파라미터를 서보 드라이브의 EEPROM 에 저장하는 명령입니다.
- (2) ExecutionMode 입력은 설정 시점을 지정합니다. 0 일 때는 명령 실행 즉시 설정되며, 1 일 경우에는 순차운전 설정의 'Buffered' 와 동일한 시점에 설정됩니다. 설정 불가능한 값을 설정했을 경우 “에러 16#000B” 가 발생합니다.

0(mclImmediately) : 평선 블록 실행 즉시(Execute 입력의 상승 Edge) 파라미터 값을 변경합니다. 해당 축이 구동 중이라면 동작에 영향이 있을 수 있습니다.

1(mcQueued) : Buffermode 의 'Buffered' 와 같은 시점에 변경됩니다. (6.1.4. 명령어 버퍼링 : BufferMode 참조)

- (3) ExecutionMode 를 1 로 하여 명령이 실행된 후 실행 대기 중일 때 BufferMode 0(Aborting)으로 다른 명령이 실행되어 중단된 경우는 Error 는 1, ErrorId 는 0 으로 출력됩니다.

LS_ReadEsc		적용 기종
ESC 읽기		XGF-M32E
평선 블록 형태		
 <pre> graph LR subgraph LS_ReadEsc Execute[Execute] Adp[Adp] Ado[Ado] Length[Length] EcatCmd[EcatCmd] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] Value[Value] Wkc[Wkc] end Execute --- Done Adp --- Busy Ado --- Error Length --- ErrorID EcatCmd --- Value Done --- Done Busy --- Busy Error --- Error ErrorID --- ErrorID Value --- Value Wkc --- Wkc </pre>		
입력		
BOOL	Execute	상승 Edge 에서 슬레이브 기기에 ESC 읽기 명령을 내립니다.
UINT	Adp(Address position)	EcatCmd 명령에 따라 슬레이브 기기 주소를 지정합니다.
WORD	Ado(Address offset)	슬레이브 기기의 ESC 주소를 지정합니다.
UINT	Length	읽을 데이터의 크기를 지정합니다. (1 ~ 4 Byte)
UINT	EcatCmd	EtherCAT 명령을 지정합니다. (1: APRD, 4: FPRD, 7: BRD)
출력		
BOOL	Done	ESC 읽기를 정상적으로 완료하였는지 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
UDINT	Value	슬레이브 기기의 ESC 읽기 값을 출력합니다.
UINT	Wkc(Working Counter)	명령 실행 후 Working Counter 값을 출력합니다. 지정된 슬레이브 기기에서 데이터가 성공적으로 읽어진다면 1 씩 증가 됩니다.

- (1) 이 모션 평선 블록은 지정된 슬레이브 기기의 ESC (EtherCAT Slave Controller)로부터 설정된 주소의 데이터를 읽을 수 있는 평선 블록입니다.
- (2) Value 및 Wkc(Working Counter) 출력은 모션 평선 블록이 실행되면 0 으로 소거되었다가 실행이 완료되면 (Done 출력 온(On)) 읽은 데이터 값이 Value 에 출력되고, Working Counter 값은 Wkc 에 출력됩니다.
- (3) Adp(Address position) 입력은 EtherCAT 슬레이브 기기의 주소를 지정하는 것으로, EcatCmd 설정에 따라 지정할 수 있는 값은 다음과 같습니다. EcatCmd 설정이 7(BRD)인 경우 Adp 입력 값은 무시 됩니다. Adp 입력에 범위 이외의 값을 설정했을 경우 “에러 0x0F60”가 발생합니다.

EcatCmd	Adp 범위
1 (APRD)	0x0000: 첫 번째 연결된 슬레이브(0) 0xFFFF: 두 번째 연결된 슬레이브(65535) 0xFFFE: 세 번째 연결된 슬레이브(65534) : 0xFFDD: 36 번째 번째 연결된 슬레이브(65501)
4 (FPRD)	1001 ~ 1036: 1번 슬레이브 ~ 36번 슬레이브
7 (BRD)	-

- (4) Length 에는 1~4 범위의 값을 입력할 수 있으며, 이는 1~4 Byte 를 의미합니다. 범위 이외의 값을 설정했을 경우 “에러 0x0F61”가 발생합니다.
- (5) EcatCmd 에는 ESC (EtherCAT Slave Controller)를 읽을 때 사용할 명령의 종류를 지정합니다. 다음의 3 가지 읽기 명령을 사용할 수 있습니다. EcatCmd 에 범위 이외의 값을 설정했을 경우 “에러 0xF62”가 발생합니다.
- 1) 1- 자동 증가 물리적 읽기 (APRD, Auto Increment Physical Read)

마스터에 의해 통신이 정상적으로 연결되기 전 물리적으로 연결된 순서에 의해 슬레이브 기기의 데이터를 읽을 때 사용하는 명령입니다. 0 값을 갖는 Adp 를 받은 슬레이브 기기는 Ado(Address offset) 영역에서 Length 로 지정된 크기의 데이터 읽기 동작을 실행합니다. 각 슬레이브 기기는 EtherCAT 프레임을 수신하면 Adp 를 증가합니다.. 예를 들어, EcatCmd 는 1 이고, Adp 는 0xFFFF 로 설정하여 ESC 읽기 펄스블록을 실행하면 첫 번째 슬레이브 기기에서 EtherCAT 프레임을 수신하였을 때 Adp 가 0 이 아니므로 읽기 동작을 수행하지 않고, Adp 만 1 증가합니다. 두 번째 슬레이브 기기에서 EtherCAT 프레임을 수신하였을 때 첫 번째 슬레이브 기기에서 Adp 값이 1 증가하여 0 이 되었으므로 읽기 동작을 실행합니다. 슬레이브 기기의 연결 순서에 따른 Adp 설정 값은 다음과 같습니다.

슬레이브 기기	설정 값
첫 번째 연결된 슬레이브	0
두 번째 연결된 슬레이브	0xFFFF
:	:
36 번째 연결된 슬레이브	0xFFDD

- 2) 4- 배열된 어드레스 물리적 읽기 (FPRD, Configured Address Physical Read)

마스터에 의해 통신이 정상적으로 연결된 후 슬레이브 기기의 Station Address 를 지정하여 데이터를 읽을 때 사용하는 명령입니다. EtherCAT 마스터에 의해 설정된 슬레이브 기기의 Station Address 와 전송된 Adp 가 일치하는 경우 슬레이브 기기는 Ado 영역에서 Length 로 지정된 크기의 데이터 읽기 동작을 실행합니다.

마스터에 의해 설정되는 슬레이브 기기의 Station Address 는 다음과 같습니다.

슬레이브 기기	설정 값
1번 슬레이브	1001
2번 슬레이브	1002
:	:
36번 슬레이브	1036

- 3) 7- 브로드캐스트 읽기 (BRD, Broadcast Read)

연결된 모든 슬레이브 기기는 Ado 영역에서 Length 로 지정된 크기의 데이터를 읽어 Bitwise-OR 하여(각각의 비트를 OR 연산) 결과를 저장합니다. Adp 에 지정된 주소 값은 무시되고, Wkc 는 정상적으로 읽기 수행된 모든 슬레이브에 의해 1 씩 증가됩니다.

- (6) **Wkc**는 **Working Counter**를 의미하는 것으로, 지정된 슬레이브 기기에서 데이터가 성공적으로 읽어진다면 1 증가 됩니다. **EcatCmd**가 7(BRD)인 경우에는 정상적으로 읽기 수행된 모든 슬레이브에 의해 1씩 증가 됩니다.
- (7) **ESC** 읽기 명령 실행 후 지정된 슬레이브 기기로부터 정상적으로 데이터 읽기가 실행되면 **Done** 출력이 **On**됩니다.
- (8) 상세 내용은 제품 사용설명서 8.3.4장 **EtherCAT** 통신 진단 기능을 참고하시기 바랍니다.
- (9) 자세한 레지스터 및 **Process RAM**의 설명은 **EtherCAT Slave Controller(ESC)** 데이터시트를 참조하시기 바랍니다.

LS_WriteEsc		적용 기종
ESC 쓰기		XGF-M32E
평선 블록 형태		
<pre> graph LR subgraph LS_WriteEsc Execute[Execute] Adp[Adp] Ado[Ado] Length[Length] EcatCmd[EcatCmd] Value[Value] Done[Done] Busy[Busy] Error[Error] ErrorID[ErrorID] Wkc[Wkc] end Execute --- Done Adp --- Busy Ado --- Error Length --- ErrorID EcatCmd --- Wkc Value --- Wkc </pre>		
입력		
BOOL	Execute	상승 Edge 에서 슬레이브 기기에 ESC 쓰기 명령을 내립니다.
UINT	Adp	EcatCmd 명령에 따라 슬레이브 기기 주소를 지정합니다.
WORD	Ado	슬레이브 기기의 ESC 주소를 지정합니다.
UINT	Length	쓸 데이터의 크기를 지정합니다. (1 ~ 4 Byte)
UINT	EcatCmd	EtherCAT 명령을 지정합니다. (2: APWR, 5: FPWR, 8: BWR)
UDINT	Value	슬레이브 기기의 ESC 에 쓸 데이터 값을 지정합니다.
출력		
BOOL	Done	ESC 쓰기를 정상적으로 완료하였는지 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
UINT	Wkc	명령 실행 후 Working Counter 값을 출력합니다.

- (1) 이 모션 평선 블록은 Adp 로 지정된 슬레이브 기기의 ESC (EtherCAT Slave Controller)에 Ado 로 설정된 주소로 데이터 쓰기를 실행합니다.
- (2) Wkc 출력은 모션 평선 블록이 실행되면 0 으로 소거되었다가 실행이 완료되면 (Done 출력 온(On)) Working Counter 값이 출력됩니다. Wkc 는 EcatCmd 및 Adp 에 의해 지정된 각 슬레이브 기기에 의해 1 씩 증가 됩니다.
- (3) Adp 입력은 EtherCAT 슬레이브 기기의 주소를 지정하는 것으로, EcatCmd 설정에 따라 지정할 수 있는 값은 다음과 같습니다. EcatCmd 설정이 8(BWR)인 경우 Adp 입력 값은 무시 됩니다. Adp 입력에 범위 이외의 값을 설정했을 경우 “에러 0x0F70”가 발생합니다.

EcatCmd	Adp 범위
2 (APWR)	0x0000: 첫 번째 연결된 슬레이브 0xFFFF: 두 번째 연결된 슬레이브 0xFFFE: 세 번째 연결된 슬레이브 : 0xFFDD: 36 번째 번째 연결된 슬레이브
5 (FPWR)	1001 ~ 1036: 1번 슬레이브 ~ 36번 슬레이브
8 (BWR)	-

(4) Length 에는 1~4 범위의 값을 입력할 수 있으며, 이는 1~4 Byte 를 의미합니다. 범위 이외의 값을 설정했을 경우 “에러 0x0F71”가 발생합니다.

(5) EcatCmd 에는 ESC (EtherCAT Slave Controller)를 읽을 때 사용할 명령의 종류를 지정합니다. 다음의 3 가지 쓰기 명령을 사용할 수 있습니다. EcatCmd 에 범위 이외의 값을 설정했을 경우 “에러 0x0F72”가 발생합니다.

1) 2- 자동 증가 물리적 쓰기 (APWR, Auto Increment Physical Write)

마스터에 의해 통신이 정상적으로 연결되기 전 물리적으로 연결된 순서에 의해 슬레이브 기기에 데이터를 쓸 때 사용하는 명령입니다. 0 값을 갖는 Adp 를 받은 슬레이브 기기는 Ado 영역에 Length 로 지정된 크기의 데이터 쓰기 동작을 실행합니다. 각 슬레이브 기기는 EtherCAT 프레임 수신하면 Adp 를 증가합니다.. 예를 들어, EcatCmd 는 2 이고, Adp 는 0xFFFF 로 설정하여 ESC 쓰기 펄스블록을 실행하면 첫 번째 슬레이브 기기에서 EtherCAT 프레임 수신하였을 때 Adp 가 0 이 아니므로 쓰기 동작을 수행하지 않고, Adp 만 1 증가합니다. 두 번째 슬레이브 기기에서 EtherCAT 프레임 수신하였을 때 첫 번째 슬레이브 기기에서 Adp 값이 1 증가하여 0 이 되었으므로 쓰기 동작을 실행합니다. 슬레이브 기기의 연결 순서에 따른 Adp 설정 값은 다음과 같습니다.

슬레이브 기기	설정 값
첫 번째 연결된 슬레이브	0
두 번째 연결된 슬레이브	0xFFFF
:	:
36 번째 연결된 슬레이브	0xFFDD

2) 5- 배열된 어드레스 물리적 쓰기 (FPWR, Configured Address Physical Write)

마스터에 의해 통신이 정상적으로 연결된 후 슬레이브 기기의 Station Address 를 지정하여 데이터를 쓸 때 사용하는 명령입니다. EtherCAT 마스터에 의해 설정된 슬레이브 기기의 Station Address 와 전송된 Adp 가 일치하는 경우 슬레이브 기기는 Ado 영역에 Length 로 지정된 크기의 데이터 쓰기 동작을 실행합니다. 마스터에 의해 설정되는 슬레이브 기기의 Station Address 는 다음과 같습니다.

슬레이브 기기	설정 값
1번 슬레이브	1001
2번 슬레이브	1002
:	:
36번 슬레이브	1036

3) 8- 브로드캐스트 쓰기 (BWR, Broadcast Write)

연결된 모든 슬레이브 기기의 Ado 영역에 Length 로 지정된 크기의 데이터 쓰기 동작을 실행합니다. Adp 에 지정된 주소 값은 무시되고, Wkc 는 정상적으로 쓰기 수행된 모든 슬레이브에 의해 1씩 증가됩니다.

(6) Wkc는 Working Counter를 의미하는 것으로, 지정된 슬레이브 기기의 Ado 영역에 데이터가 성공적으로 쓰인다면 1 증가 됩니다. EcatCmd가 8(BWR)인 경우에는 정상적으로 쓰기 수행된 모든 슬레이브에 의해 1씩 증가 됩니다.

- (7) ESC 쓰기 명령 실행 후 지정된 슬레이브 기기에 정상적으로 데이터 쓰기가 실행되면 Done 출력이 On됩니다.
- (8) 상세 내용은 제품 사용설명서 8.3.4장 EtherCAT 통신 진단 기능을 참고하시기 바랍니다.
- (9) 자세한 레지스터 및 Process RAM의 설명은 EtherCAT Slave Controller(ESC) 데이터시트를 참조하시기 바랍니다.

LS_CamSkip		적용 기종
캠 스킵		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> </div>		
입력-출력		
UINT	Slave	중축을 설정합니다. (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 캠 스킵 명령을 내립니다.
UINT	SkipCount	스킵할 캠 주기 횟수를 지정합니다.
출력		
BOOL	Done	캠 스킵 동작이 정상적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 해당 축이 캠 스킵 동작 중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
LREAL	CoveredSkipcount	스킵한 캠 주기 횟수가 출력됩니다.

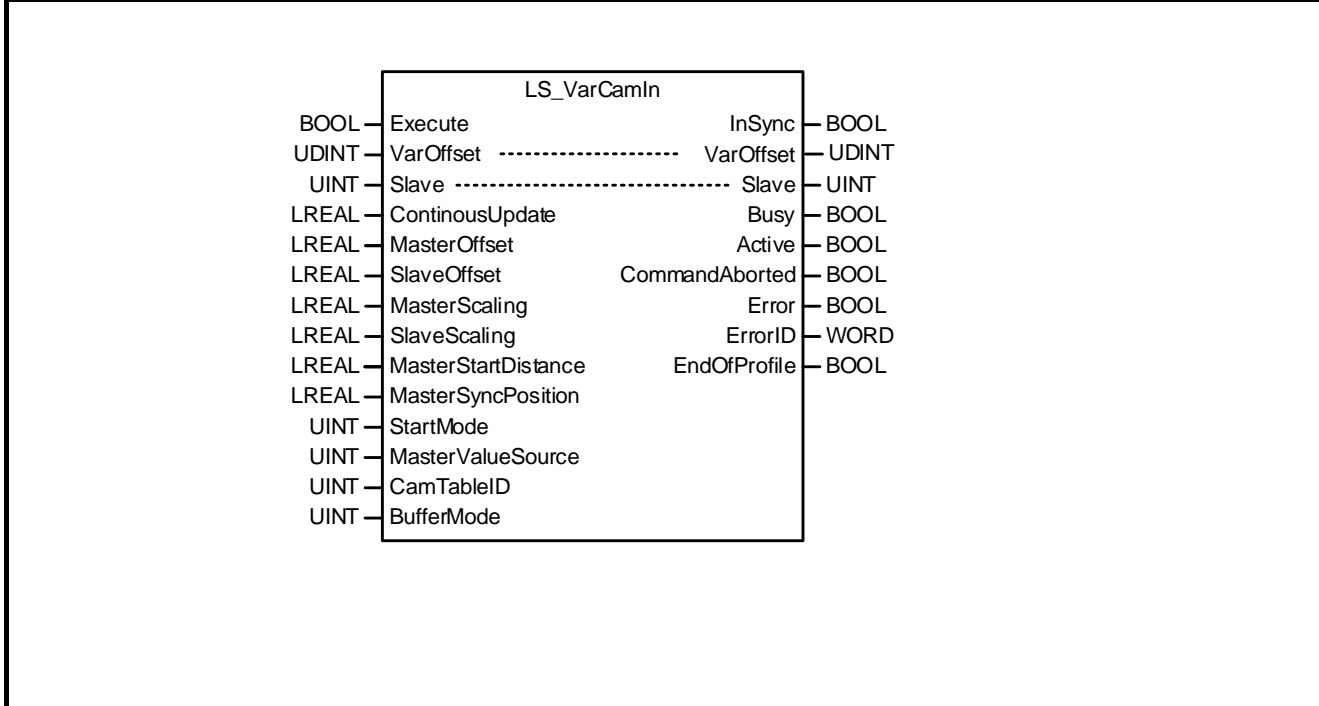
- (1) 이 모션 평선 블록은 현재 캠 운전 중인 축에 지정된 주기만큼 캠 운전을 생략하는 캠 스킵 지령을 내리는 평선 블록입니다.
- (2) SkipCount 입력에는 스킵할 캠 주기 횟수를 지정합니다. 0 이 입력될 경우 SkipCount 에러(에러 0x111E)가 출력됩니다.
- (3) 캠 운전중인 중축에 캠 스킵 명령을 내리면 현재 수행중인 캠 주기가 끝나고 스킵 동작이 시작됩니다. 캠 스킵 중인 중축은 캠 테이블의 끝 위치에서 정지상태로 대기합니다.

- (4) **CoveredSkipCount** 출력은 스킵한 캠 주기 횟수가 출력됩니다. 캠 주기를 스킵할 때 마다 증가하며, 평선 블록의 동작이 모두 완료 된 후 **Done** 출력이 **Off** 되는 시점에 0 이 됩니다. (6.1.3 기본 입출력 변수 참고)
- (5) 캠 스킵 명령 실행 후 스킵할 캠 주기 횟수만큼 캠 운전이 스킵되면 **Done** 출력이 **On**됩니다.

LS_VarCamIn	적용 기종
--------------------	-------

변수 캠 운전	XGF-M32E
---------	----------

평선 블록 형태



입력-출력

UDINT	VarOffset	주축으로 사용할 변수가 위치한 M 디바이스의 오프셋값을 설정합니다.
-------	-----------	---------------------------------------

UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축)
------	-------	-----------------------------------

입력

BOOL	Execute	상승 Edge 에서 해당 축에 캠 운전 명령을 내립니다.
------	---------	---------------------------------

BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (6.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)
------	------------------	--

LREAL	MasterOffset	주축의 오프셋 값을 설정합니다.
-------	--------------	-------------------

LREAL	SlaveOffset	종축 캠 테이블의 오프셋 값을 설정합니다.
-------	-------------	-------------------------

LREAL	MasterScaling	주축의 배율을 지정합니다.
-------	---------------	----------------

LREAL	SlaveScaling	종축 캠 테이블의 배율을 지정합니다.
-------	--------------	----------------------

LREAL	MasterStartDistance	주축의 캠 운전이 시작될 주축의 위치를 지정합니다.
-------	---------------------	------------------------------

LREAL	MasterSyncPosition	캠 운전이 시작할 때, 캠 테이블에서의 시작점을 지정합니다.
-------	--------------------	-----------------------------------

UINT	StartMode	캠 운전 모드를 설정합니다. 0: 캠 테이블이 절대값으로 적용됨 (mcAbsolute)
------	-----------	---

		1: 캠 테이블이 명령 시작 위치를 기준으로 상대값으로 적용됨 (mcRelative)
UINT	MasterValueSource	캠 운전할 주축의 소스를 선택합니다. 0: 주축의 명령 위치에 동기화합니다. 1: 주축의 현재 위치에 동기화합니다.
UINT	CamTableID	운전할 캠 테이블을 지정합니다.
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)
출력		
BOOL	InSync	캠 운전이 동기 운전 중 임을 나타냅니다. (종축이 캠 테이블을 추종(Sync) 중 임을 나타냅니다.)
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
BOOL	EndOfProfile	현재 운전중인 캠의 끝에 도달했음을 나타냅니다

- (1) 이 모션 평선 블록은 옵셋으로 지정한 변수값을 주축으로 설정하여 주축에 따라 종축 캠 운전을 시키는 모션 평선 블록입니다.
- (2) 주축으로 지정하는 변수값은 LREAL 타입을 사용해야 합니다. 예)주축값으로 %ML100 으로 메모리에 할당하는 변수를 지정하는 경우 %ML100 은 LREAL 타입으로 사용해야 하고 변수를 지정하는 옵셋값은 UDINT 타입으로 VarOffset 에 100 을 입력 합니다.
- (3) 나머지 설정과 기능은 MC_CamIn 평선 블록과 동일합니다.
- (4) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목	모듈 O/S	XG-PM
제품명		
XGF-M32E	V1.30	V3.1

LS_VarGear In		적용 기종
변수 기어 운전		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph TD subgraph LS_VarGearIn Execute[Execute] VarOffset[VarOffset] Slave[Slave] ContinuousUpdate[ContinuousUpdate] RatioNumerator[RatioNumerator] RatioDenominator[RatioDenominator] MasterValueSource[MasterValueSource] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] BufferMode[BufferMode] InGear[InGear] VarOffset2[VarOffset] Slave2[Slave] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- InGear VarOffset --- VarOffset2 Slave --- Slave2 ContinuousUpdate --- Busy RatioNumerator --- Active RatioDenominator --- CommandAborted MasterValueSource --- Error Acceleration --- ErrorID Deceleration --- ErrorID Jerk --- ErrorID BufferMode --- ErrorID </pre> </div>		
입력-출력		
UDINT	VarOffset	주축으로 사용할 변수가 위치한 M 디바이스의 오프셋값을 설정합니다.
UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 기어 운전 명령을 내립니다.
BOOL	ContinuousUpdate	입력 값의 업데이트 설정을 지정합니다. (6.1.5. 모션 평선 블록 실행 중 파라미터 변경 참조)
INT	RatioNumerator	기어비의 분자 값을 지정합니다(-32768 ~ 32767)
UINT	RatioDenominator	기어비의 분모 값을 지정합니다(0 ~ 65535)
UINT	MasterValueSource	기어 운전할 주축의 데이터를 선택합니다. 0: 주축의 명령 위치로 기어 운전을 합니다. 1: 주축의 현재 위치에 기어 운전을 합니다.
LREAL	Acceleration	기어 운전 시작시의 가속도를 지정합니다. [u/s ²]
LREAL	Deceleration	기어 운전 시작시의 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	가/감속도 변화율을 지정합니다 [u/s ³]
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)

출력		
BOOL	InGear	기어비를 적용하여 기어 운전 중임을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 옵셋으로 지정한 변수값을 주축으로 설정하여 주축과 종축을 기어 운전(속도 동기) 시키는 평선블록입니다
- (2) 주축으로 지정하는 변수값은 LREAL 타입을 사용해야 합니다. 예)주축값으로 %ML100 으로 메모리에 할당하는 변수를 지정하는 경우 %ML100 은 LREAL 타입으로 사용해야 하고 변수를 지정하는 옵셋값은 UDINT 타입으로 VarOffset 에 100 을 입력 합니다.
- (3) 나머지 설정과 기능은 MC_GearIn 평선 블록과 동일합니다.
- (4) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

LS_VarGear InPos		적용 기종																																																								
변수 위치지정 기어운전		XGF-M32E																																																								
평선 블록 형태																																																										
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">LS_VarGearInPos</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">InSync</td> <td>BOOL</td> </tr> <tr> <td>UDINT</td> <td>VarOffset</td> <td>VarOffset</td> <td>UDINT</td> </tr> <tr> <td>UINT</td> <td>Slave</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td>INT</td> <td>RatioNumerator</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>RatioDenominator</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>MasterValueSource</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterSyncPosition</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>SlaveSyncPosition</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>UINT</td> <td>SyncMode</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>MasterStartDistance</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	InSync	BOOL	UDINT	VarOffset	VarOffset	UDINT	UINT	Slave	Slave	UINT	INT	RatioNumerator	Busy	BOOL	UINT	RatioDenominator	Active	BOOL	UINT	MasterValueSource	CommandAborted	BOOL	LREAL	MasterSyncPosition	Error	BOOL	LREAL	SlaveSyncPosition	ErrorID	WORD	UINT	SyncMode			LREAL	MasterStartDistance			LREAL	Acceleration			LREAL	Deceleration			LREAL	Jerk			UINT	BufferMode		
BOOL	Execute	InSync	BOOL																																																							
UDINT	VarOffset	VarOffset	UDINT																																																							
UINT	Slave	Slave	UINT																																																							
INT	RatioNumerator	Busy	BOOL																																																							
UINT	RatioDenominator	Active	BOOL																																																							
UINT	MasterValueSource	CommandAborted	BOOL																																																							
LREAL	MasterSyncPosition	Error	BOOL																																																							
LREAL	SlaveSyncPosition	ErrorID	WORD																																																							
UINT	SyncMode																																																									
LREAL	MasterStartDistance																																																									
LREAL	Acceleration																																																									
LREAL	Deceleration																																																									
LREAL	Jerk																																																									
UINT	BufferMode																																																									
입력-출력																																																										
UDINT	VarOffset	주축으로 사용할 변수가 위치한 M 디바이스의 옵셋값을 설정합니다.																																																								
UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축)																																																								
입력																																																										
BOOL	Execute	상승 Edge 에서 해당 축에 기어 운전 명령을 내립니다.																																																								
INT	RatioNumerator	기어비의 분자 값을 지정합니다(-32768~32767)																																																								
UINT	RatioDenominator	기어비의 분모 값을 지정합니다(0~65535)																																																								
UINT	MasterValueSource	동기화될 주축 값의 기준을 선택합니다. 0(mcSetValue): 주축의 목표위치에 동기화합니다. 1(mcActualValue): 주축의 현재위치에 동기화합니다.																																																								
LREAL	MasterSyncPosition	기어운전이 시작될 주축의 위치를 지정합니다.																																																								
LREAL	SlaveSyncPosition	기어운전이 시작될 종축의 위치를 지정합니다.																																																								
LREAL	SyncMode	미사용																																																								
LREAL	MasterStartDistance	동기화를 시작할 주축의 거리를 지정합니다.																																																								
LREAL	Velocity	동기화 시작시의 종축의 최대 속도를 지정합니다. [u/s]																																																								
LREAL	Acceleration	동기화 시작시의 종축의 최대 가속도를 지정합니다. [u/s ²]																																																								

LREAL	Deceleration	동기화 시작시의 종축의 최대 감속도를 지정합니다. [u/s²]
LREAL	Jerk	미사용
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)
출력		
BOOL	InSync	지정한 기어비가 적용되어 기어운전이 정상 수행중임을 나타냅니다.
BOOL	StartSync	동기화 시작중임을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 옵셋으로 지정한 변수값을 주축으로 설정하여 특정 위치에서 설정한 기어 비에 따라 주축과 종축을 속도 동기 시키는 평선블록입니다
- (2) 주축으로 지정하는 변수값은 LREL 타입을 사용해야 합니다. 예)주축값으로 %ML100 으로 메모리에 할당하는 변수를 지정하는 경우 %ML100 은 LREAL 타입으로 사용해야 하고 변수를 지정하는 옵셋값은 UDINT 타입으로 VarOffset 에 100 을 입력 합니다.
- (3) 나머지 설정과 기능은 MC_GearInPos 평선 블록과 동일합니다.
- (4) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

LS_ReadCamTableSlavePos		적용 기종																																
캠 테이블 슬레이브 위치 읽기		XGF-M32E																																
평선 블록 형태																																		
<div style="text-align: center; border: 1px solid black; padding: 10px;"> <p>LS_ReadCamTableSlavePos</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">Done</td> <td style="width: 30%;">BOOL</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td>Axis</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>CamTableID</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterPos</td> <td>SlavePos</td> <td>LREAL</td> </tr> <tr> <td></td> <td></td> <td>SlaveVel</td> <td>LREAL</td> </tr> <tr> <td></td> <td></td> <td>SlaveAccel</td> <td>LREAL</td> </tr> <tr> <td></td> <td></td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td>ErrorID</td> <td>WORD</td> </tr> </table> </div>			BOOL	Execute	Done	BOOL	UINT	Axis	Axis	UINT	UINT	CamTableID	Busy	BOOL	LREAL	MasterPos	SlavePos	LREAL			SlaveVel	LREAL			SlaveAccel	LREAL			Error	BOOL			ErrorID	WORD
BOOL	Execute	Done	BOOL																															
UINT	Axis	Axis	UINT																															
UINT	CamTableID	Busy	BOOL																															
LREAL	MasterPos	SlavePos	LREAL																															
		SlaveVel	LREAL																															
		SlaveAccel	LREAL																															
		Error	BOOL																															
		ErrorID	WORD																															
입력-출력																																		
UINT	Axis	지령 축을 설정합니다. (1~32: 실축, 37~40: 가상축)																																
입력																																		
BOOL	Execute	상승 Edge 에서 해당 축에 캠테이블 슬레이브 위치 읽기 명령을 내립니다.																																
UINT	CamTableID	읽을 캠 테이블의 번호를 지정합니다. (1~32)																																
LREAL	MasterPos	캠 테이블 상의 주축위 위치를 지정합니다.																																
출력																																		
BOOL	Done	캠테이블 슬레이브 위치 읽기 동작이 성공적으로 완료되었음을 나타냅니다.																																
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.																																
LREAL	SlavePos	슬레이브의 위치를 출력 합니다.																																
LREAL	SlaveVel	슬레이브의 속도를 출력 합니다. [u/s]																																
LREAL	SlaveAccel	슬레이브의 가속도를 출력 합니다. [u/s ²]																																
BOOL	Error	에러 발생 여부를 나타냅니다.																																
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.																																

- (1) 이 모션 평선 블록은 지정한 캠 테이블에서 주축의 위치에 따른 종축의 위치를 출력하는 기능입니다.
- (2) 캠 테이블에서 읽고자 하는 주축의 위치값을 MasterPos 값으로 설정합니다. 명령축에 적용된 Offset/기어비/위상보정운전 등은 SlavePos 출력에 반영되지 않습니다.
- (3) 캠테이블 슬레이브 위치읽기 동작이 완료되면 Done 출력이 On됩니다.

