OMRON

머신 자동화 컨트롤러

NJ 시리즈

명령 참조 설명서
모션편

NJ501-15□□
NJ501-14□□
NJ501-13□□
NJ301-12□□
NJ301-11□□

SYSMAC
always in control

SBCE-364G-K6

부탁의 말

(1) 본 설명서 내용의 일부 또는 전체를 무단으로 복사, 복제, 전재하는 것을 금지합니다.

(2) 본 설명서의 내용은 개선을 위해 예고 없이 사양 등이 변경되는 경우가 있으므로 미리 양지해 주시길 바랍니다.

(3) 본 설명서 내용에 만전을 기하였으나 의문이 가는 점이나 잘못된 점을 발견하면 번거로우시겠지만 당사 지점 또는 영업소로 문의해 주십시오.
문의하실 때는 설명서 말미에 기재된 설명서 번호도 함께 알려주십시오.

상표

- Sysmac은 일본 및 기타 국가에서 오므론 주식회사의 FA 기기 제품의 상표 또는 등록 상표입니다.
- Windows, Windows XP, Windows Vista, Windows 7은 미국 및 기타 국가에서 미국 Microsoft Corporation의 등록 상표입니다.
- EtherCAT®은 독일의 Beckhoff Automation이 라이센스를 제공한 등록 상표이며, 특허 취득이 완료된 기술입니다.
- ODVA, CIP, CompoNet, DeviceNet, EtherNet/IP는 ODVA의 상표입니다.
- SD, SDHC 로고는 SD-3C, LLC의 상표입니다.

본문에 게재된 기타 시스템 이름 및 제품명은 각 회사의 상표 또는 등록 상표입니다.

머리말

NJ 시리즈 CPU 유니트를 구입해 주셔서 감사합니다.

이 설명서는 모션 제어 명령에 대해서 자세히 설명한 해설서입니다. 기능, 사용 방법 등에 대해 충분히 숙지한 후 모션 제어 기능 모듈(이후부터는 MC 기능 모듈이라고 합니다)을 올바르게 사용하십시오.
사용 시에는 NJ 시리즈 CPU 유니트 사용 설명서를 함께 참조하십시오.
또한 읽은 후에도 이 설명서를 가까운 곳에 잘 보관해 두고 사용하십시오.

대상 독자

이 설명서는 다음 사용자를 대상으로 합니다.

전기 지식(전기 공사 또는 이와 동등한 지식)이 있는
- FA 기기 도입 담당자
- FA 시스템 설계자
- FA 기기를 설치, 연결하는 기술자
- FA 현장 관리자

또한 프로그래밍 언어에 관해 국제 표준 규격 IEC61131-3 또는 일본 표준 규격 JIS B3503에 규정된 내용을 숙지하고 있는 사람을 대상으로 기술하였습니다.

대상 제품

이 설명서는 다음 제품을 대상으로 합니다.
NJ 시리즈 CPU 유니트
- NJ501-15□□
- NJ501-14□□
- NJ501-13□□
- NJ301-12□□
- NJ301-11□□

각 제품의 사양 중 일부는 다른 설명서에 기재되어 있는 경우가 있습니다. 🕮 '설명서의 구성(2페이지)' 및 🕮 '관련 설명서(24페이지)'를 확인하십시오.

설명서의 구성

본 제품에서는 'NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편', 'NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편', 'NJ 시리즈 명령 참조 설명서 기본편'의 3가지 설명서에 NJ 시리즈를 사용하기 위한 기본 정보가 기재되어 있습니다.
각종 조작에는 주로 자동화 소프트웨어 Sysmac Studio를 사용합니다. Sysmac Studio에 대해서는 📖 "Sysmac Studio Version 1 조작 설명서"(SBCA-362)를 참조하십시오.
그 밖의 설명서는 시스템 구성이나 용도에 맞춰 참조할 수 있게 구성되었습니다.
목적에 따라 해당 설명서를 읽은 후 본 제품을 이용하시기 바랍니다 .

