OMRON

머신 오토메이션 컨트롤러

NJ/NX 시리즈

CPU 유니트

사용자 매뉴얼
모션 제어편

NX701-17□□
NX701-16□□
NX102-12□□
NX102-11□□
NX102-10□□
NX102-90□□
NX1P2-11□□□□
NX1P2-10□□□□
NX1P2-90□□□□
NJ501-□5□□
NJ501-□4□□
NJ501-□3□□
NJ301-12□□
NJ301-11□□
NJ101-10□□

CPU 유니트



SYSMAC
always in control

SBCE-363T-K6

주의

- 본 매뉴얼 내용의 일부 또는 전부를 무단으로 복사, 복제, 전재하는 것을 금지합니다.
- 본 매뉴얼의 내용에 관해서는 업데이트를 위해 예고 없이 사양 등을 변경하는 경우가 있습니다.
  양해해 주시기 바랍니다.
- 본 매뉴얼의 내용에 관해 오류 등이 있다면 본사 또는 대리점에 연락해 주십시오.
  이때, 매뉴얼 마지막 페이지에 기재된 Man.No.(매뉴얼 No.)도 함께 알려 주십시오.

상표

- Sysmac은 오므론 주식회사 FA 기기 제품의 일본 및 기타 국가에서의 상표 또는 등록 상표입니다.
- Microsoft, Windows, Windows Vista, Excel, Visual Basic은 미국 Microsoft Corporation의 미국 및 기타 국가에서의 등록 상표 또는 상표입니다.
- EtherCAT®은 독일 Beckhoff Automation GmbH에 의해 라이선스를 부여 받은 특허 취득 완료 기술로, 등록 상표입니다.
- ODVA, CIP, CompoNet, DeviceNet, EtherNet/IP는 ODVA의 상표입니다.
- SD, SDHCD 로고는 SD-3C, LLC의 상표입니다.

그 밖에 본문 중에 기재된 회사명 및 제품명은 각 회사의 상표 또는 등록 상표입니다.

저작권

스크린샷은 마이크로소프트의 허가를 얻어 사용하고 있습니다.

머리말

NJ/NX 시리즈 CPU 유니트를 구입해 주셔서 진심으로 감사드립니다.
본 매뉴얼에는 NJ/NX 시리즈 CPU 유니트의 모션 제어 기능 모듈을 사용하는 데 필요한 정보가 기재되어 있습니다. 사용하기 전에 본 매뉴얼을 자세히 읽고 기능 · 성능 등을 충분히 이해한 뒤 시스템 구축에 활용해 주십시오.
또한, 다 읽은 뒤에도 본 매뉴얼을 잘 보관하고 항상 가까운 곳에 놓고 사용해 주십시오.

대상 독자

본 매뉴얼은 다음과 같은 분을 대상으로 기술하고 있습니다.
전기에 대한 지식(전기 공사사 또는 그와 동등한 지식)을 보유한 분을 대상으로 합니다.
- FA 기기 도입을 담당하시는 분
- FA 시스템을 설계하시는 분
- FA 기기를 설치, 접속하시는 분
- FA 현장을 관리하시는 분

또한, 프로그래밍 언어와 관련하여 국제 표준 규격 IEC 61131-3 또는 일본 표준 규격 JIS B 3503에 규정되어 있는 내용을 이해하신 분을 대상으로 기술합니다.

대상 상품

본 매뉴얼은 아래와 같은 제품을 대상으로 합니다.
NX 시리즈 CPU 유니트
- NX701-17□□
- NX701-16□□
- NX102-12□□
- NX102-11□□
- NX102-10□□
- NX102-90□□
- NX1P2-11□□□□
- NX1P2-11□□□□1
- NX1P2-10□□□□
- NX1P2-10□□□□1
- NX1P2-90□□□□
- NX1P2-90□□□□1

NJ 시리즈 CPU 유니트
- NJ501-□5□□
- NJ501-□4□□
- NJ501-□3□□
- NJ301-12□□
- NJ301-11□□
- NJ101-10□□

각 제품의 사양 일부나 제약 사항이 다른 매뉴얼에 기재되어 있는 경우가 있습니다. 🕮「분책 구성 (P.2)」 및 🕮「관련 매뉴얼 (P.26)」을 확인해 주십시오.

분책 구성

본 제품의 매뉴얼은 아래 표와 같이 구성되어 있습니다. 목적에 따라 해당하는 내용을 읽은 후 본 제품을 활용해 주십시오.
본 제품의 조작에는 주로 오토메이션 소프트웨어 Sysmac Studio를 사용합니다. Sysmac Studio에 대해서는 🕮『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.

사용 목적	매뉴얼																		
	기본 정보																		
	NX 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편	NX 시리즈 NX102 CPU 유니트 사용자 매뉴얼 하드웨어편	NX 시리즈 NX1P2 CPU 유니트 사용자 매뉴얼 하드웨어편	NJ 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편	NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편	NX 시리즈 NX1P2 CPU 유니트 사용자 매뉴얼 내장 I/O, 옵션 보드 기능편	NJ/NX 시리즈 명령 참조 매뉴얼 기본편	NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 모션 제어편	NJ/NX 시리즈 명령 참조 매뉴얼 모션편	NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용자 매뉴얼	NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용자 매뉴얼	NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 OPC UA편	NX 시리즈 CPU 유니트 사용자 매뉴얼 FINS 기능편	NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 데이터베이스 접속 CPU 유니트	NJ 시리즈 SECS/GEM 탑재 CPU 유니트 사용자 매뉴얼	NJ 시리즈 NJ Robotics CPU 유니트 사용자 매뉴얼	NJ/NY 시리즈 NC 종합 컨트롤러 사용자 매뉴얼	NJ/NX 시리즈 트러블슈팅 매뉴얼
NX701의 개요 숙지	●																		
NX102의 개요 숙지		●																	
NX1P2의 개요 숙지			●																
NJ 시리즈의 개요 숙지				●															
장착, 설치, 하드웨어 설정	●	●	●	●															
모션 제어를 실행하는 경우								●											
EtherCAT을 사용하는 경우										●									
EtherNet/IP를 사용하는 경우											●								
소프트웨어 설정					●														
모션 제어를 실행하는 경우								●											
EtherCAT을 사용하는 경우										●									
EtherNet/IP를 사용하는 경우											●								
OPC UA를 사용하는 경우												●							
FINS를 사용하는 경우													●						
데이터베이스 접속 서비스를 사용하는 경우														●					
GEM 서비스를 사용하는 경우															●				
로봇 제어를 실행하는 경우																●			
수치 제어를 실행하는 경우																	●		
NX1P2의 기능을 사용하는 경우						●													

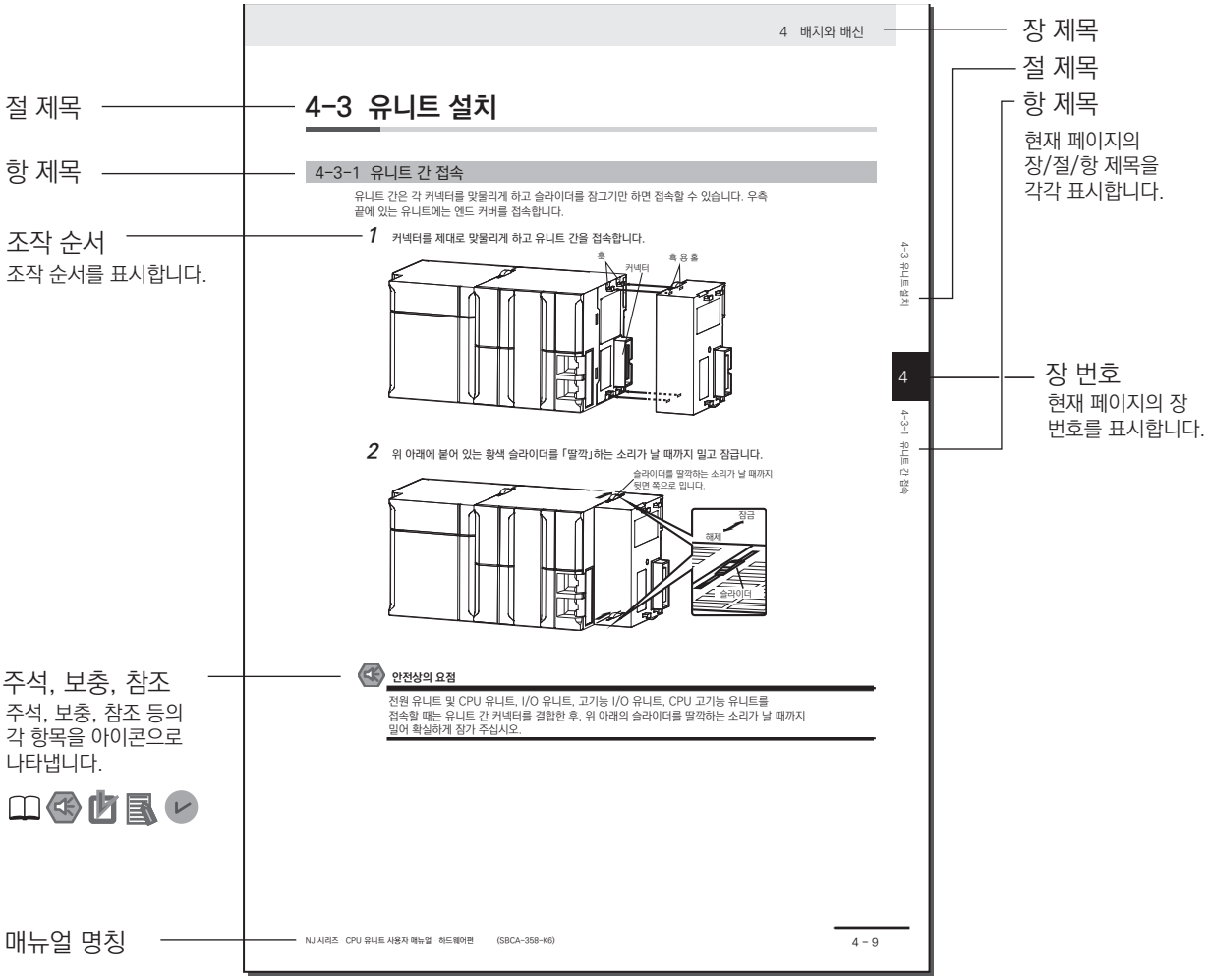
사용 목적	매뉴얼																	
	기본 정보																	
	NX 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편	NX 시리즈 NX102 CPU 유니트 사용자 매뉴얼 하드웨어편	NX 시리즈 NX1P2 CPU 유니트 사용자 매뉴얼 하드웨어편	NJ 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편	NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편	NX 시리즈 NX1P2 CPU 유니트 사용자 매뉴얼 내장 I/O, 옵션 보드 기능편	NJ/NX 시리즈 명령 참조 매뉴얼 기본편	NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 모션 제어편	NJ/NX 시리즈 명령 참조 매뉴얼 모션편	NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용자 매뉴얼	NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP 포트 사용자 매뉴얼	NJ/NX 시리즈 CPU 유니트 OPC UA편	NX 시리즈 CPU 유니트 사용자 매뉴얼 FINS 기능편	NJ/NX 시리즈 데이터베이스 접속 CPU 유니트 사용자 매뉴얼	NJ 시리즈 SECS/GEM 탑재 CPU 유니트 사용자 매뉴얼	NJ 시리즈 NJ Robotics CPU 유니트 사용자 매뉴얼	NJ/NY 시리즈 NC 종합 컨트롤러 사용자 매뉴얼	NJ/NX 시리즈 트러블 슈팅 매뉴얼
사용자 프로그램 작성																		
모션 제어를 실행하는 경우								●	●									
EtherCAT을 사용하는 경우										●								
EtherNet/IP를 사용하는 경우											●							
OPC UA를 사용하는 경우												●						
FINS를 사용하는 경우													●					
데이터베이스 접속 서비스를 사용하는 경우					●		●							●				
GEM 서비스를 사용하는 경우															●			
로봇 제어를 실행하는 경우																●		
수치 제어를 실행하는 경우																	●	
이상 처리를 실행하는 경우																		●
NX1P2의 기능을 사용하는 경우						●												
동작 확인 및 디버그 실행																		
모션 제어를 실행하는 경우								●										
EtherCAT을 사용하는 경우										●								
EtherNet/IP를 사용하는 경우											●							
OPC UA를 사용하는 경우												●						
FINS를 사용하는 경우													●					
데이터베이스 접속 서비스를 사용하는 경우					●									●				
GEM 서비스를 사용하는 경우															●			
로봇 제어를 실행하는 경우																●		
수치 제어를 실행하는 경우																	●	
NX1P2의 기능을 사용하는 경우						●												
이상 관리 기능 및 트러블 발생 시의 대처 방법 숙지 *1												△	△	△	△	△	△	●
보수 작업 숙지	●	●	●	●														
모션 제어를 실행하는 경우								●										
EtherCAT을 사용하는 경우										●								
EtherNet/IP를 사용하는 경우											●							

*1. 이상 관리의 개념과 이상 항목에 대해서는 🕮 『NJ/NX 시리즈 트러블 슈팅 매뉴얼(SBCA-361)』을 참조해 주십시오.
단, △ 마크가 있는 매뉴얼 제품의 특이점에 관한 상세한 내용은 △ 마크가 있는 매뉴얼을 참조해 주십시오.

매뉴얼 읽는 방법

페이지 구성

본 매뉴얼의 각 페이지 구성은 다음과 같습니다.



이 페이지는 설명용 견본입니다. 실제 내용과는 다릅니다.

아이콘

본 매뉴얼에서 사용되고 있는 아이콘에는 다음과 같은 의미가 있습니다.



안전상의 요점

제품을 안전하게 사용하기 위해 실시하거나 회피해야 할 사항을 나타냅니다.



사용상의 주의

제품의 작동 불능, 오작동 또는 성능 및 기능에 대한 악영향을 예방하기 위해 실시하거나 회피해야 할 사항을 나타냅니다.



참고

필요에 따라 읽을 필요가 있는 항목입니다.
알아두면 편리한 정보와 사용하는 데 있어 참고가 되는 내용에 대해 설명하고 있습니다.



버전 관련 정보

CPU 유니트와 Sysmac Studio의 버전에 따른 성능 및 기능의 차이에 대해 설명하고 있습니다.



상세한 정보와 관련 정보가 기재되어 있는 위치를 나타냅니다.

표기에 대한 유의 사항

본 매뉴얼에서는 Sysmac Studio에서 실제 장비 컨트롤러에 데이터를 전송하는 것을「다운로드」, 실제 장비 컨트롤러에서 데이터를 Sysmac Studio로 전송하는 것을「업로드」라고 기술하고 있습니다.
Sysmac Studio에서는「다운로드」및「업로드」모두「Synchronization」기능을 사용합니다.
Sysmac Studio의「Synchronization」기능은 PC상의 데이터와 실제 장비 컨트롤러의 데이터를 자동 비교하여 어느 방향으로 전송할지를 사용자가 선택하는 기능입니다.

목차 구성

1 모션 제어 기능 모듈의 개요

2 모션 제어계의 구성과 원리

3 축과 축 그룹의 설정

4 Sysmac Studio를 이용한 배선 확인

5 모션 제어 파라메터

6 모션 제어 프로그램

7 수동 운전

8 원점 결정

9 모션 제어 기능

10 샘플 프로그램

11 이상 확인과 조치

A 부록

I 색인

목차

머리말 1
대상 독자 1
대상 상품 1

분책 구성 2

매뉴얼 읽는 방법 4
페이지 구성 4
아이콘 5
표기에 대한 유의 사항 5

목차 구성 7

목차 8

동의 사항 14

안전상의 주의 16

안전상의 요점 17

사용상의 주의 18

법규와 규격 19
해외에서의 사용 19
EU 지령에 대한 적합 19
KC 규격에 대한 적합 20
선급 규격에 대한 적합 20
소프트웨어의 라이센스와 저작권 20

버전 21
버전 확인 방법 21
CPU 유니트의 유니트 버전과 Sysmac Studio의 버전 25

관련 매뉴얼 26

매뉴얼 개정 이력 29

제1장 모션 제어 기능 모듈의 개요

1-1 특징 1-2
1-2 시스템 구성 1-3
1-3 기본적인 사용 흐름도 1-5
1-4 사양 1-7
1-4-1 일반 사양 1-7
1-4-2 성능 사양 1-7
1-4-3 기능 사양 1-11

제2장 모션 제어계의 구성과 원리

2-1 CPU 유니트 내부 구성 2-2
2-2 모션 제어계의 구성 2-3

2-3 모션 제어계의 원리 2-5
2-3-1 CPU 유니트의 태스크 2-5
2-3-2 모션 제어의 태스크 동작 예 2-12
2-4 EtherCAT 통신과의 관계 2-19
2-4-1 CAN application protocol over EtherCAT(CoE) 2-19
2-4-2 EtherCAT 마스터 기능과 모션 제어 기능 모듈의 관계 2-20
2-4-3 프로세스 데이터 통신 주기와 모션 제어 주기와의 관계 2-23

제3장 축과 축 그룹의 설정

3-1 축 3-2
3-1-1 축의 개요 3-2
3-1-2 축 파라메터의 개요 3-3
3-1-3 축 변수의 개요 3-6
3-1-4 축 변수의 동기 3-8
3-1-5 사용자 프로그램에서 축 지정하는 방법 3-8
3-2 축 설정 순서 3-9
3-2-1 축 설정 흐름도 3-9
3-2-2 설정 순서 3-9
3-3 축 그룹 3-19
3-3-1 축 그룹의 개요 3-19
3-3-2 축 그룹 파라메터의 개요 3-20
3-3-3 축 그룹 변수의 개요 3-21
3-3-4 사용자 프로그램에서 축 그룹 지정하는 방법 3-22
3-4 축 그룹 설정 순서 3-24
3-4-1 축 그룹 설정 흐름도 3-24
3-4-2 설정 순서 3-24

제4장 Sysmac Studio를 이용한 배선 확인

4-1 Sysmac Studio의 기능 4-2
4-1-1 MC 시운전 기능 일람 4-2
4-1-2 사용 순서 4-3
4-1-3 축 파라메터 설정 예 4-4
4-1-4 MC 시운전 기능의 기동 순서 4-5
4-2 센서 신호 모니터 4-7
4-3 모터 동작 확인 4-8
4-3-1 서보 ON 4-8
4-3-2 조그 전송 4-9
4-3-3 원점 복귀 4-9
4-3-4 절대값 위치 결정 4-10
4-3-5 상대값 위치 결정 4-11

제5장 모션 제어 파라메터

5-1 개요 5-2
5-2 축 파라메터 5-4
5-2-1 축 파라메터 일람 5-4
5-2-2 축 기본 설정 5-6
5-2-3 단위 변환 설정 5-11
5-2-4 동작 설정 5-18
5-2-5 확장 동작 설정 5-21
5-2-6 리미트 설정 5-22
5-2-7 위치 카운트 설정 5-23
5-2-8 서보 드라이버 설정 5-25
5-2-9 원점 복귀 설정 5-26

5-2-10 축 파라메터 설정 예 5-28
5-3 축 그룹 파라메터 5-30
5-3-1 축 그룹 파라메터의 일람 5-30
5-3-2 축 그룹 기본 설정 5-31
5-3-3 축 그룹 동작 설정 5-33
5-3-4 축 그룹의 유효 5-34

제6장 모션 제어 프로그램

6-1 개요 6-2
6-2 모션 제어 명령 6-5
6-2-1 PLCopen® 모션 제어용 펑션 블록 6-5
6-2-2 모션 제어 기능 모듈의 모션 제어 명령 6-5
6-3 상태 변이 6-6
6-3-1 모션 제어 기능 모듈의 상태 6-6
6-3-2 축 상태 6-6
6-3-3 축 그룹의 상태 6-8
6-4 모션 제어 명령의 기동과 상태 6-9
6-4-1 기동에 대한 공통 규칙 6-9
6-4-2 기동의 타이밍 차트 6-11
6-4-3 모션 제어 명령 재기동 시의 타이밍 차트 6-13
6-4-4 모션 제어 명령 다중 기동 시의 타이밍 차트 6-14
6-5 위치의 취급 6-15
6-5-1 위치의 종류 6-15
6-5-2 각 축 종별에서 사용 가능한 위치의 종류 6-16
6-6 모션 제어 시스템 변수 6-17
6-6-1 모션 제어 시스템 변수의 개요 6-17
6-6-2 모션 제어 시스템 변수의 구조 6-19
6-6-3 모션 제어 시스템 변수의 일람 6-20
6-7 캠 테이블과 캠 데이터 변수 6-31
6-8 모션 제어 프로그램의 작성 방법 6-35
6-9 캠 테이블의 작성 방법 6-37

제7장 수동 운전

7-1 개요 7-2
7-2 서보 ON 7-3
7-2-1 서보 ON 순서 7-4
7-2-2 축 파라메터의 설정 7-4
7-2-3 프로그램 예 7-5
7-3 조그 전송 7-6
7-3-1 조그 전송 순서 7-6
7-3-2 축 파라메터의 설정 7-7
7-3-3 입력 변수의 설정 예 7-7
7-3-4 프로그램 예 7-8

제8장 원점 결정

8-1 개요 8-2
8-2 원점 복귀 순서 8-5
8-2-1 원점 복귀 파라메터의 설정 8-6
8-2-2 원점 복귀 모니터 8-12

8-3 원점 복귀 동작 8-13
8-4 절대값 엔코더 원점 설정 8-14
8-4-1 기능 개요 8-15
8-4-2 설정 순서 8-16
8-5 고속 원점 복귀 8-18

제9장 모션 제어 기능

9-1 단축 위치 제어 9-3
9-1-1 동작 개요 9-3
9-1-2 절대값 위치 결정 9-4
9-1-3 상대값 위치 결정 9-4
9-1-4 인터럽트 피딩 위치 결정 9-5
9-1-5 사이클릭 동기 위치 결정 9-6
9-1-6 정지 9-6
9-1-7 오버라이드 9-11
9-2 단축 동기 제어 9-12
9-2-1 동기 제어의 개요 9-12
9-2-2 기어 동작 9-12
9-2-3 위치 지정 기어 동작 9-13
9-2-4 캠 동작 9-14
9-2-5 캠 테이블 9-15
9-2-6 사다리꼴 패턴 캠 동작 9-23
9-2-7 가감산 위치 결정 9-24
9-2-8 주축 위상 보정 9-25
9-2-9 종축 위치 보정 9-26
9-2-10 멀티 모션에서 동기 제어 실현하는 방법 9-26
9-3 단축 속도 제어 9-28
9-3-1 속도 제어 9-28
9-3-2 사이클릭 동기 속도 제어 9-29
9-4 단축 토크 제어 9-30
9-5 단축 제어의 공통 기능 9-31
9-5-1 위치 9-31
9-5-2 속도 9-33
9-5-3 가속과 감속 9-34
9-5-4 저크 9-36
9-5-5 동작 방향 지정 9-37
9-5-6 모션 제어 명령의 재기동 9-40
9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) 9-45
9-6 다축 협조 제어 9-51
9-6-1 동작 개요 9-51
9-6-2 직선 보간 9-54
9-6-3 원호 보간 9-55
9-6-4 축 그룹 사이클릭 동기 위치 제어 9-55
9-6-5 다축 협조 제어에서의 정지 9-56
9-6-6 다축 협조 제어에서의 오버라이드 9-58
9-7 다축 협조 제어의 공통 기능 9-59
9-7-1 다축 협조 제어에서의 속도 9-59
9-7-2 다축 협조 제어에서의 가속과 감속 9-60
9-7-3 다축 협조 제어에서의 저크 9-61
9-7-4 다축 협조 제어의 모션 제어 명령 재기동 9-62
9-7-5 다축 협조 제어의 모션 제어 명령 다중 기동(버퍼 모드) 9-63
9-8 기타 기능 9-70
9-8-1 현재 위치 변경 기능 9-70
9-8-2 토크 제한 기능 9-71
9-8-3 래치 기능 9-71
9-8-4 구역 기능 9-72
9-8-5 소프트웨어 리미트 기능 9-73

9-8-6 위치 편차 감시 기능 9-75
9-8-7 편차 카운터 리셋 9-75
9-8-8 축 간 편차 감시 기능 9-76
9-8-9 인포지션 체크 9-77
9-8-10 축 사용 변경 기능 9-79
9-8-11 디지털 캠 스위치 유효 기능 9-79
9-8-12 임의 좌표계의 3D 모션 모니터 표시 9-80

제10장 샘플 프로그램

10-1 샘플 프로그램의 개요 10-2
10-1-1 사용 기기 10-2
10-1-2 설치와 배선 10-2
10-1-3 설정 10-2

10-2 기본 프로그램 예 10-3
10-2-1 EtherCAT 통신의 감시와 서보 ON 10-3
10-2-2 마스터 컨트롤 명령에 의한 축 동작의 인터락 10-5
10-2-3 이상 감시와 이상 해제(단축 동작, 동기 동작) 10-7
10-2-4 이상 감시와 이상 해제(다축 협조 동작) 10-9
10-2-5 명령 이상 감시 방법 10-14
10-2-6 이상 해제 감시 방법 10-16
10-2-7 단축 동작 중의 축 정지 10-18
10-2-8 다축 협조 동작 중의 축 그룹 정지 10-22
10-2-9 원점 복귀와 절대값 위치 결정 10-28
10-2-10 명령 재기동으로 목표 위치 변경하기 10-32
10-2-11 인터럽트 피딩 위치 결정 10-37
10-2-12 명령 재기동으로 캠 테이블 변환하기 10-42
10-2-13 캠 프로파일 곡선을 사용하여 동기 위치 보정하기 10-51
10-2-14 캠 동작 시에 주축의 위상 보정하기 10-60
10-2-15 속도 제어 중에 현재 위치 변경하기 10-68
10-2-16 캠 데이터 변수의 변경과 캠 테이블 저장 10-74
10-2-17 축 파라메터의 일시 갱신 10-83
10-2-18 캠 테이블의 종료점 인덱스 갱신 10-86

제11장 이상 확인과 조치

11-1 이상 확인과 조치의 개요 11-2

부록

A-1 서보 드라이버 1S 시리즈와의 접속 A-2
A-1-1 서보 드라이버와의 배선 A-2
A-1-2 서보 드라이버의 설정 A-2

A-2 서보 드라이버 G5 시리즈와의 접속 A-11
A-2-1 서보 드라이버와의 배선 A-11
A-2-2 서보 드라이버의 설정 A-11

A-3 엔코더 입력 터미널과의 접속 A-21
A-3-1 엔코더 입력 터미널과의 배선 A-21
A-3-2 엔코더 입력 터미널 설정 A-21

A-4 NX 유니트와의 접속 A-27

A-5 PDS 상태 변이 A-28
A-5-1 PDS 상태 제어 방식 A-29
A-5-2 주회로 전원 OFF 검출 A-30

A-6 용어 설명 A-31
A-6-1 NJ/NX 시리즈 공통 A-31
A-6-2 모션 제어 기능 A-32

A-6-3 EtherCAT 통신 A-33
A-7 버전 관련 정보 A-34

색인

동의 사항

「당사 상품」에 대해 특별한 합의가 없는 경우에는 고객의 구입처에 관계없이 본 동의 사항의 기재 조건을 적용합니다.

● 정의

본 동의 사항 속 용어의 정의는 다음과 같습니다.

- 「당사 상품」: 「당사」의 FA 시스템 기기, 범용 제어 기기, 센싱 기기, 전자 · 기구 부품
- 「카탈로그 등」: 「당사 상품」에 관한 베스트 제어 기기 오므론, 전자 · 기구 부품 종합 카탈로그, 기타 카탈로그, 사양서, 취급 설명서, 매뉴얼 등으로 전자적 방법으로 제공되는 것도 포함합니다.
- 「이용 조건 등」: 「카탈로그 등」에 기재된 「당사 상품」의 이용 조건, 정격, 성능, 작동 환경, 취급 방법, 이용상의 주의, 금지 사항, 기타
- 「고객 용도」: 「당사 상품」을 고객께서 이용하는 방법으로, 고객께서 제조하는 부품, 전자 기판, 기기, 설비 또는 시스템 등에 대한 「당사 상품」의 내장 또는 이용을 포함합니다.
- 「적합성 등」: 「고객 용도」에서 「당사 상품」의 (a)적합성, (b)동작, (c)제3자의 지적 재산 비침해, (d)법령 준수 및 (e)각종 규격 준수

● 기재 사항의 주의

「카탈로그 등」의 기재 내용에 대해서는 다음 사항을 이해해 주십시오.

- 정격값 및 성능값은 단독 시험의 각 조건하에서 얻은 값으로, 각 정격값 및 성능값의 복합적인 조건하에서 얻을 수 있는 값을 보증하는 것은 아닙니다.
- 참고 데이터는 참고로 제공하는 것으로 그 범위에서 항상 정상으로 작동하는 것을 보증하지는 않습니다.
- 이용 사례는 참고이므로 「당사」는 「적합성 등」에 대해 보증하지 않습니다.
- 「당사」는 개선이나 당사 사정 등에 의해 「당사 상품」의 생산을 중지 또는 「당사 상품」의 사양을 변경할 수 있습니다.

● 이용 시의 주의

채택 및 이용 시에는 다음 사항을 이해해 주십시오.

- 정격 · 성능 외에 「이용 조건 등」을 준수하여 이용해 주십시오.
- 고객께서 직접 「적합성 등」을 확인하고 「당사 상품」의 이용 여부를 판단해 주십시오. 「당사」는 「적합성 등」을 일절 보증하지 않습니다.
- 「당사 상품」이 고객의 시스템 전체에 대해 의도한 용도로 적절히 배전 · 설치되었는지 고객께서 직접, 반드시 사전에 확인해 주십시오.
- 「당사 상품」을 사용할 때는 (ⅰ)정격 및 성능에 대해 여유가 있는「당사 상품」의 이용, 여유 설계 등의 안전 설계, (ⅱ)「당사 상품」이 고장 나도 「고객 용도」의 위험을 최소화하는 안전 설계, (ⅲ)이용자에게 위험을 알리기 위한 안전 대책 시스템을 전체적으로 구축, (ⅳ)「당사 상품」 및 「고객 용도」의 정기적인 유지 보수 등 각 사항을 실시해 주십시오.
- 「당사」는 DDoS 공격(분산형 DoS 공격), 컴퓨터 바이러스 및 기타 기술적인 유해 프로그램, 부정 접속으로 인해 「당사 상품」, 설치된 소프트웨어 또는 모든 컴퓨터 기기, 컴퓨터 프로그램, 네트워크, 데이터베이스가 감염되어도 그로 인해 직 · 간접적으로 발생한 손실, 손해, 기타 비용에 대해 그 어떠한 책임도 지지 않습니다.
  고객께서 직접 ①안티바이러스 보호, ②데이터 입출력, ③분실 데이터 복원, ④「당사 상품」 또는 설치된 소프트웨어에 대한 컴퓨터 바이러스 감염 방지, ⑤「당사 상품」에 대한 부정 접속 방지에 대한 충분한 조치를 마련해 주십시오.

- 「당사 상품」은 일반 공업 제품을 위한 범용품으로 설계, 제조되었습니다. 따라서 다음에 게재하는 용도로 사용하는 경우는 의도하지 않고 있으므로 고객께서 「당사 상품」을 이러한 용도로 사용할 때 「당사」는 「당사 상품」에 대해 그 어떠한 보증도 하지 않습니다. 단, 다음에 게재된 용도라도 「당사」가 의도한 특별한 상품 용도인 경우나 특별한 합의가 있는 경우에는 제외합니다.
  (a) 높은 안전성을 필요로 하는 용도(예: 원자력 제어 설비, 연소 설비, 항공 · 우주 설비, 철도 설비, 승강 설비, 오락 설비, 의료용 기기, 안전 장비, 기타 생명 · 신체에 위험이 미칠 수 있는 용도)
  (b) 높은 신뢰성을 필요로 하는 용도(예: 가스 · 수도 · 전기 등의 공급 시스템, 24시간 연속 운전 시스템, 결제 시스템 외 권리 · 재산을 취급하는 용도 등)
  (c) 까다로운 조건 또는 환경에서의 용도(예: 실외에 설치하는 설비, 화학적 오염을 입는 설비, 전자적 방해를 받는 설비, 진동 · 충격을 받는 설비 등)
  (d) 「카탈로그 등」에 기재되어 있지 않은 조건이나 환경에서의 용도
- 상기 (a)부터 (d)에 기재된 것 이외에 「본 카탈로그 등에 기재된 상품」은 자동차(오토바이 포함. 이하 동일)용이 아닙니다. 자동차에 탑재하는 용도로는 이용하지 마십시오. 자동차 탑재용 상품에 대해서는 당사 영업 담당자에게 문의해 주십시오.

● 보증 조건

「당사 상품」의 보증 조건은 다음과 같습니다.

- 보증 기간: 구입 후 1년간으로 합니다.
  (단, 「카탈로그 등」에 별도 기재가 있는 경우를 제외합니다.)
- 보증 내용: 고장난 「당사 상품」에 대해 아래 중 하나를 「당사」가 임의로 판단하여 실시합니다.
  (a) 당사 유지 보수 서비스 거점에서 고장난 「당사 상품」의 무상 수리
  (단, 전자 · 기구 부품에 대해서는 수리 대응하지 않습니다.)
  (b) 고장난 「당사 상품」과 동일한 수의 대체품 무상 제공
- 보증 대상 외: 고장의 원인이 다음 중 하나에 해당할 경우에는 보증하지 않습니다.
  (a) 「당사 상품」 본래의 사용 방법 이외의 이용
  (b) 「이용 조건 등」에서 벗어난 이용
  (c) 본 동의 사항 「이용 시의 주의」에 반하는 이용
  (d) 「당사」 이외에서의 개조, 수리에 의한 경우
  (e) 「당사」 이외의 자가 이용한 소프트웨어 프로그램에 의한 경우
  (f) 「당사」 출하 시의 과학 · 기술 수준으로는 예견할 수 없었던 원인
  (g) 상기 이외 「당사」 또는 「당사 상품」 이외의 원인(자연재해 등의 불가항력을 포함)

● 책임의 제한

본 동의 사항에 기재되어 있는 보증이 「당사 상품」에 관련된 보증의 전부입니다.
「당사 상품」과 관련하여 발생한 손해에 대해 「당사」 및 「당사 상품」의 판매점은 책임을 지지 않습니다.

● 수출 관리

「당사 상품」 또는 기술 자료를 수출 또는 비거주자에게 제공하는 경우에는 안전보장무역관리에 관한 국내 및 관계 각국의 법령 · 규제를 준수해 주십시오. 고객께서 법령 · 규칙을 위반할 경우에는 「당사 상품」 또는 기술 자료를 제공할 수 없는 경우가 있습니다.

안전상의 주의

안전상의 주의 내용에 대해서는 아래 매뉴얼을 참조해 주십시오.

- 🕮『NX 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-418)』
- 🕮『NX 시리즈 NX102 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-462)』
- 🕮『NX 시리즈 NX1P2 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-448)』
- 🕮『NJ 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-358)』

안전상의 요점

안전상의 요점 내용에 대해서는 아래 매뉴얼을 참조해 주십시오.

- 🕮『NX 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-418)』
- 🕮『NX 시리즈 NX102 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-462)』
- 🕮『NX 시리즈 NX1P2 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-448)』
- 🕮『NJ 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-358)』

사용상의 주의

사용상의 주의 내용에 대해서는 아래 매뉴얼을 참조해 주십시오.

- 🕮『NX 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-418)』
- 🕮『NX 시리즈 NX102 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-462)』
- 🕮『NX 시리즈 NX1P2 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-448)』
- 🕮『NJ 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-358)』

법규와 규격

해외에서의 사용

본 제품 중 외환 및 외국 무역 관리법에서 정한 수출 허가, 승인 대상 화물(또는 기술)에 해당하는 것을 수출(또는 비거주자에게 제공)할 경우에는 동법에 근거하는 수출 허가, 승인(또는 역무 거래 허가)이 필요합니다.

EU 지령에 대한 적합

적합 지령

- EMC 지령
- 저전압 지령

적합의 개념

● EMC 지령

오므론 상품은 각종 기계, 제조 장비에 내장해서 사용되는 전기 기기이므로 내장한 기계 · 장비가 문제없이 EMC 규격에 적합할 수 있도록 상품 자체를 관련된 EMC 규격(*1)에 맞춰 제조하고 있습니다. 그러나 고객의 기계 · 장비가 매우 다양하고 EMC의 성능도 EU 지령 적합 상품을 내장한 기기 · 제어반의 구성, 배선 상태, 배치 상태 등에 따라 달라지므로 고객의 사용 상태에 대한 적합성은 확인할 수 없습니다. 따라서 기계 · 장비 전체에 대한 최종적인 EMC 적합성의 확인은 고객께서 직접 실시해 주십시오.

*1. EMC(Electro-Magnetic Compatibility: 전자 환경 양립성) 관련 규격 중 EMS(Electro-Magnetic Susceptibility: 전자 감수성)에 관해서는 EN61131-2.
EMI(Electro-Magnetic Interference: 전자 방해)에 관해서는 EN61131-2.
또한, Radiated emission은 10m법에 따릅니다.

● 저전압 지령

전원 전압 AC50V~1000V 및 DC75V~1500V에서 작동하는 기기에 대해 필요한 안전성이 확보될 것을 요구하고 있습니다. 적용 규격은 EN61010-2-201입니다.

● EU 지령에 대한 적합

NJ/NX 시리즈는 EU 지령에 적합합니다. 그러나 고객의 기계 · 장비를 EU 지령에 적합하게 사용하려면 아래와 같은 주의가 필요합니다.

- NJ/NX 시리즈는 반드시 제어반 내에 설치해 주십시오.
- DC 전원 유니트 및 I/O 유니트에 접속하는 DC 전원은 SELV 사양의 전원을 사용해 주십시오.
- NJ/NX 시리즈의 EU 지령 적합 상품은 EMI와 관련해 공통 이미션 규격에 적합하지만, 특히 Radiated emission(10m법)과 관련해서는 사용하는 제어반의 구성, 접속되는 다른 기기와의 관계, 배선 등에 따라 변할 수 있습니다.
  따라서 EU 지령 적합품인 NJ/NX 시리즈를 사용하는 경우에도 고객께서 기계 · 장비 전체적으로 EU 지령의 적합성을 확인 · 대응할 필요가 있습니다.

KC 규격에 대한 적합

본 제품을 한국에서 사용할 경우에는 아래와 같은 주의 사항을 지켜 주십시오.

A 급 기기 (업무용방송통신기자재)
이 기기는 업무용(A 급) 전저파작합기기로서 판매자
또는 사용자는 이 점을 주의하시기 바라며, 가정외의
지역에서 사용하는 것을 목적으로 합니다.

ClassA 기기(업무용 방송 통신 기기)
이 기기는 업무용 전자파 발생 기기(Class A)이며, 가정 이외의 장소에서의 사용을 의도했습니다.
판매자와 사용자는 이 점에 주의해 주십시오.

선급 규격에 대한 적합

본 제품은 각종 선급 규격에 적합합니다. 각 선급 규격에 적용할 때에는 사용 조건이 설정되어 있어 설치 장소에 따라서는 사용할 수 없는 경우도 있으므로 사용할 때는 반드시 당사 영업 담당자에게 문의해 주십시오.

각 선급 규격의 사용 조건(NK, LR)

- 본 제품은 반드시 제어반에 설치해 주십시오.
- 제어반 개폐구 등의 틈새는 개스킷 등으로 완전히 차폐시켜 주십시오.
- 전원 라인에 아래와 같은 노이즈 필터를 접속해 주십시오.

● 노이즈 필터

제조 회사	형식
코셀사	TAH-06-683

소프트웨어의 라이센스와 저작권

본 제품에는 서드 파티의 소프트웨어가 내장되어 있습니다. 이 소프트웨어와 관련된 라이센스와 저작권에 대해서는 http://www.ia.omron.co.kr을 참조해 주십시오.

버전

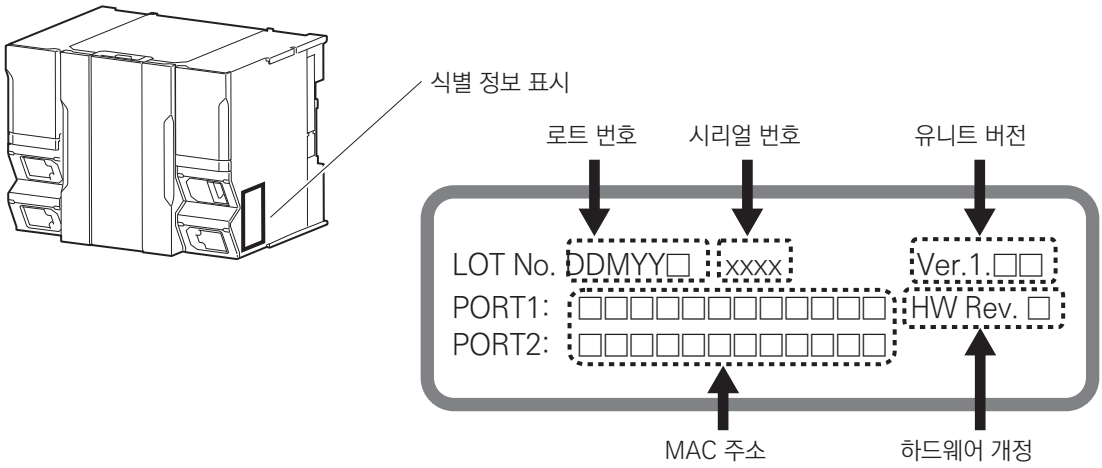
NJ/NX 시리즈의 각 유니트 및 각 EtherCAT 슬레이브는 하드웨어와 소프트웨어가 하드웨어 개정이나 유니트 버전 등의 수정 번호로 버전 관리되고 있습니다. 하드웨어 또는 소프트웨어에 사양 변경이 있을 때마다 하드웨어 개정 또는 유니트 버전이 갱신됩니다. 따라서 동일 형식의 유니트나 EtherCAT 슬레이브라도 하드웨어 개정 또는 유니트 버전이 다르면 탑재 기능이나 성능이 달라집니다.

버전 확인 방법

버전은 식별 정보 표시 또는 Sysmac Studio로 확인할 수 있습니다.

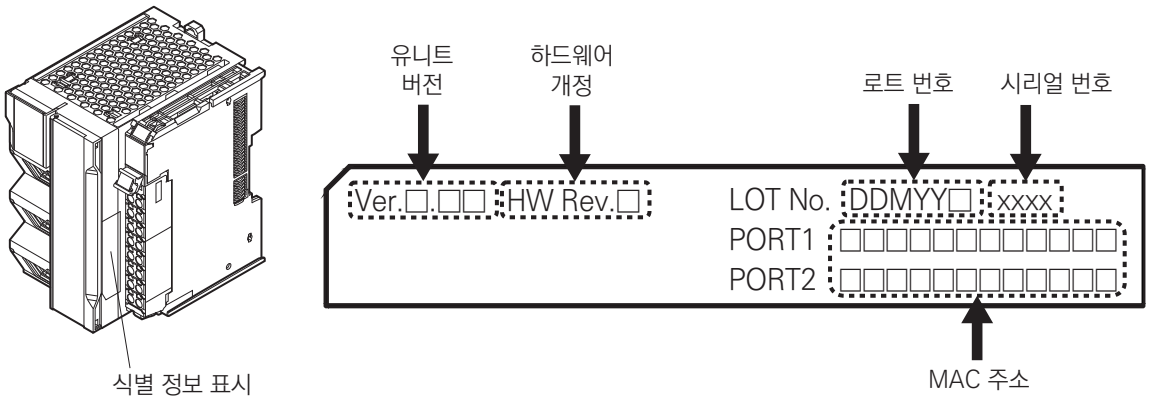
식별 정보 표시를 이용한 확인

제품 측면에 있는 식별 정보 표시로 버전을 확인할 수 있습니다.
NX 시리즈 CPU 유니트 NX701-□□□□의 식별 정보 표시는 아래 그림과 같습니다.



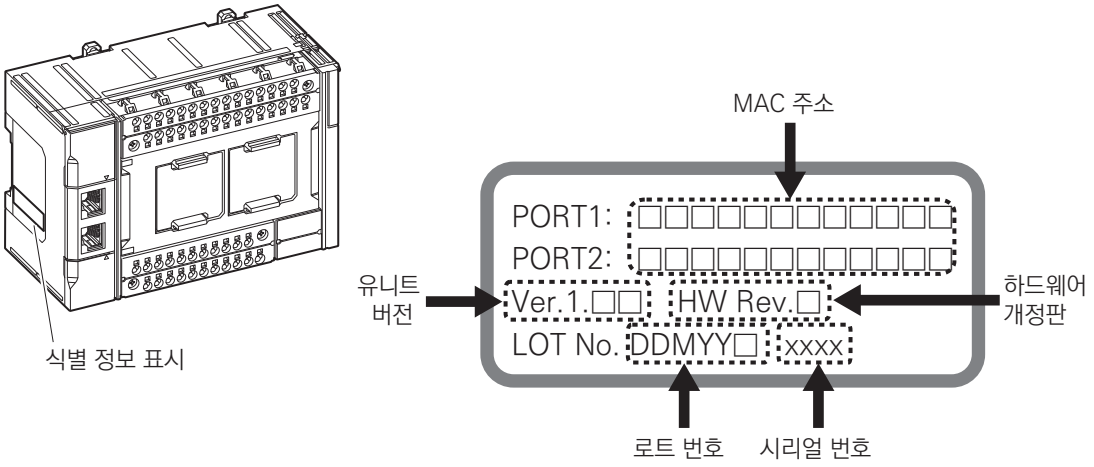
(주) 하드웨어가 개정되지 않은 유니트에는 하드웨어 개정 표시가 없습니다.

NX 시리즈 CPU 유니트 NX102-□□□□의 식별 정보 표시는 아래 그림과 같습니다.



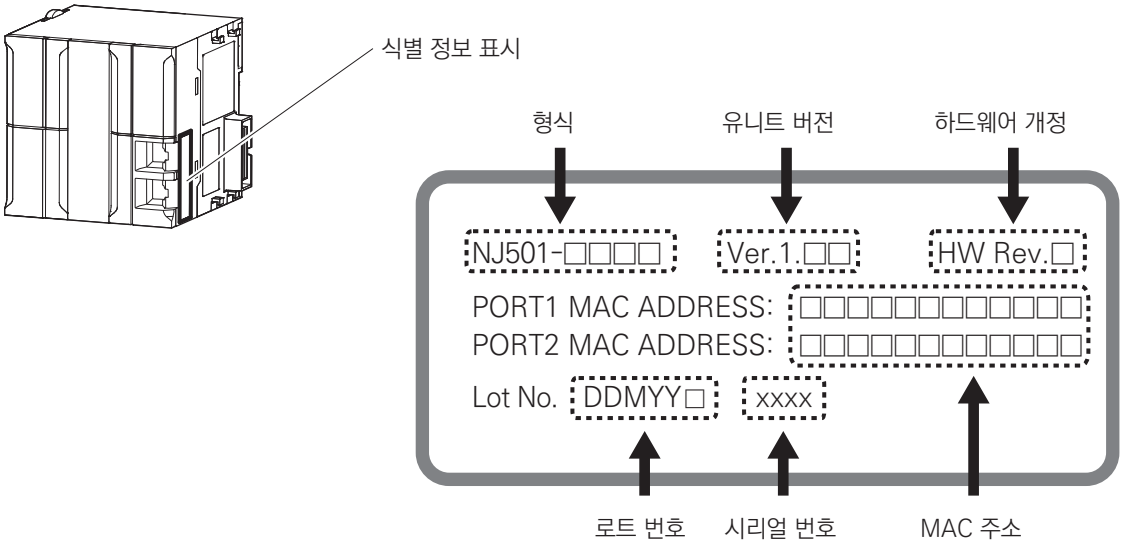
(주) 하드웨어가 개정되지 않은 유니트에는 하드웨어 개정 표시가 없습니다.

NX 시리즈 CPU 유니트 NX1P2-□□□□□□□□의 식별 정보 표시는 아래 그림과 같습니다.



(주) 하드웨어가 개정되지 않은 유니트에는 하드웨어 개정 표시가 없습니다.

NJ 시리즈 CPU 유니트 NJ501-□□□□의 식별 정보 표시는 아래 그림과 같습니다.



(주) 하드웨어가 개정되지 않은 유니트에는 하드웨어 개정 표시가 없습니다.

Sysmac Studio를 이용한 확인 방법

Sysmac Studio로 버전을 확인할 수 있습니다. 유니트와 EtherCAT 슬레이브는 확인 방법이 다릅니다.

● NX 시리즈 CPU 유니트에서 버전 확인하는 방법

유니트의 버전은 온라인 상태의 [Production Information]에서 확인할 수 있습니다. 버전을 확인할 수 있는 유니트는 아래와 같습니다.

형식	버전을 확인할 수 있는 유니트
NX701-□□□□	CPU 유니트
NX102-□□□□	CPU 유니트, CPU 랙상의 NX 유니트
NX1P2-□□□□	CPU 유니트, CPU 랙상의 NX 유니트, 옵션 보드

- 멀티 뷰 익스플로러에서 [Configurations and Setup]-[CPU/Expansion Racks]-[CPU Rack]을 우클릭하고 [Production Information]을 선택합니다.
  [Production Information] 대화 상자가 표시됩니다.

● NJ 시리즈 CPU 유니트에서 버전 확인하는 방법

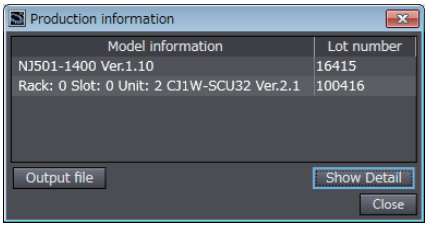
유니트의 버전은 온라인 상태의 [Production Information]에서 확인할 수 있습니다. 단, 버전을 확인할 수 있는 유니트는 CPU 유니트, CJ 시리즈의 고기능 I/O 유니트 및 CJ 시리즈의 CPU 고기능 유니트뿐입니다. CJ 유니트의 기본 I/O 유니트 버전은 Sysmac Studio에서는 확인할 수 없습니다.
버전 확인 방법은 아래와 같습니다.

1 **멀티 뷰 익스플로러에서 [Configurations and Setup]-[CPU/Expansion Racks]를 더블 클릭합니다. 또는 [Configurations and Setup]-[CPU/Expansion Racks]를 우클릭하고 [Edit]를 선택합니다.**
유니트 에디터가 표시됩니다.

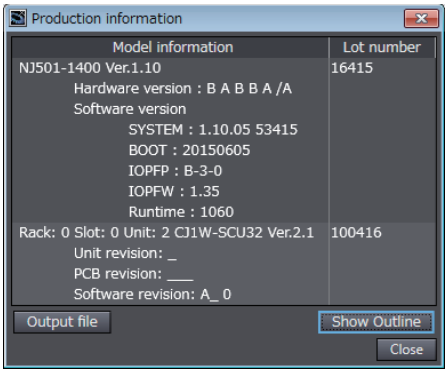
2 **유니트 에디터의 공란 위에서 우클릭하여 [Production Information]을 선택합니다.**
[Production Information] 대화 상자가 표시됩니다.

● 생산 정보 표시 내용의 변환

[Production Information] 대화 상자 우측 아래에 있는 [Show Outline] 또는 [Show Detail]을 선택합니다. [Production Information]의 간단 표시와 상세 표시가 변환됩니다.



간단 표시



상세 표시

표시 내용은 간단 표시와 상세 표시가 다릅니다. 상세 표시에서는 유니트 버전, 하드웨어 버전 및 소프트웨어 버전이 표시됩니다. 간단 표시에서는 유니트 버전만 표시됩니다.

(주) 하드웨어 개정은 하드웨어 버전 오른쪽 끝에 「/」로 구분하여 표시됩니다. 하드웨어가 개정되지 않은 유니트에는 하드웨어 개정 표시가 없습니다.

● EtherCAT 슬레이브의 버전 확인 방법

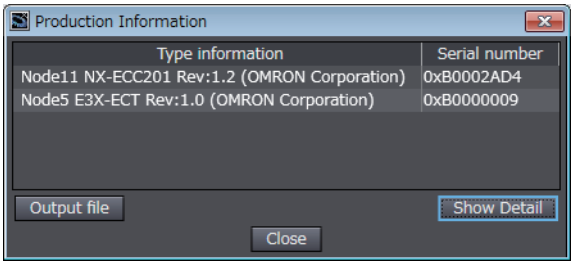
EtherCAT 슬레이브의 버전은 온라인 상태의 [Production Information]에서 확인할 수 있습니다. 확인 방법은 아래와 같습니다.

1 **멀티 뷰 익스플로러에서 [Configurations and Setup]-[EtherCAT]을 더블 클릭합니다. 또는 [Configurations and Setup]-[EtherCAT]을 우클릭하고 [Edit]를 선택합니다.**
EtherCAT 구성 편집 화면이 표시됩니다.

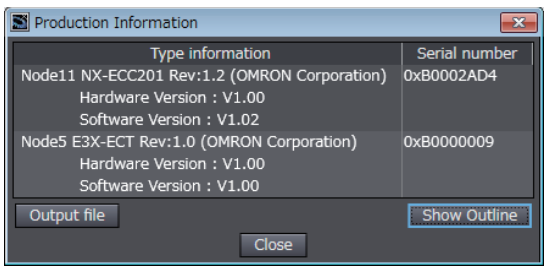
2 **EtherCAT 구성 편집 화면에서 마스터를 우클릭하고 [Display Production Information]을 선택합니다.**
생산 정보 대화 상자가 표시됩니다.
유니트 버전은 「Rev」를 붙여 표시됩니다.

● 생산 정보 표시 내용의 변환

[Production Information] 대화 상자 우측 아래에 있는 [Show Outline] 또는 [Show Detail]을 선택합니다. [Production Information]의 간단 표시와 상세 표시가 변환됩니다.



간단 표시



상세 표시

CPU 유니트의 유니트 버전과 Sysmac Studio의 버전

NJ/NX 시리즈 CPU 유니트의 버전에 따라 탑재된 기능이 다릅니다. 업그레이드로 추가된 기능을 사용할 때는 대응하는 버전의 Sysmac Studio를 사용해야 합니다.

CPU 유니트의 유니트 버전 종류와 Sysmac Studio 버전과의 관계에 대해서는 🕮『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

유니트 버전에 따른 지원 기능 일람에 대해서는 🕮「A-7 버전 관련 정보 (P.A-34)」를 참조해 주십시오.

관련 매뉴얼

본 매뉴얼과 관련된 매뉴얼은 아래 표와 같습니다. 함께 참조해 주십시오.

매뉴얼 명칭	Man.No.	형식	용도	내용
NX 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편	SBCA-418	NX701-□□□□	NX701 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고 싶을 때. 주로 하드웨어에 관한 정보.	NX701의 시스템 전체 개요 및 CPU 유니트에 관하여 아래의 내용을 설명합니다. ・특징 및 시스템 구성 ・개요 ・각부의 명칭과 기능 ・일반 사양 ・설치와 배선 ・보수 점검
NX 시리즈 NX102 CPU 유니트 사용자 매뉴얼 하드웨어편	SBCA-462	NX102-□□□□	NX102 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고 싶을 때. 주로 하드웨어에 관한 정보.	NX102의 시스템 전체 개요 및 CPU 유니트에 관하여 아래의 내용을 설명합니다. ・특징 및 시스템 구성 ・개요 ・각부의 명칭과 기능 ・일반 사양 ・설치와 배선 ・보수 점검
NX 시리즈 NX1P2 CPU 유니트 사용자 매뉴얼 하드웨어편	SBCA-448	NX1P2-□□□□	NX1P2 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고 싶을 때. 주로 하드웨어에 관한 정보.	NX1P2의 시스템 전체 개요 및 CPU 유니트에 관하여 아래의 내용을 설명합니다. ・특징 및 시스템 구성 ・개요 ・각부의 명칭과 기능 ・일반 사양 ・설치와 배선 ・보수 점검
NJ 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편	SBCA-358	NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ 시리즈 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고 싶을 때. 주로 하드웨어에 관한 정보.	NJ 시리즈의 시스템 전체 개요 및 CPU 유니트에 관하여 아래의 내용을 설명합니다. ・특징 및 시스템 구성 ・개요 ・각부의 명칭과 기능 ・일반 사양 ・설치와 배선 ・보수 점검
NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편	SBCA-467	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈 CPU 유니트의 프로그래밍/시스템 기동에 대해 알고 싶을 때. 주로 소프트웨어에 관한 정보.	NJ/NX 시리즈 CPU 유니트에 관하여 아래의 내용을 설명합니다. ・CPU 유니트의 동작 ・CPU 유니트의 기능 ・초기 설정 ・IEC 61131-3 베이스의 언어 사양과 프로그래밍
NX 시리즈 NX1P2 CPU 유니트 사용자 매뉴얼 내장 I/O, 옵션 보드 기능편	SBCA-449	NX1P2-□□□□	NX 시리즈 NX1P2 CPU 유니트에만 탑재된 기능의 상세한 내용과 NJ/NX 시리즈의 기능에 관한 개요를 알고 싶을 때.	NX1P2 CPU 유니트의 기능 중 아래의 내용을 설명합니다. ・내장 I/O ・시리얼 통신용 옵션 보드 ・아날로그 입출력용 옵션 보드 이와 함께 아래에 제시한 NJ/NX 시리즈 CPU 유니트의 기능에 관한 개요를 설명합니다. ・모션 제어 기능 ・EtherNet/IP 통신 기능 ・EtherCAT 통신 기능
NJ/NX 시리즈 명령 참조 매뉴얼 기본편	SBCA-360	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈의 기본 명령 사양의 상세한 내용에 대해 알고 싶을 때.	각 명령(IEC 61131-3 사양)에 관한 상세한 내용을 설명합니다.

매뉴얼 명칭	Man.No.	형식	용도	내용
NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 모션 제어편	SBCE-363	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	모션 제어의 설정 및 프로그래밍의 개념에 대해 알고 싶을 때.	모션 제어를 위한 CPU 유니트의 설정 및 동작, 프로그래밍의 개념에 대해 설명합니다.
NJ/NX 시리즈 명령 참조 매뉴얼 모션편	SBCE-364	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	모션 명령 사양의 상세한 내용에 대해 알고 싶을 때.	각 모션 명령의 상세한 내용을 설명합니다.
NJ/NX 시리즈 CPU 유니트 내장 EtherCAT® 포트 사용자 매뉴얼	SBCD-358	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈 CPU 유니트의 내장 EtherCAT 포트를 사용할 때.	내장 EtherCAT 포트에 관해 설명합니다. 개요, 구성, 기능, 셋업에 대해 기술되어 있습니다.
NJ/NX 시리즈 CPU 유니트 내장 EtherNet/IP™ 포트 사용자 매뉴얼	SBCD-359	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈 CPU 유니트의 내장 EtherNet/IP 포트를 사용할 때.	내장 EtherNet/IP 포트에 관해 설명합니다. 기본 설정, 태그 데이터 링크, 기타 기능에 대해 기술되어 있습니다.
NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 OPC UA편	SBCD-374	NX102-□□□□ NJ501-1□00	OPC UA를 사용할 때.	OPC UA에 관해 설명합니다.
NX 시리즈 CPU 유니트 사용자 매뉴얼 FINS 기능편	SBCD-375	NX701-□□20 NX102-□□□□	NX 시리즈 CPU 유니트의 FINS 기능을 사용할 때.	NX 시리즈 CPU 유니트의 FINS 기능에 대해 설명합니다.
NJ/NX 시리즈 데이터베이스 접속 CPU 유니트 사용자 매뉴얼	SBCA-411	NX701-□□20 NJ501-□□20 NJ101-□□20	NJ/NX 시리즈에서 데이터베이스 접속 서비스 기능을 사용할 때.	데이터베이스 접속 서비스 기능에 대해 설명합니다.
NJ 시리즈 SECS/GEM 탑재 CPU 유니트 사용자 매뉴얼	SBCA-412	NJ501-1340	NJ 시리즈에서 GEM 서비스 기능을 사용할 때.	GEM 서비스 기능에 대해 설명합니다.
NJ 시리즈 NJ Robotics CPU 유니트 사용자 매뉴얼	SBCA-421	NJ501-4□□□	NJ 시리즈에서 로봇 제어를 실행할 때.	로봇 제어 기능에 대해 설명합니다.
NJ/NY 시리즈 NC 통합 컨트롤러 사용자 매뉴얼	SBCE-428	NJ501-5300 NY532-5400	NJ/NY 시리즈에서 수치 제어를 실행할 때.	수치 제어에 대해 설명합니다.
NJ/NY 시리즈 명령 참조 매뉴얼 G 코드편	SBCE-429	NJ501-5300 NY532-5400	G 코드/M 코드 명령 사양의 상세한 내용에 대해 알고 싶을 때.	G 코드/M 코드 명령의 상세한 내용을 설명합니다.
NJ/NX 시리즈 트러블 슈팅 매뉴얼	SBCA-361	NX701-□□□□ NX102-□□□□ NX1P2-□□□□ NJ501-□□□□ NJ301-□□□□ NJ101-□□□□	NJ/NX 시리즈에서 검출하는 이상에 대한 상세한 내용에 대해 알고 싶을 때.	NJ/NX 시리즈 시스템에서 검출하는 이상 관리의 개념과 각 이상 항목에 대해 설명합니다.
Sysmac Studio Version 1 오퍼레이션 매뉴얼	SBCA-470	SYSMAC- SE2□□□	Sysmac Studio의 조작 방법, 기능에 대해 알고 싶을 때.	Sysmac Studio의 조작 방법에 대해 설명합니다.
CNC Operator 오퍼레이션 매뉴얼	SBCE-430	SYSMAC -RTNC0□□□D	CNC Operator의 개요, 사용 방법에 대해 알고 싶을 때.	CNC Operator의 개요, 설치 방법, 기본 조작, 접속 조작, 주요 기능의 조작 방법 등에 대해 설명합니다.

매뉴얼 명칭	Man.No.	형식	용도	내용
NX 시리즈 EtherCAT® 커플러 유니트 사용자 매뉴얼	SBCD-361	NX-ECC□□□	NX 시리즈 EtherCAT 커플러 유니트 및 EtherCAT 슬레이브 터미널의 사용 방법에 대해 알고 싶을 때.	NX 시리즈 EtherCAT 커플러 유니트와 NX 유니트로 구성된 EtherCAT 슬레이브 터미널의 시스템 개요 및 구성 방법, EtherCAT을 통해 NX 유니트를 설정, 제어, 모니터하기 위한 EtherCAT 커플러 유니트의 하드웨어와 설정 방법, 기능에 대해 설명합니다.
NX 시리즈 데이터 참조 매뉴얼	SBCA-410	NX-□□□□□□	NX 시리즈의 각 유니트의 시스템 구성에 필요한 데이터를 일람에서 열람하고 싶을 때.	NX 시리즈의 각 유니트의「소비 전력」,「무게」 등 시스템 구축에 필요한 데이터를 모아 기재했습니다.
NX 시리즈 NX 유니트 사용자 매뉴얼	SBCA-407	NX-ID□□□□ NX-IA□□□□ NX-OC□□□□ NX-OD□□□□ NX-MD□□□□	NX 유니트의 사용 방법에 대해 알고 싶을 때.	NX 유니트의 하드웨어와 설정 방법, 기능에 대해 설명합니다. 아래 유니트의 매뉴얼이 있습니다. 디지털 I/O 유니트, 아날로그 I/O 유니트, 시스템 유니트, 위치 인터페이스 유니트, 커뮤니케이션 인터페이스 유니트, 로드셀 입력 유니트, IO-Link 마스터 유니트
	SBCA-408	NX-AD□□□□ NX-DA□□□□		
	SBCA-440	NX-TS□□□□ NX-HB□□□□		
	SBCA-409	NX-PD1□□□ NX-PF0□□□ NX-PC0□□□ NX-TBX01		
	SBCE-374	NX-EC0□□□ NX-ECS□□□ NX-PG0□□□		
	SBCA-422	NX-CIF□□□		
	SBCA-439	NX-RS□□□□		
	SBCD-370	NX-ILM□□□		
EtherCAT 리모트 I/O 터미널 GX 시리즈 EtherCAT 슬레이브 사용자 매뉴얼	SBCD-350	GX-ID□□□□ GX-OD□□□□ GX-OC□□□□ GX-MD□□□□ GX-AD□□□□ GX-DA□□□□ GX-EC□□□□ XWT-ID□□ XWT-OD□□	EtherCAT 리모트 I/O 터미널의 사용 방법에 대해 알고 싶을 때.	EtherCAT 리모트 I/O 터미널의 하드웨어와 설정 방법, 기능에 대해 설명합니다.
AC 서보 모터/드라이버 1S 시리즈 EtherCAT® 통신 내장 타입 사용자 매뉴얼	SBCE-377	R88M-1□ R88D-1SN□-ECT	EtherCAT 통신 내장 타입의 서보 모터/드라이버의 사용 방법에 대해 알고 싶을 때.	EtherCAT 통신 내장 타입의 서보 모터/드라이버의 하드웨어와 설정 방법, 기능에 대해 설명합니다.
AC 서보 모터/드라이버 G5 시리즈 EtherCAT® 통신 내장 타입 사용자 매뉴얼	SBCE-360	R88M-K□ R88D-KN□-ECT-R	EtherCAT 통신 내장 타입의 서보 모터/드라이버의 사용 방법에 대해 알고 싶을 때.	EtherCAT 통신 내장 타입의 서보 모터/드라이버의 하드웨어와 설정 방법, 기능에 대해 설명합니다. G5 시리즈에는 리니어 모터 타입과 위치 제어 한정 모델이 있습니다.
	SBCE-365	R88M-K□ R88D-KN□-ECT		
	SBCE-366	R88L-EC-□ R88D-KN□-ECT-L		

매뉴얼 개정 이력

매뉴얼 개정 기호는 앞표지 및 뒤표지에 기재되어 있는 Man.No.에 부기됩니다.



개정 기호	개정 연월	개정 이유
A	2011년 7월	초판
B	2012년 1월	NJ301-□□□□ 추가 CPU 유니트 Ver.1.01 업그레이드에 따른 개정
C	2012년 5월	CPU 유니트 Ver.1.02 업그레이드에 따른 개정 오기 정정
D	2012년 8월	CPU 유니트 Ver.1.03 업그레이드에 따른 개정
E	2013년 2월	CPU 유니트 Ver.1.04 업그레이드에 따른 개정
F	2013년 4월	CPU 유니트 Ver.1.05 업그레이드에 따른 개정 오기 정정
G	2013년 6월	CPU 유니트 Ver.1.06 업그레이드에 따른 개정 오기 정정
H	2013년 12월	CPU 유니트 Ver.1.08 업그레이드에 따른 개정 오기 정정
J	2014년 7월	CPU 유니트 Ver.1.09 업그레이드에 따른 개정 오기 정정
K	2015년 1월	CPU 유니트 Ver.1.10 업그레이드에 따른 개정 오기 정정
L	2015년 4월	NX 시리즈 CPU 유니트 NX701-□□□□ 및 NJ 시리즈 CPU 유니트 NJ101-10□□의 추가에 따른 개정 오기 정정
M	2016년 4월	CPU 유니트 Ver.1.11 업그레이드에 따른 개정 오기 정정
N	2016년 7월	CPU 유니트 Ver.1.12 업그레이드에 따른 개정 오기 정정
P	2016년 10월	NX 시리즈 NX1P2 CPU 유니트의 추가에 따른 개정 CPU 유니트 Ver.1.13 업그레이드에 따른 개정 오기 정정
R	2017년 1월	오기 정정에 의한 개정
S	2017년 10월	오기 정정에 의한 개정
T	2018년 4월	NX 시리즈 NX102 CPU 유니트의 추가에 따른 개정 이벤트 코드 및 이상에 관한 설명을『NJ/NX 시리즈 트러블 슈팅 매뉴얼』로 이동함에 따른 개정 오기 정정

본 매뉴얼은 다음 개정부터 Man.No.가 아래와 같이 변경될 예정입니다. 최신 매뉴얼 입수 시에 주의해 주십시오.

현재 Man.No.	변경 후의 Man.No.
SBCE-363	SBCE-433

1

모션 제어 기능 모듈의 개요

모션 제어 기능 모듈의 특징과 시스템 구성, 사용 플로우에 대해 설명합니다.

1-1 특징 1-2
1-2 시스템 구성 1-3
1-3 기본적인 사용 흐름도 1-5
1-4 사양 1-7
1-4-1 일반 사양 1-7
1-4-2 성능 사양 1-7
1-4-3 기능 사양 1-11

1-1 특징

모션 제어 기능 모듈(이하,「MC 기능 모듈」로 생략하는 경우가 있습니다)이란 CPU 유니트에 내장된 소프트웨어의 기능 모듈을 나타냅니다.
MC 기능 모듈은 CPU 유니트의 내장 EtherCAT 포트를 통해 최대 256축의 모션 제어를 실행할 수 있습니다. EtherCAT 포트에 접속된 서보 드라이버 등과 사이클릭 통신을 실행하면 고속, 고정밀도가 요구되는 머신 제어를 실현할 수 있습니다.

PLCopen®에 준거한 모션 제어 명령

MC 기능 모듈의 모션 제어 명령은 PLCopen®으로 표준화된 모션 제어용 펑션 블록을 기본으로 합니다. 단축에서 PTP 위치를 결정하거나 보간 제어, 전자 캠 등의 동기 제어, 속도 제어, 토크 제어를 표준에 준거한 명령으로 프로그래밍할 수 있습니다.
또한, 모션 제어 명령의 기동별로 속도, 가속도, 감속도 및 저크를 설정할 수 있어 어플리케이션에 맞춰 유연하게 동작을 제어할 수 있습니다.



참고

PLCopen®란

PLCopen®는 유럽에 본부를 둔 IEC 61131-3의 보급 단체로, 전 세계에 회원을 둔 글로벌 조직입니다. PLCopen®에서는 모션 제어용 펑션 블록을 표준화하여 IEC 61131-3 (JIS B 3503) 규격인 언어를 이용한 프로그램 인터페이스를 정의하고 있습니다.

저크란

저크는 가속도와 감속도가 변화하는 비율로「약동」,「약도」 혹은「가가속도」라고도 합니다. 저크를 지정하면 가감속 시의 속도 파형이 S자형이 됩니다.

EtherCAT 통신을 이용한 데이터 전송

오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입 또는 G5 시리즈 EtherCAT 통신 내장 타입과 조합하면 모든 제어 정보를 고속 데이터 통신을 이용하여 교환할 수 있습니다.
각종 제어 지령이 데이터 통신으로 전달되어 엔코더 피드백 펄스의 응답 주파수 등 인터페이스의 사양에 제약을 받지 않고 서보 모터의 성능을 최대한 끌어낼 수 있습니다.
또한, 서보 드라이버의 각종 제어 파라메터와 모니터 정보를 상위 컨트롤러 측에서 다룰 수 있게 되어 시스템 정보를 통합하여 관리할 수 있습니다.



참고

EtherCAT이란

EtherCAT은 Ethernet(IEEE802.3)에 준거하는 오픈된 초고속 산업용 네트워크 시스템입니다. 각 노드는 Ethernet 프레임을 고속으로 전송하기 때문에 짧은 통신 사이클 타임을 실현할 수 있습니다. 또한, 시계 정보를 공유하는 구조이므로 낮은 통신 지터로 고정밀도 동기 제어가 가능합니다.

1-2 시스템 구성

MC 기능 모듈은 장비나 제어반에서 보내는 센서 신호의 상태와 사용자 프로그램의 모션 제어 명령이 보내는 지령을 받아 서보 드라이버, 엔코더 입력 터미널 및 NX 시리즈 위치 인터페이스 유니트의 모션 제어를 실시합니다.

EtherCAT 네트워크에서의 모션 제어 구성

MC 기능 모듈에서는 EtherCAT 네트워크 구성, EtherCAT 커플러 유니트의 슬레이브 터미널 구성 및 Sysmac Studio를 사용합니다.

● EtherCAT 네트워크 구성

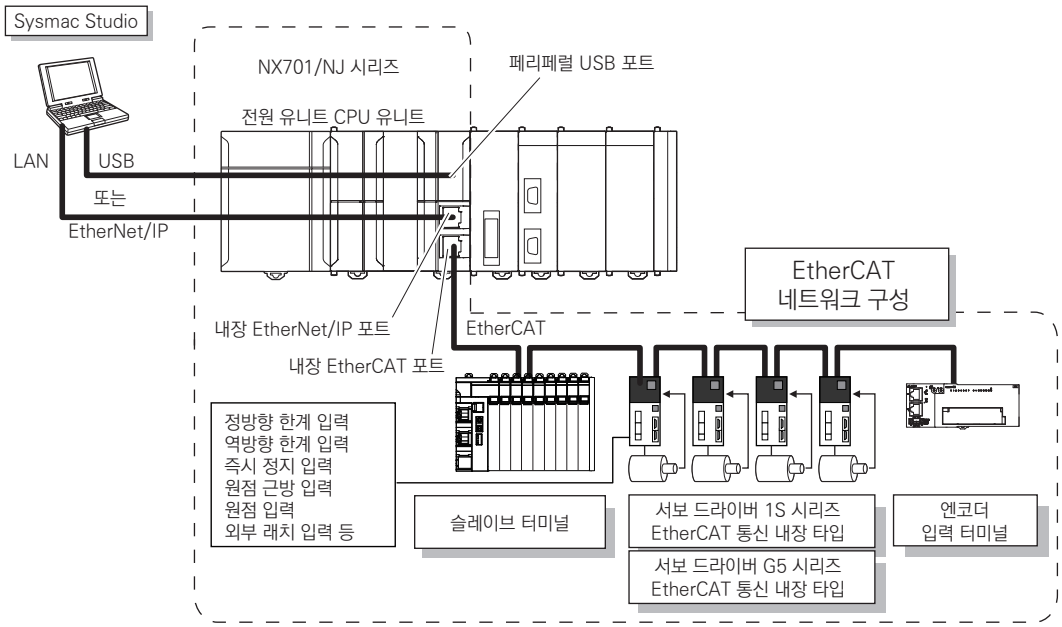
MC 기능 모듈은 CPU 유니트에 내장된 EtherCAT 통신 마스터 기능 포트를 사용하고 서보 드라이버, 엔코더 입력 터미널을 제어합니다.
이 구성을 사용하면 정주기로 편차가 작은 고정밀도 모션 제어를 실현합니다.

● EtherCAT 커플러 유니트의 슬레이브 터미널 구성

MC 기능 모듈은 EtherCAT 커플러 유니트에 장착한 위치 인터페이스 유니트를 사용하여 모터 제어용 펄스를 출력하거나 엔코더 입력을 읽어들입니다.
이 구성을 사용하면 정주기로 편차가 작은 고정밀도 모션 제어를 실현합니다.

● Sysmac Studio

CPU 유니트의 페리페럴 USB 포트에 시판 USB 케이블을 접속합니다. 또는 CPU 유니트 내장 EtherNet/IP 포트에 Ethernet 케이블로 접속합니다.



사용상의 주의

NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 모션 제어 기능의 일부가 다릅니다.
상세한 내용은 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

버전 관련 정보

NX 시리즈 위치 인터페이스 유니트는 유니트 버전 Ver.1.05 이상인 CPU 유니트와 Sysmac Studio Ver.1.06 이상의 조합으로 사용할 수 있습니다.

CPU 유니트상에서의 모션 제어 구성

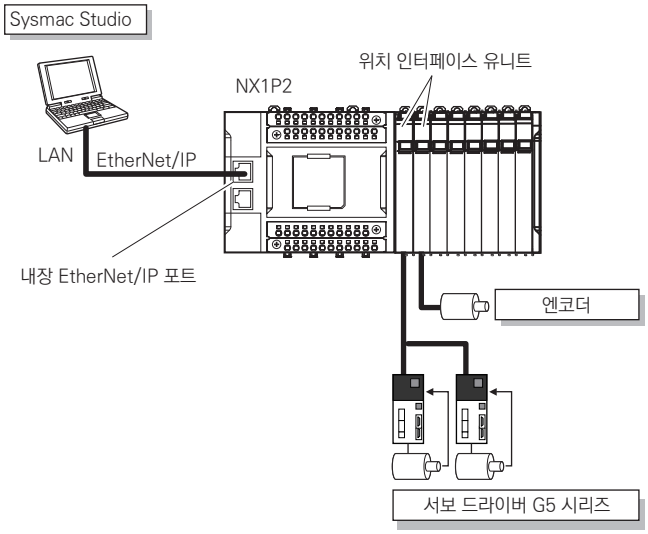
MC 기능 모듈에서는 위치 인터페이스 유니트 및 Sysmac Studio를 사용합니다.

● 위치 인터페이스 유니트

위치 인터페이스 유니트를 사용하여 모터 제어용 펄스를 출력하거나 엔코더 입력을 읽어들입니다.

● Sysmac Studio

CPU 유니트 내장 EtherNet/IP 포트에 Ethernet 케이블로 접속합니다.

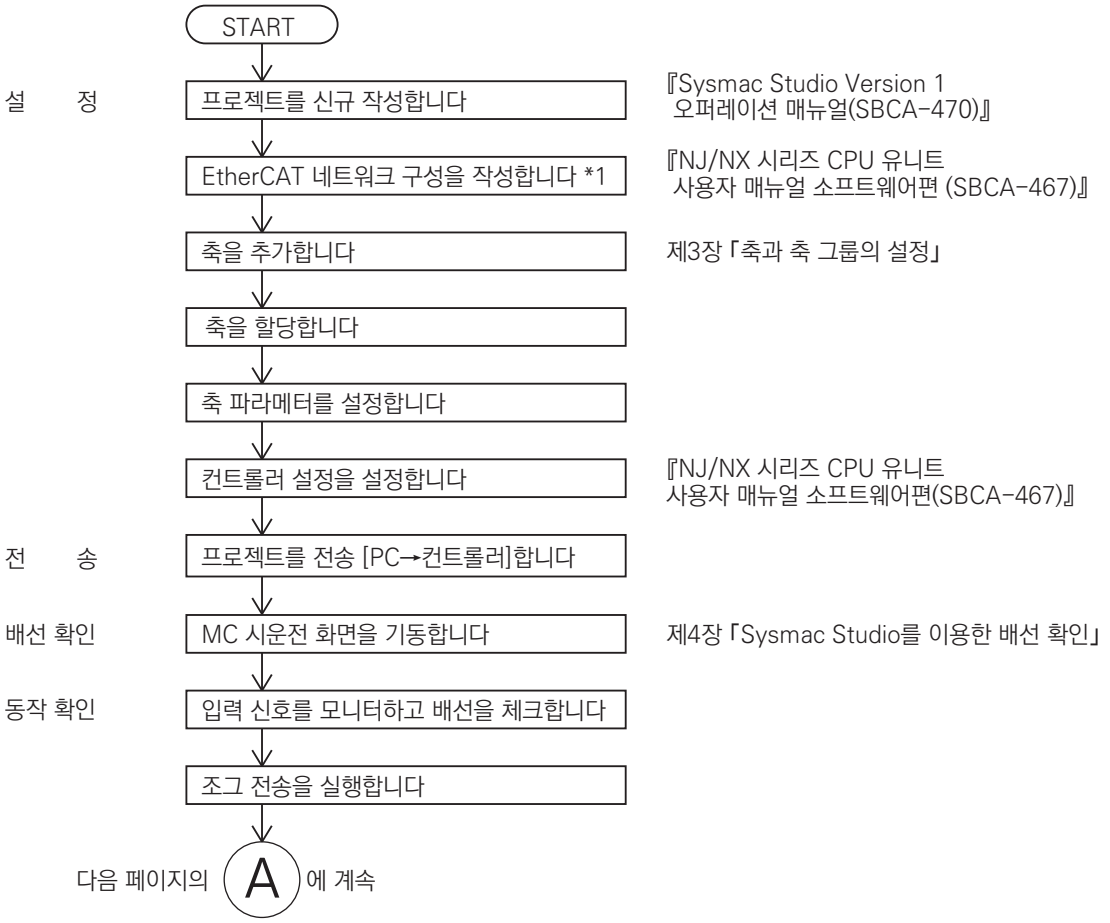


사용상의 주의

NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 모션 제어 기능의 일부가 다릅니다. 상세한 내용은 『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

1-3 기본적인 사용 흐름도

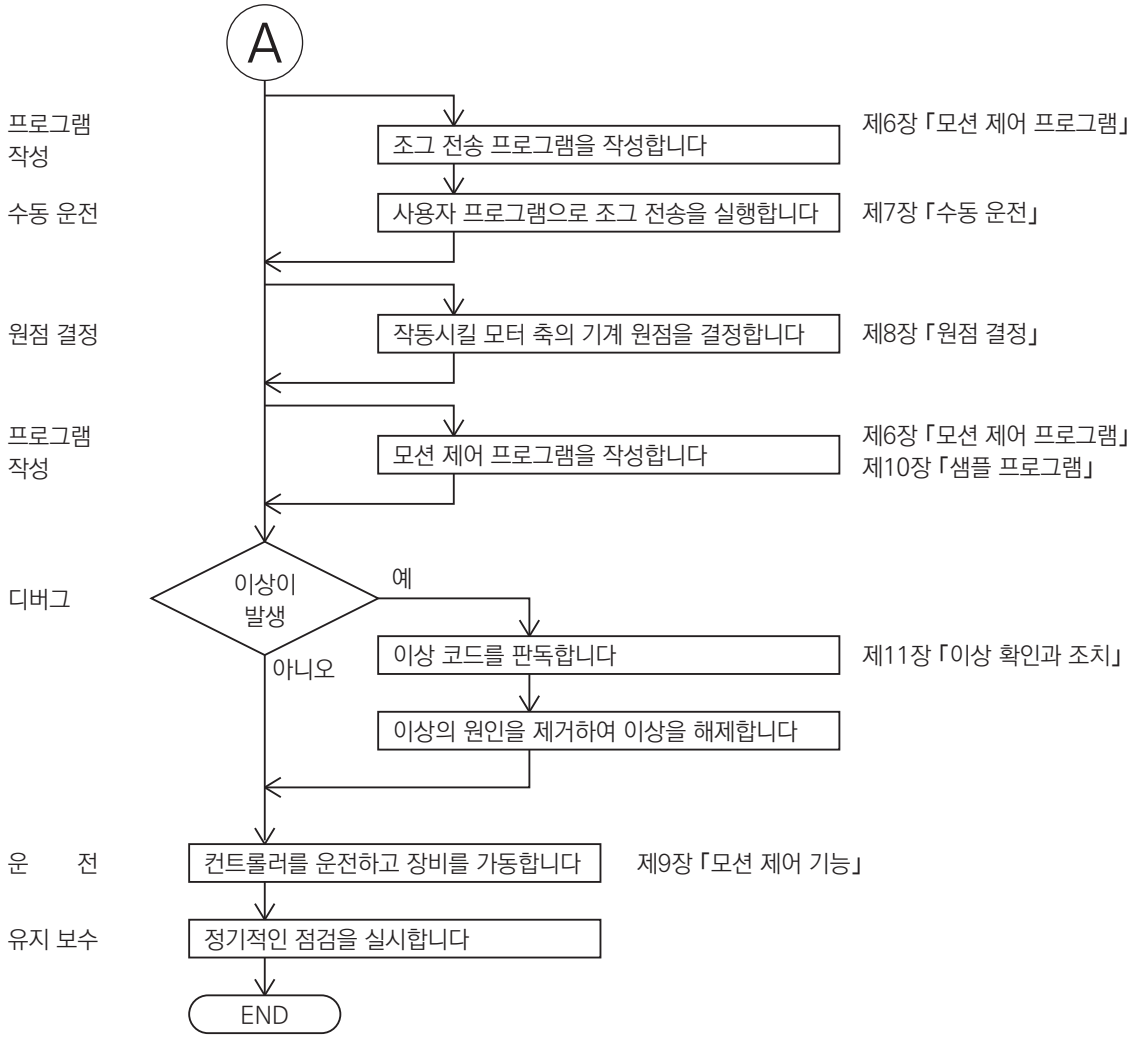
여기에서는 MC 기능 모듈에서 모션 제어를 실행하기 위한 기본적인 사용 순서에 대해 설명합니다.



*1. 실제 기기를 접속한 경우에는 온라인으로 실기 구성을 설정할 수 있습니다.
실제 기기가 없는 경우에는 오프라인에서 사용하는 EtherCAT 슬레이브를 선택하여 설정할 수 있습니다.

참고

NX 시리즈 위치 인터페이스 유니트를 사용하는 경우의 사용 순서는 『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.


A
프로그램 작성
조그 전송 프로그램을 작성합니다
제6장「모션 제어 프로그램」
수동 운전
사용자 프로그램으로 조그 전송을 실행합니다
제7장「수동 운전」
원점 결정
작동시킬 모터 축의 기계 원점을 결정합니다
제8장「원점 결정」
프로그램 작성
모션 제어 프로그램을 작성합니다
제6장「모션 제어 프로그램」
제10장「샘플 프로그램」
디버그
이상이 발생
예
아니오
이상 코드를 판독합니다
제11장「이상 확인과 조치」
이상의 원인을 제거하여 이상을 해제합니다
운 전
컨트롤러를 운전하고 장비를 가동합니다
제9장「모션 제어 기능」
유지 보수
정기적인 점검을 실시합니다
END

1-4 사양

여기에서는 MC 기능 모듈의 사양에 대해 설명합니다.

사용상의 주의

NX102-90□□ 및 NJ101-90□□는 모션 제어 기능 모듈이 없습니다.

1-4-1 일반 사양

일반 사양은 CPU 유니트의 일반 사양에 준합니다.
상세한 내용은 🕮『NX 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-418)』 또는 🕮『NJ 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-358)』을 참조해 주십시오.

1-4-2 성능 사양

CPU 유니트별 성능 사양에 대해 설명합니다.

사양 항목			사양(NX701-)	
			17□□	16□□
제어 축 수	제어 축 최대 수 *1		256축	128축
		모션 제어 축 *2	256축	128축
		단축 위치 제어 축 *3	—	
	사용 실축(Real axis) 최대 수 *4		256축	128축
		사용 모션 제어 서보 축	256축	128축
		사용 단축 위치 제어 서보 축	—	
	직선 보간 제어 최대 축 수		1축 그룹당 4축	
	원호 보간 제어 축 수		1축 그룹당 2축	
축 그룹 최대 수			64그룹	
오버라이드			0.00, 0.01~500.00%	
모션 제어 주기			EtherCAT 통신의 PDO 통신 주기와 동일	
멀티 모션			가능	
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점	
		전체 캠 테이블의 최대 점수	1,048,560점	
	캠 테이블 최대 테이블 수		640테이블	

*1. 모든 축 종별을 합한 값입니다. 축 종별에 대한 상세한 내용은 🕮「축 종별 (P.5-8)」을 참조해 주십시오.
*2. 모션 제어 기능의 모든 기능을 사용할 수 있는 축을 말합니다.
*3. 모션 제어 기능 중「단축 위치 제어」만 사용할 수 있는 축을 말합니다. 단축 위치 제어 축에 대한 상세한 내용은 🕮「제어 기능 (P.5-9)」을 참조해 주십시오.
*4. 축 종별이 [Servo Axis] 또는 [Encoder Axis]이며, 축 사용이 [Used Axis]로 설정된 축의 합계입니다.

사양 항목			사양(NX102-)			
			12□□	11□□	10□□	90□□
제어 축 수	제어 축 최대 수 *1		15축			4축
		모션 제어 축 *2	11축			—
		단축 위치 제어 축 *3	4축			
	사용 실축 최대 수 *4		12축	8축	6축	4축
		사용 모션 제어 서보 축	8축	4축	2축	—
		사용 단축 위치 제어 서보 축	4축			
	직선 보간 제어 최대 축 수		1축 그룹당 4축			—
	원호 보간 제어 축 수		1축 그룹당 2축			—
축 그룹 최대 수			8그룹			—
오버라이드			0.00, 0.01~500.00%			
모션 제어 주기			프라이머리 정주기 태스크의 주기와 동일			
멀티 모션			불가			
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점			
		전체 캠 테이블의 최대 점수	262,140점			
	캠 테이블 최대 테이블 수		160테이블			

*1. 모든 축 종별을 합한 값입니다. 축 종별에 대한 상세한 내용은 「축 종별 (P.5-8)」을 참조해 주십시오.
*2. 모션 제어 기능의 모든 기능을 사용할 수 있는 축을 말합니다.
*3. 모션 제어 기능 중「단축 위치 제어」만 사용할 수 있는 축을 말합니다. 단축 위치 제어 축에 대한 상세한 내용은 「제어 기능 (P.5-9)」을 참조해 주십시오.
*4. 축 종별이 [Servo Axis] 또는 [Encoder Axis]이며, 축 사용이 [Used Axis]로 설정된 축의 합계입니다.

사양 항목			사양(NX1P2-)		
			11□□	10□□	90□□
제어 축 수	제어 축 최대 수 *1		12축	10축	4축
		모션 제어 축 *2	8축	6축	—
		단축 위치 제어 축 *3	4축		
	사용 실축 최대 수 *4		8축	6축	4축
		사용 모션 제어 서보 축	4축	2축	—
		사용 단축 위치 제어 서보 축	4축		
	직선 보간 제어 최대 축 수		1축 그룹당 4축		—
	원호 보간 제어 축 수		1축 그룹당 2축		—
축 그룹 최대 수			8그룹		—
오버라이드			0.00, 0.01~500.00%		
모션 제어 주기			프라이머리 정주기 태스크의 주기와 동일		
멀티 모션			불가		
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점		—
		전체 캠 테이블의 최대 점수	262,140점		—
	캠 테이블 최대 테이블 수		80테이블		—

*1. 모든 축 종별을 합한 값입니다. 축 종별에 대한 상세한 내용은 「축 종별 (P.5-8)」을 참조해 주십시오.
*2. 모션 제어 기능의 모든 기능을 사용할 수 있는 축을 말합니다.
*3. 모션 제어 기능 중「단축 위치 제어」만 사용할 수 있는 축을 말합니다. 단축 위치 제어 축에 대한 상세한 내용은 「제어 기능 (P.5-9)」을 참조해 주십시오.
*4. 축 종별이 [Servo Axis] 또는 [Encoder Axis]이며, 축 사용이 [Used Axis]로 설정된 축의 합계입니다.

사양 항목			사양(NJ501-)		
			□5□□	□4□□	□3□□
제어 축 수	제어 축 최대 수 *1		64축	32축	16축
		모션 제어 축 *2	64축	32축	16축
		단축 위치 제어 축 *3	—		
	사용 실축 최대 수 *4		64축	32축	16축
		사용 모션 제어 서보 축	64축	32축	16축
		사용 단축 위치 제어 서보 축	—		
	직선 보간 제어 최대 축 수		1축 그룹당 4축		
	원호 보간 제어 축 수		1축 그룹당 2축		
축 그룹 최대 수			32그룹		
오버라이드			0.00, 0.01~500.00%		
모션 제어 주기			EtherCAT 통신의 PDO 통신 주기와 동일		
멀티 모션			불가		
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점		
		전체 캠 테이블의 최대 점수	1,048,560점		
	캠 테이블 최대 테이블 수		640테이블		

*1. 모든 축 종별을 합한 값입니다. 축 종별에 대한 상세한 내용은 「축 종별 (P.5-8)」을 참조해 주십시오.
*2. 모션 제어 기능의 모든 기능을 사용할 수 있는 축을 말합니다.
*3. 모션 제어 기능 중「단축 위치 제어」만 사용할 수 있는 축을 말합니다. 단축 위치 제어 축에 대한 상세한 내용은 「제어 기능 (P.5-9)」을 참조해 주십시오.
*4. 축 종별이 [Servo Axis] 또는 [Encoder Axis]이며, 축 사용이 [Used Axis]로 설정된 축의 합계입니다.

사양 항목			사양(NJ301-)		사양(NJ101-)
			12□□	11□□	10□□
제어 축 수	제어 축 최대 수 *1		15축 *2 *3	15축 *2 *4	6축
		모션 제어 축 *5	15축 *2 *3	15축 *2 *4	6축
		단축 위치 제어 축 *6	—		
	사용 실축 최대 수 *7		8축	4축	2축
		사용 모션 제어 서보 축	8축	4축	2축
		사용 단축 위치 제어 서보 축	—		
	직선 보간 제어 최대 축 수		1축 그룹당 4축		
	원호 보간 제어 축 수		1축 그룹당 2축		
축 그룹 최대 수			32그룹		
오버라이드			0.00, 0.01~500.00%		
모션 제어 주기			EtherCAT 통신의 PDO 통신 주기와 동일		
멀티 모션			불가		
캠	캠 데이터 점수	1캠 테이블당 최대 점수	65,535점		
		전체 캠 테이블의 최대 점수	262,140점		
	캠 테이블 최대 테이블 수		160테이블		

*1. 모든 축 종별을 합한 값입니다. 축 종별에 대한 상세한 내용은 「축 종별 (P.5-8)」을 참조해 주십시오.
*2. CPU 유니트의 버전 업그레이드로 추가된 기능입니다. 버전 업그레이드에 관한 정보는 「A-7 버전 관련 정보 (P.A-34)」를 참조해 주십시오.
*3. 유니트 버전 Ver.1.06 이상인 CPU 유니트와 Sysmac Studio Ver.1.07 이상을 조합한 경우의 축 수입니다. 그 이외의 경우는 8축입니다.
*4. 유니트 버전 Ver.1.06 이상인 CPU 유니트와 Sysmac Studio Ver.1.07 이상을 조합한 경우의 축 수입니다. 그 이외의 경우는 4축입니다.
*5. 모션 제어 기능의 모든 기능을 사용할 수 있는 축을 말합니다.
*6. 모션 제어 기능 중「단축 위치 제어」만 사용할 수 있는 축을 말합니다. 단축 위치 제어 축에 대한 상세한 내용은 「제어 기능 (P.5-9)」을 참조해 주십시오.
*7. 축 종별이 [Servo Axis] 또는 [Encoder Axis]이며, 축 사용이 [Used Axis]로 설정된 축의 합계입니다.

1-4-3 기능 사양

오므론 제어 대상 기기와 접속하는 경우의 기능을 아래에 설명합니다.

기능 항목			내용
제어 대상 서보 드라이버			오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입 또는 G5 시리즈 EtherCAT 통신 내장 타입 *1 *2
제어 대상 엔코더 입력 터미널			오므론 EtherCAT 리모트 I/O 터미널 GX 시리즈 GX-EC0211/EC0241 *3
제어 대상 위치 인터페이스 유니트 *2 *4			오므론 인크리멘털 엔코더 입력 유니트 NX-EC0□□□ 오므론 SSI 입력 유니트 NX-ECS□□□ 오므론 펄스 출력 유니트 NX-PG0□□□
제어 방식			EtherCAT 통신을 이용한 제어 명령 발행
제어 모드			위치 제어(cyclic synchronous position) 속도 제어(cyclic synchronous velocity) 토크 제어(cyclic synchronous torque)
단위 변환		표시 단위	펄스, mm, μm, nm, degree, inch
		전자 기어비	모터 1회전의 펄스 수/모터 1회전의 이동량 또는(모터 1회전의 펄스 수×모터 측 기어비)/(워크 측 1회전의 이동량×워크 측 기어비)
관리 가능한 위치			지령 위치, 피드백 위치
축 종별			서보 축, 가상 서보 축, 엔코더 축, 가상 엔코더 축
위치 지령			배정밀도 실수형(LREAL형)의 음수, 양수, 0(지령 단위 *5)
속도 지령			배정밀도 실수형(LREAL형)의 음수, 양수, 0(지령 단위/s)
가속도 지령, 감속도 지령			배정밀도 실수형(LREAL형)의 양수, 0(지령 단위/s^2)
저크 지령			배정밀도 실수형(LREAL형)의 양수, 0(지령 단위/s^3)
단축	단축 위치 제어	절대값 위치 결정	절대 좌표의 목표 위치를 지정하고 위치를 결정하는 기능
		상대값 위치 결정	지령 현재 위치에서의 이동 거리를 지정하고 위치를 결정하는 기능
		인터럽트 피딩 위치 결정	외부 입력에 의한 인터럽트 입력이 발생한 위치에서의 이동 거리를 지정하고 위치를 결정하는 기능
		사이클릭 동기 절대 위치 제어 *2	위치 제어 모드에서 제어 주기마다 지령 위치를 출력하는 기능
	단축 속도 제어	속도 제어	위치 제어 모드에서 속도를 제어하는 기능
		사이클릭 동기 속도 제어	속도 제어 모드에서 제어 주기마다 속도 지령을 출력하는 기능
	단축 토크 제어	토크 제어	모터의 토크를 제어하는 기능
	단축 동기 제어	캠 동작 시작	지정한 캠 테이블을 이용하여 캠 동작을 시작하는 기능
		캠 동작 해제	입력 파라메터로 지정한 축의 캠 동작을 종료하는 기능
		기어 동작 시작	주축과 종축 간의 기어비를 설정하여 기어 동작을 실행하는 기능
		위치 지정 기어 동작	주축과 종축 간의 기어비와 동기하는 위치를 설정하여 기어 동작을 실행하는 기능
		기어 동작 해제	실행 중인 기어 동작, 위치 지정 기어 동작을 중지하는 기능
		사다리꼴 패턴 캠	지정한 주축에 동기하여 위치를 결정하는 기능
		주축 상대값 위상 보정	동기 제어 중인 주축의 위상을 보정하는 기능
		가감산 위치 결정	2축의 지령 위치를 가산한 값 또는 감산한 값을 지령 위치로 출력하는 기능
	단축 수동 조작	운전 가능	서보 드라이버의 상태를 서보 ON 상태로 변환하여 축 동작을 가능하게 하는 기능
		조그 전송	지정한 목표 속도에 따라 조그를 전송하는 기능

기능 항목			내용
단축	단축 제어 보조	축 에러 리셋	축의 이상을 해제하는 기능
		원점 복귀	모터를 움직여 한계 신호, 원점 근방 신호, 원점 신호를 이용하여 기계 원점을 정하는 기능
		파라메터 지정 원점 복귀 *2	파라메터를 지정하여 모터를 움직여 한계 신호, 원점 근방 신호, 원점 신호를 이용하여 기계 원점을 정하는 기능
		고속 원점 복귀	절대 좌표인「0」을 목표 위치로 하여 위치를 결정하고 원점으로 되돌리는 기능
		강제 정지	축을 감속 정지시키는 기능
		즉시 정지	축을 즉시 정지시키는 기능
		오버라이드값 설정	축의 목표 속도를 변경하는 기능
		현재 위치 변경	축의 지령 현재 위치와 피드백 현재 위치를 임의의 값으로 변경하는 기능
		외부 래치 유효	트리거 발생을 통해 축의 위치를 기록하는 기능
		외부 래치 무효	실행 중인 래치를 무효화하는 기능
		구역 감시	축의 지령 위치 또는 피드백 현재 위치가 지정된 범위(구역) 내에 존재하는지를 판정하는 기능
		디지털 캠 스위치 유효 *2	축의 위치에 따라 디지털 출력을 ON 또는 OFF하는 기능
		축 간 편차 감시	지정한 2축의 지령 위치 또는 피드백 위치의 차분이 허용값을 초과하지 않는지를 감시하는 기능
		편차 카운터 리셋	지령 현재 위치와 피드백 현재 위치 간의 편차를 제로로 하는 기능
		토크 제한	서보 드라이버 토크 제한 기능의 유효/무효 변환과 토크 제한값을 설정하여 출력 토크를 제한하는 기능
		축 사용 변경 *2	축 파라메터 [Axis Use]를 일시적으로 변환하는 기능
		기동 속도 *2	축 동작이 시작될 때의 첫 속도를 설정하는 기능
축 그룹	다축 협조 제어	절대값 직선 보간	절대 위치를 지정하여 직선 보간을 실행하는 기능
		상대값 직선 보간	상대 위치를 지정하여 직선 보간을 실행하는 기능
		2축 원호 보간	2축의 원호 보간을 실행하는 기능
		축 그룹 사이클릭 동기 절대 위치 제어 *2	위치 제어 모드에서 제어 주기마다 지령 위치를 출력하는 기능
	다축 협조 제어 보조	축 그룹 에러 리셋	축 그룹 및 축의 이상을 해제하는 기능
		축 그룹 유효	축 그룹을 유효화하는 기능
		축 그룹 무효	축 그룹을 무효화하는 기능
		축 그룹 구성 축 기록 *2	축 그룹 파라메터 [Composition Axes]를 일시적으로 갱신하는 기능
		축 그룹 강제 정지	보간 동작 중인 모든 축을 감속 정지시키는 기능
		축 그룹 즉시 정지	보간 동작 중인 모든 축을 즉시 정지시키는 기능
		축 그룹 오버라이드값 설정	보간 동작 중인 합성 목표 속도를 변경하는 기능
		축 그룹 위치 취득 *2	축 그룹의 지령 현재 위치와 피드백 현재 위치를 취득하는 기능
공통	캠 *6	캠 테이블 속성(Property) 갱신	입력 파라메터로 지정한 캠 테이블의 종료점 인덱스를 갱신하는 기능
		캠 테이블 저장	입력 파라메터로 지정한 캠 테이블을 CPU 유니트에 내장된 불휘발성 메모리에 저장하는 기능
		캠 테이블 생성 *2	입력 파라메터로 지정한 캠 속성과 캠 노드에서 캠 테이블을 생성하는 기능
	파라메터	MC 설정 기록	축 파라메터 및 축 그룹 파라메터의 일부를 일시적으로 갱신하는 기능
		축 파라메터의 변경 *2	축 파라메터를 사용자 프로그램에서 참조 · 변경하는 기능

기능 항목			내용
보조 기능	카운트 모드		리니어 모드(유한 길이) 또는 로터리 모드(무한 길이) 선택 가능
	단위 변환		각 축의 표시 단위를 기계에 맞춰 설정 가능
	가감속 제어	자동 가감속 제어	축이나 축 그룹 동작 시의 가감속 커브를 저크로 설정하는 기능 (사다리꼴 커브, S자 커브)
		가감 속도 변경	가감속 동작 중에 가감 속도를 변경하는 기능
	인포지션 체크		위치 결정 완료를 체크하기 위한 인포지션 폭과 인포지션 체크 시간을 설정하는 기능
	정지 방법 선택		즉시 정지 입력 신호 및 한계 입력 신호가 유효해졌을 때 정지 방법을 설정하는 기능
	모션 제어 명령 재기동		실행 중인 모션 제어 명령의 입력 변수를 변경하여 재기동함으로써 동작 중에 목표값을 변경하는 기능
	모션 제어 명령의 다중 기동 (버퍼 모드)		동작 중에 다른 모션 제어 명령을 기동한 경우의 실행 시작 타이밍과 동작 간의 속도 접속 방법을 지정하는 기능
	감시 기능	소프트웨어 리미트	축의 동작 범위를 감시하는 기능
		위치 편차	축별로 지령 현재값과 피드백 현재값 사이의 위치 편차를 감시하는 기능
		속도/가감 속도/토크/보간 속도/보간 가감 속도	축 또는 축 그룹별로 경고값을 설정하여 감시하는 기능
	절대값 엔코더 대응		오므론 서보 모터 1S 시리즈 또는 G5 시리즈의 절대값 엔코더 부속 서보 모터를 사용하여 전원 투입 시의 원점 복귀가 불필요한 기능 *7
	입력 신호의 논리 반전 *2		즉시 정지 입력 신호, 정방향 한계 입력 신호, 역방향 한계 입력 신호, 원점 근방 입력 신호의 논리를 반전하는 기능
외부 I/F 신호			서보 드라이버 측의 하기 입력 신호를 사용 가능 ・원점 신호, 원점 근방 신호, 정방향 한계 신호, 역방향 한계 신호, 즉시 정지 신호, 할당 입력 신호

*1. 권장 서보 드라이버 G5 시리즈의 버전은 모터가 실린더 타입인 경우 Ver.2.1 이상, 리니어 모터 타입인 경우 Ver.1.1 이상입니다.
*2. CPU 유니트의 버전 업그레이드로 추가된 기능입니다. 버전 업그레이드에 관한 정보는 「A-7 버전 관련 정보 (P.A-34)」를 참조해 주십시오.
*3. 권장 유니트 버전은 Ver.1.1 이상입니다.
*4. NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 모션 제어 기능의 일부가 다릅니다. 상세한 내용은 『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.
*5. 펄스 단위로 변환한 값을 부호가 있는 정수형 40비트 범위에서 설정할 수 있습니다.
*6. 캠 테이블은 Sysmac Studio의 캠 에디터 또는 사용자 프로그램 속 캠 테이블 생성 명령으로 작성할 수 있습니다. 주축의 위상과 종축의 변위를 지정합니다. 구간별 위상 간격 폭을 변경할 수 있습니다. 사용자 프로그램 안에서 캠 데이터를 갱신할 수 있습니다.
*7. 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입에서는 절대값 타입의 외부 스케일도 사용 가능합니다.

2

모션 제어계의 구성과 원리

이 장에서는 CPU 유니트의 내부 구성 개요와 모션 제어 기능 모듈의 구성 및 원리에 대해 설명합니다.

2-1 CPU 유니트 내부 구성 2-2
2-2 모션 제어계의 구성 2-3
2-3 모션 제어계의 원리 2-5
2-3-1 CPU 유니트의 태스크 2-5
2-3-2 모션 제어의 태스크 동작 예 2-12
2-4 EtherCAT 통신과의 관계 2-19
2-4-1 CAN application protocol over EtherCAT(CoE) 2-19
2-4-2 EtherCAT 마스터 기능과 모션 제어 기능 모듈의 관계 2-20
2-4-3 프로세스 데이터 통신 주기와 모션 제어 주기와의 관계 2-23

2-1 CPU 유니트 내부 구성

여기에서는 NJ/NX 시리즈 CPU 유니트의 내부 구조 개요에 대해 설명합니다.
CPU 유니트는 아래와 같은 소프트웨어 구조로 이루어져 있습니다.
모션 제어 기능 모듈은 모션 제어를 실행하는 소프트웨어입니다.

	모션 제어 기능 모듈	EtherCAT 마스터 기능 모듈	기타 기능 모듈 *1
PLC 기능 모듈			
OS			

*1. 기타 기능 모듈에 대해서는 『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편 (SBCA-467)』을 참조해 주십시오.

OS상에 PLC 기능 모듈이 있고 그 위에서 각 기능 모듈이 작동합니다.
각 기능 모듈의 내용은 아래와 같습니다.

기능 모듈명	약칭	내용
PLC 기능 모듈	PLC	전체 스케줄 관리, 사용자 프로그램의 실행, 모션 제어 모듈에 대한 지령, USB 및 SD 메모리 카드와의 인터페이스 등을 실행합니다.
모션 제어 기능 모듈	MC	사용자 프로그램의 모션 제어 명령 지령에 따라 모션 제어를 실행하고, EtherCAT 마스터 기능 모듈에 데이터를 전송합니다.
EtherCAT 마스터 기능 모듈	ECAT	EtherCAT 마스터로 하여 EtherCAT 슬레이브와의 통신을 실행합니다.

(주) 기타 기능 모듈에 대해서는 『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

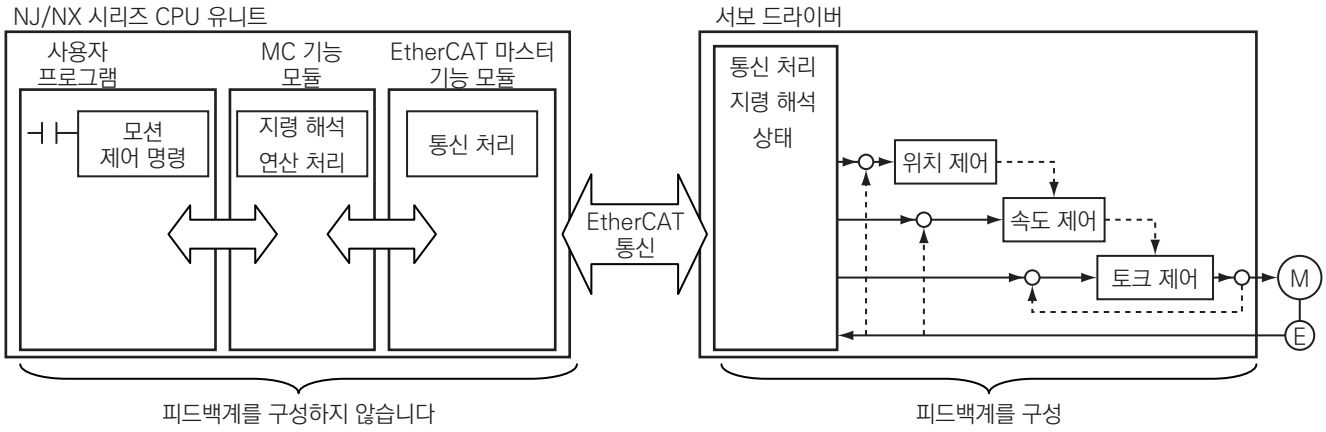
본 매뉴얼에서는 모션 제어 기능 모듈(이하,「MC 기능 모듈」로 생략하는 경우가 있습니다)의 사양, 취급에 대해 설명합니다.
MC 기능 모듈을 사용할 경우에는 NJ/NX 시리즈의 관련 사용자 매뉴얼도 함께 참조해 주십시오.

2-2 모션 제어계의 구성

서보에 의해 구축되는 제어계는 일반적으로 세미 클로즈드 루프 방식을 이용하여 모터 동작을 제어합니다. 세미 클로즈드 루프 방식은 지령값에 대한 모터의 회전량을 모터에 부속된 엔코더를 통해 검출합니다. 이를 기계의 이동량으로 피드백합니다. 지령값과 모터의 실제 회전량과의 편차를 산출하고 이 값이 「0」이 되도록 제어하는 방법입니다.

MC 기능 모듈을 사용한 기기 구성에서는 CPU 유니트의 사용자 프로그램에서의 지령에 관하여 피드백 정보는 사용하지 않습니다. 서보 드라이버 내에서 피드백계를 구축하고 있습니다.

● EtherCAT 네트워크 구성인 경우

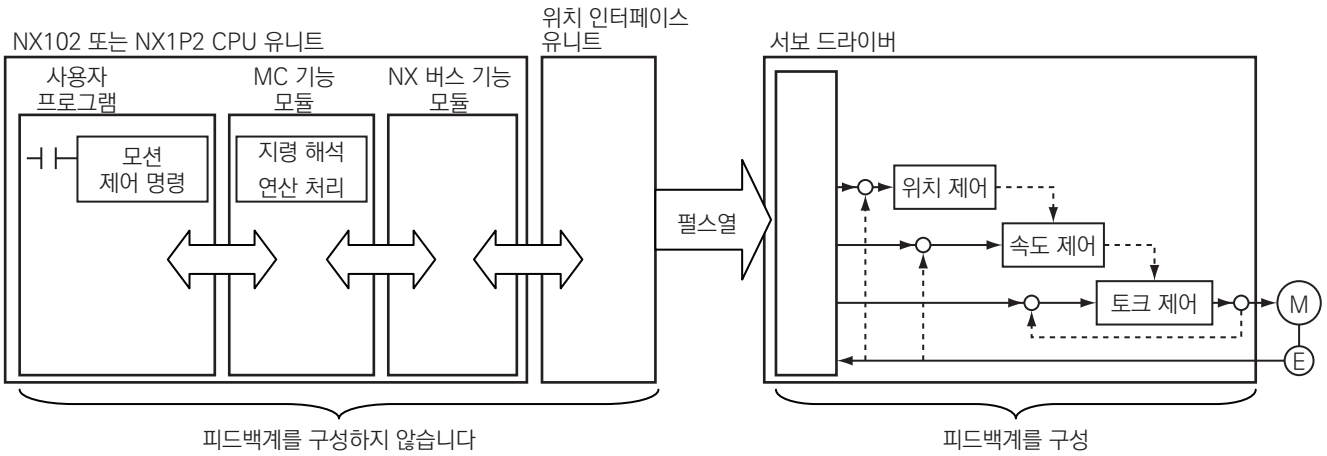


- 사용자 프로그램에서 모션 제어 명령을 기동하면 MC 기능 모듈이 지령을 해석합니다.
- MC 기능 모듈은 지령의 해석 결과를 바탕으로 일정 주기마다 모션 연산을 실행하고, 서보 드라이버에 대한 지령값을 생성합니다. 생성된 지령값은 목표 위치, 목표 속도, 목표 토크입니다.
- 생성된 지령값을 EtherCAT 통신의 프로세스 데이터 통신 주기마다 PDO 통신을 통해 송신합니다.
- 서보 드라이버는 EtherCAT 통신의 프로세스 데이터 통신 주기별 지령값에 따라 위치 루프 제어, 속도 루프 제어, 토크 루프 제어를 실행합니다.
- 엔코더의 현재값이나 서보 드라이버의 상태는 EtherCAT 통신의 프로세스 데이터 통신 주기마다 CPU 유니트로 송신합니다.

참고

- 모션 연산과 EtherCAT 통신의 프로세스 데이터 통신은 동일한 주기입니다.
- MC 기능 모듈은 위치 제어 루프, 속도 제어 루프, 토크 제어 루프가 내장된 서보 드라이버를 대상으로 합니다.
- NX 시리즈 위치 인터페이스 유니트를 사용하는 경우의 구성은 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

● CPU 유니트상의 구성인 경우



- 사용자 프로그램에서 모션 제어 명령을 기동하면 MC 기능 모듈이 지령을 해석합니다.
- MC 기능 모듈은 지령의 해석 결과를 바탕으로 일정 주기마다 모션 연산을 실행하고 위치 인터페이스 유니트(펄스 출력 유니트)에 대한 지령값을 생성합니다. 생성되는 지령값은 목표 위치, 목표 속도입니다.
- 생성된 지령값을 펄스열로 하여 서보 드라이버에 출력합니다.
- 서보 드라이버는 펄스열 지령값에 따라 위치 루프 제어, 속도 루프 제어를 실행합니다.

참고

NX 시리즈 위치 인터페이스 유니트를 사용하는 경우의 구성은 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

2-3 모션 제어계의 원리

여기에서는 CPU 유니트의 태스크와 모션 제어의 관계에 대해 설명합니다.

2-3-1 CPU 유니트의 태스크

태스크란 사용자 프로그램 등의 처리 실행 조건 및 순서의 속성을 의미합니다.
NJ/NX 시리즈 CPU 유니트에는 다음과 같은 태스크가 있습니다.

태스크의 종류	태스크명
일정 주기마다 실행되는 태스크	프라이머리 정주기 태스크
	정주기 태스크(실행 우선도 5[*1], 16[*2], 17, 18)
사전에 지정한 실행 조건이 성립했을 때 1번만 실행되는 태스크	이벤트 태스크(실행 우선도 8, 48) [*3]

*1. 정주기 태스크(실행 우선도 5)는 NX701 CPU 유니트만 사용할 수 있습니다.
*2. 정주기 태스크(실행 우선도 16)은 NX102 및 NX1P2 CPU 유니트에서는 사용할 수 없습니다.
*3. 이벤트 태스크는 유니트 버전 Ver.1.03 이상인 CPU 유니트와 Sysmac Studio Ver.1.04 이상과의 조합으로 사용할 수 있습니다.

사용자 프로그램 및 태스크의 상세한 내용, 설정 방법에 대해서는 『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

태스크의 종류와 우선도

NX701 CPU 유니트는 멀티 코어 프로세서가 탑재되어 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)를 병렬로 실행할 수 있습니다.
NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트는 여러 개의 태스크를 동시에 실행할 수 없습니다.
태스크에는 실행 우선도가 있어 실행 우선도가 높은 태스크를 우선적으로 실행합니다. 태스크를 실행하는 도중에 실행 우선도가 높은 다른 태스크의 실행 조건이 성립하면 실행 우선도가 높은 태스크를 우선적으로 실행합니다.
아래 표는 NJ/NX 시리즈 CPU 유니트에서 모션 제어 명령을 사용할 수 있는 태스크의 종류와 우선도입니다. 이벤트 태스크는 모션 제어 명령을 사용할 수 없습니다.

태스크의 종류	태스크 수	우선도	동작
프라이머리 정주기 태스크	1개	4	설정한 태스크 주기마다 I/O 리프레쉬, 사용자 프로그램 실행, 모션 제어를 실행합니다. 태스크의 실행 우선도가 가장 높고 고속 · 고정밀도로 실행할 수 있으므로 동기 제어나 고속 응답이 필요한 제어에 적합합니다. 싱글 태스크로 모든 제어를 실행할 경우에는 프라이머리 정주기 태스크를 사용합니다.
정주기 태스크	0~1개	5 [*1]	설정한 태스크 주기마다 I/O 리프레쉬, 사용자 프로그램 실행, 모션 제어를 실행합니다. 태스크의 실행 우선도가 프라이머리 정주기 태스크 다음으로 높고 고속 · 고정밀도로 실행할 수 있습니다. 장비를 구성하는 기능 중 최고속으로 제어하고 싶은 기능을 프라이머리 정주기 태스크로 제어하고, 그 외 기능을 정주기 태스크(실행 우선도 5)로 분할하여 제어하는 경우에 사용합니다. 멀티 태스크로 모션 제어를 실행할 경우에는 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)를 사용합니다. [*2]

태스크의 종류	태스크 수	우선도	동작
정주기 태스크	0~1개	16 *3	설정한 태스크 주기마다 I/O 리프레쉬와 사용자 프로그램을 실행합니다. 프라이머리 정주기 태스크나 정주기 태스크(실행 우선도 5)보다 늦은 주기로 사용자 프로그램을 실행할 경우에 정주기 태스크(실행 우선도 16)을 사용합니다. 정주기 태스크(실행 우선도 16)은 프라이머리 정주기 태스크에서 I/O 리프레쉬하는 일부 슬레이브나 유니트에 대해 사용자 프로그램을 기술할 수 있습니다. 예를 들어 동기 제어나 고속 응답이 필요한 제어는 프라이머리 정주기 태스크나 정주기 태스크(실행 우선도 5)로 제어하고, 장비 전체 제어는 정주기 태스크(실행 우선도 16)으로 분할하여 제어합니다.

*1. 정주기 태스크(실행 우선도 5)는 NX701 CPU 유니트만 사용할 수 있습니다.
*2. 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)를 사용하여 모션 제어를 병렬로 실행하는 것을 「멀티 모션」이라 합니다.
*3. 정주기 태스크(실행 우선도 16)은 NX102 및 NX1P2 CPU 유니트에서는 사용할 수 없습니다.

(주) CPU 유니트에는 실행 우선도가 17 및 18인 정주기 태스크가 있습니다. 단, 이러한 태스크에서는 모션 제어 명령을 사용할 수 없습니다. 또한, I/O 리프레쉬는 실행하지 않습니다.

사용상의 주의

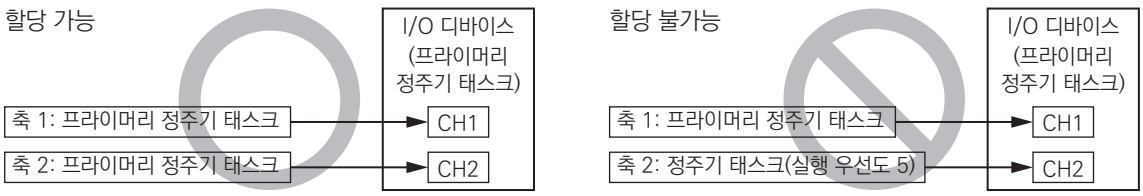
모션 제어 명령은 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5와 16)에서 사용할 수 있습니다.
위 이외의 태스크에서 모션 제어 명령을 사용하면 Sysmac Studio에서 빌드를 진행할 때 이상이 발생합니다.

참고

NX701 CPU 유니트에서는 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)에서 모션 제어를 실행할 수 있습니다. 2가지 모션 제어를 구별해야 하는 경우 프라이머리 정주기 태스크의 모션 제어를 「모션 제어 1」, 정주기 태스크(실행 우선도 5)의 모션 제어를 「모션 제어 2」라고 합니다.

태스크의 할당

- 축 및 축 그룹은 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5) 중 1가지 태스크에 할당할 수 있습니다. 축에 할당하는 I/O 디바이스 태스크는 할당하는 축과 동일한 태스크여야 합니다.



- 프라이머리 정주기 태스크에 할당한 축 및 축 그룹에 대해 정주기 태스크(실행 우선도 16)으로 동작하는 사용자 프로그램에서 모션 제어 명령을 실행할 수 있습니다.

사용상의 주의

- 프라이머리 정주기 태스크에 할당한 축 및 축 그룹에 대해 정주기 태스크(실행 우선도 5)로 동작하는 사용자 프로그램에서 모션 제어 명령을 실행할 수 없습니다. 실행한 경우는 「실행 ID 설정 범위 외(이벤트 코드: 57490000 Hex)」가 발생합니다.
  마찬가지로 정주기 태스크(실행 우선도 5)에 할당한 축 및 축 그룹에 대해 프라이머리 정주기 태스크로 동작하는 사용자 프로그램에서 모션 제어 명령을 실행할 수 없습니다.
- 정주기 태스크(실행 우선도 5)에 할당한 축 및 축 그룹에 대해 정주기 태스크(실행 우선도 16)으로 동작하는 사용자 프로그램에서 모션 제어 명령을 실행할 수 없습니다. 실행한 경우는 「실행 ID 설정 범위 외(이벤트 코드: 57490000 Hex)」가 발생합니다.

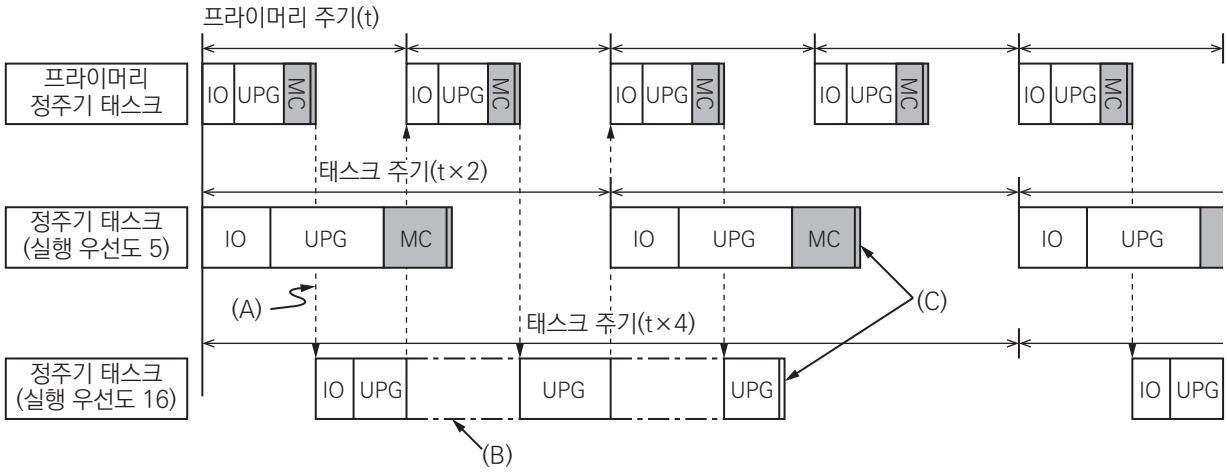
참고

축 및 축 그룹에 대해서는 「제3장 축과 축 그룹의 설정」을 참조해 주십시오.

태스크의 기본 동작

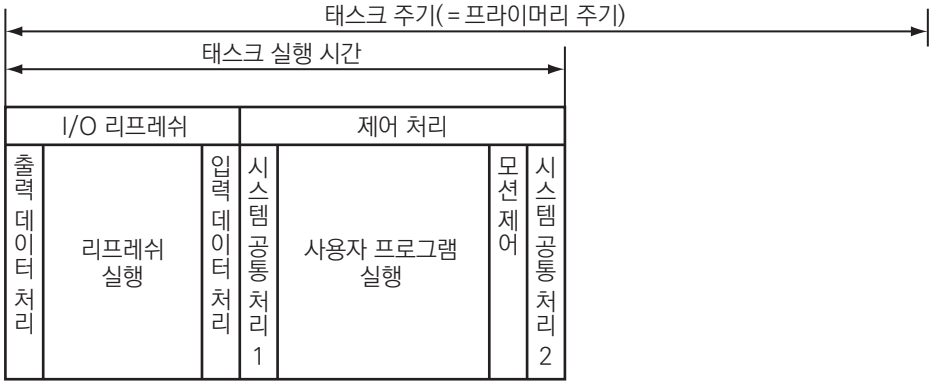
● 태스크 전체의 동작

프라이머리 정주기 태스크와 정주기 태스크는 프라이머리 정주기 태스크의 태스크 주기(이하, 「프라이머리 주기」라 합니다)를 기준으로 동작합니다.
프라이머리 정주기 태스크 및 정주기 태스크(실행 우선도 5)에는 I/O 리프레쉬, 사용자 프로그램 실행 외에 시스템 공통 처리, 모션 제어를 포함합니다.
사용자 프로그램에 기술된 모션 제어 명령은 해당 태스크의 END 명령 실행 후 다음 모션 제어(MC) 타이밍에서 연산을 실행합니다.



기호	내용
IO	I/O 리프레쉬
UPG	사용자 프로그램 실행
MC	모션 제어
(A)	점선은 태스크 변이를 나타냅니다.
(B)	일점쇄선은 태스크 처리 중단 상태를 나타냅니다.
(C)	이중선은 태스크 내 처리의 실행 완료를 나타냅니다.

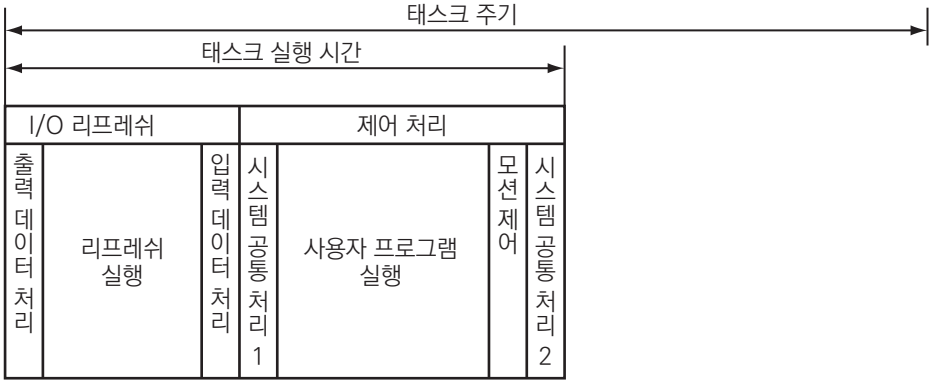
● 프라이머리 정주기 태스크의 동작



처리	처리 내용
출력 데이터 처리	・I/O 리프레쉬를 실행할 I/O(출력)에 대한 OUT 리프레쉬 데이터를 생성합니다. ・강제값 리프레쉬가 설정된 경우에는 OUT 리프레쉬 데이터에 강제값(출력)을 반영합니다.
리프레쉬 실행	I/O와의 데이터 교환을 실행합니다.
입력 데이터 처리	・이벤트 태스크 조건식의 일치 판정을 실행합니다. ・I/O 리프레쉬를 실행한 I/O(입력) 유니트의 IN 리프레쉬 데이터를 가져옵니다. ・강제값 리프레쉬가 설정된 경우에는 가져온 IN 리프레쉬 데이터에 강제값(출력)을 반영합니다.
시스템 공통 처리 1	・변수의 태스크 간 배타 제어 처리(참조 태스크의 경우) ・모션 입력 처리 *1 ・데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행	태스크에 할당된 사용자 프로그램을 할당 순으로 실행합니다.
모션 제어 1 *2	프라이머리 정주기 태스크 및 정주기 태스크(실행 우선도 16)의 사용자 프로그램의 모션 제어 명령에서 모션 제어 지령을 실행합니다. 모션 출력 처리 *3
시스템 공통 처리 2	・변수 태스크 간 배타 제어 처리(갱신 태스크의 경우) ・태스크 실행과의 동시성을 확보하기 위한 컨트롤러 외부에서의 변수 액세스 처리([Task Settings]에서 설정한 「variable access time」) ・EtherNet/IP 태그 데이터 링크 처리인 경우는 태그(네트워크 공개 속성의 변수)에 대해 갱신 태스크를 설정하면 상기 변수 액세스 처리로 실행합니다.

*1. 서보 드라이버 등에서 수신한 데이터를 바탕으로 모션 제어 시스템 변수의 서보 드라이버 상태 및 축 현재값 등이 갱신됩니다.
*2. 모션 제어 1에 할당한 축의 시스템 정의 변수는 「_MC_AX[0-255]」 또는 「_MC1_AX[0-255]」를 사용합니다.
마찬가지로 축 그룹의 시스템 정의 변수는 「_MC_GRP[0-63]」 또는 「_MC1_GRP[0-63]」을 사용합니다.
축 시스템 정의 변수는 「3-1-3 축 변수의 개요 (P.3-6)」, 축 그룹의 시스템 정의 변수는 「3-3-3 축 그룹 변수의 개요 (P.3-21)」를 참조해 주십시오.
*3. 다음 프라이머리 정주기 태스크의 「I/O 리프레쉬」에서 서보 드라이버 등으로 송신됩니다.

● 실행 우선도 5인 정주기 태스크 동작

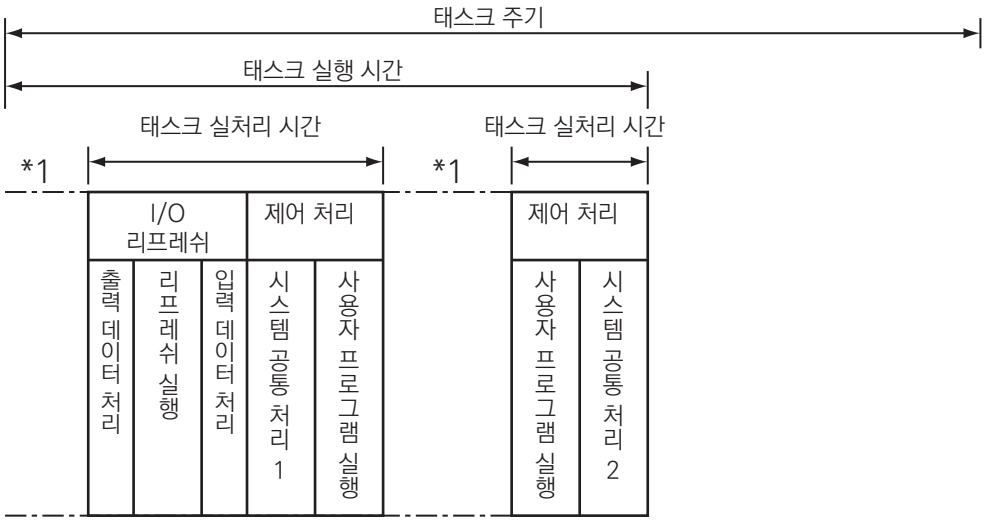


처리	처리 내용
출력 데이터 처리	・I/O 리프레쉬를 실행할 I/O(출력)에 대한 OUT 리프레쉬 데이터를 생성합니다. ・강제값 리프레쉬가 설정된 경우에는 OUT 리프레쉬 데이터에 강제값(출력)을 반영합니다.
리프레쉬 실행	I/O와의 데이터 교환을 실행합니다.
입력 데이터 처리	・I/O 리프레쉬를 실행한 I/O(입력) 유니트의 IN 리프레쉬 데이터를 가져옵니다. ・강제값 리프레쉬가 설정된 경우에는 가져온 IN 리프레쉬 데이터에 강제값(출력)을 반영합니다.
시스템 공통 처리 1	・변수의 태스크 간 배타 제어 처리(참조 태스크의 경우) ・모션 입력 처리 *1 ・데이터 트레이스 처리(샘플링, 트리거 판정)
사용자 프로그램 실행	태스크에 할당된 사용자 프로그램을 할당 순으로 실행합니다.
모션 제어 2 *2	・정주기 태스크(실행 우선도 5)의 사용자 프로그램의 모션 제어 명령에서 모션 제어 지령을 실행합니다. ・모션 출력 처리 *3
시스템 공통 처리 2	・변수 태스크 간 배타 제어 처리(갱신 태스크의 경우) ・태스크 실행과의 동시성을 확보하기 위한 컨트롤러 외부에서의 변수 액세스 처리([Task Settings]에서 설정한 「variable access time」)

*1. 서보 드라이버 등에서 수신한 데이터를 바탕으로 모션 제어 시스템 변수의 서보 드라이버 상태 및 축 현재값 등이 갱신됩니다.
*2. 모션 제어 2에 할당한 축의 시스템 정의 변수는 「_MC2_AX[0-255]」를 사용합니다. 마찬가지로 축 그룹의 시스템 정의 변수는 「_MC2_GRP[0-63]」을 사용합니다.
축 시스템 정의 변수는 「3-1-3 축 변수의 개요 (P.3-6)」, 축 그룹의 시스템 정의 변수는 「3-3-3 축 그룹 변수의 개요 (P.3-21)」를 참조해 주십시오.
*3. 다음 정주기 태스크(실행 우선도 5)의 「I/O 리프레쉬」에서 서보 드라이버 등으로 송신됩니다.

● 실행 우선도 16인 정주기 태스크 동작

정주기 태스크(실행 우선도 16)은 I/O 리프레쉬를 실행할 수 있습니다.



*1. 실행 우선도가 높은 태스크를 실행하기 위해 태스크 실행을 일시적으로 중단합니다.

태스크의 주기

싱글 태스크의 경우는 프라이머리 정주기 태스크의 태스크 주기인 프라이머리 주기로 동작합니다.
싱글 태스크의 경우 프라이머리 주기가 자동으로 모션 제어 주기(= EtherCAT 통신의 프로세스 데이터 통신 주기)가 됩니다.
NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트에서는 싱글 태스크만 제어합니다.

멀티 모션의 경우는 프라이머리 주기와 정주기 태스크(실행 우선도 5)의 태스크 주기로 동작합니다.
멀티 모션의 경우 모션 제어 주기도 2개가 되며, EtherCAT 통신의 프로세스 데이터 통신 주기도 자동으로 각각의 태스크 주기가 됩니다.

정주기 태스크는 프라이머리 주기에 동기하여 실행합니다. 정주기 태스크의 태스크 주기는 프라이머리 주기의 정수배를 설정해 주십시오.
예를 들어 프라이머리 주기를 1ms, 정주기 태스크(실행 우선도 5)의 태스크 주기를 2ms, 정주기 태스크(실행 우선도 16)의 태스크 주기를 4ms로 설정한 경우에는 프라이머리 주기 2회에 1회 간격으로 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)의 주기 선두가 모입니다. 마찬가지로 정주기 태스크(실행 우선도 16)과 프라이머리 정주기 태스크와는 4회에 1회 간격으로 주기 선두가 모입니다.
태스크 주기에 대한 상세한 내용은 📖『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편 (SBCA-467)』을 참조해 주십시오.

참고

2가지 프로세스 데이터 통신 주기를 구별해야 할 경우 프라이머리 정주기 태스크의 통신 주기를 「프로세스 데이터 통신 주기 1」, 정주기 태스크(실행 우선도 5)의 통신 주기를 「프로세스 데이터 통신 주기 2」라고 합니다.

● NX701 CPU 유니트의 설정 가능한 태스크 주기

NX701 CPU 유니트의 경우 태스크 종류에 따라 설정 가능한 태스크 주기는 다음과 같습니다.

태스크	설정 가능한 태스크 주기
프라이머리 정주기 태스크	125μs, 250μs~8ms(250μs 단위로 지정 가능)
정주기 태스크(실행 우선도 5)	125μs, 250μs~100ms(250μs 단위로 지정 가능)
정주기 태스크(실행 우선도 16)	1ms~100ms(250μs 단위로 지정 가능)

(주) 프라이머리 정주기 태스크와 정주기 태스크의 태스크 주기를 설정하는 조건은 다음과 같습니다.
정주기 태스크의 태스크 주기는 프라이머리 정주기 태스크의 정수배로 설정해 주십시오.
각 태스크의 태스크 주기 최소 공배수가 600ms 이하가 되도록 설정해 주십시오.

● NX102 CPU 유니트의 설정 가능한 태스크 주기

NX102 CPU 유니트의 경우 태스크 종류는 프라이머리 정주기 태스크만 가능합니다. 설정 가능한 태스크 주기는 다음과 같습니다.

태스크	설정 가능한 태스크 주기
프라이머리 정주기 태스크	1ms~32ms(250μs 단위로 설정 가능)

● NX1P2 CPU 유니트의 설정 가능한 태스크 주기

NX1P2 CPU 유니트의 경우 태스크 종류는 프라이머리 정주기 태스크만 가능합니다. 설정 가능한 태스크 주기는 다음과 같습니다.

태스크	설정 가능한 태스크 주기
프라이머리 정주기 태스크	2ms~8ms(250μs 단위로 설정 가능)

● NJ 시리즈 CPU 유니트의 설정 가능한 태스크 주기

NJ 시리즈 CPU 유니트의 경우 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 16)의 태스크 주기가 가능한 조합은 아래 표와 같습니다.

프라이머리 주기	정주기 태스크의 설정 가능한 태스크 주기
500μs *1	1ms, 2ms, 3ms, 4ms, 5ms, 8ms, 10ms, 15ms, 20ms, 25ms, 30ms, 40ms, 50ms, 60ms, 75ms, 100ms
1ms	1ms, 2ms, 3ms, 4ms, 5ms, 8ms, 10ms, 15ms, 20ms, 25ms, 30ms, 40ms, 50ms, 60ms, 75ms, 100ms
2ms	2ms, 4ms, 8ms, 10ms, 20ms, 30ms, 40ms, 50ms, 60ms, 100ms
4ms	4ms, 8ms, 20ms, 40ms, 60ms, 100ms

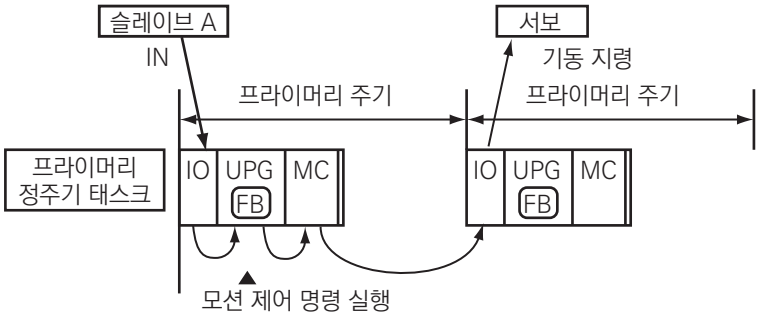
*1. NJ301 CPU 유니트에서는 유니트 버전 Ver.1.03 이상에서 사용할 수 있습니다.
NJ101-10□□에서는 사용할 수 없습니다.

2-3-2 모션 제어의 태스크 동작 예

모션 제어 명령은 프라이머리 정주기 태스크 및 정주기 태스크(실행 우선도 5와 16)에 기술할 수 있습니다. 여기에서는 각각의 경우의 태스크 동작 예에 대해 설명합니다.

프라이머리 정주기 태스크에서 모션 제어 명령을 사용할 경우

고속 모션 제어가 요구되는 경우에는 프라이머리 정주기 태스크에 모션 제어 명령(FB)을 기술합니다.



1 **데이터 가져오기**

I/O 리프레쉬(IO)에서 EtherCAT 슬레이브(슬레이브 A)의 입력 데이터를 가져옵니다.

2 **명령 기동**

사용자 프로그램 실행(UPG)에서 가져온 입력 데이터를 바탕으로 모션 제어 명령(FB)이 기동됩니다.

모션 제어 명령의 출력 변수는 이 시점에서 갱신됩니다.

3 **지령 생성**

기동된 모션 제어 명령(FB)에 따라 해당 프라이머리 정주기 태스크에 있는 직후의 모션 제어(MC)에서 모션 연산이 실행되고 서보 드라이버 등에 송신할 기동 지령이 생성됩니다.

4 **지령 송신**

생성된 기동 지령은 다음 주기의 I/O 리프레쉬(IO)에서 서보 드라이버 등에 송신됩니다.

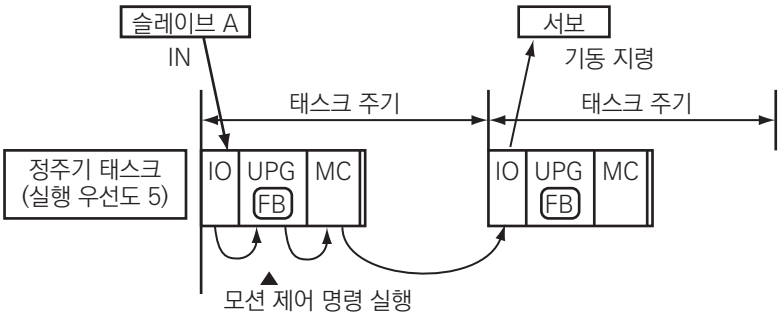
참고

입력에서 서보 드라이버 등에 송신할 기동 지령이 출력되기까지가 고속으로 처리되므로 모션 제어 명령은 프라이머리 정주기 태스크에 기술할 것을 권장합니다.

실행 우선도 5인 정주기 태스크에서 모션 제어 명령을 사용할 경우

프라이머리 정주기 태스크 다음으로 빠른 모션 제어가 요구되는 경우에는 정주기 태스크(실행 우선도 5)에 모션 제어 명령(FB)을 기술합니다.
기본적인 동작은 프라이머리 정주기 태스크와 동일합니다.

● 처리 타이밍



1 **데이터 가져오기**
I/O 리프레쉬(IO)에서 EtherCAT 슬레이브(슬레이브 A)의 입력 데이터를 가져옵니다.

2 **명령 기동**
사용자 프로그램 실행(UPG)에서 가져온 입력 데이터를 바탕으로 모션 제어 명령(FB)이 기동됩니다.
모션 제어 명령의 출력 변수는 이 시점에서 갱신됩니다.

3 **지령 생성**
기동된 모션 제어 명령(FB)에 따라 해당 프라이머리 정주기 태스크에 있는 직후의 모션 제어(MC)에서 모션 연산이 실행되고 서보 드라이버 등에 송신할 기동 지령이 생성됩니다.

4 **지령 송신**
생성된 기동 지령은 다음 주기의 I/O 리프레쉬(IO)에서 서보 드라이버 등에 송신됩니다.

참고

- 정주기 태스크(실행 우선도 5)는 NX701 CPU 유니트만 사용할 수 있습니다.
- 장비를 구성하는 기능 중 최고속으로 제어하고 싶은 기능을 프라이머리 정주기 태스크, 그 외 기능을 정주기 태스크(실행 우선도 5)로 분할하여 병렬 제어하는 경우에 정주기 태스크(실행 우선도 5)를 사용합니다.

● 멀티 모션에서 축 변수를 반영하는 타이밍

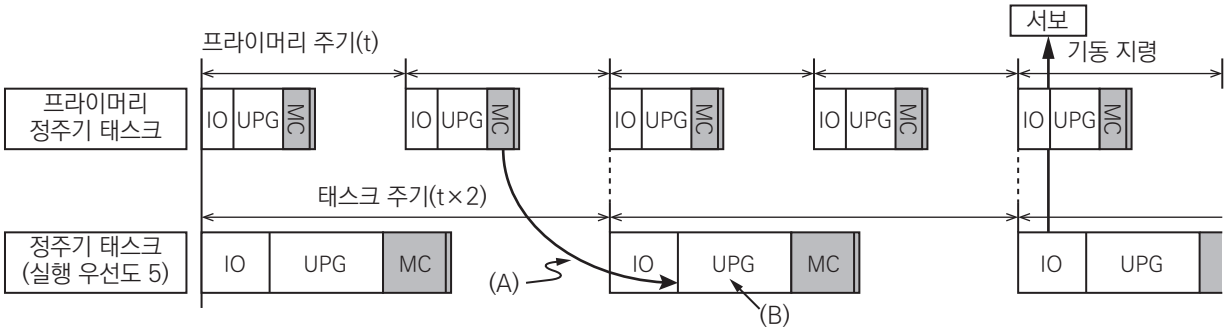
멀티 모션이란 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)를 사용하여 병렬 제어하는 것을 말합니다.
멀티 모션에서 프라이머리 정주기 태스크에서 제어하는 축의 축 변수값을 정주기 태스크(실행 우선도 5)의 사용자 프로그램에서 참조할 수 있습니다. 또한, 반대로 정주기 태스크(실행 우선도 5)에서 제어하는 축의 축 변수값을 프라이머리 정주기 태스크의 사용자 프로그램에서 참조할 수 있습니다.

참고

- 축 변수의 동기에 대해서는 「3-1-4 축 변수의 동기 (P.3-8)」를 참조해 주십시오.
- 프라이머리 정주기 태스크에서 제어하는 축의 축 변수값을 정주기 태스크(실행 우선도 16)의 사용자 프로그램에서 참조할 수 있습니다. 상세한 내용은 「실행 우선도 16의 정주기 태스크에서 모션 제어 명령을 사용할 경우 (P.2-16)」 및 「3-1-4 축 변수의 동기 (P.3-8)」를 참조해 주십시오.

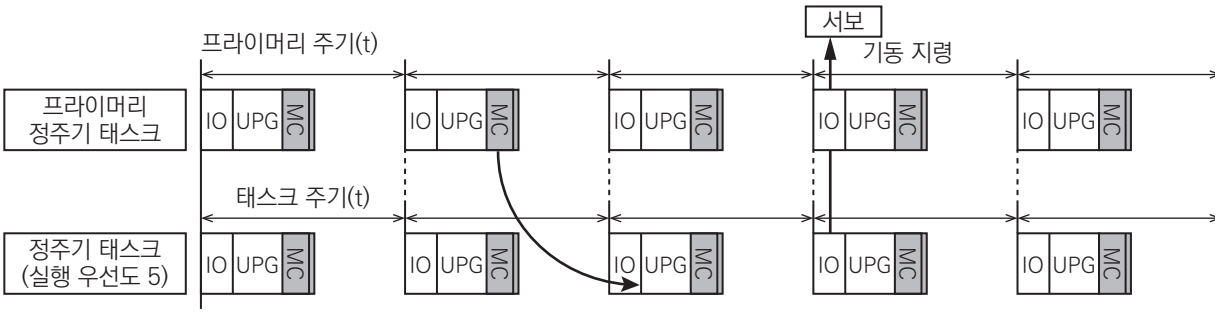
축 변수값은 참조 측 태스크 주기에 동기하여 실행해 주십시오.
축 변수값을 참조하는 측 태스크의 사용자 프로그램이 실행되고 있는 도중에는 참조하고 있는 축 변수값이 갱신되지 않습니다.

프라이머리 정주기 태스크의 축 변수값을 정주기 태스크(실행 우선도 5)에 반영하는 타이밍은 다음과 같습니다.



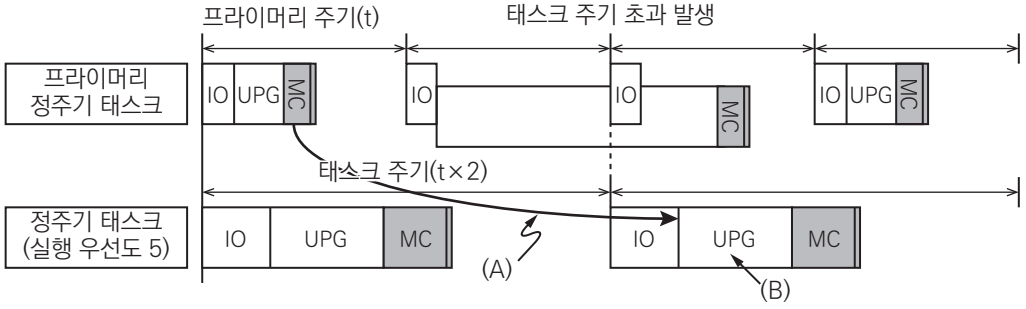
기호	설명
(A)	축 변수의 갱신. 정주기 태스크(실행 우선도 5)의 사용자 프로그램 실행 시작 위치와 상관없이 태스크 주기의 선두가 모이는 주기 직전에 프라이머리 정주기 태스크를 실행한 결과를 반영합니다.
(B)	사용자 프로그램 실행 도중에는 축 변수값이 갱신되지 않습니다.

태스크 주기가 동일한 경우도 마찬가지입니다.



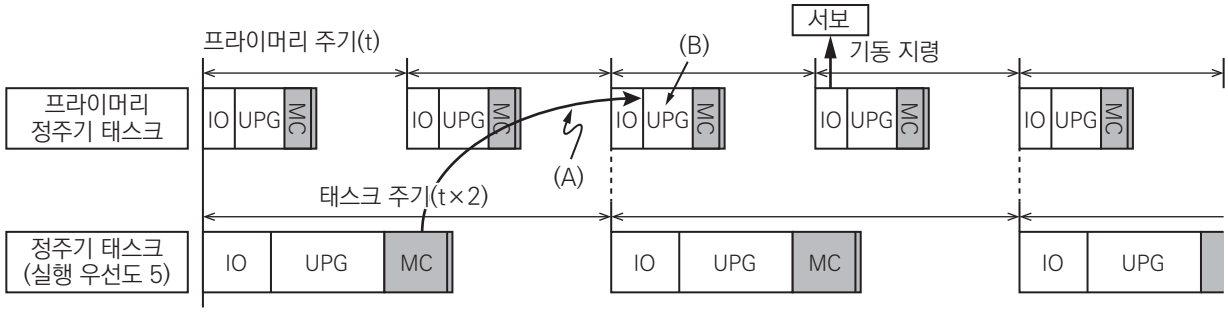
참고

태스크 주기 초과가 발생한 경우 태스크 주기의 선두가 모이는 주기 직전에 태스크 주기를 실행한 결과가 아닌 그보다 하나 더 앞에 위치한 주기를 실행한 결과가 반영됩니다.



기호	설명
(A)	축 변수의 갱신. 태스크 주기 초과가 발생한 경우 하나 더 앞에 위치한 주기의 값을 반영합니다.
(B)	사용자 프로그램 실행 도중에는 축 변수값이 갱신되지 않습니다.

정주기 태스크(실행 우선도 5)의 축 변수값을 프라이머리 정주기 태스크에 반영하는 타이밍은 다음과 같습니다.



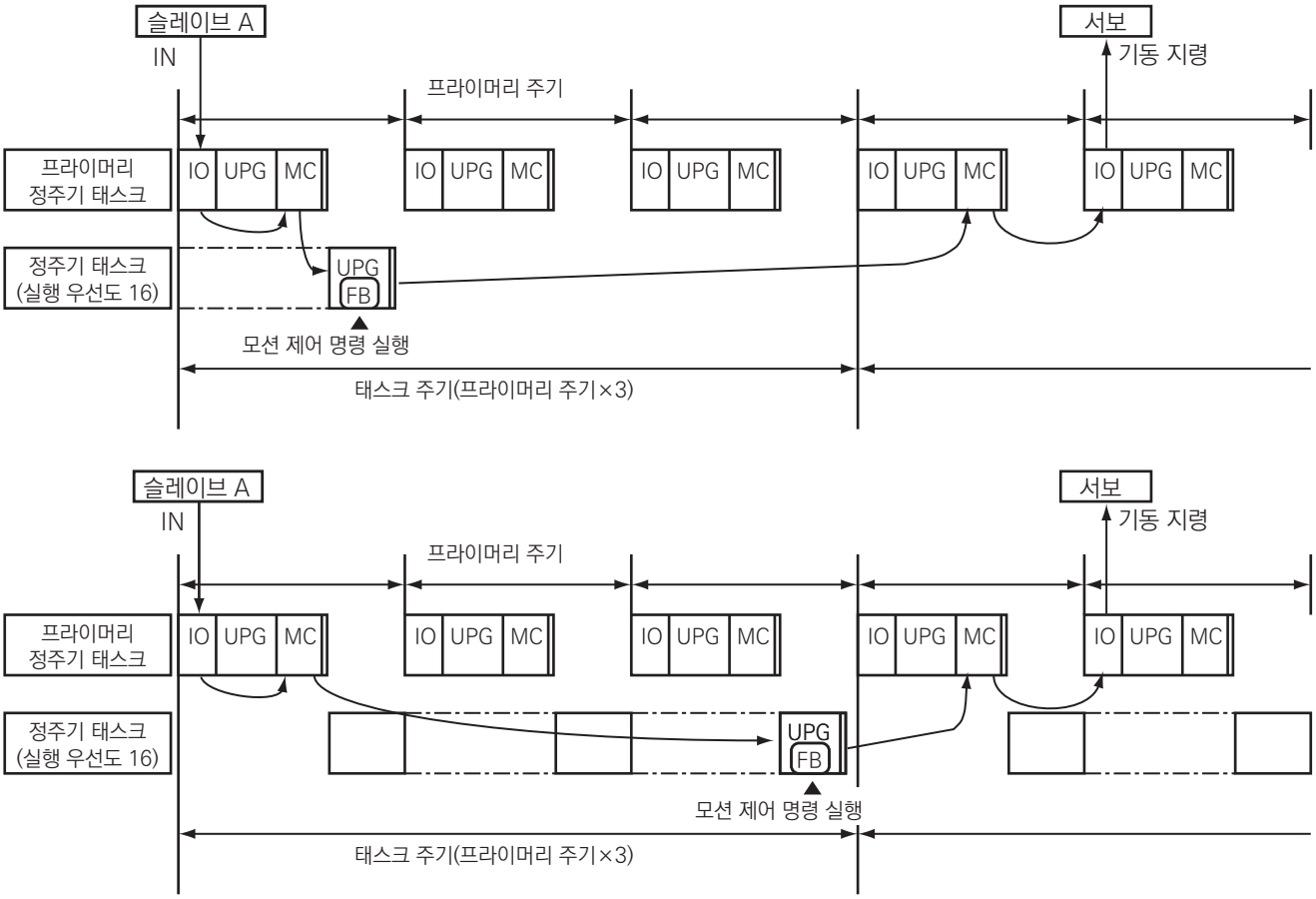
기호	설명
(A)	축 변수의 갱신. 태스크 주기의 선두가 모이는 주기 직전에 정주기 태스크(실행 우선도 5)를 실행한 결과를 반영합니다.
(B)	사용자 프로그램 실행 도중에는 축 변수값이 갱신되지 않습니다.

실행 우선도 16의 정주기 태스크에서 모션 제어 명령을 사용할 경우

고속 모션 제어가 요구되지 않고 사용자 프로그램이 커지는 경우 등에는 정주기 태스크(실행 우선도 16)에 모션 제어 명령을 기술합니다.

● 처리 타이밍

정주기 태스크(실행 우선도 16)의 동일한 태스크 주기에서 실행한 모션 제어 명령(FB)은 같은 타이밍에서 모션 제어(MC)를 실행합니다.
따라서 축이 여러 개여도 동시에 기동하거나 중단할 수 있습니다.



1 **데이터 가져오기**

I/O 리프레쉬(IO)에서 EtherCAT 슬레이브(슬레이브 A)의 입력 데이터를 가져옵니다.

2 **명령 기동**

정주기 태스크(실행 우선도 16)의 사용자 프로그램 실행(UPG)에서 가져온 입력 데이터를 바탕으로 모션 제어 명령(FB)이 기동됩니다.
모션 제어 명령의 출력 변수는 이 시점에서 갱신됩니다.

3 **지령 생성**

기동된 모션 제어 명령(FB)에 따라 정주기 태스크(실행 우선도 16)의 태스크 주기에서 모션 제어 명령(FB)이 실행됩니다.
해당 정주기 태스크의 다음 프라이머리 정주기 태스크에 있는 모션 제어(MC)에서 모션 연산이 실행되고 서보 드라이버 등에 송신할 기동 지령이 생성됩니다.

4 지령 송신

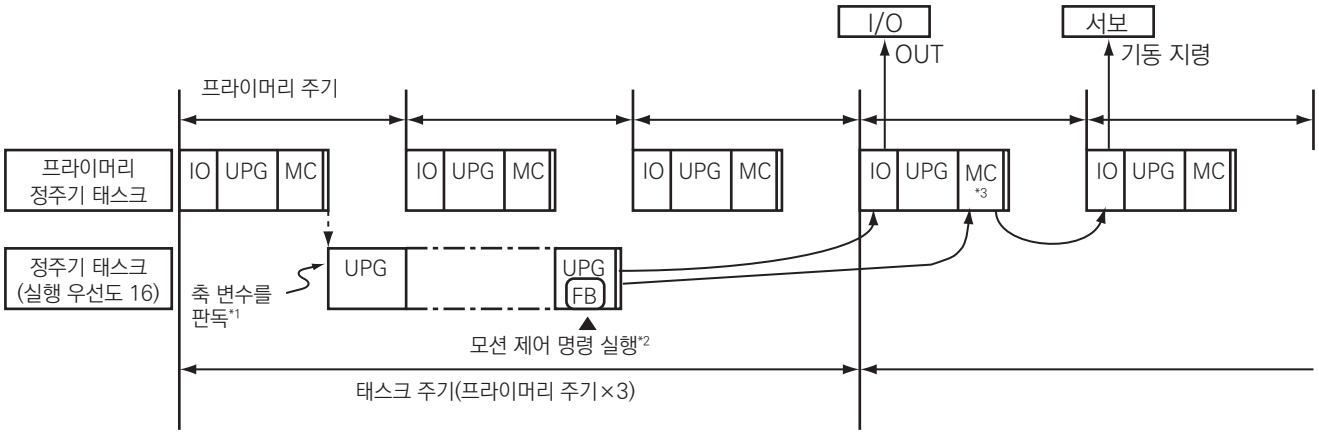
생성된 기동 지령은 하나 더 다음 주기의 I/O 리프레쉬(IO)에서 서보 드라이버 등에 송신됩니다.

● 축 변수의 반영 타이밍

축 변수는 MC 기능 모듈의 축 파라메터 일부와 현재 위치 및 이상 정보 등의 모니터 정보로 이루어진 시스템 정의 변수입니다.

정주기 태스크(실행 우선도 16)에서 프라이머리 정주기 태스크의 축 변수를 참조할 경우 정주기 태스크(실행 우선도 16)의 실행 시작 시 값이 판독됩니다.

또한, 모션 제어 명령(FB)의 실행 시점에서는 축 변수값은 갱신되지 않습니다. 다음 정주기 태스크(실행 우선도 16)이 처음으로 처리하는 모션 제어(MC)에서 갱신됩니다.



*1. 정주기 태스크(실행 우선도 16)의 사용자 프로그램 실행 시작 시 프라이머리 정주기 태스크의 축 변수값이 판독됩니다.

*2. 정주기 태스크(실행 우선도 16)의 모션 제어 명령(FB) 실행 시점에서는 축 변수값이 갱신되지 않습니다.

*3. 이 모션 제어(MC)에서 갱신됩니다.

사용상의 주의

- 모션 제어 명령을 정주기 태스크(실행 우선도 16)에 기술할 경우 정주기 태스크(실행 우선도 16)의 태스크 주기가 길어지면 서보 드라이버 등의 응답 시간도 길어집니다.
- 비상 정지와 같이 외부에서 실시하는 긴급 정지에 대해서는 안전하게 정지할 수 있도록 사전에 고려해 주십시오.
- 정주기 태스크(실행 우선도 16)에 기술한 모션 제어 명령의 실행 타이밍과 I/O 제어의 실행 타이밍이 다르므로, 이 점에 주의해서 프로그램을 작성해 주십시오.
- 정주기 태스크(실행 우선도 16)은 NX102 및 NX1P2 CPU 유니트에서는 사용할 수 없습니다.

참고

축 변수에 대해서는 「3-1-3 축 변수의 개요 (P.3-6)」를 참조해 주십시오.