(4) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

LS_SyncMoveVelocity		적용 기종
속도 제어 운전(CSV 모드)		XGF-M32E
평선 블록 형태		
입력-출력		
UINT	Axis	지령을 내릴 축을 설정합니다. (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 CSV mode 를 이용한 속도 제어 운전을 합니다.
LREAL	Velocity	운전 속도를 설정합니다.
BOOL	CmdPosMode	0: 현재 위치를 명령 위치로 적용합니다.
UINT	BufferMode	모션 평선 블록의 순차 운전 설정을 지정합니다. (0: Aborting, 1: Buffered) (6.1.4 BufferMode 입력 참조)
출력		
BOOL	Done	평선 블록 완료 상태를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 축을 제어하는 중인지 여부를 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 설정된 축에 CiA402 profile 의 CSV(Cyclic Synchronous Velocity) 모드를 이용하여 속도 제어를 하도록 하는 모션 평선 블록입니다.
- (2) 지정 속도 운전을 정지하기 위해서는, 정지 명령을 내리거나 다른 모션 평선 블록을 수행함으로써 가능합니다.
- (3) Velocity 입력에는 운전할 속도를 지정합니다. 운전 속도 값의 부호가 양(+ 또는 부호 없음)일 때는 정방향, 음(-)일 때는 역방향으로 이동하게 됩니다.
- (4) CmdPosMode 에는 명령 시 현재 위치 갱신 방법을 설정합니다. 초기값인 0 만 사용 가능하며 피드백 현재 위치를

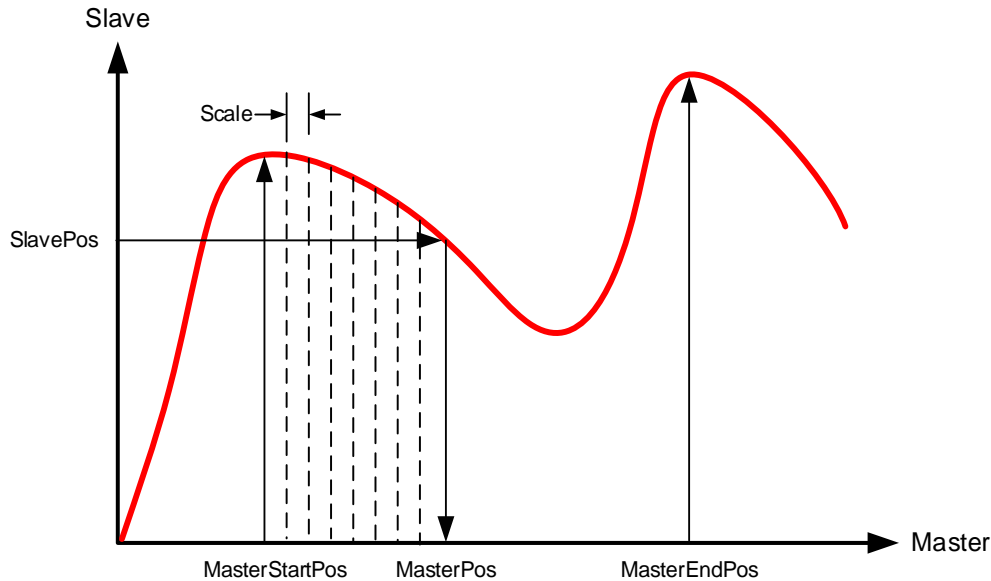
사용해 명령 현재 위치를 갱신합니다.

- (5) 출력 **InVelocity** 는 해당 축이 지정한 속도에 도달했을 때 온(On)되며, 지정속도 운전이 중단되면 오프(Off)됩니다.
- (6) 이 모션 평선 블록이 실행 중일 때 축 상태는 'ContinuousMotion'입니다.
- (7) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

LS_ReadCamTableMasterPos		적용 기종																												
캠 테이블 마스터 위치 읽기		XGF-M32E																												
평선 블록 형태																														
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">LS_ReadCamTableMasterPos</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">Done</td> <td style="width: 30%;">BOOL</td> </tr> <tr> <td>UINT</td> <td>Axis</td> <td>Axis</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>CamTableID</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>MasterStartPos</td> <td>MasterPos</td> <td>LREAL</td> </tr> <tr> <td>LREAL</td> <td>MasterEndPos</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>SlavePos</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Scale</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	Done	BOOL	UINT	Axis	Axis	UINT	UINT	CamTableID	Busy	BOOL	LREAL	MasterStartPos	MasterPos	LREAL	LREAL	MasterEndPos	Error	BOOL	LREAL	SlavePos	ErrorID	WORD	LREAL	Scale		
BOOL	Execute	Done	BOOL																											
UINT	Axis	Axis	UINT																											
UINT	CamTableID	Busy	BOOL																											
LREAL	MasterStartPos	MasterPos	LREAL																											
LREAL	MasterEndPos	Error	BOOL																											
LREAL	SlavePos	ErrorID	WORD																											
LREAL	Scale																													
입력-출력																														
UINT	Axis	지령 축을 설정합니다. (1~32: 실축, 37~40: 가상축)																												
입력																														
BOOL	Execute	상승 Edge 에서 해당 축에 캠테이블 마스터 위치 읽기 명령을 내립니다.																												
UINT	CamTableID	읽을 캠 테이블의 번호를 지정합니다. (1~32)																												
LREAL	MasterStartPos	캠 주축의 위치 읽기 시작위치																												
LREAL	MasterEndPos	캠 주축의 위치 읽기 종료위치																												
LREAL	SlavePos	캠 종축의 위치																												
LREAL	Scale	주축의 위치 읽기 정밀도																												
출력																														
BOOL	Done	캠테이블 주축 위치 읽기 동작이 성공적으로 완료되었음을 나타냅니다.																												
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.																												
LREAL	MasterPos	종축의 위치에 대응하는 주축의 위치를 출력 합니다.																												
BOOL	Error	에러 발생 여부를 나타냅니다.																												
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.																												

(1) 이 모션 평선 블록은 지정한 캠 테이블에서 MasterStartPos 부터 MasterEndPos 까지의 값중에서 SlavePos 에 설정된 종축의 위치에 대응하는 주축의 위치를 출력하는 기능입니다.



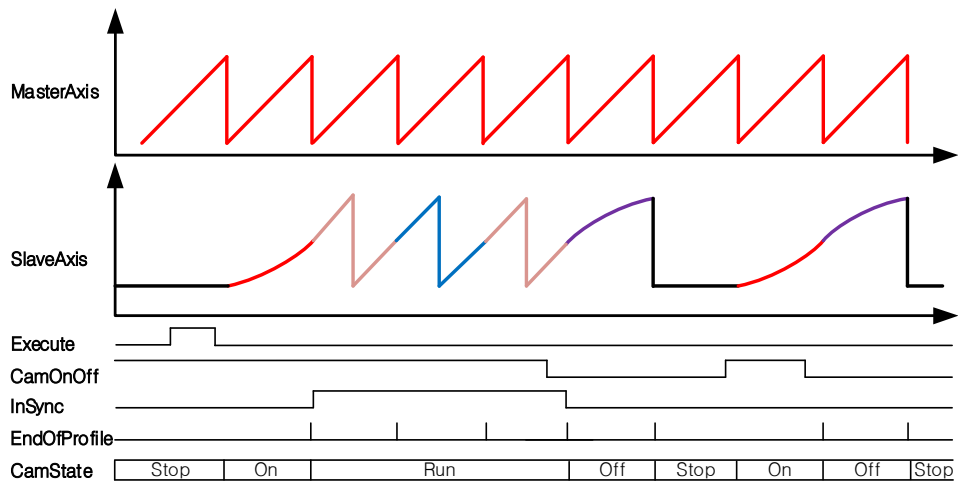
- (2) 캠 테이블에서 읽고자 하는 종축의 위치값을 **SlavePos** 값으로 설정합니다. 명령축에 적용된 **Offset/기어비/위상보정운전** 등은 **MasterPos** 출력에 반영되지 않습니다.
- (3) 캠테이블 마스터 위치읽기 동작이 완료되면 **Done** 출력이 **On**됩니다.
- (4) 캠테이블 마스터 위치 읽기의 정밀도값인 **Scale**에는 0을 입력할수 없습니다. **Scale**에 0을 입력한 경우 에러(에러번호: 0x0B)가 발생합니다. **Scale**값이 큰 경우에 축력된 **MasterPos**값과 실제 주축의 위치간에 오차가 발생할 수 있습니다. 또한 **Scale**값이 작은 경우에는 평선블록 수행시간이 길어질수 있습니다.
- (5) **MasterStartPos**와 **MasterEndPos**값 사이에 **SlavePos**에 설정한 종축에 해당하는 주축의 위치가 없는 경우에 **Error**가 **On**되고 **ErrorID**에 "0x1124" 이 발생합니다.
- (6) **MasterEndPos**값은 **MasterStartPos**값 보다 큰 값을 사용해야 합니다. **MasterEndPos**값이 **MasterStartPos**보다 작거나 같은 경우 **Error**가 **On**되고 **ErrorID**에 "0x0B" 이 발생합니다.
- (7) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목	모듈 O/S	XG-PM
제품명		
XGF-M32E	V1.30	V3.1

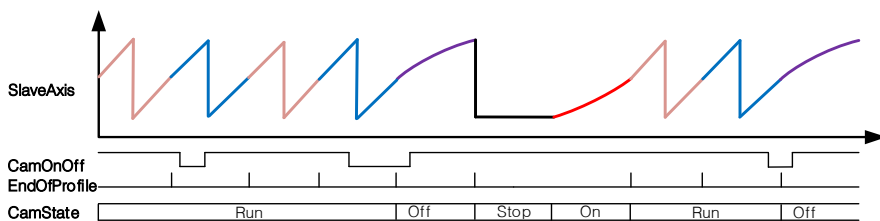
LS_OnOffCam		적용 기종																																																								
온 오프 캠 운전		XGF-M32E																																																								
평선 블록 형태																																																										
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">LS_OnOffCam</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">InSync</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>Master</td> <td>Master</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>Slave</td> <td>Slave</td> <td>UINT</td> </tr> <tr> <td>BOOL</td> <td>CamOnOff</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>BOOL</td> <td>SkipOnCam</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>BOOL</td> <td>SkipRunCam</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>MasterValueSource</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>OnCam_ID</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>UINT</td> <td>RunCam_ID</td> <td>EndOfProfile</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>OffCam_ID</td> <td>CamState</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>StartMode</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>StartModeParam</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>MasterScale</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>SlaveScale</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	InSync	BOOL	UINT	Master	Master	UINT	UINT	Slave	Slave	UINT	BOOL	CamOnOff	Busy	BOOL	BOOL	SkipOnCam	Active	BOOL	BOOL	SkipRunCam	CommandAborted	BOOL	UINT	MasterValueSource	Error	BOOL	UINT	OnCam_ID	ErrorID	WORD	UINT	RunCam_ID	EndOfProfile	BOOL	UINT	OffCam_ID	CamState	UINT	UINT	StartMode			LREAL	StartModeParam			LREAL	MasterScale			LREAL	SlaveScale		
BOOL	Execute	InSync	BOOL																																																							
UINT	Master	Master	UINT																																																							
UINT	Slave	Slave	UINT																																																							
BOOL	CamOnOff	Busy	BOOL																																																							
BOOL	SkipOnCam	Active	BOOL																																																							
BOOL	SkipRunCam	CommandAborted	BOOL																																																							
UINT	MasterValueSource	Error	BOOL																																																							
UINT	OnCam_ID	ErrorID	WORD																																																							
UINT	RunCam_ID	EndOfProfile	BOOL																																																							
UINT	OffCam_ID	CamState	UINT																																																							
UINT	StartMode																																																									
LREAL	StartModeParam																																																									
LREAL	MasterScale																																																									
LREAL	SlaveScale																																																									
입력-출력																																																										
UINT	Master	주축을 설정합니다. (1~32: 실축, 37~40: 가상축, 41~42: 엔코더), 변수만 설정 가능																																																								
UINT	Slave	종축을 설정합니다. (1~32: 실축, 37~40: 가상축), 변수만 설정 가능																																																								
입력																																																										
BOOL	Execute	상승 Edge 에서 해당 축에 온 오프 캠 운전 지령을 내립니다.																																																								
BOOL	CamOnOff	캠 동작의 OnOff 여부를 설정합니다. 1: OnCam 수행후 RunCam 으로 전환동작을 수행합니다. 0: RunCam 에서 OffCam 수행 후 캠운전이 정지상태로 전환됩니다.																																																								
BOOL	SkipOnCam	OnOff 캠운전에서 OnCam 을 제외하고 RunCam->OffCam 순서로 운전을 수행합니다.																																																								
BOOL	SkipRunCam	OnOff 캠운전에서 RunCam 을 제외하고 OnCam->OffCam 순서로 운전을 수행합니다.																																																								
UINT	MasterValueSource	캠 운전할 주축의 소스를 선택합니다. 0: 주축의 명령 위치에 동기화합니다. 1: 주축의 현재 위치에 동기화합니다.																																																								
UINT	OnCam_ID	OnCam 상태에서 운전할 캠 테이블을 지정합니다.																																																								
UINT	RunCam_ID	RunCam 상태에서 운전할 캠 테이블을 지정합니다.																																																								

UINT	OffCam_ID	OffCam 상태에서 운전할 캠 테이블을 지정합니다.
UINT	StartMode	캠운전을 시작하는 방법을 선택합니다. 0: CamOnOff 가 1 일 때 운전을 시작 1: CamOnOff 가 1 이고 주축이 StartModeParam 에 설정한 위치에 도달하면 시작 2: CamOnOff 가 1 이 입력된 시점에서 StartModeParam 에 설정한 거리만큼 이동하면 시작 3: LS_CrossSealCamGen 으로 생성한 프로파일 사용
LREAL	StartModeParam	캠운전 시작 방법에 따른 파라미터를 설정합니다.
LREAL	MasterScale	주축의 배율을 지정합니다.
LREAL	SlaveScale	종축의 배율을 지정합니다.
출력		
BOOL	InSync	RunCam 상태에 진입하였음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선블록이 축을 제어하는 중인지 여부를 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 다른 명령에 의해 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
BOOL	EndOfProfile	현재 운전중인 캠의 끝에 도달했음을 나타냅니다.
UINT	CamState	0: Stop 상태 1: OnCam 수행중 2: RunCam 수행중 3: OffCam 수행중

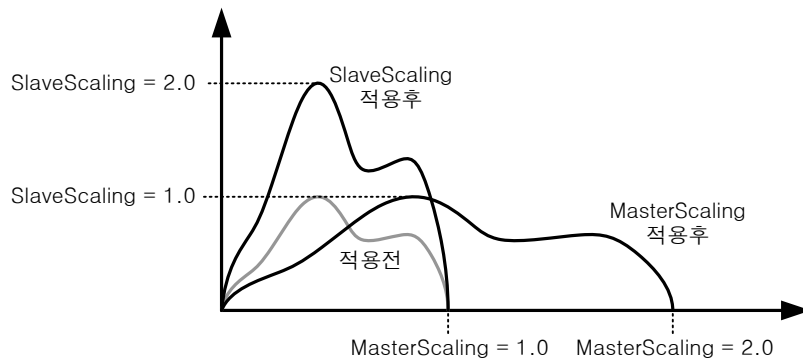
- (1) 이 모션 평선 블록은 캠테이블을 3 개를 사용하여 CamOnOff 입력에 따라 Stop 상태->OnCam->RunCam 또는 RunCam->OffCam->Stop 상태로 전환 되는 캠 운전을 수행하는 모션 평선 블록입니다.



- (2) Execute 가 상승 Edge 인 상태에서 캠 운전이 실행됩니다. 운전중에 Execute 를 오프(Off)로 변경하여도 캠 운전이 정지되지 않습니다. 온오프캠을 운전을 정지하기 위해서는 MC_CamOut 명령을 내리거나 다른 모션평선블록을 동작시켜야 합니다.
- (3) StartMode 가 0 으로 설정된 경우 CamOnOff 에 1 이 입력되면 즉시 OnCam 이 실행됩니다. StartMode 가 1 로 설정된 경우 CamOnOff 에 1 이 입력되어도 즉시 OnCam 이 실행되지 않고 주축의 위치가 StartModeParam 에 설정된 위치를 지나갈 때 OnCam 이 실행됩니다. StartMode 가 2 로 설정된 경우 CamOnOff 에 1 이 입력된 위치에서 StartModeParam 에 설정된 거리만큼 이동 후 OnCam 이 실행됩니다.
- (4) LS_CrossSealCamGen 평선블록으로 생성한 캠을 사용하는 경우 StartMode 를 3 으로 설정합니다. StartMode 가 3 으로 설정된 경우에는 OnCam_ID 의 길이가 270 인 경우 StartMode=1, StartModeParam=270 으로 설정된 것과 동일한 동작을 수행합니다. OnCam_ID 의 길이가 180 인 경우 StartMode=1, StartModeParam=0 으로 설정된 것과 동일한 동작을 수행합니다.
- (5) EndOfProfile 신호는 OnCam/OffCam/RunCam 의 각 캠 프로파일을 운전할 때 프로파일의 끝을 지날 때 출력이 온(On)됩니다.
- (6) CamOnOff 신호가 오프(Off)이면 RunCam->OffCam->Stop 상태로 전환되는 운전을 수행합니다. RunCam 상태에서 CamOnOff 신호가 Off 에서 On 으로 전환되는 경우 아직 OffCam 이 실행되지 않은 상태이면 RunCam 상태가 유지됩니다. OffCam 이 실행된 상태에는 OffCam->Stop 상태로 전환된 이후에 다시 OnCam->RunCam 상태로 전환됩니다. (RunCam 에서 CamOnOff 를 Off 하는 경우 EndOfProfile 신호가 발생할 때까지 유지해야 합니다.)



- (7) MasterScaling 및 SlaveScaling 에는 적용될 캠 데이터의 배율을 설정합니다. MasterScaling 은 주축 데이터의 배율을, SlaveScaling 에는 종축 데이터의 배율을 설정합니다. 아래 그림을 참고하시기 바랍니다.



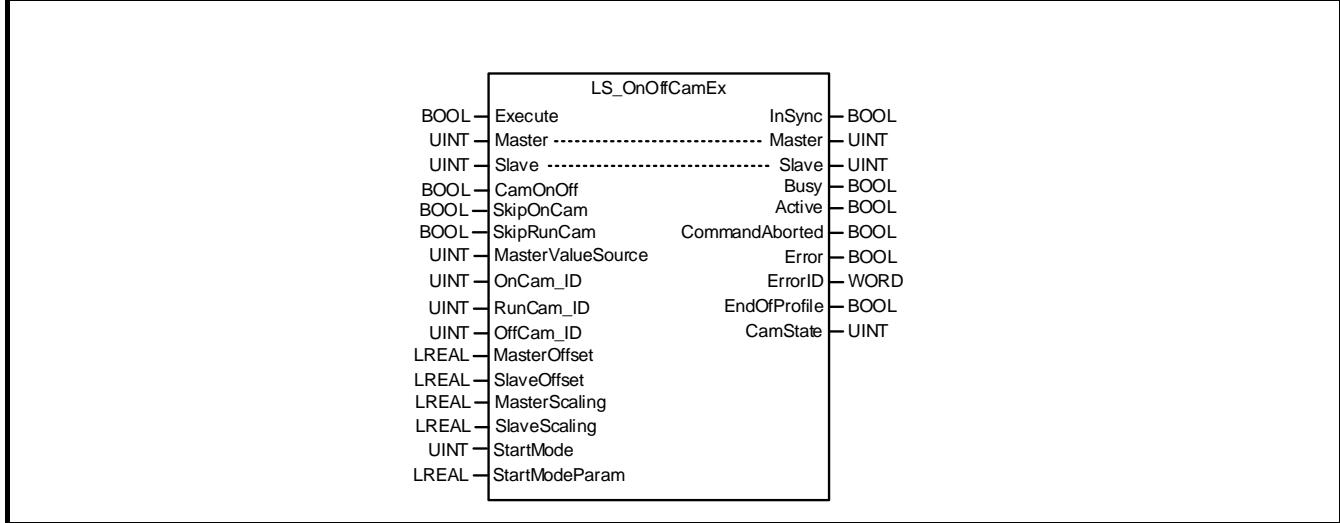
- (8) SkipOnCam 신호가 온(On)이면 OnCam 이 없이 RunCam 이 바로 실행됩니다. RunCam 실행이후 CamOnOff 신호가 오프되면 RunCam->OffCam->Stop 상태로 전환되는 운전을 수행합니다. SkipOnCam 신호가 온인 상태의 운전에서는 RunCam 의 중간부터 실행됩니다.
- (9) SkipRunCam 신호가 온(On)이면 OnCam 실행 이후 RunCam 이 실행되지 않고 OffCam 이 실행됩니다. 이때 CamOnOff 가 온이면 OnCam->OffCam->Stop->OnCam->OffCam->Stop 순으로 운전이 반복 됩니다.
- (10) 온오프 캠 운전을 완전히 정지하기 위해서는 정지(MC_Halt) 또는 즉시정지(MC_Stop)모션 평선블록을 사용하면 됩니다.
- (11) 캠운전의 상태에 따라 CamState 값이 Stop(0) /OnCam(1)/ RunCam(2)/ OffCam(3) 값으로 출력 됩니다.
- (12) RunCam_ID 에 설정된 캠 운전이 실행되면 InSync 출력이 온(On)됩니다.
- (13) MasterValueSource 는 동기할 주축의 소스를 선택합니다. 0 으로 설정하면 모션 컨트롤러에서 연산된 주축의 명령 위치를, 1 로 설정하면 주축 서보 드라이브에서 통신을 통해서 받은 현재위치를 기준으로 종축이 캠 운전을 합니다.
- (14) 온오프캠에서 운전중에 수행할 캠 프로파일을 RunCam_ID 로 설정 합니다. Stop 상태에서 RunCam 을 수행하기 전에 실행할 캠 프로파일을 OnCam_ID 로 설정합니다. OffCam_ID 에는 RunCam 이 Stop 상태로 도달하기 전에 실행할 캠 프로파일을 설정합니다. 각각의 ID 의 설정범위는 1~32 이며 입력 값이 설정 범위를 벗어나면 모션 평선 블록에서 에러“0x1115”이 발생합니다.
- (15) MasterValueSource/OnCam_ID/RunCam_ID/OffCam_ID 값은 운전중에 변경하여도 반영되지 않습니다.
- (16) 이 모션 평선 블록이 실행 중일 때 해당 축은 “동기운전(SynchronizedMotion)” 상태입니다.
- (17) 자세한 내용은 제품 사용설명서를 참조하시기 바랍니다.
- (18) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목	모듈 O/S	XG-PM
제품명		
XGF-M32E	V1.30	V3.1

LS_OnOffCamEx	적용 기종
----------------------	-------

온 오프 캠 운전	XGF-M32E
-----------	----------

평선 블록 형태



입력-출력

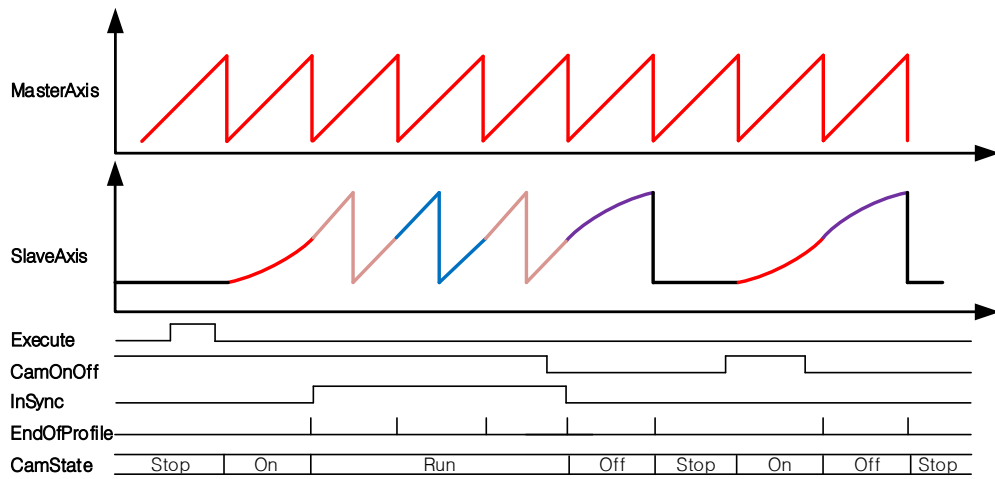
UINT	Master	주축을 설정합니다. (1~32: 실축/가상축, 33~36: 가상축, 1001~1002: 엔코더), 변수만 설정 가능
UINT	Slave	종축을 설정합니다. (1~32: 실축/가상축, 33~36: 가상축), 변수만 설정 가능

입력

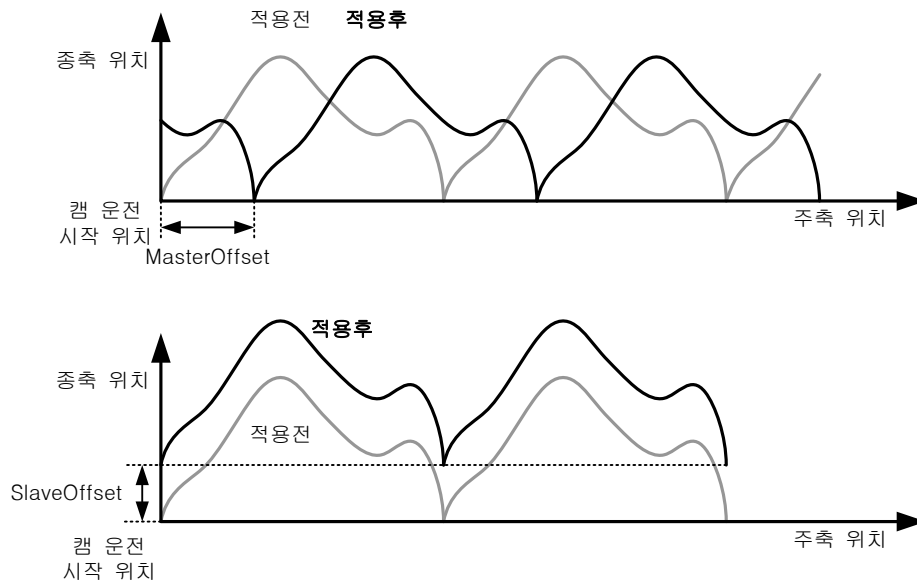
BOOL	Execute	상승 Edge 에서 해당 축에 온오프캠 운전 지령을 내립니다.
BOOL	CamOnOff	캠동작의 OnOff 여부를 설정합니다. 1: OnCam 수행후 RunCam 으로 전환동작을 수행합니다. 0: RunCam 에서 OffCam 수행 후 캠운전이 정지상태로 전환됩니다.
BOOL	SkipOnCam	OnOff 캠운전에서 OnCam 을 제외하고 RunCam->OffCam 순서로 운전을 수행합니다.
BOOL	SkipRunCam	OnOff 캠운전에서 RunCam 을 제외하고 OnCam->OffCam 순서로 운전을 수행합니다.
UINT	MasterValueSource	캠 운전할 주축의 소스를 선택합니다. 0: 주축의 명령 위치에 동기화합니다. 1: 주축의 현재 위치에 동기화합니다.
UINT	OnCam_ID	OnCam 상태에서 운전할 캠 테이블을 지정합니다.
UINT	RunCam_ID	RunCam 상태에서 운전할 캠 테이블을 지정합니다.

UINT	OffCam_ID	OffCam 상태에서 운전할 캠 테이블을 지정합니다.
LREAL	MasterOffset	주축의 오프셋 값을 설정합니다.
LREAL	SlaveOffset	종축의 오프셋 값을 설정합니다.
LREAL	MasterScaling	주축의 배율을 지정합니다.
LREAL	SlaveScaling	종축의 배율을 지정합니다.
UINT	StartMode	캠운전을 시작하는 방법을 선택합니다. 0: CamOnOff 가 1 일 때 운전을 시작 1: CamOnOff 가 1 이고 주축이 StartModeParam 에 설정한 위치에 도달하면 시작 2: CamOnOff 가 1 이 입력된 시점에서 StartModeParam 에 설정한 거리만큼 이동하면 시작 3: LS_CrossSealCamGen 으로 생성한 프로파일 사용
LREAL	StartModeParam	캠운전 시작 방법에 따른 파라미터를 설정합니다.
출력		
BOOL	InSync	RunCam 상태에 진입하였음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선블록이 축을 제어하는 중인지 여부를 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 다른 명령에 의해 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.
BOOL	EndOfProfile	현재 운전중인 캠의 끝에 도달했음을 나타냅니다.
UINT	CamState	0: Stop 상태 1:OnCam 수행중 2:RunCam 수행중 3:OffCam 수행중

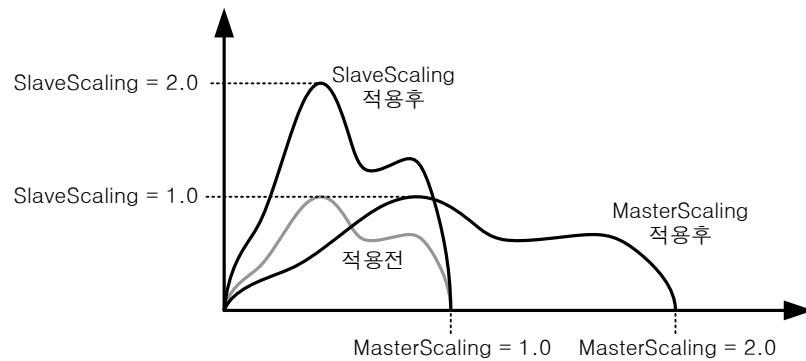
- (1) 이 모션 평선 블록은 캠테이블을 3 개를 사용하여 CamOnOff 입력에 따라 Stop 상태->OnCam->RunCam 또는 RunCam->OffCam->Stop 상태로 전환 되는 캠 운전을 수행하는 모션 평선 블록입니다.



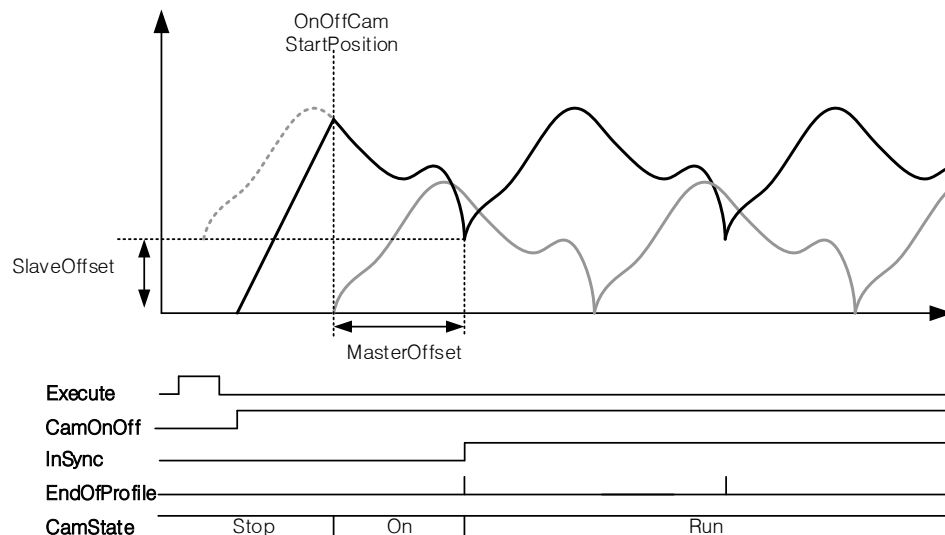
- (2) Execute 가 상승 Edge 인 상태에서 캠 운전이 실행됩니다. 운전중에 Execute 를 오프(Off)로 변경하여도 캠 운전이 정지되지 않습니다. 온오프캠을 운전을 정지하기 위해서는 MC_CamOut 명령을 내리거나 다른 모션평선블록을 동작시켜야 합니다.
- (3) MasterOffset 및 SlaveOffset 에는 적용할 캠 테이블의 오프셋을 설정합니다. MasterOffset 은 주축 시작점과의 오프셋을, SlaveOffset 은 종축 시작점과의 오프셋을 설정합니다. 아래 그림을 참고하시기 바랍니다.



- (4) MasterScaling 및 SlaveScaling 에는 적용될 캠 데이터의 배율을 설정합니다. MasterScaling 은 주축 데이터의 배율을, SlaveScaling 에는 종축 데이터의 배율을 설정합니다. 아래 그림을 참고하시기 바랍니다.