사용 목적	설명서															
	NJ 시리즈										NX 시리즈					
	기본 정보															
	NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편	NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편	NJ 시리즈 명령 참조 설명서 기본편	NJ 시리즈 CPU 유니트 사용 설명서 모션제어편	NJ 시리즈 CPU 유니트 Ether CAT 포트 사용설명서	NJ 시리즈 CPU 유니트 명령 참조 설명서 모션편	NJ 시리즈 내장 EtherNet/IP 포트 사용설명서	NJ 시리즈 데이터 베이스 연결 CPU 유니트 사용설명서	NJ 시리즈 트러블 슈팅 설명서	NX 시리즈 커플러 유니트 사용설명서	NX 시리즈 각 NX 유니트 사용설명서	NX 시리즈 세이프티 컨트롤 유니트 사용 설명서	NX 시리즈 세이프티 컨트롤 유니트 명령 참조 설명서	NX 시리즈 데이터 참조 설명서	각 CJ 시리즈 고성능 유니트 사용설명서 NJ 시리즈 연결편
NJ 시리즈의 개요 파악	●															
장치, 설치, 하드웨어를 설정																
모션 제어를 실행하는 경우				●								●				
EtherCAT을 사용하는 경우					●											
EtherNet/IP를 사용하는 경우							●									
데이터베이스 연결 서비스를 사용하는 경우	●							●								
NX 시리즈를 사용하는 경우										●	●			●		
세이프티 컨트롤을 실행하는 경우												●		●		
CJ 시리즈 유니트를 사용하는 경우																●
소프트웨어 설정																
모션 제어를 실행하는 경우				●								●				
EtherCAT을 사용하는 경우					●											
EtherNet/IP를 사용하는 경우		●					●									
데이터베이스 연결 서비스를 사용하는 경우								●								
NX 시리즈를 사용하는 경우										●	●			●		
세이프티 컨트롤을 실행하는 경우										●		●		●		
사용자 프로그램 작성																
모션 제어를 실행하는 경우				●		●					●					
EtherCAT을 사용하는 경우					●											
데이터베이스 연결 서비스를 사용하는 경우		●	●					●								
NX 시리즈를 사용하는 경우										●	●					
세이프티 컨트롤을 실행하는 경우												●	●			
CJ 시리즈 유니트를 사용하는 경우																●
이상 처리를 포함하는 경우									●							

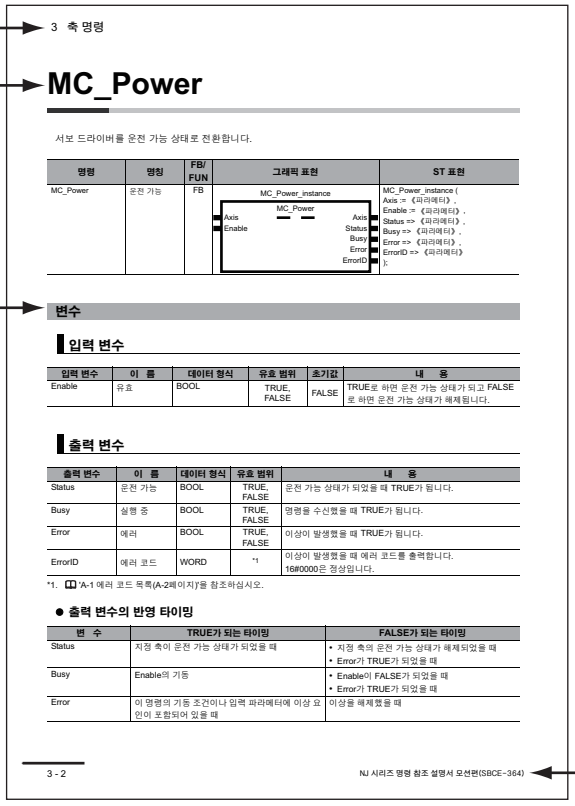
사용 목적	설명서															
	NJ 시리즈										NX 시리즈					
	기본 정보															
	NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편	NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편	NJ 시리즈 명령 참조 설명서 기본편	NJ 시리즈 CPU 유니트 사용 설명서 모션제어편	NJ 시리즈 CPU 유니트 내장 Ether CAT 포트 사용설명서	NJ 시리즈 CPU 유니트 명령 참조 설명서 모션편	NJ 시리즈 내장 EtherNet/IP 포트 사용설명서	NJ 시리즈 데이터 베이스 연결 CPU 유니트 사용설명서	NJ 시리즈 트러블 슈팅 설명서	NX 시리즈 EtherCAT 커플러 유니트 사용설명서	NX 시리즈 각 NX 유니트 사용설명서	NX 시리즈 세이프티 컨트롤 유니트 사용 설명서	NX 시리즈 세이프티 컨트롤 유니트 명령 참조 설명서	NX 시리즈 데이터 참조 설명서	각 CJ 시리즈 고성능 유니트 사용설명서 NJ 시리즈 연결편
동작 확인이나 디버그 실행		●														
모션 제어를 실행하는 경우				●								●				
EtherCAT을 사용하는 경우					●											
EtherNet/IP를 사용하는 경우							●									
데이터베이스 연결 서비스를 사용하는 경우								●								
NX 시리즈를 사용하는 경우										●	●					
세이프티 컨트롤을 실행하는 경우													●			
이상 관리 기능이나 문제 발생 시 대처 방법 파악*1	△	△		△	△		△	△	●	△	△	△				△
보수 작업 파악	●															
모션 제어를 실행하는 경우				●								●				
EtherCAT을 사용하는 경우					●											
EtherNet/IP를 사용하는 경우							●									
NX 시리즈를 사용하는 경우										●	●					
세이프티 컨트롤을 실행하는 경우													●			
CJ 시리즈 유니트를 사용하는 경우																●