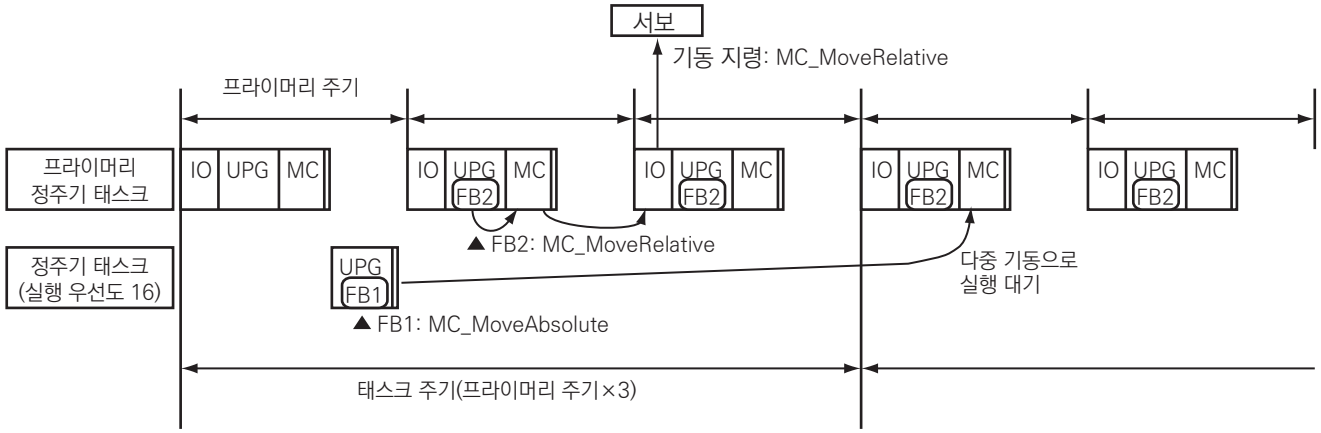
2종류의 태스크로 모션 제어 명령을 사용할 경우

동일한 축에서 빠른 모션 제어가 요구되는 처리와 요구되지 않는 처리가 있는 경우는 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 16) 양쪽에 모션 제어 명령(FB)을 기술합니다.

정주기 태스크(실행 우선도 16)의 태스크 주기 내에서 양쪽 태스크의 모션 제어 명령(FB)을 기동하면 MC 기능 모듈이 프라이머리 정주기 태스크의 명령인 모션 연산을 먼저 실행합니다.

예를 들어 정주기 태스크(실행 우선도 16)에서 MC_MoveAbsolute를 기동합니다. 그 후 프라이머리 정주기 태스크에서 동일한 축에 대해 MC_MoveRelative를 기동하는 경우는 아래와 같은 동작이 됩니다.

- MC 기능 모듈은 MC_MoveRelative를 먼저 실행하고, MC_MoveAbsolute를 명령의 다중 기동으로 실행합니다.



모션 제어 명령의 출력 변수값이나 모션 제어 시스템 변수값은 그 명령을 기동한 태스크의 I/O 리프레쉬 타이밍에서 바뀝니다. 그러므로 복수의 태스크에서 동일한 축에 대해 모션 제어 명령을 사용하면 태스크에 따라 상태가 다르게 보일 수 있습니다.
각각의 태스크 처리를 충분히 이해한 후에 프로그램을 작성해 주십시오.

사용상의 주의

- 프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 16) 양쪽에 동일한 축의 모션 제어 명령을 기술할 경우에는 모션 제어 명령의 실행 순서와 모션 제어 시스템 변수의 갱신 타이밍, 지령값의 출력 타이밍에 주의하여 프로그램을 작성해 주십시오.
- 동일한 축의 모션 제어 시스템 변수를 복수의 태스크에서 사용할 경우 모션 제어 시스템 변수의 갱신 타이밍이 다르므로, 이 점에 주의해서 프로그램을 작성해 주십시오.
- 정주기 태스크(실행 우선도 16)은 NX102 및 NX1P2 CPU 유니트에서는 사용할 수 없습니다.

참고

다중 기동에 대해서는 「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.

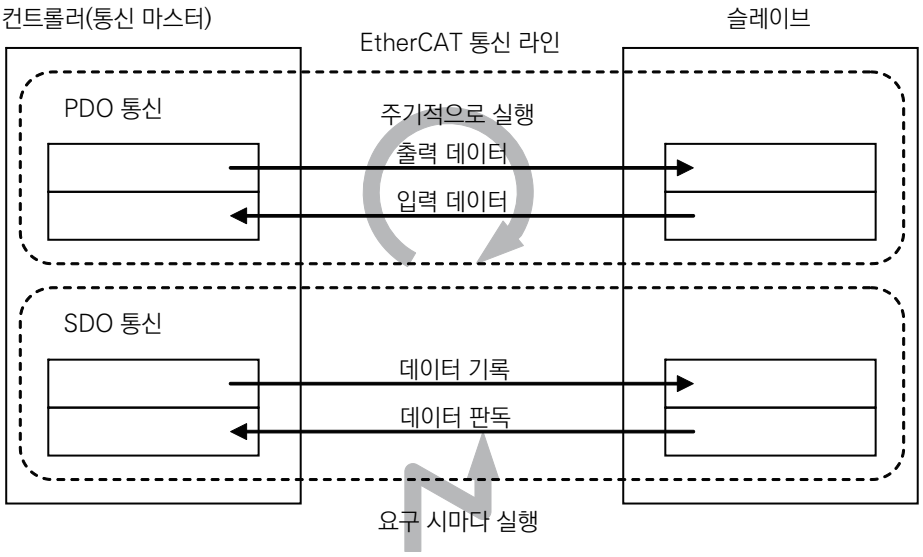
2-4 EtherCAT 통신과의 관계

MC 기능 모듈은 CPU 유니트에 내장된 EtherCAT 마스터 기능 모듈의 PDO 통신을 통해 서보 드라이버, 카운터 및 NX 시리즈 위치 인터페이스 유니트를 제어합니다.
여기에서는 EtherCAT 통신에 대해 MC 기능 모듈과 관련된 항목을 설명합니다.

2-4-1 CAN application protocol over EtherCAT(CoE)

MC 기능 모듈은 EtherCAT상에서 슬레이브와 정보를 교환할 때 CAN application protocol over EtherCAT(CoE) 프로토콜을 사용합니다.
CoE에서는 각종 슬레이브가 보유한 파라메터 및 제어 정보가 오브젝트 딕셔너리(OD)라 불리는 데이터 사양에 의해 규정됩니다.
이러한 데이터를 컨트롤러/통신 마스터와 슬레이브 사이에서 정보로 전달하기 위해 정주기로 실시간 정보를 교환하는 프로세스 데이터 오브젝트(Process Data Objects: PDO)와 임의의 타이밍으로 정보를 전달하는 서비스 데이터 오브젝트(Service Data Objects: SDO) 등 2가지 방법을 사용할 수 있습니다.

MC 기능 모듈에서는 서보 모터의 위치 제어 등 일정한 제어 주기로 입출력 데이터를 갱신하는 지령에 PDO 통신을 사용합니다.
이와는 달리 파라메터 전송 등 지정된 타이밍에서 데이터의 판독 및 기록을 실행하는 지령에는 SDO 통신을 사용합니다.



2-4-2 EtherCAT 마스터 기능과 모션 제어 기능 모듈의 관계

NJ/NX 시리즈 CPU 유니트에서는 EtherCAT 슬레이브를 접속하여 시퀀스 제어나 모션 제어를 실행할 수 있습니다.

시퀀스 제어

- Sysmac Studio의 EtherCAT 구성 편집 화면에서 EtherCAT 구성을 작성하면 구성 슬레이브의 「I/O 포트」가 자동 생성됩니다.
- Sysmac Studio의 I/O 맵에서 디바이스 변수를 할당합니다.
- 모션 제어 명령 이외의 명령으로 시퀀스 제어를 실행합니다.

모션 제어

- Sysmac Studio의 EtherCAT 구성 편집 화면에서 EtherCAT 구성을 작성하면 구성 슬레이브의 「I/O 포트」가 자동 생성됩니다.
- 모션 제어 설정 뷰에서 축 변수를 작성하고 모션 제어의 대상이 되는 EtherCAT 슬레이브를 할당합니다.
- 모션 제어 명령으로 모션 제어를 실행합니다.

축 변수에 할당할 수 있는 기기는 EtherCAT 슬레이브의 서보 드라이버와 엔코더 입력 터미널 및 NX 시리즈 위치 인터페이스 유니트입니다.

참고

- 모션 제어 명령 이외의 명령에서는 축 변수에 할당된 EtherCAT 슬레이브나 NX 시리즈 위치 인터페이스 유니트에 대해 PDO 통신으로 직접 지령할 수 없습니다. 단, 축 변수를 통해 간접적으로 상태를 참조할 수는 있습니다.
- SDO 통신을 사용하면 축 변수에 할당된 EtherCAT 슬레이브나 NX 시리즈 위치 인터페이스 유니트의 오브젝트도 판독/기록할 수 있습니다.

  단, PDO 통신에 매핑된 오브젝트에 대해서는 SDO 통신에서 기록하지 마십시오. 기록한 경우의 동작은 슬레이브 사양에 따라 다릅니다. 오므론 슬레이브의 경우 SDO 통신 이상이 발생합니다.
- EtherCAT 슬레이브의 서보 드라이버 및 엔코더 입력 터미널, NX 시리즈 위치 인터페이스 유니트를 축 변수에 할당하지 않은 경우는 범용 EtherCAT 슬레이브와 마찬가지로 시퀀스 제어를 실행해 주십시오.

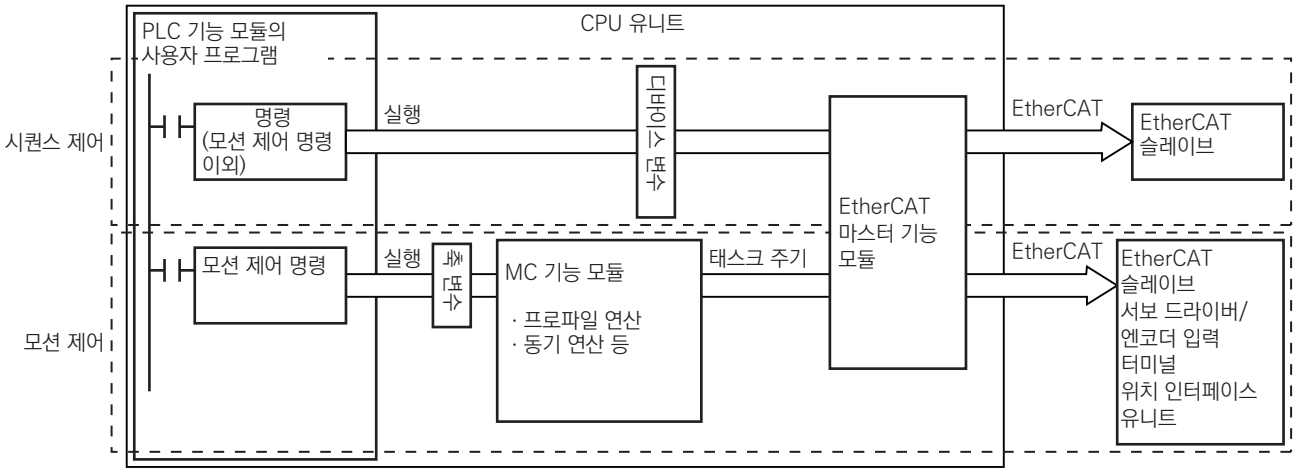
버전 관련 정보

Sysmac Studio Ver.1.09 이상을 사용하면 축 변수에 할당한 슬레이브나 유니트의 I/O 포트에 디바이스 변수를 할당할 수 있습니다.
할당 가능한 I/O 포트의 조건은 다음과 같습니다.

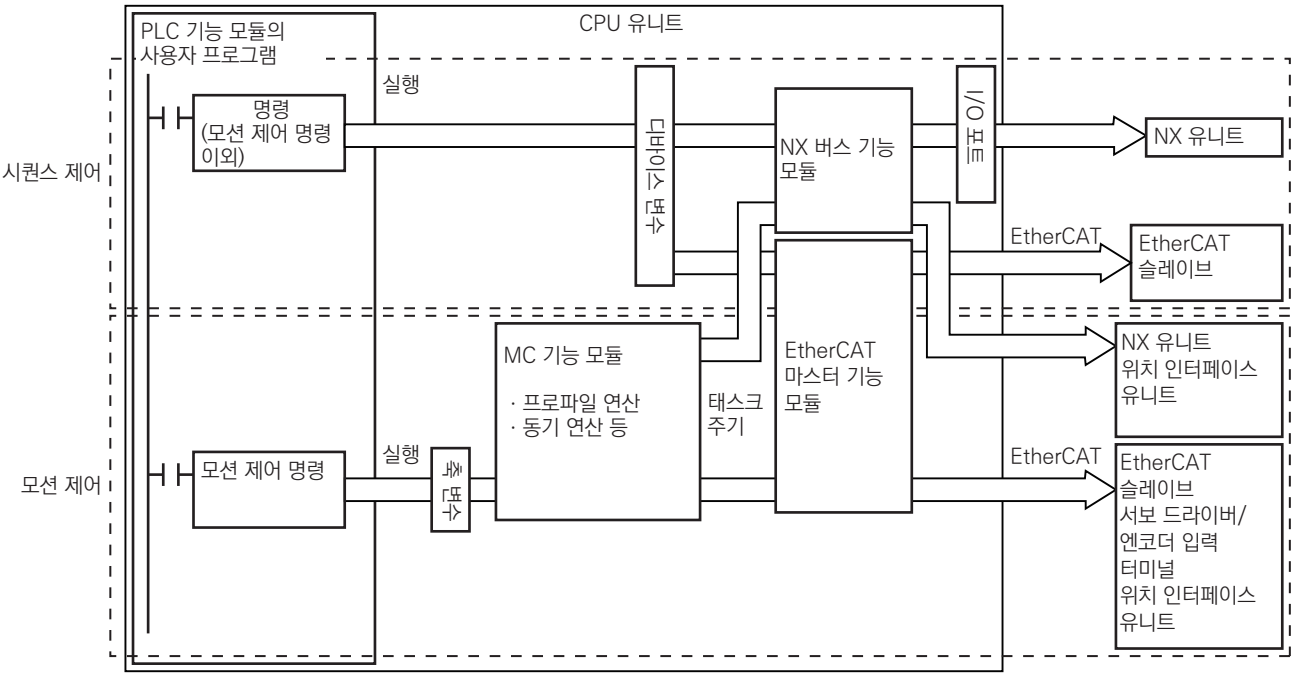
- I/O 포트의 Read/Write 속성이 Read 속성(R: 판독 전용)인 포트
- I/O 포트의 Read/Write 속성이 Write 속성(W: 기록 전용)이고, Sysmac Studio의 [Axis Basic Settings] 화면의 [Detailed Settings]에서 프로세스 데이터를 [<Not assigned>]로 한 포트

Sysmac Studio Ver.1.09 이상을 사용하여 축 변수에 디바이스 변수를 할당한 프로젝트를 Sysmac Studio Ver.1.08 이하에서 열면 디바이스 변수의 할당이 해제됩니다.

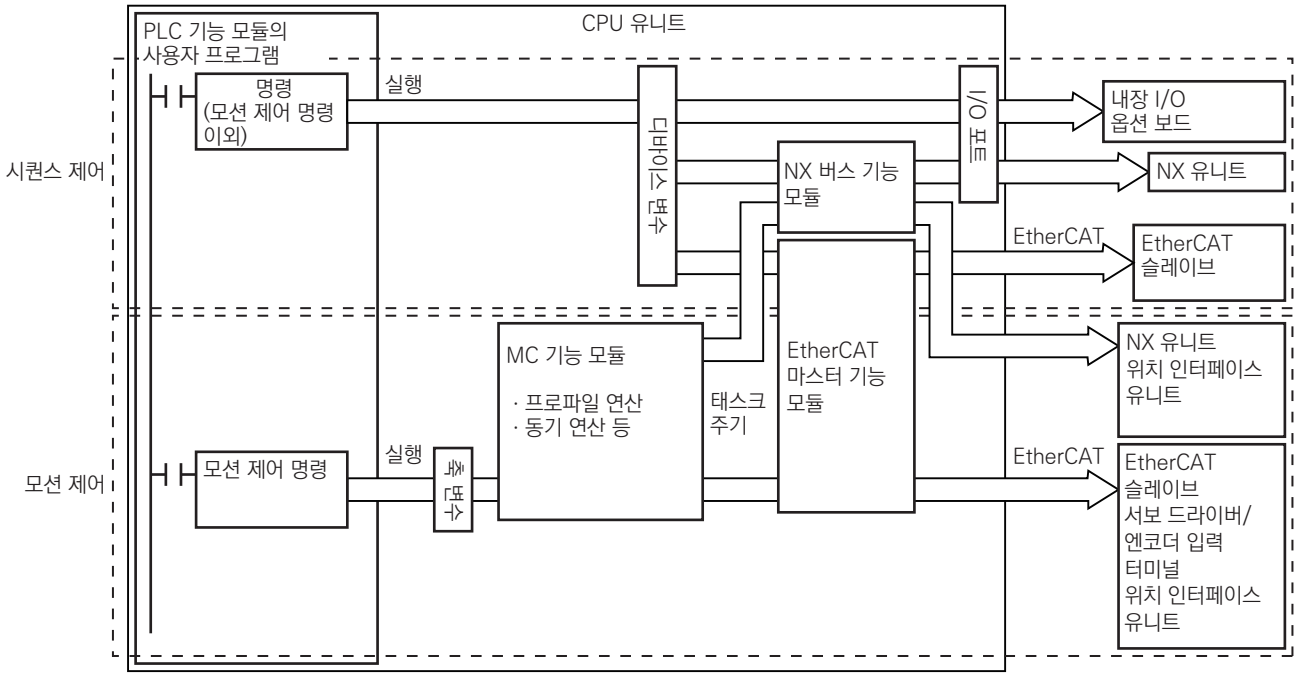
● NX701 CPU 유니트의 경우



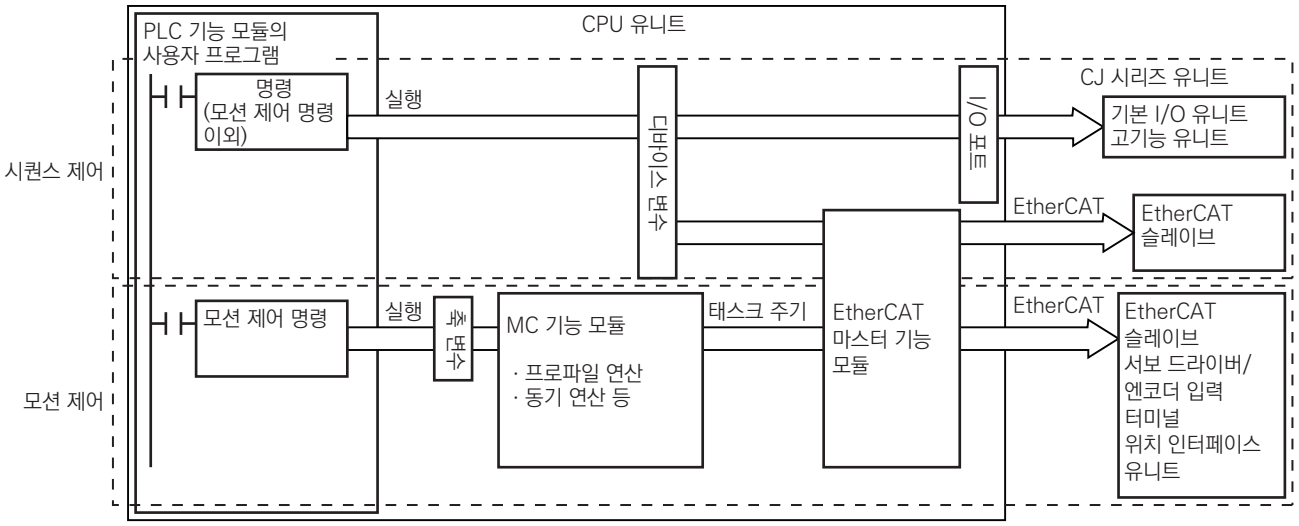
● NX102 CPU 유니트의 경우



● NX1P2 CPU 유니트의 경우



● NJ 시리즈 CPU 유니트의 경우



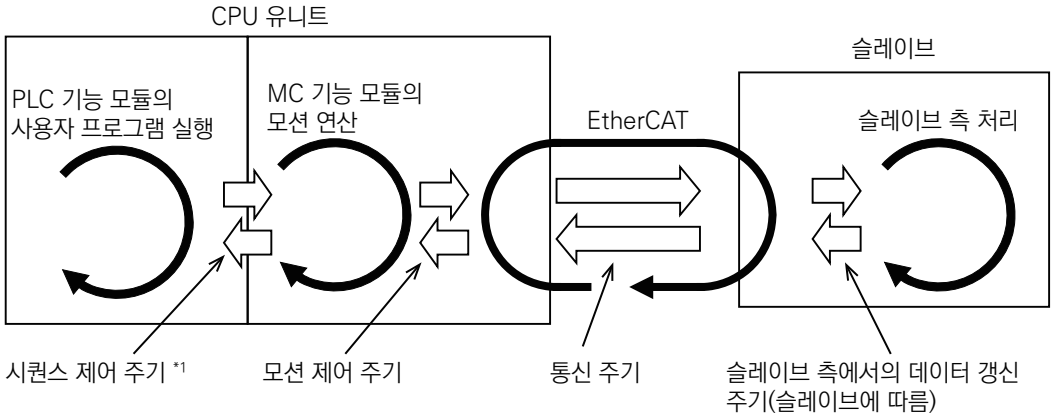
2-4-3 프로세스 데이터 통신 주기와 모션 제어 주기와의 관계

PLC 기본 기능 모듈은 사용자 프로그램상의 모션 제어 명령을 기동함으로써 모션 제어 기능 모듈에 모션 제어의 지령을 발행합니다. 이를 바탕으로 모션 제어 기능 모듈은 모션 연산을 실행하고, 연산 결과를 EtherCAT상의 서보 드라이버 등에 지령합니다.

이러한 데이터 교환은 아래와 같은 주기로 갱신됩니다.

● NX701 CPU 유니트의 경우

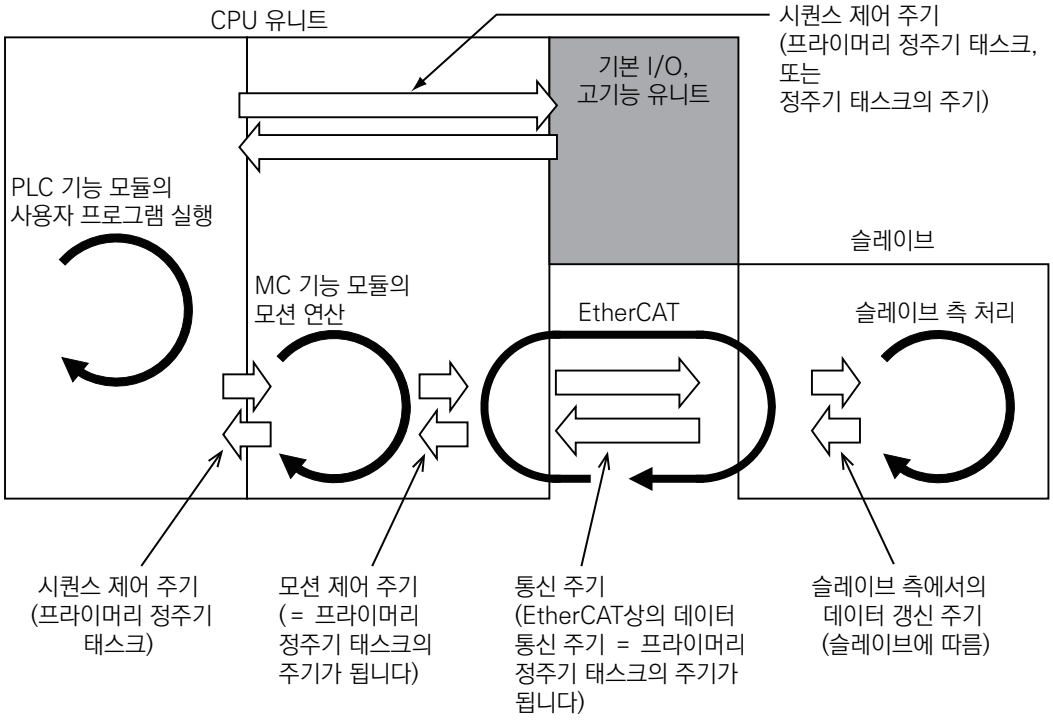
- 「프라이머리 주기」=「모션 제어 1의 주기」=「EtherCAT 통신의 프로세스 데이터 통신 주기 1」
- 「정주기 태스크(실행 우선도 5)의 태스크 주기」=「모션 제어 2의 주기」=「EtherCAT 통신의 프로세스 데이터 통신 주기 2」



*1. 시퀀스 제어 주기가 프라이머리 주기인 경우 모션 제어 주기 및 통신 주기는 프라이머리 주기가 됩니다. 시퀀스 제어 주기가 정주기 태스크(실행 우선도 5)의 태스크 주기인 경우는 모션 제어 주기 및 통신 주기가 정주기 태스크(실행 우선도 5)의 태스크 주기가 됩니다.

● NX102 CPU 유니트, NX1P2 CPU 유니트, NJ 시리즈 CPU 유니트의 경우

・「프라이머리 주기」=「모션 제어의 주기」=「EtherCAT 통신의 프로세스 데이터 통신 주기」



3

축과 축 그룹의 설정

이 장에서는 축과 축 그룹의 개요 설명, Sysmac Studio의 MC 시운전 기능으로 동작 시키는 데 필요한 축의 설정 항목 및 Sysmac Studio를 이용한 작성과 설정 순서에 대해 설명합니다.

3-1 축 3-2
3-1-1 축의 개요 3-2
3-1-2 축 파라메터의 개요 3-3
3-1-3 축 변수의 개요 3-6
3-1-4 축 변수의 동기 3-8
3-1-5 사용자 프로그램에서 축 지정하는 방법 3-8

3-2 축 설정 순서 3-9
3-2-1 축 설정 흐름도 3-9
3-2-2 설정 순서 3-9

3-3 축 그룹 3-19
3-3-1 축 그룹의 개요 3-19
3-3-2 축 그룹 파라메터의 개요 3-20
3-3-3 축 그룹 변수의 개요 3-21
3-3-4 사용자 프로그램에서 축 그룹 지정하는 방법 3-22

3-4 축 그룹 설정 순서 3-24
3-4-1 축 그룹 설정 흐름도 3-24
3-4-2 설정 순서 3-24

3-1 축

여기에서는 MC 기능 모듈의 축에 대해 설명합니다.

3-1-1 축의 개요

모션 제어 시스템에서는 모션 제어의 대상을「축」이라 부릅니다.
축에는 EtherCAT으로 접속하는 실제 서보 드라이버와 엔코더 및 MC 기능 모듈 내부의 가상적인 서보 드라이버와 엔코더가 있습니다.

또한, NX701 CPU 유니트나 NJ 시리즈 CPU 유니트에서는 모든「축」이 모션 제어 축인데, NX102 및 NX1P2 CPU 유니트에서는 모션 제어 축에「단축 위치 제어 축」을 추가했습니다.「단축 위치 제어 축」으로 변환하려면 축 기본 설정의 [Control Function]에서 실행합니다.
[Control Function]에 대한 상세한 내용은 「제어 기능 (P.5-9)」을 참조해 주십시오.

MC 기능 모듈 축의 종별은 다음 표와 같습니다.

축 종별	내용
서보 축	EtherCAT 슬레이브 서보 드라이버나 NX 시리즈 위치 인터페이스 유니트*1를 사용하는 축입니다. 실제 서보 드라이버 등으로 할당하여 사용합니다. 1대의 서보 모터를 1축으로 사용합니다. NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 펄스 출력 유니트와 디지털 입력 유니트 등 복수의 디바이스를 1개의 축에 할당할 수 있습니다.
가상 서보 축	MC 기능 모듈 내의 가상 축입니다. 실제 서보 드라이버는 사용하지 않습니다. 동기 제어의 주축 등으로 사용합니다.
엔코더 축	EtherCAT 슬레이브 엔코더 입력 터미널 및 NX 시리즈 위치 인터페이스 유니트 *1를 사용하는 축입니다. 실제 엔코더 입력 터미널 등으로 할당하여 사용합니다. 1대의 엔코더 입력 터미널에 2개의 카운터가 있는 경우에는 각각 1개의 카운터를 1축으로 사용합니다.
가상 엔코더 축	엔코더 동작을 가상으로 실행하는 축입니다. 엔코더의 실물이 없는 경우 등 엔코더 축 대신에 일시적으로 사용합니다. *2

*1. 제어 대상 기기는 「1-4-3 기능 사양 (P.1-11)」을 참조해 주십시오.
*2. 가상 엔코더 축의 현재 위치를 갱신하는 모션 제어 명령과 조합하여 사용합니다. 이 명령에 대응하지 않는 버전의 MC 기능 모듈에서는 카운트 기능을 사용할 수 없습니다.

MC 기능 모듈의 축에는 아래와 같은 관련 요소가 있습니다.
이 요소들은 각 형식의 제어 축 최대 수만큼 있습니다. 제어 축 최대 수는 형식에 따라 다릅니다. 상세한 내용은 「1-4-2 성능 사양 (P.1-7)」을 참조해 주십시오.

구성 요소	내용	참조 페이지
축 파라메터	MC 기능 모듈로 제어하는 각 축에 관하여 최고 속도나 조그 전송, 원점 복귀와 같은 축 동작을 설정합니다. 축 파라메터를 설정하려면 Sysmac Studio를 사용합니다.	P. 3-3
축 변수	축 변수는 MC 기능 모듈로 제어하는 각 축에 대해 현재 위치 및 이상 정보 등의 모니터 정보로 이루어진 시스템 정의 변수입니다. Sysmac Studio의 멀티 뷰 익스플로러에서 축을 추가하면 축 변수가 작성됩니다. 축 변수의 이름(이하, 축 변수명이라 합니다)은 이 뷰에서 설정할 수 있습니다.	P. 3-6

구성 요소	내용	참조 페이지
사용자 프로그램에서 축 지정하는 방법	사용자 프로그램에서는 모션 제어 명령을 사용하여 모션 제어를 실시합니다. 단축을 제어하는 모션 제어 명령을 「축 지령」이라 부릅니다. 축 지령으로 어떤 축을 제어할지는 명령의 입출력 변수 「Axis(축)」에 시스템 정의 변수의 축 변수명 또는 Sysmac Studio에서 설정한 축 변수명을 지정합니다.	P. 3-8

3-1-2 축 파라메터의 개요

● 축 파라메터 일람

구분	파라메터의 명칭
축 기본 설정	축 번호
	모션 제어 *1
	축 사용
	축 종별
	제어 기능 *2
	입력 디바이스/출력 디바이스
단위 변환 설정	표시 단위
	모터 1회전의 펄스 수
	모터 1회전의 이동량
	감속기 사용 *3
	워크 측 1회전의 이동량 *3
	워크 측 기어비 *3
	모터 측 기어비 *3
동작 설정	최고 속도
	기동 속도 *4
	조그 최고 속도
	최대 가속도
	최대 감속도
	가감속 초과
	반전 시 동작
	속도 경고값
	가속도 경고값
	감속도 경고값
	정방향 토크 경고값 *5
	역방향 토크 경고값 *5
	피드백 속도 필터 시정수
	인포지션 폭
	인포지션 체크 시간
	원점 위치 범위
확장 동작 설정	즉시 정지 입력 정지 방법
	한계 입력 정지 방법
	드라이버 에러 리셋 감시 시간
	정방향 토크 제한 상한값
	역방향 토크 제한 상한값
	즉시 정지 입력 논리 반전 *4
	정방향 한계 입력 논리 반전 *4
	역방향 한계 입력 논리 반전 *4
	원점 근방 입력 논리 반전 *4

구분	파라메터의 명칭
리미트 설정	소프트웨어 리미트 기능
	정방향 소프트웨어 리미트
	역방향 소프트웨어 리미트
	위치 편차 초과값
	위치 편차 경고값
위치 카운트 설정	카운트 모드
	링 카운터 상한 설정값
	링 카운터 하한 설정값
	엔코더 종별
서보 드라이버 설정	링 카운터 상한 설정값
	링 카운터 하한 설정값
	PDS 상태 제어 방식
원점 복귀 설정	원점 복귀 동작
	원점 입력 신호
	원점 복귀 시작 방향
	원점 검출 방향
	정방향 한계 입력 시 동작
	역방향 한계 입력 시 동작
	원점 복귀 속도
	원점 복귀 접근 속도
	원점 복귀 가속도
	원점 복귀 감속도
	원점 복귀 저크
	원점 입력 마스크 거리
	원점 위치 오프셋
	원점 복귀 보류 시간
	원점 복귀 보정값
	원점 복귀 보정 속도

*1. NX701 CPU 유니트의 경우에 설정합니다.
*2. NX102 및 NX1P2 CPU 유니트의 경우에 설정합니다.
*3. 유니트 버전 Ver.1.11 이상인 CPU 유니트와 Sysmac Studio Ver.1.15 이상의 조합으로 사용할 수 있습니다.
*4. 유니트 버전 Ver.1.05 이상인 CPU 유니트와 Sysmac Studio Ver.1.06 이상의 조합으로 사용할 수 있습니다.
*5. 토크 제어 시에만 유효합니다.

축 파라메터에 대한 상세한 내용에 대해서는 「5-2 축 파라메터 (P.5-4)」를 참조해 주십시오.

● 축을 사용하기 위한 필수 설정 항목

Sysmac Studio에서 작성한 축에 대해 다음 항목을 설정하면 축을 사용할 수 있습니다.

구분	파라메터의 명칭	설정 내용	참조 페이지
축 기본 설정	축 번호	축을 작성한 순서대로 자동으로 설정됩니다.	P. 5-6
	모션 제어 *1	[Primary periodic task]를 선택합니다.	
	축 사용	[Used axis]를 선택합니다.	
	축 종별	제어할 축의 종별을 선택합니다.	
	제어 기능 *2	[All]을 선택합니다. *3	
	입력 디바이스/ 출력 디바이스	축에 할당할 EtherCAT 슬레이브 디바이스의 노드 주소를 선택합니다. 축 종별이 가상 축인 경우에는 선택할 수 없습니다.	

*1. NX701 CPU 유니트의 경우에 설정합니다.
*2. NX102 및 NX1P2 CPU 유니트의 경우에 설정합니다.
*3. 모션 제어 축으로 사용할 경우에는 「All」을 선택합니다. 단축 위치 제어 축으로 사용할 경우에는 「Single-axis position control only」를 선택해 주십시오.

● Sysmac Studio에서 서보 드라이버를 시운전하기 위한 필수 설정 항목

Sysmac Studio의 MC 시운전 기능에서 EtherCAT 접속한 서보 드라이버 등을 작동시키려면 다음 항목을 설정합니다.

구분	파라메터의 명칭	설정 내용	참조 페이지
축 기본 설정	축 번호	축을 작성한 순서대로 자동으로 설정됩니다.	P. 5-6
	모션 제어 *1	[Primary periodic task]를 선택합니다.	
	축 사용	[Used axis]를 선택합니다.	
	축 종별	[Servo axis]를 선택합니다.	
	제어 기능 *2	[All]을 선택합니다. *3	
	입력 디바이스/ 출력 디바이스	축에 할당할 EtherCAT 슬레이브 디바이스의 노드 주소를 선택합니다. 축 종별이 가상 축인 경우에는 선택할 수 없습니다.	
단위 변환 설정	표시 단위	[mm], [degree] 등 표시 단위를 선택합니다.	P. 5-11
	모터 1회전의 펄스 수	엔코더 분해능에 맞춰 모터 1회전당 펄스 수를 설정합니다. *4	
	모터 1회전의 이동량	기계 사양에 맞춰 모터 1회전당 워크 이동량을 설정합니다.	
	감속기 사용 *5	감속기 설정을 사용할지 지정합니다.	
	워크 측 1회전의 이동량 *5	워크 측이 1회전한 경우의 이동량을 설정합니다.	
	워크 측 기어비 *5	워크 측의 기어비를 설정합니다.	
	모터 측 기어비 *5	모터 측의 기어비를 설정합니다.	
위치 카운트 설정	카운트 모드	기계 사양에 맞춰 선택합니다.	P. 5-23
리미트 설정	소프트웨어 리미트 기능	장비의 사양에 맞춰 선택합니다.	P. 5-22

*1. NX701 CPU 유니트의 경우에 설정합니다.
*2. NX102 및 NX1P2 CPU 유니트의 경우에 설정합니다.
*3. 모션 제어 축으로 사용할 경우에는 「All」을 선택합니다. 단축 위치 제어 축으로 사용할 경우에는 「Single-axis position control only」를 선택해 주십시오.
*4. 예를 들어 엔코더 분해능이 10,000펄스/회전인 경우에는 10,000을 설정해 주십시오.
*5. 유니트 버전 Ver.1.11 이상인 CPU 유니트와 Sysmac Studio Ver.1.15 이상의 조합으로 사용할 수 있습니다.

사용상의 주의

- 실제 모터를 작동시키는 경우 최고 속도 및 최대 가감 속도, 정지 설정 등은 기계의 동작 조건에 따라 적절한 값을 설정해 주십시오.
- 오므론 서보 드라이버 1S 시리즈 및 G5 시리즈는 노드 주소를 드라이버 전면의 로터리 스위치를 통해 고정값을 설정할 수 있습니다. 이 로터리 스위치를 「00」으로 설정한 경우 노드 주소는 Sysmac Studio의 EtherCAT 에디터에서 설정한 값을 따릅니다.

  접속된 모든 드라이버에서 로터리 스위치를 「00」으로 설정한 경우 드라이버의 접속 위치를 변경해도 이상이 발생하지는 않습니다. 장비의 각 제어 대상에 대해 특정 서보 드라이버를 할당하는 경우에는 드라이버의 노드 주소를 로터리 스위치로 설정해 주십시오.

3-1-3 축 변수의 개요

축 변수는 MC 기능 모듈의 축 파라메터 일부와 현재 위치 및 이상 정보 등의 모니터 정보로 이루어진 시스템 정의 변수입니다.
Sysmac Studio에서 축을 작성하면 작성한 순서대로 축 변수가 변수 테이블에 등록됩니다.
축 변수의 데이터형은 구조체형으로 데이터형명은 「_sAXIS_REF」입니다.

축 변수명

MC 기능 모듈의 축 변수명에는 시스템 정의 변수의 축 변수명과 Sysmac Studio에서 축을 추가했을 때의 축 변수명 등 2종류가 있습니다.
시스템 정의 변수의 축 변수명은 「_MC_AX[0]~_MC_AX[255]」, 「_MC1_AX[0]~_MC1_AX[255]」, 「_MC2_AX[0]~_MC2_AX[255]」입니다.
Sysmac Studio에서 축을 추가하면 「_MC_AX[0]~_MC_AX[255]」에 「MC_Axis000~MC_Axis255」라는 디폴트 명칭이 설정됩니다. 번호는 추가한 순서대로 지정됩니다. 이러한 축 변수명은 Sysmac Studio에서 축마다 임의의 변수명으로 변경할 수 있습니다.
사용자 프로그램에서는 시스템 정의 변수의 축 변수명, Sysmac Studio에서 추가한 축 변수명을 모두 사용할 수 있습니다.

● _MC_AX[0-255]를 사용한 경우의 예

시스템 정의 변수의 축 변수명 (글로벌 변수 뷰의 할당처 *1)	Sysmac Studio에서 추가한 축 변수명의 디폴트	축 번호 예
_MC_AX[0]	MC_Axis000	축 0
_MC_AX[1]	MC_Axis001	축 1
:	:	:
_MC_AX[255]	MC_Axis255	축 255

*1. Sysmac Studio의 글로벌 변수 뷰에 있는 할당처란의 명칭을 변경하면 이상이 발생합니다.

참고

- NX701 CPU 유니트의 경우는 「_MC_AX[0-255]」 또는 「_MC1_AX[0-255]」와 「_MC2_AX[0-255]」를 사용할 수 있습니다.
  NX102 CPU 유니트의 경우는 「_MC_AX[0-14]」만 사용할 수 있습니다.
  NX1P2 CPU 유니트의 경우는 「_MC_AX[0-11]」만 사용할 수 있습니다.
  NJ 시리즈 CPU 유니트의 경우는 「_MC_AX[0-63]」만 사용할 수 있습니다.
- NX701 CPU 유니트의 경우 「_MC_AX[0-255]」와 「_MC1_AX[0-255]」는 축 번호가 같으면 같은 값을 참조합니다. 어떠한 축 변수라도 사용 가능하며 혼재시켜 사용하는 것도 가능합니다.
- 프라이머리 정주기 태스크에 할당한 축 변수는 「_MC_AX[0-255]」 또는 「_MC1_AX[0-255]」가 됩니다.
  정주기 태스크(실행 우선도 5)에 할당한 축 변수는 「_MC2_AX[0-255]」가 됩니다.

축 변수의 계층과 축 변수명의 변경 예

_MC_AX[0]을 예로 들어 설명합니다. 다른 축 변수도 동일합니다.

```
_MC_AX[0]                          : 축 변수
  _MC_AX[0].Status                 :「축 상태」를 나타내는 계층
    _MC_AX[0].Status.Ready         :「축 기동 준비 완료」를 나타내는 변수
    _MC_AX[0].Status.Disabled      :「축 무효」를 나타내는 변수
        :
  _MC_AX[0].Details                :「축 제어 상태」를 나타내는 계층
    _MC_AX[0].Details.Idle         :「정지 중」을 나타내는 변수
    _MC_AX[0].Details.InPosWaiting :「인포지션 대기」를 나타내는 변수
        :
  _MC_AX[0].Cmd                    :「축 지령값」을 나타내는 계층
    _MC_AX[0].Cmd.Pos              :「지령 현재 위치」를 나타내는 변수
    _MC_AX[0].Cmd.Vel              :「지령 현재 속도」를 나타내는 변수
    _MC_AX[0].Cmd.AccDec           : 축 모니터의「지령 현재 가감 속도」를 나타내는 변수
        :
  _MC_AX[0].Act                    :「축 현재값」을 나타내는 계층
    _MC_AX[0].Act.Pos              :「피드백 현재 위치」를 나타내는 변수
    _MC_AX[0].Act.Vel              :「피드백 현재 속도」를 나타내는 변수
        :
  _MC_AX[0].Cfg                    :「축 기본 설정」을 나타내는 계층
    _MC_AX[0].Cfg.AxNo             :「축 번호」를 나타내는 변수
    _MC_AX[0].Cfg.AxEnable         :「축 사용」을 나타내는 변수
    _MC_AX[0].Cfg.AxType           :「축 종별」을 나타내는 변수
        :
    _MC_AX[0].Scale.Units          :「표시 단위」를 나타내는 변수
_MC_AX[1]                          : 축 변수
        :
```

● 예

「MC_Axis000」을「MyAxis1」로 변경한 경우「피드백 현재 위치」를 나타내는 변수는「MyAxis1.Act.Pos」와「_MC_AX[0].Act.Pos」를 모두 취급할 수 있습니다.

축 변수에 대한 상세한 내용은 📖「축 변수 (P.6-21)」를 참조해 주십시오.

3-1-4 축 변수의 동기

프라이머리 정주기 태스크에서 제어하는 축의 축 변수값을 정주기 태스크(실행 우선도 5)의 사용자 프로그램에서 참조할 수 있습니다. 또한, 반대로 정주기 태스크(실행 우선도 5)에서 제어하는 축의 축 변수값을 프라이머리 정주기 태스크의 사용자 프로그램에서 참조할 수 있습니다.
또한, 프라이머리 정주기 태스크에서 제어하는 축 변수의 값을 정주기 태스크(실행 우선도 16)의 사용자 프로그램에서 참조할 수도 있습니다.
참조할 태스크의 종류 및 축 변수의 관계는 다음과 같습니다.

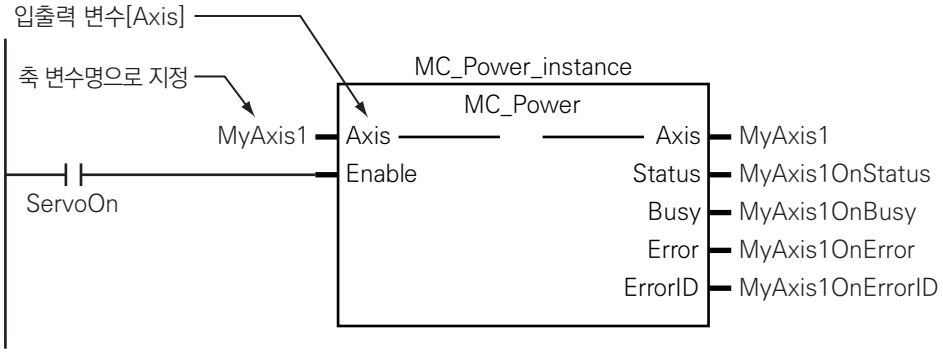
○: 참조 가능 ×: 참조 불가

태스크의 종류	모션 제어 1의 축 변수		모션 제어 2의 축 변수	
	사용자 정의 *1	시스템 정의 *2	사용자 정의*1	시스템 정의 *3
프라이머리 정주기 태스크	○	○	○ *4	×
정주기 태스크 (실행 우선도 5)	○ *5	×	○	○
정주기 태스크 (실행 우선도 16)	○ *5	○ *6	○ *4	×
정주기 태스크 (실행 우선도 17, 18)	○ *5	○ *7	○ *4	○ *8
이벤트 태스크(8, 48)	○ *5	○ *7	○ *4	○ *8

*1. Sysmac Studio에서 「축」을 작성했을 때 자동 생성되는 사용자 정의의 축 변수 「MC_Axis000」이나 「Loader1」 등을 말합니다. 사용자 정의의 축 변수를 사용하는 것을 권장합니다.
*2. 「_MC_AX[*]」나 「_MC1_AX[*]」를 말합니다.
*3. 「_MC2_AX[*]」를 말합니다.
*4. 정주기 태스크(실행 우선도 5)의 태스크 주기마다 1축분의 축 변수를 처리합니다.
*5. 프라이머리 주기마다 1축분의 축 변수를 처리합니다.
*6. 프라이머리 주기마다 최대 축 수분의 축 변수를 처리합니다. 프라이머리 정주기 태스크의 태스크 실행 시간이 증가하므로 권장하지 않습니다.
*7. 프라이머리 주기마다 최대 축 수분의 축 변수를 처리합니다.
*8. 정주기 태스크(실행 우선도 5)의 태스크 주기마다 최대 축 수분의 축 변수를 처리합니다.

3-1-5 사용자 프로그램에서 축 지정하는 방법

사용자 프로그램에서는 모션 제어 명령의 입출력 변수 「Axis」에 축 변수명을 지정합니다.
아래 그림은 시스템 정의 변수의 축 변수명이 「_MC_AX[0]」인 축에 대해 Sysmac Studio에서 추가했을 때의 축 변수명을 「MyAxis1」로 변경한 경우의 예입니다.



「MyAxis1」 대신에 시스템 정의 변수인 「_MC_AX[0]」도 사용할 수 있습니다.

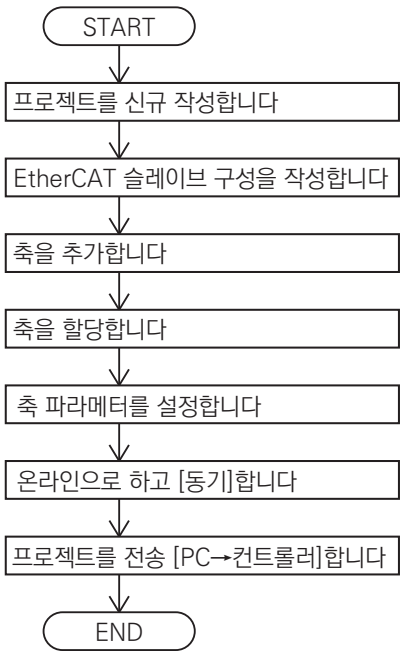
모션 제어 명령에 대한 상세한 내용은 📖 「6-2 모션 제어 명령 (P.6-5)」을 참조해 주십시오.

각 모션 제어 명령의 상세 기능에 대해서는 📖 『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』에서 각 명령의 설명을 참조해 주십시오.

3-2 축 설정 순서

여기에서는 Sysmac Studio를 사용하여 새로운 서보 축을 설정하는 순서에 대해 설명합니다.

3-2-1 축 설정 흐름도



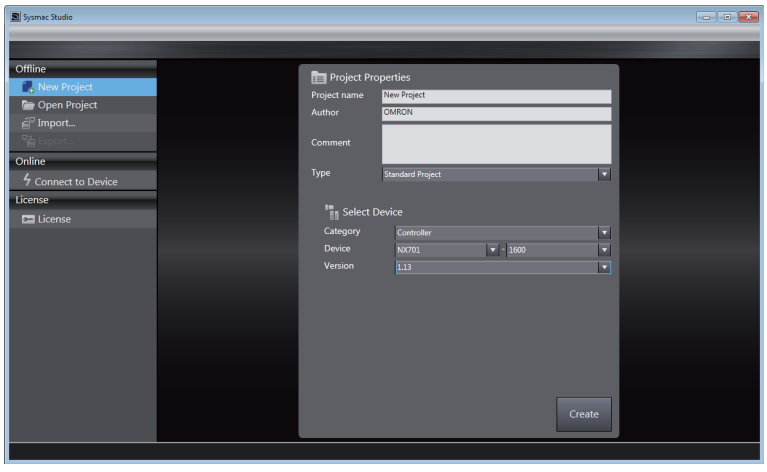
3-2-2 설정 순서

여기에서는 축을 설정하는 방법에 대해 설명합니다.

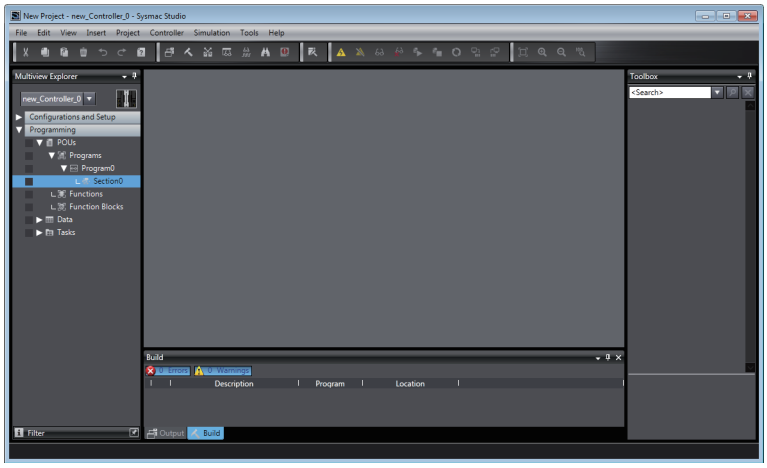
Sysmac Studio의 기동

1 Sysmac Studio를 기동하고 [New Project] 버튼을 클릭합니다.

2 프로젝트의 속성과 디바이스 선택을 설정하고 [Create] 버튼을 클릭합니다.



신규 프로젝트 화면이 표시됩니다.



EtherCAT 네트워크 구성의 작성

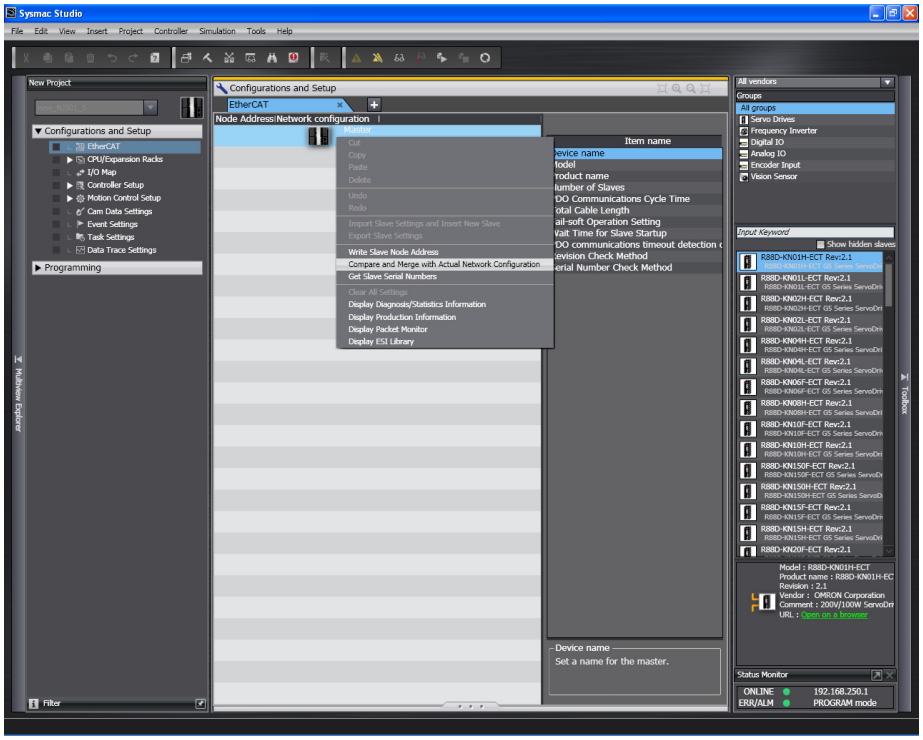
EtherCAT 네트워크 구성을 작성하는 방법은 온라인에서 실행하는 방법과 오프라인에서 실행하는 방법 등 2가지가 있습니다.

● 온라인을 이용한 방법

1 **멀티 뷰 익스플로러에서 [EtherCAT]을 더블 클릭합니다.**
EtherCAT 구성 편집 화면이 표시됩니다.

2 **[Controller] 메뉴의 [Online]을 클릭하여 온라인 상태로 설정합니다.**

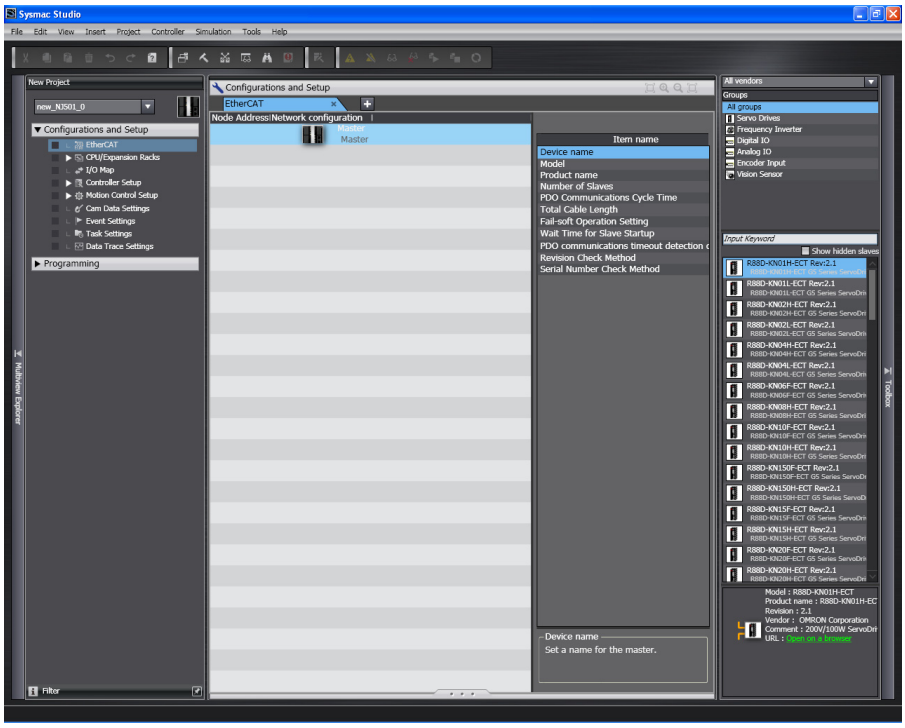
3 **화면 내 [Master] 아이콘 위에서 우클릭하고 [Compare and Merge with Actual Network Configuration]을 선택합니다.**



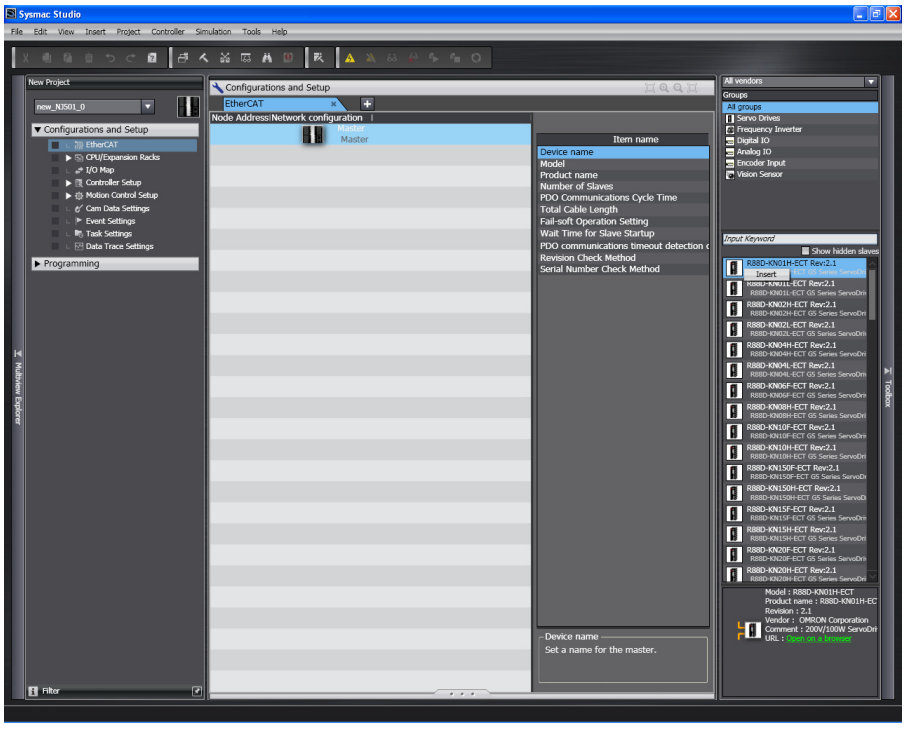
정보 취득이 종료되면 EtherCAT 슬레이브의 실제 장비 구성이 표시됩니다.
표시된 실제 장비 구성상에서 우클릭하고 [Apply actual network configuration]을 클릭합니다.

● 오프라인에서 작성하는 경우

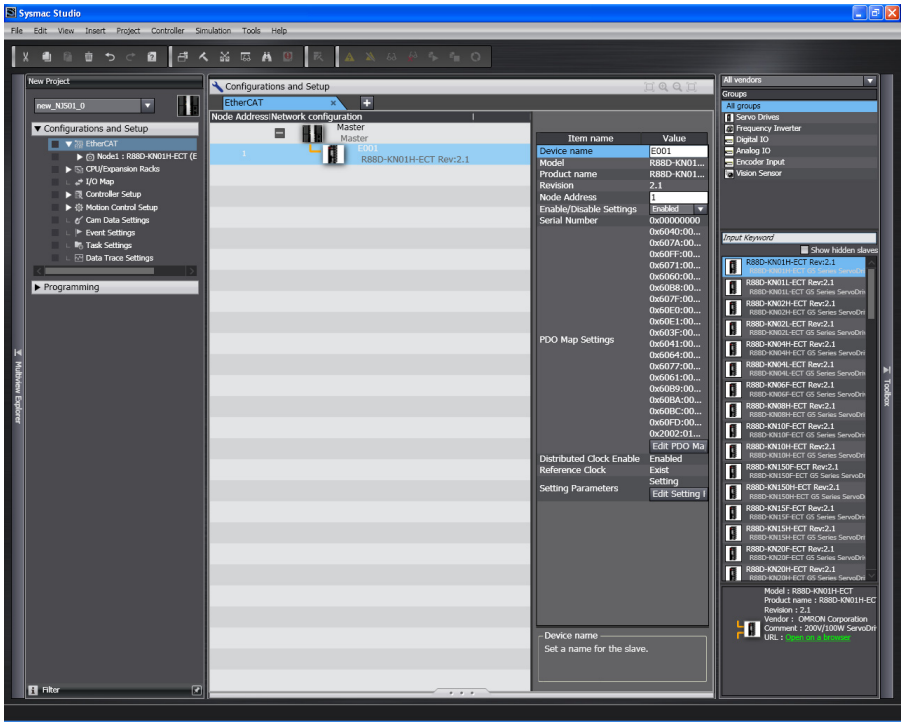
1 멀티 뷰 익스플로러에서 [EtherCAT]을 더블 클릭합니다.
EtherCAT 구성 편집 화면이 표시됩니다.



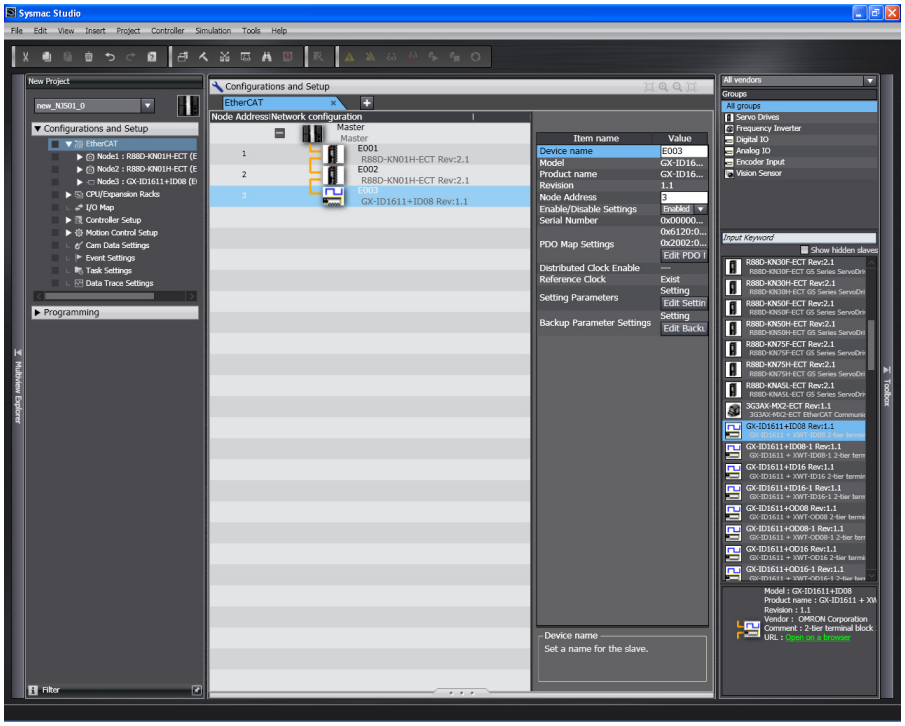
2 접속할 슬레이브를 우클릭하고 [Insert]를 클릭합니다.



삽입한 슬레이브가 표시됩니다.



3 사용할 모든 슬레이브를 삽입합니다.

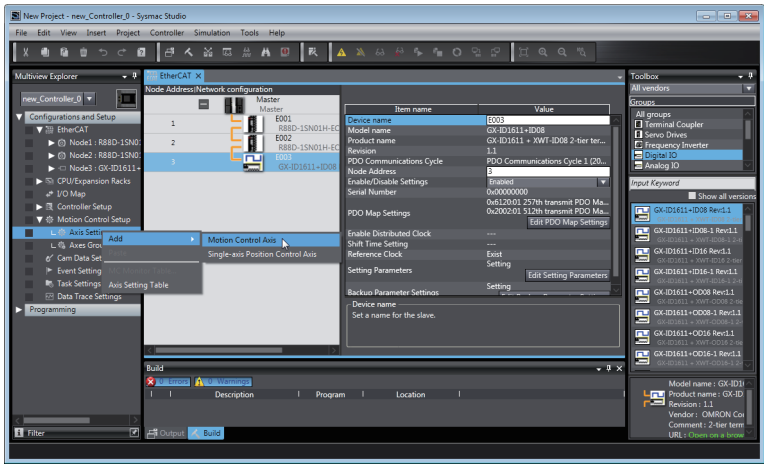


축 추가

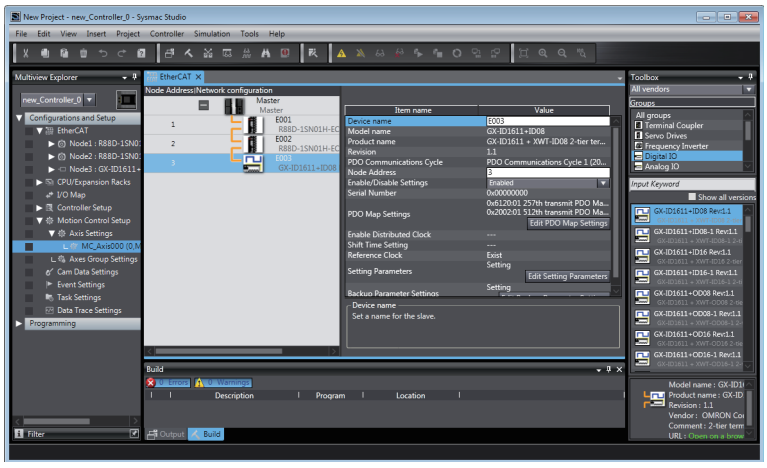
1 멀티 뷰 익스플로러에서 [Axis Settings]를 우클릭하고 [Add]에서 [Motion Control Axis] 또는 [Single-axis Position Control Axis]를 클릭합니다.

참고

[Single-axis Position Control Axis]는 NX102 및 NX1P2 CPU 유니트인 경우에 표시됩니다.



멀티 뷰 익스플로러에 축이 추가됩니다.
추가된 축 변수명의 디폴트는 「MC_Axis000」입니다.

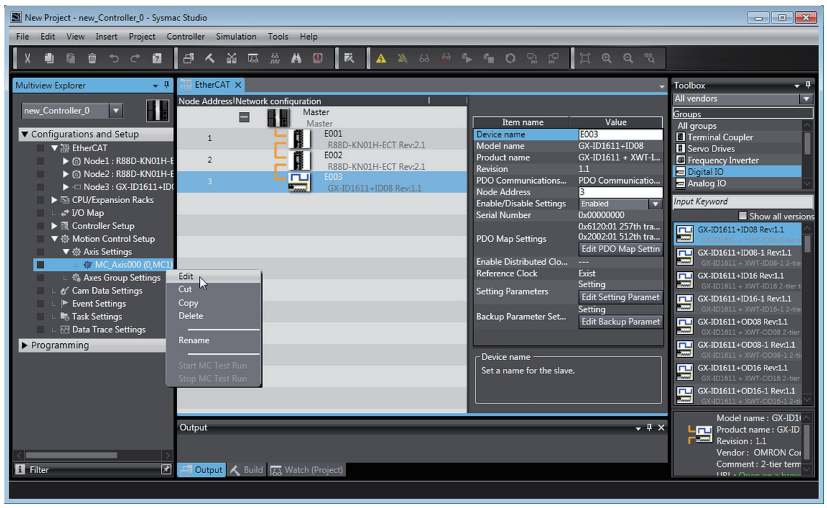


● 복사를 이용한 축 추가

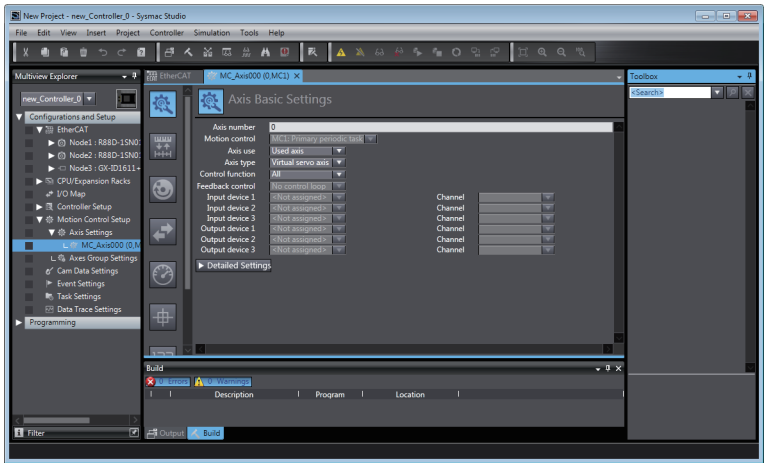
기존의 축 설정을 복사하여 붙여넣는 방법으로도 축을 추가할 수 있습니다.

축 할당

1 멀티 뷰 익스플로러에 추가된 축을 우클릭하고 [Edit]를 클릭합니다.



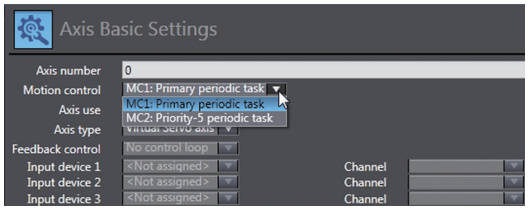
축 파라메터 설정 화면의 [Axis Basic Settings] 화면이 표시됩니다.



참고

[Control Function]은 NX102 및 NX1P2 CPU 유니트인 경우에 표시됩니다.

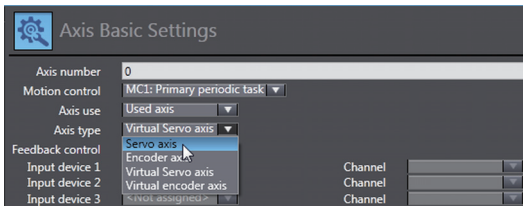
2 [Motion control]에서 [Primary periodic task] 또는 [Priority-5 periodic task]를 선택합니다.



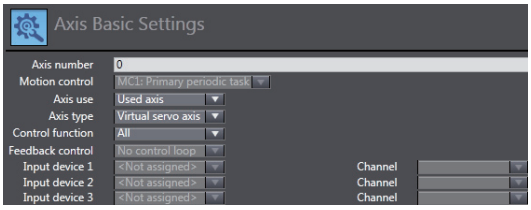
참고

NX701 CPU 유니트의 경우에 설정합니다.

3 [Axis type]에서 [Servo axis]를 선택합니다.



4 [Control Function]에서 [All]을 선택합니다.

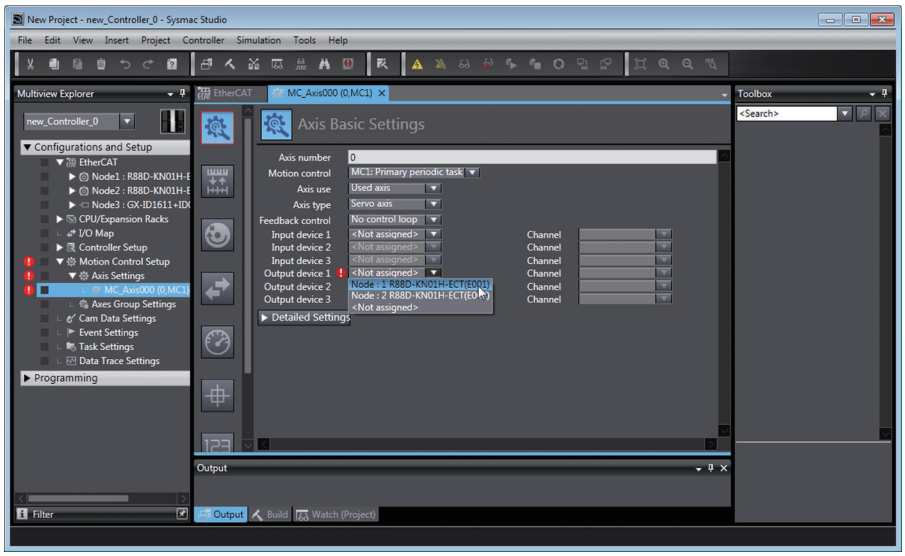


참고

- NX102 및 NX1P2 CPU 유니트의 경우에 설정합니다.
- 모션 제어 축으로 사용할 경우에는 「All」을 선택합니다. 단축 위치 제어 축으로 사용할 경우에는 「Single-axis position control only」를 선택해 주십시오.

5 **사용할 서보 드라이버를 선택합니다.**

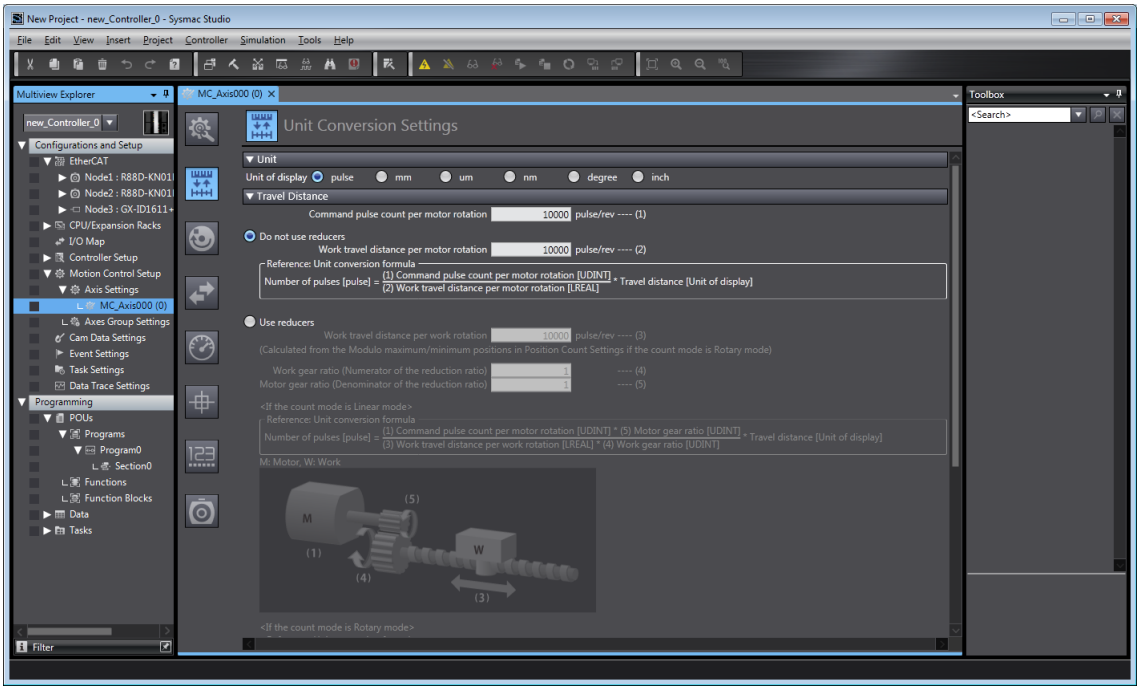
이 조작에 의해 서보 드라이버를 축으로 사용할 수 있습니다.



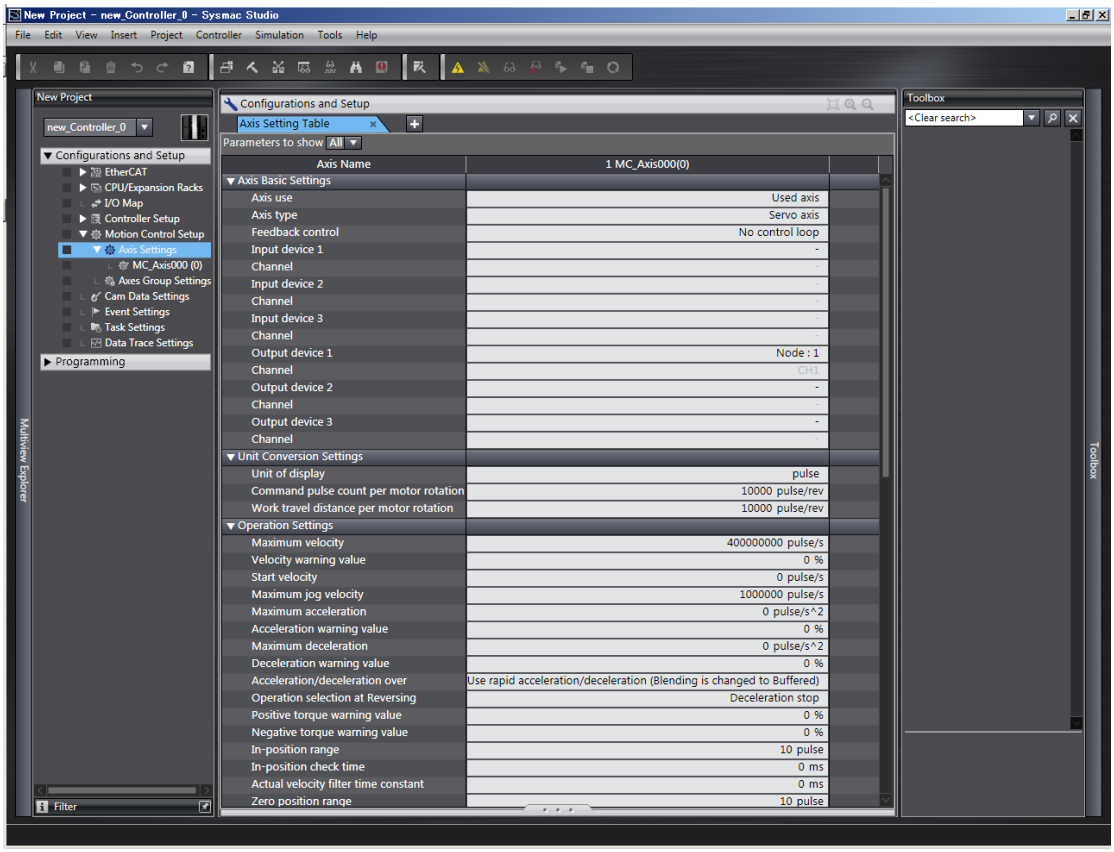
축 파라메터의 설정

축 파라메터 설정 화면에 있는 아이콘을 클릭합니다.

아이콘마다 분류된 축 파라메터 설정 화면이 표시됩니다.



멀티 뷰 익스플로러의 [Axis Settings]를 우클릭하고 [Axis Setting Table]을 클릭하면 전체 축의 축 파라메터를 일람 화면에서 설정할 수도 있습니다.



사용상의 주의

표시 단위나 전자 기어(단위 변환식) 및 최고 속도와 최대 가감 속도 등의 동작 설정은 장비의 동작 조건에 따라 적절한 값을 설정해 주십시오.

참고

사용자 프로그램에서 사용 중인 축 변수명의 변경

이미 사용 중인 축 변수명을 변경하려면 다음 2가지 조작을 실행해 주십시오.

- 변수 선언 내의 변수 테이블에 있는 축 변수명 변경.
- 사용자 프로그램의 축 변수명 변경.

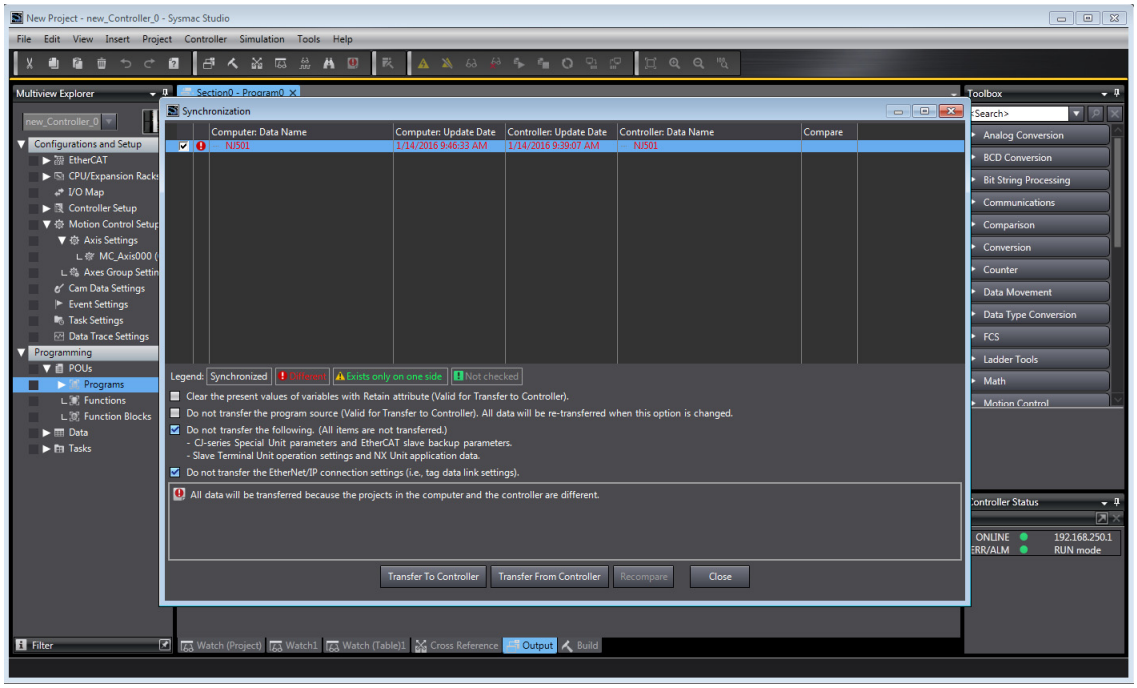
변수 테이블의 축 변수명을 변경해도 사용자 프로그램의 축 변수명에는 반영되지 않습니다.
변수 테이블에서 선언되지 않은 변수명을 사용자 프로그램에서 사용하면 이상이 발생합니다.
반드시 둘 다 변경해 주십시오.

CPU 유니트로 다운로드하기

CPU 유니트에 다운로드하는 경우는 Sysmac Studio의「Synchronization」기능을 사용합니다.

1 [Controller] 메뉴의 [Online]을 클릭하여 온라인 상태로 설정합니다.

2 [Controller] 메뉴에서 [Synchronization]을 선택하고 [Transfer To Controller]을 클릭합니다.



참고

MC 기능 모듈은 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입, G5 시리즈 EtherCAT 통신 내장 타입, NX 시리즈 펄스 출력 유니트를 접속 대상으로 합니다.

서보 드라이버의 접속 대상 기종

- 서보 드라이버의 대상 기종 형식은 R88D-1SN□□□-ECT 및 R88D-KN□□□-ECT, R88D-KN□□□-ECT-L입니다.
- R88D-KN□□□-ECT-R은 위치 제어(cyclic synchronous position) 모드에만 대응하는 서보 드라이버이므로, 속도 제어(cyclic synchronous velocity) 모드 또는 토크 제어(cyclic synchronous torque) 모드를 사용하는 기능은 사용할 수 없습니다.

서보 드라이버 측의 설정

- MC 기능 모듈은 서보 드라이버 측의 입력 신호나 일부 기능을 사용합니다. MC 기능 모듈의 기능을 올바르게 사용하기 위해서는 서보 드라이버 측의 신호를 배선하거나 오브젝트를 설정해야 합니다.

  설정 내용에 대해서는 「A-1 서보 드라이버 1S 시리즈와의 접속 (P.A-2)」 또는 「A-2 서보 드라이버 G5 시리즈와의 접속 (P.A-11)」을 참조해 주십시오.

NX 시리즈 펄스 출력 유니트를 사용할 경우의 설정은 『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

3-3 축 그룹

여기에서는 MC 기능 모듈의 축 그룹에 대해 설명합니다.

3-3-1 축 그룹의 개요

직선 보간이나 원호 보간 등 복수의 축을 동시에 작동시킬 때 축 그룹을 사용합니다.
따라서 축 그룹은 복수의 축으로 구성되어 있습니다.
Sysmac Studio를 사용하여 축 그룹 변수를 설정하면 축 그룹의 모션 제어 명령을 기동하거나 축 그룹의 상태를 취득할 수 있습니다.
MC 기능 모듈에서는 최대로 64그룹까지 취급할 수 있습니다.
축 그룹의 사양 개요는 아래와 같습니다.

항목	사양			
	NX701	NX102	NX1P2	NJ 시리즈
축 그룹 수	최대 64그룹	최대 8그룹	최대 8그룹	최대 32그룹
구성 축 수	최대 4축/축 그룹			

MC 기능 모듈의 축 그룹에는 아래와 같은 관련 요소가 있습니다.

구성 요소	내용	참조 페이지
축 그룹 파라메터	MC 기능 모듈로 제어하는 각 축 그룹에 관하여 최고 보간 속도와 최대 보간 가감 속도와 같은 축 그룹 동작에 관한 설정을 합니다. 축 그룹 파라메터를 설정하려면 Sysmac Studio를 사용합니다.	P. 3-20
축 그룹 변수	축 그룹 변수는 MC 기능 모듈로 제어하는 각 축 그룹에 대해 축 그룹 파라메터의 일부와 지령 보간 속도 및 이상 정보 등의 모니터 정보로 이루어진 시스템 정의 변수입니다. Sysmac Studio의 멀티 뷰 익스플로러에서 축 그룹을 추가하면 축 그룹 변수가 작성됩니다. 축 그룹 변수의 이름(이하, 축 그룹 변수명이라 합니다)은 이 뷰에서 설정할 수 있습니다.	P. 3-21
사용자 프로그램에서 축 그룹 지정하는 방법	사용자 프로그램에서는 모션 제어 명령을 사용하여 모션 제어를 실시합니다. 다축 협조 제어를 하는 모션 제어 명령을「축 그룹 지령」이라 부릅니다. 축 그룹 지령으로 어떤 축 그룹을 제어할지는 명령의 입출력 변수「AxesGroup(축 그룹)」에 시스템 정의 변수의 축 그룹 변수명 또는 Sysmac Studio에서 설정한 축 그룹 변수명을 지정합니다.	P. 3-22

사용상의 주의

EtherCAT 네트워크상의 축과 CPU 유니트상의 축으로 축 그룹을 구성하지 마십시오.
네트워크끼리 동기되지 않아 지령 타이밍에 오차가 발생하여 의도치 않은 동작을 할 가능성이 있습니다.

3-3-2 축 그룹 파라메터의 개요

● 축 그룹 파라메터의 일람

구분	파라메터의 명칭
축 그룹 기본 설정	축 그룹 번호
	모션 제어 *1
	축 그룹 사용
	축 구성
	구성 축
축 그룹 동작 설정	최고 보간 속도
	최대 보간 가속도
	최대 보간 감속도
	보간 가감속 초과
	보간 속도 경고값
	보간 가속도 경고값
	보간 감속도 경고값
	축 그룹 정지 방법
	중심점 보정 허용률

*1. NX701 CPU 유니트의 경우에 설정합니다.

축 그룹 파라메터에 대한 상세한 내용은 「5-3 축 그룹 파라메터 (P.5-30)」를 참조해 주십시오.

● 축 그룹을 사용하기 위한 필수 설정 항목

Sysmac Studio에서 작성한 축 그룹에 대해 다음 항목을 설정하면 축 그룹을 사용할 수 있습니다.

구분	파라메터의 명칭	설정 내용	참조페이지
축 그룹 기본 설정	축 그룹 번호	축 그룹을 작성한 순서대로 자동으로 설정됩니다.	P. 5-31
	모션 제어 *1	[Primary periodic task]를 선택합니다.	
	축 그룹 사용	[Use]를 선택합니다.	
	축 구성	제어할 축 구성을 선택합니다.	
	구성 축	축 그룹에 할당할 축을 선택합니다.	

*1. NX701 CPU 유니트의 경우에 설정합니다.

사용상의 주의

- 최고 보간 속도나 정지 방법 등은 동작 조건에 따라 적절한 값을 설정해 주십시오.
- [Control Function] 설정이 [Single-axis position control only]인 축은 축 그룹의 구성 축에 할당되지 않습니다.

3-3-3 축 그룹 변수의 개요

축 그룹 변수는 MC 기능 모듈 각 축 그룹의 설정 정보와 현재 위치 및 이상 정보 등의 모니터 정보로 이루어진 시스템 정의 변수입니다.
Sysmac Studio에서 축 그룹을 작성하면 작성한 순서대로 축 그룹 변수가 변수 테이블에 등록됩니다. 축 그룹 변수의 데이터형은 구조체형으로 데이터형명은 「_sGROUP_REF」입니다.

축 그룹 변수명

MC 기능 모듈의 축 그룹 변수명에는 시스템 정의 변수의 축 그룹 변수명과 Sysmac Studio에서 축 그룹을 추가했을 때의 축 그룹 변수명 등 2종류가 있습니다.
시스템 정의 변수의 축 그룹 변수명은 「_MC_GRP[0]~_MC_GRP[63]」, 「_MC1_GRP[0]~_MC1_GRP[63]」, 「_MC2_GRP[0]~_MC2_GRP[63]」입니다.
Sysmac Studio에서 축 그룹을 추가하면 「_MC_GRP[0]~_MC_GRP[63]」에 「MC_Group000~MC_Group063」이라는 디폴트 명칭이 설정됩니다. 번호는 추가한 순서대로 지정됩니다. 이러한 축 그룹 변수명은 Sysmac Studio에서 축 그룹마다 임의의 변수명으로 변경할 수 있습니다.
사용자 프로그램에서는 시스템 정의 변수의 축 그룹 변수명, Sysmac Studio에서 추가한 축 그룹 변수명을 모두 사용할 수 있습니다.

● _MC_GRP[0-63]을 사용한 경우의 예

시스템 정의 변수의 축 그룹 변수명 (글로벌 변수 뷰의 할당처 *1)	Sysmac Studio에서 추가한 축 그룹 변수명의 디폴트	축 그룹 번호 예
_MC_GRP[0]	MC_Group000	축 그룹 0
_MC_GRP[1]	MC_Group001	축 그룹 1
:	:	:
_MC_GRP[63]	MC_Group063	축 그룹 63

*1. Sysmac Studio의 글로벌 변수 뷰에 있는 할당처란의 명칭을 변경하면 이상이 발생합니다.

참고

- NX701 CPU 유니트의 경우는 「_MC_GRP[0-63]」 또는 「_MC1_GRP[0-63]」과 「_MC2_GRP[0-63]」을 사용할 수 있습니다.
  NX102 및 NX1P2 CPU 유니트의 경우는 「_MC_GRP[0-7]」만 사용할 수 있습니다.
  NJ 시리즈 CPU 유니트의 경우는 「_MC_GRP[0-31]」만 사용할 수 있습니다.
- NX701 CPU 유니트의 경우 「_MC_GRP[0-63]」과 「_MC1_GRP[0-63]」은 축 그룹 번호가 같으면 같은 값을 참조합니다. 어떠한 축 그룹 변수라도 사용 가능하며 혼재시켜 사용하는 것도 가능합니다.
- 프라이머리 정주기 태스크에 할당한 축 그룹 변수는 「_MC_GRP[0-63]」 또는 「_MC1_GRP[0-63]」이 됩니다.
  정주기 태스크(실행 우선도 5)에 할당한 축 그룹 변수는 「_MC2_GRP[0-63]」이 됩니다.

축 그룹 변수의 계층과 축 그룹 변수명의 변경 예

_MC_GRP[0]을 예로 들어 설명합니다. 다른 축 그룹 변수도 동일합니다.

_MC_GRP[0]	: 축 그룹 변수
_MC_GRP[0].Status	:「축 그룹 상태」를 나타내는 계층
:	
_MC_GRP[0].Cmd	:「축 그룹 지령값」을 나타내는 계층
_MC_GRP[0].Cmd.Vel	:「지령 보간 속도」를 나타내는 변수
_MC_GRP[0].Cmd.AccDec	:「지령 보간 가감 속도」를 나타내는 변수
:	
_MC_GRP[0].Cfg	:「축 그룹 기본 설정」을 나타내는 계층
_MC_GRP[0].Cfg.GrNo	:「축 그룹 번호」를 나타내는 변수
_MC_GRP[0].Cfg.GrEnable	:「축 그룹 사용」을 나타내는 변수
_MC_GRP[0].Kinematics	:「운동학 변환 설정」을 나타내는 계층
_MC_GRP[0].Kinematics.GrType	:「축 구성」을 나타내는 변수
_MC_GRP[0].Kinematics.Axis[0]	:「구성 축(축 A0)」의 축 번호를 나타내는 변수
:	
_MC_GRP[0].Kinematics.Axis[3]	:「구성 축(축 A3)」의 축 번호를 나타내는 변수
_MC_GRP[1]	: 축 그룹 변수
:	

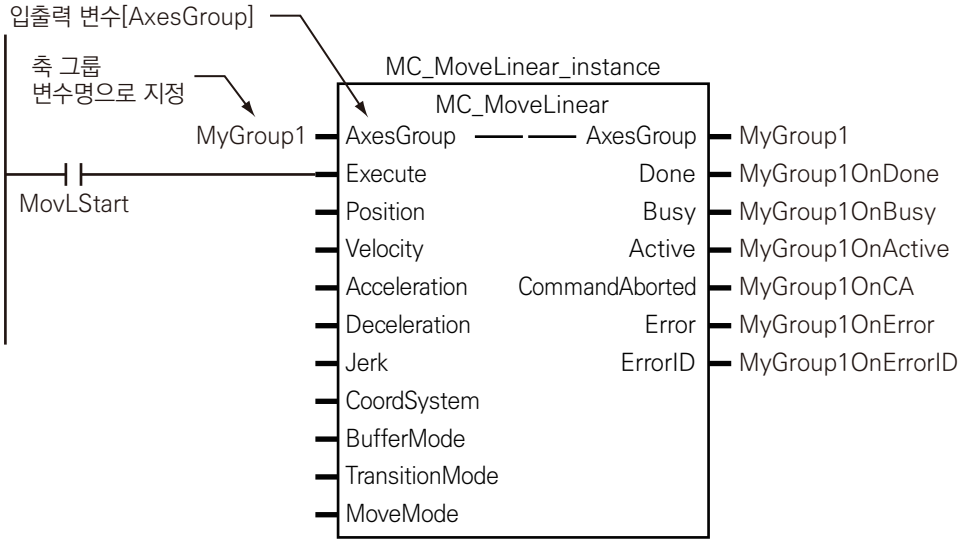
● **예**

「MC_Group000」을「MyGroup1」로 변경한 경우「지령 보간 속도」를 나타내는 변수는「MyGroup1.Cmd.Vel」과「_MC_GRP[0].Cmd.Vel」을 모두 취급할 수 있습니다.

축 그룹 변수에 대한 상세한 내용은 🕮「축 그룹 변수 (P.6-28)」를 참조해 주십시오.

3-3-4 사용자 프로그램에서 축 그룹 지정하는 방법

사용자 프로그램에서는 모션 제어 명령의 입출력 변수「AxesGroup」에 축 그룹 변수명을 지정합니다. 아래 그림은 시스템 정의 변수의 축 변수명이「_MC_GRP[0]」인 축 그룹에 대해 Sysmac Studio에서 추가했을 때의 축 그룹 변수명을「MyGroup1」로 변경한 경우의 예입니다.



「MyGroup1」대신에 시스템 정의 변수인「_MC_GRP[0]」도 사용할 수 있습니다.

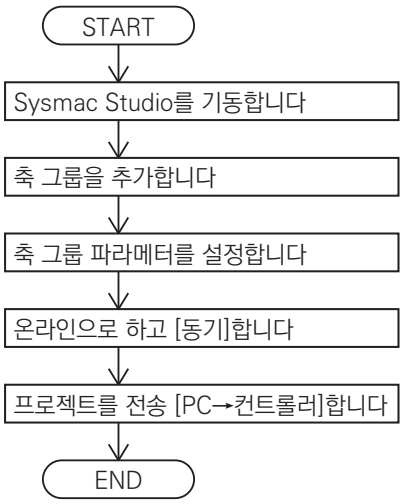
모션 제어 명령에 대한 상세한 내용은 「6-2 모션 제어 명령 (P.6-5)」을 참조해 주십시오.

각 모션 제어 명령의 상세 기능에 대해서는 『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』에서 각 명령의 설명을 참조해 주십시오.

3-4 축 그룹 설정 순서

여기에서는 Sysmac Studio를 사용하여 축 그룹을 설정하는 순서에 대해 설명합니다.
직선 보간 및 원호 보간 등의 축 그룹 지령 명령을 사용하지 않는 경우에는 설정할 필요가 없습니다.

3-4-1 축 그룹 설정 흐름도

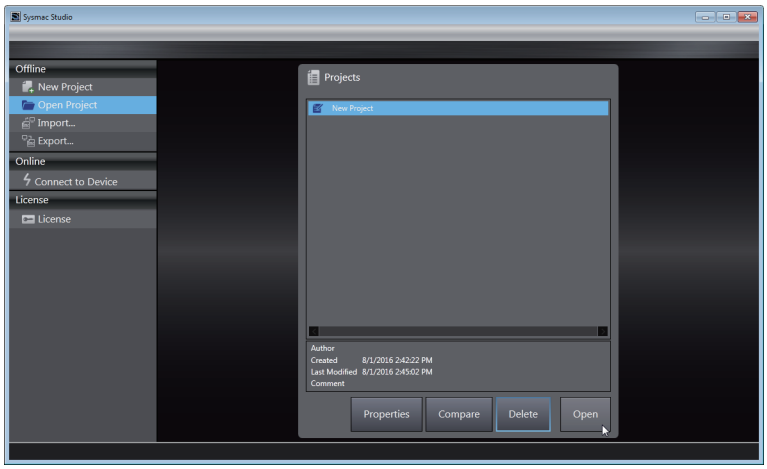


3-4-2 설정 순서

여기에서는 이미 축이 추가된 기존 프로젝트에서 축 그룹을 설정하는 방법에 대해 설명합니다.

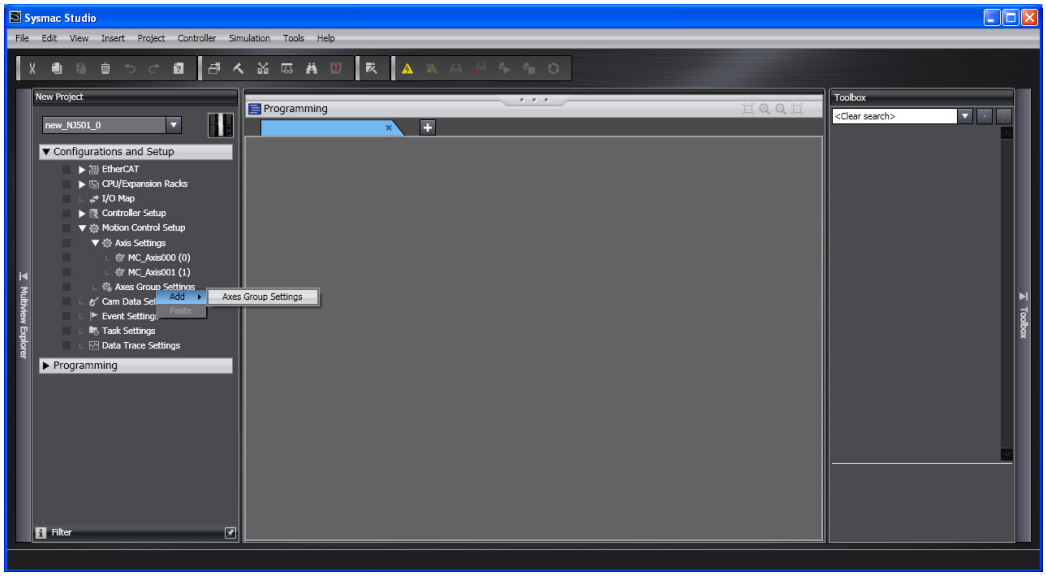
Sysmac Studio의 기동

Sysmac Studio를 기동시켜 기존 프로젝트를 엽니다.

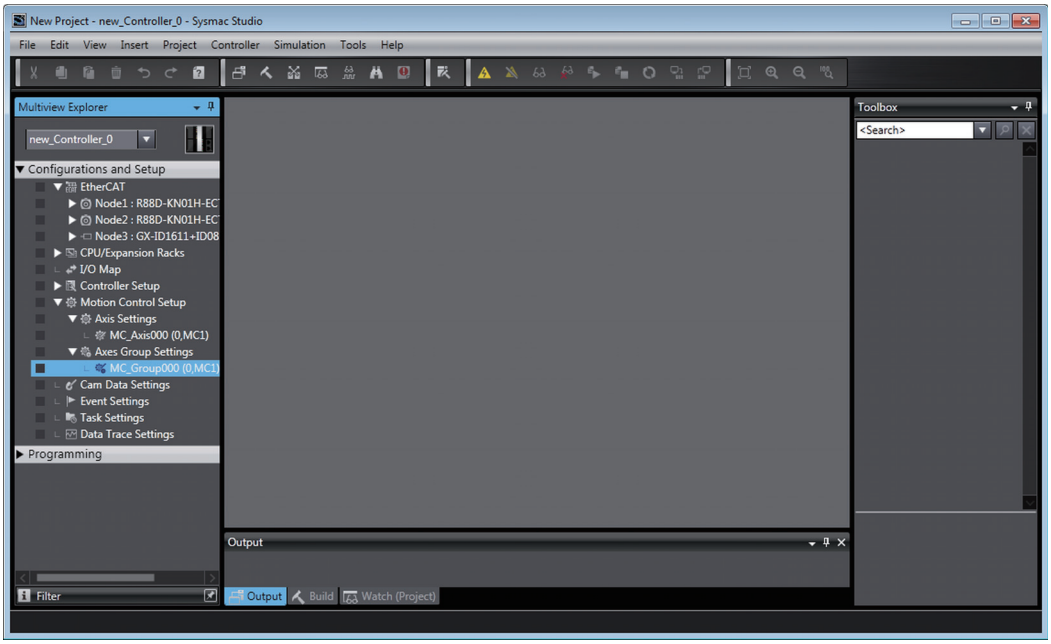


축 그룹 추가

멀티 뷰 익스플로러에서 [Axes Group Settings]를 우클릭하고 [Add]에서 [Axes Group Settings]를 클릭합니다.



멀티 뷰 익스플로러에 축 그룹이 추가됩니다.
추가된 축 그룹 변수명의 디폴트는「MC_Group000」입니다.

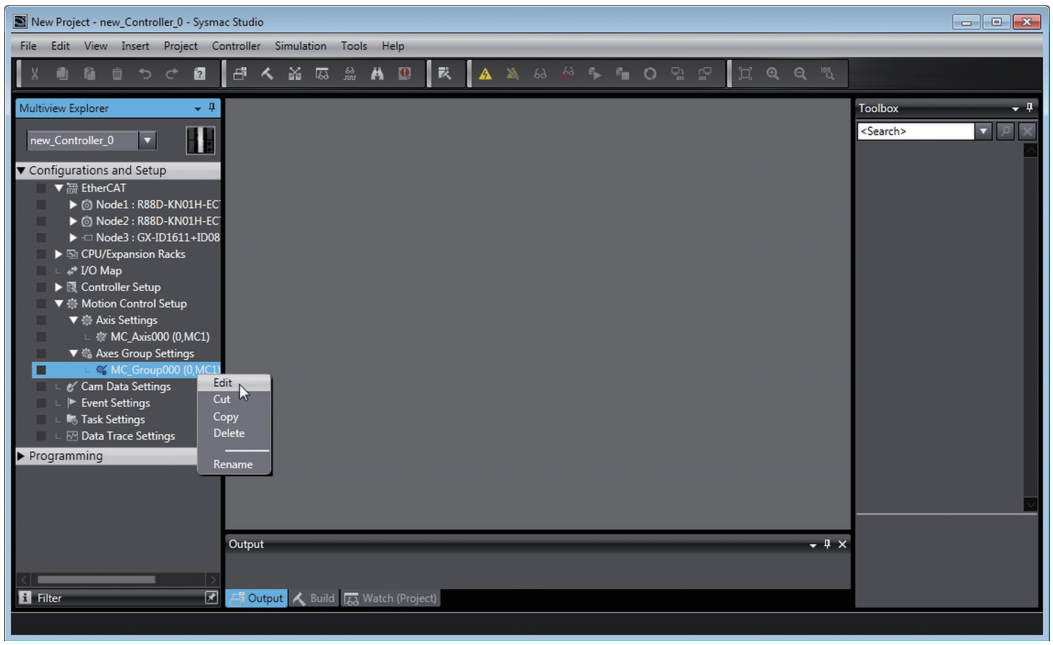


● 복사를 이용한 축 작성

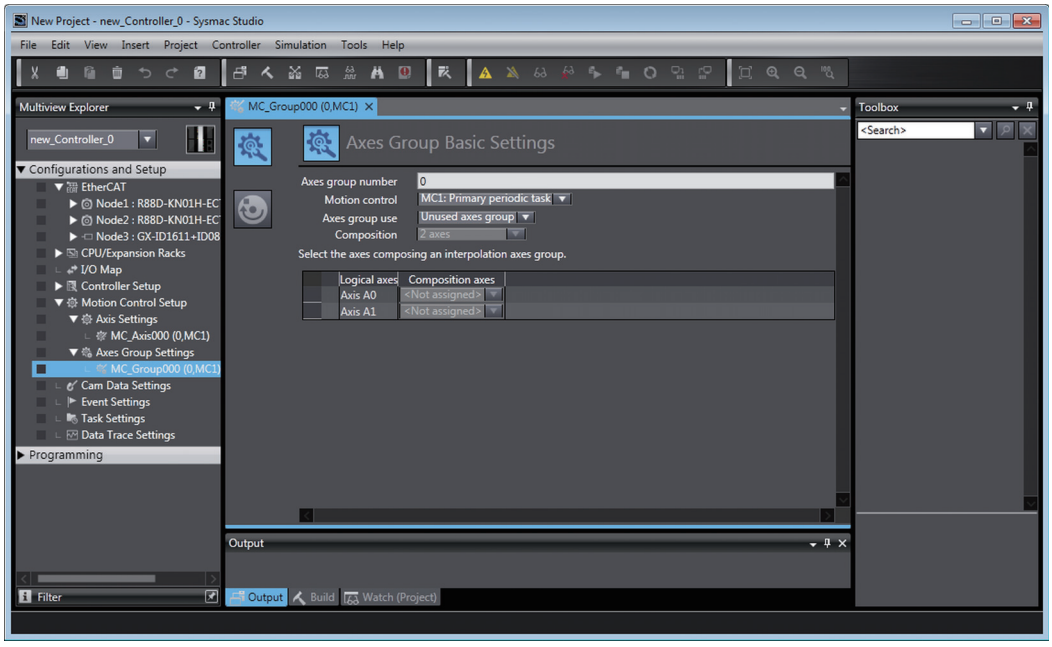
기존 프로젝트의 축 그룹을 복사하여 축 그룹을 새로 작성할 수도 있습니다.

축 그룹 파라메터의 설정

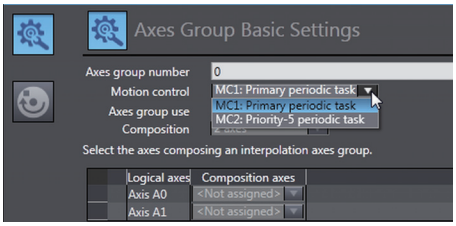
1 멀티 뷰 익스플로러에 추가된 축 그룹을 우클릭하고 [Edit]를 클릭합니다.



축 그룹 파라메터 설정 화면의 [Axes Group Basic Settings] 화면이 표시됩니다.



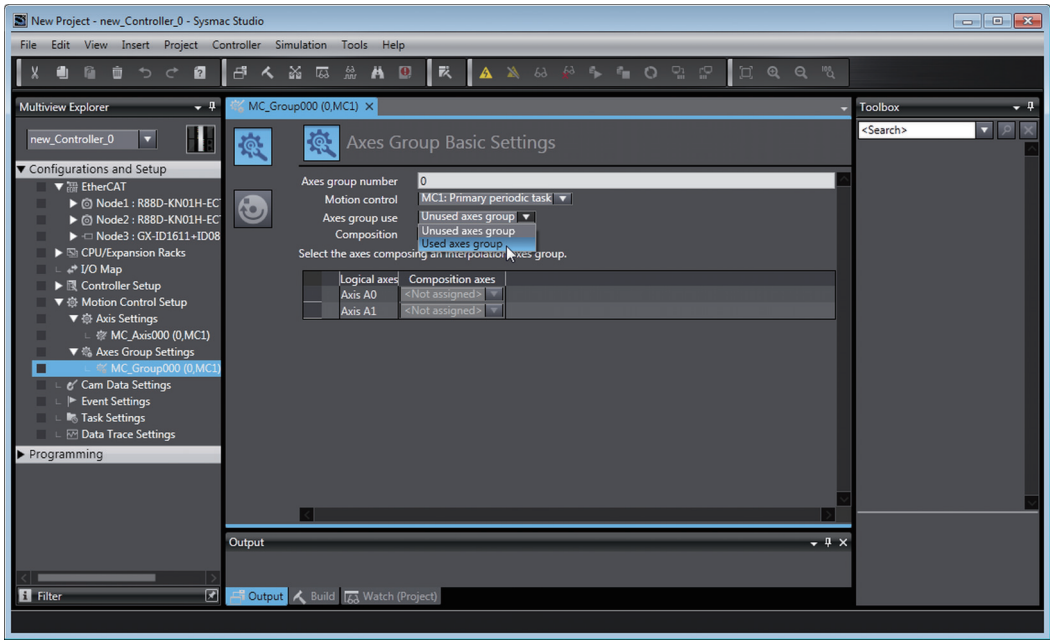
2 [Motion control]에서 [Primary periodic task] 또는 [Priority-5 periodic task]를 선택합니다.



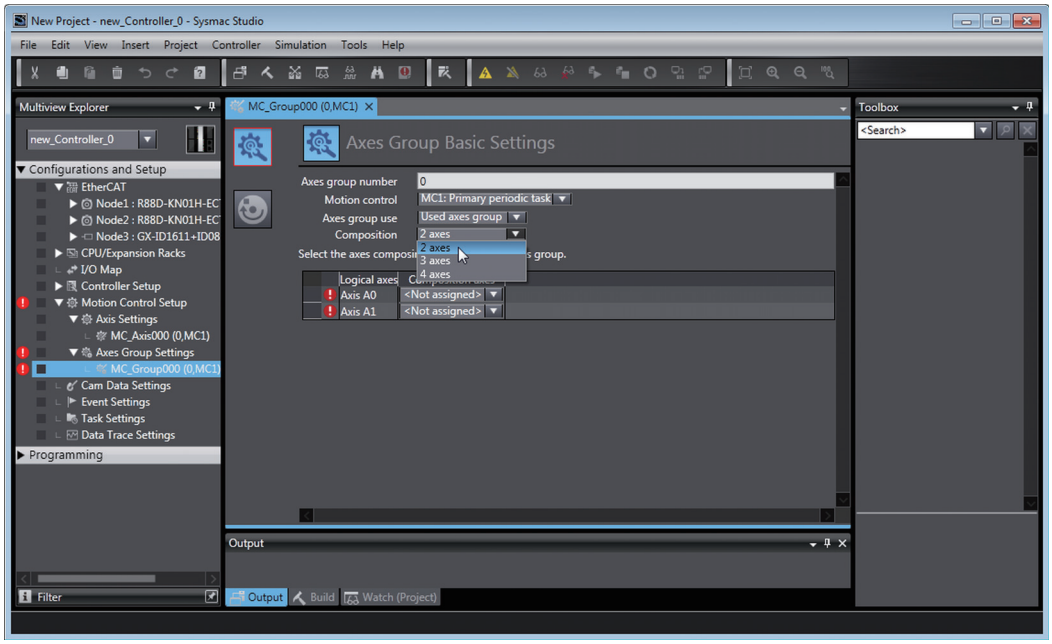
참고

NX701 CPU 유니트의 경우에 설정합니다.

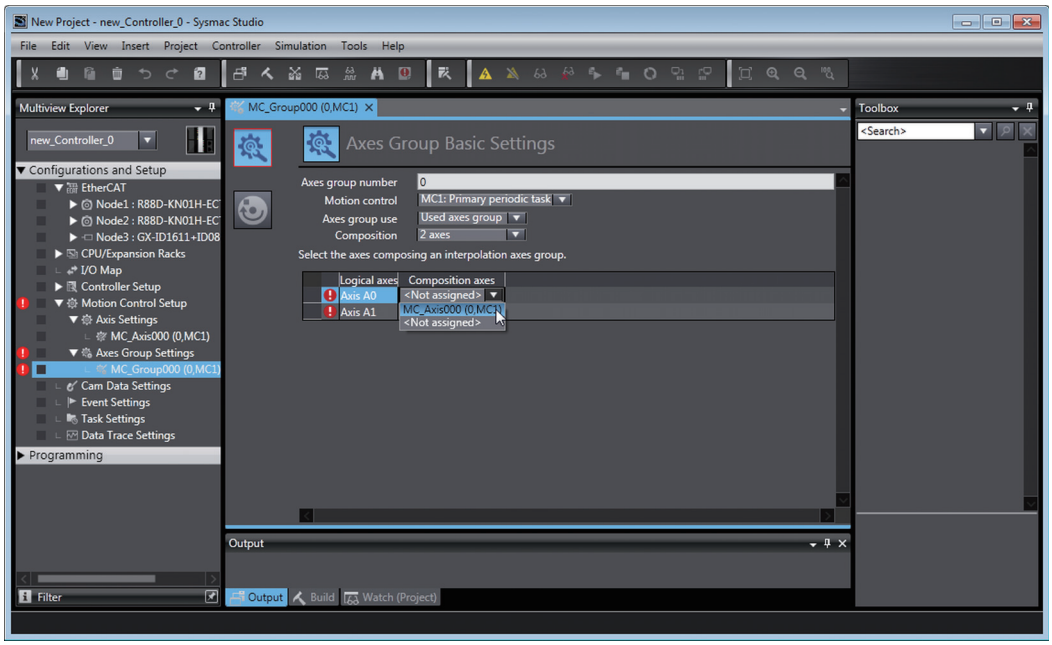
3 [Axes group use]에서 [Used axes group]을 선택합니다.



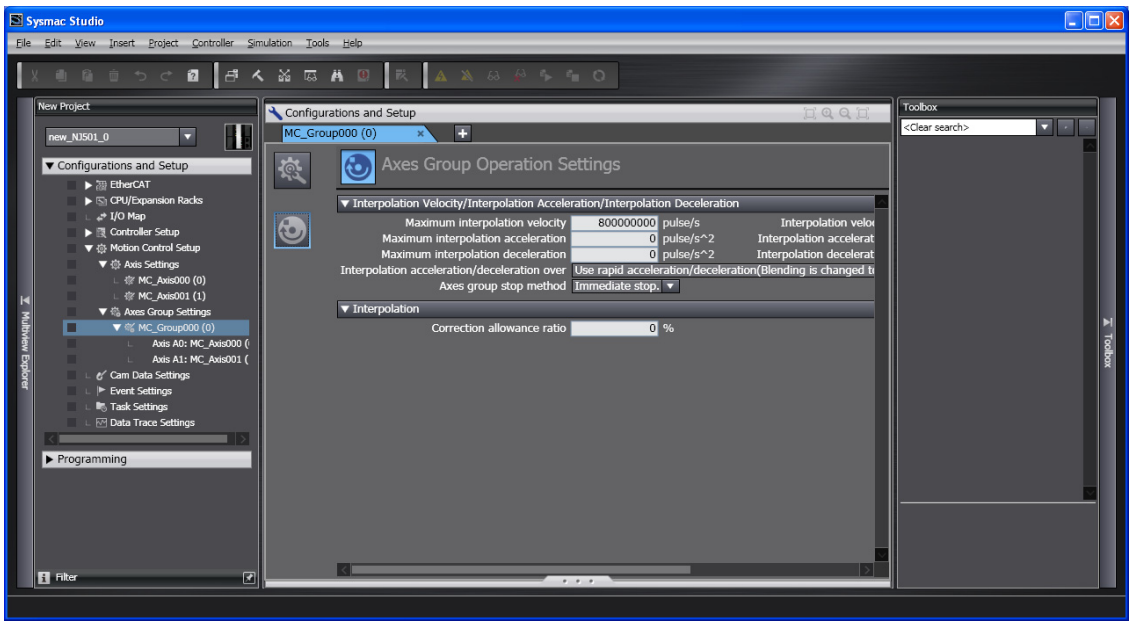
4 [Composition]에서 축 그룹의 구성을 선택합니다. 아래는 2축 구성을 선택한 경우입니다.



5 [Logical axes]에 사용할 축을 할당합니다.



6 **아래 아이콘을 클릭하면 [Axes Group Operation Settings] 화면이 표시됩니다.**
각 설정은 장비의 동작 조건에 따라 적절한 값을 설정해 주십시오.



참고

사용자 프로그램에서 사용 중인 축 그룹 변수명의 변경
이미 사용 중인 축 그룹 변수명을 변경할 경우 다음 2가지 조작을 실행해 주십시오.
- 변수 선언 내의 변수 테이블에 있는 축 그룹 변수명 변경.
- 사용자 프로그램의 축 그룹 변수명 변경.

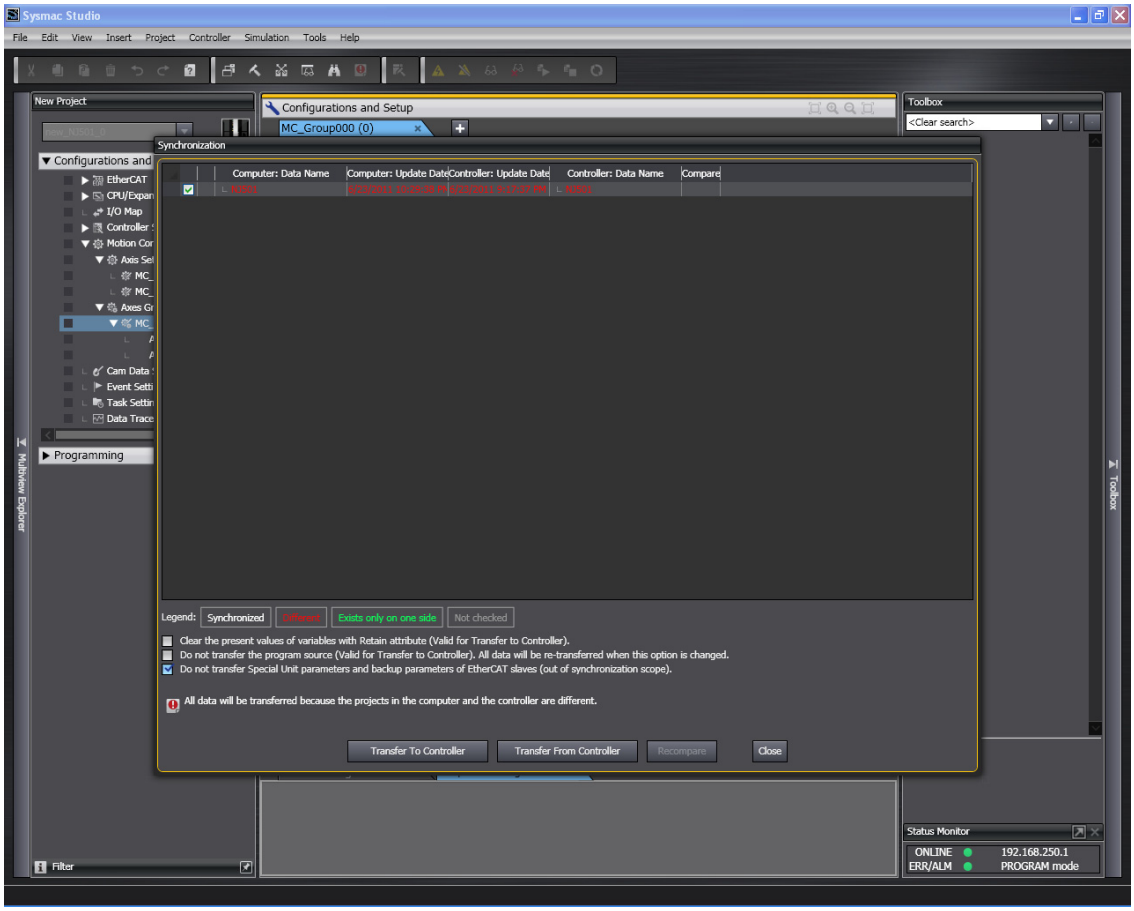
변수 테이블의 축 그룹 변수명을 변경해도 사용자 프로그램의 축 그룹 변수명에는 반영되지 않습니다.
변수 테이블에서 선언되지 않은 변수명을 사용자 프로그램에서 사용하면 이상이 발생합니다.
반드시 둘 다 변경해 주십시오.

CPU 유니트로 다운로드하기

CPU 유니트에 다운로드하는 경우는 Sysmac Studio의「Synchronization」기능을 사용합니다.

1 [Controller] 메뉴의 [Online]을 클릭하여 온라인 상태로 설정합니다.

2 [Controller] 메뉴에서 [Synchronization]을 선택하고 [Transfer To Controller]을 클릭합니다.



4

Sysmac Studio를 이용한 배선 확인

Sysmac Studio의 MC 시운전 기능에 대해 설명합니다. MC 시운전 기능을 이용하여 Programless로 센서 신호의 모니터 및 모터 배선을 확인할 수 있습니다.

4-1 Sysmac Studio의 기능 4-2
4-1-1 MC 시운전 기능 일람 4-2
4-1-2 사용 순서 4-3
4-1-3 축 파라메터 설정 예 4-4
4-1-4 MC 시운전 기능의 기동 순서 4-5
4-2 센서 신호 모니터 4-7
4-3 모터 동작 확인 4-8
4-3-1 서보 ON 4-8
4-3-2 조그 전송 4-9
4-3-3 원점 복귀 4-9
4-3-4 절대값 위치 결정 4-10
4-3-5 상대값 위치 결정 4-11

4-1 Sysmac Studio의 기능

여기에서는 MC 시운전 기능을 사용하여 배선 및 기본적인 설정을 확인하는 방법에 대해 설명합니다.
Sysmac Studio의 MC 시운전 기능을 이용하면 Programless로 배선을 확인할 수 있습니다.

4-1-1 MC 시운전 기능 일람

MC 시운전 기능에는 아래의 기능이 있습니다.

분류	기능	설명	설정 항목/모니터 항목
축 동작	감속 정지	MC 시운전 중에 감속 정지합니다.	—
	서보 ON/OFF	서보 ON/OFF합니다.	—
	이상 해제	MC 기능 모듈에서 발생한 이상을 해제합니다.	—
	조그 전송	정방향 또는 역방향으로 조그를 전송합니다.	목표 속도 가속도/감속도
	절대값 위치 결정	절대값 위치 결정을 실행합니다. *1	목표 위치 목표 속도 가속도/감속도 저크
	상대값 위치 결정	상대값 위치 결정을 실행합니다.	이동 거리 목표 속도 가속도/감속도 저크
	원점 복귀	원점 복귀 파라메터의 설정으로 원점 복귀를 실행합니다.	원점 복귀 파라메터
모니터	이상 일람	MC 기능 모듈에서 발생하는 이상을 모니터합니다.	MC에 공통된 이상 축 이상 축 그룹 이상
	축 상태	축 상태를 모니터합니다.	축 기동 준비 완료 정지 중 위치 결정 동작 중 연속 동작 중 원점 복귀 중 감속 정지 중 원점 확정 원점 정지
	현재 위치 모니터	현재 위치를 모니터합니다.	현재 위치(지령/피드백)
	현재 속도 모니터	현재 속도를 모니터합니다.	현재 속도(지령/피드백)
	서보 드라이버 상태	서보 드라이버의 상태를 모니터합니다.	서보 ON/OFF 서보 준비 주회로 전원
	입력 신호	입력 신호의 상태를 모니터합니다.	정방향 한계 입력/역방향 한계 입력 즉시 정지 입력 원점 근방 입력 원점 입력 외부 래치 입력 1~2

*1. 축의 카운트 모드가 로터리 모드일 때는 정방향 목표 위치를 향해 위치 결정을 실행합니다.
상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_MoveAbsolute(절대값 위치 결정) 명령을 참조해 주십시오.

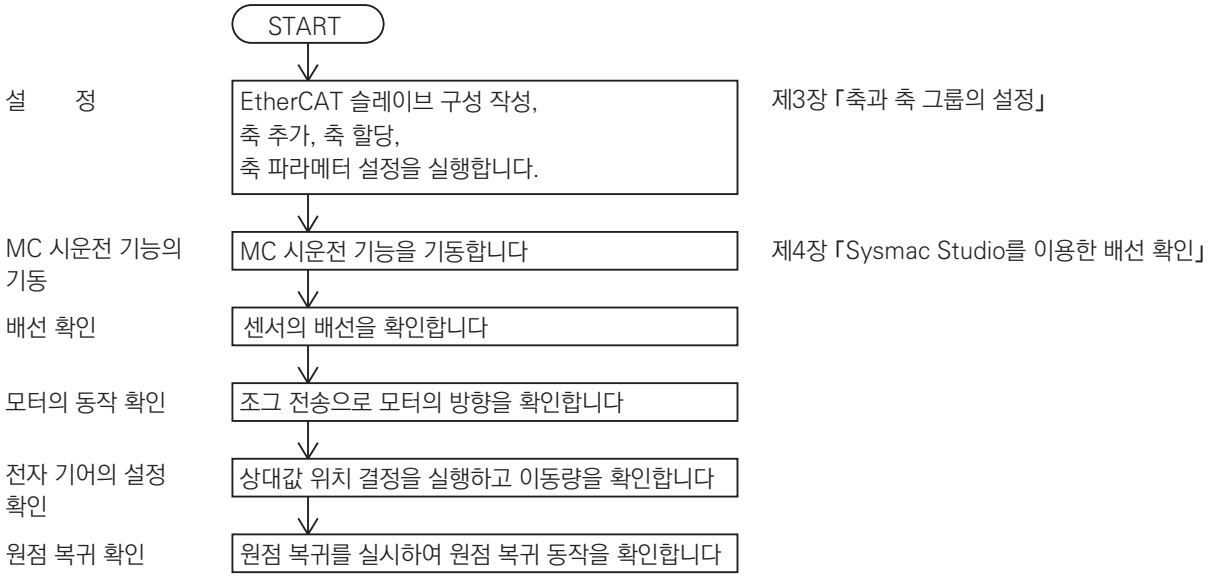
(주) MC 시운전 기능은 오므론 서보 드라이버 1S 시리즈 및 G5 시리즈, NX 시리즈 펄스 출력 유니트를 대상으로 합니다. 타사 드라이버에서는 사용하지 마십시오.

4-1-2 사용 순서

MC 시운전을 실행하려면 우선 아래 2가지 사항을 확인해 주십시오.

- Sysmac Studio와 컨트롤러 본체가 온라인으로 연결되어 있습니까?
- 다른 Sysmac Studio에서 MC 시운전을 기동하지 않았습니까?

확인이 끝나면 아래의 순서에 따라 조작해 주십시오.



사용상의 주의

- Sysmac Studio의 지령에 따라 서보 ON, 조그 전송, 상대값 위치 결정, 절대값 위치 결정, 원점 복귀를 실행하면 설정된 속도로 모터가 동작합니다.
  모터 동작으로 인한 위험이 없는지 충분히 확인한 후 실행해 주십시오.
- Sysmac Studio에서 조작할 경우에는 안전하게 모터를 정지할 수 있도록 외부에 비상 정지 회로를 설치해 주십시오. PC에 이상이 발생하면 Sysmac Studio에서 지령이 발행되지 않는 경우가 있습니다.
- Sysmac Studio에서 조작할 때는 EtherCAT 통신을 설정하고 통신을 확립한 후 실행해 주십시오.
- NX 시리즈 위치 인터페이스 유니트를 사용하는 경우의 순서는 📖『NX 시리즈 위치 인터페이스 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

참고

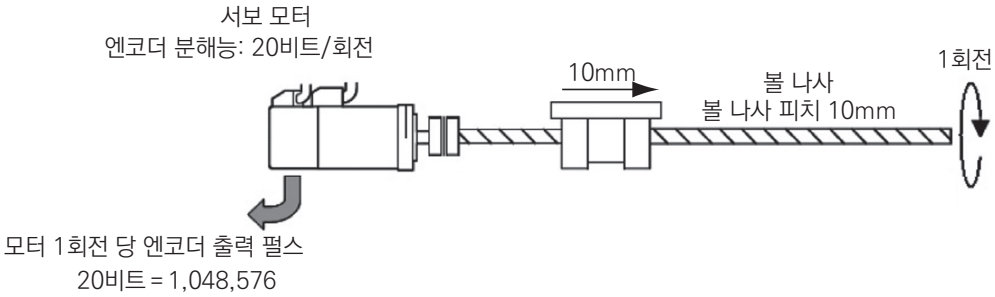
다음 조작을 하면 언제라도 MC 시운전 기능을 종료할 수 있습니다.

- Sysmac Studio의 [Controller] 메뉴에 있는 [MC Test Run]에서 [Stop]을 선택한다.
- Sysmac Studio의 멀티 뷰 익스플로러의 대상 축에서 우클릭하고 [Stop MC Test Run]을 선택한다.
- Sysmac Studio의 MC 시운전 화면을 닫는다.
- Sysmac Studio를 종료한다.

조작 방법에 대한 상세한 내용은 📖『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.

4-1-3 축 파라메터 설정 예

Sysmac Studio의 MC 시운전 기능을 실행할 경우 다음 축 파라메터를 설정합니다.
1축 장비를 예로 들어 설정 예를 설명합니다.



설정 항목	설정 내용
축 변수명	Axis1 *1
축 번호	1 *2
축 사용	사용 축
축 종별	서보 축
입력 디바이스/출력 디바이스	1 *3
표시 단위	μm
모터 1회전의 펄스 수	1,048,576 *4
모터 1회전의 이동량	10,000 *4
최고 속도	500,000 *5
JOG 최고 속도	50,000 *6
최대 가속도	5,000,000 *7
최대 감속도	5,000,000 *7
소프트웨어 리미트 기능	지령 위치에 대해 유효하면 즉시 정지시킨다.
정방향 소프트웨어 리미트	500,000 *8
역방향 소프트웨어 리미트	0 *8
카운트 모드	리니어 모드

*1. 축이 여러 개인 경우는 축마다 다른 변수명을 설정합니다.
*2. 축이 여러 개인 경우는 축마다 다른 값을 설정합니다.
*3. 서보 드라이버의 노드 주소와 같은 값을 설정합니다. 축이 여러 개인 경우는 축마다 다른 값을 설정합니다.
*4. 위치의 지령 단위는 1(μm)입니다.
*5. 최고 속도는 3,000r/min = 30m/min = 0.5m/s = 500,000μm/s로 합니다.
*6. 조그 최고 속도는 최고 속도의 10%인 0.05m/s = 50,000μm/s로 합니다.
*7. 최대 가속도 및 최대 감속도는 5m/s^{2}입니다. 최고 속도 3,000r/min까지의 가속 시간을 0.1s로 합니다.
*8. 장비 가동 범위의 안쪽 위치를 설정합니다. 정방향 소프트웨어 리미트 설정값은 50cm = 500,000μm로 합니다.

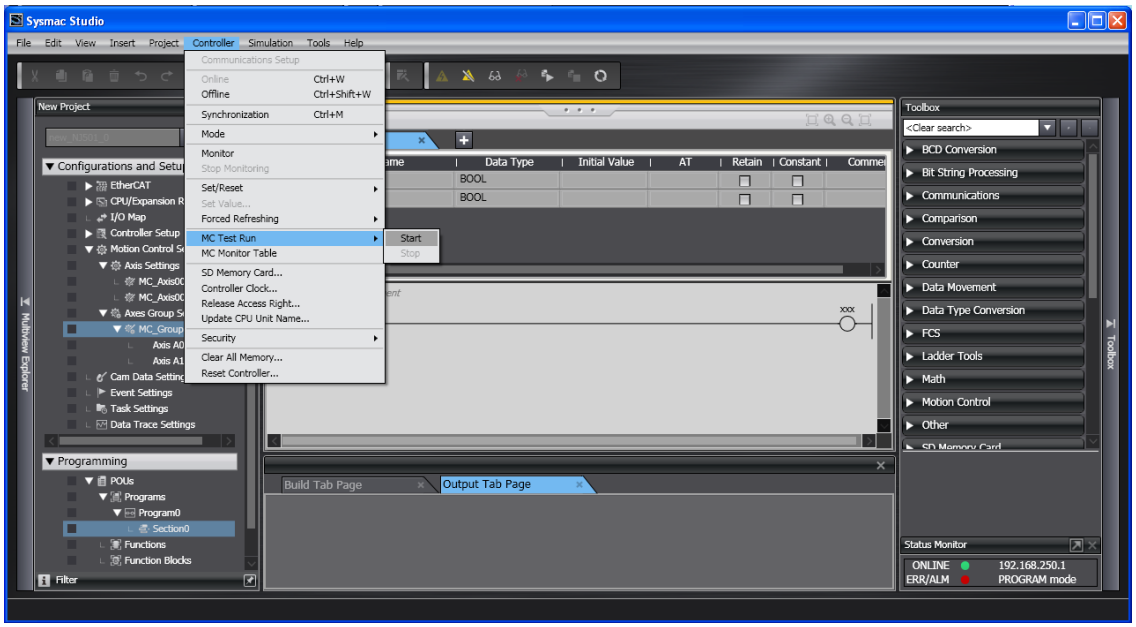
4-1-4 MC 시운전 기능의 기동 순서

MC 시운전 기능은 Sysmac Studio에서 기동합니다.

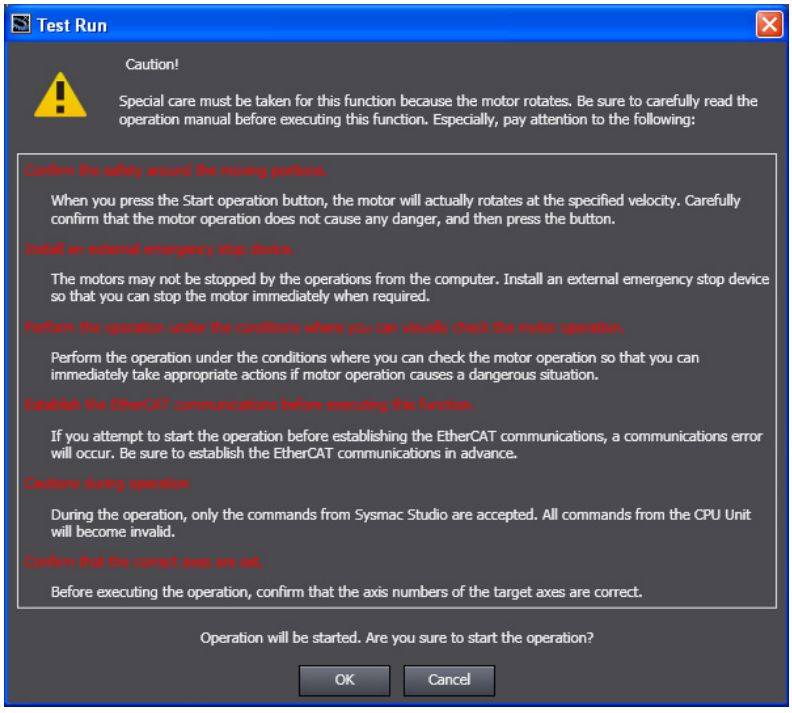
1 Sysmac Studio를 기동하여 서보 축 설정이 완료된 기존 프로젝트를 엽니다.

2 [Controller] 메뉴의 [Online]을 클릭하여 온라인 상태로 설정합니다.

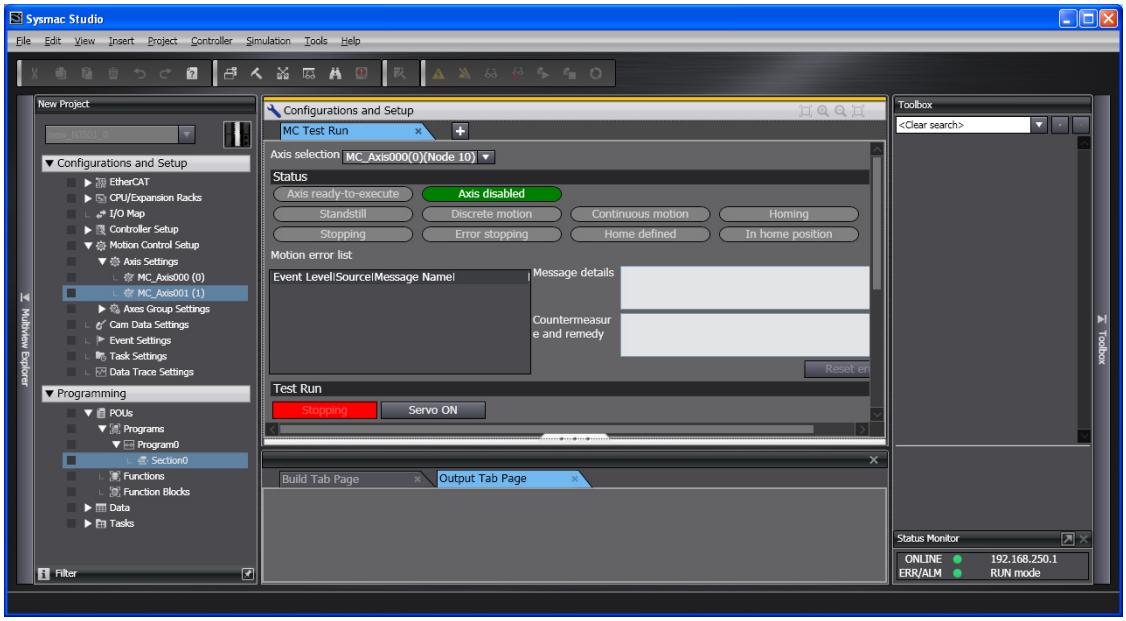
3 [Controller] 메뉴의[MC Test Run]에서 [Start]를 클릭합니다.



아래의 임의 대화 상자가 표시되므로 내용을 잘 읽고 안전한지 확인한 후 [OK]를 클릭해 주십시오.



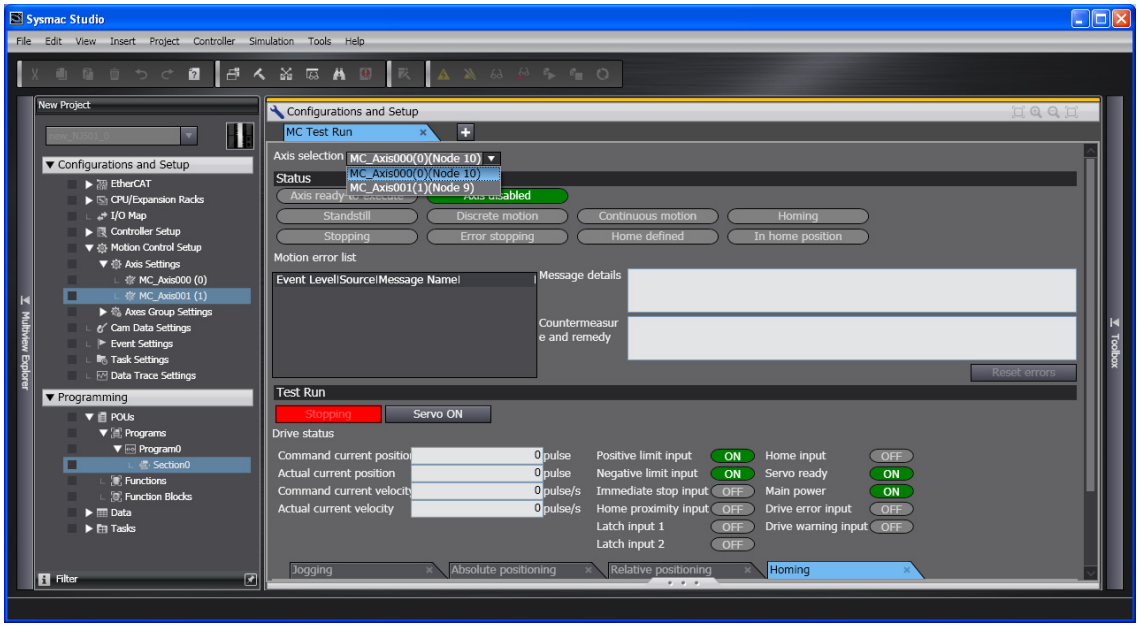
MC 시운전 화면이 표시됩니다.



4-2 센서 신호 모니터

센서 신호의 배선 확인은 각 입력 신호의 표시를 사용합니다.

1 MC 시운전 화면에서 확인할 축을 선택합니다.



2 각 입력 신호에 접속한 센서를 ON/OFF하여 모니터 화면의 해당 신호가 올바르게 ON/OFF 하는지 확인합니다.

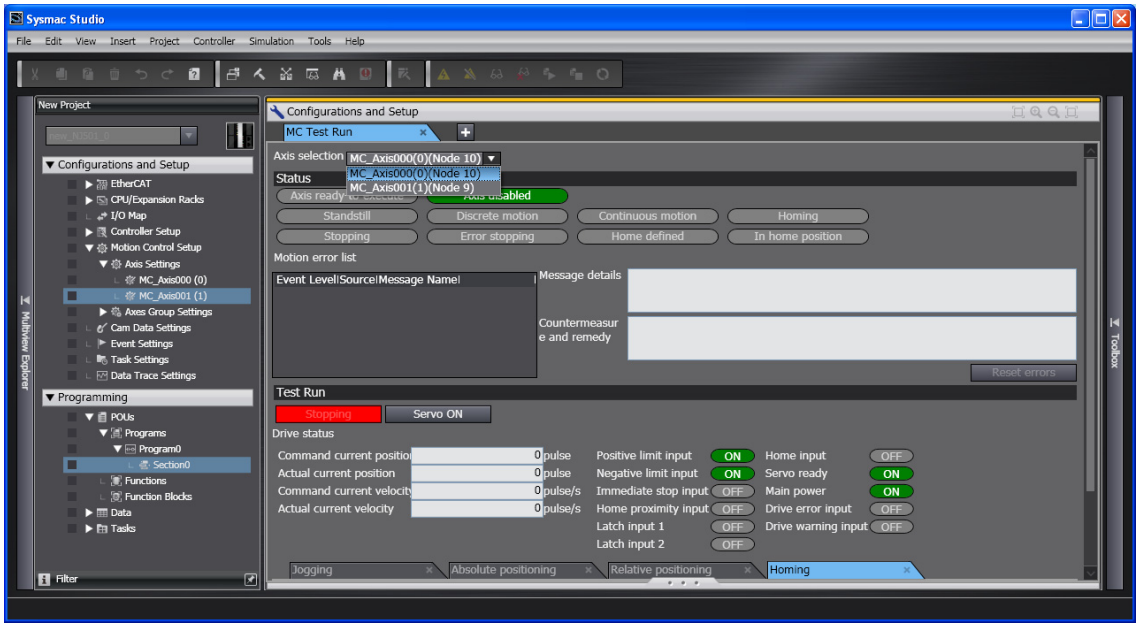
4-3 모터 동작 확인

모터 동작을 확인하려면 MC 시운전의 각 기능을 사용합니다.

4-3-1 서보 ON

[Servo ON] 버튼을 누르면 서보 ON과 서보 OFF를 변환할 수 있습니다.

1 서보 ON할 축을 선택합니다.



2 [Servo ON] 버튼을 누르고 서보 ON 상태로 만듭니다.

3 [Servo OFF] 버튼을 누르면 서보 OFF 상태로 되돌아갑니다.

사용상의 주의

- Sysmac Studio의 지령에 따라 서보 ON, 조그 전송, 상대값 위치 결정, 절대값 위치 결정, 원점 복귀를 실행하면 설정된 속도로 모터가 동작합니다.
  모터 동작으로 인한 위험이 없는지 충분히 확인한 후 실행해 주십시오.
- Sysmac Studio에서 조작할 경우에는 안전하게 모터를 정지할 수 있도록 외부에 비상 정지 회로를 설치해 주십시오. PC에 이상이 발생하면 Sysmac Studio에서 지령이 발행되지 않는 경우가 있습니다.
- Sysmac Studio에서 조작할 때는 EtherCAT 통신을 설정하고 통신을 확립한 후 실행해 주십시오.
- NX 시리즈 펄스 출력 유니트를 사용할 때는 모터 드라이버의 통전 ON/OFF를 별도로 실행해야 합니다. 상세한 내용은 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

4-3-2 조그 전송

1 MC 시운전 화면의 [Jogging] 탭에서 조그 전송할 축을 선택합니다.

2 [Servo ON] 버튼을 누르고 서보 ON 상태로 만듭니다.

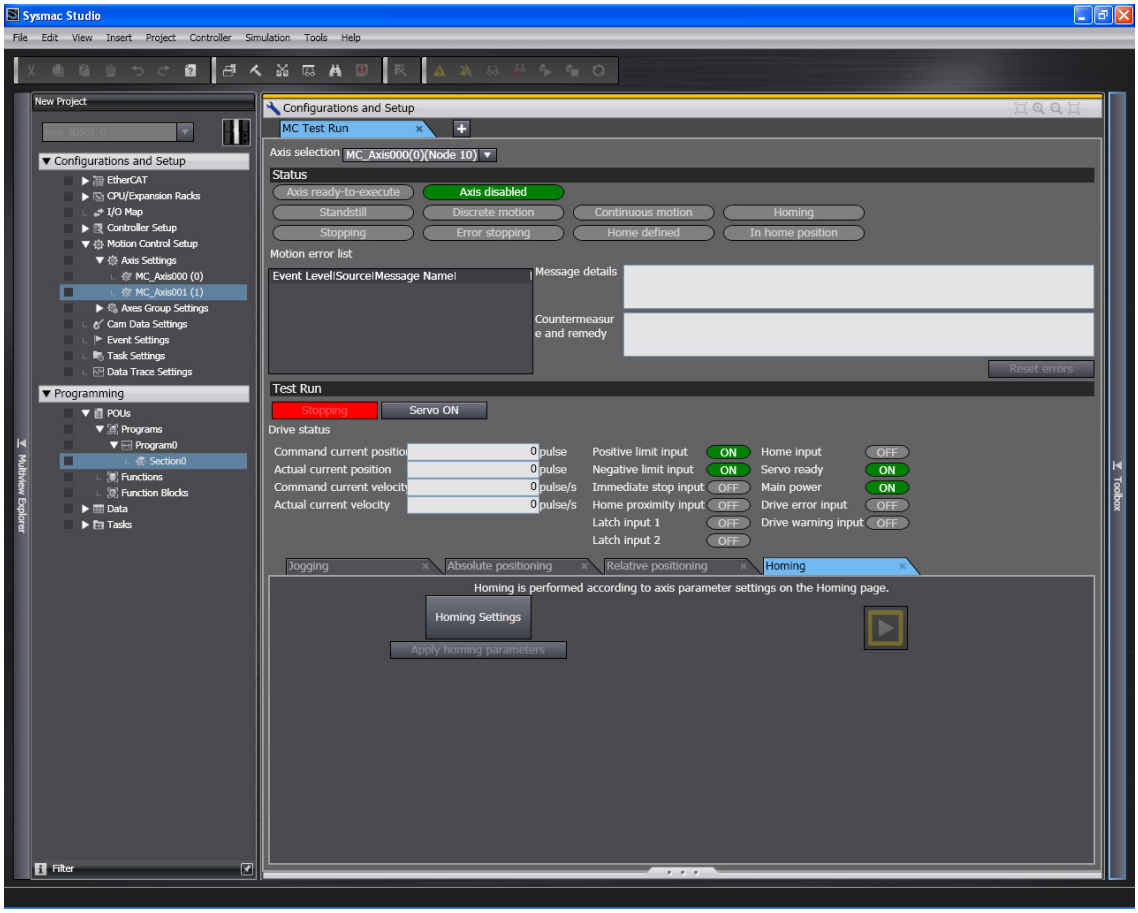
3 목표 속도, 가속도, 감속도 항목에 값을 입력하고 [Apply] 버튼을 누릅니다.

4  버튼 혹은  버튼을 누릅니다.
버튼을 누르고 있는 동안 정방향 혹은 역방향으로 작동합니다.
설정한 방향으로 움직이는지 확인해 주십시오.

4-3-3 원점 복귀

1 미리 축 파라메터 설정 화면의 [Homing Settings]에서 원점 복귀 파라메터를 설정해 둡니다.

2 시운전 화면의 [Homing] 탭을 선택합니다.
아래와 같은 화면이 표시됩니다



3 원점 복귀를 실행할 축을 선택합니다.

4 [Servo ON] 버튼을 누르고 서보 ON 상태로 만듭니다.

5 [Apply homing parameters] 버튼을 누릅니다.

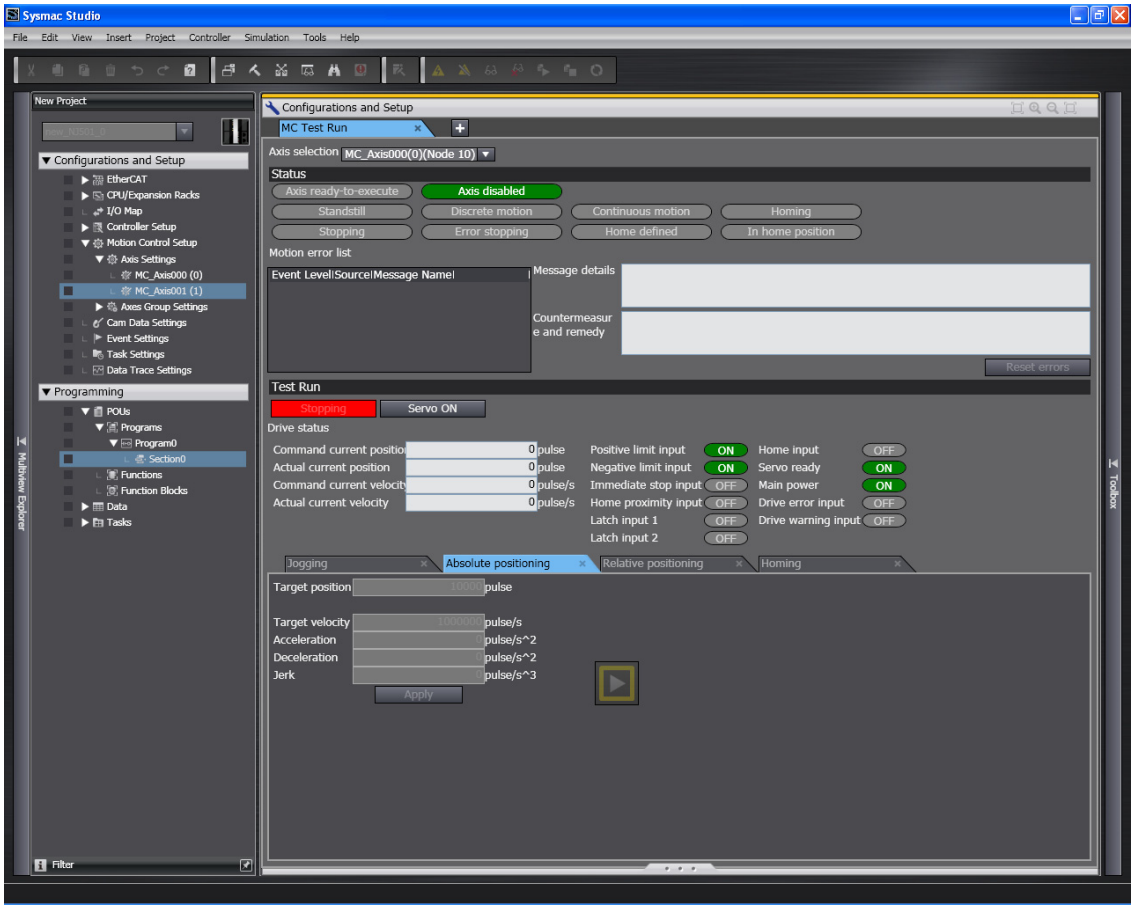
6  버튼을 누릅니다.
원점 복귀 동작이 설정한 대로 움직이는지 확인해 주십시오.

참고

- [Homing Settings] 버튼을 누르면 [Axis Parameter Settings]의 [Homing Settings] 화면이 표시되므로 원점 복귀 파라메터를 설정해 주십시오.
- 미리 원점 복귀 파라메터를 설정해 두면 [Apply homing parameters] 버튼을 눌렀을 때 해당 설정이 적용됩니다.

4-3-4 절대값 위치 결정

1 MC 시운전 화면에서 [Absolute positioning] 탭을 선택합니다.
아래와 같은 화면이 표시됩니다



2 절대값 위치 결정을 실행할 축을 선택합니다.

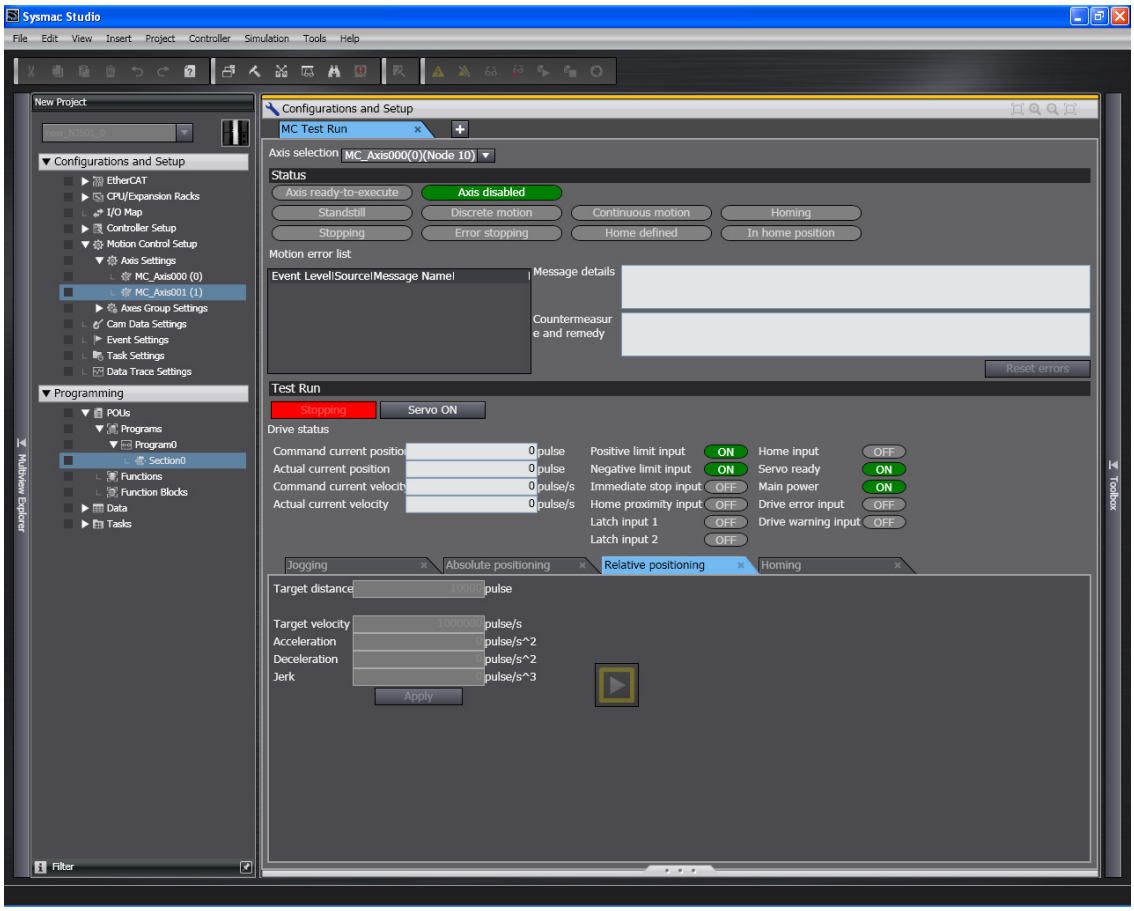
3 [Servo ON] 버튼을 누르고 서보 ON 상태로 만듭니다.

4 목표 위치, 목표 속도, 가속도, 감속도, 저크 항목에 값을 입력하고 [Apply] 버튼을 누릅니다.

5 버튼을 누르면 절대값 위치 결정을 시작합니다.
위치 결정이 완료된 위치가 설정한 위치인지 확인해 주십시오.

4-3-5 상대값 위치 결정

1 MC 시운전 화면에서 [Relative positioning] 탭을 선택합니다.
아래와 같은 화면이 표시됩니다



2 상대값 위치 결정을 실행할 축을 선택합니다.

3 [Servo ON] 버튼을 누르고 서보 ON 상태로 만듭니다.

4 목표 이동 거리, 목표 속도, 가속도, 감속도, 저크 항목에 값을 입력하고 [Apply] 버튼을 누릅니다.

5 ▶ 버튼을 누르면 상대값 위치 결정을 시작합니다.
이동 거리가 설정한 대로 되어 있는지 확인해 주십시오.

5

모션 제어 파라메터

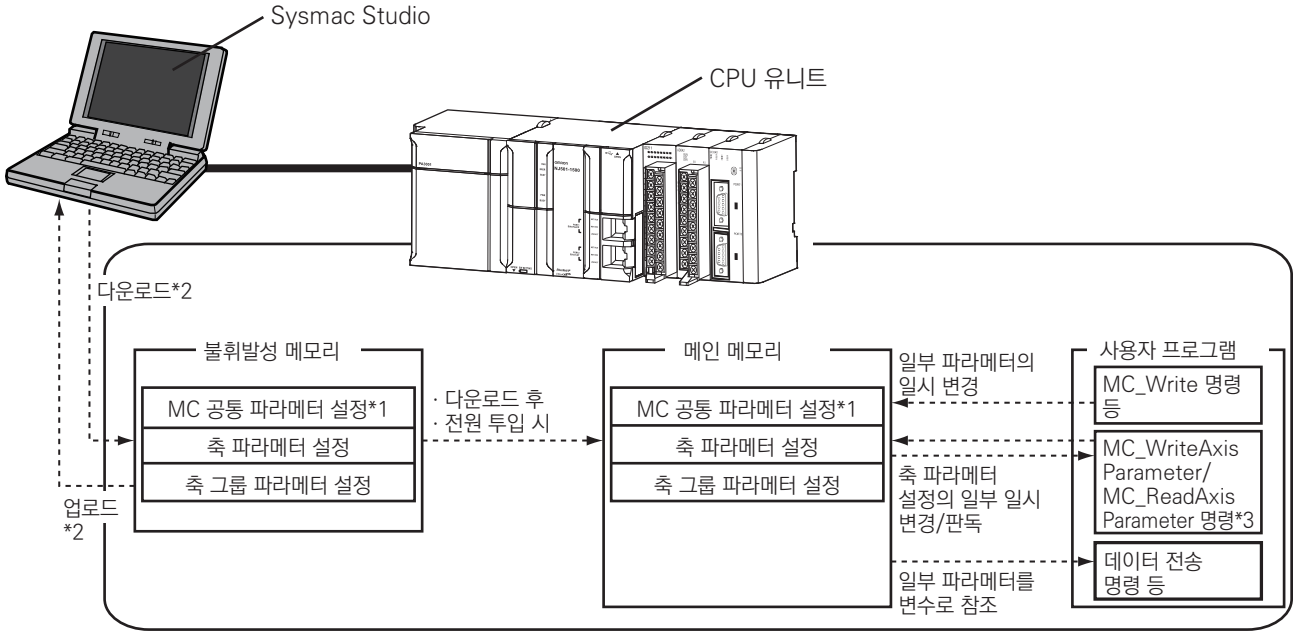
이 장에서는 모션 제어에서 사용하는 축 파라메터와 축 그룹 파라메터에 대해 설명합니다.

5-1 개요 5-2
5-2 축 파라메터 5-4
5-2-1 축 파라메터 일람 5-4
5-2-2 축 기본 설정 5-6
5-2-3 단위 변환 설정 5-11
5-2-4 동작 설정 5-18
5-2-5 확장 동작 설정 5-21
5-2-6 리미트 설정 5-22
5-2-7 위치 카운트 설정 5-23
5-2-8 서보 드라이버 설정 5-25
5-2-9 원점 복귀 설정 5-26
5-2-10 축 파라메터 설정 예 5-28
5-3 축 그룹 파라메터 5-30
5-3-1 축 그룹 파라메터의 일람 5-30
5-3-2 축 그룹 기본 설정 5-31
5-3-3 축 그룹 동작 설정 5-33
5-3-4 축 그룹의 유효 5-34

5-1 개요

NJ/NX 시리즈 CPU 유니트의 MC 기능 모듈에서는 모션 제어 명령으로 단축에서의 동작과 축 그룹에서의 다축 동작을 실행할 수 있습니다.
축 파라메터와 축 그룹 파라메터는 이러한 동작을 설정하기 위한 파라메터입니다.
축 파라메터는 항상 설정해야 하나 축 그룹에서의 다축 동작을 사용하지 않는 경우에는 축 그룹 파라메터를 설정할 필요가 없습니다.