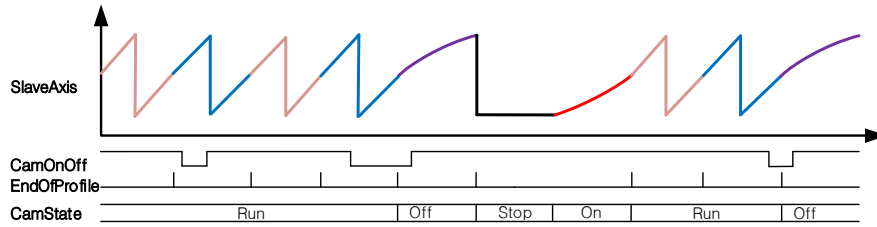


- (5) StartMode 가 0 으로 설정된 경우 CamOnOff 에 1 이 입력되면 즉시 OnCam 이 실행됩니다. StartMode 가 1 로 설정된 경우 CamOnOff 에 1 이 입력되어도 즉시 OnCam 이 실행되지 않고 주축의 위치가 StartModeParam 에 설정된 위치를 지나갈 때 OnCam 이 실행됩니다. StartMode 가 2 로 설정된 경우 CamOnOff 에 1 이 입력된 위치에서 StartModeParam 에 설정된 거리만큼 이동 후 OnCam 이 실행됩니다.
- (6) LS_CrossSealCamGen 평선블록으로 생성한 캠을 사용하는 경우 StartMode 를 3 으로 설정합니다. StartMode 가 3 으로 설정된 경우에는 OnCam_ID 의 길이가 270 인 경우 StartMode=1, StartModeParam=270 으로 설정된 것과 동일한 동작을 수행합니다. OnCam_ID 의 길이가 180 인 경우 StartMode=1, StartModeParam=0 으로 설정된 것과 동일한 동작을 수행합니다.
- (7) MasterOffset/SlaveOffset 이 설정된 경우 CamOnOff 에 1 이 입력되면 StartMode 와 StartModeParam 으로 설정된 OnOffCam 시작위치로 운전이 시작됩니다. OnOffCam 의 시작위치에 도달하면 OnOffCam 운전을 수행합니다. MasterOffset/SlaveOffset 이 설정하고 StartMode 가 0 으로 OnOffCam 운전을 수행하는 경우 운전 시작시 충격이 발생할수 있습니다.



- (8) EndOfProfile 신호는 OnCam/OffCam/RunCam 의 각 캠 프로파일을 운전할 때 프로파일의 끝을 지날 때 출력이 온(On)됩니다.
- (9) CamOnOff 신호가 오프(Off)이면 RunCam->OffCam->Stop 상태로 전환되는 운전을 수행합니다. RunCam 상태에서

CamOnOff 신호가 Off 에서 On 으로 전환되는 경우 아직 OffCam 이 실행되지 않은 상태이면 RunCam 상태가 유지됩니다. OffCam 이 실행된 상태에는 OffCam->Stop 상태로 전환된 이후에 다시 OnCam->RunCam 상태로 전환됩니다. (RunCam 에서 CamOnOff 를 Off 하는 경우 EndOfProfile 신호가 발생할 때까지 유지해야 합니다.)



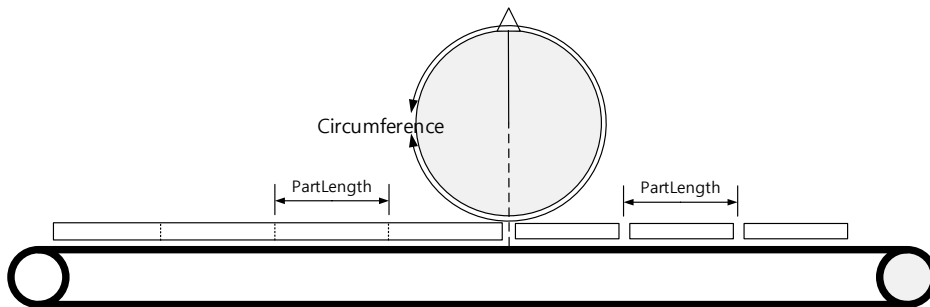
- (10) SkipOnCam 신호가 온(On)이면 OnCam 이 없이 RunCam 이 바로 실행됩니다. RunCam 실행이후 CamOnOff 신호가 오프되면 RunCam->OffCam->Stop 상태로 전환되는 운전을 수행합니다. SkipOnCam 신호가 온인 상태의 운전에서는 RunCam 의 중간부터 실행됩니다.
- (11) SkipRunnCam 신호가 온(On)이면 OnCam 실행 이후 RunCam 이 실행되지 않고 OffCam 이 실행됩니다. 이때 CamOnOff 가 온이면 OnCam->OffCam->Stop->OnCam->OffCam->Stop 순으로 운전이 반복 됩니다.
- (12) 온오프 캠 운전을 완전히 정지하기 위해서는 정지(MC_Halt) 또는 즉시정지(MC_Stop)모션 평선블록을 사용하면 됩니다.
- (13) 캠운전의 상태에 따라 CamState 값이 Stop(0) /OnCam(1)/ RunCam(2)/ OffCam(3) 값으로 출력 됩니다.
- (14) RunCam_ID 에 설정된 캠 운전이 실행되면 InSync 출력이 온(On)됩니다.
- (15) MasterValueSource 는 동기할 주축의 소스를 선택합니다. 0 으로 설정하면 모션 컨트롤러에서 연산된 주축의 명령 위치를, 1 로 설정하면 주축 서보 드라이브에서 통신을 통해서 받은 현재위치를 기준으로 종축이 캠 운전을 합니다.
- (16) 온오프캠에서 운전중에 수행할 캠 프로파일을 RunCam_ID 로 설정 합니다. Stop 상태에서 RunCam 을 수행하기 전에 실행할 캠 프로파일을 OnCam_ID 로 설정합니다. OffCam_ID 에는 RunCam 이 Stop 상태로 도달하기 전에 실행할 캠 프로파일을 설정합니다. 각각의 ID 의 설정범위는 1~32 이며 입력 값이 설정 범위를 벗어나면 모션 평선 블록에서 에러“ 0x1115”이 발생합니다.
- (17) MasterValueSource/OnCam_ID/RunCam_ID/OffCam_ID 값은 운전중에 변경하여도 반영되지 않습니다.
- (18) 이 모션 평선 블록이 실행 중일 때 해당 축은 “동기운전(SynchronizedMotion)” 상태입니다.
- (19) 자세한 내용은 제품 사용설명서를 참조하시기 바랍니다.
- (20) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목	모듈 O/S	XG-PM
제품명		
XGF-M32E	V1.30	V3.1

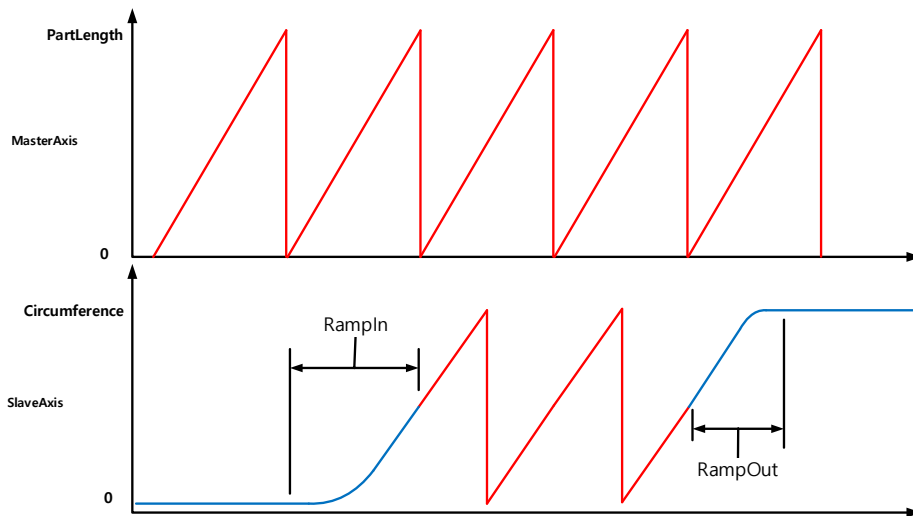
LS_RotaryKnifeCamGen		적용 기종
로터리커터 캠 프로파일 생성		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_RotaryKnifeCamGen Execute[BOOL Execute] Axis[UINT Axis] CamTableID[UINT CamTableID] PartLength[LREAL PartLength] Circumference[LREAL Circumference] CuttingStart[LREAL CuttingStart] CuttingEnd[LREAL CuttingEnd] CuttingSpdRatio[LREAL CuttingSpdRatio] CamType[UINT CamType] CamCurve[UINT CamCurve] CamPointNum[UINT CamPointNum] Done[BOOL Done] AxisOut[UINT Axis] Busy[BOOL Busy] Error[BOOL Error] ErrorID[WORD ErrorID] end Execute --- Done Axis --- AxisOut CamTableID --- Busy PartLength --- Error Circumference --- ErrorID CuttingStart --- Done CuttingEnd --- Done CuttingSpdRatio --- Done CamType --- Done CamCurve --- Done CamPointNum --- Done </pre> </div>		
입력-출력		
UINT	Axis	명령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)
입력		
BOOL	Execute	상승 Edge 에서 캠 프로파일 생성 동작을 수행합니다.
UINT	CamTableID	프로파일이 저장된 캠 테이블 ID 를 설정합니다.
LREAL	PartLength	로터리커터가 자르는 물체의 길이를 설정합니다.
LREAL	Circumference	로터리커터의 둘레의 길이을 설정합니다.
LREAL	CuttingStart	로터리커터가 커팅을 시작하는 위치를 설정합니다.
LREAL	CutingEnd	로터리커터가 커팅을 끝내는 위치를 설정합니다.
LREAL	CuttingSpdRatio	로터리커터가 커팅을 수행하는 동안 동기속도를 %단위로 조절합니다.(100 입력시 커팅속도는 주축과 1:1 동기)
UINT	CamType	생성하고자 하는 캠 프로파일의 타입을 설정합니다. (0:ALL 1:RampIn 2:Running 3:RampOut) (4:sALL 5:sRampIn 6:Running 7:sRampOut)
UINT	CamCurve	캠프로파일 생성에 사용하는 캠 커브 형식을 설정합니다.(0:Linear 1:Cubic)
UINT	CamPointNum	캠 프로파일에 사용되는 캠포인트의 개수를 설정합니다.
출력		

BOOL	Done	캠 프로파일이 성공적으로 생성되었음을 나타냅니다.
BOOL	Busy	평선블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	평선블록 실행 중 발생한 에러번호를 출력합니다.

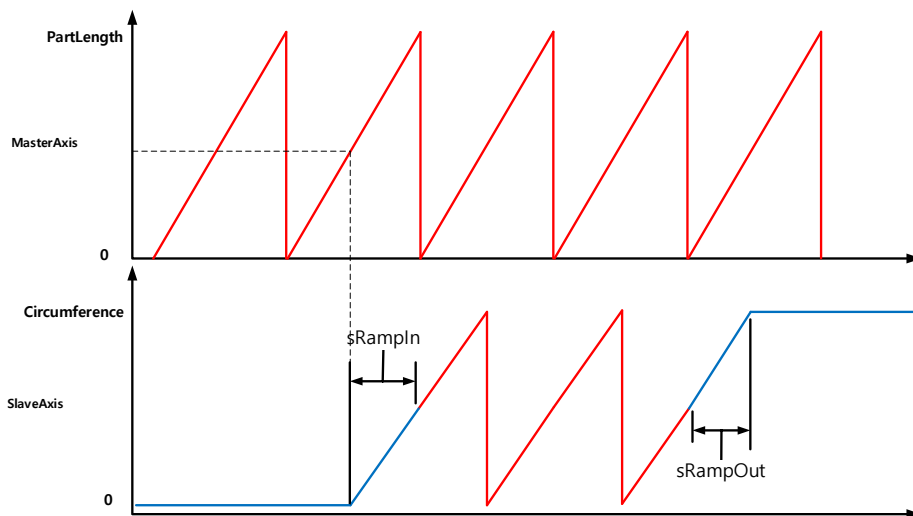
- (1) 이 모션 평선블록은 로터리커터동작을 수행하는 캠 프로파일을 생성하는 평선블록입니다.
- (2) LS_RotaryKnifeCamGen 을 통해 생성된 캠 프로파일을 LS_OnOffCam 평선블록에서 사용합니다.
- (3) PartLength 입력에 로터리커터를 사용하여 커팅을 수행하는 물체의 길이를 입력합니다.
- (4) Circumference 입력에 로터리커터의 둘레의 길이를 입력합니다.



- (5) CuttingStart 입력에 로터리커터가 커팅을 수행하는 위치를 입력합니다. CuttingEnd 입력에 로터리커터가 커팅을 완료하는 위치를 입력합니다. CuttingStart 와 CuttingEnd 사이에는 컨베이어와 로터리커터의 속도가 동기 됩니다. (Circumference 가 360 일 때 커팅구간을 10 을 주고 싶은 경우 CuttingStart 는 175 CuttingEnd 는 185 을 설정합니다.)
- (6) 생성되는 캠 프로파일에서 주축이 PartLength 만큼 이동은 360Degree 입니다. 축 파라미터의 모터축 기어비/기계축 기어비를 설정하여 실제 주축의 1 회전이 PartLength 와 같도록 설정하여야 합니다.
- (7) 생성되는 캠 프로파일에서 종축의 Circumference 만큼의 이동은 360Degree 입니다. 축 파라미터의 모터축 기어비/기계축 기어비를 설정하여 실제 종축의 1 회전이 Circumference 와 같도록 설정하여야 합니다.
- (8) CuttingStart 는 Circumference 의 1/8 보다 작거나 CuttingEnd 보다 큰 값을 입력할 수 없습니다. CuttingStart 값에 오류가 있는경우 0x1172 에러가 발생합니다.
- (9) CuttingEnd 는 CuttingEnd 보다 작거나 Circumference 의 7/8 보다 큰값을 입력할 수 없습니다. CuttingEnd 값에 오류가 있는경우 0x1173 에러가 발생합니다. Cutting 구간을 최소로 설정하려면 CuttingEnd 와 CuttingStart 를 동일한 값을 설정합니다.
- (10) CamType 입력에 생성할 캠 프로파일의 타입을 입력합니다. CamType 에는 1:Rampln 2:Running 3:RampOut 5:sRampln 6:Running 7:sRampOut 이 설정 가능합니다. CamType 에 0 을 입력하면 Rampln/Running/RampOut 이 한번에 생성됩니다. Running 은반복적으로 커팅동작을 수행하는 캠 프로파일을 생성합니다. Rampln 은 정지상태에서 커팅동작을 수행하는 Running 캠프로파일의 동작까지 들어가는 프로파일을 생성합니다. RampOut 은 Running 상태에서 로터리커터를 정지상태로 빠져 나오는 프로파일을 생성합니다. CamType 값이 범위를 벗어난 경우 0x1176 에러가 발생합니다.



- (11) **sRampIn** 과 **sRampOut** 은 **RampIn** 과 **RampOut** 보다 단축된 형태의 캠 프로파일을 생성합니다. **sRampIn** 와 **sRampOut** 을 사용하여 운전을 수행할 때 주축이 **PartLength** 위치에서 종축이 $1/2 \text{Circumference}$ 위치에 도달하기 위해서는 주축이 **PartLength** 의 $1/2$ 위치에서 시작해야 합니다.



- (12) **CuttingSpdRatio** 입력에 **Cutting** 구간에서의 속도비율을 설정합니다. **CuttingSpdRatio** 에 100 이 설정된 경우 **Cutting** 구간에서 주축의 속도와 1:1 동기하여 운전하는 캠 프로파일이 생성됩니다. **CuttingSpdRatio** 값이 커질수록 **Cutting** 구간에서의 동기속도가 빨라지게 됩니다. **CuttingSpdRatio** 값은 50~200 사이의 값을 설정할 수 있으며 **CuttingSpdRatio** 값에 오류가 있는 경우 0x1174 에러가 발생합니다.
- (13) **CamCurve** 입력에 생성할 캠 프로파일의 커브를 입력합니다. 0:Linear 을 입력하면 직선보간을 사용하여 캠 프로파일을 생성합니다. 직선보간을 선택한 경우 **CamPointNum** 을 선택하여 생성할 캠 프로파일의 포인트 수를 지정해 주어야 합니다. 캠 포인트수가 적은 경우 캠운전시 가감속에 따른 충격이 발생할 수 있으며 캠 포인트수가 많은 경우 캠 프로파일 생성과 캠 프로파일 저장에 많은 연산이 필요하기 때문에 전체 프로그램에 부담을 줄 수 있습니다. 1:Cubic 을 입력하면 Cubic 보간을 사용하는 캠 프로파일을 생성합니다. **CamCurve** 값이 범위를 벗어난 경우 0x1178 에러가 발생합니다.
- (14) **CamPointNum** 에 설정가능한 최소 캠포인트수는 10, 최대 캠포인트수는 1024 입니다. **CamPointNum** 에 오류가 있는 경우 0x1177 에러가 발생합니다.

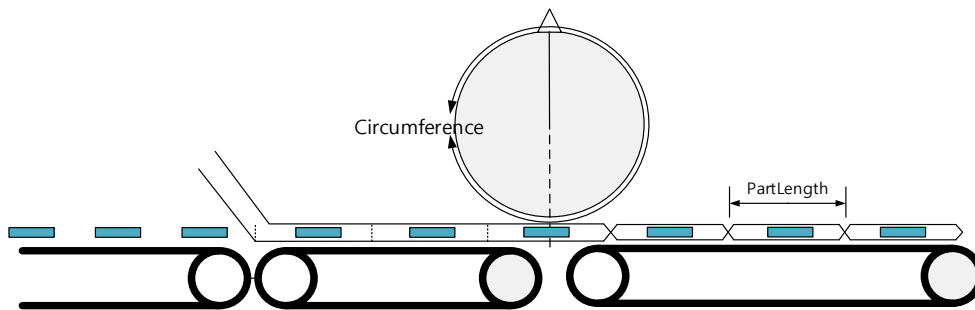
(15) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

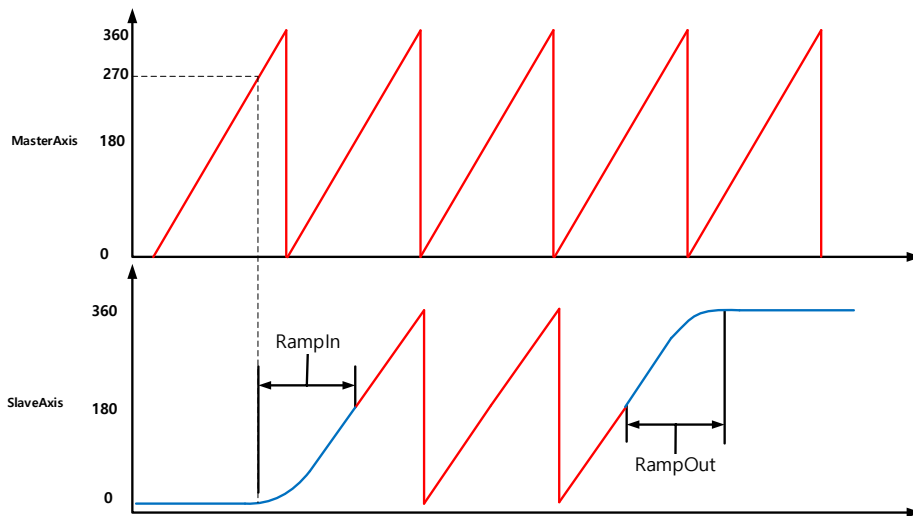
LS_CrossSea CamGen		적용 기종																																												
크로스 실러 캠 프로파일 생성		XGF-M32E																																												
평선 블록 형태																																														
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;"> <p style="text-align: center;">LS_CrossSealCamGen</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL Execute</td> <td style="width: 40%;"></td> <td style="width: 30%;">Done</td> <td style="width: 10%;">BOOL</td> </tr> <tr> <td>UINT Axis</td> <td style="text-align: center;">-----</td> <td>Axis</td> <td>UINT</td> </tr> <tr> <td>UINT CamTableID</td> <td></td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL PartLength</td> <td></td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL Circumference</td> <td></td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL SealStart</td> <td></td> <td></td> <td></td> </tr> <tr> <td>LREAL SealEnd</td> <td></td> <td></td> <td></td> </tr> <tr> <td>LREAL SealSpdRatio</td> <td></td> <td></td> <td></td> </tr> <tr> <td>UINT CamType</td> <td></td> <td></td> <td></td> </tr> <tr> <td>UINT CamCurve</td> <td></td> <td></td> <td></td> </tr> <tr> <td>UINT CamPointNum</td> <td></td> <td></td> <td></td> </tr> </table> </div>			BOOL Execute		Done	BOOL	UINT Axis	-----	Axis	UINT	UINT CamTableID		Busy	BOOL	LREAL PartLength		Error	BOOL	LREAL Circumference		ErrorID	WORD	LREAL SealStart				LREAL SealEnd				LREAL SealSpdRatio				UINT CamType				UINT CamCurve				UINT CamPointNum			
BOOL Execute		Done	BOOL																																											
UINT Axis	-----	Axis	UINT																																											
UINT CamTableID		Busy	BOOL																																											
LREAL PartLength		Error	BOOL																																											
LREAL Circumference		ErrorID	WORD																																											
LREAL SealStart																																														
LREAL SealEnd																																														
LREAL SealSpdRatio																																														
UINT CamType																																														
UINT CamCurve																																														
UINT CamPointNum																																														
입력-출력																																														
UINT	Axis	명령을 내릴 축 지정 (1~32: 실축, 37~40: 가상축)																																												
입력																																														
BOOL	Execute	상승 Edge 에서 캠 프로파일 생성 동작을 수행합니다.																																												
UINT	CamTableID	캠프로파일이 저장될 캠 테이블 ID 를 설정합니다.																																												
LREAL	PartLength	크로스 실러가 실링하는 물체의 길이를 설정합니다.																																												
LREAL	Circumference	크로스 실러의 둘레의 길이를 설정합니다.																																												
LREAL	SealStart	크로스 실러가 실링을 시작하는 위치를 설정합니다.																																												
LREAL	SealEnd	크로스 실러가 실링을 끝내는 위치를 설정합니다.																																												
LREAL	SealSpdRatio	크로스 실러가 실링을 수행하는 동안 동기속도를 %단위로 조절합니다.(100 입력시 실링속도는 주축과 1:1 동기)																																												
UINT	CamType	생성하고자 하는 캠 프로파일의 타입을 설정합니다. (0:ALL 1:RampIn 2:Running 3:RampOut) (4:sALL 5:sRampIn 6:Running 7:sRampOut)																																												
UINT	CamCurve	캠프로파일 생성에 사용하는 캠 커브 형식을 설정합니다.(0:Linear 1:Cubic)																																												
UINT	CamPointNum	캠 프로파일에 사용되는 캠포인트의 개수를 설정합니다.																																												
출력																																														

BOOL	Done	캠 프로파일이 성공적으로 생성되었음을 나타냅니다.
BOOL	Busy	평션블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	평션블록 실행 중 발생한 에러번호를 출력합니다.

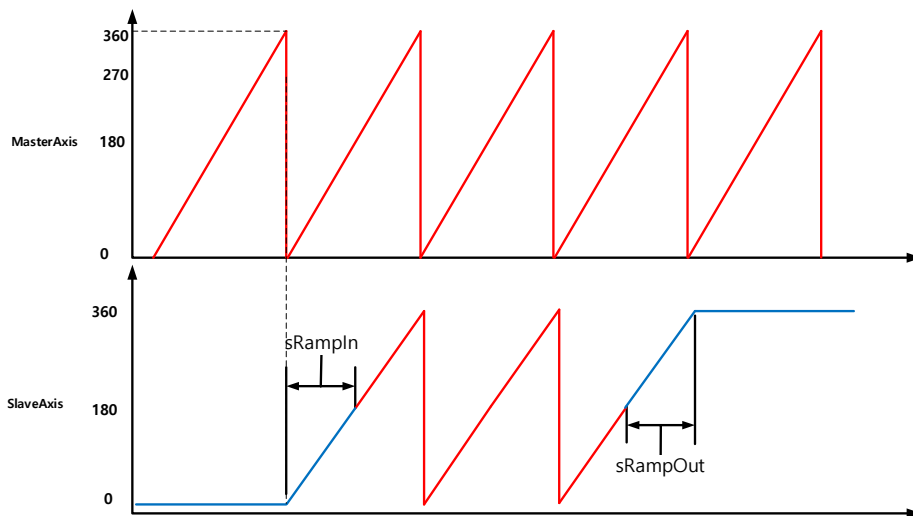
- (1) 이 모션 평션블록은 크로스 실러동작을 수행하는 캠 프로파일을 생성하는 평션블록입니다. LS_CrossSealCamGen 을 통해 생성된 캠 프로파일을 LS_OnOffCam 평션블록에서 사용합니다.
- (2) PartLength 입력에 크로스실러를 사용하여 실링을 수행하는 물체의 길이를 입력합니다.
- (3) Circumference 입력에 크로스실러의 둘레의 길이를 입력합니다.
- (4) 생성되는 캠 프로파일은 주축/종축 모두 0~360 의 범위로 출력됩니다. PartLength 와 Circumference 값은 주축과 종축이 360 값 이동시 실제 주축이 이동하는 거리값을 입력해야 합니다.



- (5) SealStart 입력에 크로스실러가 실링을 수행하는 위치를 입력합니다. SealEnd 입력에 크로스실러가 실링을 완료하는 위치를 입력합니다. SealStart 와 SealEnd 사이에는 컨베이어와 크로스실러의 속도가 동기 됩니다. (Circumference 가 360 일 때 실링구간을 10 을 주고 싶은 경우 SealStart 는 175 SealEnd 는 185 을 설정합니다.)
- (6) 생성되는 캠 프로파일에서 주축이 PartLength 만큼 이동은 360 입니다. 축 파라미터의 모터측 기어비/기계측 기어비를 설정하여 실제 주축이 360 이동할 때 실제이동이 PartLength 와 같도록 설정하여야 합니다.
- (7) 생성되는 캠 프로파일에서 종축의 Circumference 만큼의 이동은 360 입니다. 축 파라미터의 모터측 기어비/기계측 기어비를 설정하여 종축이 360 이동할 때 실제 종축의 이동거리가 Circumference 와 같도록 설정하여야 합니다.
- (8) SealStart 는 Circumference 의 1/8 보다 작거나 SealEnd 보다 큰 값을 입력할 수 없습니다. SealStart 값에 오류가 있는 경우 0x1172 에러가 발생합니다.
- (9) SealEnd 는 SealEnd 보다 작거나 Circumference 의 7/8 보다 큰값을 입력할 수 없습니다. SealEnd 값에 오류가 있는 경우 0x1173 에러가 발생합니다. 실링구간을 최소로 설정하려면 SealEnd 와 SealStart 를 동일한 값을 설정합니다.
- (10) CamType 입력에 생성할 캠 프로파일의 타입을 입력합니다. CamType 에는 1:RampIn 2:Running 3:RampOut 5:sRampIn 6:Running 7:sRampOut 이 설정 가능합니다. CamType 에 0 을 입력하면 RampIn/Running/RampOut 이 한번에 생성됩니다. Running 은반복적으로 실링동작을 수행하는 캠 프로파일을 생성합니다. RampIn 은 정지상태에서 실링동작을 수행하는 Running 캠프로파일의 동작까지 들어가는 프로파일을 생성합니다. RampOut 은 Running 상태에서 크로스실러를 정지상태로 빠져 나오는 프로파일을 생성합니다. CamType 값이 범위를 벗어난 경우 0x1176 에러가 발생합니다.



- (11) LS_CrossSealCamGen 평선에서 생성되는 캠 프로파일은 LS_RotaryCutCamGen 에서 생성되는 캠 프로파일과 유사합니다. RampIn 프로파일의 경우 주축이 0 위치에서 시작되지 않고 주축의 위치가 270 일 때 시작되는 운전으로 생성되어 있습니다. Seal 을 수행하는 위치도 주축의 위치가 180 도 위치에서 실링을 수행하도록 생성됩니다.
- (12) sRampIn 과 sRampOut 은 RampIn 과 RampOut 보다 단축된 형태의 캠 프로파일을 생성합니다. sRampIn 와 sRampOut 을 사용하여 운전을 수행할때는 주축이 0 위치에서 캠운전을 시작합니다.



- (13) SealSpdRatio 입력에 실링 구간에서의 속도비율을 설정합니다. SealSpdRatio 에 100 이 설정된 경우 실링구간에서 주축의 속도와 1:1 동기하여 운전하는 캠 프로파일이 생성됩니다. SealSpdRatio 값이 커질수록 Cutting 구간에서의 동기속도가 빨라지게 됩니다. SealSpdRatio 값은 50~200 사이의 값을 설정할 수 있으며 SealSpdRatio 값에 오류가 있는 경우 0x1174 에러가 발생합니다.
- (14) CamCurve 입력에 생성할 캠 프로파일의 커브를 입력합니다. 0:Linear 을 입력하면 직선보간을 사용하여 캠 프로파일을 생성합니다. 직선보간을 선택한 경우 CamPointNum 을 선택하여 생성할 캠 프로파일의 포인트 수를 지정해 주어야 합니다. 캠 포인트수가 적은 경우 캠운전시 가감속에 따른 충격이 발생할 수 있으며 캠 포인트수가 많은 경우 캠 프로파일 생성과 캠 프로파일 저장에 많은 연산이 필요하기 때문에 전체 프로그램에 부담을 줄 수 있습니다. 1:Cubic 을 입력하면 Cubic 보간을 사용하는 캠 프로파일을 생성합니다. CamCurve 값이 범위를 벗어난 경우 0x1178 에러가 발생합니다.

- (15) CamPointNum 에 설정가능한 최소 캠포인트수는 10, 최대 캠포인트수는 100 입니다. CamPointNum 에 오류가 있는 경우 0x1177 에러가 발생합니다.
- (16) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

MC_SetKinTransform		적용 기종
기구정보 설정		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_SetKinTransform Execute[Execute] AxesGroup[AxesGroup] KinType[KinType] KinExtParam[KinExtParam] KinParam[KinParam] ToolOffsetX[ToolOffsetX] ToolOffsetY[ToolOffsetY] ToolOffsetZ[ToolOffsetZ] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup KinType --- KinType KinExtParam --- KinExtParam KinParam --- KinParam ToolOffsetX --- ToolOffsetX ToolOffsetY --- ToolOffsetY ToolOffsetZ --- ToolOffsetZ Busy --- Busy Active --- Active CommandAborted --- CommandAborted Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	기구정보를 설정할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 축에 기구정보 설정 명령을 내립니다.
UINT	KinType	기구물 타입을 설정합니다.
UINT	KinExtParam	미사용
LREAL[]	KinParam	기구정보를 설정합니다.
LREAL	ToolOffsetX	기구물 끝의 X 축 오프셋을 설정합니다.
LREAL	ToolOffsetY	기구물 끝의 Y 축 오프셋을 설정합니다.
LREAL	ToolOffsetZ	기구물 끝의 Z 축 오프셋을 설정합니다.
출력		
BOOL	Done	기구정보 설정 동작이 성공적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 해당 축이 기구정보 설정이 동작 중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

(1) 이 모션 평선 블록은 AxesGroup 에 사전에 정의된 기구모델에 기반하여 ACS 와 MCS 기구학 변환을 설정합니다.

- (2) XG-PM 내 축 그룹 파라미터 설정에서도 동일하게 설정이 가능합니다.
- (3) KinType입력은 기구물의 타입을 설정하는데 사용되며 아래와 같이 기구물을 설정할수 있습니다.
 - 1) 0: 없음
 - 2) 1: XYZ
 - 3) 2: Delta3
 - 4) 3: Delta3R
 - 5) 4: LinearDelta3
 - 6) 5: LinearDelta3R
 - 7) 6: T-Gantry
 - 8) 7: T-GantryR
- (4) KinParam입력은 기구물의 정보를 설정하는데 사용됩니다.(XYZ타입의 경우 설정하지 않습니다.)
- (5) ToolOffsetX/ToolOffsetY/ToolOffsetZ입력은 기구물의 끝점에 오프셋을 설정하는 기능입니다.
- (6) 자세한 내용은 제품 사용설명서를 참고하시기 바랍니다.