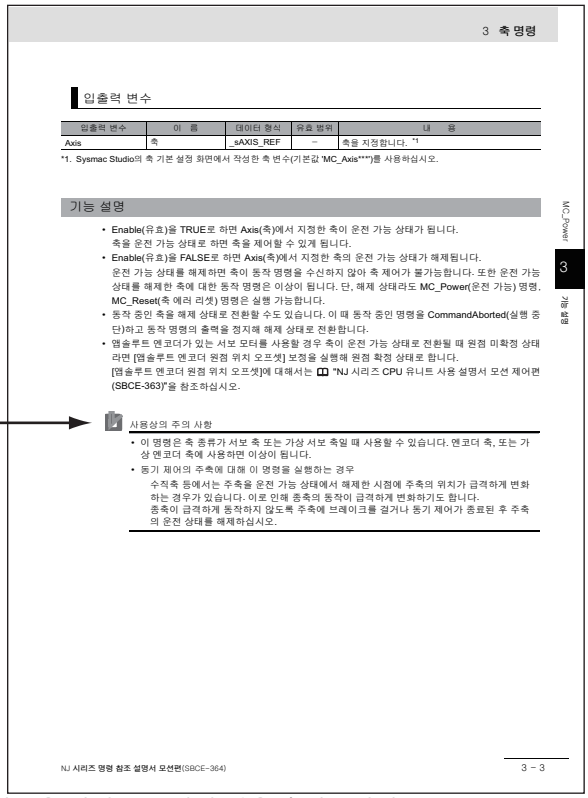
*1. NJ 시리즈 트러블 슈팅 설명서에는 이상 관리의 개념과 이상 항목의 개요가 나와 있습니다. 이상에 대한 자세한 내용은 이상의 내용에 따라 △ 표시된 설명서를 참조하십시오.

설명서 읽는 방법

페이지 구성

본 설명서의 각 페이지 구성과 기호는 다음과 같습니다.





(주의) 이 페이지는 설명용 예이므로 실제 내용과 다릅니다.

아이콘

본 설명서에서 사용하는 아이콘에는 다음과 같은 의미가 있습니다.



안전상의 중요 사항

제품을 안전하게 사용하기 위해 실시하거나 피해야 하는 사항을 나타냅니다.



사용상의 주의 사항

제품의 동작 불능, 오작동 또는 성능 및 기능에 대한 악영향을 예방하기 위해 실시하거나 피해야 하는 사항을 나타냅니다.



참고

필요에 따라 읽어야 하는 항목입니다.
알아두면 편리한 정보나 사용 시 참고가 되는 내용에 대해 설명합니다.



버전 관련 정보

CPU 유니트나 Sysmac Studio의 버전에 따른 성능이나 기능의 차이에 대해 설명합니다.



자세한 정보 또는 관련 정보가 기재되어 있는 위치를 나타냅니다.

목차 구성

1 모션 제어 명령의 개요
2 변수와 명령
3 축 명령
4 축 그룹 명령
5 공통 명령
A 부록
I 색인

목차

머리말 **1**
설명서의 구성 **2**
설명서 읽는 방법 **4**
목차 구성 **7**
사용 시 동의 사항 **14**
안전상의 주의 사항 **16**
안전상의 중요 사항 **17**
사용상의 주의 사항 **18**
법규와 규격 **19**
유니트 버전 **21**
관련 설명서 **24**
설명서 개정 이력 **26**

제1장 모션 제어 명령의 개요

1-1 모션 제어 명령 **1-2**
PLCopen 모션 제어용 펑션 블록 1-2
모션 제어 명령의 개요 1-3
동기 제어의 주축 및 보조축의 주의 사항 1-6
1-2 모션 제어 명령의 기본 지식 **1-8**
모션 제어 명령의 명칭 1-8
모션 제어 명령 언어 1-8
모션 제어 명령의 배치 1-9
모션 제어 명령의 동시 기동 1-18
모션 제어 명령의 온라인 편집 1-19
CPU 유니트의 동작 모드 전환 1-19

제2장 변수와 명령

2-1 변수 목록 **2-2**
MC 공통 변수 2-4
축 변수 2-5
축 그룹 변수 2-9
모션 제어 명령의 입력 변수 2-11
모션 제어 명령의 출력 변수 2-21
모션 제어 명령의 입출력 변수 2-23
2-2 명령 목록 **2-24**
공통 명령 2-24
축 명령 2-25
축 그룹 명령 2-26