이러한 파라메터를 모션 제어 파라메터 설정(약칭: MC 파라메터 설정)이라 부릅니다.



*1 현 버전의 MC 기능 모듈에서는「MC 공통 파라메터 설정」의 설정 항목은 없습니다.
*2 「다운로드」 및「업로드」는 Sysmac Studio의「Synchronization」 기능을 사용합니다.
*3 MC_WriteAxisParameter(축 파라메터 기록) 명령과 MC_ReadAxisParameter(축 파라메터 판독) 명령은 유니트 버전 Ver.1.08 이상인 CPU 유니트와 Sysmac Studio Ver.1.09 이상의 조합으로 사용할 수 있습니다.

MC 파라메터 설정 데이터 플로우

- Sysmac Studio에서 설정한 MC 파라메터 설정을 CPU 유니트에 다운로드하면 CPU 유니트에 내장된 불휘발성 메모리에 저장됩니다. 또한, Sysmac Studio에 MC 파라메터 설정을 업로드하면 불휘발성 메모리에 저장된 설정이 업로드됩니다.
- 불휘발성 메모리에 저장된 설정은 다운로드 후나 전원 투입 시에 메인 메모리에 반영됩니다.
- 설정 내용이 올바르면 MC 기능 모듈은 메인 메모리의 설정 내용으로 제어를 실행합니다.
- 일부 파라메터는 모션 제어 시스템 변수로 설정 내용을 참조할 수 있습니다.
- MC 파라메터 설정의 업로드 및 다운로드는 CPU 유니트의 모드나 MC 기능 모듈의 상태와 관계없이 실행할 수 있습니다.
- 다운로드를 시작하면 동작 중인 축은 즉시 정지합니다.

버전 관련 정보

유니트 버전 Ver.1.13 이상인 CPU 유니트와 Sysmac Studio Ver.1.17 이상을 조합하면 다운로드 중에도 I/O 디바이스로 지령을 계속 보낼 수 있습니다.
유니트 버전 Ver.1.12 이하인 CPU 유니트의 경우는 다운로드 중에 I/O 디바이스로 보내는 지령을 정지합니다.
다운로드 시작 시에 I/O 디바이스로 보내는 지령을 정지할지 계속 보낼지에 대한 설정은『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

I/O 디바이스로 보내는 지령을 정지할 경우

다운로드 중에는 서보 OFF가 되고 축 상태는「Disable(축 무효)」이 됩니다.

I/O 디바이스로 지령을 계속 보낼 경우

다운로드 중에는 다운로드 직전의 MC_Power 명령을 이용한 서보 ON 상태 및 MC_SetTorqueLimit 명령을 이용한 출력 토크 제한을 계속합니다.
또한, 다운로드를 통해 사용자 프로그램이 갱신되어 MC_Power 명령 및 MC_SetTorqueLimit 명령이 삭제된 경우에도 서보 ON 상태 및 토크 제한을 계속합니다.

MC 파라메터 설정 명령에 의한 갱신

- 모션 제어 명령의 MC_Write(MC 설정 기록) 명령, MC_ChangeAxesInGroup(축 그룹 구성 축 기록) 명령 및 MC_WriteAxisParameter(축 파라메터 기록) 명령 등은 메인 메모리에 있는 일부 MC 파라메터 설정의 내용을 사용자 프로그램 실행 중에 변경할 수 있습니다.
- 지정된 설정값이 범위 외라면 명령의 출력 변수「Error(에러)」가 TRUE가 되고 MC 파라메터 설정은 변경되지 않습니다.
- 변경한 MC 파라메터 설정 내용이 유효해지는 타이밍은 아래 중 하나일 때입니다.
  a) 축이나 축 그룹이 정지된 상태에서 축 지령 또는 축 그룹 지령의 명령을 기동했을 때.
  b) 명령의 버퍼 모드 선택을 Aborting(중단)으로 하고 다중 기동했을 때.
- MC_Write(MC 설정 기록) 명령, MC_ChangeAxesInGroup(축 그룹 구성 축 기록) 명령 및 MC_WriteAxisParameter(축 파라메터 기록) 명령 등에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

사용상의 주의

- MC 파라메터 설정의 MC_Write(MC 설정 기록) 명령 등을 이용한 변경은 CPU 유니트 내 메인 메모리에 저장됩니다. CPU 유니트에 내장된 불휘발성 메모리에는 저장되지 않습니다. 따라서 전원을 재투입하거나 Sysmac Studio에서 다운로드하여 불휘발성 메모리의 파라메터 설정 내용으로 되돌아갑니다. 또한, Sysmac Studio에서 이러한 데이터를 업로드할 수 없습니다.
  불휘발성 메모리에 저장해야 하는 경우에는 Sysmac Studio에서 파라메터 설정을 변경하고 CPU 유니트로 다운로드 해 주십시오.
- 다운로드 후에도 MC_Power(운전 가능) 명령 및 MC_SetTorqueLimit(토크 제한) 명령을 유지할 경우 명령 입력 접점의 유지 속성을 유지해 주십시오.
- MC 파라메터 설정을 변경할 수 있는 명령은 아래와 같습니다.
  MC_Write(MC 설정 기록) 명령
  MC_ChangeAxesInGroup(축 그룹 구성 축 기록) 명령
  MC_ChangeAxisUse(축 사용 변경) 명령
  MC_WriteAxisParameter(축 파라메터 기록) 명령
- NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

5-2 축 파라메터

축 파라메터는 MC 기능 모듈로 제어하는 축에 관하여 최고 속도나 저크, 원점 복귀와 같은 축 동작에 관한 설정을 합니다.
축 파라메터는 각 형식의 제어 축 최대 수만큼 있습니다. 제어 축 최대 수는 형식에 따라 다릅니다.
상세한 내용은 「1-4-2 성능 사양 (P.1-7)」을 참조해 주십시오.
또한, 각 축의 파라메터 설정은 동일하므로 여기에서는 1축에 해당하는 부분에 대해서만 설명합니다.

5-2-1 축 파라메터 일람

축 파라메터는 Sysmac Studio를 사용하여 축별로 설정합니다.

구분	파라메터의 명칭	일시 변경 *1		변수 판독 *2	참조 페이지
		가능 여부	사용 명령		
축 기본 설정	축 번호			가능	P. 5-6
	모션 제어 *3			가능 *4	
	축 사용	가능 *5	MC_ChangeAxisUse	가능	
	축 종별			가능	
	제어 기능 *6				
	입력 디바이스/출력 디바이스			가능	
단위 변환 설정	표시 단위	가능 *7	MC_WriteAxisParameter *8	가능	P. 5-11
	모터 1회전의 펄스 수			가능	
	모터 1회전의 이동량			가능	
	감속기 사용 *9				
	워크 측 1회전의 이동량 *9				
	워크 측 기어비 *9				
	모터 측 기어비 *9				
동작 설정	최고 속도	가능 *7	MC_WriteAxisParameter *8		P. 5-18
	기동 속도 *10				
	조그 최고 속도				
	최대 가속도				
	최대 감속도				
	가감속 초과				
	반전 시 동작				
	속도 경고값	가능	MC_Write MC_WriteAxisParameter *8		
	가속도 경고값				
	감속도 경고값				
	정방향 토크 경고값 *11				
	역방향 토크 경고값 *11				
	인포지션 폭	가능 *12			
	인포지션 체크 시간	가능			
	피드백 속도 필터 시정수	가능 *7	MC_WriteAxisParameter *8		
	원점 위치 범위				
확장 동작 설정	즉시 정지 입력 정지 방법	가능 *7	MC_WriteAxisParameter *8		P. 5-21
	한계 입력 정지 방법				
	드라이버 에러 리셋 감시 시간				
	정방향 토크 제한 상한값				
	역방향 토크 제한 상한값				
	즉시 정지 입력 논리 반전 *10				
	정방향 한계 입력 논리 반전 *10				

구분	파라메터의 명칭	일시 변경 *1		변수 판독 *2	참조 페이지
		가능 여부	사용 명령		
확장 동작 설정	역방향 한계 입력 논리 반전 *10	가능 *7	MC_WriteAxisParameter *8		P. 5-21
	원점 근방 입력 논리 반전 *10				
리미트 설정	소프트웨어 리미트 기능	가능	MC_Write MC_WriteAxisParameter *8		P. 5-22
	정방향 소프트웨어 리미트				
	역방향 소프트웨어 리미트				
	위치 편차 초과값				
	위치 편차 경고값				
위치 카운트 설정	카운트 모드	가능 *7	MC_WriteAxisParameter *8		P. 5-23
	링 카운터 상한 설정값			가능 *4	
	링 카운터 하한 설정값			가능 *4	
	엔코더 종별				
서보 드라이버 설정	링 카운터 상한 설정값				P. 5-25
	링 카운터 하한 설정값				
	PDS 상태 제어 방식 *4				
원점 복귀 설정	원점 복귀 동작	가능 *7	MC_WriteAxisParameter *8		P. 5-26
	원점 입력 신호				
	원점 복귀 시작 방향				
	원점 검출 방향				
	정방향 한계 입력 시 동작				
	역방향 한계 입력 시 동작				
	원점 복귀 속도				
	원점 복귀 접근 속도				
	원점 복귀 가속도				
	원점 복귀 감속도				
	원점 복귀 저크				
	원점 입력 마스크 거리				
	원점 위치 오프셋				
	원점 복귀 보류 시간				
	원점 복귀 보정값				
	원점 복귀 보정 속도				

*1. 명령을 이용한 일시 변경의 가능 여부를 나타냅니다.
*2. 모션 제어 시스템 변수로 사용자 프로그램에서의 판독 가능 여부를 나타냅니다.
*3. NX701 CPU 유니트의 경우에 설정합니다.
*4. 유니트 버전 Ver.1.10 이상인 CPU 유니트와 Sysmac Studio Ver.1.12 이상의 조합으로 사용할 수 있습니다.
*5. 유니트 버전 Ver.1.04 이상인 CPU 유니트와 Sysmac Studio Ver.1.05 이상의 조합으로 일시 변경할 수 있습니다.
*6. NX102 및 NX1P2 CPU 유니트의 경우에 설정합니다.
*7. 유니트 버전 Ver.1.08 이상인 CPU 유니트와 Sysmac Studio Ver.1.09 이상의 조합으로 일시 변경할 수 있습니다.
*8. MC_WriteAxisParameter 명령은 유니트 버전 Ver.1.08 이상인 CPU 유니트와 Sysmac Studio Ver.1.09 이상의 조합으로 사용할 수 있습니다.
MC_WriteAxisParameter 명령으로 일시 변경할 수 있는 파라메터는 MC_ReadAxisParameter 명령으로 판독할 수 있습니다.
*9. 유니트 버전 Ver.1.11 이상인 CPU 유니트와 Sysmac Studio Ver.1.15 이상의 조합으로 사용할 수 있습니다.
*10.유니트 버전 Ver.1.05 이상인 CPU 유니트와 Sysmac Studio Ver.1.06 이상의 조합으로 사용할 수 있습니다.
*11.토크 제어 시에만 유효합니다.
*12.MC_Write 명령에서의 일시 변경은 유니트 버전 Ver.1.01 이상인 CPU 유니트와 Sysmac Studio Ver.1.02 이상의 조합으로 가능합니다.
MC_WriteAxisParameter 명령을 이용한 일시 변경은 유니트 버전 Ver.1.08 이상인 CPU 유니트와 Sysmac Studio Ver.1.09 이상의 조합으로 가능합니다.

축 파라메터의 설정 방법에 대해서는 「3-2 축 설정 순서 (P.3-9)」를 참조해 주십시오.
MC_Write(MC 설정 기록) 명령 등 각 명령에 대한 상세한 내용은 『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.
모션 제어 시스템 변수에 대해서는 「6-6 모션 제어 시스템 변수 (P.6-17)」를 참조해 주십시오.

5-2-2 축 기본 설정

축의 사용 여부, 사용하는 경우의 축 종별 및 EtherCAT 슬레이브 디바이스의 노드 주소를 설정합니다.

파라메터의 명칭	기능	설정 범위	초기값
축 번호	축의 논리 번호를 설정합니다. 시스템 정의 변수의 변수명 「_MC_AX[0-255]」, 「_MC1_AX[0-255]」, 「_MC2_AX[0-255]」의 번호는 여기에서 설정한 번호가 됩니다.	—	—
모션 제어 *1	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5) 중 1가지 태스크에 할당합니다. 1: 프라이머리 정주기 태스크 2: 정주기 태스크(실행 우선도 5)	1~2	1
축 사용	축의 사용 여부를 설정합니다. *2 0: 미작성 축 1: 미사용 축 *3 2: 사용 축	0~2	0
축 종별	축 종별을 설정합니다. 가상 축에서는 입출력 배선이 필요 없습니다. 0: 서보 축 1: 엔코더 축 2: 가상 서보 축 3: 가상 엔코더 축	0~3	2
제어 기능 *4	제어할 축의 기능을 선택합니다. *5 0: 모두 1: 단축 위치 제어만	0, 1	0
입력 디바이스/ 출력 디바이스	축에 할당할 EtherCAT 슬레이브 디바이스의 노드 주소를 선택합니다. *6 축 종별이 가상 축인 경우에는 선택할 수 없습니다.	0~ 65535	—

*1. NX701 CPU 유니트의 경우에 설정합니다.
*2. 사용하지 않거나 작성하지 않은 축으로 모션 제어 명령을 실행하면 Busy(실행 중)가 TRUE가 됩니다. Execute(기동) 혹은 Enable(유효)을 FALSE로 하면 Busy(실행 중)도 FALSE가 됩니다. 「미사용 축」으로 설정하면 1개의 사용자 프로그램을 축 구성이 다른 장비에서 함께 사용할 때 사용하지 않는 축의 프로그램을 삭제할 필요가 없습니다.
*3. 유니트 버전 Ver.1.04 이상인 CPU 유니트와 Sysmac Studio Ver.1.05 이상의 조합인 경우 Sysmac Studio의 설정 화면에서는 [Unused axis (changeable to used axis)], [Unused axis (unchangeable to used axis)]로 표시됩니다. [Unused axis (changeable to used axis)]로 설정하면 축 파라메터를 설정할 수 있으며, MC_ChangeAxisUse(축 사용 변경) 명령을 이용하여 일시적으로 [Axis Use]를 변경할 수 있습니다.
축 사용 변경 기능에 대해서는 「9-8-10 축 사용 변경 기능 (P.9-79)」을 참조해 주십시오.
*4. NX102 및 NX1P2 CPU 유니트의 경우에 설정합니다.
*5. 모션 제어 축으로 사용할 경우에는 「All」을 선택합니다. 단축 위치 제어 축으로 사용할 경우에는 「Single-axis position control only」를 선택해 주십시오.
*6. NX 시리즈 위치 인터페이스 유니트는 EtherCAT 커플러 유니트의 노드 주소 및 위치 인터페이스 유니트의 NX 유니트 번호를 선택합니다.

사용상의 주의

절대값 엔코더를 사용할 경우
절대값 엔코더 원점 위치 오프셋이 축 번호에 맞춰 축별로 배터리 백업 메모리에 저장됩니다. 축 번호를 변경하면 저장된 오프셋량이 소실됩니다. 축 번호를 변경한 경우는 다시 원점 복귀를 실행해 주십시오.

축 번호

축 번호로 설정할 수 있는 번호는 최대 제어 축 수까지입니다. 또한, 사용 축으로 변환할 수 있는 실축 수는 최대 사용 실축 수까지입니다.

사양 항목	NX701-17□□	NX701-16□□
설정 가능한 축 번호	0~255	0~127
사용 실축 최대 수	256축	128축

사양 항목		NX102-12□□	NX102-11□□	NX102-10□□	NX102-90□□
설정 가능한 축 번호		0~14	0~14	0~14	0~3
사용 실축 최대 수		12축	8축	6축	4축
	사용 모션 제어 서보 축	8축	4축	2축	—
	사용 단축 위치 제어 서보 축	4축	4축	4축	4축

사양 항목		NX1P2-11□□□□	NX1P2-10□□□□	NX1P2-90□□□□
설정 가능한 축 번호		0~11	0~9	0~3
사용 실축 최대 수		8축	6축	4축
	사용 모션 제어 서보 축	4축	2축	—
	사용 단축 위치 제어 서보 축	4축	4축	4축

사양 항목	NJ501-□5□□	NJ501-□4□□	NJ501-□3□□
설정 가능한 축 번호	0~63	0~31	0~15
사용 실축 최대 수	64축	32축	16축

사양 항목	NJ301-12□□	NJ301-11□□	NJ101-10□□
설정 가능한 축 번호	0~14 *1	0~14 *2	0~5
사용 실축 최대 수	8축	4축	2축

*1. 유니트 버전 Ver.1.05 이하인 CPU 유니트 축 번호는「0~7」입니다.
*2. 유니트 버전 Ver.1.05 이하인 CPU 유니트는「0~3」입니다.

모션 제어

NX701 CPU 유니트의 경우 사용할 축을 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5) 중 1가지 태스크에 할당합니다.

참고

NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트에는 프라이머리 정주기 태스크만 있으므로「Motion control」설정은 없습니다.

축 종별

축 종별에서 선택할 수 있는 축 종별의 내용은 다음 표와 같습니다.

축 종별	내용
서보 축	EtherCAT 슬레이브 서보 드라이버나 NX 시리즈 위치 인터페이스 유니트*1를 사용하는 축입니다. 실제 서보 드라이버 등으로 할당하여 사용합니다. 1대의 서보 모터를 1축으로 사용합니다. NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 펄스 출력 유니트와 디지털 입력 유니트 등 복수의 디바이스를 1개의 축에 할당할 수 있습니다.
가상 서보 축	MC 기능 모듈 내의 가상 축입니다. 실제 서보 드라이버는 사용하지 않습니다. 동기 제어의 주축 등으로 사용합니다.
엔코더 축	EtherCAT 슬레이브 엔코더 입력 터미널 및 NX 시리즈 위치 인터페이스 유니트 *1를 사용하는 축입니다. 실제 엔코더 입력 터미널 등으로 할당하여 사용합니다. 1대의 엔코더 입력 터미널에 2개의 카운터가 있는 경우에는 각각 1개의 카운터를 1축으로 사용합니다.
가상 엔코더 축	엔코더 동작을 가상으로 실행하는 축입니다. 엔코더의 실물이 없는 경우 등 엔코더 축 대신에 일시적으로 사용합니다. *2

*1. 제어 대상 기기는 「1-4-3 기능 사양 (P.1-11)」을 참조해 주십시오.
*2. 가상 엔코더 축의 현재 위치를 갱신하는 모션 제어 명령과 조합하여 사용합니다. 이 명령에 대응하지 않는 버전인 MC 기능 모듈에서는 엔코더 축 대신 사용할 수 없습니다.

● 가상 서보 축

가상 서보 축은 물리적인 엔코더나 외부 입출력 신호가 없기 때문에 서보 축과 비교하여 아래와 같은 차이가 있습니다.

- 상시 서보 ON 상태로 취급됩니다.
- 피드백 현재 위치 = 지령 현재 위치가 됩니다. *1
- 피드백 현재 속도 = 지령 현재 속도가 됩니다. *1
- 외부 입력 신호는 사용할 수 없습니다.
- MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령으로 원점 복귀할 경우 축 파라메터의 원점 복귀 동작 설정과 관계없이 원점 프리셋으로 취급됩니다.
- 래치 기능을 사용하는 모션 제어 명령에서는 트리거 입력 조건의 모드를 컨트롤러 모드로 설정해야 합니다. 드라이버 모드로 설정해도 이상은 없으나 래치의 트리거가 발생하지 않으므로 명령 실행은 완료되지 않습니다.
  래치 기능을 사용하는 명령에서는 MC_TouchProbe(외부 래치 유효) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령 및 MC_MoveLink(사다리꼴 패턴 캠) 명령 등이 있습니다.
- 즉시 정지 입력 및 한계 입력 신호가 없으므로 즉시 정지 입력 및 정방향/역방향 한계 입력 검출에 관한 이상은 발생하지 않습니다.

● 엔코더 축/가상 엔코더 축

엔코더 축/가상 엔코더 축은 서보 축/가상 서보 축에 대해 아래와 같은 차이가 있습니다.

- 지령 위치를 갖지 않습니다. 피드백 위치만 가집니다.
- 동작계 모션 제어 명령은 사용할 수 없습니다.

사용상의 주의

단축 위치 제어 축의 축 종별에 엔코더 축/가상 엔코더 축은 지정할 수 없습니다.

*1. MC 기능 모듈 내부에서는 배정밀도 실수형으로 처리하고 있으므로 연산 오차를 포함하는 경우가 있습니다.

제어 기능

제어할 축의 기능을 선택합니다.
NX102 또는 NX1P2 CPU 유니트를 사용할 경우에만 선택할 수 있습니다.

제어 기능	내용
모두	전체 기능을 사용할 수 있는 축인「모션 제어 축」으로 설정됩니다.
단축 위치 제어만	단축 위치 제어만 사용할 수 있는「단축 위치 제어 축」으로 설정됩니다.

제어 기능 선택을 통해 사용할 수 있는 축 종별은 아래와 같습니다.

축 종별	제어 기능	
	[All]	[Single-axis position control only]
서보 축	○	○
가상 서보 축	○	○
엔코더 축	○	×
가상 엔코더 축	○	×

또한, 동작 가능한 기능은 아래와 같습니다.

모션 제어 기능	제어 기능	
	[All]	[Single-axis position control only]
단축 위치 제어	○	○
단축 동기 제어	○	×
단축 속도 제어	○	○ *1
단축 토크 제어	○	×
다축 협조 제어	○	×

*1. MC_MoveVelocity(속도 제어) 명령만 사용할 수 있습니다.

모션 제어 기능에 대한 상세한 내용은「제9장 모션 제어 기능」을 참조해 주십시오.

참고

NX102-90□□ 또는 NX1P2-90□□□□를 사용할 경우에는 [Single-axis position control only]를 선택해 주십시오.

입력 디바이스/출력 디바이스

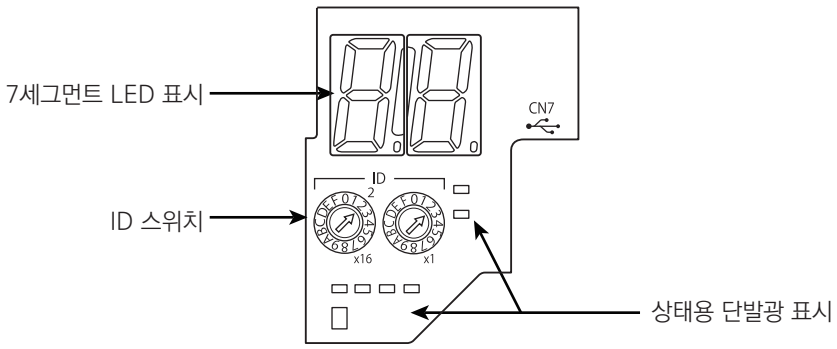
축 종별이 서보 축 또는 엔코더 축일 때 축에 할당할 EtherCAT 디바이스의 노드 주소를 선택합니다. NX 시리즈 위치 인터페이스 유니트는 EtherCAT 커플러 유니트의 노드 주소 및 위치 인터페이스 유니트의 NX 유니트 번호를 선택합니다.
축 종별이 가상 축인 경우에는 선택할 수 없습니다.

사용상의 주의

- 오므론 서보 드라이버 1S 시리즈 및 G5 시리즈는 노드 주소를 드라이버 전면의 노드 주소 스위치를 통해 고정값을 설정할 수 있습니다. 이 노드 주소 스위치를「00」으로 설정한 경우 노드 주소는 Sysmac Studio의 EtherCAT 에디터에서 설정한 값에 따릅니다.
  접속된 모든 드라이버에서 노드 주소 스위치를「00」으로 설정한 경우 드라이버의 접속 위치를 변경해도 이상이 발생하지는 않습니다. 장비의 각 제어 대상에 대해 특정 서보 드라이버를 할당할 때는 드라이버의 노드 주소는 노드 주소 스위치로 설정해 주십시오.
- 서보 드라이버의 노드 주소 스위치 설정값은 서보 드라이버의 제어 전원을 투입할 때 한 번만 판독됩니다.
  전원 투입 후에 변경해도 제어에 반영되지 않고 다음 전원 투입 시에 유효해집니다.
  전원 투입 후에는 노드 주소 스위치의 값을 변경하지 마십시오.
- 노드 주소가 중복된 경우에는 이상이 발생합니다.

참고

- EtherCAT 디바이스의 노드 주소 설정 예로 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입의 경우를 아래에 설명합니다.



- 서보 드라이버 표시부의 로터리 스위치를 사용하여 EtherCAT의 노드 주소를 설정합니다.

로터리 스위치의 설정값	노드 주소의 설정값	
	오므론 슬레이브	타사 슬레이브 *1
00	Sysmac Studio에서 설정한 값 (1~512) *2	Sysmac Studio에서 설정한 값 (1~512) *2
01~FF	노드 주소 스위치의 설정값	

*1. 타사 슬레이브에서는 슬레이브 측의 설정 유무와 관계없이 Sysmac Studio에서 설정한 값이 유효해집니다.
*2. NJ 시리즈 CPU 유니트의 경우 설정값은「1~192」입니다. 단, NJ101 CPU 유니트만 접속할 수 있는 슬레이브의 최대 수는 64슬레이브입니다.
NX102 CPU 유니트의 경우 설정값은「1~192」, 접속할 수 있는 슬레이브의 최대 수는 64슬레이브입니다.
NX1P2 CPU 유니트의 경우 설정값은「1~192」, 접속할 수 있는 슬레이브의 최대 수는 16슬레이브입니다.

5-2-3 단위 변환 설정

위치의 단위계를 설정합니다.

파라메터의 명칭	기능	설정 범위	초기값
표시 단위	지령 위치의 단위를 설정합니다. ・pulse ・mm ・μm ・nm ・degree ・inch	0~5	0
모터 1회전의 펄스 수 *1	엔코더 분해능에 맞춰 지령 위치의 모터 1회전당 펄스 수를 설정합니다. *2 전자 기어비를 바탕으로 지령값에서 펄스량으로 변환합니다.	1~4294967295	10000
모터 1회전의 이동량 *3	지령 위치의 모터 1회전당 워크의 이동량을 설정합니다.	0.000000001~4294967295	10000
감속기 사용 *4 *5	감속기 설정을 사용할지 지정합니다. TRUE: 사용함 FALSE: 사용하지 않음	TRUE, FALSE	FALSE
워크 측 1회전의 이동량 *4 *6 *7	워크 측이 1회전한 경우의 이동량을 설정합니다.	배정밀도 실수형의 양수	10000
워크 측 기어비 *4 *6	워크 측의 기어비를 설정합니다.	1~4294967295	1
모터 측 기어비 *4 *6	모터 측의 기어비를 설정합니다.	1~4294967295	1

*1. 전자 기어비(단위 변환식)의 분자입니다.
*2. 예를 들어 엔코더 분해능이 10,000펄스/회전인 경우에는 10,000을 설정해 주십시오.
*3. 감속기 사용을「사용하지 않음」으로 설정한 경우의 전자 기어비(단위 변환식)의 분모입니다.
감속기 사용을「사용함」으로 설정한 경우는 무효가 됩니다.
*4. 유니트 버전 Ver.1.11 이상인 CPU 유니트와 Sysmac Studio Ver.1.15 이상의 조합으로 사용할 수 있습니다.
*5. 감속기 사용을「사용함」으로 설정할 경우 단위 변환 설정의 설정에는 조건이 있습니다. 조건에 대해서는 「감속기 사용 조건 (P.5-12)」을 참조해 주십시오.
*6. 감속기 사용을「사용함」으로 설정한 경우에 유효해집니다.
*7. 카운터 모드가 리니어 모드인 경우에만 설정할 수 있습니다. 로터리 모드인 경우에는 링 카운터 상한 설정값 및 링 카운터 하한 설정값으로 산출됩니다.

사용상의 주의

- 카운트 모드를 로터리 모드로 사용할 때는 감속기 사용을「사용함」으로 설정해 주십시오.
  감속기 사용을「사용하지 않음」으로 설정하면 링 카운터의 1사이클을 펄스 단위로 변환할 때의 연산 오차에 의해 링 카운터 1사이클당 펄스 수가 기대한 정수가 되지 않는 경우가 있습니다. 기대한 정수가 되지 않으면 위치 이탈의 원인이 됩니다.
- 표시 단위의 변환을 제외하고 단위 변환 설정을 변경한 경우에는 물리적인 기계 위치와 MC 기능 모듈의 지령 현재 위치와의 관계에 차이가 발생합니다. 단위 변환 설정을 변경한 경우에는 원점 복귀 명령을 실행하고 다시 원점을 확정해 주십시오.

서보 드라이버 및 엔코더 입력 터미널과 MC 기능 모듈 간은 기본적으로 펄스 단위의 위치가 사용됩니다. 모션 제어 명령에서 사용하는 위치는 실제 동작을 알기 쉽도록 하기 위해 mm, degree 등의 표시 단위를 사용합니다.



표시 단위와 전자 기어(단위 변환식)의 설정에 따라 펄스 단위의 표기를 mm, degree 표기로 변환할 수 있습니다.

참고

가상 서보 축의 경우 지령 현재값을 펄스 단위로 변환하고, 그 값을 다시 표시 단위로 변환한 값을 피드백 현재값으로 합니다.



표시 단위

표시 단위는 Sysmac Studio에 표시하는 표시 단위의 설정입니다. 표시는 위치의 표시 단위입니다.
설정할 수 있는 단위에 대해 설명합니다.

설정 가능한 단위	설명
pulse	펄스 단위로 표현하는 경우에 사용합니다.
mm	비교적 긴 거리를 직동하는 경우에 사용합니다.
μm	정밀도가 높은 직동을 실행하는 경우에 사용합니다.
nm	μm보다 정밀도가 높은 직동을 실행하는 경우에 사용합니다.
degree	로터리 테이블 등 회전하는 축에 사용합니다.
inch	직동 등에서 사용합니다.

감속기 사용 조건

감속기 사용을「사용함」으로 설정할 경우 단위 변환 설정의 설정에는 다음 조건을 만족해야 합니다.

- 워크 측 1회전의 이동량×워크 측 기어비÷모터 측 기어비의 값이 0.000000001 이상, 4294967295 이하일 것.

카운트 모드가 로터리 모드인 경우에는 다음 조건 또한 충족해야 합니다.

- 모터 1회전의 펄스 수×모터 측 기어비가 1099511627775 이하일 것.

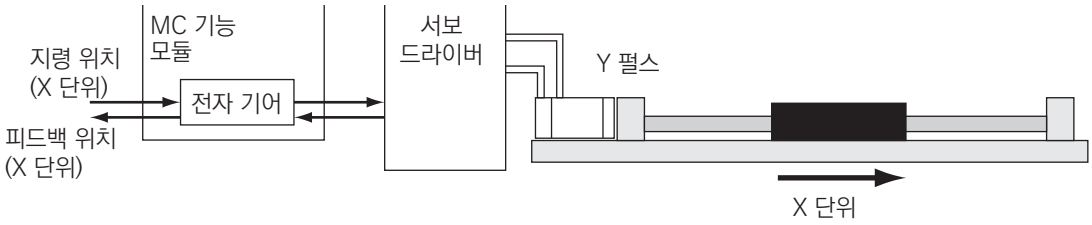
단, 이러한 조건의 워크 측 기어비와 모터 측 기어비의 값은 양쪽의 최대 공약수로 나는 값으로 계산하고 있습니다.
따라서 설정한 값 그대로 계산하면 조건이 만족하지 않아도 실제로는 조건을 만족하는 경우가 있습니다.

● **예**

워크 측 기어비가「4」, 모터 측 기어비가「6」인 경우 각각 워크 측 기어비를「2」, 모터 측 기어비를「3」으로 간주하여 계산합니다.

전자 기어비(단위 변환식)

MC 기능 모듈에서는 표시 단위와 펄스 단위의 관계를 설정하기 위해 전자 기어를 사용합니다.
Sysmac Studio를 사용하여 전자 기어비를 설정합니다.



지령 위치(펄스)값 = 지령 위치(X 단위)×전자 기어비

● 감속기 사용을「사용하지 않음」으로 설정한 경우

감속기 사용을「사용하지 않음」으로 설정한 경우 전자 기어비는 다음 식으로 나타냅니다.

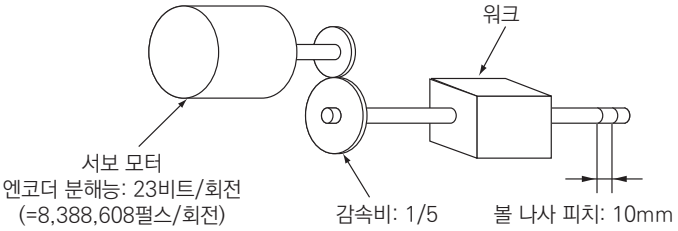
$$\text{전자 기어비} = \frac{\text{모터 1회전의 펄스 수}^{*1}\text{(Y 펄스)}}{\text{모터 1회전의 이동량}^{*2}\text{(X 단위)}}$$

*1. 엔코더 축은 엔코더 1회전당 펄스 수가 됩니다.
*2. 엔코더 축은 엔코더 1회전당 이동량이 됩니다.

사용상의 주의

MC 기능 모듈이나 모션 제어 명령에서 사용하는 위치는 전자 기어로 단위 변환한 값을 사용합니다.
모션 제어 명령에서는 목표 위치를 LREAL형으로 지정하나 전자 기어에서 펄스 단위로 변환한 후의 지령 위치가 40비트를 초과할 경우에는 명령에서 이상이 발생합니다.

오므론 서보 모터 1S 시리즈의 23비트 절대값 엔코더 부속 제품을 사용하는 경우를 예로 들어 설명합니다.
기계계는 감속기의 감속비가 1/5이며 볼나사가 1회전하면 워크가 10mm 이동하도록 합니다.



표시 단위는 mm로 합니다. 모터 1회전의 펄스 수에는 서보 모터의 엔코더 분해능을 설정합니다. 감속비가 1/5인 감속기를 사용하므로 서보 모터가 5회전하면 볼나사는 1회전합니다. 이때 워크는 2mm(=10mm×1/5) 이동하므로 모터 1회전의 이동량에는 「2」를 설정합니다.

설정 항목	설정 내용
표시 단위	mm
모터 1회전의 펄스 수	8,388,608
모터 1회전의 이동량	2

이 설정에 따라 사용자 프로그램에서의 위치 지령 단위는 1(mm)이 됩니다.
예를 들어 100.5mm인 위치로 절대값 위치를 결정할 경우에는 MC_MoveAbsolute(절대값 위치 결정) 명령의 입력 변수 「Position(목표 위치)」에 「100.5」를 설정합니다.

참고

예와 동일한 기계에서 감속비가 1/9인 감속기를 사용하는 경우
서보 모터가 1회전한 경우의 워크 이동량은 10mm×1/9 = 1.111···mm가 되어 나누어 떨어지지 않습니다.
이처럼 나누어 떨어지지 않는 숫자가 되는 경우에는 [모터 1 회전의 펄스 수]와 [모터 1 회전의 이동량]에 같은 계수를 곱한 값을 설정합니다.
감속비가 1/9이므로 계수를 「9」로 합니다.

- [모터 1 회전의 펄스 수] : 75,497,472 (=8,388,608×9)
- [모터 1 회전의 이동량] : 10 (=10×1/9×9)

● 감속기 사용을 「사용함」으로 설정하고 카운트 모드가 리니어 모드인 경우

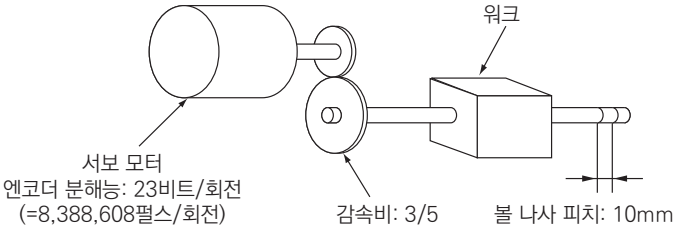
감속기 사용을 「사용함」으로 설정하고 카운트 모드가 리니어 모드인 경우 전자 기어비는 다음 식으로 나타냅니다.

$$\text{전자 기어비} = \frac{\text{모터 1회전의 펄스 수} \times \text{모터 측 기어비}}{\text{워크 1회전의 이동량} \times \text{워크 측 기어비}}$$

사용상의 주의

MC 기능 모듈이나 모션 제어 명령에서 사용하는 위치는 전자 기어로 단위 변환한 값을 사용합니다.
모션 제어 명령에서는 목표 위치를 LREAL형으로 지정하나 전자 기어에서 펄스 단위로 변환한 후의 지령 위치가 40비트를 초과할 경우에는 명령에서 이상이 발생합니다.

오므론 서보 모터 1S 시리즈의 23비트 절대값 엔코더 부속 제품을 사용하는 경우를 예로 들어 설명합니다.
기계계는 감속기의 감속비가 3/5이며 볼나사가 1회전하면 워크가 10mm 이동하도록 합니다.



표시 단위는 mm로 합니다. 모터 1회전의 펄스 수에는 서보 모터의 엔코더 분해능을 설정합니다.
워크 1회전의 이동량에는 볼나사 피치인 10mm를 설정합니다.
감속비가 3/5인 감속기를 사용하므로 서보 모터가 5회전하면 볼나사는 3회전합니다. 이때의 감속비 설정은 워크 측 기어비에 「3」, 모터 측 기어비에 「5」를 설정합니다.

설정 항목	설정 내용
표시 단위	mm
모터 1회전의 펄스 수	8,388,608
워크 1회전의 이동량	10
워크 측 기어비	3
모터 측 기어비	5

이 설정에 따라 사용자 프로그램에서의 위치 지령 단위는 1(mm)이 됩니다.
예를 들어 100.5mm인 위치로 절대값 위치를 결정할 경우에는 MC_MoveAbsolute(절대값 위치 결정) 명령의 입력 변수 「Position(목표 위치)」에 「100.5」를 설정합니다.

● 감속가 사용을 「사용함」으로 설정하고 카운트 모드가 로터리 모드인 경우

감속기 사용을 「사용함」으로 설정하고 카운트 모드가 로터리 모드인 경우 전자 기어비는 다음 식으로 나타냅니다.
로터리 모드의 경우 워크 1회전의 이동량은 「｜링 카운터 상한값 - 링 카운터 하한값｜」으로 자동 설정됩니다.

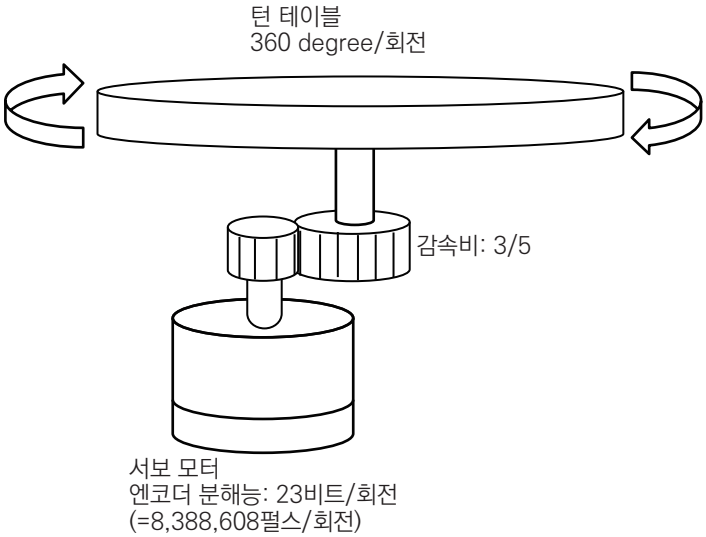
$$\text{전자 기어비} = \frac{\text{모터 1회전의 펄스 수} \times \text{모터 측 기어비}}{\text{워크 1회전의 이동량} \times \text{워크 측 기어비}}$$

$$= \frac{\text{모터 1회전의 펄스 수} \times \text{모터 측 기어비}}{|\text{링 카운터 상한값} - \text{링 카운터 하한값}| \times \text{워크 측 기어비}}$$

사용상의 주의

MC 기능 모듈이나 모션 제어 명령에서 사용하는 위치는 전자 기어로 단위 변환한 값을 사용합니다.
모션 제어 명령에서는 목표 위치를 LREAL형으로 지정하나 전자 기어에서 펄스 단위로 변환한 후의 지령 위치가 40비트를 초과할 경우에는 명령에서 이상이 발생합니다.

오므론 서보 모터 1S 시리즈의 23비트 절대값 엔코더 부속 제품을 사용하는 경우를 예1로 들어 설명합니다.
기계계는 감속기의 감속비가 3/5이며 턴테이블이 1회전하면 워크가 360degree 이동합니다.



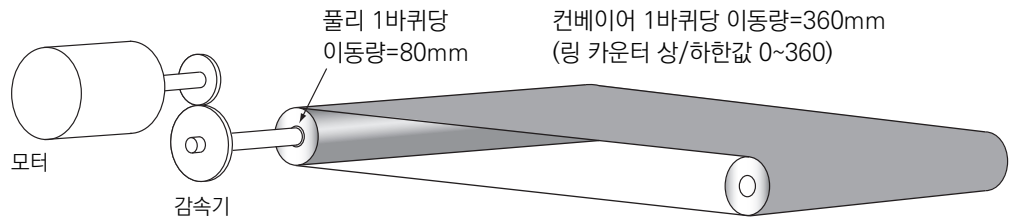
표시 단위는 degree로 합니다. 모터 1회전의 펄스 수에는 서보 모터의 엔코더 분해능을 설정합니다.
워크 1회전의 이동량은「｜링 카운터 상한값 - 링 카운터 하한값｜」으로 자동 설정됩니다.
감속비가 3/5인 감속기를 사용하므로 서보 모터가 5회전하면 턴테이블(워크)은 3회전합니다. 이때의 감속비 설정은 워크 측 기어비에「3」, 모터 측 기어비에「5」를 설정합니다.

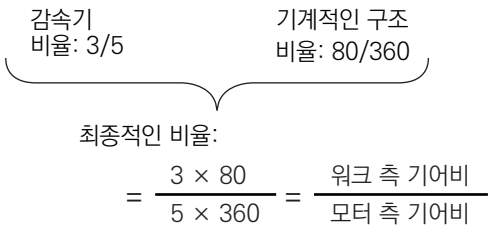
설정 항목	설정 내용
표시 단위	degree
모터 1회전의 펄스 수	8,388,608
워크 1회전의 이동량	｜링 카운터 상한값 - 링 카운터 하한값｜
워크 측 기어비	3
모터 측 기어비	5

이 설정에 따라 사용자 프로그램에서의 위치 지령 단위는 1(degree)이 됩니다.
예를 들어 100.5degree인 위치로 절대값 위치를 결정할 경우에는 MC_MoveAbsolute(절대값 위치 결정) 명령의 입력 변수「Position(목표 위치)」에「100.5」를 설정합니다.

오므론 서보 모터 1S 시리즈의 23비트 절대값 엔코더 부속 제품을 사용하는 경우를 예2로 들어 설명합니다.
기계계는 감속기의 감속비가 3/5이며 풀리가 1회전하면 컨베이어가 80mm 이동합니다. 컨베이어는 360mm마다 1바퀴씩 돕니다.

서보 모터
엔코더 분해능: 23비트/회전
(=8,388,608펄스/회전)





표시 단위는 mm로 합니다. 모터 1회전의 펄스 수에는 서보 모터의 엔코더 분해능을 설정합니다.
컨베이어 1회전의 이동량은 「｜링 카운터 상한값 - 링 카운터 하한값｜」으로 자동 설정됩니다.

설정 항목	설정 내용
링 카운터 상한값	360
링 카운터 하한값	0

감속비가 3/5인 감속기를 사용하므로 서보 모터가 5회전하면 풀리는 3회전합니다. 또한, 풀리 1바퀴당 컨베이어의 이동량이 80mm, 컨베이어가 1바퀴 도는 거리가 360mm이므로 감속비의 설정은 워크 측 기어비에 「3×80」인 「240」, 모터 측 기어비에 「5×360」인 「1,800」을 설정합니다.

설정 항목	설정 내용
표시 단위	mm
모터 1회전의 펄스 수	8,388,608
워크 1회전의 이동량	｜링 카운터 상한값 - 링 카운터 하한값｜
워크 측 기어비	240
모터 측 기어비	1800

참고

- 풀리 1회전당 이동량이 80.1mm 등 정수가 아닐 때는 정수가 되는 계수 「10」을 곱한 비율을 사용하여 워크 측 기어비, 모터 측 기어비를 산출해 주십시오. 위의 예를 바탕으로 하면 워크 측 기어 비율이 「3×80.1×10」인 「2,403」이 되고, 모터 측 기어 비율은 「5×360×10」인 「18,000」이 됩니다.
- 감속기가 다단인 경우에도 위 예2와 같이 복수의 비율을 곱한 최종적인 비율을 워크 측 기어비, 모터 측 기어비로 설정해 주십시오.

5-2-4 동작 설정

제어하는 장비의 사양에 맞춰 최고 속도 및 최대 가감 속도 등의 축 동작에 대해 설정합니다.

파라메터의 명칭	기능	설정 범위	초기값
최고 속도	각 축의 최고 속도를 설정합니다. *1 사용하는 모터의 최대 회전수를 초과하지 않도록 설정해 주십시오. (단위: 지령 단위/s)	배정밀도 실수형의 양수 *2	400,000,000
기동 속도 *3	각 축의 기동 속도를 설정합니다. 최고 속도를 초과하지 않도록 설정해 주십시오. (단위: 지령 단위/s)	배정밀도 실수형의 양수	0
조그 최고 속도	각 축의 조그 최고 속도를 설정합니다. *4 최고 속도를 초과하지 않도록 설정해 주십시오. (단위: 지령 단위/s)	배정밀도 실수형의 양수	1,000,000
최대 가속도	축 동작 지령 시 각 축의 최대 가속도를 설정합니다. 「0」을 설정하면 가속도의 제한이 없어집니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
최대 감속도	축 동작 지령 시 각 축의 최대 감속도를 설정합니다. 「0」을 설정하면 감속도의 제한이 없어집니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
가감속 초과	축 가감속 제어에서 목표 위치에 대한 정지를 우선한 결과 감속이 초과되어 최대 가감 속도를 초과한 경우의 동작을 지정합니다. 0: 가감 속도를 빠르게 사용한다(블렌딩을 버퍼드로 변환한다). *5 1: 가감 속도를 빠르게 사용한다. 2: 이상으로 인식하여 정지한다. *6	0~2	0
반전 시 동작	명령 다중 기동 시, 명령 재기동 시, 인터럽트 피딩 위치 결정 시의 반전 시 동작을 지정합니다. *7 0: 감속 정지 1: 즉시 정지	0~1	0
속도 경고값	각 축의 속도 경고를 출력하기 위한 최고 속도에 대한 비율을 설정합니다. 「0」을 설정하면 속도 경고는 출력하지 않습니다. (단위: %)	0~100	0
가속도 경고값	각 축의 가속도 경고를 출력하기 위한 최고 가속도에 대한 비율을 설정합니다. 「0」을 설정하면 가속도 경고는 출력하지 않습니다. (단위: %)	0~100	0
감속도 경고값	각 축의 감속도 경고를 출력하기 위한 최고 감속도에 대한 비율을 설정합니다. 「0」을 설정하면 감속도 경고는 출력하지 않습니다. (단위: %)	0~100	0
정방향 토크 경고값 *8	각 축의 정방향 토크 경고를 출력하기 위한 토크 지령값을 설정합니다. 「0」을 설정하면 정방향 토크 경고는 출력하지 않습니다. (단위: %)	0~1000	0
역방향 토크 경고값 *8	각 축의 역방향 토크 경고를 출력하기 위한 토크 지령값을 설정합니다. 「0」을 설정하면 역방향 토크 경고는 출력하지 않습니다. (단위: %)	0~1000	0

파라메터의 명칭	기능	설정 범위	초기값
피드백 속도 필터 시정수	피드백 속도의 이동 평균을 산출하는 기간을 밀리초 단위로 설정합니다. 「0」을 설정하면 이동 평균은 산출하지 않습니다. (단위: ms) 축 속도가 느릴 때 피드백 현재 속도의 편차를 줄이기 위해 사용합니다.	0~100	0
인포지션 폭 *9	위치 결정 완료 폭을 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0	10
인포지션 체크 시간 *9	위치 결정 완료 체크 시간을 밀리초 단위로 설정합니다. 「0」을 설정하면 원점 복귀 시에 원점을 확정했을 때에만 위치 결정 완료 체크를 계속하고, 원점 복귀 시 이외에는 위치 결정 완료 체크는 하지 않습니다. (단위: ms)	0~10,000	0
원점 위치 범위	원점 위치 검출 폭을 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0	10

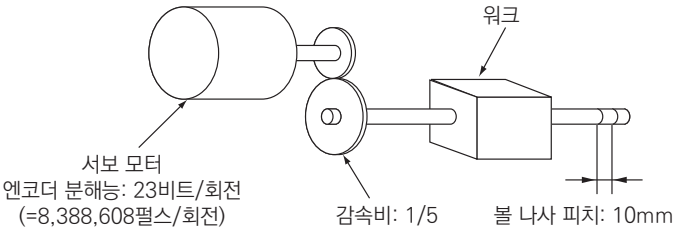
*1. 최고 속도를 초과하는 속도 지령값을 지정한 경우에는 최고 속도로 지령합니다. 보간 제어 동작 시에도 유효합니다.
축에 접속하는 서보 모터/드라이버가 출력 가능한 모터 최고 속도(최고 회전수)를 초과하는 값은 설정하지 마십시오. 접속하는 서보 모터/드라이버의 사양을 확인한 후 설정해 주십시오.
*2. 설정할 수 있는 최대값은 펄스 단위로 변환한 값으로 다음과 같습니다.
유니트 버전 Ver.1.02 이하인 CPU 유니트는 400,000,000 [pulse/s],
유니트 버전 Ver.1.03 이상인 CPU 유니트는 500,000,000 [pulse/s],
유니트 버전 Ver.1.11 이상인 CPU 유니트는 2,147,483,647 [pulse/s]입니다.
*3. 유니트 버전 Ver.1.05 이상인 CPU 유니트와 Sysmac Studio Ver.1.06 이상의 조합으로 사용할 수 있습니다.
*4. 조그 최고 속도를 초과하는 속도 지령값을 지정한 경우에는 조그 최고 속도로 지령합니다.
*5. 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 블렌딩을 버퍼드로 변환하지 않습니다. 상세한 내용은 「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.
*6. 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 블렌딩 동작의 경우 이상으로 정지하지 않습니다. 상세한 내용은 「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.
*7. 반전 시 동작에 대한 상세한 내용은 「9-5-6 모션 제어 명령의 재기동 (P.9-40)」 및 「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.
*8. 토크 제어 시에만 유효합니다.
*9. 인포지션 체크는 MC 기능 모듈 측에서 처리합니다. 서보 드라이버 측 기능은 사용하지 않습니다.

최고 속도

최고 속도의 설정 예에 대해 설명합니다.

● 최고 속도의 설정 예

앞에 설명한 「감속기 사용을 「사용하지 않음」으로 설정한 경우 (P.5-13)」와 같은 기계에서 서보 모터의 최대 회전수가 6,000r/min인 경우에 대해 설명합니다.



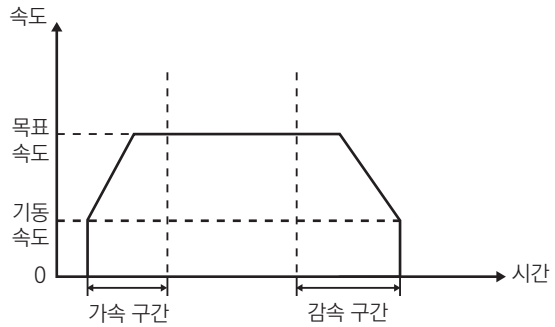
최대 회전수: 6,000r/min, 감속비: 1/5, 볼나사 피치: 10mm인 각 조건을 바탕으로 계산하면 6,000r/min × 1/5 × 10mm = 12,000mm/min = 200mm/s가 되므로 최고 속도에는「200」을 설정합니다.
초기값 400,000,000은 모터의 최대 회전수를 초과했으므로 반드시 설정값을 변경해 주십시오.

설정 항목	설정 내용
표시 단위	mm
모터 1회전의 펄스 수	8,388,608
모터 1회전의 이동량	2
최고 속도	200

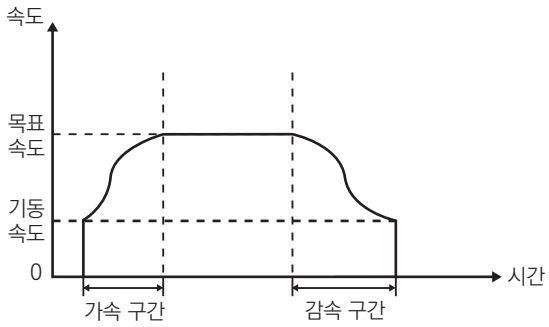
기동 속도

서보 모터를 사용할 때는「0」으로 해 주십시오.
스테핑 모터를 사용할 때는 기동 시의 탈조 방지를 위해 일반적으로 최대 자기동 주파수의 1/10~1/2인 값을 설정합니다. 부하 상황에 따라 바뀌므로 상세한 내용은 스테핑 모터 매뉴얼을 확인해 주십시오.

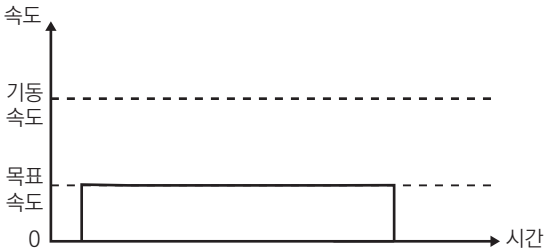
저크를 지정하지 않을 때



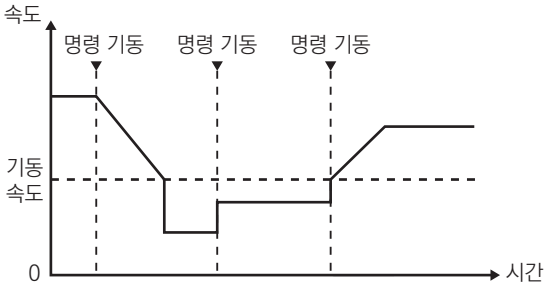
저크를 지정했을 때



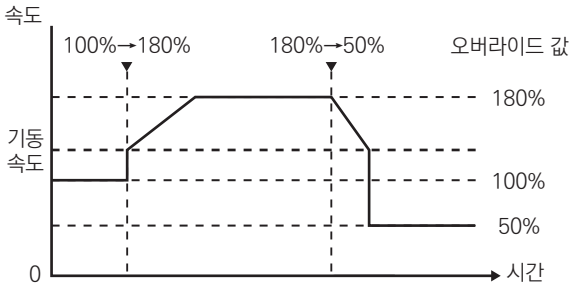
목표 속도가 기동 속도 이하일 때는 가감속을 하지 않고 목표 속도로 동작합니다.



동작 중에 모션 제어 명령의 재기동이나 다중 기동으로 목표 속도를 변경한 경우에도 기동 속도가 적용됩니다. 목표 속도가 기동 속도를 초과한 경우에는 지정한 가속도와 감속도로 가감속 동작을 실행합니다. 목표 속도가 기동 속도 이하일 때는 가감속하지 않고 동작합니다.



MC_SetOverride(오버라이드값 설정) 명령으로 속도를 변경한 경우에도 기동 속도가 적용됩니다. 목표 속도가 기동 속도를 초과한 경우에는 지정한 가속도와 감속도로 가감속 동작을 실행합니다. 목표 속도가 기동 속도 이하일 때는 가감속하지 않고 동작합니다.



다음 기능에서는 기동 속도가 적용되지 않습니다.

- 토크 제어 모드
- 사이클릭 동기 속도 제어
- 사이클릭 동기 위치 결정
- 동기 제어
  단, MC_GearOut(기어 동작 해제), MC_CamOut(캠 동작 해제)에서는 기동 속도가 적용됩니다.
- 다축 협조 제어
  단,「Decelerate axes to a stop at maximum」으로 감속할 때는 각 축의 기동 속도가 적용됩니다.

5-2-5 확장 동작 설정

각종 입력 신호가 유효해졌을 때의 정지 방법이나 토크 제한값 등을 설정합니다.

파라메터의 명칭	기능	설정 범위	초기값
즉시 정지 입력 정지 방법	즉시 정지 입력이 유효해졌을 때 MC 기능 모듈 측에서 정지하는 방법을 설정합니다. 0: 지령값을 즉시 정지 2: 지령값을 즉시 정지하고 동시에 편차 카운터 리셋을 실행 3: 지령값을 즉시 정지하고 동시에 서보 OFF를 실행	0, 2, 3	0
한계 입력 정지 방법	정방향 한계 입력 또는 역방향 한계 입력이 유효해졌을 때 MC 기능 모듈 측에서 정지하는 방법을 설정합니다. 0: 지령값을 즉시 정지 1: 지령값을 감속 정지 2: 지령값을 즉시 정지하고 동시에 편차 카운터 리셋을 실행 3: 지령값을 즉시 정지하고 동시에 서보 OFF를 실행	0~3	0
드라이버 에러 리셋 감시 시간	드라이버 에러 리셋 실행 시의 감시 시간을 설정합니다. (단위: ms) 감시 시간이 지나면 드라이버 이상이 리셋되지 않아도 리셋 처리를 종료합니다.	1~1000	200
정방향 토크 제한 상한값	정방향 토크 제한값의 상한을 설정합니다. *1 (단위: %)	0.0~1000.0	300.0
역방향 토크 제한 상한값	역방향 토크 제한값의 상한을 설정합니다. *1 (단위: %)	0.0~1000.0	300.0
즉시 정지 입력 논리 반전 *2	즉시 정지 입력 신호의 논리 반전을 설정합니다. FALSE: 반전하지 않음 TRUE: 반전함	FALSE, TRUE	FALSE *3
정방향 한계 입력 논리 반전 *2	정방향 한계 입력 신호의 논리 반전을 설정합니다. FALSE: 반전하지 않음 TRUE: 반전함	FALSE, TRUE	FALSE *3
역방향 한계 입력 논리 반전 *2	역방향 한계 입력 신호의 논리 반전을 설정합니다. FALSE: 반전하지 않음 TRUE: 반전함	FALSE, TRUE	FALSE *3

파라메터의 명칭	기능	설정 범위	초기값
원점 근방 입력 논리 반전*2	원점 근방 입력의 논리 반전을 설정합니다. FALSE: 반전하지 않음 TRUE: 반전함	FALSE, TRUE	FALSE

*1. PDO 매핑에 「정방향 토크 제한값(60E0Hex)」 및 「역방향 토크 제한값(60E1Hex)」을 매핑했을 때 이러한 설정값이 EtherCAT 프로세스 데이터 통신으로 송신됩니다. MC_SetTorqueLimit(토크 제한) 명령으로 토크 제한을 유효하게 만든 경우에는 명령의 입력 변수에 지정한 값이 송신됩니다.
*2. 유니트 버전 Ver.1.05 이상인 CPU 유니트와 Sysmac Studio Ver.1.06 이상의 조합으로 사용할 수 있습니다.
유니트 버전 Ver.1.04 이하인 CPU 유니트에서는 논리를 반전하지 않습니다.
NX 시리즈 디지털 입력 유니트 등 입력 신호의 논리를 설정할 수 없는 기기를 할당한 경우에 설정합니다.
오므론 서보 드라이버 1S 시리즈 등 입력 신호의 논리를 설정할 수 있는 기기에서는 일반적으로 「반전하지 않음」으로 설정합니다.
*3. NX 시리즈 펄스 출력 유니트를 할당한 축은 초기값이 「TRUE」가 됩니다.

5-2-6 리미트 설정

위치 편차에 관한 리미트 기능과 소프트웨어를 통한 리미트 기능을 선택합니다.

파라메터의 명칭	기능	설정 범위	초기값
소프트웨어 리미트 기능 *1	소프트웨어 리미트의 기능을 선택합니다. 0: 무효 1: 지령 위치에 대해 유효하면 감속 정지 2: 지령 위치에 대해 유효하면 즉시 정지 3: 피드백 위치에 대해 유효하면 감속 정지 4: 피드백 위치에 대해 유효하면 즉시 정지	0~4	0
정방향 소프트웨어 리미트	정방향 측 소프트웨어 리미트를 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 음수, 양수, 0	2147483647
역방향 소프트웨어 리미트	역방향 측 소프트웨어 리미트를 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 음수, 양수, 0	-2147483648
위치 편차 초과값	위치 편차 초과 체크값을 설정합니다. 「0」을 설정하면 위치 편차 초과 체크가 무효해집니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0	0
위치 편차 경고값	위치 편차 경고 체크값을 설정합니다. 「0」을 설정하면 위치 편차 경고 체크가 무효해집니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0 (위치 편차 초과값 이하)	0

*1. 이 기능은 카운트 모드가 리니어 모드에서 원점 확정된 경우에만 기능합니다.
소프트웨어 리미트 기능에 대한 상세한 내용은 📖 「9-8-5 소프트웨어 리미트 기능 (P.9-73)」을 참조해 주십시오.

5-2-7 위치 카운트 설정

위치의 카운트 모드를 설정합니다.

NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

파라메터의 명칭	기능	설정 범위	초기값
카운트 모드	위치의 카운트 모드를 설정합니다. 0: 리니어 모드(유한 길이) 1: 로터리 모드(무한 길이)	0~1	0
링 카운터 상한 설정값	카운트 모드를 로터리 모드로 설정한 경우의 링 카운터 상한값을 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 음수, 양수, 0	2147483647
링 카운터 하한 설정값	카운트 모드를 로터리 모드로 설정한 경우의 링 카운터 하한값을 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 음수, 양수, 0	-2147483648
엔코더 종별	엔코더 종별을 선택합니다. *1 *2 0: 인크리멘털 엔코더(INC) 1: 절대값 엔코더(ABS)	0~1	0

*1. 아래와 같이 사용할 때는 엔코더 종별로 1(절대값 엔코더(ABS))을 설정해 주십시오.
오므론 서보 모터/드라이버 1S 시리즈를 사용하는 경우
오므론 서보 모터/드라이버 G5 시리즈에서 절대값 엔코더 부속 서보 모터를 사용하는 경우
오므론 서보 모터/드라이버 G5 시리즈에서 풀 클로즈 제어용으로 절대값 타입의 외부 스케일을 사용하는 경우
오므론 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입에서 절대값 타입의 외부 스케일을 사용하는 경우

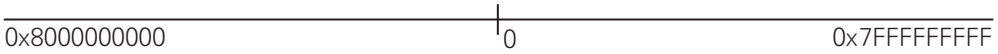
*2. 오므론 서보 모터/드라이버 G5 시리즈에서 풀 클로즈 제어용으로 외부 스케일을 사용하는 경우 또는 오므론 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 타입을 사용하는 경우에는 각각 아래와 같이 바꾸어 읽어 주십시오.
0: 인크리멘털 타입의 외부 스케일
1: 절대값 타입의 외부 스케일

카운트 모드

축 전송 모드를 말합니다. 축마다 지령 위치의 카운트 모드를 선택합니다. 축 전송 범위가 유한 길이인 리니어 모드와 무한 길이인 로터리 모드가 있습니다.

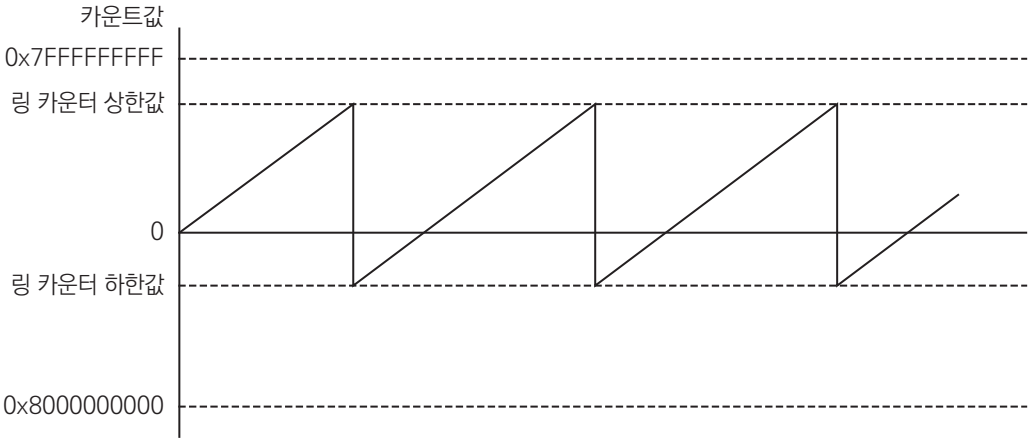
● 리니어 모드(유한 길이 축)

- 「0」을 중심으로 하는 직선적인 설정이 됩니다.
  X-Y 스테이지 등 기계적인 동작 범위가 한정되는 장비 등에서 사용합니다.
- 설정 범위는 펄스 단위 변환 후에 부호가 있는 정수형 40비트(0x8000000000~7FFFFFFFFF)입니다.
- 상대값 이동이나 절대값 이동 등의 목표 위치를 지정하는 위치 결정에서는 범위를 초과하는 목표 위치를 지령할 수 없습니다.
- 속도 제어 및 원점 복귀, 토크 제어 등의 목표 위치를 지정하지 않은 동작에서 동작 범위를 초과하면 감시 정보(지령 위치 오버 플로우/언더 플로우)가 됩니다. 지령 위치의 출력은 계속되나 현재 위치는 갱신되지 않고 상한값 또는 하한값으로 고정됩니다.
- 현재 위치가 고정된 상태이면 목표 위치를 지정하지 않는 동작으로 범위 내에 들어가는 방향에 대한 지령 및 정지를 기동할 수 있습니다. 범위 내에서 멀어지는 방향에 대한 지령은 명령 기동 시에 이상이 발생합니다.
- 오버 플로우/언더 플로우 상태를 해제할 때까지 현재 위치는 갱신할 수 없습니다.



● 로터리 모드(무한 길이 축)

- 설정 범위에서 반복하여 무한으로 카운트하는 링 카운터 형식의 모드입니다.
  로터리 테이블이나 와인딩 축 등에서 사용합니다.
- 링 카운터 범위는 Sysmac Studio에서 링 카운터 상한값과 링 카운터 하한값을 설정합니다.
- 설정 범위는 펄스 단위 변환 후에 부호가 있는 정수형 40비트(0x8000000000~7FFFFFFFFF)입니다.



- 링 카운터의 1사이클을 펄스 단위로 변환할 때의 연산 오차에 의해 링 카운터 1사이클당 펄스 수가 기대한 정수가 되지 않는 경우가 있습니다. 기대한 정수가 되지 않으면 위치 이탈의 원인이 됩니다.

버전 관련 정보

유니트 버전 V1.11 이상의 CPU 유니트를 사용할 때는 감속기 사용을 「사용함」으로 설정해 주십시오.
「사용함」으로 설정하면 상기 위치 이탈의 요인을 해소할 수 있습니다.
감소기 사용에 대해서는 「5-2-3 단위 변환 설정 (P.5-11)」을 참조해 주십시오.

링 카운터 상한 설정값과 링 카운터 하한 설정값

카운트 모드를 로터리 모드로 설정한 경우에 유효해지는 설정값입니다.
링 카운터의 상한값과 하한값을 설정합니다.

사용상의 주의

- 링 카운터의 상한값과 하한값 사이에 「0」이 포함되지 않는 경우 MC_MoveZeroPosition(고속 원점 복귀) 명령을 실행하면 이상이 발생합니다.
- MC_MoveAbsolute(절대값 위치 결정) 명령이나 MC_Move(위치 결정) 명령으로 절대값 위치 결정을 실행하는 경우에는 목표 위치를 링 카운터의 범위 내로 설정해 주십시오. 범위 외일 때는 이상이 발생합니다.
  단, 명령의 입력 변수 Direction(방향 선택)에서 「4: 방향 지정 없음」을 지정한 경우에는 범위 외도 목표 위치로 설정할 수 있습니다. 이때의 동작은 목표 위치와 지령 현재 위치의 차분을 목표 거리로 하는 상대값 위치 결정이 됩니다.
- 위치 카운트 설정을 변경한 경우에는 물리적인 기계 위치와 MC 기능 모듈의 지령 현재 위치와의 관계에 차이가 발생합니다. 위치 카운트를 설정한 경우에는 원점 복귀 명령을 실행하고 다시 원점을 확정해 주십시오.

엔코더 종별

피드백 입력하는 엔코더의 종별을 선택합니다.
아래와 같이 사용할 때는 엔코더 종별로 1(절대값 엔코더(ABS))을 설정해 주십시오.

- 오므론 서보 모터/드라이버 1S 시리즈를 사용하는 경우
- 오므론 서보 모터/드라이버 G5 시리즈에서 절대값 엔코더 부속 서보 모터를 사용하는 경우
- 오므론 서보 모터/드라이버 G5 시리즈에서 풀 클로즈 제어용으로 절대값 타입의 외부 스케일을 사용하는 경우
- 오므론 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입에서 절대값 타입의 외부 스케일을 사용하는 경우

5-2-8 서보 드라이버 설정

접속된 서보 드라이버 또는 엔코더 입력 터미널에서 설정된 값을 설정합니다.

파라메터의 명칭	기능	설정 범위	초기값 *1
링 카운터 상한 설정값	서보 드라이버 또는 엔코더 입력 터미널에서 설정된 링 카운터 상한값을 설정합니다. *2	-2^{63}~2^{63}-1 (단위: 펄스)	2147483647
링 카운터 하한 설정값	서보 드라이버 또는 엔코더 입력 터미널에서 설정된 링 카운터 하한값을 설정합니다. *2	-2^{63}~2^{63}-1 (단위: 펄스)	-2147483648
PDS 상태 제어 방식 *3	MC_Power 명령으로 서보를 OFF할 때 PDS 상태가 바뀌는 곳을 설정합니다. *4 0: 서보 OFF일 때 Switeched on 1: 서보 OFF일 때 Ready to switch on	0~1	0

*1. 초기값은 정수형(DINT)의 전체 범위로 되어 있습니다. 오므론 서보 드라이버 1S 시리즈 및 G5 시리즈는 정수형(DINT)의 전체 범위이므로 초기값 그대로 사용할 수 있습니다.
NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.
*2. 오므론 GX 시리즈 EtherCAT 슬레이브 엔코더 입력 터미널을 사용할 경우 엔코더 입력 터미널의 링 카운터 최대값(대상 인덱스 0x4003)과 링 카운터 상한 설정값을 일치시켜야 합니다. 또한, 링 카운터 하한 설정값을「0」으로 설정해야 합니다.
*3. 유니트 버전 Ver.1.10 이상인 CPU 유니트와 Sysmac Studio Ver.1.12 이상의 조합으로 사용할 수 있습니다.
*4. 오므론 서보 드라이버 G5 시리즈 사용 시「1」로 설정하면 서보 드라이버의 서보 준비(Switeched on)를 사용할 수 없습니다. 서보 준비(Switeched on)를 사용할 경우에는「0」으로 설정해 주십시오. PDS 상태의 변이에 대해서는 📖「A-5 PDS 상태 변이 (P.A-28)」를 참조해 주십시오.

5-2-9 원점 복귀 설정

기계 원점을 결정하기 위한 모터 동작에 대해 설정합니다.

파라메터의 명칭	기능	설정 범위	초기값
원점 복귀 동작 *1 *2	원점 복귀 동작 방법을 설정합니다. 0: 근방 퇴피 · 원점 근방 입력 OFF 지정 1: 근방 퇴피 · 원점 근방 입력 ON 지정 4: 원점 근방 입력 OFF 지정 5: 원점 근방 입력 ON 지정 8: 한계 입력 OFF 지정 9: 근방 퇴피 · 원점 입력 마스터 거리 지정 11: 한계 입력만 12: 근방 퇴피 · 보류 시간 지정 13: 원점 근방 입력 없음 · 보류 원점 입력 지정 14: 원점 프리셋	0, 1, 4, 5, 8, 9, 11~14	14
원점 입력 신호	원점 입력 신호를 선택합니다. 0: Z상 입력을 사용한다. 1: 외부 원점 입력을 사용한다. *3	0~1	0
원점 복귀 시작 방향	원점 복귀 기동 시의 시작 방향을 설정합니다. 0: 정방향 2: 역방향	0, 2	0
원점 검출 방향	원점 복귀의 원점 입력 검출 방향을 설정합니다. 0: 정방향 2: 역방향	0, 2	0
정방향 한계 입력 시 동작	원점 복귀 중에 정방향 한계 입력이 「ON」한 경우의 정지 방법을 설정합니다. 0: 반전하지 않음 · 축 이상으로 정지 (정지 방법은 한계 입력 정지 방법에 따름) 1: 반전함 · 즉시 정지 2: 반전함 · 감속 정지	0~2	1
역방향 한계 입력 시 동작	원점 복귀 중에 역방향 한계 입력이 「ON」한 경우의 정지 방법을 설정합니다. 0: 반전하지 않음 · 축 이상으로 정지 (정지 방법은 한계 입력 정지 방법에 따름) 1: 반전함 · 즉시 정지 2: 반전함 · 감속 정지	0~2	1
원점 복귀 속도	원점 복귀 중인 속도를 설정합니다. (단위: 지령 단위/s)	배정밀도 실수형의 양수	10,000
원점 복귀 접근 속도	원점 근방 입력이 「ON」한 후의 속도를 설정합니다. (단위: 지령 단위/s)	배정밀도 실수형의 양수	1,000
원점 복귀 가속도	원점 복귀 시의 가속도를 지정합니다. 「0」을 설정하면 가속하지 않고 원점 복귀 속도 등의 목표로 하는 속도에 도달합니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
원점 복귀 감속도	원점 복귀 시의 감속도를 지정합니다. 「0」을 설정하면 감속하지 않고 원점 복귀 접근 속도 등의 목표로 하는 속도에 도달합니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
원점 복귀 저크	원점 복귀 시의 저크를 지정합니다. 「0」을 설정하면 저크 없음이 됩니다. (단위: 지령 단위/s^3)	배정밀도 실수형의 양수, 0	0
원점 입력 마스크 거리	원점 복귀 동작 모드가 근방 퇴피 · 원점 입력 마스크 거리 지정인 경우의 원점 입력의 마스크 거리를 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0	10,000
원점 위치 오프셋	원점 복귀 완료 시에 설정한 값으로 현재 위치를 프리셋합니다. (단위: 지령 단위)	배정밀도 실수형의 음수, 양수, 0	0
원점 복귀 보류 시간	원점 복귀 동작 모드가 근방 퇴피 · 보류 시간 지정인 경우의 보류 시간을 밀리초 단위로 설정합니다. (단위: ms)	0~10,000	100

파라메터의 명칭	기능	설정 범위	초기값
원점 복귀 보정값	원점 복귀의 원점 확정 후 원점 복귀 보정량을 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 음수, 양수, 0	0
원점 복귀 보정 속도	원점 복귀 보정 시의 속도를 설정합니다. (단위: 지령 단위/s)	배정밀도 실수형의 양수	1,000

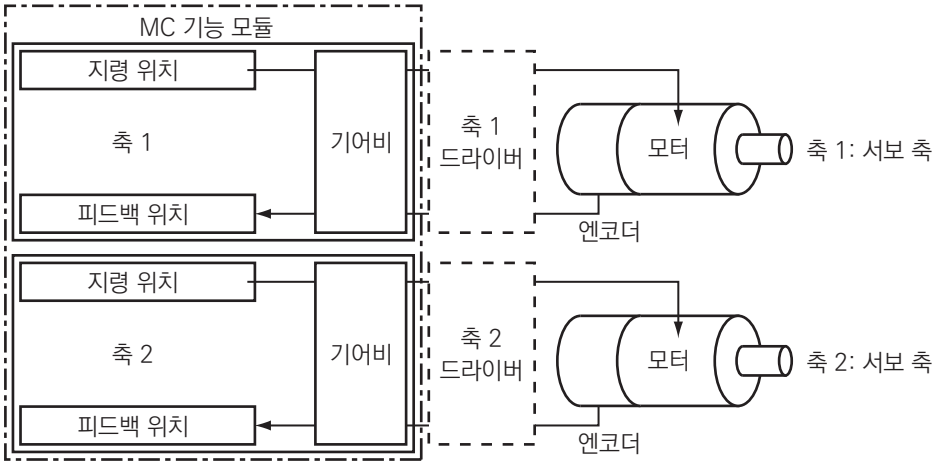
*1. 원점 복귀 동작의 설정입니다. 상세한 내용에 대해서는 🕮「제8장 원점 결정」을 참조해 주십시오.
*2. 오므론 서보 드라이버 G5 시리즈 중 리니어 모터 타입은 Z상 입력을 PDO에 매핑할 수 없습니다. PDO에 매핑한 Z상 입력을 사용하는 원점 복귀 동작인「원점 근방 입력 없음 · 보류 원점 입력 지정」을 사용할 때는 원점 입력 신호에 Z상 입력을 선택하지 마십시오.
*3. 오므론 서보 드라이버 1S 시리즈 및 G5 시리즈를 사용할 때 지정할 수 있습니다. 서보 드라이버의 래치 기능 1에 할당된 입력을 외부 원점 입력으로 사용합니다. 1S 시리즈 및 G5 시리즈의 초기 상태에서는 래치 기능 1에 외부 래치 입력 1이 할당되어 있습니다.
상세한 내용에 대해서는 🕮『AC 서보 모터/드라이버 1S 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼(SBCE-377)』 및 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼(SBCE-365)』, 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입 사용자 매뉴얼(SBCE-366)』을 참조해 주십시오.
NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

5-2-10 축 파라메터 설정 예

축 파라메터 중 위치에 관한 항목의 설정 예를 설명합니다.

단축에서의 위치 결정

2축을 각각 단축으로 위치 결정하는 장비를 예를 들어 설명합니다.

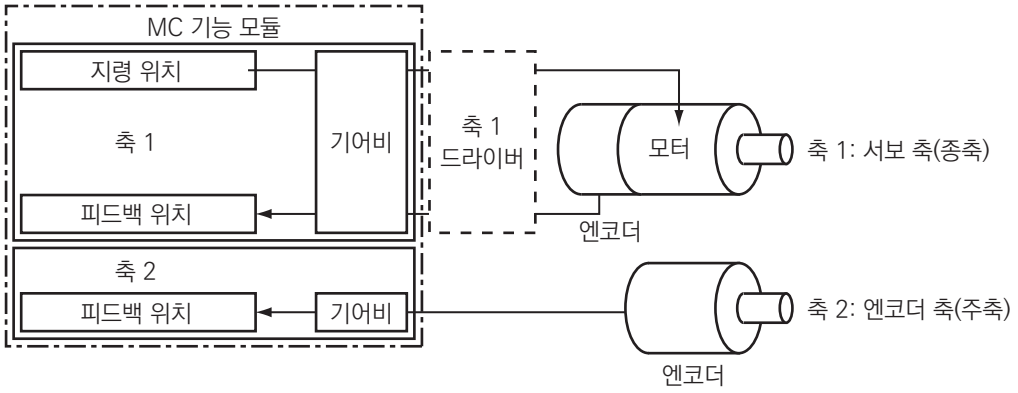


설정 항목	설정 내용	
	축 1	축 2
축 변수	Axis1	Axis2
축 번호	1	2
축 사용	사용 축	사용 축
축 종별	서보 축	서보 축
입력 디바이스/출력 디바이스	1	2
표시 단위	μm *1	μm
모터 1회전의 펄스 수	1,048,576	1,048,576
모터 1회전의 이동량	10,000	10,000
최고 속도	500,000 *2	500,000 *2
조그 최고 속도	50,000 *3	50,000 *3
최대 가속도	5,000,000 *4	5,000,000 *4
최대 감속도	5,000,000 *4	5,000,000 *4
소프트웨어 리미트 기능	지령 위치에 대해 유효하면 즉시 정지	지령 위치에 대해 유효하면 즉시 정지
정방향 소프트웨어 리미트	500,000 *5	500,000 *5
역방향 소프트웨어 리미트	0 *5	0 *5
카운트 모드	리니어 모드	리니어 모드

*1. 위치의 지령 단위는 1(μm)입니다.
*2. 최고 속도는 3,000r/min = 30m/min = 0.5m/s = 500,000μm/s로 합니다.
*3. 조그 최고 속도는 최고 속도의 10%인 0.05m/s = 50,000μm/s로 합니다.
*4. 최대 가속도 및 최대 감속도는 5m/s^{2}입니다. 최고 속도 3,000r/min까지의 가속 시간을 0.1s로 합니다.
*5. 장비 가동 범위의 안쪽 위치를 설정합니다.
정방향 소프트웨어 리미트 설정값은 50cm = 500,000μm로 합니다.

엔코더가 주축인 동기 제어

컨베이어에 설치된 축 2(엔코더)의 피드백 위치를 주축으로 하여 축 1의 서보 드라이버가 유한 길이 범위 내에서 동기하는 장비를 예로 들어 설명합니다.



설정 항목	설정 내용	
	축 1	축 2
축 변수	Axis1	Axis2
축 번호	1	2
축 사용	사용 축	사용 축
축 종별	서보 축	엔코더 축
입력 디바이스/출력 디바이스	1	2
표시 단위	μm *1	μm *1
모터 1회전의 펄스 수	1,048,576	1,048,576
모터 1회전의 이동량	10,000	10,000
최고 속도	500,000 *2	—
조그 최고 속도	50,000 *3	—
최대 가속도	5,000,000 *4	—
최대 감속도	5,000,000 *4	—
소프트웨어 리미트 기능	지령 위치에 대해 유효하면 즉시 정지	무효
정방향 소프트웨어 리미트	500,000 *5	—
역방향 소프트웨어 리미트	0 *5	—
카운트 모드	리니어 모드	로터리 모드
링 카운터 상한값	—	1,000,000 *6
링 카운터 하한값	—	0 *6

*1. 위치의 지령 단위는 1(μm)입니다.
*2. 최고 속도는 3,000r/min = 30m/min = 0.5m/s = 500,000μm/s로 합니다.
*3. 조그 최고 속도는 최고 속도의 10%인 0.05m/s = 50,000μm/s로 합니다.
*4. 최대 가속도 및 최대 감속도는 5m/s^{2}입니다. 최고 속도 3,000r/min까지의 가속 시간을 0.1s로 합니다.
*5. 장비 가동 범위의 안쪽 위치를 설정합니다.
정방향 소프트웨어 리미트 설정값은 50cm = 500,000μm로 합니다.
*6. 위치의 반복 범위를 0~1m(1,000,000μm)로 합니다.

참고

주축은 장비 구성에 따라 축 종별을 선택할 수 있습니다. 축 종별에는 서보 축, 가상 서보 축, 엔코더 축, 가상 엔코더 축 등 4종류가 있습니다.
이 예에서는 주축의 축 종별이 엔코더 축이므로 모션 제어 명령의 입력 변수 「ReferenceType (위치 종별 선택)」은 피드백 위치를 지정해 주십시오.

5-3 축 그룹 파라메터

축 그룹 파라메터는 MC 기능 모듈로 제어하는 축 그룹에 관하여 축 구성 및 최고 보간 속도, 축 그룹 정지 방법과 같은 축 그룹 동작에 관한 설정을 합니다.
축 그룹 파라메터는 축 그룹 수만큼 존재합니다. 축 그룹 수는 형식에 따라 다릅니다. 상세한 내용은 「1-4-2 성능 사양 (P.1-7)」을 참조해 주십시오.
또한, 각 축 그룹의 파라메터 설정은 동일하므로 여기에서는 1그룹에 해당하는 부분에 대해서만 설명합니다.

5-3-1 축 그룹 파라메터의 일람

축 그룹 파라메터는 Sysmac Studio를 사용하여 축 그룹별로 설정합니다.

구분	파라메터의 명칭	일시 변경 *1		변수 판독 *2	참조 페이지
		가능 여부	사용 명령		
축 그룹 기본 설정	축 그룹 번호			가능	P. 5-31
	모션 제어 *3			가능 *4	
	축 그룹 사용			가능	
	축 구성			가능	
	구성 축	가능 *5	MC_ChangeAxesInGroup	가능	
축 그룹 동작 설정	최고 보간 속도				P. 5-33
	최대 보간 가속도				
	최대 보간 감속도				
	보간 가감속 초과				
	보간 속도 경고값	가능	MC_Write		
	보간 가속도 경고값				
	보간 감속도 경고값				
	축 그룹 정지 방법				
	중심점 보정 허용률				

*1. 명령을 이용한 일시 변경의 가능 여부를 나타냅니다.
*2. 변수로 사용자 프로그램에서의 판독 가능 여부를 나타냅니다.
*3. NX701 CPU 유니트의 경우에 설정합니다.
*4. 유니트 버전 Ver.1.10 이상인 CPU 유니트와 Sysmac Studio Ver.1.12 이상의 조합으로 사용할 수 있습니다.
*5. 유니트 버전 Ver.1.01 이상인 CPU 유니트와 Sysmac Studio Ver.1.02 이상의 조합으로 사용할 수 있습니다.

축 그룹 파라메터의 설정 방법에 대해서는 「3-4 축 그룹 설정 순서 (P.3-24)」를 참조해 주십시오.
MC_Write(MC 설정 기록) 명령과 MC_ChangeAxesInGroup(축 그룹 구성 축 기록) 명령에 대한 상세한 내용은 『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.
모션 제어 시스템 변수에 대해서는 「6-6 모션 제어 시스템 변수 (P.6-17)」를 참조해 주십시오.

5-3-2 축 그룹 기본 설정

축 그룹의 사용 여부, 사용할 경우의 축 구성과 사용할 축을 설정합니다.

파라메터의 명칭	기능	설정 범위	초기값
축 그룹 번호	축 그룹의 논리 번호를 설정합니다. 시스템 정의 변수의 변수명 「_MC_GRP[0-63]」, 「_MC1_GRP[0-63]」, 「_MC2_GRP[0-63]」의 번호는 여기에서 지정한 번호가 됩니다.	—	—
모션 제어 *1	프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5) 중 1가지 태스크에 할당합니다. 1: 프라이머리 정주기 태스크 2: 정주기 태스크(실행 우선도 5)	1~2	1
축 그룹 사용	축 그룹의 사용 여부를 설정합니다. 작성하지 않거나 사용하지 않은 축 그룹에 모션 제어 명령을 실행하면 이상이 발생합니다. *2 0: 미작성 축 그룹 1: 미사용 축 그룹 2: 사용 축 그룹	0~2	0
축 구성	축 그룹의 축 구성을 설정합니다. 0: 2축 1: 3축 2: 4축	0~2	0
구성 축	축 그룹에 할당할 축 번호를 설정합니다. Sysmac Studio에서는 축 A0~A3에 대해 축 변수명을 선택합니다.	2~4축	0

*1. NX701 CPU 유니트의 경우에 설정합니다.
*2. 사용하지 않는 축을 포함하는 축 그룹에 MC_GroupEnable(축 그룹 유효) 명령을 실행하면 이상이 발생합니다.

축 구성

다음 표는 MC 기능 모듈에서 사용할 수 있는 축 구성입니다.
Sysmac Studio를 사용하여 실제 장비에 맞춰 축 구성을 설정합니다.

축 구성	설명
2축	2축의 축 구성입니다. 예를 들면 2축 직교 좌표계의 기계입니다.
3축	3축의 축 구성입니다. 예를 들면 3축 직교 좌표계의 기계입니다.
4축	4축의 축 구성입니다. 예를 들면 3축 직교 좌표계와 선단 툴의 회전 축이 있는 기계입니다.

구성 축

축 그룹에 속하는 축을 구성 축이라 부릅니다. 또한, 축 그룹 지령의 보간 명령인 경우 프로그램의 재이용성을 높이기 위해 축 번호(축 0~축 255) 대신에 논리 축(축 A0~축 A3)을 사용합니다.
구성 축은 축 그룹에 속하는 축의 축 번호와 논리 축 번호의 할당을 설정합니다.
논리 축은 서보 축 또는 가상 서보 축에서 선택할 수 있습니다.
Sysmac Studio를 이용하여 축 구성에서 선택한 축 수만큼 축 A0부터 순서대로 축을 설정합니다.
복수의 축 그룹을 작성할 때에도 축 그룹마다 축 A0부터 순서대로 축 번호를 설정합니다.
또한, 동일한 축 번호를 복수의 축 그룹에 설정할 수도 있습니다.

축 구성의 설정	구성 축 설정
2축	축 A0, 축 A1의 축 변수명(축 번호)을 설정합니다.
3축	축 A0, 축 A1, 축 A2의 축 변수명(축 번호)을 설정합니다.
4축	축 A0, 축 A1, 축 A2, 축 A3의 축 변수명(축 번호)을 설정합니다.

사용상의 주의

- 축 그룹의 구성 축은 모두 동일한 태스크에 할당해 주십시오.
- 동일한 축 번호를 복수의 축 그룹에 설정한 경우 동일한 축 번호가 설정된 축 그룹을 동시에 유효하게 할 수는 없습니다.
- [Control Function] 설정이 [Single-axis position control only]인 축은 축 그룹의 구성 축에 할당되지 않습니다.

버전 관련 정보

유니트 버전 Ver.1.04 이상인 CPU 유니트와 Sysmac Studio Ver.1.05 이상의 조합인 경우 서보 축 또는 가상 서보 축이면 사용 축뿐만 아니라 사용하지 않는 축(사용 축으로 변환 가능)으로 설정된 축도 축 그룹으로 설정할 수 있습니다.

● 구성 축 설정 예

- 축 그룹에 축 번호 1, 2, 5, 8의 4축을 할당할 경우에는 아래와 같이 설정합니다.

논리 축	축 번호
축 A0	축 1
축 A1	축 2
축 A2	축 5
축 A3	축 8

- 축 그룹에 축 번호 1, 8, 2의 3축을 할당할 경우에는 아래와 같이 설정합니다.

논리 축	축 번호
축 A0	축 1
축 A1	축 8
축 A2	축 2
축 A3	(없음)

5-3-3 축 그룹 동작 설정

제어하는 장비의 사양에 맞춰 최고 보간 속도 및 축 그룹 정지 방법 등 축 그룹의 동작에 대해 설정합니다.
NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

파라메터의 명칭	기능	설정 범위	초기값
최고 보간 속도	궤적의 최고 보간 속도를 설정합니다. 「0」을 설정하면 보간 속도의 제한이 없어집니다. 축 그룹 동작 명령으로 최고 보간 속도를 초과하는 보간 목표 속도가 지정된 경우에는 최고 보간 속도로 동작합니다. (단위: 지령 단위/s)	배정밀도 실수형의 양수, 0	800000000
최대 보간 가속도	궤적의 최대 보간 가속도를 설정합니다. 「0」을 설정하면 보간 가속도의 제한이 없어집니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
최대 보간 감속도	궤적의 최대 보간 감속도를 설정합니다. 「0」을 설정하면 보간 감속도의 제한이 없어집니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
보간 가감속 초과	축 그룹의 가감속 제어에서 목표 위치에 대한 정지를 우선한 결과 감속이 초과되어 최대 보간 가감 속도를 초과한 경우의 동작을 지정합니다. 0: 가감 속도를 빠르게 사용한다(블렌딩을 버퍼드로 변환한다). *1 1: 가감 속도를 빠르게 사용한다. 2: 이상으로 인식하여 정지한다. *2	0~2	0
보간 속도 경고값	보간 속도 경고를 출력하기 위한 최고 보간 속도에 대한 비율을 설정합니다. 「0」을 설정하면 보간 속도 경고는 출력하지 않습니다. (단위: %)	0~100	0
보간 가속도 경고값	보간 가속도 경고를 출력하기 위한 최고 보간 가속도에 대한 비율을 설정합니다. 「0」을 설정하면 보간 가속도 경고는 출력하지 않습니다. (단위: %)	0~100	0
보간 감속도 경고값	보간 감속도 경고를 출력하기 위한 최고 보간 감속도에 대한 비율을 설정합니다. 「0」을 설정하면 보간 감속도 경고는 출력하지 않습니다. (단위: %)	0~100	0
축 그룹 정지 방법	다축 협조 동작 중에 있는 축에서 즉시 정지하는 이상이 발생한 경우 이상이 발생하지 않는 구성 축의 정지 방법을 설정합니다. 0: 즉시 정지 1: 각 축을 감속 정지(각 축의 최대 감속도) 3: 즉시 정지하고 동시에 서보 OFF를 실행	0, 1, 3	0
중심점 보정 허용률	원호 보간 명령의 중심점 지정 방식에서 시작점과 중심점, 종료점과 중심점의 거리가 다를 경우에는 거리가 같아지도록 보정합니다. 그 보정의 허용 범위를 반경에 대한 비율로 설정합니다. 비율은 0.1% 이상으로 설정합니다. 「0」을 설정하면 오차 체크를 실행하지 않습니다.	단정밀도 부동소수점형에서 0~100	0

*1. 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 블렌딩을 버퍼드로 변환하지 않습니다. 상세한 내용은 🕮「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.
*2. 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 블렌딩 동작의 경우 이상으로 정지하지 않습니다. 상세한 내용은 🕮「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.

5-3-4 축 그룹의 유효

사용자 프로그램에서 축 그룹에 대한 동작 명령을 유효화하려면 MC_GroupEnable(축 그룹 유효) 명령으로 유효화할 축 그룹 번호를 지정합니다.
유효하지 않은 축 그룹에 모션 제어 명령을 기동하면 명령에서 이상이 발생합니다.
또한, 동시에 복수의 축 그룹을 유효하게 할 수 있으나 동일한 축을 포함하는 축 그룹을 유효화하면 명령에서 이상이 발생합니다.

축 1개를 작업 공정마다 다른 축 그룹에서 동작시키고 싶을 때는 그 축이 포함된 복수의 축 그룹을 작성해 둡니다. 그 후 MC_GroupEnable(축 그룹 유효) 명령과 MC_GroupDisable(축 그룹 무효) 명령을 순차적으로 변환하면 실현 가능합니다.

다축에서 동작하는 도중에 MC_GroupDisable(축 그룹 무효) 명령을 기동하면 감속 정지합니다.

6

모션 제어 프로그램

이 장에서는 모션 제어 프로그램의 사양과 프로그램을 작성하는 조작 순서에 대해 설명합니다.

6-1 개요 6-2
6-2 모션 제어 명령 6-5
6-2-1 PLCopen® 모션 제어용 펑션 블록 6-5
6-2-2 모션 제어 기능 모듈의 모션 제어 명령 6-5
6-3 상태 변이 6-6
6-3-1 모션 제어 기능 모듈의 상태 6-6
6-3-2 축 상태 6-6
6-3-3 축 그룹의 상태 6-8
6-4 모션 제어 명령의 기동과 상태 6-9
6-4-1 기동에 대한 공통 규칙 6-9
6-4-2 기동의 타이밍 차트 6-11
6-4-3 모션 제어 명령 재기동 시의 타이밍 차트 6-13
6-4-4 모션 제어 명령 다중 기동 시의 타이밍 차트 6-14
6-5 위치의 취급 6-15
6-5-1 위치의 종류 6-15
6-5-2 각 축 종별에서 사용 가능한 위치의 종류 6-16
6-6 모션 제어 시스템 변수 6-17
6-6-1 모션 제어 시스템 변수의 개요 6-17
6-6-2 모션 제어 시스템 변수의 구조 6-19
6-6-3 모션 제어 시스템 변수의 일람 6-20
6-7 캠 테이블과 캠 데이터 변수 6-31
6-8 모션 제어 프로그램의 작성 방법 6-35
6-9 캠 테이블의 작성 방법 6-37

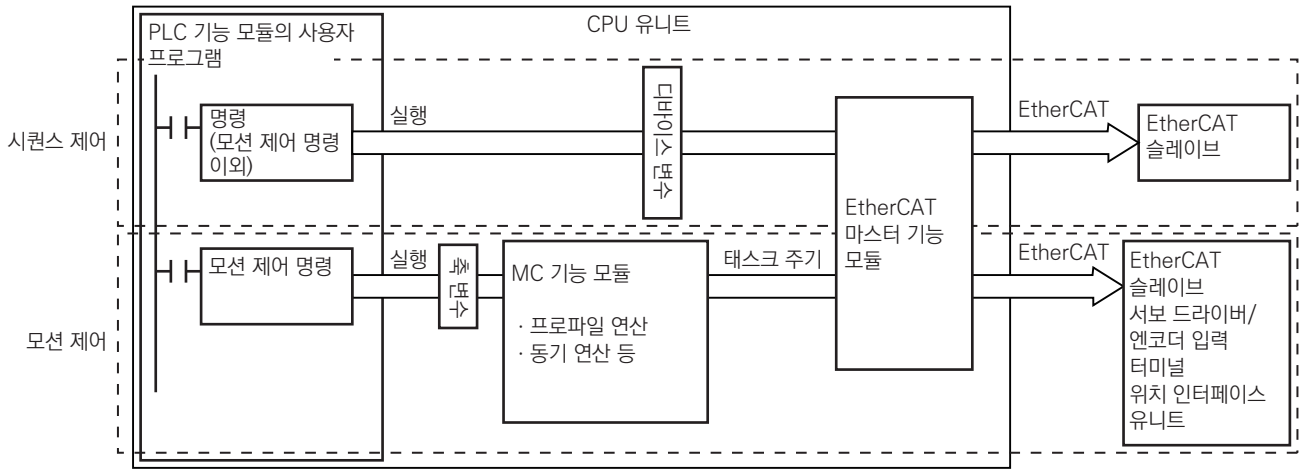
6-1 개요

NJ/NX 시리즈 CPU 유니트에서는 시퀀스 제어 및 모션 제어를 실행할 수 있습니다.
사용자 프로그램에 모션 제어 명령을 기술하면 EtherCAT 슬레이브 서보 드라이버와 NX 시리즈 위치 인터페이스 유니트 등을 대상으로 모션 제어를 실행합니다.
본 매뉴얼에서는 모션 제어 명령을 기술한 프로그램을 모션 제어 프로그램이라 부릅니다.
EtherCAT 슬레이브 서보 드라이버나 NX 시리즈 위치 인터페이스 유니트에는 축 변수를 할당해야 합니다.
축 변수에 할당하지 않을 때는 범용 슬레이브와 같이 I/O 디바이스 변수에 할당합니다.
모션 제어 명령은 프라이머리 정주기 태스크 또는 정주기 태스크(실행 우선도 5와 16)에서 사용할 수 있습니다.

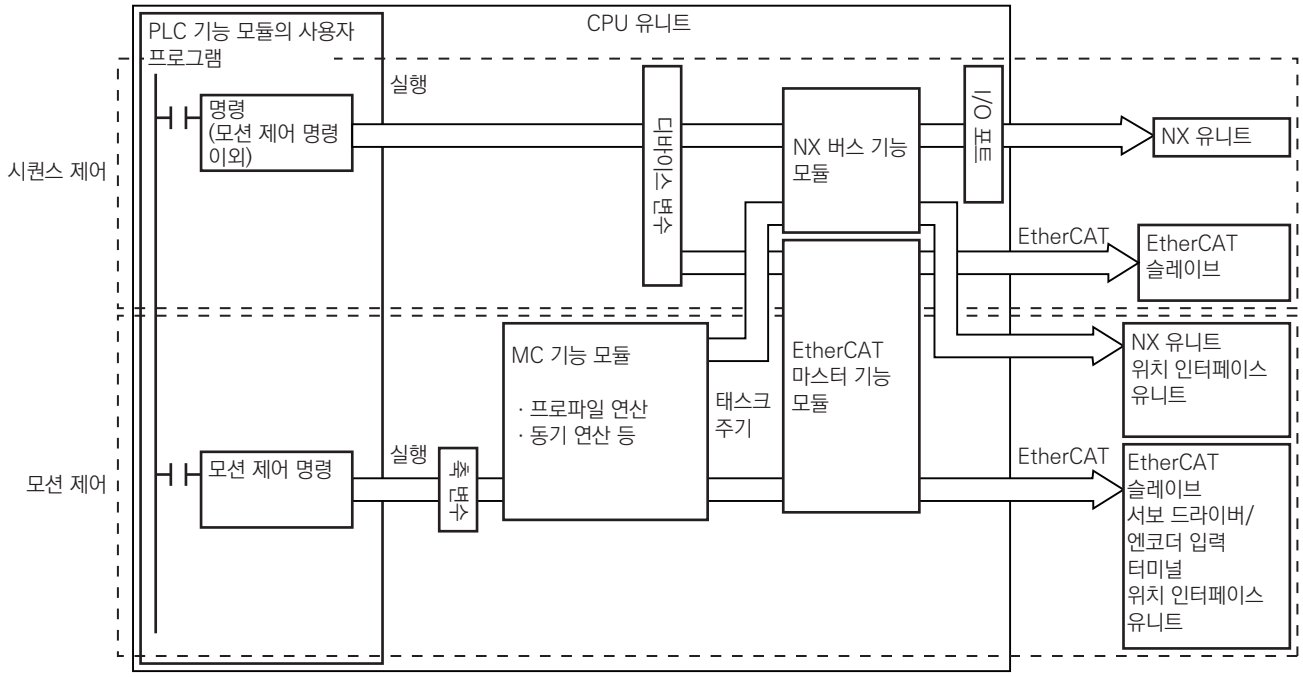
버전 관련 정보

Sysmac Studio Ver.1.09 이상을 사용하면 축 변수에 할당한 슬레이브나 유니트의 I/O 포트에 디바이스 변수를 할당할 수 있습니다.
상세한 내용은 「2-4-2 EtherCAT 마스터 기능과 모션 제어 기능 모듈의 관계 (P.2-20)」를 참조해 주십시오.

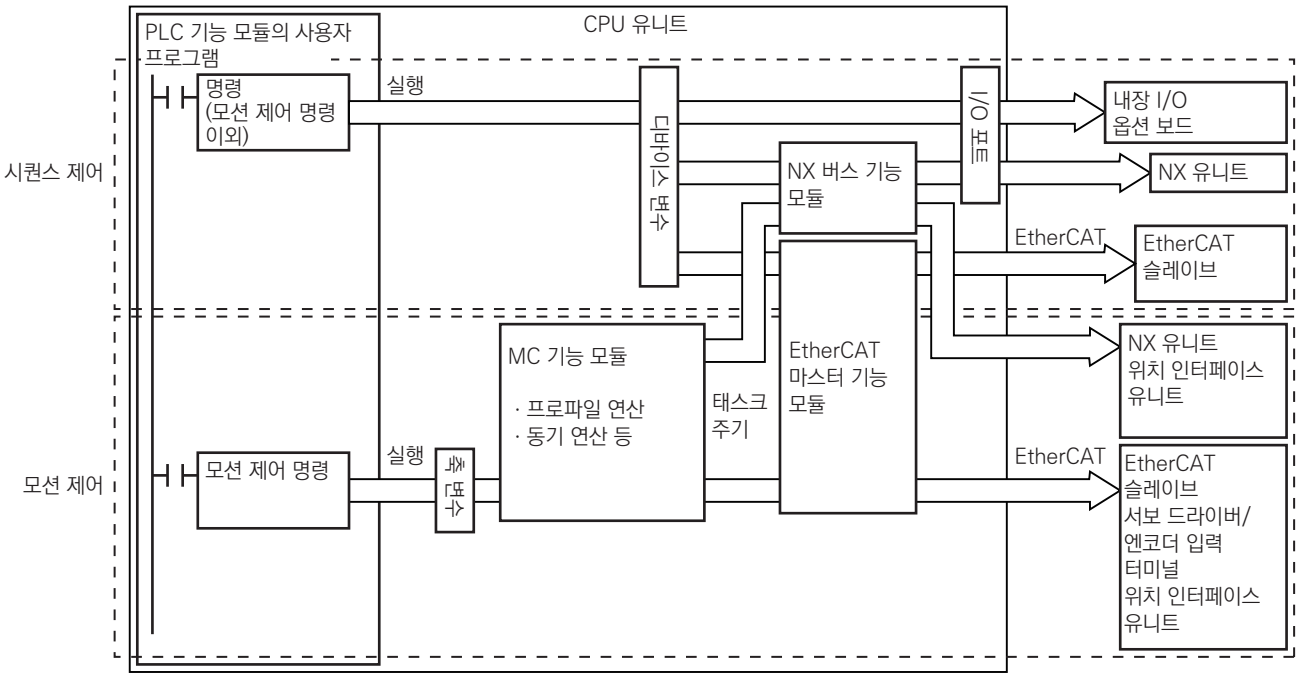
● NX701 CPU 유니트의 경우



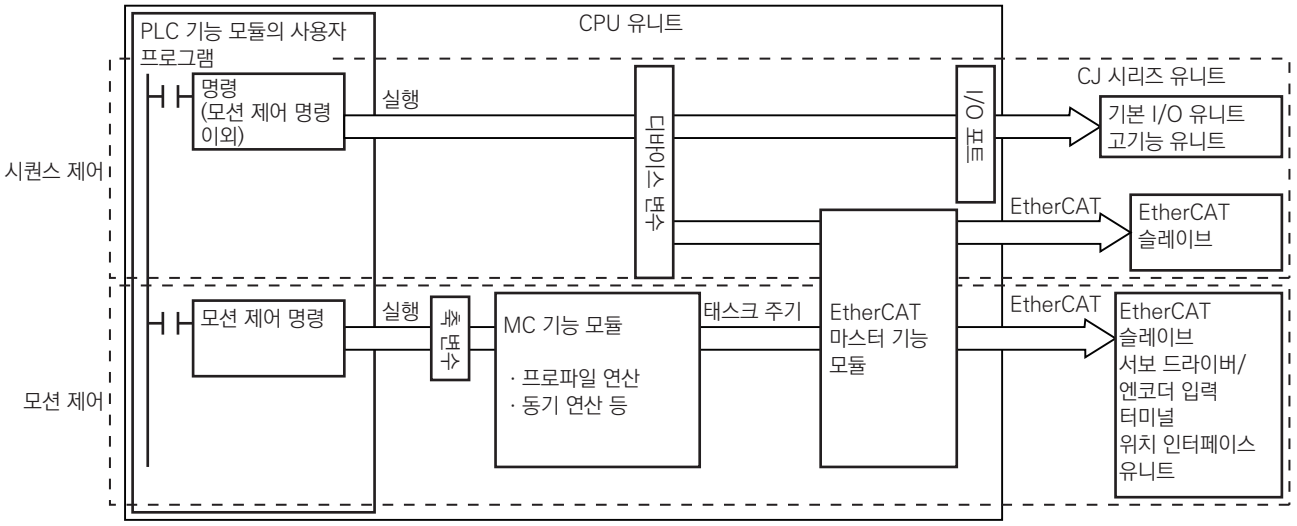
● NX102 CPU 유니트의 경우



● NX1P2 CPU 유니트의 경우



● NJ 시리즈 CPU 유니트의 경우



6-2 모션 제어 명령

NJ/NX 시리즈의 사용자 프로그램에서 모션 제어 기능을 실행하려면 펑션 블록(이하, FB로 생략하는 경우가 있습니다)으로 정의된 모션 제어 명령을 사용합니다.
MC 기능 모듈의 모션 제어 명령은 PLCopen®의 모션 제어용 펑션 블록 기술 사양을 기본으로 합니다.
모션 제어 명령은 PLCopen®에서 정의된 명령과 MC 기능 모듈의 독자적인 명령 등 2가지가 있습니다.
여기에서는 PLCopen® 모션 제어용 FB의 개요와 MC 기능 모듈의 사양에 대해 설명합니다.

NJ/NX 시리즈의 펑션 블록(FB) 기본 사항에 대해서는 🕮『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

유니트 버전 Ver.1.05 이상인 CPU 유니트와 Sysmac Studio Ver.1.06 이상의 조합으로 NX 시리즈 위치 인터페이스 유니트를 사용할 수 있습니다.
NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

6-2-1 PLCopen® 모션 제어용 펑션 블록

PLCopen®에서는 모션 제어용 펑션 블록을 표준화하여 IEC 61131-3(JIS B 3503) 규격 언어를 이용한 프로그램 인터페이스를 정의하고 있습니다.
단축의 위치 결정이나 전자 캠 기능, 다축에서의 보간 제어 기능과 명령을 기동하기 위한 기본적인 순서에 대해서도 정의합니다.

PLCopen® 모션 제어용 펑션 블록을 이용하면 하드웨어의 의존성을 줄이고 사용자 프로그램의 재이용성을 높일 수 있습니다.
또한, 트레이닝이나 서포트에 드는 비용을 줄일 수도 있습니다.

참고

PLCopen®란
PLCopen®는 유럽에 본부를 둔 IEC 61131-3의 보급 단체로, 전 세계에 회원을 둔 글로벌 조직입니다.
IEC 61131-3이란 PLC 프로그래밍의 국제 표준 규격입니다.
PLCopen® Japan은 일본 시장의 프로모션 위원회로 일본 시장에 관심이 있는 회원으로 구성되어 있습니다.
- PLCopen® Japan 웹 사이트, http://www.plcopen-japan.jp/
- PLCopen® 유럽 본부 웹 사이트, http://www.plcopen.org/

6-2-2 모션 제어 기능 모듈의 모션 제어 명령

모션 제어 명령은 아래와 같이 3가지 종류가 있습니다.

종류	개요
공통 지령	MC 기능 모듈 공통 명령
축 지령	MC 기능 모듈이 단축 제어를 실행하기 위한 명령
축 그룹 지령	MC 기능 모듈이 다축 협조 제어를 실행하기 위한 명령

MC 기능 모듈에서 사용할 수 있는 명령의 일람과 상세한 내용에 대해서는 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

6-3 상태 변이

MC 기능 모듈의 축과 축 그룹의 상태 및 명령 기동에 의한 상태 변이는 PLCopen®의 모션 제어용 펑션 블록 기술 사양을 기본으로 합니다.
여기에서는 MC 기능 모듈 전체를 포함하여 각각의 상태와 상태 변이에 대해 설명합니다.

6-3-1 모션 제어 기능 모듈의 상태

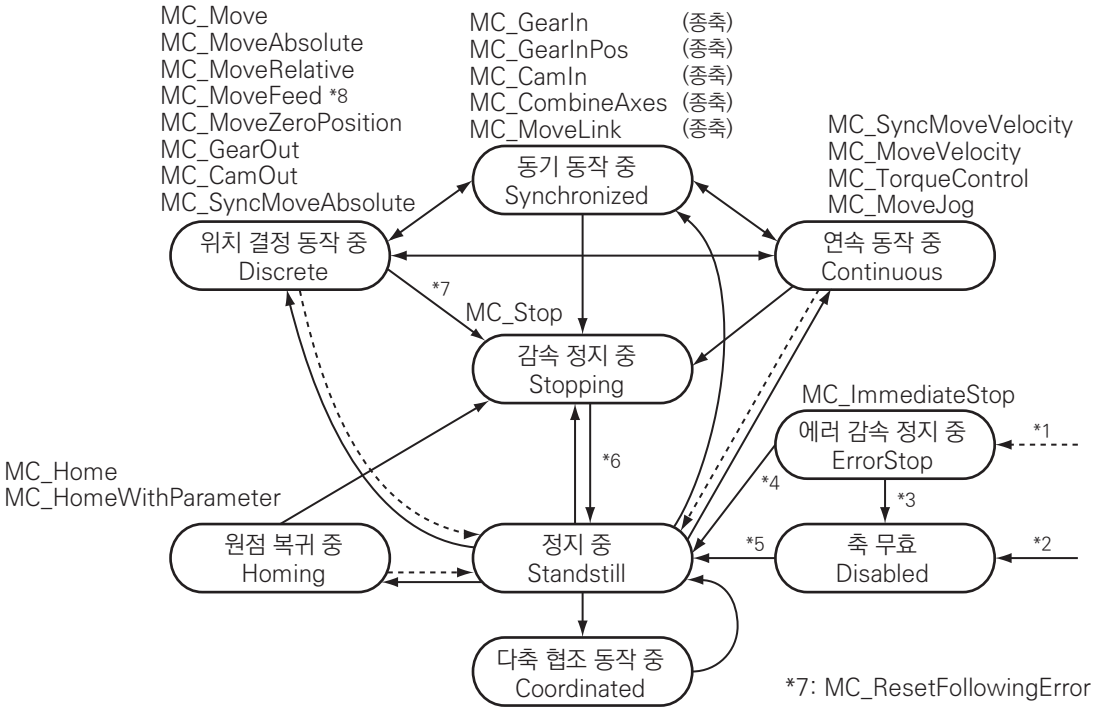
MC 기능 모듈 전체의 상태를 아래 표에 표시합니다.

상태명	정의
MC 기능 모듈 운전 중 *1	모션 제어 명령이 유효한 상태입니다. 사용자 프로그램의 모션 제어 명령을 해석하고 모션 제어를 실행합니다. CPU 유니트의 동작 모드와 관계없이 「MC 기능 모듈 운전 중」으로 설정할 수 있습니다.
MC 기능 모듈 시운전 중 *2	Sysmac Studio에서 시운전을 하고 있는 상태입니다.
캠 테이블 파일 저장 중 *3	캠 테이블 파일 저장 중 또는 대기 중 상태입니다.
캠 테이블 생성 실행 중 *4	캠 테이블을 생성 중인 상태입니다. *5

*1. MC 공통 변수인 「_MC_COM.Status.RunMode」로 모니터할 수 있습니다.
*2. MC 공통 변수인 「_MC_COM.Status.TestMode」로 모니터할 수 있습니다.
*3. MC 공통 변수인 「_MC_COM.Status.CamTabeleBusy」로 모니터할 수 있습니다.
*4. MC 공통 변수인 「_MC_COM.Status.GenerateCamBusy」로 모니터할 수 있습니다.
*5. CPU 유니트의 전원을 OFF할 때는 캠 테이블을 생성하고 있지 않은지 확인해 주십시오. 캠 테이블을 생성하고 있는 도중에 CPU 유니트 전원을 OFF하면 캠 테이블이 올바르게 생성되지 않습니다.

6-3-2 축 상태

축에 대한 모션 제어 명령이 기동되었을 때의 축 동작은 아래 그림과 같습니다.
원칙적으로 모션 제어 명령은 차례로 실행되고 축은 아래 표에 기술된 상태 중 하나가 됩니다.



*1. 「다축 협조 동작 중」이 아닌 상태에서도 축 이상 발생으로 변이합니다.

*2. 어떠한 상태에서도 MC_Power 명령의 「Status(운전 가능)」 출력이 「FALSE」이고 축 이상이 발생하지 않았을 때 변이합니다. (서보 OFF)
*3. 서보 OFF 상태에서 MC_Reset 명령 또는 ResetMCError 명령에 의한 이상이 해제되면 변이합니다.
*4. 서보 ON 상태에서 MC_Reset 명령 또는 ResetMCError 명령에 의한 이상이 해제되면 변이합니다.
*5. MC_Power 명령의 「Enable(유효)」 입력을 「TRUE」로 하고, MC_Power 명령의 「Status(운전 가능)」 출력이 「TRUE」로 변이합니다. (서보 ON)
*6. MC_Stop 명령의 「Done(완료)」 출력이 「TRUE」이고 MC_Stop 명령의 「Execute(기동)」 입력이 「FALSE」일 때 변이합니다.
*7. MC_ResetFollowingError 명령에 의해 감속 정지 중으로 바뀝니다.
*8. MC_MoveFeed 명령의 입력 변수 「MoveMode(이동 방법 선택)」에 「2: _mcVelocity(속도 제어)」를 설정한 경우 트리거 입력이 검출될 때까지는 「연속 동작 중」이 됩니다.

상태명		정의
서보 OFF		축이 서보 OFF된 상태입니다. 이 상태로 변이하면 명령 다중 기동의 대기 상태를 해제합니다.
	축 무효	축이 서보 OFF이고 정지 중에 기동 준비가 완료된 상태입니다.
	에러 감속 정지 중 *1	축이 서보 OFF이고 정지 중에 축 이상이 발생하고 있는 상태입니다.
서보 ON		축이 서보 ON된 상태입니다.
	정지 중	축이 서보 ON이고 정지하고 있는 상태입니다.
	위치 결정 동작 중	목표 위치가 지정된 위치 결정을 실행하고 있는 상태입니다. 위치 결정 완료 대기 상태 및 위치 결정 동작 중에 오버라이드값을 「0」으로 설정하여 속도 「0」이 된 상태도 포함합니다.
	연속 동작 중	목표 위치가 지정되지 않은 연속 동작을 실행하고 있는 상태입니다. 속도 제어 또는 토크 제어일 때 이 상태가 됩니다. 목표 속도를 「0」으로 설정하여 속도 「0」이 된 상태, 연속 동작 중에 오버라이드값을 「0」으로 설정하여 속도 「0」이 된 상태도 포함합니다.
	동기 동작 중	동기 제어 지령에 의해 축을 동기 제어하고 있는 상태입니다. 동기 제어 명령 변환 후의 동기 대기 상태도 포함합니다.
	감속 정지 중	MC_Stop(강제 정지) 명령 또는 MC_TouchProbe(외부 래치 유효) 명령에 의해 축이 정지하기 전까지인 상태입니다. MC_Stop(강제 정지) 명령으로 정지 후 Execute(기동)가 TRUE인 상태도 포함합니다. 이 상태에서는 축 지령을 이용한 동작을 실행할 수 없습니다. 실행하면 명령은 CommandAborted(실행 중단)가 TRUE가 됩니다.
	에러 감속 정지 중 *1	축이 서보 ON이고 이상이 발생한 상태입니다. MC_ImmediateStop(즉 정지) 명령이 실행된 상태 및 축에 이상이 발생하여 축이 감속 중인 상태도 포함합니다. 이 상태에서는 축 동작 지령을 실행할 수 없습니다. 실행하면 명령은 중단 (CommandAborted = 1) 상태가 됩니다.
	원점 복귀 중	MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령에 의해 원점을 찾고 있는 상태입니다.
	다축 협조 동작 중	축이 축 그룹 지령의 명령에 의해 축 그룹이 유효해진 상태입니다. 축 그룹의 상태의 「그룹 동작 중」, 「감속 정지 중」 또는 「에러 정지 중」으로 동작 중인 상태입니다.

*1. 「에러 감속 정지 중」에는 축이 서보 OFF일 때와 서보 ON일 때 등 2가지 경우가 있습니다.

(주) 축 상태는 축 변수인 「_MC_AX[0-255].Status」의 각 멤버 변수로 모니터할 수 있습니다.

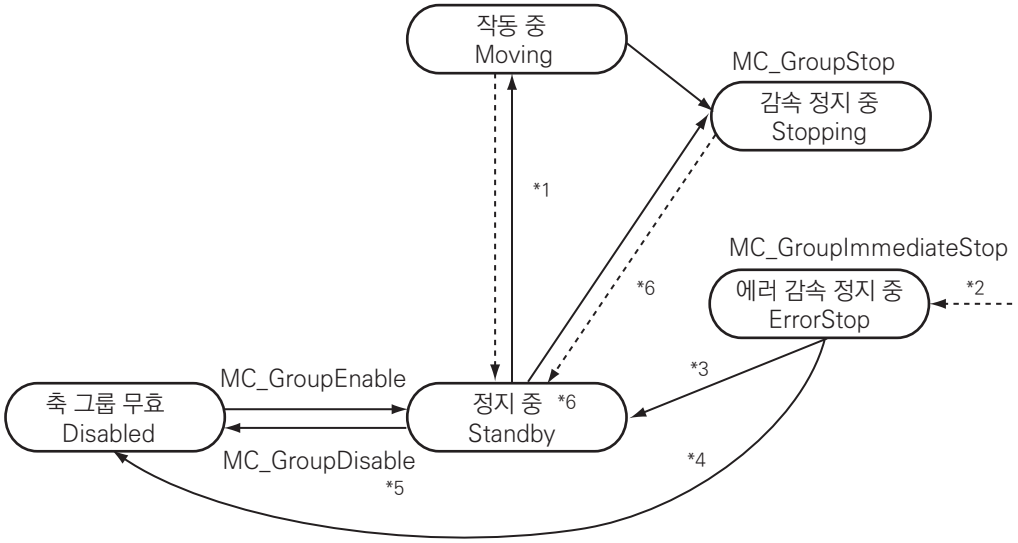
버전 관련 정보

NX701 CPU 유니트의 경우, _MC_AX[*]로 시작되는 변수명은 _MC1_AX[*], _MC2_AX[*]인 경우가 있습니다.

NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

6-3-3 축 그룹의 상태

축 그룹에 대한 모션 제어 명령이 기동되었을 때의 축 그룹 동작은 아래 그림과 같습니다.



*1. 모든 축 그룹 동작 명령으로 변이합니다.
*2. 모든 상태에서 변이합니다. 축 그룹이 무효인 상태에서도 이상이 발생하면 변이합니다.
*3. 축 그룹이 유효한 경우에 MC_GroupReset 명령 또는 ResetMCError 명령으로 변이합니다.
*4. 축 그룹이 무효인 경우에 MC_GroupReset 명령 또는 ResetMCError 명령으로 변이합니다.
*5. 에러 감속 정지 중 상태일 때 MC_GroupDisable 명령을 기동해도 같은 상태에 머뭅니다.
*6. MC_GroupStop 명령의 출력 변수「Done」= TRUE, 입력 변수「Execute」= FALSE일 때 변이합니다.
*7. 정지 중(Standby) 상태에서 아래의 모든 조건을 만족할 때 기동 준비 완료(Ready)가 됩니다.
모든 구성 축이 서보 ON하고 있다.
모든 구성 축에서 MC_Stop 명령이 실행되지 않는다.
모든 구성 축이 원점을 확정하고 있다.

상태명		정의
축 그룹 무효		축 그룹이 무효한 상태입니다. 본 상태로 변이하면 명령 다중 기동의 대기 상태를 해제합니다.
	에러 감속 정지 중 *1	축 그룹이 무효인 상태에서 축 그룹의 이상이 발생하고 있는 상태입니다.
축 그룹 유효		축 그룹이 유효한 상태입니다.
	정지 중	축 그룹 지령의 동작 명령을 실행하고 있지 않은 상태입니다. (축 그룹 구성 축의 서보 ON/OFF 상태에 따르지 않습니다.)
	동작 중	축 그룹 지령의 동작 명령에 의해 목표 위치가 지정된 위치 결정을 실행하고 있는 상태입니다. 인포지션 대기 상태 및 그룹 동작 중에 오버라이드값을「0」으로 설정하여 속도「0」이 된 상태도 포함됩니다.
	감속 정지 중	MC_GroupStop(축 그룹 강제 정지) 명령이 실행되고 있는 상태입니다. MC_GroupStop(축 그룹 강제 정지) 명령으로 정지 후 Execute(기동)가 TRUE인 상태도 포함합니다. 이 상태에서는 축 그룹 지령을 이용한 동작을 실행할 수 없습니다. 실행하면 명령은 CommandAborted(실행 중단)가 TRUE가 됩니다.
	에러 감속 정지 중 *1	축 그룹에서 이상이 발생하고 있는 상태입니다. MC_GroupImmediateStop(축 그룹 즉시 정지) 명령이 실행된 상태 및 축 그룹에 이상이 발생하여 축 그룹이 감속 중인 상태도 본 상태에 포함합니다. 이 상태에서는 다축 협조 지령의 실행을 금지합니다. 실행 시에는 CommandAborted(실행 중단)가 TRUE가 됩니다.

*1.「에러 감속 정지 중」에는 축 그룹 무효일 때와 축 그룹 유효일 때 등 2가지 경우가 있습니다.

(주) 축 그룹의 상태는 축 그룹 변수인「_MC_GRP[0-63].Status」,「_MC1_GRP[0-63].Status」,「_MC2_GRP[0-63].Status」의 각 멤버 변수로 모니터할 수 있습니다.

6-4 모션 제어 명령의 기동과 상태

MC 기능 모듈의 공통 규칙으로 모션 제어 명령을 기동하는 변수나 명령의 실행 상태를 나타내는 변수를 정의합니다.
모션 제어 명령의 기능을 기동하기 위한 입력 변수에는 「Execute」와 「Enable」 등 2종류가 있습니다.
명령의 실행 상태를 나타내는 출력 변수에는 「Busy」, 「Done」, 「CommandAborted」, 「Error」 등이 있습니다.

6-4-1 기동에 대한 공통 규칙

MC 기능 모듈의 공통 규칙을 아래 표에 기재했습니다.
기동 예를 「6-4-2 기동의 타이밍 차트 (P.6-11)」에 기재했습니다.
함께 참조해 주십시오.

항목	원칙
출력의 배타성	출력 변수 Busy(실행 중), Done(완료), Error(에러), CommandAborted(실행 중단)는 상호 배타적으로 여러 개가 동시에 TRUE가 되는 경우는 없습니다. 마찬가지로 출력 변수 Active(제어 중), Done(완료), Error(에러), CommandAborted(실행 중단)에 대해서도 여러 개가 동시에 TRUE가 되는 경우는 없습니다. Busy(실행 중)와 Active(제어 중)는 동시에 TRUE가 되는 경우가 있습니다.
출력 상태	출력 변수 Done(완료), InGear(기어비 도달), InSync(동기 중), InVelocity(목표 속도 도달) 및 CommandAborted(실행 중단)는 입력 변수 Execute(기동)의 하강 엣지에서 FALSE가 됩니다. Execute(기동)의 하강 엣지는 실제 모션 제어 명령의 실행을 정지시키지 않습니다. 명령 인스턴스가 실행 종료하기 전에 Execute(기동)가 FALSE로 되어 있어도 명령 인스턴스의 상태가 바뀌면 적어도 1주기 동안 대응하는 출력 변수가 TRUE가 됩니다. 출력 변수 Error(에러)와 ErrorID(에러 코드)는 MC_Reset(축 에러 리셋) 명령, MC_GroupReset(축 그룹 에러 리셋) 명령 혹은 ResetMCError(전체 에러 리셋) 명령을 실행할 때까지 제로 클리어되지 않습니다. *1 모션 제어 명령이 실행 중에 같은 명령 인스턴스의 Execute(기동)를 다시 FALSE→TRUE로 상승시켜도(명령 재기동), CommandAborted(실행 중단)는 TRUE가 되지 않습니다.
입력 파라메터	입력 변수 Execute(기동)에서 기동하는 모션 제어 명령의 입력 파라메터는 Execute(기동)의 상승 엣지 발생 시의 값이 사용됩니다. 입력 변수 Enable(유효)에서 기동하는 모션 제어 명령의 입력 파라메터는 Enable(유효)이 TRUE인 동안 각 주기의 값이 사용됩니다.
입력 파라메터의 생략	명령 인스턴스의 입력 파라메터를 생략한 경우 초기값이 적용됩니다. *2
「Position(목표 위치)」과 「Distance(이동 거리)」	입력 변수 Position(목표 위치)은 좌표계에서 정의된 값입니다. 입력 변수 Distance(이동 거리)는 상대적인 길이로 2개의 위치 간 차이입니다.
부호 규칙	입력 변수 Acceleration(가속도), Deceleration(감속도), Jerk(저크)는 +값 또는 「0」입니다. Position(목표 위치), Distance(이동 거리), Velocity(목표 속도)는 +값, -값 또는 「0」도 될 수 있습니다.
이상 처리 동작	명령 인스턴스 실행 중에 발생하는 이상을 나타내는 출력 변수는 2가지가 있습니다. 이러한 출력은 아래와 같이 정의됩니다. ・Error(에러): 출력 변수 「Error」의 상승 엣지는 명령 인스턴스 실행 중에 이상이 발생했음을 나타냅니다. ・ErrorID(에러 코드): 이상 원인을 나타내는 코드입니다. 출력 변수 Done(완료), InVelocity(목표 속도 도달), InGear(기어비 도달), InSync(동기 중)은 정상 완료 또는 정상으로 동작하고 있음을 의미하므로 출력 변수 Error(에러)에 대해 배타적으로 됩니다.

항목	원칙
이상 처리 동작 (계속)	이상의 종류는 아래와 같이 2가지가 있습니다. ・명령 인스턴스 이상(예: 파라메터 범위 외, 상태 변이의 조건 위반) ・축 이상(예: 위치 편차 초과, 서보 드라이버 측 이상) 명령 인스턴스의 이상에서는 축 이상을 발생시키지 않고 축을 정지시키는 경우가 있습니다.
출력 변수 「Done(완료)」의 동작	출력 변수 Done(완료), InGear(기어비 도달), InSync(동기 중)은 명령 동작이 정상 완료 또는 지령한 상태에 도달한 경우에 TRUE가 됩니다. 목표 위치를 지정하는 동작 명령의 경우 축 파라메터의 「인포지션 체크 시간」 설정에 따라 TRUE가 되는 타이밍이 다릅니다. ・「0 이외」인 경우 위치 결정이 완료된 다음 주기에서 TRUE가 됩니다. ・「0」인 경우 지령 종료(할당 완료) 다음 주기에서 TRUE가 됩니다. 동일한 축에서 작동하는 복수의 명령의 경우, 축 동작이 최종 목표 위치에 도달하기 전에 다른 동작 명령으로 인터럽트되면 최초 명령의 출력 변수 「Done(완료)」은 TRUE가 되지 않습니다.
출력 변수 「CommandAborted (실행 중단)」의 동작	출력 변수 CommandAborted(실행 중단)는 지령된 동작이 다른 동작 명령으로 인터럽트 되었을 때 TRUE가 됩니다. 본 MC 기능 모듈에서는 모션 제어 명령 기동 시에 대상 축이나 축 그룹이 이상 발생 중 또는 감속 정지 중일 때 TRUE가 됩니다. CommandAborted(실행 중단)가 TRUE가 되면 다른 출력 변수는 FALSE가 됩니다.
유력 범위 외의 입력	입력 변수값이 유효 범위 외에서 기동된 경우 명령 인스턴스는 이상을 발생합니다.
출력 변수 「Busy(실행 중)」의 동작	출력 변수 Busy(실행 중)는 명령 인스턴스가 실행 중임을 나타냅니다. Busy(실행 중)는 입력 변수 Execute(기동) 상승 엣지에서 TRUE가 되고, 출력 변수 Done(완료), CommandAborted(실행 중단), Error(에러)가 TRUE가 되었을 때 FALSE가 됩니다. 위 출력 변수는 언제 바뀔지 모르므로 Busy(실행 중)가 TRUE인 동안에는 출력 변수의 변화를 감시하기 위해 명령 인스턴스가 주기마다 실행*3되도록 프로그램해 주십시오. 하나의 축 또는 하나의 축 그룹에서 복수의 명령 인스턴스 Busy(실행 중)가 TRUE로 될 수 있으나 출력 변수 Active(제어 중)가 TRUE가 되는 명령 인스턴스는 한 번에 1개뿐입니다. 단, MC_Phasing(주축 상대값 위상 보정) 명령은 예외입니다.
출력 변수 「Active(제어 중)」	출력 변수 Active(제어 중)는 명령 인스턴스가 해당하는 축의 제어권을 획득했을 때 TRUE가 됩니다. 출력 변수 Active(제어 중)는 Busy(실행 중)보다 늦게 바뀌는 경우가 있습니다. *4

*1. 이 사양은 PLCopen®와는 다릅니다. PLCopen®에서는 Execute(기동)의 하강 엣지에서 Error(에러)와 ErrorID(에러 코드)가 「0」이 됩니다.
또한, Error(에러)가 TRUE인 동안에는 모션 제어 명령은 기동하지 않습니다. 이상을 해제했을 때 Execute(기동)가 TRUE라도 상승 엣지가 아니므로 명령은 기동하지 않습니다. Enable형 모션 제어 명령은 이상을 해제했을 때 Enable(유효)이 TRUE이면 기동합니다.

*2. 래더도 언어로 프로그램을 작성할 경우에는 입력 변수 Execute(기동) 또는 Enable(유효)과 좌측 모선과의 사이에 접점을 배치해 주십시오. 접점을 배치하지 않고 좌측 모선과 직접 접속하면 빌드 실시 시에 이상이 발생합니다.
예약(Reserved)으로 취급하는 입력 변수에 대해서는 입력 파라메터를 생략하거나 초기값을 설정해 주십시오.

*3. ST 구조 명령에서 조건식이나 선택값이 일치하지 않는 경우 해당 문장에 기술한 명령은 실행되지 않습니다. 상세한 내용에 대해서는 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

*4. 출력 변수의 타이밍 차트에 대해서는 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

사용상의 주의

- 모션 제어 명령을 기동할 때는 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중인지 확인해 주십시오.
  상세한 내용은 📖「10-2-1 EtherCAT 통신의 감시와 서보 ON (P.10-3)」를 참조해 주십시오.
- Execute(기동)는 사용자 프로그램을 실행하는 첫 번째 주기에서는 「FALSE」가 되도록 사용자 프로그램을 작성해 주십시오.

6-4-2 기동의 타이밍 차트

MC 기능 모듈의 모션 제어 명령은 상시 실행형인 FB입니다.
여기에서는 모션 제어 명령의 기능을 기동하기 위한 입력 변수가 「Execute(기동)」인 명령을 Execute형, 「Enable(유효)」인 명령을 Enable형으로 설명합니다.

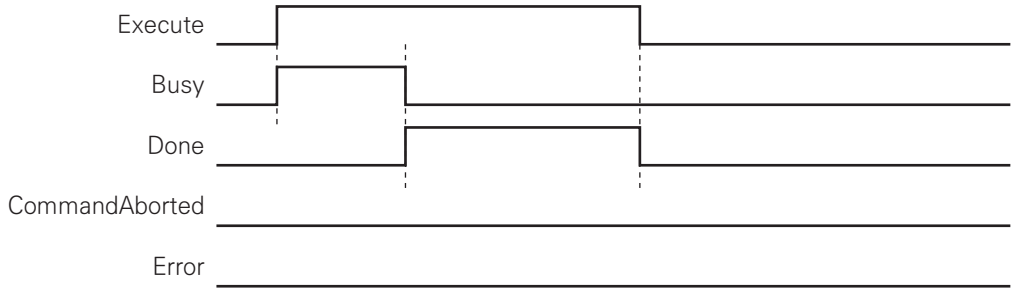
기동 조건	내용
Execute형	모션 제어 명령 입력 변수 「Execute(기동)」의 상승 시에 기동하는 명령입니다. 다음과 같은 상태가 될 때까지 명령을 계속 실행합니다. ・명령으로 지정한 동작이 완료 ・다른 모션 제어 명령을 기동하여 동작 중단 ・Execute(기동)의 재상승 엣지에 의한 재기동 그 외의 입력 변수는 Execute(기동) 상승 시에 값이 저장됩니다.
Enable형	모션 제어 명령의 입력 변수 「Enable(유효)」가 TRUE인 동안 주기마다 실행하는 명령입니다. 「Enable(유효)」이 TRUE인 동안 그 외의 입력 변수도 주기마다 입력됩니다. 단, 예외로 MC_MoveJog(조그 전송)의 입력 변수 「Velocity(목표 속도)」, 「Acceleration(가속도)」, 「Deceleration(감속도)」은 「PositiveEnable(정방향 유효」 또는 「NegativeEnable(역방향 유효)」 상승 시의 값이 입력됩니다.

사용상의 주의

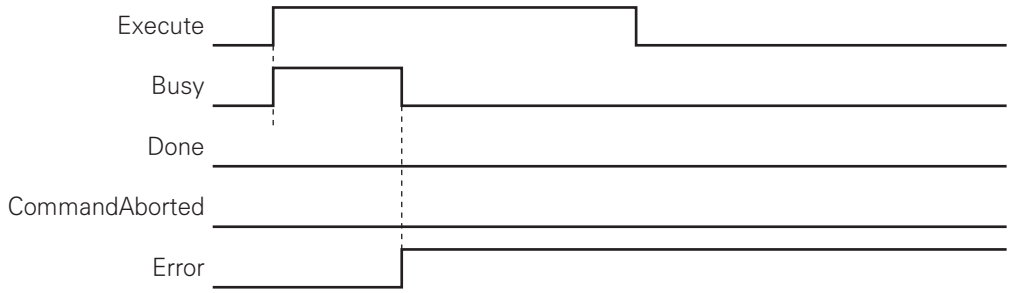
매뉴얼에 기술된 타이밍 차트는 Sysmac Studio의 데이터 트레이스에서 표시한 타이밍과 다른 경우가 있습니다.
데이터 트레이스에 대한 상세한 내용은 🕮『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

Execute형의 명령 실행 타이밍 차트

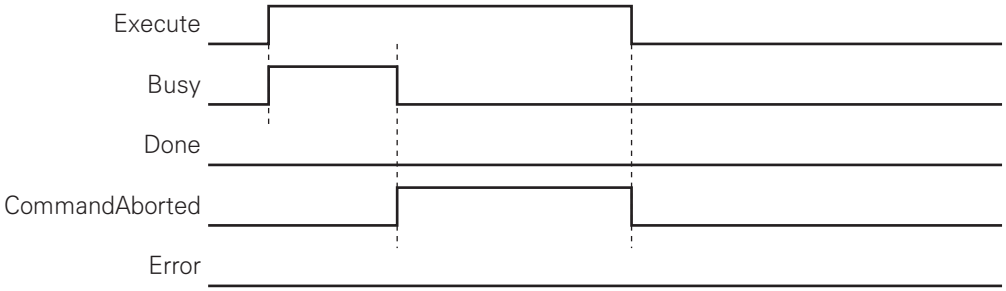
・ 입력 변수 Execute(기동)가 TRUE인 상태에서 명령 동작이 완료됩니다.
그 후 Execute(기동)를 FALSE로 할 때까지 이상이 발생하지 않은 경우를 아래에 그림으로 나타냅니다.



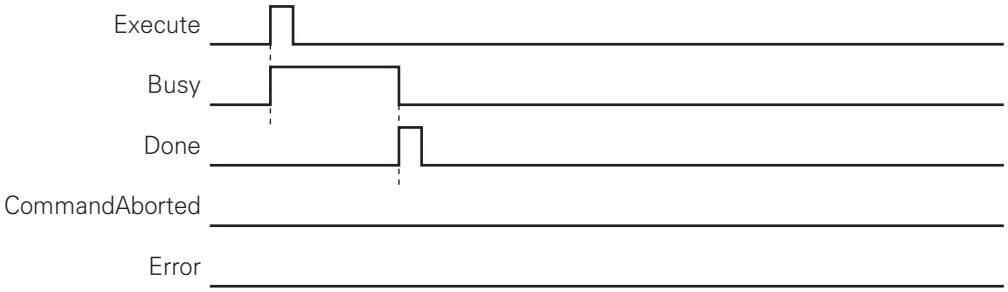
・ 입력 변수 Execute(기동)가 TRUE인 상태에서 이상이 발생한 경우를 아래에 그림으로 나타냅니다.
그 후 Execute(기동)를 FALSE로 해도 출력 변수 Error(에러)는 TRUE인 상태 그대로입니다.



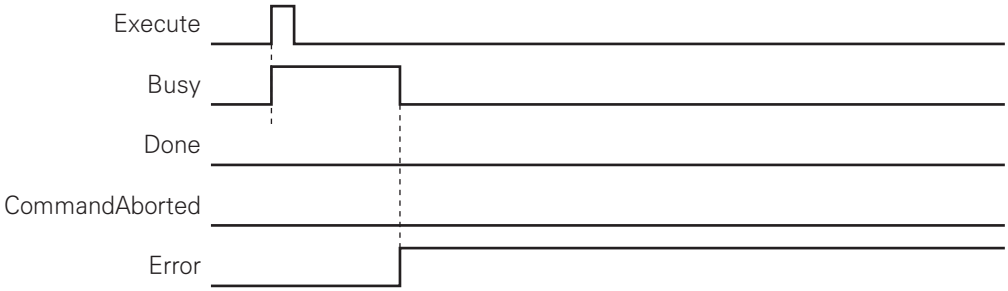
- 입력 변수 Execute(기동)가 TRUE인 상태에서 명령 실행이 도중에 중단된 경우를 아래에 그림으로 나타냅니다.



- 입력 변수 Execute(기동)를 1주기만 TRUE로 하고 명령에서 이상이 발생하지 않은 경우를 아래에 그림으로 나타냅니다.
  명령 동작이 완료되면 출력 변수 Done(완료)은 1주기만 TRUE가 됩니다.

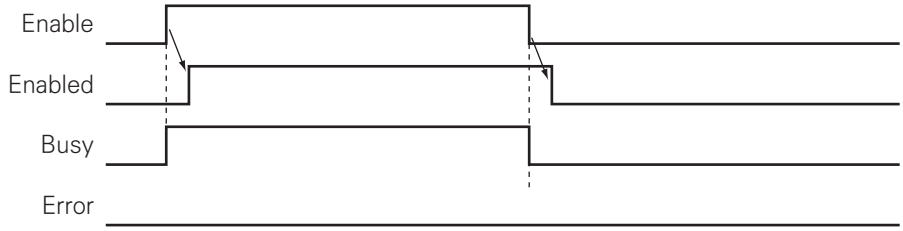


- 입력 변수 Execute(기동)를 1주기만 TRUE로 하고 명령에서 이상이 발생한 경우를 아래에 그림으로 나타냅니다.
  출력 변수 Error(에러)는 TRUE인 상태 그대로입니다.

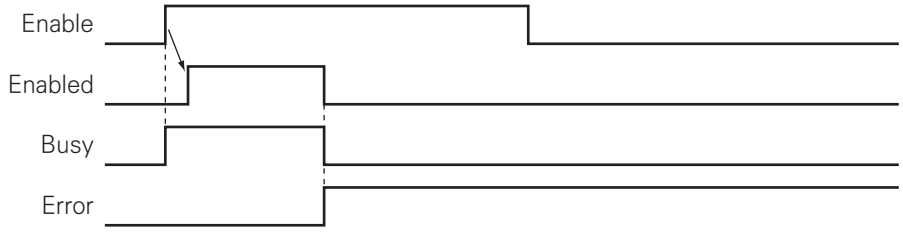


Enable형의 명령 실행 타이밍 차트

- 입력 변수 「Enable(유효)」을 TRUE로 하고 명령에서 이상이 발생하지 않은 경우를 아래에 그림으로 나타냅니다.



- 입력 변수 「Enable(유효)」을 TRUE로 하고 명령에서 이상이 발생한 경우를 아래에 그림으로 나타냅니다.



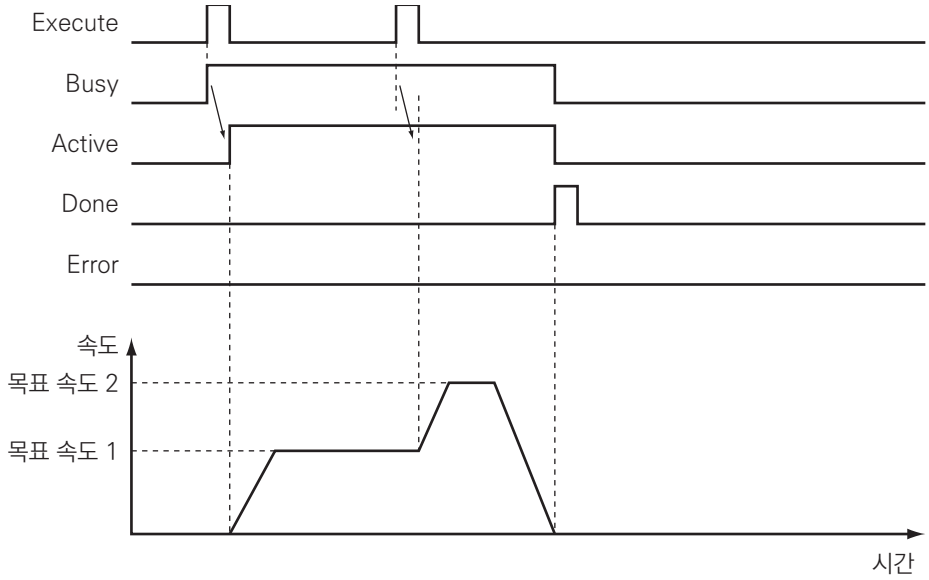
참고

MC_ZoneSwitch(구역 감시) 명령이나 MC_AxesObserve(축 간 편차 감시) 명령 등 Enable과 Enabled가 같은 타이밍이 되는 명령도 있습니다. 각 명령 타이밍에 대한 상세한 내용은 『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

6-4-3 모션 제어 명령 재기동 시의 타이밍 차트

모션 제어 명령 실행 중에 동일한 인스턴스 입력 변수의 값을 변경하고, 입력 변수 「Execute(기동)」를 TRUE→FALSE→TRUE로 하면 변경한 입력 변수의 값으로 동작합니다.

MC_MoveAbsolute(절대값 위치 결정)로 속도를 변경한 경우를 아래에 그림으로 나타냅니다.

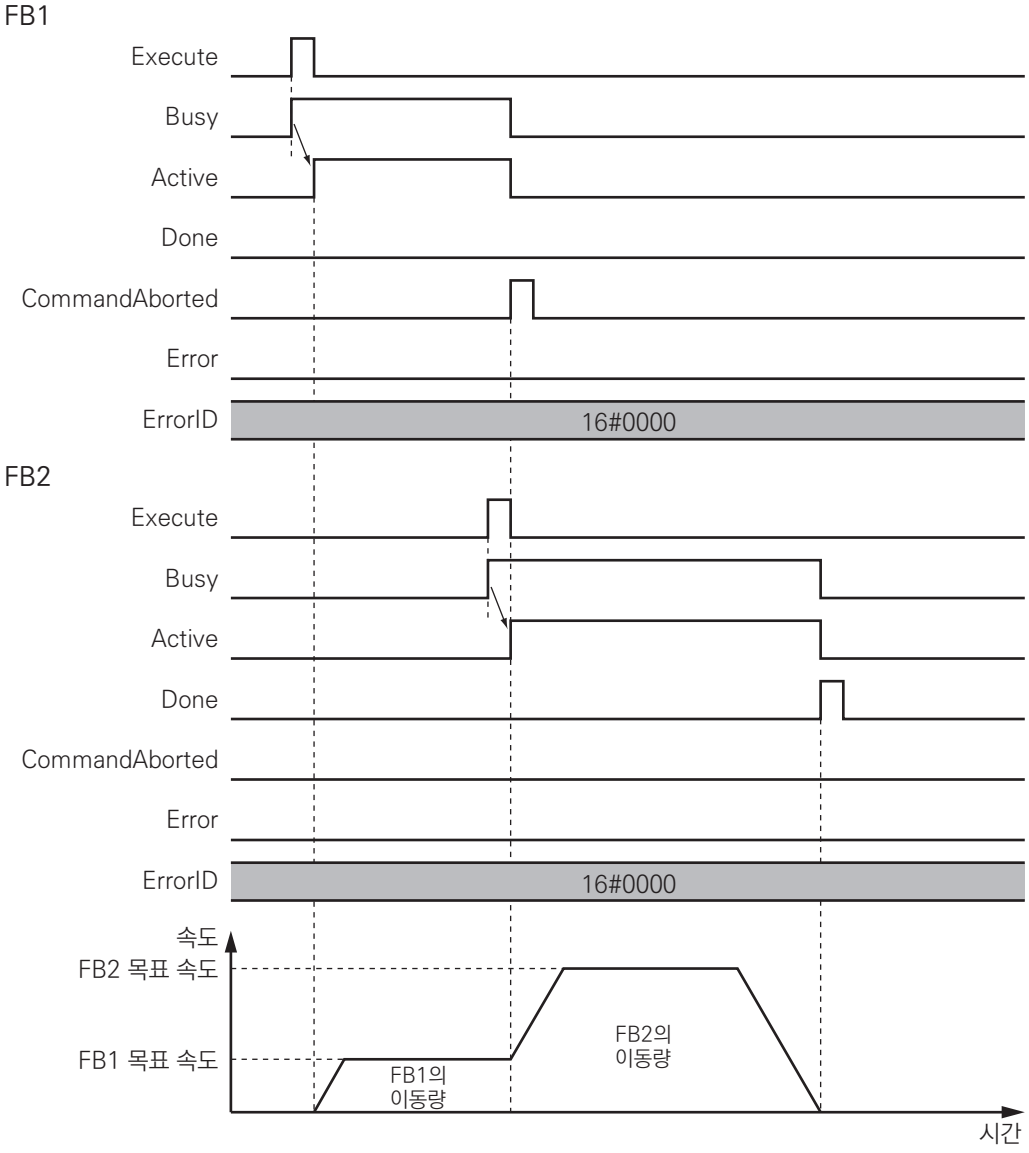


MC 기능 모듈의 명령 재기동에 관한 상세한 내용은 「9-5-6 모션 제어 명령의 재기동 (P.9-40)」, 「9-7-4 다축 협조 제어의 모션 제어 명령 재기동 (P.9-62)」을 참조해 주십시오.

6-4-4 모션 제어 명령 다중 기동 시의 타이밍 차트

동작 중인 축에 다른 인스턴스를 기동할 수 있습니다.
입력 변수의 BufferMode(버퍼드 선택)를 설정하면 동작 시작 타이밍을 지정할 수 있습니다.

아래 그림은 BufferMode(버퍼드 선택)를 Aborting(중단)으로 설정하고, 2개의 MC_MoveAbsolute(절대값 위치 결정) 명령을 다중 기동한 경우입니다.
그림 속의「FB1」및「FB2」는 명령의 인스턴스명입니다.



MC 기능 모듈의 명령 다중 기동에 대해서는 📖「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」, 📖「9-7-5 다축 협조 제어의 모션 제어 명령 다중 기동(버퍼 모드) (P.9-63)」을 참조해 주십시오.

6-5 위치의 취급

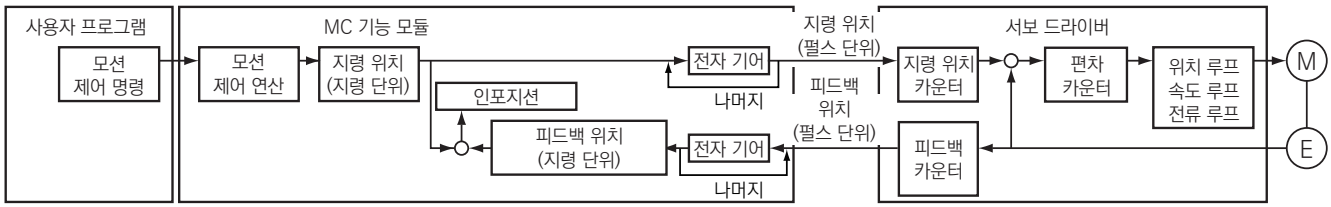
여기에서는 모션 제어 프로그램에서 다루는 위치에 대해 설명합니다.

6-5-1 위치의 종류

MC 기능 모듈에는 다음과 같이 2종류의 위치가 있습니다.

위치 종별	정의
지령 위치	MC 기능 모듈이 축을 제어하기 위해 출력하는 위치입니다.
피드백 위치	서보 드라이버 및 카운터에서 입력한 실제 위치입니다.

EtherCAT 슬레이브 서보 드라이버인 경우의 지령 위치와 피드백 위치의 관계는 아래 그림과 같습니다.



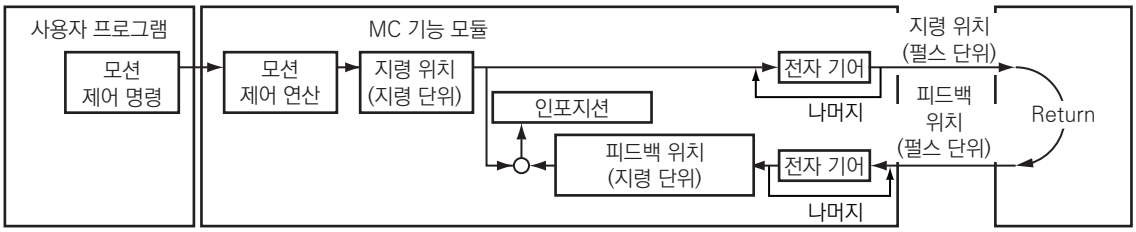
지령 위치와 피드백 위치는 아래의 항목을 공유합니다.

항목	지령 위치	피드백 위치
카운트 모드	리니어 모드 또는 로터리 모드를 설정합니다.	지령 위치와 동일한 카운트 모드가 됩니다.
위치의 단위	mm, μm, nm, inch, degree, pulse 중에서 1가지를 설정합니다.	지령 위치와 동일한 단위가 됩니다.
소프트웨어 리미트	소프트웨어적으로 동작 가능한 범위를 설정합니다.	지령 위치와 동일한 범위가 됩니다.
현재 위치 변경	현재 위치를 임의의 위치로 변경합니다.	지령 위치와 동일한 값이 동시에 설정됩니다. *1
원점 확정	원점 확정 또는 원점 미확정 상태입니다.	지령 위치와 동일한 원점 상태가 됩니다.

*1. 변경 전에 위치 편차가 있으면 위치 편차값을 유지한 값이 피드백 위치에 설정됩니다.

참고

- 가상 서보 축의 경우 지령 현재값을 펄스 단위로 변환하고, 그 값을 다시 표시 단위로 변환한 값을 피드백 현재값으로 합니다.



- NX 시리즈 위치 인터페이스 유니트의 경우는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

6-5-2 각 축 종별에서 사용 가능한 위치의 종류

축 종별에서 사용 가능한 위치의 종별은 아래와 같습니다.

축 종별	위치의 종류	
	지령 위치	피드백 위치
서보 축	사용 가능	사용 가능
가상 서보 축	사용 가능	사용 가능 *1
엔코더 축	사용 불가	사용 가능
가상 엔코더 축	사용 불가	사용 가능 *2

*1. 가상 서보 축에서는「피드백 위치 = 지령 위치」가 됩니다.
MC 기능 모듈 내부에서는 배정밀도 실수형으로 처리하고 있으므로 연산 오차를 포함하는 경우가 있습니다.
*2. 실제 엔코더가 존재하지 않을 때 사용합니다.

6-6 모션 제어 시스템 변수

여기에서는 MC 기능 모듈에 속하는 변수에 대해 설명합니다.

6-6-1 모션 제어 시스템 변수의 개요

NJ/NX 시리즈는 IEC 61131-3에 준거한 컨트롤러입니다.
NJ/NX 시리즈의 사용자 프로그램에서는 파라메터 설정 및 상태 정보 등의 데이터를 변수로 취급합니다.
이러한 변수 중 MC 기능 모듈에 속하는 시스템 정의 변수를「모션 제어 시스템 변수」라 부릅니다.

모션 제어 시스템 변수의 종류

아래 표는 모션 제어 시스템 변수 종류의 일람입니다.

제1 계층	제2 계층	제3 계층	내용
시스템 정의 변수	모션 제어 시스템 변수	MC 공통 변수	MC 기능 모듈의 공통 상태를 모니터할 수 있습니다.
		축 변수	각 축의 상태와 일부 축 파라메터의 설정 내용을 모니터할 수 있습니다.
		축 그룹 변수	각 축 그룹의 상태와 일부의 축 그룹 파라메터의 설정 내용을 모니터할 수 있습니다.

● MC 공통 변수

MC 기능 모듈의 공통 상태를 모니터하는 변수입니다. 변수명은「_MC_COM」입니다.

● 축 변수

EtherCAT 슬레이브의 서보 드라이버와 엔코더 입력 터미널, NX 시리즈 위치 인터페이스 유니트, 가상 서보 드라이버, 엔코더 입력을 다루기 위한 변수입니다.
사용자 프로그램에서는 시스템 정의 변수의 변수명과 Sysmac Studio에서 작성한 변수명을 모두 사용할 수 있습니다.
Sysmac Studio에서 작성한 변수명은 축마다 임의의 변수명으로 변경할 수 있습니다.

a) 시스템 정의 변수의 변수명은 다음과 같습니다.
- _MC_AX[0]~_MC_AX[255]
- _MC1_AX[0]~_MC1_AX[255]
- _MC2_AX[0]~_MC2_AX[255]

b) Sysmac Studio에서 작성한 변수명은 다음과 같습니다.
- MC_Axis000~MC_Axis255(디폴트)

● 축 그룹 변수

여러 개의 축을 한데 모은 그룹을 다루기 위한 변수입니다.
사용자 프로그램에서는 시스템 정의 변수의 변수명과 Sysmac Studio에서 작성한 변수명을 모두 사용할 수 있습니다.
Sysmac Studio에서 작성한 변수명은 축 그룹마다 임의의 변수명으로 변경할 수 있습니다.

a) 시스템 정의 변수의 변수명은 다음과 같습니다.
- _MC_GRP[0]~_MC_GRP[63]
- _MC1_GRP[0]~_MC1_GRP[63]
- _MC2_GRP[0]~_MC2_GRP[63]

b) Sysmac Studio에서 작성한 변수명은 다음과 같습니다.
- MC_Group000~MC_Group063(디폴트)

NJ/NX 시리즈에서 취급하는 변수에 대해서는 📖『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

모션 제어 시스템 변수로 취급하는 데이터형

모션 제어 시스템 변수에는 기본 데이터형과 파생 데이터형이 있습니다.

● 기본 데이터형

분류	데이터형	사이즈	값의 범위	기술 방법
부울	BOOL	2 *1	FALSE, TRUE	FALSE, TRUE
정수	UINT	2	0~+65535	2진 표현은 선두에 「2#」을 붙입니다. *2 8진 표현은 선두에 「8#」을 붙입니다. *3 10진 표현은 선두에 「10#」을 붙입니다. *4 16진 표현은 선두에 「16#」을 붙입니다. *5 선두에 아무것도 붙이지 않으면 10진수로 해석됩니다.
	UDINT	4	0~+4294967295	
부동 소수점	LREAL	8	-1.79769313486231e+308~ -2.22507385850721e-308, 0, 2.22507385850721e-308~ 1.79769313486231e+308, +∞/-∞	(부호)+정수부+(소수점)+(소수부)+(지수부)로 기술합니다. *6 ( ) 안 내용은 생략 가능

*1. BOOL형 데이터 사이즈는 1비트지만 메모리는 2바이트만큼 소비합니다.
*2. 2진 표현 예: 2#1111_1111, 2#1110_0000
*3. 8진 표현 예: 8#377, 8#340
*4. 10진 표현 예: -12, 0, 123_456, +986, 10#1234
*5. 16진 표현 예: 16#FF, 16#ff, 16#E0, 16#e0
*6. 예: 2, -12.0, 0.0, 0.4560, 3.14159_26, -1.34E-12, -1.34e-12, 1.0E+6, 1.0e+6, 1.234E6, 1.234e6

● 파생 데이터형

종류	내용
열거형	미리 준비된 이름 리스트 중 하나를 값으로 사용하는 데이터형입니다. 데이터형의 선두가 「_e」인 타입입니다.
구조체형	여러 개의 데이터형을 하나의 계층으로 묶은 데이터형입니다. 데이터형의 선두가 「_s」인 타입입니다.

NJ/NX 시리즈에서 취급하는 기타 데이터형에 대해서는 📖『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

모션 제어 시스템 변수의 속성

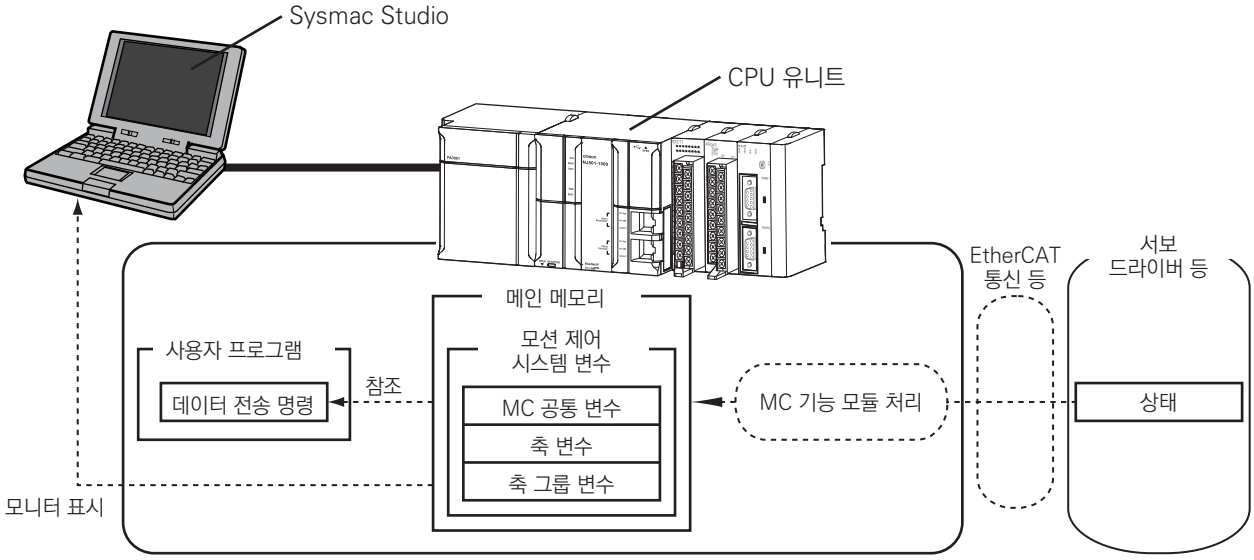
다음과 같은 속성에 대해 모션 제어 시스템 변수는 모두 동일한 속성을 가집니다.