MC_SetCartesianTransform		적용 기종
PCS 설정		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph MC_SetCartesianTransform Execute[Execute] AxesGroup[AxesGroup] TransX[TransX] TransY[TransY] TransZ[TransZ] RotAngleA[RotAngleA] RotAngleB[RotAngleB] RotAngleC[RotAngleC] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup TransX --- Busy TransY --- Active TransZ --- CommandAborted RotAngleA --- Error RotAngleB --- Error RotAngleC --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	PCS 를 설정할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 축그룹에 PCS 설정 명령을 내립니다.
LREAL	TransX	MCS 로부터 X 축 이동(mm)
LREAL	TransY	MCS 로부터 Y 축 이동(mm)
LREAL	TransZ	MCS 로부터 Z 축 이동(mm)
LREAL	RotAngleA	X 축 회전량(Degree)(예약)
LREAL	RotAngleB	Y 축 회전량(Degree)(예약)
LREAL	RotAngleC	Z 축 회전량(Degree)
출력		
BOOL	Done	PCS 설정 동작이 성공적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 해당 축이 기구정보 설정이 동작 중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 **AxesGroup** 에 **MCS** 와 **PCS** 사이의 직교좌표 변환을 설정합니다.
- (2) **XG-PM** 내 축 그룹 파라미터 설정에서도 동일하게 축 그룹 설정이 가능합니다.
- (3) 자세한 내용은 제품 사용설명서를 참고하시기 바랍니다.

LS_SetWorkspaceTransform		적용 기종
워크스페이스 설정		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_SetWorkspaceTransform Execute[Execute] AxesGroup[AxesGroup] WorkspaceType[WorkspaceType] WorkspaceErrorLevel[WorkspaceErrorLevel] WorkspaceParam[WorksapceParam] Done[Done] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup WorkspaceType --- Busy WorkspaceErrorLevel --- Active WorkspaceParam --- CommandAborted CommandAborted --- Error Error --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	워크스페이스를 설정할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 축그룹에 작업반경설정 명령을 내립니다.
UINT	WorkspaceType	작업반경의 형태를 설정합니다. (0:Rectangle 1:Cylinder 2:Delta3 3:Sector)
BOOL	WorkspaceErrorLevel	작업반경을 벗어나는 운전시 에러 발생 여부를 설정합니다. (0: 금지, 1: 허용)
LREAL[]	WorkspaceParam	작업반경의 파라미터를 설정합니다.
출력		
BOOL	Done	PCS 설정 동작이 성공적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 해당 축이 기구정보 설정이 동작 중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 AxesGroup 입력에 지정한 축 그룹에 좌표계기반 운전시 작업반경을 설정합니다.
- (2) XG-PM 내 축 그룹 파라미터 설정에서도 동일하게 설정이 가능합니다.
- (3) WorkspaceType는 (0:Rectangle 1:Cylinder 2:Delta3 3:Sector) 4가지 형태가 설정이 가능합니다.

- (4) WorkspaceErrorLevel 입력은 좌표계운전이 작업반경을 벗어나는 운전을 수행하는 경우 에러 발생 여부를 설정 합니다.
- (5) WorkspaceParam 입력은 작업반경의 타입에 따른 파라미터를 설정합니다.
- (6) 자세한 내용은 제품 사용설명서를 참고하시기 바랍니다.

LS_MoveLinearTimeAbsolute		적용 기종
좌표계 절대위치 시간직선보간운전		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_MoveLinearTimeAbsolute Execute[Execute] AxesGroup[AxesGroup] CoordSystem[CoordSystem] Position[Position] TrajType[TrajType] TrajTime[TrajTime] BufferMode[BufferMode] TransitionMode[TransitionMode] TransitionParameter[TransitionParameter] Done[Done] Busy[Busy] Active[Active] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup CoordSystem --- CoordSystem Position --- Position TrajType --- TrajType TrajTime --- TrajTime BufferMode --- BufferMode TransitionMode --- TransitionMode TransitionParameter --- TransitionParameter Done --- Done Busy --- Busy Active --- Active Error --- Error ErrorID --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	절대위치 시간직선보간운전을 수행할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 축그룹에 절대위치 시간직선보간운전 명령을 내립니다.
UINT	CoordSystem	좌표계 타입을 선택 합니다.(1:MCS 2:PCS)
LREAL[]	Position	기구물 끝점의 목표위치를 입력합니다.
UINT	TrajType	운전의 가/감속 타입을 입력합니다.(0:Trapezoid 1:Sine1 2:Sine2)
LREAL	TrajTime	목표위치에 도달하는 운전시간을 입력합니다.(msec)
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)
UINT	TransitionMode	미사용
LREAL	TransitionParameter	미사용
출력		
BOOL	Done	PCS 설정 동작이 성공적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 해당 축이 기구정보 설정이 동작 중임을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 **AxesGroup** 입력에 지정한 축 그룹에 좌표계기반의 절대위치 시간직선보간 명령을 시키는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록이 실행되면 각 축그룹의 현재 기구물의 끝점에서 목표위치까지 직선의 궤적으로 보간 제어를 실행합니다.
- (3) **TrajType** 입력은 보간 경로의 속도/가속도/감속도의 형태를 설정합니다. **Trapezoid/Sine1/Sine2** 3개의 타입이 선택 가능합니다.
- (4) **TrajTime** 입력은 목표위치에 도달하는 시간을 설정 합니다.
- (5) 자세한 내용은 제품 사용설명서를 참고하시기 바랍니다.

LS_MoveLinearTimeRelative		적용 기종
좌표계 상대위치 시간직선보간운전		XGF-M32E
평선 블록 형태		
<pre> graph LR subgraph LS_MoveLinearTimeRelative Execute[Execute] AxesGroup[AxesGroup] CoordSystem[CoordSystem] Position[Position] TrajType[TrajType] TrajTime[TrajTime] BufferMode[BufferMode] TransitionMode[TransitionMode] TransitionParameter[TransitionParameter] Done[Done] Busy[Busy] Active[Active] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup CoordSystem --- CoordSystem Position --- Position TrajType --- TrajType TrajTime --- TrajTime BufferMode --- BufferMode TransitionMode --- TransitionMode TransitionParameter --- TransitionParameter Done --- Done Busy --- Busy Active --- Active Error --- Error ErrorID --- ErrorID </pre>		
입력-출력		
UINT	AxesGroup	상대위치 시간직선보간운전을 수행할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 축그룹에 상대위치 시간직선보간운전 명령을 내립니다.
UINT	CoordSystem	좌표계 타입을 선택 합니다.(1:MCS 2:PCS)
LREAL[]	Position	기구물 끝점의 목표위치를 입력합니다.
UINT	TrajType	운전의 가/감속 타입을 입력합니다.(0:Trapezoid 1:Sine1 2:Sine2)
LREAL	TrajTime	목표위치에 도달하는 운전시간을 입력합니다.(msec)
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)
UINT	TransitionMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)
LREAL	TransitionParameter	그룹 운전의 경로변경 설정의 파라미터를 지정합니다. (6.1.4 BufferMode 입력 참조)
출력		
BOOL	Done	PCS 설정 동작이 성공적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 해당 축이 기구정보 설정이 동작 중임을 나타냅니다.

BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 **AxesGroup** 입력에 지정한 축 그룹에 좌표계기반의 상대위치 시간직선보간 명령을 시키는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록이 실행되면 각 축그룹의 현재 기구물의 끝점에서 목표위치까지 직선의 궤적으로 보간 제어를 실행합니다.
- (3) **TrajType** 입력은 보간 경로의 속도/가속도/감속도의 형태를 설정합니다. **Trapezoid/Sine1/Sine2** 3개의 타입이 선택 가능합니다.
- (4) **TrajTime** 입력은 목표위치에 도달하는 시간을 설정 합니다.

LS_MoveLinearAbsolute		적용 기종
좌표계 절대위치 직선보간운전		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> </div>		
입력-출력		
UINT	AxesGroup	절대위치 직선보간운전을 수행할 그룹을 설정합니다. (1 ~ 16: 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 축그룹에 절대위치 직선보간운전 명령을 내립니다.
LREAL[]	Position	기구물 끝점의 목표위치를 입력합니다.
LREAL	Velocity	최대 속도를 지정합니다. [u/s]
LREAL	Acceleration	최대 가속도를 지정합니다. [u/s ²]
LREAL	Deceleration	최대 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	최대 가/감속도 변화율을 지정합니다 [u/s ³]
UINT	CoordSystem	좌표계 타입을 선택 합니다.(1:MCS 2:PCS)
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)
UINT	TransitionMode	그룹 운전의 경로변경 모드를 지정합니다. (6.1.6. 그룹운전 경로 변경 설정 : TransitionMode 참조)
LREAL	TransitionParameter	그룹 운전의 경로변경 설정의 파라미터를 지정합니다. (6.1.6. 그룹운전 경로 변경 설정 : TransitionMode 참조)
출력		

BOOL	Done	좌표계 절대위치 직선보간 운전이 성공적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 해당 축이 기구정보 설정이 동작 중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 **AxesGroup** 입력에 지정한 축 그룹에 좌표계기반의 절대위치 직선보간 명령을 시키는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록이 실행되면 각 축그룹의 현재 기구물의 끝점에서 목표위치까지 직선의 궤적으로 보간 제어를 실행합니다.
- (3) **Velocity**, **Acceleration**, **Deceleration**, **Jerk** 입력에는 보간운전의 속도, 가속도, 감속도, 가감속 변화율을 각각 지정합니다.
- (4) **Velocity** 는 보간 속도를 설정하는 것으로 목표위치의 위치값(**Position[0]**, **Position[1]**, **Position[2]**)의 합성거리에 대한 속도가 됩니다. 목표위치의 위치값이 현재위치와 동일한 경우 목표위치의 각도값(**Position[3]**, **Position[4]**, **Position[4]**)의 합성각도에 대한 속도가 됩니다.
- (5) 명령이 종료되기 전에 평선 블록을 재 실행(**Execute** 입력을 **On**) 시키면 변경된 파라미터가 적용됩니다.
- (6) **Velocity**, **Acceleration**, **Deceleration**, **Jerk**, **Position** 입력만 업데이트 가능합니다.
- (7) **Velocity** 입력은 0으로 설정 또는 변경 가능합니다.
- (8) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

LS_MoveLinearRelative		적용 기종
좌표계 상대위치 직선보간운전		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph LS_MoveLinearRelative Execute[Execute] AxesGroup[AxesGroup] Distance[Distance] Velocity[Velocity] Acceleration[Acceleration] Deceleration[Deceleration] Jerk[Jerk] CoordSystem[CoordSystem] BufferMode[BufferMode] TransitionMode[TransitionMode] TransitionParameter[TransitionParameter] Done[Done] AxesGroup_Out[AxesGroup] Busy[Busy] Active[Active] CommandAborted[CommandAborted] Error[Error] ErrorID[ErrorID] end Execute --- Done AxesGroup --- AxesGroup_Out Distance --- Busy Velocity --- Active Acceleration --- CommandAborted Deceleration --- Error Jerk --- ErrorID CoordSystem --- Done BufferMode --- Done TransitionMode --- Done TransitionParameter --- Done </pre> </div>		
입력-출력		
UINT	AxesGroup	상대위치 직선보간운전을 수행할 그룹을 설정합니다. (1 ~ 16: 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 축그룹에 상대위치 직선보간운전 명령을 내립니다.
LREAL[]	Distance	기구물 끝점의 이동거리를 입력합니다.
LREAL	Velocity	최대 속도를 지정합니다. [u/s]
LREAL	Acceleration	최대 가속도를 지정합니다. [u/s ²]
LREAL	Deceleration	최대 감속도를 지정합니다. [u/s ²]
LREAL	Jerk	최대 가/감속도 변화율을 지정합니다 [u/s ³]
UINT	CoordSystem	좌표계 타입을 선택 합니다.(1:MCS 2:PCS)
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)
UINT	TransitionMode	그룹 운전의 경로변경 모드를 지정합니다. (6.1.6. 그룹운전 경로 변경 설정 : TransitionMode 참조)
LREAL	TransitionParameter	그룹 운전의 경로변경 설정의 파라미터를 지정합니다. (6.1.6. 그룹운전 경로 변경 설정 : TransitionMode 참조)
출력		

BOOL	Done	좌표계 상태위치 직선보간 운전이 성공적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 해당 축이 기구정보 설정이 동작 중임을 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

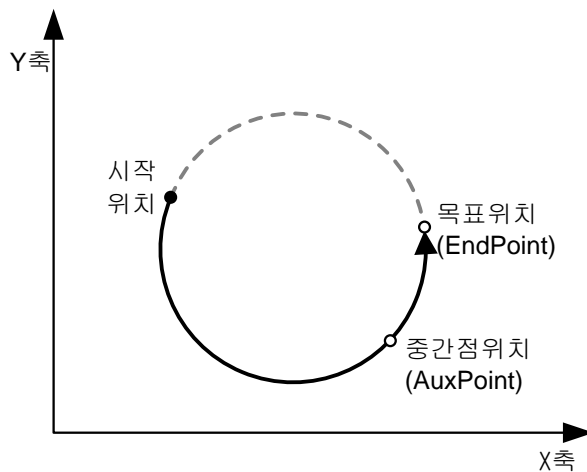
- (1) 이 모션 평선 블록은 **AxesGroup** 입력에 지정한 축 그룹에 좌표계기반의 상대위치 직선보간 명령을 시키는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록이 실행되면 각 축그룹의 현재 기구물의 끝점에서 목표위치까지 직선의 궤적으로 보간 제어를 실행합니다.
- (3) **Velocity, Acceleration, Deceleration, Jerk** 입력에는 보간운전의 속도, 가속도, 감속도, 가감속 변화율을 각각 지정합니다.
- (4) **Velocity** 는 보간 속도를 설정하는 것으로 목표위치의 이동거리값(**Distance [0], Distance [1], Distance [2]**)의 합성거리에 대한 속도가 됩니다. 목표위치의 위치값이 현재위치와 동일한 경우 목표위치의 이동각도값(**Distance[3], Distance [4], Distance [4]**)의 합성각도에 대한 속도가 됩니다.
- (5) 명령이 종료되기 전에 평선 블록을 재 실행(**Execute** 입력을 **On**) 시키면 변경된 파라미터가 적용됩니다.
- (6) **Velocity, Acceleration, Deceleration, Jerk, Position** 입력만 업데이트 가능합니다.
- (7) **Velocity** 입력은 0으로 설정 또는 변경 가능합니다.
- (8) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

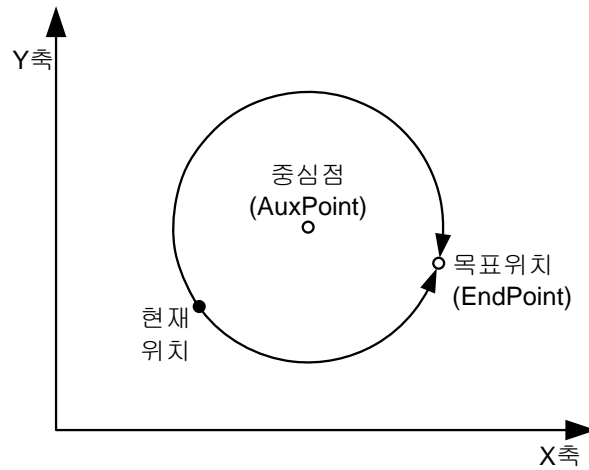
MC_MoveCircularAbsolute2D		적용 기종																																																								
좌표계 절대위치 원호보간 운전		XGF-M32E																																																								
평선 블록 형태																																																										
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <p style="text-align: center;">MC_MoveCircularAbsolute2D</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td style="width: 40%;">Execute</td> <td style="width: 30%;">Done</td> <td style="width: 10%;">BOOL</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>AxesGroup</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>CircMode</td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>AuxPoint</td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>EndPoint</td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>PathChoice</td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>CoordSystem</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>TransitionMode</td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>TransitionParameter</td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute	Done	BOOL	UINT	AxesGroup	AxesGroup	UINT	UINT	CircMode	Busy	BOOL	LREAL[]	AuxPoint	Active	BOOL	LREAL[]	EndPoint	CommandAborted	BOOL	UINT	PathChoice	Error	BOOL	LREAL	Velocity	ErrorID	WORD	LREAL	Acceleration			LREAL	Deceleration			LREAL	Jerk			UINT	CoordSystem			UINT	BufferMode			UINT	TransitionMode			LREAL	TransitionParameter		
BOOL	Execute	Done	BOOL																																																							
UINT	AxesGroup	AxesGroup	UINT																																																							
UINT	CircMode	Busy	BOOL																																																							
LREAL[]	AuxPoint	Active	BOOL																																																							
LREAL[]	EndPoint	CommandAborted	BOOL																																																							
UINT	PathChoice	Error	BOOL																																																							
LREAL	Velocity	ErrorID	WORD																																																							
LREAL	Acceleration																																																									
LREAL	Deceleration																																																									
LREAL	Jerk																																																									
UINT	CoordSystem																																																									
UINT	BufferMode																																																									
UINT	TransitionMode																																																									
LREAL	TransitionParameter																																																									
입력-출력																																																										
UINT	AxesGroup	절대위치 원호보간 운전을 할 그룹을 설정합니다. (1 ~ 16: 1 그룹 ~ 16 그룹)																																																								
입력																																																										
BOOL	Execute	상승 Edge 에서 해당 그룹에 절대위치 원호보간 운전 명령을 내립니다.																																																								
UINT	CircMode	원호보간 방법 설정 [0: 중간점, 1: 중심점, 2: 반지름]																																																								
LREAL[]	AuxPoint	원호보간 방법에 따른 보조점 위치를 절대좌표로 지정합니다.																																																								
LREAL[]	EndPoint	원호 끝 위치를 절대좌표로 지정합니다.																																																								
BOOL	PathChoice	원호 경로 선택 0: 시계방향, 1: 반시계 방향																																																								
LREAL	Velocity	경로의 최대 속도를 지정합니다. [u/s]																																																								
LREAL	Acceleration	최대 가속도를 지정합니다. [u/s ²]																																																								
LREAL	Deceleration	최대 감속도를 지정합니다. [u/s ²]																																																								
LREAL	Jerk	최대 가/감속도 변화율을 지정합니다 [u/s ³]																																																								
UINT	CoordSystem	좌표계 타입을 선택 합니다.(1:MCS 2:PCS)																																																								
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)																																																								

UINT	TransitionMode	미사용
LREAL	TransitionParameter	미사용
출력		
BOOL	Done	지정한 위치에 도달 완료 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 그룹을 제어하는 중인지 여부를 나타냅니다
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 **AxesGroup** 입력으로 지정한 축 그룹에 좌표계 기반의 절대위치 원호보간 명령을 내리는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록이 시작되면 각 축은 입력된 보조점을 참고로 하는 원호궤적 보간 제어를 실행하며 이동 방향은 **PathChoice** 입력에 의해 결정됩니다. **PathChoice** 입력을 0 으로 설정하면 시계방향, 1 로 설정하면 반시계방향으로 원호 보간 운전을 하게 됩니다.
- (3) **AuxPoint**, **EndPoint** 입력에는 각 축 원호보간시 참고할 보조점의 절대 위치를 배열로 지정합니다. **MC_MoveCircularAbsolute** 와 달리 입력은 X,Y,Z 순서로 대응됩니다.
- (4) **Velocity**, **Acceleration**, **Deceleration**, **Jerk** 입력에는 보간 경로의 속도, 가속도, 감속도, 가속도 변화율을 각각 지정합니다.
- (5) **CircMode** 입력에는 원호보간 방식을 설정합니다. **CircMode** 에 지정한 값에 다른 원호보간 방식은 아래와 같습니다.
 - (a) 중간점 지정 방식 원호보간(**CircMode = 0**)
 시작 위치에서 운전을 시작하고 지정된 중간 점 위치를 거쳐 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 시작 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 중간점 위치에는 **AuxPoint** 에 입력된 좌표가, 목표 위치에는 **EndPoint** 에 입력된 좌표가 절대값으로 대응됩니다.

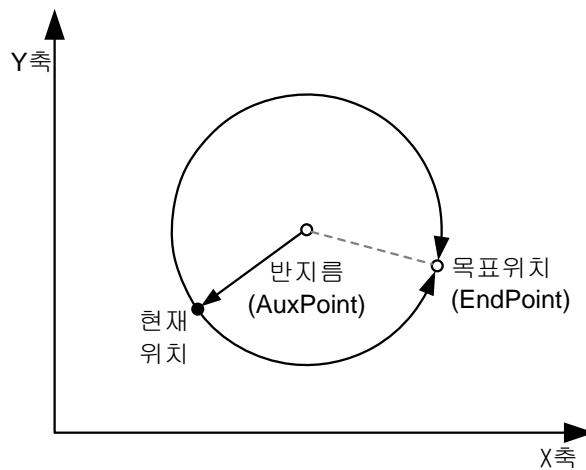


- (b) 중심점 지정 방식을 이용한 원호 보간(**CircMode = 1**)
 현재 위치에서 운전을 시작하고 지정된 중심점 위치까지의 거리를 반지름으로 하는 원호 궤적을 따라 목표 위치 까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 중심점 위치에는 **AuxPoint** 에 입력된 좌표가, 목표점에는 **EndPoint** 에 입력된 좌표가 절대값으로 대응됩니다.



(c) 반지름 지정 방식을 이용한 원호 보간 제어(CircMode = 2)

현재 위치에서 운전을 시작하고 현재 위치에서 반지름에 지정된 값을 반지름으로 하는 원호의 궤적을 따라 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 반지름에는 AuxPoint의 X 축에 입력된 값이, 목표점에는 EndPoint에 입력된 좌표가 절대값으로 대응됩니다



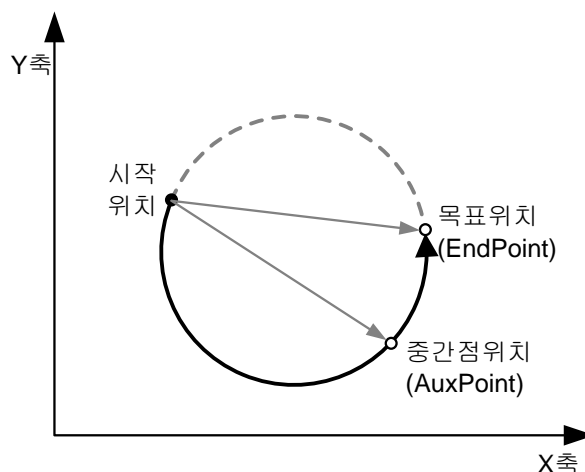
(6) 자세한 내용은 제품 사용설명서를 참고하시기 바랍니다.

(7) 명령이 종료되기 전에 평선 블록을 재 실행(Execute 입력을 On) 시키면 변경된 파라미터가 적용됩니다.
Velocity, Acceleration, Deceleration, Jerk, AuxPoint, Endpoint 입력만 업데이트 가능합니다.

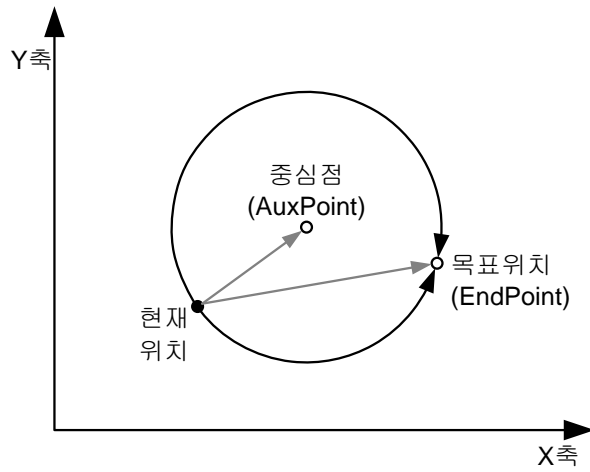
MC_MoveCircularRelative2D		적용 기종																																																																						
좌표계 상대위치 원호보간 운전		XGF-M32E																																																																						
평선 블록 형태																																																																								
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">MC_MoveCircularRelative2D</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">BOOL</td> <td>Execute</td> <td style="width: 30%;"></td> <td>Done</td> <td style="width: 30%;">BOOL</td> </tr> <tr> <td>UINT</td> <td>AxesGroup</td> <td>-----</td> <td>AxesGroup</td> <td>UINT</td> </tr> <tr> <td>UINT</td> <td>CircMode</td> <td></td> <td>Busy</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>AuxPoint</td> <td></td> <td>Active</td> <td>BOOL</td> </tr> <tr> <td>LREAL[]</td> <td>EndPoint</td> <td></td> <td>CommandAborted</td> <td>BOOL</td> </tr> <tr> <td>UINT</td> <td>PathChoice</td> <td></td> <td>Error</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>Velocity</td> <td></td> <td>ErrorID</td> <td>WORD</td> </tr> <tr> <td>LREAL</td> <td>Acceleration</td> <td></td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Deceleration</td> <td></td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>Jerk</td> <td></td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>CoordSystem</td> <td></td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>BufferMode</td> <td></td> <td></td> <td></td> </tr> <tr> <td>UINT</td> <td>TransitionMode</td> <td></td> <td></td> <td></td> </tr> <tr> <td>LREAL</td> <td>TransitionParameter</td> <td></td> <td></td> <td></td> </tr> </table> </div>			BOOL	Execute		Done	BOOL	UINT	AxesGroup	-----	AxesGroup	UINT	UINT	CircMode		Busy	BOOL	LREAL[]	AuxPoint		Active	BOOL	LREAL[]	EndPoint		CommandAborted	BOOL	UINT	PathChoice		Error	BOOL	LREAL	Velocity		ErrorID	WORD	LREAL	Acceleration				LREAL	Deceleration				LREAL	Jerk				UINT	CoordSystem				UINT	BufferMode				UINT	TransitionMode				LREAL	TransitionParameter			
BOOL	Execute		Done	BOOL																																																																				
UINT	AxesGroup	-----	AxesGroup	UINT																																																																				
UINT	CircMode		Busy	BOOL																																																																				
LREAL[]	AuxPoint		Active	BOOL																																																																				
LREAL[]	EndPoint		CommandAborted	BOOL																																																																				
UINT	PathChoice		Error	BOOL																																																																				
LREAL	Velocity		ErrorID	WORD																																																																				
LREAL	Acceleration																																																																							
LREAL	Deceleration																																																																							
LREAL	Jerk																																																																							
UINT	CoordSystem																																																																							
UINT	BufferMode																																																																							
UINT	TransitionMode																																																																							
LREAL	TransitionParameter																																																																							
입력-출력																																																																								
UINT	AxesGroup	상대위치 원호보간 운전을 할 그룹을 설정합니다. (1 ~ 16: 1 그룹 ~ 16 그룹)																																																																						
입력																																																																								
BOOL	Execute	상승 Edge 에서 해당 그룹에 상대위치 원호보간 운전 명령을 내립니다.																																																																						
UINT	CircMode	원호보간 방법 설정 [0: 중간점, 1: 중심점, 2: 반지름]																																																																						
LREAL[]	AuxPoint	원호보간 방법에 따른 보조점 위치를 시작점을 기준으로 한 상대좌표로 지정합니다.																																																																						
LREAL[]	EndPoint	원호 끝 위치를 시작점을 기준으로 한 상대좌표로 지정합니다.																																																																						
BOOL	PathChoice	원호 경로 선택 0: 시계방향, 1: 반시계 방향																																																																						
LREAL	Velocity	경로의 최대 속도를 지정합니다. [u/s]																																																																						
LREAL	Acceleration	최대 가속도를 지정합니다. [u/s ²]																																																																						
LREAL	Deceleration	최대 감속도를 지정합니다. [u/s ²]																																																																						
LREAL	Jerk	최대 가/감속도 변화율을 지정합니다 [u/s ³]																																																																						
UINT	CoordSystem	좌표계 타입을 선택 합니다.(1:MCS 2:PCS)																																																																						
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)																																																																						

UINT	TransitionMode	미사용
LREAL	TransitionParameter	미사용
출력		
BOOL	Done	지정한 위치에 도달 완료 여부를 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 모션 평선 블록이 그룹을 제어하는 중인지 여부를 나타냅니다.
BOOL	CommandAborted	현재 모션 평선 블록이 실행 중 중지되었음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 **AxesGroup** 입력으로 지정한 축 그룹에 좌표계 기반의 상대위치 원호보간 명령을 내리는 모션 평선 블록입니다.
- (2) 이 모션 평선 블록이 시작되면 각 축은 입력된 보조점을 참고로 하는 원호궤적 보간 제어를 실행하며 이동 방향은 **PathChoice** 입력에 의해 결정됩니다. **PathChoice** 입력을 0 으로 설정하면 시계방향, 1 로 설정하면 반시계방향으로 원호 보간 운전을 하게 됩니다.
- (3) **AuxPoint**, **EndPoint** 입력에는 각 축 원호보간시 참고할 보조점의 상대 위치를 배열로 지정합니다. 입력한 배열과 그룹 내 축은 지정한 축 ID [**ID1**, **ID2**, **ID3**, ...]의 순서로 대응됩니다. (모션 제어 모듈에서 원호보간 운전 명령을 내릴 그룹을 구성하는 축 수는 3 개이므로 **Position** 입력에는 **LEAL** 형 3 개 크기의 배열이 입력되어야 합니다.)
- (4) **Velocity**, **Acceleration**, **Deceleration**, **Jerk** 입력에는 보간 경로의 속도, 가속도, 감속도, 가속도 변화율을 각각 지정합니다.
- (5) **CircMode** 입력에는 원호보간 방식을 설정합니다. **CircMode** 에 지정한 값에 다른 원호보간 방식은 아래와 같습니다.
 - (a) 중간점 지정 방식 원호보간(**BORDER**, **CircMode** = 0)
현재 위치에서 운전을 시작하고 지정된 중간 점 위치를 거쳐 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 중간점 위치에는 **AuxPoint** 에 입력된 좌표가, 목표점 에는 **EndPoint** 에 입력된 좌표가 상대값으로 대응됩니다.

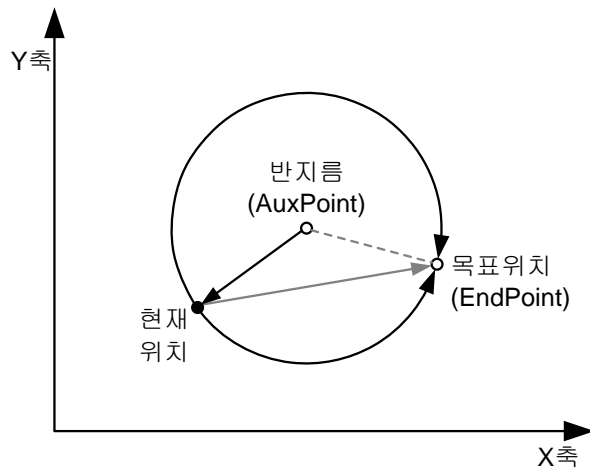


- (b) 중심점 지정 방식을 이용한 원호 보간
현재 위치에서 운전을 시작하고 지정된 중심점 위치까지의 거리를 반지름으로 하는 원호 궤적을 따라 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 중심점 위치에는 **AuxPoint** 에 입력된 좌표가, 목표점에는 **EndPoint** 에 입력된 좌표가 상대값으로 대응됩니다.



(c) 반지름 지정 방식을 이용한 원호 보간 제어

현재 위치에서 운동을 시작하고 현재 위치에서 반지름에 지정된 값을 반지름으로 하는 원호의 궤적을 따라 목표 위치까지 원호 보간을 하는 방식입니다. 아래 그림에서 현재 위치에는 명령 시작시의 축 그룹 좌표가 대응되며, 반지름에는 AuxPoint 의 X 축에 입력된 값이, 목표점에는 EndPoint 에 입력된 좌표가 상대값으로 대응됩니다



(6) 자세한 내용은 제품 사용설명서를 참고하시기 바랍니다.