2-3 PDO 매핑 2-27
필수 오브젝트 2-27
명령 종류에 따른 설정 오브젝트 2-28

제3장 축 명령

MC_Power 3-3
변수 3-3
기능 설명 3-4
MC_MoveJog 3-8
변수 3-8
기능 설명 3-9
MC_Home 3-15
변수 3-15
기능 설명 3-17
MC_HomeWithParameter 3-36
변수 3-36
기능 설명 3-39
MC_Move 3-41
변수 3-41
기능 설명 3-43
MC_MoveAbsolute 3-47
변수 3-47
기능 설명 3-49
샘플 프로그램 1 3-57
샘플 프로그램 2 3-64
MC_MoveRelative 3-73
변수 3-73
기능 설명 3-75
MC_MoveVelocity 3-80
변수 3-80
기능 설명 3-82
샘플 프로그램 3-87
MC_MoveZeroPosition 3-95
변수 3-95
기능 설명 3-97
MC_MoveFeed 3-102
변수 3-103
기능 설명 3-106
샘플 프로그램 3-118
MC_Stop 3-129
변수 3-129
기능 설명 3-131
MC_ImmediateStop 3-138
변수 3-138
기능 설명 3-139
MC_SetPosition 3-143
변수 3-143
기능 설명 3-145
MC_SetOverride 3-149
변수 3-149
기능 설명 3-150
MC_ResetFollowingError 3-154
변수 3-154
기능 설명 3-156
MC_CamIn 3-161
변수 3-161

기능 설명 3-164
샘플 프로그램 1 3-183
샘플 프로그램 2 3-194
MC_CamOut 3-210
변수 3-210
기능 설명 3-211
MC_GearIn 3-215
변수 3-215
기능 설명 3-217
샘플 프로그램 3-225
MC_GearInPos 3-235
변수 3-235
기능 설명 3-238
샘플 프로그램 3-246
MC_GearOut 3-257
변수 3-257
기능 설명 3-259
MC_MoveLink 3-262
변수 3-262
기능 설명 3-265
샘플 프로그램 3-273
MC_CombineAxes 3-283
변수 3-283
기능 설명 3-285
MC_Phasing 3-294
변수 3-294
기능 설명 3-296
MC_TorqueControl 3-300
변수 3-300
기능 설명 3-302
MC_SetTorqueLimit 3-311
변수 3-311
기능 설명 3-313
MC_ZoneSwitch 3-318
변수 3-318
기능 설명 3-319
MC_TouchProbe 3-324
변수 3-324
기능 설명 3-326
샘플 프로그램 3-337
MC_AbortTrigger 3-344
변수 3-344
기능 설명 3-346
MC_AxesObserve 3-348
변수 3-348
기능 설명 3-350
MC_SyncMoveVelocity 3-354
변수 3-354
기능 설명 3-356
MC_SyncMoveAbsolute 3-363
변수 3-363
기능 설명 3-365
MC_Reset 3-369
변수 3-369
기능 설명 3-370

MC_ChangeAxisUse 3-373
변수 3-373
기능 설명 3-374
MC_DigitalCamSwitch 3-377
변수 3-377
기능 설명 3-378
샘플 프로그램 3-386
MC_TimeStampToPos 3-393
변수 3-393
기능 설명 3-394
샘플 프로그램 3-398

제4장 축 그룹 명령

MC_GroupEnable 4-2
변수 4-2
기능 설명 4-3
MC_GroupDisable 4-6
변수 4-6
기능 설명 4-7
MC_MoveLinear 4-11
변수 4-11
기능 설명 4-14
샘플 프로그램 4-24
MC_MoveLinearAbsolute 4-37
변수 4-37
기능 설명 4-39
MC_MoveLinearRelative 4-40
변수 4-40
기능 설명 4-42
MC_MoveCircular2D 4-43
변수 4-43
기능 설명 4-46
샘플 프로그램 4-55
MC_GroupStop 4-67
변수 4-67
기능 설명 4-69
MC_GroupImmediateStop 4-75
변수 4-75
기능 설명 4-76
MC_GroupSetOverride 4-79
변수 4-79
기능 설명 4-80
MC_GroupReadPosition 4-83
변수 4-83
기능 설명 4-85
MC_ChangeAxesInGroup 4-87
변수 4-87
기능 설명 4-89
MC_GroupSyncMoveAbsolute 4-91
변수 4-91
기능 설명 4-93
MC_GroupReset 4-97
변수 4-97
기능 설명 4-98

제5장 공통 명령

MC_SetCamTableProperty **5-2**
변수 5-2
기능 설명 5-3
MC_SaveCamTable **5-8**
변수 5-8
기능 설명 5-10
MC_Write **5-13**
변수 5-13
기능 설명 5-16

부록

A-1 에러 코드 목록 **A-2**
A-2 에러 코드에 대한 자세한 내용 **A-20**
A-2 컨트롤러 이상에 대한 설명 보는 방법 A-20
A-2 이상 설명 A-21
A-3 다중 기동의 가능 여부 **A-78**
A-3-1 축의 상태, 축 그룹의 상태 A-79
A-3-2 상태 이동과 다중 기동의 가능 여부 A-81
A-4 버전 관련 정보 **A-86**

색인

사용 시 동의 사항

● 보증 내용

(1) 보증 기간
본 제품의 보증 기간은 구입 후 또는 지정 장소에서 납품 후 1년입니다.