속성	모션 제어 시스템 변수의 속성
글로벌/로컬	글로벌 변수
R/W 액세스	판독 전용
유지	비유지
네트워크 공개	공개 *1

*1. 시스템 정의 변수의 변수명으로 네트워크가 공개됩니다. Sysmac Studio에서 축 또는 축 그룹을 작성했을 때의 변수명은 네트워크 비공개입니다.

6-6-2 모션 제어 시스템 변수의 구조

모션 제어 시스템 변수는 MC 기능 모듈의 상태를 나타내는 정보, EtherCAT 통신으로 접속된 슬레이브 기기 및 CPU 유니트상의 NX 유니트 상태 정보, 모션 제어를 실행하기 위한 MC 파라메터 설정의 일부로 구성되어 있습니다.
모션 제어 시스템 변수는 사용자 프로그램에서 변수로 참조할 수 있으며 Sysmac Studio에서 모니터 표시할 수도 있습니다.



모션 제어 시스템 변수의 갱신 타이밍

모션 제어 시스템 변수는 NX701 CPU 유니트와 NX102 CPU 유니트, NX1P2 CPU 유니트 및 NJ 시리즈 CPU 유니트에서는 다음과 같이 갱신 타이밍이 다릅니다.

● NX701 CPU 유니트의 경우

프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)에 축을 할당한 경우는 아래와 같습니다.

- 프라이머리 주기마다 프라이머리 정주기 태스크에 할당한 태스크용 영역과 정주기 태스크(실행 우선도 5)의 기본 설정 영역만 갱신됩니다.
- 정주기 태스크(실행 우선도 5)의 그 외 영역은 초기값인 상태로 갱신되지 않고 정주기 태스크(실행 우선도 5)의 태스크 주기로 갱신됩니다.

갱신 대상의 변수 영역	프라이머리 정주기 태스크	정주기 태스크(실행 우선도 5)
_MC_AX[0-255]	○	
_MC1_AX[0-255]	○	
_MC2_AX[0-255]		○
_MC_GRP[0-63]	○	
_MC1_GRP[0-63]	○	
_MC2_GRP[0-63]		○

● NX102 CPU 유니트, NX1P2 CPU 유니트, NJ 시리즈 CPU 유니트의 경우

모션 제어 시스템 변수는 프라이머리 주기마다 갱신됩니다.

6-6-3 모션 제어 시스템 변수의 일람

여기에서는 모션 제어 시스템 변수의 종류마다 일람표로 설명합니다.

MC 공통 변수

MC 공통 변수는 변수명이 _MC_COM이고 데이터형이 _sCOMMON_REF형인 구조체형 변수입니다.
여기에서는 MC 공통 변수의 구조와 그 멤버에 대한 상세한 내용을 설명합니다.

변수명	데이터형	명칭	기능
_MC_COM	_sCOMMON_REF	MC 공통 변수	
Status	_sCOMMON_REF_STA	MC 공통 상태	
RunMode	BOOL	MC 운전 중	MC 기능 모듈이 운전 중일 때 TRUE가 됩니다.
TestMode	BOOL	MC 시운전 중	Sysmac Studio에서 시운전 중일 때 TRUE가 됩니다.
CamTableBusy	BOOL	캠 테이블 파일 저장 실행 중/대기 중	캠 테이블 파일 저장 실행 중 또는 대기 중일 때 TRUE가 됩니다.
GenerateCam Busy *1	BOOL	캠 테이블 생성 실행 중	캠 테이블 생성 실행 중일 때 TRUE가 됩니다.
PFaultLvl	_sMC_REF_EVENT	MC 공통 부분 정지 폴트	
Active	BOOL	MC 공통 부분 정지 폴트 발생 중	MC 공통 부분 정지 폴트가 발생했을 때 TRUE가 됩니다.
Code	WORD	MC 공통 부분 정지 폴트 코드	MC 공통 부분 정지 폴트의 코드를 출력합니다. 이벤트 코드의 상위 4자리와 같은 값입니다.
MFaultLvl	_sMC_REF_EVENT	MC 공통 경도 폴트	
Active	BOOL	MC 공통 경도 폴트 발생 중	MC 공통 경도 폴트가 발생했을 때 TRUE가 됩니다.
Code	WORD	MC 공통 경도 폴트 코드	MC 공통 경도 폴트의 코드를 출력합니다. 이벤트 코드의 상위 4자리와 같은 값입니다.
Obsr	_sMC_REF_EVENT	MC 공통 감시 정보	
Active	BOOL	MC 공통 감시 정보 발생 중	MC 공통 감시 정보가 발생했을 때 TRUE가 됩니다.
Code	WORD	MC 공통 감시 정보 코드	MC 공통 감시 정보의 코드를 출력합니다. 이벤트 코드의 상위 4자리와 같은 값입니다.

*1. 유니트 버전 Ver.1.08 이상인 CPU 유니트와 Sysmac Studio Ver.1.09 이상의 조합으로 사용할 수 있습니다.

축 변수

축 변수는 시스템 정의 변수의 변수명이 _MC_AX[0-255] 또는 _MC1_AX[0-255], _MC2_AX[0-255]이고 데이터형이 _sAXIS_REF형인 구조체형 변수입니다.
여기에서는 _MC_AX[0-255]를 예로 들어 축 변수의 구성과 그 멤버에 대한 상세한 내용을 설명합니다. _MC1_AX[0-255], _MC2_AX[0-255]에 대해서도 동일합니다.

변수명	데이터형	명칭	기능
_MC_AX[0-255]	_sAXIS_REF	축 변수	
Status	_sAXIS_REF_STA	축 상태	
Ready	BOOL	축 기동 준비 완료	축의 기동 준비가 완료되고 정지 중일 때 TRUE가 됩니다. _MC_AX[*].Status.Standstill이 「TRUE: 정지 중」과 동일한 상태를 나타냅니다.
Disabled	BOOL	축 무효	축이 서보 OFF이고 정지 중인 상태에서 TRUE가 됩니다. 아래의 축 상태는 배타적으로 이 중에서 하나만 TRUE가 됩니다. Disabled/ Standstill/ Discrete/ Continuous/ Synchronized/ Homing/ Stopping/ ErrorStop/ /Coordinated
Standstill	BOOL	정지 중	축이 서보 ON이고 정지 중인 상태에서 TRUE가 됩니다.
Discrete	BOOL	위치 결정 동작 중	목표 위치를 향해 위치 결정 제어를 실행하고 있을 때 TRUE가 됩니다. 위치 결정 동작 중에 오버라이드값을 「0」으로 설정하여 속도 「0」이 된 상태도 포함합니다.
Continuous	BOOL	연속 동작 중	목표 위치 없이 연속 동작 중일 때 TRUE가 됩니다. 속도 제어 또는 토크 제어일 때 이 상태가 됩니다. 목표 속도를 「0」으로 설정하여 속도 「0」이 된 상태 및 연속 동작 중에 오버라이드값을 「0」으로 설정하여 속도 「0」이 된 상태도 포함합니다.
Synchronized	BOOL	동기 동작 중	동기 제어를 실행하고 있을 때 TRUE가 됩니다. 동기 제어 명령 변환 후의 동기 대기 상태도 포함합니다.
Homing	BOOL	원점 복귀 중	MC_Home 명령 또는 MC_HomeWithParameter 명령에 의해 원점 복귀를 실행하고 있을 때 TRUE가 됩니다.
Stopping	BOOL	감속 정지 중	MC_Stop 명령, MC_TouchProbe 명령에 의해 축이 정지하기 전까지인 상태일 때 TRUE가 됩니다. MC_Stop 명령으로 축이 정지한 후에 Execute가 TRUE일 때도 포함합니다. 감속 정지 중 상태에서는 축 동작 명령을 기동할 수 없습니다. (CommandAborted=TRUE)

변수명	데이터형	명칭	기능
ErrorStop	BOOL	에러 감속 정지 중	MC_ImmediateStop 명령 또는 축 경도 폴트 발생 중(_MC_AX[*].MFaultLvl.Active)이 「TRUE: 축 경도 폴트 발생 중」이고 축이 정지할 때까지 또는 정지하고 있는 상태. 이 상태에서는 축 동작 명령을 기동할 수 없습니다. (CommandAborted=TRUE)
Coordinated	BOOL	다축 협조 동작 중	다축 협조 명령에 의해 축 그룹 유효가 된 상태에서 TRUE가 됩니다.
Details	_sAXIS_REF_DET	축 제어 상태	지령의 제어 상태를 나타낸다.
Idle	BOOL	정지 중	인포지션 대기를 제외하고 지령값을 연산하지 않을 때 TRUE가 됩니다. *1 Idle과 InPosWaiting은 배타적으로 동시에는 TRUE가 되지 않습니다.
InPosWaiting	BOOL	인포지션 대기	인포지션 대기일 때 TRUE가 됩니다. 인포지션 체크를 실행하는 위치 결정에서 인포지션 체크를 실행하고 있는 상태입니다.
Homed	BOOL	원점 확정	원점 확정 상태 시에 TRUE가 됩니다. *2 FALSE: 원점 미확정 TRUE: 원점 확정
InHome	BOOL	원점 정지	원점 위치의 범위 내에서 TRUE가 됩니다. 아래의 AND 조건입니다. · 원점 확정 상태 · 피드백 현재 위치가 원점을 중심으로 「원점 위치 범위」의 폭에 들어갔을 때 지령 상태에서 축 동작 중에 원점 위치를 통과하는 경우에도 TRUE가 됩니다.
VelLimit *3	BOOL	지령 속도 포화	동기 제어 중에 지령 속도가 최고 속도로 제한되었을 때 TRUE가 됩니다.
Dir	_sAXIS_REF_DIR	지령 방향 상태	지령을 내린 이동 방향을 나타냅니다.
Posi	BOOL	정방향 지령 중	정방향으로 지령을 내리고 있을 때 TRUE가 됩니다.
Nega	BOOL	역방향 지령 중	역방향으로 지령을 내리고 있을 때 TRUE가 됩니다.
DrvStatus	_sAXIS_REF_STA_DRV	서보 드라이버 상태	드라이버 등의 상태를 나타냅니다.
ServoOn	BOOL	서보 ON	서보 모터에 통전 중일 때 TRUE가 됩니다.
Ready	BOOL	서보 준비	서보 준비 상태*4에서 TRUE가 됩니다.
MainPower	BOOL	주회로 전원	서보 드라이버의 주회로 전원이 통전 중일 때 TRUE가 됩니다.
P_OT	BOOL	정방향 한계 입력	정방향 한계 입력이 유효할 때 TRUE가 됩니다.
N_OT	BOOL	역방향 한계 입력	역방향 한계 입력이 유효할 때 TRUE가 됩니다.
HomeSw	BOOL	원점 근방 입력	원점 근방 입력이 유효할 때 TRUE가 됩니다.
Home	BOOL	원점 입력	원점 입력이 유효할 때 TRUE가 됩니다. *5 *6
ImdStop	BOOL	즉시 정지 입력	즉시 정지 입력이 유효할 때 TRUE가 됩니다.

변수명	데이터형	명칭	기능
Latch1	BOOL	외부 래치 입력 1	래치 입력 1이 유효할 때 TRUE가 됩니다.
Latch2	BOOL	외부 래치 입력 2	래치 입력 2가 유효할 때 TRUE가 됩니다.
DrvAlarm	BOOL	드라이버 에러 입력	드라이버 에러가 발생 중일 때 TRUE가 됩니다.
DrvWarning	BOOL	드라이버 워닝 입력	드라이버 워닝이 발생 중일 때 TRUE가 됩니다.
ILA	BOOL	드라이버 내부 기능 제한 중	서보 드라이버 측의 제한 기능에 의해 실제로 제한 중일 때 TRUE가 됩니다. *7
CSP	BOOL	사이클릭 동기 위치 (CSP) 모드 중	서보 드라이버 측이 서보 ON이고 CSP 모드일 때 TRUE가 됩니다. *8
CSV	BOOL	사이클릭 동기 속도 (CSV) 모드 중	서보 드라이버 측이 서보 ON이고 CSV 모드일 때 TRUE가 됩니다. *6
CST	BOOL	사이클릭 동기 토크 (CST) 모드 중	서보 드라이버 측이 서보 ON이고 CST 모드일 때 TRUE가 됩니다. *6
Cmd	_sAXIS_REF_CMD_DATA	축 지령값	
Pos	LREAL	지령 현재 위치	지령 위치의 현재값을 출력합니다. (단위: 지령 단위) 서보 OFF 시나 위치 제어 모드 이외일 때는 피드백 현재 위치를 출력합니다. *9
Vel	LREAL	지령 현재 속도	지령 속도의 현재값을 출력합니다. (단위: 지령 단위/s) 정방향 이동 시에는 + 부호, 역방향 이동 시에는 - 부호가 붙습니다. 지령 현재 위치의 차분으로 산출한 속도입니다. 서보 OFF 시나 위치 제어 모드 이외일 때는 피드백 현재 위치를 바탕으로 산출한 속도입니다.
AccDec	LREAL	지령 현재 가감 속도	지령 가감 속도의 현재값을 출력합니다. (단위: 지령 단위/s^2) 지령 현재 속도의 차분으로 산출한 가감속도입니다. 가속 시에는 + 부호, 감속 시에는 - 부호가 붙습니다. 실행 중 명령의 지령 가감 속도가 「0」인 경우는 「0」이 됩니다.
Jerk	LREAL	지령 현재 저크	지령 저크의 현재값을 출력합니다. (단위: 지령 단위/s^3) 가감 속도의 절대값 증가 시에는 + 부호, 감소 시에는 - 부호가 붙습니다. 실행 중 명령의 지령 가감 속도, 지령 저크가 「0」인 경우는 「0」이 됩니다.
Trq	LREAL	지령 현재 토크	지령 토크의 현재값을 출력합니다. (단위: %) 정방향 이동 시에는 + 부호, 역방향 이동 시에는 - 부호가 붙습니다. 토크 제어 모드 이외에서는 피드백 현재 토크와 동일한 값을 출력합니다. *10

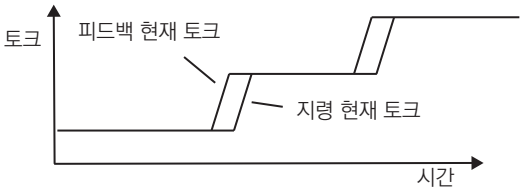
변수명	데이터형	명칭	기능
Act	_sAXIS_REF_ACT_DATA	축 현재값	
Pos	LREAL	피드백 현재 위치	피드백 위치의 현재값을 출력합니다. (단위: 지령 단위) *9
Vel	LREAL	피드백 현재 속도	피드백 속도의 현재값을 출력합니다. (단위: 지령 단위/s) 정방향 이동 시에는 + 부호, 역방향 이동 시에는 - 부호가 붙습니다.
Trq	LREAL	피드백 현재 토크	피드백 토크의 현재값을 출력합니다. (단위: %) 정방향 이동 시에는 + 부호, 역방향 이동 시에는 - 부호가 붙습니다.
TimeStamp *11	ULINT	타임 스탬프	축의 현재 위치를 갱신한 시각을 출력합니다. 변화 시각 취득 기능이 동작 중인 축에서 유효합니다. (단위: ns)
MFaultLvl	_sMC_REF_EVENT	축 경도 폴트	
Active	BOOL	축 경도 폴트 발생 중	축 경도 폴트가 발생 중일 때 TRUE가 됩니다.
Code	WORD	축 경도 폴트 코드	축 경도 폴트의 코드를 출력합니다. 이벤트 코드의 상위 4자리와 같은 값입니다.
Obsr	_sMC_REF_EVENT	축 감시 정보	
Active	BOOL	축 감시 정보 발생 중	축 감시 정보가 발생 중일 때 TRUE가 됩니다.
Code	WORD	축 감시 정보 코드	축 감시 정보의 코드를 출력합니다. 이벤트 코드의 상위 4자리와 같은 값입니다.
Cfg	_sAXIS_REF_CFG	축 기본 설정	축 기본 설정의 설정 내용을 나타냅니다.
AxNo	UINT	축 번호	축의 논리 번호를 나타냅니다.
AxEnable	_eMC_AXIS_USE	축 사용	축의 사용 여부를 나타냅니다. 0: _mcNoneAxis(미작성 축) 1: _mcUnusedAxis(미사용 축) 2: _mcUsedAxis(사용 축)
AxType	_eMC_AXIS_TYPE	축 종별	축 종별을 나타냅니다. 가상 축에서는 입출력 배선이 필요 없습니다. 0: _mcServo(서보 축) 1: _mcEncdr(엔코더 축) 2: _mcVirServo(가상 서보 축) 3: _mcVirEncdr(가상 엔코더 축)
NodeAddress	UINT	노드 주소	EtherCAT의 슬레이브 주소를 나타냅니다. *12 16#FFFF는 없음을 나타냅니다.
ExecID *13	UINT	실행 ID	태스크별 실행 ID를 나타냅니다. 0: 태스크 할당 없음(미작성 축) 1: 프라이머리 정주기 태스크에 할당 2: 정주기 태스크(실행 우선도 5)에 할당
Scale	_sAXIS_REF_SCALE	단위 변환	전자 기어비의 설정값을 나타냅니다.
Num	UDINT	모터 1회전의 펄스 수	지령 위치의 모터 1회전 펄스 수를 나타냅니다. 전자 기어비를 바탕으로 지령값에서 펄스량으로 변환합니다.
Den	LREAL	모터 1회전의 이동량	지령 위치의 모터 1회전당 워크의 이동량을 설정합니다. *14

변수명			데이터형	명칭	기능
		Units	_eMC_UNITS	표시 단위	지령 위치의 표시 단위를 나타냅니다. 0: _mcPls (pulse) 1: _mcMm (mm) 2: _mcUm (μm) 3: _mcNm (nm) 4: _mcDeg (degree) 5: _mcInch (inch)
		CountMode *13	_eMC_COUNT_MODE	카운트 모드	카운트 모드를 나타냅니다. 0: _mcCountModeLinear(리니어 모드) 1: _mcCountModeRotary(로터리 모드)
		MaxPos *13	LREAL	현재 위치 상한값	현재 위치 표시의 상한값을 나타냅니다. *15
		MinPos *13	LREAL	현재 위치 하한값	현재 위치 표시의 하한값을 나타냅니다. *16

*1. 속도「0」에서의 동작, 편차 카운터 리셋 중, 동기 제어 중, 다축 협조 동작 중도 연산하고 있는 상태에 포함됩니다.
*2. TRUE라도 아래의 경우에는 원점을 재확정해야 합니다.
위치 카운트 설정 또는 단위 변환 설정을 변경한 경우.
서보 드라이버 측에서 절대 위치 정보를 상실하는 이상 발생이나 조작을 실시한 경우. 예: 엔코더 케이블의 단선이나 절대값 엔코더 데이터의 삭제 등.
*3. VelLimit는 동기 제어 중인 종축에만 사용해 주십시오.
*4. 서보 드라이버의 PDS 상태가「Ready to switch on」,「Switched on」,「Operation enabled」중 하나인 경우이며 주회로 전원(voltage enabled)이 ON인 경우에 TRUE가 됩니다. 단, [Main circuit power supply OFF detection]이 [Do not detect]로 설정된 경우 주회로 전원의 ON/OFF는 무시합니다. PDS 상태, 주회로 전원 OFF 검출에 대한 상세한 내용은 「A-5 PDS 상태 변이 (P.A-28)」를 참조해 주십시오.
*5. Sysmac Studio의 [Axis Basic Settings] 화면의 [Detailed Settings]에서 디지털 입력의 엔코터 Z상 검출로 설정한 신호의 상태입니다. 타사 드라이버에서는 이 신호를 PDO에 매핑할 수 없는 경우가 있습니다. 접속할 드라이버의 매뉴얼을 확인해 주십시오.
*6. 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입은 이 신호를 PDO에 매핑할 수 없습니다.
*7. PDO에 매핑된「상태 워드(6041Hex)」의 비트 11(내부 제한 기능 유효)의 상태를 표시합니다. TRUE가 되는 조건은 서보 드라이버 사양에 따라 다릅니다. 접속한 서보 드라이버의 매뉴얼을 참조해 주십시오. 오므론 서보 드라이버 1S 시리즈 및 G5 시리즈에서는 토크 제한/속도 제한/구동 금지 입력/소프트웨어 리미트 중 하나입니다.
*8. PDO에 매핑된「상태 워드(6061Hex)」의 값을 바탕으로 표시합니다. CSP, CSV, CST가 TRUE가 되는 조건은 서보 드라이버 사양에 따라 다릅니다. 접속한 서보 드라이버의 매뉴얼을 참조해 주십시오.
PDO에「오퍼레이션 모드 표시(6061Hex)」가 매핑되지 않은 경우 CPU 유니트의 버전에 따라 다음과 같이 됩니다.
CPU 유니트 Ver.1.09 이하에서는 항상 FALSE가 됩니다.
CPU 유니트 Ver.1.10 이상에서는 PDO에 매핑된「상태 워드(6041Hex)」의 상태가 Operation Enabled일 때 TRUE가 됩니다.
*9. 축에 할당한 EtherCAT 슬레이브 또는 NX 유니트와 CPU 유니트 간의 프로세스 데이터 통신이 확립되지 않은 경우 CPU 유니트의 버전에 따라 다음과 같이 됩니다.
CPU 유니트 Ver.1.09 이하에서는 축 변수의 피드백 현재 위치 및 지령 현재 위치는「0」또는 하한값이 됩니다. 하한값이 되는 조건은 카운트 모드가 로터리 모드에서 위치 범위에「0」을 포함하지 않는 경우입니다.
CPU 유니트 Ver.1.10 이상에서는 프로세스 데이터 통신이 확립되기 직전의 피드백 현재 위치가 축 변수의 피드백 현재 위치 및 지령 현재 위치입니다.

*10.지령 현재 토크와 피드백 현재 토크를 데이터 트레이스로 비교 표시한 경우 토크 제어 모드 이외에서는 지령 현재 토크가 1태스크 주기 후부터 바뀌도록 표시됩니다. 이는 데이터 트레이스의 처리 타이밍에 의한 영향입니다. 데이터 트레이스의 처리 타이밍에 대해서는 📖『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』의 데이터 트레이스 내용을 참조해 주십시오.

데이터 트레이스 화면의 예



지령 현재 토크가 피드백 현재 토크보다
늦게 바뀌도록 표시됩니다.

*11.유니트 버전 Ver.1.06 이상인 CPU 유니트와 Sysmac Studio Ver.1.07 이상의 조합으로 사용할 수 있습니다.
*12.NX 시리즈 위치 인터페이스 유니트의 경우는 위치 인터페이스 유니트를 장착한 EtherCAT 커플러 유니트의 노드 주소를 나타냅니다.
*13.유니트 버전 Ver.1.10 이상인 CPU 유니트와 Sysmac Studio Ver.1.12 이상의 조합으로 사용할 수 있습니다.
*14.유니트 버전 Ver.1.11 이상인 CPU 유니트에서 추가된 단위 변환 설정의 감속기 사용을「사용함」으로 설정한 경우 이 파라메터는 무효해집니다. 대신에 유효한 파라메터가 되는「워크 측 1회전의 이동량」,「워크 측 기어비」,「모터 측 기어비」의 값을 확인할 경우에는 MC_ReadAxisParameter(축 파라메터 판독) 명령을 사용해 주십시오.
*15.카운트 모드가 리니어 모드인 경우에는 오버 플로우 직전의 위치를 표시합니다. 로터리 모드인 경우에는 링 카운터 상한값을 표시합니다.
*16.카운트 모드가 리니어 모드인 경우에는 언더 플로우 직전의 위치를 표시합니다. 로터리 모드인 경우에는 링 카운터 하한값을 표시합니다.

● 축 변수와 축 종별의 관계

축 변수에 대해서는 축 종별마다 유효/무효한 멤버가 다릅니다. 무효한 멤버를 참조하면「FALSE」혹은「0」이 판독됩니다.

여기에서는 _MC_AX[0-255]를 예로 들어 설명합니다. _MC1_AX[0-255], _MC2_AX[0-255]의 경우도 동일합니다. 또한, 축 종별마다 각 멤버의 유효/무효가 일람으로 기재되어 있습니다.

○: 유효, —: 무효

변수명	데이터형	명칭	서보 축	가상 서보 축	엔코더 축	가상 엔코더 축
_MC_AX[0-255]	_sAXIS_REF	축 변수				
Status	_sAXIS_REF_STA	축 상태				
Ready	BOOL	축 기동 준비 완료	○	○	—	—
Disabled	BOOL	축 무효	○	○	○	○
Standstill	BOOL	정지 중	○	○	—	—
Discrete	BOOL	위치 결정 동작 중	○	○	—	—
Continuous	BOOL	연속 동작 중	○	○	—	—
Synchronized	BOOL	동기 동작 중	○	○	—	—
Homing	BOOL	원점 복귀 중	○	○	—	—
Stopping	BOOL	감속 정지 중	○	○	—	—
ErrorStop	BOOL	에러 감속 정지 중	○	○	—	—
Coordinated	BOOL	다축 협조 동작 중	○	○	—	—
Details	_sAXIS_REF_DET	축 제어 상태				
Idle	BOOL	정지 중	○	○	○	○
InPosWaiting	BOOL	인포지션 대기	○	○	—	—
Homed	BOOL	원점 확정	○	○	—	—
InHome	BOOL	원점 정지	○	○	—	—
VelLimit	BOOL	지령 속도 포화	○	○	—	—

변수명	데이터형	명칭	서보 축	가상 서보 축	엔코더 축	가상 엔코더 축
Dir	_sAXIS_REF_DIR	지령 방향 상태				
Posi	BOOL	정방향 지령 중	○	○	—	—
Nega	BOOL	역방향 지령 중	○	○	—	—
DrvStatus	_sAXIS_REF_STA_DRV	서보 드라이버 상태				
ServoOn	BOOL	서보 ON	○	—	—	—
Ready	BOOL	서보 준비	○	—	—	—
MainPower	BOOL	주회로 전원	○	—	—	—
P_OT	BOOL	정방향 한계 입력	○	—	—	—
N_OT	BOOL	역방향 한계 입력	○	—	—	—
HomeSw	BOOL	원점 근방 입력	○	—	—	—
Home	BOOL	원점 입력	○	—	—	—
ImdStop	BOOL	즉시 정지 입력	○	—	—	—
Latch1	BOOL	외부 래치 입력 1	○	—	—	—
Latch2	BOOL	외부 래치 입력 2	○	—	—	—
DrvAlarm	BOOL	드라이버 에러 입력	○	—	—	—
DrvWarning	BOOL	드라이버 워닝 입력	○	—	—	—
ILA	BOOL	드라이버 내부 기능 제한 중	○	—	—	—
CSP	BOOL	사이클릭 동기 위치(CSP) 모드 중	○	—	—	—
CSV	BOOL	사이클릭 동기 속도(CSV) 모드 중	○	—	—	—
CST	BOOL	사이클릭 동기 토크(CST) 모드 중	○	—	—	—
Cmd	_sAXIS_REF_CMD_DATA	축 지령값				
Pos	LREAL	지령 현재 위치	○	○	—	—
Vel	LREAL	지령 현재 속도	○	○	—	—
AccDec	LREAL	지령 현재 가감 속도	○	○	—	—
Jerk	LREAL	지령 현재 저크	○	○	—	—
Trq	LREAL	지령 현재 토크	○	○	—	—
Act	_sAXIS_REF_ACT_DATA	축 현재값				
Pos	LREAL	피드백 현재 위치	○	○	○	○
Vel	LREAL	피드백 현재 속도	○	○	○	○
Trq	LREAL	피드백 현재 토크	○	○	—	—
TimeStamp	ULINT	타임 스탬프	○	—	○	—
MFaultLvl	_sMC_REF_EVENT	축 경도 폴트				
Active	BOOL	축 경도 폴트 발생 중	○	○	○	○
Code	WORD	축 경도 폴트 코드	○	○	○	○
Obsr	_sMC_REF_EVENT	축 감시 정보				
Active	BOOL	축 감시 정보 발생 중	○	○	○	○
Code	WORD	축 감시 정보 코드	○	○	○	○
Cfg	_sAXIS_REF_CFG	축 기본 설정				
AxNo	UINT	축 번호	○	○	○	○
AxEnable	_eMC_AXIS_USE	축 사용	○	○	○	○
AxType	_eMC_AXIS_TYPE	축 종별	○	○	○	○
NodeAddress	UINT	노드 주소	○	—	○	—
ExecID *1	UINT	실행 ID	○	○	○	○

변수명	데이터형	명칭	서보 축	가상 서보 축	엔코더 축	가상 엔코더 축
Scale	_sAXIS_REF_SCALE	단위 변환				
Num	UDINT	모터 1회전의 펄스 수	○	○	○	○
Den *2	LREAL	모터 1회전의 이동량	○	○	○	○
Units	_eMC_UNITS	표시 단위	○	○	○	○
CountMode *1	_eMC_COUNT_MODE	카운트 모드	○	○	○	○
MaxPos *1	LREAL	현재 위치 상한값	○	○	○	○
MinPos *1	LREAL	현재 위치 하한값	○	○	○	○

*1. 유니트 버전 Ver.1.10 이상인 CPU 유니트와 Sysmac Studio Ver.1.12 이상의 조합으로 사용할 수 있습니다.
*2. 유니트 버전 Ver.1.11 이상인 CPU 유니트에서 추가된 단위 변환 설정의 감속기 사용을「사용함」으로 설정한 경우 이 파라메터는 무효해집니다.

축 그룹 변수

축 그룹 변수는 시스템 정의 변수의 변수명이 _MC_GRP[0-63] 및 _MC1_GRP[0-63], _MC2_GRP[0-63]이고 데이터형이 _sGROUP_REF형인 구조체형 변수입니다.
여기에서는 _MC_GRP[0-63]을 예로 들어 축 그룹 변수의 구성과 그 멤버에 대한 상세한 내용을 설명합니다. _MC1_GRP[0-63], _MC2_GRP[0-63]에 대해서도 동일합니다.
또한, 기능 설명에서 축 변수는 _MC_AX[*]를 예로 들어 설명하고 있는데 _MC1_AX[*], _MC2_AX[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_GRP[0-63]	_sGROUP_REF	축 그룹 변수	
Status	_sGROUP_REF_STA	축 그룹 상태	
Ready	BOOL	기동 준비 완료	축 그룹이 정지 중이며 기동 준비가 완료되었을 때 TRUE가 됩니다. 아래의 AND 조건이 기동 준비가 완료된 조건입니다. ・구성 축에서 MC_Stop 명령이 실행되지 않는다. ・_MC_GRP[*].Status.Standby가 TRUE(정지 중) ・구성 축이 서보 ON 상태 ・구성 축의 _MC_AX[*].Details.Homed가 TRUE(원점 확정)
Disabled	BOOL	축 그룹 무효	축 그룹 무효로 정지 중일 때 TRUE가 됩니다. 아래의 축 그룹 상태는 배타적으로 이 중에서 하나만 TRUE가 됩니다. Disabled/ Standby/ Moving/ Stopping/ ErrorStop
Standby	BOOL	정지 중	축 그룹 동작 명령이 정지 중일 때 TRUE가 됩니다. 축 그룹 구성 축의 서버 ON/OFF 상태에 따르지 않습니다.
Moving	BOOL	동작 중	목표 위치를 향해 축 그룹 동작 명령을 실행하고 있을 때 TRUE가 됩니다. 인포지션 대기 상태 및 오버라이드에 의한 속도「0」의 상태도 포함됩니다.

변수명	데이터형	명칭	기능
Stopping	BOOL	감속 정지 중	MC_GroupStop 명령에 의해 축 그룹이 정지하기 전까지인 상태일 때 TRUE가 됩니다. MC_GroupStop 명령으로 축이 정지한 후에 Execute가 TRUE일 때도 포함합니다. 본 상태에서는 축 그룹 동작 명령을 기동할 수 없습니다. (CommandAborted=TRUE)
ErrorStop	BOOL	에러 감속 정지 중	MC_GroupImmediateStop 명령 또는 축 그룹 경도 폴트 발생 (_MC_GRP[*].MFaultLvl.Active=TRUE)으로 인해 축 그룹이 정지할 때까지 또는 정지 중일 때 TRUE가 됩니다. 이 상태에서는 축 그룹 동작 명령을 기동할 수 없습니다. (CommandAborted=TRUE)
Details	_sGROUP_REF_DET	축 그룹 제어 상태	지령의 제어 상태를 나타냅니다.
Idle	BOOL	정지 중	인포지션 대기를 제외하고 지령값을 연산하지 않을 때 TRUE가 됩니다. *1 Idle과 InPosWaiting은 배타적으로 동시에는 TRUE가 되지 않습니다.
InPosWaiting	BOOL	인포지션 대기	어느 한 구성 축이 인포지션 대기일 때 TRUE가 됩니다. 인포지션 체크를 실행하는 위치 결정에서 인포지션 체크를 실행하고 있는 상태입니다. *2
Cmd	_sGROUP_REF_CMD_DATA	축 그룹 지령값	
Vel	LREAL	지령 보간 속도	지령 보간 속도의 현재값을 출력합니다. 보간 지령 현재 위치의 차분으로 산출한 보간 속도입니다. 정방향 이동 시에는 + 부호, 역방향 이동 시에는 - 부호가 붙습니다. 축 그룹이 무효인 경우는 「0」이 됩니다.
AccDec	LREAL	지령 보간 가감 속도	지령 보간 가감 속도의 현재값을 출력합니다. 지령 보간 속도의 차분으로 산출한 보간 가감 속도입니다. 가속 시에는 + 부호, 감속 시에는 - 부호가 붙습니다. 축 그룹이 무효이거나 실행 중인 축 그룹 동작 명령의 지령 가감 속도 지정이 「0」인 경우는 「0」이 됩니다.
MFaultLvl	_sMC_REF_EVENT	축 그룹 경도 폴트	
Active	BOOL	축 그룹 경도 폴트 발생 중	축 그룹 경도 폴트가 발생 중일 때 TRUE가 됩니다.
Code	WORD	축 그룹 경도 폴트 코드	축 그룹 경도 폴트의 코드를 출력합니다. 이벤트 코드의 상위 4자리와 같은 값입니다.
Obsr	_sMC_REF_EVENT	축 그룹 감시 정보	
Active	BOOL	축 그룹 감시 정보 발생 중	축 그룹 감시 정보가 발생 중일 때 TRUE가 됩니다.
Code	WORD	축 그룹 감시 정보 코드	축 그룹 감시 정보의 코드를 출력합니다. 이벤트 코드의 상위 4자리와 같은 값입니다.

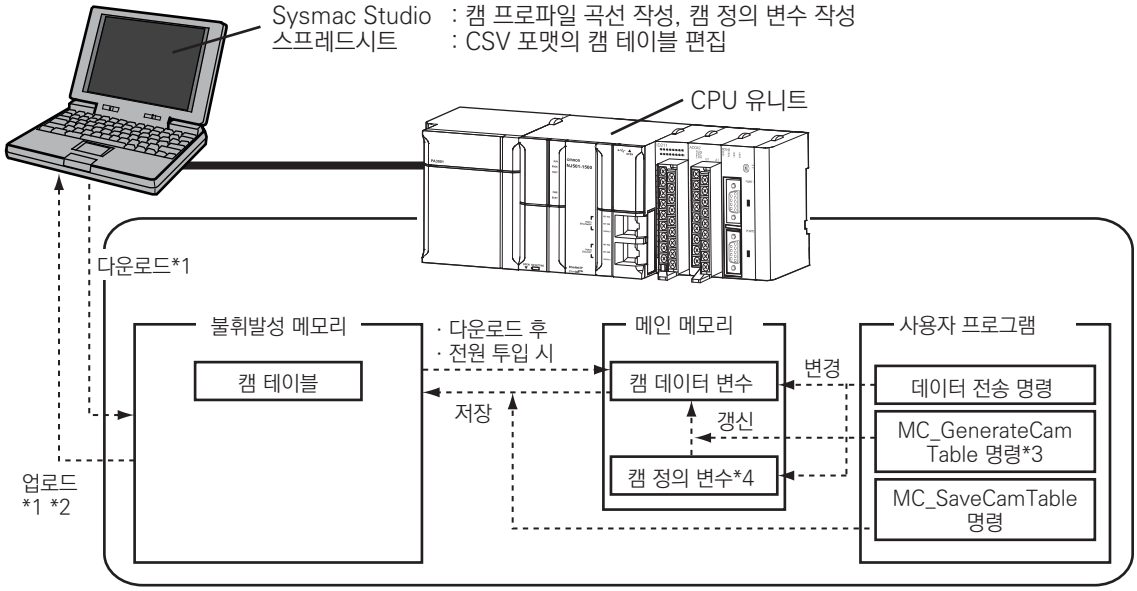
변수명	데이터형	명칭	기능
Cfg	_sGROUP_REF_CFG	축 그룹 기본 설정	축 그룹 기본 설정의 설정 내용을 나타냅니다.
GrpNo	UINT	축 그룹 번호	축 그룹의 논리 번호를 나타냅니다.
GrpEnable	_eMC_GROUP_USE	축 그룹 사용	축 그룹의 사용 여부를 나타냅니다. 0: _mcNoneGroup(미작성 축 그룹) 1: _mcUnusedGroup(미사용 축 그룹) 2: _mcUsedGroup(사용 축 그룹)
ExecID *3	UINT	실행 ID	할당한 태스크별 실행 ID를 나타냅니다. 0: 태스크 할당 없음(미작성 그룹) 1: 프라이머리 정주기 태스크에 할당 2: 정주기 태스크(실행 우선도 5)에 할당
Kinematics	_sGROUP_REF_KIM	운동학 변환 설정	축 그룹의 운동학 변환을 위한 정의를 나타냅니다.
GrpType	_eMC_TYPE	축 구성	다축 협조 제어의 축 구성을 나타냅니다. 0: _mcXY(2축) 1: _mcXYZ(3축) 2: _mcXYZU(4축)
Axis[0]	UINT	구성 축(축 A0)	축 A0에 할당된 축 번호를 나타냅니다.
Axis[1]	UINT	구성 축(축 A1)	축 A1에 할당된 축 번호를 나타냅니다.
Axis[2]	UINT	구성 축(축 A2)	축 A2에 할당된 축 번호를 나타냅니다.
Axis[3]	UINT	구성 축(축 A3)	축 A3에 할당된 축 번호를 나타냅니다.

*1. 속도「0」에서의 동작도 연산 중인 상태에 포함됩니다.
*2. 축 그룹 내의 모든 구성 축이 각 축 파라메터의 인포지션 폭 범위 내에 들어가면 FALSE가 됩니다.
*3. 유니트 버전 Ver.1.10 이상인 CPU 유니트와 Sysmac Studio Ver.1.12 이상의 조합으로 사용할 수 있습니다.

6-7 캠 테이블과 캠 데이터 변수

MC 기능 모듈에서는 Sysmac Studio의 캠 에디터 기능으로 작성한 캠 프로파일 곡선을 캠 테이블로 사용할 수 있습니다.
NJ/NX 시리즈의 사용자 프로그램에서는 캠 테이블의 데이터를 캠 데이터 변수로 취급합니다.

캠 테이블의 작성과 저장



*1. 「다운로드」 및 「업로드」는 Sysmac Studio의 「Synchronization」 기능을 사용합니다.
*2. 업로드한 캠 데이터 변수는 캠 에디터 기능으로 편집할 수 없습니다. 사용자 프로그램으로 편집한 캠 데이터 변수를 PC에서 재편집하는 방법은 다음 페이지를 참조해 주십시오.
또한, Sysmac Studio로 리빌드하거나 캠 프로파일 곡선을 편집하면 업로드한 캠 데이터 변수는 캠 프로파일 곡선 데이터로 덮어쓰기됩니다.
*3. MC_GenerateCamTable(캠 테이블 생성) 명령은 유니트 버전 Ver.1.08 이상인 CPU 유니트와 Sysmac Studio Ver.1.09 이상의 조합으로 사용할 수 있습니다.
*4. 캠 정의 변수는 캠 속성 변수와 캠 노드 변수의 총칭입니다.

● 캠 테이블의 데이터 플로우

- Sysmac Studio에서 작성한 캠 프로파일 곡선을 CPU 유니트에 다운로드하면 캠 테이블로 CPU 유니트에 내장된 불휘발성 메모리에 저장됩니다. 또한, Sysmac Studio에 캠 테이블을 업로드하면 불휘발성 메모리에 저장된 캠 테이블이 업로드됩니다.
- 불휘발성 메모리에 저장된 캠 테이블은 다운로드 후나 전원 투입 시에 캠 데이터 변수로 메인 메모리에 반영됩니다.
- 사용자 프로그램을 이용하여 메인 메모리의 캠 데이터 변수 및 캠 정의 변수를 편집할 수 있습니다. 캠 데이터 변수 및 캠 정의 변수에 대해서는 「9-2-5 캠 테이블 (P.9-15)」을 참조해 주십시오.
- 사용자 프로그램 속 MC_GenerateCamTable(캠 테이블 생성) 명령은 캠 정의 변수값에 따라 메인 메모리의 캠 데이터 변수를 갱신할 수 있습니다.
- 모션 제어 명령의 MC_SaveCamTable(캠 테이블 저장) 명령은 메인 메모리의 캠 데이터 변수를 불휘발성 메모리에 저장합니다.
- Sysmac Studio의 캠 데이터 설정에서 작성한 캠 정의 변수는 사용자 프로그램에서 변경한 경우에도 업로드할 수 있습니다. 사용자 정의 변수로 작성한 캠 정의 변수는 사용자 프로그램에서 변경한 경우에는 업로드할 수 없습니다.

- Sysmac Studio에서 캠 정의 변수를 작성, 전송하는 내용에 대해서는 🕮『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.
- MC_GenerateCamTable(캠 테이블 생성) 명령 및 MC_SaveCamTable(캠 테이블 저장) 명령에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.
- 캠 테이블의 다운로드 및 업로드는 CPU 유니트 모드나 MC 기능 모듈의 상태와 관계없이 실행할 수 있습니다. 단, 캠 테이블 저장 중에는 캠 데이터의 다운로드 및 업로드 외에 온라인 시작 조작 및 온라인 에디트, 데이터 트레이스 시작 조작 등을 사용할 수 없습니다. 또한, 온라인 에디트 등을 실행 중인 경우에는 MC_SaveCamTable(캠 테이블 저장) 명령은 사용할 수 없습니다.
- 다운로드를 시작하면 동작 중인 축은 최대 감속도로 감속합니다.

사용상의 주의

사용자 프로그램에서 변경한 캠 데이터는 전원을 재투입하거나 Sysmac Studio에서 다운로드하여 불휘발성 메모리의 캠 테이블 내용으로 되돌아갑니다. 또한, Sysmac Studio에서는 이러한 데이터를 업로드할 수 없습니다.

버전 관련 정보

유니트 버전 Ver.1.13 이상인 CPU 유니트와 Sysmac Studio Ver.1.17 이상을 조합하면 다운로드 중에도 I/O 디바이스로 지령을 계속 보낼 수 있습니다.
유니트 버전 Ver.1.12 이하인 CPU 유니트의 경우는 다운로드 중에 I/O 디바이스로 보내는 지령을 정지합니다.
다운로드 시작 시에 I/O 디바이스로 보내는 지령을 정지할지 계속 보낼지에 대한 설정은『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

I/O 디바이스로 보내는 지령을 정지할 경우

다운로드 중에는 서보 OFF가 되고 축 상태는「Disable(축 무효)」이 됩니다.

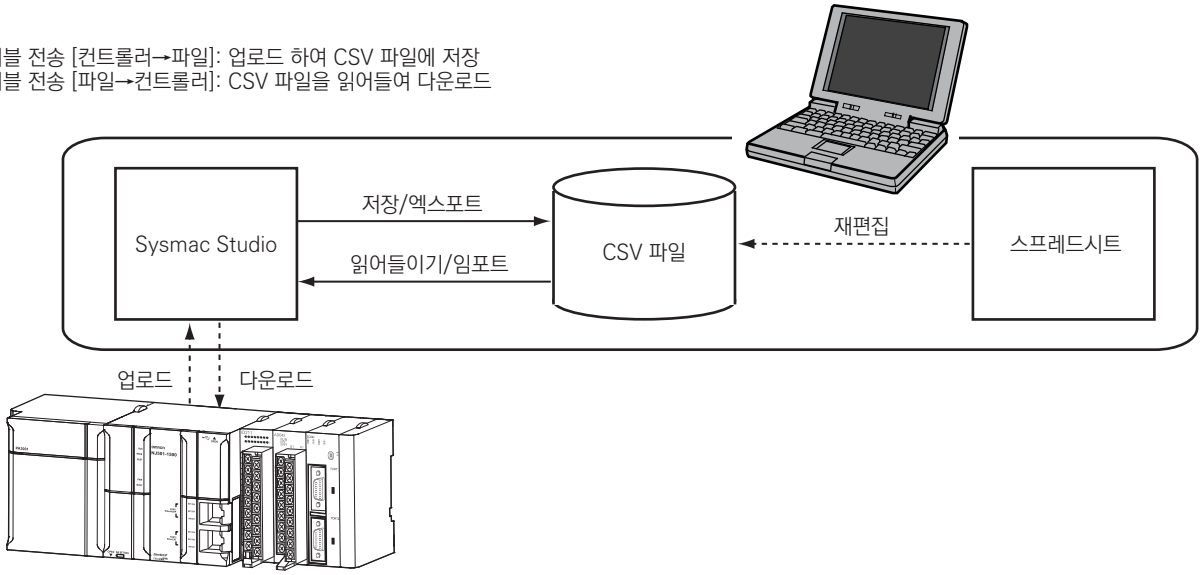
I/O 디바이스로 지령을 계속 보낼 경우

MC_Power(운전 가능) 명령 및 MC_SetTorqueLimit(토크 제어) 명령은 다운로드 중에도 동작 가능합니다.
다운로드 중에는 다운로드 직전의 MC_Power 명령을 이용한 서보 ON 상태 및 MC_SetTorqueLimit 명령을 이용한 출력 토크 제한을 계속합니다.
또한, 다운로드를 통해 사용자 프로그램이 갱신되어 MC_Power 명령 및 MC_SetTorqueLimit 명령이 삭제된 경우에도 서보 ON 상태 및 토크 제한을 계속합니다.

사용자 프로그램으로 편집한 캠 데이터 변수를 PC로 재편집하기

사용자 프로그램에서 캠 데이터 변수를 편집 또는 갱신하고 MC_SaveCamTable(캠 테이블 저장) 명령으로 불휘발성 메모리에 저장한 캠 테이블은 Sysmac Studio의 캠 에디터 기능으로 데이터를 재편집할 수 있습니다. 여기에서는 스프레드시트 등으로 데이터를 재편집하여 캠 테이블로 사용하는 방법에 대해 설명합니다.

캠 테이블 전송 [컨트롤러→파일]: 업로드 하여 CSV 파일에 저장
캠 테이블 전송 [파일→컨트롤러]: CSV 파일을 읽어들여 다운로드



*1. 「다운로드」 및 「업로드」는 Sysmac Studio의 「Synchronization」 기능을 사용합니다.

● 불휘발성 메모리에 저장한 캠 테이블을 CSV 파일로 저장하기

- Sysmac Studio의 [Cam Data Settings]에서 재편집할 캠 프로파일을 우클릭하고 메뉴에서 [Transfer Cam Table[Controller to File]]을 선택합니다.
- [Save] 대화 상자에서 파일 저장처와 파일명을 지정하고 [Save] 버튼을 클릭합니다.

● CSV 파일 재편집하기

- 스프레드시트 등을 사용하여 데이터를 편집하고 CSV 파일로 저장합니다.

● CSV 파일을 CPU 유니트로 전송하기

- 다운로드할 캠 프로파일을 우클릭하고 메뉴에서 [Transfer Cam Table[File to Controller]]를 선택합니다.
- [Open File] 대화 상자에서 전송할 파일명을 지정하고 [OK] 버튼을 클릭합니다.
- 전송한 캠 테이블을 유효화하려면 전송 완료 후에 컨트롤러를 리셋하거나 컨트롤러의 전원을 다시 투입해 주십시오.

사용상의 주의

- [Transfer Cam Table[Controller to File]]을 실행하기 전에 컨트롤러와 동기해 주십시오.
- [Transfer Cam Table[Controller to File]]로 캠 테이블을 전송한 후에 동기 기능으로 캠 데이터 설정을 컨트롤러로 전송하면 캠 데이터 설정 데이터로 변경됩니다.
  이때 다시 [Transfer Cam Table[Controller to File]]로 전송하거나 캠 데이터 설정을 동기 대상 외로 해 주십시오.

캠 에디터에서 작성한 캠 데이터 설정을 CSV 파일로 엑스포트할 수도 있습니다.
캠 데이터 설정과 조작 방법에 대한 상세한 내용은 📖『Sysmac Studio Version 1 오퍼레이션 매뉴얼 (SBCA-470)』을 참조해 주십시오.

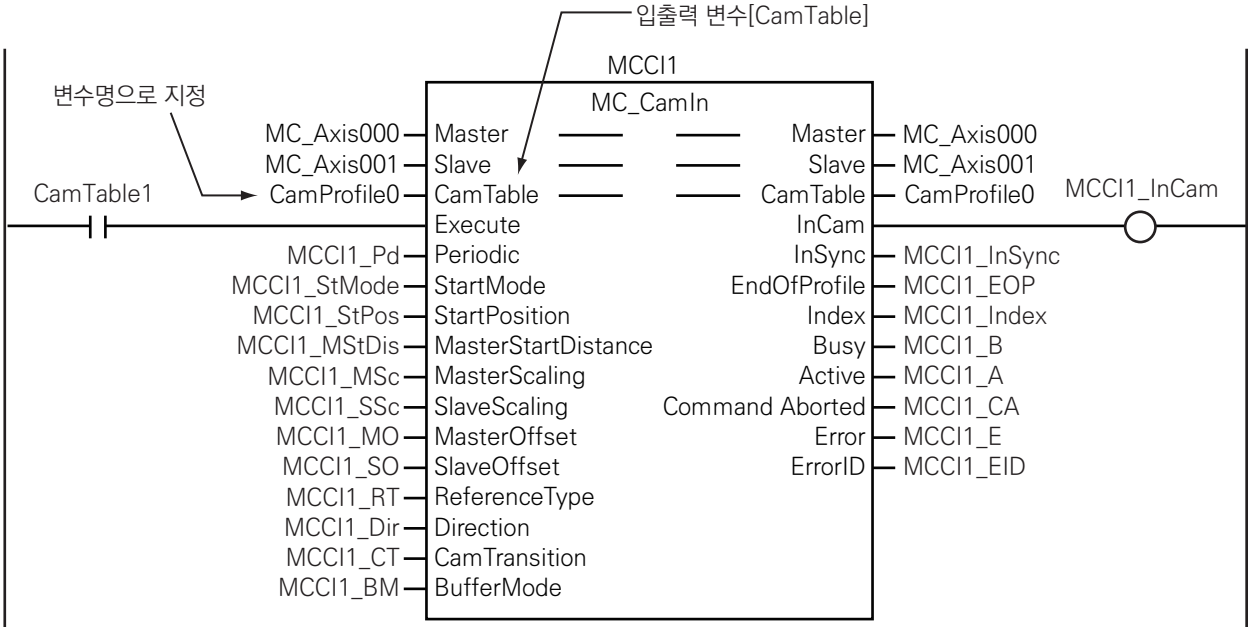
캠 프로파일 곡선의 이름

Sysmac Studio에서 캠 프로파일을 추가하면「CamProfile0」이라는 이름이 디폴트로 설정됩니다.
뒷부분 숫자에는 캠 프로파일을 추가할 때마다 추가 번호가 설정됩니다.
이러한 이름은 Sysmac Studio에서 캠 프로파일마다 임의의 이름으로 변경할 수 있습니다.

여기에서 설정한 캠 프로파일의 이름이 캠 테이블명으로 사용됩니다.

사용자 프로그램에서 지정하는 방법

사용자 프로그램에서는 모션 제어 명령의 입출력 변수「CamTable」에 캠 테이블명을 지정합니다.



6-8 모션 제어 프로그램의 작성 방법

NJ/NX 시리즈의 사용자 프로그램 안에 모션 제어 명령을 기술하면 모션 제어를 실행할 수 있습니다. 모션 제어 명령을 기술한 프로그램을 모션 제어 프로그램이라 합니다.

사용상의 주의

- Sysmac Studio에서는 어떠한 형식의 CPU 유니트라도 최대 256축의 설정과 프로그래밍이 가능합니다. 단, CPU 형식의 최대 제어 축 수를 초과하는 프로젝트는 CPU 유니트에 다운로드 할 수 없습니다.
- 프로젝트를 유용하는 경우 등에는 CPU 유니트의 형식을 확인하고 최대 제어 축 수를 초과하지 않도록 해 주십시오.
- 축 사용이 [Unused axes]인 축도 제어 축 수의 대상입니다.

프로그래밍에 대한 상세한 내용은 『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편 (SBCA-467)』을 참조해 주십시오.

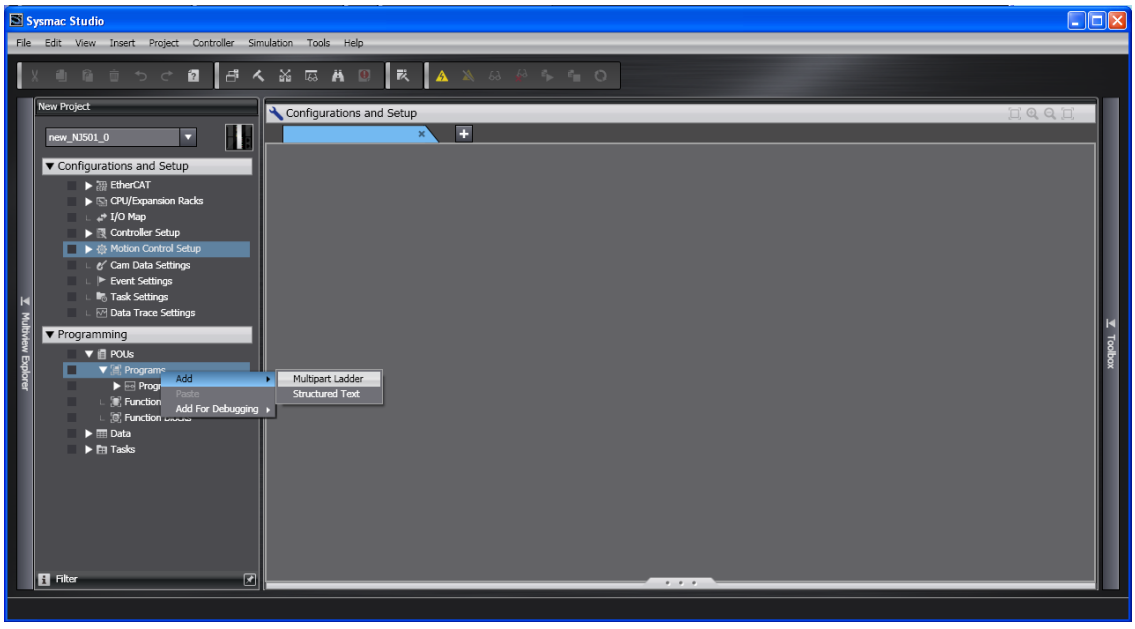
여기에서는 Sysmac Studio를 사용하여 기존의 프로젝트에 프로그램을 추가하는 순서에 대해 설명합니다.

1 **Sysmac Studio의 기동**

Sysmac Studio를 기동시켜 기존 프로젝트를 엽니다.

2 **프로그램 추가**

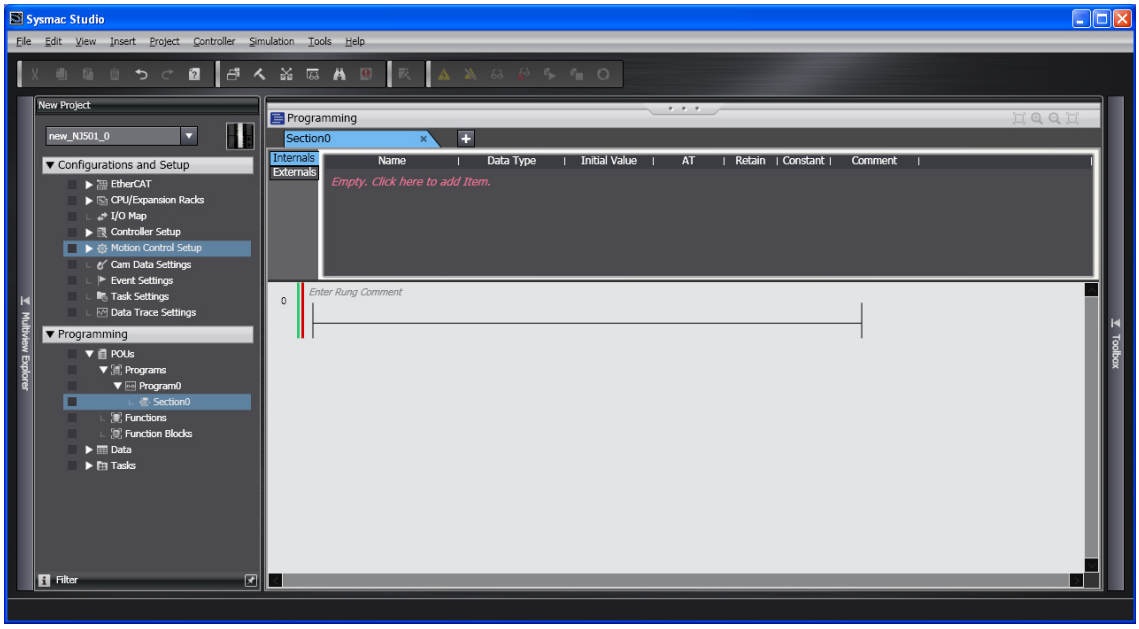
멀티 뷰 익스플로러의 [Programs]를 우클릭하고 [Add]에서 [Multipart Ladder] 또는 [Structured Text]를 클릭합니다.



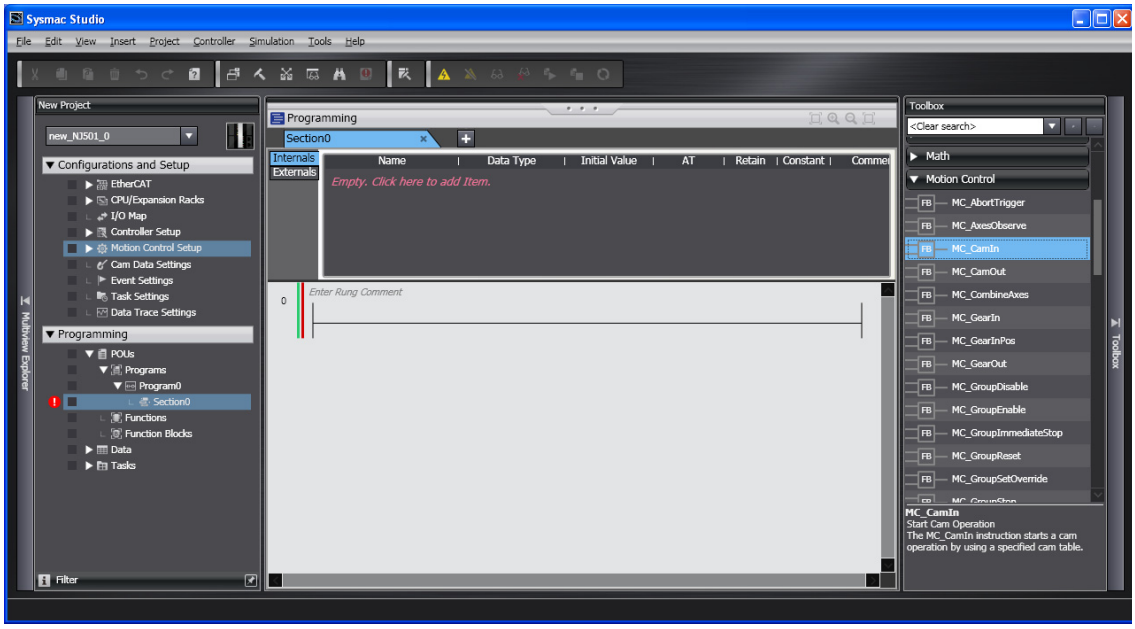
멀티 뷰 익스플로러에 프로그램이 추가됩니다.

3 프로그램 편집

추가된 프로그램 아래에 있는 섹션을 우클릭하고 [Edit]를 클릭하면 프로그램 편집 화면이 열립니다.



도구 상자에서 명령을 선택하고 필요한 프로그램을 작성해 주십시오.



프로그래밍에 대한 상세한 내용은 🕮『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편 (SBCA-467)』을 참조해 주십시오.

조작 방법에 대한 상세한 내용은 🕮『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.

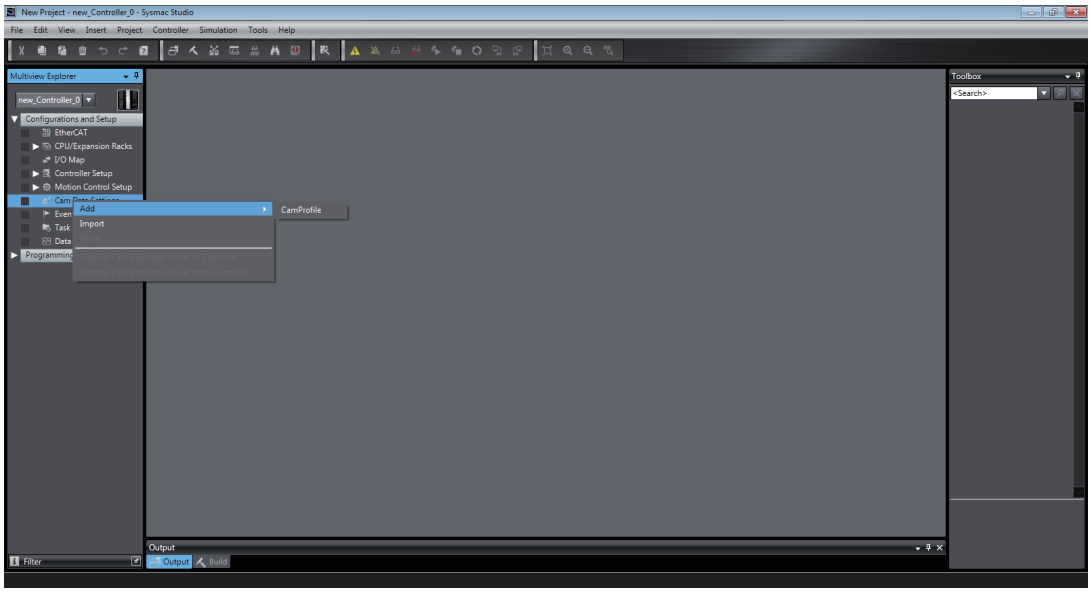
6-9 캠 테이블의 작성 방법

여기에서는 Sysmac Studio의 캠 에디터 기능을 사용하여 캠 테이블을 작성하는 순서에 대해 설명합니다.

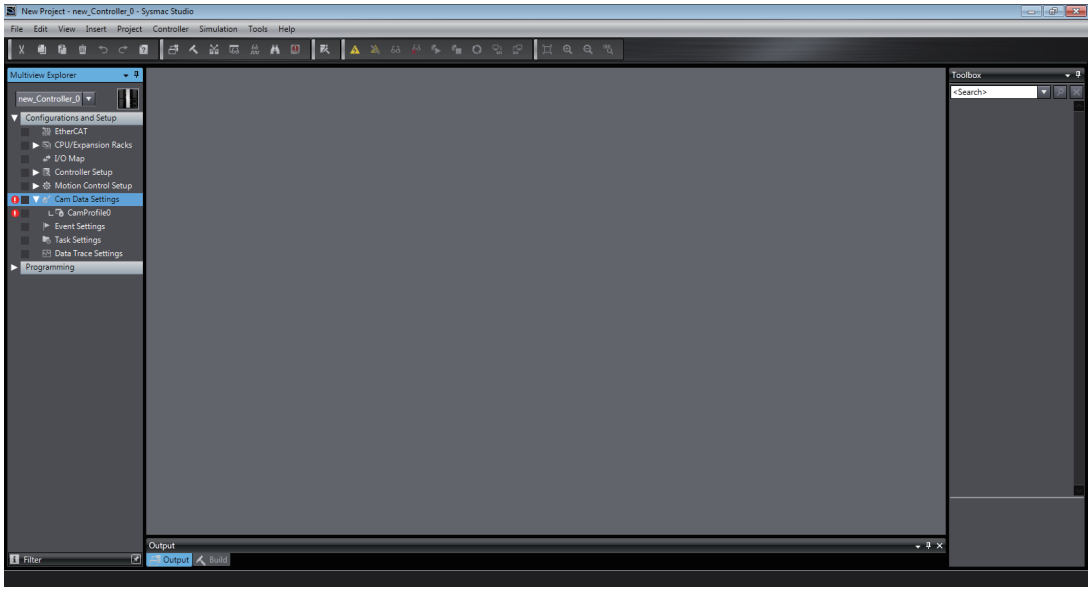
캠 에디터 기능에 대한 상세한 내용은 『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.

1 캠 프로파일 추가

멀티 뷰 익스플로러의 [Cam Data Settings]를 우클릭하고 [Add]에서 [CamProfile]을 클릭합니다.

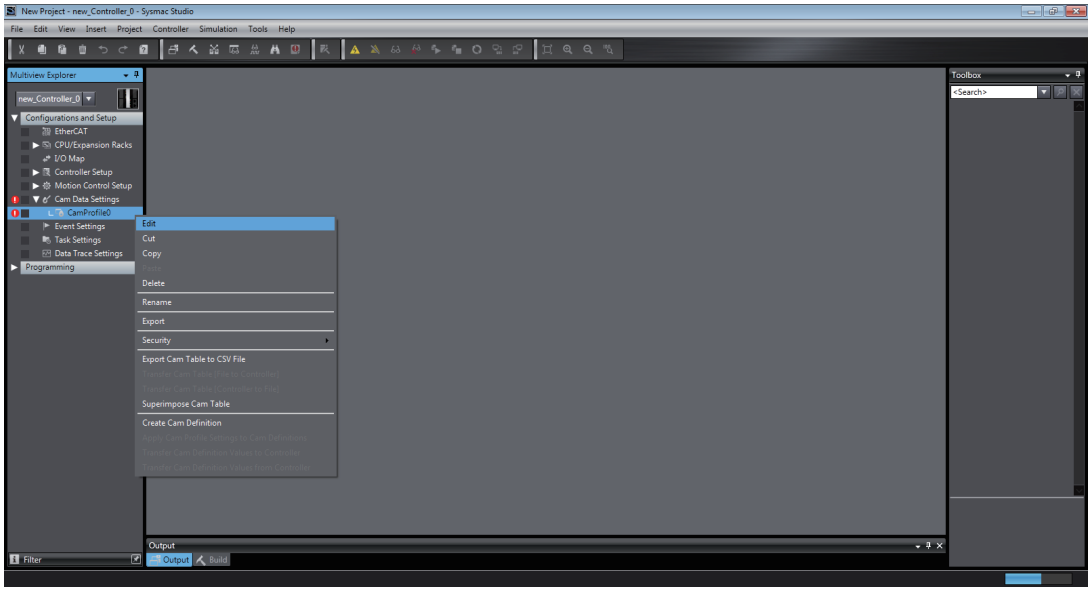


멀티 뷰 익스플로러에 캠 프로파일이 추가됩니다. 추가된 프로파일의 이름은 디폴트인 「CamProfile0」에서 임의의 이름으로 변경할 수 있습니다.

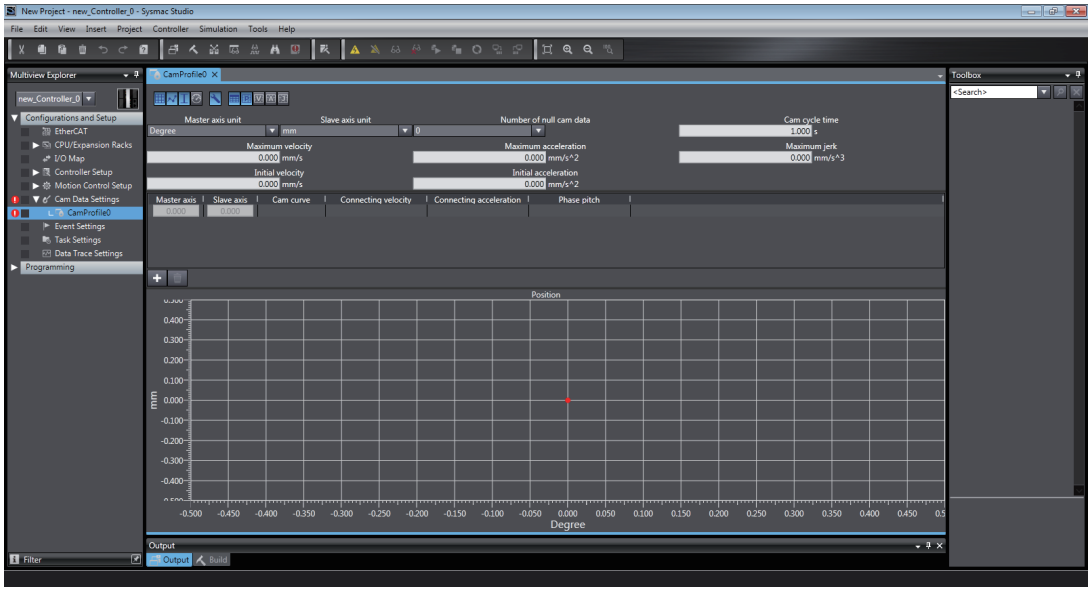


2 캠 프로파일 편집

멀티 뷰 익스플로러에 추가된 캠 프로파일을 우클릭하고 [Edit]를 클릭합니다.



캠 프로파일의 편집 화면이 표시됩니다.



각 항목을 설정하여 캠 프로파일을 편집해 주십시오.

조작 방법에 대한 상세한 내용은 📖『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.

7

수동 운전

오므론 서보 드라이버 1S 시리즈와의 조합으로 수동 조작을 위한 기능에 대해 설명합니다.

7-1 개요 7-2
7-2 서보 ON 7-3
7-2-1 서보 ON 순서 7-4
7-2-2 축 파라메터의 설정 7-4
7-2-3 프로그램 예 7-5
7-3 조그 전송 7-6
7-3-1 조그 전송 순서 7-6
7-3-2 축 파라메터의 설정 7-7
7-3-3 입력 변수의 설정 예 7-7
7-3-4 프로그램 예 7-8

7-1 개요

MC 기능 모듈과 오므론 서보 드라이버 1S 시리즈와의 조합으로 사용자 프로그램에서 모션 제어 명령을 사용하여 수동 운전을 실행하는 경우에 대해 설명합니다.

수동 운전 모션 제어 명령은 서보 드라이버를 서보 ON 상태로 하는 「MC_Power(운전 가능)」 명령과 조그 전송을 실행하는 「MC_MoveJog(조그 전송)」 명령입니다.

사용상의 주의

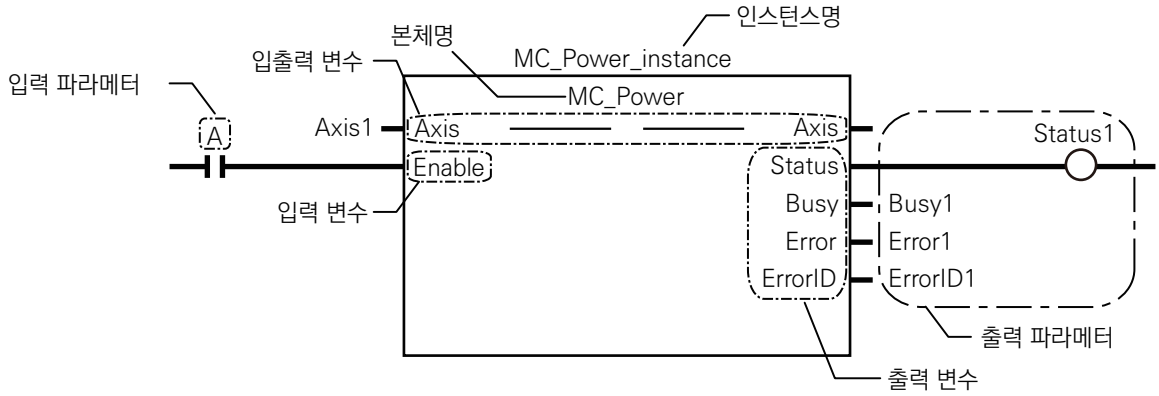
수동 운전을 실행하려면 축 설정이 필요합니다. 축의 설정 방법에 대해서는 「제3장 축과 축 그룹의 설정」을 설정해 주십시오.

참고

- 프로그래밍 없이 수동 운전을 실행하려면 Sysmac Studio를 사용해 주십시오.
  Sysmac Studio를 사용한 수동 운전에 대해서는 「4-3 모터 동작 확인 (P.4-8)」을 참조해 주십시오.
- 사용자 프로그램의 작성 방법은 「제6장 모션 제어 프로그램」을 참조해 주십시오.
- 오므론 서보 드라이버 G5 시리즈를 사용하는 경우에도 순서와 조작은 동일합니다.

7-2 서보 ON

서보 드라이버에 대한 운전 지령을 ON/OFF하는 기능입니다.
모션 제어 명령의 MC_Power(운전 가능) 명령을 사용합니다.



동작시키고 싶은 축은 입출력 변수인 Axis(축)로 지정합니다.
MC_Power(운전 가능) 명령의 입력 변수인 Enable(유효)을 TRUE로 하면 서보 ON 상태가 됩니다. Enable(유효)을 FALSE로 하면 서보 OFF 상태가 됩니다.

사용상의 주의

- Enable(유효)을 축 동작 중에 FALSE로 한 경우는 지령을 즉시 정지시켜 그 축에 대한 모션 제어 명령이 무효해집니다.
- NX 시리즈 펄스 출력 유니트를 사용할 때는 모터 드라이버의 통전 ON/OFF를 별도로 실행해야 합니다. 상세한 내용은 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

참고

오므론 서보 모터 1S 시리즈 및 G5 시리즈의 절대값 엔코더 부속 서보 모터 또는 오므론 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입에서 절대값 타입의 외부 스케일을 사용할 경우 MC_Power(운전 가능) 명령의 입력 변수 Enable(유효)을 TRUE로 하면 원점을 확정합니다.

버전 관련 정보

유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 오므론 서보 모터 1S 시리즈 및 G5 시리즈의 절대값 엔코더 부속 서보 모터 또는 오므론 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입에서 절대값 타입의 외부 스케일을 사용할 경우 상기「참고」내용과 함께 EtherCAT의 프로세스 데이터 통신이 확립되지 않은 상태에서 확립된 상태로 바뀐 경우에도 원점을 확정합니다.

7-2-1 서보 ON 순서

1 **축의 추가 및 설정**

Sysmac Studio에서 축을 추가, 설정합니다.

상세한 내용에 대해서는 「3-2-2 설정 순서 (P.3-9)」를 참조해 주십시오.

2 **축 파라메터 설정**

Sysmac Studio에서 축 파라메터를 설정합니다.

상세한 내용에 대해서는 「3-2-2 설정 순서 (P.3-9)」를 참조해 주십시오.

3 **사용자 프로그램 작성**

Sysmac Studio에서 사용자 프로그램을 작성합니다.

상세한 내용에 대해서는 「6-8 모션 제어 프로그램의 작성 방법 (P.6-35)」를 참조해 주십시오.

4 **축 파라메터와 사용자 프로그램 다운로드**

설정한 축 파라메터와 사용자 프로그램을 CPU 유니트로 다운로드합니다.

CPU 유니트에 다운로드하는 경우는 Sysmac Studio의「Synchronization」기능을 사용합니다.

상세한 내용에 대해서는 「3-2-2 설정 순서 (P.3-9)」를 참조해 주십시오.

5 **사용자 프로그램 기동**

사용자 프로그램을 기동하고「MC_Power」명령의 입력 변수「Enable」을 TRUE로 합니다.

서보 드라이버가 서보 ON 상태가 됩니다.

7-2-2 축 파라메터의 설정

서보 ON 상태로만 바꾸려면 아래와 같이 축 파라메터의 설정 항목을 설정해야 합니다.

아래 표는 설정 예입니다.

설정 항목	설정 내용
축 변수명	Axis1 *1
축 번호	1 *2
축 사용	사용 축
축 종별	서보 축
입력 디바이스/출력 디바이스	1 *3

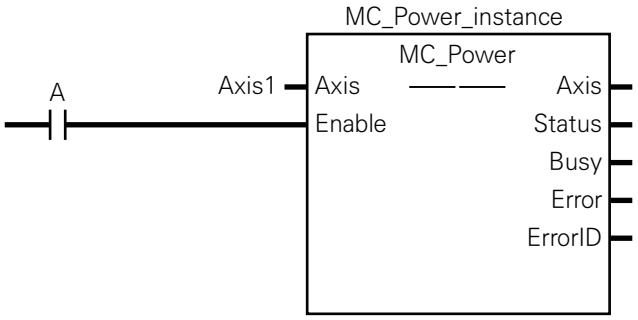
*1. 축이 여러 개인 경우는 축마다 다른 변수명을 설정해 주십시오.

*2. 축이 여러 개인 경우는 축마다 다른 값을 설정해 주십시오.

*3. 서보 드라이버의 노드 주소와 같은 값을 설정해 주십시오. 축이 여러 개인 경우는 축마다 다른 값을 설정해 주십시오.

7-2-3 프로그램 예

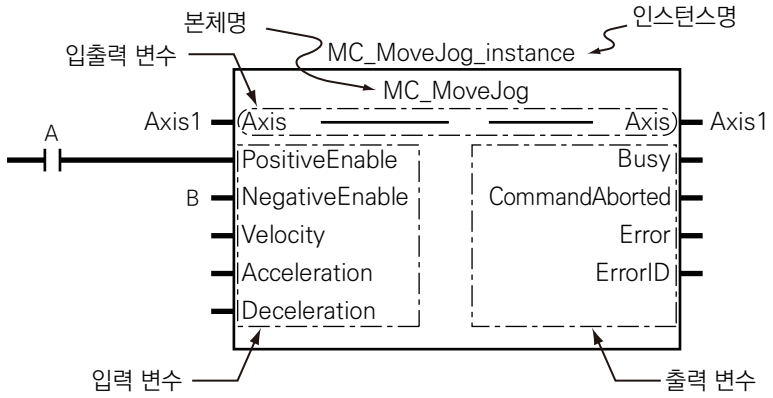
접점 A에 의해 Axis 1이라는 축의 서보 ON/OFF를 변환하는 샘플 프로그램은 아래와 같습니다.



MC_Power(운전 가능) 명령에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편 (SBCE-364)』을 참조해 주십시오.

7-3 조그 전송

조그 전송은 모션 제어 명령의 MC_MoveJog(Jog 전송) 명령을 사용합니다.



조그 전송하고 싶은 축은 입출력 변수 Axis(축)로 지정합니다.
입력 변수의 PositiveEnable을 TRUE로 하면 정방향으로 지정한 Velocity(목표 속도), Acceleration(가속도)으로 축을 기동합니다. PositiveEnable을 FALSE로 하면 지정한 Deceleration(감속도)으로 감속하여 정지합니다.
마찬가지로 입력 변수 NegativeEnable을 TRUE로 하면 역방향으로 기동하고 FALSE에서 정지합니다.
조그 전송은 원점이 확정되지 않은 상태에서도 실행할 수 있습니다.

7-3-1 조그 전송 순서

1 **축의 추가 및 설정**
Sysmac Studio에서 축을 추가, 설정합니다.
상세한 내용에 대해서는 「3-2-2 설정 순서 (P.3-9)」를 참조해 주십시오.

2 **축 파라메터 설정**
Sysmac Studio에서 축 파라메터를 설정합니다.
상세한 내용에 대해서는 「3-2-2 설정 순서 (P.3-9)」를 참조해 주십시오.

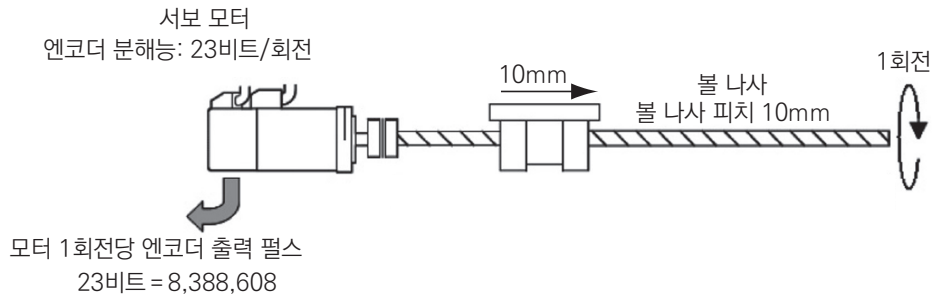
3 **사용자 프로그램 작성**
Sysmac Studio에서 사용자 프로그램을 작성합니다.
상세한 내용에 대해서는 「6-8 모션 제어 프로그램의 작성 방법 (P.6-35)」를 참조해 주십시오.

4 **축 파라메터와 사용자 프로그램 다운로드**
설정한 축 파라메터와 사용자 프로그램을 CPU 유니트로 다운로드합니다.
CPU 유니트에 다운로드하는 경우는 Sysmac Studio의「Synchronization」기능을 사용합니다.
상세한 내용에 대해서는 「3-2-2 설정 순서 (P.3-9)」를 참조해 주십시오.

5 **사용자 프로그램 기동**
사용자 프로그램을 기동하여 MC_Power(운전 가능) 명령의 입력 변수「Enable」을 TRUE로 하고 서보 드라이버를 서보 ON 상태로 합니다.
MC_MoveJog(Jog 전송) 명령의 입력 변수「PositiveEnable」또는「NegativeEnable」을 TRUE로 하고 조그 전송을 실행합니다.

7-3-2 축 파라메터의 설정

원점이 확정되지 않은 상태에서 조그 전송을 실행할 경우 다음과 같은 축 파라메터를 설정해 주십시오. 1축 장비를 예로 들어 설정 예를 설명합니다.



설정 항목	설정 내용
축 변수명	Axis1 *1
축 번호	1 *2
축 사용	사용 축
축 종별	서보 축
입력 디바이스/출력 디바이스	1 *3
모터 1회전의 펄스 수	8,388,608 *4
모터 1회전의 이동량	10,000 *4
소프트웨어 리미트 기능	피드백 위치에 대해 유효
표시 단위	μm
카운트 모드	리니어 모드
최고 속도	500,000 *5
JOG 최고 속도	50,000 *6
최대 가속도	5,000,000 *7
최대 감속도	5,000,000 *7

*1. 축이 여러 개인 경우는 축마다 다른 변수명을 설정해 주십시오.
*2. 축이 여러 개인 경우는 축마다 다른 값을 설정해 주십시오.
*3. 서보 드라이버의 노드 주소와 같은 값을 설정해 주십시오. 축이 여러 개인 경우는 축마다 다른 값을 설정해 주십시오.
*4. 위치의 지령 단위는 1(μm)입니다.
*5. 최고 속도는 3,000r/min = 30m/min = 0.5m/s = 500,000μm/s로 합니다.
*6. 조그 최고 속도는 최고 속도의 10%인 0.05m/s = 50,000μm/s로 합니다.
*7. 최대 가속도 및 최대 감속도는 5m/s^{2}입니다. 최고 속도 3,000r/min까지의 가속 시간을 0.1s로 합니다.

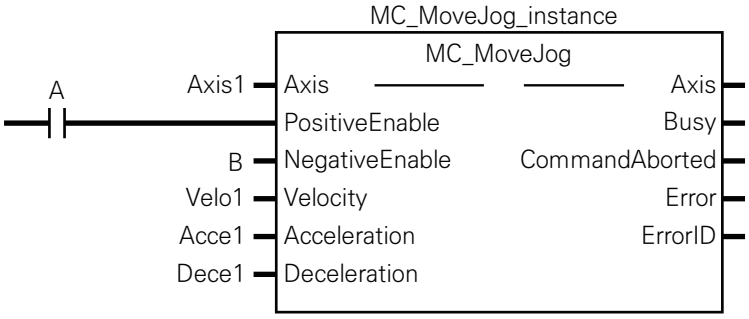
7-3-3 입력 변수의 설정 예

여기에서는 「MC_MoveJog(조그 전송)」 명령의 입력 변수 「Velocity(목표 속도)」, 「Acceleration(가속도)」, 「Deceleration(감속도)」의 설정에 대해 설명합니다.

- 예를 들어 0.03m/s(30,000μm/s) 속도로 조그를 전송하려면 「Velocity」에 30,000을 지정합니다.
- 또한, 3m/s^2 (3,000,000μm/s^2)으로 가감속하려면 「Acceleration」과 「Deceleration」에 3,000,000을 지정합니다.

7-3-4 프로그램 예

접점 A에 의해 Axis 1이라는 축을 정방향으로, 접점 B에 의해 역방향으로 조그를 전송시키는 샘플 프로그램은 다음과 같이 됩니다.



이때 Velocity(목표 속도)는 Velo1, Acceleration(가속도)은 Acce1, Deceleration(감속도)은 Dece1이 됩니다.

입력 변수의 설정 예 조건으로 동작시키려면 먼저 사용자 프로그램에서 각 변수값을 설정해 둡니다.

- Velo1 = 30,000
- Acce1 = 3,000,000
- Dece1 = 3,000,000

MC_MoveJog(조그 전송) 명령에 대한 상세한 내용은 『NJ/NX 시리즈 명령 참조 모션편(SBCE-364)』을 참조해 주십시오.

8

원점 결정

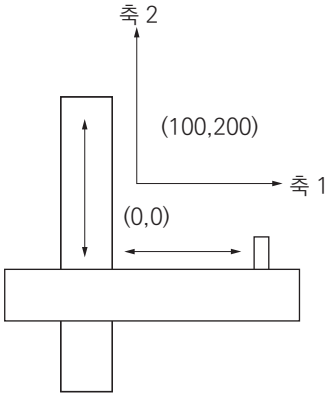
이 장에서는 원점 결정에 대해 설명합니다.

8-1 개요 8-2
8-2 원점 복귀 순서 8-5
8-2-1 원점 복귀 파라메터의 설정 8-6
8-2-2 원점 복귀 모니터 8-12
8-3 원점 복귀 동작 8-13
8-4 절대값 엔코더 원점 설정 8-14
8-4-1 기능 개요 8-15
8-4-2 설정 순서 8-16
8-5 고속 원점 복귀 8-18

8-1 개요

MC 기능 모듈과 오므론 서보 드라이버 G5 시리즈를 조합한 상태에서의 조작에 대해 설명합니다.

위치 결정을 실행하는 시스템에서 시스템의 절대적인 위치에 대해 위치 결정을 실행하는 경우 가장 먼저 원점 결정을 실행해야 합니다.
예를 들어 아래 그림과 같은 XY 스테이지에서 (축 1, 축 2) = (100mm, 200mm)의 위치에 위치 결정을 하고 싶은 경우 기계의 원점(0, 0)을 결정해야 합니다.
이 기계 원점을 결정하는 것을 원점 결정이라 부릅니다.



MC 기능 모듈에서는 기계 원점을 결정하는 방법으로 모션 제어 명령의 「MC_Home(원점 복귀)」 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령을 사용합니다.

명칭	내용
원점 복귀	모터를 실제로 움직여 한계 센서, 원점 근방 센서, 원점 입력 신호를 이용해 기계 원점을 결정합니다. 원점 입력 신호로 근방 센서 및 엔코더의 Z상 등을 사용합니다.

사용상의 주의

- 다음의 경우 원점이 확정되지 않습니다.
  MC_SetPosition(현재 위치 변경) 명령의 실행
  리니어 모드 시의 오버 플로우 및 언더 플로우
  원점 복귀 시작 시
  EtherCAT 통신의 제어 상태가 오퍼레이셔널 상태 이외
- NX 시리즈 위치 인터페이스 유니트를 사용할 경우에는 일부 기능에 제약이 있습니다. 상세한 내용은 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

참고

오므론 서보 모터 1S 시리즈 및 G5 시리즈의 절대값 엔코더 부속 서보 모터 또는 오므론 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입에서 절대값 타입의 외부 스케일을 사용할 경우 MC_Power(운전 가능) 명령의 입력 변수 Enable(유효)을 TRUE로 하면 원점을 확정합니다.

버전 관련 정보

유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 오므론 서보 모터 1S 시리즈 및 G5 시리즈의 절대값 엔코더 부속 서보 모터, G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입에서 절대값 타입의 외부 스케일을 사용할 경우 앞 페이지 「참고」 내용과 함께 EtherCAT의 프로세스 데이터 통신이 확립되지 않은 상태에서 확립된 상태로 바뀐 경우에도 원점을 확정합니다.

또한, 앞의 방법으로 결정한 원점에 위치 결정하는 기능으로는 아래의 MC_MoveZeroPosition(고속 원점 복귀) 명령도 탑재되어 있습니다.

명칭	내용
고속 원점 복귀	절대 좌표인 「0」을 목표 위치로 하여 위치를 결정하고 원점으로 돌아갑니다.

또한, MC 기능 모듈은 원점 미확정 상태에서도 모터를 움직일 수 있습니다(고속 원점 복귀 제외).

기능	동작
조그 전송, 속도 제어	원점이 확정되지 않은 상태인 경우에는 전원 투입 시를 「0」으로 하여 이동합니다.
고속 원점 복귀	사용할 수 없습니다. 사용한 경우 명령 이상이 발생합니다.
위치 결정	원점이 확정되지 않은 상태인 경우에는 전원 투입 시를 「0」으로 하여 이동합니다.
인터럽트 피딩 위치 결정	
캠 동작 시작	
기어 동작 시작	
사다리꼴 패턴 캠	
가감산 위치 결정	
토크 제어	
구역 감시	
직선 보간	사용할 수 없습니다. 사용한 경우 명령 이상이 발생합니다.
원호 보간	

참고

원점 미확정 시에는 소프트웨어 리미트 기능은 무효해집니다.

각 조작 또는 이벤트에 대한 원점 확정 상태의 변화를 아래에 설명합니다.

조작 또는 이벤트	변화 조건	원점 확정 상태의 변화
서보 ON 축 유효 시	인크리멘털 엔코더	변화 없음
	절대값 엔코더에서 절대값 데이터 정상 판독 시	원점 확정
	절대값 엔코더에서 절대값 데이터 판독 실패 시	원점 미확정
현재 위치 변경	기동 시	원점 미확정
원점 복귀	기동 시	원점 미확정
	완료 시	원점 확정
오버 플로우/언더 플로우	발생 시	원점 미확정

사용상의 주의

- 가상 축에 대해서는 축 파라메터 원점 복귀 동작의 설정과 관계없이 원점 프리셋으로 원점을 확정합니다.
- MC 기능 모듈은 서보 드라이버의 정회전 측 구동 금지 입력(POT), 역회전 측 구동 금지 입력(NOT), 원점 근방 입력(DEC)을 각각 정방향 한계 입력, 역방향 한계 입력, 원점 근방 입력으로 사용합니다.

  이러한 입력 신호에 관해서는 서보 드라이버를 검출할 수 있는 신호 폭을 확보하여 MC 기능 모듈의 제어 주기 이상이 되도록 해 주십시오.

  입력 신호가 제어 주기 이하인 폭이 될 경우 MC 기능 모듈이 입력 신호를 검출하지 못하고 정상적인 동작을 실행할 수 없게 됩니다.
- 서보 드라이버의 각 입력 신호는 서보 드라이버 측의 파라메터를 설정해야 합니다. 접속할 서보 드라이버의 매뉴얼과 부록을 참조하여 올바르게 설정해 주십시오.

8-2 원점 복귀 순서

원점 복귀를 실행하는 순서에 대해 설명합니다.

1 **축의 추가 및 설정**
Sysmac Studio에서 축을 추가, 설정합니다.

2 **축 파라메터 설정**
원점 복귀 파라메터로 원점 복귀 방향을 설정합니다.

3 **사용자 프로그램 작성**
Sysmac Studio에서 사용자 프로그램을 작성합니다.
상세한 내용에 대해서는 「6-8 모션 제어 프로그램의 작성 방법 (P.6-35)」을 참조해 주십시오.

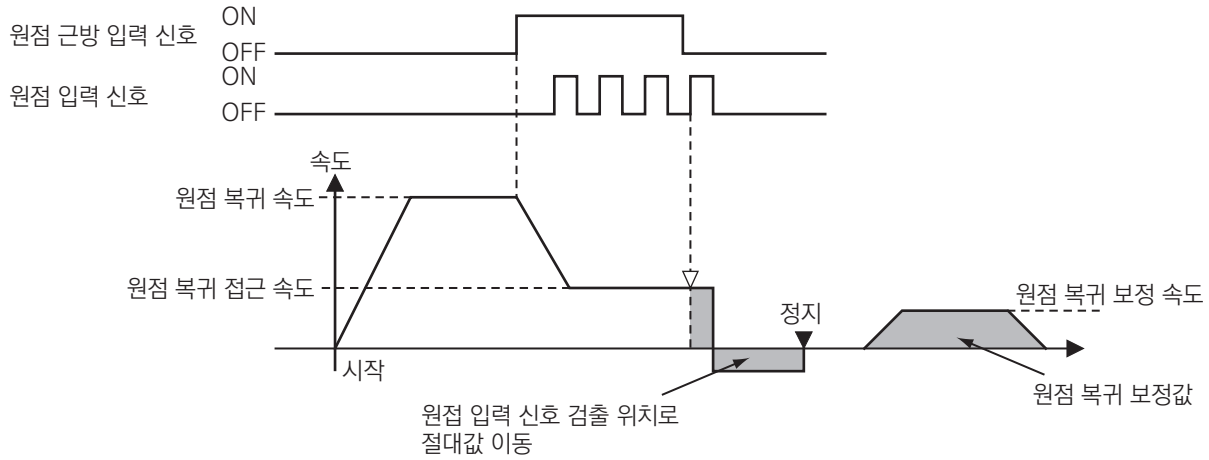
4 **축 파라메터와 사용자 프로그램 다운로드**
설정한 축 파라메터와 사용자 프로그램을 CPU 유니트로 다운로드합니다.
CPU 유니트에 다운로드하는 경우는 Sysmac Studio의「Synchronization」기능을 사용합니다.
상세한 내용에 대해서는 「3-2-2 설정 순서 (P.3-9)」을 참조해 주십시오.

5 **사용자 프로그램 기동**
사용자 프로그램을 기동하여 MC_Power(운전 가능) 명령의 입력 변수「Enable」을 TRUE로 하고 서보 드라이버를 서보 ON 상태로 합니다.
MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령의 입력 변수「Execute」를 TRUE로 하고 원점 복귀를 실행합니다.

8-2-1 원점 복귀 파라메터의 설정

원점 복귀 방법은 원점 복귀 파라메터의 설정값에 따라 지정합니다.
원점 복귀 파라메터는 Sysmac Studio로 설정합니다.

설정	개요
원점 복귀 동작	원점 복귀 동작 방법을 선택합니다.
원점 입력 신호	원점 입력 신호의 입력처를 선택합니다.
원점 복귀 시작 방향	원점 복귀 기동 시의 시작 방향을 설정합니다.
원점 검출 방향	원점 복귀의 원점 입력 검출 방향을 설정합니다.
정방향 한계 입력 시 동작	원점 복귀 중에 정방향 한계 입력이 「ON」한 경우의 정지 방법을 설정합니다.
역방향 한계 입력 시 동작	원점 복귀 중에 역방향 한계 입력이 「ON」한 경우의 정지 방법을 설정합니다.
원점 복귀 속도	원점 복귀 중인 속도를 설정합니다. (단위: 지령 단위/s)
원점 복귀 접근 속도	원점 근방 입력이 「ON」한 후의 속도를 설정합니다. (단위: 지령 단위/s)
원점 복귀 가속도	원점 복귀 시의 가속도를 지정합니다. 「0」을 설정하면 가속하지 않고 원점 복귀 속도에 도달합니다. (단위: 지령 단위/s^2)
원점 복귀 감속도	원점 복귀 시의 감속도를 지정합니다. 「0」을 설정하면 감속하지 않고 제로 속도에 도달합니다. (단위: 지령 단위/s^2)
원점 복귀 저크	원점 복귀 시의 저크를 지정합니다. 「0」을 설정하면 저크 없음이 됩니다. (단위: 지령 단위/s^3)
원점 입력 마스크 거리	원점 복귀 동작 모드가 근방 퇴피 · 원점 입력 마스크 거리 지정인 경우 원점 입력의 마스크 거리를 설정합니다. (단위: 지령 단위)
원점 위치 오프셋	원점 복귀 완료 시에 설정한 값으로 현재 위치를 프리셋합니다. (단위: 지령 단위)
원점 복귀 보류 시간	원점 복귀 동작 모드가 근방 퇴피 · 보류 시간 지정인 경우의 보류 시간을 밀리초 단위로 설정합니다. (단위: ms)
원점 복귀 보정값	원점 복귀의 원점 확정 후 원점 복귀 보정량을 설정합니다. (단위: 지령 단위)
원점 복귀 보정 속도	원점 복귀 보정 시의 속도를 설정합니다. (단위: 지령 단위/s)



각 원점 복귀 파라메터에 대해 아래에 설명합니다.

원점 복귀 동작

기계 원점을 결정하는 동작으로 10종류의 동작 중에서 선택할 수 있습니다.

- 근방 퇴피 · 원점 근방 입력 OFF 지정
- 근방 퇴피 · 원점 근방 입력 ON 지정
- 원점 근방 입력 OFF 지정
- 원점 근방 입력 ON 지정
- 한계 입력 OFF 지정
- 근방 퇴피 · 원점 입력 마스크 거리 지정
- 한계 입력만
- 근방 퇴피 · 보류 시간 지정
- 원점 근방 입력 없음 · 보류 원점 입력 지정
- 원점 프리셋

원점 복귀 동작별로 사용하는 원점 복귀 파라메터는 아래 표와 같습니다.

○: 파라메터를 사용함, ×: 파라메터를 사용하지 않음

	원점 복귀 파라메터														
원점 복귀 동작	원점 입력 신호	원점 복귀 시작 방향	원점 검출 방향	정방향 한계 입력 시 동작	역방향 한계 입력 시 동작	원점 복귀 속도	원점 복귀 접근 속도	원점 복귀 가속도	원점 복귀 감속도	원점 복귀 저크	원점 입력 마스크 거리	원점 위치 오프셋	원점 복귀 보류 시간	원점 복귀 보정값	원점 복귀 보정 속도
근방 퇴피 · 원점 근방 입력 OFF 지정	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
근방 퇴피 · 원점 근방 입력 ON 지정	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
원점 근방 입력 OFF 지정	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
원점 근방 입력 ON 지정	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
한계 입력 OFF 지정	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
근방 퇴피 · 원점 입력 마스크 거리 지정	○	○	○	○	○	○	○	○	○	○	○	○	×	○	○
한계 입력만	×	○	○	○	○	○	○	○	○	○	×	○	×	○	○
근방 퇴피 · 보류 시간 지정	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○
원점 근방 입력 없음 · 보류 원점 입력 지정	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
원점 프리셋	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×

각 원점 복귀 동작에 대한 상세한 내용은 『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

사용상의 주의

오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입은 Z상 입력을 PDO에 매핑할 수 없습니다. PDO에 매핑한 Z상 입력을 사용하는 원점 복귀 동작의「원점 근방 입력 없음 · 보류 원점 입력 지정」을 사용할 때는 원점 입력 신호에 Z상 입력을 선택하지 마십시오.

원점 입력 신호

원점 입력 신호를 사용하는 원점 복귀 동작 모드에서 원점 결정하기 위한 신호를 서보 드라이버의 Z상 신호, 외부 원점 신호 중에서 선택합니다.

사용상의 주의

이 설정에서는 오므론 서보 드라이버 1S 시리즈 또는 G5 시리즈를 접속한 경우에만 원점 입력 신호를 선택할 수 있습니다.

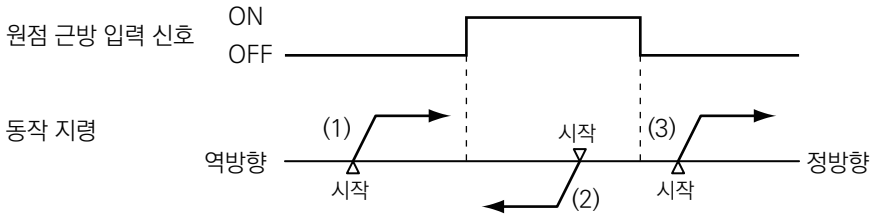
원점 복귀 시작 방향

원점 복귀 기동 시에 움직일 방향을 정방향과 역방향 중에서 선택합니다.
단, 근방 퇴피에 의한 반전 동작을 포함한 원점 복귀 동작 모드에서 원점 근방 신호가 「ON」인 상태로 원점 복귀를 기동한 경우에는 원점 복귀 시작 방향의 설정과 관계없이 원점 검출 방향과 반대 방향으로 동작합니다.

근방 퇴피에 의한 반전 동작을 포함한 원점 복귀 동작 모드는 아래 4가지입니다.

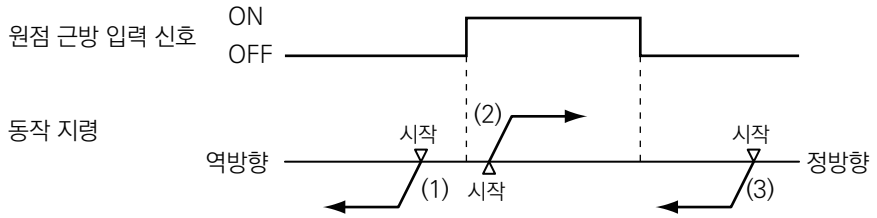
- 0: 근방 퇴피 · 원점 근방 입력 OFF 지정
- 1: 근방 퇴피 · 원점 근방 입력 ON 지정
- 9: 근방 퇴피 · 원점 입력 마스크 거리 지정
- 12: 근방 퇴피 · 보류 시간 지정

원점 복귀 시작 방향: 정방향
원점 검출 방향: 정방향



(1), (3) : 원점 근방 신호 「OFF」 상태에서 시작하므로 원점 복귀 시작 방향으로 동작 시작
(2) : 원점 근방 신호 「ON」 상태에서 시작하므로 원점 검출 방향과 반대 방향으로 동작 시작

원점 복귀 시작 방향: 역방향
원점 검출 방향: 역방향



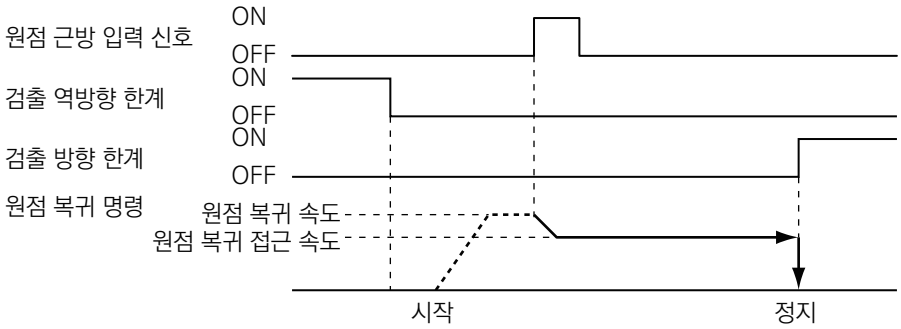
(1), (3) : 원점 근방 신호 「OFF」 상태에서 시작하므로 원점 복귀 시작 방향으로 동작 시작
(2) : 원점 근방 신호 「ON」 상태에서 시작하므로 원점 검출 방향과 반대 방향으로 동작 시작

원점 검출 방향

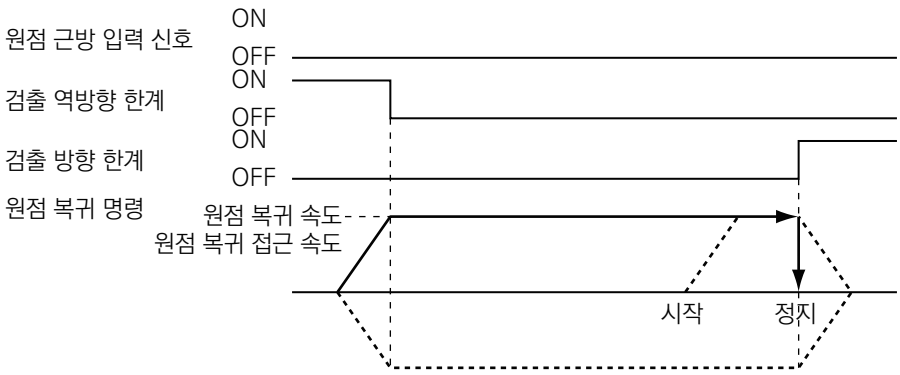
원점 검출하는 방향을 정방향과 역방향 중에서 선택합니다.
원점 검출 방법과 원점 복귀 기동 시에 움직일 방향과의 관계에 대해서는 「원점 복귀 시작 방향 (P.8-8)」을 참조해 주십시오.

정방향 한계 입력 시 동작/역방향 한계 입력 시 동작

- 원점 복귀 중에 동작 방향의 한계 입력까지 도달했을 때 반전해서 원점 복귀를 계속할지 반전시키지 않고 이상 정지할지를 선택합니다. 반전시킬 경우에는 그 정지 방법도 선택합니다.
- 반전하도록 설정한 경우에 원점 복귀 접근 속도에서 원점 검출 방향의 한계 신호가「ON」되면 이상 정지합니다. 단, 근방 신호를 사용하지 않는 원점 복귀 동작 모드 [13: 원점 근방 입력 없음 · 보류 원점 입력 지정]의 경우는 이상 정지되지 않습니다.



- 또한, 양방향의 한계 입력 시 동작을 반전하도록 설정한 경우 원점 검출 방향에 대해 반대 측 한계 입력에서 다른 쪽 한계 입력까지 동작하고 원점이 검출되지 않았을 때는 이상 정지합니다.



원점 복귀 속도

원점 복귀 중인 속도를 지령 단위/s로 설정합니다.

원점 복귀 접근 속도

원점 근방 입력이「ON」한 후의 속도를 지령 단위/s로 설정합니다.

원점 복귀 가속도

원점 복귀 시의 가속도를 지령 단위/s^2로 설정합니다.
「0」을 설정하면 가속하지 않고 원점 복귀 속도 등의 목표로 하는 속도에 도달합니다.

원점 복귀 감속도

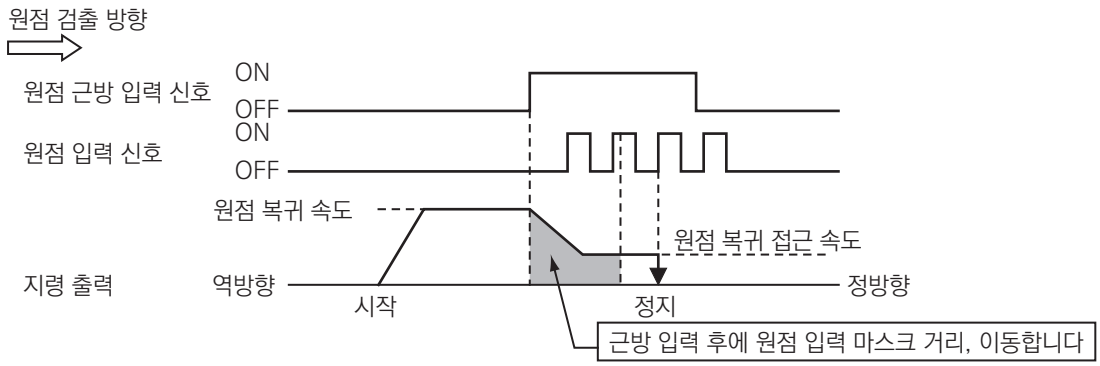
원점 복귀 시의 감속도를 지령 단위/s^2로 설정합니다.
「0」을 설정하면 감속하지 않고 원점 복귀 접근 속도 등의 목표로 하는 속도에 도달합니다.

원점 복귀 저크

원점 복귀 시의 저크를 지령 단위/s^3로 설정합니다.
「0」을 설정하면 저크가 없는 가감속이 됩니다.

원점 입력 마스크 거리

원점 복귀 동작이 [9: 근방 퇴피 · 원점 입력 마스크 거리 지정]인 경우의 원점 입력의 마스크 거리를 지령 단위로 설정합니다.
원점 근방 입력 신호의「OFF」→「ON」에 의한 감속 시작부터의 거리입니다.

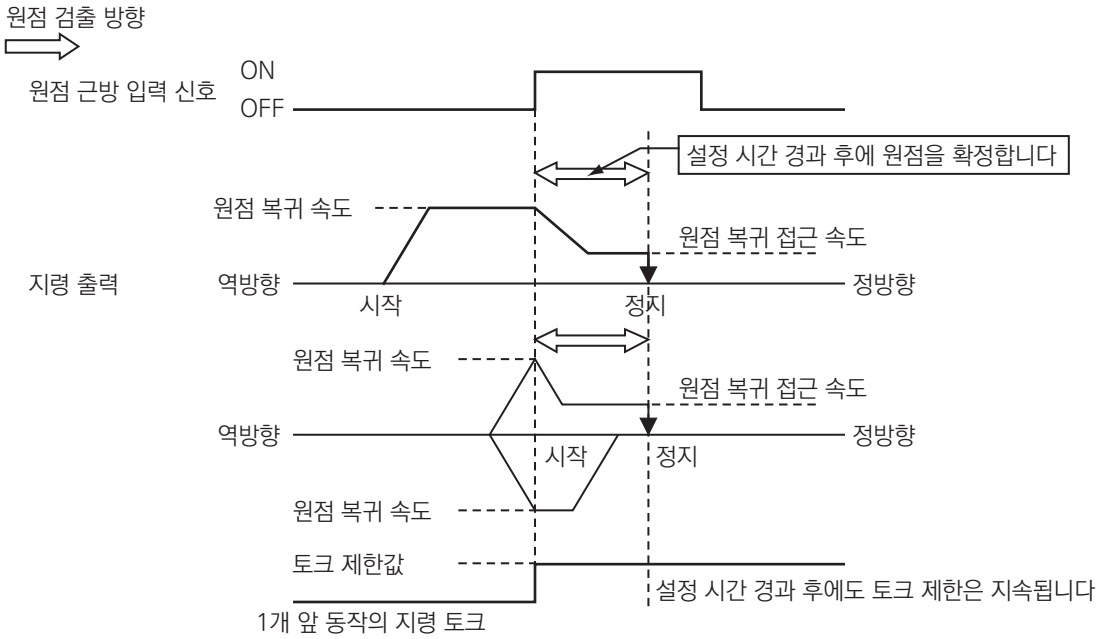


원점 위치 오프셋

기계 원점이 확정된 후 원점 복귀 보정값이 설정된 경우에는 원점 복귀 보정의 동작 완료 후에 설정한 값으로 현재값을 프리셋합니다.
기계 원점을「0」으로 하지 않고 임의의 값으로 사용하기 위한 기능입니다.
또한, 절대값 엔코더 시스템의 경우는 동시에 절대값 엔코더 원점 위치 오프셋값을 자동으로 계산하여 CPU 유니트의 배터리 백업 메모리에 저장합니다.

원점 복귀 보류 시간

원점 복귀 동작이「12: 근방 퇴피 · 보류 시간 지정」인 경우의 보류 시간을 밀리초 단위로 설정합니다.
원점 근방 입력 신호의「OFF」→「ON」에 의한 감속 시작부터의 시간입니다.



원점 복귀 보정값

기계 원점이 확정된 후에 설정한 값으로 상대 위치 결정을 실행하고 기계 원점을 보정합니다. 이때의 이동 속도는 원점 복귀 보정 속도가 됩니다.
기계 원점이 확정된 후에는 워크를 조정하기 어려우므로 미세 조정이 필요한 경우에는 처음 확정한 기계 원점에서의 이탈량을 보정 데이터로 설정합니다.
원점 근방 센서를 다시 설치하기 어렵거나 모터 교환으로 인해 기계 원점이 어긋난 경우에 사용하면 편리합니다.

원점 복귀 보정 속도

원점 복귀 보정값을 설정한 경우의 속도를 지령 단위/s로 설정합니다.

8-2-2 원점 복귀 모니터

사용자 프로그램에서는 축 변수를 판독하여 원점 복귀 상태나 입력 신호 상태를 모니터할 수 있습니다. 또한, 변수명은 _MC_AX[*]를 예로 들어 설명하고 있으나 _MC1_AX[*], _MC2_AX[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_AX[0-255].Status.Homing	BOOL	원점 복귀 중	MC_Home 명령 또는 MC_HomeWithParameter 명령에 의해 원점 복귀를 실행하고 있을 때 TRUE가 됩니다.
_MC_AX[0-255].Dtails.Homed	BOOL	원점 확정	원점 확정 상태 시에 TRUE가 됩니다. FALSE: 원점 미확정 TRUE: 원점 확정
_MC_AX[0-255].Dtails.InHome	BOOL	원점 정지	원점 위치의 범위 내에서 TRUE가 됩니다. 아래의 AND 조건입니다. ・원점 확정 상태 ・피드백 현재 위치가 원점을 중심으로 「원점 위치 범위」의 폭에 들어갔을 때 지령 상태에서 축 동작 중에 원점 위치를 통과하는 경우에도 TRUE가 됩니다.
_MC_AX[0-255].DrvStatus.P_OT	BOOL	정방향 한계 입력	정방향 한계 입력이 유효할 때 TRUE가 됩니다.
_MC_AX[0-255].DrvStatus.N_OT	BOOL	역방향 한계 입력	역방향 한계 입력이 유효할 때 TRUE가 됩니다.
_MC_AX[0-255].DrvStatus.HomeSw	BOOL	원점 근방 입력	원점 근방 입력이 유효할 때 TRUE가 됩니다.
_MC_AX[0-255].DrvStatus.Home	BOOL	원점 입력	원점 입력이 유효할 때 TRUE가 됩니다.

8-3 원점 복귀 동작

원점 복귀는 위치 결정 시스템의 구성이나 용도에 따라 기계 원점의 확정 방법을 선택할 필요가 있습니다.
MC 기능 모듈에서는 10종류의 원점 복귀 동작 모드가 준비되어 있습니다.
또한, 원점을 확정한 후에 원점 복귀 보정값으로 미세 조정할 수도 있습니다.



참고

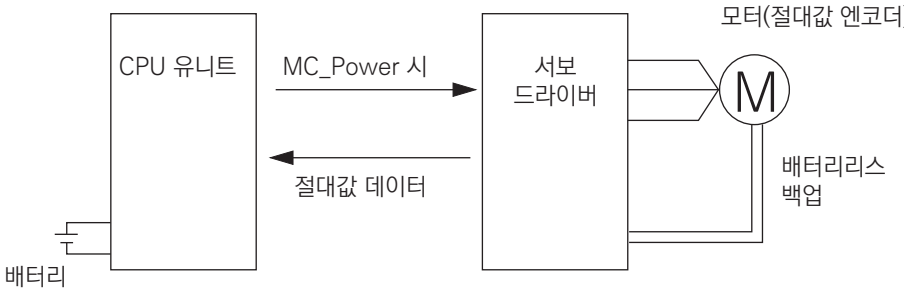
- 어떤 모드가 적합한지는 위치 결정 시스템의 구성이나 용도에 따라 다릅니다.
  리니어 모드(유한 길이)에서 원점 근방 센서와 정방향 한계 입력, 역방향 한계 입력을 갖춘 기계계에서는「근방 퇴피 · 원점 근방 입력 OFF 지정 동작」을 선택하는 것이 일반적입니다.
- 원점 복귀에서는 오버라이드는 무효해집니다.
- 원점 확정 시 또는 원점 복귀 보정 이동 완료 시에만 인포지션 체크 설정에 따라 인포지션 체크를 실행합니다.
- 원점 복귀 중에 다른 모션 제어 명령을 이용한 다중 기동을 실행할 경우 버퍼드 및 블렌딩은 실행할 수 없습니다.

원점 복귀에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

8-4 절대값 엔코더 원점 설정

여기에서는 오므론 서보 모터/드라이버 1S 시리즈 EtherCAT 통신 내장 타입을 사용할 경우에 대해 설명합니다.

절대값 엔코더는 CPU 유니트의 전원이 끊겨도 절대값 데이터를 유지하므로 MC_Power(운전 가능) 명령 실행 시에 절대값 엔코더에서 현재 위치를 판독하여 위치를 확정시킵니다.
일단 원점을 확정해 두면 인크리멘털 엔코드와 같이 원점 복귀를 조작하지 않아도 됩니다.



사용상의 주의

- 오므론 서보 드라이버 1S 시리즈를 사용하는 경우에는 CPU 유니트의 배터리를 접속해 주십시오.
- 오므론 서보 드라이버 G5 시리즈에서 절대값 엔코더를 사용하는 경우에는 CPU 유니트의 배터리와 서보 드라이버의 절대값 엔코더 백업용 배터리 양쪽을 접속해 주십시오.
- 절대값 엔코더를 처음 사용하거나 모터를 교환할 때, 오므론 서보 드라이버 G5 시리즈를 사용할 때는 절대값 엔코더의 배터리 방전 등 절대값 데이터가 손상되면 반드시 「MC_Home(원점 복귀)」 명령 또는 「MC_HomeWithParameter(파라메터 지정 원점 복귀)」 명령을 실행하여 원점을 확정해 주십시오. 그 이후에는 MC_Power(운전 가능) 명령의 입력 변수 Enable(유효)을 TRUE로 하면 원점을 확정합니다. 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 앞 설명과 함께 EtherCAT 프로세스 데이터 통신이 확립되지 않은 상태에서 확립된 상태로 바뀐 경우에도 원점을 확정합니다.
- CPU 유니트 측의 배터리가 이상한 상태에서 CPU 유니트의 전원을 투입하면 「절대값 엔코더 원점 위치 오프셋 판독 이상(이벤트 코드: 14600000Hex)」이 발생합니다. 이 경우에도 「ResetMCError(MC 전체 에러 리셋)」 명령으로 이상을 해제하면 서보 ON할 수 있습니다.

  유니트 버전 Ver.1.09 이하인 CPU 유니트에서는 이상 해제 후 서보가 ON되었을 때 절대값 엔코더 원점 위치 오프셋값을 「0」으로 하여 원점을 확정합니다.

  유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 이상 해제 후에 「MC_Home(원점 복귀)」 명령 또는 「MC_HomeWithParameter(파라메터 지정 원점 복귀)」 명령을 실행하여 원점을 확정할 때까지 원점이 확정되지 않은 상태를 유지합니다.

  올바른 위치에서 동작을 실행하려면 「MC_Home(원점 복귀)」 명령 또는 「MC_HomeWithParameter(파라메터 지정 원점 복귀)」 명령을 실행하여 올바른 위치에서 원점을 확정해 주십시오.

  CPU 유니트의 전원을 끊으면 원점이 확정되지 않은 상태가 됩니다.

참고

오므론 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입을 사용할 경우에도 절대값 엔코더 원점 설정을 할 수 있습니다. 리니어 모터 타입을 사용할 때는 아래의 사항에 주의하여 본 절(8-4)의 설명을 읽어 주십시오.

- 리니어 모터 타입은「엔코더」는 사용하지 않으나 동일한 기능인「외부 스케일」을 사용합니다.
- 본 절(8-4)의 설명 속「절대값 엔코더」는「절대값 타입의 외부 스케일」로 읽을 수 있습니다.
- 절대값 타입의 외부 스케일은 절대값 엔코더에는 있는 회전량 데이터가 없기 때문에 절대값 엔코더에서 필요했던 셋업이 불필요합니다. 또한, 회전량 데이터를 유지하는 배터리는 사용하지 않습니다.
- 리니어 모터 타입의 사양에 대한 상세한 내용은 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입 사용자 매뉴얼(SBCE-366)』을 참조해 주십시오.

8-4-1 기능 개요

절대값 엔코더 시스템의 원점 설정은 MC_Power(운전 가능) 명령 실행 시에 절대값 엔코더의 오프셋을 보정하고 원점을 확정합니다.
유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 앞 설명과 함께 EtherCAT 프로세스 데이터 통신이 확립되지 않은 상태에서 확립된 상태로 바뀐 경우에도 원점을 확정합니다.
또한, 인크리멘털 엔코더와 마찬가지로 원점 복귀를 통해 원점을 확정시킬 수도 있습니다. 이 경우 원점 확정 후의 지령 위치와 절대값 엔코더에서 판독한 절대값 데이터와의 차이를 절대값 엔코더 원점 위치 오프셋으로 CPU 유니트의 배터리 백업 메모리에 저장합니다.

MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령에 의한 현재 위치 프리셋을 실행한 경우에도 원점 확정 후의 지령 위치와 절대값 데이터와의 차이를 오프셋량으로 하여 절대값 엔코더 원점 위치 오프셋에 설정합니다.
MC 기능 모듈에서는 절대값 엔코더 원점 위치 오프셋을 CPU 유니트 내의 배터리 백업 메모리에 자동 저장하고 있으므로 Sysmac Studio 등에서 저장 조작을 실행할 필요가 없습니다.

사용상의 주의

- 절대값 엔코더 원점 위치 오프셋은 축 번호에 맞춰 축별로 배터리 백업 메모리에 저장됩니다. 축 번호를 변경하면 저장된 오프셋량이 소실됩니다. 축 번호를 변경한 경우는 다시 원점 복귀를 실행해 주십시오.
- CPU 유니트 또는 CPU 유니트 측의 배터리를 교환할 때는 교환 전에 원점 확정 상태에서 절대값 엔코더 원점 위치 오프셋을 백업해 주십시오.
- 교환 후에 복원하면 교환 전에 확정한 원점을 사용할 수 있습니다.
- 백업/복원은 Sysmac Studio를 사용합니다.

  조작 방법에 대한 상세한 내용은 🕮『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.

대상 서보 모터

절대값 엔코더 원점 설정을 사용할 수 있는 서보 모터를 아래에 나타냅니다.

제조 회사명	시리즈명	서보 드라이버 형식	서보 모터 형식
오므론	1S 시리즈	R88D-1SN□□□-ECT	R88M-1□□□□□□S R88M-1□□□□□□T R88M-1□□□□□□C
	G5 시리즈	R88D-KN□□□-ECT	R88M-K□□□□□S R88M-K□□□□□T R88M-K□□□□□C
		R88D-KN□□□-ECT-L	R88L-EC□□□□□□□□□

사용상의 주의

NX 시리즈 펄스 출력 유니트에서는 절대값 엔코더를 사용할 수 없습니다.

서보 드라이버와의 접속

🕮『NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용자 매뉴얼(SBCD-358)』에 따라 올바르게 접속해 주십시오.

8-4-2 설정 순서

절대값 엔코더 시스템의 원점 설정 순서에 대해 설명합니다.

1 **절대값 엔코더 셋업**

셋업 방법에 대해서는 🕮『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.

2 **축 파라메터 설정**

MC 기능 모듈의 축 파라메터에 있는 [Position Count Settings]의 [Encoder Type]에서 [1: Absolute encoder (ABS)]를 선택합니다.

상세한 내용에 대해서는 🕮「5-2-7 위치 카운트 설정 (P.5-23)」을 참조해 주십시오.

3 **원점 복귀 실행**

원점 복귀를 실행합니다.

원점 확정 방법은 MC 기능 모듈의 축 파라메터 [Homing Settings]의 [Homing Method]를 이용하여 선택할 수 있습니다.

원점 확정 후 지령 위치와 절대값 엔코더에서 판독한 절대값 데이터와의 차이를 절대값 엔코더 원점 위치 오프셋으로 배터리 백업 메모리에 저장합니다.

절대값 엔코더 셋업

절대값 엔코더를 처음 사용하거나 회전량 데이터를「0」으로 초기화하고 싶을 때 또는 절대값 엔코더에 배터리를 접속하지 않고 장기간 방치한 경우 등에는 절대값 엔코더의 셋업이 필요합니다.
셋업 방법에 대한 상세한 내용은 🕮『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.



사용상의 주의

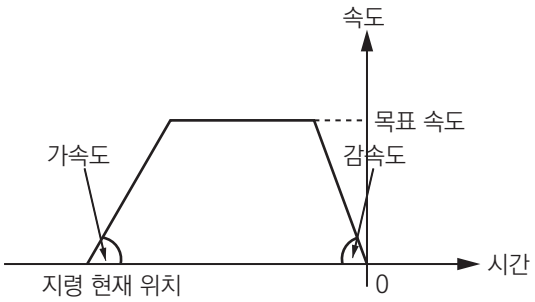
오므론 서보 드라이버 1S 시리즈 및 G5 시리즈는 절대값 엔코더를 셋업한 후 전원을 다시 투입해야 합니다. 절대값 엔코더의 셋업 처리가 완료되면 서보 드라이버에 절대값 클리어 에러(A27.1)가 발생합니다. 서보 드라이버의 제어 전원을 한번 OFF한 후 다시 전원을 투입하면 이상은 해제되고, 절대값 엔코더의 셋업 순서가 종료됩니다.

로터리 모드에서 절대값 엔코더 사용하기

축 파라메터 [Count Mode]에 [Rotary Mode]를 설정한 경우 현재 위치는 링 카운터 상한 설정값과 링 카운터 하한 설정값의 범위에서 링 형상의 카운터가 됩니다.
절대값 엔코더를 사용하여 로터리 모드로 설정한 경우 MC 기능 모듈은 모션 제어 주기마다 자동으로 절대값 엔코더 원점 위치 오프셋을 계산하여 갱신합니다.
갱신된 절대값 엔코더 원점 위치 오프셋은 CPU 유니트의 전원이 OFF될 때 CPU 유니트에 내장된 배터리 백업 메모리에 자동 저장됩니다.
이로 인해 CPU 유니트나 서보 드라이버의 전원이 OFF되어도 다음 전원 투입 시에 절대값 엔코더에서 회전축의 현재 위치를 복원할 수 있습니다.

8-5 고속 원점 복귀

원점 확정 상태에서 원점 위치로 빠르게 위치 결정하는 기능입니다.
MC_MoveZeroPosition(고속 원점 복귀) 명령을 사용하여 목표 속도, 가속도, 감속도, 저크를 지정할 수 있습니다.
원점 미확정 상태에서 실행하면 명령에서 이상이 발생합니다.



참고

MC_MoveZeroPosition(고속 원점 복귀) 명령에 대한 상세한 내용에 대해서는 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

9

모션 제어 기능

여기에서는 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입과 접속할 경우의 모션 제어 기능에 대해 설명합니다.

9-1 단축 위치 제어 9-3
9-1-1 동작 개요 9-3
9-1-2 절대값 위치 결정 9-4
9-1-3 상대값 위치 결정 9-4
9-1-4 인터럽트 피딩 위치 결정 9-5
9-1-5 사이클릭 동기 위치 결정 9-6
9-1-6 정지 9-6
9-1-7 오버라이드 9-11

9-2 단축 동기 제어 9-12
9-2-1 동기 제어의 개요 9-12
9-2-2 기어 동작 9-12
9-2-3 위치 지정 기어 동작 9-13
9-2-4 캠 동작 9-14
9-2-5 캠 테이블 9-15
9-2-6 사다리꼴 패턴 캠 동작 9-23
9-2-7 가감산 위치 결정 9-24
9-2-8 주축 위상 보정 9-25
9-2-9 종축 위치 보정 9-26
9-2-10 멀티 모션에서 동기 제어 실현하는 방법 9-26

9-3 단축 속도 제어 9-28
9-3-1 속도 제어 9-28
9-3-2 사이클릭 동기 속도 제어 9-29

9-4 단축 토크 제어 9-30

9-5 단축 제어의 공통 기능 9-31
9-5-1 위치 9-31
9-5-2 속도 9-33
9-5-3 가속과 감속 9-34
9-5-4 저크 9-36
9-5-5 동작 방향 지정 9-37
9-5-6 모션 제어 명령의 재기동 9-40
9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) 9-45

9-6 다축 협조 제어 9-51
9-6-1 동작 개요 9-51
9-6-2 직선 보간 9-54
9-6-3 원호 보간 9-55
9-6-4 축 그룹 사이클릭 동기 위치 제어 9-55

9-6-5 다축 협조 제어에서의 정지 9-56
9-6-6 다축 협조 제어에서의 오버라이드 9-58

9-7 다축 협조 제어의 공통 기능 9-59
9-7-1 다축 협조 제어에서의 속도 9-59
9-7-2 다축 협조 제어에서의 가속과 감속 9-60
9-7-3 다축 협조 제어에서의 저크 9-61
9-7-4 다축 협조 제어의 모션 제어 명령 재기동 9-62
9-7-5 다축 협조 제어의 모션 제어 명령 다중 기동(버퍼 모드) 9-63

9-8 기타 기능 9-70
9-8-1 현재 위치 변경 기능 9-70
9-8-2 토크 제한 기능 9-71
9-8-3 래치 기능 9-71
9-8-4 구역 기능 9-72
9-8-5 소프트웨어 리미트 기능 9-73
9-8-6 위치 편차 감시 기능 9-75
9-8-7 편차 카운터 리셋 9-75
9-8-8 축 간 편차 감시 기능 9-76
9-8-9 인포지션 체크 9-77
9-8-10 축 사용 변경 기능 9-79
9-8-11 디지털 캠 스위치 유효 기능 9-79
9-8-12 임의 좌표계의 3D 모션 모니터 표시 9-80

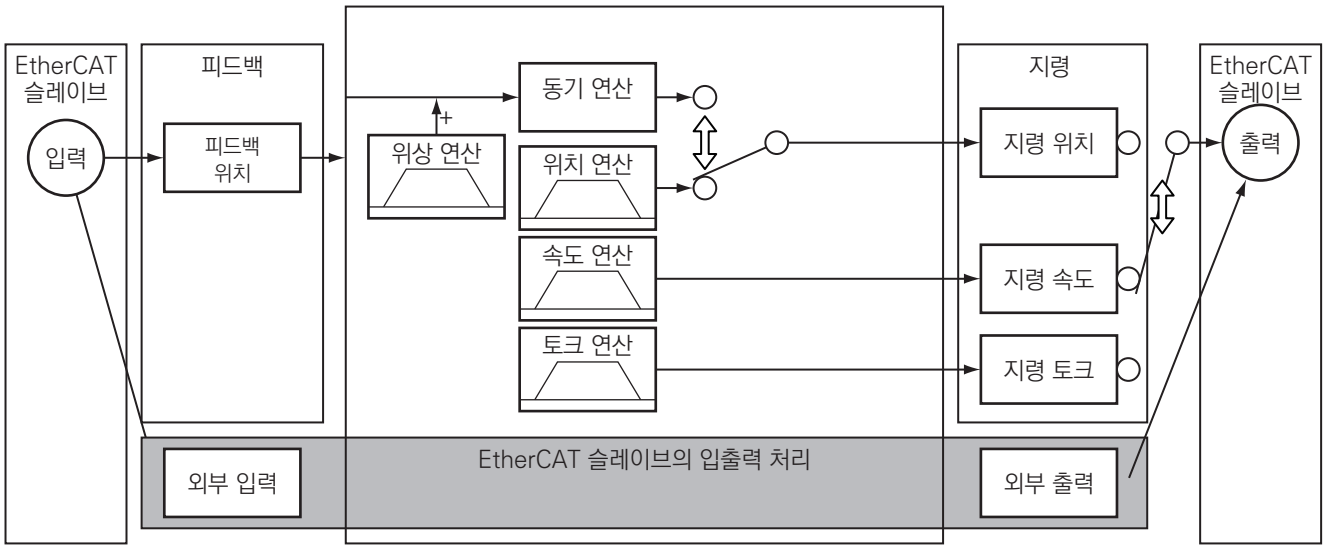
9-1 단축 위치 제어

MC 기능 모듈은 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입 및 G5 시리즈 EtherCAT 통신 내장 타입과 접속하여 위치 제어, 속도 제어, 토크 제어를 실행할 수 있습니다.
여기에서는 단축에서의 위치 결정 동작에 대해 설명합니다.

NX 시리즈 펄스 출력 유니트를 사용할 경우에는 모션 제어 기능의 일부가 다릅니다. 상세한 내용은 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

9-1-1 동작 개요

MC 기능 모듈의 단축 제어 기능에는 동작 프로파일을 지령하는 제어와 동기 제어가 있습니다.
동작 프로파일을 지령하는 제어에서는 위치 제어, 속도 제어, 토크 제어 등 3가지 제어 모드로 지령할 수 있습니다.
동기 제어에서는 제어 대상인 종축이 주축에 대한 캠 프로파일 곡선 및 기어비 등으로 표현되는 동기 관계를 맺으며 동작합니다.
또한, 조그 전송 등의 수동 운전 및 원점 복귀에도 대응합니다.



(주) 동기 연산에는 지령 위치 및 피드백 위치를 입력으로 사용할 수 있습니다.

축 에러 리셋

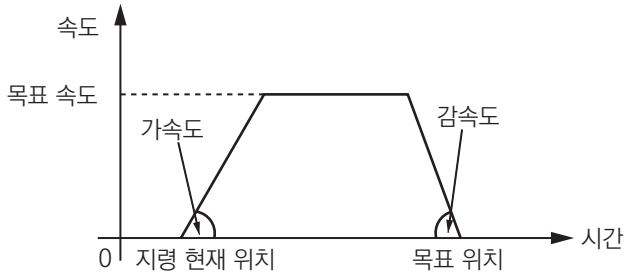
축에서 이상이 발생한 경우 이상의 원인을 제거한 후에 MC_Reset(축 에러 리셋) 명령으로 이상을 해제합니다.

축 에러 리셋에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_Reset(축 에러 리셋) 명령을 참조해 주십시오.

NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

9-1-2 절대값 위치 결정

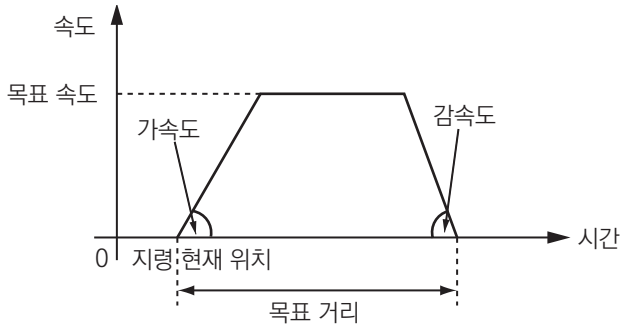
원점을 기준으로 하는 절대 좌표의 목표 위치를 지정하는 위치 결정입니다.
카운트 모드를 로터리 모드로 하여 동작 방향을 지정하면 로터리 테이블에서 최단거리 등의 위치 결정을 할 수 있습니다.



기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_MoveAbsolute(절대값 위치 결정) 명령 및 MC_Move(위치 결정) 명령을 참조해 주십시오.

9-1-3 상대값 위치 결정

현재 위치를 기준으로 거리를 지정하는 위치 결정입니다.
카운트 모드를 로터리 모드로 설정하면 링 카운터 범위를 초과하는 이동 거리를 지정할 수 있습니다.



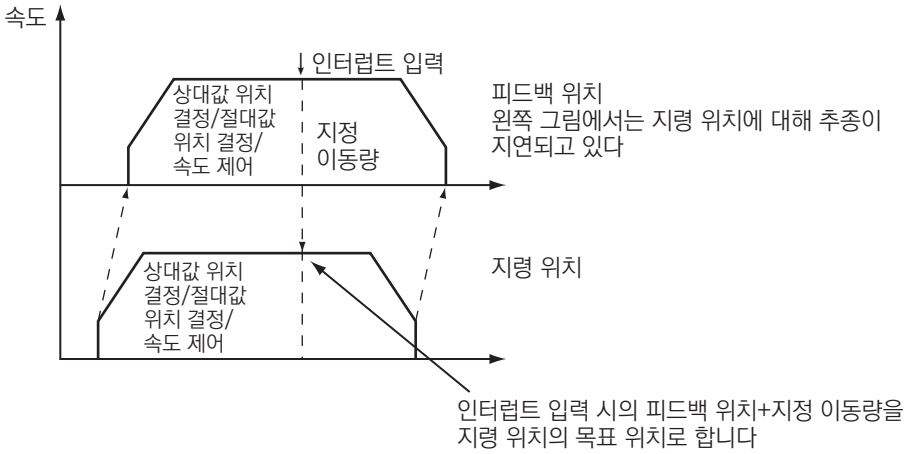
기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_MoveRelative(상대값 위치 결정) 명령 및 MC_Move(위치 결정) 명령을 참조해 주십시오.

9-1-4 인터럽트 피딩 위치 결정

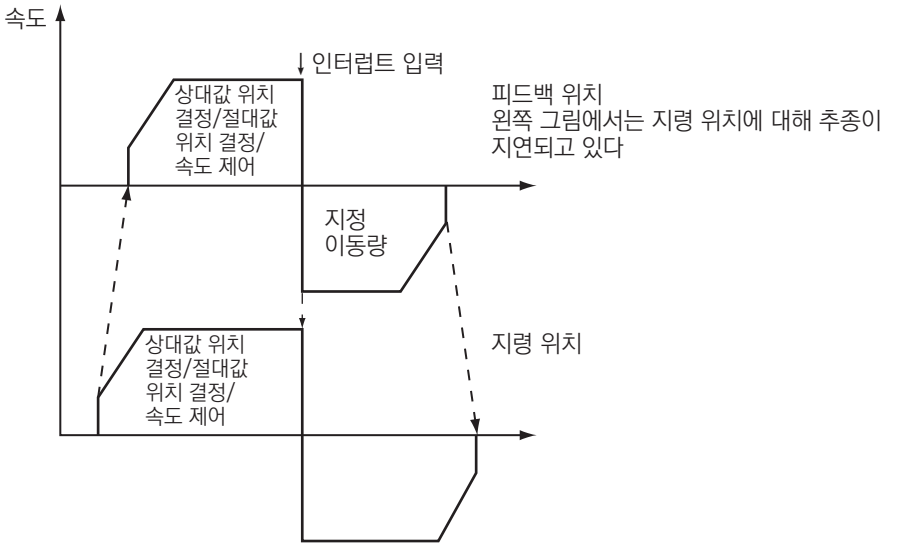
지정한 속도로 축을 보내고 트리거 신호가 발생한 피드백 위치에서 지정한 거리를 이동하는 위치 결정입니다.
절대값 위치 결정 또는 상대값 위치 결정의 경우 지정한 이동량 이내에 트리거 신호가 발생하지 않을 때 이상을 출력하도록 선택할 수 있습니다.
트리거 신호 발생 시의 피드백 위치는 서보 드라이버 측의 래치 기능을 사용하므로 위치 편차의 영향을 받지 않는 피딩 위치 결정을 실현할 수 있습니다 또한, 지정한 위치 범위 외의 트리거 신호 발생을 무효화하는 윈도우 기능에 의해 포장기 등에서 필름의 마크 인쇄 이외의 영향을 제거할 수 있습니다.

기준: 피드백 위치

● **진행 방향으로 지정량 이동**



● **진행 방향과 역방향으로 지정량 이동**



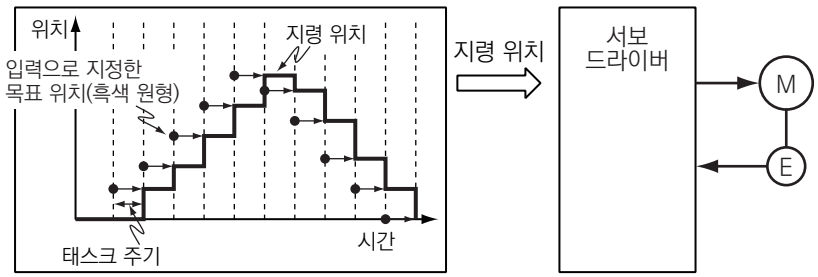
축 파라메터의 반전 시 동작으로 반전 시 감속 정지를 선택하면 가감속 커브로 반전합니다.

기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_MoveFeed (인터럽트 피딩 위치 결정) 명령을 참조해 주십시오.

NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

9-1-5 사이클릭 동기 위치 결정

사이클릭 동기 위치 결정은 축에 지정한 목표 위치를 프라이머리 주기 또는 정주기 태스크의 태스크 주기별로 출력하는 기능입니다. 목표 위치는 절대 위치에서 지정합니다.
사용자가 작성한 임의의 궤도에서 동작시키고 싶을 때 사용합니다.



기능에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_SyncMoveAbsolute(사이클릭 동기 절대 위치 결정) 명령을 참조해 주십시오.

버전 관련 정보

사이클릭 동기 위치 결정 기능은 유니트 버전 Ver.1.03 이상인 CPU 유니트와 Sysmac Studio Ver.1.04 이상의 조합으로 사용할 수 있습니다.

9-1-6 정지

축 동작을 정지하는 기능에는 서보 드라이버에 접속된 즉시 정지 입력 신호 및 한계 입력 신호와 사용자 프로그램의 모션 제어 명령에 의한 정지 기능, 이상 발생 등에 의한 정지가 있습니다. NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼 (SBCE-374)』을 참조해 주십시오.

서보 드라이버의 입력 신호에 의한 정지

서보 드라이버의 즉시 정지 입력 신호 또는 한계 입력 신호에 의해 축 동작을 정지합니다.
Sysmac Studio를 사용하여 정지 방법을 선택할 수 있습니다.

● 즉시 정지 입력

서보 드라이버 측 입력 신호의 상태에 따라 MC 기능 모듈 측에서 정지 처리를 실행합니다.
MC 기능 모듈의 정지 방법은 다음 중에서 선택할 수 있습니다.

- 지령값을 즉시 정지
- 지령값을 즉시 정지하고 동시에 편차 카운터 리셋을 실행
- 지령값을 즉시 정지하고 동시에 서보 OFF를 실행

사용상의 주의

오므론 서보 드라이버 1S 시리즈 및 G5 시리즈의 즉시 정지 입력은 서보 드라이버 측에서도 이상이 되어 정지 처리를 실행합니다.

● 한계 입력(정방향 한계 입력 또는 역방향 한계 입력)

서보 드라이버 측 입력 신호의 상태에 따라 MC 기능 모듈 측에서 정지 처리를 실행합니다.
MC 기능 모듈의 정지 방법은 다음 중에서 선택할 수 있습니다.

- 지령값을 즉시 정지
- 지령값을 감속 정지
- 지령값을 즉시 정지하고 동시에 편차 카운터 리셋을 실행
- 지령값을 즉시 정지하고 동시에 서보 OFF를 실행



사용상의 주의

- 한계 입력 신호가 ON인 경우 축 지령 명령은 ON 상태인 한계 입력 신호와 동일한 방향으로는 기동하지 마십시오.
- 축 그룹 중 어느 한 축의 한계 입력이 ON인 경우 축 그룹 지령 명령은 기동하지 마십시오.
- 입력 변수「Deceleration(감속도)」을 가진 동기 동작 명령을 실행하는 도중에 감속 정지되는 신호가 입력되면「Deceleration(감속도)」으로 감속 정지합니다.
- 입력 변수「Deceleration(감속도)」을 갖지 못한 동기 동작 명령을 실행하는 도중에 감속 정지되는 신호가 입력되면 축 파라메터의 최대 감속도로 감속 정지합니다.



참고

- 서보 드라이버의 입력 신호를 사용하려면 서보 드라이버를 설정해야 합니다 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입 및 G5 시리즈 EtherCAT 통신 내장 타입에서는 서보 드라이버 측의 초기값으로 즉시 정지 입력과 한계 입력이 할당되어 있습니다.
- 오므론 서보 드라이버 1S 시리즈와 접속하는 경우의 설정 예에 대해서는 🕮「A-1 서보 드라이버 1S 시리즈와의 접속 (P.A-2)」을 참조해 주십시오.
- 오므론 서보 드라이버 G5 시리즈와 접속하는 경우의 설정 예에 대해서는 🕮「A-2 서보 드라이버 G5 시리즈와의 접속 (P.A-11)」을 참조해 주십시오.

모션 제어 명령에 의한 정지

단축 동작 정지에는 MC_Stop(강제 정지) 명령 또는 MC_ImmediateStop(즉시 정지) 명령을 사용합니다.

● MC_Stop(강제 정지) 명령

단축 제어 및 동기 제어 기능에 대해 감속도와 저크를 지정하여 감속 정지할 수 있습니다.
감속도에「0」을 지정하면 즉시 정지하는 지령을 서보 드라이버에 송신합니다.
본 명령으로 감속 정지 중과 입력 변수 Execute(기동)가 TRUE인 동안에는 다른 동작 지령을 받지 않습니다.

● MC_ImmediateStop(즉시 정지) 명령

단축 제어 및 동기 제어 기능에 대해 즉시 정지할 수 있습니다.
축 그룹이 유효한 축에 대해서도 실행할 수 있습니다.

기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_Stop(강제 정지) 명령, MC_ImmediateStop(즉시 정지) 명령을 참조해 주십시오.



참고

MC_Power(운전 가능) 명령의 입력 변수 Enable(유효)을 FALSE로 하면 MC 기능 모듈은 지령값을 즉시 정지하고 서보 OFF를 실행합니다.
서보가 OFF일 때의 서보 드라이버 동작은 서보 드라이버 측의 설정에 따릅니다.

이상 발생 등에 의한 정지

● 단축 동작 중 이상 발생에 의한 정지

단축 동작 중에 이상이 발생하면 이상 내용에 따라 즉시 정지 또는 감속 정지합니다.
이상별 정지 방법에 대해서는 🕮『NJ/NX 시리즈 트러블 슈팅 매뉴얼(SBCA-361)』을 참조해 주십시오.

● 소프트웨어 리미트에 의한 정지

소프트웨어 리미트에 의한 정지는 축 파라메터 설정의「소프트웨어 리미트 기능」에서 설정합니다.
소프트웨어 리미트에 의한 정지 방법은 다음 중에서 선택할 수 있습니다.

- 지령 위치에 대해 유효. 감속 정지한다.
- 지령 위치에 대해 유효. 즉시 정지한다.
- 피드백 위치에 대해 유효. 감속 정지한다.
- 피드백 위치에 대해 유효. 즉시 정지한다.

소프트웨어 리미트 기능에 대해서는 🕮「9-8-5 소프트웨어 리미트 기능 (P.9-73)」을 참조해 주십시오.

● 모션 제어 주기 초과 발생에 의한 정지

모션 제어의 처리가 2주기 이내에 완료하지 않고 제어 주기 초과가 발생하면 즉시 정지합니다.



사용상의 주의

NX701 CPU 유니트를 사용하여 멀티 모션으로 동작 중인 경우에 어느 한쪽의 태스크에서 모션 제어 주기 초과가 발생했을 때는 양쪽 태스크의 모든 축이 즉시 정지합니다.
멀티 모션은 🕮「A-6-2 모션 제어 기능 (P.A-32)」을 참조해 주십시오.

● 서보 OFF가 되는 이상 발생

서보 OFF가 되는 이상이 발생한 경우에는 즉시 정지합니다. 서보가 OFF일 때의 서보 드라이버 동작은 서보 드라이버 측의 설정에 따릅니다.

● MC 시운전 기능의 시작에 의한 정지

Sysmac Studio의 MC 시운전 기능을 시작했을 때 각 축은「최대 감속도」로 감속 정지합니다.

● MC 시운전 기능의 종료에 의한 정지

Sysmac Studio의 MC 시운전 기능을 종료했을 때 각 축은「최대 감속도」로 감속 정지합니다.

- Sysmac Studio의 MC 시운전 화면에 있는 [Stop MC Test Run]을 선택한다.
- Sysmac Studio의 MC 시운전 화면을 닫는다.
- Sysmac Studio를 종료한다.

● CPU 유니트의 모드 변환에 의한 정지

CPU 유니트의 모드를 변환하면 각 축은「최대 감속도」로 감속 정지합니다.

사용상의 주의

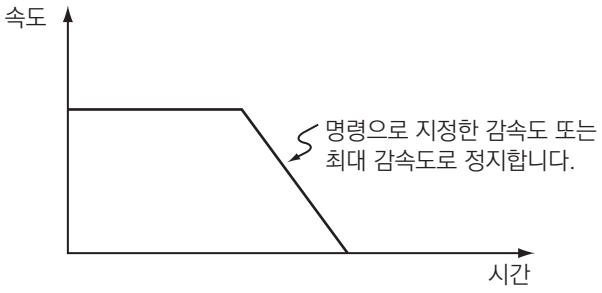
- 입력 변수「Deceleration(감속도)」을 가진 동기 동작 명령을 실행하는 도중에 감속 정지되는 이상이 발생하면「Deceleration(감속도)」으로 감속 정지합니다.
- 입력 변수「Deceleration(감속도)」을 갖지 못한 동기 동작 명령을 실행하는 도중에 감속 정지되는 이상이 발생하면 축 파라메터의 최대 감속도로 감속 정지합니다.

참고

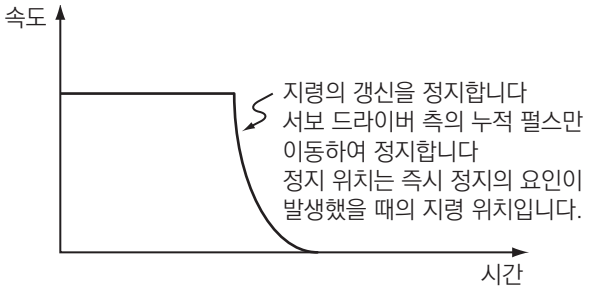
- 운전 모드에서 프로그램 모드로 변환했을 때는 동작 중인 모션 제어 명령은 중단됩니다. 이때 명령의 출력 변수「CommandAborted(실행 중단)」는 FALSE인 상태입니다. 프로그램 모드로 변환한 후에도 서보 ON/OFF 상태를 유지합니다.
- 운전 모드에서 프로그램 모드로 변환하여 감속 정지했을 때 운전 모드로 되돌리면 모션 제어 명령의 출력 변수가 클리어되므로 모션 제어 명령의 출력 변수「CommandAborted(실행 중단)」는 FALSE 상태가 됩니다.
- MC_SaveCamTable(캠 테이블 저장) 명령으로 저장 중인 경우 저장 처리는 계속됩니다.
- MC_GenerateCamTable(캠 테이블 생성) 명령으로 캠 테이블 생성을 실행 중인 경우 생성 처리는 계속됩니다.

정지 방법

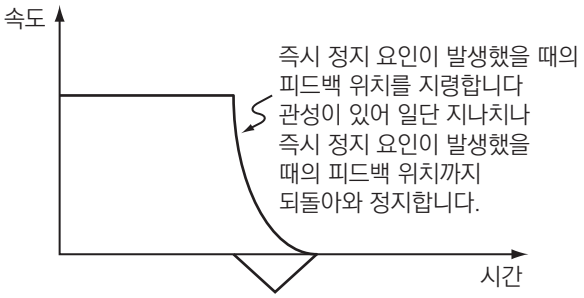
● 지령값을 감속 정지



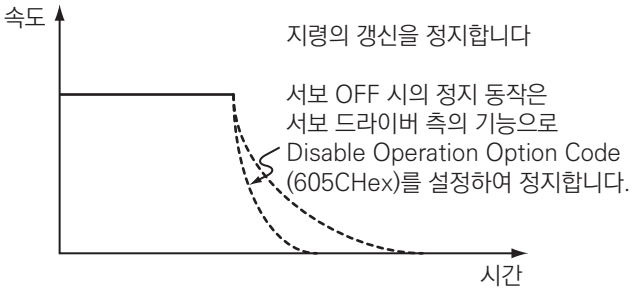
● 지령값을 즉시 정지



● **지령값을 즉시 정지하고 동시에 편차 카운터 리셋을 실행**



● **지령값을 즉시 정지하고 동시에 서보 OFF를 실행**



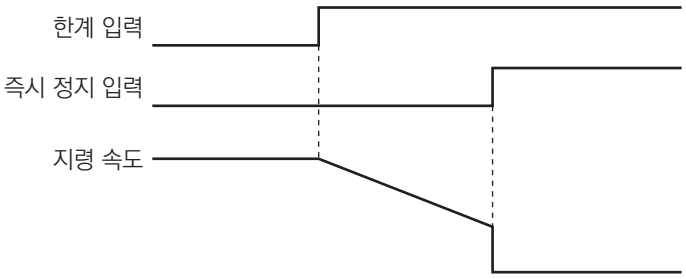
정지 우선순위

정지 우선순위는 다음 표와 같습니다. 정지 중에 우선순위가 높은 정지가 발생하면 우선순위가 높은 정지 방법으로 바뀝니다.

정지 방법	우선순위 (숫자가 큰 쪽이 우선)
지령값을 즉시 정지하고 동시에 서보 OFF를 실행	4
지령값을 즉시 정지하고 동시에 편차 카운터 리셋을 실행	3
지령값을 즉시 정지	2
지령값을 감속 정지	1

● **예**

아래의 그림은 한계 입력 신호 ON에서 감속 정지 도중에 즉시 정지 입력이 ON되어 즉시 정지하는 경우의 예입니다.



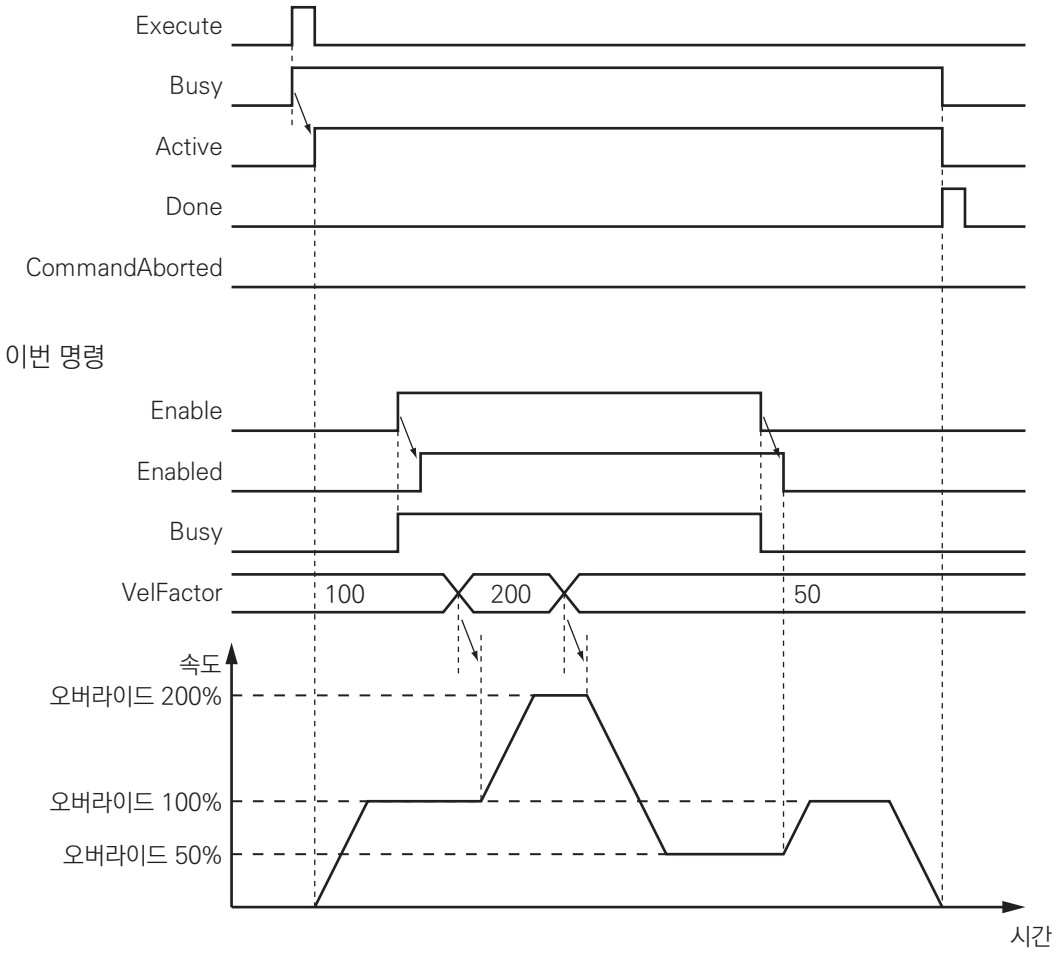
9-1-7 오버라이드

MC_SetOverride(오버라이드값 설정) 명령으로 기동 중인 축의 동작에 대해 오버라이드값을 설정하는 기능입니다.
오버라이드값은 목표 속도에 대한 비율로 0~500%를 설정할 수 있습니다.
목표 속도에 대해 0%를 설정하면 속도 「0」에서 정지한 채 동작 상태를 계속합니다.

오버라이드가 유효인 동안 설정한 오버라이드값이 입력됩니다.
오버라이드를 무효화하면 오버라이드값은 100%로 되돌아갑니다.
오버라이드값으로 변경한 후의 값이 최고 속도를 초과하면 각 축의 「최고 속도」가 됩니다.

● MC_MoveAbsolute(절대값 위치 결정) 명령에 대한 오버라이드

MC_MoveAbsolute(절대값 위치 결정) 명령에 오버라이드 명령을 사용한 경우의 타이밍 차트 예입니다.



기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의
MC_SetOverride(오버라이드값 설정) 명령을 참조해 주십시오.

9-2 단축 동기 제어

여기에서는 단축의 동기 제어 동작에 대해 설명합니다.

9-2-1 동기 제어의 개요

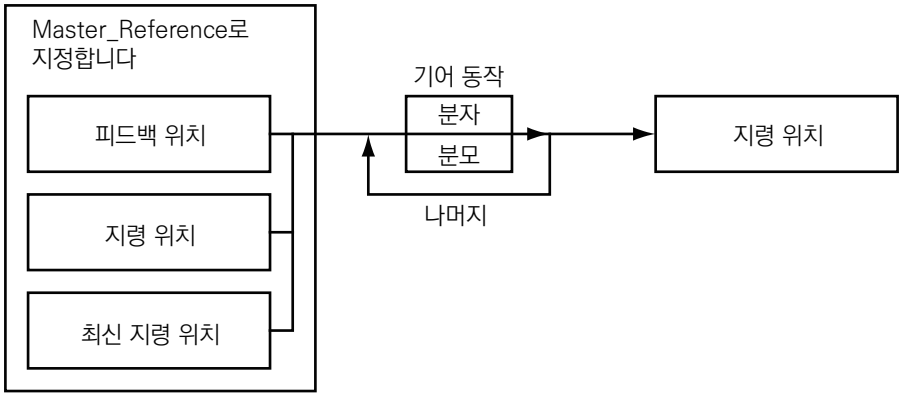
주축의 위치에 동기해서 종축의 위치를 제어하는 기능입니다.
주축에는 각 축의 지령 위치 또는 피드백 위치를 지정할 수 있습니다.
종축에 대한 지령 속도가 축 파라메터의 최고 속도를 초과한 경우에는 최고 속도로 지령합니다.
이 경우 부족한 이동량은 다음 주기 이후로 분배하여 출력합니다.