MC_TrackConveyorBelt		적용 기종
컨베이어벨트 동기설정		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <pre> graph LR subgraph Inputs Execute[BOOL] AxesGroup[UINT] ConveyorAxis[UINT] ConveyorOrigin[ARRAY[0..5] OF LREAL[]] ObjectPosition[ARRAY[0..5] OF LREAL[]] CoordSystem[UINT] BufferMode[UINT] end subgraph Block [MC_TrackConveyorBelt] Done[BOOL] Busy[BOOL] Active[BOOL] Error[BOOL] ErrorID[WORD] end subgraph Outputs Done Busy Active Error ErrorID end Execute --- Block AxesGroup --- Block ConveyorAxis --- Block ConveyorOrigin --- Block ObjectPosition --- Block CoordSystem --- Block BufferMode --- Block Block --- Done Block --- Busy Block --- Active Block --- Error Block --- ErrorID </pre> </div>		
입력-출력		
UINT	AxesGroup	컨베이어벨트 동기설정을 수행할 그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 축그룹에 컨베이어벨트 동기설정 명령을 내립니다.
UINT	ConveyorAixs	컨베이어축을 설정합니다.(1 ~ 32 :1 축~32 축)
LREAL[]	ConveyorOrigin	MCS 원점으로부터 컨베이어 원점까지의 위치 입력합니다.
LREAL[]	ObjectPosition	컨베이어 원점으로부터 작업할 물체까지의 위치를 입력합니다.
UINT	CoordSystem	좌표계 타입을 선택 합니다.(2:PCS)
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)
출력		
BOOL	Done	PCS 설정 동작이 성공적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	현재 해당 축이 기구정보 설정이 동작 중임을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 AxesGroup 입력에 지정한 축 그룹에 컨베이어벨트 동기운전을 설정하는 평선 블록 입니다.
- (2) 이 모션 평선 블록은 직접운전을 수행하지 않으며 평선블록이 실행되면 이후 PCS좌표계를 사용하는 좌표계운전이 설정한 컨베이어벨트 축에 동기 하여 수행 됩니다.
- (3) ConveyorAxis는 1축~32축까지 설정이 가능하며 AxesGroup로 설정한 축그룹에 속하는 축으로 설정할 수 없습니다.

- (4) ConveyorAxis 로 설정하는 축의 운전파라미터는 반드시 mm/inch로 설정되어야 합니다.
- (5) ConveyorAxis로 설정하는 축의 운전파라미터는 무한길이 반복이 설정되어야 합니다.
- (6) 컨베이어 동기운전의 해제는 MCS좌표계를 사용하는 좌표계운전을 수행하거나 MC_SetCartesianTransform 펄스블록을 사용하여 PCS 설정을 수행하는 동작으로 수행됩니다.
- (7) 자세한 내용은 제품 사용설명서를 참고하시기 바랍니다.

MC_TrackRotaryTable		적용 기종
로터리테이블 동기설정		XGF-M32E
평선 블록 형태		
<div style="text-align: center;"> <p>The diagram shows a central box labeled 'MC_TrackRotaryTable'. On the left side, there are inputs: 'Execute' (BOOL), 'AxesGroup' (UINT), 'RotaryAxis' (UINT), 'RotaryOrigin' (ARRAY[0..5] OF LREAL[]), 'ObjectPosition' (ARRAY[0..5] OF LREAL[]), 'CoordSystem' (UINT), and 'BufferMode' (UINT). On the right side, there are outputs: 'Done' (BOOL), 'Busy' (BOOL), 'Active' (BOOL), 'Error' (BOOL), and 'ErrorID' (WORD). A dashed line connects 'AxesGroup' on the left to 'AxesGroup' on the right.</p> </div>		
입력-출력		
UINT	AxesGroup	로터리테이블 동기설정을 수행할 그룹을 설정합니다. (1 ~ 16: 1 그룹 ~ 16 그룹)
입력		
BOOL	Execute	상승 Edge 에서 해당 축그룹에 로터리테이블 동기설정 명령을 내립니다.
UINT	RotaryAxis	로터리테이블 구동축을 설정합니다.(1 ~ 36 :1 축~36 축)
LREAL[]	RotaryOrigin	MCS 원점으로부터 로터리테이블 원점까지의 위치 입력합니다.
LREAL[]	ObjectPosition	로터리테이블의 원점에서부터 작업할 물체까지의 위치를 입력합니다.
UINT	CoordSystem	좌표계 타입을 선택 합니다.(2:PCS)
UINT	BufferMode	모션 평선 블록의 순차운전 설정을 지정합니다. (6.1.4 BufferMode 입력 참조)
출력		
BOOL	Done	로터리테이블 동기설정이 성공적으로 완료되었음을 나타냅니다.
BOOL	Busy	모션 평선 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Active	모션 평선 블록이 동작 중임을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 평선 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 평선 블록은 AxesGroup 입력에 지정한 축 그룹에 로터리테이블 동기운전을 설정하는 평선 블록 입니다.
- (2) 이 모션 평선 블록은 직접운전을 수행하지 않으며 평선블록이 실행되면 이후 PCS좌표계를 사용하는 좌표계운전이 설정한 로터리테이블 축에 동기 하여 수행 됩니다.

- (3) RotaryAxis는 1축~36축까지 설정이 가능하며 AxesGroup로 설정한 축그룹에 속하는 축으로 설정할 수 없습니다.
- (4) RotaryAxis로 설정하는 축의 운전파라미터는 반드시 degree로 설정되어야 합니다.
- (5) RotaryAxis로 설정하는 축의 운전파라미터는 무한길이 반복이 설정되어야 합니다.
- (6) 로터리테이블 동기운전의 해제는 MCS좌표계를 사용하는 좌표계운전을 수행하거나 MC_SetCartesianTransform 평선블록을 사용하여 PCS 설정을 수행하는 동작으로 수행됩니다.
- (7) 자세한 내용은 제품 사용설명서를 참고하시기 바랍니다.
- (8) 이 모션 평선 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

LS_Robot Jog	적용 기종
좌표계 조그 운전	XGF-M32E

평선 블록 형태

LS_RobotJog

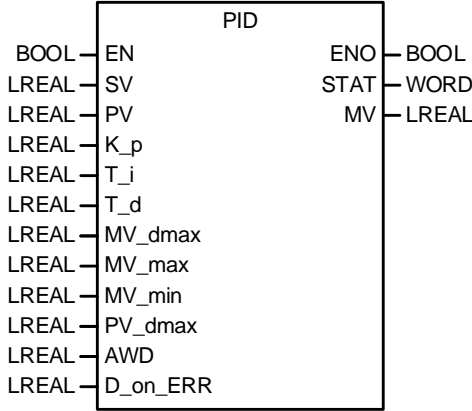
BOOL - Enable		Enabled - BOOL
UINT - AxesGroup	-----	AxesGroup - UINT
BOOL - Low_High		Busy - BOOL
BOOL - Pos_X		Error - BOOL
BOOL - Neg_X		ErrorID - WORD
BOOL - Pos_Y		
BOOL - Neg_Y		
BOOL - Pos_Z		
BOOL - Neg_Z		
BOOL - Pos_A		
BOOL - Neg_A		
BOOL - Pos_B		
BOOL - Neg_B		
BOOL - Pos_C		
BOOL - Neg_C		

입력-출력		
UINT	AxesGroup	명령을 내릴 축그룹을 설정합니다. (1 ~ 16 : 1 그룹 ~ 16 그룹)
입력		
BOOL	Enable	입력이 온(On) 되어 있는 동안 해당 축그룹에 조그 운전 명령을 내립니다.
BOOL	Low_High	조그 운전시 조그 속도를 설정합니다. (0: 조그 저속 운전, 1: 조그 고속 운전)
BOOL	Pos_X	조그 운전시 직선운전 방향을 설정합니다. (X 축 + 방향)
BOOL	Neg_X	조그 운전시 직선운전 방향을 설정합니다. (X 축 - 방향)
BOOL	Pos_Y	조그 운전시 직선운전 방향을 설정합니다. (Y 축 + 방향)
BOOL	Neg_Y	조그 운전시 직선운전 방향을 설정합니다. (Y 축 - 방향)
BOOL	Pos_Z	조그 운전시 직선운전 방향을 설정합니다. (Z 축 + 방향)
BOOL	Neg_Z	조그 운전시 직선운전 방향을 설정합니다. (Z 축 - 방향)
BOOL	Pos_A	조그 운전시 회전운전 방향을 설정합니다. (X 축 반시계방향 회전)
BOOL	Neg_A	조그 운전시 회전운전 방향을 설정합니다. (X 축 시계방향 회전)
BOOL	Pos_B	조그 운전시 회전운전 방향을 설정합니다. (Y 축 반시계방향 회전)
BOOL	Neg_B	조그 운전시 회전운전 방향을 설정합니다. (Y 축 시계방향 회전)

BOOL	Pos_C	조그 운전시 회전운전 방향을 설정합니다. (Z 축 반시계방향 회전)
BOOL	Neg_C	조그 운전시 회전운전 방향을 설정합니다. (Z 축 시계방향 회전)
출력		
BOOL	Enabled	해당 축그룹이 조그 운전 중임을 나타냅니다.
BOOL	Busy	모션 펄스 블록 수행이 완료되지 않았음을 나타냅니다.
BOOL	Error	에러 발생 여부를 나타냅니다.
WORD	ErrorID	모션 펄스 블록 실행 중 발생한 에러번호를 출력합니다.

- (1) 이 모션 펄스 블록은 해당 축그룹에 좌표계 조그 운전을 시키는 모션 펄스 블록입니다.
- (2) 조그 운전은 테스트를 위한 수동 운전 기능으로 시스템의 동작, 배선상태 검사 및 티칭을 위한 위치 어드레스 확인용으로 사용되며, 속도를 고속과 저속으로 구분하여 사용할 수 있습니다.
- (3) Enable 입력이 온(On)인 상태(조그 운전 중인 상태)에서 Low/High 에 설정된 값을 변경하면 조그 운전을 멈추지 않은 상태에서 속도 변경이 이루어 집니다.
- (4) 동일한 축에 대하여 정방향(Pos_)/역방향(Neg_) 입력을 모두 설정한 경우 해당 축의 운전이 정지 합니다.
- (5) 이 모션 펄스 블록을 사용할 수 있는 버전 정보는 아래와 같습니다.

항목 제품명	모듈 O/S	XG-PM
XGF-M32E	V1.30	V3.1

PID		적용 기종
PID 연산		XGF-M32E
평선 블록 형태		
<div style="text-align: center;">  <pre> graph LR subgraph PID EN[EN] SV[SV] PV[PV] Kp[K_p] Ti[T_i] Td[T_d] MVdmax[MV_dmax] MVmax[MV_max] MVmin[MV_min] PVdmax[PV_dmax] AWD[AWD] D_on_ERR[D_on_ERR] end EN --> ENO[ENO] SV --> STAT[STAT] PV --> MV[MV] </pre> </div>		
입력		
BOOL	EN	평선 블록을 실행합니다.
LREAL	SV	목표 값 (SV)
LREAL	PV	현재 값 (PV)
LREAL	K _p	P 상수 (K _p)
LREAL	T _i	I 상수 (T _i)[sec]
LREAL	T _d	D 상수 (T _d)[sec]
LREAL	MV _{dmax}	MV 변화량 제한
LREAL	MV _{max}	MV 최대값 제한
LREAL	MV _{min}	MV 최소값 제한
LREAL	PV _{dmax}	PV 변화량 제한
BOOL	AWD	Anti Wind-up 금지 (0:동작, 1:금지)
BOOL	D _{on_ERR}	미분 계산 소스 선택(0:PV, 1:ERR)
출력		
BOOL	ENO	PID 연산이 정상적으로 수행 중임을 나타냅니다.
WORD	STAT	PID 상태알람
LREAL	MV	출력값 (MV)

- (1) 이 평선 블록은 목표 값(SV)과 제어 대상의 현재 값(PV)을 받아 PID 연산을 수행하여 MV 로 출력하는 평선 블록입니다.
- (2) 목표 값 SV 입력은 제어 대상의 현재 상태입니다. 이 상태는 숫자로 표현되며, 시스템의 계인에 따라 PV 의 기준으로

변환하여 입력해야 합니다. 예를 들어, 온도가 50℃ 일 때 PV 가 5000 으로 센싱되는 시스템에서는 온도를 50℃로 제어할 때 SV 를 5000 으로 설정합니다.

- (3) 현재 값 PV 입력은 제어 대상의 현재 상태를 나타내는 지표로서, 일반적으로 센서로부터의 입력은 A/D 변환 모듈 등의 입력 장치를 거쳐 CPU 의 디바이스에 저장되고, 이 값을 전달받아 PV 입력으로 입력시켜 주어야 합니다.
- (4) K_p 입력은 현재 PID 연산기의 비례 상수를 설정합니다. K_p 는 PID 제어 효과 중 P, I, D (비례, 적분, 미분)항에 모두 곱해 지므로 K_p 가 커지면 비례, 미분 효과가 커지고 적분효과는 줄어듭니다. 특별히 K_p 입력이 0 인 경우에는 PID 제어를 하지 않습니다.
- (5) T_i 입력은 해당 루프의 적분상수를 설정합니다. T_i 는 PID 제어 효과 중 I(적분) 항을 나누므로 T_i 가 커지면 적분 효과가 작아 집니다. T_i 입력이 0 인 경우에는 I 제어를 하지 않습니다.
- (6) T_d 입력은 해당 루프의 미분 상수를 설정합니다. T_d 는 PID 제어 효과 중 D(미분) 항에 곱해 지므로 T_d 가 커지면 미분 효과가 커집니다. T_d 입력이 0 인 경우에는 D 제어를 하지 않습니다.
- (7) PV_{dmax} 입력은 해당 루프의 PV 변화량을 제한합니다. 실제 제어에 있어서 PV 는 항상 정확한 시스템의 상태를 반영하는 것은 아닙니다. 센서의 오동작이나 잡음, 외란 등의 원하지 않는 신호가 섞여서 PV 에 반영될 수 있습니다. 이와 같이 PV 가 순간적으로 급격히 변화하여 PID 출력 상에 큰 변화를 야기하는 경우도 있습니다. 이러한 현상을 방지하고자 _PID[B]_[L]dPV_{max} 에 설정된 값 이상의 PV 변화가 생기면, 이를 1 차적으로 막아서, 설정 값 이상의 변화를 막아 줍니다.

반면, PV_{dmax} 가 너무 작게 설정될 경우에는 시스템의 변화가 늦게 반영되어 수렴 시간이 오래 걸릴 수도 있으니, 시스템의 특성에 맞게 설정하시기 바랍니다. 특별히 해당 설정값이 0 으로 설정된 경우 PV 변화량을 제한기능은 동작하지 않습니다.

- (8) MV_{dmax} 입력은 해당 루프의 MV 변화량을 제한합니다. 제어 시스템의 출력이 급격하게 변하는 경우 시스템이 불안정 해지거나, 구동기에 큰 로드가 걸려서 고장 또는 동작이 불안정한 경우가 있습니다. 이를 막기 위해 제어기 출력 변화량을 제한하는 항목입니다. 특별히 해당 설정값이 0 으로 설정된 경우 본 기능은 동작하지 않습니다.
- (9) MV_{max} 입력은 해당 루프의 MV 최대값을 제한합니다. 출력 기기로 전달되는 제어기 출력의 최대값을 제한하여 과부하를 막아주고 시스템의 오류를 사전에 차단합니다. 또한 오버플로우 등으로 원치 않는 값이 전달되는 것을 막아 줍니다.
- (10) MV_{min} 입력은 해당 루프의 MV 최소값을 제한합니다. 출력 기기로 전달되는 제어기 출력의 최소값을 제한하여 시스템의 오류를 사전에 차단합니다. 또한 오버플로우 등으로 원치 않는 값이 전달되는 것을 막아 줍니다.
- (11) D_{on_EPR} 입력은 해당 PID 루프의 D 연산 소스를 EPR 로 설정합니다. D 연산은 EPR 또는 PV 로 연산되는데, EPR 을 이용하여 D 연산을 하게 될 경우 SV 가 유저에 의해 변경 되는 순간 D 응답이 급격한 변화를 보이므로 순간적으로 구동기 측에 과도한 입력이 가해질 수 있습니다. 이를 막기 위해 D 연산에서 PV 를 이용하는 방법이 사용되며 기본값 역시 PV 를 이용한 D 연산을 하도록 설정 되었습니다. 이러한 알고리즘 없이 EPR 을 사용할 경우 본 비트를 0n 시켜줍니다. 해당 비트가 0ff 인 경우 PID 는 PV 값으로 D 연산을 수행하고, 0n 상태에서는 EPR 값으로 D 연산을 수행합니다.
- (12) AWD 입력은 Anti Wind-up 기능을 활성화 하거나 비활성화 시키는 입력입니다. 해당 입력을 0n 하게 되면 Anti Wind-up 기능을 비활성화합니다.
- (13) STAT 출력의 각 비트들은 해당 PID 제어기의 상태 또는 이상상태를 표시합니다. 각 비트는 해당 동작이 일어난 순간에만 0n 되고 해당 동작이 해제되면 0ff 로 복귀합니다.

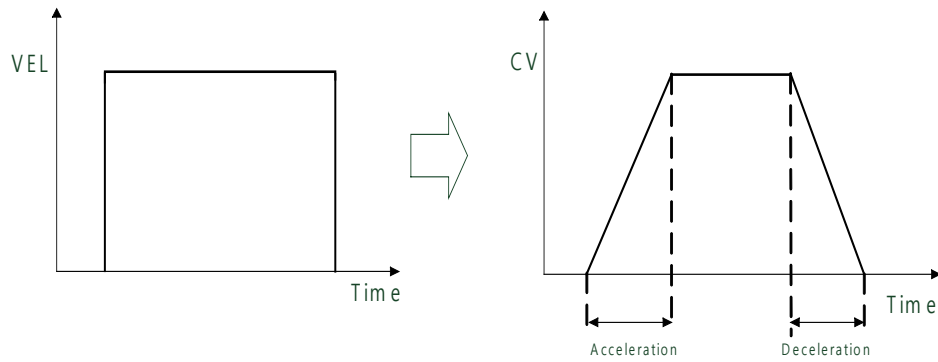
STAT 의 하위 8 비트는 해당 루프의 각종 이상 상태를 표시하며, 상위 8 비트는 해당 루프의 제어 상태를 표시합니다. 각각의 비트들의 할당은 다음과 같습니다.

비트	상태
0	T _s 설정이 너무 작아서 연산을 건너뛰는 중 임을 알립니다
2	PV 변화량이 제한되고 있음을 알립니다.
3	MV 변화량이 제한되고 있음을 알립니다.
4	MV 최대값이 제한되고 있음을 알립니다.
5	MV 최소값이 제한되고 있음을 알립니다.

8	PID 연산이 이루어 지고 있음을 알립니다.
15	PID 연산 중 Anti Wind-up 이 동작중 임을 알립니다

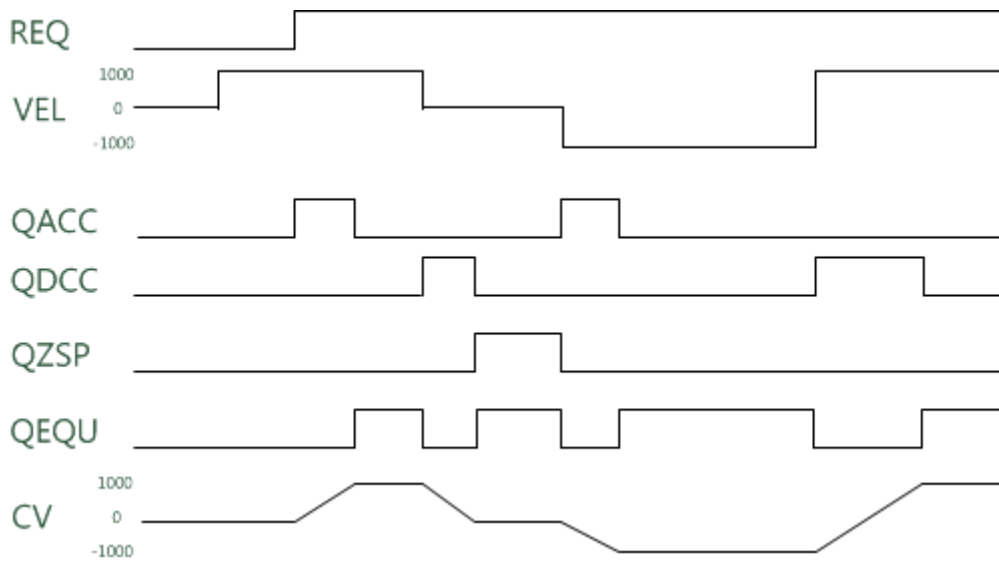
LINAC		적용 기종																															
가감속기 명령		XGF-M32E																															
평선 블록 형태																																	
<table border="1"> <tr> <td colspan="3">LINAC</td> </tr> <tr> <td>BOOL</td> <td>REQ</td> <td>DONE</td> <td>BOOL</td> </tr> <tr> <td>BOOL</td> <td>QS</td> <td>QACC</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>VEL</td> <td>QDCC</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>ACC</td> <td>QZSP</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>DCC</td> <td>QEQU</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>QCC</td> <td>CV</td> <td>LREAL</td> </tr> <tr> <td></td> <td></td> <td>DVDT</td> <td>LREAL</td> </tr> </table>			LINAC			BOOL	REQ	DONE	BOOL	BOOL	QS	QACC	BOOL	LREAL	VEL	QDCC	BOOL	LREAL	ACC	QZSP	BOOL	LREAL	DCC	QEQU	BOOL	LREAL	QCC	CV	LREAL			DVDT	LREAL
LINAC																																	
BOOL	REQ	DONE	BOOL																														
BOOL	QS	QACC	BOOL																														
LREAL	VEL	QDCC	BOOL																														
LREAL	ACC	QZSP	BOOL																														
LREAL	DCC	QEQU	BOOL																														
LREAL	QCC	CV	LREAL																														
		DVDT	LREAL																														
입력																																	
BOOL	REQ	LINAC 지령을 내립니다.																															
BOOL	QS	급정지 입력																															
LREAL	VEL	목표속도를 지정합니다. [u/s]																															
LREAL	ACC	가속도를 지정합니다. [u/s ²]																															
LREAL	DCC	감속도를 지정합니다. [u/s ²]																															
LREAL	QCC	급정지 감속도를 지정합니다. [u/s ²]																															
출력																																	
BOOL	DONE	평선블록 수행 완료 상태를 나타냅니다.																															
BOOL	QACC	가속중 여부를 나타냅니다.																															
BOOL	QDCC	감속중 여부를 나타냅니다.																															
BOOL	QZSP	현재속도의 영속 여부를 나타냅니다.																															
BOOL	QEQU	목표속도와 현재속도의 일치 여부를 나타냅니다.																															
LREAL	CV	현재속도를 나타냅니다.																															
LREAL	DVDT	현재 가감속도를 나타냅니다.																															

(1) 이 평선 블록은 입력속도까지 일정한 가/감속을 적용하여 도달하는 속도값을 출력하는 평선블록입니다.

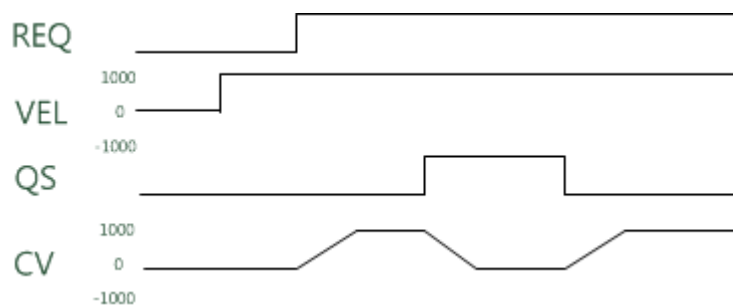


(2) REQ 입력이 상승엣지에서 ACC/DCC/QCC값이 평선블록에서 사용되며 동작중에 ACC/DCC/QCC값은 변경되지 않습니다.

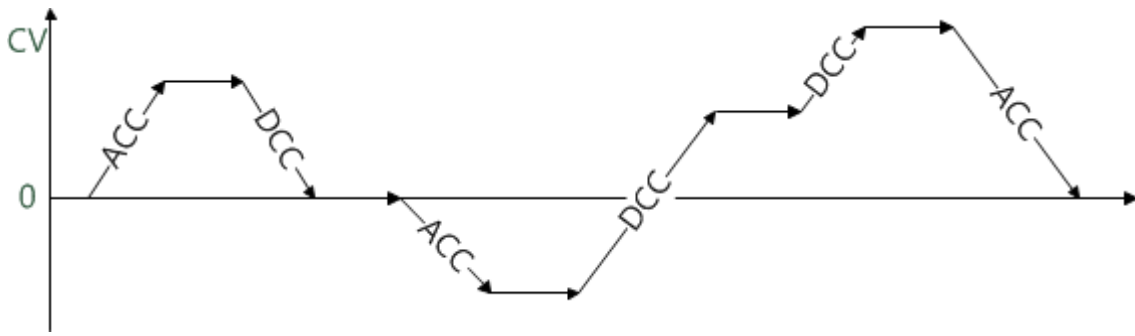
(3) 운전중 QACC/QDCC/QZSP/QEQU 출력은 다음과 같습니다.



(4) QS값이 1인 경우 QCC에서 설정한 감속도로 감속(급감속) 합니다. QS값이 0으로 변경되면 감속이 해제되며 입력된 목표속도로 가속/감속 합니다.

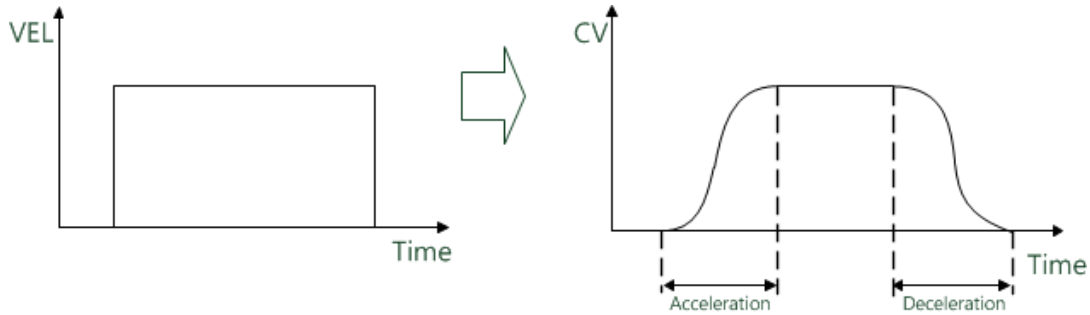


(5) 정지상태의 속도가 0일 때 입력된 목표속도의 방향으로 가속하고 반대방향으로는 감속합니다. 0 속도에서 정지하는 동작을 하는 경우 가속/감속의 방향이 변경됩니다.



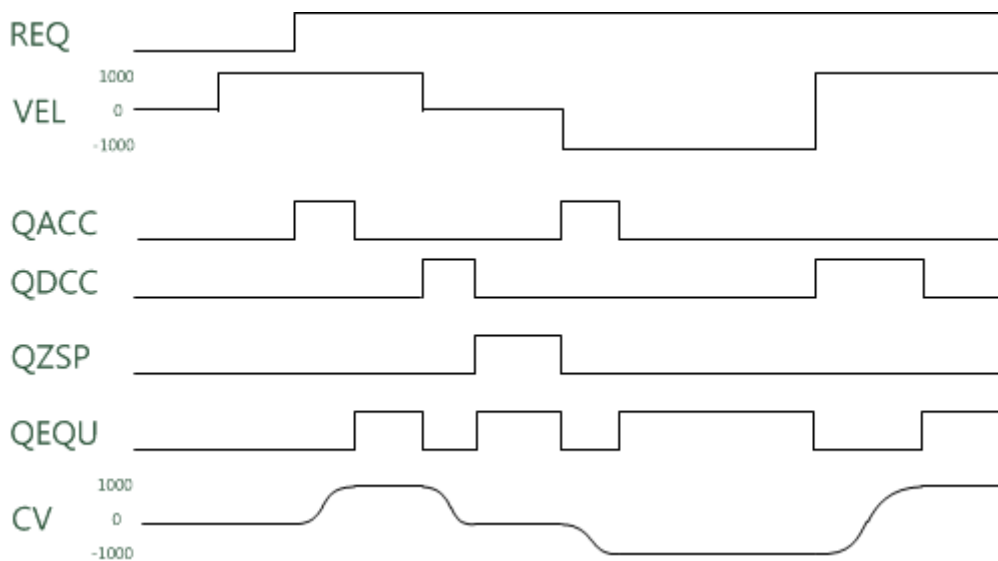
SINAC		적용 기종																																
S-커브 가감속기 명령		XGF-M32E																																
평선 블록 형태																																		
<table border="1"> <thead> <tr> <th colspan="4">SLINAC</th> </tr> </thead> <tbody> <tr> <td>BOOL</td> <td>REQ</td> <td>DONE</td> <td>BOOL</td> </tr> <tr> <td>BOOL</td> <td>QS</td> <td>QACC</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>VEL</td> <td>QDCC</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>ACC</td> <td>QZSP</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>DCC</td> <td>QEQU</td> <td>BOOL</td> </tr> <tr> <td>LREAL</td> <td>QCC</td> <td>CV</td> <td>LREAL</td> </tr> <tr> <td>LREAL</td> <td>JERK</td> <td>DVDT</td> <td>LREAL</td> </tr> </tbody> </table>			SLINAC				BOOL	REQ	DONE	BOOL	BOOL	QS	QACC	BOOL	LREAL	VEL	QDCC	BOOL	LREAL	ACC	QZSP	BOOL	LREAL	DCC	QEQU	BOOL	LREAL	QCC	CV	LREAL	LREAL	JERK	DVDT	LREAL
SLINAC																																		
BOOL	REQ	DONE	BOOL																															
BOOL	QS	QACC	BOOL																															
LREAL	VEL	QDCC	BOOL																															
LREAL	ACC	QZSP	BOOL																															
LREAL	DCC	QEQU	BOOL																															
LREAL	QCC	CV	LREAL																															
LREAL	JERK	DVDT	LREAL																															
입력																																		
BOOL	REQ	SLINAC 지령을 내립니다.																																
BOOL	QS	급정지 입력																																
LREAL	VEL	목표속도를 지정합니다. [u/s]																																
LREAL	ACC	가속도를 지정합니다. [u/s ²]																																
LREAL	DCC	감속도를 지정합니다. [u/s ²]																																
LREAL	QCC	급정지 감속도를 지정합니다. [u/s ²]																																
LREAL	JERK	가/감속도 변화율을 지정합니다. [u/s ³]																																
출력																																		
BOOL	DONE	평선블록 수행 완료 상태를 나타냅니다.																																
BOOL	QACC	가속중 여부를 나타냅니다.																																
BOOL	QDCC	감속중 여부를 나타냅니다.																																
BOOL	QZSP	현재속도의 영속 여부를 나타냅니다.																																
BOOL	QEQU	목표속도와 현재속도의 일치 여부를 나타냅니다.																																
LREAL	CV	현재속도를 나타냅니다.																																
LREAL	DVDT	현재 가감속도를 나타냅니다.																																
BOOL	DONE	평선블록 수행 완료 상태를 나타냅니다.																																

(1) 이 평선 블록은 입력속도까지 JERK가 적용된 가/감속을 적용하여 도달하는 속도값을 출력하는 평선블록입니다.



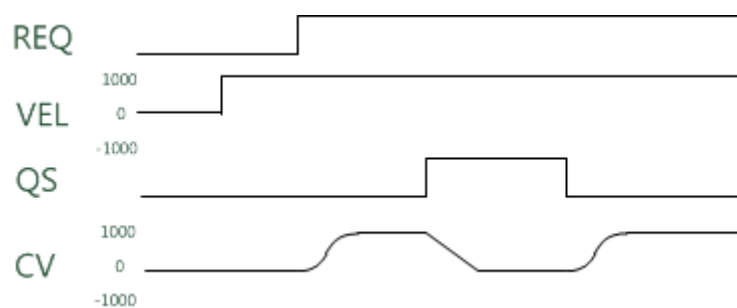
(2) REQ 입력이 상승엣지에서 ACC/DCC/QCC값이 평선블록에서 사용되며 동작중에는 ACC/DCC/QCC값은 변경되지 않습니다.