(2) 보증 범위
상기 보증 기간 중에 당사의 책임으로 본 제품에 고장이 발생한 경우에는 대체품을 제공하거나 또는 제품 구입처에서 무상으로 수리해 드립니다.
단, 고장의 원인이 다음 항목에 해당하는 경우에는 이 보증 대상 범위에서 제외됩니다.
(a) 카탈로그 또는 사용 설명서 등에 기재되어 있는 이외의 조건, 환경, 취급 및 사용에 의한 경우
(b) 본 제품 이외의 원인인 경우
(c) 당사 직원 이외의 사람에 의한 개조 또는 수리에 의한 경우
(d) 본 제품의 본래 사용 방법 이외의 방법으로 사용한 경우
(e) 제품 출하 당시 당사의 과학 기술 수준으로는 예견할 수 없었던 문제인 경우
(f) 그 외 천재지변, 재해 등 당사의 책임이 아닌 원인에 의한 경우
또한 여기서의 보증은 본 제품에 대한 보증만을 의미하는 것으로, 본 제품의 고장에 의해 유발되는 손해는 보증의 대상에서 제외됩니다.

● 책임의 제한

(1) 본 제품에 기인하여 발생한 특별 손해, 간접 손해 또는 소극적 손해에 관해서는 어떤 경우라도 책임을 지지 않습니다.

(2) 프로그래밍 가능한 본 제품에 대해서는 당사 직원 이외의 사람이 실시한 프로그램 또는 그로 인해 발생한 결과에 대해 당사는 책임을 지지 않습니다.

● 적합 용도의 조건

(1) 본 제품을 다른 제품과 조합하여 사용하는 경우, 고객이 적합한 규격이나 법규 또는 규제를 확인하십시오. 또한 고객이 사용하는 시스템, 기계, 장치에 대한 본 제품의 적합성은 고객이 직접 하십시오. 이러한 내용이 이루어지지 않은 경우 당사는 본 제품의 적합성에 대해 책임을 지지 않습니다.

(2) 다음 용도로 사용되는 경우, 당사 영업 담당자에게 상담한 후 사양서 등에서 확인하는 한편, 정격 및 성능에 대해 여유가 있는 사용 방법이나 혹시 고장이 일어나도 위험을 최소화하는 안전 회로 등의 안전 대책을 마련하십시오.
(a) 옥외의 용도, 잠재적인 화학적 오염 또는 전기적인 방해를 받는 용도 또는 카탈로그 및 사용 설명서에 기재되지 않은 조건이나 환경에서의 사용
(b) 원자력 제어 설비, 소각 설비, 철도 · 항공 · 차량 설비, 의료용 기계, 오락 기계, 안전 장치 및 행정 기관이나 개별 업계의 규제에 따르는 설비
(c) 인명이나 재산에 위험이 미치는 시스템, 기계, 장치
(d) 가스, 수도, 전기 공급 시스템이나 24시간 연속 운전 시스템 등 높은 신뢰성이 요구되는 설비
(e) 그 외 상기 (a) ~ (d)에 준하는 고도의 안전성이 필요한 용도

(3) 고객이 본 제품을 인명이나 재산에 중대한 영향을 미치는 용도로 사용하는 경우에는 시스템 전체에서 위험을 알리거나 여유 있는 설계로 안전성을 확보해야 하며, 본 제품이 전체적으로 의도한 용도에 적합하게 배선 및 설치되었는지 반드시 사전에 확인하십시오.

(4) 카탈로그 등에 기재된 활용 사례는 참고용이므로 이를 적용하고자 할 때에는 기기나 장치의 기능이나 안전성을 확인한 후 사용하십시오.

(5) 본 제품이 올바르게 사용되지 않아 고객 또는 제3자에게 예측하지 못한 손해가 발생하지 않도록 사용상의 금지 사항 및 주의 사항을 모두 숙지하고 준수하십시오.

● 사양 변경

카탈로그와 사용 설명서에 기재된 제품의 사양 및 부속품은 개선 또는 그 밖의 이유로 필요에 따라 변경되는 경우가 있습니다. 당사 영업 담당자에게 상담하여 본 제품의 실제 사양을 확인하시기 바랍니다.

● 적용 범위

이상의 내용은 일본 국내에서의 거래 및 사용을 전제로 합니다.
일본 이외의 국가에서 거래 및 사용하는 경우에 대해서는 당사 영업 담당자에게 상담하시기 바랍니다.

안전상의 주의 사항

안전상의 주의 사항에 대해서는 다음 설명서를 참조하십시오.

- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"
- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"

안전상의 중요 사항

안전상의 주의 사항에 대해서는 다음 설명서를 참조하십시오.

- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"
- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"

사용상의 주의 사항

사용상의 주의 사항에 대해서는 다음 설명서를 참조하십시오.

- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편(SBCA-358)"
- 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"

법규와 규격

해외에서 사용 시

본 제품 중 외환 및 해외 무역법의 규정에 따라 수출허가, 승인 대상 화물(또는 기술)에 해당하는 것을 수출(또는 비거주자에게 제공)하는 경우, 동법에 규정된 수출 허가, 승인(또는 직무 거래 허가)이 필요합니다.

EC 지령에 대한 적합성

적합 지령

- EMC 지령
- 저전압 지령

적합의 개념

● EMC 지령

오므론의 제품은 각종 기계, 제조 장치에 조립하여 사용하는 전기 기기이므로 조립한 기계 및 장치가 좀 더 간단하게 EMC 규격에 적합한 상태가 되도록 제품 자체와 관련된 EMC 규격*을 준수하기 위해 노력하고 있습니다.

그러나 고객의 기계 및 장치가 다양하며 EMC의 성능은 EC 지령 적합 제품을 조립한 기기 및 제어반의 구성, 배선 상태, 배치 상태 등에 따라 달라지므로 고객의 사용 상태에서의 적합성은 확인할 수 없습니다. 따라서 기계 및 장치 전체의 최종적인 EMC 적합성을 고객 스스로 확인하시기 바랍니다.

* EMC(Electro-Magnetic Compatibility: 전자 환경 양립성) 관련 규격 중
EMS(Electro-Magnetic Susceptibility: 전자파 내성)에 관해서는 EN61131-2 또는 EN61000-6-2,
EMI(Electro-Magnetic Interference: 전자파 간섭)에 대해서는 EN61131-2 또는 EN61000-6-4,
또한 EN61000-6-4 Radiated emission(복사성 방출)은 10m법에 따릅니다.

● 저전압 지령

전원 전압 50V AC~1000V AC 및 75V DC~1500V DC에서 작동하는 기기에 대해 필요한 안전성을 확보해야 합니다. 적용 규격은 EN61131-2입니다.

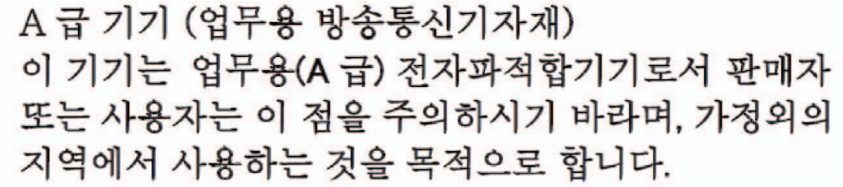
● EC 지령에 대한 적합성

NJ 시리즈는 EC 지령에 적합합니다. 그러나 고객의 기계 및 장치가 EC 지령에 적합하도록 하기 위해서는 다음 사항에 주의해야 합니다.

- NJ 시리즈는 반드시 제어반 내에 설치하십시오.
- DC 전원 유니트 및 I/O 유니트에 연결하는 DC 전원은 강화 절연 또는 이중 절연된 것을 사용하십시오.
- NJ 시리즈의 EC 지령 적합 제품은 EMI에 관한 공통 방출 규격에 적합하지만, 특히 Radiated emission(복사성 방출: 10m법)에 관해서는 사용하는 제어반의 구성, 연결되는 다른 기기와의 관계, 배선 등에 따라 달라질 수 있습니다.
  따라서 EC 지령 적합품인 NJ 시리즈를 사용하는 경우에도 고객이 기계 및 장비 전체에서 EC 지령의 적합성을 확인하여 대처해야 합니다.

KC 규격에 대한 적합성

본 제품을 한국에서 사용하는 경우는 다음 주의 사항을 준수하십시오.



ClassA 기기(업무용 방송 통신 기기)
이 기기는 업무용 전자파 발생 기기(ClassA)로, 가정 외의 장소에서 사용하기 위한 용도로 제작되었습니다. 판매자 또는 사용자는 이 점에 주의하십시오.

선급 규격에 대한 적합성

본 제품은 각종 선급 규격에 적합합니다. 각 선급 규격에 적용하는 경우 사용 조건이 설정되어 있어서 설치 장소에 따라서는 사용할 수 없는 경우도 있으므로 당사 영업 담당자에게 문의하십시오.

각 선급 규격의 사용 조건(NK, LR)

- 본 제품은 반드시 제어반에 설치하십시오.
- 제어반의 개폐구 등의 간격은 개스킷 등으로 완전히 차폐하십시오.
- 전원 라인에 다음과 같은 노이즈 필터를 연결하십시오.

노이즈 필터

제조업체	형식
Cosel(코셀)사	TAH-06-683

소프트웨어의 라이센스와 저작권

이 제품에는 타사의 소프트웨어가 조립되어 있습니다. 이 소프트웨어와 관련된 라이센스와 저작권은 http://www.ia.omron.co.kr/를 참조하십시오.