사용상의 주의

- 종축에는 엔코더 축, 가상 엔코더 축, 단축 위치 제어 축을 지정할 수 없습니다.
- NX701 CPU 유니트를 사용하여 멀티 모션으로 동작 중인 경우에 동기 제어할 때는 구성하는 주축과 종축을 동일한 태스크로 해 주십시오.
  MC_GearIn(기어 동작 시작) 명령 및 MC_CamIn(캠 동작 시작) 명령 등의 동기 제어 명령에 의해 태스크가 다른 축을 주축, 종축으로 지정한 경우 「주축 지정 오류(이벤트 코드: 54620000 Hex)」가 발생합니다.
  태스크가 다른 축을 주축, 종축으로 지정하고 싶은 경우에는 「9-2-10 멀티 모션에서 동기 제어 실현하는 방법 (P.9-26)」을 참조해 주십시오.

9-2-2 기어 동작

주축과 종축 사이의 기어비를 설정하고 기어 동작을 실행하는 기능입니다.
MC_GearIn(기어 동작 시작) 명령으로 기어 동작을 시작하고 MC_GearOut(기어 동작 해제) 명령 및 MC_Stop(강제 정지) 명령으로 동기를 해제합니다.



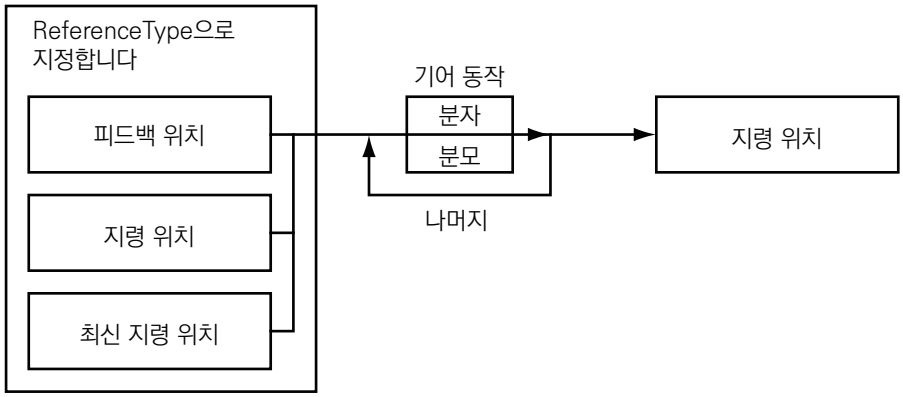
동작 대상의 종축에 대해 기어비 분자, 기어비 분모, 위치 종별 선택, 가속도, 감속도를 지정할 수 있습니다.
또한, 주축에는 지령 위치, 피드백 위치, 최신 지령 위치 중 하나를 지정할 수 있습니다.

동작 시작 후 종축은 주축의 속도에 기어비를 곱한 속도를 목표 속도로 하여 가감속 동작을 실시합니다.
목표 속도에 도달할 때까지를 Catching phase(추적 중)라 하고, 도달 후를 InGear phase(기어 동기 중)라 합니다.
기어비가 양수일 때는 Slave(종축)가 Master(주축)와 동일한 방향으로 이동하고, 음수일 때는 Slave(종축)가 Master(주축)의 역방향으로 이동합니다.

기어 동작에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_GearIn(기어 동작 시작) 명령 및 MC_GearOut(기어 동작 해제) 명령, MC_Stop(강제 정지) 명령을 참조해 주십시오.

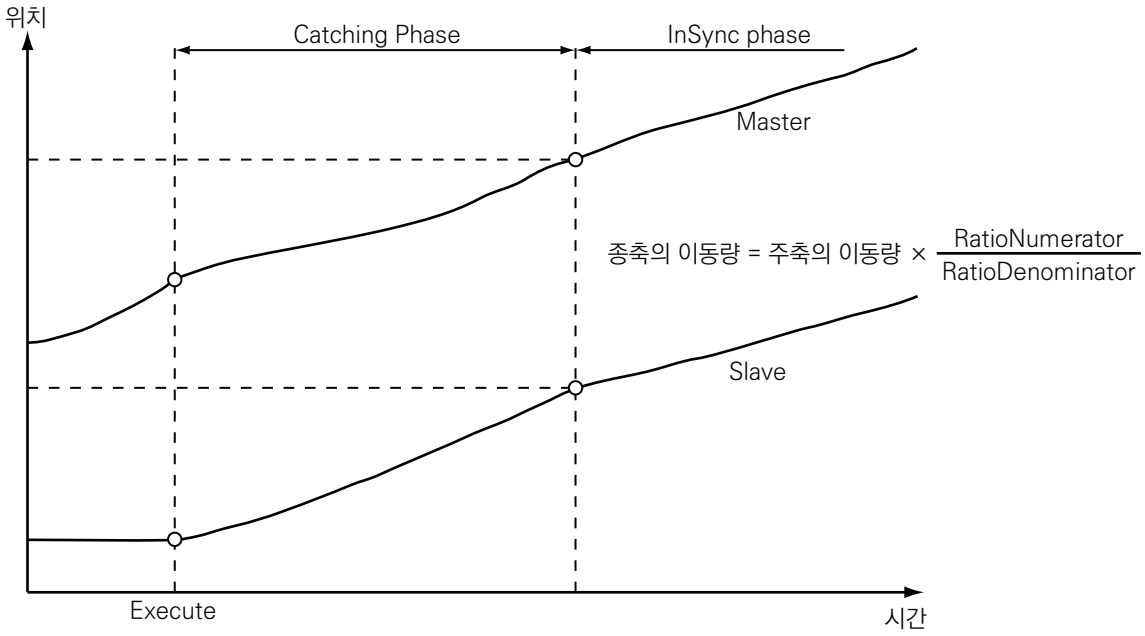
9-2-3 위치 지정 기어 동작

주축과 종축 사이의 기어비를 설정하고 기어 동작을 실행하는 기능입니다.
위치 지정 기어 동작에서는 동기를 시작하는 주축의 위치와 종축의 위치를 지정할 수 있습니다.
MC_GearInPos(위치 지정 기어 동작) 명령으로 위치 지정 기어 동작을 시작하고 MC_GearOut(기어 동작 해제) 명령 또는 MC_Stop(강제 정지) 명령으로 동기를 종료합니다.



동작 대상의 종축에 대해 기어비 분자, 기어비 분모, 위치 종별 선택, 가속도, 감속도를 지정할 수 있습니다.
또한, 주축에는 지령 위치, 피드백 위치, 최신 지령 위치 중 하나를 지정할 수 있습니다.

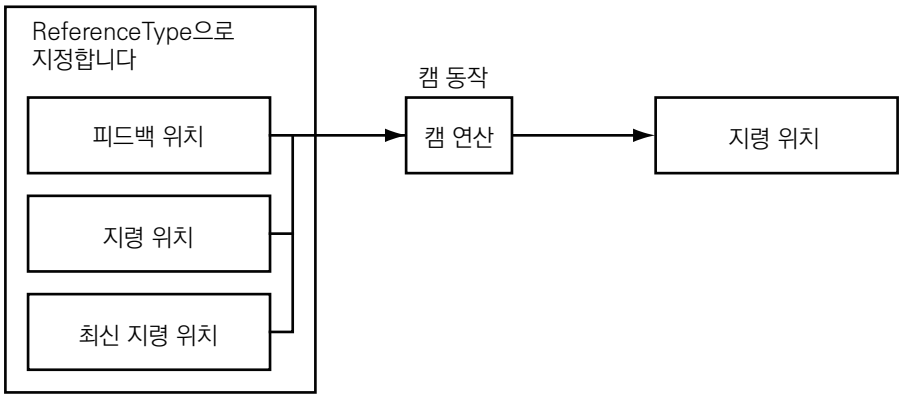
동작 시작 후 종축은 주축의 속도에 기어비를 곱한 속도를 목표로 하여 가감속 동작을 실행합니다.
종축이 종축 동기 위치에 도달할 때까지가 Catching Phase(추적 중)이며, 종축이 종축 동기 시작 위치에 도달하면 InSync(기어 동기 중)가 됩니다. 양쪽 모두 종축은 주축의 위치에 동기합니다
기어비가 양수일 때는 Slave(종축)가 Master(주축)와 동일한 방향으로 이동하고, 음수일 때는 Slave(종축)가 Master(주축)의 역방향으로 이동합니다. 아래 그림은 기어비가 양수일 때의 동작을 나타냅니다.

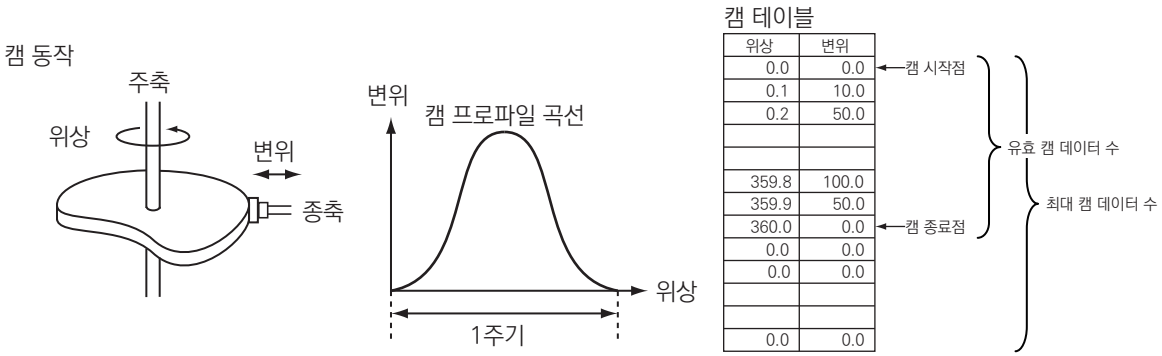


위치 지정 기어 동작에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_GearInPos(위치 지정 기어 동작) 명령, MC_GearOut(기어 동작 해제) 명령 및 MC_Stop(강제 정지) 명령을 참조해 주십시오.

9-2-4 캠 동작

캠 테이블에 따라서 종축이 주축에 위치 동기하여 동작하는 기능입니다.
MC_CamIn(캠 동작 시작) 명령으로 캠 동작을 시작하고 MC_CamOut(캠 동작 해제) 명령 및 MC_Stop (강제 정지) 명령으로 캠 동작을 해제합니다.
캠 테이블은 Sysmac Studio의 캠 에디터 기능을 사용하여 작성하고 CPU 유니트 본체에 다운로드해 둡니다.
CPU 유니트에 다운로드하는 경우는 Sysmac Studio의「Synchronization」기능을 사용합니다.





유니트 버전 Ver.1.06 이상인 CPU 유니트와 Sysmac Studio Ver.1.07 이상의 조합에서는 MC_CamIn (캠 동작 시작) 명령 실행 중에 블렌딩을 지정하여 별도의 MC_CamIn(캠 동작 시작) 명령을 다중 기동하면 종축을 정지시키지 않고 캠 테이블을 바꾸어가며 작동시킬 수 있습니다.

캠 동작에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_CamIn (캠 동작 시작) 명령 및 MC_CamOut(캠 동작 해제) 명령, MC_Stop(강제 정지) 명령을 참조해 주십시오.
또한, 캠 에디터 기능에 대한 상세한 내용은 🕮『Sysmac Studio Version 1 오퍼레이션 매뉴얼 (SBCA-470)』을 참조해 주십시오.

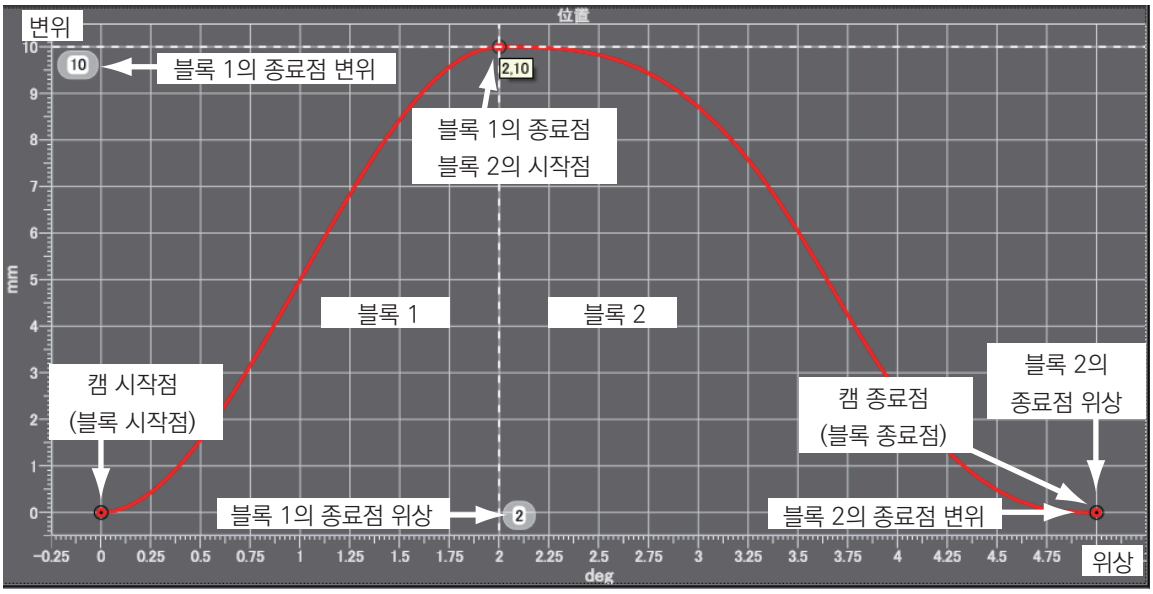
9-2-5 캠 테이블

여기에서는 캠 동작에서 사용하는 캠 테이블에 대해 설명합니다.

캠 테이블의 용어

용어	설명
캠 동작	주축 1개와 종축 1개를 가지고 캠 프로파일 곡선을 따라 주축의 위상에서 종축의 변위를 구해 움직이게 하는 기능입니다.
캠 프로파일 곡선	캠 동작을 실현하기 위한 위상과 변위의 관계를 나타내는 곡선입니다. Sysmac Studio의 캠 에디터에서 작성합니다. 캠 프로파일 곡선을 CPU 유니트에 다운로드 하면 사용자 프로그램에서 캠 데이터 변수로 취급할 수 있습니다. CPU 유니트에 다운로드하는 경우는 Sysmac Studio의 「Synchronization」 기능을 사용합니다.
캠 블록	캠 곡선을 선택할 수 있는 구간입니다. 이전 캠 블록의 종료점과 자체 캠 블록의 종료점 사이를 나타냅니다.
캠 곡선	캠 특성을 나타내는 곡선입니다. 캠 블록별로 캠 곡선을 선택할 수 있습니다. 지정한 점에서 Sysmac Studio가 산출하는 위상 폭, 변위 폭을 이용하여 실제 캠 프로파일 곡선을 Sysmac Studio가 작성합니다. 직선, 등가속도, 사다리꼴(Trapecloid) 곡선 등이 있습니다.
캠 데이터	캠 동작을 실현하기 위한 위상(주축)과 변위(종축)로 구성되는 데이터입니다.
캠 데이터 변수	캠 데이터를 구조체 배열화한 변수입니다.
캠 테이블	캠 데이터의 데이터 테이블을 나타냅니다. 위상이 오름차순이 아니면 올바르지 않은 캠 테이블로 간주합니다.
캠 시작점	캠 데이터의 선두 데이터를 나타냅니다.
캠 종료점	캠 데이터의 유효 캠 데이터의 마지막을 나타냅니다. 캠 종료점 < 캠 데이터 수의 경우 캠 종료점보다 뒤에 있는 위상과 변위는 「0」입니다.
캠 블록 시작점	캡 블록의 시작점입니다. 캠 동작의 시작부에서는 「캠 시작점」과 동일한 점이 됩니다. 캠 프로파일 곡선이 연속일 때는 「캠 블록 종료점」과 동일한 점이 됩니다.
캠 블록 종료점	캠 블록의 종료점입니다. 캠 동작의 종료부에서는 「캠 종료점」과 동일한 점이 됩니다. 캠 프로파일 곡선이 연속할 때는 「캠 블록 시작점」과 동일한 점이 됩니다. 캠 블록 종료점은 (가로축, 세로축) = (위상 종료점, 변위 종료점)으로 정의됩니다.
오리지널 캠 데이터	캠 에디터로 캠 프로파일 곡선을 분할하여 작성한 캠 데이터입니다.
프로그램 변경 캠 데이터	CPU 운전 중에 사용자 프로그램으로 변경한 캠 데이터입니다.
주축	캠 동작의 입력이 되는 축으로 리니어 모드/로터리 모드 모두 지정할 수 있습니다.
종축	캠 동작의 출력이 되는 축으로 리니어 모드/로터리 모드 모두 지정할 수 있습니다.
위상	캠 테이블 시작점 위치에서의 주축의 상대량입니다.
변위	주축 추종 거리에서의 종축의 상대량입니다.
유효 캠 데이터	캠 시작점 이외에서 위상이 「0」이 아닌 수치를 가진 캠 데이터입니다.
무효 캠 데이터	캠 시작점 이외에서 위상이 「0」인 캠 데이터입니다.
유효 캠 데이터 수	캠 데이터의 세트 수를 나타냅니다
최대 캠 데이터 수	캠 테이블을 가질 수 있는 캠 데이터의 최대 세트 수입니다.
캠 데이터 인덱스	캠 데이터가 실행되는 번호입니다.
캠 테이블 시작 위치	캠 시작점(위상 = 0)에 해당하는 주축의 절대 위치입니다.
주축 추종 거리	종축이 캠 동작을 시작하는 주축 위치에서 절대 위치 또는 상대 위치로 나타냅니다. 상대 위치는 캠 시작점 위치가 기준입니다.
시작 위치 방식 지정	주축 추종 거리를 절대 위치로 나타낼지 상대 위치로 나타낼지 지정합니다.
비어 있는 캠 데이터	종료점 뒤로 설정할 수 있는 위상과 변위가 「0」인 캠 데이터입니다.
접속 속도	캠 프로파일 곡선 간을 접속할 때의 속도. 지정할 수 없는 곡선도 있습니다.
접속 가속도	캠 프로파일 곡선 간을 접속할 때의 가속도입니다. 지정할 수 없는 곡선도 있습니다.

용어	설명
위상 간격 폭	캠 프로파일 곡선을 위상(가로축)으로 분할할 때의 폭입니다. 위상 간격 폭으로 곡선을 분할했을 때의 각 점이 캠 테이블의 캠 데이터가 됩니다. 반드시 캠 블록별로 지정해야 합니다.

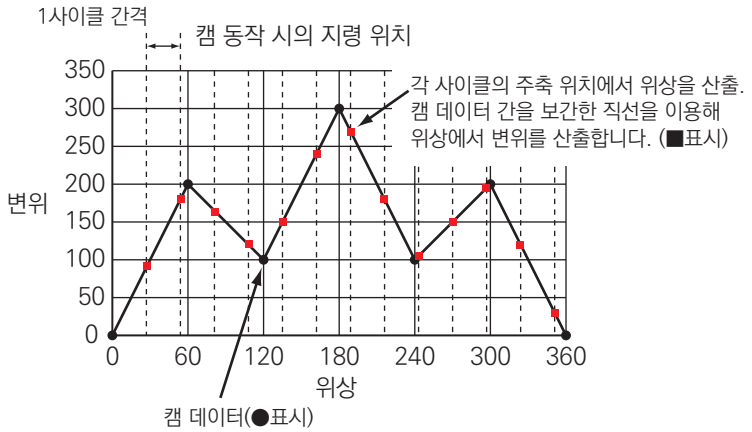


캠 테이블이란

MC 기능 모듈에서는 주축의 위상과 종축의 변위로 이루어진 1대의 데이터를 캠 데이터, 복수의 캠 데이터 조합을 캠 테이블로 정의합니다.
캠 테이블은 Sysmac Studio의 캠 에디터에서 작성합니다 또한, 사용자 프로그램에서 캠 테이블 내의 캠 데이터를 변경할 수 있습니다.
캠 테이블에 있는 캠 데이터의 위상과 변위의 값은 「0.0」인 시작점에서의 상대적인 양으로 나타냅니다.
캠 동작 중에는 주축의 위상에 인접하는 2개의 캠 데이터를 직선으로 보간한 변위가 종축에 대한 지령 위치가 됩니다.
캠 테이블의 캠 데이터 수를 늘리면 궤도 정밀도가 높고 매끄러운 캠 프로파일 곡선이 됩니다.

캠 테이블

	위상	변위	캠 데이터 인덱스
캠 시작점	0	0	0
캠 데이터	60	200	1
	120	100	2
	180	300	3
	240	100	4
	300	200	5
캠 종료점	360	0	6



사용상의 주의

- 캠 테이블의 캠 데이터는 위상의 값이 오름차순이 되도록 작성합니다. 위상이 오름차순이 아닌 경우에는 캠 동작 명령을 실행할 때 명령에서 이상이 발생합니다.
- 캠 데이터 변수는 글로벌 변수로, 여러 개의 태스크에서 캠 데이터 변수의 값을 참조 및 변경할 수 있습니다. 여러 개의 태스크에서 캠 데이터 변수의 값을 변경하는 경우 복수의 태스크에 의한 갱신 처리로 경합이 발생하지 않도록 프로그램을 작성해 주십시오.
- 「글로벌 변수의 태스크 간 배타 제어」 기능을 이용하여 캠 데이터 변수를 배타 제어하는 경우 배타 대상인 캠 데이터 변수로 모션 명령을 기동하지 마십시오. 경도 폴트 레벨 이상 「Incorrect Cam Table Specification(이벤트 코드: 54390000 Hex)」이 발생합니다.

캠 테이블 사양

항목	설명
캠 테이블 1개당 최대 캠 데이터 수	65,535점
전체 캠 데이터의 사이즈 상한	1,048,560점 *1
캠 테이블 수의 상한	640테이블 *2
캠 동작 변환	모션 제어 명령을 이용하여 다른 캠 동작으로 변환 가능
캠 데이터 변경	사용자 프로그램에서 편집 가능 캠 테이블 생성 명령을 이용하여 갱신 가능 *3
캠 데이터 저장	캠 테이블 저장 명령을 이용하여 불휘발성 메모리에 저장 가능
캠 데이터에 부속되는 정보	캠 에디터 표시용 정보를 다운로드, 업로드 가능 *4
캠 데이터가 메인 메모리에 반영되는 타이밍	・Sysmac Studio에서 다운로드할 때 ・전원 투입 시

*1. 캠 테이블 1개가 65,535점인 경우 최대 16캠이 됩니다. 0.1deg 분해능의 경우 캠 테이블 1개는 3600점이 되고 최대 291캠이 됩니다.
*2. 합계 10MB 이하로 해 주십시오.
*3. 캠 테이블 생성 명령은 유니트 버전 Ver.1.08 이상인 CPU 유니트와 Sysmac Studio Ver.1.09 이상의 조합으로 사용할 수 있습니다.
*4. 「다운로드」 및 「업로드」는 Sysmac Studio의 「Synchronization」 기능을 사용합니다.

캠 테이블의 데이터형

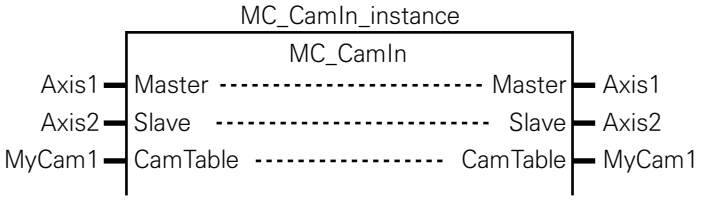
캠 테이블은 캠 데이터 구조체를 배열 선언하여 사용합니다. 캠 데이터 구조체는 사전에 다음과 같이 형(Type) 선언되어 있습니다.

```
TYPE
        (*캠 데이터 구조체*)
        _sMC_CAM_REF :
        STRUCT
                Phase       : REAL;         (*위상*)
                Distance    : REAL;         (*변위*)
        END_STRUCT;
END_TYPE
```

Sysmac Studio의 캠 에디터로 캠 데이터를 작성하고 캠 테이블명과 캠 데이터 수(배열 사이즈)를 지정해야 합니다.
예를 들어 캠 테이블명을「MyCam1」, 캠 데이터 수를 1000개로 할 경우 다음과 같이 선언합니다.

```
VAR
        (*캠 테이블*)
        MyCam1      :       ARRAY [0..999] OF _sMC_CAM_REF;
END_VAR
```

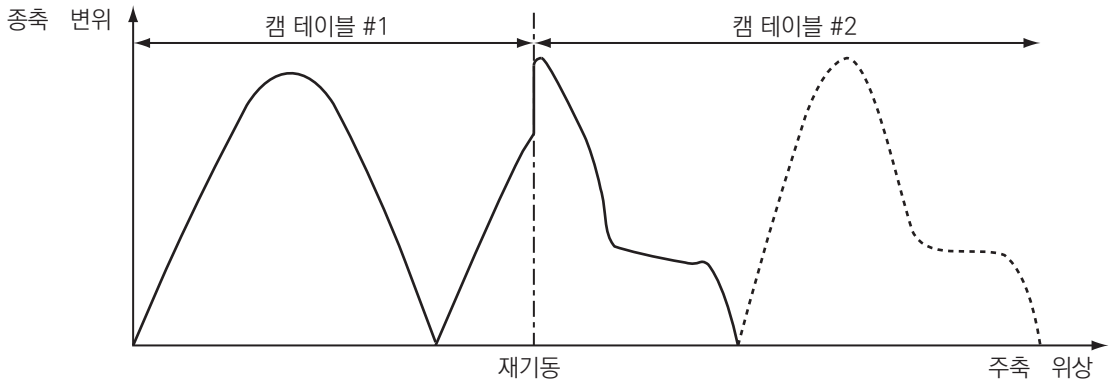
캠 동작 명령으로「MyCam1」을 지정할 경우 다음과 같이 기술합니다. 여기에서는 주축을 Axis1, 종축을 Axis2로 합니다.



지정한 캠 테이블이 컨트롤러에 존재하지 않으면 이상을 감지합니다. 또한, 복수의 축에서 동일한 캠 테이블을 지정할 수 있습니다.

캠 테이블 변환

캠 동작 중에 캠 동작 명령을 재기동하면 캠 테이블을 변환할 수 있습니다.
변환 후에는 재기동 시에 지정한 캠 테이블로 캠 동작을 실행합니다.
MC_CamIn(캠 동작 시작) 명령의 출력 변수「EndOfProfile(종료점 실행)」및「Index(인덱스)」는 변환한 캠 테이블에 따라 출력됩니다.



사용상의 주의

- 변환하고 싶은 캠 테이블은 미리 불휘발성 메모리에 저장해 두어야 합니다.
- 캠 동작 중에 캠을 변환하면 속도가 연속되지 않습니다. 과도한 속도 불연속이 되지 않도록 변환 타이밍을 조정해 주십시오.

캠 데이터의 판독/기록과 캠 테이블의 저장

다른 변수와 같이 사용자 프로그램에서 캠 데이터를 판독/기록할 수 있습니다.
예를 들어 캠 테이블명이 「MyCam1」인 첫 번째 배열 요소에 대해서 위상을 「MyCam1[0].Phase」, 변위를 「MyCam1[0].Distance」로 지정할 수 있습니다.
사용자 프로그램에서 갱신한 캠 데이터는 MC_SaveCamTable(캠 테이블 저장) 명령을 이용하여 CPU 유니트의 불휘발성 메모리에 캠 테이블로 저장할 수 있습니다.

사용상의 주의

- 캠 테이블 저장 명령이 기동되기 전에 CPU 유니트의 전원이 OFF되거나 본 명령에 의한 저장이 정상으로 완료되지 않은 경우 또는 Sysmac Studio에서 다운로드를 실행한 경우에는 갱신된 캠 데이터가 소멸됩니다.
- CPU 유니트상의 사용자 프로그램에서 캠 데이터를 갱신하는 경우에는 갱신 데이터가 소실되지 않도록 주의해 주십시오.
- 불휘발성 메모리에 저장한 캠 데이터는 Sysmac Studio의 업로드 기능으로 가져올 수 있습니다.
- 「다운로드」 및 「업로드」는 Sysmac Studio의 「Synchronization」 기능을 사용합니다.

배열 지정에 대해서는 📖『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

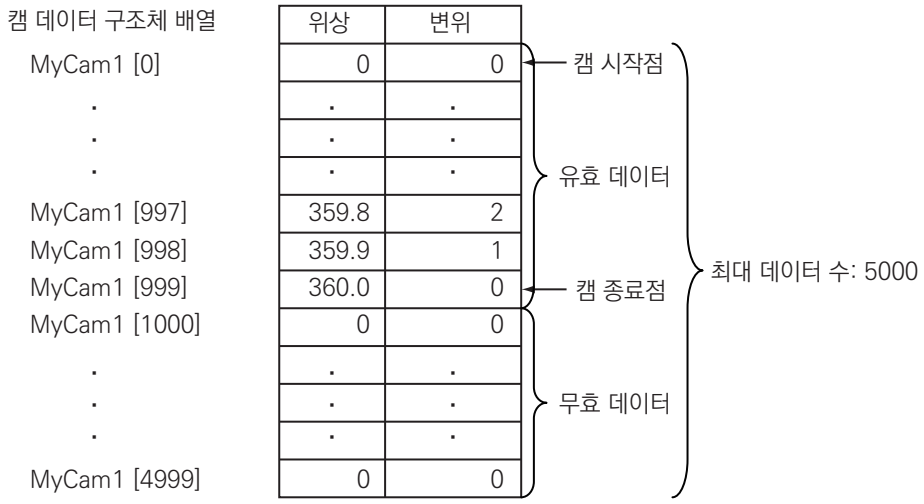
캠 테이블 저장 명령에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_SaveCamTable(캠 테이블 저장) 명령을 참조해 주십시오.

캠 테이블의 속성 갱신

MC 기능 모듈에서는 캠 테이블의 캠 종료점을 식별해야 합니다.
캠 동작 중에 사용자 프로그램에서 유효 캠 데이터 수를 변경하는 갱신을 실행한 경우에는 유효 캠 데이터 수를 최신으로 갱신해야 합니다.
이를 위해 MC_SetCamTableProperty(캠 테이블 속성 갱신) 명령을 사용합니다.

캠 종료점이란 캠 테이블을 시작점부터 순서대로 검색했을 때 위상이 처음으로 「0」이 되는 캠 데이터의 바로 앞에 배열된 데이터를 말합니다. 위상 「0」을 검출한 이후의 캠 데이터는 무효가 됩니다.

예를 들면 다음과 같은 캠 테이블의 경우 MC_SetCamTableProperty(캠 테이블 속성 갱신) 명령의 출력 변수 「EndPointIndex(종료점 인덱스)」는 「999」, 「MaxDataNumber(최대 데이터 수)」는 「5000」이 됩니다.



사용상의 주의

- 최대 캠 데이터 수는 사용자 프로그램에서 변경할 수 없습니다.
- 유효 캠 데이터 수를 변경하는 갱신을 실행한 후에는 본 명령을 실행해 주십시오.
  유효 캠 데이터 수를 갱신하지 않으면 캠 동작 및 MC_CamIn의 EndOfProfile(캠 사이클 완료)이 의도한 동작을 하지 않습니다.

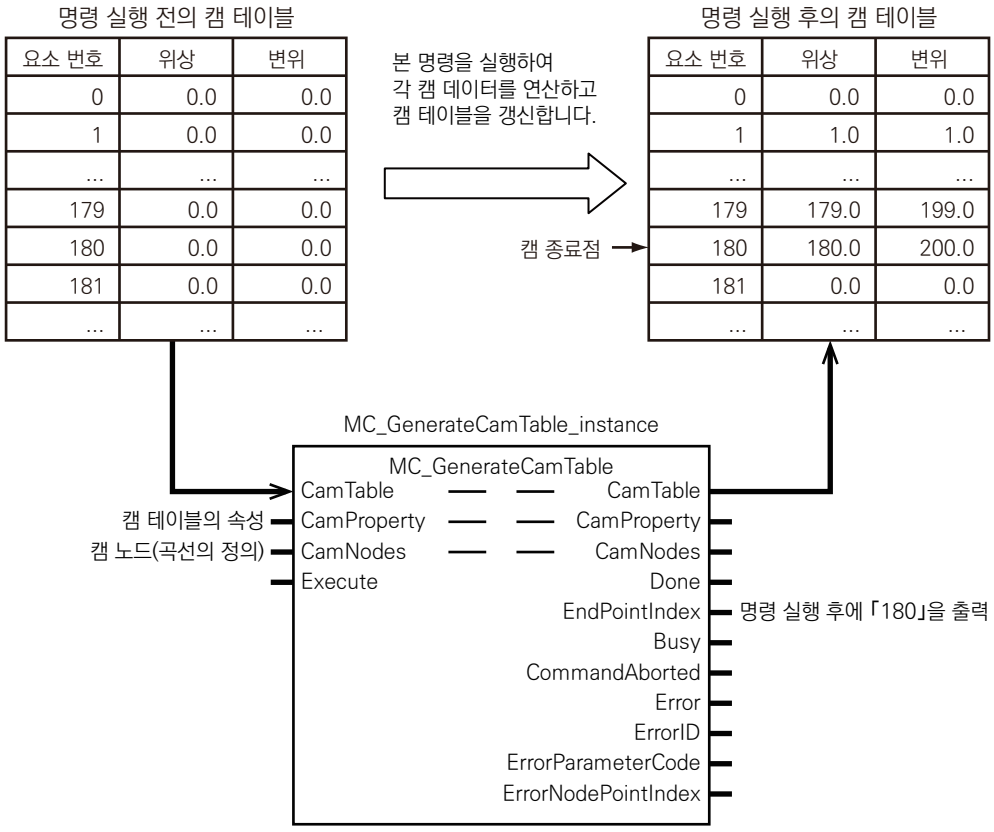
캠 테이블 속성 갱신에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_SetCamTableProperty(캠 테이블 속성 갱신) 명령을 참조해 주십시오.

캠 테이블 생성

유니트 버전 Ver.1.08 이상인 CPU 유니트와 Sysmac Studio Ver.1.09 이상을 조합하면 사용자 프로그램 내 MC_GenerateCamTable(캠 테이블 생성) 명령으로 캠 테이블을 생성할 수 있습니다.

MC_GenerateCamTable(캠 테이블 생성) 명령의 CamProperty(캠 속성)와 CamNodes(캠 노드)에 지정한 값에 따라 캠 데이터의 값을 연산하고 명령의 입출력 변수 CamTable(캠 테이블) 에 지정한 캠 데이터 변수를 갱신합니다.

갱신이 완료되면 캠 테이블의 종료점 인덱스를 갱신하고 캠 종료점의 요소 번호를 EndPointIndex(종료점 인덱스)에 출력합니다.
MC_GenerateCamTable(캠 테이블 생성) 명령 완료 후에 MC_SetCamTableProperty(캠 테이블 속성 갱신) 명령을 실행할 필요는 없습니다.



캠 데이터 변수는 캠 데이터 구조체 _sMC_CAM_REF형의 배열 변수로, Sysmac Studio의 캠 에디터에서 작성합니다.
CamProperty(캠 속성)에는 캠 속성 변수를 지정합니다 캠 속성 변수는 캠 속성 구조체 _sMC_CAM_PROPERTY형의 변수로, Sysmac Studio의 글로벌 변수 테이블에서 사용자 정의로 작성하거나 또는 Sysmac Studio의 캠 데이터 설정을 이용하여 작성합니다.
CamNodes(캠 노드)에는 캠 노드 변수를 지정합니다. 캠 노드 변수는 캠 노드 구조체 _sMC_CAM_NODE형의 배열 변수로, Sysmac Studio의 글로벌 변수 테이블에서 사용자 정의 변수로 작성하거나 Sysmac Studio의 캠 데이터 설정을 이용하여 작성합니다.
캠 속성 변수와 캠 노드 변수를 총칭하여 「캠 정의 변수」라 합니다.

캠 정의 변수를 사용자 정의 변수로 작성한 경우 변수의 유지 속성 초기값은 「Non-retain」입니다.
변수값을 변경하여 프로그램 모드로 변환한 후나 전원 재투입 후에 다시 사용할 때는 변수의 유지 속성을 「Retain」로 설정해 주십시오. 표시기 등에서 매번 설정할 때는 「Non-retain」 상태라도 괜찮습니다.
캠 정의 변수를 Sysmac Studio의 캠 데이터 설정으로 작성한 경우 변수의 유지 속성은 「Retain」로 고정됩니다.

MC_GenerateCamTable(캠 테이블 생성) 명령에 지정할 값을 표시기 등에서 설정하면 Sysmac Studio를 사용하지 않고 캠 데이터 변수를 작성하거나 캠 동작을 조정할 수 있습니다.
이어서 캠 동작을 조정하는 순서의 개요에 대해 설명합니다.

1 **다음과 같은 처리를 미리 사용자 프로그램으로 작성해 둡니다.**
- 표시기에서 설정한 캠 정의 변수의 값을 캠 테이블 생성 명령에 지정하는 처리
- 캠 테이블 생성 명령으로 생성된 캠 데이터 변수를 표시기에 그래프 표시하는 처리
- EndPointIndex(종료점 인덱스)의 값을 표시기에 표시하는 처리

2 **캠 정의 변수의 값을 표시기로 설정합니다.**

3 **캠 테이블 생성 명령을 실행합니다.**

4 **생성된 캠 테이블의 형상과 종료점 인덱스의 값을 표시기에서 확인합니다.**

5 **캠 테이블의 외형과 캠 데이터 수에 문제가 없으면 캠 동작을 실행합니다.**

6 **캠 동작의 결과를 확인하고 캠 정의 변수값의 변경을 검토합니다**

7 **2~6의 순서를 반복합니다.**

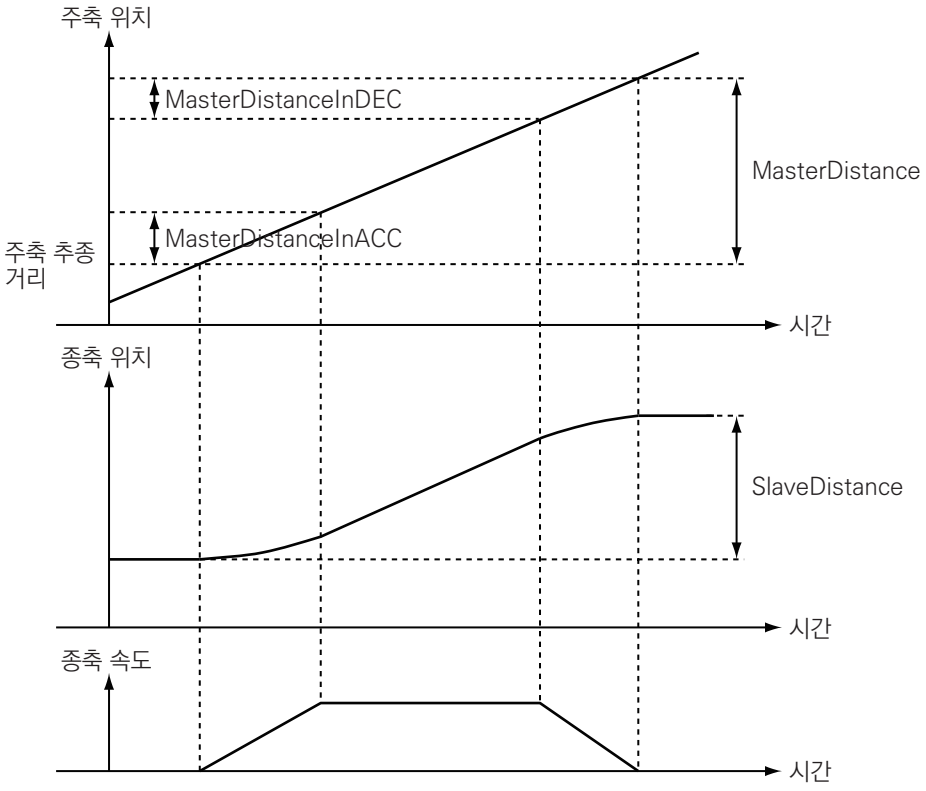
캠 정의 변수와 캠 테이블 생성 명령에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편 (SBCE-364)』의 MC_GenerateCamTable(캠 테이블 생성) 명령을 참조해 주십시오.

Sysmac Studio에서 캠 정의 변수를 작성, 전송하는 내용에 대해서는 📖『Sysmac Studio Version 1 오퍼레이션 매뉴얼(SBCA-470)』을 참조해 주십시오.

9-2-6 사다리꼴 패턴 캠 동작

지정한 종축이 지정한 주축에 동기하면서 사다리꼴 커브로 위치 결정을 하는 기능입니다.
전자 캠의 일종이지만 캠 에디터에서 작성한 캠 테이블은 사용하지 않습니다.
MC_MoveLink(사다리꼴 패턴 캠) 명령을 기동하여 동작을 시작합니다.
동작 중인 축을 정지하려면 MC_Stop(강제 정지) 명령을 사용합니다.
동작 대상인 축을「Slave(종축)」로 하고「Master(주축)」,「MasterDistance(주축 이동 거리)」,「MasterDistanceInACC(주축 가속 이동 거리),「MasterDistanceInDEC(주축 감속 이동 거리)」,「SlaveDistance(종축 이동 거리)」,「MasterStartDistance(주축 추종 거리)」를 지정합니다.
주축에는 지령 위치 또는 피드백 위치를 지정할 수 있습니다.
동기 동작의 시작 조건은「명령 시작 시」,「트리거 검출 시」,「주축이 주축 추종 거리에 도달 시」 중에서 지정할 수 있습니다.

종축의 속도와 위치는 아래 그림과 같이 주축과 종축의 이동량 비율로 결정됩니다.
아래 그림의 동기 시작 위치는 동기 시작 조건이 유효해졌을 때의 위치를 나타냅니다.

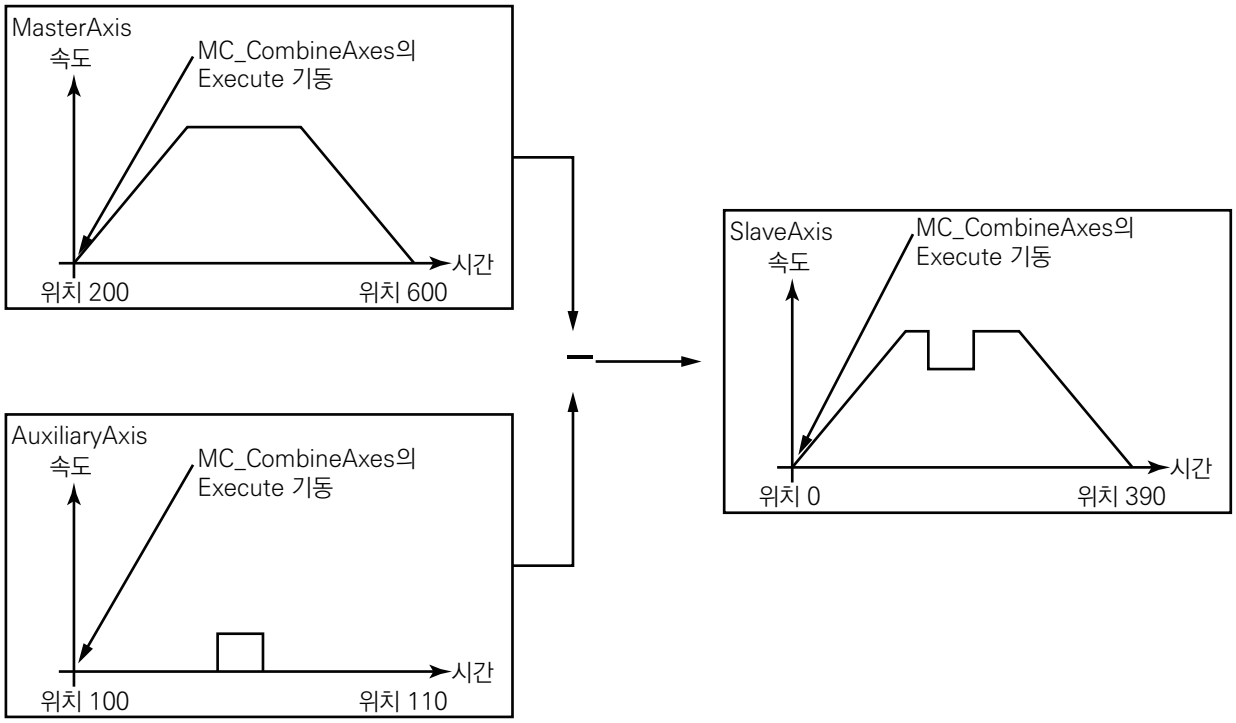


사다리꼴 패턴 동작에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_MoveLink(사리리꼴 패턴 캠) 명령 및 MC_Stop(강제 정지) 명령을 참조해 주십시오.

9-2-7 가감산 위치 결정

2개의 위치를 가산하거나 감산한 값을 종축의 지령 위치로 하는 기능입니다.
MC_CombineAxes(가감산 위치 결정) 명령을 기동하여 동작을 시작합니다.
동작 중인 축을 정지하려면 MC_Stop(강제 정지) 명령을 사용합니다.

아래 그림은 감산한 경우의 동작 예입니다.
SlaveAxis(종축) 지령 현재 위치 = MasterAxis(주축) 지령 현재 위치 - AuxiliaryAxis(보조 축) 지령 현재 위치

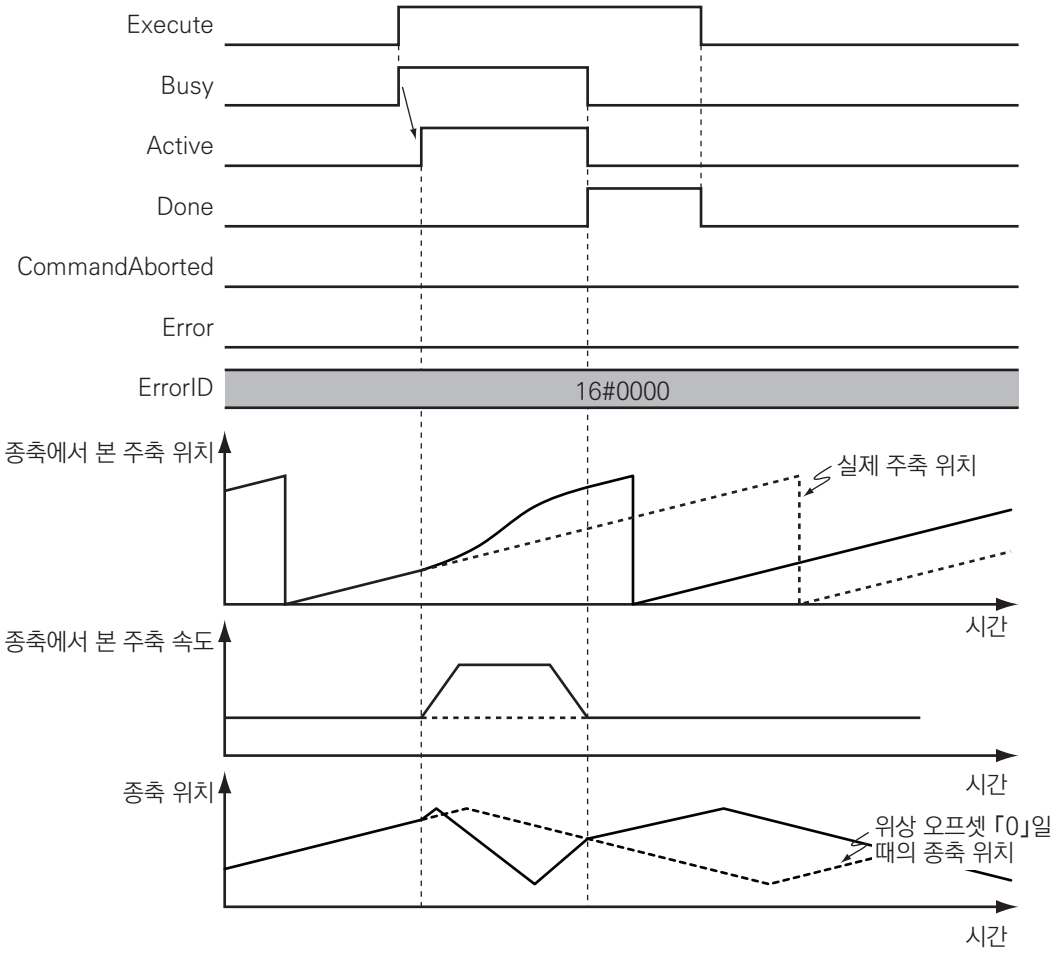


가감산 위치 결정 동작에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_CobineAxes(가감산 위치 결정) 명령 및 MC_Stop(강제 정지) 명령을 참조해 주십시오.

9-2-8 주축 위상 보정

동작 중인 명령에 대해 종축에서 본 주축의 위상을 보정하는 기능입니다.
종축에서 본 보정량은 상대량이며 동기 상태인 동안에 종축은 주축의 상대량에 대해 동기합니다.
MC_Phasing(주축 상대값 위상 보정) 명령을 기동하면 동기 제어 명령에 대해 위상을 보정할 수 있습니다.

MC_Phasing(주축 상대값 위상 보정) 명령은「위상 보정량」,「목표 속도」,「가속도」,「감속도」,「저크」를 지정할 수 있습니다.



주축 위상 보정이 가능한 동기 제어 명령 및 주축 위상 보정 기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_Phasing(주축 상대값 위상 보정) 명령을 참조해 주십시오.

9-2-9 종축 위치 보정

동기 제어 중인 종축에 대해 위치 보정하는 기능입니다.
동기 제어 중인 종축에 대해 입력 변수값을 바탕으로 연산한 오프셋양을 지령 현재 위치에 가산하여 서보 드라이버에 출력합니다.
MC 기능 모듈이 2축에 대해 동일한 양의 이동량을 지령해도 기계적인 변형 등으로 인해 이동량이 다를 때 보정할 수 있습니다.
MC_SyncOffsetPosition(사이클릭 동기 위치 오프셋 보정) 명령을 기동하면 동기 제어 중인 종축의 위치 보정을 실행할 수 있습니다.

종축 위치 보정 기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_SyncOffsetPosition(사이클릭 동기 위치 오프셋 보정) 명령을 참조해 주십시오.

9-2-10 멀티 모션에서 동기 제어 실현하는 방법

MC 기능 모듈의 표준 기능에서는 멀티 모션에서 다른 태스크에 할당한 축 간 동기 제어 명령을 실행하면 「주축 지정 오류(이벤트 코드: 54620000 Hex)」가 발생합니다.
그러나 MC_PeriodicSyncVariables(축 변수 태스크 간 정주기 동기) 명령을 사용하면 프라이머리 정주기 태스크에서 제어하는 주축과 정주기 태스크(실행 우선도 5)에서 제어하는 종축을 동기 제어할 수 있습니다.

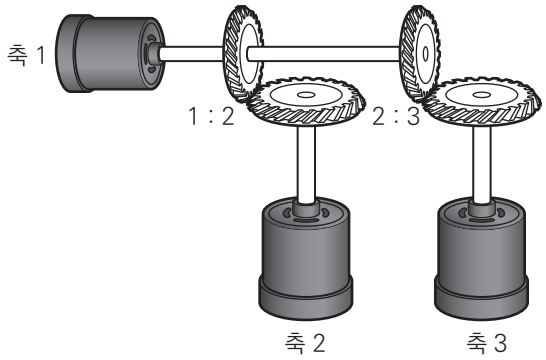
주축과 종축이 동기하는 캠 동작과 기어 동작은 다음과 같은 조합으로 사용할 수 있습니다.

주축 태스크	종축 태스크	
	프라이머리 정주기 태스크	정주기 태스크(실행 우선도 5)
프라이머리 정주기 태스크	모션 제어 명령에 의한 동기	프라이머리 정주기 태스크에서 MC_PeriodicSyncVariables(축 변수 태스크 간 정주기 동기) 명령을 실행하고 정주기 태스크(실행 우선도 5)에서 「가상 주축」을 사용하여 실현 가능.
정주기 태스크(실행 우선도 5)	사용할 수 없습니다	모션 제어 명령에 의한 동기

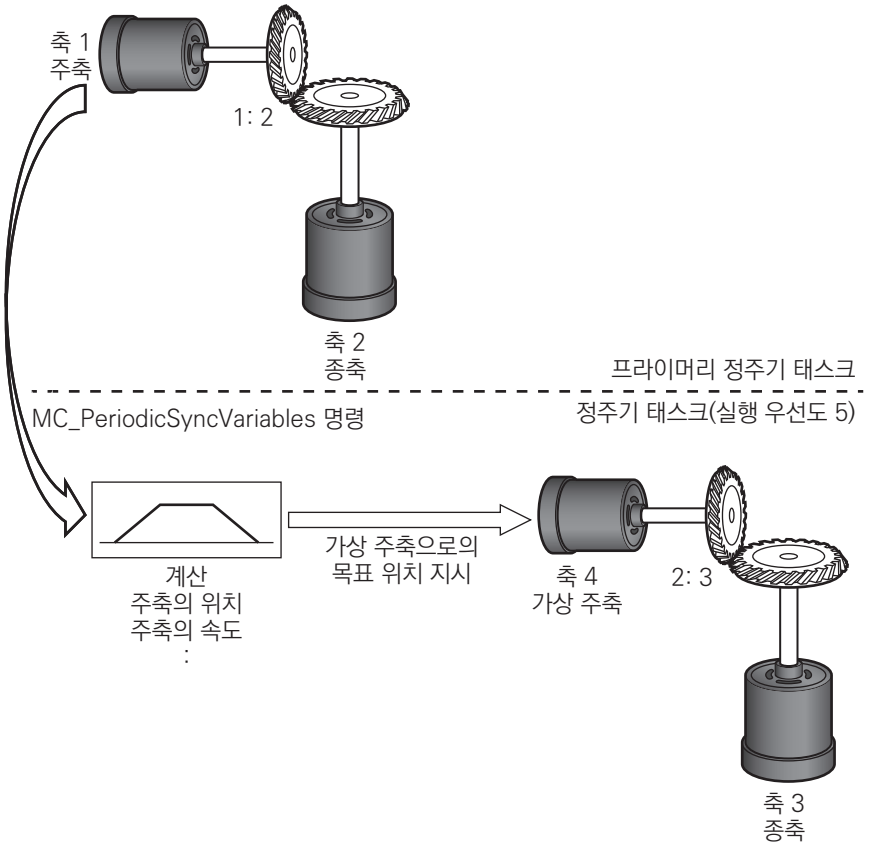
동작 예의 축 구성

아래 그림과 같이 주축(축 1)에 대해 고속으로 고정밀도가 필요한 종축(축 2)을 프라이머리 정주기 태스크에 할당하고 정밀도가 필요하지 않은 종축(축 3)을 정주기 태스크(실행 우선도 5)의 태스크에 할당하여 제어합니다. 또한, 주축(축 1)은 프라이머리 정주기 태스크에 할당합니다.

● **물리적인 축 구성**



● **논리적인 축 구성**



프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)에 각각 프로그램을 편성하면 상기 어플리케이션 동작을 실현합니다.

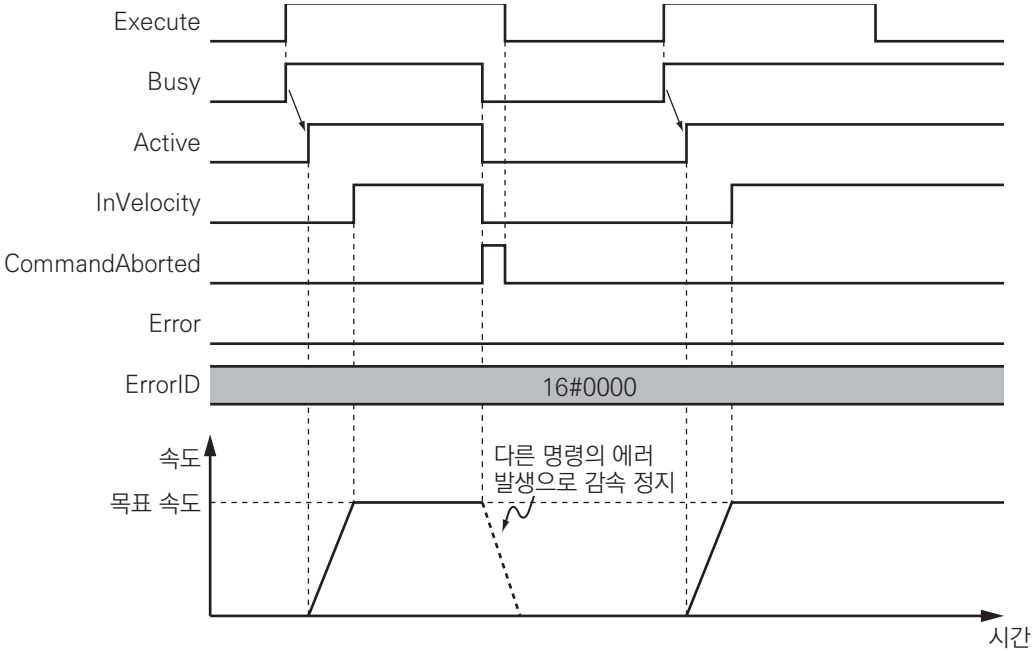
MC_PeriodicSyncVariables(축 변수 태스크 간 정주기 동기) 명령의 상세한 설명은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

9-3 단축 속도 제어

여기에서는 단축의 속도 제어 동작에 대해 설명합니다.
NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

9-3-1 속도 제어

속도 제어란 지령한 속도로 축을 계속 보내는 기능입니다.
가속도, 감속도 및 저크도 지정할 수 있습니다.
축을 정지하려면 MC_Stop(강제 정지) 명령을 사용하거나 다른 동작 명령을 기동합니다.
목표 속도로「0」을 지정하면 축은 이동하지 않지만, 축의 상태는 동작 중인 것으로 나타납니다.
속도 제어 중에 다른 모션 제어 명령의 다중 기동으로 인터럽트된 경우에는 목표 속도에 도달한 후의 상태에서 동작을 변환합니다.



MC 기능 모듈에서는 서보 드라이버 등을 위치 제어 모드로 지정한 목표 속도가 되도록 목표 위치를 지령합니다.
서보 드라이버 등에서는 위치 제어 루프가 유효하므로 외란 등으로 인해 지령 속도보다 느려져 위치 편차가 커지면 위치 편차를 없애기 위한 속도 변동이 발생합니다.

기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의
MC_MoveVelocity(속도 제어) 명령을 참조해 주십시오.

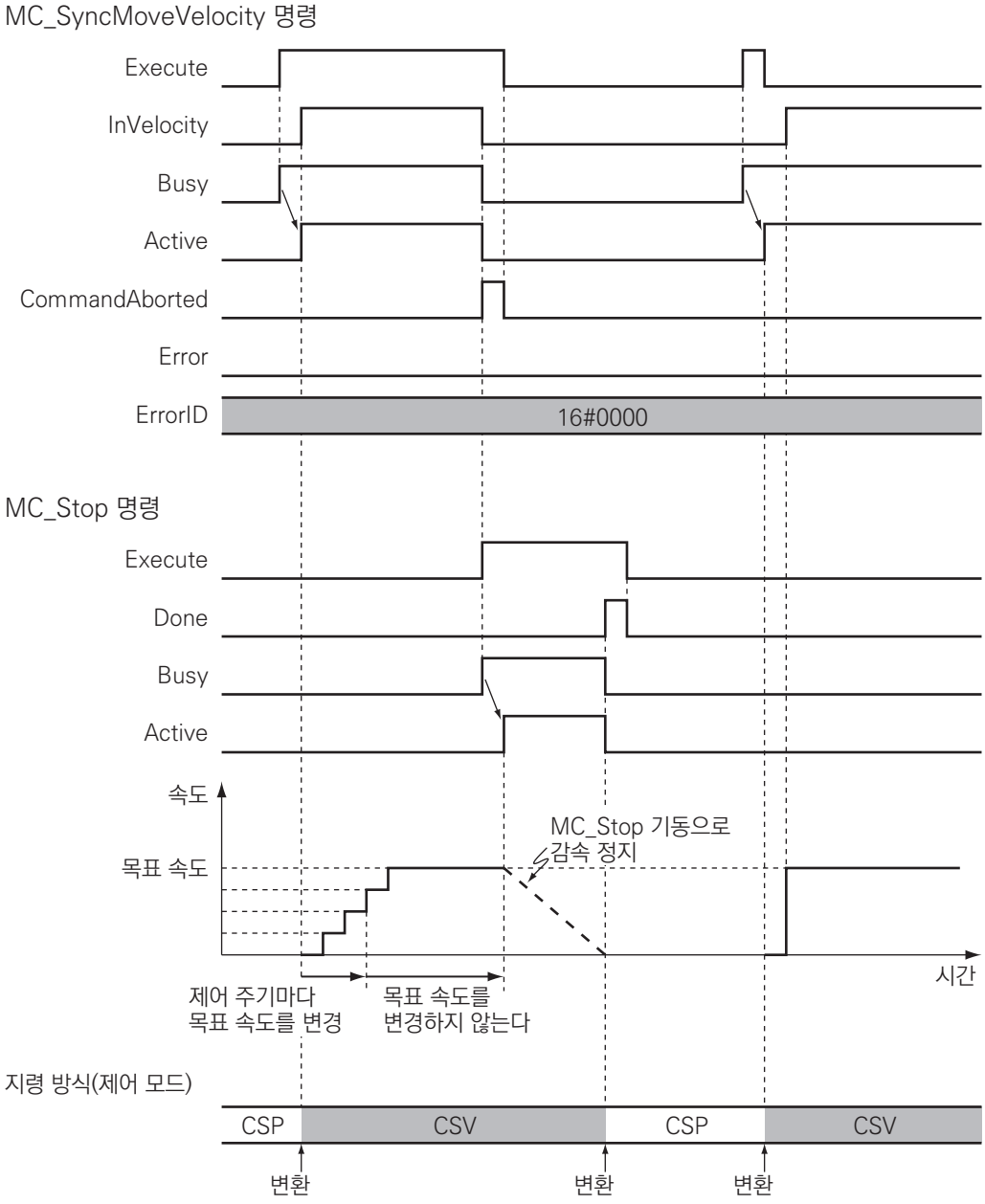
9-3-2 사이클릭 동기 속도 제어

서보 드라이버의 제어 모드를 속도 제어 모드로 하여 제어 주기별로 지령 속도를 출력하는 기능입니다.

사용상의 주의

NX 시리즈 펄스 출력 유니트를 사용할 경우에는 이 기능을 사용할 수 없습니다.

축을 정지하려면 MC_Stop(강제 정지) 명령을 사용하거나 다른 동작 명령을 기동합니다.
목표 속도로 「0」을 지정하면 축은 이동하지 않지만, 축의 상태는 동작 중인 것으로 나타납니다.



서보 드라이버 측은 속도 제어 루프에서 지령을 받으므로 외란 등으로 인해 지령 속도보다 느려져도 위치 편차를 없애기 위한 속도 변동은 발생하지 않습니다.

기능에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_SyncMoveVelocity(사이클릭 동기 속도 제어) 명령을 참조해 주십시오.

9-4 단축 토크 제어

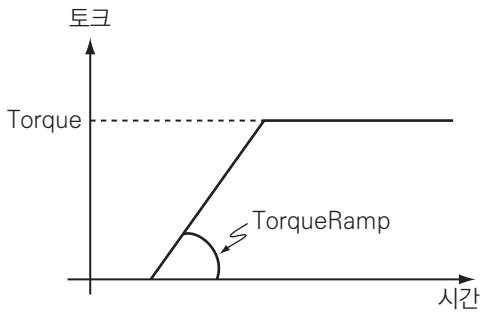
토크 제어란 지령한 토크를 계속 발생시키는 기능입니다.
Torque(목표 토크)에 도달할 때까지의 토크 변화율을 TorqueRamp(토크 램프)에서 지정할 수도 있습니다.

사용상의 주의

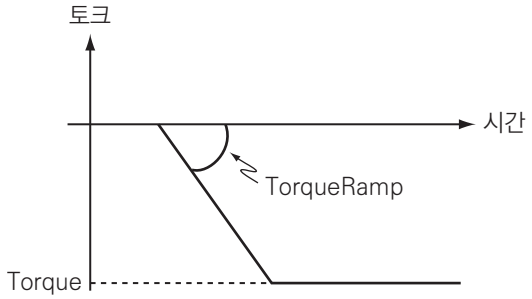
- 토크 제어 시에는 안전을 위해 반드시 속도 제한값을 설정해 주십시오.
- NX 시리즈 펄스 출력 유니트를 사용할 경우에는 이 기능을 사용할 수 없습니다.

축을 정지하려면 MC_Stop(강제 정지) 명령을 사용하거나 다른 동작 명령을 기동합니다.
Torque(목표 토크)로 「0」을 지정하면 축은 이동하지 않지만, 축의 상태는 동작 중인 것으로 나타납니다.

● **방향 지정 = 정방향의 경우**



● **방향 지정 = 역방향의 경우**



MC 기능 모듈에서는 서보 드라이버의 토크 제어 모드를 사용합니다.
서보 드라이버 측은 MC 기능 모듈의 토크 지령값을 토크 제어 루프에서 받아 제어합니다.
모션 제어 명령의 입력 변수 「Velocity(속도 제한)」로 서보 드라이버 측의 「속도 제한값」을 지정할 수 있습니다. 이를 통해 토크 제어 모드에서 모터의 부하가 가벼워진 경우와 같이 모터의 고속 회전을 제한할 수 있습니다.

기능에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의
MC_TorqueControl(토크 제어) 명령을 참조해 주십시오.

9-5 단축 제어의 공통 기능

여기에서는 단축 제어의 공통 기능에 대해 설명합니다.
NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

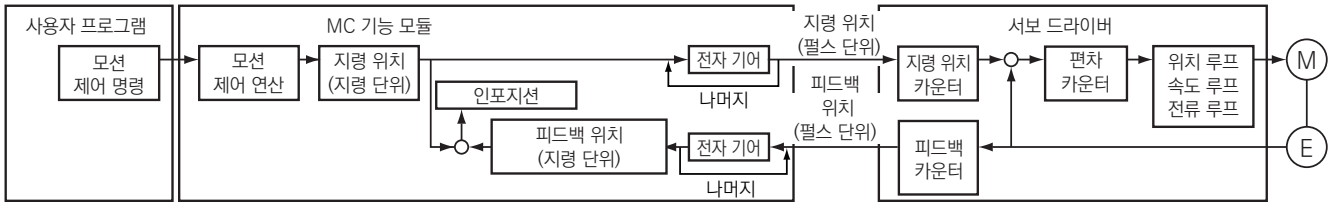
9-5-1 위치

위치의 종류

MC 기능 모듈에는 다음과 같이 2종류의 위치가 있습니다.

위치 종별	정의
지령 위치	MC 기능 모듈이 축을 제어하기 위해 출력하는 위치입니다.
피드백 위치	서보 드라이버 및 카운터에서 입력한 실제 위치입니다.

EtherCAT 슬레이브 서보 드라이버인 경우의 지령 위치와 피드백 위치의 관계는 아래 그림과 같습니다.



지령 위치와 피드백 위치는 아래의 항목을 공유합니다.

항목	지령 위치	피드백 위치
카운트 모드	리니어 모드 또는 로터리 모드를 설정합니다.	지령 위치와 동일한 카운트 모드가 됩니다.
위치의 단위	mm, μm, nm, inch, degree, pulse 중에서 1가지를 설정합니다.	지령 위치와 동일한 단위가 됩니다.
소프트웨어 리미트	소프트웨어적으로 동작 가능한 범위를 설정합니다.	지령 위치와 동일한 범위가 됩니다.
현재 위치 변경	현재 위치를 임의의 위치로 변경합니다.	지령 위치와 동일한 값이 동시에 설정됩니다. *1
원점 확정	원점 확정 또는 원점 미확정 상태입니다.	지령 위치와 동일한 원점 상태가 됩니다.

*1. 변경 전에 위치 편차가 있으면 위치 편차값을 유지한 값이 피드백 위치에 설정됩니다.

참고

NX 시리즈 위치 인터페이스 유니트의 경우는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

위치에 관한 축 파라메터

파라메터 명칭	기능	설정 범위	초기값
인포지션 폭	위치 결정 완료 폭을 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0	10
인포지션 체크 시간	위치 결정 완료 체크 시간을 밀리초 단위로 설정합니다. 「0」을 설정하면 원점 복귀 시에 원점을 확정했을 때에만 위치 결정 완료 체크를 계속하고, 원점 복귀 시 이외에는 위치 결정 완료 체크는 하지 않습니다. (단위: ms)	0~10000	0
소프트웨어 리미트 기능	소프트웨어 리미트의 기능을 선택합니다. 0: 무효 1: 지령 위치에 대해 유효하면 감속 정지 2: 지령 위치에 대해 유효하면 즉시 정지 3: 피드백 위치에 대해 유효하면 감속 정지 4: 피드백 위치에 대해 유효하면 즉시 정지	0~4	0
정방향 소프트웨어 리미트	정방향 측 소프트웨어 리미트값을 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 음수, 양수, 0	2147483647
역방향 소프트웨어 리미트	역방향 측 소프트웨어 리미트값을 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 음수, 양수, 0	-2147483648
위치 편차 초과값	위치 편차 초과 체크값을 설정합니다. 「0」을 설정하면 위치 편차 초과 체크가 무효해집니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0	0
위치 편차 경고값	위치 편차 경고 체크값을 설정합니다. 「0」을 설정하면 위치 편차 경고 체크가 무효해집니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0 (위치 편차 초과값 이하)	0

축 동작의 목표 위치 지정

실제 위치 결정 동작의 위치 및 거리는 모션 제어 명령의 입력 변수 「Position(목표 위치)」, 「Distance(이동 거리)」로 지정합니다.

위치 모니터

사용자 프로그램에서 축 변수를 판독하여 위치를 모니터할 수 있습니다.
또한, 변수명은 _MC_AX[*]를 예로 들어 설명하고 있으나 _MC1_AX[*], _MC2_AX[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_AX[0-255].Cmd.Pos	LREAL	지령 현재 위치	지령 위치의 현재값입니다. 서보 OFF 시나 위치 제어 모드 이외일 때는 피드백 현재 위치를 출력합니다.
_MC_AX[0-255].Act.Pos	LREAL	피드백 현재 위치	피드백 위치의 현재값입니다.

9-5-2 속도

속도의 종류

MC 기능 모듈에서 사용할 축의 속도에는 다음 2종류가 있습니다.

속도 종별	정의
지령 속도	MC 기능 모듈이 축을 제어하기 위해 출력하는 속도입니다.
피드백 속도	서보 드라이버 및 카운터에서 입력한 피드백 위치를 바탕으로 MC 기능 모듈 내에서 산출된 속도입니다. *1

*1. PDO 매핑에 피드백 속도(606CHex)를 매핑해서「Actual Current Velocity」기능에 할당한 경우에는 이 값을 표시합니다.

속도의 단위

속도의 단위는「지령 단위/s」입니다.
지령 단위는 위치의 표시 단위와 전자 기어로 단위 변환된 값입니다.

속도에 관한 축 파라메터

파라메터 명칭	기능	설정 범위	초기값
최고 속도	축의 최고 속도를 지정합니다. 축 동작 명령으로 최고 속도를 넘는 목표 속도가 지정된 경우에는 최고 속도로 동작합니다.	배정밀도 실수형의 양수	400,000,000
기동 속도 *1	각 축의 기동 속도를 설정합니다. 최고 속도를 초과하지 않도록 설정해 주십시오. (단위: 지령 단위/s)	배정밀도 실수형의 양수	0
조그 최고 속도	각 축의 조그 최고 속도를 설정합니다. *2 최고 속도를 초과하지 않도록 설정해 주십시오. (단위: 지령 단위/s)	배정밀도 실수형의 양수	1,000,000
속도 경고값	각 축의 속도 경고를 출력하기 위한 최고 속도에 대한 비율을 설정합니다. 「0」을 설정하면 속도 경고는 출력하지 않습니다. (단위: %)	0~100	0
피드백 속도 필터 시정수	피드백 속도의 이동 평균을 산출하는 기간을 밀리초 단위로 설정합니다. 「0」을 설정하면 이동 평균은 산출하지 않습니다. (단위: ms) 축 속도가 느릴 때 피드백 현재 속도의 편차를 줄이기 위해 사용합니다.	0~100	0

*1. 유니트 버전 Ver.1.05 이상인 CPU 유니트와 Sysmac Studio Ver.1.06 이상의 조합으로 사용할 수 있습니다.
*2. 조그 최고 속도를 초과하는 속도 지령값을 지정한 경우에는 조그 최고 속도로 지령합니다.

축 동작의 목표 속도 지정

실제 위치 결정 동작의 속도는 모션 제어 명령의 입력 변수「Velocity(목표 속도)」로 지정합니다.

속도 모니터

사용자 프로그램에서 축 변수를 판독하여 속도를 모니터할 수 있습니다.
또한, 변수명은 _MC_AX[*]를 예로 들어 설명하고 있으나 _MC1_AX[*], _MC2_AX[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_AX[0-255].Cmd.Vel	LREAL	지령 현재 속도	지령 속도의 현재값입니다. 정방향으로 이동 시에는 + 부호, 역방향으로 이동 시에는 - 부호가 붙습니다.
_MC_AX[0-255].Act.Vel	LREAL	피드백 현재 속도	피드백 속도의 현재값입니다. 정방향으로 이동 시에는 + 부호, 역방향으로 이동 시에는 - 부호가 붙습니다.

9-5-3 가속과 감속

가속도와 감속도의 단위

가속도와 감속도의 단위는 「지령 단위/s^2」입니다.
지령 단위는 위치의 표시 단위와 전자 기어로 단위 변환된 값입니다.

가속도와 감속도에 관한 축 파라메터

파라메터 명칭	기능	설정 범위	초기값
최대 가속도	축 동작 지령 시 각 축의 최대 가속도를 설정합니다. 「0」을 설정하면 가속도의 제한이 없어집니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
최대 감속도	축 동작 지령 시 각 축의 최대 감속도를 설정합니다. 「0」을 설정하면 감속도의 제한이 없어집니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
가감속 초과	축 가감속 제어에서 목표 위치에 대한 정지를 우선한 결과 감속이 초과되어 최대 가감 속도를 초과한 경우의 동작을 지정합니다. 0: 가감 속도를 빠르게 사용한다(블렌딩을 버퍼드로 변환한다) *1 1: 가감 속도를 빠르게 사용한다 2: 이상으로 인식하여 정지한다 *2	0~2	0
가속도 경고값	각 축의 가속도 경고를 출력하기 위한 최고 가속도에 대한 비율을 설정합니다. 「0」을 설정하면 가속도 경고는 출력하지 않습니다. (단위: %)	0~100	0
감속도 경고값	각 축의 감속도 경고를 출력하기 위한 최고 감속도에 대한 비율을 설정합니다. 「0」을 설정하면 감속도 경고는 출력하지 않습니다. (단위: %)	0~100	0

*1. 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 블렌딩을 버퍼드로 변환하지 않습니다. 상세한 내용은 「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.
*2. 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 블렌딩 동작의 경우 이상으로 정지하지 않습니다. 상세한 내용은 「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.

축 동작의 가속도와 감속도의 지정

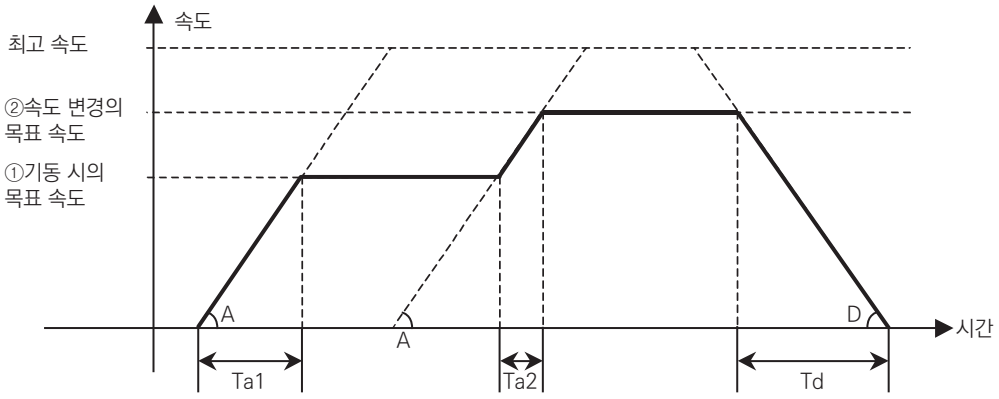
실제 위치 결정 동작의 가속도 및 감속도는 모션 제어 명령의 입력 변수 「Acceleration(가속도)」 및 「Deceleration(감속도)」으로 지정합니다.

가속도와 감속도 모니터

사용자 프로그램에서는 축 변수를 판독하여 가속도 및 감속도를 모니터할 수 있습니다.
또한, 변수명은 _MC_AX[*]를 예로 들어 설명하고 있으나 _MC1_AX[*], _MC2_AX[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_AX[0-255].Cmd.AccDec	LREAL	지령 현재 가감 속도	지령 가감 속도의 현재값입니다. 가속 시에는 + 부호, 감속 시에는 - 부호가 붙습니다.

가감속 동작 예



【기동 시】
Ta1: 실제 가속 시간
A: 가속도

【속도 변경 시】
Ta2: 실제 가속 시간
A: 가속도

【감속 시】
Td: 실제 감속 시간
D: 감속도

지령한 이동량이 적거나 가감 속도가 작은 경우에는 목표 속도에 도달하지 않는 경우가 있습니다.
모션 제어 명령을 재기동하여 변경된 새로운 가속도(감속도)로 동작하면 목표 위치를 넘게 되었을 때 목표 위치에 정지 가능한 가속도 또는 감속도로 위치 결정을 실시합니다.

9-5-4 저크

저크는 가속도/감속도의 변화 비율을 지정합니다. 저크를 지정하면 가감속 시의 속도 파형이 S자형이 되므로 기계의 충격이나 진동을 억제할 수 있습니다.

참고

저크는 「약동」 또는 「약도」, 「가가속도」라고도 합니다.

저크의 단위

저크의 단위는 「지령 단위/s^3」입니다.
지령 단위는 위치의 표시 단위와 전자 기어로 단위 변환된 값입니다.

축 동작의 저크 지정

실제 위치 결정 동작의 저크는 모션 제어 명령의 입력 변수 「Jerk(저크)」로 지정합니다.
가속과 감속에서 공통된 값을 사용합니다.

저크에는 아래의 식으로 계산한 값을 설정합니다.

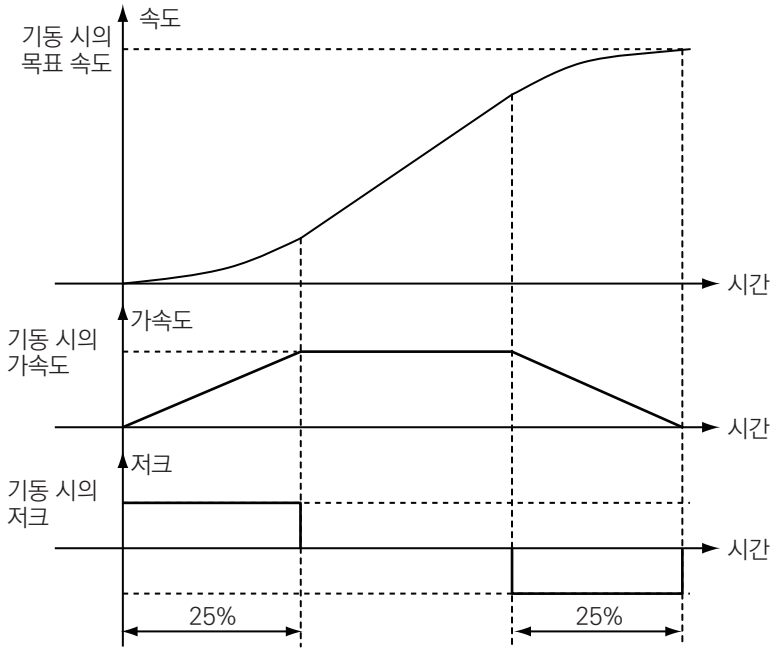
저크 = 가속도/(가속 시간×가속 중인 시간 중 저크를 적용하는 시간의 비율/2)

저크를 적용하는 시간은 가속이 시작되는 시간부터 종료될 때까지의 시간으로 2등분됩니다. 따라서 저크를 적용하는 시간의 비율을 2로 나눕니다.

● 저크를 지정했을 때의 가속 제어의 예

저크가 적용되는 구간의 가속도는 일정한 비율로 변하므로 지령 속도는 S자형으로 완만해집니다.
저크가 「0」인 구간의 가속도는 고정이므로 지령 속도는 직선 모양이 됩니다.

예: 가속도가 25,000mm/s^2이고, 가속 시간이 0.1s, 저크를 적용하는 시간 비율이 50%인 경우
저크 = 25000/(0.1 × 0.5/2) = 1,000,000(mm/s^3)



저크 모니터

사용자 프로그램에서는 축 변수를 판독하여 저크를 모니터할 수 있습니다.
또한, 변수명은 _MC_AX[*]를 예로 들어 설명하고 있으나 _MC1_AX[*], _MC2_AX[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_AX[0-255].Cmd.Jerk	LREAL	지령 현재 저크	지령 저크의 현재값입니다.

9-5-5 동작 방향 지정

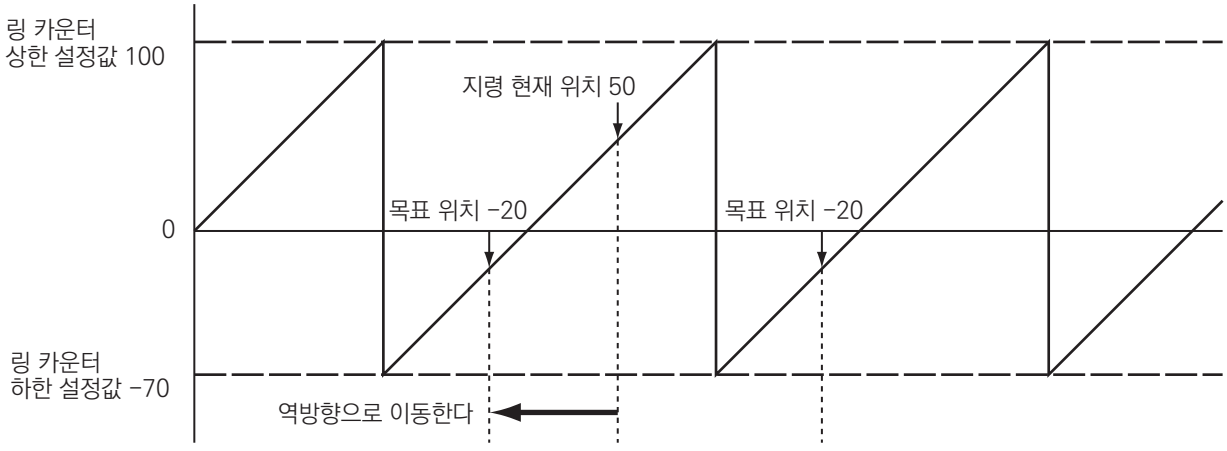
인덱스 테이블 등에서 최단거리 등의 회전 방향을 지정하여 동작시키려면 카운트 모드를 로터리 모드로 설정합니다. 그 후 절대 위치를 지정하는 모션 제어 명령의 입력 변수 「Direction(방향 선택)」으로 동작 방향을 지정합니다.
방향 선택이 「최단거리 지정」, 「정방향 지정」, 「역방향 지정」, 「현재 방향 지정」인 경우 목표 위치는 링 카운터의 1회전 미만이며 링 카운터의 하한값 이상, 상한값 미만인 위치를 지정할 수 있습니다.
카운트 모드가 리니어일 때 방향 선택은 무시되고 목표 위치로 위치 결정을 합니다.

MC 기능 모듈에서 지정할 수 있는 방향 선택은 아래와 같습니다.

방향 선택	동작
최단거리 지정	지령 현재 위치와 목표 위치와의 거리가 짧은 방향을 향해 동작을 시작합니다.
정방향 지정	정방향을 향해 동작을 시작합니다.
역방향 지정	역방향을 향해 동작을 시작합니다.
현재 방향 지정	이전 동작과 같은 방향을 향해 동작을 시작합니다.
방향 지정 없음	링 카운터의 상하한값을 통과하지 않는 방향을 향해 동작을 시작합니다. 이 방향 선택에서는 링 카운터의 상하한값을 초과한 목표 위치를 지정할 수 있습니다. 이때의 동작은 목표 위치와 지령 현재 위치의 차분을 목표 거리로 하는 상대값 위치 결정입니다. 이로 인해 링 카운터의 다회전 위치로 위치 결정을 할 수 있습니다.

최단거리 지정의 예

지령 현재 위치가 「50」일 때 목표 위치 「-20」을 향해 위치 결정하는 예입니다.

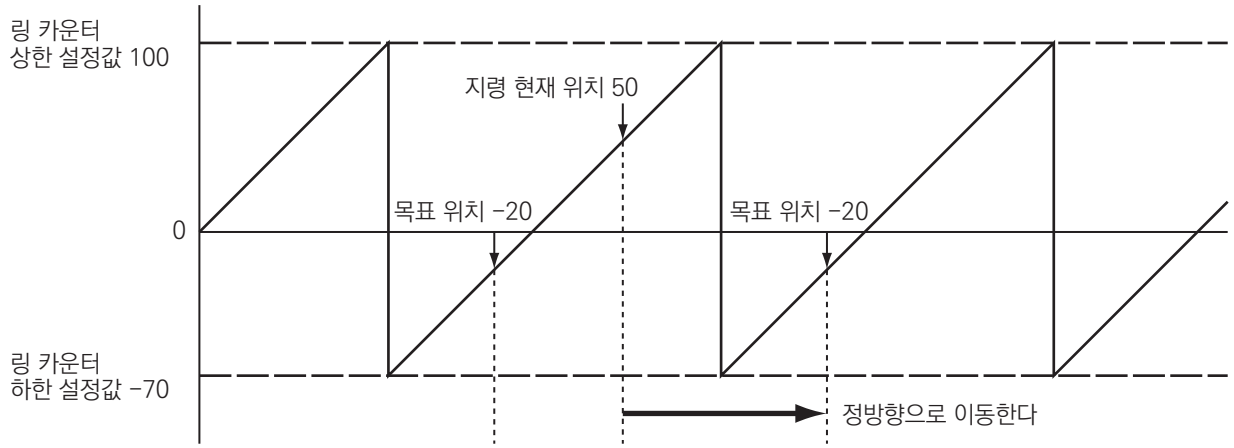


참고

정방향의 이동 거리와 역방향의 이동 거리가 모두 같은 경우에는 「현재 방향 지정」과 동작이 동일합니다.

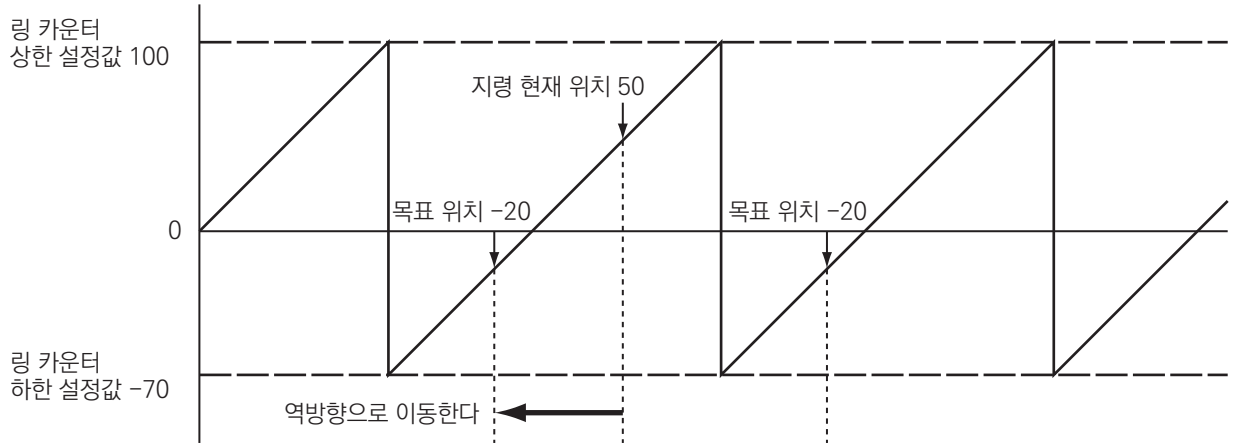
정방향 지정의 예

지령 현재 위치가「50」일 때 목표 위치「-20」을 향해 위치 결정하는 예입니다.



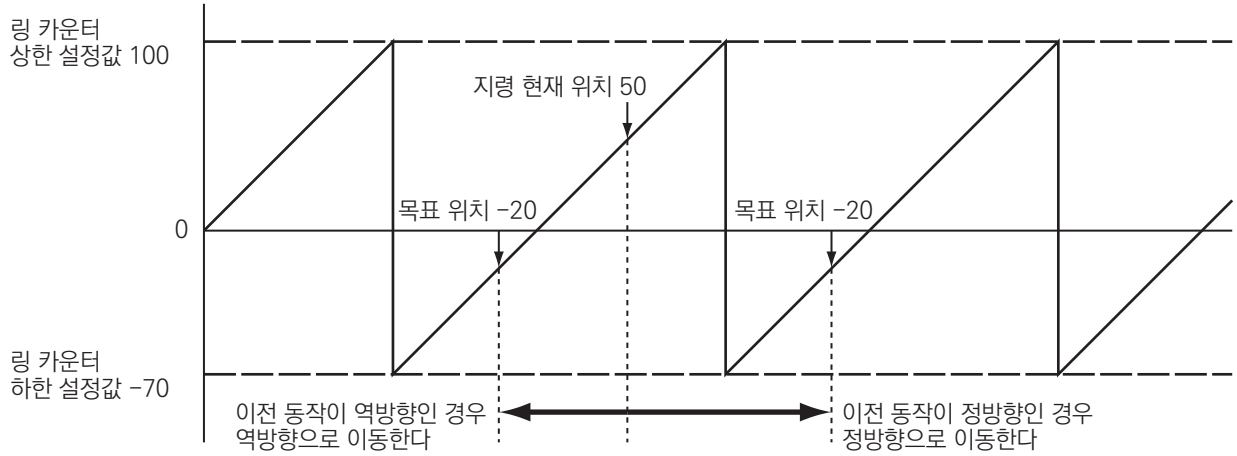
역방향 지정의 예

지령 현재 위치가「50」일 때 목표 위치「-20」을 향해 위치 결정하는 예입니다.



현재 방향 지정의 예

지령 현재 위치가「50」일 때 목표 위치「-20」을 향해 위치 결정하는 예입니다.



이전의 동작 방향은 축 변수의「지령 방향 상태」로 판별할 수 있습니다.

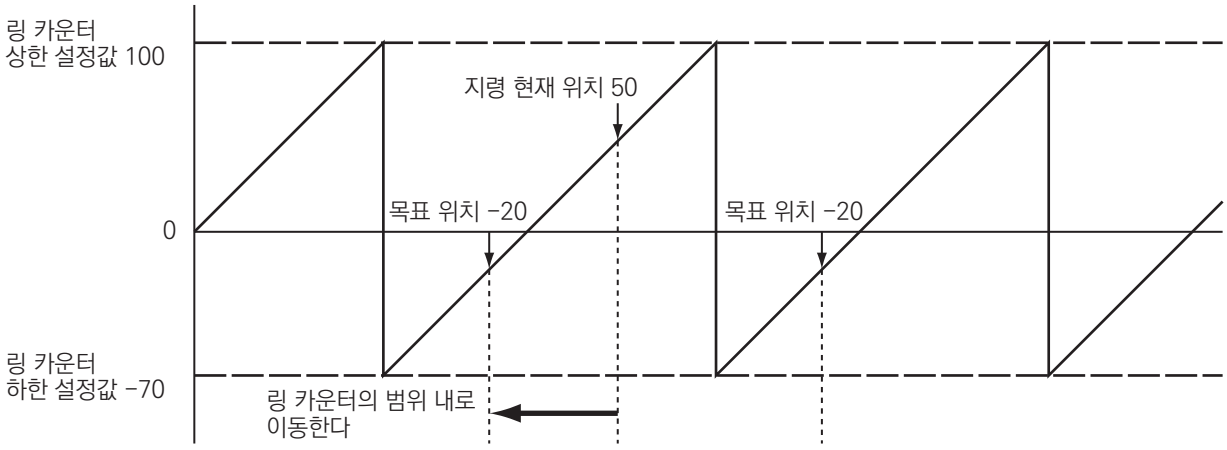
사용상의 주의

이전 동작이 아래와 같은 경우에는 동작 방향에 주의해 주십시오.

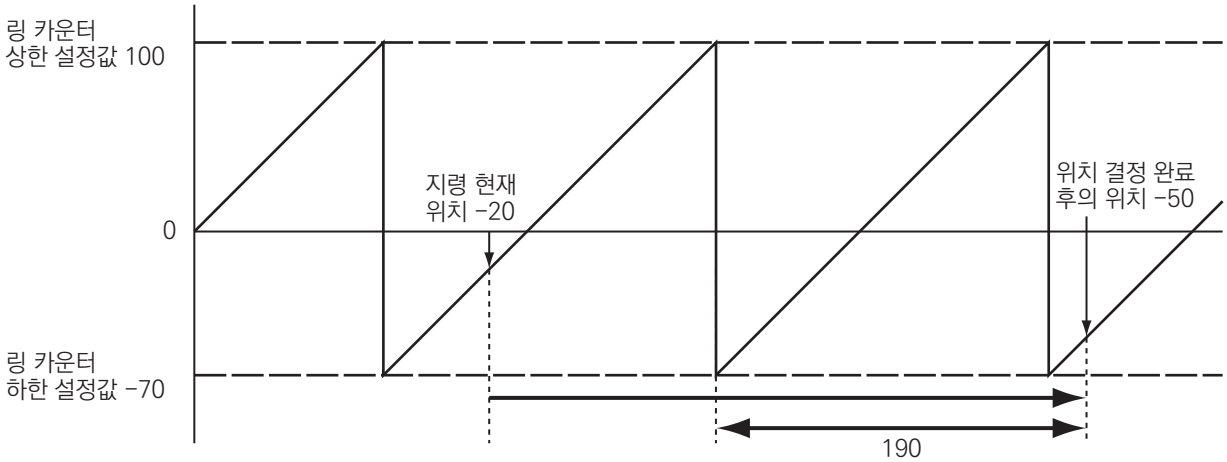
- MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령으로 원점 입력을 검출한 위치를 지나쳐 반전 동작할 경우 원점 검출 방향과 반대 방향이 됩니다.
- MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령으로 원점 복귀 보정값을 설정한 경우 보정값으로 동작하는 방향이 됩니다.
- MC_TouchProbe(외부 래치 유효) 명령으로 즉시 정지를 지정한 경우에 래치 위치를 지나쳐 반전 동작하는 경우.
- MC_MoveFeed(인터럽트 피딩 위치 결정)로 반전 동작하는 경우.
- MC_ResetFollowingError(편차 카운터 리셋) 명령을 실행한 경우에는 편차를 「0」으로 하기 위해 주어진 지령 위치의 방향이 됩니다.
- 외부 입력 신호의 즉시 정지나 한계 입력 정지로 편차 카운터 리셋을 지정한 경우에 외부 입력 신호가 발생한 위치로 반전 동작하는 경우.

방향 지정이 없는 경우의 예

지령 현재 위치가 「50」일 때 목표 위치 「-20」을 향해 위치 결정하는 예입니다.



마찬가지로 링 카운터 상한값이 「100」, 하한값이 「-70」이고 지령 현재 위치가 「-20」일 때 목표 위치 「290」을 향해 위치 결정하는 예입니다.



목표 거리가 (290(목표 위치)-100(상한값))=190으로
상대값 위치 결정을 실행한다

9-5-6 모션 제어 명령의 재기동

단축에서 동작하는 도중에 모션 제어 명령의 동일한 인스턴스의 입력 변수를 변경하고 재기동한 경우에 대해 설명합니다.
재기동으로 변경 가능한 입력 변수에는 Position(목표 위치), Distance(이동 거리), Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도), Torque(목표 토크) 등이 있습니다.
변경할 수 없는 입력 변수를 변경하고 재기동하면 명령에서 이상이 발생합니다.
모션 제어 명령의 다중 기동에 의해 버퍼링되고 있는 명령에 대해 재기동하면 버퍼 중인 명령의 입력 변수가 변경됩니다.

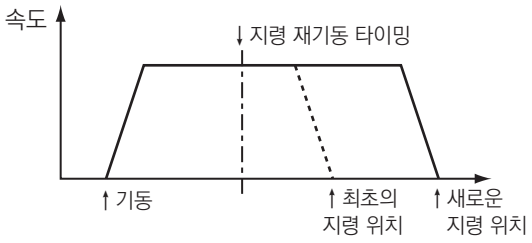
변경 가능한 입력 변수에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 각 명령을 참조해 주십시오.

목표 위치 변경

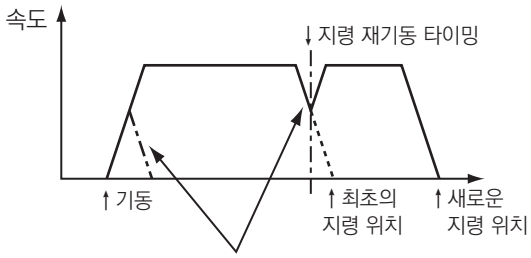
재기동으로 목표 위치를 변경할 경우 변경한 목표 위치나 변경 타이밍에 따라 동작이 다릅니다.
목표 위치를 변경하여 동작 방향이 반전하는 경우의 동작은 축 파라메터 반전 시 동작에서 반전 시 감속 정지 또는 반전 시 즉시 정지를 선택할 수 있습니다.

● 새로운 지령값에 의해 반전되지 않는 경우

등속 동작 중의 재기동



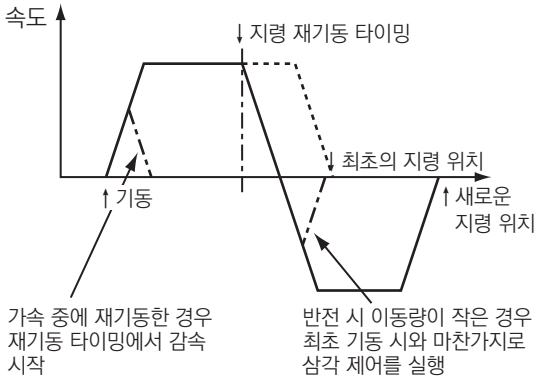
가감속 동작 중의 재기동



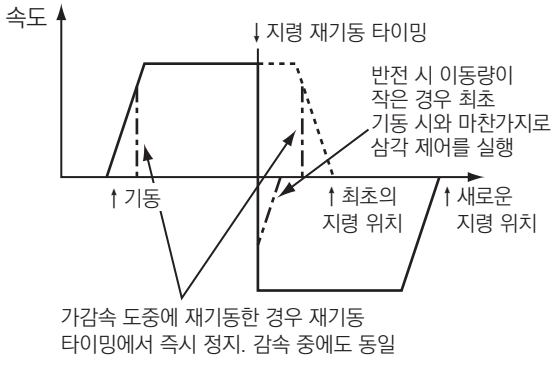
삼각 제어 혹은 감속 중에 재기동한 경우는
다시 목표 속도까지의 가속으로 이행합니다.
이때 목표 속도까지 가속하지 않는 경우도 있습니다.

● 새로운 지령값에 의해 반전되는 경우

반전 시 감속 정지 설정



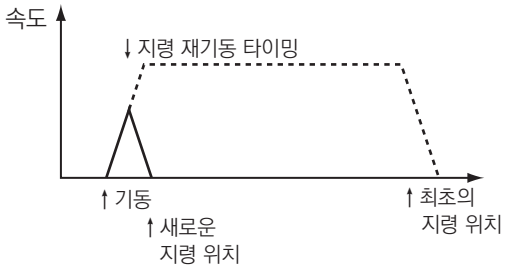
반전 시 즉시 정지 설정



● 삼각 제어 패턴

목표 위치를 변경하여 이동 거리가 짧아지면 아래의 그림과 같이 삼각 제어가 되는 경우가 있습니다.

반전하지 않는 경우

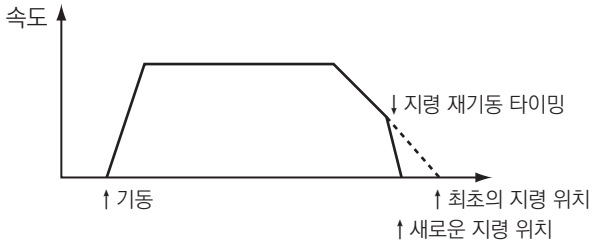


● 감속 초과 패턴

다음의 경우에는 목표 위치로의 정지를 우선하므로 감속도가 지정한 감속도를 초과합니다.
감속도가 축 파라메터「최대 감속도」를 초과할 때는 축 파라메터 설정「가감속 초과」에서 설정한 대로 동작합니다.

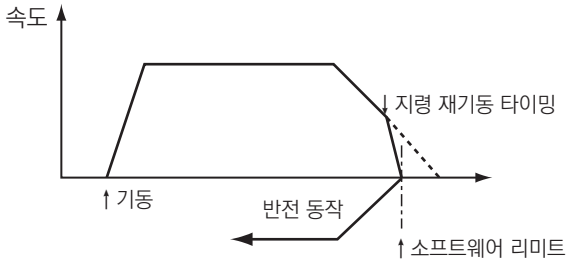
- 동작이 반전하지 않으며, 지정한 감속도로 인해 목표 위치를 지나쳐 버리는 경우에는 아래와 같이 동작합니다.

반전하지 않는 경우



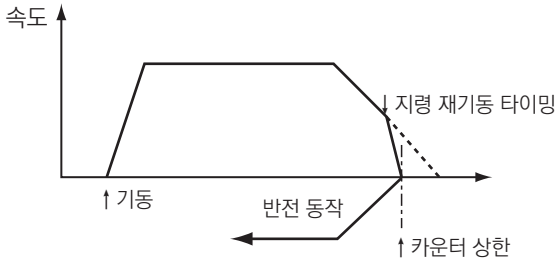
- 동작이 반전하며, 감속 정지로 인해 소프트웨어 리미트를 초과할 때는 아래와 같이 동작합니다.

반전하지 않는 경우



- 동작이 반전하며, 감속 정지로 인해 지령 현재 위치가 오버 플로우 또는 언더 플로우하는 경우에는 아래와 같이 동작합니다.

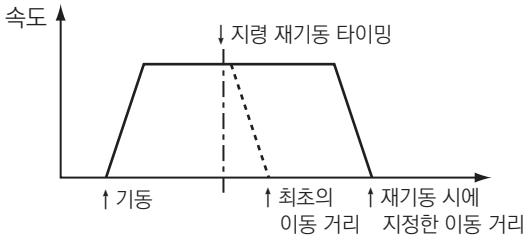
반전하지 않는 경우



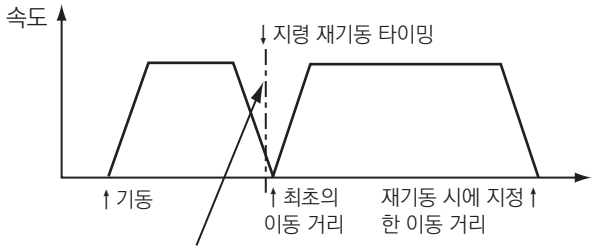
이동 거리 변경

MC_MoveRelative(상대값 위치 결정) 명령으로 이동 거리를 변경하여 재기동한 경우라도 이동을 시작한 위치를 기준으로 하여 변경한 이동 거리로 위치 결정을 합니다.
단, 위치 결정 완료 직전에 재기동하면 새로운 명령의 기동으로 이동하는 경우가 있습니다.

동작 중의 재기동



위치 결정 완료 직전에서의 재기동



위치 결정 완료 직전에서 재기동하면 최초 이동 거리에서 위치 결정한 위치를 기준으로 하여 재기동 시에 지정한 이동 거리를 이동하는 경우가 있습니다.

사용상의 주의

이동 거리를 변경하는 재기동은 위치 결정 완료 직전에 실행하지 마십시오.

목표 속도 변경

목표 속도에 대해서는 가속 중, 등속 중이거나 삼각 제어로 가속 중일 때 동작이 변경됩니다.
감속 중일 때도 변경을 받아들이지만 실제 동작에는 영향을 끼치지 않습니다.

가속도 변경

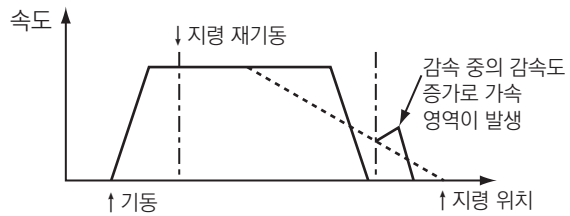
가속도에 대해서는 가속 중이거나 삼각 제어로 가속 중일 때 동작이 변경됩니다.
등속 중일 때 변경하면 오버라이드 등으로 가속합니다.
감속 중일 때도 변경을 받아들이지만 실제 동작에는 영향을 끼치지 않습니다.

감속도 변경

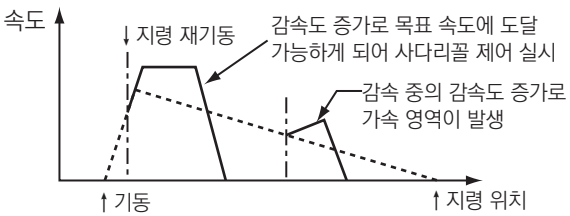
감속도에 대해서는 가속 중, 등속 중, 감속 중, 삼각 제어 중, 감속 초과 제어 중에 변경됩니다
감속도가 변경되어 목표 위치를 지나쳐 버리는 경우 목표 위치로의 정지를 우선하므로 감속도가 지정한 감속도를 초과합니다.

● 감속도 증가 패턴

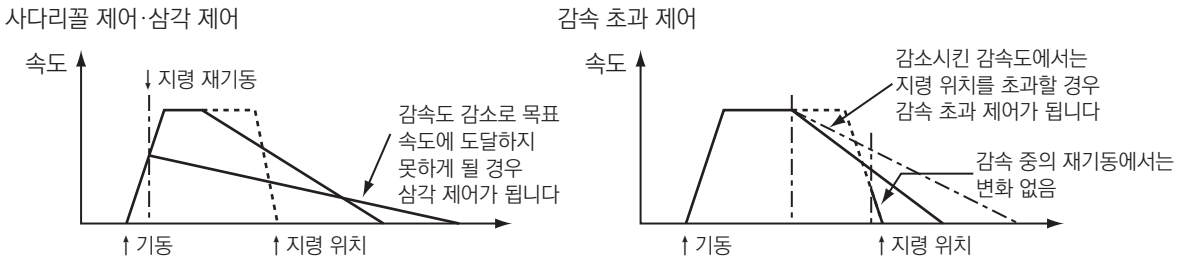
사다리꼴 제어



삼각 제어→사다리꼴 제어



● **감속도 감소 패턴**

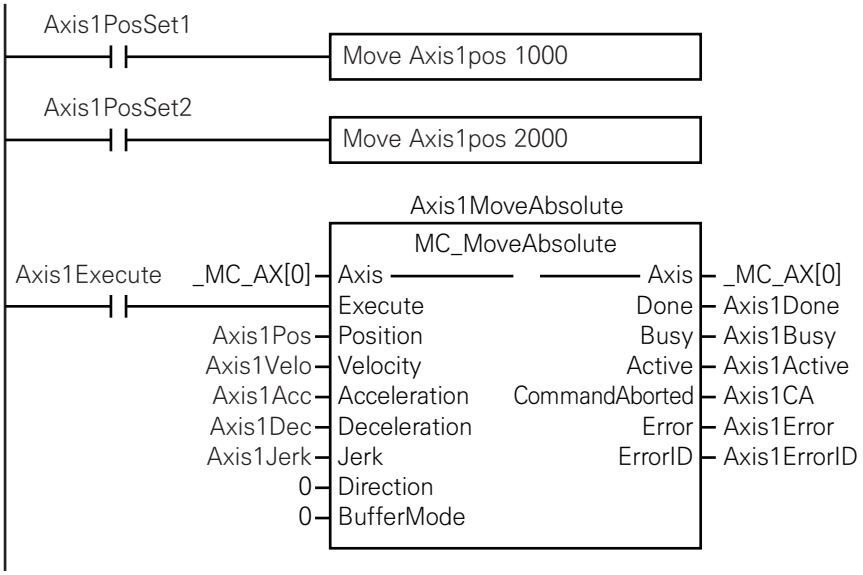


토크 지령 변경

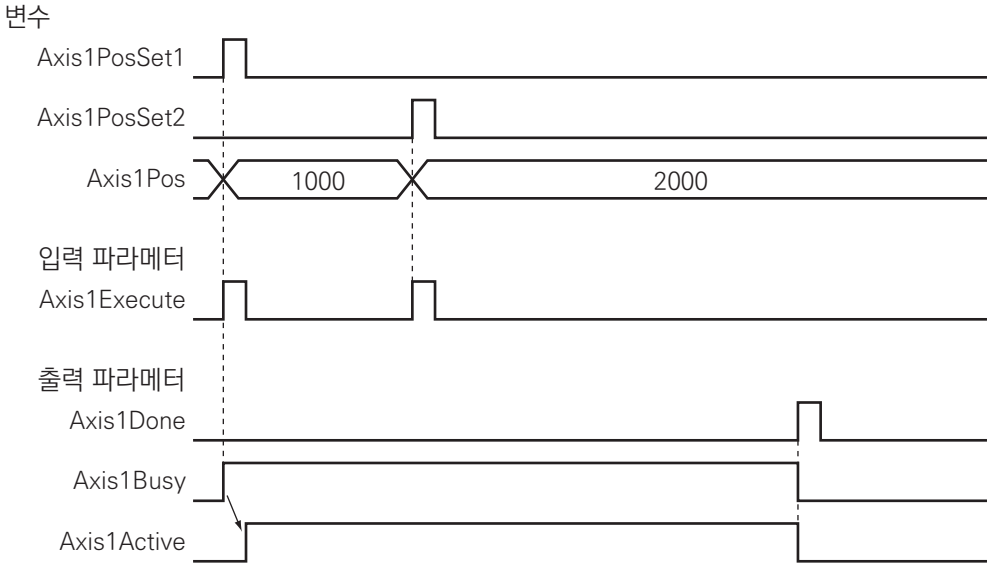
모션 제어 명령을 재기동했을 때 토크 램프의 지정에 따라서 토크 지령값이 변경됩니다.

재기동 프로그램 예

절대값 위치 결정에서 목표 위치를「1000」에서「2000」으로 변경하는 예입니다.
목표 위치에 대한 입력 파라메터에는 변수「Axis1Pos」를 사용합니다.
MOV 명령으로 목표 위치를「1000」으로 설정하고 Axis1Execute를 FALSE→TRUE로 하면 위치 결정을 시작합니다.
동작 중에 목표 위치를「2000」으로 설정하고 Axis1Execute를 다시 FALSE→TRUE로 하면 목표 위치「2000」으로 위치 결정 동작이 변경됩니다.



● **타이밍 차트**



사용상의 주의

변경하지 않는 입력 변수는 재기동하기 전의 값을 유지해 주십시오.

9-5-7 모션 제어 명령의 다중 기동(버퍼 모드)

축 동작 중에 다른 모션 제어 명령을 기동할 수 있습니다.
PLCopen® 기술 사양서에서는 버퍼 모드로 정의하지만 MC 기능 모듈에서는「다중 기동」으로 설명합니다.
다중 기동을 사용하면 복수의 모션 제어 명령 동작을 정지하지 않고 연속적으로 실행할 수 있습니다.

MC 기능 모듈에서는 다중 기동에 대해 다음과 같은 용어를 사용합니다.

용어		의미
본 매뉴얼	PLCopen®	
현 명령	이전 FB	다중 기동을 실행하기 직전에 동작하던 모션 제어 명령
버퍼링 명령	다음 FB	축 동작 중에 기동되어 실행 대기 상태인 모션 제어 명령
중계 속도	블렌딩(혼합)	블렌딩(혼합) 지정에서 현 명령으로 지정된 목표 위치까지 동작할 때 사용하는 지령 속도.

모션 제어 명령의 입력 변수「BufferMode(버퍼 모드 선택)」를 지정하면 아래의 다중 기동 모드를 선택할 수 있습니다.
이러한 모드의 주된 차이는 다중 기동된 명령의 실행 시작 타이밍과 중계 속도입니다.

다중 기동 모드	동작 설명
Aborting(중단)	현 명령이 중단되고 다중 기동한 모션 제어 명령이 실행됩니다.
버퍼드(대기)	현 명령의 동작이 정상 종료한 후 버퍼링 명령이 실행됩니다.
블렌딩(혼합)	현 명령의 목표 위치까지 도달한 후 버퍼링 명령이 실행됩니다. 이 모드에서는 현 명령과 버퍼 명령 사이에서 정지하지 않습니다. 현 명령이 목표 위치에 도달할 때의 중계 속도를 아래에서 선택할 수 있습니다.
블렌딩 로우(저속도)	현 명령과 버퍼링 명령의 목표 속도 중 낮은 쪽을 중계 속도로 합니다.
블렌딩 프리비어스 (이전 속도)	현 명령의 목표 속도를 중계 속도로 합니다.
블렌딩 넥스트 (다음 속도)	버퍼링 명령의 목표 속도를 중계 속도로 합니다.
블렌딩 하이(고속도)	현 명령과 버퍼링 명령의 목표 속도 중 높은 쪽을 중계 속도로 합니다.

다중 기동된 명령은 버퍼드(대기)와 블렌딩(혼합)의 경우 MC 기능 모듈 내에서 버퍼되고 지정한 다중 기동 모드의 타이밍과 중계 속도로 실행됩니다.
이 버퍼는 축마다 1개씩 있습니다.
Aborting(중단)에서는 다중 기동된 명령이 즉시 실행되므로 명령은 버퍼되지 않습니다.

사용상의 주의

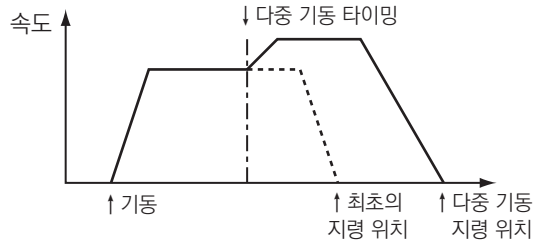
- 1축에서 버퍼할 수 있는 다중 기동은 1개뿐입니다. 2개 이상의 명령을 다중 기동하면 명령에서 이상이 발생합니다.
- 단축 제어로 동작 중인 축에 대해 다축 협조 제어 명령(축 그룹 명령)을 다중 기동할 수 없습니다. 또한, 다축 협조 제어(축 그룹 명령)로 동작 중인 축에 대해 단축 제어 명령(축 명령)을 다중 기동할 수 없습니다. 이러한 다중 기동을 실행한 경우 명령에서 이상이 발생합니다.

Aborting(중단)

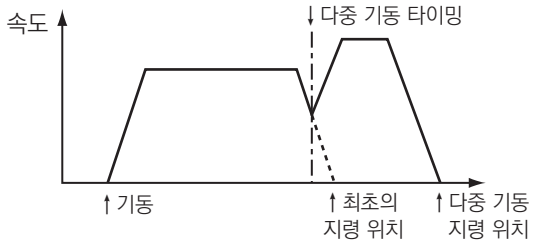
버퍼를 실시하지 않은 디폴트 모드입니다.
실행 중인 지령은 즉시 중단되고 다중 기동된 새로운 명령이 실행됩니다.
Aborting(중단)은 단축 제어와 동기 제어의 모션 제어 명령으로 다중 기동할 수 있습니다.

● 다중 기동 지령 위치에 의해 반전되지 않는 경우

등속 동작 중의 다중 기동



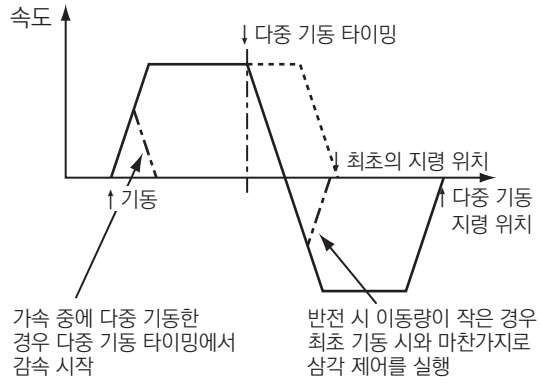
가감속 동작 중의 다중 기동



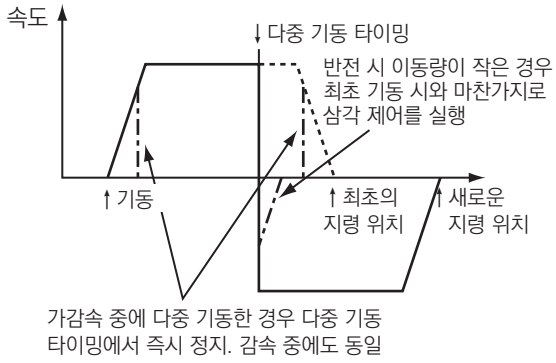
삼각 제어 혹은 감속 중에 다중 기동한 경우는
버퍼링 명령의 목표 속도까지의 가속으로 이행합니다.
이때 목표 속도까지 가속하지 않는 경우도 있습니다.

● 다중 기동 지령 위치에 의해 반전되는 경우

반전 시 감속 정지 설정

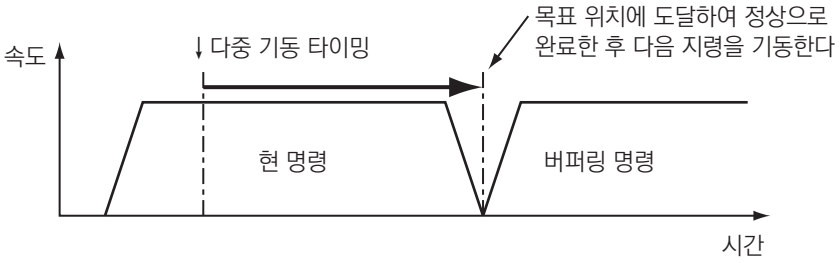


반전 시 즉시 정지 설정



버퍼드(대기)

다중 기동된 명령은 실행 중인 현 명령의 동작이 끝날 때까지 버퍼(대기)됩니다.
현 명령의 동작이 정상 완료한 후 버퍼링 명령이 실행됩니다.



블렌딩(혼합)

다중 기동된 명령은 현 명령의 목표 위치에 도달할 때까지 버퍼(대기)됩니다.
현 명령의 목표 위치에 도달한 후 버퍼 명령이 실행되지만 이때 정지하지 않고 입력 변수 「BufferMode (버퍼 모드 선택)」에서 지정한 속도로 2개의 동작을 연속합니다.
상대값 이동의 경우 최종적인 도달 위치는 두 명령의 합계값이 됩니다.
절대값 이동에서는 2번째 명령의 목표 위치입니다.
축 파라메터 「최대 가속도」 혹은 「최대 감속도」가 설정되어 목표 위치를 지나쳐 버리는 경우의 동작은 축 파라메터의 「가감속 초과」로 다음 중 하나를 설정할 수 있습니다.

- 가감 속도를 빠르게 사용한다(블렌딩을 버퍼드로 변환한다)
- 가감 속도를 빠르게 사용한다
- 이상으로 인식하여 정지한다

사용상의 주의

블렌딩(혼합)은 단축 제어와 동기 제어의 조합으로는 실행할 수 없습니다.

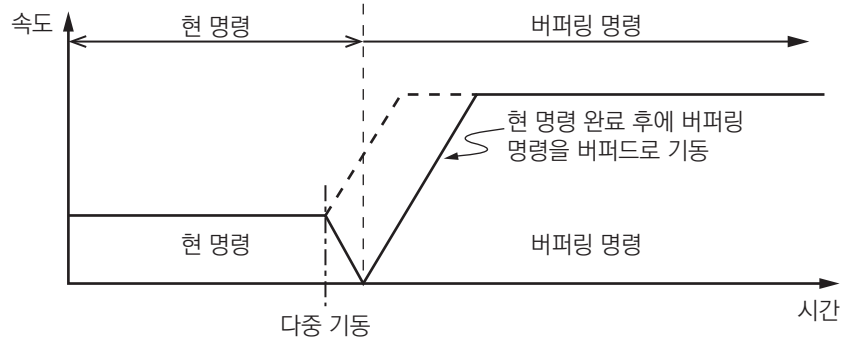
버전 관련 정보

유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 [Use rapid acceleration/deceleration. (Blending is changed to Buffered.)]를 설정해도 블렌딩을 버퍼드로 변환하지 않습니다.
이때 최대 가감 속도로 가감속하고 블렌딩을 계속합니다.
또한, [Minor fault stop]을 설정해도 이상으로 정지하지 않습니다. 이 경우에도 최대 가감 속도로 가감속하고 블렌딩을 계속합니다.

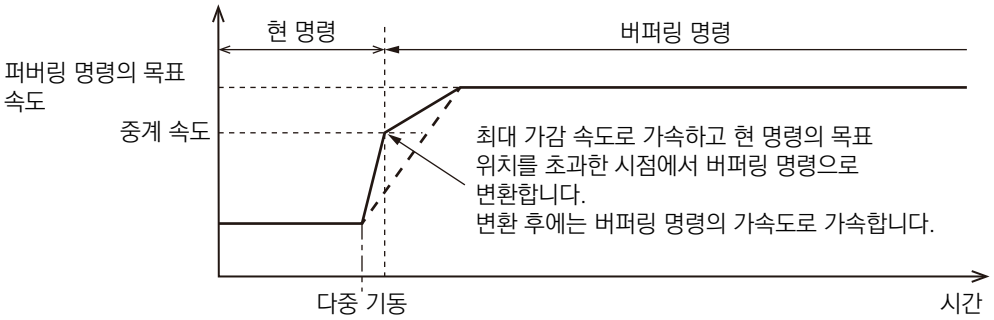
「가감속 초과」의 동작 예는 다음과 같습니다.

● 가감 속도를 빠르게 사용한다(블렌딩을 버퍼드로 변환한다)

- 유니트 버전 Ver.1.09 이하인 CPU 유니트는 다음과 같이 동작합니다.

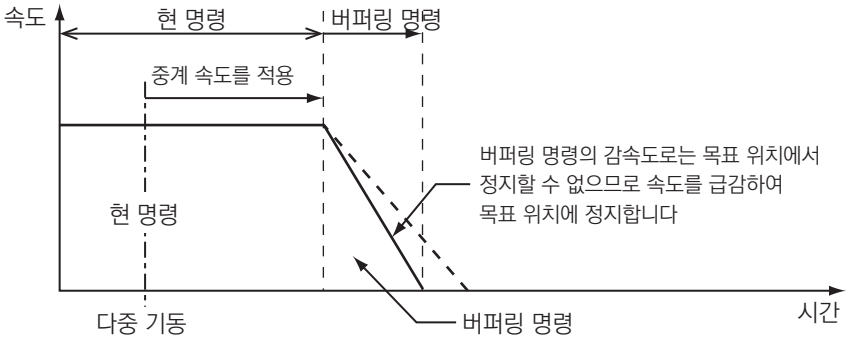


・ 유니트 버전 Ver.1.10 이상의 CPU 유니트는 다음과 같이 동작합니다.
또한, [Minor fault stop]으로 설정해도 동일하게 동작합니다.
이 예에서는 BufferMode를 블렌딩 넥스트로 설정합니다.

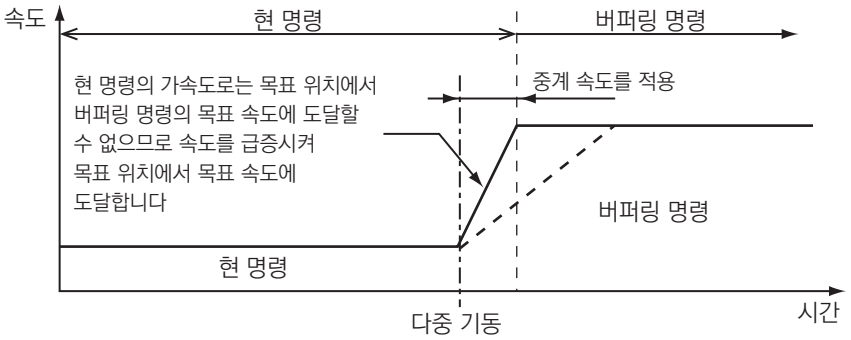


● **가감 속도를 빠르게 사용한다**

・「BufferMode(버퍼드 선택)」을 블렌딩 넥스트로 설정한 경우에는 다음과 같이 동작합니다.



・「BufferMode(버퍼드 선택)」을 블렌딩 넥스트로 설정한 경우에는 다음과 같이 동작합니다.



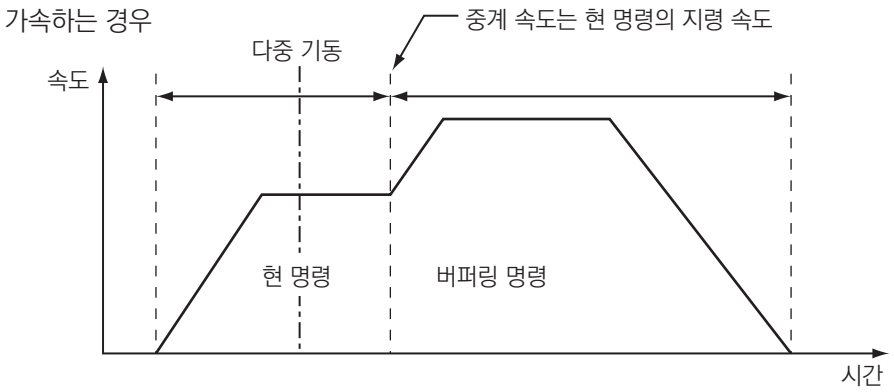
블렌딩 로우(저속도)

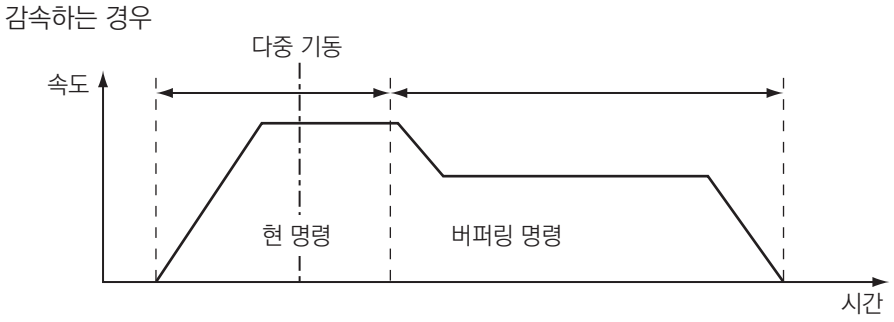
현 명령의 목표 위치에서 현 명령과 버퍼링 명령 중 낮은 쪽의 목표 속도가 되도록 동작합니다.

블렌딩 프리비어스(이전 속도)

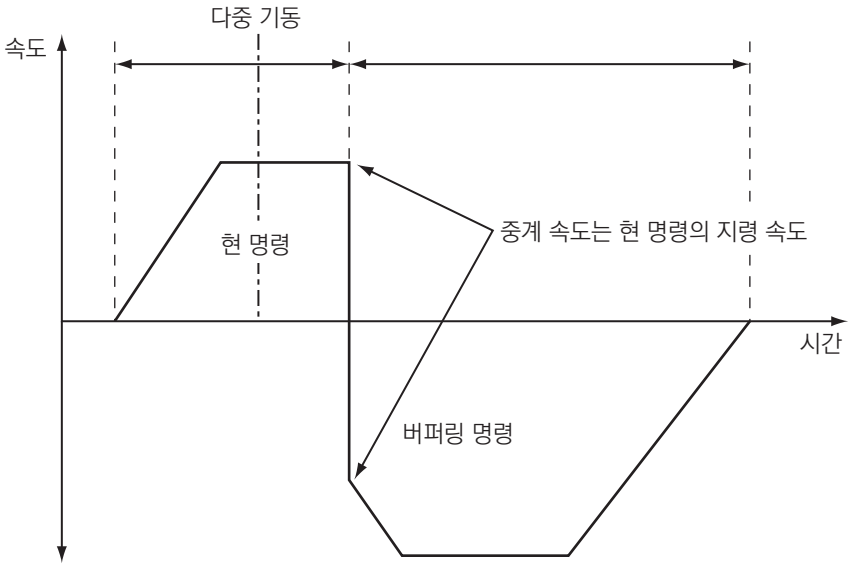
현 명령의 목표 위치까지는 현 명령의 목표 속도로 동작합니다.
목표 위치에 도달한 후에는 버퍼링 명령의 목표 속도로 가감속하여 동작합니다.

● 동작 방향이 반전하지 않는 경우



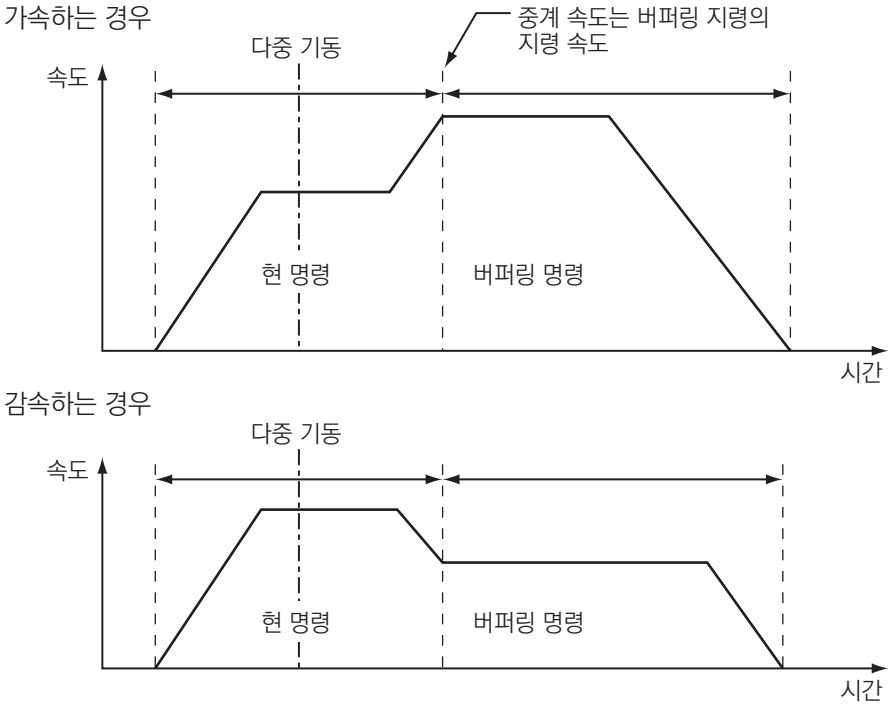


● 동작 방향이 반전하는 경우



블렌딩 넥스트(다음 속도)

현 명령의 목표 위치에서 버퍼링 명령의 목표 속도가 되도록 동작합니다.



블렌딩 하이(고속도)

현 명령의 목표 위치에서 현 명령과 버퍼링 명령 중 높은 쪽의 목표 속도가 되도록 동작합니다.

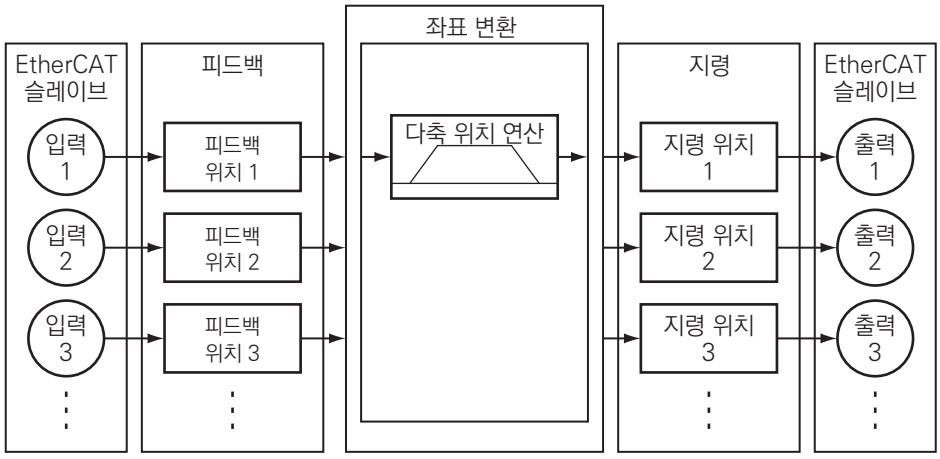
9-6 다축 협조 제어

여기에서는 다축의 협조 제어 동작에 대해 설명합니다.
MC 기능 모듈에서는 사전에 Sysmac Studio에서 축 그룹을 설정해 두면 다축에서 보간 제어를 할 수 있습니다.

9-6-1 동작 개요

다축 협조 제어란 공구 끝부분과 같은 제어 대상의 궤적을 제어하기 위해 관련된 여러 축을 그룹화하여 협조 동작시키는 기능입니다.
MC 기능 모듈에서는 협조 동작을 실행할 축의 조합을 Sysmac Studio를 이용하여 축 그룹으로 설정합니다.
사용자 프로그램에서는 각 축을 서보 ON한 후 실제로 다축 협조 제어를 실행할 축 그룹을 유효화합니다.
다축 협조 제어는 축 그룹에 속하는 축의 협조 동작이 목적이므로 유효한 축 그룹의 축에 대해 단축 동작의 모션 제어 명령을 기동할 수 없습니다. 또한, 축 그룹에 속하는 축에서 이상이 발생한 경우에는 축 그룹에 속하는 다른 축은 축 그룹 파라메터 [Axis Group Stop Method] 설정에 따라 정지합니다.

MC 기능 모듈에서는 2축~4축까지의 직선 보간과 2축의 원호 보간을 실행할 수 있습니다.



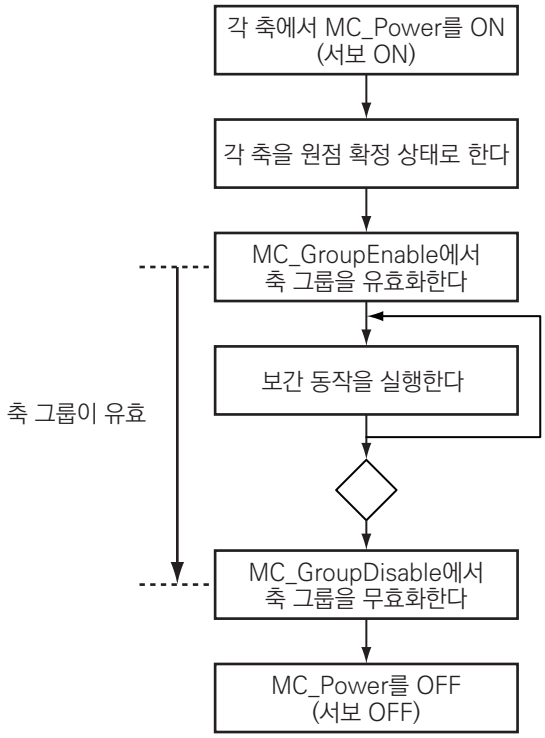
참고

동작 중에 축의 조합을 변경하여 보간을 제어해야 하는 장비는 조합을 변경할 축을 포함한 여러 개의 축 그룹을 미리 Sysmac Studio에서 작성해 둡니다. 그 후, 동작 도중에 사용자 프로그램에서 유효화할 축 그룹을 지정하면 실현할 수 있습니다.
유니트 버전 Ver.1.01 이상인 CPU 유니트와 Sysmac Studio Ver.1.02 이상을 조합하면 MC_ChangeAxesInGroup(축 그룹 구성 축 기록)을 사용하여 무효한 축 그룹의 구성 축을 갱신할 수 있습니다.

축 그룹에 대한 상세한 내용은 「3-3 축 그룹 (P.3-19)」을 참조해 주십시오.

축 그룹 유효/무효

축 그룹을 유효화하려면 MC_GroupEnable(축 그룹 유효) 명령으로 유효화할 축 그룹을 지정합니다. 무효한 상태에서 축 그룹 명령을 기동하면 명령에서 이상이 발생하여 기동할 수 없습니다.
유효한 상태인 축 그룹을 무효하게 하려면 MC_GroupDisable(축 그룹 무효) 명령으로 무효화할 축 그룹을 지정합니다.
축 그룹 동작 중에 축 그룹 무효 명령을 기동하면 축 그룹에 속하는 축은 축 파라메터 설정의 최대 감속도로 감속 정지합니다.



축 그룹 유효/무효 기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_GroupEnable(축 그룹 유효) 명령 및 MC_GroupDisable(축 그룹 무효) 명령을 참조해 주십시오.

축 그룹 구성 축 기록

MC_ChangeAxesInGroup(축 그룹 구성 축 기록) 명령을 이용하여 무효화한 축 그룹의 구성 축을 일시적으로 갱신하는 기능입니다. 축 그룹이 유효한 경우에는 MC_GroupDisable(축 그룹 무효) 명령으로 무효화한 후 구성 축을 갱신해 주십시오.
유니트 버전 Ver.1.01 이상인 CPU 유니트와 Sysmac Studio Ver.1.02 이상의 조합으로 사용할 수 있습니다.

사용상의 주의

MC_ChangeAxesInGroup(축 그룹 구성 축 기록)에 의한 변경은 CPU 유니트에 내장된 불휘발성 메모리에 저장되지 않습니다. 전원을 재투입하거나 Sysmac Studio에서 다운로드하여 불휘발성 메모리의 파라메터 설정 내용으로 되돌아옵니다.

축 그룹 구성 축 기록에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_ChangeAxesInGroup(축 그룹 구성 축 기록) 명령을 참조해 주십시오.

축 그룹 위치 취득

MC_GroupReadPosition(축 그룹 위치 취득) 명령을 이용하여 축 그룹의 지령 현재 위치와 피드백 현재 위치를 취득하는 기능입니다.
유니트 버전 Ver.1.01 이상인 CPU 유니트와 Sysmac Studio Ver.1.02 이상의 조합으로 사용할 수 있습니다.

축 그룹 위치 취득 기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_GroupReadPosition(축 그룹 위치 취득) 명령을 참조해 주십시오.

축 그룹 에러 리셋

축 그룹에서 이상이 발생한 경우 이상의 원인을 제거한 후 MC_GroupReset(축 그룹 에러 리셋) 명령으로 이상을 해제합니다.

축 그룹 에러 리셋에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_GroupReset(축 그룹 에러 리셋) 명령을 참조해 주십시오.

NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

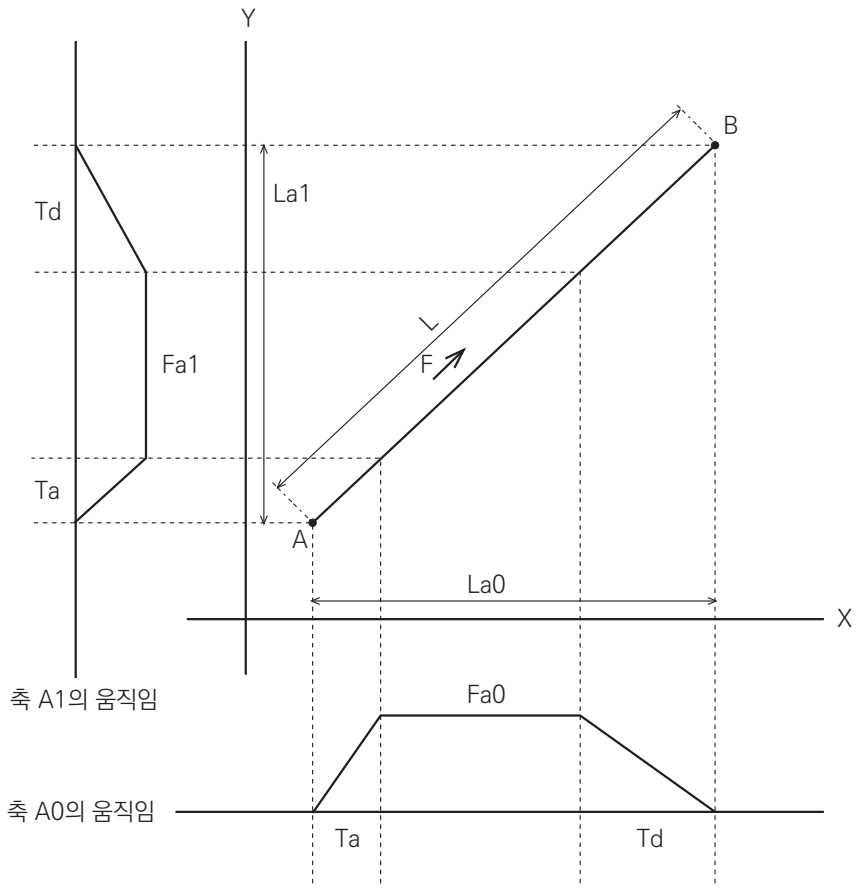
9-6-2 직선 보간

직선 보간은 축 그룹의 축 A0~축 A3인 논리 축 중에서 2축~4축을 사용하여 시작점과 종료점을 직선으로 위치 결정합니다.
절대값 위치 결정과 상대값 위치 결정이 가능하며 보간 속도, 보간 가속도, 보간 감속도 및 저크를 지정할 수 있습니다.

MC 기능 모듈에서는 다음과 같이 3종류의 직선 보간 명령이 있습니다.

- MC_MoveLinear(직선 보간)
  입력 변수「MoveMode(이동 방법 선택)」를 지정하여 절대값 직선 보간과 상대값 직선 보간을 선택할 수 있습니다. MC 기능 모듈의 독자적인 명령입니다.
- MC_MoveLinearAbsolute(절대값 직선 보간)
  절대값 직선 보간을 실행합니다. PLCopen® 기술 사양서에서 정의된 명령입니다
- MC_MoveLinearRelative(상대값 직선 보간)
  상대값 직선 보간을 실행합니다. PLCopen® 기술 사양서에서 정의된 명령입니다

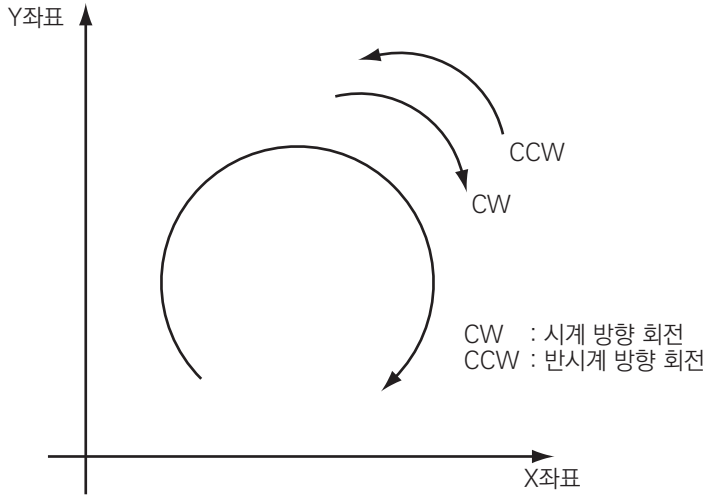
2축을 사용한 A점부터 B점까지의 직선 보간은 다음 그림과 같습니다.



직선 보간에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_MoveLinear(직선 보간) 명령, MC_MoveLinearAbsolute(절대값 직선 보간) 명령 및 MC_MoveLinearRelative(상대값 직선 보간) 명령을 참조해 주십시오.

9-6-3 원호 보간

원호 보간은 축 그룹의 축 A0~축 A3인 논리 축 중에서 2축을 사용하여 2차원 평면상에서 원호를 그려 위치 결정합니다.
절대값 위치 결정과 상대값 위치 결정이 가능하며 원호 보간 모드, 경로의 방향, 보간 속도, 보간 가속도, 보간 감속도 및 2축의 합성값인 저크를 지정할 수 있습니다.



MC 기능 모듈에서는 입력 변수「CircMode(원호 보간 모드)」에 의해 원호 보간 방식을 아래의 3종류 중에서 지정할 수 있습니다.

- 통과점 지정
- 중심점 지정
- 반경 지정

사용상의 주의

원호 보간에서 사용하는 축의 카운트 모드는「Linear Mode」로 해 주십시오.
「Rotary Mode」로 기동하면 명령에서 이상이 발생합니다.

9-6-4 축 그룹 사이클릭 동기 위치 제어

이 기능은 축 그룹의 각 축에 지정한 목표 위치를 주기적으로 출력합니다.
임의의 궤적으로 동작시키고 싶은 경우 등에 사용자 프로그램에서 연산한 목표 위치를 절대 위치로 지정합니다.
유니트 버전 Ver.1.01 이상인 CPU 유니트와 Sysmac Studio Ver.1.02 이상의 조합으로 사용할 수 있습니다.

축 그룹 사이클릭 동기 위치 제어 기능에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션 편(SBCE-364)』의 MC_GroupSyncMoveAbsolute(축 그룹 사이클릭 동기 위치 제어) 명령을 참조해 주십시오.

9-6-5 다축 협조 제어에서의 정지

축 그룹의 동작을 정지하는 기능에는 사용자 프로그램의 모션 제어 명령에 의한 정지 기능과 이상 발생 등에 의한 정지가 있습니다.

모션 제어 명령에 의한 정지

축 그룹 동작의 동작 정지에는 MC_GroupStop(축 그룹 강제 정지) 명령 또는 MC_GroupImmediateStop (축 그룹 즉시 정지) 명령을 사용합니다.

● MC_GroupStop(축 그룹 강제 정지) 명령

축 그룹에서의 직선 보간 및 원호 보간에 대해 감속도와 저크를 지정하여 궤적상에서 감속 정지할 수 있습니다.
감속도에 「0」을 지정하면 즉시 정지하는 지령을 서보 드라이버 등에 송신합니다.
본 명령으로 감속 정지 중과 입력 변수 「Execute(기동)」가 TRUE인 동안에는 다른 동작 지령을 받지 않습니다.

● MC_GroupImmediateStop(축 그룹 즉시 정지) 명령

축 그룹의 모든 축을 즉시 정지시킬 수 있습니다. 즉시 정지 방법은 각 축의 축 파라메터 「즉시 정지 입력 정지 방법」에 따릅니다.
MC_GroupStop(축 그룹 강제 정지) 명령에 의해 감속 정지 중인 축 그룹에 대해서도 실행 가능합니다.

기능에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_GroupStop (축 그룹 강제 정지) 명령, MC_GroupImmediateStop(축 그룹 즉시 정지) 명령을 참조해 주십시오.

이상 발생 등에 의한 정지

● 축 그룹 동작 중 이상 발생에 의한 정지

축 그룹 동작 중에 축 그룹 중 하나의 구성 축에서 감속 정지하는 이상이 발생한 경우 보간의 궤적상에서 보간 감속 정지합니다. 이때 보간 감속도는 제어 중인 명령의 감속도 지정에 따릅니다.
또한, 어느 한 구성 축에서 즉시 정지되는 이상이 발생한 경우 축 그룹의 다른 구성 축은 축 그룹 파라메터 설정의 「축 그룹 정지 방법」 설정에 따라 정지합니다.

축 그룹의 정지 방법은 다음 중에서 선택할 수 있습니다.
- 즉시 정지
- 각 축을 감속 정지(각 축의 최대 감속도)
- 즉시 정지하고 동시에 서보 OFF를 실행

● 모션 제어 주기 초과 발생에 의한 정지

모션 제어의 처리가 2주기 이내에 완료하지 않고 제어 주기 초과가 발생하면 즉시 정지합니다.

사용상의 주의

NX701 CPU 유니트를 사용하여 멀티 모션으로 동작 중인 경우에 어느 한쪽의 태스크에서 모션 제어 주기 초과가 발생했을 때는 양쪽 태스크의 모든 축이 즉시 정지합니다.
멀티 모션은 「A-6-2 모션 제어 기능 (P.A-32)」을 참조해 주십시오.

● MC 시운전 기능의 시작에 의한 정지

Sysmac Studio의 MC 시운전 기능을 시작했을 때 각 축은「최대 감속도」로 감속 정지합니다.

● CPU 유니트의 모드 변환에 의한 정지

CPU 유니트의 모드를 변환하면 각 축은「최대 감속도」로 감속 정지합니다.

참고

- 축 그룹에서 동작 중에 MC_GroupDisable(축 그룹 무효) 명령을 기동하면 각 축은 최대 감속도로 감속 정지합니다.
- 축 그룹 동작 중에 MC_Stop(강제 정지) 명령을 기동하면 축과 축 그룹에 이상이 발생하여 축 그룹 동작이 보간 감속 정지합니다. 이때 보간 감속도는 제어 중인 명령의 감속도 지정에 따릅니다.
- 축 그룹 동작 중에 MC_Power(운전 가능) 명령의 입력 변수「Enable(유효)」을 FALSE로 하면 MC 기능 모듈은 해당 축의 지령값을 즉시 정지하고 서보 OFF를 실행합니다. 서보가 OFF일 때 서보 드라이버 등의 동작은 서보 드라이버 등의 설정에 따릅니다.

  축 그룹의 다른 축은 축 그룹 파라메터 설정의「Axes Group Stop Method」설정에 따라 정지합니다. 축 그룹에 이상이 발생합니다.
- 운전 모드에서 프로그램 모드로 변환했을 때는 동작 중인 모션 제어 명령은 중단됩니다. 이때 명령의 출력 변수「CommandAborted(실행 중단)」는 FALSE인 상태입니다. 프로그램 모드로 변환한 후에도 서보 ON/OFF 상태를 유지합니다.
- 운전 모드에서 프로그램 모드로 변환하여 감속 정지하고 있을 때 운전 모드로 되돌리면 모션 제어 명령의 출력 변수「CommandAborted(실행 중단)」가 TRUE가 됩니다.
- MC_SaveCamTable(캠 테이블 저장) 명령으로 저장 중인 경우 저장 처리는 계속됩니다.
- MC_GenerateCamTable(캠 테이블 생성) 명령으로 캠 테이블 생성을 실행 중인 경우 생성 처리는 계속됩니다.

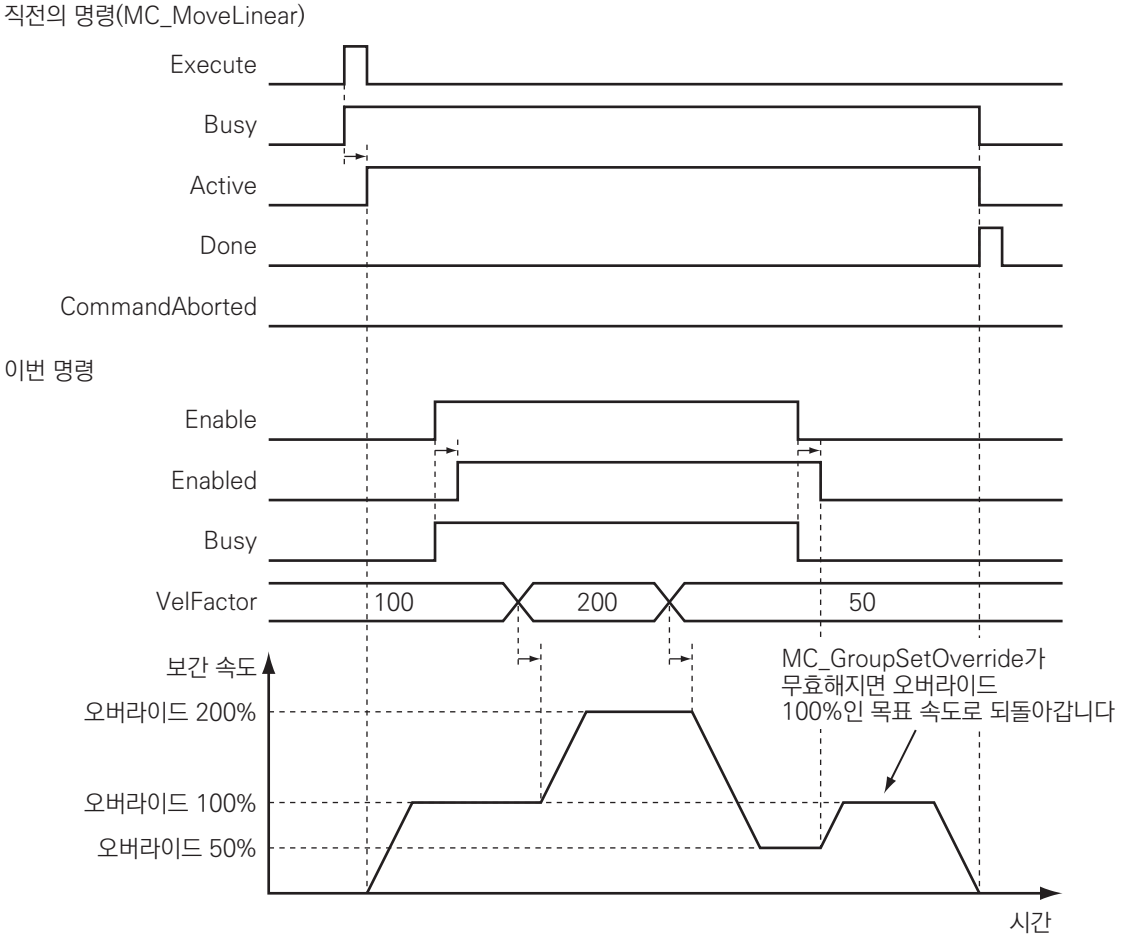
9-6-6 다축 협조 제어에서의 오버라이드

축 그룹에서 다축 협조 제어하는 경우에는 MC_GroupSetOverride(축 그룹 오버라이드값 설정) 명령을 이용하여 기동 중인 보간 동작에 대한 오버라이드값을 설정할 수 있습니다.
오버라이드값은 보간의 목표 속도에 대한 비율로 0~500%를 설정할 수 있습니다.
보간의 목표 속도에 대해 0%로 설정하면 속도 「0」으로 정지한 채 동작 상태를 계속합니다.

오버라이드가 유효인 동안 설정한 오버라이드값이 입력됩니다. 오버라이드를 무효화하면 오버라이드값은 100%로 되돌아갑니다.
오버라이드값으로 변경한 후의 값이 최고 보간 속도를 초과하면 축 그룹 파라메터의 「최고 보간 속도」가 됩니다.

● MC_MoveLinear(직선 보간) 명령에 대한 오버라이드

MC_MoveLinear(직선 보간) 명령에 오버라이드 명령을 사용한 경우의 타이밍 차트 예는 다음과 같습니다.



기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_GroupSetOverride(축 그룹 오버라이드값 설정) 명령을 참조해 주십시오.

9-7 다축 협조 제어의 공통 기능

여기에서는 다축 협조 제어의 공통 기능에 대해 설명합니다.

9-7-1 다축 협조 제어에서의 속도

다축 협조 제어에서는 궤적상에서의 보간 속도로 속도를 지정합니다.
단위는 단축의 경우와 같이 「지령 단위/s」를 사용합니다.

속도의 종류

MC 기능 모듈에서 취급하는 축 그룹의 보간 속도는 아래의 1종류입니다.

속도 종별	정의
지령 보간 속도	MC 기능 모듈이 축 그룹을 제어하기 위해 출력 중인 지령 보간 속도의 현재값입니다.

속도에 관한 축 파라메터

파라메터 명칭	기능	설정 범위	초기값
최고 보간 속도	궤적의 최고 보간 속도를 설정합니다. 「0」을 설정하면 보간 속도의 제한이 없어집니다. 축 그룹 동작 명령으로 최고 보간 속도를 초과하는 보간 목표 속도가 지정된 경우에는 최고 보간 속도로 동작합니다.	배정밀도 실수형의 양수, 0	800,000,000
보간 속도 경고값	보간 속도 경고를 출력하기 위한 최고 보간 속도에 대한 비율을 설정합니다. 「0」을 설정하면 보간 속도 경고는 출력하지 않습니다. (단위: %)	0~100	0

축 동작의 목표 속도 지정

실제 위치 결정 동작의 보간 속도는 모션 제어 명령의 입력 변수 「Velocity(목표 속도)」로 지정합니다.

속도 모니터

사용자 프로그램에서 축 그룹 변수를 판독하여 보간 속도를 모니터할 수 있습니다.
또한, 변수명은 _MC_GRP[*]를 예로 들어 설명하고 있으나 _MC1_GRP[*], _MC2_GRP[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_GRP[0-63].Cmd.Vel	LREAL	지령 보간 속도	지령 보간 속도의 현재값입니다. 정방향으로 이동 시에는 + 부호, 역방향으로 이동 시에는 - 부호가 붙습니다.

9-7-2 다축 협조 제어에서의 가속과 감속

다축 협조 제어에서는 궤적상에서의 보간 가속도와 보간 감속도로 제어합니다.
단위는 단축의 경우와 같이 「지령 단위/s^2」를 사용합니다.

보간 가속도와 보간 감속도에 관한 축 파라메터

파라메터 명칭	기능	설정 범위	초기값
최대 보간 가속도	궤적의 최대 보간 가속도를 설정합니다. 「0」을 설정하면 보간 가속도의 제한이 없어집니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
최대 보간 감속도	궤적의 최대 보간 감속도를 설정합니다. 「0」을 설정하면 보간 감속도의 제한이 없어집니다. (단위: 지령 단위/s^2)	배정밀도 실수형의 양수, 0	0
보간 가감속 초과	축 그룹의 가감속 제어에서 목표 위치에 대한 정지를 우선한 결과 감속이 초과되어 최대 보간 가감 속도를 초과한 경우의 동작을 지정합니다. 0: 가감 속도를 빠르게 사용한다(블렌딩을 버퍼드로 변환한다) *1 1: 가감 속도를 빠르게 사용한다 2: 이상으로 인식하여 정지한다 *2	0~2	0
보간 가속도 경고값	보간 가속도 경고를 출력하기 위한 최고 보간 가속도에 대한 비율을 설정합니다. 「0」을 설정하면 보간 가속도 경고는 출력하지 않습니다. (단위: %)	0~100	0
보간 감속도 경고값	보간 감속도 경고를 출력하기 위한 최고 보간 감속도에 대한 비율을 설정합니다. 「0」을 설정하면 보간 감속도 경고는 출력하지 않습니다. (단위: %)	0~100	0

*1. 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 블렌딩을 버퍼드로 변환하지 않습니다. 상세한 내용은 「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.
*2. 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 블렌딩 동작의 경우 이상으로 정지하지 않습니다. 상세한 내용은 「9-5-7 모션 제어 명령의 다중 기동(버퍼 모드) (P.9-45)」을 참조해 주십시오.

축 그룹 동작의 보간 가속도와 보간 감속도 지정

실제 위치 결정 동작의 보간 가속도 및 보간 감속도는 모션 제어 명령의 입력 변수 「Acceleration(가속도)」 및 「Deceleration(감속도)」으로 지정합니다.

보간 가속도와 보간 감속도의 모니터

사용자 프로그램에서 축 그룹 변수를 판독하여 보간 가속도 및 보간 감속도를 모니터할 수 있습니다.
또한, 변수명은 _MC_GRP[*]를 예로 들어 설명하고 있으나 _MC1_GRP[*], _MC2_GRP[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_GRP[0-63].Cmd.AccDec	LREAL	지령 보간 가감속도	지령 보간 가감 속도의 현재값입니다. 가속 시에는 + 부호, 감속 시에는 - 부호가 붙습니다.

9-7-3 다축 협조 제어에서의 저크

다축 협조 제어에서의 Jerk(저크)는 보간 궤적상에서의 보간 가감속을 S자형으로 완만하게 하여 기계의 충격 및 진동을 억제하는 기능입니다.