(3) 운전중 QACC/QDCC/QZSP/QEQU 출력은 다음과 같습니다.



(4) 목표속도에 도달하기 전에 목표속도 변경되는 경우 오버슈트나 언더슈트가 발생할수 있습니다.

(5) QS값이 1인 경우 QCC에서 설정한 감속도로 감속(급감속) 합니다. QS값이 0으로 변경되면 감속이 해제되며 입력된 목표속도로 가속/감속 합니다.



제11장 확장 평선

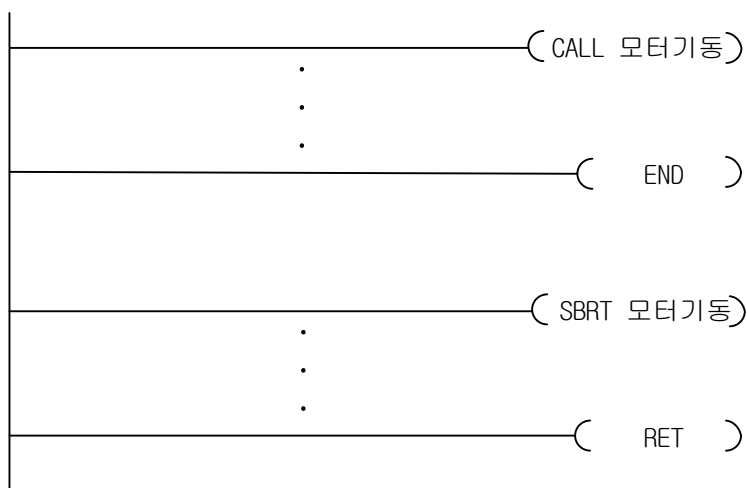
1. 각각의 확장 평선에 대한 설명입니다.
2. 사용자 프로그램 수행 도중 특정한 부분의 프로그램 처리(FOR ~ NEXT, CALL 명령 등을 사용)를 할 때 사용합니다.

CALL/SBRT/RET	적용 기종	발생플래그
호출 명령	XGF-M32E	-
평 선	설 명	
_____ (CALL NAME)	SBRT 루틴 호출	
_____ (SBRT NAME)	CALL 에 의해 호출될 루틴 지정	
_____ (RET)	RETURN	

■ 기능

1. 프로그램 수행 중 입력 조건이 성립하면 CALL n 명령에 따라 해당 SBRT n ~ RET 명령 사이의 프로그램을 수행합니다.
2. CALL n 은 중첩되어 사용 가능하며 반드시 SBRT n ~ RET 명령 사이에 프로그램은 END 명령 뒤에 있어야 합니다.
3. SBRT 내에서 다른 SBRT 를 CALL 하는 것이 가능합니다. 대신 SBRT 내에서는 END 명령을 사용하지 않습니다.
4. FOR ~ NEXT 루프를 빠져 나오는 방법은 BREAK 명령을 사용합니다.

■ 프로그램 예



- (1) 프로그램이 실행 도중 CALL 명령어를 만나면 모터기동이라는 이름의 SBRT 를 호출하게 됩니다.
- (2) SBRT 명령어는 반드시 END 뒤에 위치하여야 합니다.
- (3) 모터기동이라는 SBRT 가 호출되면 해당 SBRT 내의 프로그램이 실행되며 RET 를 만나면 다시 모터기동이라는 CALL 을 실행한 자리로 되돌아 갑니다.

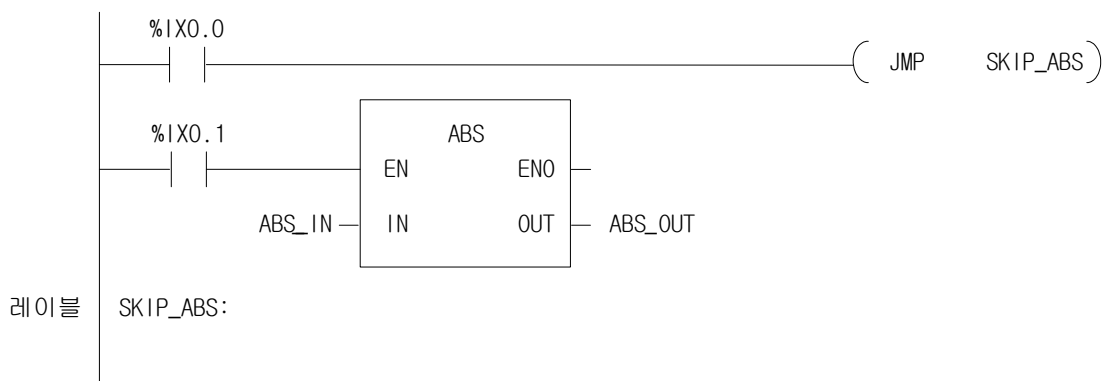
JMP	적용 기종	발생플래그
분기 명령	XGF-M32E	-
평 선	설 명	
	LABEL 위치로 점프	

■ 기능

1. JMP (레이블) 명령의 입력 접점이 On 되면 지정 레이블(LABEL) 이후로 Jump 하며 JMP 와 레이블 사이의 모든 명령 은 처리되지 않습니다.
2. 레이블은 중복되게 사용할 수 없습니다. JMP 는 중복사용 가능합니다.
3. 비상사태 발생시 처리해서는 안 되는 프로그램을 JMP 와 레이블 사이에 넣으면 좋습니다.

■ 프로그램 예

(1) 입력 신호 %IX0.0을 On 하였을 시 ABS 평선을 실행하지 않는 프로그램



END	적용 기종	발생플래그
종료 명령	XGF-M32E	-
평 선	설 명	
	프로그램의 종료	

■ 기능

1. 프로그램 종료를 표시합니다.
2. END 명령 처리 후 프로그램의 처음으로 돌아가 처리합니다.

제12장 ST (Structured Text)

12.1 개요

- 1) ST 프로그램은 모든 텍스트 편집기를 사용 가능하여 이식성이 뛰어납니다.
- 2) 복잡한 수식 및 알고리즘을 잘 표현할 수 있습니다.
- 3) 컴퓨터 언어에 익숙한 사람이 쉽게 프로그램 할 수 있습니다.

```

1
2 //FUNCTION 예제
3 CMD_TMR(IN:=%IX0.0 , PT:=T#300ms) ;
4 bb := CMD_TMR.Q ;
5
6 // IF문 예제
7 A := 1.0;
8 B := 1.000e+3;
9 C := 2.0;
10 D := B*B - 4*A*C ;
11 IF D < 0.0 THEN NROOTS := 0 ;
12 ELSIF D = 0.0 THEN
13     NROOTS := 1 ;
14     X1 := - B/(2.0*A) ;
15 ELSE
16     NROOTS := 2 ;
17     X1 := (- B + SQRT(D))/(2.0*A) ;
18     X2 := (- B - SQRT(D))/(2.0*A) ;
19 END_IF ;
20

```

12.2 비 실행문(설명문)

- 1) 비 실행문(설명문)은 2 가지 형태를 제공 합니다. 한라인 비 실행문과 블록 비 실행문 형태가 있습니다.
- 2) 한라인 비 실행문은 “//” 사용하며, 해당 라인의 끝까지 비 실행문 처리 됩니다.
- 3) 블록 비 실행문은 “(*” 과 “*)” 사이의 문자들을 비 실행문 처리 합니다.

예)


```

2
3 // 한라인 주석입니다.
4
5 (* 여러 라인
6   주석 입니다.
7 *)

```

12.3 표현식

- 1) 표현식은 연산자들과 피연산자들로 구성되어 있습니다. 피연산자는 정의된 문자(숫자 문자, 문자열, 시간문자), 정의된 변수(일반변수, 직접변수), 정의된 함수(평선, 평선 블록) 또는 다른 표현식일 수 있습니다. ST 언어의 연산자들은 <표 1>에 요약합니다.
- 2) 표현식의 계산은 <표 1>에 있는 연산자 우선 순위로 정의된 순서대로 연산자들을 피연산자에 적용함으로써 이루어집니다. 표현식에서 가장 높은 우선 순위를 가진 연산자가 첫 번째로 수행하고, 다음 우선순위를 가진 연산자가 차례로 수행합니다. 이러한 순서가 계산이 끝날 때까지 계속됩니다.
 예) $A+B*C$: 먼저 B와 C를 곱하고, 그 연산 결과를 A에 더합니다.

번호	연산	기호	우선순위
1	괄호로 묶음	(표현식)	가장 높음  가장 낮음
2	함수 계산	함수 식별자 (파라미터 리스트) 예) ADD(X, Y)	
3	부정 보수	- NOT	
4	지수	**	
5	곱셈 나눗셈 나머지	* / MOD	
6	더하기 빼기	+ -	
7	비교	<, >, <=, >=	
8	동등 (일치) 부등 (불일치)	= ◇	
9	부울 논리곱 부울 논리곱	& AND	
10	부울 배타적 논리합	XOR	
11	부울 논리합	OR	

<표 1> ST 언어의 연산자들

- 3) 동일한 우선순위를 가진 연산자들은 표현식의 왼쪽에서 오른쪽 순으로 수행합니다.
 예) $A+B-C$: 먼저 A와 B를 더하고, 그 연산 결과에 C를 뺍니다.
- 4) 연산자가 2개의 피연산자를 가질 때 왼쪽 끝의 피연산자가 먼저 수행합니다.
 예) $SIN(A)*COS(B)$: SIN(A)를 먼저 실행 후 COS(B)를 연산합니다.

5) 연산자를 수행 할 경우, 다음 조건은 에러로 처리합니다.

(1) 0의 값으로 나눌 경우

예) $A/(B * C)$: $B * C$ 연산 결과가 0일 경우 CPU에서 연산 에러 발생

(2) 피연산자가 연산을 위한 정확한 데이터 타입이 아닌 경우

예) $ADD(1, 2, 3)$: 숫자의 데이터 타입을 결정할 수 없어서 컴파일할 때 에러 발생

(3) 산술 연산 결과가 데이터 타입의 값의 범위를 벗어난 경우.

예) $B * C$: B, C가 UINT 타입일 경우, 연산 결과가 65,535 이상이면 CPU에서 연산 에러 발생

12.3.1 + 연산자

1) 덧셈 연산자는 연산자 양쪽의 두 값을 더하는데 사용합니다.

2) 문법

$result := expression1 + expression2$

항목	설명
<i>Result</i>	변수 또는 직접변수
<i>expression1</i>	ANY_NUM 타입의 표현식
<i>expression2</i>	ANY_NUM 타입의 표현식

사용 예	설명
Val1 := 20; Val2 := 4; Result := Val1 + Val2;	오른쪽의 두 개의 변수 값을 찾아 더하고 그 결과를 변수 Result에 대입 하라고 지시합니다. 결과 값은 24가 됩니다. 피연산자(Val1, Val2)는 상수와 변수가 모두 사용 가능합니다.

알아두기
ANY_NUM 타입은 ANY_REAL 타입과 ANY_INT 타입을 포함합니다. 자세한 사항은 3.2.2의 데이터 타입 계층도를 참고하시기 바랍니다.

12.3.2 - 연산자

1) 뺄셈 연산자는 - 연산자 앞에 있는 값에서 뒤에 있는 값을 빼는데 사용합니다.

2) 문법

$result := expression1 - expression2$

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY_NUM 타입의 표현식
<i>expression2</i>	ANY_NUM 타입의 표현식

사용 예	설명
<pre>Val1 := 20; Val2 := 4; Result := Val1 - Val2;</pre>	<p>- 기호 앞의 변수 값에서 뒤에 있는 변수 값을 빼서 그 결과를 변수 Result에 대입하라고 지시합니다.</p> <p>결과 값은 160이 됩니다.</p> <p>피연산자(Val1, Val2)는 상수와 변수가 모두 사용 가능합니다.</p>

12.3.3 * 연산자

- 1) 곱셈 연산자는 연산자 양쪽의 두 값을 곱하는데 사용합니다.
- 2) 문법

*result := expression1 * expression2*

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY_NUM 타입의 표현식
<i>expression2</i>	ANY_NUM 타입의 표현식

사용 예	설명
<pre>In1 := 2 ; Result := 20 * In1 ;</pre>	<p>20과 변수 In1의 값을 곱하여 Result 값에 대입합니다.</p> <p>결과 값은 400이 됩니다.</p> <p>피연산자는 상수와 변수가 모두 사용 가능합니다.</p>

12.3.4 / 연산자

- 1) 나눗셈 연산자는 / 연산자의 왼쪽에 있는 값을 연산자 오른쪽에 있는 값으로 나눕니다.
- 2) 나눗셈은 연산자와 함께 사용된 변수의 타입이 정수형인지 실수형 인지에 따라서 다르게 계산됩니다. 실수형 나눗셈은 실수형 값을 출력하고, 정수형 나눗셈은 정수형 값을 출력합니다. 정수는 소수점이 없으므로 5를 3으로 나누면, 일반적인 수학적 개념으로는 소수점이 있는 실수가 되므로 정수형이 아니라서 소수점 이하 부분을 버린 결과가 출력됩니다.

```
7 Result := 20 / INT_TYPE ;
8
9 Result1 := 20 / REAL_TYPE ;
```

```
7 Result = 6, INT_TYPE = 3
8
9 Result1 = 6.666666508e+000, REAL_TYPE = 3.000000000e+000
```

3) 문법

result := expression1 / expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY_NUM 타입의 표현식
<i>expression2</i>	ANY_NUM 타입의 표현식

사용 예	설명
In1 := 2 ; Result := 20 / In1 ;	20을 변수 In1의 값으로 나눠서 Result 값에 대입합니다. 결과 값은 10이 됩니다. 피연산자는 상수와 변수가 모두 사용 가능합니다.

알아두기
0 값으로 나눈 경우에 연산 에러 플래그(_ERR)가 On 됩니다. 이 경우에도 CPU는 런 모드를 유지합니다.

12.3.5 MOD 연산자

1) 나머지 연산자는 MOD 연산자의 왼쪽에 있는 값을 연산자 오른쪽에 있는 값으로 나누었을 때의 나머지를 결과로 출력합니다.

2) 문법

result := expression1 MOD expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY_NUM 타입의 표현식
<i>expression2</i>	ANY_NUM 타입의 표현식

사용 예	설명
In1 := 10 ; Result := 12 MOD In1 ;	12를 변수 In1의 값으로 나눈 나머지 2 값을 Result 값에 대입합니다. 피연산자는 상수와 변수가 모두 사용 가능합니다.

알아두기
0 값으로 나눈 경우에 연산 에러 플래그(_ERR)가 On 됩니다. 이 경우에도 CPU는 런 모드를 유지합니다.

12.3.6 ** 연산자

- 1) 지수 연산자는 **연산자 왼쪽의 값을 연산자 오른쪽의 승수만큼 곱하는 것입니다.
- 2) 문법

*result := expression1 ** expression2*

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY_REAL 타입의 표현식
<i>expression2</i>	ANY_REAL 타입의 표현식

사용 예	설명
In1 := 3 ; Result := 10 ** In1 ;	10를 변수 In1의 값만큼 곱하여 Result 값에 대입합니다. 결과 값은 1000이 됩니다. 피연산자는 상수와 변수가 모두 사용 가능합니다.

12.3.7 AND 또는 & 연산자

- 1) 이항 연산자(AND 또는 &)는 두 피연산자 사이의 비트끼리 비교합니다. 두 피연산자에 대응하는 비트가 모두 1이면 결과 비트도 1이 됩니다.
- 2) 문법

result := expression1 AND expression2 또는 *result := expression1 & expression2*

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY_BIT 타입의 표현식
<i>expression2</i>	ANY_BIT 타입의 표현식

AND 비트 연산 수행은 아래와 같습니다.

<i>expression1</i>	<i>expression2</i>	<i>result</i>
0	0	0
0	1	0
1	0	0
1	1	1

사용 예	설명
Result := 2#10010011 AND 2#00111101 ;	두 피연산자에서 5번째 비트와 1번째 비트만 둘다 1이므로 Result 값은 2#00010001이 됩니다. 피연산자는 상수와 변수가 모두 사용 가능합니다.

12.3.8 OR 연산자

1) 이항 연산자(OR)는 두 피연산자 사이의 비트끼리 비교합니다. 두 피연산자에 대응하는 비트 중 어느 하나라도 1 이면 결과 비트도 1이 됩니다.

2) 문법

result := expression1 OR expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY_BIT 타입의 표현식
<i>expression2</i>	ANY_BIT 타입의 표현식

OR 비트 연산 수행은 아래와 같습니다.

<i>expression1</i>	<i>expression2</i>	<i>result</i>
0	0	0
0	1	1
1	0	1
1	1	1

사용 예	설명
Result := 2#10010011 OR 2#00111101 ;	7번째 비트를 제외한 모든 비트 위치에서 적어도 하나의 1이 존재하므로 Result 값은 2#10111111이 됩니다.

12.3.9 XOR 연산자

1) 이항 연산자(XOR)는 두 피연산자 사이의 비트끼리 비교합니다. 두 피연산자에 대응하는 비트 중 어느 하나만 1 이면 (둘다 1 이면 안됨) 결과 비트도 1이 됩니다.

2) 문법

result := expression1 XOR expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY_BIT 타입의 표현식
<i>expression2</i>	ANY_BIT 타입의 표현식

XOR 비트 연산 수행은 아래와 같습니다.

<i>expression1</i>	<i>expression2</i>	<i>result</i>
0	0	0
0	1	1
1	0	1
1	1	0

사용 예	설명
Result := 2#10010011 XOR 2#00111101;	두 피연산자의 1번째 비트는 모두 1이므로 그 연산 결과의 1번째 비트는 0이 되는 것입니다 Result 값은 2#10101110 이 됩니다.

12.3.10 = 연산자

- 1) 비교 연산자(=)는 두 피연산자가 같은지 비교합니다.
- 2) 문법

result := expression1 = expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY 타입의 표현식
<i>expression2</i>	ANY 타입의 표현식

= 비트 연산 수행은 아래와 같습니다.

<i>expression1</i>	<i>expression2</i>	<i>result</i>
0	0	1
0	1	0
1	0	0
1	1	1

사용 예	설명
Val1 := 20; Val2 := 20 ; Result := Val1 = Val2 ;	두 피연산자 Val1과 Val2가 같은지 비교한 결과를 변수 Result에 대입합니다. 결과 값은 1이 됩니다.

12.3.11 ◇ 연산자

- 1) 비교 연산자(◇)는 두 피연산자가 같지 않은지 비교합니다.
- 2) 문법

result := expression1 ◇ expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY 타입의 표현식
<i>expression2</i>	ANY 타입의 표현식

◇ 비트 연산 수행은 아래와 같습니다.

<i>expression1</i>	<i>expression2</i>	<i>result</i>
0	0	0
0	1	1
1	0	1
1	1	0

사용 예	설명
Val1 := 20; Val2 := 20 ; Result := Val1 ◇ Val2 ;	두 피연산자 Val1과 Val2가 같지 않은지 비교한 결과를 변수 Result에 대입합니다. 결과 값은 0이 됩니다.

12.3.12 > 연산자

- 1) 관계 연산자(>)는 연산자 왼쪽의 피연산자가 오른쪽의 피연산자 보다 큰지 비교합니다.
- 2) 문법

result := expression1 > expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY 타입의 표현식
<i>expression2</i>	ANY 타입의 표현식

> 비트 연산 수행은 아래와 같습니다.

<i>expression1</i>	<i>expression2</i>	<i>result</i>
0	0	0
0	1	0
1	0	1
1	1	0

사용 예	설명
Val1 := 20; Val2 := 10 ; Result := Val1 > Val2 ;	피연산자 Val1이 피연산자 Val2보다 큰지 비교한 결과를 변수 Result에 대입합니다. 결과 값은 1이 됩니다.

12.3.13 < 연산자

- 1) 관계 연산자(<)는 < 연산자 앞의 피연산자가 연산자 뒤의 피연산자 보다 작은지 비교합니다.
- 2) 문법

result := expression1 < expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY 타입의 표현식
<i>expression2</i>	ANY 타입의 표현식

< 비트 연산 수행은 아래와 같습니다.

<i>expression1</i>	<i>expression2</i>	<i>result</i>
0	0	0
0	1	1
1	0	0
1	1	0

사용 예	설명
Val1 := 20; Val2 := 10 ; Result := Val1 < Val2 ;	피연산자 Val1이 피연산자 Val2보다 작은지 비교한 결과를 변수 Result에 대입하라고 지시합니다. 결과 값은 0이 됩니다.

12.3.14 >= 연산자

- 1) 관계 연산자(>=)는 앞의 피연산자가 뒤의 피연산자 보다 크거나 같은지 비교합니다.
- 2) 문법

result := expression1 >= expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY 타입의 표현식
<i>expression2</i>	ANY 타입의 표현식

>= 비트 연산 수행은 아래와 같습니다.

<i>expression1</i>	<i>expression2</i>	<i>result</i>
0	0	1
0	1	0
1	0	1
1	1	1

사용 예	설명
Val1 := 20; Val2 := 20 ; Result := Val1 >= Val2 ;	피연산자 Val1이 피연산자 Val2보다 크거나 같은지 비교한 결과를 변수 Result에 대입합니다. 결과 값은 1이 됩니다.

12.3.15 <= 연산자

- 1) 관계 연산자(<=)는 앞의 피연산자가 뒤의 피연산자 보다 작거나 같은지 비교합니다.
- 2) 문법

result := expression1 <= expression2

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression1</i>	ANY 타입의 표현식
<i>expression2</i>	ANY 타입의 표현식

<= 비트 연산 수행은 아래와 같습니다.

<i>expression1</i>	<i>expression2</i>	<i>result</i>
0	0	1
0	1	0
1	0	1
1	1	1

사용 예	설명
Val1 := 2; Val2 := 20 ; Result := Val1 <= Val2 ;	피연산자 Val1이 피연산자 Val2보다 작거나 같은지 비교한 결과를 변수 Result에 대입합니다. 결과 값은 1이 됩니다.

12.3.16 NOT 연산자

- 1) NOT 연산자는 비트 값을 반전합니다.
- 2) 문법

result := NOT expression

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression</i>	ANY_BIT 타입의 표현식

사용 예	설명
Val1 = 2#1100; Result := NOT Val1 ;	Val1의 값을 반전하여 Result값에 넣습니다. 결과 값은2#00110이 됩니다.

12.3.17 - 연산자

- 1) 연산자는 부호를 변경 시킵니다.
- 2) 문법

result := - expression

항목	설명
<i>result</i>	변수 또는 직접변수
<i>expression</i>	ANY_NUM 타입의 표현식

사용 예	설명
Val1 = 10; Result := - Val1 ;	Val1의 부호를 반전하여 Result값에 넣습니다. 결과 값은 -10이 됩니다.

12.4 명령문

- 1) ST 언어의 명령문들이 <표 2>에 요약되어 있습니다.
- 2) 명령문들은 세미 콜론(;)에 의해 마칩니다.

12.4.1 할당문

- 1) 할당문은 왼쪽에 변수, 그 뒤를 따르는 할당문 연산자(:=), 마지막으로 연산할 표현식으로 구성됩니다.
예) A := B + C ;
- 2) 할당문은 평선 이름을 할당문 연산자 왼쪽에 위치 시킴으로써 평선의 리턴값을 대입하는데도 사용합니다.

12.4.2 선택문

- 1) 선택문은 IF 문과 CASE 문 두가지 종류가 있습니다.
- 2) 선택문은 특정 조건에 기초하여 수행하는 동안 선택문을 구성하는 명령문들 중 하나(혹은 그룹)를 선택합니다.
 - (1) IF 문
 - (a) 관련 부울 표현식이 1의 값(참)으로 결과가 나오면 명령문 그룹을 실행합니다.
 - (b) 조건이 거짓이면 어떠한 명령문도 실행되지 않습니다. 그러나 ELSE가 있는 경우 ELSE를 따르는 명령문 그룹이 실행됩니다. 만약 ELSIF 관련 조건이 참인 경우 ELSIF를 따르는 명령문 그룹이 실행됩니다.

(2) CASE 문

- (a) INT 형의 변수(“선택자”)를 계산하는 표현식과 명령문 그룹의 리스트로 구성됩니다.
- (b) 각 그룹의 라벨은 하나 이상의 정수와 정수 값의 범위로 설정할 수 있습니다.
- (c) 선택자의 계산 값을 포함하는 범위 내에 있는 명령문의 그룹이 수행되며, 선택자의 어떠한 값도 CASE 문의 각 경우에 해당되지 않으면 ELSE 를 뒤따르는 명령문 그룹이 수행됩니다. 만일 ELSE 가 없다면 어떠한 명령문도 수행되지 않습니다.

12.4.3 반복문

- 1) 반복문은 FOR 문, WHILE 문 그리고 REPEAT 문 세가지 종류가 있습니다.
- 2) 반복문은 관련 명령문의 그룹이 반복적으로 수행되는 것을 말합니다.

(1) FOR 문

- (a) 반복 횟수가 미리 결정되어 있는 경우에 사용됩니다.
- (b) FOR 문에서는 명령문 시퀀스가 END_FOR 까지 반복적으로 수행되며 값의 진행은 FOR 루프의 제어 변수에 지정됩니다.
- (c) 제어 변수, 초기값과 최종 값은 같은 정수형(SINT, INT, DINT)의 표현식으로 나타내며 반복되는 문장에 의해 바뀌지 않습니다. 종료 조건에 대한 검사는 각 반복의 시작시 행해져 초기값이 최종값을 초과한다면 명령문 시퀀스가 더 이상 수행되지 않습니다.

(2) WHILE 문과 REPEAT 문

- (a) WHILE 문은 관련 부울 표현식이 거짓일 때까지 END_WHILE 까지의 명령문 시퀀스를 반복적으로 수행됩니다.
- (b) REPEAT 문에서는 관련 부울 조건이 참일 때까지 UNTIL 까지의 명령문 시퀀스를 반복적으로 (최소한 한번은) 수행됩니다.
- (c) WHILE 문과 REPEAT 문은 외부적으로 결정되는 종료 조건을 가진 “대기 루프(wait loop)”와 같은 프로세스간 동기화 하는데 사용하지 않습니다.
- (d) EXIT 문은 종료 조건이 만족되기 전에 반복을 중단하는데 사용됩니다. EXIT 문이 중첩 반복 구조내에 사용될 때 해당 EXIT 는 그 EXIT 가 위치한 가장 안쪽 루프에 적용됩니다. 따라서, 제어는 EXIT 문의 뒤에 위치하는 첫번째 루프 종료자(END_FOR, END_WHILE, END_REPEAT) 이후의 명령문으로 전달됩니다.
- (e) WHILE 문과 REPEAT 문은 루프 종료 조건의 만족하거나 EXIT 문의 수행이 보장될 수 없는 알고리즘에 사용되면 에러입니다.

번호	명령문 형태	예
1	할당문	A:=B; CV:= CV+1; C:=SIN(X);
2	평선 블록 호출 평선 블록 출력 사용	CMD_TMR(IN:=%IX0.0, PT:= T#300ms); A:=CMD_TMR.Q;
3	리턴	RETURN;
4	IF 문	D:=B*B -4*A*C; IF D<1.0 THEN NROOTS :=0; ELSIF D= 0.0 THEN

번호	명령문 형태	예
		<pre> NROOTS := 1; X1:= -B/(2.0*A); ELSE X1:= (-B+SQRT(D))/(2.0*A); X2:= (-B-SQRT(D))/(2.0*A); END_IF; </pre>
5	CASE 문	<pre> TW := WORD_BCD_TO_INT(THUMBWHEEL); TW_ERROR := 0; CASE TW OF 1,5: DISPLAY := OVEN_TEMP; 2: DISPLAY := MOTOR_SPEED; 3: DISPLAY := GROSS - TARE; 4, 6..10: DISPLAY := ADD(TW , 4); ELSE DISPLAY := 0 ; TW_ERROR := 1; END_CASE; %MW100 := INT_TO_BCD_WORD(DISPLAY); </pre>
6	FOR 문	<pre> J := 101; FOR I := 1 TO 100 BY 2 DO IF WORDS[I] = 'KEY' THEN J := I; EXIT; END_IF; END_FOR ; </pre>
7	WHILE 문	<pre> J := 1; WHILE J <= 100 & WORDS[J] <> 'KEY' DO J := J+2; END_WHILE; </pre>
8	REPEAT 문	<pre> J := -1; REPEAT J := J+2; UNTIL J = 101 OR WORDS[J] = 'KEY' END_REPEAT ; </pre>
9	EXIT 문	EXIT;
10	널/공백 명령문	;

EXIT 문은 지원되는 모든 반복문(FOR, WHILE, REPEAT)에 사용됩니다.

<표 3> ST 언어 명령문들

12.4.4 IF 문

- 1) IF 문은 프로그램이 한 가지 이상 선택해야 하는 경우에 갈림길을 제공하기 때문에 분기문이라 불립니다.
- 2) 문법

```
IF condition THEN statements [ELSE elstatements ] END_IF
```

또는 아래와 같이 사용할 수 있습니다.

```
IF condition THEN
    statements
[ELSIF condition-n THEN
    elseifstatements] . . .
[ELSE
    elstatements]
END_IF
```

항목	설명
<i>condition</i>	<i>condition</i> 이 TRUE 이면 THEN 이하의 <i>statements</i> 를 수행합니다. FALSE 인 경우는 ELSIF 또는 ELSE 로 분기됩니다.
<i>statements</i>	만약 <i>condition</i> 이 TRUE 이면 하나 이상의 상태문을 수행합니다..
<i>condition-n</i>	<i>condition</i> 을 N개 사용할 수 있습니다.
<i>elseifstatements</i>	만약 <i>condition-n</i> 이 TRUE이면 하나 이상의 상태문을 수행 합니다.
<i>elstatements</i>	만약에 이전 <i>condition</i> 또는 <i>condition-n</i> 이 FALSE 인 경우 하나 이상의 상태문을 수행합니다.

사용 예	설명
<pre>IF Val1 <= 10 THEN Result := 10; END_IF;</pre>	조건 Val1 <= 10 이 참이면 Result에 10의 값을 할당합니다.
<pre>IF Val1 <= 10 THEN Result := 10; ELSE Result := 20; END_IF;</pre>	조건 Val1 <= 10 이 참이면 Result에 10의 값을 할당합니다. 만약 FALSE 이면 Result에 20의 값을 할당합니다.
<pre>IF Val1 <= 10 THEN Result := 10; ELSIF Val1 <= 20 THEN Result := 20; ELSE Result := 30; END_IF;</pre>	조건 Val1 <= 10 이 참이면 Result에 10의 값을 할당합니다. 만약 FALSE 이면 ELSIF의 조건을 수행합니다. 두번째 조건 Val1 <= 20 이 참이면 Result에 20의 값을 할당합니다. 만약 FALSE 이면 ELSE 이하의 상태문을 수행합니다. 즉, Result에 30의 값을 할당합니다.

12.4.5 CASE 문

- 1) 프로그램의 제어가 CASE 다음에 오는 표현식의 값으로 분기합니다. 표현식은 모두 정수값(INT 형)이어야 합니다. 표현식의 값이 케이스 목록(case list) 범위에 포함되지 않는 경우 ELSE 이후에 있는 상태문이 실행됩니다. 만약 ELSE 가 없는 경우 케이스 목록 내의 어떠한 상태문 리스트(statement list)도 실행하지 않습니다. 다른 분기 명령이 없다면 나머지 모든 문을 실행하면서 프로그램의 흐름을 진행합니다.
- 2) 문법

```

CASE expression OF
  case_list : statement_list
{ case_list : statement_list}
[ELSE
  statement_list]
END_CASE
    
```

항목	설명
<i>expression</i>	INT형 상태문만 가능합니다.
<i>case_list</i>	<i>case_list_element</i> {',' <i>case_list_element</i> } 위와 같이 조건이 여러 개 올 수 있습니다.
<i>case_list_element</i>	<i>subrange</i> 또는 <i>signed_integer</i> 만 가능합니다.
<i>subrange</i>	<i>signed_integer</i> .. <i>signed_integer</i> 형태입니다.
<i>statement_list</i>	하나 이상의 상태문을 수행합니다.