-

유니트 버전

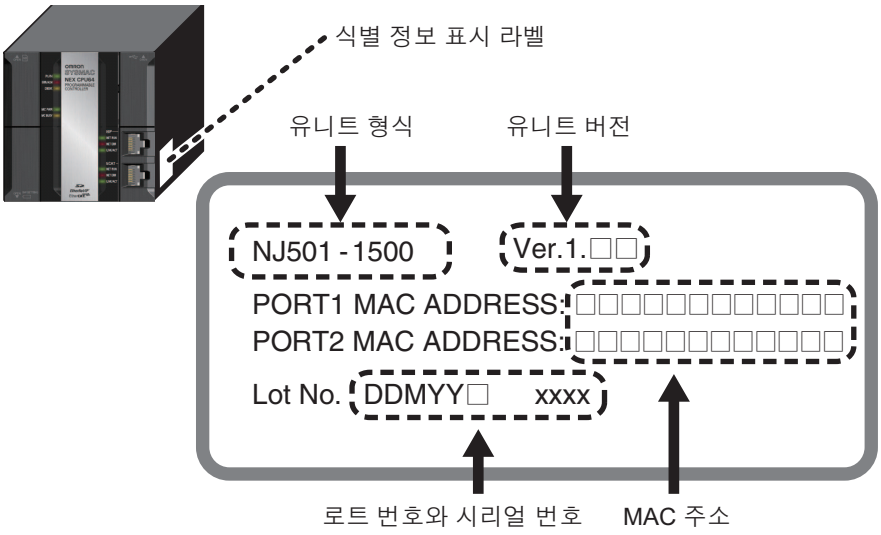
유니트 버전이란?

유니트 버전이란 NJ 시리즈 각 유니트의 개정 번호를 나타냅니다. 같은 형식의 유니트라도 유니트 버전이 다르면 탑재 기능에 몇 가지 차이점이 있습니다.

제품에서의 표기

제품 측면의 식별 정보 표시 라벨에서 '유니트 버전'을 확인할 수 있습니다.

NJ 시리즈 NJ501-□□□□의 경우



식별 정보 표시 라벨에는 다음 정보가 기재되어 있습니다.

이 름	기 능
유니트 형식	본 유니트의 형식을 표시합니다.
유니트 버전	본 유니트의 유니트 버전을 표시합니다.
로트 번호와 시리얼 번호	본 유니트의 로트 번호와 시리얼 번호를 표시합니다. DDMYY: 로트 번호, □: 오므론에서 사용, xxxx: 시리얼 번호 M은 월을 나타내며 X: 10월, Y: 11월, Z: 12월을 의미합니다.
MAC 주소	본 유니트에 내장된 포트의 MAC 주소를 표시합니다.

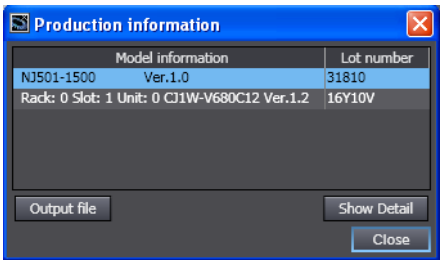
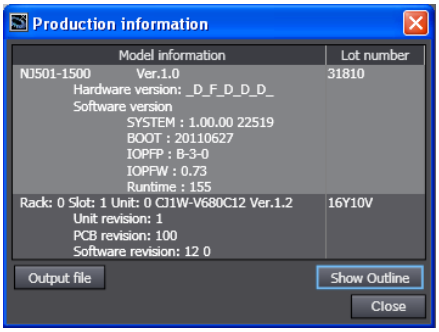
Sysmac Studio에 의한 확인 방법

Sysmac Studio에서는 [유니트 생산 정보]에서 CPU 유니트 및 CJ 시리즈의 고기능 I/O 유니트, CPU 고기능 유니트, EtherCAT 슬레이브의 유니트 버전을 확인할 수 있습니다.
CJ 시리즈의 기본 I/O 유니트의 유니트 버전은 Sysmac Studio에서는 확인할 수 없습니다.

● CPU 유니트 및 CJ 시리즈의 유니트인 경우

1 **멀티 뷰 탐색기로 [구성 · 설정] | [CPU · 증설 랙]을 두 번 클릭합니다. 또는 [구성 · 설정] | [CPU · 증설 랙]을 오른쪽 버튼으로 클릭하고 [편집]을 선택합니다.**
구성 · 설정 레이어에 유니트 편집기가 표시됩니다.

2 **유니트 편집기의 공란에 대고 오른쪽 버튼으로 클릭해 [생산 정보 표시]를 선택합니다.**
[생산 정보] 대화 상자가 표시됩니다.

간단히	자세히
	

유니트 형식의 우측에 예) 'Ver.1.00'으로 표시됩니다.

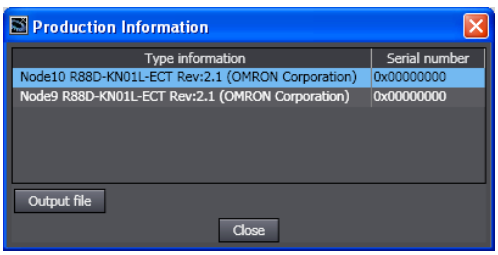
표시 내용은 다음과 같습니다.