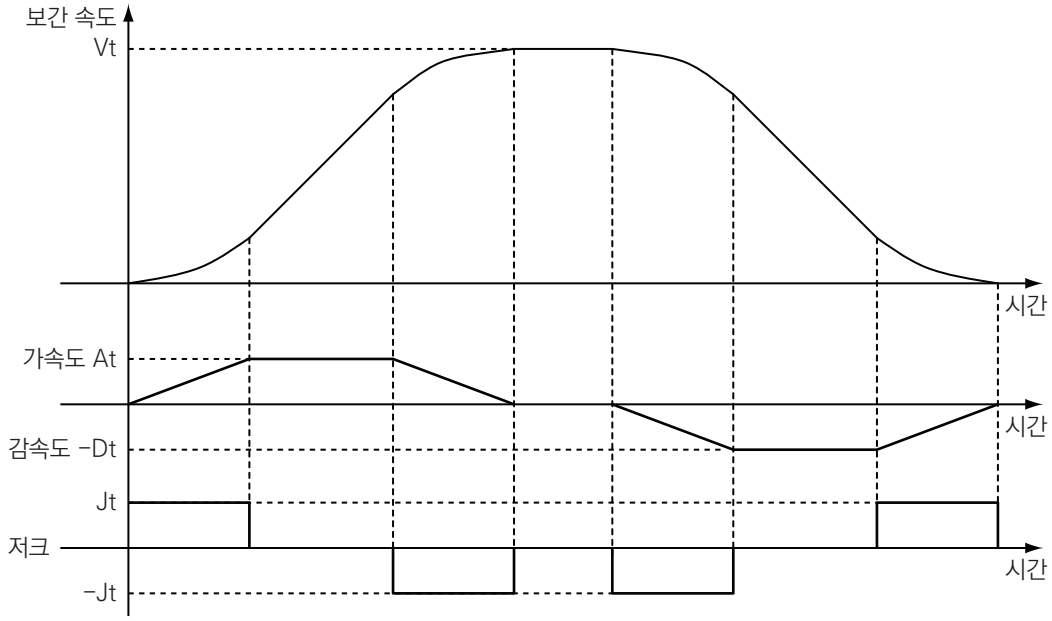
단위는 단축의 경우와 같이 「지령 단위/s^3」를 사용합니다.

축 그룹 동작의 저크 지정

실제 보간 동작의 저크는 모션 제어 명령의 입력 변수 「Jerk(저크)」로 지정합니다.

Jerk(저크)가 「0」 이외인 경우의 예

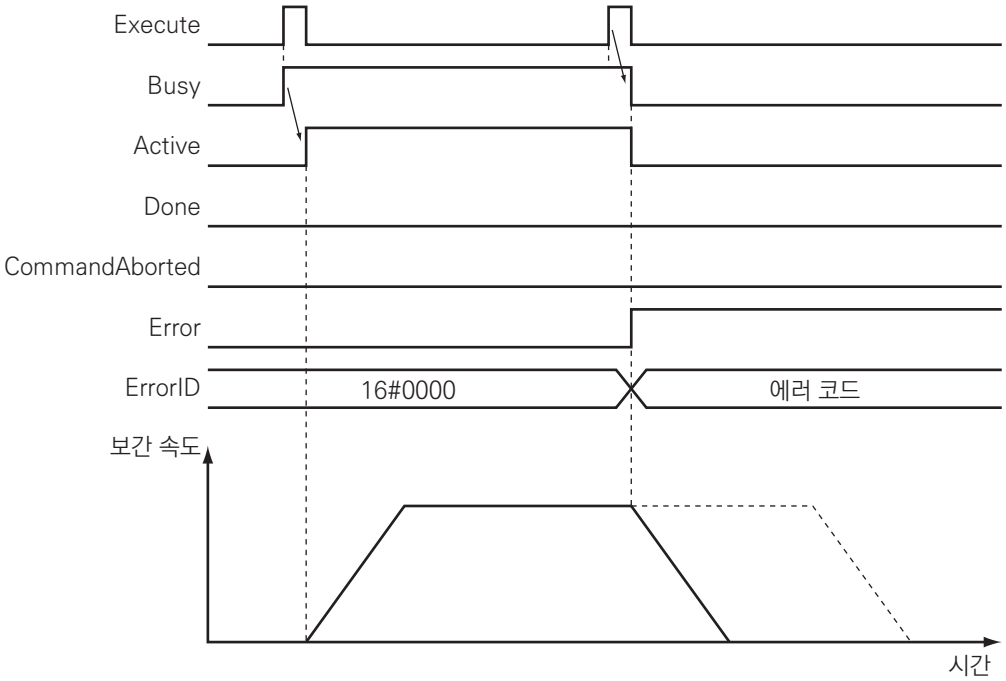
저크가 적용되는 구간의 가감 속도는 일정한 비율로 변하므로 지령 보간 속도는 S자형으로 완만해집니다.
저크가 「0」인 구간의 보간 가속도는 고정이므로 지령 보간 속도는 직선 모양이 됩니다.



Vt: 보간 속도 지정값, At: 가속도 지정값, Dt: 감속도 지정값, Jt: 저크 지정값

9-7-4 다축 협조 제어의 모션 제어 명령 재기동

직선 보간이나 원호 보간 명령을 재기동한 경우 명령에서 이상이 발생합니다.



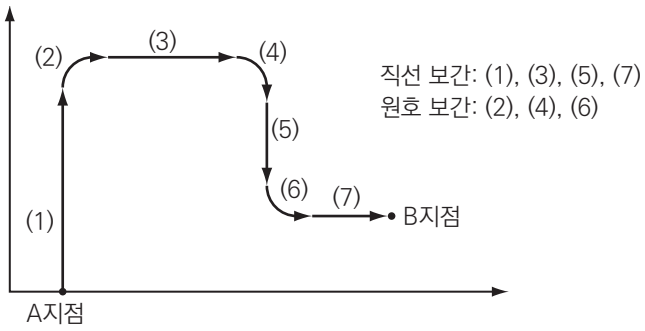
MC_GroupStop(축 그룹 강제 정지) 명령을 재기동할 때는 감속도를 변경할 수 있지만 저크는 변경할 수 없습니다.

MC_GroupReset(축 그룹 에러 리셋) 명령을 재기동한 경우 재기동 지령은 무시되고 이상 해제 처리를 계속합니다.

각 모션 제어 명령의 재기동에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편 (SBCE-364)』의 각 명령을 참조해 주십시오.

9-7-5 다축 협조 제어의 모션 제어 명령 다중 기동(버퍼 모드)

축 동작과 마찬가지로 축 그룹에서의 다축 협조 제어를 다중 기동할 수 있습니다.
다축 협조 제어에서 다중 기동을 사용하면 여러 개의 직선 및 원호가 연속하는 궤적 제어를 실행할 수 있습니다.



모션 제어 명령의 입력 변수「BufferMode(버퍼 모드 선택)」를 지정하면 단축일 때와 동일한 다중 기동 모드를 선택할 수 있습니다.
축 그룹의 명령 버퍼는 축 그룹별로 동작 중인 것이 1개, 다중 기동용이 7개로 총 8개입니다.
축 동작의 명령과 축 그룹 동작의 명령은 서로 다중 기동할 수 없습니다.

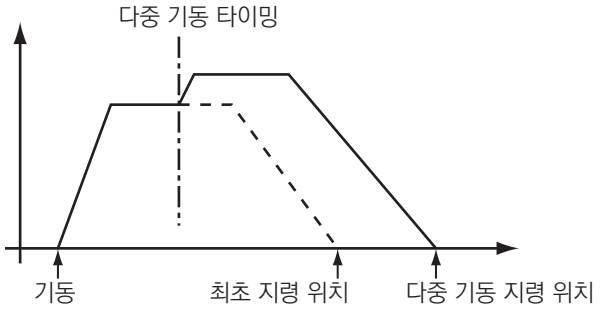
사용상의 주의

- 축 그룹 1개에서 버퍼할 수 있는 다중 기동은 최대 7개입니다. 8개 이상의 명령을 다중 기동하면 명령에서 이상이 발생합니다.
- 단축 제어로 동작 중인 축에 대해 다축 협조 제어 명령(축 그룹 명령)을 다중 기동할 수 없습니다. 또한, 다축 협조 제어(축 그룹 명령)로 동작 중인 축에 대해 단축 제어 명령(축 명령)을 다중 기동할 수 없습니다. 이러한 다중 기동을 실행한 경우 명령에서 이상이 발생합니다.

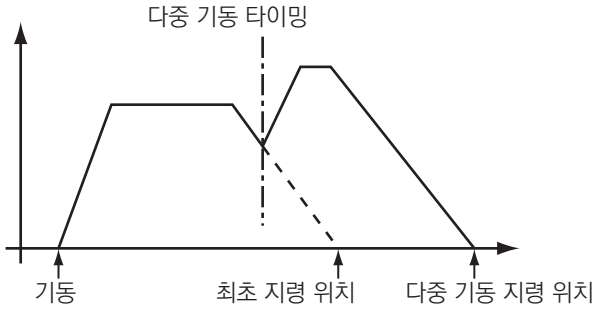
Aborting(중단)

버퍼를 실시하지 않은 디폴트 모드입니다.
실행 중인 명령은 즉시 중단되고 다중 기동된 새로운 명령이 실행됩니다.
입력 변수「BufferMode(버퍼 모드 선택)」가 없는 모션 제어 명령의 다중 기동은 Aborting(중단)으로 동작합니다.
다중 기동된 시점의 보간 속도를 기점으로 하여 다중 기동된 명령의 동작이 시작됩니다.
Aborting(중단)은 동기 제어를 포함하는 단축 제어와 축 그룹 제어의 조합으로는 실행할 수 없습니다.
단축 동작 중인 축에 축 그룹 동작을 기동하면 다중 기동 시에 명령에서 이상이 발생합니다.
축 그룹 동작 중에 단축 동작을 기동하면 축과 축 그룹에 모두 이상이 발생하여 정지합니다.

등속 동작 중의 다중 기동

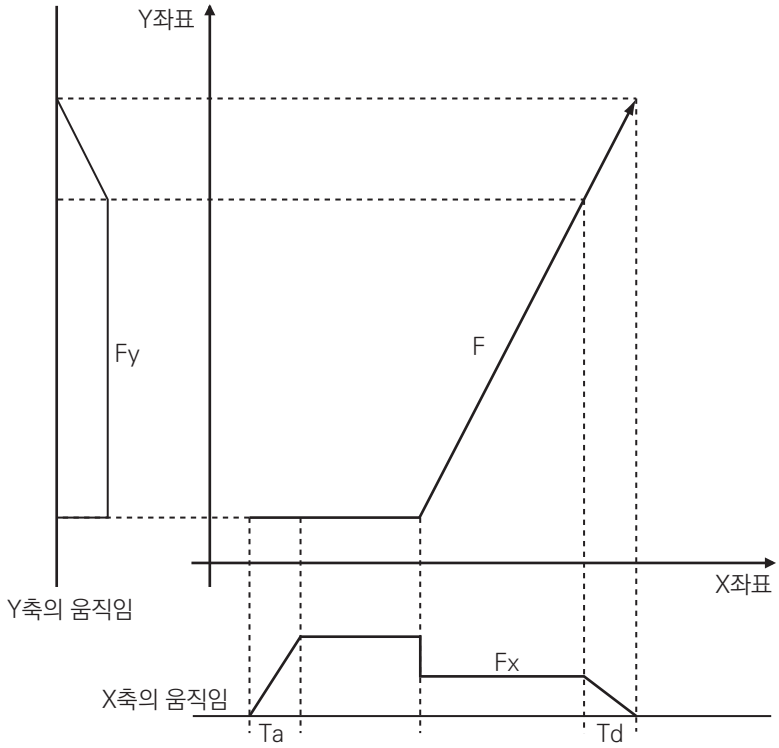


가감속 동작 중의 다중 기동



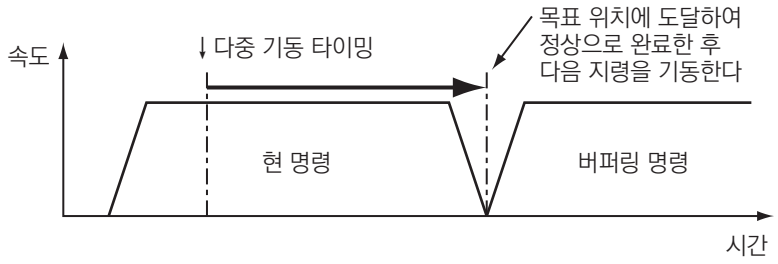
축 그룹에서의 다중 기동은 보간 속도가 연속되도록 실행됩니다. 이동량이「0」인 축을 포함하는명령과 연속할 경우 각 축의 속도 변화는 연속되지 않습니다.

● **2축 직교 좌표에서의 보간 속도와 각 축의 속도 예**



버퍼드(대기)

다중 기동된 명령은 실행 중인 동작이 끝날 때까지 버퍼(대기)됩니다.
현 명령의 동작이 정상 완료한 후 버퍼링 명령이 실행됩니다.



블렌딩(혼합)

축 그룹에서의 블렌딩(혼합)도 단축 동작의 블렌딩(혼합)과 같은 동작이 됩니다.
다중 기동된 명령은 현 명령의 목표 위치에 도달할 때까지 버퍼(대기)됩니다.
현 명령의 목표 위치에 도달한 후 버퍼링 명령이 실행됩니다. 이때 도달 위치에서 정지하지 않고 입력 변수「BufferMode(버퍼 모드 선택)」에서 지정한 보간 속도로 2가지 동작이 연속됩니다.
버퍼 명령이 가속도 혹은 감속도에서 목표 위치를 지나쳐 버리는 경우의 동작은 축 그룹 파라메터 [Interpolation Acceleration/Deceleration Over]의 아래 항목 중에서 선택할 수 있습니다.

- 가감 속도를 빠르게 사용한다(블렌딩을 버퍼드로 변환한다)
- 가감 속도를 빠르게 사용한다
- 이상으로 인식하여 정지한다

버전 관련 정보

- 유니트 버전 Ver.1.10 이상인 CPU 유니트에서는 [Use rapid acceleration/deceleration. (Blending is changed to Buffered.)]를 설정해도 블렌딩을 버퍼드로 변환하지 않습니다. 이때 최대 가감 속도로 가감속하고 블렌딩을 계속합니다.
  또한, [Minor fault stop]을 설정해도 이상으로 정지하지 않습니다. 이 경우에도 최대 가감 속도로 가감속하고 블렌딩을 계속합니다.
- 유니트 버전 Ver.1.09 이하인 CPU 유니트를 사용할 때는 다음 사항에 주의해 주십시오.
  다축 협조 제어에서의 블렌딩(혼합)으로 프로파일 연산 결과, 현 명령의 실행 시간이 4제어 주기 미만일 때는 버퍼드(대기) 동작이 됩니다. 이때 감시 정보인「블렌딩이 중계 속도에 도달할 때까지의 이동량 부족 통지」가 발생합니다.
  유니트 버전 Ver.1.10 이상인 CPU 유니트를 사용할 때는 위와 같은 제약은 없습니다.

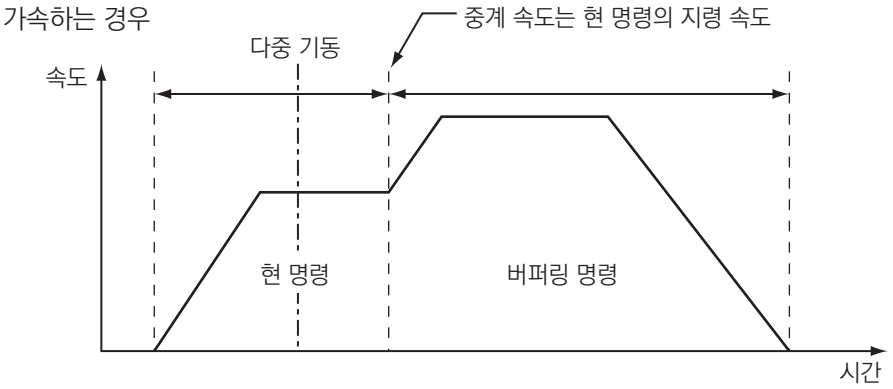
● 블렌딩 로우(저속도)

현 명령의 목표 위치에서 현 명령과 버퍼링 명령 중 낮은 쪽의 목표 속도가 되도록 동작합니다.

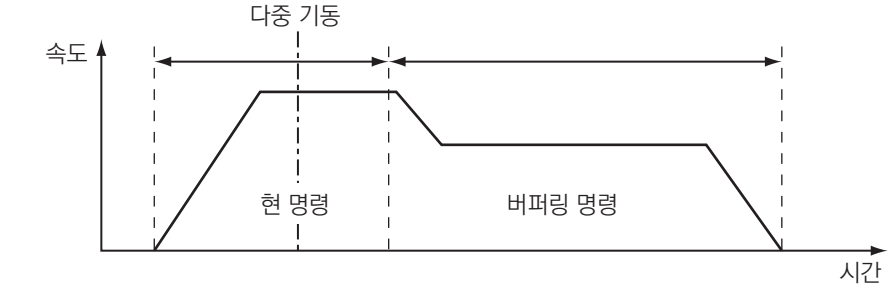
● 블렌딩 프리비어스(이전 속도)

현 명령의 목표 위치까지는 현 명령의 목표 속도로 동작합니다. 목표 위치에 도달한 후에는 버퍼링 명령의 목표 속도로 가감속하여 동작합니다.

가속하는 경우



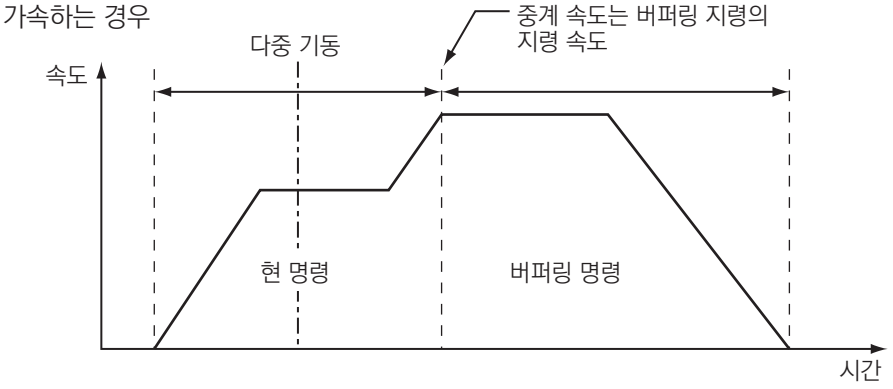
감속하는 경우



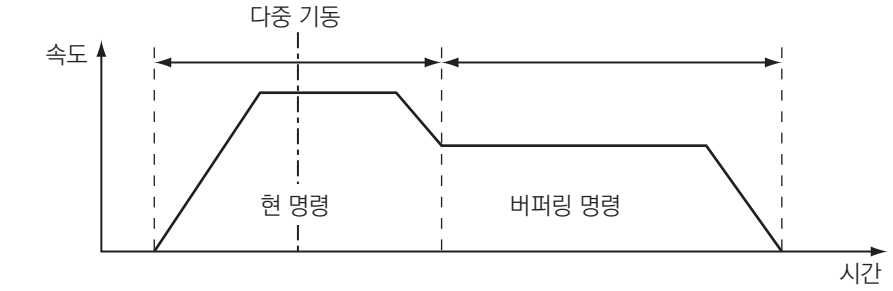
● 블렌딩 넥스트(다음 속도)

현 명령의 목표 위치에서 버퍼링 명령의 목표 속도가 되도록 동작합니다.

가속하는 경우



감속하는 경우



● 블렌딩 하이(고속도)

현 명령의 목표 위치에서 현 명령과 버퍼링 명령 중 높은 쪽의 목표 속도가 되도록 동작합니다.

트랜지션 모드

축 그룹의 다중 기동에서는 보간 궤적의 방향이 바뀌면 장비나 워크에 충격이 발생하는 경우가 있습니다. 이 충격을 완화하기 위한 방법으로 명령 간 연결 동작 방법을 모션 제어 명령의 입력 변수 「TransitionMode(트랜지션 모드)」로 지정할 수 있습니다.
MC 기능 모듈에서 선택할 수 있는 트랜지션 모드는 다음과 같습니다.

No.	트랜지션 모드	설명
0	트랜지션 무효 (_mcTMNone)	트랜지션을 위한 처리를 실행하지 않습니다(디폴트). 충격은 완화되지 않지만 동작 시간은 짧아집니다.
10	코너 슈퍼임포즈 (_mcTMCornerSuperimposed)	현 명령의 감속과 버퍼 명령의 가속을 중첩합니다. 보간 궤적의 선 속도를 일정하게 할 수 있습니다.

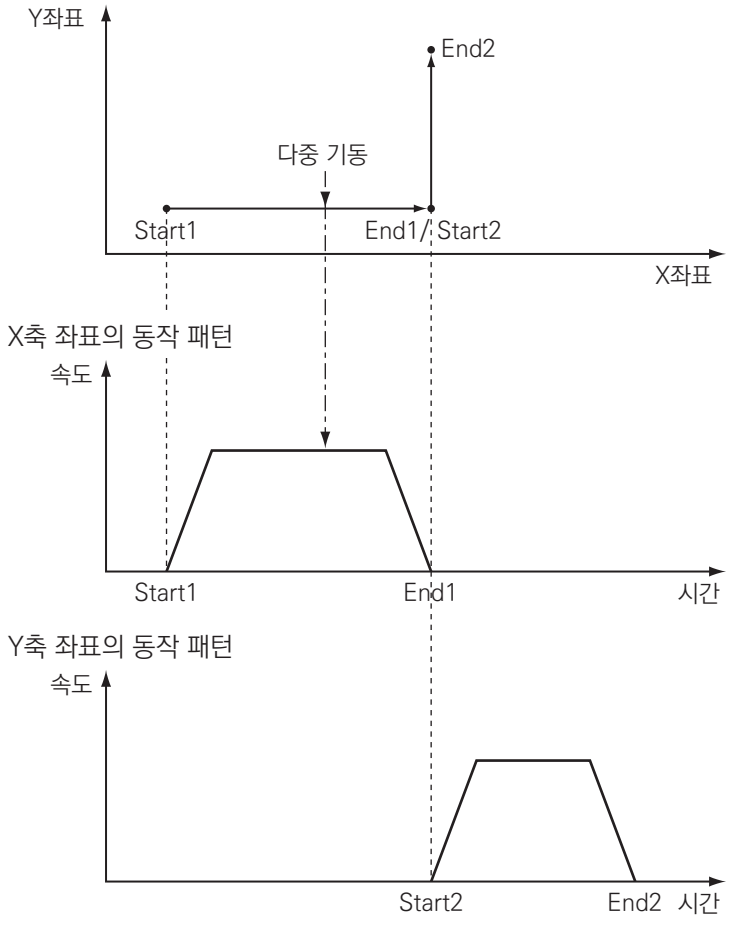
참고

PLCopen®의 기술 사양서에서는 No.0~9가 정의되어 있습니다. No.10은 MC 기능 모듈의 독자적인 사양입니다.

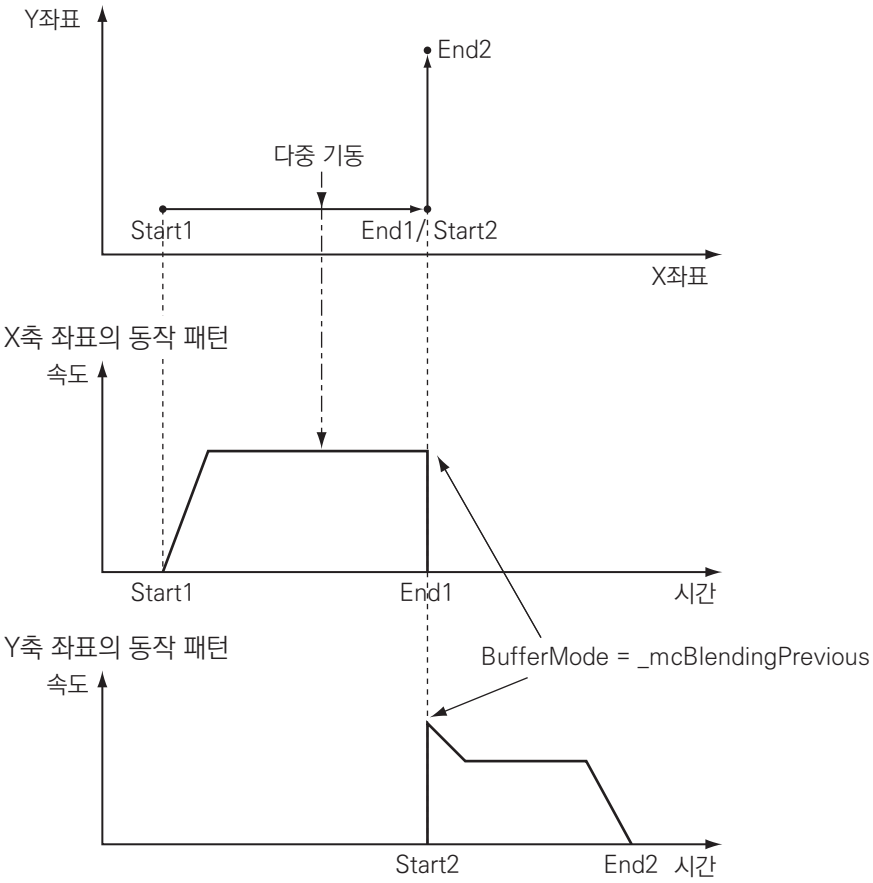
● 트랜지션 무효(0: _mcTMNone)의 경우

두 위치를 연결하기 위한 처리는 실행하지 않습니다.

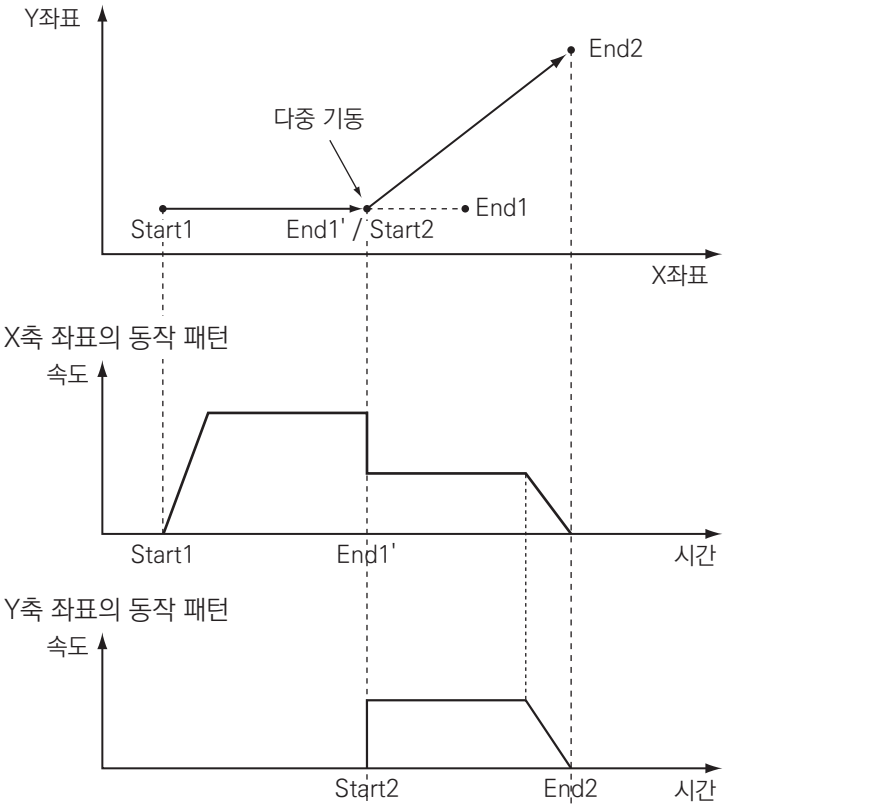
- 「BufferMode(버퍼 모드 선택)」가 [1: _mcBuffered]인 경우에는 다음과 같이 위치 End1까지 이동하고 정지한 후에 위치 End2로 이동합니다.



- 「BufferMode(버퍼 모드 선택)」가 블렌딩인 경우에는 다음과 같이 위치 End1까지 이동한 후에 위치 End2로 이동합니다.

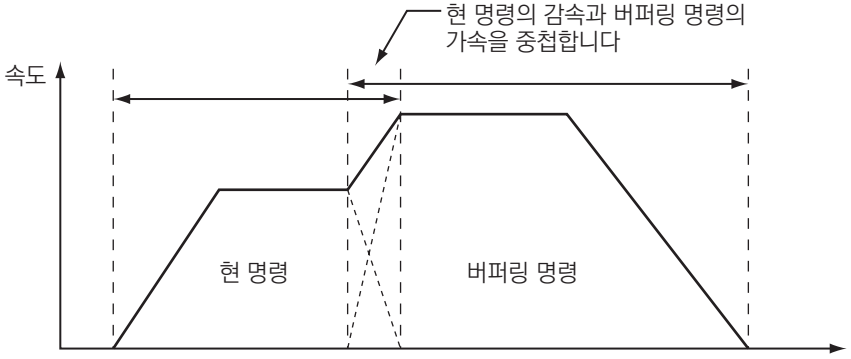


- 「BufferMode(버퍼 모드 선택)」가 [0: _mcAborting]인 경우에는 다음과 같이 다중 기동한 위치 End1에서 위치 End2로 이동합니다.



● **코너 슈퍼임포즈(10: _mcTMCornerSuperimposed)의 경우**

현 명령의 감속과 버퍼링 명령의 가속을 중첩합니다.
버퍼 명령의 가속도 지정과 상관없이 현 명령의 감속도에 의한 감속 시간과 같은 시간으로 동작합니다.
중첩되는 구간에서는 저크 지정과 상관없이 저크 없음이 됩니다.



_mcTMCornerSuperimposed의 경우 모션 제어 명령의 완료를 나타내는 출력 변수 「Done(완료)」은 중첩의 구간이 종료하는 타이밍에서 TRUE가 됩니다.

참고

아래의 2가지 조건을 동시에 만족하면 궤적의 선 속도가 일정해집니다.
- 현 명령과 버퍼링 명령의 목표 속도를 같은 값으로 한다.
- 현 명령의 감속도와 버퍼링 명령의 가속도를 같은 값으로 한다.

트랜지션 모드와 다중 기동의 조합

TransitionMode(트랜지션 모드)와 BufferMode(버퍼 모드 선택)에 의한 명령 다중 기동의 조합입니다.

○: 동작 가능 —: 이상이 발생하여 정지

트랜지션 모드	버퍼 모드					
	Aborting(중단)	버퍼드	블렌딩 로우	블렌딩 프리비어스	블렌딩 넥스트	블렌딩 하이
트랜지션 무효 (_mcTMNone)	○	○	○	○	○	○
코너 슈퍼임포즈 *1 (_mcTMCornerSuperimposed)	—	—	○	○	○	○

*1. 코너 슈퍼임포즈는 블렌딩의 종류와 관계없이 현 명령의 감속과 버퍼링 명령의 가속이 중첩됩니다.

9-8 기타 기능

여기에서는 MC 기능 모듈에 있는 기타 기능에 대해 설명합니다.
NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

9-8-1 현재 위치 변경 기능

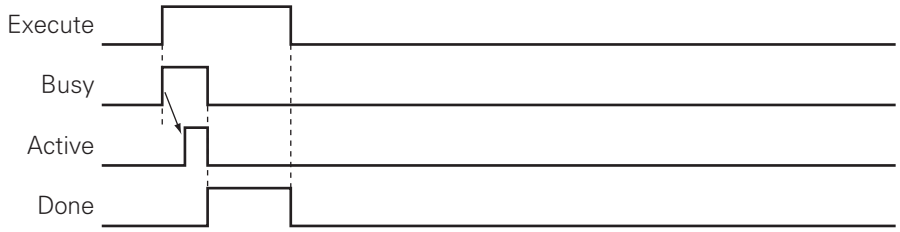
서보 축의 지령 현재 위치를 임의의 값으로 변경하고 피드백 현재 위치는 지령 현재 위치와의 위치 편차 값을 유지한 값으로 변경하는 기능입니다.
엔코더 축은 피드백 현재 위치를 변경합니다.
MC_SetPosition(현재 위치 변경) 명령을 사용하여 변경할 현재 위치를 지정합니다.

축 동작 중에 현재 위치를 변경할 수 있습니다.
절대값에 대한 위치 결정을 실행 중인 경우에는 변경한 절대 좌표상의 목표 위치로 위치를 결정합니다.
상대값으로 위치 결정을 실행 중인 경우 이동 거리는 변경하지 않습니다.

사용상의 주의

- 카운트 모드가 로터리 모드인 경우 링 카운터 범위 밖의 위치를 지정하면 명령에서 이상이 발생합니다.
- 현재 위치 변경이 끝나면 원점 미확정이 되므로 아래의 기능 및 명령은 사용할 수 없게 됩니다.
  소프트웨어 리미트 기능
  고속 원점 복귀
  보간 명령(직선 보간과 원호 보간)

● 축이 정지 중일 때 실행한 경우의 타이밍 차트



참고

MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령으로 「원점 프리셋」을 지정하면 원점 확정 상태에서 현재 위치를 변경할 수 있습니다.

MC_SetPosition(현재 위치 변경) 명령에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션 편(SBCE-364)』을 참조해 주십시오.

9-8-2 토크 제한 기능

서보 드라이버 토크 제한 기능의 유효/무효 변환과 토크 제한값을 설정하여 출력 토크를 제한하는 기능입니다.
정방향에 대한 토크 제한과 역방향에 대한 토크 제한을 따로 지정할 수 있습니다.

기능에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_SetTorqueLimit(토크 제한) 명령을 참조해 주십시오.

사용상의 주의

NX 시리즈 펄스 출력 유니트를 사용할 경우에는 이 기능을 사용할 수 없습니다.

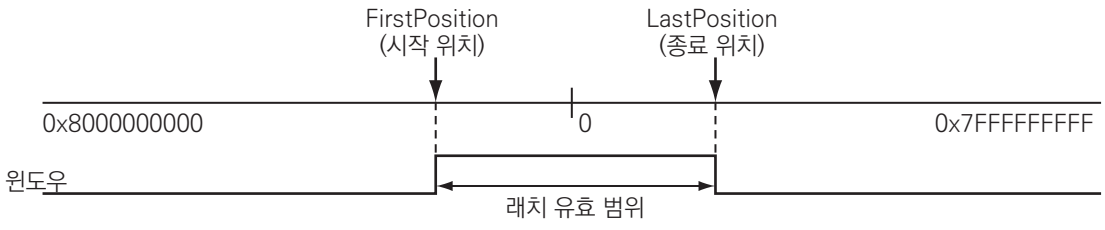
9-8-3 래치 기능

센서 입력 등의 트리거 신호 발생을 기점으로 한 위치 제어를 목적으로, 축의 위치를 트리거 신호 발생에 따라 기록(래치)하는 기능입니다.
축마다 동시에 2점의 트리거 신호를 설정할 수 있습니다.
MC_TouchProbe(외부 래치 유효) 명령을 사용하여 래치하고 싶은 축에 대해 「트리거 입력 조건」, 「윈도우 유효」, 「정지 방법 선택」을 지정할 수 있습니다.
트리거에는 서보 드라이버에 접속된 신호 외에 사용자 프로그램에서 취급하는 변수도 지정할 수 있습니다.
래치 기능을 중단하려면 MC_AbortTrigger(외부 래치 무효) 명령을 사용합니다.
래치 기능은 오므론 서보 드라이버 1S 시리즈 등 래치 기능(Touch probe)에 대응하는 서보 드라이버와 엔코더 입력 터미널 GX-EC0211/EC0241에서 사용할 수 있습니다.

WindowOnly(윈도우 유효)를 사용하여 시작점과 종료점의 범위 내에서만 트리거 신호를 검출합니다.
카운트 모드의 차이로 인해 범위는 아래와 같습니다.

● 리니어 모드

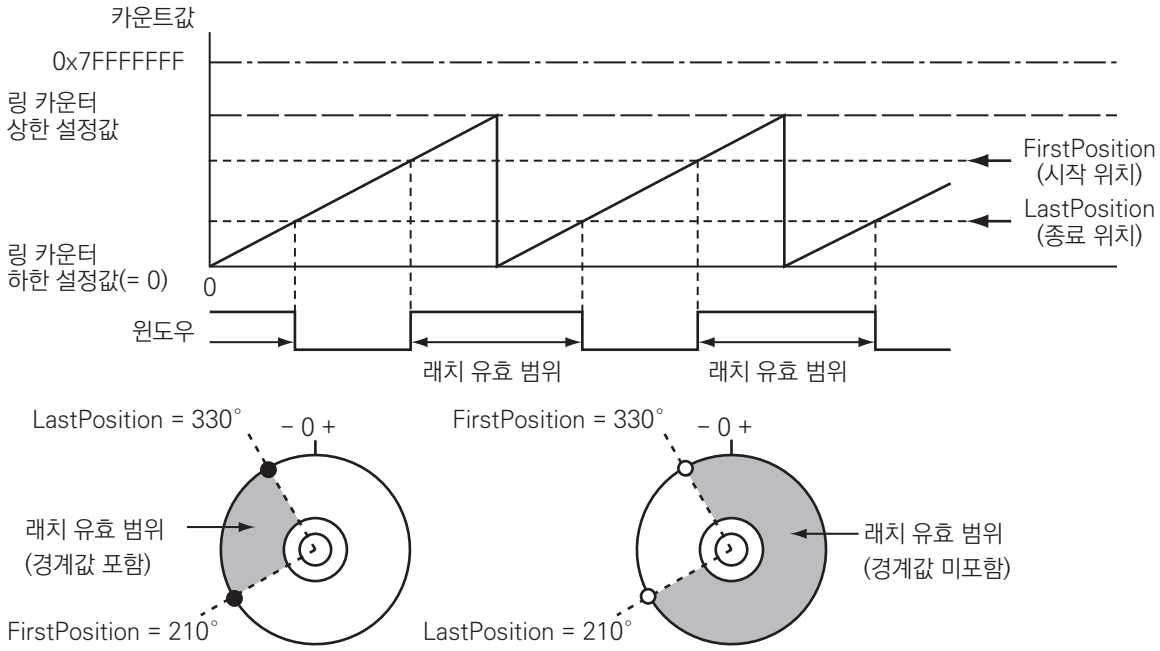
- FirstPosition(시작 위치) ≦ LastPosition(종료 위치)일 때만 가능합니다.
- FirstPosition(시작 위치) > LastPosition(종료 위치)으로 지정한 경우는 명령에서 이상이 발생합니다.
- 리니어 모드의 위치 범위를 초과해서 지정한 경우에는 명령에서 이상이 발생합니다.



● 로터리 모드

- FirstPosition(시작 위치) ≦ LastPosition(종료 위치),
  FirstPosition(시작 위치) > LastPosition(종료 위치) 모두 지정할 수 있습니다. 후자의 경우는 링 카운터 하한 설정값을 넘는 설정이 됩니다.
- 링 카운터 상하한의 범위를 초과하여 지정한 경우에는 명령에서 이상이 발생합니다.

	First Position ≦ Last Position	First Position > Last Position
유효 범위	FirstPosition~LastPosition	LastPosition~FirstPosition



래치 기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_TouchProbe(외부 래치 유효) 명령 및 MC_AbortTrigger(외부 래치 무효) 명령을 참조해 주십시오.

NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

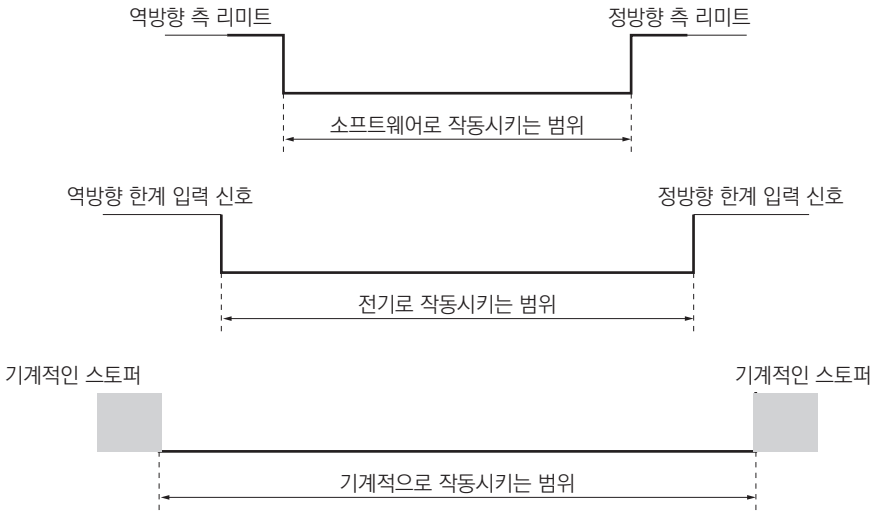
9-8-4 구역 기능

지정된 범위(구역) 내에 축의 지령 위치 혹은 피드백 위치가 존재하는지 판정하는 기능입니다.
MC_ZoneSwitch(구역 감시) 명령을 사용하여 판정할 범위의 시작 위치와 종료 위치를 지정합니다.
축의 위치가 지정한 구역 내에 있으면 구역 감시 명령의 출력 변수「InZone(구역 내)」가 TRUE가 됩니다.
1축에서 여러 구역을 지정할 수 있고 구역을 중복할 수도 있습니다.

구역 기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_ZoneSwitch(구역 감시) 명령을 참조해 주십시오.

9-8-5 소프트웨어 리미트 기능

하드웨어의 한계 입력 신호 입력과는 별도로 MC 기능 모듈의 소프트웨어로 현재 위치를 감시하는 기능입니다.
소프트웨어 리미트로 감시하는 범위는 축 파라메터 [Positive Software Limit]와 [Negative Software Limit]를 설정합니다.
일반적인 위치 결정 기능에서는 이 소프트웨어 리미트의 범위 내에서 동작할 수 있습니다.
소프트웨어 리미트를 설정하면 사용자 프로그램의 오류 및 오조작에 의한 기기 파손을 방지할 수 있습니다.



● 소프트웨어 리미트 기능에 관한 축 파라메터

파라메터 명칭	기능	설정 범위	초기값
소프트웨어 리미트 기능	소프트웨어 리미트의 기능을 선택합니다. 0: 무효 1: 지령 위치에 대해 유효하면 감속 정지 *1 2: 지령 위치에 대해 유효하면 즉시 정지 3: 피드백 위치에 대해 유효하면 감속 정지 *1 4: 피드백 위치에 대해 유효하면 즉시 정지	0~4	0
정방향 소프트웨어 리미트	정방향 소프트웨어 리미트값을 설정합니다. 단위는 지령 단위입니다.	배정밀도 실수형의 음수, 양수, 0 *2	2147483647
역방향 소프트웨어 리미트	역방향 소프트웨어 리미트값을 설정합니다. 단위는 지령 단위입니다.		-2147483648

*1. 입력 변수「Deceleration(감속도)」을 가진 명령을 실행하는 도중에 현재 위치가 소프트웨어 리미트 범위 외로 나오면「Deceleration(감속도)」으로 감속 정지합니다. 입력 변수「Deceleration(감속도)」을 갖지 못한 명령을 실행하는 도중에 현재 위치가 소프트웨어 리미트 범위 외로 나오면 축 파라메터의 최대 감속도로 감속 정지합니다.
*2. 펄스 단위로 변환한 값을 부호가 있는 정수형 40비트 범위에서 설정할 수 있습니다.

상기 축 파라메터는 Sysmac Studio의 축 설정, MC_Write(MC 설정 기록) 명령, MC_WriteAxisParameter(축 파라메터 기록) 명령으로 설정할 수 있습니다.
축 및 축 그룹 동작 중에 설정값을 변경하면 다음 동작 시작 시에 변경한 설정이 유효해집니다.

소프트웨어 리미트는 축의 동작 상태나 모션 제어 명령을 이용하여 다음 2가지인 경우에 기능합니다.

● 모션 명령 기동 시

- 현재 위치가 소프트웨어 리미트 범위 내인 경우에는 목표 위치가 소프트웨어 리미트 범위 외에 있으면 명령에서 이상이 발생합니다.
- 현재 위치가 소프트웨어 리미트 범위 외인 경우에는 소프트웨어 리미트 범위 내로 이동할 수 있습니다. 이때 목표 위치가 소프트웨어 리미트 범위 내에 있지 않아도 됩니다.



사용상의 주의

축 지령 명령은 목표 위치를 소프트웨어 리미트 범위 외로 하여 기동하지 마십시오.

● 축이 동작 중일 때

위치 결정 동작 중, 동기 동작 중, 연속 동작 중, 다축 협조 동작 중인 축이 움직이고 있는 경우에는 아래와 같이 동작합니다.

- 소프트웨어 리미트 기능이 지령 위치에 대해 유효인 경우 지령 위치가 범위 내에서 범위 외로 나오면 축에 이상이 발생합니다.
- 소프트웨어 리미트 기능이 피드백 위치에 대해 유효인 경우 피드백 위치가 범위 내에서 범위 외로 나오면 축에 이상이 발생합니다.



참고

소프트웨어 리미트 기능은 카운트 모드가 리니어 모드이고 원점 확정 상태일 때 유효하게 설정할 수 있습니다.
다음의 경우에는 축 파라메터 설정과 상관없이 소프트웨어 리미트 기능이 무효합니다.

- 카운트 모드가 로터리 모드인 경우
- 원점이 미확정 상태인 경우
- 원점 복귀 동작 중인 경우

MC 설정 기록 명령 및 축 파라메터 기록 명령에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_Write(MC 설정 기록) 명령 및 MC_WriteAxisParameter(축 파라메터 기록) 명령을 참조해 주십시오.

9-8-6 위치 편차 감시 기능

축의 지령 위치와 피드백 위치의 차분이 위치 편차입니다. MC 기능 모듈은 모션 제어 주기별로 위치 편차를 감시합니다.
위치 편차의 값이 축 파라메터의「위치 편차 초과값」을 초과한 경우 경도 폴트 레벨의「위치 편차 초과」가 발생합니다. 또한,「위치 편차 경고값」을 초과한 경우에는 감시 정보의「위치 편차 경고」가 발생합니다.
원점 복귀에서 보류 동작을 실행 중인 경우에는 위치 편차 감시가 무효해집니다.

● 위치 편차 감시 기능에 관한 축 파라메터

축 파라메터 설정으로 위치 편차 감시의 체크값을 설정할 수 있습니다.「위치 편차 경고값」은「위치 편차 초과값」이하가 되도록 설정해 주십시오.
이러한 축 파라메터는 Sysmac Studio로 설정합니다.

파라메터 명칭	기능	설정 범위	초기값
위치 편차 초과값	위치 편차 초과 체크값을 설정합니다. 「0」을 설정하면 위치 편차 초과 체크가 무효해집니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0	0
위치 편차 경고값	위치 편차 경고 체크값을 설정합니다. 「0」을 설정하면 위치 편차 경고 체크가 무효해집니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0 (위치 편차 초과값 이하)	0

NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

9-8-7 편차 카운터 리셋

편차 카운터 리셋은 위치 편차를「0」으로 하는 기능입니다.
사용자 프로그램에서 편차 카운터 리셋을 하려면「MC_ResetFollowingError(편차 카운터 리셋)」명령을 사용합니다.
편차 카운터 리셋 명령은 단축 위치 결정 중일 때와 원점 복귀 중일 때 사용할 수 있습니다.
축 동작 중에 편차 카운터 리셋을 실행하면 동작 중인 모션 제어 명령은 동작을 중단하고 지령 위치를 피드백 위치와 같은 값으로 변경합니다.
편차 카운터 리셋을 실행해도 원점 확정 상태는 변하지 않습니다.

편차 카운터 리셋 명령에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_ResetFollowingError(편차 카운터 리셋) 명령을 참조해 주십시오.

● 편차 카운터 리셋에 관한 축 파라메터

축 파라메터 설정으로 즉시 정지 시, 한계 입력 정지 시, 원점 복귀 완료 시에 편차 카운터 리셋을 하도록 선택할 수 있습니다.
이러한 축 파라메터는 Sysmac Studio로 설정합니다.

파라메터 명칭	기능	설정 범위	초기값
즉시 정지 입력 정지 방법	즉시 정지 입력이 유효해졌을 때 MC 기능 모듈 측에서 정지하는 방법을 설정합니다. 0: 지령값을 즉시 정지 2: 지령값을 즉시 정지하고 동시에 편차 카운터 리셋을 실행 3: 지령값을 즉시 정지하고 동시에 서보 OFF를 실행	0, 2, 3	0
한계 입력 정지 방법	정방향 한계 입력 또는 역방향 한계 입력이 유효해졌을 때 MC 기능 모듈 측에서 정지하는 방법을 설정합니다. 0: 지령값을 즉시 정지 1: 지령값을 감속 정지 2: 지령값을 즉시 정지하고 동시에 편차 카운터 리셋을 실행 3: 지령값을 즉시 정지하고 동시에 서보 OFF를 실행	0~3	0

NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

9-8-8 축 간 편차 감시 기능

2축 간 지령 위치 혹은 피드백 위치의 편차량을 감시하는 기능입니다.
MC_AxesObserve(축 간 편차 감시) 명령을 사용하여 감시 대상인 2축과 축간 편차 허용값을 지정합니다.
허용값을 초과하면 축 간 편차 감시 명령의 출력 변수「Invalid(편차 초과)」가 TRUE가 됩니다.

갠트리 제어 등 2축에서 같은 동작을 하는 장비는 축 간 감시 기능을 사용하여 축 간 편차가 과대해진 경우의 처리를 자유롭게 프로그램할 수 있습니다.

사용상의 주의

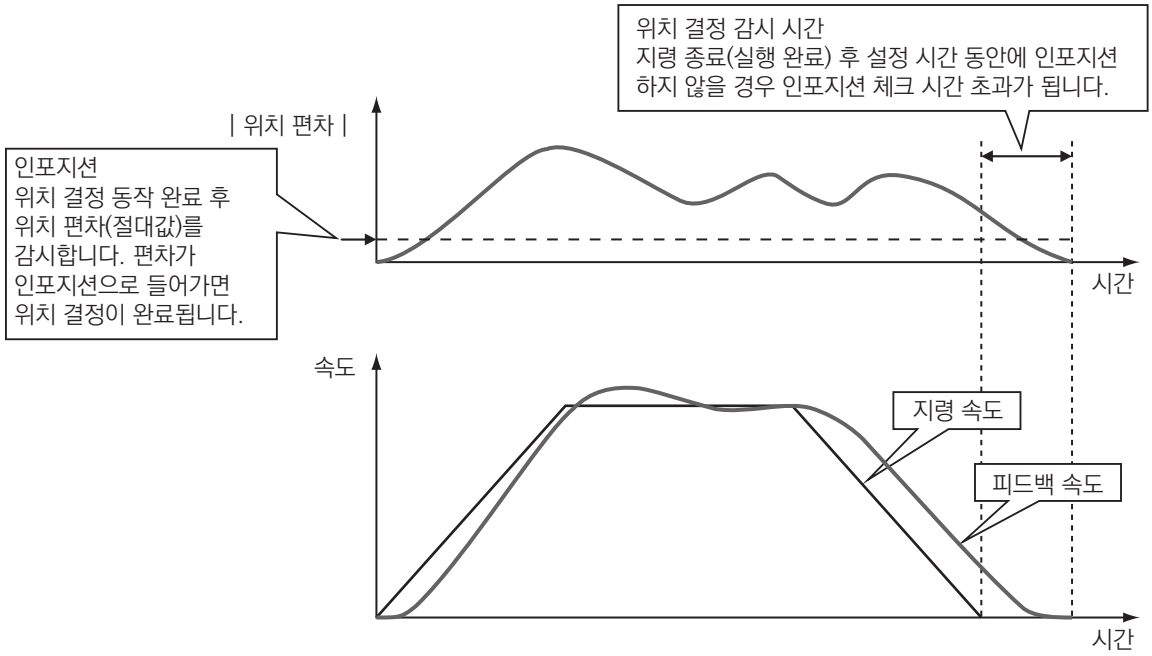
축간 편차 허용값을 초과해도 MC 기능 모듈은 이상이 되지 않습니다.
출력 변수「Invalid(편차 초과)」로 사용자 프로그램에서 축 동작을 정지하는 등의 프로그램 처리를 실행해 주십시오.

축 간 편차 감시 기능 기능에 대한 상세한 내용은 🕮『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_AxesObserve(축 간 편차 감시 기능) 명령을 참조해 주십시오.

NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

9-8-9 인포지션 체크

위치 결정 동작 및 원점 복귀 동작 시 실제 피드백 현재 위치가 목표 위치에 대해 지정한 범위 내에 도달했는지를 체크하는 기능입니다.
목표 위치의 지령 출력 완료 후에 목표 위치와 피드백 현재 위치의 차이가 인포지션 폭 이하가 되면 위치 결정이 완료됩니다.
인포지션 체크 시간 이내에 위치 결정을 완료하지 않으면 명령에서 이상이 발생합니다.



● 인포지션 체크에 관한 축 파라메터

축 파라메터로 위치 결정 완료 체크 조건을 설정할 수 있습니다.
위치 결정을 확실히 완료한 후에 다음 동작을 시작하려면 인포지션 체크 시간을 설정합니다.

파라메터 명칭	기능	설정 범위	초기값
인포지션 폭	위치 결정 완료 폭을 설정합니다. (단위: 지령 단위)	배정밀도 실수형의 양수, 0	10
인포지션 체크 시간	위치 결정 완료 체크 시간을 밀리초 단위로 설정합니다. 「0」을 설정하면 원점 복귀 시에 원점을 확정했을 때에만 위치 결정 완료 체크를 계속하고, 원점 복귀 시 이외에는 위치 결정 완료 체크는 하지 않습니다. (단위: ms)	0~10000	0

상기 축 파라메터는 Sysmac Studio의 축 설정, MC_Write(MC 설정 기록) 명령, MC_WriteAxisParameter(축 파라메터 기록) 명령으로 설정할 수 있습니다.

참고

- 인포지션 체크는 MC 기능 모듈 측에서 처리합니다. 서보 드라이버 측 기능은 사용하지 않습니다.
- 조금이라도 빨리 동작시키려면 위치 결정 완료를 기다리지 않고 다음 동작으로 이행시키기 위해 인포지션 체크 시간을 설정하지 않습니다.
- CPU 유니트의 전원을 재투입하거나 Sysmac Studio의「Synchronization」기능으로 프로그램을 다운로드하면 Sysmac Studio에서 설정한 값으로 돌아갑니다.
  MC_Write(MC 설정 기록) 명령 및 MC_WriteAxisParameter(축 파라메터 기록) 명령은 일시적으로 인포지션 체크 시간을 변경할 때 사용합니다.

● 인포지션 체크에 관한 모니터

사용자 프로그램에서는 축 변수를 판독하여 위치 결정 완료를 모니터할 수 있습니다.
또한, 변수명은 _MC_AX[*]를 예로 들어 설명하고 있으나 _MC1_AX[*], _MC2_AX[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_AX[0-255].Details.Idle	BOOL	정지 중	인포지션 대기를 제외하고 지령값을 연산하지 않을 때 TRUE가 됩니다. *1 Idle과 InPosWaiting은 배타적으로 동시에는 TRUE가 되지 않습니다.
_MC_AX[0-255].Details.InPosWaiting	BOOL	인포지션 대기	인포지션 대기일 때 TRUE가 됩니다. 인포지션 체크를 실행하는 위치 결정에서 인포지션 체크를 실행하고 있는 상태입니다.

*1. 속도「0」에서의 동작, 편차 카운터 리셋 중, 동기 제어 중, 다축 협조 동작 중도 연산하고 있는 상태에 포함됩니다.

축 그룹에서 위치 결정을 할 때는 축 그룹 변수를 판독하여 위치 결정 완료를 모니터할 수 있습니다.
또한, 변수명은 _MC_GRP[*]를 예로 들어 설명하고 있으나 _MC1_GRP[*], _MC2_GRP[*]의 경우도 동일합니다.

변수명	데이터형	명칭	기능
_MC_GRP[0-63].Details.Idle	BOOL	정지 중	인포지션 대기를 제외하고 지령값을 연산하지 않을 때 TRUE가 됩니다. *1 Idle과 InPosWaiting은 배타적으로 동시에는 TRUE가 되지 않습니다.
_MC_GRP[0-63].Details.InPosWaiting	BOOL	인포지션 대기	어느 한 구성 축이 인포지션 대기일 때 TRUE가 됩니다. *2 인포지션 체크를 실행하는 위치 결정에서 인포지션 체크를 실행하고 있는 상태입니다.

*1. 속도「0」에서의 동작도 연산 중인 상태에 포함됩니다.
*2. 축 그룹 내의 모든 구성 축이 각 축 파라메터의 인포지션 폭 범위 내에 들어가면 FALSE가 됩니다.

MC 설정 기록 명령 및 축 파라메터 기록 명령에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_Write(MC 설정 기록) 명령 및 MC_WriteAxisParameter(축 파라메터 기록) 명령을 참조해 주십시오.

NX 시리즈 펄스 출력 유니트를 사용할 경우의 차이에 대해서는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

9-8-10 축 사용 변경 기능

MC_ChangeAxisUse(축 사용 변경) 명령을 이용하여 축 파라메터 [Axis Use]를 일시적으로 변환하는 기능입니다. 변환할 수 있는 축은 축 파라메터 [Axis Use]가 [Unused axis (changeable to used axis)] 또는 [Used axis]로 설정된 축입니다.
[Axis Use]가 [Unused axis (changeable to used axis)]이고 [Axis Type]이 [servo axis] 또는 [virtual servo axis]인 경우 축 그룹에 할당할 수 있습니다.
유니트 버전 Ver.1.04이상인 CPU 유니트와 Sysmac Studio Ver.1.05 이상의 조합으로 사용할 수 있습니다.

사용상의 주의

- [Unused axis (unchangeable to used axis)]로 설정된 축을 [Used axis]로 변환하지 마십시오.
- 축 파라메터 [Axis Use]가 [Unused axis(unchangeable to used axis)]로 설정된 경우 축 그룹에 할당할 수 없습니다.

기능에 대한 상세한 내용은 『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_ChangeAxisUse(축 사용 변경) 명령을 참조해 주십시오.

MC_ChangeAxisUse(축 사용 변경) 명령의 어플리케이션 예는 『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

9-8-11 디지털 캠 스위치 유효 기능

MC_DigitalCamSwitch(디지털 캠 스위치 유효) 명령을 이용하여 축 위치에 적합한 디지털 출력을 ON 또는 OFF합니다.
명령의 입력 변수「ValueSource」의 설정에 따라 가감 속도를 추가할 수도 있습니다.
이 기능은 반드시 NX_AryDOutTimeStamp 명령 및 타임 스탬프 방식 대응 디지털 출력 유니트와 조합하여 사용합니다. NX_AryDOutTimeStamp 명령은 지정한 타임 스탬프의 타이밍으로 지정한 디지털 출력을 ON/OFF합니다.
유니트 버전 Ver.1.06 이상인 CPU 유니트와 Sysmac Studio Ver.1.07 이상의 조합으로 사용할 수 있습니다.

사용상의 주의

본 명령은 NX 시리즈 위치 인터페이스 유니트를 할당하고 있는 축에 대해 유효합니다.
사용 가능한 NX 유니트는 NX-EC0□□□, NX-ECS□□□에서 변화 시각 취득 기능이 작동하고 있어야 합니다.

디지털 캠 스위치 유효 기능에 대한 상세한 내용은 『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』의 MC_DigitalCamSwitch(디지털 캠 스위치 유효) 명령을 참조해 주십시오.

NX_AryDOutTimeStamp 명령에 대한 상세한 내용은 『NJ/NX 시리즈 명령 참조 기본편(SBCA-360)』을 참조해 주십시오.

타임 스탬프 방식에 대응하는 디지털 출력 유니트에 대해서는 『NX 시리즈 디지털 I/O 유니트 사용자 매뉴얼(SBCA-407B 이상)』을 참조해 주십시오.

변화 시각 취득 기능 및 타임 스탬프에 대해서는 『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374B 이상)』을 참조해 주십시오.

9-8-12 임의 좌표계의 3D 모션 모니터 표시

스칼라 로봇이나 수직 다관절 로봇 등 직교 좌표계 이외의 좌표계를 사용자 프로그램에서 실현하고 있는 경우에 로봇의 손끝 등의 궤적을 Sysmac Studio에서 3D로 표시하는 기능입니다.
_sMC_POSITION_REF형의 사용자 정의 변수를 작성하고 3D 모션 모니터 표시로 표시합니다.

이 기능은 유니트 버전 Ver.1.06 이상인 CPU 유니트와 Sysmac Studio Ver.1.07 이상의 조합으로 사용할 수 있습니다.

● _sMC_POSITION_REF형

_sMC_POSITION_REF형의 멤버는 아래와 같습니다.

멤버	데이터형	명칭
CommandPosition	ARRAY [0..5] OF LREAL	지령 현재 위치
ActualPosition	ARRAY [0..5] OF LREAL	피드백 현재 위치

각 멤버에 대해 설명합니다.

멤버	설명
사용자 정의 변수.CommandPosition[0]	지령 현재 위치의 X축 성분입니다. 사용자 프로그램에서 생성하는 지령 현재 위치의 X축상의 위치를 나타내는 사용자 정의 변수를 대입합니다.
사용자 정의 변수.CommandPosition[1]	지령 현재 위치의 Y축 성분입니다. 사용자 프로그램에서 생성하는 지령 현재 위치의 Y축상의 위치를 나타내는 사용자 정의 변수를 대입합니다.
사용자 정의 변수.CommandPosition[2]	지령 현재 위치의 Z축 성분입니다. 사용자 프로그램에서 생성하는 지령 현재 위치의 Z축상의 위치를 나타내는 사용자 정의 변수를 대입합니다.
사용자 정의 변수.CommandPosition[3]~[5]	미사용
사용자 정의 변수.ActualPosition[0]	피드백 현재 위치의 X축 성분입니다. 사용자 프로그램에서 취급하는 피드백 현재 위치의 X축상의 위치를 나타내는 사용자 정의 변수를 대입합니다.
사용자 정의 변수.ActualPosition[1]	피드백 현재 위치의 Y축 성분입니다. 사용자 프로그램에서 취급하는 피드백 현재 위치의 Y축상의 위치를 나타내는 사용자 정의 변수를 대입합니다.
사용자 정의 변수.ActualPosition[2]	피드백 현재 위치의 Z축 성분입니다. 사용자 프로그램에서 취급하는 피드백 현재 위치의 Z축상의 위치를 나타내는 사용자 정의 변수를 대입합니다.
사용자 정의 변수.ActualPosition[3]~[5]	미사용

각 멤버에는 사용자 정의 변수를 대입합니다. 다음에 예를 나타냅니다.

명칭	데이터형	설명
3D_position	_sMC_POSITION_REF	3D로 표시하기 위한 사용자 정의 변수
MCS_Cmd_TransX	LREAL	사용자 프로그램에서 생성하는 지령 현재 위치의 X축상의 위치를 나타내는 사용자 정의 변수
MCS_Cmd_TransY	LREAL	사용자 프로그램에서 생성하는 지령 현재 위치의 Y축상의 위치를 나타내는 사용자 정의 변수
MCS_Cmd_TransZ	LREAL	사용자 프로그램에서 생성하는 지령 현재 위치의 Z축상의 위치를 나타내는 사용자 정의 변수
MCS_Act_TransX	LREAL	사용자 프로그램에서 취급하는 피드백 현재 위치의 X축상의 위치를 나타내는 사용자 정의 변수
MCS_Act_TransY	LREAL	사용자 프로그램에서 취급하는 피드백 현재 위치의 Y축상의 위치를 나타내는 사용자 정의 변수
MCS_Act_TransZ	LREAL	사용자 프로그램에서 취급하는 피드백 현재 위치의 Z축상의 위치를 나타내는 사용자 정의 변수

```
3D_position.CommandPosition[0] := MCS_Cmd_TransX;
3D_position.CommandPosition[1] := MCS_Cmd_TransY;
3D_position.CommandPosition[2] := MCS_Cmd_TransZ;
3D_position.ActualPosition[0] := MCS_Act_TransX;
3D_position.ActualPosition[1] := MCS_Act_TransY;
3D_position.ActualPosition[2] := MCS_Act_TransZ;
```

● 사용 순서의 개요

1 **_sMC_POSITION_REF형의 사용자 정의 변수를 작성합니다.**

2 **작성한 사용자 정의 변수의 각 멤버에 3D 표시할 「지령 현재 위치」와 「피드백 현재 위치」를 나타내는 사용자 정의 변수를 대입할 프로그램을 작성합니다.**

3 **3D 기구 모델 리스트의 종별에서 [Specified coordinate]를 선택합니다.**
3D 기구 모델 파라메터 설정에 _sMC_POSITION_REF형이 표시됩니다.

4 **3D 기구 모델 파라메터 설정의 설정값란에 작성한 사용자 정의 변수를 설정합니다.**

5 **사용자 프로그램을 실행합니다.**

6 **데이터 트레이스로 트레이스를 시작하고 데이터 샘플링을 실행합니다.**

7 **데이터 트레이스 화면에서 트레이스 결과를 확인합니다.**

3D 모션 모니터 표시에 대한 상세한 내용은 🕮『Sysmac Studio Version 1 오퍼레이션 매뉴얼 (SBCA-470)』을 참조해 주십시오.

10

샘플 프로그램

이 장에서는 원점 복귀 및 이상 감시 등의 기본적인 사용 방법과 절대값 위치 결정, 캠 동작 등 축 동작에 관한 프로그램 예가 기재되어 있습니다.

10-1 샘플 프로그램의 개요 10-2
10-1-1 사용 기기 10-2
10-1-2 설치와 배선 10-2
10-1-3 설정 10-2

10-2 기본 프로그램 예 10-3
10-2-1 EtherCAT 통신의 감시와 서보 ON 10-3
10-2-2 마스터 컨트롤 명령에 의한 축 동작의 인터락 10-5
10-2-3 이상 감시와 이상 해제(단축 동작, 동기 동작) 10-7
10-2-4 이상 감시와 이상 해제(다축 협조 동작) 10-9
10-2-5 명령 이상 감시 방법 10-14
10-2-6 이상 해제 감시 방법 10-16
10-2-7 단축 동작 중의 축 정지 10-18
10-2-8 다축 협조 동작 중의 축 그룹 정지 10-22
10-2-9 원점 복귀와 절대값 위치 결정 10-28
10-2-10 명령 재기동으로 목표 위치 변경하기 10-32
10-2-11 인터럽트 피딩 위치 결정 10-37
10-2-12 명령 재기동으로 캠 테이블 변환하기 10-42
10-2-13 캠 프로파일 곡선을 사용하여 동기 위치 보정하기 10-51
10-2-14 캠 동작 시에 주축의 위상 보정하기 10-60
10-2-15 속도 제어 중에 현재 위치 변경하기 10-68
10-2-16 캠 데이터 변수의 변경과 캠 테이블 저장 10-74
10-2-17 축 파라메터의 일시 갱신 10-83
10-2-18 캠 테이블의 종료점 인덱스 갱신 10-86

10-1 샘플 프로그램의 개요

여기에서는 본 장에 기재된 샘플 프로그램의 공통 사항에 대해 설명합니다

사용상의 주의

- 본 장의 샘플 프로그램은 MC 기능 모듈의 기능을 사용하는 부분만 기재되어 있습니다.
- 실제 장비의 프로그램을 작성할 경우에는 장비의 동작에 관한 인터락 프로그램 및 기타 기기에서의 입출력, 제어 순서 등에 관한 프로그램을 작성해 주십시오.
- 실제 장비가 의도한 대로 동작하도록 사용자 프로그램을 작성해 주십시오.
- 작성한 사용자 프로그램은 충분히 동작을 확인한 후 본 운전으로 이행해 주십시오.
- NX 시리즈 위치 인터페이스 유니트의 사용 예는 🕮『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

10-1-1 사용 기기

여기에서 설명하는 샘플 프로그램에서는 다음과 같은 디바이스의 사용을 전제로 합니다.

기기	서보 구성의 예
CPU 유니트	NJ501-1□00 (Ver.1.0)
전원 유니트	NJ-Px3001
서보 드라이버	R88D-1SN□□□-ECT
서보 모터	R88M-1□□□□□□□
엔코더 입력 터미널	GX-EC0211 (Ver.1.1)

10-1-2 설치와 배선

각 사용 기기의 설치와 배선에 대해서는 아래의 매뉴얼을 참조해 주십시오.

사용 기기	참조 매뉴얼
CPU 유니트와 전원 유니트	NJ 시리즈 CPU 유니트 사용자 매뉴얼 하드웨어편(SBCA-358)
서보 드라이버/모터	AC 서보 모터/드라이버 1S 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼 (SBCE-377)
엔코더 입력 터미널	GX 시리즈 EtherCAT 슬레이브 사용자 매뉴얼(SBCD-350)
EtherCAT 통신의 배선	

10-1-3 설정

각종 설정에 대해서는 아래의 매뉴얼을 참조해 주십시오.

사용 기기	참조 매뉴얼
컨트롤러 설정	NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)
모션 제어 설정	「3-2 축 설정 순서 (P.3-9)」, 「5-2 축 파라메터 (P.5-4)」, 「A-1 서보 드라이버 1S 시리즈와의 접속 (P.A-2)」
서보 파라메터	AC 서보 모터/드라이버 1S 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼 (SBCE-377)

10-2 기본 프로그램 예

여기에서는 MC 기능 모듈의 기본 기능에 관한 사용자 프로그램 예를 설명합니다.

사용상의 주의

- 본 프로그램 예를 참고할 경우에는 장비의 동작 조건에 맞는 적절한 인터락 프로그램을 추가해 주십시오.
- 프로그램에서 사용하는 변수는 Sysmac Studio의 [Programming] 화면에서 입력해 주십시오.

10-2-1 EtherCAT 통신의 감시와 서보 ON

대상인 서보 드라이버와 EtherCAT 통신의 프로세스 데이터 통신이 확립되면 MC_Power(운전 가능) 명령으로 서보 드라이버를 서보 ON합니다.

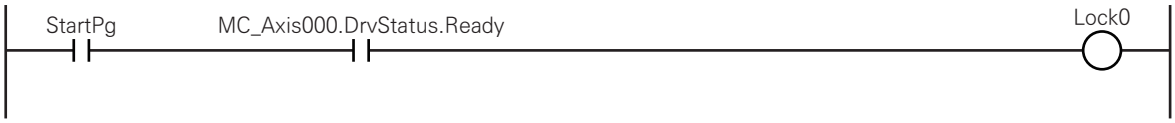
래더도 프로그램의 샘플과 ST 프로그램의 샘플을 기재합니다.

프로그램에서 사용하는 주요 변수

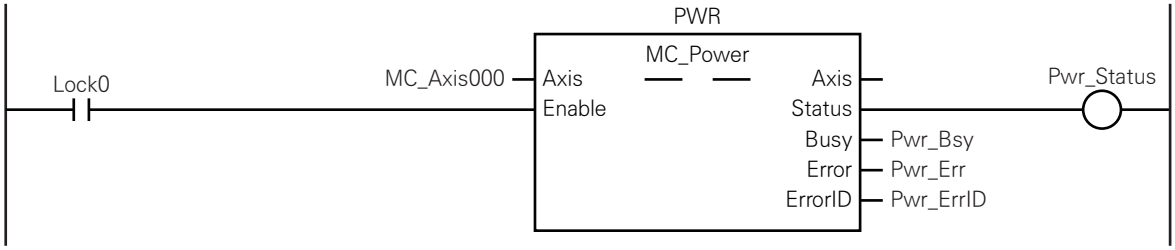
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Cfg.NodeAddress	UINT	—	노드 주소입니다
_EC_PDSlavTbl[N]	BOOL	FALSE	TRUE일 때 노드 주소 N의 EtherCAT 통신의 프로세스 데이터 통신이 오퍼레이셔널 상태임을 나타냅니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON 합니다.

래더도 프로그램

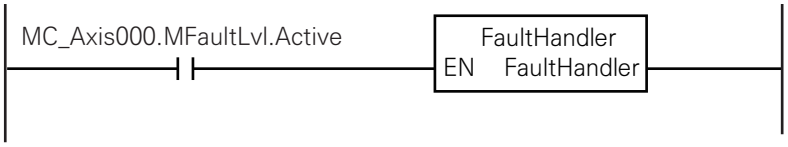
접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태인지 확인합니다.



서보 드라이버가 서보 준비 상태이면 축 0을 서버 ON합니다.



축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



ST 프로그램

```
//접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr_En:=TRUE;
ELSE
   Pwr_En:=FALSE;
END_IF;


// 축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;

// MC_Power
PWR(
   Axis := MC_Axis000,
   Enable := Pwr_En,
   Status => Pwr_Status,
   Busy => Pwr_Bsy,
   Error => Pwr_Err,
   ErrorID => Pwr_ErrID
);
```

10-2-2 마스터 컨트롤 명령에 의한 축 동작의 인터락

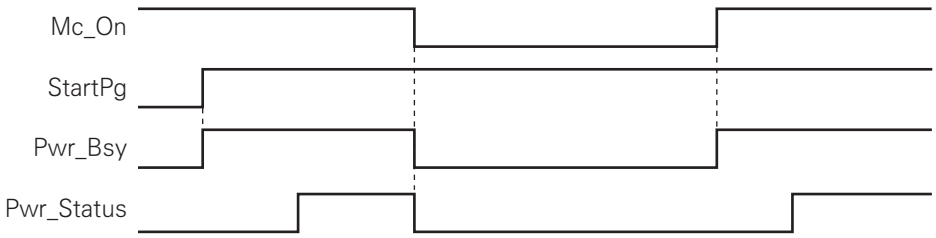
래더도 프로그램에서 MC(마스터 컨트롤) 명령과 MCR(마스터 컨트롤 리셋) 명령 사이에 MC_Power(운전 가능) 명령을 배치하여 축 동작의 인터락을 제어합니다.
접점 Mc_On을 FALSE로 하면 MC 명령과 MCR 명령 사이에 기술한 MC_Power(운전 가능) 명령이 무효해지고 서보 OFF됩니다.
이때 실행 중인 동작 명령의 출력 변수「CommandAborted(실행 중단)」가 TRUE가 되고 축 동작은 정지합니다.

MC 명령은 ST 프로그램에서 사용할 수 없습니다.
따라서 래더도 프로그램의 샘플만 기재합니다.

프로그램에서 사용하는 주요 변수

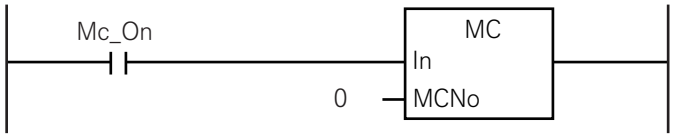
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
Mc_On	BOOL	FALSE	MC 명령의 유효 및 무효를 제어합니다. 이 샘플 프로그램에서는 제어할 기술이 기재되어 있지 않습니다. 실제 프로그램에서는 장비의 동작에 맞춰 제어해 주십시오.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON 합니다.

타이밍 차트



래더도 프로그램

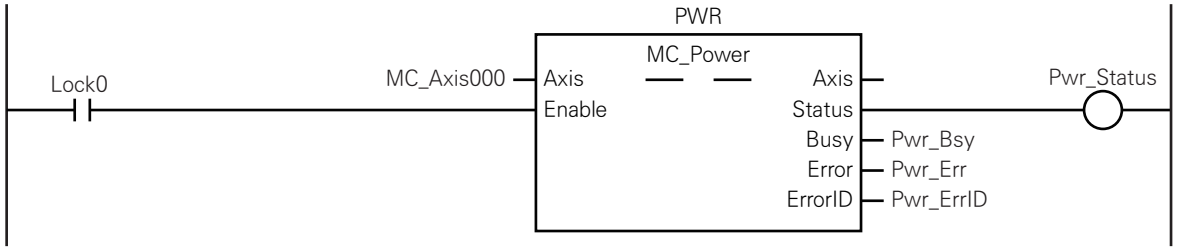
접점 Mc_On이 TRUE가 되면 마스터 컨트롤을 시작합니다.



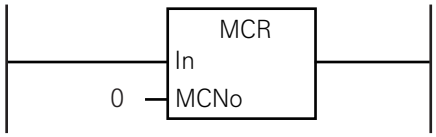
접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태인지 확인합니다.



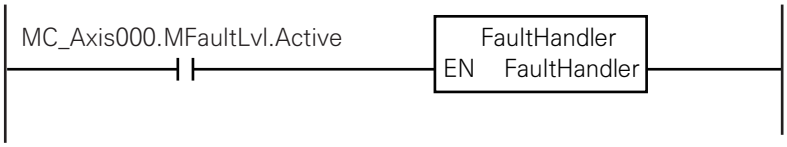
서보 드라이버가 서보 준비 상태이면 축 0을 서버 ON합니다.



마스터 컨트롤러를 종료합니다.



축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



10-2-3 이상 감시와 이상 해제(단축 동작, 동기 동작)

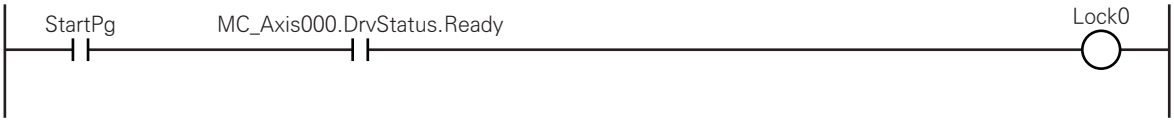
축 변수에 있는「축 경도 폴트 발생 중」인 상태를 통해 이상 여부를 감시합니다.
경도 폴트 레벨의 이상이 발생한 경우에는 MC_Power(운전 가능) 명령의 입력 변수「Enable(유효)」이 FALSE가 되어 서보 OFF합니다.
외부 버튼이 ON이고 지령 현재 속도가 제로이면 MC_Reset(축 에러 리셋) 명령으로 이상을 해제합니다.
래더도 프로그램의 샘플과 ST 프로그램의 샘플을 기재합니다.

프로그램에서 사용하는 주요 변수

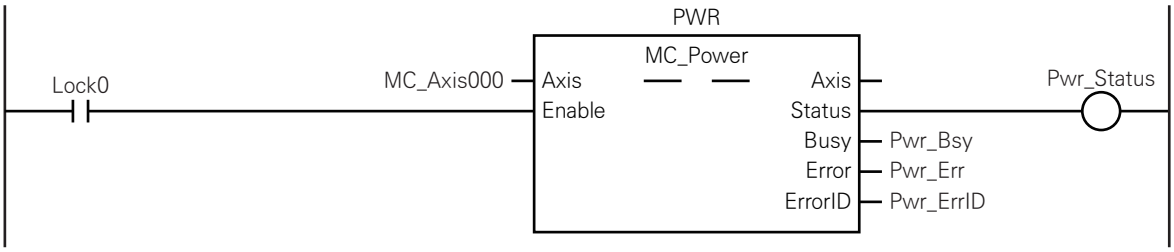
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Status.ErrorStop	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생하여 축이 감속 정지 중이거나 정지 중인 경우에 TRUE가 됩니다.
MC_Axis000.Details.Idle	BOOL	FALSE	인포지션 대기가 아닐 때 축 0의 지령 현재 속도가 0인 경우 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON 합니다.
ResetON	BOOL	FALSE	이상을 해제하기 위한 외부 버튼의 상태를 나타냅니다.

래더도 프로그램

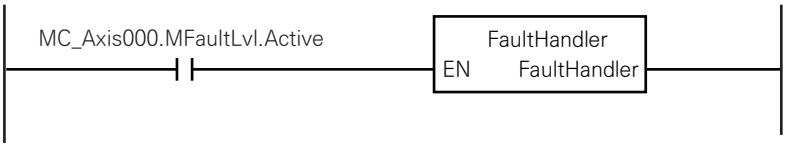
접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태인지 확인합니다.



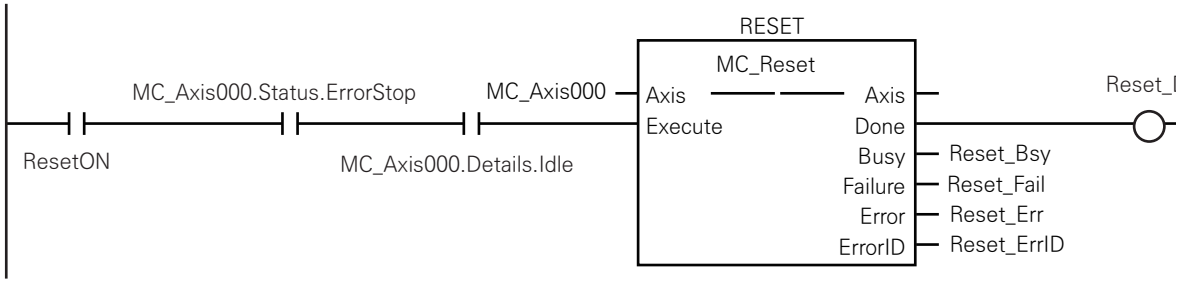
서보 드라이버가 서보 준비 상태이면 축 0을 서버 ON합니다.



축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



외부 버튼을 ON하여 ResetON이 TRUE가 되고 지령 현재 속도가 제로이면 이상을 해제합니다.



ST 프로그램

```
//StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태인지 확인한 후 축 0을 서보 ON합니다.
//서보 준비 상태가 되지 않은 경우에는 축 0을 서보 OFF로 합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr_En:=TRUE;
ELSE
   Pwr_En:=FALSE;
END_IF;


// 축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


// 외부 버튼을 ON하여 ResetON이 TRUE가 되고 지령 현재 속도가 제로이면 이상을 해제합니다.
IF (ResetOn=TRUE)
 AND (MC_Axis000.Status.ErrorStop=TRUE)
 AND (MC_Axis000.Details.Idle=TRUE) THEN
   Reset_Ex := TRUE;   //경도 폴트 해제
END_IF;

// MC_Power
PWR(
   Axis := MC_Axis000,
   Enable := Pwr_En,
   Status => Pwr_Status,
   Busy => Pwr_Bsy,
   Error => Pwr_Err,
   ErrorID => Pwr_ErrID
);

// MC_Reset
RESET(
   Axis := MC_Axis000,
   Execute := Reset_Ex,
   Done => Reset_D,
   Busy => Reset_Bsy,
   Failure => Reset_Fai,
   Error => Reset_Err,
   ErrorID => Reset_ErrID
);
```

10-2-4 이상 감시와 이상 해제(다축 협조 동작)

축 변수에 있는「축 경도 폴트 발생 중」인 상태 및 축 그룹 변수에 있는「축 그룹 경도 폴트 발생 중」인 상태를 통해 이상 여부를 감시합니다.
경도 폴트 레벨의 이상이 발생한 경우에는 MC_GroupDisable(축 그룹 무효) 명령의 입력 변수「Execute (기동)」가 TRUE가 되어 축 그룹을 무효화합니다.
외부 버튼이 ON이고 축 그룹의 지령 현재 속도가 제로이면 MC_GroupReset(축 그룹 에러 리셋) 명령으로 이상을 해제합니다.

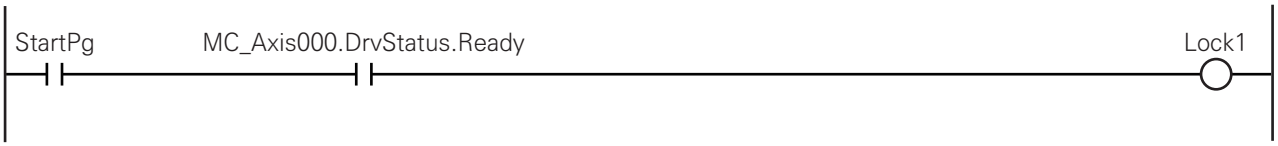
래더도 프로그램의 샘플과 ST 프로그램의 샘플을 기재합니다.

프로그램에서 사용하는 주요 변수

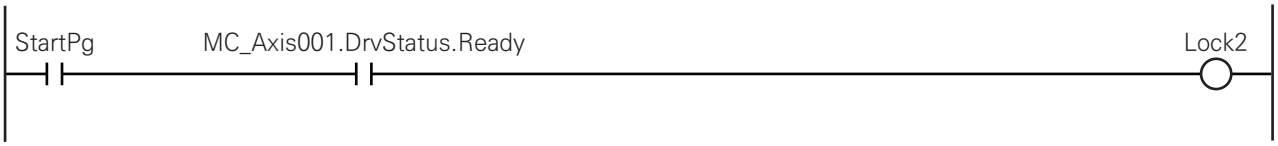
명칭	데이터형	초기값	코멘트
MC_Group000	_sGROUP_REF	—	축 그룹 0의 축 그룹 변수입니다.
MC_Group000.MFaultLvl.Active	BOOL	FALSE	축 그룹 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Group000.Details.Idle	BOOL	FALSE	인포지션 대기가 아닐 때 축 그룹 0의 지령 보간 속도가 0인 경우 TRUE가 됩니다.
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 0이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	—	축 1의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 1에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr1_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON 합니다.
ResetON	BOOL	FALSE	이상을 해제하기 위한 외부 버튼의 상태를 나타냅니다.

래더도 프로그램

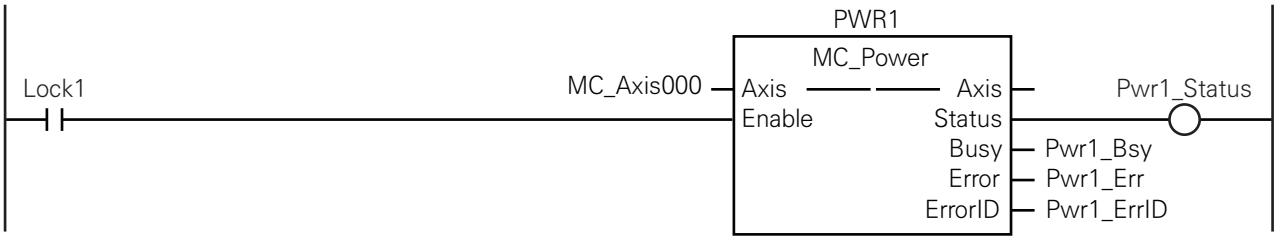
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



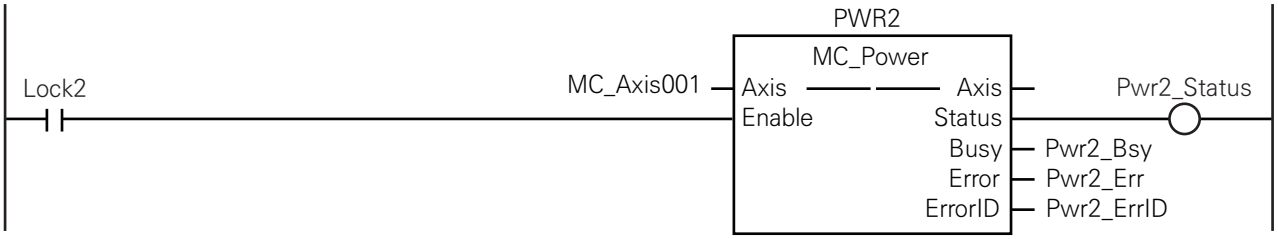
접점 StartPg가 TRUE이고 축 1의 서보 드라이버가 서보 준비 상태인지 확인합니다.



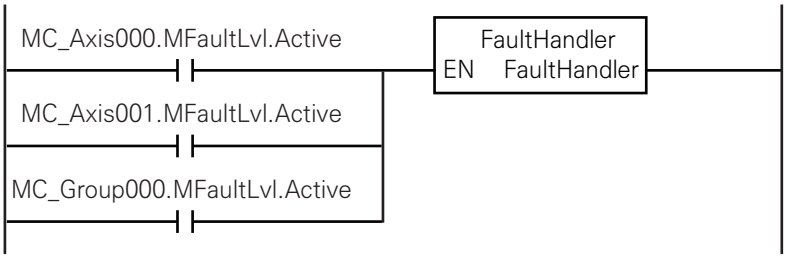
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



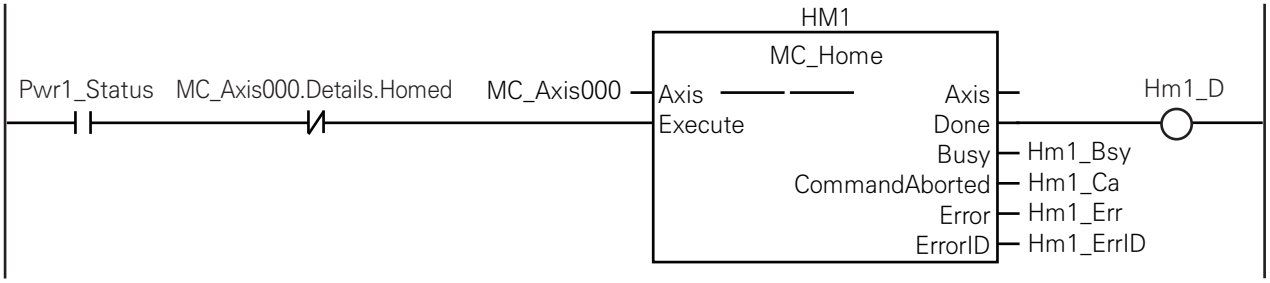
축 1의 서보 드라이버가 서보 준비 상태이면 축 1을 서버 ON합니다.



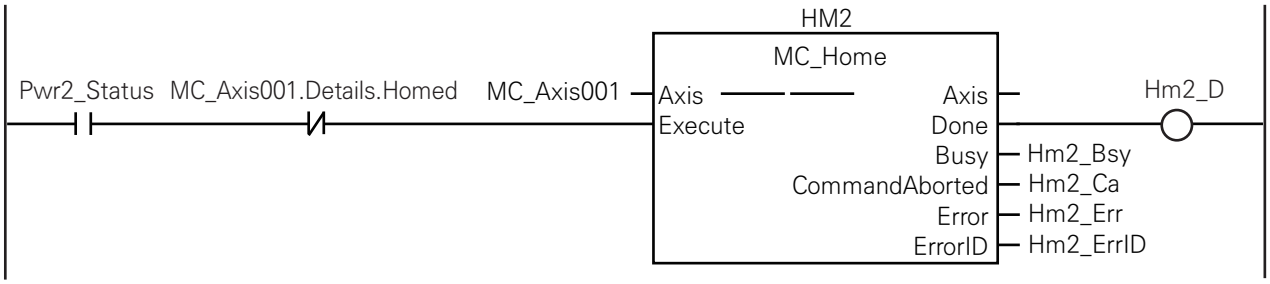
축 그룹 혹은 구성 축에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



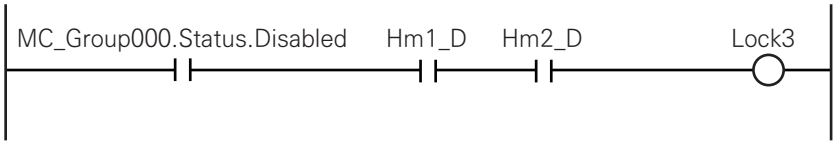
축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



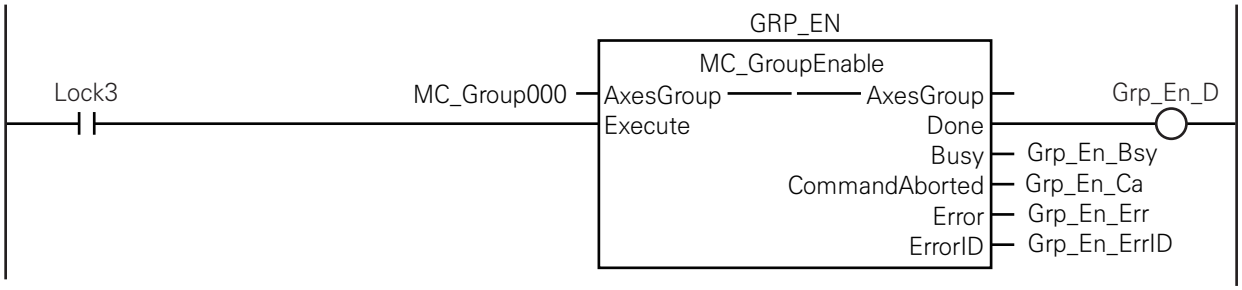
축 10이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



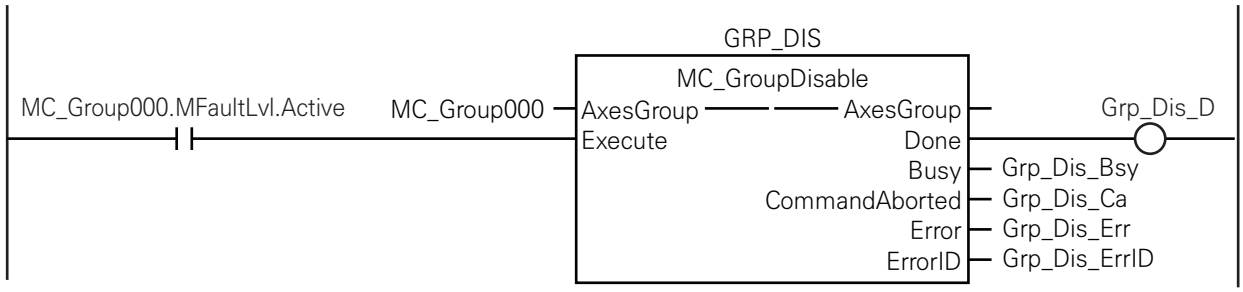
축 그룹의 무효 상태 및 축 0과 축 1의 원점 확정을 확인합니다.



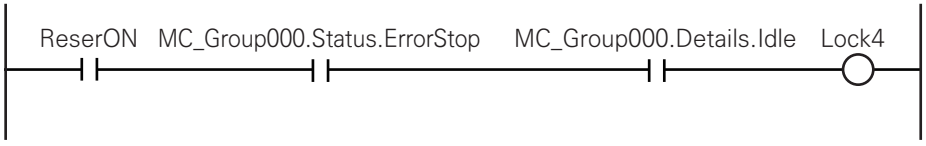
축 0과 축 1이 원점 확정되어 있으면 축 그룹을 유효화합니다.



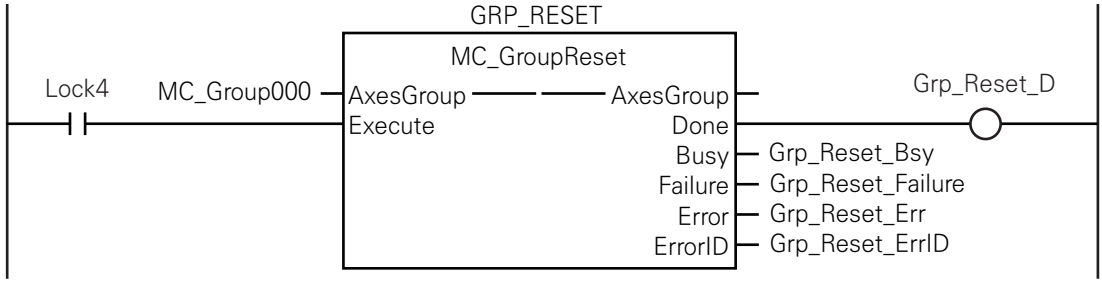
축 그룹에서 경도 폴트 레벨의 이상이 발생한 경우 축 그룹을 무효화합니다.



외부 버튼을 ON하여 Reset ON의 상태와 축 그룹의 정지를 확인합니다.



Reset ON이 TRUE이고 축 그룹이 정지해 있으면 이상을 해제합니다



ST 프로그램

```
//StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태인지 확인한 후 축 0을 서보 ON합니다.
//서보 준비 상태가 되지 않은 경우에는 축 0을 서보 OFF로 합니다.
IF (StartPg =TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr1_En :=TRUE;   //축 0을 서보 ON합니다.
ELSE
   Pwr1_En :=FALSE;   //축 0을 서보 OFF합니다.
END_IF;


//StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태인지 확인한 후 축 1을 서보 ON합니다.
//서보 준비 상태가 되지 않은 경우에는 축 1을 서보 OFF로 합니다.
IF (StartPg =TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
   Pwr2_En :=TRUE;   //축 1을 서보 ON합니다.
ELSE
   Pwr2_En :=FALSE;   //축 1을 서보 OFF합니다.
END_IF;


//축 그룹 혹은 구성 축에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
IF (MC_Axis000.MFaultLvl.Active=TRUE)
 OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (MC_Group000.MFaultLvl.Active=TRUE) THEN
   FaultHandler();   //이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
END_IF;


//축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex := TRUE;
END_IF;


//축 1이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
   Hm2_Ex := TRUE;
END_IF;


축 그룹이 무효이고 축 0과 축 1이 원점 확정되어 있으면 축 그룹을 유효화합니다.
IF (MC_Group000.Status.Disabled=TRUE) AND (Hm1_D=TRUE) AND (Hm2_D=TRUE) THEN
   Grp_En_Ex := TRUE;
END_IF;


//축 그룹에서 경도 폴트 레벨의 이상이 발생한 경우 축 그룹을 무효화합니다.
IF MC_Group000.MFaultLvl.Active=TRUE THEN
   Grp_Dis_Ex :=TRUE;
END_IF;


//외부 버튼을 ON하여 Reset ON이 TRUE가 되고 축 그룹이 정지해 있으면 이상을 해제합니다.
IF (ResetON=TRUE)
```

```
 AND (MC_Group000.Status.ErrorStop=TRUE)
 AND (MC_Group000.Details.Idle=TRUE) THEN
   Grp_Reset_Ex := TRUE;
END_IF;

//MC_Power1
PWR1(
   Axis := MC_Axis000,
   Enable := Pwr1_En,
   Status => Pwr1_Status,
   Busy => Pwr1_Bsy,
   Error => Pwr1_Err,
   ErrorID => Pwr1_ErrID
);

//MC_Power2
PWR2(
   Axis := MC_Axis001,
   Enable := Pwr2_En,
   Status => Pwr2_Status,
   Busy => Pwr2_Bsy,
   Error => Pwr2_Err,
   ErrorID => Pwr2_ErrID
);

// MC_Home1
HM1(
   Axis := MC_Axis000,
   Execute := Hm1_Ex,
   Done => Hm1_D,
   Busy => Hm1_Bsy,
   CommandAborted => Hm1_Ca,
   Error => Hm1_Err,
   ErrorID => Hm1_ErrID
);

// MC_Home2
HM2(
   Axis := MC_Axis001,
   Execute := Hm2_Ex,
   Done => Hm2_D,
   Busy => Hm2_Bsy,
   CommandAborted => Hm2_Ca,
   Error => Hm2_Err,
   ErrorID => Hm2_ErrID
);

//MC_GroupEnable
GRP_EN(
   AxesGroup := MC_Group000,
   Execute := Grp_En_Ex,
   Done => Grp_En_D,
   Busy => Grp_En_Bsy,
   CommandAborted => Grp_En_Ca,
   Error => Grp_En_Err,
   ErrorID => Grp_En_ErrID
);

//MC_GroupDisable
```

```
GRP_DIS(
   AxesGroup := MC_Group000,
   Execute := Grp_Dis_Ex,
   Done => Grp_Dis_D,
   Busy => Grp_Dis_Bsy,
   CommandAborted => Grp_Dis_Ca,
   Error => Grp_Dis_Err,
   ErrorID => Grp_Dis_ErrID
);

//MC_GroupReset
GRP_RESET(
   AxesGroup := MC_Group000,
   Execute := Grp_Reset_Ex,
   Done => Grp_Reset_D,
   Busy => Grp_Reset_Bsy,
   Failure => Grp_Reset_Fai,
   Error => Grp_Reset_Err,
   ErrorID => Grp_Reset_ErrID
);
```

10-2-5 명령 이상 감시 방법

운전 가능(MC_Power) 명령을 기동할 때 이상이 발생한 경우 이후의 처리는 실행하지 않습니다.
본 샘플 프로그램에서는 이후의 처리 허가를 변수「UpgOn」으로 나타냅니다.

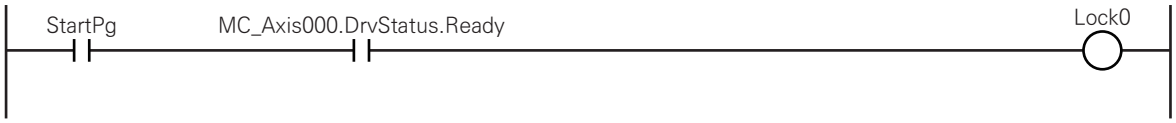
래더도 프로그램의 샘플과 ST 프로그램의 샘플을 기재합니다.

프로그램에서 사용하는 주요 변수

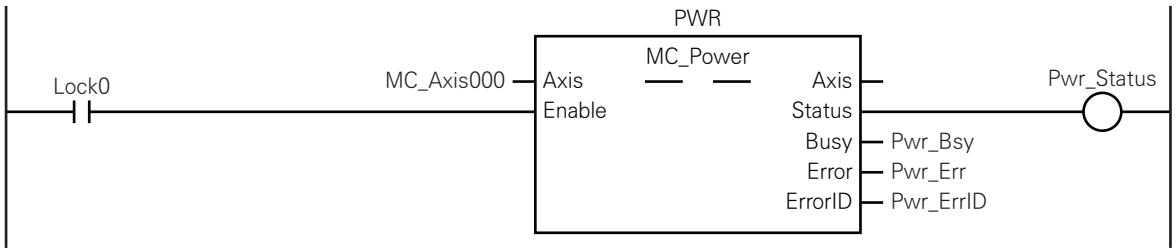
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
Pwr_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Pwr_Bsy	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR의 출력 Busy에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON합니다.
UpgOn	BOOL	FALSE	TRUE일 때 사용자 프로그램 이후의 처리를 실행합니다.

래더도 프로그램

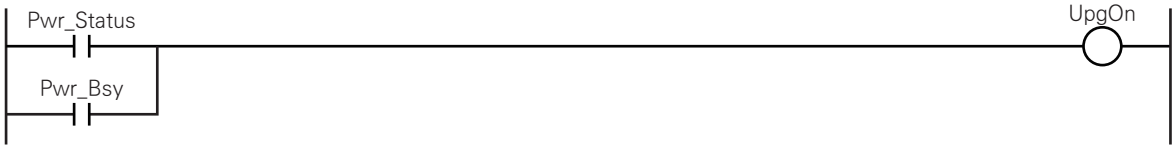
접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태인지 확인합니다.



서보 드라이버가 서보 준비 상태이면 축 0을 서버 ON합니다.



MC_Power에서 이상이 발생하지 않았는지 확인하고 다음 처리를 실행합니다.



ST 프로그램

```
// 접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
//서보 준비 상태가 되지 않은 경우에는 축 0을 서보 OFF로 합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;

IF (Pwr_Status=TRUE) OR (Pwr_Bsy=TRUE) THEN
   UpgOn := TRUE;   //이후의 처리를 실행합니다.
ELSE
   UpgOn := FALSE;   //이후의 처리를 실행하지 않습니다.
END_IF;

// MC_Power
PWR(
   Axis := MC_Axis000,
   Enable := Pwr_En,
   Status => Pwr_Status,
   Busy => Pwr_Bsy,
   Error => Pwr_Err,
   ErrorID => Pwr_ErrID
);
```

10-2-6 이상 해제 감시 방법

경도 폴트 레벨의 이상이 발생 중인 상태에서 외부 버튼을 ON하면 축 에러 리셋(MC_Reset) 명령을 기동합니다. 축 에러 리셋(MC_Reset) 명령의 출력「Done」이 TRUE가 될 때까지 다음 정상 처리를 실행하지 않습니다.
출력「Failure」가 TRUE가 되면 축이 감속 정지 중이거나 MC 공통 이상이 발생하는 상태입니다. 축 에러 리셋(MC_Reset) 명령의「Failure」가 TRUE가 된 요인을 취득합니다.

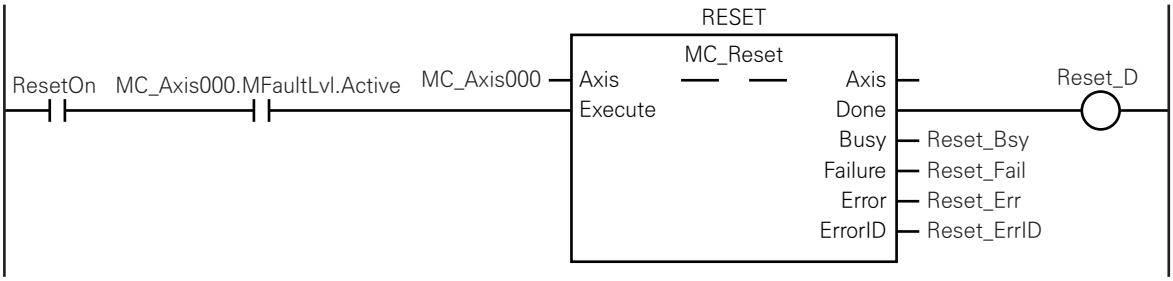
래더도 프로그램의 샘플과 ST 프로그램의 샘플을 기재합니다.

프로그램에서 사용하는 주요 변수

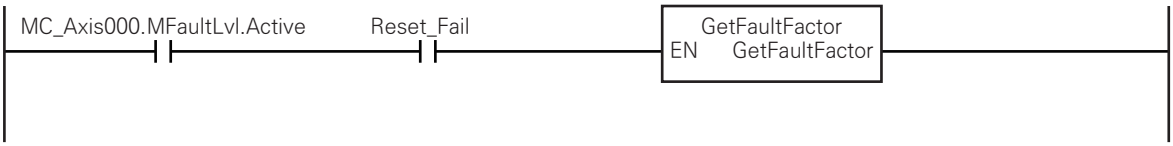
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
ResetON	BOOL	FALSE	이상을 해제하기 위한 외부 버튼의 상태를 나타냅니다.
GetFaultFactor	—	—	이상 요인의 취득 처리입니다. 장비에 맞춰 프로그래밍해 주십시오.
RegularProcess	—	—	정상 처리입니다. 장비에 맞춰 프로그래밍해 주십시오.

래더도 프로그램

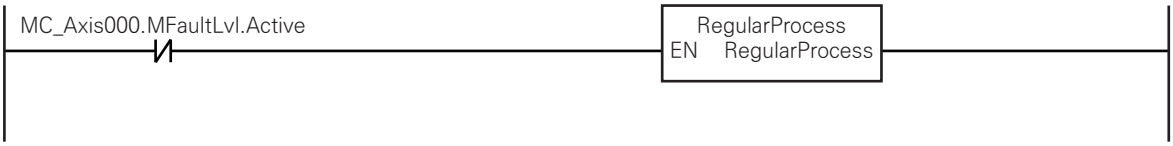
경도 폴트 레벨의 이상이 발생한 상태에서 외부 버튼(ResetOn)을 ON하면
축 에러 리셋(MC_Reset) 명령을 실행합니다.



축 에러 리셋(MC_Reset) 명령의 출력 변수「Failure」가 TRUE이면 장비의 이상 요인 취득 처리
(GetFaultFactor)를 실행합니다. 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



경도 폴트 레벨의 이상이 발생하지 않았거나 경도 폴트 레벨의 이상이 해제된 경우에는 장비의 정상 처리
(RegularProcess)를 실행합니다.
처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



ST 프로그램

```
//경도 폴트 레벨의 이상이 발생한 상태에서 외부 버튼(ResetOn)을 ON하면 축 에러 리셋(MC_Reset) 명령을 실행합니다.
IF (MC_Axis000.MFaultLvl.Active=TRUE) AND (ResetOn=TRUE) THEN
   Reset_Ex := TRUE;   //경도 폴트 해제
ELSE
   Reset_Ex := FALSE;
END_IF;


//축 에러 리셋(MC_Reset) 명령의 출력 변수「Failure」가 TRUE이면 이상 요인의 취득 처리(GetFaultFactor)를 실행합니다.
//처리 내용은 장비에 맞춰 프로그래밍해 주십시오
IF (MC_Axis000.MFaultLvl.Active=TRUE) AND (Reset_Fail=TRUE) THEN
   GetFaultFactor();
END_IF;


//경도 폴트 레벨의 이상이 발생하지 않았거나 경도 폴트 레벨의 이상이 해제된 경우에는 장비의 정상 처리(RegularProcess)를 실행합니다.
//처리 내용은 장비에 맞춰 프로그래밍해 주십시오
IF MC_Axis000.MFaultLvl.Active=FALSE THEN
   RegularProcess();
END_IF;

// MC_Reset
RESET(
   Axis := MC_Axis000,
   Execute := Reset_Ex,
   Done => Reset_D,
   Busy => Reset_Bsy,
   Failure => Reset_Fail,
   Error => Reset_Err,
   ErrorID => Reset_ErrID
);
```

10-2-7 단축 동작 중의 축 정지

절대값 위치 결정(MC_MoveAbsolute) 명령을 실행하는 도중에 외부 버튼을 ON하면 강제 정지(MC_Stop) 명령을 실행하여 감속 정지합니다. 경도 폴트 레벨의 이상이 발생 중인 경우에는 강제 정지(MC_Stop) 명령의 출력「CommandAborted」가 TRUE가 됩니다.
이때 즉시 정지(MC_ImmediateStop)를 실행하여 즉시 정지합니다.
또한, 어떠한 이유로 강제 정지(MC_Stop) 명령의 출력「Error」가 TRUE가 된 경우에도 즉시 정지(MC_ImmediateStop) 명령을 실행하여 즉시 정지합니다. 즉시 정지(MC_ImmediateStop) 명령을 실행하면 축은 에러 감속 정지 중이 됩니다.

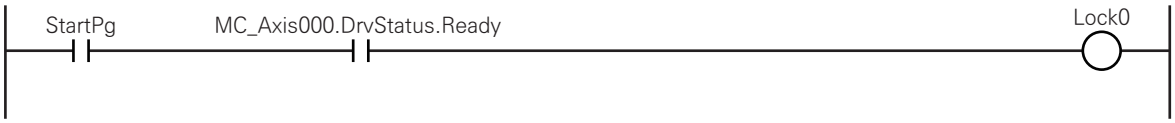
래더도 프로그램의 샘플과 ST 프로그램의 샘플을 기재합니다.

프로그램에서 사용하는 주요 변수

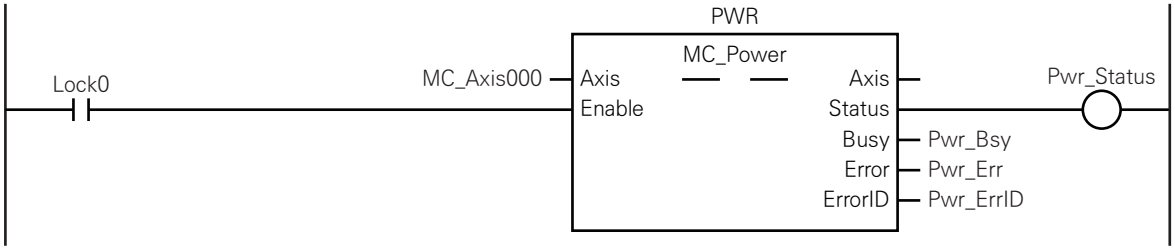
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 0이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Stp_Ca	BOOL	FALSE	MC_Stop의 인스턴스 STP의 출력 변수 CommandAborted에 할당한 변수입니다.
Stp_Err	BOOL	FALSE	MC_Stop의 인스턴스 STP의 출력 변수 Error에 할당한 변수입니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON 합니다.
StopOn	BOOL	FALSE	강제 정지를 위한 외부 버튼의 상태입니다. TRUE일 때 MC_ S top으로 강제 정지합니다.

래더도 프로그램

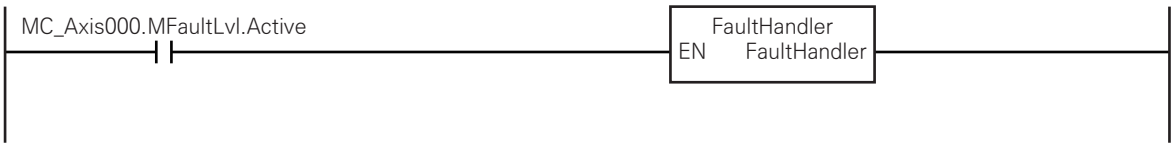
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



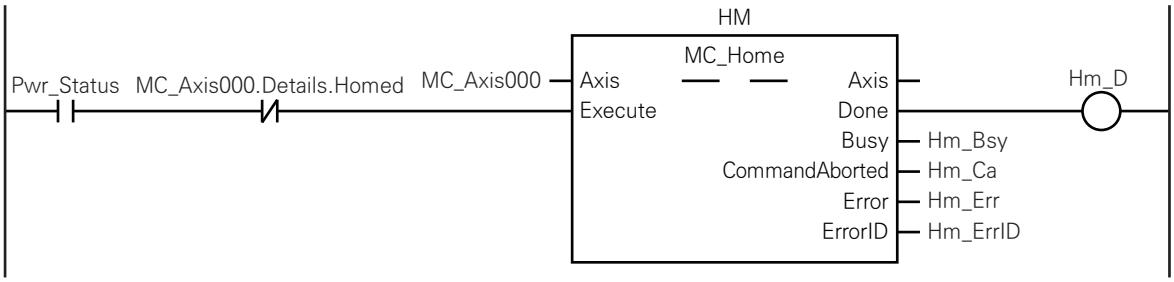
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



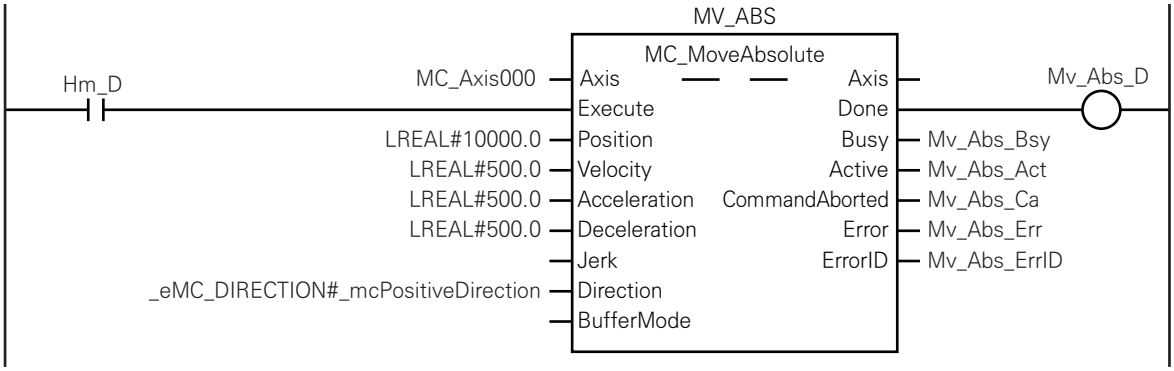
축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



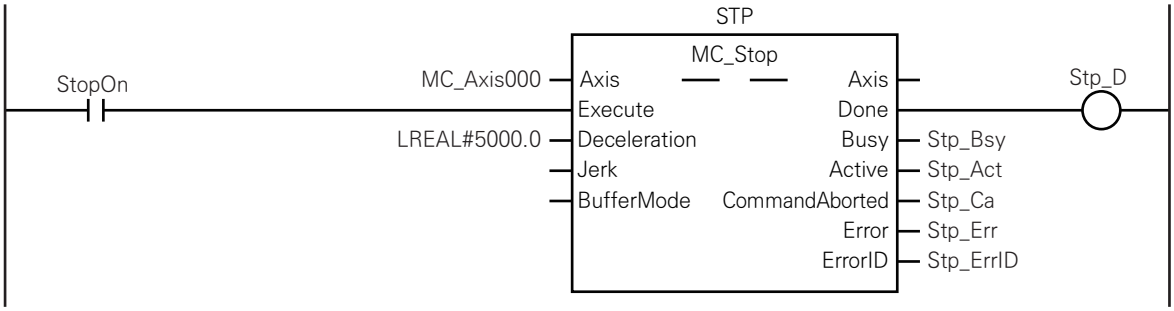
축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



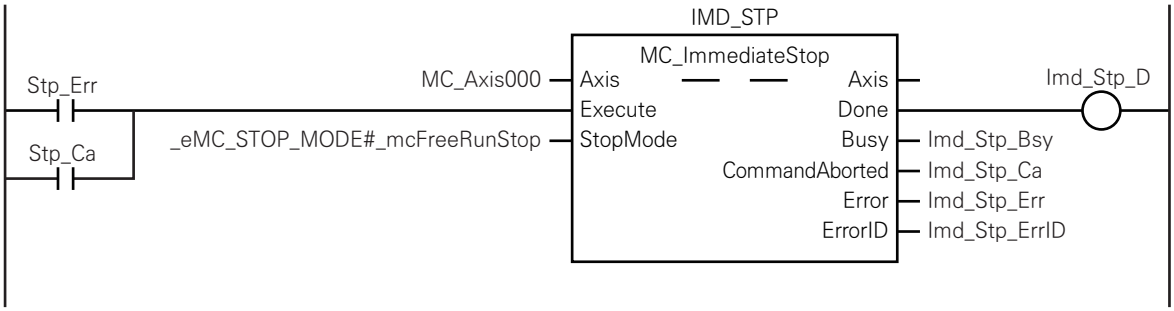
축 0의 원점 복귀가 완료되면 절대값 위치 결정(MC_MoveAbsolute) 명령을 실행합니다.



StopOn이 TRUE일 때 강제 정지(MC_Stop) 명령을 실행합니다.



강제 정지(MC_Stop) 명령의 출력 「Error」 또는 「CommandAborted」가 TRUE일 때
즉시 정지(MC_ImmediateStop) 명령을 실행하여 즉시 정지합니다.



ST 프로그램

```
//절대값 위치 결정과 강제 정지의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag=FALSE THEN
   //절대값 위치 결정(MC_MoveAbsolute)의 입력 파라메터에 값을 설정합니다.
   Mv_Abs_Pos := LREAL#10000.0;
   Mv_Abs_Vel := LREAL#500.0;
   Mv_Abs_Acc := LREAL#500.0;
   Mv_Abs_Dec := LREAL#500.0;
   Mv_Abs_Dir := _eMC_DIRECTION#_mcPositiveDirection;

   //강제 정지(MC_Stop)의 입력 파라메터에 값을 설정합니다.
   Stp_Dec:=LREAL#5000.0;

   //즉시 정지(MC_ImmediateStop)의 입력 파라메터에 값을 설정합니다.
   Imd_Stp_SM :=_eMC_STOP_MODE#_mcFreeRunStop;

   //입력 파라메터의 설정 완료 플래그를 TRUE로 합니다.
   InitFlag := TRUE;
END_IF;


//접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;


//축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


//축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm_Ex :=TRUE;
END_IF;


//원점 복귀가 완료되면 절대값 위치 결정을 실행합니다.
IF Hm_D=TRUE THEN
   Mv_Abs_Ex := TRUE;
END_IF;


//StopOn이 TRUE가 되면 강제 정지를 실행합니다.
IF StopOn=TRUE THEN
   Stp_Ex :=TRUE;
END_IF;


//강제 정지(MC_Stop)의 출력「Error」또는「CommandAborted」가 TRUE일 때 즉시 정지(MC_ImmediateStop)를
실행하여 즉시 정지합니다.
IF (Stp_Err=TRUE) OR (Stp_Ca=TRUE) THEN
```

```
    Imd_Stp_Ex :=TRUE;
END_IF;

//MC_Power
PWR(
    Axis := MC_Axis000,
    Enable := Pwr_En,
    Status => Pwr_Status,
    Busy => Pwr_Bsy,
    Error => Pwr_Err,
    ErrorID => Pwr_ErrID
);

//MC_Home
HM(
    Axis := MC_Axis000,
    Execute := Hm_Ex,
    Done => Hm_D,
    Busy => Hm_Bsy,
    CommandAborted => Hm_Ca,
    Error => Hm_Err,
    ErrorID => Hm_ErrID
);

//MC_MoveAbsolute
MV_ABS(
    Axis := MC_Axis000,
    Execute := Mv_Abs_Ex,
    Position := Mv_Abs_Pos,
    Velocity := Mv_Abs_Vel,
    Acceleration := Mv_Abs_Acc,
    Deceleration := Mv_Abs_Dec,
    Direction := Mv_Abs_Dir,
    Done => Mv_Abs_D,
    Busy => Mv_Abs_Bsy,
    Active => Mv_Abs_Act,
    CommandAborted => Mv_Abs_Ca,
    Error => Mv_Abs_Err,
    ErrorID => Mv_Abs_ErrID
);

//MC_Stop
STP(
    Axis := MC_Axis000,
    Execute := Stp_Ex,
    Deceleration := Stp_Dec,
    Done => Stp_D,
    Busy => Stp_Bsy,
    Active => Stp_Act,
    CommandAborted => Stp_Ca,
    Error => Stp_Err,
    ErrorID => Stp_ErrID
);

//MC_ImmediateStop
IMD_STP(
    Axis := MC_Axis000,
    Execute := Imd_Stp_Ex,
    StopMode := Imd_Stp_SM,
```

```
    Done => Imd_Stp_D,
    Busy => Imd_Stp_Bsy,
    CommandAborted => Imd_Stp_Ca,
    Error => Imd_Stp_Err,
    ErrorID => Imd_Stp_ErrID
);
```

10-2-8 다축 협조 동작 중의 축 그룹 정지

절대값 직선 보간(MC_MoveLinearAbsolute) 명령을 실행 중일 때 외부 버튼을 ON하면 강제 정지(MC_GroupStop) 명령을 실행하여 감속 정지합니다. 경도 폴트 레벨의 이상이 발생 중일 때는 축 그룹 강제 정지(MC_GroupStop)의 출력 「CommandAborted」가 TRUE가 됩니다. 이때 축 그룹 즉시 정지(MC_GroupImmediateStop)를 실행하여 즉시 정지합니다.
또한, 어떠한 이유로 축 그룹 강제 정지(MC_GroupStop)의 출력 「Error」가 TRUE가 된 경우에도 축 그룹 즉시 정지(MC_GroupImmediateStop) 명령을 실행하여 즉시 정지합니다. 축 그룹 즉시 정지(MC_GroupImmediateStop)를 실행하면 축 그룹은 에러 감속 정지 중이 됩니다.

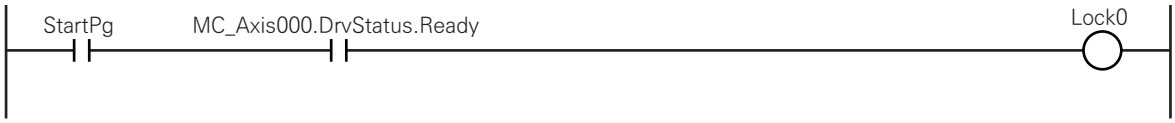
프로그램에서 사용하는 주요 변수

명칭	데이터형	초기값	코멘트
MC_Group000	_sGROUP_REF	—	축 그룹 0의 축 그룹 변수입니다.
MC_Group000.Status.Disabled	BOOL	FALSE	축 그룹 0이 무효일 때 TRUE가 됩니다.
MC_Group000.MFaultLvl.Active	BOOL	FALSE	축 그룹 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 0이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	—	축 1의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 1에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr1_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Grp_Stp_Ca	BOOL	FALSE	MC_GroupStop의 인스턴스 GRP_EN의 출력 변수 CommandAborted에 할당한 변수입니다.
Grp_Stp_Err	BOOL	FALSE	MC_GroupStop의 인스턴스 GRP_EN의 출력 변수 Error에 할당한 변수입니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON 합니다.
StopOn	BOOL	FALSE	강제 정지를 위한 외부 버튼의 상태입니다. TRUE 일 때 MC_Group S top으로 강제 정지합니다.
InitFlag	BOOL	FALSE	절대값 직선 보간 명령과 축 그룹 강제 정지의 입력 파라메터 설정 완료를 나타내는 변수입니다.
Grp_En_Ex	BOOL	FALSE	MC_GroupEnable의 인스턴스 GRP_EN을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.

명칭	데이터형	초기값	코멘트
Mv_Lin_Abs_Ex	BOOL	FALSE	MC_MoveLinear 인스턴스 MV_LIN_ABS를 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.
Grp_Stp_Ex	BOOL	FALSE	MC_GroupStop의 인스턴스 GRP_STP를 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.
Grp_Imd_Stp_Ex	BOOL	FALSE	MC_GroupImmediateStop의 인스턴스 GRP_IMD_STP를 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.

래더도 프로그램

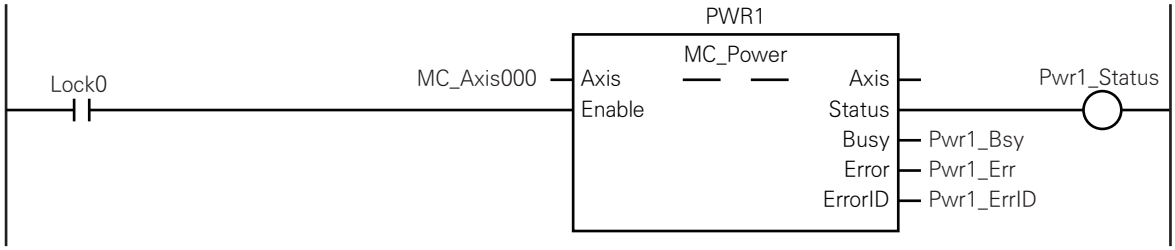
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



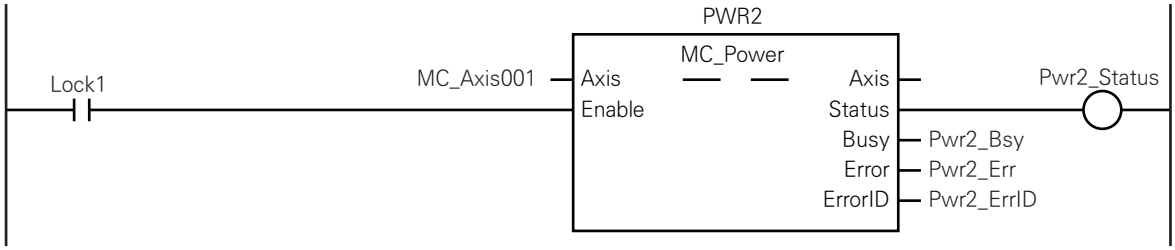
접점 StartPg가 TRUE이고 축 1의 서보 드라이버가 서보 준비 상태인지 확인합니다.



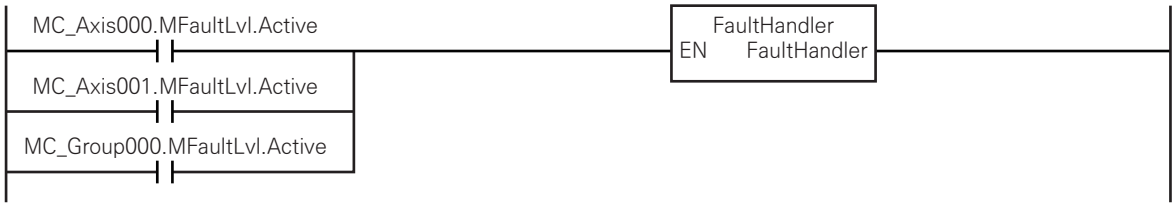
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



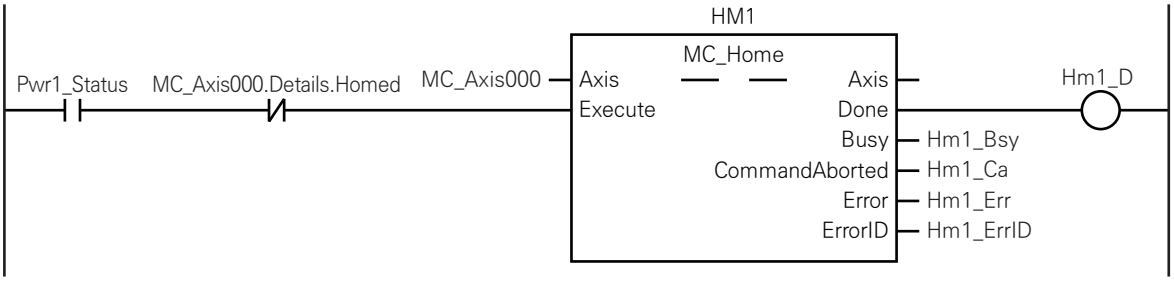
축 1의 서보 드라이버가 서보 준비 상태이면 축 1을 서버 ON합니다.



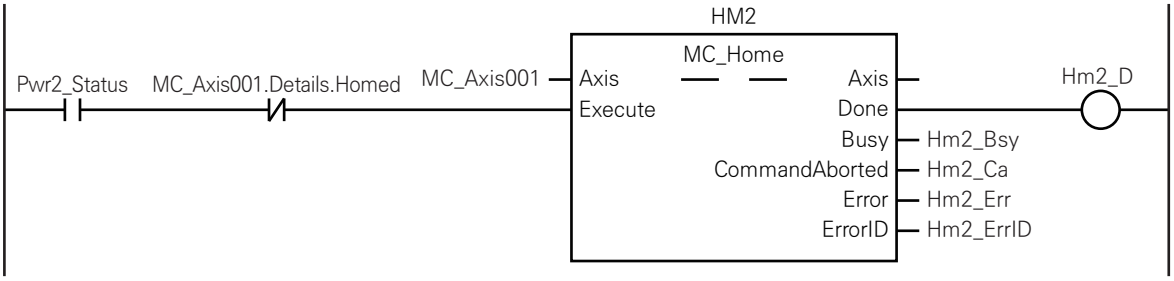
축 그룹 혹은 구성 축에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다. 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



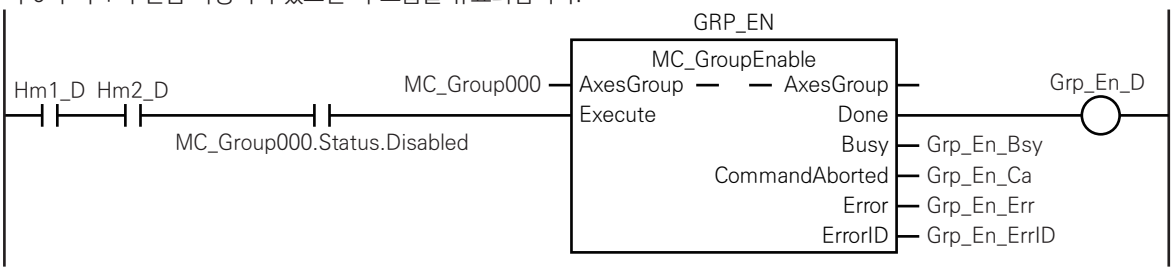
축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



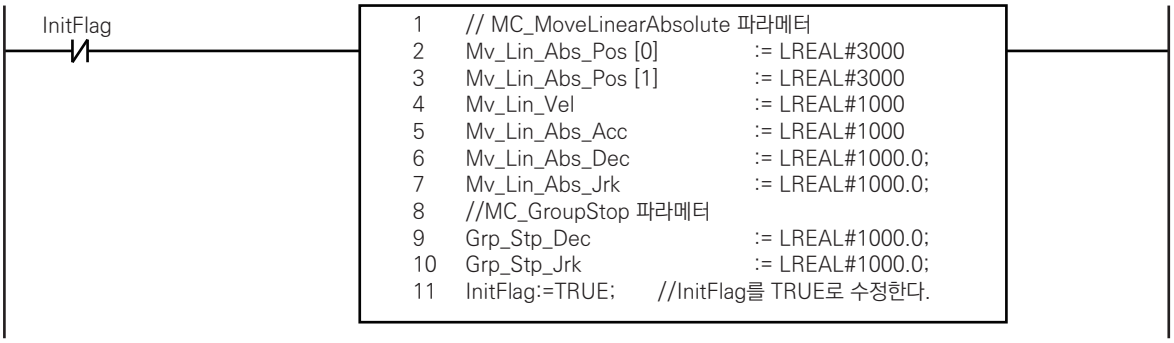
축 1이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



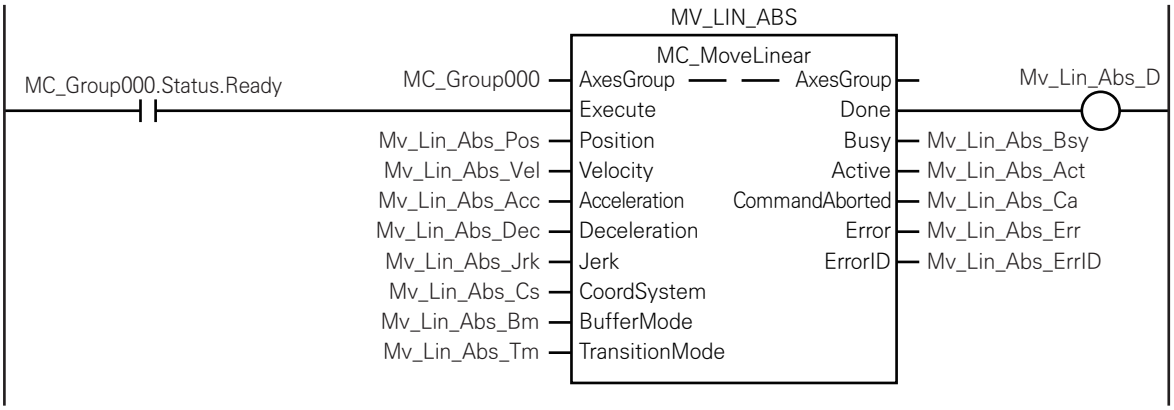
축 0과 축 1이 원점 확정되어 있으면 축 그룹을 유효화합니다.



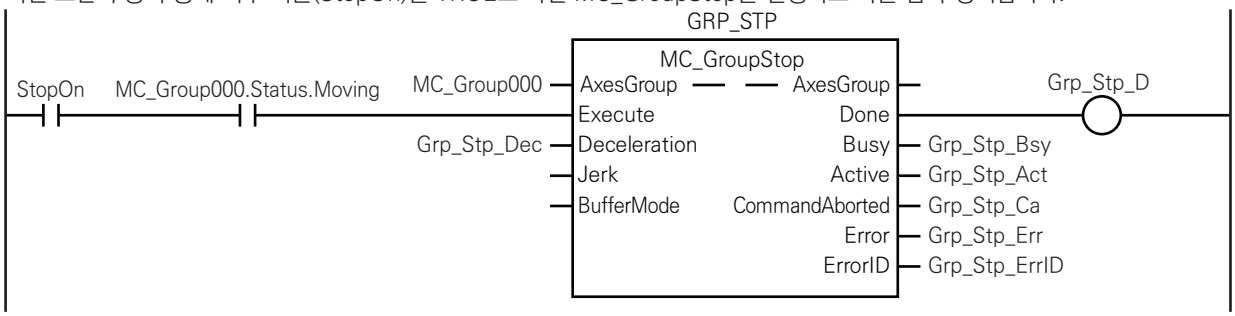
절대값 직선 보간과 축 그룹 강제 정지 입력 파라메터를 설정합니다.



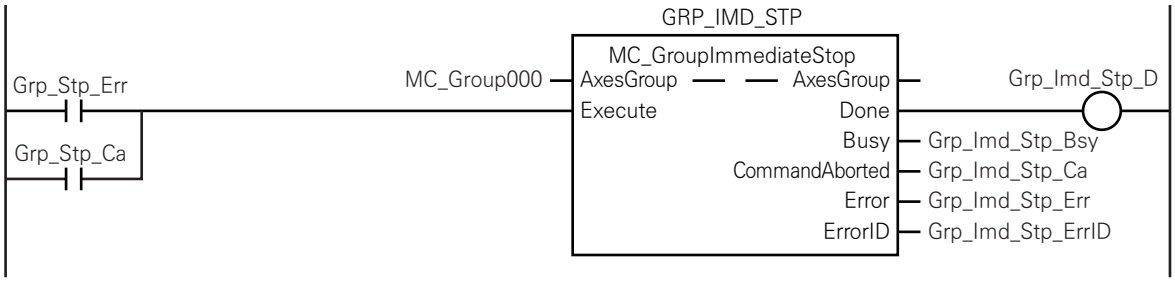
축 그룹이 유효해졌는지 확인하고 직선 보간을 실행합니다.



직선 보간이 동작 중에 외부 버튼(StopOn)을 TRUE로 하면 MC_GroupStop을 실행하고 축을 감속 정지합니다.



MC_GroupStop Error 혹은 CommandAborted가 TRUE일 때 MC_GroupImmediateStop을 실행하고 즉시 정지합니다.



ST 프로그램

```
//절대값 직선 보간과 축 그룹 강제 정지의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag=FALSE THEN

  //절대값 직선 보간(MC_MoveLinearAbsolute)의 입력 파라메터에 값을 설정합니다.
  Mv_Lin_Abs_Pos[0] := LREAL#3000.0;
  Mv_Lin_Abs_Pos[1] := LREAL#3000.0;
  Mv_Lin_Abs_Vel := LREAL#1000.0;
  Mv_Lin_Abs_Acc := LREAL#1000.0;
  Mv_Lin_Abs_Dec := LREAL#1000.0;
  Mv_Lin_Abs_Jrk := LREAL#1000.0;

  //축 그룹 강제 정지(MC_GroupStop)의 입력 파라메터에 값을 설정합니다.
  Grp_Stp_Dec := LREAL#1000.0;
  Grp_Stp_Jrk := LREAL#1000.0;

  //입력 파라메터의 설정 완료 플래그를 TRUE로 합니다.
  InitFlag := TRUE;
END_IF;


//StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
//서보 준비 상태가 되지 않은 경우에는 축 0을 서보 OFF로 합니다.
IF (StartPg =TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
  Pwr1_En :=TRUE;   //축 0을 서보 ON합니다.
ELSE
  Pwr1_En :=FALSE;   //축 0을 서보 OFF합니다.
END_IF;


//StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 1을 서보 ON합니다.
//서보 준비 상태가 되지 않은 경우에는 축 1을 서보 OFF로 합니다.
IF (StartPg =TRUE) AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
  Pwr2_En :=TRUE;   //축 1을 서보 ON합니다.
```

```
ELSE
   Pwr2_En :=FALSE;   //축 1을 서보 OFF합니다.
END_IF;


//경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
//이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE)
 OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (MC_Group000.MFaultLvl.Active=TRUE) THEN
   FaultHandler();
END_IF;


//축 0이 서보 ON 상태에서 원점 미확정인 경우 축 0의 원점 복귀를 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex :=TRUE;
END_IF;


//축 1이 서보 ON 상태에서 원점 미확정인 경우 축 1의 원점 복귀를 실행합니다.
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
   Hm2_Ex :=TRUE;
END_IF;


//축 0과 축 1이 원점 확정했을 때 축 그룹이 무효이면 축 그룹을 유효화합니다.
IF (MC_Group000.Status.Disabled=TRUE) AND (Hm1_D=TRUE) AND (Hm2_D=TRUE) THEN
   Grp_En_Ex := TRUE;
END_IF;


//축 그룹이 유효일 때 절대값 직선 보간을 실행합니다.
IF MC_Group000.Status.Ready=TRUE THEN
   Mv_Lin_Abs_Ex :=TRUE;
END_IF;


//절대값 직선 보간을 실행하는 도중에 외부 버튼으로 StopOn을 TRUE로 하면 강제 정지를 실행합니다.
IF (MC_Group000.Status.Moving=TRUE) AND (StopOn=TRUE) THEN
   Grp_Stp_Ex := TRUE;
END_IF;


//축 그룹 강제 정지가 CommandAborted 또는 Error가 된 경우 축 그룹을 즉시 정지합니다.
IF (Grp_Stp_Ca=TRUE) OR (Grp_Stp_Err=TRUE) THEN
   Grp_Imd_Stp_Ex :=TRUE;
END_IF;

//MC_Power
PWR1(
   Axis := MC_Axis000,
   Enable := Pwr1_En,
   Status => Pwr1_Status,
   Busy => Pwr1_Bsy,
   Error => Pwr1_Err,
   ErrorID => Pwr1_ErrID
);
```

```
PWR2(
   Axis := MC_Axis001,
   Enable := Pwr2_En,
   Status => Pwr2_Status,
   Busy => Pwr2_Bsy,
   Error => Pwr2_Err,
   ErrorID => Pwr2_ErrID
);

//MC_Home
HM1(
   Axis := MC_Axis000,
   Execute := Hm1_Ex,
   Done => Hm1_D,
   Busy => Hm1_Bsy,
   CommandAborted => Hm1_Ca,
   Error => Hm1_Err,
   ErrorID => Hm1_ErrID
);

HM2(
   Axis := MC_Axis001,
   Execute := Hm2_Ex,
   Done => Hm2_D,
   Busy => Hm2_Bsy,
   CommandAborted => Hm2_Ca,
   Error => Hm2_Err,
   ErrorID => Hm2_ErrID
);

//MC_GroupEnable
GRP_EN(
   AxesGroup := MC_Group000,
   Execute := Grp_En_Ex,
   Done => Grp_En_D,
   Busy => Grp_En_Bsy,
   CommandAborted => Grp_En_Ca,
   Error => Grp_En_Err,
   ErrorID => Grp_En_ErrID
);

//MC_MoveLinearAbsolute
MV_LIN_ABS(
   AxesGroup := MC_Group000,
   Execute := Mv_Lin_Abs_Ex,
   Position := Mv_Lin_Abs_Pos,
   Velocity := Mv_Lin_Abs_Vel,
   Acceleration := Mv_Lin_Abs_Acc,
   Deceleration := Mv_Lin_Abs_Dec,
   Jerk := Mv_Lin_Abs_Jrk,
   Done => Mv_Lin_Abs_D,
   Busy => Mv_Lin_Abs_Bsy,
   Active => Mv_Lin_Abs_Act,
   CommandAborted => Mv_Lin_Abs_Ca,
   Error => Mv_Lin_Abs_Err,
   ErrorID => Mv_Lin_Abs_ErrID
);

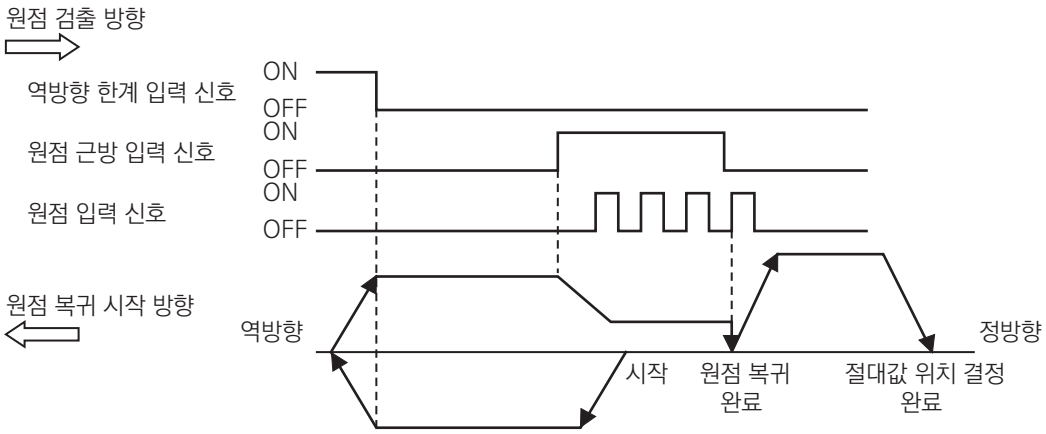
//MC_GroupStop
```

```
GRP_STP(
   AxesGroup := MC_Group000,
   Execute := Grp_Stp_Ex,
   Deceleration := Grp_Stp_Dec,
   Done => Grp_Stp_D,
   Busy => Grp_Stp_Bsy,
   Active => Grp_Stp_Act,
   CommandAborted => Grp_Stp_Ca,
   Error => Grp_Stp_Err,
   ErrorID => Grp_Stp_ErrID
);

//MC_GroupImmediateStop
GRP_IMD_STP(
   AxesGroup := MC_Group000,
   Execute := Grp_Imd_Stp_Ex,
   Done => Grp_Imd_Stp_D,
   Busy => Grp_Imd_Stp_Bsy,
   CommandAborted => Grp_Imd_Stp_Ca,
   Error => Grp_Imd_Stp_Err,
   ErrorID => Grp_Imd_Stp_ErrID
);
```

10-2-9 원점 복귀와 절대값 위치 결정

원점 복귀 시작 위치는 원점 근방 입력 ON으로 합니다. 원점 복귀는「4: 원점 근방 입력 OFF 지정」으로 실행합니다. 원점 복귀가 정상적으로 완료되어 원점이 확정되면 절대값 위치 결정을 실행합니다.



래더도 프로그램의 샘플과 ST 프로그램의 샘플을 기재합니다.

원점 복귀에 관한 축 파라메터 설정

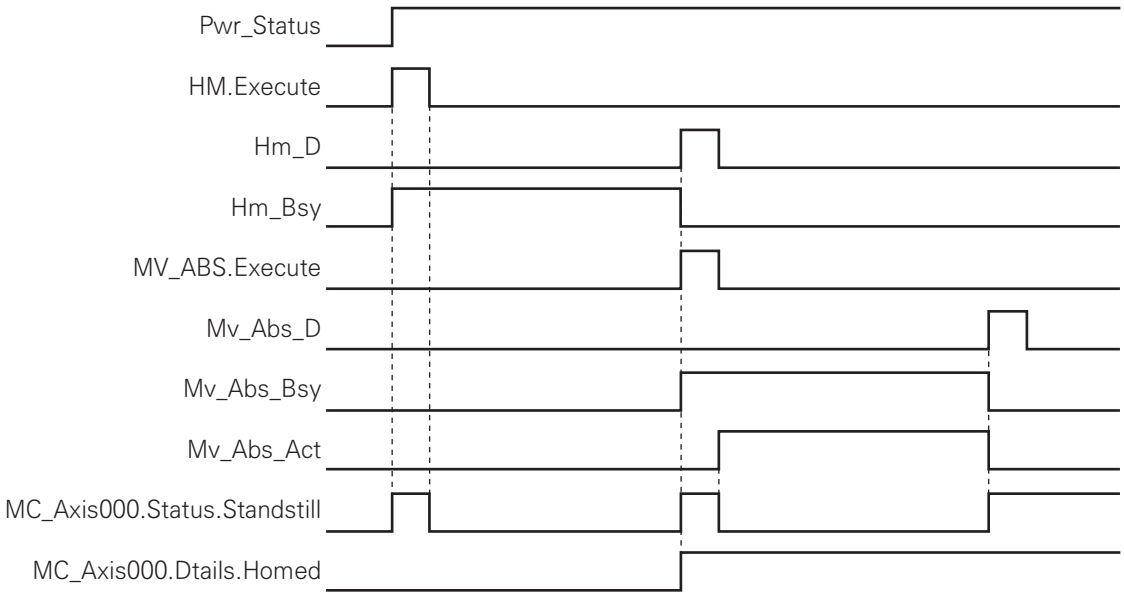
설정 항목	설정 내용	설명
원점 복귀 동작	4: 원점 근방 입력 OFF 지정	원점 근방 입력 OFF 지정으로 원점을 확정합니다.
정방향 한계 입력 시 동작	1: 반전함 · 즉시 정지	정방향 한계 입력을 사용하지 않으므로 초기값인 상태입니다.
역방향 한계 입력 시 동작	2: 반전함 · 감속 정지	역방향 한계 입력이 유효해지면 감속 정지하여 반전합니다.
원점 복귀 시작 방향	2: 역방향	원점 복귀를 기동하면 역방향으로 동작을 시작합니다.
원점 검출 방향	1: 정방향	원점을 검출할 때의 동작 방향은 정방향입니다.

프로그램에서 사용하는 주요 변수

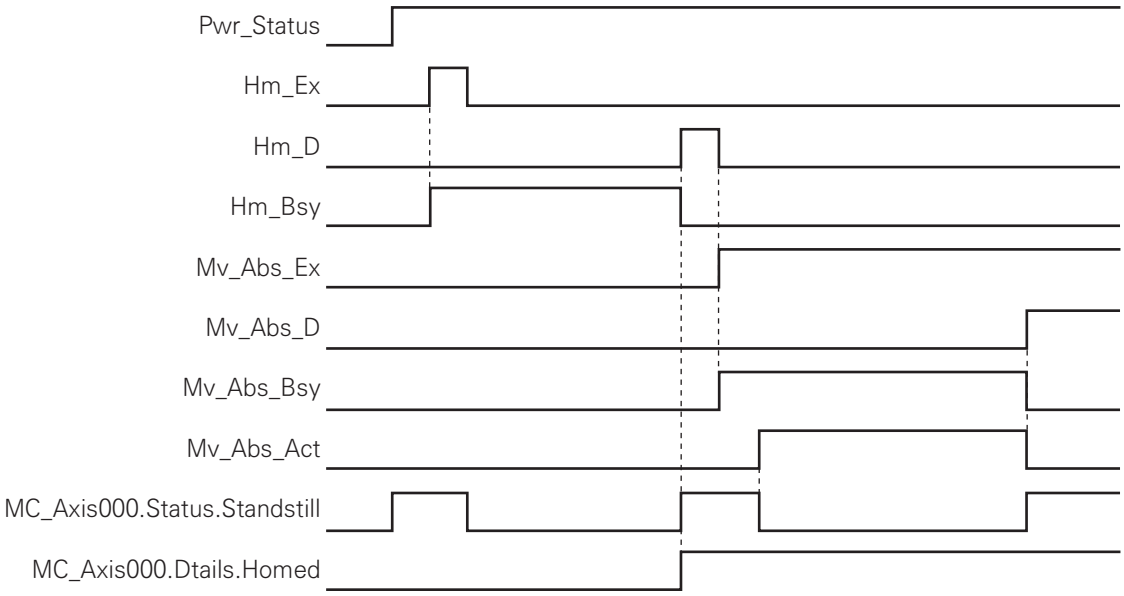
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.Status.StandStill	BOOL	FALSE	축 0이 서보 ON이고 정지 중인 상태에서 TRUE가 됩니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 0이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON 합니다.
Hm_Ex	BOOL	FALSE	MC_Home(원점 복귀) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.
Mv_Abs_Ex	BOOL	FALSE	MC_MoveAbsolute(절대값 위치 결정) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.

타이밍 차트

● **래더도 프로그램의 경우**

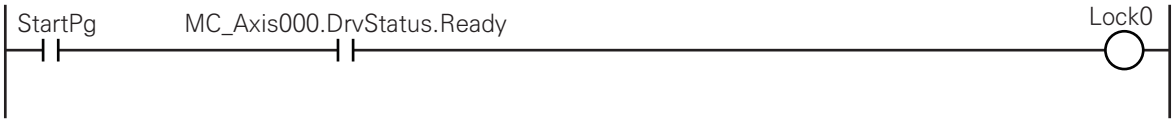


● ST 프로그램의 경우

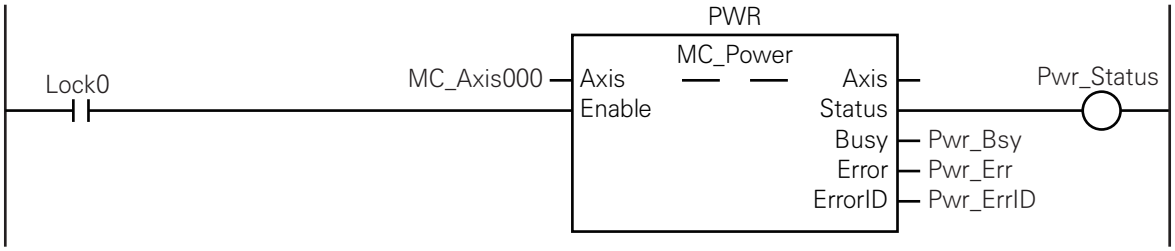


래더도 프로그램

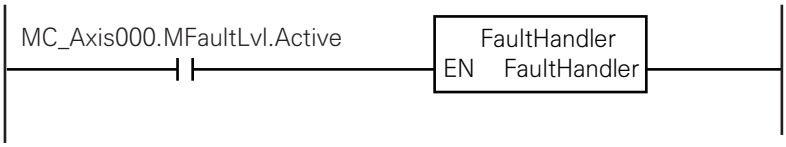
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



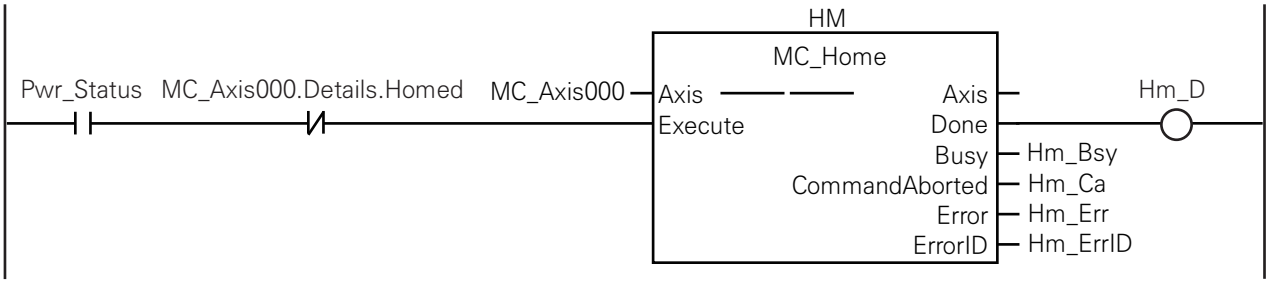
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



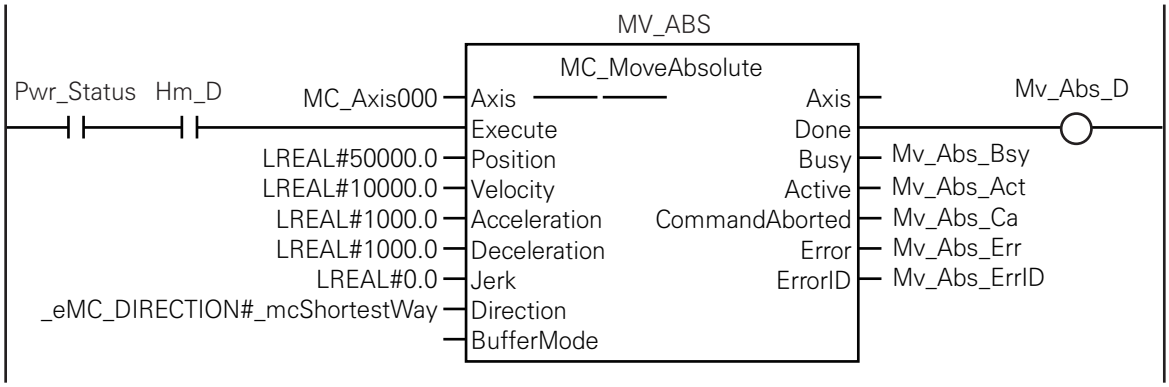
축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



서보 ON 상태에서 원점이 확정되어 있으면 절대값 위치 결정을 실행합니다.



ST 프로그램

```
//절대값 위치 결정의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag=FALSE THEN
   //절대값 위치 결정(MC_MoveAbsolute)의 입력 파라메터에 값을 설정합니다.
   Mv_Abs_Pos := LREAL#50000.0;
   Mv_Abs_Vel := LREAL#10000.0;
   Mv_Abs_Acc := LREAL#1000.0;
   Mv_Abs_Dec := LREAL#1000.0;
   Mv_Abs_Dir := _eMC_DIRECTION#_mcShortestWay;

   //입력 파라메터의 설정 완료 플래그를 유효화합니다.
   InitFlag := TRUE;
END_IF;


// 접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;


// 축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


//축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm_Ex :=TRUE;
ELSE
   Hm_Ex :=FALSE;
END_IF;


//서보 ON 상태에서 원점이 확정되어 있으면 절대값 위치 결정을 실행합니다.
IF (Pwr_Status=TRUE) AND (Hm_D=TRUE) THEN
   Mv_Abs_Ex :=TRUE;
END_IF;
```

```
//MC_Power
PWR(
   Axis := MC_Axis000,
   Enable := Pwr_En,
   Status => Pwr_Status,
   Busy => Pwr_Bsy,
   Error => Pwr_Err,
   ErrorID => Pwr_ErrID
);

//MC_Home
HM(
   Axis := MC_Axis000,
   Execute := Hm_Ex,
   Done => Hm_D,
   Busy => Hm_Bsy,
   CommandAborted => Hm_Ca,
   Error => Hm_Err,
   ErrorID => Hm_ErrID
);

//MC_MoveAbsolute
MV_ABS(
   Axis := MC_Axis000,
   Execute := Mv_Abs_Ex,
   Position := Mv_Abs_Pos,
   Velocity := Mv_Abs_Vel,
   Acceleration := Mv_Abs_Acc,
   Deceleration := Mv_Abs_Dec,
   Direction := Mv_Abs_Dir,
   Done => Mv_Abs_D,
   Busy => Mv_Abs_Bsy,
   Active => Mv_Abs_Act,
   CommandAborted => Mv_Abs_Ca,
   Error => Mv_Abs_Err,
   ErrorID => Mv_Abs_ErrID
);
```

10-2-10 명령 재기동으로 목표 위치 변경하기

목표 위치 1000을 향해 절대값 위치 결정 동작을 하는 도중에 같은 명령 인스턴스를 사용해서 목표 위치를 2000으로 변경합니다.

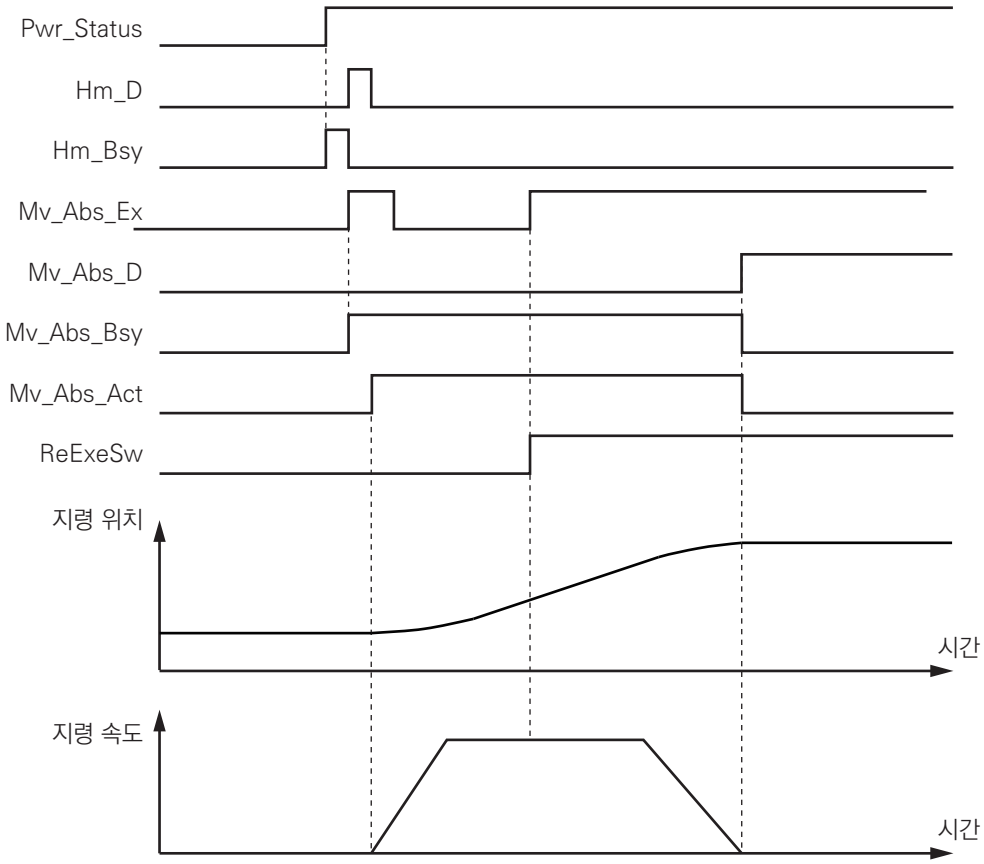
프로그램에서 사용하는 주요 변수

명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 0이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.