사용 예	설명
<pre> CASE Val1 OF 1 : Result := 10 ; 2..5 : Result := 20 ; 7, 10 : Result := 30 ; ELSE Result := 40 ; END_CASE ; </pre>	<p>만약 Val1의 값이 정수값 1이면 Result 변수에 10의 값을 할당합니다. 만약 2와 5의 범위의 값이면 Result 변수에 20의 값을 할당합니다. 만약 Val1의 값이 정수값 7 또는 10 이면 Result 변수에 30의 값을 할당합니다. 이외의 다른 값일 경우에는 Result 변수에 40의 값을 할당합니다.</p>

12.4.6 FOR 문

- 1) FOR 문은 순환 과정을 처리하기 위해서 구분자로 분리된 세 개의 제어문을 사용합니다. FOR 문의 구성 요소 중 먼저 초기화하는 수식이 실행됩니다. 만약 TO 수식이 참이면(현재 counter 의 값이 end 값보다 작은 경우) 루프는 한번 실행됩니다. 그 다음 BY 수식의 값만큼 counter 의 값이 갱신되고 조건문을 다시 검사합니다. FOR 문은 선 조건 검사 루프입니다. 즉, 루프를 통과하기 전에 순환 여부를 검사합니다. 따라서 루프를 전혀 수행하지 않을 수도 있습니다.
- 2) 문법

```
FOR counter := start TO end [BY step] DO
    statements
END_FOR
```

항목	설명
<i>counter</i>	정수형(SINT, INT, DINT) 변수입니다. start, end, step 은 모두 동일한 타입이어야 합니다.
<i>start</i>	<i>counter</i> 의 초기값
<i>end</i>	<i>counter</i> 의 마지막 값
<i>step</i>	<i>counter</i> 변수가 루프 수행할 때 마다 증가되는 수를 표시합니다. 만약 사용하지 않았다면 기본적으로 1이 증가됩니다.
<i>statements</i>	세개의 제어문에 의하여 설정된 수만큼 수행되는 하나 이상의 상태문입니다.

사용 예	설명
SUM := 0; FOR counter := 0 TO 10 DO SUM := SUM + 1; END_FOR ;	counter 변수가 0에서 10까지 1씩 자동 증가 시킵니다. SUM 변수에 1의 값을 계속 더하는 프로그램입니다. 최종 SUM의 값은 11 입니다.
SUM := 0; FOR counter = 0 TO 10 BY 2 DO SUM := SUM + 1; END_FOR ;	counter 변수가 0에서 10까지 2씩 증가 시킵니다. SUM 변수에 1의 값을 계속 더하는 프로그램입니다. 최종 SUM의 값은 6 입니다.

알아두기
<ol style="list-style-type: none"> 스캔 타입이 오래 걸려 위치독이 걸릴 수 있습니다. BY 부분은 생략 가능합니다. 생략된 경우 루프 카운터는 기본적으로 1씩 증가합니다. 초기값이 마지막 값보다 큰 경우 FOR 문을 수행하지 않습니다.

12.4.7 WHILE 문

- 1) WHILE 문은 조건 수식이 거짓이나 0 이 될 때까지 반복하는 순환문을 만듭니다. WHILE 문은 선 조건 검사 루프입니다. 즉, 루프를 통과하기 전에 순환 여부를 검사합니다. 따라서 조건이 만족하지 않는 경우 WHILE 문 내의 상태문을 전혀 수행하지 않을 수 있습니다.

- 2) 문법

```

WHILE condition DO
    statements
END_WHILE
    
```

항목	설명
<i>condition</i>	<i>condition</i> 이 TRUE 이면 DO 이하의 <i>statements</i> 를 수행합니다. FALSE 인 경우는 순환문을 빠져 나갑니다.
<i>statements</i>	<i>condition</i> 이 TRUE 이면 하나 이상 상태문이 처리됩니다.

사용 예	설명
<pre> Counter := 0 WHILE Counter < 20 DO Counter := Counter + 1; END_WHILE ; </pre>	Counter 변수가 20보다 작다는 조건이 맞으면 상태문을 수행 합니다. Counter 변수가 20이 되면 조건이 FALSE 가 되어 순환문을 빠져 나갑니다.

알아두기
WHILE 문은 조건 수식이 거짓이나 0 이 되지 않는 경우 무한 루프에 빠질 수 있습니다. 이 경우 스캔 타임이 오래 걸려 스캔 위치독이 걸릴 수 있습니다. 따라서 조건 수식이 항상 TRUE 가 되지 않도록 주의하시기 바랍니다.

12.4.8 REPEAT 문

- 1) REPEAT 문은 조건식이 TRUE 가 될 때까지 반복하는 순환문을 만듭니다. REPEAT 문(후 조건 검사 루프)은 루프 통과 후에 루프 반복 여부를 검사해서 결정합니다. 따라서 루프는 적어도 한번은 실행합니다.

- 2) 문법

```

REPEAT
    statements
UNTIL condition
END_REPEAT
    
```

항목	설명
----	----

<i>condition</i>	<i>condition</i> 이 FALSE 면 반복 수행하고 TRUE 면 빠져 나갑니다.
<i>statements</i>	조건이 TRUE 일 때까지 반복 수행합니다.

사용 예	설명
<pre>Counter := 0; REPEAT Counter := Counter + 1; UNTIL Counter > 20 END_REPEAT ;</pre>	<p>우선 Counter 변수가 1 증가됩니다. Counter 변수가 20 보다 크다는 조건을 만족하면 순환문을 빠져 나가고 아니면 상태문을 수행합니다. Counter 변수가 21이 되면 조건이 TRUE 가 되어 순환문을 빠져 나갑니다.</p>

알아두기

REPEAT 문은 조건 수식이 거짓이나 0 이 되지 않는 경우 무한 루프에 빠질 수 있습니다. 이 경우 스캔 타임이 오래 걸려 스캔 위치독이 걸릴 수 있습니다. 따라서 조건 수식이 항상 TRUE 가 되지 않도록 주의하시기 바랍니다.

12.4.9 EXIT 문

- EXIT 문은 반복문(WHILE, FOR, REPEAT)에서 빠져 나가는데 사용됩니다.
- EXIT 문을 반복문 밖에서 사용하는 경우 에러가 발생합니다.
- 문법

EXIT

사용 예	설명
<pre>SUM := 0; FOR Counter := 0 TO 10 DO SUM := SUM + 1; EXIT; END_FOR ;</pre>	<p>Counter 변수가 0 에서 10 까지 1씩 자동 증가하는 FOR 문 입니다. 그러나 상태문에 EXIT를 사용하여 바로 빠져 나옵니다. Counter 변수는 0의 되고, SUM 변수는 1의 됩니다.</p>
<pre>Counter := 0; WHILE Counter < 20 DO Counter := Counter + 1 ; IF Counter = 10 THEN EXIT; END_IF; END_WHILE ;</pre>	<p>Counter 변수가 20보다 작다는 조건이 맞으면 상태문을 수행 합니다. Counter 변수가 20이 되면 조건이 FALSE가 되어 순환문을 빠져 나갈 수 있습니다. 그러나 상태문에 IF문과 EXIT문을 사용하여 Counter 변수가 10일 경우 반복문을 빠져 나갑니다.</p>
<pre>Counter := 0; REPEAT Counter := Counter + 1 ; IF Counter = 10 THEN EXIT; END_IF; UNTIL Counter > 20 END_REPEAT ;</pre>	<p>우선 Counter 변수가 1만큼 증가합니다. Counter 변수가 20보다 크다는 조건이 맞으면 순환문을 빠져나가고 아니면 상태문을 수행합니다. 그러나 상태문에 IF문과 EXIT문을 사용하여 Counter 변수가 10 일 경우 반복문을 빠져 나갑니다.</p>

12.5 평선 및 평선 블록

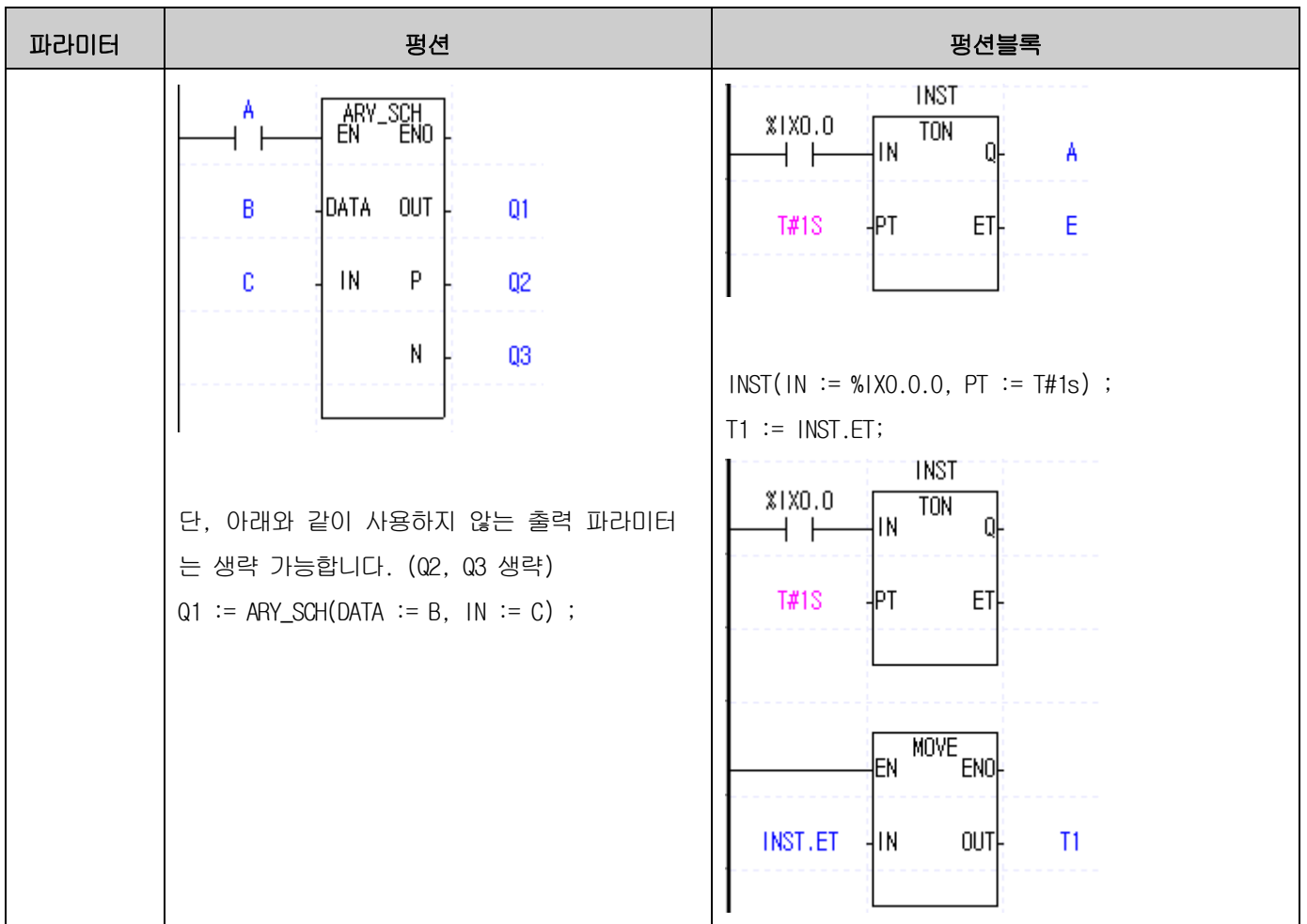
12.5.1 사용 방법

1) 평선 및 평선 블록 입력 방법은 정형화 형태, 비정형화 형태 2 가지가 있습니다. 상황에 따라 어느 형태도 사용 가능합니다.

(1) 정형화 형태:

평선 및 평선블록의 입력, 출력 파라미터 이름을 표기하는 형태입니다.

파라미터	평선	평선블록
공통	<p>파라미터 순서는 임의의 순서로 사용 가능.</p> <p>Q1 := LIMIT(MN := B, MX := 20, IN := 10) ;</p> <p>Q1 := LIMIT(MX := 20, MN := B, IN := 10) ;</p> <p>EN, ENO은 사용 또는 생략 가능</p> <p>Q1 := LIMIT(EN := A, MN := B, MX := 20, IN := 10, ENO => Q2) ;</p>	<p>파라미터 순서는 임의의 순서로 사용 가능.</p> <p>INST(IN := %IX0.0.0, PT := T#1s, Q => A, ET => E) ;</p> <p>INST(PT := T#1s, IN := %IX0.0.0, Q => A, ET => E) ;</p>
입력	<p>입력, 입출력 파라미터 할당에 := 기호 사용.</p> <p>C := LIMIT(MN := B, MX := 20, IN := 10) ;</p>	<p>입력, 입출력 파라미터 할당에 := 기호 사용.</p> <p>INST(IN := %IX0.0, PT := T#1s, Q => A, ET => B) ;</p>
출력	<p>출력 파라미터 이름이 OUT 또는 Y(사용자 정의 평선 이름)일 경우 리턴값 할당.</p> <p>나머지 출력 파라미터 할당은 => 기호 사용.</p> <p>Q1 := ARY_SCH(DATA := B, IN := C, P => Q2, N => Q3) ;</p>	<p>모든 출력 파라미터 할당에 => 기호 사용</p> <p>출력 파라미터 할당은 생략 가능.</p> <p>INST(IN := %IX0.0, PT := T#1s, Q => A, ET => E) ;</p>



알아두기

평선 블록은 인스턴스 이름으로 사용합니다. 즉, 평선 블록을 변수처럼 선언하고 이 변수 명(인스턴스 명)을 사용하여야 합니다.

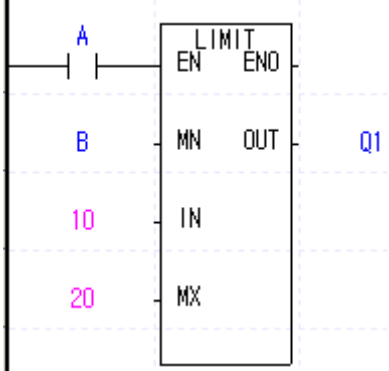
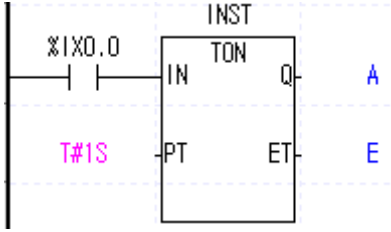
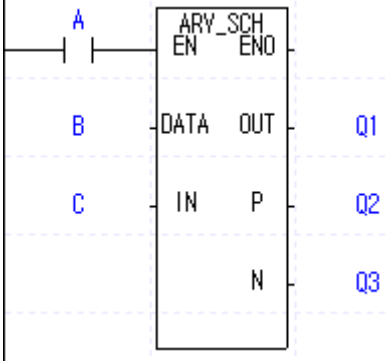
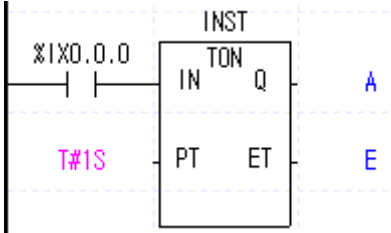
예) 타이머 사용

	변수 종류	변수	타입
1	VAR	INST_TON1	TON

```
INST_TON1(IN := TRUE, PT := T#100MS, Q => Q_OUT, ET => ET_OUT) ;
```

(2) 비 정형화된 형태

평선 및 평선블록의 입력, 출력 파라미터 이름을 생략하는 형태입니다.

파라미터	평선	평선블록
공통	<p>모든 파라미터 순서는 변경 불가. 모든 파라미터는 생략 불가 Q1 := LIMIT(B, 20, 10) ;</p>  <p>EN, ENO는 사용 불가.</p>	<p>모든 파라미터 순서는 변경 불가. 모든 파라미터는 생략 불가. INST(%IX0.0, T#1s, A, E) ;</p> 
입력	<p>입력 파라미터 순서는 변경 불가. C := LIMIT(B, 20, IN := 10) ;</p>	<p>입력 파라미터 순서는 변경 불가. INST(%IX0.0, T#1s, A, E) ;</p>
출력	<p>출력 파라미터 이름이 OUT 또는 Y(사용자 정의 평선은 평선 이름)일 경우 리턴값 할당. 나머지 출력 파라미터 할당은 순서대로 입력. Q1 := ARY_SCH(B, C, Q2, Q3) ;</p> 	<p>모든 출력 파라미터 할당은 순서대로 입력. INST(%IX0.0.0, T#1s, A, E) ;</p> 
알아두기		

파라미터 타입이 가변인 평선은 입력 파라미터 타입이 결정 되어야 합니다.

사용 예	설명
INT1 := ADD(1, 2, 3);	평선 타입 결정 중에 에러가 발생합니다.

정상 동작 하기 위해서는 아래 중에 한가지 방법으로 입력하여야 합니다.

사용 예	설명
INT1 := ADD(INT#1, 2, 3);	상수에 타입을 설정할 수 있습니다.
INT1 := ADD(B, 2, 3);	변수(B)를 사용할 수 있습니다.
INT1 := ADD_INT(1, 2, 3);	타입이 설정된 평선을 사용할 수 있습니다.

알아두기

1. 입력 파라미터 EN 은 평선을 실행하기 위한 조건입니다. 다음과 같이 EN 을 사용할 경우 A 값이 1 일 경우에만 LIMIT 평선이 실행됩니다.
 OUT := LIMIT(EN := A, MX := 20, MN := B, IN := 10) ;
2. ENO 파라미터는 평선이 에러 없이 실행될 경우에 1 이 됩니다. ST 에서는 사용할 수 없고 LD 에서만 사용 가능합니다.

알아두기

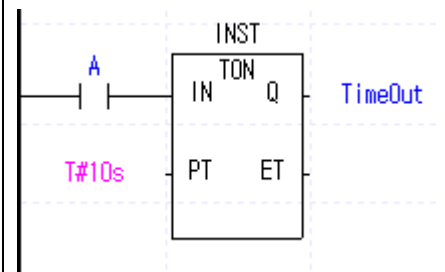
1. ST 는 확장 명령어(BREAK, CALL, END, FOR, INIT_DONE, JMP, NEXT, RET, SBRT)를 지원하지 않습니다.
2. 연산자 명과 동일한 이름의 평선은 사용할 수 없습니다.(OR, XOR, AND, MOD, NOT)

12.5.2 사용 예

1) 평선

LD 사용 예	ST 사용 예
	<p>1) 정형화된 형태 EN 사용 <code>OutValue := ADD(EN := A, IN1 := Value1, IN2 := Value2);</code> EN 사용 안 함 <code>OutValue := ADD(IN1 := Value1, IN2 := Value2);</code></p> <p>2) 비정형화된 형태 <code>OutValue := ADD(Value1, Value2);</code> EN, ENO를 사용할 수 없습니다.</p>

2) 평선 블록

LD 사용 예	ST 사용 예
	<p>1) 정형화된 형태 <code>INST(IN := A, PT := T#10S, Q => TimeOut);</code></p> <p>2) 비정형화된 형태 <code>INST(A, T#10S, TimeOut, TimeValue);</code> 출력 변수를 생략할 수 없습니다. 따라서 출력 파라미터 ET에 해당하는 변수를 연결해야 합니다. (TimeValue)</p>

3) 응용

LD 사용 예	ST 사용 예
<p>The diagram shows two function blocks. The first block, labeled 'INST1', has three inputs: 'CD' (set to '_T1S'), 'PV' (set to '10'), and 'RST' (set to '리셋'). It has one output 'Q' (set to '완료'). The second block is an 'LT' (Limit) block with two inputs: 'IN1' (set to '현재값') and 'IN2' (set to '5'). It has one output 'OUT' (set to '미달'). A normally open contact labeled '완료' is connected to a coil labeled '%QX0.0'.</p>	<pre> INST1(CD := _T1S, PV := 10, RST := 리셋, Q => 완료, CV => 현재값); %QX0.0 := 완료; 미달 := LT(IN1 := 현재값, IN2 := 5); </pre>

부록 1장 수치 체계 및 데이터 구조

부1.1. 수치(데이터)의 표현

모션 제어 모듈에서는 모든 정보를 On 과 Off, 또는 ‘1’ 과 ‘0’ 의 상태로 기억하고 처리합니다. 따라서 수치 연산도 1 과 0 으로 처리된 수치, 즉 2 진수 (Binary number: BIN)로 처리합니다. 한편, 일상 생활에서는 10 진수가 알기 쉽고 가장 널리 사용되고 있습니다. 그래서 PLC 에 수치를 Write 할 경우, 또는 PLC 의 수치정보를 Read 할 경우에는 10 진수에서 16 진수로, 16 진수에서 10 진수로 변환이 필요합니다. 여기에서는 10 진수와 2 진수, 16 진수, 2 진화 10 진수(BCD)의 표현과 그 상호관계에 대해 설명합니다.

1) 10 진수(Decimal)

10 진수란 “0~9 의 종류의 기호를 사용하여 순서와 크기(량)를 표현하는 수” 를 말합니다.

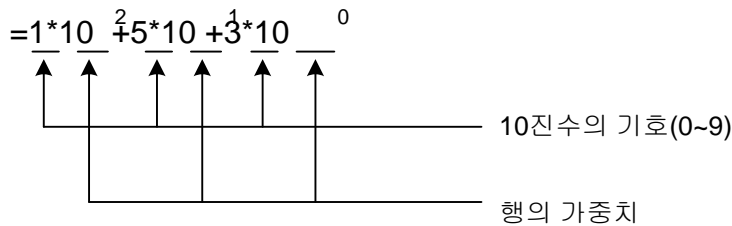
그리고 0, 1, 2, 3, 4, ..., 9 다음에 ‘10’ 으로 자리올림하고 계속 진행됩니다.

예를 들면, 10 진수 153 을 행과 “행의 가중치” 란 측면에서 보면 아래와 같습니다.

$$153=100+50+3$$

$$=1*100+5*10+3*1$$

$$=1*10^2+5*10^1+3*10^0$$



2) 2 진수 (Binary …… Bin)

2 진수란 “0 과 1 의 두 종류 기호를 사용하여 순서와 크기를 나타내는 수” 를 말합니다. 그래서 0, 1 다음에 ‘10’ 으로 자리올림을 하고, 계속 진행됩니다.

즉, 0,1 의 한 자리 수를 비트라고 합니다.

2 진수	10 진수
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
.....

예를 들면 다음의 2진수는 10진수로 얼마나 되는지 생각해 봅시다.

“10011101”

10진수에서 행 번호와 행의 가중치를 고려하였듯이 우측부터 비트번호와 비트가중치를 붙여 봅시다.

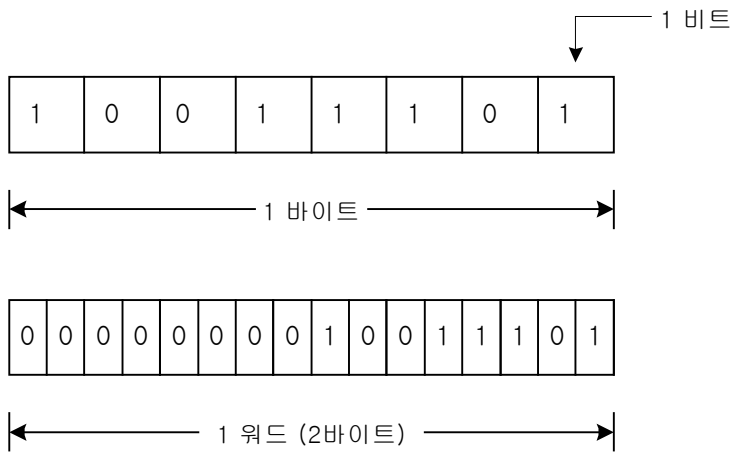
7	6	5	4	3	2	1	0	←	비트번호 2진수
1	0	0	1	1	1	0	1		
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		
128	64	32	16	8	4	2	1		비트의 가중치

10진수와 같이 각 비트의 코드의 가중치의 곱의 합을 생각해 봅시다.

$$\begin{aligned}
 &= 1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\
 &= 128 + 16 + 8 + 4 + 1 \\
 &= 157
 \end{aligned}$$

즉, 2진수는 “코드가 1인, 비트의 가중치를 가산한 것” 이 10진수로 되는 것입니다.

일반적으로 8비트를 1바이트, 16비트(2바이트)를 1워드라 말합니다.

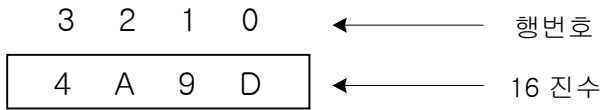
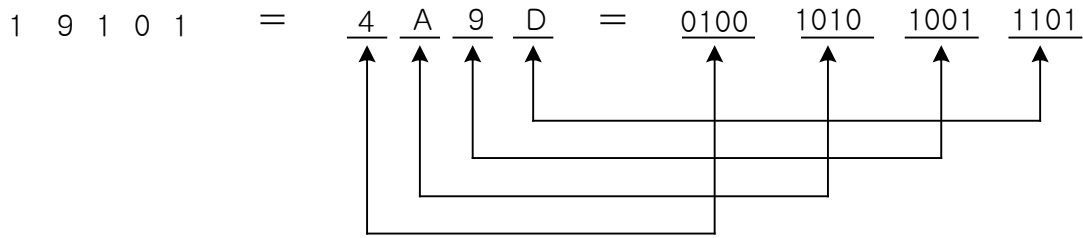


3) 16 진수 (Hexadecimal HEX)

16 진수도 10 진수, 2 진수와 동일하게 생각하여 “0 ~ 9, A ~ F 의 종류의 기호를 사용하여 순서와 크기를 나타내는 수” 를 말합니다.

그리고 0, 1, 2,D, E, F 다음에 ‘10’ 으로 자리올림을 하고 계속 진행됩니다.

10 진수	16 진수	2 진수
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000
17	11	10001
18	12	10010



$$\begin{aligned}
 &= (4) \times 16^3 + (A) \times 16^2 + (9) \times 16^1 + (D) \times 16^0 \\
 &= 4 \times 4096 + 10 \times 256 + 9 \times 16 + 13 \times 1 \\
 &= 19101
 \end{aligned}$$

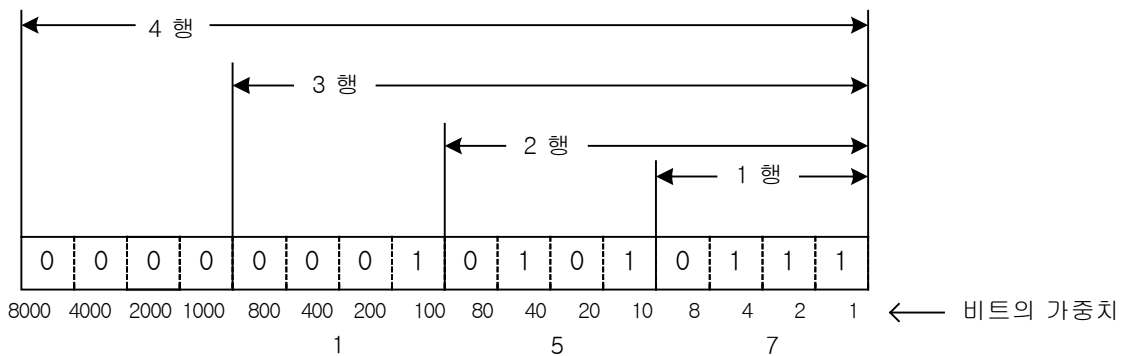
16 진수의 한자리는 2 진수의 4 비트로 대응됩니다.

4) 2 진화 10 진수 (Binary Coded Decimal BCD)

2 진화 10 진수는 “10 진수의 각행의 숫자를 2 진수로 나타낸 수” 를 말합니다.

따라서, 2 진화 10 진수는 10 진수의 0 ~ 9,999 (4 행의 최대치)를 16 비트로 나타냅니다.

예를 들면, 10 진수의 157 는 다음과 같이 나타낼 수 있으며, 각 비트의 가중치는 다음과 같습니다.



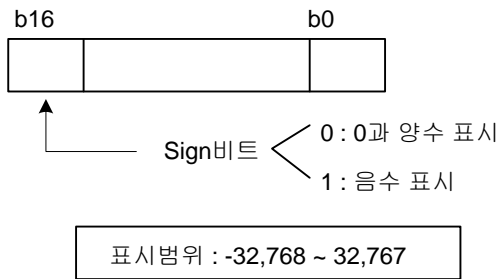
5) 수치 체계표

2 진화 10 진수 (Binary coded Decimal) BCD		2 진수 (Binary) BIN		10 진수 (Decimal)	16 진수 (Hexadecimal) H
0000000	0000000	0000000	0000000	0	0000
0000000	0000001	0000000	0000001	1	0001
0000000	0000010	0000000	0000010	2	0002
0000000	0000011	0000000	0000011	3	0003
0000000	0000100	0000000	0000100	4	0004
0000000	0000101	0000000	0000101	5	0005
0000000	0000110	0000000	0000110	6	0006
0000000	0000111	0000000	0000111	7	0007
0000000	0001000	0000000	0001000	8	0008
0000000	0001001	0000000	0001001	9	0009
0000000	0001000	0000000	0001010	10	000A
0000000	0001001	0000000	0001011	11	000B
0000000	0001010	0000000	0001100	12	000C
0000000	0001011	0000000	0001101	13	000D
0000000	0001010	0000000	0001110	14	000E
0000000	0001011	0000000	0001111	15	000F
0000000	0000110	0000000	0001000	16	0010
0000000	0000111	0000000	0001001	17	0011
0000000	0000100	0000000	00010010	18	0012
0000000	0000101	0000000	00010011	19	0013
0000000	0010000	0000000	00010100	20	0014
0000000	0010001	0000000	00010101	21	0015
0000000	0010010	0000000	00010110	22	0016
0000000	0010011	0000000	00010111	23	0017
0000001	0000000	0000000	01100100	100	0064
0000001	00100111	0000000	01111111	127	007F
0000010	01010101	0000000	11111111	255	00FF
0001000	0000000	0000000	11100000	1,000	03E8
0010000	01000111	0000000	11111111	2,047	07FF
0100000	10010101	0000000	11111111	4,095	0FFF
10011001	10011001	0000111	00001111	9,999	270F
		00100111	00010000	10,000	2710
		01111111	11111111	32,767	7FFF

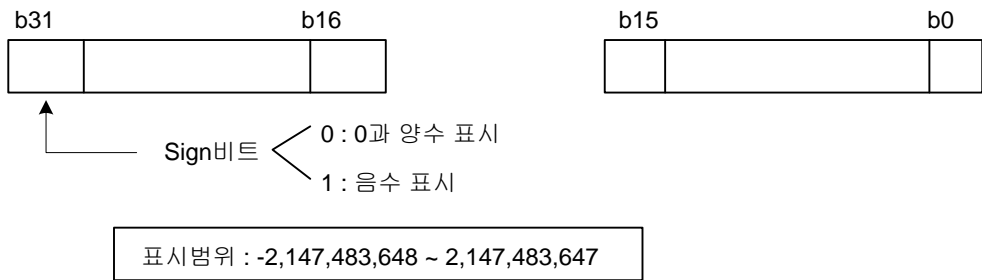
부1.2. 정수표현

모션 제어 모듈의 명령어에서는 음수체계연산(Signed)을 기본으로 합니다.
 이때 정수표시는 최상위 비트(MSB)가 0 이 되면 양수를 나타내고 1 이면 음수로 나타나게 됩니다.
 음수, 양수를 표시하는 최상위 비트를 Sign 비트라고 합니다.
 16 비트, 32 비트에서는 MSB 의 위치가 다르기 때문에 Sign 비트 위치에 주의해야 합니다.