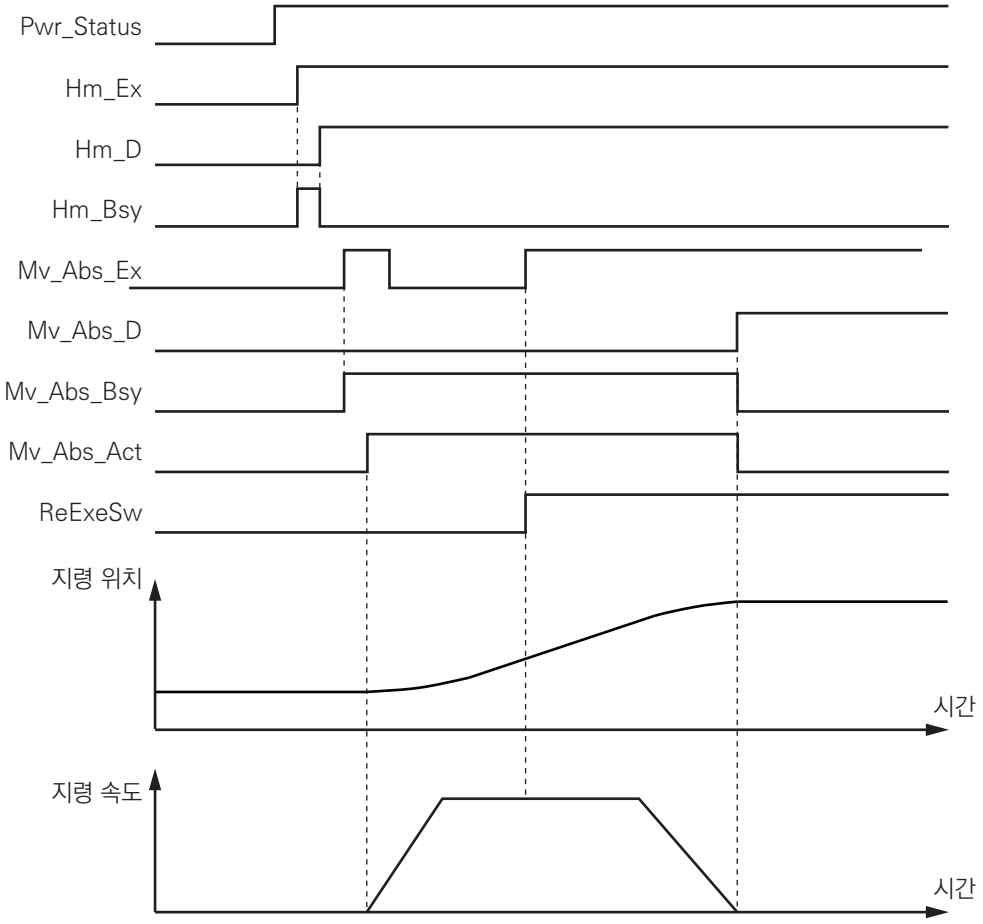
명칭	데이터형	초기값	코멘트
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON 합니다.
ReExeSw	BOOL	FALSE	명령을 재기동하기 위한 변수입니다.
Hm_Ex	BOOL	FALSE	MC_Home(원점 복귀) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.
Mv_Abs_Ex	BOOL	FALSE	MC_MoveAbsolute(절대값 위치 결정) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.

타이밍 차트

● **래더도 프로그램의 경우**



● ST 프로그램의 경우

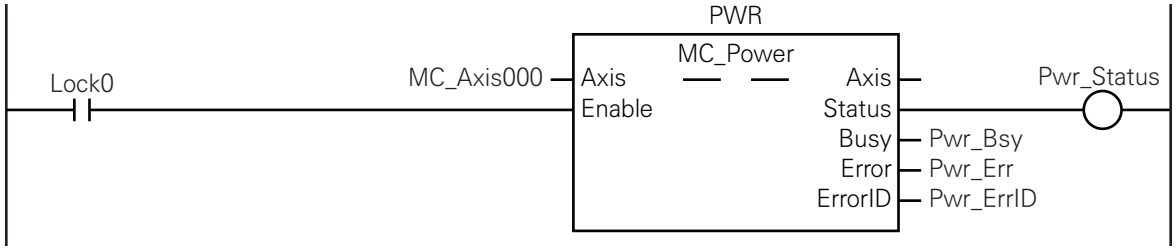


래더도 프로그램

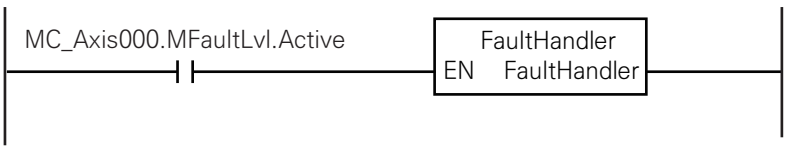
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



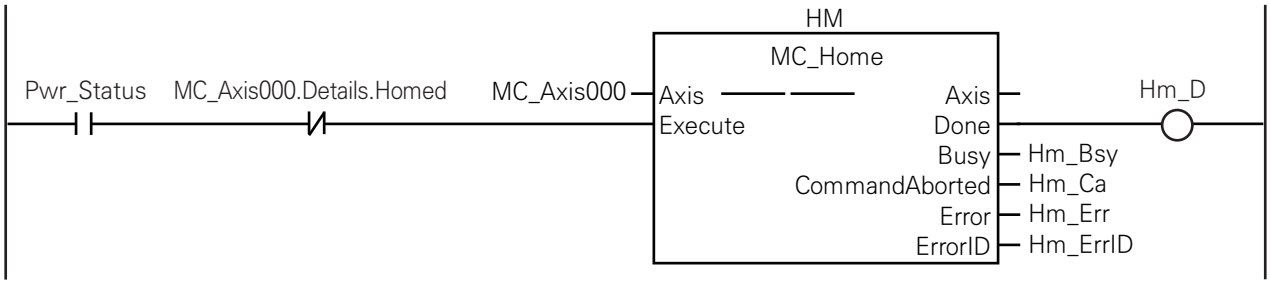
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



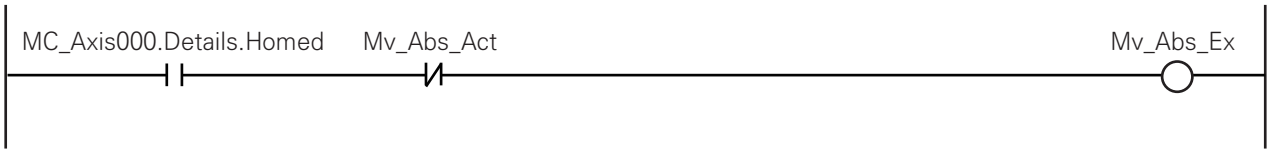
축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



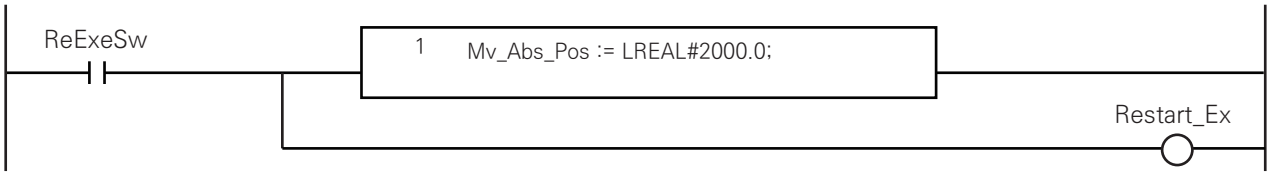
축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



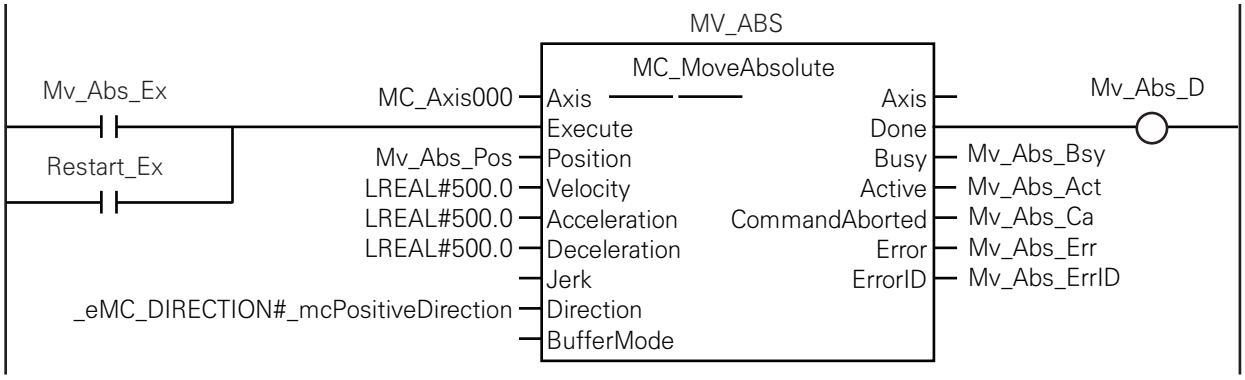
축 0이 원점 확정 후 절대값 위치 결정이 실행되고 있지 않으면 실행합니다.



접점 ReExeSw를 TRUE로 하면 목표 위치를 2000으로 변경하고 절대값 위치 결정을 재기동합니다.



접점 Mv_Abs_Ex의 상태에 따라 절대값 위치 결정을 실행합니다.



ST 프로그램

```
//절대값 위치 결정의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag = FALSE THEN
   //MoveAbsolute 파라메터
   Mv_Abs_Pos := LREAL#1000.0;
   Mv_Abs_Vel := LREAL#500.0;
   Mv_Abs_Acc := LREAL#500.0;
   Mv_Abs_Dec := LREAL#500.0;
   Mv_Abs_Dir := _eMC_DIRECTION#_mcPositiveDirection;

   //입력 파라메터의 설정 완료 플래그를 유효화합니다.
   InitFlag :=TRUE;
END_IF;


// 접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
```

```
END_IF;


// 축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


//축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm_Ex :=TRUE;
END_IF;


//축 0이 원점 복귀 완료 후에 절대값 위치 결정이 실행되고 있지 않으면 실행합니다.
IF (Hm_D=TRUE) AND (Mv_Abs_Act=FALSE) THEN
   Mv_Abs_Ex := TRUE;
ELSE
   Mv_Abs_Ex := FALSE;
END_IF;


//접점 ReExeSw를 TRUE로 하면 목표 위치를 2000으로 변경하고 절대값 위치 결정을 재기동합니다.
IF ReExeSw=TRUE THEN
   Mv_Abs_Pos :=LREAL#2000.0;
   Mv_Abs_Ex := TRUE;
END_IF;

//MC_Power
PWR(
   Axis := MC_Axis000,
   Enable := Pwr_En,
   Status => Pwr_Status,
   Busy => Pwr_Bsy,
   Error => Pwr_Err,
   ErrorID => Pwr_ErrID
);

//MC_Home
HM(
   Axis := MC_Axis000,
   Execute := Hm_Ex,
   Done => Hm_D,
   Busy => Hm_Bsy,
   CommandAborted => Hm_Ca,
   Error => Hm_Err,
   ErrorID => Hm_ErrID
);

//MC_MoveAbsolute
MV_ABS(
   Axis := MC_Axis000,
   Execute := Mv_Abs_Ex,
   Position := Mv_Abs_Pos,
   Velocity := Mv_Abs_Vel,
   Acceleration := Mv_Abs_Acc,
   Deceleration := Mv_Abs_Dec,
```

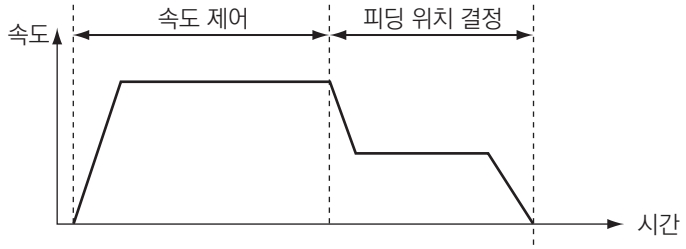
```
    Direction := Mv_Abs_Dir,
    Done => Mv_Abs_D,
    Busy => Mv_Abs_Bsy,
    Active => Mv_Abs_Act,
    CommandAborted => Mv_Abs_Ca,
    Error => Mv_Abs_Err,
    ErrorID => Mv_Abs_ErrID
);
```

10-2-11 인터럽트 피딩 위치 결정

속도 제어 중에 인터럽트 입력이 발생하면 인터럽트 피딩으로 바뀝니다.
로터리 모드에서 속도를 제어할 때 방향 선택(Direction)에서는 다음 중 하나를 지정합니다.

- _mcPositiveDirection(정방향 지정)
- _mcNegativeDirection(역방향 지정)
- _mcCurrentDirection(현재 방향 지정)

이 샘플 프로그램에서는 _mcCurrentDirection(현재 방향 지정)을 지정합니다.
인터럽트 입력 발생 전과 같은 방향으로 인터럽트 피딩하는 경우에는 입력 변수 「FeedDistance(피딩 거리)」에 양수를 설정합니다. 인터럽트 입력 발생 전과 반대 방향으로 인터럽트 피딩하는 경우에는 입력 변수 「FeedDistance(피딩 거리)」에 음수를 설정합니다.
예를 들어 인터럽트 입력 발생 전의 동작이 역방향이며, 입력 변수 「FeedDistance(피딩 거리)」에 양수를 설정하면 역방향으로 인터럽트 피딩합니다. 입력 변수 「FeedDistance(피딩 거리)」에 음수를 설정하면 정방향으로 인터럽트 피딩합니다.



축 파라메터 설정

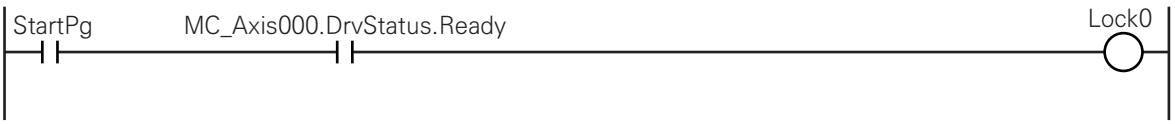
설정 항목	설정 내용	설명
카운트 모드	로터리 모드	위치의 카운트 모드를 로터리 모드로 설정합니다.
링 카운터 상한 설정값	360	링 카운터의 상한값을 「360」으로 설정합니다.
링 카운터 하한 설정값	0	링 카운터의 하한값을 「0」으로 설정합니다.
원점 복귀 동작	원점 프리셋	원점 프리셋으로 원점을 확정합니다.

프로그램에서 사용하는 주요 변수

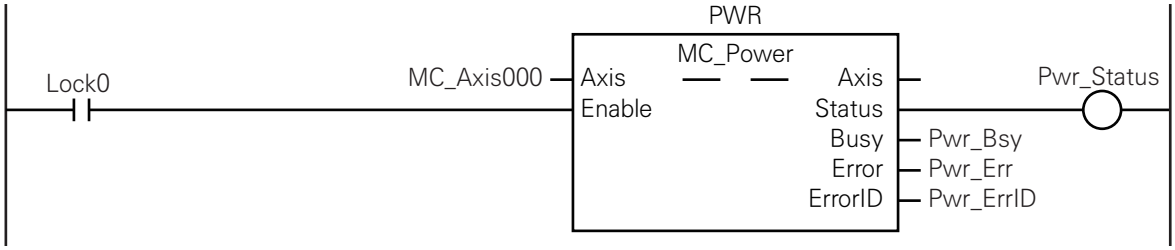
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 0이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON합니다.
TrigRef	_sTRIGGER_REF	—	인터럽트 입력의 트리거 입력 조건을 지정합니다. 이 샘플 프로그램에서는 서보 드라이버의 래치 기능1을 사용합니다.
Hm_Ex	BOOL	FALSE	MC_Home(원점 복귀) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.
Mv_Feed_Ex	BOOL	FALSE	MC_MoveFeed(인터럽터 피딩 위치 결정) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.
InitFlag	BOOL	FALSE	MC_MoveFeed(인터럽터 피딩 위치 결정) 명령의 입력 파라메터 설정 완료를 나타내는 변수입니다.

래더도 프로그램

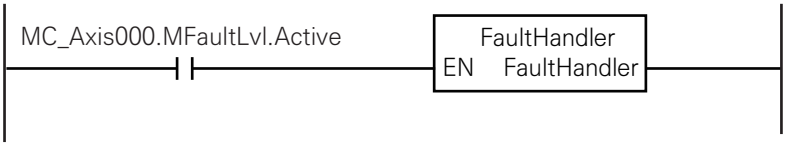
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



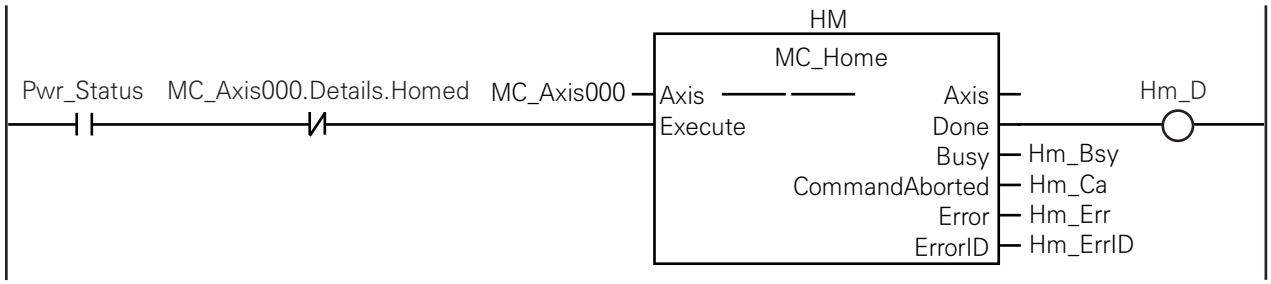
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



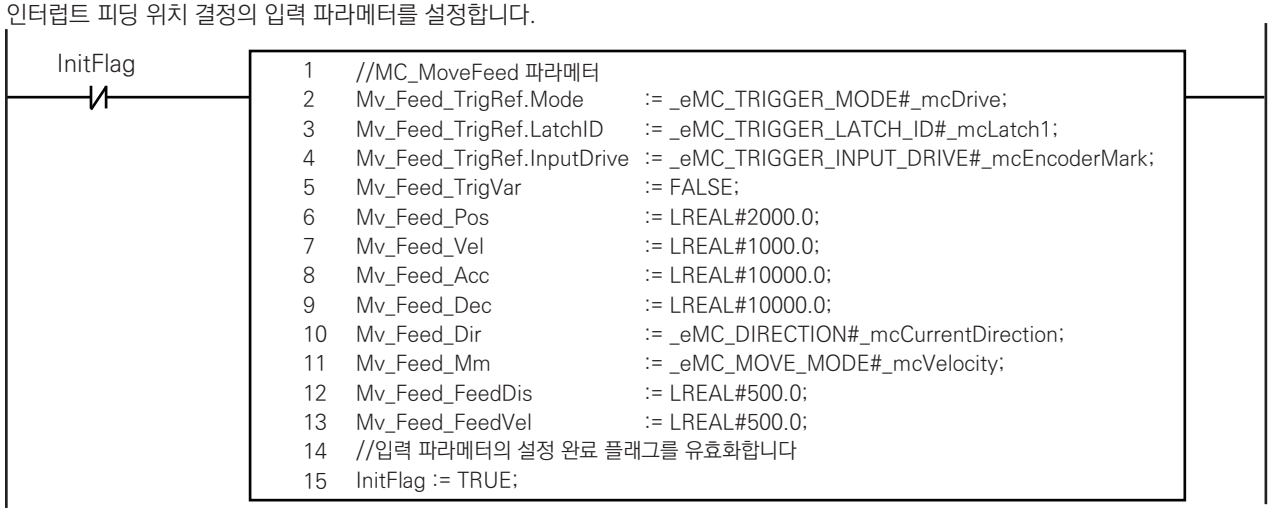
축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



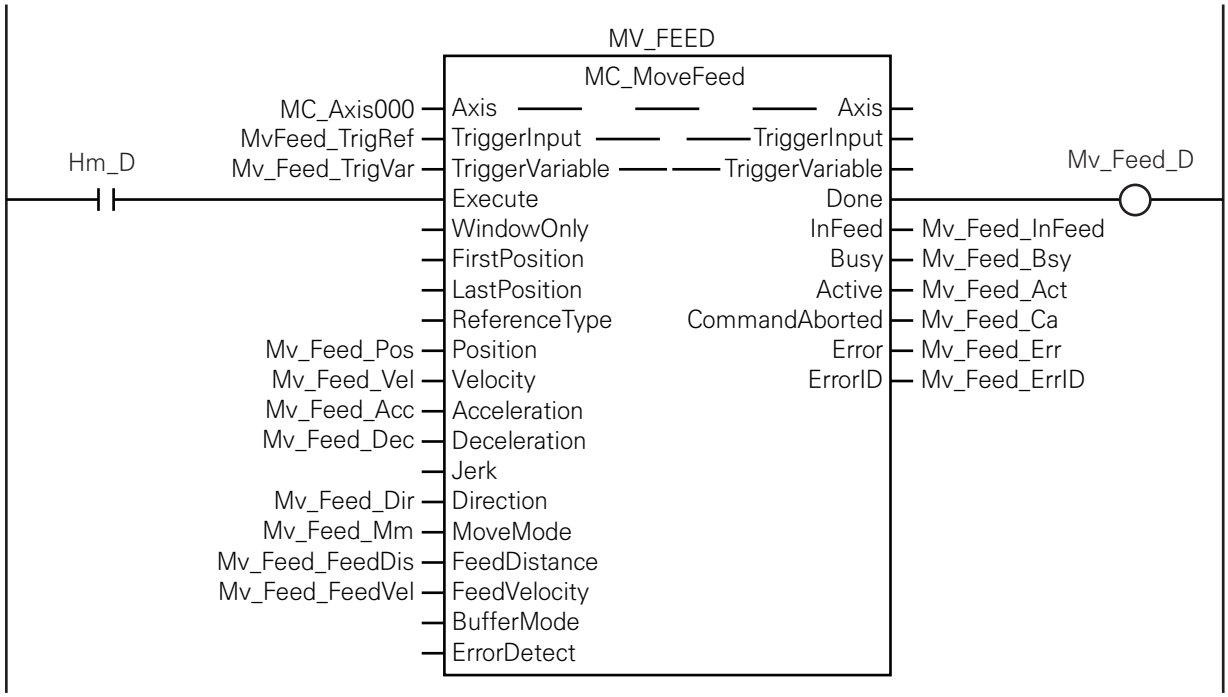
축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



인터럽트 피딩 위치 결정의 입력 파라메터를 설정합니다.



원점 복귀가 완료되면 인터럽트 피딩 위치 결정을 실행합니다.



ST 프로그램

```
//인터럽트 피딩 위치 결정의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag=FALSE THEN
   //MC_MoveFeed 파라메터
   Mv_Feed_TrigRef.Mode := _eMC_TRIGGER_MODE#_mcDrive;
   Mv_Feed_TrigRef.LatchID := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
   Mv_Feed_TrigRef.InputDrive := _eMC_TRIGGER_INPUT_DRIVE#_mcEncoderMark;
   Mv_Feed_TrigVar := FALSE;
   Mv_Feed_Pos := LREAL#2000.0;
   Mv_Feed_Vel := LREAL#1000.0;
   Mv_Feed_Acc := LREAL#10000.0;
   Mv_Feed_Dec := LREAL#10000.0;
   Mv_Feed_Dir := _eMC_DIRECTION#_mcCurrentDirection;
   Mv_Feed_Mm := _eMC_MOVE_MODE#_mcVelocity;
   Mv_Feed_FeedDis := LREAL#500.0;
   Mv_Feed_FeedVel := LREAL#500.0;

   //입력 파라메터의 설정 완료 플래그를 유효화합니다.
   InitFlag := TRUE;
END_IF;


// 접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;


// 축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


//축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm_Ex :=TRUE;
END_IF;


원점을 확정하면 인터럽트 피딩 위치 결정을 실행합니다
IF Hm_D=TRUE THEN
   Mv_Feed_Ex :=TRUE;
END_IF;

// MC_Power
PWR(
   Axis := MC_Axis000,
   Enable := Pwr_En,
   Status => Pwr_Status,
   Busy => Pwr_Bsy,
   Error => Pwr_Err,
   ErrorID => Pwr_ErrID
);
```

```
//MC_Home
HM(
   Axis := MC_Axis000,
   Execute := Hm_Ex,
   Done => Hm_D,
   Busy => Hm_Bsy,
   CommandAborted => Hm_Ca,
   Error => Hm_Err,
   ErrorID => Hm_ErrID
);

//MC_MoveFeed
MV_FEED(
   Axis := MC_Axis000,
   TriggerInput := Mv_Feed_TrigRef,
   TriggerVariable := Mv_Feed_TrigVar,
   Execute := Mv_Feed_Ex,
   Position := Mv_Feed_Pos,
   Velocity := Mv_Feed_Vel,
   Acceleration := Mv_Feed_Acc,
   Deceleration := Mv_Feed_Dec,
   Direction := Mv_Feed_Dir,
   MoveMode := Mv_Feed_Mm,
   FeedDistance := Mv_Feed_FeedDis,
   FeedVelocity := Mv_Feed_FeedVel,
   Done => Mv_Feed_D,
   InFeed => Mv_Feed_InFeed,
   Busy => Mv_Feed_Bsy,
   Active => Mv_Feed_Act,
   CommandAborted => Mv_Feed_Ca,
   Error => Mv_Feed_Err,
   ErrorID => Mv_Feed_ErrID
);
```

10-2-12 명령 재기동으로 캠 테이블 변환하기

캠 동작 실행 중에 캠 테이블을 변환합니다.
축 0의 지령 위치가 5000 이하일 때는 CamProfile0을 사용하고 5000을 초과하면 CamTable1을 사용합니다.

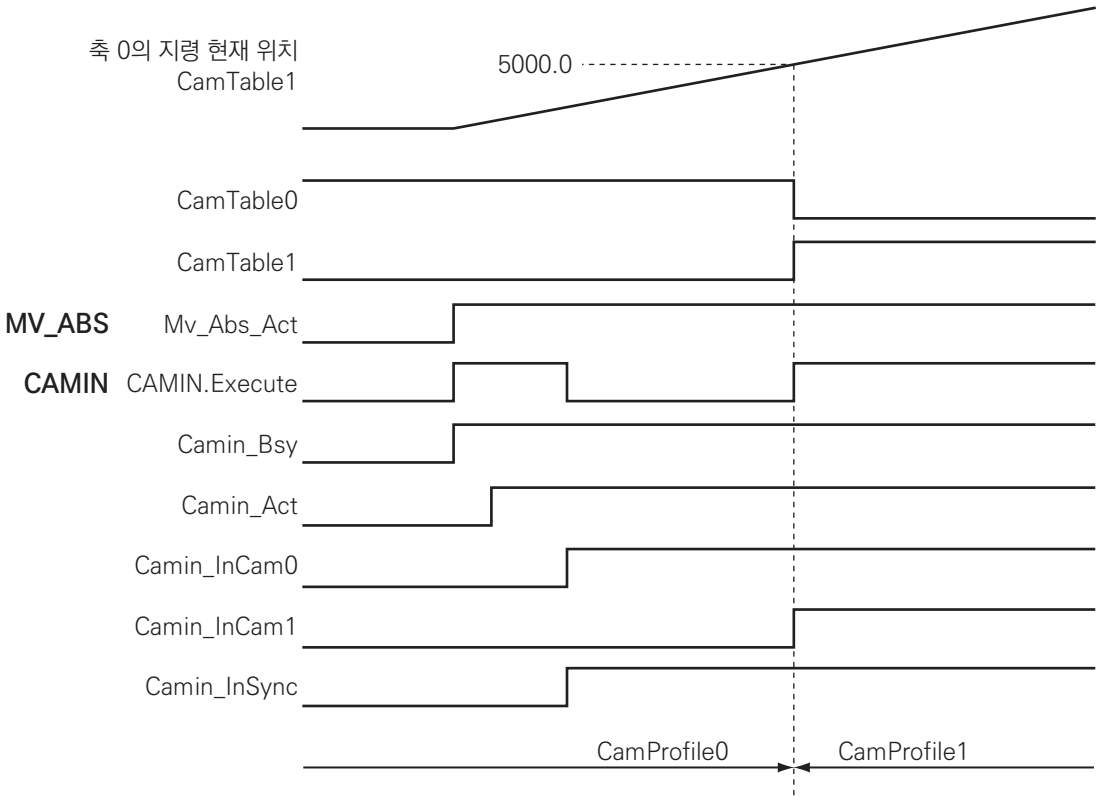
프로그램에서 사용하는 주요 변수

명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis001	_sAXIS_REF	—	축 1의 축 변수입니다.
CamProfile0	ARRAY[0..100] OF _sMC_CAM_REF	—	캠 데이터 변수입니다. *1
CamProfile1	ARRAY[0..10] OF _sMC_CAM_REF	—	캠 데이터 변수입니다. *1
Pwr1_S	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Pwr2_S	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
CamTable0	BOOL	FALSE	CamProfile0을 캠 테이블로 사용하는 경우에 TRUE로 합니다.
CamTable1	BOOL	FALSE	CamProfile1을 캠 테이블로 사용하는 경우에 TRUE로 합니다.
Camin_InCam0	BOOL	FALSE	MC_CamIn 명령의 인스턴스 CAMIN의 출력 InCam에 할당한 변수입니다. CamProfile0에서 캠 동작 중에 TRUE가 됩니다. 일단 TRUE가 되면 동작을 종료 또는 중단할 때까지 TRUE를 유지합니다.
Camin_InCam1	BOOL	FALSE	MC_CamIn 명령의 인스턴스 CAMIN의 출력 InCam에 할당한 변수입니다. CamProfile1에서 캠 동작 중에 TRUE가 됩니다. 일단 TRUE가 되면 동작을 종료 또는 중단할 때까지 TRUE를 유지합니다.
Mv_Abs_Act	BOOL	FALSE	MC_MoveAbsolute 명령의 인스턴스 MV_ABS의 출력 Active에 할당한 변수입니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON합니다.
Hm1_Ex	BOOL	FALSE	축 0에서 MC_Home(원점 복귀) 명령을 기동하기 위한 변수입니다.
Hm2_Ex	BOOL	FALSE	축 1에서 MC_Home(원점 복귀) 명령을 기동하기 위한 변수입니다.
Mv_Abs_Ex	BOOL	FALSE	MC_MoveAbsolute(절대값 위치 결정) 명령을 기동하기 위한 변수입니다.
Camin_Ex	BOOL	FALSE	MC_CamIn(캠 동작 시작) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.

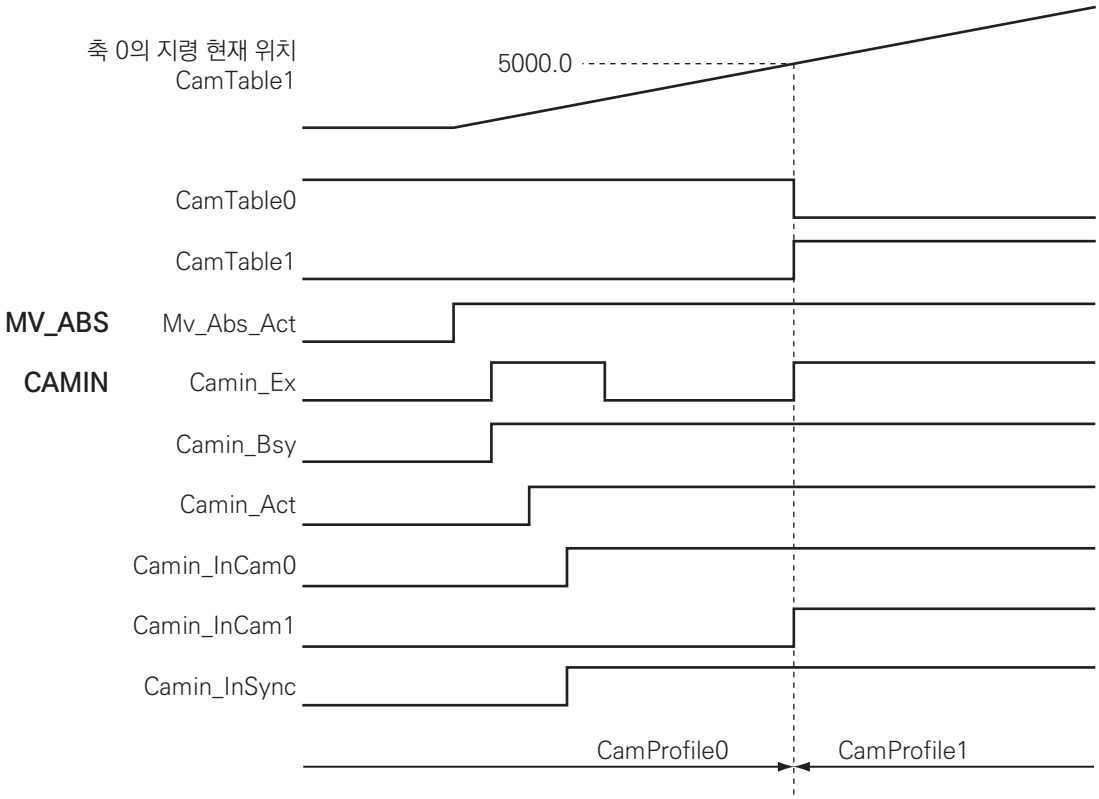
*1. 배열의 요소 수 ARRAY[0..N]는 Sysmac Studio의 캠 에디터 기능으로 설정합니다. 본 샘플 프로그램에서는 [0~109]로 합니다.

타이밍 차트

● 래더도 프로그램의 경우



● ST 프로그램의 경우



래더도 프로그램

2개의 캠 테이블을 변환하여 사용할 경우에는 MC_CamIn(캠 동작 시작) 명령과 명칭이 같은 인스턴스를 2개 배치합니다.
이 경우 각 인스턴스의 출력 변수「InCam(캠 동작 중)」에는 다른 출력 파라메터를 할당합니다. 동일한 출력 파라메터를 할당하면 이상이 발생합니다.
이 샘플 프로그램에서는 JMP(점프) 명령을 이용하여 2개의 인스턴스가 동시에 실행되지 않도록 합니다.

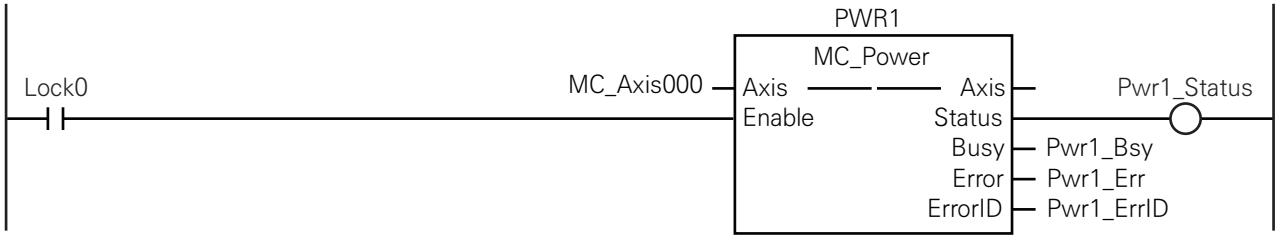
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



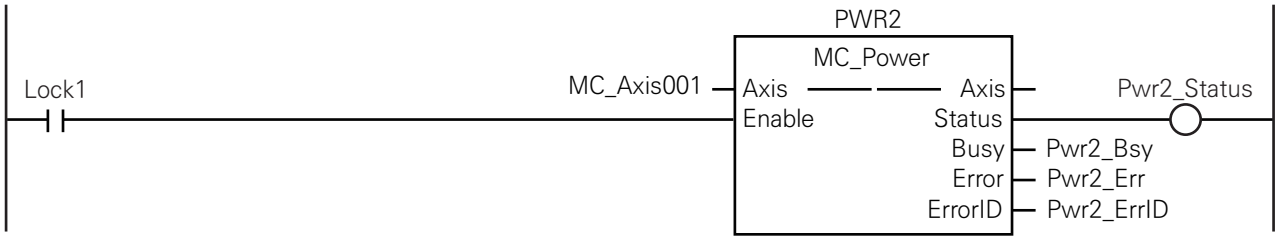
접점 StartPg가 TRUE이고 축 1의 서보 드라이버가 서보 준비 상태인지 확인합니다.



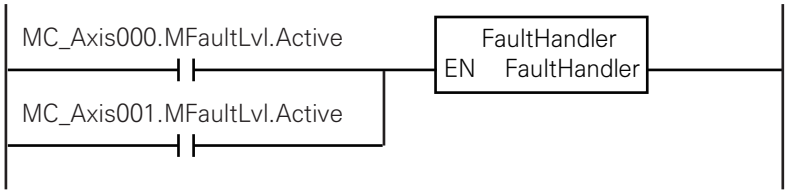
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



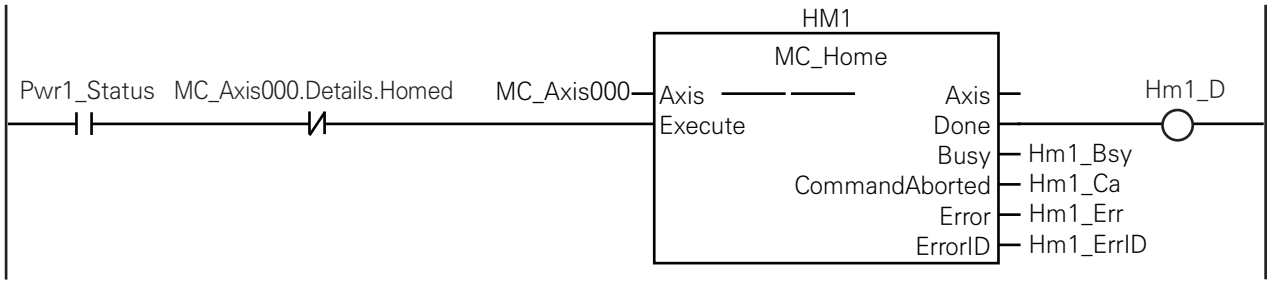
축 1의 서보 드라이버가 서보 준비 상태이면 축 1을 서버 ON합니다.



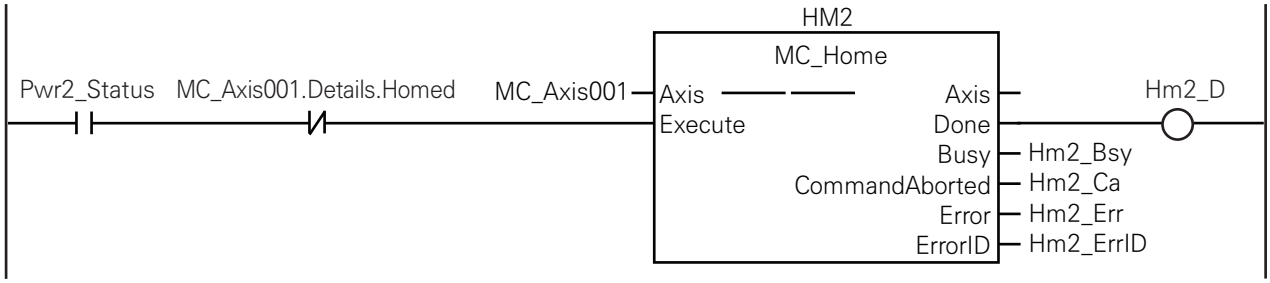
축 0 또는 축 1에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



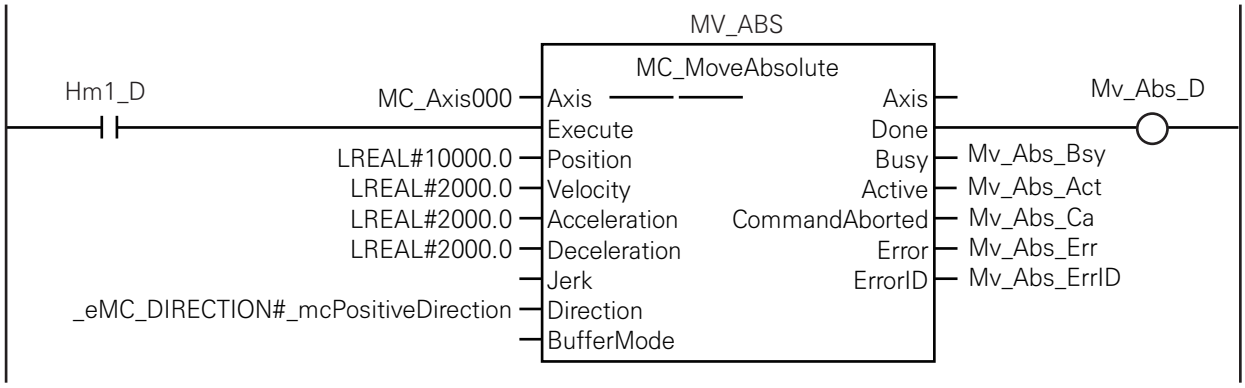
축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



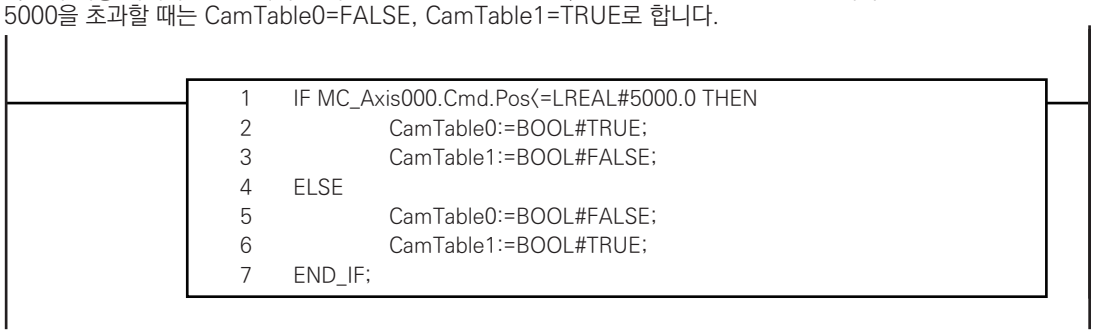
축 1이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



축 0의 원점 복귀가 완료되면 절대값 위치 결정(MC_MoveAbsolute) 명령을 실행합니다.

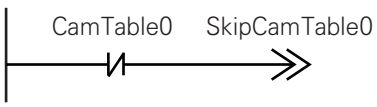


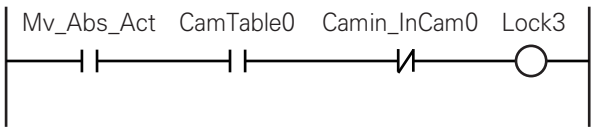
축 0의 지령 위치가 5000 이하일 때는 CamTable0=TRUE, CamTable1=FALSE로 합니다.
5000을 초과할 때는 CamTable0=FALSE, CamTable1=TRUE로 합니다.

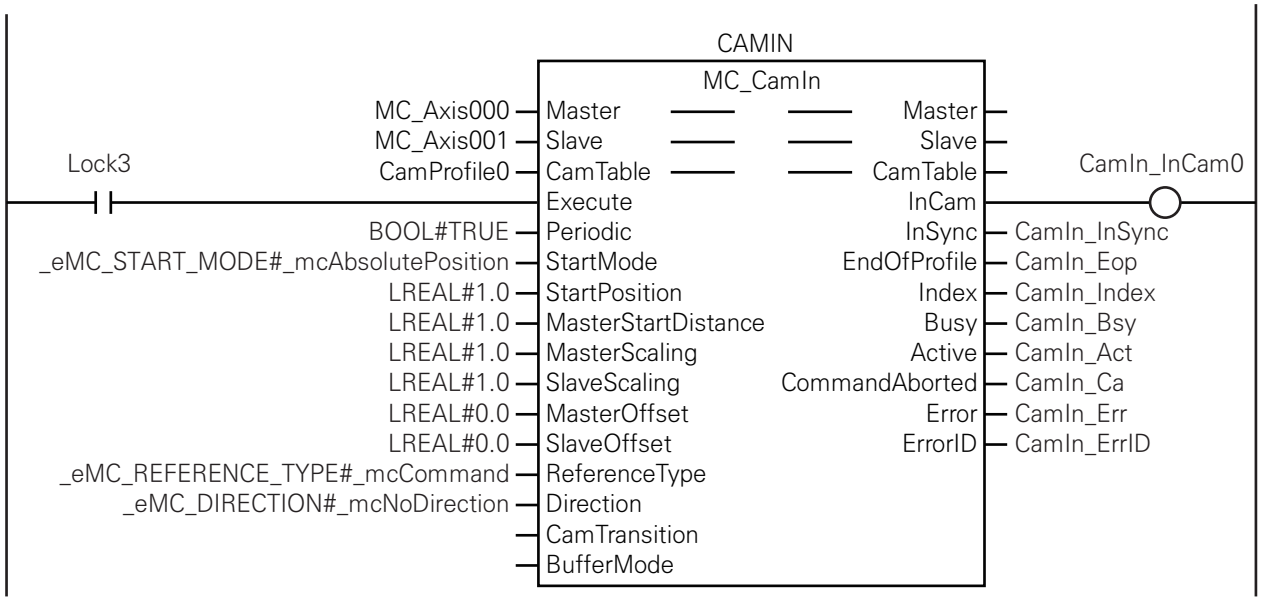


```
1    IF MC_Axis000.Cmd.Pos<=LREAL#5000.0 THEN
2            CamTable0:=BOOL#TRUE;
3            CamTable1:=BOOL#FALSE;
4    ELSE
5            CamTable0:=BOOL#FALSE;
6            CamTable1:=BOOL#TRUE;
7    END_IF;
```

절대값 위치 결정이 실행 중이고 CamTable0=TRUE인 경우에는 캠 테이블에 CamProfile0을 사용한다.
인스턴스를 실행합니다.
InCam=TRUE일 때 Execute를 FALSE로 합니다.

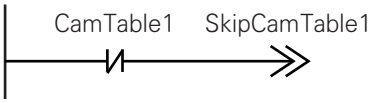


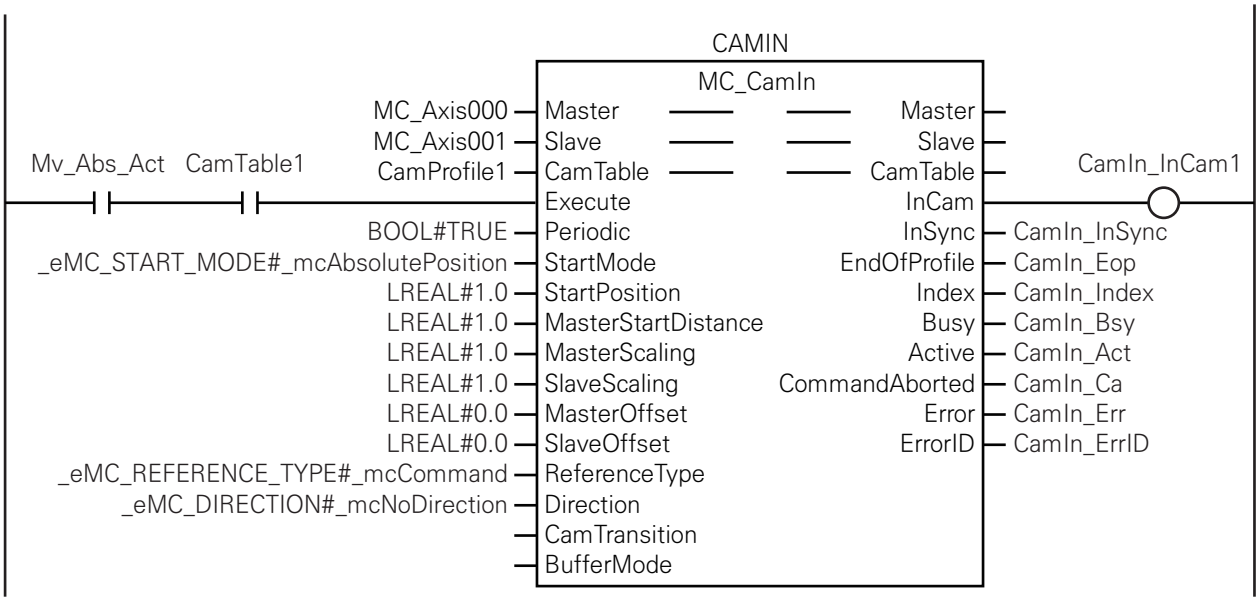




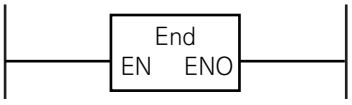
절대값 위치 결정이 실행 중이고 CamTable1=TRUE인 경우에는 캠 테이블에 CamProfile1을 사용한다.
인스턴스를 실행합니다.

SkipCamTable0





SkipCamTable1



ST 프로그램

```
//절대값 위치 결정과 캠 동작 시작의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag=FALSE THEN
   //절대값 위치 결정(MC_MoveAbsolute)의 입력 파라메터를 설정합니다.
   Mv_Abs_Pos := LREAL#10000.0;
   Mv_Abs_Vel := LREAL#2000.0;
   Mv_Abs_Acc := LREAL#2000.0;
   Mv_Abs_Dec := LREAL#2000.0;

   //캠 동작 시작(MC_CamIn)의 입력 파라메터를 설정합니다.
   Camin_EM := TRUE;
   Camin_StMode := _eMC_START_MODE#_mcAbsolutePosition;
   Camin_StPos := LREAL#1.0;
   Camin_MStDis := LREAL#1.0;
   Camin_MSc := LREAL#1.0;
   Camin_SSc := LREAL#1.0;
   Camin_MO := LREAL#0.0;
   Camin_SO := LREAL#0.0;
   Camin_RT := _eMC_REFERENCE_TYPE#_mcCommand;
   Camin_Dir := _eMC_DIRECTION#_mcNoDirection;

   //캠 테이블 선택
   CamTable0 :=BOOL#TRUE;
   CamTable1 :=BOOL#FALSE;

   //입력 파라메터의 설정 완료 플래그를 유효화합니다.
   InitFlag := TRUE;
END_IF;


// 접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr1_En :=TRUE;
ELSE
   Pwr1_En :=FALSE;
END_IF;


//접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 1을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
   Pwr2_En :=TRUE;
ELSE
   Pwr2_En :=FALSE;
END_IF;


//축 0 또는 축 1에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
   FaultHandler();
END_IF;


//축 0이 서보 ON이고 원점 미확정인 경우 원점 복귀를 실행합니다.
IF (Pwr1_S=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex :=TRUE;
```

```
END_IF;


//축 1이 서보 ON이고 원점 미확정인 경우 원점 복귀를 실행합니다.
IF (Pwr2_S=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
   Hm2_Ex :=TRUE;
END_IF;


//축 0의 원점 복귀가 완료되면 절대값 위치 결정을 실행합니다.
IF Hm1_D=TRUE  THEN
   Mv_Abs_Ex := TRUE;
END_IF;


//축 0의 지령 위치가 5000 이하일 때는 CamTable0=TRUE, CamTable1=FALSE로 합니다.
//5000을 초과할 때는 CamTable0=FALSE, CamTable1=TRUE로 합니다.
IF MC_Axis000.Cmd.Pos<=LREAL#5000.0 THEN
   CamTable0 :=BOOL#TRUE;
   CamTable1 :=BOOL#FALSE;
ELSE
   CamTable0 :=BOOL#FALSE;
   CamTable1 :=BOOL#TRUE;
END_IF;


//절대값 위치 결정이 실행 중이고 CamTable0=TRUE인 경우에는 캠 테이블에 CamProfile0을 사용한다.
//인스턴스를 실행합니다.
//InCam=TRUE일 때 Execute를 FALSE로 합니다.
IF (Mv_Abs_Act=TRUE)
 AND (CamTable0=TRUE)
 AND (Camin_InCam0=FALSE) THEN
   Camin_Ex := TRUE;
ELSE
   Camin_Ex := FALSE;
END_IF;


//절대값 위치 결정이 실행 중이고 CamTable1=TRUE인 경우에는 캠 테이블에 CamProfile1을 사용한다.
//인스턴스를 실행합니다.
IF (Mv_Abs_Act=TRUE) AND (CamTable1=TRUE) THEN
   Camin_Ex := TRUE;
END_IF;

//MC_Camin
IF CamTable0=TRUE THEN
   CAMIN(
      Master := MC_Axis000,
      Slave := MC_Axis001,
      CamTable := CamProfile0,
      Execute := Camin_Ex,
      Periodic := Camin_EM,
      StartMode := Camin_StMode,
      StartPosition := Camin_StPos,
      MasterStartDistance := Camin_MStDis,
      MasterScaling := Camin_MSc,
      SlaveScaling := Camin_SSc,
      MasterOffset := Camin_MO,
      SlaveOffset := Camin_SO,
```

```
        ReferenceType := Camin_RT,
        Direction := Camin_Dir,
        InCam => Camin_InCam0,
        InSync => Camin_InSync,
        EndOfProfile => Camin_EOP,
        Index => Camin_Index,
        Busy => Camin_Bsy,
        Active => Camin_Act,
        CommandAborted => Camin_Ca,
        Error => Camin_Err,
        ErrorID => Camin_ErrID
    );
END_IF;

IF CamTable1=TRUE THEN
    CAMIN(
        Master := MC_Axis000,
        Slave := MC_Axis001,
        CamTable := CamProfile1,
        Execute := Camin_Ex,
        Periodic := Camin_EM,
        StartMode := Camin_StMode,
        StartPosition := Camin_StPos,
        MasterStartDistance := Camin_MStDis,
        MasterScaling := Camin_MSc,
        SlaveScaling := Camin_SSc,
        MasterOffset := Camin_MO,
        SlaveOffset := Camin_SO,
        ReferenceType := Camin_RT,
        Direction := Camin_Dir,
        InCam => Camin_InCam1,
        InSync => Camin_InSync,
        EndOfProfile => Camin_EOP,
        Index => Camin_Index,
        Busy => Camin_Bsy,
        Active => Camin_Act,
        CommandAborted => Camin_Ca,
        Error => Camin_Err,
        ErrorID => Camin_ErrID
    );
END_IF;

//축 0의 MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_S,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

//축 1의 MC_Power
PWR2(
    Axis := MC_Axis001,
    Enable := Pwr2_En,
    Status => Pwr2_S,
    Busy => Pwr2_Bsy,
    Error => Pwr2_Err,
```

```
    ErrorID => Pwr2_ErrID
);

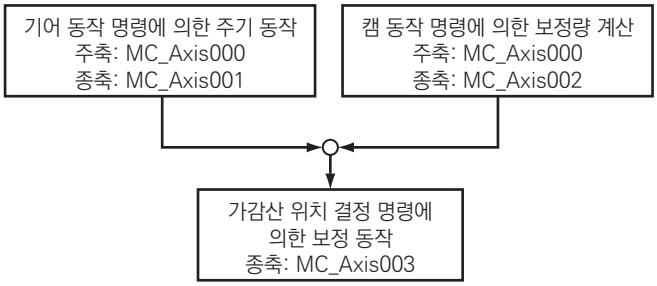
//축 0의 MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

//축 1의 MC_Home
HM2(
    Axis := MC_Axis001,
    Execute := Hm2_Ex,
    Done => Hm2_D,
    Busy => Hm2_Bsy,
    CommandAborted => Hm2_Ca,
    Error => Hm2_Err,
    ErrorID => Hm2_ErrID
);

//MC_MoveAbsolute
MV_ABS(
    Axis := MC_Axis000,
    Execute := Mv_Abs_Ex,
    Position := Mv_Abs_Pos,
    Velocity := Mv_Abs_Vel,
    Acceleration := Mv_Abs_Acc,
    Deceleration := Mv_Abs_Dec,
    Direction := Mv_Abs_Dir,
    Done => Mv_Abs_D,
    Busy => Mv_Abs_Bsy,
    Active => Mv_Abs_Act,
    CommandAborted => Mv_Abs_Ca,
    Error => Mv_Abs_Err,
    ErrorID => Mv_Abs_ErrID
);
```

10-2-13 캠 프로파일 곡선을 사용하여 동기 위치 보정하기

캠 프로파일 곡선을 사용하여 기어 동작을 하는 종축을 보정합니다.
기어 동작의 종축은 가상 서보 축의 MC_Axis001이고 캠 동작의 종축은 가상 서보 축의 MC_Axis002입니다. 이러한 종축을 MC_CombineAxes로 가산하고 그 결과를 서보 축의 MC_Axis003에 출력합니다.
주축은 서보 축의 MC_Axis000입니다.
처리의 흐름은 다음과 같습니다.



● 축 종별 설정

축 종별은 각 축의 축 파라메터에서 아래와 같이 설정합니다.

설정 항목	설정 내용			
	축 1	축 2	축 3	축 4
축 변수명	MC_Axis000	MC_Axis001	MC_Axis002	MC_Axis003
축 종별	서보 축	가상 서보 축	가상 서보 축	서보 축

프로그램에서 사용하는 주요 변수

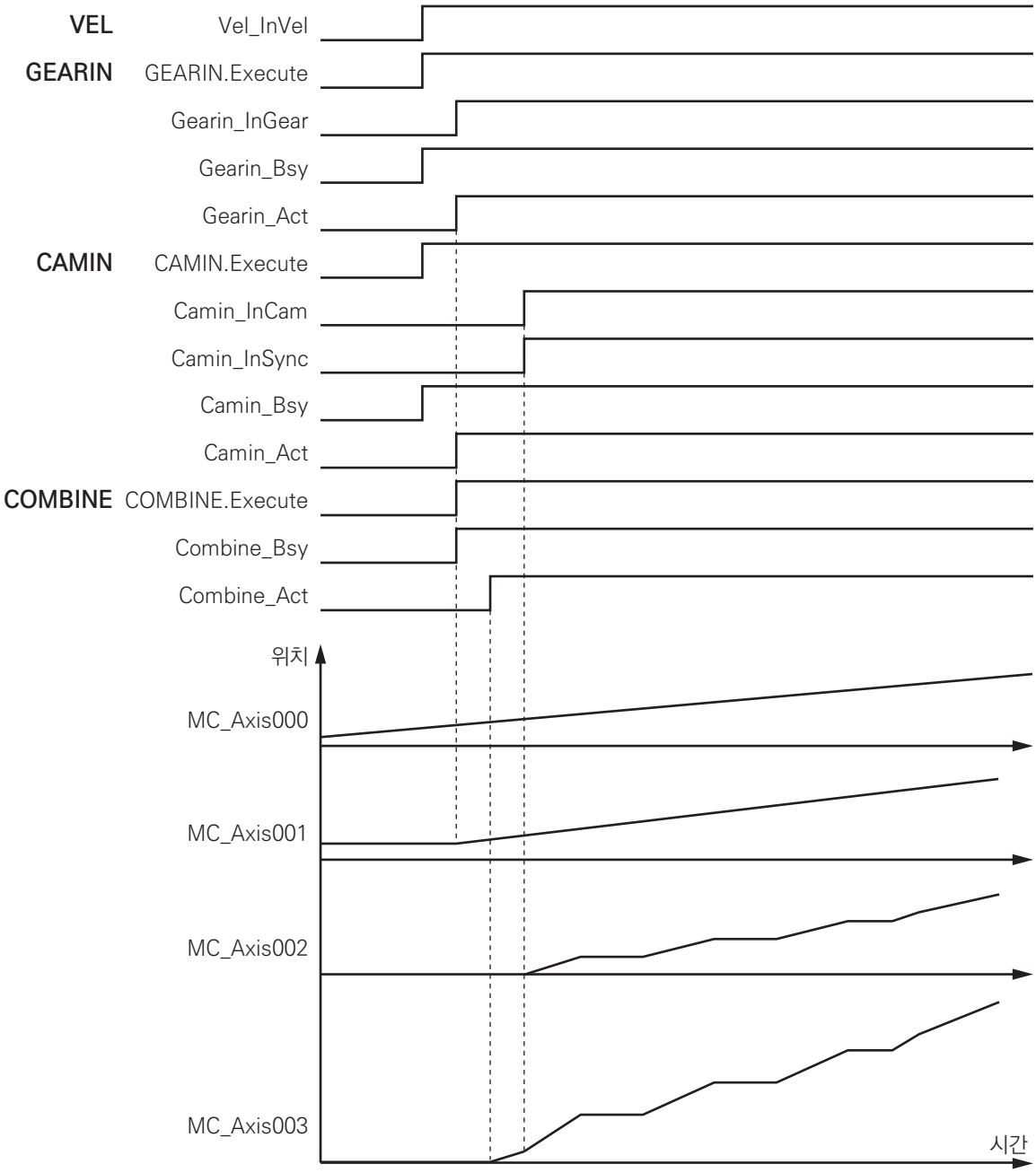
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 0이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	—	축 1의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 1에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis002	_sAXIS_REF	—	축 2의 축 변수입니다.
MC_Axis002.MFaultLvl.Active	BOOL	FALSE	축 2에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis003	_sAXIS_REF	—	축 3의 축 변수입니다.
MC_Axis003.MFaultLvl.Active	BOOL	FALSE	축 3에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis003.Details.Homed	BOOL	FALSE	축 3이 원점 확정 상태일 때 TRUE가 됩니다.
CamProfile0	ARRAY[0..109] OF _sMC_CAM_REF	—	캠 데이터 변수입니다. *1
Pwr1_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Pwr4_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR4의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Vel_InVel	BOOL	FALSE	축 0의 MC_MoveVelocity가 목표 속도에 도달한 경우에 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON합니다.

명칭	데이터형	초기값	코멘트
Gearin_Ex	BOOL	FALSE	MC_GearIn(기어 동작 시작) 명령을 기동하기 위한 변수입니다. *2
Camin_Ex	BOOL	FALSE	MC_CamIn(캠 동작 시작) 명령을 기동하기 위한 변수입니다. *2
Combine_Ex	BOOL	FALSE	MC_CombineAxes(가감산 위치 결정) 명령을 기동하기 위한 변수입니다. *2

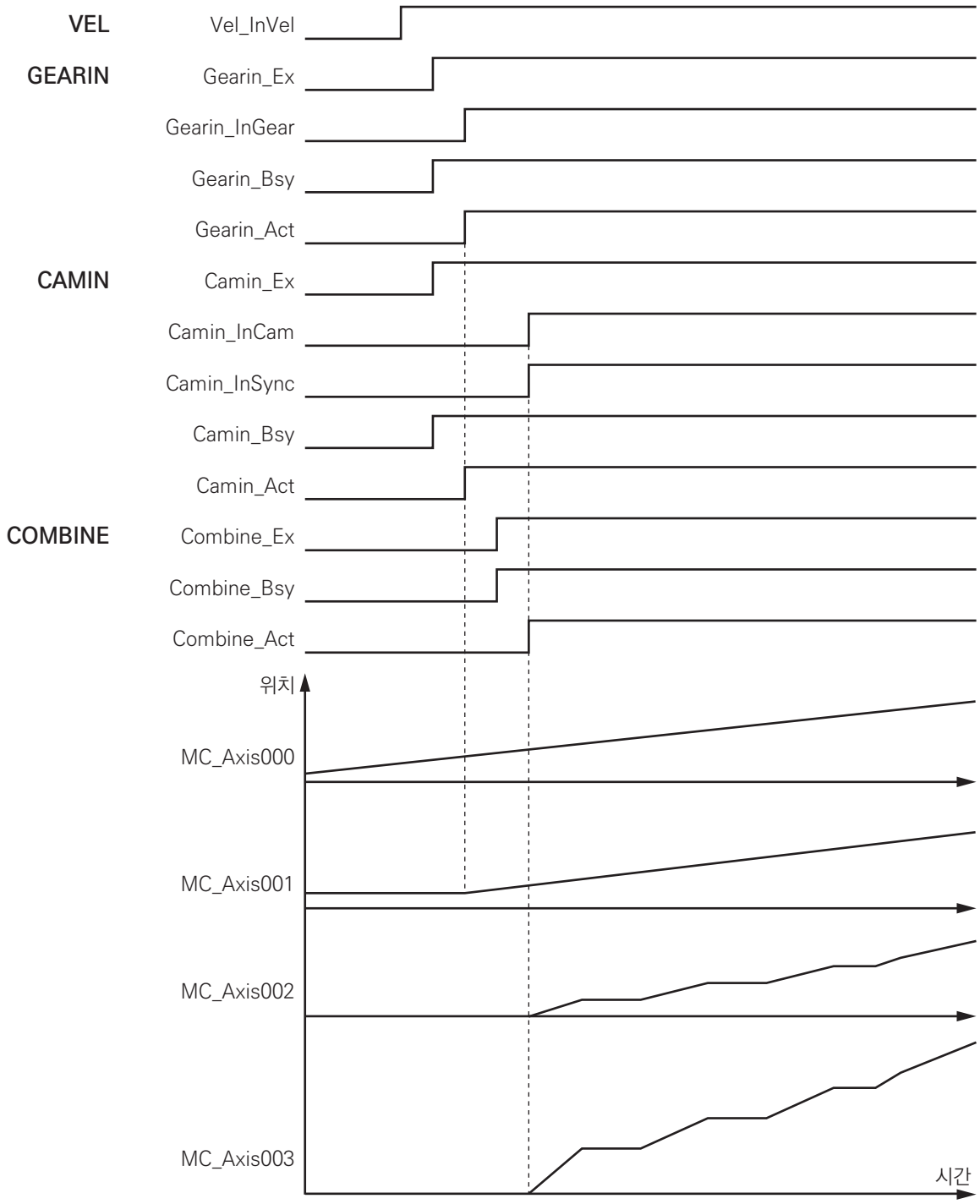
*1. 배열의 요소 수 ARRAY[0..N]는 Sysmac Studio의 캠 에디터 기능으로 설정합니다. 본 샘플 프로그램에서는 [0~109]로 합니다.
*2. ST 프로그램에서 사용합니다.

타이밍 차트

● **래더도 프로그램의 경우**

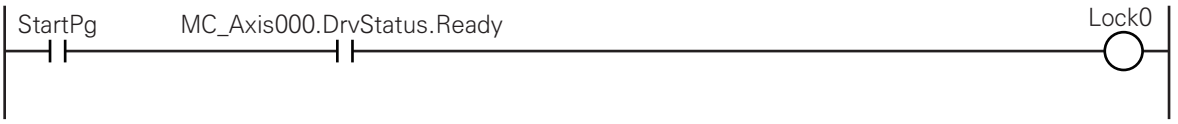


● ST 프로그램의 경우

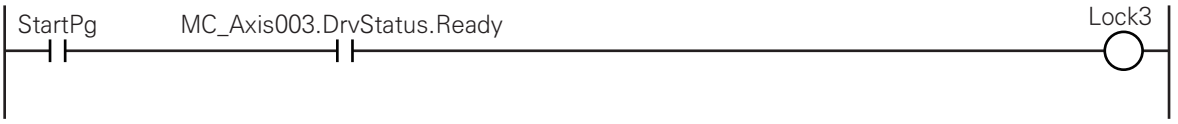

VEL
Vel_InVel
GEARIN
Gearin_Ex
Gearin_InGear
Gearin_Bsy
Gearin_Act
CAMIN
Camin_Ex
Camin_InCam
Camin_InSync
Camin_Bsy
Camin_Act
COMBINE
Combine_Ex
Combine_Bsy
Combine_Act
위치
MC_Axis000
MC_Axis001
MC_Axis002
MC_Axis003
시간

래더도 프로그램

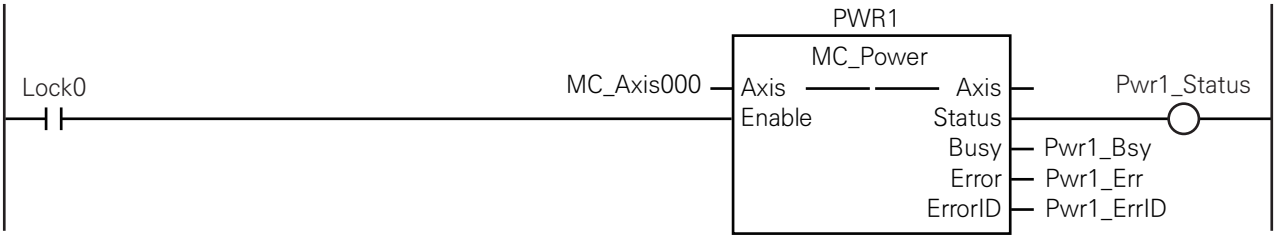
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



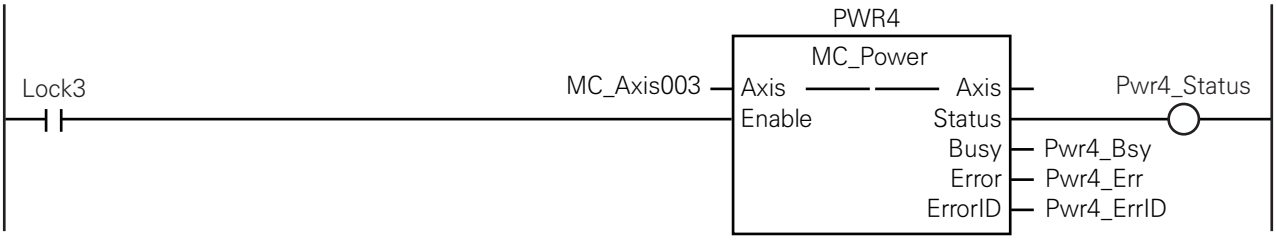
접점 StartPg가 TRUE이고 축 3의 서보 드라이버가 서보 준비 상태인지 확인합니다.



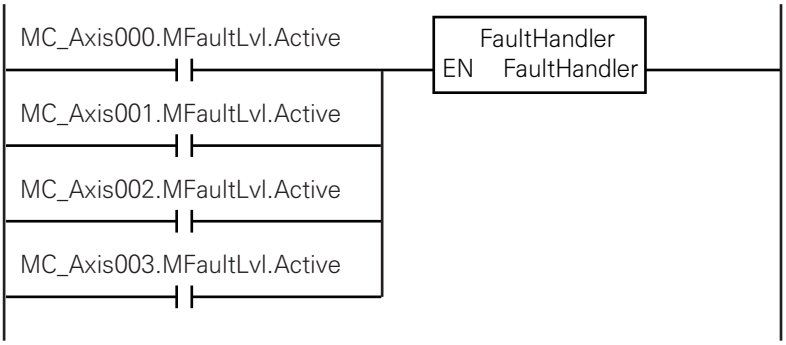
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



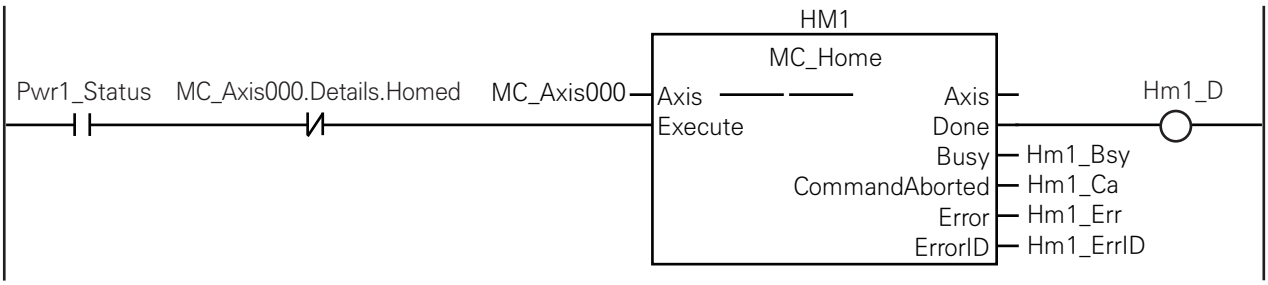
축 3의 서보 드라이버가 서보 준비 상태이면 축 3을 서버 ON합니다.



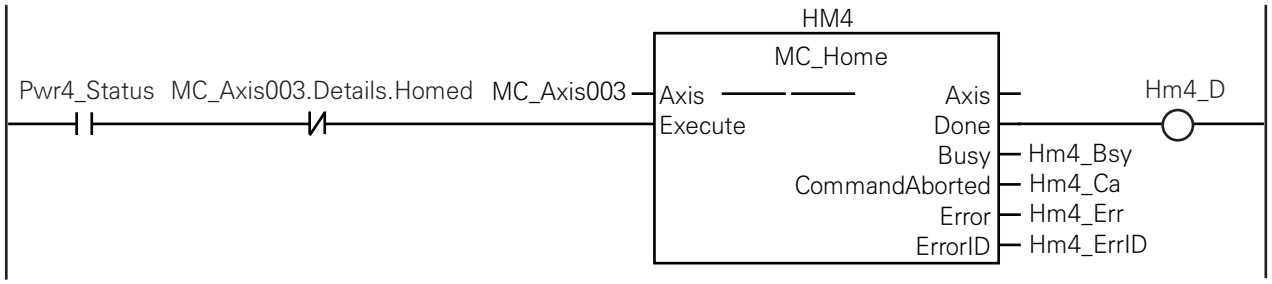
축 그룹 혹은 구성 축에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



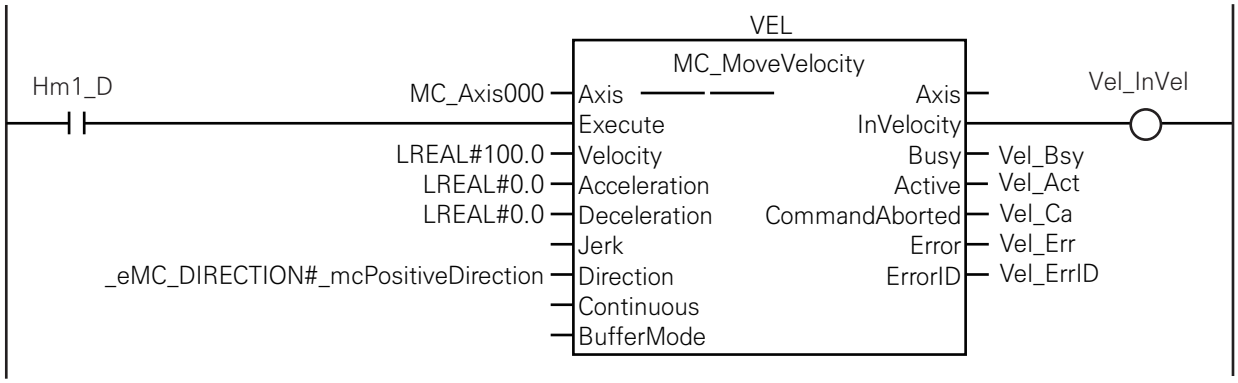
축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



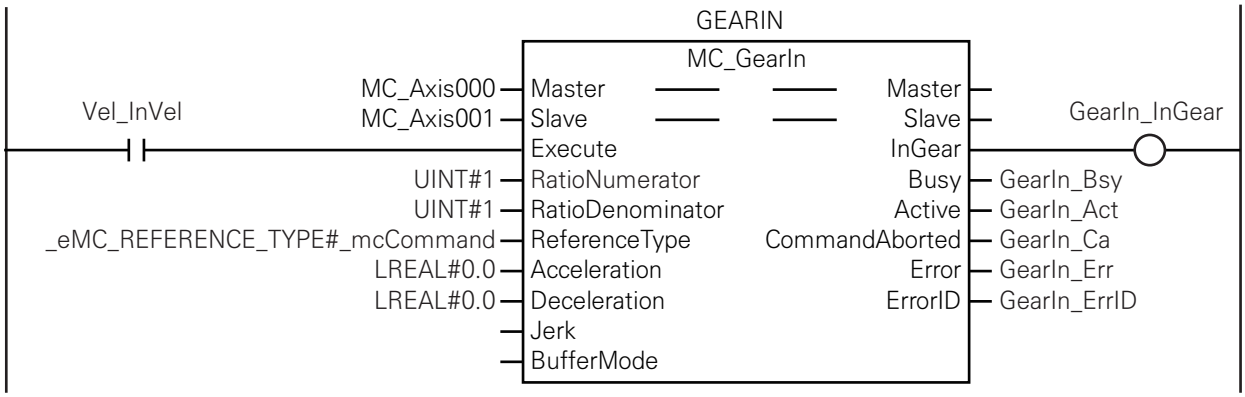
축 3이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



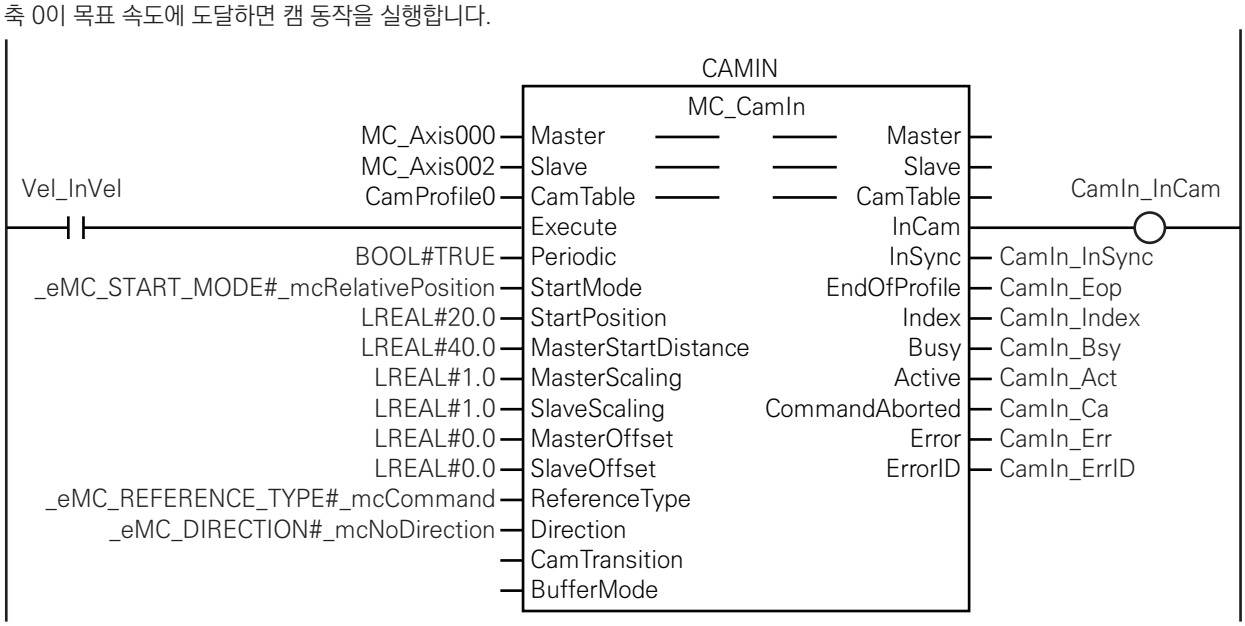
축 0의 원점 복귀가 완료되면 속도 제어를 실행합니다.



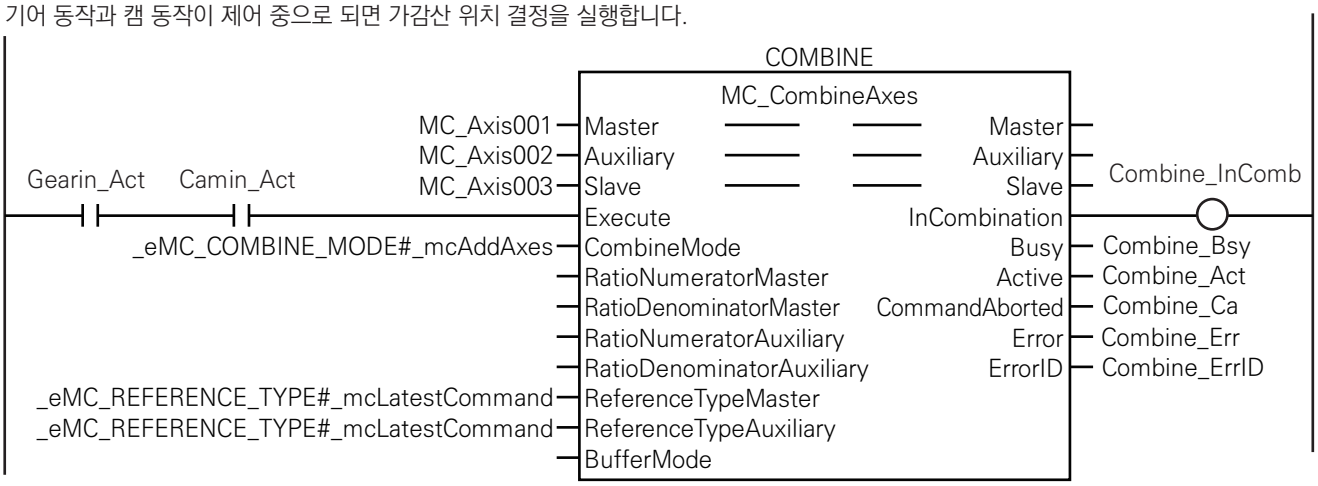
축 0이 목표 속도에 도달하면 기어 동작을 실행합니다.



축 0이 목표 속도에 도달하면 캠 동작을 실행합니다.



기어 동작과 캠 동작이 제어 중으로 되면 가감산 위치 결정을 실행합니다.



ST 프로그램

```
//동작 명령의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag=FALSE THEN
  //속도 제어(MC_MoveVelocity)의 입력 파라메터에 값을 설정합니다.
  Vel_Vel := LREAL#100.0;
  Vel_Acc := LREAL#0.0;
  Vel_Dec := LREAL#0.0;
  Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

  //캠 동작 시작(MC_CamIn)의 입력 파라메터에 값을 설정합니다.
  Camin_Em := TRUE;
  Camin_Sm := _eMC_START_MODE#_mcRelativePosition;
  Camin_Sp := LREAL#20.0;
  Camin_Msd := LREAL#40.0;
  Camin_Ms := LREAL#1.0;
  Camin_Ss := LREAL#1.0;
  Camin_Mo := LREAL#0.0;
  Camin_So := LREAL#0.0;
  Camin_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
  Camin_Dir := _eMC_DIRECTION#_mcNoDirection;

  //기어 동작 시작(MC_GearIn)의 입력 파라메터에 값을 설정합니다.
  Gearin_RatN := UINT#1;
  Gearin_RatD := UINT#1;
  Gearin_RefTyp := _eMC_REFERENCE_TYPE#_mcCommand;
  Gearin_Acc := LREAL#0.0;
  Gearin_Dec := LREAL#0.0;

  //가감산 위치 결정(MC_CombineAxes)의 입력 파라메터에 값을 설정합니다.
  Combine_Cm := _eMC_COMBINE_MODE#_mcAddAxes;
  Combine_RefMas :=_eMC_REFERENCE_TYPE#_mcLatestCommand;
  Combine_RefAux :=_eMC_REFERENCE_TYPE#_mcLatestCommand;
  //입력 파라메터의 설정 완료 플래그를 유효화합니다.
  InitFlag := TRUE;
END_IF;


//접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
  Pwr1_En :=TRUE;
```

```
ELSE
   Pwr1_En :=FALSE;
END_IF;


//접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 3을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis003.DrvStatus.Ready=TRUE) THEN
   Pwr4_En :=TRUE;
ELSE
   Pwr4_En :=FALSE;
END_IF;


//축 0~축 3에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리 FaultHandler를 실행합니다.
//이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE)
 OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (MC_Axis002.MFaultLvl.Active=TRUE)
 OR (MC_Axis003.MFaultLvl.Active=TRUE) THEN
   FaultHandler();
END_IF;


//축 0이 서보 ON 상태에서 원점 미확정인 경우 축 0의 원점 복귀를 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex :=TRUE;
END_IF;


//축 3이 서보 ON 상태에서 원점 미확정인 경우 축 3의 원점 복귀를 실행합니다.
IF (Pwr4_Status=TRUE) AND (MC_Axis003.Details.Homed=FALSE) THEN
   Hm4_Ex :=TRUE;
END_IF;


//축 0의 원점 복귀가 완료되면 속도 제어를 실행합니다.
IF Hm1_D=TRUE THEN
   Vel_Ex :=TRUE;
END_IF;


//축 0이 목표 속도에 도달하면 기어 동작을 실행합니다.
IF Vel_InVel=TRUE THEN
   Gearin_Ex := TRUE;
END_IF;


//축 0이 목표 속도에 도달하면 캠 동작을 실행합니다.
IF Vel_InVel=TRUE THEN
   Camin_Ex := TRUE;
END_IF;


//기어 동작과 캠 동작이 제어 중으로 되면 가감산 위치 결정을 실행합니다.
IF (Gearin_Act=TRUE) AND (Camin_Act=TRUE) THEN
   Combine_Ex :=TRUE;
END_IF;
```

```
//축 0의 MC_Power
PWR1(
   Axis := MC_Axis000,
   Enable := Pwr1_En,
   Status => Pwr1_Status,
   Busy => Pwr1_Bsy,
   Error => Pwr1_Err,
   ErrorID => Pwr1_ErrID
);

//축 3의 MC_Power
PWR4(
   Axis := MC_Axis003,
   Enable := Pwr4_En,
   Status => Pwr4_Status,
   Busy => Pwr4_Bsy,
   Error => Pwr4_Err,
   ErrorID => Pwr4_ErrID
);

//축 0의 MC_Home
HM1(
   Axis := MC_Axis000,
   Execute := Hm1_Ex,
   Done => Hm1_D,
   Busy => Hm1_Bsy,
   CommandAborted => Hm1_Ca,
   Error => Hm1_Err,
   ErrorID => Hm1_ErrID
);

//축 3의 MC_Home
HM4(
   Axis := MC_Axis003,
   Execute := Hm4_Ex,
   Done => Hm4_D,
   Busy => Hm4_Bsy,
   CommandAborted => Hm4_Ca,
   Error => Hm4_Err,
   ErrorID => Hm4_ErrID
);

//MC_MoveVelocity
VEL(
   Axis := MC_Axis000,
   Execute := Vel_Ex,
   Velocity := Vel_Vel,
   Acceleration := Vel_Acc,
   Deceleration := Vel_Dec,
   Direction := Vel_Dir,
   InVelocity => Vel_InVel,
   Busy => Vel_Bsy,
   Active => Vel_Act,
   CommandAborted => Vel_Ca,
   Error => Vel_Err,
   ErrorID => Vel_ErrID
);

//MC_CamIn
```

```
CAMIN(
   Master := MC_Axis000,
   Slave := MC_Axis002,
   CamTable := CamProfile0,
   Execute := Camin_Ex,
   Periodic := Camin_Em,
   StartMode := Camin_Sm,
   StartPosition := Camin_Sp,
   MasterStartDistance := Camin_Msd,
   MasterScaling := Camin_Ms,
   SlaveScaling := Camin_Ss,
   MasterOffset := Camin_Mo,
   SlaveOffset := Camin_So,
   ReferenceType := Camin_Rt,
   Direction := Camin_Dir,
   InCam => Camin_InCam,
   InSync => Camin_InSync,
   EndOfProfile => Camin_Eop,
   Index => Camin_Index,
   Busy => Camin_Bsy,
   Active => Camin_Act,
   CommandAborted => Camin_Ca,
   Error => Camin_Err,
   ErrorID => Camin_ErrID
);

//MC_GearIn
GEARIN(
   Master := MC_Axis000,
   Slave := MC_Axis001,
   Execute := Gearin_Ex,
   RatioNumerator := Gearin_RatN,
   RatioDenominator := Gearin_RatD,
   ReferenceType := Gearin_RefTyp,
   Acceleration := Gearin_Acc,
   Deceleration := Gearin_Dec,
   InGear => Gearin_InGear,
   Busy => Gearin_Bsy,
   Active => Gearin_Act,
   CommandAborted => Gearin_Ca,
   Error => Gearin_Err,
   ErrorID => Gearin_ErrID
);

//MC_CombineAxes
COMBINE(
   Master := MC_Axis001,
   Auxiliary := MC_Axis002,
   Slave := MC_Axis003,
   Execute := Combine_Ex,
   CombineMode := Combine_CM,
   ReferenceTypeMaster := Combine_RefMas,
   ReferenceTypeAuxiliary := Combine_RefAux,
   InCombination => Combine_InComb,
   Busy => Combine_Bsy,
   Active => Combine_Act,
   CommandAborted => Combine_Ca,
   Error => Combine_Err,
   ErrorID => Combine_ErrID
);
```

10-2-14 캠 동작 시에 주축의 위상 보정하기

캠 동작의 종축은 속도 제어로 동작 중인 주축에 동기하여 동작합니다. StartOn을 TRUE로 하면 MC_Phasing(주축 상대값 위상 보정) 명령으로 주축의 위상을 보정합니다. 종축은 보정된 위상에 대해서 동기를 실행합니다.

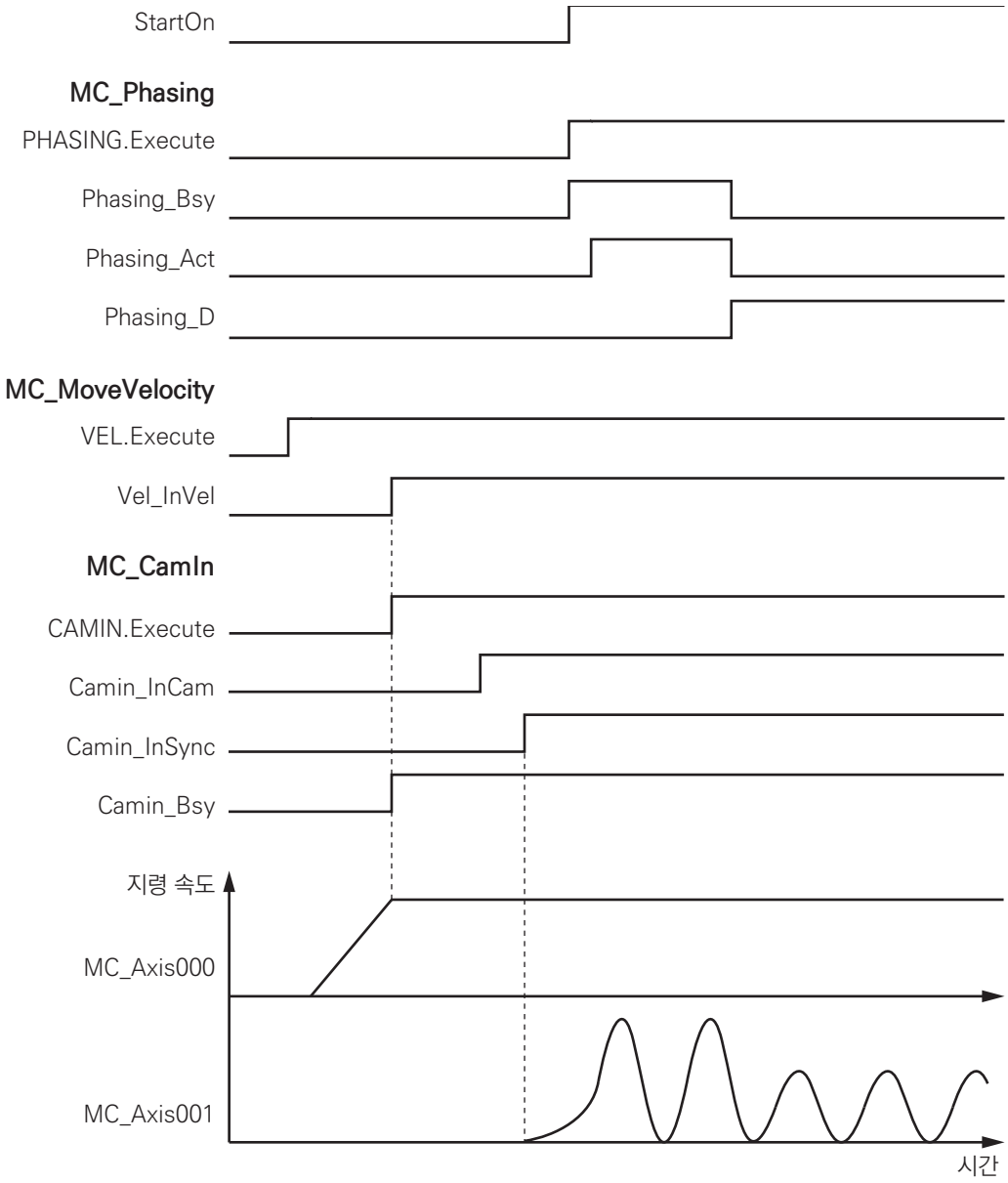
프로그램에서 사용하는 주요 변수

명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 0이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	—	축 1의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 1에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	—	캠 데이터 변수입니다. *1
Pwr1_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
StartOn	BOOL	FALSE	주축의 위상 보정을 시작하기 위한 변수입니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON합니다.
Camin_Ex	BOOL	FALSE	MC_CamIn(캠 동작 시작) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.
Vel_Ex	BOOL	FALSE	MC_MoveVelocity(속도 제어) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.
Phasing_Ex	BOOL	FALSE	MC_Phasing(주축 상대값 위상 보정) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.

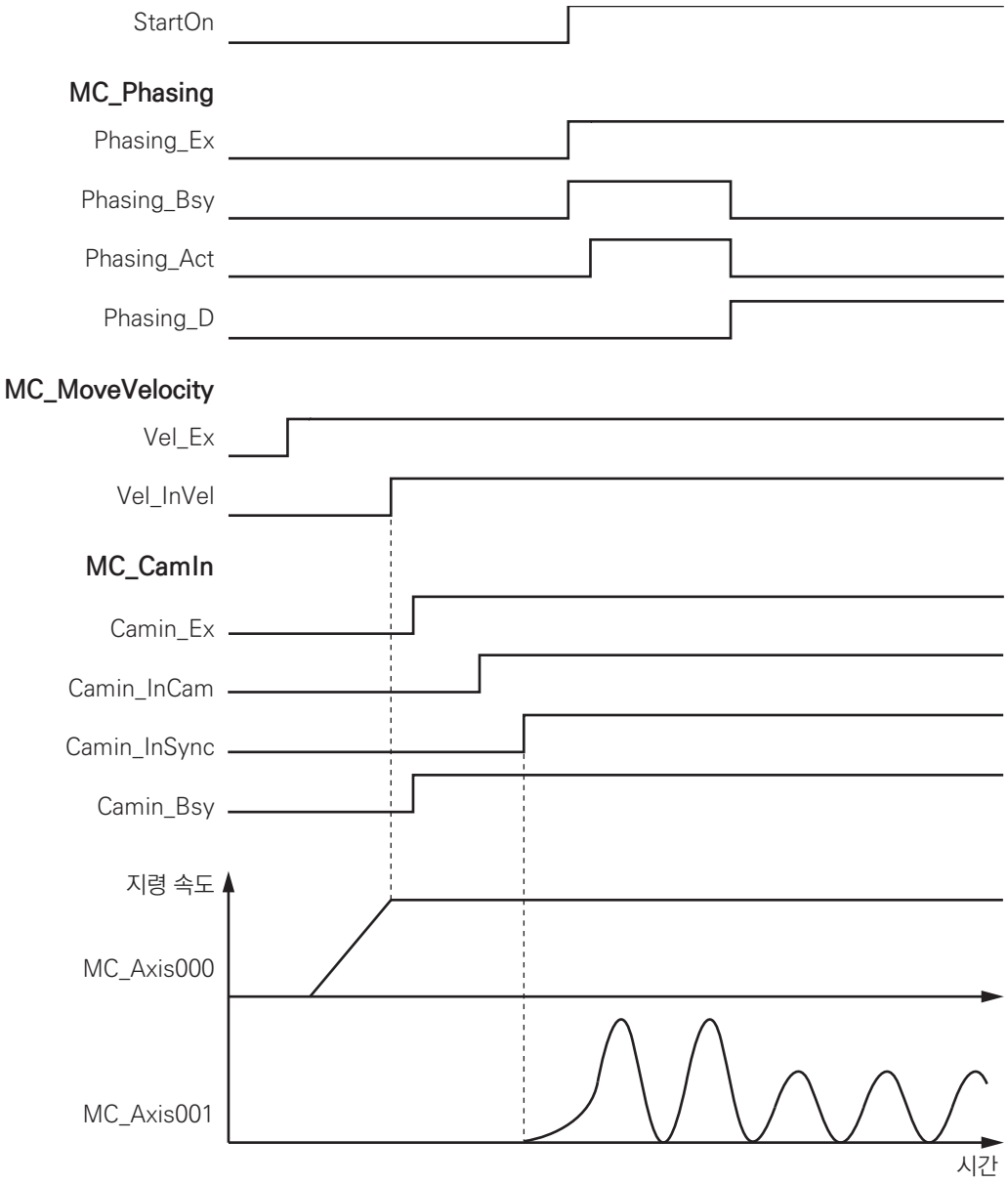
*1. 배열의 요소 수 ARRAY[0..N]는 Sysmac Studio의 캠 에디터 기능으로 설정합니다. 본 샘플 프로그램에서는 [0~360]로 합니다.

타이밍 차트

● 래더도 프로그램의 경우

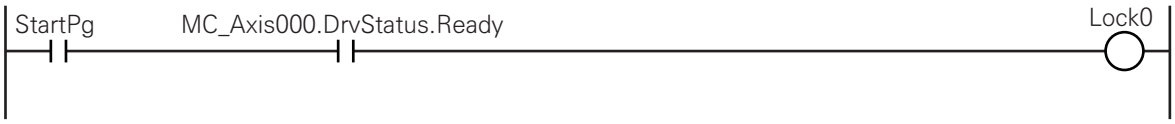


● ST 프로그램의 경우

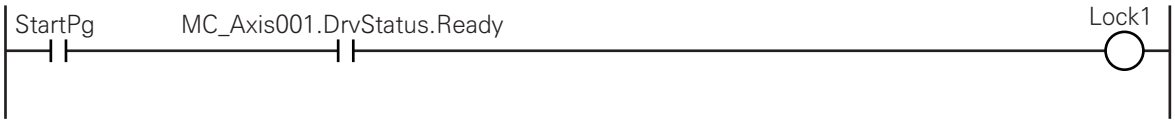

StartOn
MC_Phasing
Phasing_Ex
Phasing_Bsy
Phasing_Act
Phasing_D
MC_MoveVelocity
Vel_Ex
Vel_InVel
MC_CamIn
Camin_Ex
Camin_InCam
Camin_InSync
Camin_Bsy
지령 속도
MC_Axis000
MC_Axis001
시간

래더도 프로그램

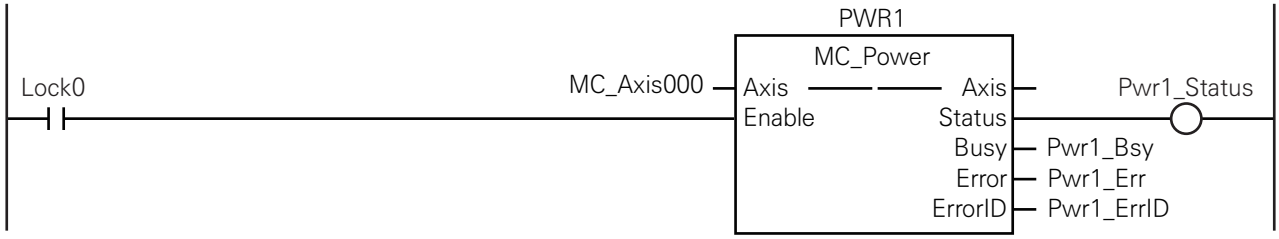
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



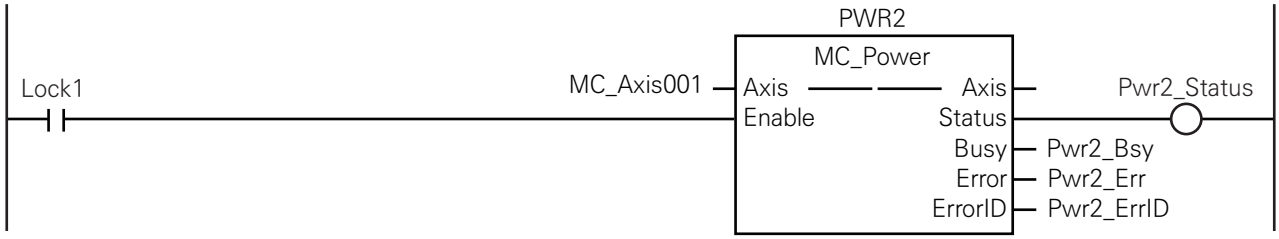
접점 StartPg가 TRUE이고 축 1의 서보 드라이버가 서보 준비 상태인지 확인합니다.



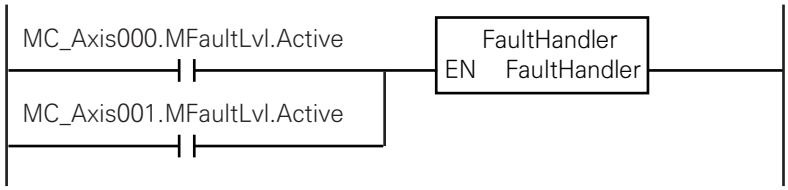
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



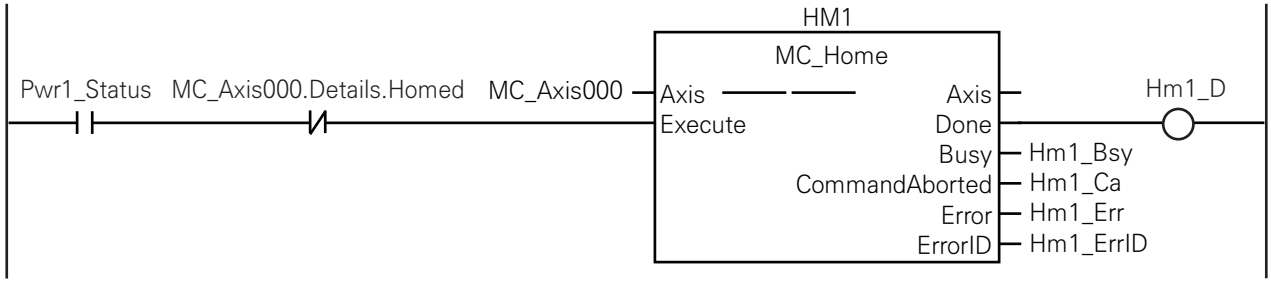
축 1의 서보 드라이버가 서보 준비 상태이면 축 1을 서버 ON합니다.



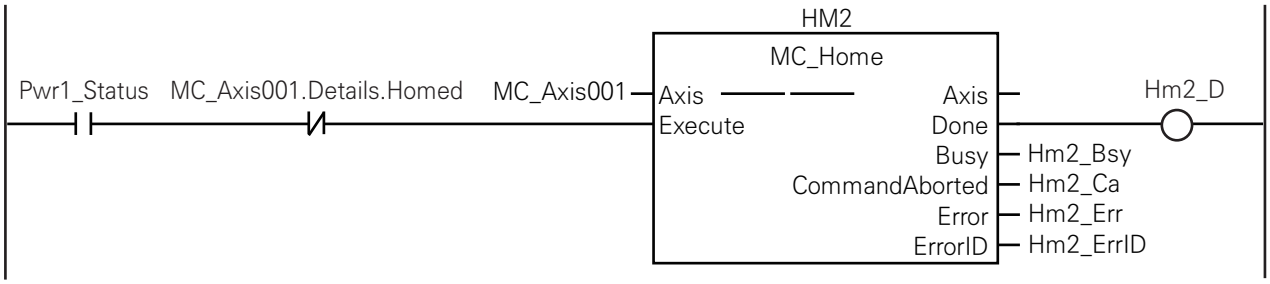
축 그룹 혹은 구성 축에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



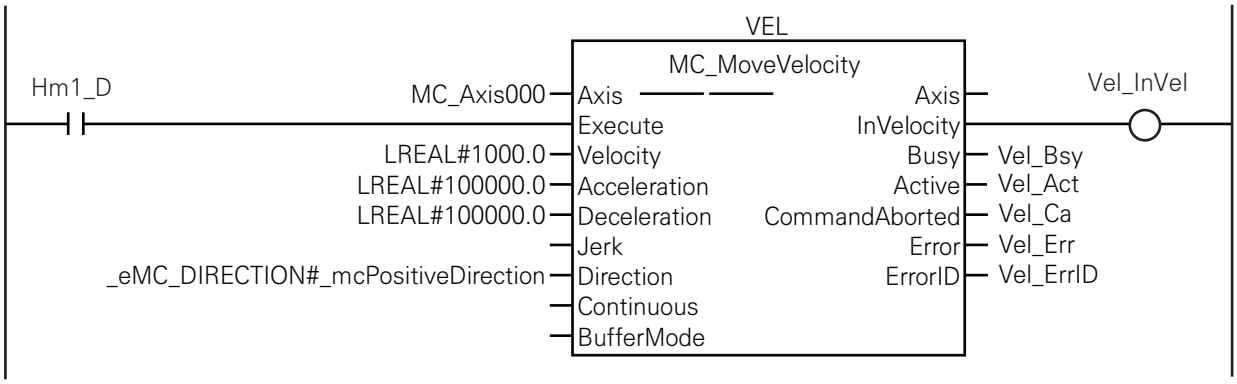
축 00이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



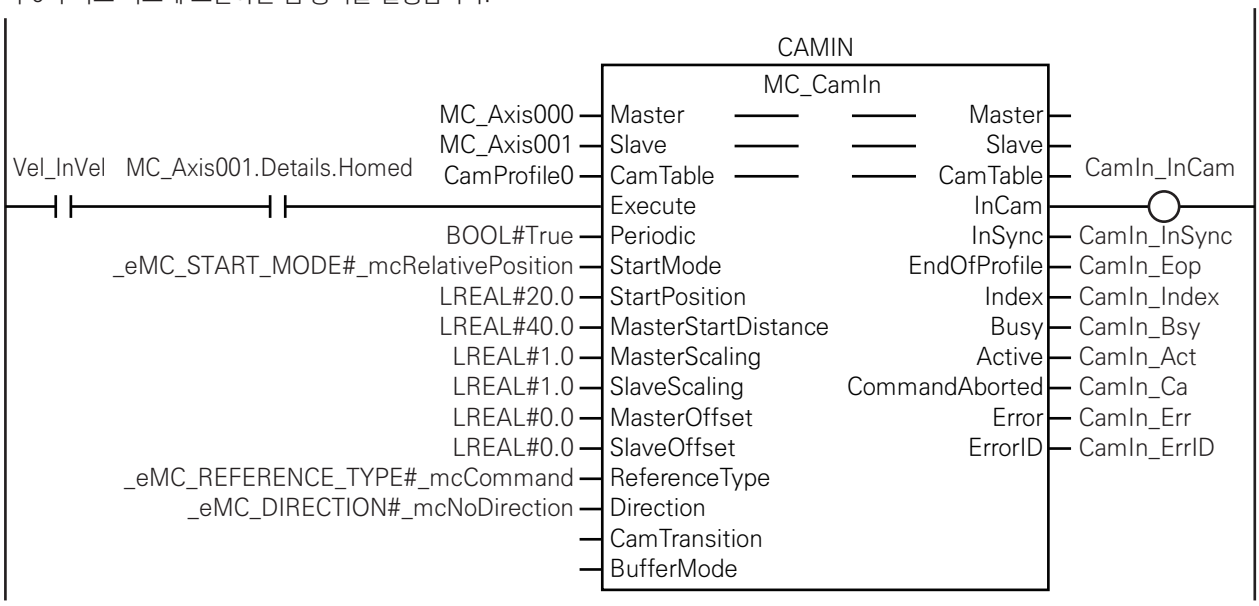
축 1이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



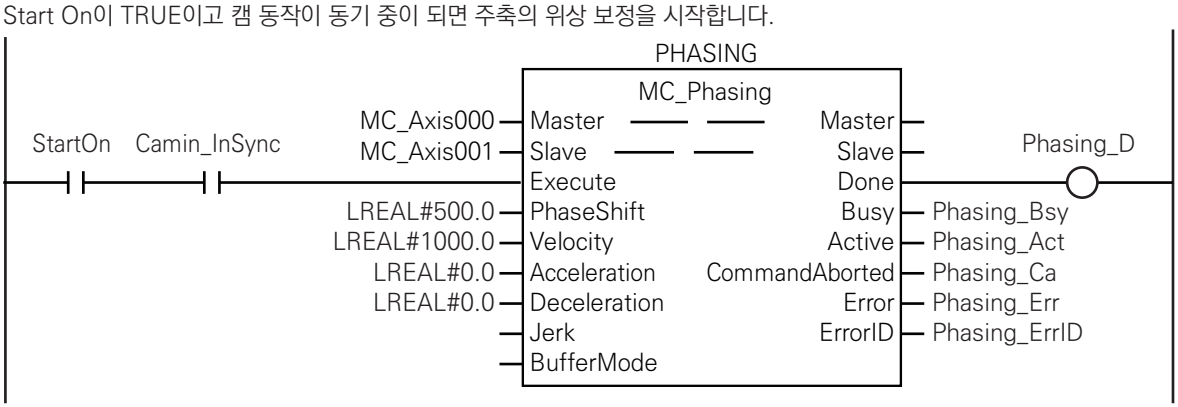
축 0의 원점 복귀가 완료되면 속도 제어를 실행합니다.



축 0이 목표 속도에 도달하면 캠 동작을 실행합니다.



Start On이 TRUE이고 캠 동작이 동기 중이 되면 주축의 위상 보정을 시작합니다.



ST 프로그램

```
//동작 명령의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag=FALSE THEN
   //속도 제어(MC_MoveVelocity)의 입력 파라메터에 값을 설정합니다.
   Vel_Vel := LREAL#1000.0;
   Vel_Acc := LREAL#100000.0;
   Vel_Dec := LREAL#100000.0;
   Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

   //주축 상대값 위상 보정(MC_Phasing)의 입력 파라메터에 값을 설정합니다.
   Phasing_Ps := LREAL#500.0;
   Phasing_Vel := LREAL#1000.0;
   Phasing_Acc := LREAL#0.0;
   Phasing_Dec := LREAL#0.0;

   //캠 동작 시작(MC_CamIn)의 입력 파라메터에 값을 설정합니다.
   Camin_Em := TRUE;
   Camin_Sm := _eMC_START_MODE#_mcRelativePosition;
   Camin_Sp := LREAL#20.0;
   Camin_Msd := LREAL#40.0;
   Camin_Ms := LREAL#1.0;
   Camin_Ss := LREAL#1.0;
   Camin_Mo := LREAL#0.0;
   Camin_So := LREAL#0.0;
   Camin_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
   Camin_Dir := _eMC_DIRECTION#_mcNoDirection;

    //입력 파라메터의 설정 완료 플래그를 유효화합니다.
   InitFlag := TRUE;
END_IF;


// 접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr1_En:=TRUE;
ELSE
   Pwr1_En:=FALSE;
END_IF;


//접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 1을 서보 ON합니다.
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
   Pwr2_En:=TRUE;
ELSE
   Pwr2_En:=FALSE;
END_IF;


//축 0~축 1에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리 FaultHandler를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
   FaultHandler();
END_IF;
```

```
// 축 0이 서보 ON 상태에서 원점 미확정인 경우 축 0의 원점 복귀를 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex:=TRUE;
END_IF;


// 축 1이 서보 ON 상태에서 원점 미확정인 경우 축 1의 원점 복귀를 실행합니다.
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
   Hm2_Ex:=TRUE;
END_IF;


// 축 0의 원점 복귀가 완료되면 속도 제어를 실행합니다.
IF Hm1_D=TRUE THEN
   Vel_Ex :=TRUE;
END_IF;


// 축 0이 목표 속도에 도달하고 축 1이 원점 확정되어 있으면 동작을 실행합니다.
IF (Vel_InVel=TRUE) AND (MC_Axis001.Details.Homed=TRUE) THEN
   Camin_Ex := TRUE;
END_IF;


// Start On이 TRUE이고 캠 동작이 동기 중이 되면 주축의 위상 보정을 시작합니다.
IF (StartOn=TRUE) AND (Camin_InSync=TRUE) THEN
   Phasing_Ex :=TRUE;
END_IF;

//축 0의 MC_Power
PWR1(
   Axis := MC_Axis000,
   Enable := Pwr1_En,
   Status => Pwr1_Status,
   Busy => Pwr1_Bsy,
   Error => Pwr1_Err,
   ErrorID => Pwr1_ErrID
);

//축 1의 MC_Power
PWR2(
   Axis := MC_Axis001,
   Enable := Pwr2_En,
   Status => Pwr2_Status,
   Busy => Pwr2_Bsy,
   Error => Pwr2_Err,
   ErrorID => Pwr2_ErrID
);

//축 0의 MC_Home
HM1(
   Axis := MC_Axis000,
   Execute := Hm1_Ex,
   Done => Hm1_D,
   Busy => Hm1_Bsy,
   CommandAborted => Hm1_Ca,
   Error => Hm1_Err,
   ErrorID => Hm1_ErrID
);
```

```
//축 1의 MC_Home
HM2(
   Axis := MC_Axis001,
   Execute := Hm2_Ex,
   Done => Hm2_D,
   Busy => Hm2_Bsy,
   CommandAborted => Hm2_Ca,
   Error => Hm2_Err,
   ErrorID => Hm2_ErrID
);

//MC_MoveVelocity
VEL(
   Axis := MC_Axis000,
   Execute := Vel_Ex,
   Velocity := Vel_Vel,
   Acceleration := Vel_Acc,
   Deceleration := Vel_Dec,
   Direction := Vel_Dir,
   InVelocity => Vel_InVel,
   Busy => Vel_Bsy,
   Active => Vel_Act,
   CommandAborted => Vel_Ca,
   Error => Vel_Err,
   ErrorID => Vel_ErrID
);

//MC_Phasing
PHASING(
   Master := MC_Axis000,
   Slave := MC_Axis001,
   Execute := Phasing_Ex,
   PhaseShift := Phasing_Ps,
   Velocity := Phasing_Vel,
   Acceleration := Phasing_Acc,
   Deceleration := Phasing_Dec,
   Done => Phasing_D,
   Busy => Phasing_Bsy,
   Active => Phasing_Act,
   CommandAborted => Phasing_Ca,
   Error => Phasing_Err,
   ErrorID => Phasing_ErrID
);

//MC_CamIn
CAMIN(
   Master := MC_Axis000,
   Slave := MC_Axis001,
   CamTable := CamProfile0,
   Execute := Camin_Ex,
   Periodic := Camin_Em,
   StartMode := Camin_Sm,
   StartPosition := Camin_Sp,
   MasterStartDistance := Camin_Msd,
   MasterScaling := Camin_Ms,
   SlaveScaling := Camin_Ss,
   MasterOffset := Camin_Mo,
   SlaveOffset := Camin_So,
```

```
    ReferenceType := Camin_Rt,
    Direction := Camin_Dir,
    InCam => Camin_InCam,
    InSync => Camin_InSync,
    EndOfProfile => Camin_Eop,
    Index => Camin_Index,
    Busy => Camin_Bsy,
    Active => Camin_Act,
    CommandAborted => Camin_Ca,
    Error => Camin_Err,
    ErrorID => Camin_ErrID
);
```

10-2-15 속도 제어 중에 현재 위치 변경하기

속도 제어로 동작 중인 축의 지령 현재 위치와 피드백 현재 위치의 절대 위치를 변경합니다.

사용상의 주의

- 동작 중인 축에 대해 MC_SetPosition(현재 위치 변경) 명령을 기동하면 기동한 다음부터 실제로 현재 위치가 변경될 때까지의 이동 거리가 오차로 남습니다.
- MC_SetPosition(현재 위치 변경) 명령을 기동하면 원점이 확정되지 않습니다.

축 파라메터 설정

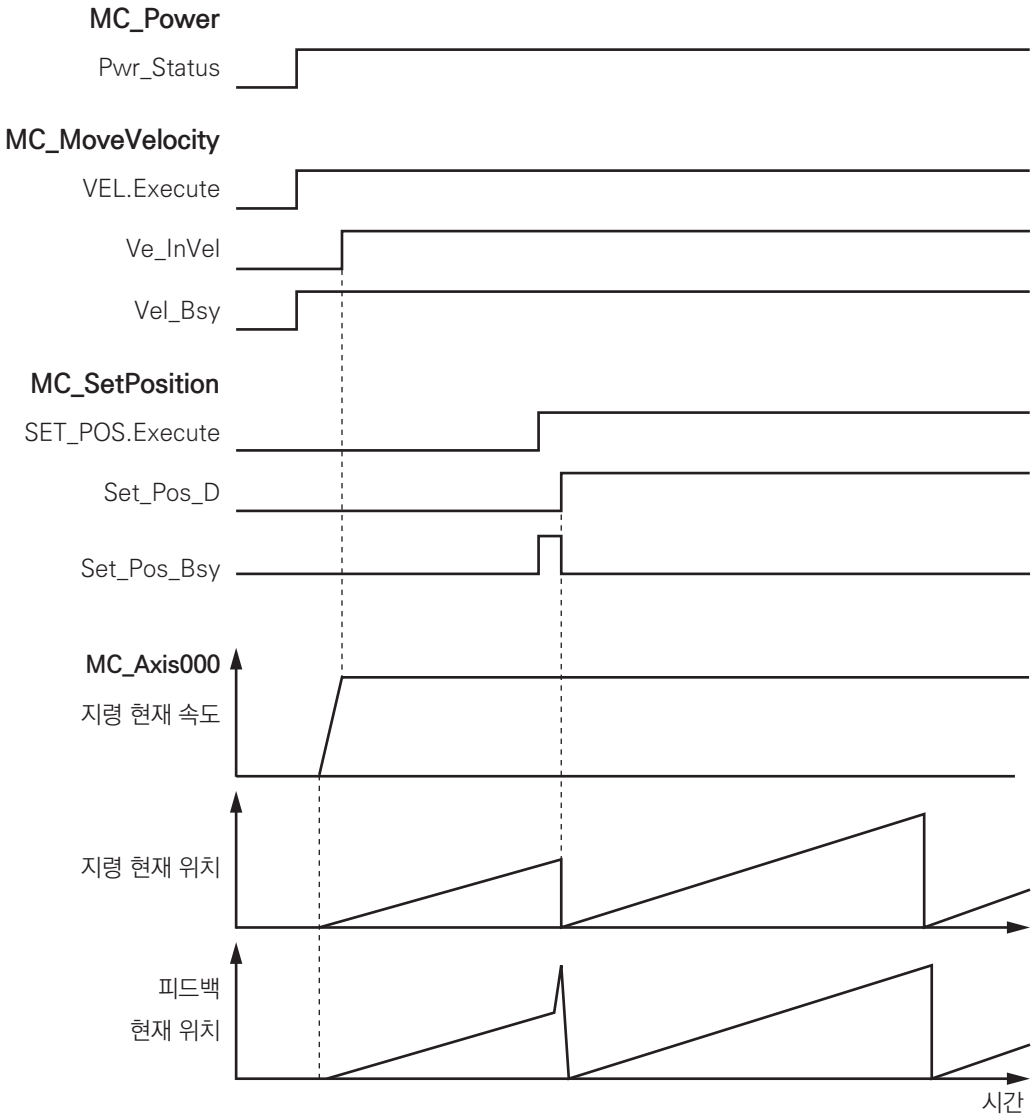
설정 항목	설정 내용	설명
카운트 모드	로터리 모드	위치의 카운트 모드를 로터리 모드로 설정합니다.
링 카운터 상한 설정값	360	링 카운터의 상한값을 「360」으로 설정합니다.
링 카운터 하한 설정값	0	링 카운터의 하한값을 「0」으로 설정합니다.
원점 복귀 동작	원점 프리셋	원점 프리셋으로 원점을 확정합니다.

프로그램에서 사용하는 주요 변수

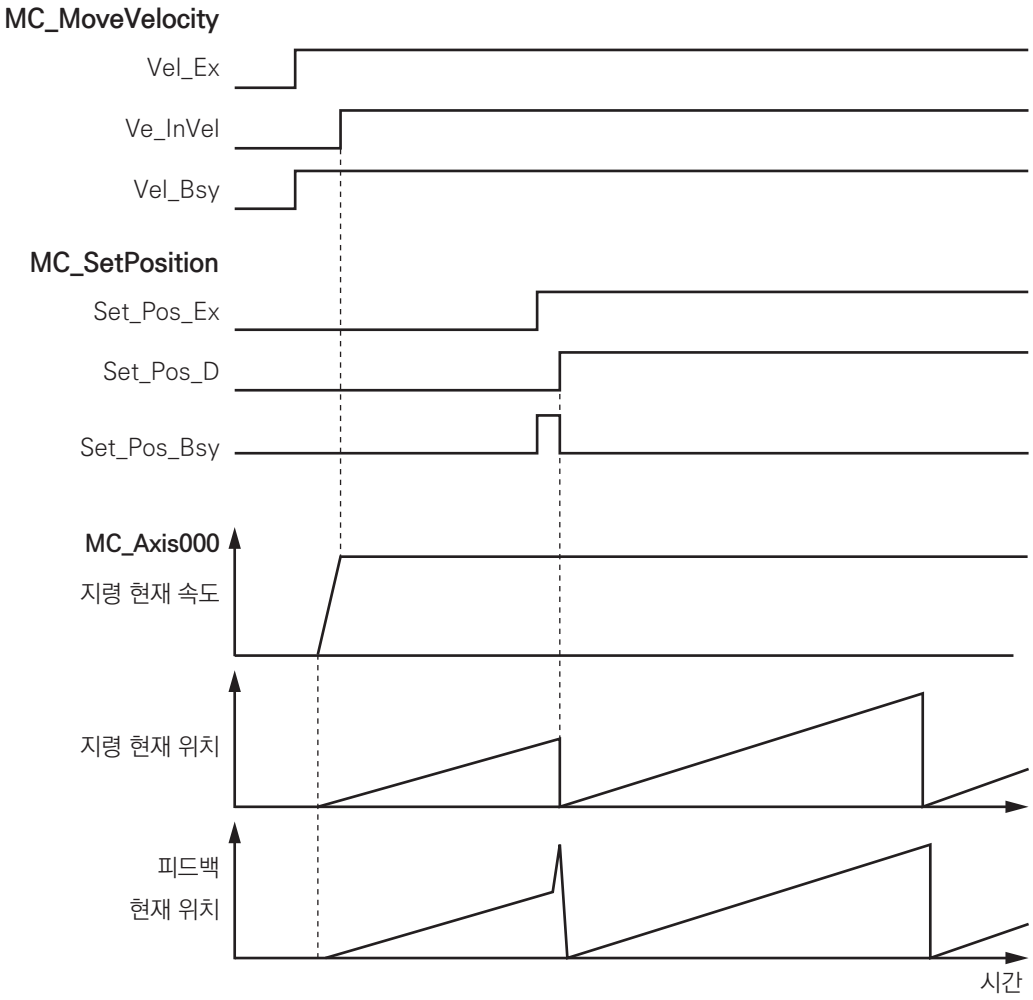
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 0에서 경도 폴트 레벨의 이상이 발생한 경우에 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
StartSetPos	BOOL	FALSE	현재 위치를 변경하기 위한 외부 버튼입니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON합니다.
Vel_Ex	BOOL	FALSE	MC_MoveVelocity(속도 제어) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.
SetPos_Ex	BOOL	FALSE	MC_SetPosition(현재 위치 변경) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.

타이밍 차트

● 래더도 프로그램의 경우

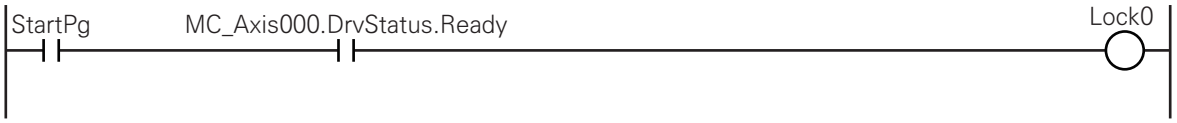


● ST 프로그램의 경우

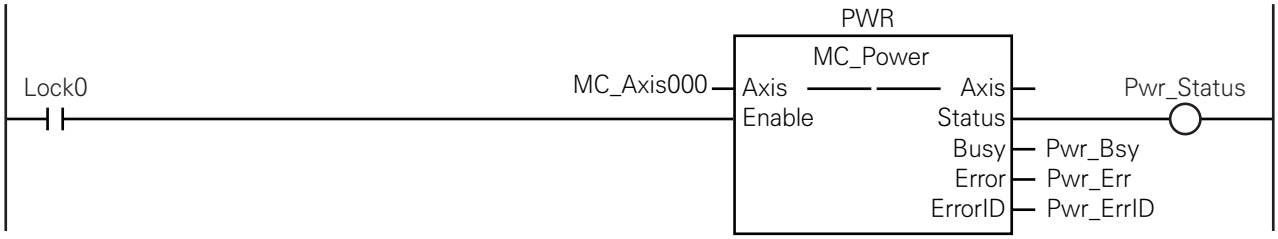

MC_MoveVelocity
Vel_Ex
Ve_InVel
Vel_Bsy
MC_SetPosition
Set_Pos_Ex
Set_Pos_D
Set_Pos_Bsy
MC_Axis000
지령 현재 속도
지령 현재 위치
피드백
현재 위치
시간

래더도 프로그램

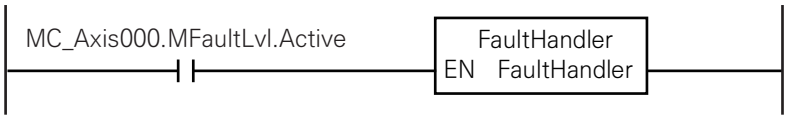
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



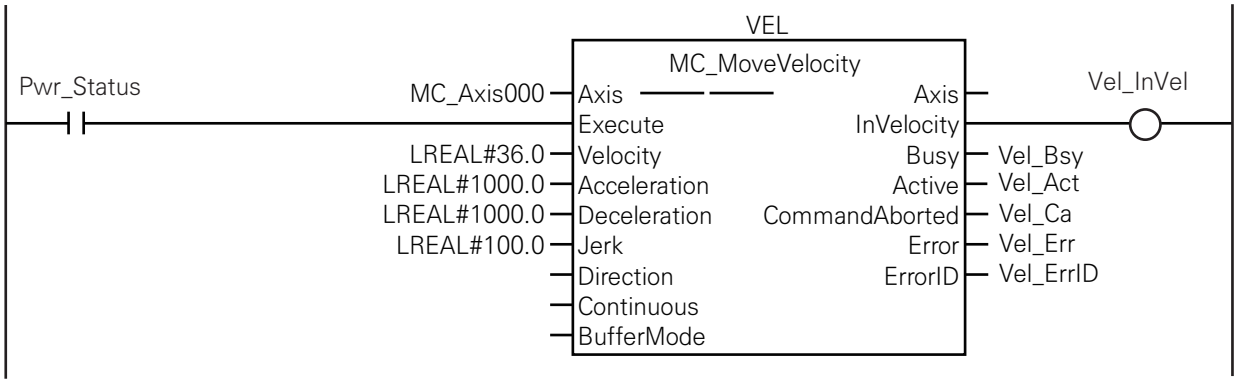
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



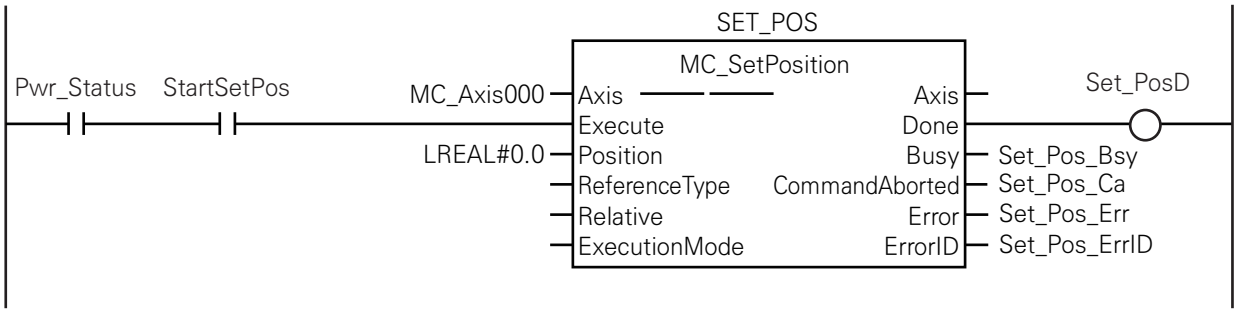
축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



축 00이 서보 ON되면 속도 제어를 실행합니다.



서보 ON 상태에서 접점 StartSetPos가 TRUE가 되면 현재 위치 변경을 실행합니다.



ST 프로그램

```
//명령의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag=FALSE THEN
   //속도 제어(MC_MoveVelocity)의 입력 파라메터에 값을 설정합니다.
   Vel_Vel := LREAL#36.0;
   Vel_Acc := LREAL#1000.0;
   Vel_Dec := LREAL#1000.0;
   Vel_Jrk := LREAL#100.0;

   //현재 위치 변경(MC_SetPosition)의 입력 파라메터에 값을 설정합니다.
   Set_Pos_Pos := LREAL#0.0;

   //입력 파라메터의 설정 완료 플래그를 유효화합니다.
   InitFlag :=TRUE;
END_IF;


// 접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
//서보 준비 상태가 되지 않은 경우에는 서보 OFF로 합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr_En :=TRUE;
ELSE
   Pwr_En :=FALSE;
END_IF;


//축 0에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
   FaultHandler();
END_IF;


//축 0이 서보 ON되면 축 0의 속도 제어를 실행합니다.
IF Pwr_Status=TRUE THEN
   Vel_Ex :=TRUE;
END_IF;


//서보 ON 상태에서 접점 StartSetPos가 TRUE가 되면 현재 위치 변경을 실행합니다.
IF (Pwr_Status=TRUE) AND (StartSetPos=TRUE) THEN
   Set_Pos_Ex :=TRUE;
END_IF;

//MC_Power
PWR(
   Axis := MC_Axis000,
   Enable := Pwr_En,
   Status => Pwr_Status,
   Busy => Pwr_Bsy,
   Error => Pwr_Err,
   ErrorID => Pwr_ErrID
);

//MC_MC_MoveVelocity
VEL(
   Axis := MC_Axis000,
```

```
    Execute := Vel_Ex,
    Velocity := Vel_Vel,
    Acceleration := Vel_Acc,
    Deceleration := Vel_Dec,
    Jerk := Vel_Jrk,
    InVelocity => Vel_InVel,
    Busy => Vel_Bsy,
    Active => Vel_Act,
    CommandAborted => Vel_Ca,
    Error => Vel_Err,
    ErrorID => Vel_ErrID
);

//MC_SetPosition
SET_POS(
    Axis := MC_Axis000,
    Execute := Set_Pos_Ex,
    Position := Set_Pos_Pos,
    Done => Set_Pos_D,
    Busy => Set_Pos_Bsy,
    CommandAborted => Set_Pos_Ca,
    Error => Set_Pos_Err,
    ErrorID => Set_Pos_ErrID
);
```

10-2-16 캠 데이터 변수의 변경과 캠 테이블 저장

Sysmac Studio의 캠 에디터에서 작성한 캠 데이터 변수의 값을 사용자 프로그램으로 변경합니다.
위상 0도~180도까지는 변위를 2배로 하고 181도~360도에서는 변위를 0.5배로 합니다.

캠 데이터의 변경이 끝나면 MC_SaveCamTable(캠 테이블 저장) 명령을 이용하여 CPU 유니트의 불휘발성 메모리에 저장합니다.
저장이 끝나면 MC_CamIn(캠 동작 시작) 명령을 이용하여 캠 동작을 시작합니다.

사용상의 주의

- 위상이 오름차순이 아닌 경우 MC_CamIn(캠 동작 시작) 명령 기동 시 이상이 발생합니다.
  이 샘플 프로그램에서는 위상의 오름차순 체크를 하지 않습니다. 체크를 할 경우에는 MC_SetCamTableProperty(캠 테이블 속성(Property) 갱신) 명령을 실행해 주십시오.
- CPU 유니트의 불휘발성 메모리는 기록 횟수에 제한이 있습니다. 캠 테이블의 저장은 필요한 경우에만 실행해 주십시오.
- MC_SaveCamTable(캠 테이블 저장) 명령으로 저장하기 전에 CPU 유니트의 전원을 OFF하면 사용자 프로그램에서 변경한 캠 데이터 변수가 변경 전의 내용으로 돌아갑니다.

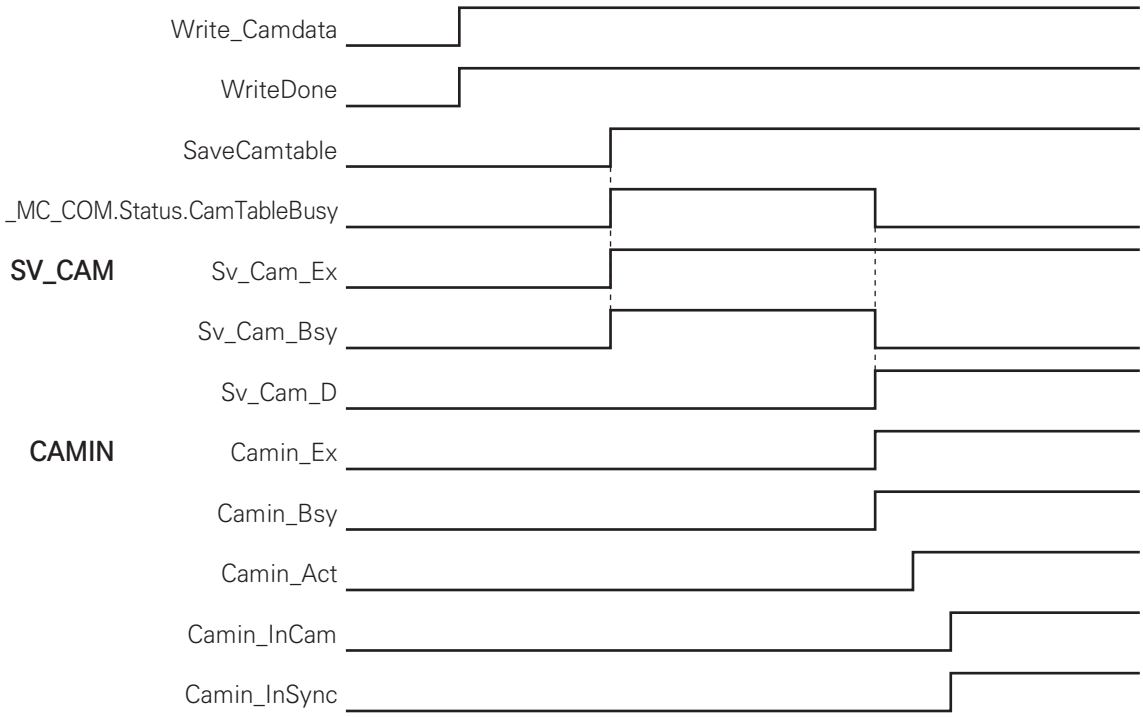
프로그램에서 사용하는 주요 변수

명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
MC_Axis001	_sAXIS_REF	—	축 1의 축 변수입니다.
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	—	캠 데이터 변수입니다. *1
Pwr1_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 명령의 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON일 때 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT 통신의 프로세스 데이터 통신이 정상으로 실행 중이면 서보 ON합니다.
WriteCamdata	BOOL	FALSE	캠 데이터 변경을 기동하기 위한 변수입니다. 편집을 시작할 때 TRUE로 해 주십시오.
WriteDone	BOOL	FALSE	캠 데이터 변경이 끝났음을 나타내기 위한 변수입니다. 변경이 끝났을 때 TRUE로 해 주십시오.
SaveCamtable	BOOL	FALSE	캠 테이블 저장을 기동하기 위한 변수입니다.
_MC_COM.Status.CamTableBusy	BOOL	FALSE	캠 테이블 저장 실행 중을 나타내는 시스템 정의 변수입니다.
Sv_Cam_Ex	BOOL	FALSE	MC_SaveCamTable(캠 테이블 저장) 명령을 기동하기 위한 변수입니다.
Camin_Ex	BOOL	FALSE	MC_CamIn(캠 동작 시작) 명령을 기동하기 위한 변수입니다. ST 프로그램에서 사용합니다.

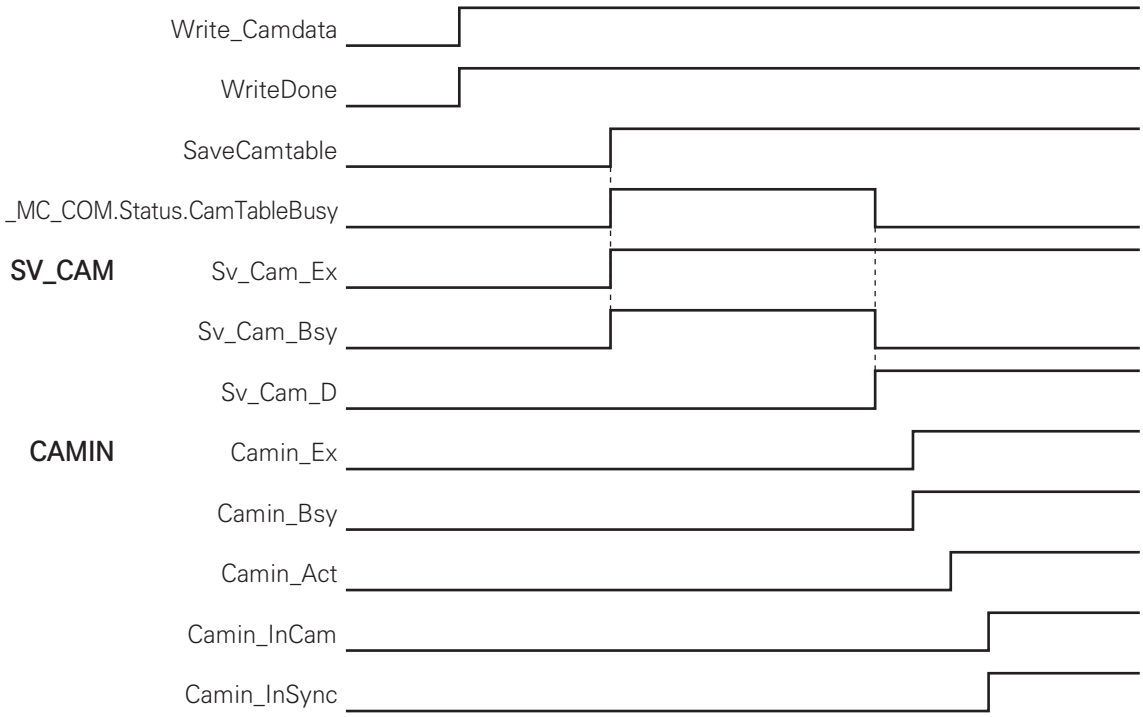
*1. 배열의 요소 수 ARRAY[0..N]는 Sysmac Studio의 캠 에디터 기능으로 설정합니다. 본 샘플 프로그램에서는 [0~360]로 합니다.

타이밍 차트

● 래더도 프로그램의 경우

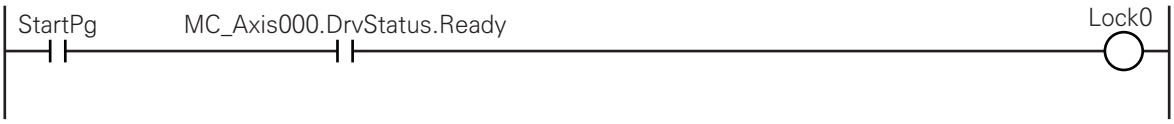


● ST 프로그램의 경우

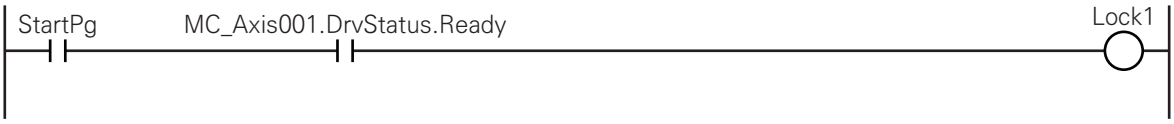


래더도 프로그램

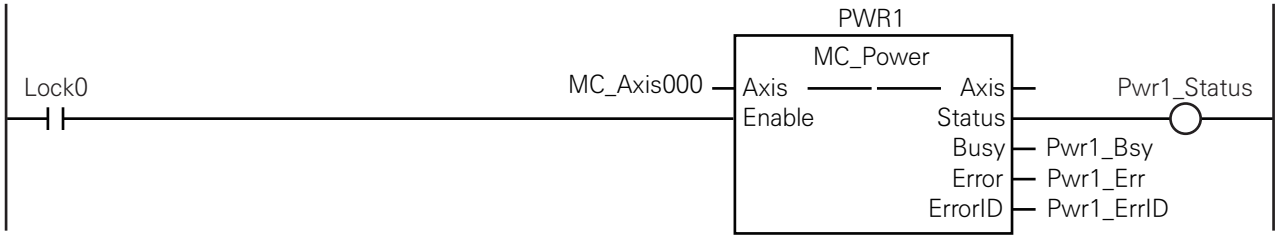
접점 StartPg가 TRUE이고 축 0의 서보 드라이버가 서보 준비 상태인지 확인합니다.



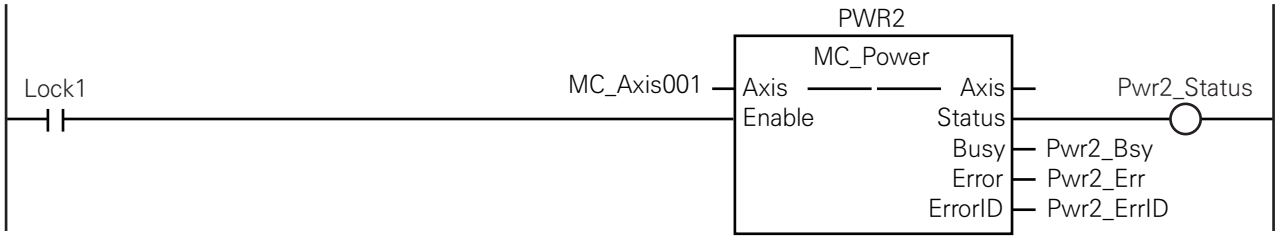
접점 StartPg가 TRUE이고 축 1의 서보 드라이버가 서보 준비 상태인지 확인합니다.



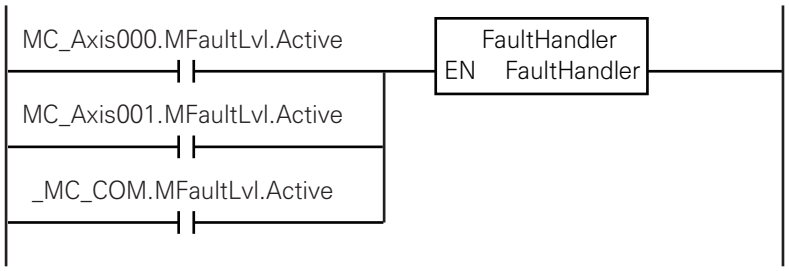
축 0의 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.



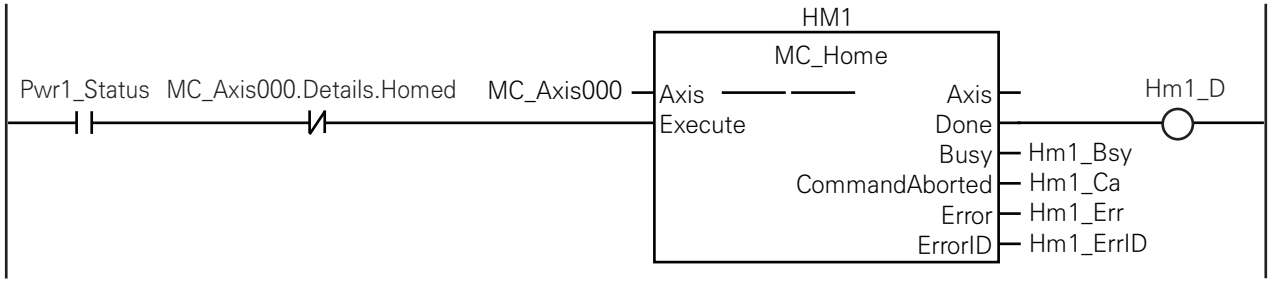
축 1의 서보 드라이버가 서보 준비 상태이면 축 1을 서버 ON합니다.



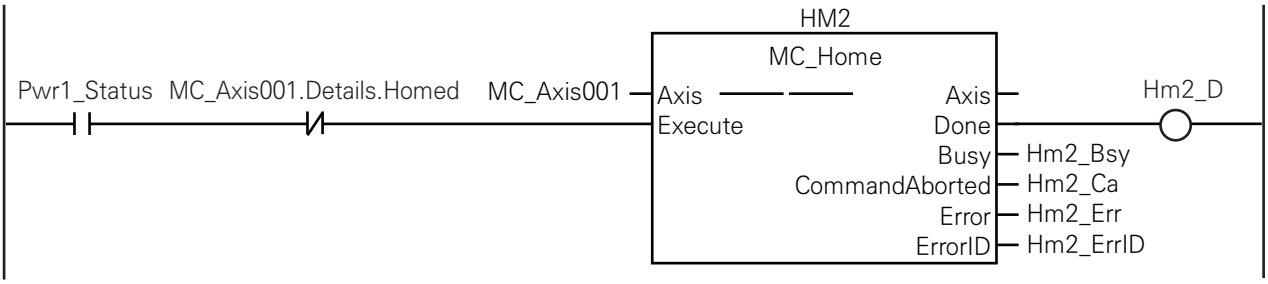
MC 공통 혹은 축에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



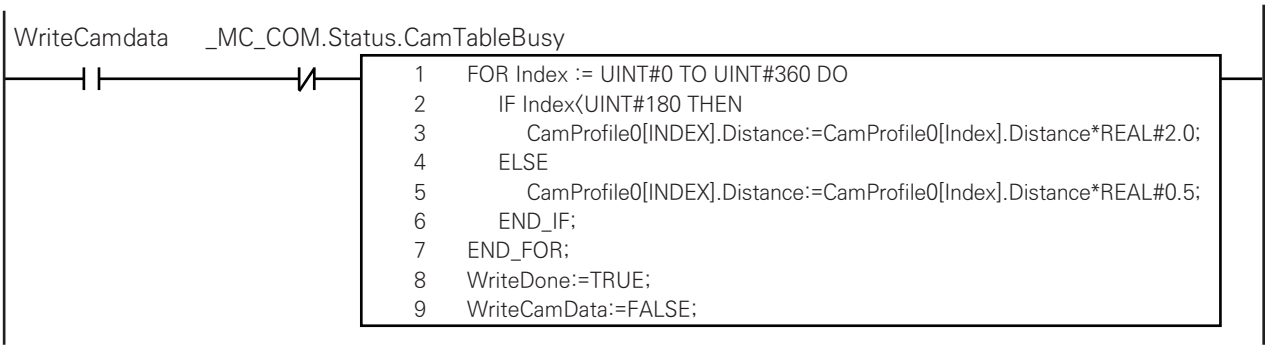
축 0이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



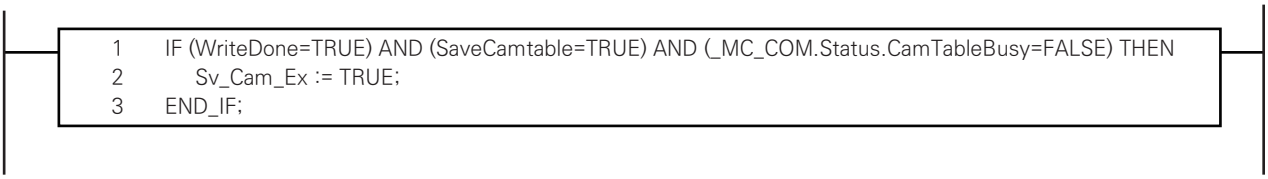
축 1이 서보 ON 상태에서 원점 미확정인 경우 원점 복귀를 실행합니다.



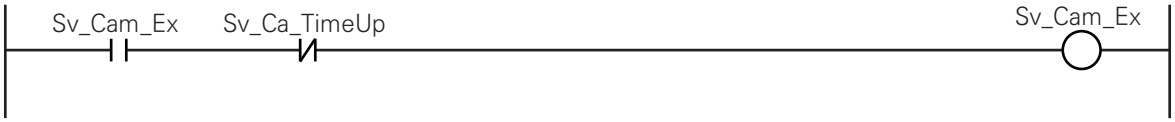
WriteCamData가 TRUE이고 캠 테이블 파일 저장이 실행 중이 아닌 경우 캠 데이터 변수값을 변경합니다.
위상 0도~180도까지는 변위를 2배로 하고 181도~360도에서는 변위를 0.5배로 합니다.
변경이 완료되었을 때 WriteDone을 TRUE로 합니다.



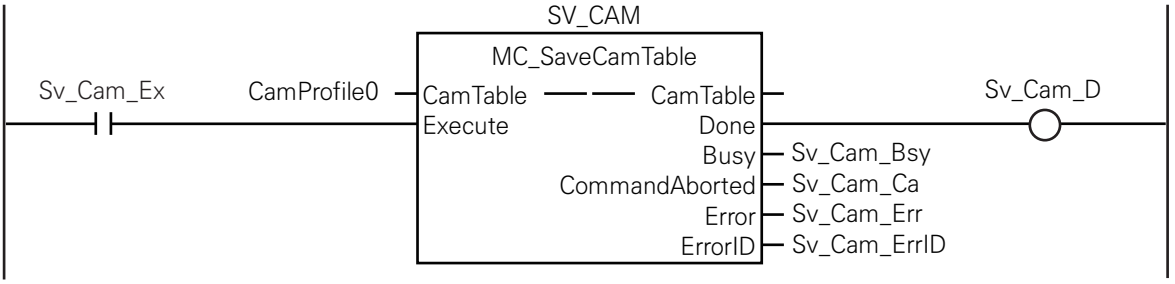
캠 데이터 변수의 변경이 완료되어 SaveCamtable이 TRUE이고 캠 테이블 파일 저장이 실행 중이 아닌 경우
Sv_Cam_Ex를 TRUE로 합니다.



Sv_Ca_TimeUp이 TRUE인 경우 Sv_Cam_Ex를 FALSE로 합니다.
Sv_Cam_Ex가 FALSE가 되면 Sv_Ca_TimeUp이 FALSE가 되고 Sv_Cam_Ex를 TRUE로 합니다.
MC_SaveCamTable(캠 테이블 저장) 명령을 재실행합니다.



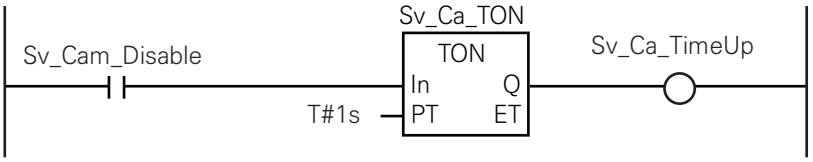
Sv_Cam_Ex가 TRUE가 되면 MC_SaveCamTable(캠 테이블 저장) 명령을 실행합니다.



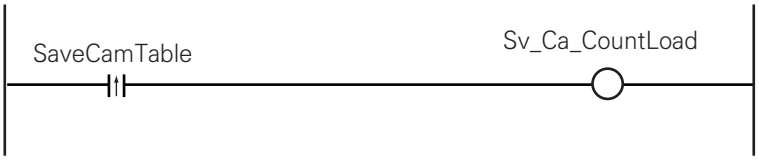
Sv_Ca_CountUp이 FALSE이고 캠 테이블 저장 실행 불가가 발생한 경우
Sv_Cam_Disable을 TRUE로 합니다.



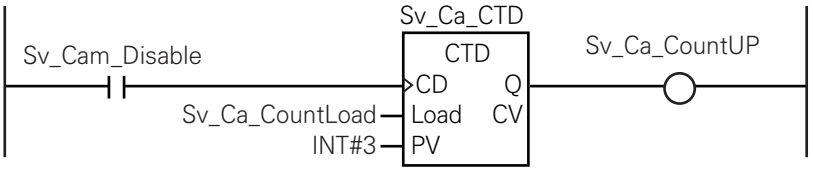
캠 테이블 저장 실행 불가가 발생하고 1초 후에 Sv_Ca_TimeUp을 TRUE로 합니다.
Sv_Ca_TimeUp이 TRUE가 되면 Sv_Cam_Ex가 FALSE가 됩니다.



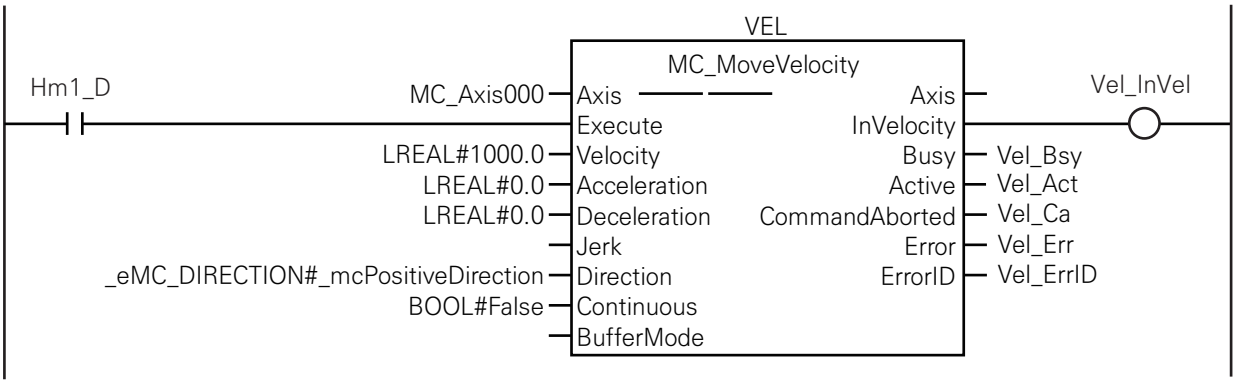
캠 테이블 저장 실행 시에 Sv_Ca_CountLoad를 1주기 동안 TRUE로 합니다.
Sv_Ca_CountLoad가 TRUE가 되면 재실행 카운터를 리셋합니다.



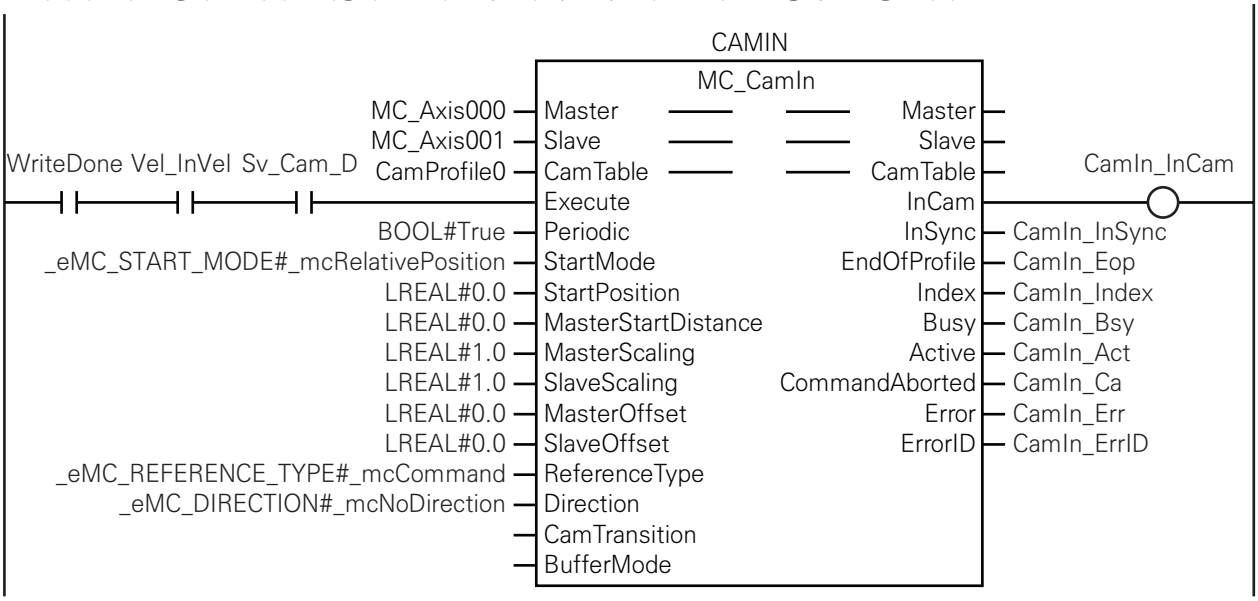
캠 테이블 저장 실행 불가가 3회 발생한 경우에 Sv_Ca_CountUp을 TRUE로 합니다.
Sv_Ca_CountUp이 TRUE가 되면 Sv_Cam_Disable을 FALSE로 합니다.
MC_SaveCamTable(캠 테이블 저장) 명령의 재실행을 종료합니다.



축 0의 원점 복귀가 완료되면 속도 제어를 실행합니다.



캠 데이터 변수 변경과 캠 테이블 저장이 완료되고 축 0이 목표 속도에 도달하면 캠 동작을 실행합니다.



ST 프로그램

```
//명령의 입력 파라메터를 설정하지 않은 경우에는 목표값 등을 설정합니다.
IF InitFlag=FALSE THEN
   //속도 제어(MC_MoveVelocity)의 입력 파라메터에 값을 설정합니다.
   Vel_Vel := LREAL#1000.0;
   Vel_Acc := LREAL#0.0;
   Vel_Dec := LREAL#0.0;
   Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;

   //캠 동작 시작(MC_CamIn)의 입력 파라메터에 값을 설정합니다.
   Camin_Em := TRUE;
   Camin_Sm := _eMC_START_MODE#_mcRelativePosition;
   Camin_Sp := LREAL#0.0;
   Camin_Msd := LREAL#0.0;
   Camin_Ms := LREAL#1.0;
   Camin_Ss := LREAL#1.0;
   Camin_Mo := LREAL#0.0;
   Camin_So := LREAL#0.0;
   Camin_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
   Camin_Dir := _eMC_DIRECTION#_mcNoDirection;

   //입력 파라메터의 설정 완료 플래그를 유효화합니다.
   InitFlag: =TRUE;
END_IF;


//접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 0을 서보 ON합니다.
//서보 준비 상태가 되지 않은 경우에는 서보 OFF로 합니다.
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
   Pwr1_En :=TRUE;
ELSE
   Pwr1_En :=FALSE;
END_IF;


// 접점 StartPg가 TRUE이고 서보 드라이버가 서보 준비 상태이면 축 1을 서보 ON합니다.
//서보 준비 상태가 되지 않은 경우에는 서보 OFF로 합니다.
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
   Pwr2_En :=TRUE;
ELSE
   Pwr2_En :=FALSE;
END_IF;


//MC 공통 혹은 축에서 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
//이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE)
 OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (_MC_COM.MFaultLvl.Active=TRUE) THEN
   FaultHandler();
END_IF;


//축 0이 서보 ON 상태에서 원점 미확정인 경우 축 0의 원점 복귀를 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
   Hm1_Ex :=TRUE;
```

```
END_IF;


//축 1이 서보 ON 상태에서 원점 미확정인 경우 축 1의 원점 복귀를 실행합니다.
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
   Hm2_Ex :=TRUE;
END_IF;


//WriteCamData가 TRUE이고 캠 테이블 파일 저장이 실행 중이 아닌 경우 캠 데이터 변수값을 변경합니다.
//위상 0도~180도까지는 변위를 2배로 하고 181도~360도에서는 변위를 0.5배로 합니다.
//변경이 완료되었을 때 WriteDone을 TRUE로 합니다.
IF (WriteCamdata=TRUE) AND (_MC_COM.Status.CamTableBusy=FALSE) THEN
   FOR Index := UINT#0 TO UINT#360 DO
      IF Index<UINT#180 THEN
         CamProfile0[Index].Distance:=CamProfile0[Index].Distance*REAL#2.0;
      ELSE
         CamProfile0[Index].Distance:=CamProfile0[Index].Distance*REAL#0.5;
      END_IF;
   END_FOR;
   WriteDone :=TRUE;
   WriteCamdata :=FALSE;
END_IF;


//축 0의 원점 복귀가 완료되면 속도 제어를 실행합니다.
IF Hm1_D=TRUE THEN
   Vel_Ex :=TRUE;
END_IF;


// 캠 데이터 변수의 변경이 완료되어 SaveCamtable이 TRUE이고 캠 테이블 파일 저장이 실행 중이 아닌 경우
Sv_Cam_Ex를 TRUE로 합니다.
// Sv_Cam_Ex가 TRUE가 되면 MC_SaveCamTable(캠 테이블 저장) 명령을 실행합니다.
IF (WriteDone=TRUE)
 AND (SaveCamtable=TRUE)
 AND (_MC_COM.Status.CamTableBusy=FALSE) THEN
   Sv_Cam_Ex := TRUE;
END_IF;

// Sv_Ca_TimeUp이 TRUE인 경우 Sv_Cam_Ex를 FALSE로 합니다.
// Sv_Cam_Ex가 FALSE가 되면 Sv_Ca_TimeUp이 FALSE가 되고 Sv_Cam_Ex를 TRUE로 합니다.
//MC_SaveCamTable(캠 테이블 저장) 명령을 재실행합니다.
IF (Sv_Cam_Ex=TRUE) AND (Sv_Ca_TimeUp=FALSE) THEN
   Sv_Cam_Ex := TRUE;
ELSE
   Sv_Cam_Ex := FALSE;
END_IF;

// Sv_Ca_CountUP가 FALSE이고 캠 테이블 저장 실행 불가가 발생한 경우 Sv_Cam_Disable을 TRUE로 합니다.
IF (Sv_Ca_CountUP=FALSE) AND (Sv_Cam_Ca=TRUE) THEN
   Sv_Cam_Disable := TRUE;
ELSE
   Sv_Cam_Disable := FALSE;
END_IF;

// 캠 테이블 저장 실행 불가가 발생하고 1초 후에 Sv_Ca_TimeUp을 TRUE로 합니다.
// Sv_Ca_TimeUp이 TRUE가 되면 Sv_Cam_Ex가 FALSE가 됩니다.
```

```
Sv_Ca_TON(
   In := Sv_Cam_Disable ,
   PT := T#1s ,
   Q => Sv_Ca_TimeUp
);


// 캠 테이블 저장 실행 시에 Sv_Ca_CountLoad를 1주기 동안 TRUE로 합니다.
// Sv_Ca_CountLoad가 TRUE가 되면 재실행 카운터를 리셋합니다.
R_TRIG1(SaveCamtable, Sv_Ca_CountLoad);


// 캠 테이블 저장 실행 불가가 3회 발생한 경우 Sv_Ca_CountUP을 TRUE로 합니다.
// Sv_Ca_CountUP이 TRUE가 되면 Sv_Cam_Disable을 FALSE로 합니다.
// MC_SaveCamTable(캠 테이블 저장) 명령의 재실행을 종료합니다.
Sv_Ca_CTD(
   CD := Sv_Cam_Disable ,
   LOAD := Sv_Ca_CountLoad ,
   PV := INT#3 ,
   Q => Sv_Ca_CountUP
);


// 캠 데이터 변수 변경과 캠 테이블 저장이 완료되고 축 0이 목표 속도에 도달하면 캠 동작을 실행합니다.
IF (Vel_InVel=TRUE)
 AND (WriteDone=TRUE)
 AND (Sv_Cam_D=TRUE) THEN
   Camin_Ex :=TRUE;
END_IF;

//MC_SaveCamTable
SV_CAM(
   CamTable :=CamProfile0,
   Execute := Sv_Cam_Ex,
   Done => Sv_Cam_D,
   Busy => Sv_Cam_Bsy,
   CommandAborted => Sv_Cam_Ca,
   Error => Sv_Cam_Err,
   ErrorID => Sv_Cam_ErrID
);

CAMIN(
   Master := MC_Axis000,
   Slave := MC_Axis001,
   CamTable := CamProfile0,
   Execute := Camin_Ex,
   Periodic := Camin_Em,
   StartMode := Camin_Sm,
   StartPosition := Camin_Sp,
   MasterStartDistance := Camin_Msd,
   MasterScaling := Camin_Ms,
   SlaveScaling := Camin_Ss,
   MasterOffset := Camin_Mo,
   SlaveOffset := Camin_So,
   ReferenceType := Camin_Rt,
   Direction := Camin_Dir,
   InCam => Camin_InCam,
   InSync => Camin_InSync,
   EndOfProfile => Camin_Eop,
```

```
    Index => Camin_Index,
    Busy => Camin_Bsy,
    Active => Camin_Act,
    CommandAborted => Camin_Ca,
    Error => Camin_Err,
    ErrorID => Camin_ErrID
);

//축 0의 MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

//축 1의 MC_Power
PWR2(
    Axis := MC_Axis001,
    Enable := Pwr2_En,
    Status => Pwr2_Status,
    Busy => Pwr2_Bsy,
    Error => Pwr2_Err,
    ErrorID => Pwr2_ErrID
);

//축 0의 MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

//축 1의 MC_Home
HM2(
    Axis := MC_Axis001,
    Execute := Hm2_Ex,
    Done => Hm2_D,
    Busy => Hm2_Bsy,
    CommandAborted => Hm2_Ca,
    Error => Hm2_Err,
    ErrorID => Hm2_ErrID
);

//MC_MoveVelocity
VEL(
    Axis := MC_Axis000,
    Execute := Vel_Ex,
    Velocity := Vel_Vel,
    Acceleration := Vel_Acc,
    Deceleration := Vel_Dec,
    Direction := Vel_Dir,
    InVelocity => Vel_InVel,
```

```
    Busy => Vel_Bsy,
    Active => Vel_Act,
    CommandAborted => Vel_Ca,
    Error => Vel_Err,
    ErrorID => Vel_ErrID
);
```

10-2-17 축 파라메터의 일시 갱신

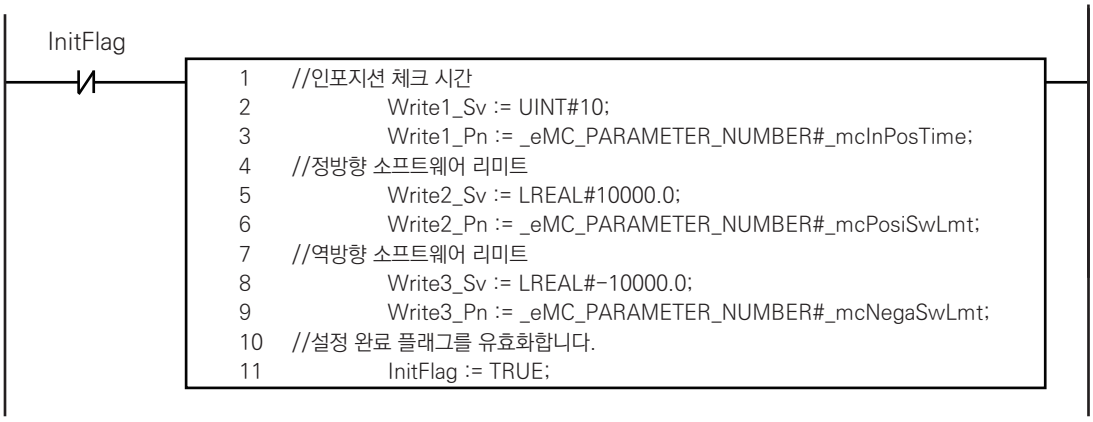
MC 설정 기록(MC_Write) 명령을 사용하여 인포지션 체크 시간, 정방향 소프트웨어 리미트, 역방향 소프트웨어 리미트의 값을 변경합니다.

프로그램에서 사용하는 주요 변수

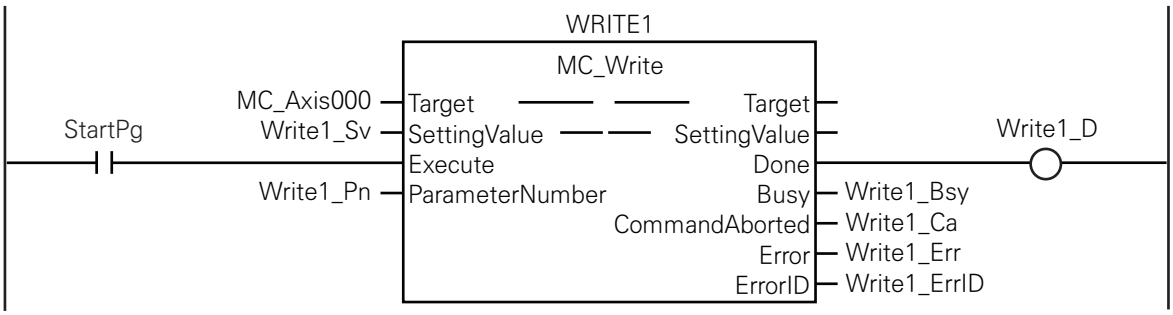
명칭	데이터형	초기값	코멘트
MC_Axis000	_sAXIS_REF	—	축 0의 축 변수입니다.
InitFlag	BOOL	FALSE	파라메터의 설정 상태를 나타내는 변수입니다. FALSE에서 파라메터 변경을 실행하고 완료했을 때 TRUE로 합니다.
StartPg	BOOL	FALSE	MC 설정 기록을 실행하기 위한 변수입니다.

래더도 프로그램

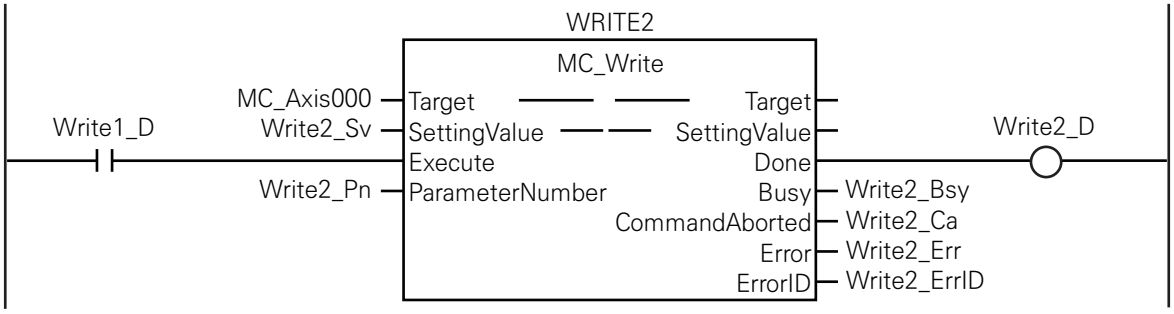
파라메터를 설정합니다. 설정이 완료되면 InitFlag를 TRUE로 합니다.



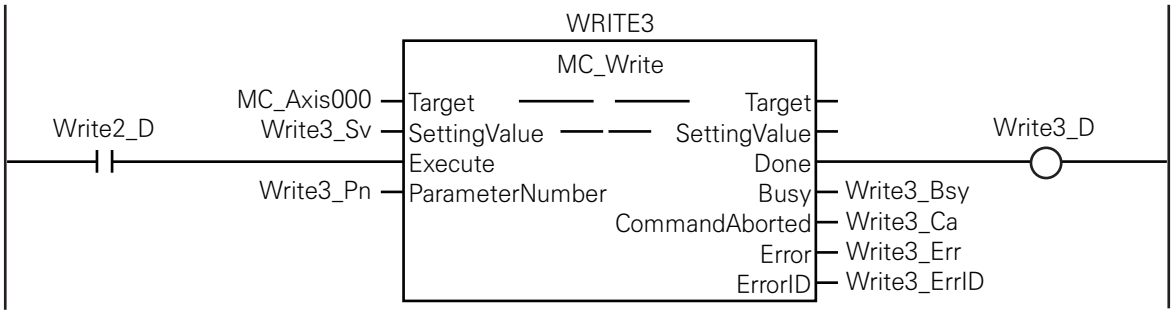
StartPg가 TRUE가 되면 인포지션 체크 시간을 변경합니다.



인포지션 체크 시간의 설정이 완료되면 정방향 소프트웨어 리미트를 변경합니다.



정방향 소프트웨어 리미트의 설정이 완료되면 역방향 소프트웨어 리미트를 변경합니다.



ST 프로그램

```
//파라메터를 설정합니다.  설정이 완료되면 InitFlag를 TRUE로 합니다.
IF InitFlag=FALSE THEN

  //인포지션 체크 시간
    Write1_Sv := UINT#10;
    Write1_Pn := _eMC_PARAMETER_NUMBER#_mcInPosTime;

  //정방향 소프트웨어 리미트
    Write2_Sv := LREAL#10000.0;
    Write2_Pn := _eMC_PARAMETER_NUMBER#_mcPosiSwLmt;

  //역방향 소프트웨어 리미트
    Write3_Sv := LREAL#-10000.0;
    Write3_Pn := _eMC_PARAMETER_NUMBER#_mcNegaSwLmt;

  //입력 파라메터의 설정 완료 플래그를 유효화합니다.
  InitFlag := TRUE;
END_IF;


// StartPg가 TRUE가 되면 인포지션 체크 시간을 변경합니다.
IF StartPg =TRUE THEN
  Write1_Ex := TRUE;
END_IF;


//인포지션 체크 시간의 변경이 완료되면 정방향 소프트웨어 리미트를 변경합니다.
IF Write1_D = TRUE THEN
  Write2_Ex := TRUE;
END_IF;


//정방향 소프트웨어 리미트의 변경이 완료되면 역방향 소프트웨어 리미트를 변경합니다.
IF Write2_D = TRUE THEN
```

```
    Write3_Ex := TRUE;
END_IF;

// MC_Write
WRITE1(
    Target := MC_Axis000,
    SettingValue := Write1_Sv,
    Execute := Write1_Ex,
    ParameterNumber := Write1_Pn,
    Done => Write1_D,
    Busy => Write1_Bsy,
    CommandAborted => Write1_Ca,
    Error => Write1_Err,
    ErrorID => Write1_ErrID
);

WRITE2(
    Target := MC_Axis000,
    SettingValue := Write2_Sv,
    Execute := Write2_Ex,
    ParameterNumber := Write2_Pn,
    Done => Write2_D,
    Busy => Write2_Bsy,
    CommandAborted => Write2_Ca,
    Error => Write2_Err,
    ErrorID => Write2_ErrID
);

WRITE3(
    Target := MC_Axis000,
    SettingValue := Write3_Sv,
    Execute := Write3_Ex,
    ParameterNumber := Write3_Pn,
    Done => Write3_D,
    Busy => Write3_Bsy,
    CommandAborted => Write3_Ca,
    Error => Write3_Err,
    ErrorID => Write3_ErrID
);
```

10-2-18 캠 테이블의 종료점 인덱스 갱신

최대 캠 데이터 수가 110점이고 유효 데이터 수가 100점인 캠 테이블에 대해서 유효 데이터 수를 10점 늘리고 종료점 인덱스를 갱신합니다.

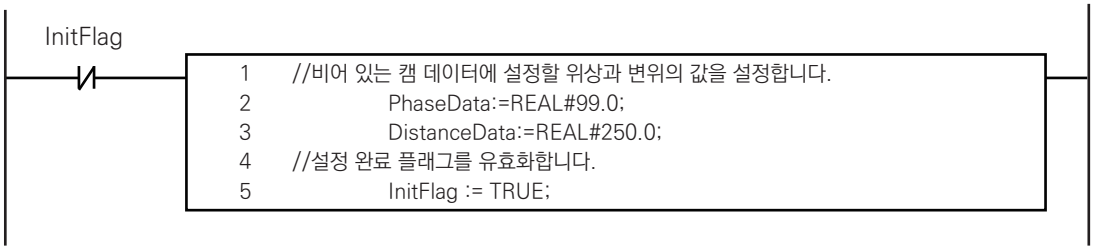
프로그램에서 사용하는 주요 변수

명칭	데이터형	초기값	코멘트
CamProfile0	ARRAY[0..109] OF _sMC_CAM_REF	—	최대 캠 데이터 수가 110점인 캠 데이터 변수입니다. *1 유효 캠 데이터 수는 100점이고 비어 있는 데이터 수는 10점입니다.
WriteCamdata	BOOL	FALSE	캠 데이터 변경을 기동하기 위한 변수입니다. 편집을 시작할 때 TRUE로 해 주십시오.
WriteDone	BOOL	FALSE	캠 데이터 변경이 끝났음을 나타내기 위한 변수입니다. 변경이 끝났을 때 TRUE로 해 주십시오.

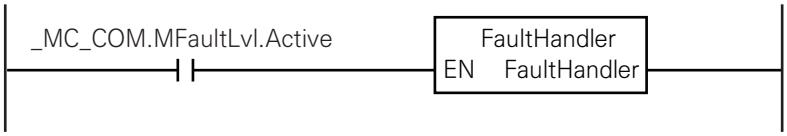
*1. 배열의 요소 수 ARRAY[0..N]는 Sysmac Studio의 캠 에디터 기능으로 설정합니다. 본 샘플 프로그램에서는 [0~109]로 합니다.

래더도 프로그램

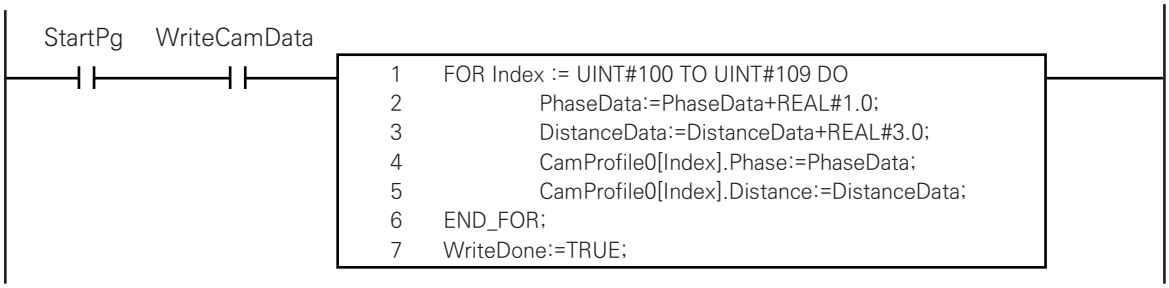
파라메터를 설정합니다. 설정이 완료되면 InitFlag를 TRUE로 합니다.



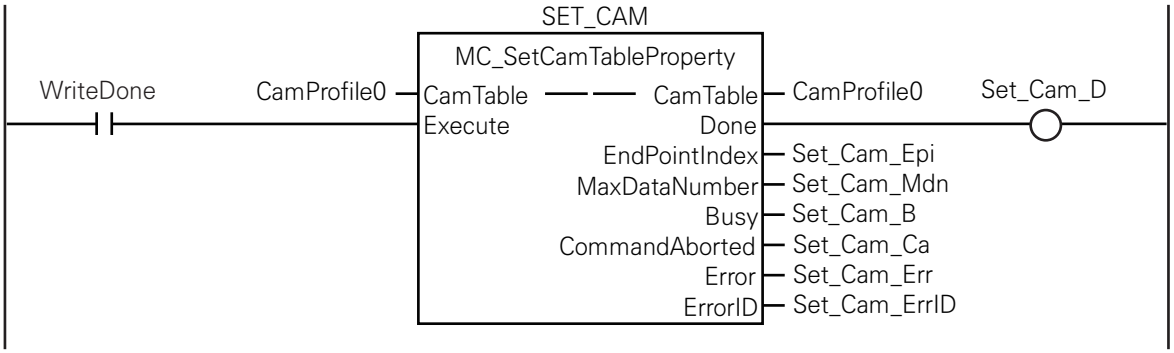
MC 공통으로 경도 폴트 레벨의 이상이 발생한 경우 이상 처리의 FaultHandler를 실행합니다.
이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.



StartPg와 WriteCamData가 TRUE가 되면 캠 데이터 변수값을 변경합니다.
CamProfile[100]~CamProfile[109]에 위상과 변위를 설정했습니다.
캠 데이터 변수의 변경이 완료되었을 때 WriteDone을 TRUE로 합니다.



캠 데이터 변수의 변경이 완료되면 캠 테이블 속성 갱신을 실행합니다.



ST 프로그램

```
//파라메터를 설정합니다.  설정이 완료되면 InitFlag를 TRUE로 합니다.
IF InitFlag=FALSE THEN
  //비어 있는 캠 데이터에 설정할 위상과 변위의 값을 설정합니다.
  PhaseData :=REAL#99.0;
  DistanceData :=REAL#250.0;

  //설정 완료 플래그를 TRUE로 합니다.
  InitFlag := TRUE;
END_IF;


// MC 공통으로 경도 폴트 레벨의 이상이 발생한 경우 이상 시 처리를 실행합니다.
// 이상 처리 내용은 장비에 맞춰 프로그래밍해 주십시오.
IF _MC_COM.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;


// StartPg와 WriteCamData가 TRUE가 되면 캠 데이터 변수값을 변경합니다.
// CamProfile[100]~CamProfile[109]에 위상과 변위를 설정했습니다.
// 캠 데이터 변수의 변경이 완료되었을 때 WriteDone을 TRUE로 합니다.
IF StartPg=TRUE
AND WriteCamData=TRUE THEN
  FOR Index := UINT#100 TO UINT#109 DO
    PhaseData :=PhaseData+REAL#1.0;
    DistanceData :=DistanceData+REAL#3.0;
    CamProfile0[Index].Phase :=PhaseData;
    CamProfile0[Index].Distance :=DistanceData;
  END_FOR;
  WriteDone :=TRUE;
END_IF;


// 캠 데이터 변수의 변경이 완료되면 캠 테이블 속성 갱신을 실행합니다.
IF WriteDone=TRUE THEN
  Set_Cam_Ex := TRUE;
END_IF;

//MC_SetCamTableProperty
SET_CAM(
  CamTable := CamProfile0,
  Execute := Set_Cam_Ex,
  Done => Set_Cam_D,
```

```
    EndPointIndex => Set_Cam_Epi,
    MaxDataNumber => Set_Cam_Mdn,
    Busy => Set_Cam_B,
    CommandAborted => Set_Cam_Ca,
    Error => Set_Cam_Err,
    ErrorID => Set_Cam_ErrID
);
```

11

이상 확인과 조치

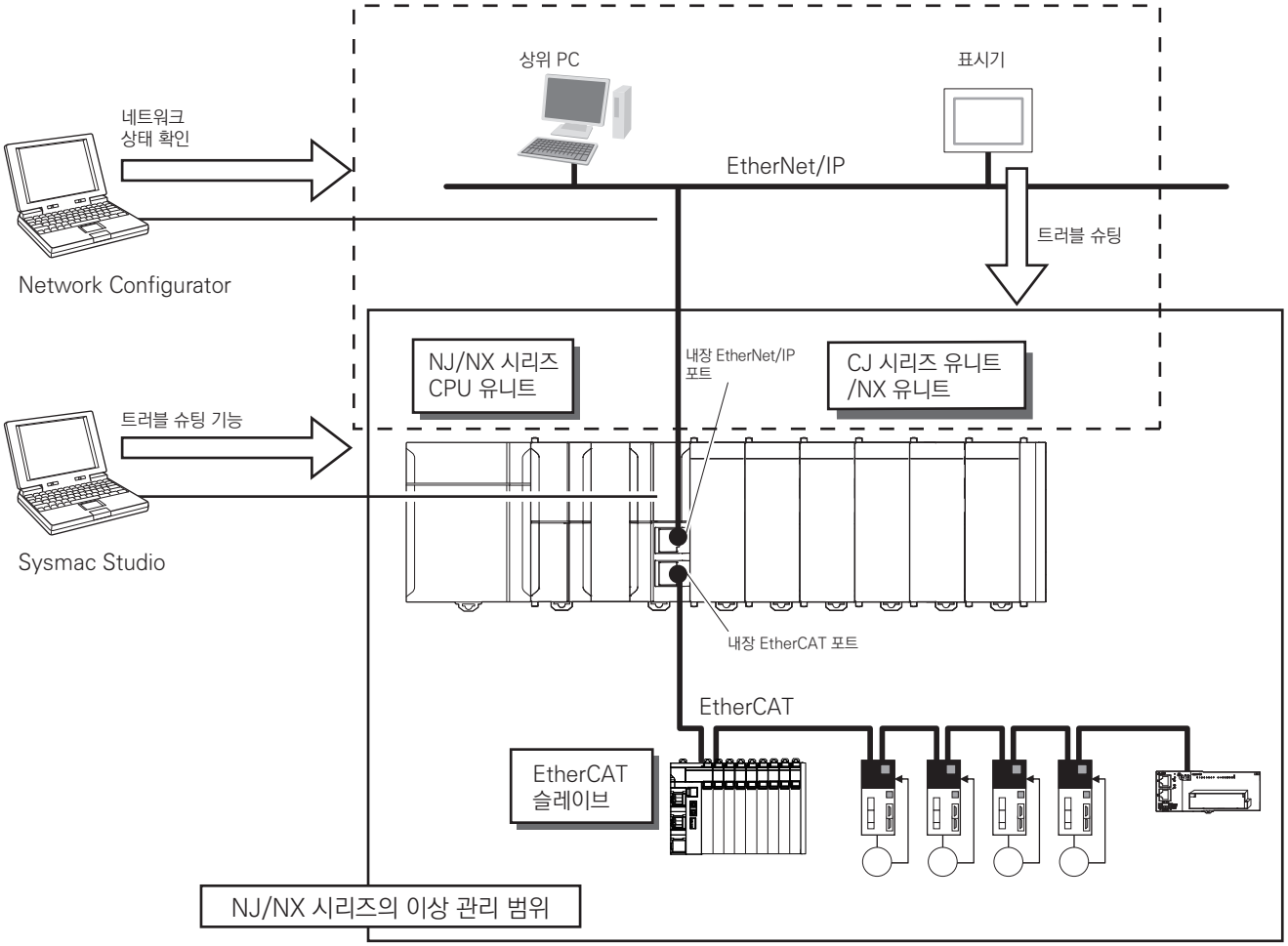
이상 확인 방법의 개요에 대해 설명합니다.

11-1 이상 확인과 조치의 개요 11-2

11-1 이상 확인과 조치의 개요

NJ/NX 시리즈는 컨트롤러에서 발생한 이상을 이벤트로 취급하여 공통의 방법으로 일원 관리합니다.
이를 통해 관리 범위(CPU 유니트, NX 유니트, NX 시리즈 슬레이브 터미널, EtherCAT 슬레이브*1, CJ 시리즈 유니트)의 이상에 대해 공통의 방법으로 이상 내용과 대처 방법을 확인할 수 있습니다.

*1. Sysmac 디바이스여야 합니다.



Sysmac Studio의 트러블 슈팅 기능 또는 표시기의 트러블 슈터를 사용하면 이상 내용부터 대처 방법까지 확인할 수 있으므로 신속하게 대응할 수 있습니다.

이상의 종류와 내용, 이상 발생 시의 구체적인 대처 방법, NJ/NX 시리즈 전체의 트러블 슈팅에 대해서는 🕮『NJ/NX 시리즈 트러블 슈팅 매뉴얼』(SBCA-361V 이상)을 참조해 주십시오.

A

부록

오므론 서보 드라이버 1S 시리즈 및 G5 시리즈의 오브젝트 설정값 및 접속 방법이 기재되어 있습니다.

A-1 서보 드라이버 1S 시리즈와의 접속 A-2
A-1-1 서보 드라이버와의 배선 A-2
A-1-2 서보 드라이버의 설정 A-2

A-2 서보 드라이버 G5 시리즈와의 접속 A-11
A-2-1 서보 드라이버와의 배선 A-11
A-2-2 서보 드라이버의 설정 A-11

A-3 엔코더 입력 터미널과의 접속 A-21
A-3-1 엔코더 입력 터미널과의 배선 A-21
A-3-2 엔코더 입력 터미널 설정 A-21

A-4 NX 유니트와의 접속 A-27

A-5 PDS 상태 변이 A-28
A-5-1 PDS 상태 제어 방식 A-29
A-5-2 주회로 전원 OFF 검출 A-30

A-6 용어 설명 A-31
A-6-1 NJ/NX 시리즈 공통 A-31
A-6-2 모션 제어 기능 A-32
A-6-3 EtherCAT 통신 A-33

A-7 버전 관련 정보 A-34

A-1 서보 드라이버 1S 시리즈와의 접속

오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입과의 접속에 대해 설명합니다.

A-1-1 서보 드라이버와의 배선

서보 드라이버와는 EtherCAT 통신으로 접속합니다.
접속 방법에 대해서는 📖『NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용자 매뉴얼(SBCD-358)』을 참조해 주십시오.

A-1-2 서보 드라이버의 설정

여기에서는 MC 기능 모듈의 제어 대상인 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입과 접속한 경우의 서보 드라이버 측 설정 개요에 대해 설명합니다.
서보 드라이버에 대한 상세한 내용은 📖『AC 서보 모터/드라이버 1S 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼(SBCE-377)』을 참조해 주십시오.

외부 입력 신호 할당

MC 기능 모듈에서는 서보 드라이버의 범용 입력을 아래의 입력 신호로 사용합니다.

- 즉시 정지 입력
- 정방향 한계 입력
- 역방향 한계 입력
- 원점 근방 입력
- 외부 래치 기능의 트리거 신호(래치 입력 1/래치 입력 2)

● 정방향 한계 입력/역방향 한계 입력/원점 근방 입력 할당

오므론 서보 드라이버 1S 시리즈의 출하 설정에서는 각 입력 신호가 아래의 표와 같이 할당되어 있습니다.

신호 명칭	입력 신호
즉시 정지 입력	서보 드라이버의 범용 입력 1(IN1: CN1 커넥터의 핀 No.12 b접점) *1
정방향 한계 입력	서보 드라이버의 범용 입력 2(IN2: CN1 커넥터의 핀 No.32 b접점) *2
역방향 한계 입력	서보 드라이버의 범용 입력 3(IN3: CN1 커넥터의 핀 No.13 b접점) *3
원점 근방 입력	서보 드라이버의 범용 입력 4(IN4: CN1 커넥터의 핀 No.33 a접점)

*1. 서보 드라이버 측의 신호 명칭은「이상 정지 입력」입니다.
*2. 서보 드라이버 측의 신호 명칭은「정방향 구동 금지 입력」입니다.
*3. 서보 드라이버 측의 신호 명칭은「역방향 구동 금지 입력」입니다.

● 외부 래치 기능의 트리거 신호 할당

오므론 서보 드라이버 1S 시리즈의 출하 시 설정의 경우 아래 표의 입력 신호가 외부 래치 기능의 트리거 신호로 할당되어 있습니다.

MC_TouchProbe 명령의 입력 변수 「TriggerInput(트리거 입력 조건)」 설정			외부 래치 기능의 트리거 신호
Mode	InputDrive	LatchID	
0: mcDrive	0: mcEncoderMark	-	엔코더의 Z상
	1: mcEXT	1: mcLatch1	서보 드라이버의 범용 입력 7 (IN7: CN1 커넥터의 핀 No.15 a접점) *1
		2: mcLatch2	서보 드라이버의 범용 입력 8 (IN8: CN1 커넥터의 핀 No.35 a접점) *2
1: mcController	-	-	TriggerVariable(트리거 변수)로 지정한 변수

*1. 서보 드라이버 측의 신호 명칭은 「외부 래치 입력 1」입니다.
*2. 서보 드라이버 측의 신호 명칭은 「외부 래치 입력 2」입니다.

백래시 보정 기능

MC 기능 모듈은 백래시 보정 기능이 없습니다.
백래시 보정이 필요한 경우에는 서보 드라이버 측의 보정 기능을 사용합니다.
서보 드라이버 측에서 설정이 필요한 오브젝트는 아래와 같습니다.

인덱스	서브 인덱스	명칭	설명
3001Hex	-	기계	-
	02Hex	백래시 보정 선택	위치 제어 시의 백래시 보정의 유효/무효, 보정 시의 동작 방향을 설정합니다. *1 0: 백래시 보정 무효 1: 서보 ON 이후 최초의 정회전 측 동작 시에 보정합니다. 2: 서보 ON 이후 최초의 역회전 측 동작 시에 보정합니다.
	03Hex	백래시 보정량	위치 제어 시의 백래시 보정량을 설정합니다.
	04Hex	백래시 보정 시정수	위치 제어 시의 백래시 보정 시정수를 설정합니다.

*1. 출하 시 설정은 「0: 백래시 보정 무효」입니다.

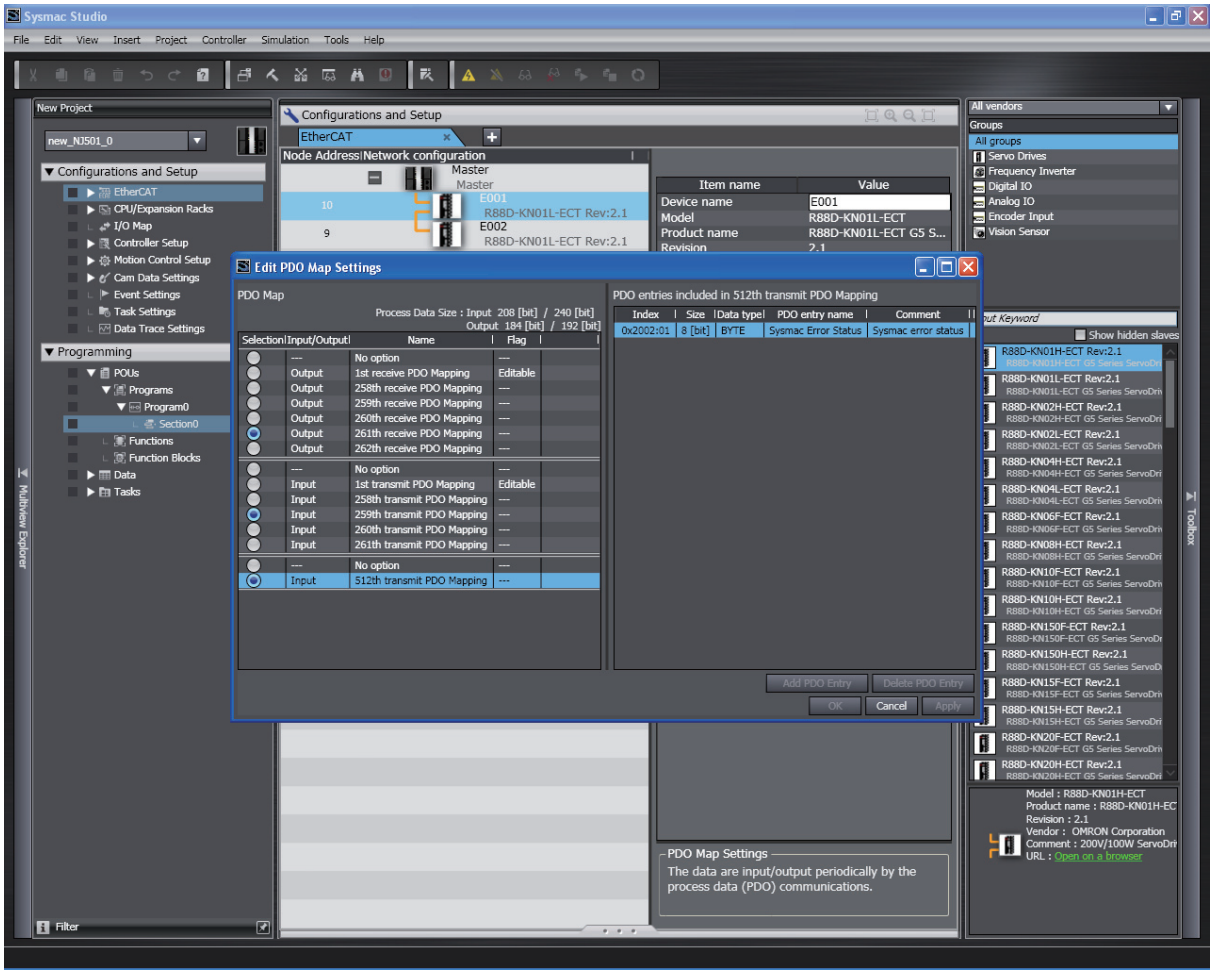
백래시 기능에 대한 상세한 내용은 🕮『AC 서보 모터/드라이버 1S 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼(SBCE-377)』을 참조해 주십시오.

PDO 매핑

MC 기능 모듈의 서보 축으로 제어하기 위한 PDO 매핑에 대해 설명합니다.

모션 제어의 각 기능을 사용하려면 사용할 기능에 필요한 오브젝트를 프로세스 데이터 통신에 매핑해야 합니다.
필요한 오브젝트를 미리 정리한 것이 PDO 맵입니다.

사용할 PDO 맵은 Sysmac Studio의 [EtherCAT] 구성 편집 화면의 [Edit PDO Map Settings]에서 선택합니다.



Sysmac Studio의 디폴트에는 아래의 PDO 맵이 선택되어 있습니다.

RxPDO: [261th receive PDO Mapping] (1704Hex)	컨트롤 워드(6040Hex), 목표 위치(607AHex), 목표 속도(60FFHex), 목표 토크(6071Hex), 오퍼레이션 모드(6060Hex), 래치 기능 (60B8Hex), 최대 프로파일 속도(607FHex), 정방향 토크 제한값(60E0Hex), 역방향 토크 제한값(60E1Hex)
TxPDO: [259th transmit PDO Mapping] (1B02Hex)	에러 코드(603FHex), 상태 워드(6041Hex), 현재 위치(6064Hex), 현재 토크(6077Hex), 오퍼레이션 모드 표시(6061Hex), 래치 상태(60B9Hex), 래치 1 위치(60BAHex), 래치 2 위치 (60BCHex), 디지털 입력(60FDHex)

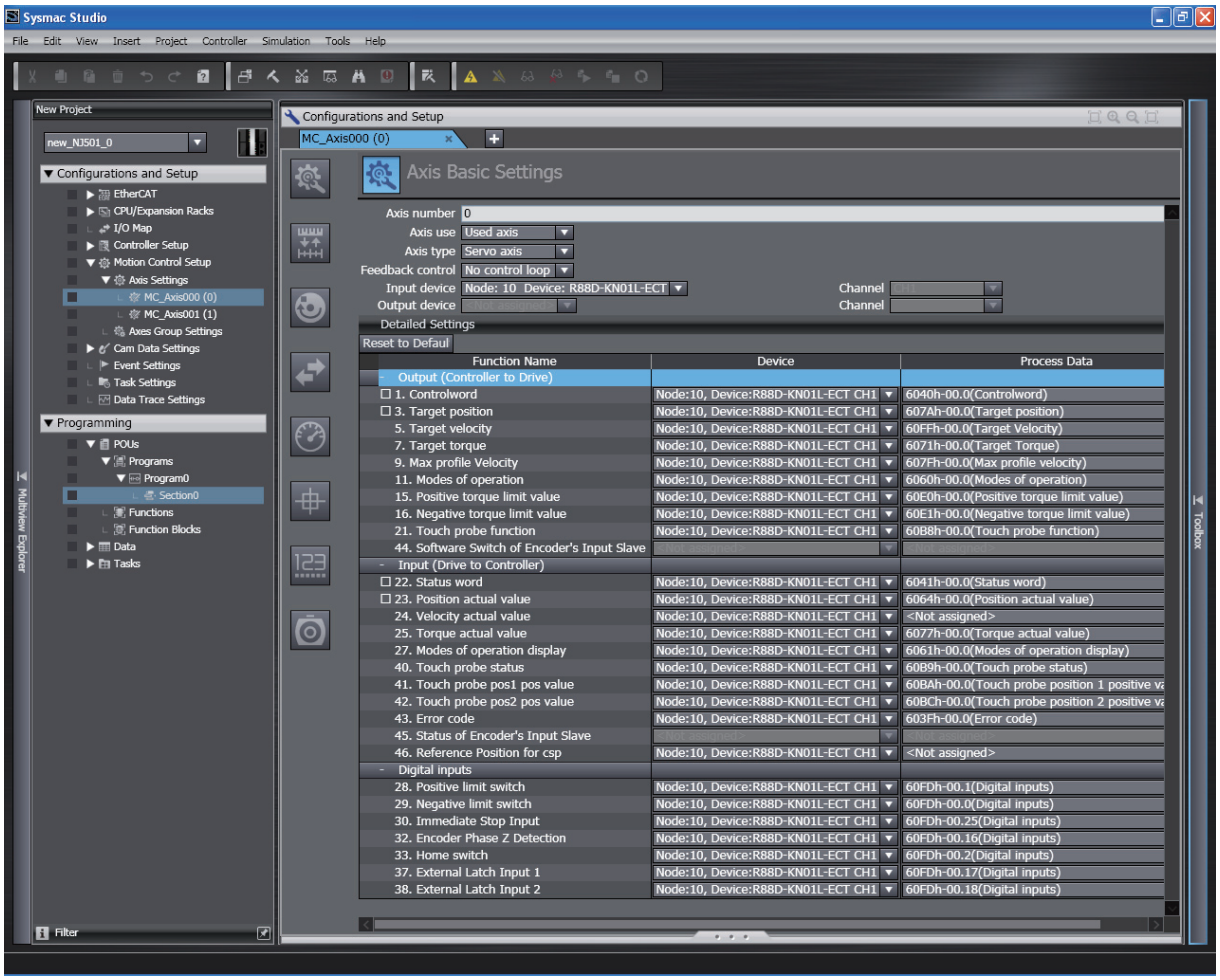
참고

- 권장 서보 드라이버인 오므론 R88D-1SN□□□-ECT의 경우 PDO 맵은 Sysmac Studio의 디폴트 상태에서도 사용할 수 있습니다.

MC 기능 모듈의 기능과 프로세스 데이터의 관계

MC 기능 모듈의 기능은 프로세스 데이터의 오브젝트 정보와 관계가 있습니다.
MC 기능 모듈이 사용하는 EtherCAT 슬레이브의 구성 및 사용 기능에 따라서는 이러한 관계를 변경해야 하는 경우가 있습니다.

Sysmac Studio의 [Axis Basic Settings] 화면의 [Detailed Settings]를 클릭하면 설정 화면이 열립니다.



참고

권장 서보 드라이버인 오므론 R88D-1SN□□□-ECT의 경우 기능과 프로세스 데이터와의 관계는 Sysmac Studio의 디폴트 상태에서도 사용할 수 있습니다.

● 출력(컨트롤러→드라이버) 설정

MC 기능 모듈에서 서보 드라이버로 보내는 지령 데이터 설정입니다.
아래는 Sysmac Studio의 디폴트 설정입니다. (○표시는 필수 오브젝트)

기능명	프로세스 데이터	설명
○ 컨트롤 워드	6040Hex-00.0 (Controlword)	서보 드라이버의 상태를 제어하기 위한 데이터입니다. [6040Hex: Controlword]를 설정해 주십시오.
○ 목표 위치	607AHex-00.0 (Target position)	위치 제어에서의 목표 위치입니다. [607AHex: Target position]을 설정해 주십시오.
목표 속도	60FFHex-00.0 (Target velocity)	속도 제어에서의 목표 속도입니다. MC_SyncMoveVelocity(사이클릭 동기 속도 제어) 명령 등 사이클릭 동기 속도 모드에서 드라이버로 출력하는 경우에 필요합니다. 일반적으로는 [60FFHex: Target velocity]를 설정해 주십시오.
목표 토크	6071Hex-00.0 (Target torque)	토크 제어에서의 목표 토크입니다. MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 토크 모드에서 드라이버로 출력하는 경우에 필요합니다. 일반적으로는 [6071Hex: Target torque]를 설정해 주십시오.
최대 프로파일 속도	607FHex-00.0 (Max profile velocity)	토크 제어 시의 속도 제한값입니다. MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 토크 모드에서 속도를 제한하는 경우에 필요합니다. 일반적으로는 [607FHex: Max profile velocity]를 설정해 주십시오.
오퍼레이션 모드	6060Hex-00.0 (Modes of operation)	제어 모드를 변환하기 위한 데이터입니다. MC_SyncMoveVelocity(사이클릭 동기 속도 제어) 명령이나 MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 위치 모드 이외의 제어 모드로 변환하는 경우에 필요합니다. 일반적으로는 [6060Hex: Modes of operation]을 설정해 주십시오. *1
정방향 토크 제한값	60E0Hex-00.0 (Positive torque limit value)	정방향 측의 토크 제한값입니다. MC_SetTorqueLimit(토크 제한) 명령 등 드라이버의 출력 토크를 제한하는 경우에 필요합니다. 일반적으로는 [60E0Hex: Positive torque limit value]를 설정해 주십시오.
역방향 토크 제한값	60E1Hex-00.0 (Negative torque limit value)	역방향 측의 토크 제한값입니다. MC_SetTorqueLimit(토크 제한) 명령 등 드라이버의 출력 토크를 제한하는 경우에 필요합니다. 일반적으로는 [60E1Hex: Negative torque limit value]를 설정해 주십시오.
래치 기능	60B8Hex-00.0 (Touch probe function)	래치 기능을 제어하기 위한 데이터입니다. MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령, MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령 등 래치 기능을 사용할 경우에 필요합니다. 일반적으로는 [60B8Hex: Touch probe function]을 설정해 주십시오.

*1. [6060Hex: Modes of operation]을 설정할 경우에는 [6061Hex: Modes of operation display]도 설정해 주십시오. 둘 중 하나만 설정하면 정상적으로 동작하지 않습니다.

사용상의 주의

- 설정 변경 시에는 MC 기능 모듈의 기능과 프로세스 데이터의 조합이 의도한 대로 동작하는지 확인해 주십시오.
- 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입 또는 G5 시리즈 EtherCAT 통신 내장 타입 이외를 사용할 경우에는 오퍼레이션 모드(6060Hex)를 반드시 설정해 주십시오.

● 입력(드라이버→컨트롤러) 설정

서보 드라이버에서 MC 기능 모듈로 보내는 상태 데이터입니다.
아래는 Sysmac Studio의 디폴트 설정입니다. (○표시는 필수 오브젝트)

기능명	프로세스 데이터	설명
○ 상태 워드	6041Hex-00.0 (Statusword)	서보 드라이버의 상태입니다. [6041Hex: Statusword]를 설정해 주십시오.
○ 현재 위치	6064Hex-00.0 (Position actual value)	현재 위치 표시입니다. [6064Hex: Position actual value]를 설정해 주십시오.
현재 속도	설정 없음 *1	현재 속도 표시입니다. 사용하려면 일반적으로 [606CHex: Velocity actual value]를 설정해 주십시오.
현재 토크	6077Hex (Torque actual value)	현재 토크 표시입니다. MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 토크 모드에서 드라이버로 출력하는 경우에 필요합니다. 일반적으로는 [6077Hex: Torque actual value]를 설정해 주십시오.
오퍼레이션 모드 표시	6061Hex-00.0 (Modes of operation display)	제어 모드 표시입니다. MC_SyncMoveVelocity(사이클릭 동기 속도 제어) 명령이나 MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 위치 모드 이외의 제어 모드로 변환하는 경우에 필요합니다. 일반적으로는 [6061Hex: Modes of operation display]를 설정해 주십시오. *2
래치 상태	60B9Hex-00.0 (Touch probe status)	래치 기능의 상태 표시입니다. MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령, MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령 등 래치 기능을 사용할 경우에 필요합니다. 일반적으로는 [60B9Hex: Touch probe status]를 설정해 주십시오.
래치 1 위치	60BAHex-00.0 (Touch probe pos1 pos value)	래치 기능 1에서 래치한 위치입니다. MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령, MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령 등 래치 기능을 사용할 경우에 필요합니다. 일반적으로는 [60BAHex: Touch probe pos1 pos value]를 설정해 주십시오.
래치 2 위치	60BCHex-00.0 (Touch probe pos2 pos value)	래치 기능 2에서 래치한 위치입니다. MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령, MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령 등 래치 기능을 사용할 경우에 필요합니다. 일반적으로는 [60BCHex: Touch probe pos2 pos value]를 설정해 주십시오.

기능명	프로세스 데이터	설명
에러 코드	603FHex-00.0 (Error code)	드라이버 내의 이상 코드입니다. 일반적으로는 [603FHex: Error code]를 설정해 주십시오.
csp용 참조 위치	설정 없음	csp 모드 변환 시의 참조 위치입니다. 속도 제어 모드(CSV)를 사용하는 명령 및 토크 제어 모드(CST)를 사용하는 명령에서 참조되는 데이터입니다. 오므론 서보 모터/드라이버 1S 시리즈에서 사용할 수 있습니다. *3

*1. 필요한 경우에는 선택할 프로세스 데이터를 PDO에 매핑한 후에 설정해 주십시오. 606CHex-00.0(Velocity actual value)이 표준 설정입니다.
*2. [6061Hex: Modes of operation display]를 설정할 경우에는 [6060Hex: Modes of operation]도 설정해 주십시오. 둘 중 하나만 설정하면 정상적으로 동작하지 않습니다.
*3. 사용하려면 [3010Hex-87Hex: Reference Position for csp]를 PDO에 매핑해 주십시오.

사용상의 주의

- 설정 변경 시에는 MC 기능 모듈의 기능과 프로세스 데이터의 조합이 의도한 대로 동작하는지 확인해 주십시오.
- 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입 또는 G5 시리즈 EtherCAT 통신 내장 타입 이외를 사용할 경우에는 오퍼레이션 모드 표시(6061Hex)를 반드시 설정해 주십시오.
- 오므론 서보 드라이버 1S 시리즈를 사용하여 MC_SyncMoveVelocity(사이클릭 동기 속도 제어) 명령으로 제어 모드를 변환할 경우에는「csp용 참조 위치」를 매핑해 주십시오.

● 디지털 입력 설정

MC 기능 모듈이 사용하는 서보 드라이버의 입력 신호입니다.

기능명	프로세스 데이터	설명
정방향 구동 금지 입력	60FDHex-00.1 (Digital inputs)	정방향 한계 입력으로 사용하는 신호입니다. 일반적으로는 [60FDHex-00: Digital inputs]의 [Bit 1: Positive limit switch]를 설정해 주십시오.
역방향 구동 금지 입력	60FDHex-00.0 (Digital inputs)	역방향 한계 입력으로 사용하는 신호입니다. 일반적으로는 [60FDHex-00: Digital inputs]의 [Bit 0: Negative limit switch]를 설정해 주십시오.
이상 정지 입력	60FDHex-00.25 (Digital inputs)	즉시 정지 입력으로 사용하는 신호입니다. 오므론 서보 드라이버 1S 시리즈는 [60FDHex-00: Digital inputs]의 [Bit 25: Error Stop Input]을 설정해 주십시오.
엔코더 Z상 검출	60FDHex-00.16 (Digital inputs)	Z상 입력의 검출 상태 표시입니다. 오므론 서보 모터/드라이버 1S 시리즈는 [60FDHex-00: Digital inputs]의 [Bit 16: Encoder Phase Z Detection]을 설정해 주십시오.
원점 근방 입력	60FDHex-00.2 (Digital inputs)	원점 근방 입력으로 사용하는 신호입니다. 일반적으로는 [60FDHex-00: Digital inputs]의 [Bit 2: Home switch]를 설정해 주십시오.
외부 래치 입력 1	60FDHex-00.17 (Digital inputs)	외부 래치 입력 1로 사용하는 신호의 상태 표시입니다. 오므론 서보 드라이버 1S 시리즈는 [60FDHex-00: Digital inputs]의 [Bit 17: External Latch Input 1]을 설정해 주십시오.
외부 래치 입력 2	60FDHex-00.18 (Digital inputs)	외부 래치 입력 2로 사용하는 신호의 상태 표시입니다. 오므론 서보 드라이버 1S 시리즈는 [60FDHex-00: Digital inputs]의 [Bit 18: External Latch Input 2]를 설정해 주십시오.

사용상의 주의

- 설정 변경 시에는 MC 기능 모듈의 기능과 프로세스 데이터의 조합이 의도한 대로 동작하는지 확인해 주십시오.

버전 관련 정보

- 유니트 버전 Ver.1.09 이하인 CPU 유니트에서 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입 이외를 사용할 경우에는 오퍼레이션 모드(6060Hex)와 오퍼레이션 모드 표시(6061Hex)가 필요합니다.
- 유니트 버전 Ver.1.10 이상인 CPU 유니트의 경우 오퍼레이션 모드(6060Hex) 및 오퍼레이션 모드 표시(6061Hex)의 매핑 유무에 따라 다음과 같이 동작합니다.

	오퍼레이션 모드 표시	
	(6061Hex) 있음	(6061Hex) 없음
오퍼레이션 모드 (6060Hex) 있음	・CSP*1, CSV*2, CST*3를 사용하는 명령을 실행할 수 있습니다. ・CSP, CSV, CST 이외일 때는 서보 OFF가 됩니다.	・CSP를 사용하는 명령을 실행할 수 있습니다. 그 외 제어 모드를 사용하는 명령은 실행 시에 「프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)」이 발생합니다. ・MC 기능 모듈은 서보 드라이버의 제어 모드를 CSP로 간주합니다. 서보 드라이버에 CSP를 지령합니다.
오퍼레이션 모드 (6060Hex) 없음	・CSP를 사용하는 명령을 실행할 수 있습니다. 그 외 제어 모드를 사용하는 명령은 실행 시에 「프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)」이 발생합니다. ・CSP 이외일 때는 서보 OFF가 됩니다.	CSP를 사용하는 명령을 실행할 수 있습니다. 그 외 제어 모드를 사용하는 명령은 실행 시에 「프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)」이 발생합니다.

*1. 서보 드라이버의 제어 모드 중 사이클릭 동기 위치 모드(CSP)를 말합니다.
*2. 서보 드라이버의 제어 모드 중 사이클릭 동기 속도 모드(CSV)를 말합니다.
*3. 서보 드라이버의 제어 모드 중 사이클릭 동기 토크 모드(CST)를 말합니다.

서보 드라이버 측의 오브젝트 설정

MC 기능 모듈의 제어 기능을 사용하기 위한 오므론 서보 드라이버 1S 시리즈 측의 설정을 아래에 표로 제시합니다.
서보 드라이버의 각종 기능을 사용할 경우에는 서보 드라이버의 매뉴얼에 따라 각종 기능과 관련된 오브젝트도 설정해 주십시오.

인덱스	서브 인덱스	명칭	권장 설정값	설명
3001Hex	-	기계	-	서보 측 기어비는 1:1로 하고 사용자 단위는 컨트롤러 측에서 설정합니다. *1
	05Hex	전자 기어비 분자	1	
	06Hex	전자 기어비 분모	1	
3330Hex	-	토크 제한	-	PCL와 NCL은 OFF 상태에서 「60E0Hex」와 「60E1Hex」의 값으로 제한합니다.
	01Hex	변환 선택	2	
3A00Hex	-	원점 복귀	-	서보 측 오프셋값은 「0」으로 사용합니다.
	06Hex	엔코더 원점 위치 오프셋	0	
3B10Hex	-	구동 금지	-	서보 측은 구동 금지 입력을 무효화하고 컨트롤러 측에서 처리합니다.
	01Hex	유효	0	
3B11Hex	-	소프트웨어 리미트	-	정방향, 역방향 모두 소프트웨어 리미트를 무효화합니다.
	01Hex	유효 선택	0	
3B30Hex	-	래치 기능 1	-	Touch probe1=외부 래치 신호 1, Touch probe2=외부 래치 신호 2로 설정합니다.
	01Hex	래치 1 트리거 선택	1	
3B31Hex	-	래치 기능 2	-	Touch probe1=외부 래치 신호 1, Touch probe2=외부 래치 신호 2로 설정합니다.
	01Hex	래치 2 트리거 선택	2	
4020Hex	-	경고 커스터마이즈	-	요인이 소멸되어 자동 리셋되되도록 설정합니다.
	04Hex	경고 유지 선택	0	
4510Hex	-	엔코더	-	절대값 엔코더로 사용합니다. 다회전 카운터 초과는 무시합니다.
	01Hex	절대값 엔코더 동작 선택	2	
4630Hex	-	정방향 구동 금지	-	정방향 구동 금지를 범용 입력 2(IN2)에 역논리(b접점)로 할당합니다.
	01Hex	포트 선택	2	
	02Hex	논리 선택	1	
4631Hex	-	역방향 구동 금지	-	역방향 구동 금지를 범용 입력 3(IN3)에 역논리(b접점)로 할당합니다.
	01Hex	포트 선택	3	
	02Hex	논리 선택	1	
4632Hex	-	외부 래치 입력 1	-	외부 래치 입력 1을 범용 입력 7(IN7)에 정논리(a접점)로 할당합니다.
	01Hex	포트 선택	7	
	02Hex	논리 선택	0	
4633Hex	-	외부 래치 입력 2	-	외부 래치 입력 2를 범용 입력 8(IN8)에 정논리(a접점)로 할당합니다.
	01Hex	포트 선택	8	
	02Hex	논리 선택	0	
4634Hex	-	원점 근방 입력	-	원점 근방 입력을 범용 입력 4(IN4)에 정논리(a접점)로 할당합니다.
	01Hex	포트 선택	4	
	02Hex	논리 선택	0	

*1. 유니트 버전 1.10 이하인 CPU 유니트의 경우 오므론 서보 모터 1S 시리즈를 최고 회전수로 동작시킬 수 없습니다.
1S 시리즈를 최고 회전수로 동작시키려면 전자 기어비를 2:1 이상으로 설정해 주십시오.

A-2 서보 드라이버 G5 시리즈와의 접속

오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 타입과의 접속에 대해 설명합니다.

A-2-1 서보 드라이버와의 배선

서보 드라이버와는 EtherCAT 통신으로 접속합니다.

접속 방법에 대해서는 🕮『NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용자 매뉴얼(SBCD-358)』을 참조해 주십시오.

A-2-2 서보 드라이버의 설정

여기에서는 MC 기능 모듈의 제어 대상인 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 타입과 접속한 경우의 서보 드라이버 측 설정 개요에 대해 설명합니다.

서보 드라이버에 대한 상세한 내용은 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼(SBCE-365)』 및 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입 사용자 매뉴얼(SBCE-366)』을 참조해 주십시오.

권장 서보 드라이버

MC 기능 모듈의 모든 기능을 사용할 수 있는 서보 드라이버의 유니트 버전을 아래에 표로 제시합니다.

제조 회사	대상 형식	대상 유니트 버전
오므론	R88D-KN□□□-ECT	Ver.2.1 이상
	R88D-KN□□□-ECT-L	Ver.1.1 이상

참고

- 권장 대상 외의 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 타입의 유니트 버전도 사용할 수 있습니다. 사용할 수 있는 기능은 서보 드라이버 측의 사양에 따라 다릅니다. Sysmac Studio에서 사용하는 기능과 오브젝트 딕셔너리를 설정해 주십시오.
- 오므론 서보 드라이버 G5 시리즈의 R88D-KN□□□-ECT-R(유니트 버전 Ver.1.0)은 위치 제어(cyclic synchronous position)에만 대응하는 서보 드라이버입니다.

  속도 제어(cyclic synchronous velocity) 또는 토크 제어(cyclic synchronous torque)가 불필요한 어플리케이션에서 사용할 수 있습니다.

  기능에 대한 상세한 내용은 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼(SBCE-360)』을 참조해 주십시오.
- 오므론 서보 모터/드라이버 G5 시리즈 중 실린더 타입의 유니트 버전 Ver.2.0 이하를 사용할 경우에는 노드 주소 스위치를「00」이외로 설정해 주십시오.「00」으로 설정하면 네트워크 구성에 이상이 발생합니다.
- 오므론 서보 드라이버 G5 시리즈의 유니트 버전별 차이는 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼(SBCE-365B 이후)』을 참조해 주십시오.

외부 입력 신호 할당

MC 기능 모듈에서는 서보 드라이버의 범용 입력을 아래의 입력 신호로 사용합니다.

- 즉시 정지 입력
- 정방향 한계 입력
- 역방향 한계 입력
- 원점 근방 입력
- 외부 래치 기능의 트리거 신호(래치 입력 1/래치 입력 2)

● 정방향 한계 입력/역방향 한계 입력/원점 근방 입력 할당

오므론 서보 드라이버 G5 시리즈의 출하 시 설정에서는 각 입력 신호가 아래의 표와 같이 할당되어 있습니다.

신호 명칭	입력 신호
즉시 정지 입력	서보 드라이버의 범용 입력 1(IN1: CN1 커넥터의 핀 No.5 b접점)
정방향 한계 입력	서보 드라이버의 범용 입력 2(IN2: CN1 커넥터의 핀 No.7 b접점) *1
역방향 한계 입력	서보 드라이버의 범용 입력 3(IN3: CN1 커넥터의 핀 No.8 b접점) *2
원점 근방 입력	서보 드라이버의 범용 입력 4(IN4: CN1 커넥터의 핀 No.9 a접점)

*1. 서보 드라이버 측의 신호 명칭은「정회전 측 구동 금지 입력」입니다.
*2. 서보 드라이버 측의 신호 명칭은「역회전 측 구동 금지 입력」입니다.

● 외부 래치 기능의 트리거 신호 할당

오므론 서보 드라이버 G5 시리즈의 출하 시 설정의 경우 아래 표의 입력 신호가 외부 래치 기능의 트리거 신호로 할당되어 있습니다.

MC_TouchProbe 명령의 입력 변수「TriggerInput(트리거 입력 조건)」 설정			외부 래치 기능의 트리거 신호
Mode	InputDrive	LatchID	
0: mcDrive	0: mcEncoderMark	-	엔코더의 Z상
	1: mcEXT	1: mcLatch1	서보 드라이버의 범용 입력 7 (IN7: CN1 커넥터의 핀 No.12 a접wja) *1
		2: mcLatch2	서보 드라이버의 범용 입력 6 (IN6: CN1 커넥터의 핀 No.11 a접점) *2
1: mcController	-	-	TriggerVariable(트리거 변수)로 지정한 변수

*1. 서보 드라이버 측의 신호 명칭은「외부 래치 입력 1」입니다.
*2. 서보 드라이버 측의 신호 명칭은「외부 래치 입력 2」입니다.

백래시 보정 기능

MC 기능 모듈은 백래시 보정 기능이 없습니다.
백래시 보정이 필요한 경우에는 서보 드라이버 측의 보정 기능을 사용합니다.
서보 드라이버 측에서 설정이 필요한 오브젝트는 아래와 같습니다.

인덱스	명칭	설명
3704Hex	백래시 보정 선택	위치 제어 시 백래시 보정의 유효/무효 및 보정 시의 동작 방향을 설정합니다. 출하 시 설정은「무효」입니다.
3705Hex	백래시 보정량	위치 제어 시의 백래시 보정량을 설정합니다.
3706Hex	백래시 보정 시정수	위치 제어 시의 백래시 보정 시정수를 설정합니다.

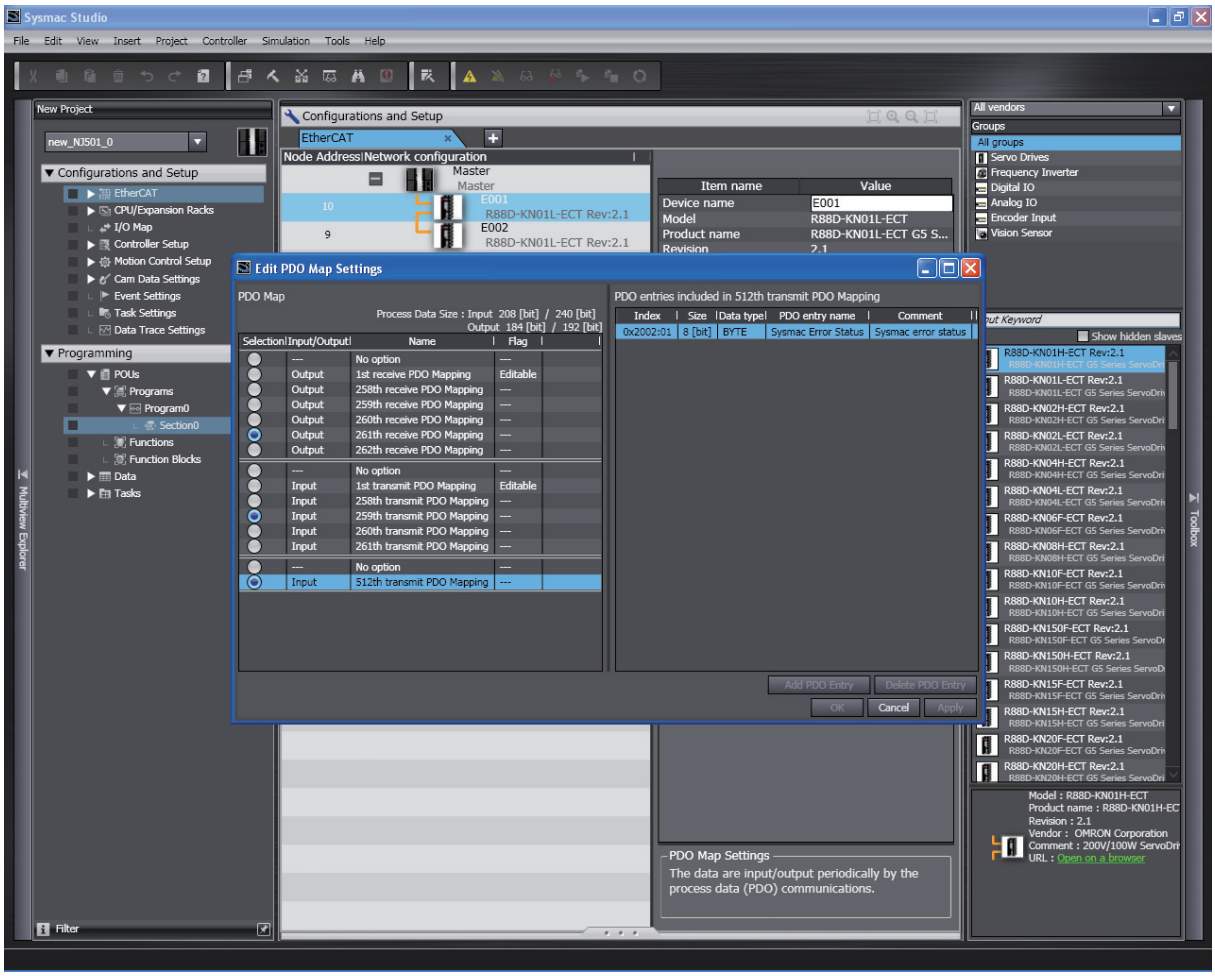
백래시 기능에 대한 상세한 내용은 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼(SBCE-365)』 및 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입 사용자 매뉴얼(SBCE-366)』을 참조해 주십시오.

PDO 매핑

MC 기능 모듈의 서보 축으로 제어하기 위한 PDO 매핑에 대해 설명합니다.

모션 제어의 각 기능을 사용하려면 사용할 기능에 필요한 오브젝트를 프로세스 데이터 통신에 매핑해야 합니다.
필요한 오브젝트를 미리 정리한 것이 PDO 맵입니다.

사용할 PDO 맵은 Sysmac Studio의 [EtherCAT] 구성 편집 화면의 [Edit PDO Map Settings]에서 선택합니다.



Sysmac Studio의 디폴트에는 아래의 PDO 맵이 선택되어 있습니다.

RxPDO: [261th receive PDO Mapping] (1704Hex)	컨트롤 워드(6040Hex), 목표 위치(607AHex), 목표 속도(60FFHex), 목표 토크(6071Hex), 오퍼레이션 모드(6060Hex), 래치 기능 (60B8Hex), 최대 프로파일 속도(607FHex), 정회전 측 토크 제한값(60E0Hex), 역회전 측 토크 제한값(60E1Hex)
TxPDO: [259th transmit PDO Mapping] (1B02Hex)	에러 코드(603FHex), 상태 워드(6041Hex), 피드백 위치(6064Hex), 피드백 토크(6077Hex), 오퍼레이션 모드 표시(6061Hex), 래치 상태(60B9Hex), 래치 위치 1(60BAHex), 래치 위치 2 (60BCHex), 디지털 입력(60FDHex)

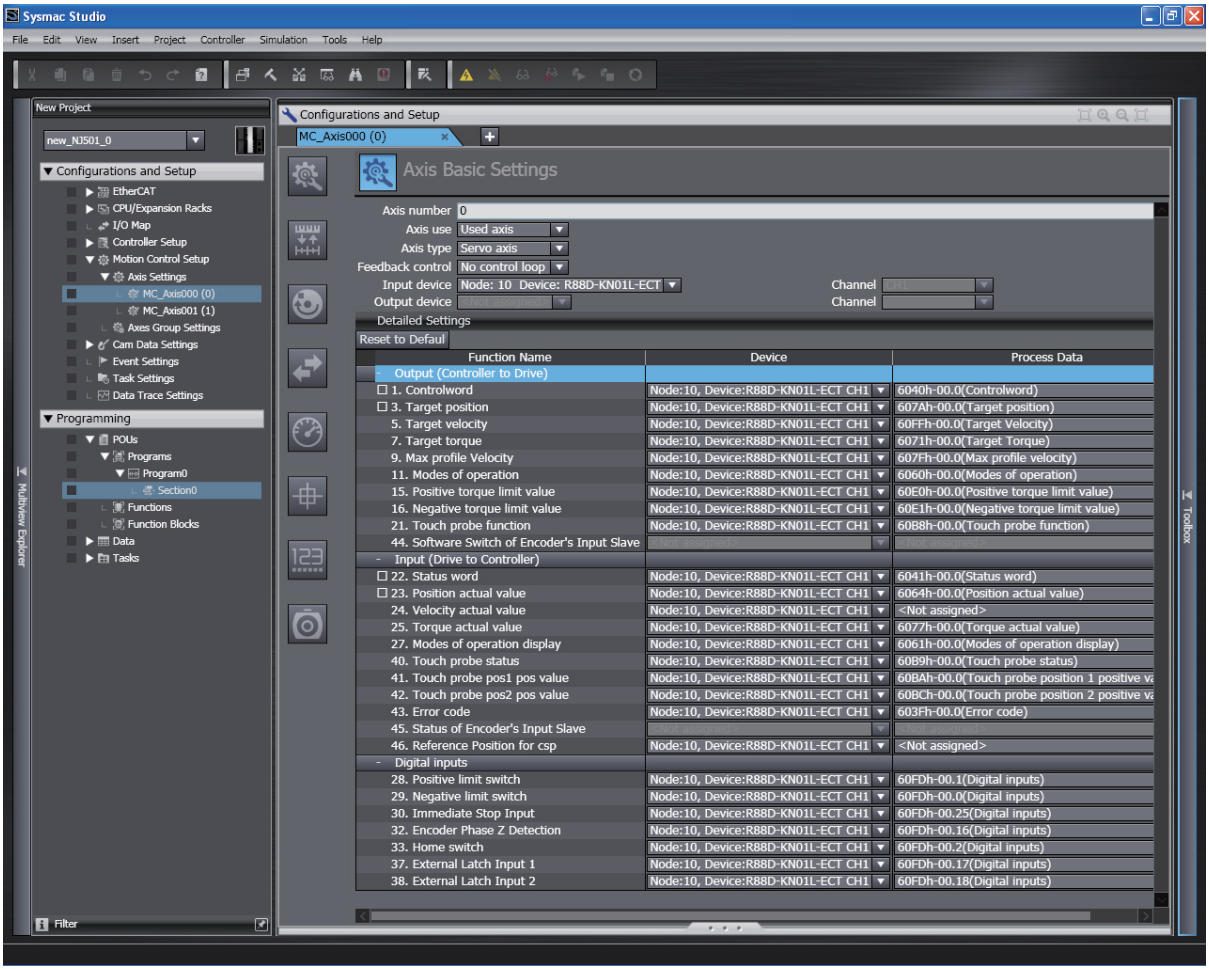
참고

- 권장 서보 드라이버인 오므론 R88D-KN□□□-ECT(Ver.2.1 이상) 또는 R88D-KN□□□-ECT-L(Ver.1.1 이상)의 경우 PDO 맵은 Sysmac Studio의 디폴트 상태에서도 사용할 수 있습니다.
- 오므론의 R88D-KN□□□-ECT에서 풀 클로즈 제어를 실행할 경우 RxPDO에는 1701Hex 또는 오브젝트의 합계 사이즈를 12바이트 이하로 한 1600Hex(Ver.2.1 이상의 경우)를 선택해 주십시오.

MC 기능 모듈의 기능과 프로세스 데이터의 관계

MC 기능 모듈의 기능은 프로세스 데이터의 오브젝트 정보와 관계가 있습니다.
MC 기능 모듈이 사용하는 EtherCAT 슬레이브의 구성 및 사용 기능에 따라서는 이러한 관계를 변경해야 하는 경우가 있습니다.

Sysmac Studio의 [Axis Basic Settings] 화면의 [Detailed Settings]를 클릭하면 설정 화면이 열립니다.



참고

권장 서보 드라이버의 오므론 R88D-KN□□□-ECT(Ver.2.1 이상) 또는 R88D-KN□□□-ECT-L(Ver.1.1 이상)의 경우 기능과 프로세스 데이터와의 관계는 Sysmac Studio의 디폴트 상태에서도 사용할 수 있습니다.

● 출력(컨트롤러→드라이버) 설정

MC 기능 모듈에서 서보 드라이버로 보내는 지령 데이터 설정입니다.
아래는 Sysmac Studio의 디폴트 설정입니다. (○표시는 필수 오브젝트)

기능명	프로세스 데이터	설명
○ 컨트롤 워드	6040Hex-00.0 (Controlword)	서보 드라이버의 상태를 제어하기 위한 데이터입니다. [6040Hex: Controlword]를 설정해 주십시오.
○ 목표 위치	607AHex-00.0 (Target position)	위치 제어에서의 목표 위치입니다. [607AHex: Target position]을 설정해 주십시오.
목표 속도	60FFHex-00.0 (Target velocity)	속도 제어에서의 목표 속도입니다. MC_SyncMoveVelocity(사이클릭 동기 속도 제어) 명령 등 사이클릭 동기 속도 모드에서 드라이버로 출력하는 경우에 필요합니다. 일반적으로는 [60FFHex: Target velocity]를 설정해 주십시오.
목표 토크	6071Hex-00.0 (Target torque)	토크 제어에서의 목표 토크입니다. MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 토크 모드에서 드라이버로 출력하는 경우에 필요합니다. 일반적으로는 [6071Hex: Target torque]를 설정해 주십시오.
최대 프로파일 속도	607FHex-00.0 (Max profile velocity)	토크 제어 시의 속도 제한값입니다. MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 토크 모드에서 속도를 제한하는 경우에 필요합니다. 일반적으로는 [607FHex: Max profile velocity]를 설정해 주십시오.
오퍼레이션 모드	6060Hex-00.0 (Modes of operation)	제어 모드를 변환하기 위한 데이터입니다. MC_SyncMoveVelocity(사이클릭 동기 속도 제어) 명령이나 MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 위치 모드 이외의 제어 모드로 변환하는 경우에 필요합니다. 일반적으로는 [6060Hex: Modes of operation]을 설정해 주십시오. *1
정회전 측 토크 제한값	60E0Hex-00.0 (Positive torque limit value)	정방향 측의 토크 제한값입니다. MC_SetTorqueLimit(토크 제한) 명령 등 드라이버의 출력 토크를 제한하는 경우에 필요합니다. 일반적으로는 [60E0Hex: Positive torque limit value]를 설정해 주십시오.
역회전 측 토크 제한값	60E1Hex-00.0 (Negative torque limit value)	역방향 측의 토크 제한값입니다. MC_SetTorqueLimit(토크 제한) 명령 등 드라이버의 출력 토크를 제한하는 경우에 필요합니다. 일반적으로는 [60E1Hex: Negative torque limit value]를 설정해 주십시오.
래치 기능	60B8Hex-00.0 (Touch probe function)	래치 기능을 제어하기 위한 데이터입니다. MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령, MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령 등 래치 기능을 사용할 경우에 필요합니다. 일반적으로는 [60B8Hex: Touch probe function]을 설정해 주십시오.

*1. [6060Hex: Modes of operation]을 설정할 경우에는 [6061Hex: Modes of operation display]도 설정해 주십시오. 둘 중 하나만 설정하면 정상적으로 동작하지 않습니다.

사용상의 주의

- 권장 서보 드라이버 이외의 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 타입의 유니트 버전을 접속한 경우 대응하는 기종이 없을 수 있습니다.
  접속할 서보 드라이버의 매뉴얼을 확인해 주십시오.
- 설정 변경 시에는 MC 기능 모듈의 기능과 프로세스 데이터의 조합이 의도한 대로 동작하는지 확인해 주십시오.
- 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입 또는 G5 시리즈 EtherCAT 통신 내장 타입 이외를 사용할 경우에는 오퍼레이션 모드(6060Hex)를 반드시 설정해 주십시오.
- 오므론 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 타입에서 풀 클로즈 제어를 실행할 경우 오브젝트의 합계 사이즈가 12바이트 이하가 되도록 설정해 주십시오.

● 입력(드라이버→컨트롤러) 설정

서보 드라이버에서 MC 기능 모듈로 보내는 상태 데이터입니다.
아래는 Sysmac Studio의 디폴트 설정입니다. (○표시는 필수 오브젝트)

	기능명	프로세스 데이터	설명
○	상태 워드	6041Hex-00.0 (Statusword)	서보 드라이버의 상태입니다. [6041Hex: Statusword]를 설정해 주십시오.
○	피드백 위치	6064Hex-00.0 (Position actual value)	현재 위치 표시입니다. [6064Hex: Position actual value]를 설정해 주십시오.
	피드백 속도	설정 없음 *1	현재 속도 표시입니다. 사용하려면 일반적으로 [606CHex: Velocity actual value]를 설정해 주십시오.
	피드백 토크	6077Hex (Torque actual value)	현재 토크 표시입니다. MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 토크 모드에서 드라이버로 출력하는 경우에 필요합니다. 일반적으로는 [6077Hex: Torque actual value]를 설정해 주십시오.
	오퍼레이션 모드 표시	6061Hex-00.0 (Modes of operation display)	제어 모드 표시입니다. MC_SyncMoveVelocity(사이클릭 동기 속도 제어) 명령이나 MC_TorqueControl(토크 제어) 명령 등 사이클릭 동기 위치 모드 이외의 제어 모드로 변환하는 경우에 필요합니다. 일반적으로는 [6061Hex: Modes of operation display]를 설정해 주십시오. *2
	래치 상태	60B9Hex-00.0 (Touch probe status)	래치 기능의 상태 표시입니다. MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령, MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령 등 래치 기능을 사용할 경우에 필요합니다. 일반적으로는 [60B9Hex: Touch probe status]를 설정해 주십시오.
	래치 위치 1	60BAHex-00.0 (Touch probe pos1 pos value)	래치 기능 1에서 래치한 위치입니다. MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령, MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령 등 래치 기능을 사용할 경우에 필요합니다. 일반적으로는 [60BAHex: Touch probe pos1 pos value]를 설정해 주십시오.

기능명	프로세스 데이터	설명
래치 위치 2	60BCHex-00.0 (Touch probe pos2 pos value)	래치 기능 2에서 래치한 위치입니다. MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령, MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령 등 래치 기능을 사용할 경우에 필요합니다. 일반적으로는 [60BCHex: Touch probe pos2 pos value]를 설정해 주십시오.
에러 코드	603FHex-00.0 (Error code)	드라이버 내의 이상 코드입니다. 일반적으로는 [603FHex: Error code]를 설정해 주십시오.
csp용 참조 위치	설정 없음	csp 모드 변환 시의 참조 위치입니다. 속도 제어 모드(CSV)를 사용하는 명령 및 토크 제어 모드(CST)를 사용하는 명령에서 참조되는 데이터입니다. 오므론 서보 모터/드라이버 G5 시리즈 중 실린더 타입의 유니트 버전 Ver.2.1 이상에서 사용할 수 있습니다. *3 오므론 서보 드라이버 G5 시리즈 중 리니어 모터 타입에서는 이 오브젝트에 대응하지 않습니다.

*1. 필요한 경우에는 선택할 프로세스 데이터를 PDO에 매핑한 후에 설정해 주십시오. 606CHex-00.0(Velocity actual value)이 표준 설정입니다.
*2. [6061Hex: Modes of operation display]를 설정할 경우에는 [6060Hex: Modes of operation]도 설정해 주십시오. 둘 중 하나만 설정하면 정상적으로 동작하지 않습니다.
*3. 사용하려면 [4020Hex: Reference Position for csp]를 PDO에 매핑해 주십시오.

사용상의 주의

- 권장 서보 드라이버 이외의 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 타입의 유니트 버전을 접속한 경우 대응하는 기종이 없을 수 있습니다. 접속할 서보 드라이버의 매뉴얼을 확인해 주십시오.
- 설정 변경 시에는 MC 기능 모듈의 기능과 프로세스 데이터의 조합이 의도한 대로 동작하는지 확인해 주십시오.
- 오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입 또는 G5 시리즈 EtherCAT 통신 내장 타입 이외를 사용할 경우에는 오퍼레이션 모드 표시(6061Hex)를 반드시 설정해 주십시오.
- 오므론 서보 드라이버 G5 시리즈를 사용하여 MC_SyncMoveVelocity(사이클릭 동기 속도 제어) 명령으로 제어 모드를 변환할 경우에는 「csp용 참조 위치」를 매핑해 주십시오.
- 오므론 서보 드라이버 G5 시리즈에서 「csp용 참조 위치」를 사용할 경우 프라이머리 주기는 1ms로 해 주십시오. 또한, 전자 기어비는 1:1로 해 주십시오.

  상세한 내용은 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 타입 사용자 매뉴얼(SBCE-365)』 및 🕮『AC 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입 사용자 매뉴얼(SBCE-366)』을 참조해 주십시오.

● 디지털 입력 설정

MC 기능 모듈이 사용하는 서보 드라이버의 입력 신호입니다.

기능명	프로세스 데이터	설명
정회전 측 구동 금지 입력	60FDHex-00.1 (Digital inputs)	정방향 한계 입력으로 사용하는 신호입니다. 일반적으로는 [60FDHex-00: Digital inputs]의 [Bit 1: Positive limit switch]를 설정해 주십시오.
역회전 측 구동 금지 입력	60FDHex-00.0 (Digital inputs)	역방향 한계 입력으로 사용하는 신호입니다. 일반적으로는 [60FDHex-00: Digital inputs]의 [Bit 0: Negative limit switch]를 설정해 주십시오.
즉시 정지 입력	60FDHex-00.25 (Digital inputs)	즉시 정지 입력으로 사용하는 신호입니다. 오므론 서보 드라이버 G5 시리즈에서는 [60FDHex-00: Digital inputs]의 [Bit 25: Immediate Stop Input]을 설정해 주십시오.
엔코더 Z상 검출	60FDHex-00.16 (Digital inputs)	Z상 입력의 검출 상태 표시입니다. 오므론 서보 모터/드라이버 G5 시리즈 중 실린더 타입의 유니트 버전 Ver.2.1 이상에서는 [60FDHex-00: Digital inputs]의 [Bit 16: Encoder Phase Z Detection]을 설정해 주십시오. 서보 드라이버 G5 시리즈 중 리니어 모터 타입에서는 이 오브젝트에 대응하지 않습니다.
원점 근방 입력	60FDHex-00.2 (Digital inputs)	원점 근방 입력으로 사용하는 신호입니다. 일반적으로는 [60FDHex-00: Digital inputs]의 [Bit 2: Home switch]를 설정해 주십시오.
외부 래치 입력 1	60FDHex-00.17 (Digital inputs)	외부 래치 입력 1로 사용하는 신호의 상태 표시입니다. 오므론 서보 드라이버 G5 시리즈에서는 [60FDHex-00: Digital inputs]의 [Bit 17: External Latch Input 1]를 설정해 주십시오.
외부 래치 입력 2	60FDHex-00.18 (Digital inputs)	외부 래치 입력 2로 사용하는 신호의 상태 표시입니다. 오므론 서보 드라이버 G5 시리즈에서는 [60FDHex-00: Digital inputs]의 [Bit 18: External Latch Input 2]를 설정해 주십시오.

사용상의 주의

- 권장 서보 드라이버 이외의 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 타입의 유니트 버전을 접속한 경우 대응하는 기종이 없을 수 있습니다. 접속할 서보 드라이버의 매뉴얼을 확인해 주십시오.
- 설정 변경 시에는 MC 기능 모듈의 기능과 프로세스 데이터의 조합이 의도한 대로 동작하는지 확인해 주십시오.

버전 관련 정보

- 유니트 버전 Ver.1.09 이하인 CPU 유니트에서 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 타입 이외를 사용할 경우에는 오퍼레이션 모드(6060Hex)와 오퍼레이션 모드 표시(6061Hex)가 필요합니다.
- 유니트 버전 Ver.1.10 이상인 CPU 유니트의 경우 오퍼레이션 모드(6060Hex) 및 오퍼레이션 모드 표시(6061Hex)의 매핑 유무에 따라 다음과 같이 동작합니다.

	오퍼레이션 모드 표시	
	(6061Hex) 있음	(6061Hex) 없음
오퍼레이션 모드 (6060Hex) 있음	・CSP*1, CSV*2, CST*3를 사용하는 명령을 실행할 수 있습니다 ・CSP, CSV, CST 이외일 때는 서보 OFF가 됩니다.	・CSP를 사용하는 명령을 실행할 수 있습니다. 그 외 제어 모드를 사용하는 명령은 실행 시에 「프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)」이 발생합니다. ・MC 기능 모듈은 서보 드라이버의 제어 모드를 CSP로 간주합니다. 서보 드라이버에 CSP를 지령합니다.
오퍼레이션 모드 (6060Hex) 없음	・CSP를 사용하는 명령을 실행할 수 있습니다. 그 외 제어 모드를 사용하는 명령은 실행 시에 「프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)」이 발생합니다. ・CSP 이외일 때는 서보 OFF가 됩니다.	CSP를 사용하는 명령을 실행할 수 있습니다. 그 외 제어 모드를 사용하는 명령은 실행 시에 「프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)」이 발생합니다.

*1. 서보 드라이버의 제어 모드 중 사이클릭 동기 위치 모드(CSP)를 말합니다.
*2. 서보 드라이버의 제어 모드 중 사이클릭 동기 속도 모드(CSV)를 말합니다.
*3. 서보 드라이버의 제어 모드 중 사이클릭 동기 토크 모드(CST)를 말합니다.

서보 드라이버 측의 오브젝트 설정

MC 기능 모듈의 제어 기능을 사용하기 위한 오므론 서보 드라이버 G5 시리즈 측의 설정을 아래에 표로 제시합니다.
서보 드라이버의 각종 기능을 사용할 경우에는 서보 드라이버의 매뉴얼에 따라 각종 기능에 관한 오브젝트도 설정해 주십시오.

인덱스	서브 인덱스	명칭	권장 설정값	설명
3013Hex	00Hex	제1 외부 토크 제한 *1	1388Hex	초기값=500.0%
3015Hex *2	00Hex	절대값 엔코더 사용 시 동작 변환	0002Hex	절대값으로 이용하여 다회전 카운터 초과를 무시합니다.
3317Hex	00Hex	속도 제한 선택	0001Hex	토크 제어 시의 속도 제한 방법은 607FHex: Max profile velocity와 3321Hex: 속도 제한 값 설정이 작은 쪽의 값으로 제한합니다.
3323Hex	00Hex	외부 피드백 펄스 타입 선택	0000Hex 또는 0001Hex, 0002Hex *3	사용하는 외부 스케일 타입의 설정입니다. 초기값인 0000Hex는 「90° 위상차 출력 타입」입니다.
3324Hex *2	00Hex	외부 피드백 펄스 분주비 분자	00000000Hex	모터 1회전당 엔코더 분해능을 설정합니다. 「0」으로 설정하면 자동 설정됩니다.
3401Hex	00Hex	입력 신호 선택 2	00818181Hex	정회전 측 구동 금지 입력(b접점)
3402Hex	00Hex	입력 신호 선택 3	00828282Hex	역회전 측 구동 금지 입력(b접점)

인덱스	서브 인덱스	명칭	권장 설정값	설명
3403Hex	00Hex	입력 신호 선택 4	00222222Hex	원점 근방 입력(a접점)
3404Hex	00Hex	입력 신호 선택 5	002B2B2BHex	외부 래치 신호 3(a접점)
3405Hex	00Hex	입력 신호 선택 6	00212121Hex	외부 래치 신호 2(a접점)
3406Hex	00Hex	입력 신호 선택 7	00202020Hex	외부 래치 신호 1(a접점)
3504Hex	00Hex	구동 금지 입력 선택	0001Hex	서보 측은 구동 금지 입력을 무효화합니다. MC 기능 모듈 측에서 실행합니다.
3508Hex	00Hex	부족 전압 에러 선택	0001Hex	주전원 부족 전압이 발생하고 정지합니다.
3521Hex	00Hex	토크 제한 선택 *1	0006Hex	정역 모두 2개씩의 제한값을 가집니다. 변환은 PCL, NCL에서 실행합니다.
3522Hex	00Hex	제2 외부 토크 제한 *1	1388Hex	초기값=500.0%
3703Hex	00Hex	토크 제한 중 출력 설정 *1	0001Hex	토크 지령값을 제외한 토크 제한에서 ON 합니다.
3801Hex	00Hex	소프트웨어 리미트 기능	0003Hex	양쪽 소프트웨어 리미트를 무효화합니다.
3758Hex	00Hex	래치 트리거 선택	0100Hex	Touch probe1 =외부 래치 신호 1, Touch probe2 =외부 래치 신호 2로 합니다.
3759Hex	00Hex	경고 유지 선택	0000Hex	경고 요인이 없어지면 경고는 자동으로 지워집니다.
607CHex	00Hex	엔코더 원점 위치 오프셋	00000000Hex	서보 측 오프셋 값은「0」으로 사용.
6091Hex	01Hex	전자 기어비 분자	00000001Hex	서보 드라이버 측의 기어비는 1:1로 합니다. 동일한 기능은 MC 기능 모듈 측에서 설정합니다.
	02Hex	전자 기어비 분모	00000001Hex	

*1. 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입에서는「토크」를「추력」으로 대체해 주십시오.
*2. 오므론 서보 드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입에서는 이 오브젝트에 대응하지 않습니다.
*3. 오므론 서보 모터/드라이버 G5 시리즈에서 풀 클로즈 제어할 경우 또는 오므론 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입의 경우 사용하는 외부 스케일 타입에 맞춰 설정해 주십시오.

A-3 엔코더 입력 터미널과의 접속

오므론 GX 시리즈 EtherCAT 슬레이브 엔코더 입력 터미널과의 접속에 대해 설명합니다.

A-3-1 엔코더 입력 터미널과의 배선

엔코더 입력 터미널이란 EtherCAT 통신으로 접속합니다.
접속 방법에 대해서는 🕮『NJ/NX 시리즈 CPU 유니트 내장 EtherCAT 포트 사용자 매뉴얼(SBCD-358)』을 참조해 주십시오.

A-3-2 엔코더 입력 터미널 설정

여기에서는 MC 기능 모듈의 제어 대상인 오므론 EtherCAT 리모트 I/O 터미널 GX 시리즈 GX-EC0211/EC0241과 접속한 경우의 엔코더 입력 터미널 측 설정 개요에 대해 설명합니다.
엔코더 입력 터미널에 대한 상세한 내용은 🕮『GX 시리즈 EtherCAT 슬레이브 사용자 매뉴얼(SBCD-350)』을 참조해 주십시오.

권장 엔코더 입력 터미널

MC 기능 모듈의 엔코더 축으로 모든 기능을 사용할 수 있는 엔코더 입력 터미널의 유니트 버전을 아래에 표로 제시합니다.

제조 회사	대상 형식	대상 유니트 버전
오므론	GX-EC0211	Ver.1.1 이상
	GX-EC0241	Ver.1.1 이상

참고

- 엔코더 축으로 사용할 수 있는 EtherCAT 슬레이브는 오므론 GX-EC0211/EC0241뿐입니다.
- 오므론 GX-EC0211/EC0241(유니트 버전 Ver.1.0)도 엔코더 축으로 사용할 수 있으나 Sysmac 디바이스에서는 사용할 수 없습니다.

  유니트 버전 Ver.1.0을 사용할 경우 노드 주소 스위치는「00」이외로 설정해 주십시오.
  「00」으로 설정하면 네트워크 구성에 이상이 발생합니다.

  기능에 대한 상세한 내용은 🕮『GX 시리즈 EtherCAT 슬레이브 사용자 매뉴얼(SBCD-350)』을 참조해 주십시오.

  Sysmac 디바이스에 대해서는 🕮『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

외부 입력 신호

엔코더 축으로 모든 기능을 사용할 경우 아래의 엔코더 입력 터미널 측 입력 신호를 사용합니다.

- 카운터 A상
- 카운터 B상
- 카운터 Z상
- 래치 입력(A/B)

카운터는 2채널이 있으며 채널마다 2개의 외부 래치가 있습니다. 실제로 사용하는 입력 신호를 배선해 주십시오.

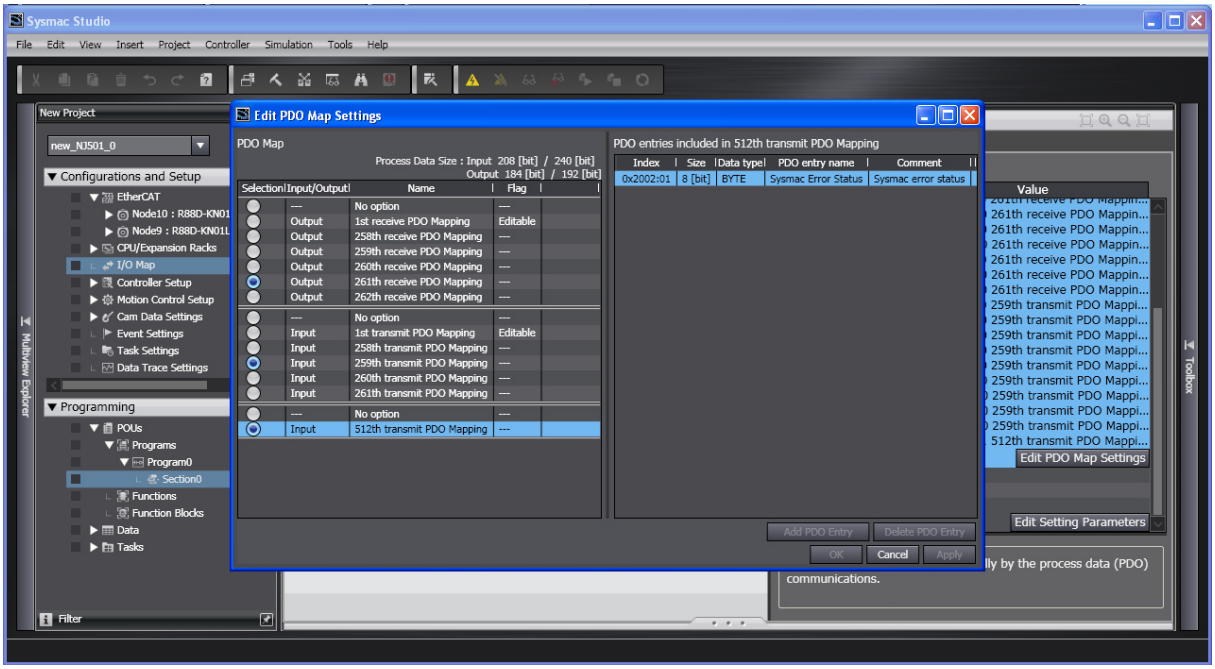
입력 신호와 배선 방법은 📖『GX 시리즈 EtherCAT 슬레이브 사용자 매뉴얼(SBCD-350)』을 참조해 주십시오.

PDO 매핑

MC 기능 모듈의 엔코더 축으로 제어하기 위한 PDO 매핑에 대해 설명합니다.

모션 제어의 각 기능을 사용하려면 사용할 기능에 필요한 오브젝트를 프로세스 데이터 통신에 매핑해야 합니다.
필요한 오브젝트를 미리 정리한 것이 PDO 맵입니다.

사용할 PDO 맵은 Sysmac Studio의 [EtherCAT] 구성 편집 화면의 [Edit PDO Map Settings]에서 선택합니다.



Sysmac Studio의 디폴트에는 아래의 PDO 맵이 선택되어 있습니다.

RxPDO(1700Hex)	채널 1 소프트 스위치(4020Hex-01Hex), 채널 2 소프트 스위치(4020Hex-02Hex)
RxPDO(1701Hex)	채널 1 프리셋 현재값(4011Hex-01Hex), 채널 2 프리셋 현재값(4011Hex-02Hex)
TxPDO(1B00Hex)	채널 1 카운터 현재값(4010Hex-01Hex), 채널 2 카운터 현재값(4010Hex-02Hex)
TxPDO(1B01Hex)	채널 1 래치값 A(4012Hex-01Hex), 채널 2 래치값 A(4012Hex-02Hex)
TxPDO(1B02Hex)	채널 1 래치값 B(4013Hex-01Hex), 채널 2 래치값 B(4013Hex-02Hex)
TxPDO(1B03Hex)	채널 1 상태 비트(4030Hex-01Hex), 채널 2 상태 비트(4030Hex-02Hex)
TxPDO(1BFFHex)	Sysmac 이상 상태(2002Hex-01Hex)

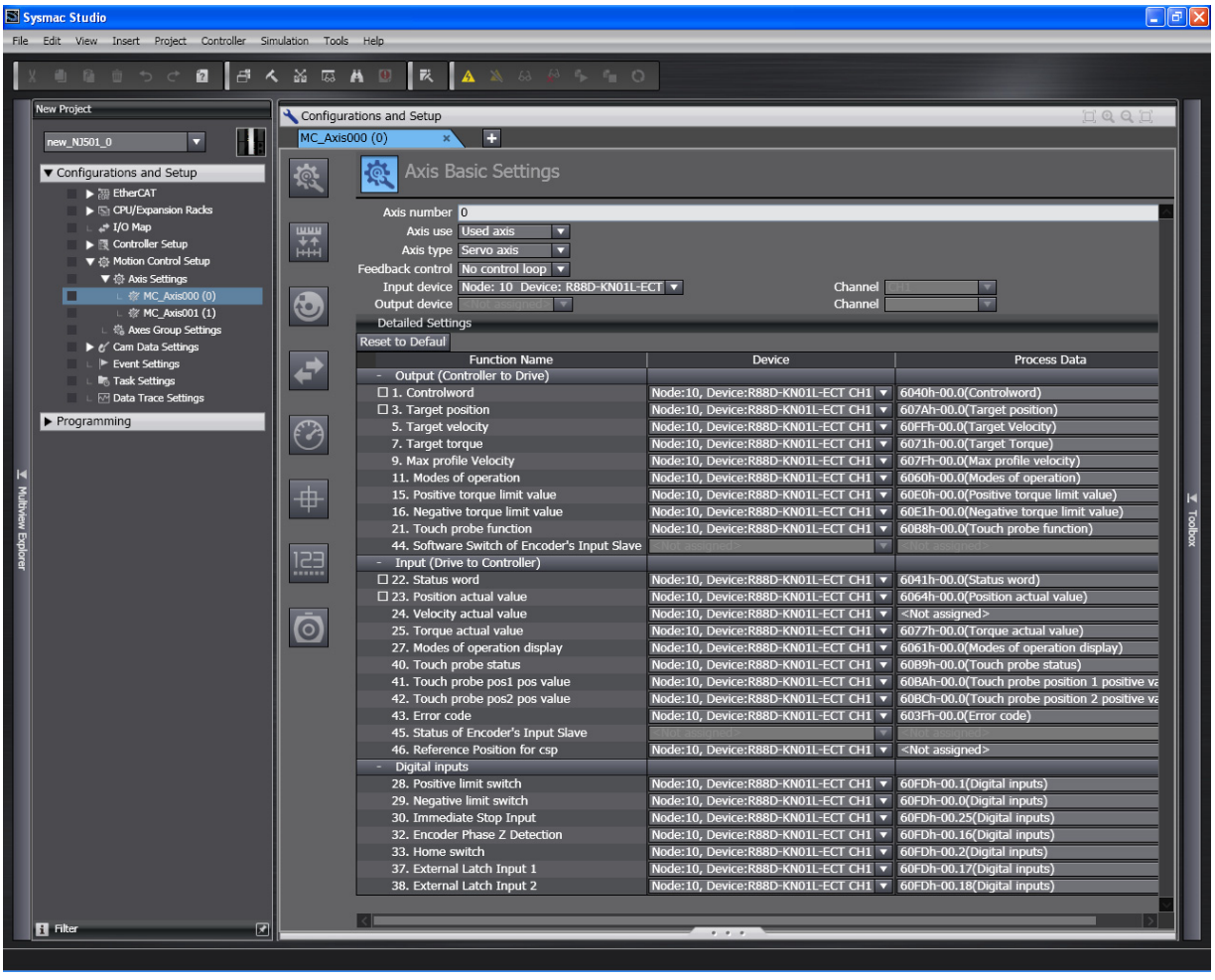
참고

권장 엔코더 입력 터미널인 GX-EC0211/EC0241(유니트 버전 Ver.1.1 이상)의 경우 PDO 맵은 Sysmac Studio의 디폴트 상태에서도 사용할 수 있습니다.

MC 기능 모듈의 기능과 프로세스 데이터의 관계

MC 기능 모듈의 기능은 프로세스 데이터의 오브젝트 정보와 관계가 있습니다.
MC 기능 모듈이 사용하는 EtherCAT 슬레이브의 구성 및 사용 기능에 따라서는 이러한 관계를 변경해야 하는 경우가 있습니다.

Sysmac Studio의 [Axis Basic Settings] 화면의 [Detailed Settings]를 클릭하면 설정 화면이 열립니다.



참고

권장 엔코더 입력 터미널인 GX-EC0211/EC0241(유니트 버전 Ver.1.1 이상)의 경우 기능과 프로세스 데이터와의 관계는 Sysmac Studio의 디폴트 상태에서도 사용할 수 있습니다.

● 출력(컨트롤러→드라이버) 설정

MC 기능 모듈에서 엔코더 입력 터미널로 보내는 지령 데이터 설정입니다.
아래는 Sysmac Studio의 디폴트 설정입니다. (○표시는 필수 오브젝트)

기능명		프로세스 데이터		설명
		채널 1	채널 2	
○	엔코더 입력 슬레이브 소프트 스위치	4020Hex-01.0 (Instruction Bits)	4020Hex-02.0 (Instruction Bits)	소프트 스위치의 내용을 설정합니다. 사용할 채널에 따라 좌측의 오브젝트를 설정해 주십시오.

사용상의 주의

설정 변경 시에는 MC 기능 모듈의 기능과 프로세스 데이터의 조합이 의도한 대로 동작하는지 확인해 주십시오.

● 입력(드라이버→컨트롤러) 설정

엔코더 입력 터미널에서 MC 기능 모듈로 보내는 상태 데이터입니다.
아래는 Sysmac Studio의 디폴트 설정입니다. (○표시는 필수 오브젝트)

기능명		프로세스 데이터		설명
		채널 1	채널 2	
○	피드백 위치	4010Hex-01.0 (Position Value)	4010Hex-02.0 (Position Value)	카운터의 현재값이 저장되어 있습니다. 사용할 채널에 따라 좌측의 오브젝트를 설정해 주십시오.
	래치 위치 1	4012Hex-01.0 (Latch Value A)	4012Hex-02.0 (Latch Value A)	래치 기능 1에서 래치한 위치입니다. 래치 A의 값이 저장되어 있습니다. MC_TouchProbe(외부 래치 유효) 명령 등 래치 기능을 사용할 경우에 필요합니다. 사용할 채널에 따라 좌측의 오브젝트를 설정해 주십시오.
	래치 위치 2	4013Hex-01.0 (Latch Value B)	4013Hex-02.0 (Latch Value B)	래치 기능 2에서 래치한 위치입니다. 래치 B의 값이 저장되어 있습니다. MC_TouchProbe(외부 래치 유효) 명령 등 래치 기능을 사용할 경우에 필요합니다. 사용할 채널에 따라 좌측의 오브젝트를 설정해 주십시오.
	엔코더 입력 슬레이브 상태	4030Hex-01.0 (Status Bits)	4030Hex-02.0 (Status Bits)	상태 비트가 저장되어 있습니다. MC_TouchProbe(외부 래치 유효) 명령 등 래치 기능을 사용할 경우에 필요합니다. 사용할 채널에 따라 좌측의 오브젝트를 설정해 주십시오.

사용상의 주의

설정 변경 시에는 MC 기능 모듈의 기능과 프로세스 데이터의 조합이 의도한 대로 동작하는지 확인해 주십시오.

● **디지털 입력 설정**

엔코더 축으로 사용할 경우 설정은 필요 없습니다.

엔코더 입력 터미널 측의 오브젝트 설정

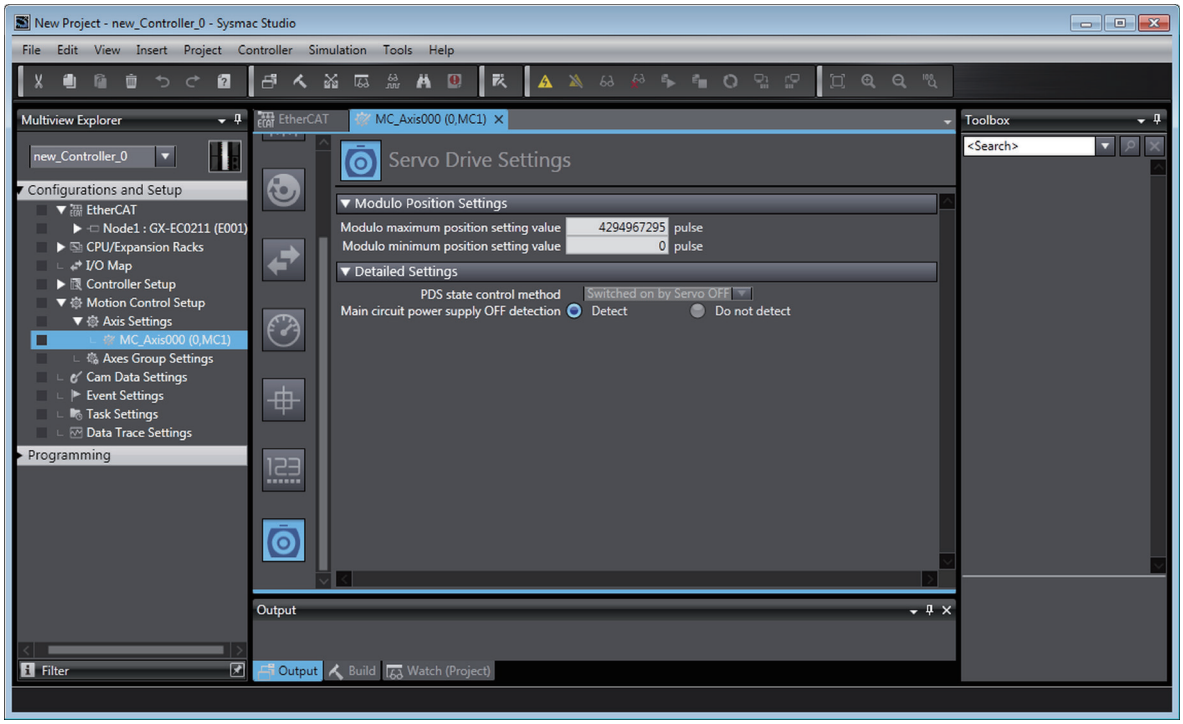
엔코더 입력 터미널에서는 설정이 필요한 오브젝트는 없습니다.

MC 기능 모듈과 엔코더 입력 터미널의 링 카운터 관계

MC 기능 모듈 축 파라메터의 서보 드라이버 설정(링 카운터 상한값, 링 카운터 하한값)과 엔코더 입력 터미널의 링 카운터 최대값 설정은 일치시켜 두어야 합니다.

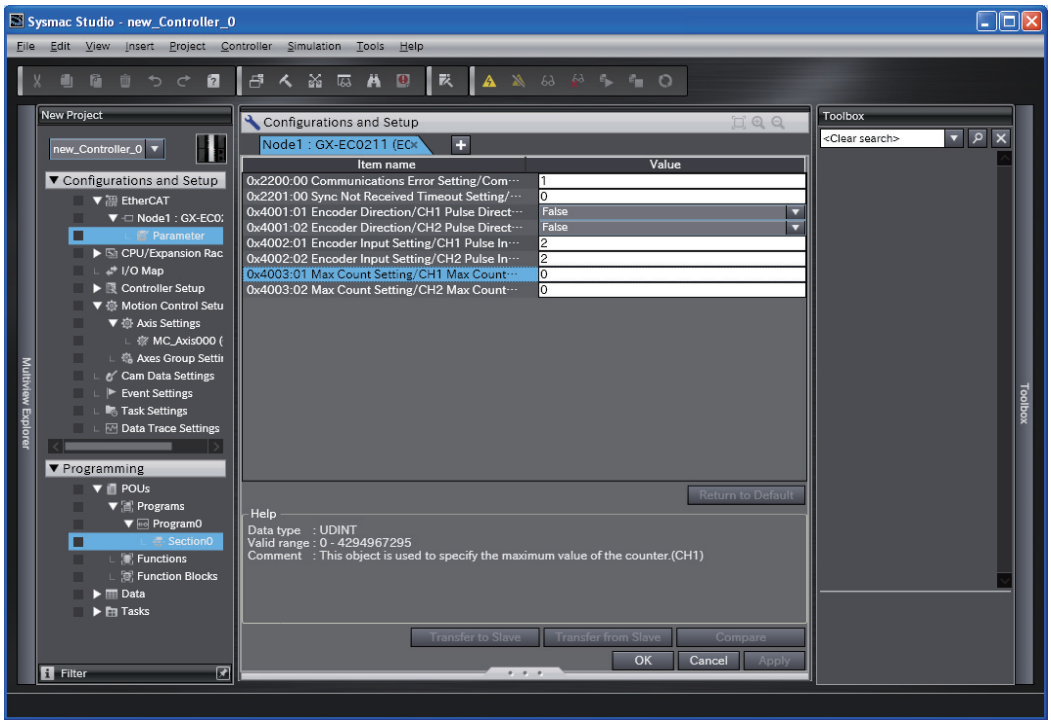
서보 드라이버 설정(링 카운터 상한값, 링 카운터 하한값)은 Sysmac Studio의 [Servo Drive Settings] 화면에서 편집합니다.
설정 내용은 아래와 같습니다.

파라메터	의미	설정값
링 카운터 상한 설정값	서보 드라이버 또는 엔코더 입력 터미널에서 설정된 링 카운터 상한값을 설정합니다.	엔코더 입력 터미널의 링 카운터 최대값 설정과 일치시켜 주십시오.
링 카운터 하한 설정값	서보 드라이버 또는 엔코더 입력 터미널에서 설정된 링 카운터 하한값을 설정합니다.	「0」을 설정해 주십시오.



엔코더 입력 터미널의 링 카운터 최대값 설정은 Sysmac Studio의 [EtherCAT] 편집 화면에서 설정합니다. 설정 내용은 아래와 같습니다.

인덱스	오브젝트명	설정값
0x4003	링 카운터 최대값 설정 (Max Count Setting)	MC 기능 모듈 축 파라메터의 서보 드라이버 설정(링 카운터 상한 설정값)과 일치시켜 주십시오.



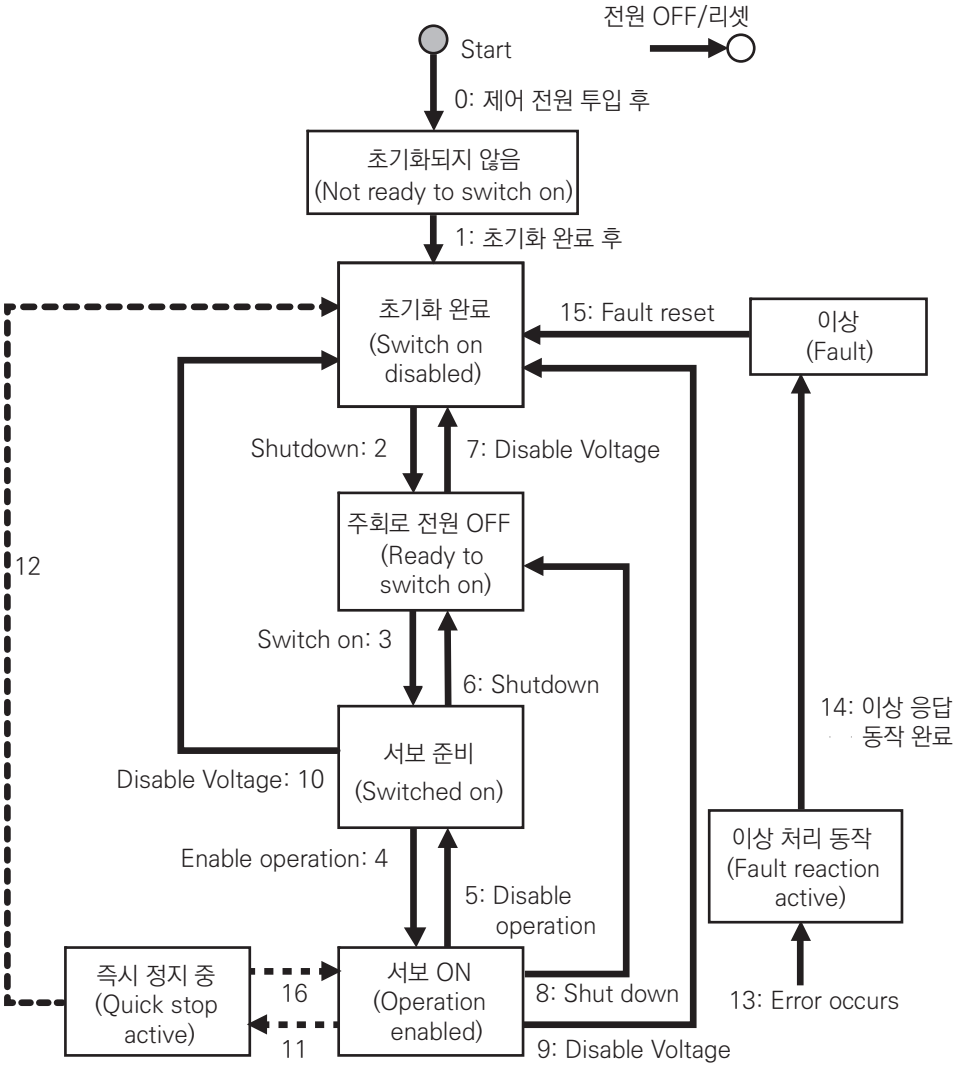
A-4 NX 유니트와의 접속

NX 시리즈 위치 인터페이스 유니트와의 접속에 대해서는 📖『NX 시리즈 위치 인터페이스 유니트 사용자 매뉴얼(SBCE-374)』을 참조해 주십시오.

A-5 PDS 상태 변이

PDS 상태 변이는 CiA402 드라이버 프로파일로 정의되어 있습니다. PDS 상태 변이 지령은 프로세스 데이터의 컨트롤 워드(6040Hex: Controlword)를 사용합니다. 실제 PDS 상태는 프로세스 데이터의 상태 워드(6041Hex: Statusword)에서 확인할 수 있습니다.

CiA402 드라이브 프로파일로 정의된 상태 플로우 차트는 다음과 같습니다.
직사각형 안은 스테이트를 나타내며 숫자는 상태 제어 명령을 나타냅니다.



A-5-1 PDS 상태 제어 방식

여기에서는 축 파라메터 [PDS State Control Method]의 설정값과 PDS 상태의 관계에 대해 설명합니다.

● [PDS State Control Method]를 「0」로 설정한 경우

[PDS State Control Method]를 「0: Switched on by Servo OFF」로 설정하면 아래와 같이 동작합니다.

- 서보 드라이버와의 통신 확립 후 MC 기능 모듈은 PDS 상태를 서보 레디(Switched on) 상태까지 자동으로 변이시킵니다.
- 서보 레디(Switched on) 상태에서 MC_Power(운전 가능) 명령의 「Enable(유효)」을 TRUE로 하면 서보 ON(Operation enabled)으로 변이합니다. 상태 플로우 차트의 「4: Enable operation」와 같이 변이됩니다.
- 서보 ON(Operation enabled) 상태에서 MC_Power(운전 가능) 명령의 「Enable(유효)」을 FALSE로 하면 서보 레디(Switched on) 상태로 변이합니다. 상태 플로우 차트의 「5: Disable operation」와 같이 변이됩니다.

● [PDS State Control Method]를 「1」로 설정한 경우

[PDS State Control Method]를 「1: Ready to switched on by Servo OFF」로 설정하면 아래와 같이 동작합니다.

- 서보 드라이버와의 통신 확립 후 MC 기능 모듈은 PDS 상태를 주회로 전원 OFF(Ready to switch on) 상태까지 자동으로 변이시킵니다.
- 주회로 전원 OFF(Ready to switch on) 상태에서 MC_Power(운전 가능) 명령의 「Enable(유효)」을 TRUE로 하면 서보 ON(Operation enabled)으로 변이합니다. 상태 플로우 차트의 「3: Switch on」과 「4: Enable operation」의 변이를 동시에 실행합니다.
- 서보 ON(Operation enabled) 상태에서 MC_Power(운전 가능) 명령의 「Enable(유효)」을 FALSE로 하면 주회로 전원 OFF(Ready to switch on) 상태로 변이합니다. 상태 플로우 차트의 「5: Disable operation」과 「6: Shutdown」의 변이를 동시에 실행합니다.

버전 관련 정보

- 유니트 버전 Ver.1.09 이하인 CPU 유니트는 [PDS State Control Method]이 「0」입니다. 유니트 버전 Ver.1.10 이상인 CPU 유니트는 [PDS State Control Method]을 선택할 수 있습니다.
- 유니트 버전 Ver.1.10 이상인 CPU 유니트의 [PDS State Control Method]의 초기값은 「0」입니다.

A-5-2 주회로 전원 OFF 검출

서보 ON 상태에서 주회로 전원 OFF를 검출할지 선택할 수 있습니다.
MC_Power(운전 가능) 명령을 실행해도 「Servo Main Circuit Power OFF」 이상이 발생하는 경우에는 「Do not detect」로 설정해 주십시오.

사용상의 주의

오므론 서보 드라이버 1S 시리즈 또는 G5 시리즈를 사용할 경우 「Do not detect」를 선택할 수 없습니다.
「Do not detect」를 선택하고 서보 ON 상태에서 서보 주회로 전원을 OFF하면 「서보 주회로 전원 OFF 검출」 이상이 발생합니다.

버전 관련 정보

- 유니트 버전 Ver.1.09 이하인 CPU 유니트는 [Main circuit power supply OFF detection]이 「Detect」입니다. 유니트 버전 Ver.1.10 이상인 CPU 유니트는 [Main circuit power supply OFF detection]을 선택할 수 있습니다.
- 유니트 버전 Ver.1.10 이상인 CPU 유니트의 [Main circuit power supply OFF detection] 초기값은 「Detect」입니다.

A-6 용어 설명

모션 제어와 관련된 용어를 설명합니다.

A-6-1 NJ/NX 시리즈 공통

용어	설명
메인 메모리 (main memory)	CPU 유니트가 OS, 사용자 프로그램 실행에 사용하는 CPU 유니트 내부의 메모리
정주기 태스크 (periodic task)	주기마다 I/O 리프레쉬와 사용자 프로그램을 실행하는 태스크
프라이머리 정주기 태스크 (primary periodic task)	가장 우선도가 높은 태스크
태스크 주기 (task period)	프라이머리 정주기 태스크 및 정주기 태스크의 실행 간격
I/O 리프레쉬 (I/O Refresh)	사전에 정해진 메모리 영역이 외부와 주기적으로 데이터를 교환하는 것.
프로그램 (program)	펑션, 펑션 블록과 함께 3종류의 POU 중 하나. 태스크에 할당되어 동작한다.
사용자 프로그램 (user program)	복수의 프로그램 집합체
인라인 ST (Inline ST)	래더도 프로그램 중에 그 일부로 ST 프로그램을 쓰거나 쓰인 ST 프로그램.
시스템 정의 변수 (system-defined variables)	모든 변수 속성이 시스템에 의해 정의되어 있어 사용자가 절대로 변경할 수 없는 변수.
글로벌 변수 (global variable)	모든 POU(프로그램, 펑션 블록, 펑션)에서 판독과 기록이 가능한 변수.
로컬 변수 (local variable)	정의된 POU 내에서만 접근할 수 있는 변수. 로컬 변수에는 내부 변수, 입력 변수, 출력 변수, 입출력 변수, 외부 변수가 포함된다.
다운로드 (download)	Sysmac Studio의「동기」기능을 통해 Sysmac Studio에서 컨트롤러로 데이터를 전송하는 것.
업로드 (upload)	Sysmac Studio의「동기」기능을 통해 컨트롤러에서 Sysmac Studio로 데이터를 전송하는 것.
전체 정지 폴트 레벨의 컨트롤러 이상 (major fault level Controller error)	NJ/NX 시리즈 컨트롤러 전체를 제어할 수 없게 되는 이상을 의미. CPU 유니트는 사용자 프로그램 실행을 바로 정지하고 리모트 I/O를 포함한 모든 슬레이브/유니트에 대해 부하를 차단한다.
부분 정지 폴트 레벨의 컨트롤러 이상 (partial fault level Controller error)	NJ/NX 시리즈 컨트롤러의 특정 기능 모듈 전체를 제어할 수 없게 되는 이상을 의미. 부분 정지 폴트 발생 후에도 NJ/NX 시리즈 CPU 유니트는 운전을 계속한다.
경도 폴트 레벨의 컨트롤러 이상 (minor fault level Controller error)	NJ/NX 시리즈 컨트롤러의 특정 기능 모듈 일부를 제어할 수 없게 되는 이상을 의미. 경도 폴트 발생 후에도 NJ/NX 시리즈 CPU 유니트는 운전을 계속한다.
감시 정보 (observation)	컨트롤러 정보 및 사용자 정보의 이벤트 중요도를 나타내는 랭크 중 하나. 제어에 영향이 없는 수준의 경미한 이상이나 이벤트 로그에 기록하여 사용자에게 통지하는 정보.

A-6-2 모션 제어 기능

용어	설명
사용 실축	축 종별이 [Servo Axis] 또는 [Encoder Axis]이며, 축 사용이 [Used Axis]로 설정된 축
사용 가상 축	축 종별이 [Virtual Servo Axis] 또는 [Virtual Encoder Axis]이며, 축 사용이 [Used Axis]로 설정된 축
모션 제어 기능 모듈 (Motion Control Function Module)	모션 제어를 실행하는 소프트웨어 구성 중 하나. 사용자 프로그램 내의 모션 제어 명령이 내린 지령에 따라 모션 제어를 실행한다. (약어: MC 기능 모듈)
모션 제어 명령 (motion control instruction)	모션 제어 기능을 실행하기 위한 펑션 블록 정의의 명령. 모션 제어 기능 모듈에서는 PLCopen®의 모션 제어용 펑션 블록에 준거한 명령과 모션 제어 기능 모듈의 독자적인 명령을 가진다.
모션 제어 축 (motion control axes)	모션 제어 기능의 모든 기능을 사용할 수 있는 축
단축 위치 제어 축 (single-axis position control axes)	모션 제어 기능 중 단축 위치 제어만 사용할 수 있는 축
단축 위치 제어 (single-axis position control)	1축에서의 위치 결정 제어
단축 속도 제어 (single-axis velocity control)	1축에서의 속도 제어. MC 기능 모듈에서는 서보 드라이버에 대해 속도 지령값을 주는 경우와 위치 지령값을 주는 경우가 있다.
단축 토크 제어 (single-axis torque control)	1축에서의 토크 제어
단축 동기 제어 (single-axis synchronized control)	하나의 주축에 대해 1축의 종축이 동기하는 제어. 기어비로 동기하는 기어 동작과 캠 테이블의 위상과 변위의 관계에 동기하는 캠 동작이 있다.
단축 수동 제어 (single-axis manual operation)	조그 전송 등 수동 조작을 위한 제어
단축 제어 보조 (auxiliary functions for single-axis control)	오버라이드값 설정 및 이상 리셋 등 1축에서의 제어를 보조하는 기능
다축 협조 제어 (multi-axes coordinated control)	직선 보간 및 원호 보간 등 복수 축이 관련되어 동작하는 제어. 축 그룹에 따라 협조할 축을 지정한다.
다축 협조 제어 보조 (auxiliary functions for multi-axes coordinated control)	축 그룹에 대한 오버라이드값 설정 및 이상 리셋 등 복수 축에서의 제어를 보조하는 기능
모션 제어 파라메터 (motion control parameters)	MC 기능 모듈의 각종 동작을 지정하기 위한 파라메터. MC 공통 파라메터, 축 파라메터, 축 그룹 파라메터의 총칭.
축 파라메터 (axis parameters)	축 단위의 파라메터
축 그룹 파라메터 (axes group parameters)	축 그룹 단위의 파라메터
모션 제어 시스템 변수 (system-defined variables for motion control)	MC 기능 모듈의 각종 상태 정보 등을 나타내는 변수. MC 공통 변수, 축 변수, 축 그룹 변수의 총칭.
MC 공통 변수 (MC common variable)	MC 기능 모듈 공통의 각종 상태 정보 등을 구조체로 정의한 시스템 정의 변수
축 변수 (axis variable)	축마다 각종 상태 정보 및 일부 축 파라메터의 설정 정보 등을 구조체로 정의한 시스템 정의 변수
축 그룹 변수 (axes group variable)	축 그룹마다 각종 상태 정보 및 일부 축 그룹 파라메터의 설정 정보 등을 구조체로 정의한 시스템 정의 변수
원점 복귀 (homing)	기계 원점을 결정하는 것. 원점 결정, 원점 서치, 캘리브레이션, Datum이라고도 한다.

용어	설명
기계 원점 (home)	장비마다 정해진 기계의 원점. 원점 복귀 동작 시에 원점 입력 신호로 결정하는 원점.
원점 (zero position)	기계 원점을 바탕으로 사용자 프로그램에서 제로 위치로 취급하는 원점. 원점 위치를 변경하지 않으면 기계 원점과 같은 위치이다.
위치 편차 (following error)	지령 현재 위치와 피드백 현재 위치의 차. 위치 제어 모드가 아닐 때는 지령 현재 위치가 없으므로 위치 편차도 존재하지 않는다. 편차 카운터값, 누적 펄스 등으로도 불린다.
편차 카운터 리셋 (following error reset)	위치 편차를 제로로 하는 것
캠 프로파일 곡선 (cam profile curve)	캠 동작을 실현하기 위한 위상과 변위의 관계를 나타내는 곡선. Sysmac Studio의 캠 에디터에서 작성한다.
캠 데이터 (cam data)	캠 동작을 실현하기 위한 위상(주축)과 변위(종축)로 구성되는 데이터
캠 데이터 변수 (cam data variable)	캠 데이터를 구조체 배열화한 변수. 위상과 변위 데이터로 구성된 구조체형 배열 지정 변수이다.
캠 테이블 (cam table)	캠 데이터의 데이터 테이블. 캠 에디터에서 작성한 캠 프로파일 곡선을 CPU 유니트에 전송하면 캠 테이블로 CPU 유니트에 내장된 불휘발성 메모리에 저장된다.
오버라이드 (override)	작업 중에 작업자가 프로그램된 값을 일시적으로 변경할 수 있는 기능
저크 (jerk)	가속도/감속도 변화의 비율. 저크를 지정하면 가감속 시의 속도 파형이 S자형이 된다. 약동 또는 약도, 가가속도라고도 한다.
멀티 모션 (Multi-motion)	프라이머리 정주기 태스크와 정주기 태스크(실행 우선도 5)를 사용하여 모션 제어를 병렬로 실행하는 것.

A-6-3 EtherCAT 통신

용어	설명
CAN application protocol over EtherCAT (CoE)	EtherCAT상에서의 CAN 어플리케이션 프로토콜 서비스
CAN in Automation (CiA)	상위층 프로토콜을 개발, 서포트하는 사용자 및 제조사로 구성된 국제단체
EtherCAT 협회(ETG) (EtherCAT Technology Group)	EtherCAT을 오픈화하여 보급 촉진을 도모하는 단체
오브젝트 (Object)	디바이스 내 특정 구성체의 추상적 표현으로 데이터, 파라메터, 방법의 집합체
오브젝트 딕셔너리 (Object Dictionary)	데이터형 오브젝트, 통신 오브젝트, 어플리케이션 오브젝트의 기술이 들어간 데이터 구조
프로세스 데이터 (Process Data)	측정 및 제어를 위해 주기적 혹은 비주기적으로 전송 지정되는 어플리케이션 오브젝트의 집합체
프로세스 데이터 오브젝트 (Process Data Object)	1개 이상의 프로세스 데이터 엔티티를 가진 파라메터를 매핑하여 기술되는 구조
서비스 데이터 오브젝트 (Service Data Object)	CoE의 비동기 메일 박스 통신으로 모든 오브젝트 딕셔너리를 판독, 기록 가능
수신 PDO (Receive PDO)	EtherCAT 슬레이브로 수신하는 프로세스 데이터 오브젝트
송신 PDO (Transmit PDO)	EtherCAT 슬레이브에서 송신하는 프로세스 데이터 오브젝트
디바이스 프로파일 (Device Profile)	디바이스급 어플리케이션용 파라메터 및 기능 기술의 집합체

A-7 버전 관련 정보

유니트 버전별의 서포트 기능에 대해 설명합니다.

CPU 유니트의 유니트 버전 종류와 Sysmac Studio 버전과의 관계는 📖『NJ/NX 시리즈 CPU 유니트 사용자 매뉴얼 소프트웨어편(SBCA-467)』을 참조해 주십시오.

모션 제어 명령의 버전 관련 정보에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

유니트 버전 Ver.1.01에서 추가된 모션 제어 기능

유니트 버전 Ver.1.01에서 추가된 기능을 사용할 때는 Sysmac Studio Ver.1.02 이상을 사용해 주십시오.

기능	개요
축 그룹 구성 축 기록	MC_ChangeAxesInGroup(축 그룹 구성 축 기록) 명령을 이용하여 축 그룹 파라메터의 [Composition Axes]를 일시적으로 갱신하는 기능.
축 그룹 위치 취득	MC_GroupReadPosition(축 그룹 위치 취득) 명령을 이용하여 축 그룹의 지령 현재 위치와 피드백 현재 위치를 취득하는 기능.
축 그룹 사이클릭 동기 절대 위치 제어	MC_GroupSyncMoveAboslute(축 그룹 사이클릭 동기 절대 위치 제어) 명령을 이용하여 각 축에 지정한 목표 위치를 주기적으로 출력하는 기능.
제어 대상 서보 드라이버	오므론 서보 모터/드라이버 G5 시리즈 EtherCAT 통신 내장 리니어 모터 타입을 추가.

유니트 버전 Ver.1.02에서 추가된 모션 제어 기능

유니트 버전 Ver.1.02에서 추가된 기능을 사용할 때는 Sysmac Studio Ver.1.03 이상을 사용해 주십시오.

유니트 버전 Ver.1.02에서 추가된 모션 제어 기능은 없으나 일부 명령 사양을 개선했습니다. 모션 제어 명령의 버전 관련 정보에 대한 상세한 내용은 📖『NJ/NX 시리즈 명령 참조 매뉴얼 모션편(SBCE-364)』을 참조해 주십시오.

유니트 버전 Ver.1.03에서 추가된 모션 제어 기능

유니트 버전 Ver.1.03에서 추가된 기능을 사용할 때는 Sysmac Studio Ver.1.04 이상을 사용해 주십시오.

기능	개요
사이클릭 동기 절대 위치 결정	MC_SyncMoveAbsolute(사이클릭 동기 절대 위치 결정) 명령을 이용하여 위치 제어 모드에서 제어 주기마다 지령 위치를 출력하는 기능.
파라메터 지정 원점 복귀	MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령을 이용하여 원점 복귀 파라메터를 지정하여 모터를 움직이고 한계 신호, 원점 근방 신호, 원점 신호를 이용하여 기계 원점을 정하는 기능.

유니트 버전 Ver.1.04에서 추가된 모션 제어 기능

유니트 버전 Ver.1.04에서 추가된 기능을 사용할 때는 Sysmac Studio Ver.1.05 이상을 사용해 주십시오.

기능	개요
축 사용 변경	MC_ChangeAxisUse(축 사용 변경) 명령을 이용하여 축 파라메터의 [Axis Use]를 일시적으로 변환하는 기능.

유니트 버전 Ver.1.05에서 추가된 모션 제어 기능

유니트 버전 Ver.1.05에서 추가된 기능을 사용할 때는 Sysmac Studio Ver.1.06 이상을 사용해 주십시오.

기능	개요
기동 속도	축 동작이 시작될 때의 첫 속도를 설정하는 기능.
입력 신호의 논리 반전	즉시 정지 입력 신호, 정방향 한계 입력 신호, 역방향 한계 입력 신호, 원점 근방 입력 신호의 논리를 반전하는 기능.

(주) 유니트 버전 Ver.1.05 이상인 CPU 유니트에서 NX 시리즈 위치 인터페이스 유니트를 축에 할당하여 모션 제어를 실행할 수 있습니다.

유니트 버전 Ver.1.06에서 추가, 변경된 모션 제어 사양

유니트 버전 Ver.1.06에서 추가, 변경된 성능 사양, 기능 사양을 사용할 때는 Sysmac Studio Ver.1.07 이상을 사용해 주십시오.

기능	개요
제어 축 최대 수	NJ301-□□□□의 제어 축 최대 수를 15축으로 확장. (사용 실축 최대수는 변경 없음)
단축 제어 최대 수	NJ301-□□□□의 단축 제어 최대 수를 15축으로 확장. (사용 실축 최대수는 변경 없음)
디지털 캠 스위치 유효	MC_DigitalCamSwitch(디지털 캠 스위치 유효) 명령을 이용하여 축 위치에 적합한 디지털 출력을 ON 또는 OFF하는 기능.
타임 스탬프→축 위치 산출	MC_TimeStampToPos(타임 스탬프→축 위치 산출) 명령을 이용하여 지정한 타임 스탬프로 축 위치를 산출하는 기능.
캠 동작 시작에 블렌딩을 추가	MC_CamIn(캠 동작 시작) 명령의 버퍼 모드 선택에 블렌딩을 추가.
_sMC_POSITION_REF형	임의 좌표계의 궤적을 3D 모션 모니터에 표시하기 위한 데이터형.

유니트 버전 Ver.1.08에서 추가, 변경된 모션 제어 기능

유니트 버전 Ver.1.08에서 추가, 변경된 기능을 사용할 때는 Sysmac Studio Ver.1.09 이상을 사용해 주십시오.

기능	개요
캠 테이블 생성	MC_GenerateCamTable(캠 테이블 생성) 명령을 이용하여 입력 파라메터로 지정한 캠 속성과 캠 노드에서 캠 테이블을 생성하는 기능.
축 파라메터의 변경	MC_WriteAxisParameter(축 파라메터 기록) 명령 및 MC_ReadAxisParameter(축 파라메터 판독) 명령을 이용하여 축 파라메터를 사용자 프로그램에서 참조 · 변경하는 기능.
디바이스 변수의 할당	축 변수에 할당한 슬레이브나 유니트의 I/O 포트에 디바이스 변수를 할당 가능. *1 상세한 내용은 「2-4-2 EtherCAT 마스터 기능과 모션 제어 기능 모듈의 관계 (P.2-20)」를 참조해 주십시오.

*1. CPU 유니트의 유니트 버전에는 의존하지 않습니다. Sysmac Studio Ver.1.09 이상을 사용하면 할당할 수 있습니다.

유니트 버전 Ver.1.09에서 추가, 변경된 모션 제어 기능

유니트 버전 Ver.1.09에서 추가된 기능을 사용할 때는 Sysmac Studio Ver.1.10 이상을 사용해 주십시오.

유니트 버전 Ver.1.09에서 추가된 모션 제어 기능은 없으나 일부 명령 사양을 개선했습니다. 모션 제어 명령의 버전 관련 정보에 대한 상세한 내용은 『NJ/NX 시리즈 명령 참조 매뉴얼(SBCE-364)』을 참조해 주십시오.

유니트 버전 Ver.1.10에서 추가, 변경된 모션 제어 기능

유니트 버전 Ver.1.10에서 추가된 아래 기능을 사용할 때는 Sysmac Studio Ver.1.12 이상을 사용해 주십시오.

기능	개요
종축 위치 보정	동기 제어 중인 종축의 위치를 보정하는 기능.
블렌딩 동작 변경	「Acceleration/Deceleration Over」의 설정이 [Use rapid acceleration/deceleration.(Blending is changed to Buffered.)], [Minor fault stop]인 경우에도 최대 가감 속도로 가감속하고 블렌딩을 계속합니다.
절대값 엔코더의 원점 확정 타이밍	기존의 원점 확정과 함께 EtherCAT 슬레이브의 프로세스 데이터 통신이 확립되지 않은 상태에서 확립 상태로 바뀐 경우에도 원점을 확정합니다.
프로세스 데이터 통신이 확립되지 않은 경우의 현재 위치	프로세스 데이터 통신이 확립되기 직전의 피드백 현재 위치가 축 변수의 피드백 현재 위치 및 지령 현재 위치입니다.
오퍼레이션 모드(6060Hex) 및 오퍼레이션 모드 표시(6061Hex)의 매핑 유무	오퍼레이션 모드(6060Hex) 및 오퍼레이션 모드 표시(6061Hex)의 매핑 유무 조합에 따라 동작이 달라집니다.
PDS 상태 제어 방식의 추가	MC_Power 명령으로 서보를 OFF할 때 PDS 상태가 바뀌는 곳을 설정합니다.
주회로 전원 OFF 검출 추가	서보 ON 지령 시 또는 서보 ON 상태에서 주회로 전원 OFF를 검출할지를 선택할 수 있습니다.

유니트 버전 Ver.1.10에서 추가된 아래 기능을 사용할 때는 Sysmac Studio Ver.1.13 이상을 사용해 주십시오.

기능	개요
멀티 모션에서의 동기 제어	멀티 모션에서 다른 태스크에 할당한 축 간 동기 제어 명령을 실행하는 기능.

유니트 버전 Ver.1.11에서 추가, 변경된 모션 제어 기능

유니트 버전 Ver.1.11에서 추가된 아래 기능을 사용할 때는 Sysmac Studio Ver.1.15 이상을 사용해 주십시오.

기능	개요
단위 변환 설정의 파라메터 추가	「감속기 사용」,「워크 측 1회전의 이동량」,「워크 측 기어비」,「모터 측 기어비」 파라메터를 추가.

유니트 버전 Ver.1.11에서 추가된 아래 기능을 사용할 때는 Sysmac Studio Ver.1.16 이상을 사용해 주십시오.

기능	개요
제어 대상 서보 드라이버	오므론 서보 드라이버 1S 시리즈 EtherCAT 통신 내장 타입을 추가.

유니트 버전 Ver.1.13에서 추가, 변경된 모션 제어 기능

유니트 버전 Ver.1.13에서 추가된 아래 기능을 사용할 때는 Sysmac Studio Ver.1.17 이상을 사용해 주십시오.

기능	개요
제어 기능	제어할 축의 기능을 선택합니다.
다운로드 중인 I/O 디바이스에 보내는 지령	다운로드를 시작할 때 I/O 디바이스에 보내는 지령을 정지할지 계속 보낼지 선택하면 그에 맞춰 MC 기능 모듈이 동작합니다.

I

색인

색인

A

Aborting(중단) 9-46, 9-64

E

EtherCAT 1-2
EtherCAT 마스터 기능 모듈 2-2

M

_MC_AX[*](축 변수) 6-21
_MC_AX[*].Act.Pos(피드백 현재 위치) 6-24
_MC_AX[*].Act.Timestamp(타임 스탬프) 6-24
_MC_AX[*].Act.Trq(피드백 현재 토크) 6-24
_MC_AX[*].Act.Vel(피드백 현재 속도) 6-24
_MC_AX[*].Cfg.AxEnable(축 사용) 6-24
_MC_AX[*].Cfg.AxNo(축 번호) 6-24
_MC_AX[*].Cfg.AxType(축 종별) 6-24
_MC_AX[*].Cfg.ExecID(실행 ID) 6-24
_MC_AX[*].Cfg.NodeAddress(노드 주소) 6-24
_MC_AX[*].Cmd.AccDec(지령 현재 가감 속도) 6-23
_MC_AX[*].Cmd.Jerk(지령 현재 저크) 6-23
_MC_AX[*].Cmd.Pos(지령 현재 위치) 6-23
_MC_AX[*].Cmd.Trq(지령 현재 토크) 6-23
_MC_AX[*].Cmd.Vel(지령 현재 속도) 6-23
_MC_AX[*].Details.Homed(원점 확정) 6-22
_MC_AX[*].Details.Idle(정지 중) 6-22
_MC_AX[*].Details.InHome(원점 정지) 6-22
_MC_AX[*].Details.InPosWaiting(인포지션 대기) 6-22
_MC_AX[*].Details.VelLimit(지령 속도 포화) 6-22
_MC_AX[*].Dir.Nega(역방향 지령 중) 6-22
_MC_AX[*].Dir.Posi(정방향 지령 중) 6-22
_MC_AX[*].DrvStatus.CSP
(사이클릭 동기 위치(CSP) 모드 중) 6-23
_MC_AX[*].DrvStatus.CST
(사이클릭 동기 토크(CST) 모드 중) 6-23
_MC_AX[*].DrvStatus.CSV
(사이클릭 동기 속도(CSV) 모드 중) 6-23
_MC_AX[*].DrvStatus.DrvAlarm
(드라이버 에러 입력) 6-23
_MC_AX[*].DrvStatus.DrvWarning
(드라이버 워닝 입력) 6-23
_MC_AX[*].DrvStatus.Home(원점 입력) 6-22
_MC_AX[*].DrvStatus.HomeSw(원점 근방 입력) 6-22
_MC_AX[*].DrvStatus.ILA
(드라이버 내부 기능 제한 중) 6-23
_MC_AX[*].DrvStatus.ImdStop(즉시 정지 입력) 6-22
_MC_AX[*].DrvStatus.Latch1(래치 입력 1) 6-23
_MC_AX[*].DrvStatus.Latch2(래치 입력 2) 6-23
_MC_AX[*].DrvStatus.MainPower(주회로 전원) 6-22
_MC_AX[*].DrvStatus.N_OT(역방향 한계 입력) 6-22
_MC_AX[*].DrvStatus.P_OT(정방향 한계 입력) 6-22
_MC_AX[*].DrvStatus.Ready(서보 준비) 6-22
_MC_AX[*].DrvStatus.ServoOn(서보ON) 6-22
_MC_AX[*].MFaultLvl.Active
(축 경도 폴트 발생 중) 6-24
_MC_AX[*].MFaultLvl.Code
(축 경도 폴트 코드) 6-24
_MC_AX[*].Obsr.Active(축 감시 정보 발생 중) 6-24
_MC_AX[*].Obsr.Code(축 감시 정보 코드) 6-24
_MC_AX[*].Scale.CountMode(카운트 모드) 6-25
_MC_AX[*].Scale.Den(모터 1회전의 이동량) 6-24
_MC_AX[*].Scale.MaxPos(현재 위치 상한값) 6-25
_MC_AX[*].Scale.MinPos(현재 위치 하한값) 6-25
_MC_AX[*].Scale.Num(모터 1회전의 펄스 수) 6-24
_MC_AX[*].Scale.Units(표시 단위) 6-25
_MC_AX[*].Status.Continuous(연속 동작 중) 6-21
_MC_AX[*].Status.Coordinated
(다축 협조 동작 중) 6-22
_MC_AX[*].Status.Disabled(축 무효) 6-21
_MC_AX[*].Status.Discrete(위치 결정 동작 중) 6-21
_MC_AX[*].Status.ErrorStop(에러 감속 정지 중) 6-22
_MC_AX[*].Status.Homing(원점 복귀 중) 6-21
_MC_AX[*].Status.Ready(축 기동 준비 완료) 6-21
_MC_AX[*].Status.Standstill(정지 중) 6-21
_MC_AX[*].Status.Stopping(감속 정지 중) 6-21
_MC_AX[*].Status.Synchronized(동기 동작 중) 6-21
_MC_COM(MC 공통 변수) 6-20
_MC_COM.MFaultLvl.Active
(MC 공통 경도 폴트 발생 중) 6-20
_MC_COM.MFaultLvl.Code
(MC 공통 경도 폴트 코드) 6-20
_MC_COM.Obsr.Active
(MC 공통 감시 정보 발생 중) 6-20
_MC_COM.Obsr.Code(MC 공통 감시 정보 코드) 6-20
_MC_COM.PFaultLvl.Active
(MC 공통 부분 정지 폴트 발생 중) 6-20
_MC_COM.PFaultLvl.Code
(MC 공통 부분 정지 폴트 코드) 6-20
_MC_COM.Status.CamTableBusy
(캠 테이블 파일 저장 실행 중/대기 중) 6-20
_MC_COM.Status.GenerateCamBusy
(캠 테이블 생성 실행 중) 6-20
_MC_COM.Status.RunMode(MC 운전 중) 6-20
_MC_COM.Status.TestMode(MC 시운전 중) 6-20
_MC_GRP[*](축 그룹 변수) 6-28
_MC_GRP[*].Cfg.ExecID(실행 ID) 6-30
_MC_GRP[*].Cfg.GrpEnable(축 그룹 사용) 6-30
_MC_GRP[*].Cfg.GrpNo(축 그룹 번호) 6-30
_MC_GRP[*].Cmd.AccDec(지령 보간 가감 속도) 6-29
_MC_GRP[*].Cmd.Vel(지령 보간 속도) 6-29
_MC_GRP[*].Details.Idle(정지 중) 6-29
_MC_GRP[*].Details.InPosWaiting
(인포지션 대기) 6-29
_MC_GRP[*].Kinematics.Axis[0]
(구성 축(축 A0)) 6-30
_MC_GRP[*].Kinematics.Axis[1]
(구성 축(축 A1)) 6-30

_MC_GRP[*].Kinematics.Axis[2]
(구성 축(축 A2)) 6-30
_MC_GRP[*].Kinematics.Axis[3]
(구성 축(축 A3)) 6-30
_MC_GRP[*].Kinematics.GrpType(축 구성) 6-30
_MC_GRP[*].MFaultLvl.Active
(축 그룹 경도 폴트 발생 중) 6-29
_MC_GRP[*].MFaultLvl.Code
(축 그룹 경도 폴트 코드) 6-29
_MC_GRP[*].Obsr.Active
(축 그룹 감시 정보 발생 중) 6-29
_MC_GRP[*].Obsr.Code
(축 그룹 감시 정보 코드) 6-29
_MC_GRP[*].Status.Disabled(축 그룹 무효) 6-28
_MC_GRP[*].Status.ErrorStop
(에러 감속 정지 중) 6-29
_MC_GRP[*].Status.Moving(동작 중) 6-28
_MC_GRP[*].Status.Ready(기동 준비 완료) 6-28
_MC_GRP[*].Status.Standby(정지 중) 6-28
_MC_GRP[*].Status.Stopping(감속 정지 중) 6-29
MC 공통 감시 정보 6-20
MC 공통 감시 정보 발생 중 6-20
MC 공통 감시 정보 코드 6-20
MC 공통 경도 폴트 6-20
MC 공통 경도 폴트 발생 중 6-20
MC 공통 경도 폴트 코드 6-20
MC 공통 변수 6-20
MC 공통 부분 정지 폴트 6-20
MC 공통 부분 정지 폴트 발생 중 6-20
MC 공통 부분 정지 폴트 코드 6-20
MC 공통 상태 6-20
MC 시운전 기능 4-2
MC 시운전 중 6-20
MC 운전 중 6-20
MC 파라메터 설정 5-2

P

PLC 기능 모듈 2-2
PLCopen® 1-2
Process Data Objects (PDO) 2-19

S

Service Data Objects (SDO) 2-19
S자 9-36

ㄱ

가감속 초과 5-18, 9-34
가상 서보 축 3-2, 5-8
가상 엔코더 축 3-2, 5-8
가속도 경고값 5-18, 9-34
가속도 단위 9-34
가속도 변경 9-42
감속 정지 중 6-21, 6-29
감속도 경고값 5-18, 9-34
감속도 단위 9-34
감속도 변경 9-42
구성 축(축 A0) 6-30
구성 축(축 A1) 6-30
구성 축(축 A2) 6-30
구성 축(축 A3) 6-30
구성 축 5-31, 5-32
구조체형 6-18
기계 원점 8-2
기동 속도 5-18, 5-20, 9-33
기동 준비 완료 6-28
기본 데이터형 6-18

ㄴ

노드 주소 6-24

ㄷ

다중 기동 9-45, 9-63
다축 협조 동작 중 6-22
단위 변환 6-24
데이터형 6-18
동기 동작 중 6-21
동작 중 6-28
드라이버 내부 기능 제한 중 6-23
드라이버 에러 리셋 감시 시간 5-21
드라이버 에러 입력 6-23
드라이버 워닝 입력 6-23

ㄹ

로터리 모드 5-24
리니어 모드 5-23
링 카운터 상한 설정값 5-23, 5-24
링 카운터 하한 설정값 5-23, 5-24

ㅁ

모션 제어 5-6, 5-7, 5-31
모션 제어 기능 모듈 2-2
모션 제어 명령
Enable형 6-11
Enable형의 명령 실행 타이밍 차트 6-13
Execute형 6-11
Execute형의 명령 실행 타이밍 차트 6-11
입력 파라메터 6-9
출력 변수「Active(제어 중)」 6-10
출력 변수「Busy(실행 중)」의 동작 6-10
출력 변수「CommandAborted(실행 중단)」의 동작 6-10
출력 변수「Done(완료)」의 동작 6-10
출력 상태 6-9
출력의 배타성 6-9
모션 제어 시스템 변수
속성 6-18
모션 제어 주기 2-23
모션 제어 파라메터 설정 5-2
모션 제어 프로그램 6-2
모터 1회전의 이동량 5-11, 6-24
모터 1회전의 펄스 수 5-11, 6-24

목표 속도 변경 9-42
목표 위치 변경 9-40
감속 초과 패턴 9-41
삼각 제어 패턴 9-41
새로운 지령값으로 반전하는 경우 9-40
새로운 지령값으로 반전하지 않는 경우 9-40
무한 길이 축 5-24
무효 캠 데이터 9-15

ㅂ

반전 시 동작 5-18
방향 지정 없음 9-37
버퍼 모드 9-45, 9-63
버퍼드(대기) 9-46, 9-65
변위 9-15
보간 가감속 초과 5-33
보간 가속도 경고값 5-33
보간 감속도 경고값 5-33
보간 속도 경고값 5-33
블렌딩(혼합) 9-47, 9-65
비어 있는 캠 데이터 9-15

ㅅ

사용자 프로그램에서 축 그룹 지정하는 방법 3-19
사용자 프로그램에서 축 지정하는 방법 3-3
사이클릭 동기 속도(CSV) 모드 중 6-23
사이클릭 동기 위치(CSP) 모드 중 6-23
사이클릭 동기 토크(CST) 모드 중 6-23
서보 ON 6-22
서보 드라이버 상태 6-22
서보 준비 6-22
서보 축 3-2, 5-8
서비스 데이터 오브젝트 2-19
소프트웨어 리미트 기능 5-22
속도 경고값 5-18, 9-33
속도의 단위 9-33
속도의 종류 9-33
시스템 정의 변수 6-17
시운전 기능 4-2
시작 위치 방식 지정 9-15
실행 ID 6-24, 6-30

ㅇ

에러 감속 정지 중 6-22, 6-29
엔코더 종별 5-23, 5-25
엔코더 축 3-2, 5-8
역방향 소프트웨어 리미트 5-22
역방향 지령 중 6-22
역방향 지정 9-37
역방향 토크 경고값 5-18
역방향 토크 제한 상한값 5-21
역방향 한계 입력 6-22
역방향 한계 입력 논리 반전 5-21
역방향 한계 입력 시 동작 5-26, 8-9
연속 동작 중 6-21
열거형 6-18
오리지널 캠 데이터 9-15
오버라이드 9-58
오브젝트 딕셔너리 2-19
외부 래치 입력 1 6-23
외부 래치 입력 2 6-23
운동학 변환 설정 6-30
원점 검출 방향 5-26, 8-9
원점 결정 8-2
원점 근방 입력 6-22
원점 근방 입력 논리 반전 5-22
원점 미확정 8-2
원점 복귀 가속도 5-26, 8-9
원점 복귀 감속도 5-26, 8-10
원점 복귀 동작 5-26, 8-7
원점 복귀 보류 시간 5-26, 8-11
원점 복귀 보정 속도 5-27, 8-11
원점 복귀 보정값 5-27, 8-11
원점 복귀 속도 5-26, 8-9
원점 복귀 시작 방향 5-26, 8-8
원점 복귀 저크 5-26, 8-10
원점 복귀 접근 속도 5-26, 8-9
원점 복귀 중 6-21
원점 위치 범위 5-19
원점 위치 오프셋 5-26, 8-10
원점 입력 6-22
원점 입력 마스크 거리 5-26, 8-10
원점 입력 신호 5-26, 8-8
원점 정지 6-22
원점 확정 6-22
위상 9-15
위상 간격 폭 9-16
위치 결정 동작 중 6-21
위치 편차 경고값 5-22
위치 편차 초과값 5-22
위치의 종류 9-31
유한 길이 축 5-23
유효 캠 데이터 9-15
유효 캠 데이터 수 9-15
이동 거리 변경 9-42
이벤트 11-2
인포지션 대기(축 그룹 제어 상태) 6-29
인포지션 대기(축 제어 상태) 6-22
인포지션 체크 시간 5-19
인포지션 폭 5-19
입력 디바이스 5-6, 5-10

ㅈ

재기동 9-40, 9-62
저크 1-2
저크의 단위 9-36
전자 기어비 5-13
절대값 엔코더
대상 서보 모터 8-16
로터리 모드에서의 사용 8-17
셋업 8-17
원점 위치 오프셋 8-15
접속 가속도 9-15
접속 속도 9-15

정방향 소프트웨어 리미트 5-22
정방향 지령 중 6-22
정방향 지정 9-37
정방향 토크 경고값 5-18
정방향 토크 제한 상한값 5-21
정방향 한계 입력 6-22
정방향 한계 입력 논리 반전 5-21
정방향 한계 입력 시 동작 5-26, 8-9
정주기 태스크 2-5, 2-6
정지
MC_GroupImmediateStop(축 그룹 즉시 정지) 명령 9-56
MC_GroupStop(축 그룹 강제 정지) 명령 9-56
MC_ImmediateStop(즉시 정지) 명령 9-7
MC_Stop(강제 정지) 명령 9-7
서보 드라이버의 입력 신호 9-6
이상 발생 등에 의한 정지 9-8, 9-56
정지 방법 9-9
지령값을 감속 정지 9-9
지령값을 즉시 정지 9-9
지령값을 즉시 정지하고 동시에 서보 OFF 9-10
지령값을 즉시 정지하고 편차 카운터 리셋 9-10
즉시 정지 입력 9-6
한계 입력 9-7
정지 우선순위 9-10
정지 중(축 그룹 상태) 6-28
정지 중(축 그룹 제어 상태) 6-29
정지 중(축 상태) 6-21
정지 중(축 제어 상태) 6-22
제어 기능 3-20, 5-9
조그 최고 속도 5-18, 9-33
종축 9-15
주축 9-15
주축 추종 거리 9-15
주회로 전원 6-22
중심점 보정 허용률 5-33
즉시 정지 입력 6-22
즉시 정지 입력 논리 반전 5-21
즉시 정지 입력 정지 방법 5-21
지령 방향 상태 6-22
지령 보간 가감 속도 6-29
지령 보간 속도 6-29
지령 속도 9-33
지령 속도 포화 6-22
지령 위치 6-15, 9-31
지령 현재 가감 속도 6-23
지령 현재 속도 6-23
지령 현재 위치 6-23
지령 현재 저크 6-23
지령 현재 토크 6-23

大

최고 보간 속도 5-33
최고 속도 5-18, 5-19, 9-33
최단거리 지정 9-37
최대 가속도 5-18, 9-34
최대 감속도 5-18, 9-34
최대 보간 가속도 5-33
최대 보간 감속도 5-33
최대 캠 데이터 수 9-15
축 감시 정보 6-24
축 감시 정보 발생 중 6-24
축 감시 정보 코드 6-24
축 경도 폴트 6-24
축 경도 폴트 발생 중 6-24
축 경도 폴트 코드 6-24
축 구성 5-31, 6-30
축 그룹 감시 정보 6-29
축 그룹 감시 정보 발생 중 6-29
축 그룹 감시 정보 코드 6-29
축 그룹 경도 폴트 6-29
축 그룹 경도 폴트 발생 중 6-29
축 그룹 경도 폴트 코드 6-29
축 그룹 구성 축 기록 9-52
축 그룹 기본 설정 6-30
축 그룹 무효 6-28
축 그룹 번호 5-31, 6-30
축 그룹 변수 3-19, 6-28
축 그룹 변수명 3-21
축 그룹 사용 5-31, 6-30
축 그룹 상태 6-28
축 그룹의 상태
감속 정지 중 6-8
동작 중 6-8
에러 감속 정지 중 6-8
정지 중 6-8
축 그룹 에러 리셋 9-53
축 그룹 위치 취득 9-53
축 그룹 유효/무효 9-52
축 그룹 정지 방법 5-33
축 그룹 제어 상태 6-29
축 그룹 지령값 6-29
축 그룹 축 구성 5-31
축 그룹 파라메터 3-19
축 기동 준비 완료 6-21
축 기본 설정 6-24
축 무효 6-21
축 번호 5-6, 5-7, 6-24
축 변수 3-2, 6-21
축 변수명 3-6
축 변수와 축 종별의 관계 6-26
축 사용 5-6, 6-24
축 상태 6-21
축의 상태
감속 정지 중 6-7
다축 협조 동작 중 6-7
동기 동작 중 6-7
에러 감속 정지 중 6-7
연속 동작 중 6-7
원점 복귀 중 6-7
위치 결정 동작 중 6-7
정지 중 6-7
축 무효 6-7
축 에러 리셋 9-3
축 제어 상태 6-22
축 종별 5-6, 5-8, 6-24
축 지령값 6-23

축 파라메터 3-2
축 현재값 6-24
출력 디바이스 5-6, 5-10

ㅋ

카운트 모드 5-23, 6-25
캠 곡선 9-15
캠 데이터 9-15
캠 데이터 변수 9-15
캠 데이터 인덱스 9-15
캠 데이터의 판독/기록 9-19
캠 동작 9-15
캠 블록 9-15
캠 블록 시작점 9-15
캠 블록 종료점 9-15
캠 시작점 9-15
캠 종료점 9-15
캠 테이블 9-16
  사용자 프로그램에서 지정하는 방법 6-34
  속성 갱신 9-20
  캠 테이블 사양 9-17
  캠 테이블 생성 9-21
  캠 테이블 저장 9-19
  캠 테이블명 6-34
  캠 테이블의 데이터형 9-17
캠 테이블 변환 9-18
캠 테이블 생성 실행 중 6-20
캠 테이블 시작 위치 9-15
캠 테이블 파일 저장 실행 중/대기 중 6-20
캠 프로파일 곡선 6-31, 9-15
  이름 6-34
코너 슈퍼임포즈 9-67, 9-69

ㅌ

타임 스탬프 6-24
태스크 2-5
태스크의 주기 2-10
토크 지령 변경 9-43
트랜지션 모드 9-67
트랜지션 무효 9-67

ㅍ

파생 데이터형 6-18
표시 단위 5-11, 5-12, 6-25
프라이머리 정주기 태스크 2-5
프라이머리 주기 2-7, 2-23, 2-24
프로그램 변경 캠 데이터 9-15
프로세스 데이터 오브젝트 2-19
프로세스 데이터 통신 주기 2-23
피드백 속도 9-33
피드백 속도 필터 시정수 5-19, 9-33
피드백 위치 6-15, 9-31
피드백 현재 속도 6-24
피드백 현재 위치 6-24
피드백 현재 토크 6-24

ㅎ

한계 입력 정지 방법 5-21
현재 방향 지정 9-37
현재 위치 상한값 6-25
현재 위치 하한값 6-25

Industrial Web ▶ http://www.ia.omron.co.kr

한국 오므론 제어기기 주식회사
서울특별시 서초구 강남대로 465, B동 18층(서초동, 교보타워)
TEL: 1522-8994 FAX: 02-3483-7788

매뉴얼 번호 SBCE-363T-K6 2018년 10월 현재

ⓒ OMRON Corporation 2018 All Rights Reserved.
예고없이 사양 등을 변경하는 경우가 있으므로 양해 부탁드립니다.