★ 16 비트 일 경우



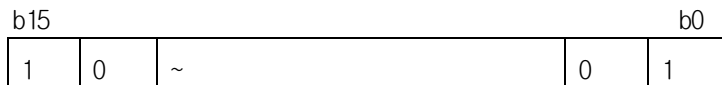
★ 32 비트 일 경우



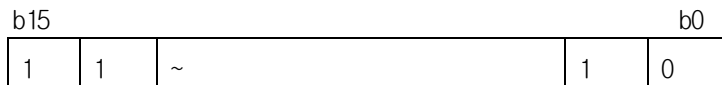
부1.3. 음수의 표현

예) - 0001 을 표기하는 방법

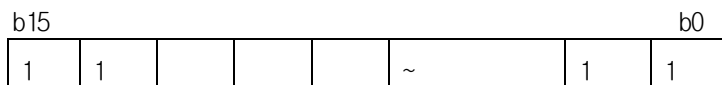
(1) 음수번호를 뺀 0001 을 표기한다. (b15 = 1)



(2) (1) 의 결과를 반전시킨다. (b15 = 제외)



(3) (2) 의 결과에 +1을 한다.



-0001 = 16#FFFF

부록 2 장 플래그 및 파라미터 일람(XGF-M32E)

부2.1. 시스템 플래그

모션제어 모듈의 동작 및 상태, 정보 등을 표시하는 플래그입니다.

변수	타입	메모리할당	설명
_RUN	BOOL	%FX0	RUN
_STOP	BOOL	%FX1	STOP
_ERROR	BOOL	%FX2	ERROR
_INIT_RUN	BOOL	%FX4	초기화 태스크 수행 중
_RUN_EDIT_ST	BOOL	%FX5	런중 수정 중 프로그램 다운로드 중
_RUN_EDIT_CHK	BOOL	%FX6	런중 수정 중 내부 처리 중
_RUN_EDIT_DONE	BOOL	%FX7	런중 수정 완료
_RUN_EDIT_NG	BOOL	%FX8	런중 수정 비정상 완료
_FLASH_BUSY	BOOL	%FX9	FLASH 쓰기 중
_FLASH_FAIL	BOOL	%FX10	FLASH 쓰기 실패
_FORCE_IN	BOOL	%FX11	강제입력
_FORCE_OUT	BOOL	%FX12	강제출력
_PB1	BOOL	%FX14	프로그램 코드 1
_PB2	BOOL	%FX15	프로그램 코드 2
_BPRM_ER	BOOL	%FX35	기본 파라미터 이상
_COMPRM_ER	BOOL	%FX36	공통 파라미터 이상
_AXISPRM_ER	BOOL	%FX37	축 파라미터 이상
_AXESPRM_ER	BOOL	%FX38	축 그룹 파라미터 이상
_COORDPARAM_ER	BOOL	%FX46	축 그룹 파라미터 - 좌표계 설정 이상
_NETPRM_ER	BOOL	%FX39	네트워크 파라미터 이상
_SDPRM_ER	BOOL	%FX40	공유변수 파라미터 이상
_PGM_ER	BOOL	%FX41	프로그램 에러
_CODE_ER	BOOL	%FX42	프로그램 코드 에러
_SWDT_ER	BOOL	%FX45	모듈 비정상 종료
_BACKUPDATA_ER	BOOL	%FX63	백업 데이터 이상
_AB_SD_ER	BOOL	%FX64	운전 이상 정지
_TASK_ER	BOOL	%FX65	태스크 충돌
_PTASK_CYCLE_ER	BOOL	%FX67	주태스크 주기 오류
_CTASK_CYCLE_ER	BOOL	%FX68	주기태스크 주기 오류
_T20MS	BOOL	%FX96	20ms 주기의 CLOCK
_T100MS	BOOL	%FX97	100ms 주기의 CLOCK
_T200MS	BOOL	%FX98	200ms 주기의 CLOCK

변수	타입	메모리할당	설명
_T1S	BOOL	%FX99	1s 주기의 CLOCK
_T2S	BOOL	%FX100	2s 주기의 CLOCK
_T10S	BOOL	%FX101	10s 주기의 CLOCK
_T20S	BOOL	%FX102	20s 주기의 CLOCK
_T60S	BOOL	%FX103	60s 주기의 CLOCK
_ON	BOOL	%FX104	상시 On
_OFF	BOOL	%FX105	상시 Off
_10N	BOOL	%FX106	첫 스캔 On
_10FF	BOOL	%FX107	첫 스캔 Off
_STOG	BOOL	%FX108	스캔 반전 (scan toggle)
_ERR	BOOL	%FX112	연산 에러 플래그
_LER	BOOL	%FX113	연산 에러 래치 플래그
_ARY_IDX_ERR	BOOL	%FX114	배열 인덱스 범위 초과 에러 플래그
_ARY_IDX_LER	BOOL	%FX115	배열 인덱스 범위 초과 래치 에러 플래그
_OS_VER	DWORD	%FD4	OS 버전 표시
_OS_DATE	DWORD	%FD5	OS 날짜 표시
_OS_VER_PATCH	DWORD	%FD6	OS 패치 버전
_SCAN_MAX	WORD	%FW14	주 태스크 최대 스캔시간(단위:100us)
_SCAN_MIN	WORD	%FW15	주 태스크 최소 스캔시간(단위:100us)
_SCAN_CUR	WORD	%FW16	주 태스크 현재 스캔시간 표시(단위:100us)
_CTASK_SCAN_MAX	WORD	%FW17	주기 태스크 최대 스캔시간(단위:100us)
_CTASK_SCAN_MIN	WORD	%FW18	주기 태스크 최소 스캔시간(단위:100us)
_CTASK_SCAN_CUR	WORD	%FW19	주기 태스크 현재 스캔시간 표시(단위:100us)
_PLC_TIME	ARRAY[0..7] OF BYTE	%FB40	현재 시각
_PLC_DATE	DATE	%FW24	RTC의 현재 날짜
_PLC_TOD	TIME_OF_DAY	%FD13	RTC의 현재 시간 (ms 단위)
_FLASH_WR_RATIO	WORD	%FW28	FLASH 쓰기 진행 정도
_SCAN_WR	BOOL	%FX464	스캔 값 초기화
_CTASK_SCAN_WR	BOOL	%FX465	정주기 태스크 스캔 값 초기화
_INIT_DONE	BOOL	%FX480	초기화 태스크 수행 완료
_FLASH_WR_CNT	DWORD	%FD16	플래시(FLASH) 쓰기 카운트

부2.2. 모션 플래그

모션 제어 모듈이 모션 제어 중 표시하는 플래그입니다. 모션 제어 상태 및 데이터 등을 표시합니다.

축 관련 플래그는 “_AXxx...” 로 표시되고(xx 는 해당 축번호), 축 그룹 관련 플래그는 “_AGxx...” 형태로 표시됩니다 (xx 는 축 그룹 번호: 01 ~ 40).

1. 모션 공통

변수	타입	메모리할당	설명
_RUN_MODE	BOOL	%JX0.0	RUN
_STOP_MODE	BOOL	%JX0.1	STOP
_TEST_MODE	BOOL	%JX0.2	TEST
_WARNING	BOOL	%JX0.3	경고 발생 상태
_ALARM	BOOL	%JX0.4	알람 발생 상태
_COM_ERR	BOOL	%JX0.8	공통 에러 발생 상태
_ERR_CODE	WORD	%JW0.1	공통 에러코드

2. EtherCAT 마스터

변수	타입	메모리할당	설명
_LINKUP_INFO	BOOL	%JX0.5	링크 업/다운 정보
_COMM	BOOL	%JX0.6	통신 연결 상태
_COMM_ERR	BOOL	%JX0.7	통신 타임아웃 에러
_EC_SLAVE_NUM	UINT	%JW0.17	EtherCAT 슬레이브 연결 개수
_EC_SLAVE_RDY	ARRAY[0..35] OF BOOL	%JX0.320	EtherCAT 슬레이브 레디
_EC_SDO_BUSY	ARRAY[0..35] OF BOOL	%JX0.384	EtherCAT 슬레이브 SDO 처리 Busy
_EC_SDO_ERR	ARRAY[0..35] OF BOOL	%JX0.448	EtherCAT 슬레이브 SDO 처리 에러
_AX_NUM	BYTE	%JB0.10	축 연결 개수
_AX_RDY	ARRAY[0..31] OF BOOL	%JX0.32	축 레디
_AX_SDO_BUSY	ARRAY[0..31] OF BOOL	%JX0.96	축 SDO 처리 Busy
_AX_SDO_ERR	ARRAY[0..31] OF BOOL	%JX0.128	축 SDO 처리 에러
_IO_NUM	BYTE	%JB0.11	I/O 연결 개수
_IO_RDY	ARRAY[0..3] OF BOOL	%JX0.64	I/O 레디
_IO_SDO_BUSY	ARRAY[0..3] OF BOOL	%JX0.160	I/O SDO 처리 Busy
_IO_SDO_ERR	ARRAY[0..3] OF BOOL	%JX0.176	I/O SDO 처리 에러
_PDO_ERR_CNT	WORD	%JW0.12	PDO 에러 카운트
_EC_LINE_FAIL	ARRAY[0..35] OF BOOL	%JX0.512	EtherCAT 케이블 단선 발생
_EC_ERR_SLAVE	UINT	%JW0.18	EtherCAT 연결 실패 슬레이브 연결 번호
_EC_AL_STS_CODE	WORD	%JW0.19	EtherCAT 연결 실패 슬레이브 AL 에러 코드

3. 모션 축

변수	타입	메모리할당	설명
_AXxx_RDY	BOOL	%JXxx.0	축 xx 준비
_AXxx_WARNING	BOOL	%JXxx.1	축 xx 경고 발생 상태
_AXxx_ALARM	BOOL	%JXxx.2	축 xx 알람 발생 상태
_AXxx_SV_ON	BOOL	%JXxx.3	축 xx 서보 온/오프 상태
_AXxx_SV_RDY	BOOL	%JXxx.4	축 xx 서보 레디 상태
_AXxx_AX_TYPE	BOOL	%JXxx.5	축 xx 주축/종축 상태
_AXxx_MST_INFO	BYTE	%JBxx.1	축 xx 주축 정보
_AXxx_LINKED_NODE	USINT	%JBxx.98	축 xx 연결 장치 노드 정보
_AXxx_LINKED_SLOT	USINT	%JBxx.99	축 xx 연결 장치 슬롯 정보
_AXxx_SVON_INCMPL	BOOL	%JXxx.16	축 xx 서보온 미완료
_AXxx_COMM_WARN	BOOL	%JXxx.17	축 xx 통신 경고
_AXxx_DEV_WARN	BOOL	%JXxx.18	축 xx 편차 이상 경고
_AXxx_SV_ERR	BOOL	%JXxx.32	축 xx 서보드라이브 이상
_AXxx_HW_POT	BOOL	%JXxx.33	축 xx 상한 검출
_AXxx_HW_NOT	BOOL	%JXxx.34	축 xx 하한 검출
_AXxx_SW_POT	BOOL	%JXxx.35	축 xx 소프트 상한 검출
_AXxx_SW_NOT	BOOL	%JXxx.36	축 xx 소프트 하한 검출
_AXxx_SV_OFF	BOOL	%JXxx.37	축 xx 서보 오프 상태에서 운전 명령 실행
_AXxx_POS_OVR	BOOL	%JXxx.38	축 xx 위치 결정 이동량 설정 범위 초과
_AXxx_VEL_OVR	BOOL	%JXxx.39	축 xx 최대 속도 초과
_AXxx_DEV_ERR	BOOL	%JXxx.40	축 xx 편차 이상 알람
_AXxx_HOME_INCMPL	BOOL	%JXxx.41	축 xx 원점 미결정 상태에서 절대 위치 운전 지령 실행
_AXxx_COMM_ERR	BOOL	%JXxx.42	축 xx 통신 알람
_AXxx_BUSY	BOOL	%JXxx.48	축 xx 모션 명령 실행 상태
_AXxx_PAUSE	BOOL	%JXxx.49	축 xx 모션 명령 일시 정지 상태(운전 중이나 속도가 0)
_AXxx_STOP	BOOL	%JXxx.50	축 xx 정지 명령에 의한 정지 상태
_AXxx_FAIL	BOOL	%JXxx.51	축 xx 명령 이상 종료 상태
_AXxx_CMPL	BOOL	%JXxx.52	축 xx 명령 실행 완료
_AXxx_DIR	BOOL	%JXxx.53	축 xx 운전 방향
_AXxx_JOG	BOOL	%JXxx.54	축 xx 조그 운전 중
_AXxx_HOME	BOOL	%JXxx.55	축 xx 원점복귀 운전 중
_AXxx_P_CTRL	BOOL	%JXxx.56	축 xx 위치 제어 운전 중
_AXxx_V_CTRL	BOOL	%JXxx.57	축 xx 속도 제어 운전 중

변수	타입	메모리할당	설명
_AXxx_T_CTRL	BOOL	%JXxx.58	축 xx 토크 제어 운전 중
_AXxx_L_INTP	BOOL	%JXxx.59	축 xx 직선보간 운전 중
_AXxx_C_INTP	BOOL	%JXxx.60	축 xx 원호보간 운전 중
_AXxx_SYNC	BOOL	%JXxx.61	축 xx 동기 제어 운전 중
_AXxx_COORD	BOOL	%JXxx.62	축 xx 좌표계 운전 중
_AXxx_SYNCVEL_CTRL	BOOL	%JXxx.63	축 xx 동기 속도 제어(CSV) 운전 중
_AXxx_POS_CMPL	BOOL	%JXxx.64	축 xx 위치 결정 완료
_AXxx_INPOS	BOOL	%JXxx.65	축 xx 인포지션 검출
_AXxx_LATCH_CMPL	BOOL	%JXxx.66	축 xx 래치 완료
_AXxx_HOME_CMPL	BOOL	%JXxx.67	축 xx 원점 결정 완료
_AXxx_BUSY_ACC	BOOL	%JXxx.77	축 xx 모션 명령 가속 운전 상태
_AXxx_BUSY_CONSTVEL	BOOL	%JXxx.78	축 xx 모션 명령 등속 운전 상태
_AXxx_BUSY_DEC	BOOL	%JXxx.79	축 xx 모션 명령 감속 운전 상태
_AXxx_Disabled	BOOL	%JXxx.80	축 xx Disabled 상태
_AXxx_Standstill	BOOL	%JXxx.81	축 xx Standstill 상태
_AXxx_Discrete	BOOL	%JXxx.82	축 xx Discrete 상태
_AXxx_Continuous	BOOL	%JXxx.83	축 xx Continuous 상태
_AXxx_Synchronized	BOOL	%JXxx.84	축 xx Synchronized 상태
_AXxx_Homing	BOOL	%JXxx.85	축 xx Homing 상태
_AXxx_Stopping	BOOL	%JXxx.86	축 xx Stopping 상태
_AXxx_Errorstop	BOOL	%JXxx.87	축 xx Errorstop 상태
_AXxx_CMDBUF_FULL	BOOL	%JXxx.95	축 xx Buffered 명령 버퍼 가득 참
_AXxx_CMDBUF_QUEUED	UINT	%JWxx.6	축 xx 명령 실행 대기 개수
_AXxx_CMDBUF_FREE	UINT	%JWxx.7	축 xx 명령 실행 가능 개수
_AXxx_TPOS	LREAL	%JLxx.2	축 xx 목표 위치
_AXxx_GPOS	LREAL	%JLxx.3	축 xx 현재스캔 명령위치
_AXxx_CVEL	LREAL	%JLxx.4	축 xx 명령 속도
_AXxx_ACCDEC	LREAL	%JLxx.5	축 xx 명령 가/감속도
_AXxx_CJERK	LREAL	%JLxx.6	축 xx 명령 저크
_AXxx_CTRQ	LREAL	%JLxx.7	축 xx 명령 토크
_AXxx_POS	LREAL	%JLxx.8	축 xx 현재 위치
_AXxx_VEL	LREAL	%JLxx.9	축 xx 현재 속도
_AXxx_TRQ	LREAL	%JLxx.10	축 xx 현재 토크
_AXxx_POS_DEV	LREAL	%JLxx.11	축 xx 위치 편차
_AXxx_ALM	BOOL	%JXxx.768	축 xx 드라이브 알람 발생
_AXxx_DRV_WARNING	BOOL	%JXxx.769	축 xx 드라이브 경고 발생

변수	타입	메모리할당	설명
_AXxx_SV_ON_STS	BOOL	%JXxx.770	축 xx 서보온 상태
_AXxx_POT	BOOL	%JXxx.771	축 xx 상한 입력 신호
_AXxx_NOT	BOOL	%JXxx.772	축 xx 하한 입력 신호
_AXxx_HOME_INPUT	BOOL	%JXxx.773	축 xx 원점 입력 신호
_AXxx_LATCH1	BOOL	%JXxx.774	축 xx 래치 1 입력
_AXxx_LATCH2	BOOL	%JXxx.775	축 xx 래치 2 입력
_AXxx_SVPARAM_BUSY	BOOL	%JXxx.776	축 xx SDO 파라미터 읽기/쓰기 처리 중
_AXxx_SDOCMD_BUSY	BOOL	%JXxx.777	축 xx SDO 파라미터 명령 실행 중
_AXxx_DRV_IN	DWORD	%JDxx.25	축 xx 서보드라이브 입력 신호
_AXxx_AX_ERR	WORD	%JWxx.52	축 xx 에러코드 번호
_AXxx_DRV_ERR	WORD	%JWxx.53	축 xx 서보드라이브 에러코드 번호

4. 모션 축 그룹

변수	타입	메모리할당	설명
_AGyy_RDY	BOOL	%CXyy.0	축 그룹 yy 준비
_AGyy_WARNING	BOOL	%CXyy.1	축 그룹 yy 경고 발생 상태
_AGyy_ALARM	BOOL	%CXyy.2	축 그룹 yy 알람 발생 상태
_AGyy_SV_ON	BOOL	%CXyy.3	축 그룹 yy 서보 온/오프 상태
_AGyy_SV_RDY	BOOL	%CXyy.4	축 그룹 yy 서보 레디 상태
_AGyy_BUSY	BOOL	%CXyy.16	축 그룹 yy 모션 명령 실행 상태
_AGyy_PAUSE	BOOL	%CXyy.17	축 그룹 yy 모션 명령 일시 정지 상태(운전 중이나 속도가 0)
_AGyy_STOP	BOOL	%CXyy.18	축 그룹 yy 정지 명령에 의한 정지 상태
_AGyy_FAIL	BOOL	%CXyy.19	축 그룹 yy 명령 이상 종료 상태
_AGyy_CMPL	BOOL	%CXyy.20	축 그룹 yy 명령 실행 완료
_AGyy_LINTP	BOOL	%CXyy.21	축 그룹 yy 직선보간 운전 중
_AGyy_CINTP	BOOL	%CXyy.22	축 그룹 yy 원호보간 운전 중
_AGyy_HOME	BOOL	%CXyy.23	축 그룹 yy 원점복귀 중
_AGyy_INSYNC	BOOL	%CXyy.24	축 그룹 yy 동기운전 중
_AGyy_TLINTP	BOOL	%CXyy.25	축 그룹 yy 좌표계 시간운전 중
_AGyy_CCINTP	BOOL	%CXyy.26	축 그룹 yy 좌표계 원호보간 운전 중
_AGyy_JOG	BOOL	%CXyy.27	축 그룹 yy 좌표계 조그 운전 중
_AGyy_CCINTP3D	BOOL	%CXyy.28	축 그룹 yy 3D 원호보간 운전 중
_AGyy_POS_CMPL	BOOL	%CXyy.32	축 그룹 yy 위치 결정 완료
_AGyy_BUSY_ACC	BOOL	%CXyy.45	축 그룹 yy 모션 명령 가속 운전 상태
_AGyy_BUSY_CONSTVEL	BOOL	%CXyy.46	축 그룹 yy 모션 명령 등속 운전 상태
_AGyy_BUSY_DEC	BOOL	%CXyy.47	축 그룹 yy 모션 명령 감속 운전 상태

변수	타입	메모리할당	설명
_AGyy_Disabled	BOOL	%CXyy.48	축 그룹 yy Disabled 상태
_AGyy_Standby	BOOL	%CXyy.49	축 그룹 yy Standby 상태
_AGyy_Moving	BOOL	%CXyy.50	축 그룹 yy Moving 상태
_AGyy_Homing	BOOL	%CXyy.51	축 그룹 yy Homing 상태
_AGyy_Stopping	BOOL	%CXyy.52	축 그룹 yy Stopping 상태
_AGyy_ErrorStop	BOOL	%CXyy.53	축 그룹 yy ErrorStop 상태
_AGyy_CMDBUF_FULL	BOOL	%CXyy.63	축 그룹 yy Buffered 명령 버퍼 가득 참
_AGyy_CMDBUF_QUEUED	UINT	%CWyy.20	축 그룹 yy Buffered 명령 실행 대기 개수
_AGyy_CMDBUF_FREE	UINT	%CWyy.21	축 그룹 yy 명령 실행 가능 개수
_AGyy_CMD_VEL	LREAL	%CLyy.1	축 그룹 yy 명령 속도
_AGyy_CMD_ACCDEC	LREAL	%CLyy.2	축 그룹 yy 명령 가/감속도
_AGyy_ACT_VEL	LREAL	%CLyy.3	축 그룹 yy 현재 속도
_AGyy_ERR	WORD	%CWyy.16	축 그룹 yy 에러 코드 번호
_AGyy_AX_NUM	WORD	%CWyy.17	축 그룹 yy 구성 축 개수
_AGyy_AX1	BYTE	%CByy.36	축 그룹 yy 구성축 1 축번호
_AGyy_AX2	BYTE	%CByy.37	축 그룹 yy 구성축 2 축번호
_AGyy_AX3	BYTE	%CByy.38	축 그룹 yy 구성축 3 축번호
_AGyy_AX4	BYTE	%CByy.39	축 그룹 yy 구성축 4 축번호

부2.3. 입출력 플래그

입출력 플래그는 내장 디지털 입력 및 내장 디지털 출력 값을 표시하고, 내장 엔코더 값을 표시합니다. 그 밖에 모션제어 모듈에 연결되어 정주기 통신을 하는 PDO Data 를 표시합니다. 아래 예는 L7N 서보드라이브를 연결하였을 때의 정주기 통신 데이터 플래그를 표시한 것입니다.

1. 내장 디지털 입출력 플래그

변수	타입	메모리할당	설명
_IN	ARRAY[0..7] OF BOOL	%IX0.0	내장 디지털 입력
_ENC1	LREAL	%ILO.1	엔코더 1 입력 위치
_ENC2	LREAL	%ILO.2	엔코더 2 입력 위치
_ENC1_UNIT	UINT	%IWO.12	엔코더 1 단위 (0: pulse, 1: mm, 2: inch, 3: degree)
_ENC2_UNIT	UINT	%IWO.13	엔코더 2 단위 (0: pulse, 1: mm, 2: inch, 3: degree)
_ENC1_VEL	LREAL	%ILO.4	엔코더 1 속도
_ENC2_VEL	LREAL	%ILO.5	엔코더 2 속도
_OUT	ARRAY[0..7] OF BOOL	%QX0.0	내장 디지털 출력

2. 슬레이브 프로세스 데이터(PDO Data) 입출력 플래그

변수	타입	메모리할당	설명
_EC001_RxPDO_1600_0_컨트롤_워드	UINT	%QW1.0	컨트롤 워드
_EC001_RxPDO_1600_1_목표_토크	INT	%QW1.1	목표 토크
_EC001_RxPDO_1600_2_목표_위치	DINT	%QD1.1	목표 위치
_EC001_RxPDO_1600_3_운전_모드	SINT	%QB1.8	운전 모드
_EC001_RxPDO_1600_4_Padding	USINT	%QB1.9	Padding
_EC001_RxPDO_1600_5_터치_프로브_기능	UINT	%QW1.5	터치 프로브 기능
_EC001_TxPDO_1A00_0_스테이터스_워드	UINT	%IW1.0	스테이터스 워드
_EC001_TxPDO_1A00_1_실제_토크값	INT	%IW1.1	실제 토크값
_EC001_TxPDO_1A00_2_실제_위치값	DINT	%ID1.1	실제 위치값
_EC001_TxPDO_1A00_3_터치_프로브_1_상승에지_위치값	DINT	%ID1.2	터치 프로브 1 상승에지 위치값
_EC001_TxPDO_1A00_4_디지털_입력	UDINT	%ID1.3	디지털 입력
_EC001_TxPDO_1A00_5_운전_모드_표시	SINT	%IB1.16	운전 모드 표시
_EC001_TxPDO_1A00_6_Padding	USINT	%IB1.17	Padding
_EC001_TxPDO_1A00_7_터치_프로브_상태	UINT	%IW1.9	터치 프로브 상태

부2.4. 모션 좌표계 플래그

좌표계 플래그는 좌표계운전 중 표시하는 플래그입니다. 좌표계 운전중 MCS 좌표계의 현재 TCP 위치 등을 표시합니다. 좌표계 플래그는 “_Kyy...” 형태로 표시됩니다(yy는 축 그룹 번호, 10 진수).

변수	타입	메모리할당	설명
_Kyy_MTCP_Px	LREAL	%KLyy.0	축 그룹 yy X축 좌표(MCS)
_Kyy_MTCP_Py	LREAL	%KLyy.1	축 그룹 yy Y축 좌표(MCS)
_Kyy_MTCP_Pz	LREAL	%KLyy.2	축 그룹 yy Z축 좌표(MCS)
_Kyy_MTCP_A	LREAL	%KLyy.3	축 그룹 yy X축 회전(MCS)
_Kyy_MTCP_B	LREAL	%KLyy.4	축 그룹 yy Y축 회전(MCS)
_Kyy_MTCP_C	LREAL	%KLyy.5	축 그룹 yy Z축 회전(MCS)
_Kyy_PTCP_Px	LREAL	%KLyy.6	축 그룹 yy X축 좌표(PCS)
_Kyy_PTCP_Py	LREAL	%KLyy.7	축 그룹 yy Y축 좌표(PCS)
_Kyy_PTCP_Pz	LREAL	%KLyy.8	축 그룹 yy Z축 좌표(PCS)
_Kyy_PTCP_A	LREAL	%KLyy.9	축 그룹 yy X축 회전(PCS)
_Kyy_PTCP_B	LREAL	%KLyy.10	축 그룹 yy Y축 회전(PCS)
_Kyy_PTCP_C	LREAL	%KLyy.11	축 그룹 yy Z축 회전(PCS)

보증 내용

1. 보증 기간

구입하신 제품의 보증 기간은 제조일로부터 18 개월입니다.

2. 보증 범위

위의 보증 기간 중에 발생한 고장에 대해서는 부분적인 교환 또는 수리를 받으실 수 있습니다. 다만, 아래에 해당하는 경우에는 그 보증 범위에서 제외하오니 양지하여 주시기 바랍니다.

- (1) 사용설명서에 명기된 이외의 부적당한 조건·환경·취급으로 발생한 경우
- (2) 고장의 원인이 당사의 제품 이외의 것으로 발생한 경우
- (3) 당사 및 당사가 정한 지정점 이외의 장소에서 개조 및 수리를 한 경우
- (4) 제품 본래의 사용 방법이 아닌 경우
- (5) 당사에서 출하 시 과학·기술의 수준에서는 예상이 불가능한 사유에 의한 경우
- (6) 기타 천재·화재 등 당사 측에 책임이 없는 경우

3. 위의 보증은 PLC 단위체만의 보증을 의미하므로 시스템 구성이나 제품 응용 시에는 안전성을 고려하여 사용하여 주십시오.

환경 방침

LS ELECTRIC 은 다음과 같이 환경 방침을 준수하고 있습니다.

환경 경영

LS ELECTRIC은 환경보전을 경영의 우선과제로 하며, 전 임직원은 쾌적한 지구환경보전을 위해 최선을 다한다.

제품 폐기에 대한 안내

LS ELECTRIC PLC는 환경을 보호할 수 있도록 설계된 제품입니다. 제품을 폐기할 경우 알루미늄, 철 합성수지(커버)류로 분리하여 재활용 할 수 있습니다.



www.lselectric.co.kr

LS ELECTRIC Co., Ltd.

■ 본사: 서울특별시 용산구 한강대로 92 LS용산타워 14층

■ 구입 문의

서울영업	TEL : (02)2034-4623~38	FAX : (02)2034-4057
부산영업	TEL : (051)310-6855~60	FAX : (051)310-6851
대구영업	TEL : (053)603-7741~8	FAX : (053)603-7788
서부영업(광주)	TEL : (062)510-1891~92	FAX : (062)526-3262
서부영업(대전)	TEL : (042)820-4240~42	FAX : (042)820-4298

■ A/S 문의

기술상담센터	TEL : (전국)1544-2080	FAX : (031)689-7290
서울/경기Global지원	TEL : (031)689-7112	FAX : (031)689-7113
천안Global지원	TEL : (041)550-8308~9	FAX : (041)554-3949
부산Global지원	TEL : (051)310-6922~3	FAX : (051)310-6851
대구Global지원	TEL : (053)603-7751~4	FAX : (053)603-7788
광주Global지원	TEL : (062)510-1885~6	FAX : (062)526-3262

■ 교육 문의


연수원	TEL : (043)268-2631~2	FAX : (043)268-4384
서울/경기교육장	TEL : (031)689-7107	FAX : (031)689-7113
부산교육장	TEL : (051)310-6860	FAX : (051)310-6851
대구교육장	TEL : (053)603-7744	FAX : (053)603-7788

■ 기술 문의

기술상담센터	TEL : (전국)1544-2080	FAX : (031)689-7290
동현산전(안양)	TEL : (031)479-4785~6	FAX : (031)479-4784
나노오트메이션(대전)	TEL : (042)336-7797	FAX : (042)636-8016
신광ENG(부산)	TEL : (051)319-1051	FAX : (051)319-1052
에이엔디시스템(부산)	TEL : (051)319-0668	FAX : (051)319-0669

■ 서비스 지정점

명산전(서울)	TEL : (02)462-3053	FAX : (02)462-3054
TPI시스템(서울)	TEL : (02)895-4803~4	FAX : (02)6264-3545
우진산전(의정부)	TEL : (031)877-8273	FAX : (031)878-8279
신진시스템(안산)	TEL : (031)508-9606	FAX : (031)508-9608
드림시스템(평택)	TEL : (031)665-7520	FAX : (031)667-7520
스마트산전(안양)	TEL : (031)430-4629	FAX : (031)430-4630
세아산전(안양)	TEL : (031)340-5228	FAX : (031)340-5229
성원M&S(인천)	TEL : (032)588-3750	FAX : (032)588-3751
파란자동화(천안)	TEL : (041)554-8308	FAX : (041)554-8310
태영시스템(대전)	TEL : (042)670-7363	FAX : (042)670-7364
디에스산전(청주)	TEL : (043)237-4816	FAX : (043)237-4817
조은시스템(부산)	TEL : (051)319-3923	FAX : (051)319-3924
산전테크(부산)	TEL : (051)319-1025	FAX : (051)319-1026
서진산전(울산)	TEL : (052)227-0335	FAX : (052)227-0337
대명시스템(대구)	TEL : (053)564-4370	FAX : (053)564-4371
제이엠산전(포항)	TEL : (054)284-6050	FAX : (054)284-6051
지이티시스템(구미)	TEL : (054)465-2304	FAX : (054)465-2315
제일시스템(창원)	TEL : (055)273-6778	FAX : (050)4005-6778
지유시스템(광주)	TEL : (062)714-1765	FAX : (062)714-1766
코리아FA(익산)	TEL : (063)838-8002	FAX : (063)838-8001
SJ주식회사(전주)	TEL : (063)213-6900~1	FAX : (063)213-6902

LS ELECTRIC  고객센터 - 신속한 서비스, 든든한 기술지원
전화. 1544-2080 | www.lselectric.co.kr

사용설명서의 사양은 지속적인 제품 개발 및 개선으로 인해 예고없이 변경될 수 있습니다.