CPU 유니트	CJ 시리즈 유니트
유니트 형식 유니트 버전 LOT 번호	유니트 형식 유니트 버전 LOT 번호 랙 번호, 슬롯 번호, 호기 번호/유니트 번호

● EtherCAT 슬레이브의 경우

1 **멀티 뷰 탐색기로 [구성 · 설정] | [EtherCAT]을 두 번 클릭합니다. 또는 [구성 · 설정] | [EtherCAT]을 오른쪽 버튼으로 클릭하고 [편집]을 선택합니다.**
구성 · 설정 레이어에 EtherCAT 구성 편집 화면이 표시됩니다.

2 **EtherCAT 구성 편집 화면에서 마스터를 오른쪽 버튼으로 클릭하고 [생산 정보 표시]를 선택합니다.**
생산 정보 대화 상자가 표시됩니다.



표시 내용은 다음과 같습니다.
노드 주소
형식 정보 *
시리얼 번호

* ESI 파일이 없는 경우와 같이 형식을 특정할 수 없는 경우는 벤더 ID, 제품 코드, 개정 번호가 표시됩니다.

참고

데이터베이스 연결 서비스 등의 기능이 추가된 CPU 유니트, 통신 커플러 유니트, NX 유니트, 세이프티 컨트롤 유니트의 유니트 버전에 대해서는 각 유니트의 설명서를 참조하십시오.

유니트 버전과 Sysmac Studio의 버전

NJ 시리즈 CPU 유니트의 버전에 따라 대응하는 기능이 다릅니다. 버전 업그레이드를 통해 추가된 기능을 사용하려면 호환되는 버전의 Sysmac Studio를 사용해야 합니다.
CPU 유니트의 유니트 버전의 종류와 Sysmac Studio 버전의 관계 및 유니트 버전에 따른 지원 기능 목록에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359)"을 참조하십시오

유니트 버전 표기

본 설명서에서는 유니트 버전에 대해서 다음과 같이 기재하고 있습니다.

제품의 명판	본 설명서에서의 기재 방법	특별 사항
로트 번호의 우측에 'Ver.1.□□' 이상	유니트 Ver.1.0 이상	본 설명서에서 유니트 버전이 특별히 명시되지 않은 경우는 모든 유니트 버전을 대상으로 설명하고 있습니다.

관련 설명서

이와 관련된 NJ 시리즈 설명서는 다음과 같습니다. 참조하시기 바랍니다.

설명서 명칭	설명서 번호	형 식	용 도	내 용
NJ 시리즈 CPU 유니트 사용 설명서 하드웨어편	SBCA-358	NJ501 -□□□□ NJ301 -□□□□	NJ 시리즈 CPU 유니트의 개요/설계/설치/보수 등의 기본적인 사양에 대해 알고 싶은 경우. 주로 하드웨어에 관한 정보	NJ 시리즈 시스템 전체 개요 및 CPU 유니트에 대해 다음 내용을 설명합니다. • 특징 및 시스템 구성 • 개요 • 각 부분의 명칭과 기능 • 일반 사양 • 설치와 배선 • 보수 점검 사용 설명서 소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편	SBCA-359	NJ501 -□□□□ NJ301 -□□□□	NJ 시리즈 CPU 유니트의 프로그래밍/시스템 기동에 대해 알고 싶은 경우. 주로 소프트웨어에 관한 정보	NJ 시리즈 CPU 유니트에 대해 다음 내용을 설명합니다 • CPU 유니트의 동작 • CPU 유니트의 기능 • 초기 설정 • IEC61131-3 기반의 언어 사양과 프로그래밍 사용 설명서 하드웨어편(SBCA-358)과 함께 사용하십시오.
NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편	SBCE-363	NJ501 -□□□□NJ301 -□□□□	모션 제어 설정이나 프로그래밍의 개념에 대해서 알고 싶은 경우	모션 제어를 위한 CPU 유니트의 설정이나 동작, 프로그래밍의 개념에 대해서 설명합니다. 프로그래밍 시에 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 명령 참조 설명서 기본편	SBCA-360	NJ501 -□□□□NJ301 -□□□□	NJ 시리즈의 기본 명령 사양에 대해 자세히 알고 싶은 경우	각 명령(IEC61131-3 사양)에 대해 자세히 설명합니다. 프로그래밍 시에 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 명령 참조 설명서 모션편	SBCE-364	NJ501 -□□□□NJ301 -□□□□	모션 명령 사양에 대해 자세히 알고 싶은 경우	각 모션 명령에 대해 자세히 설명합니다. 프로그래밍 시에 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359), 사용 설명서 모션 제어편(SBCE-363)과 함께 사용하십시오.
NJ 시리즈 CPU 유니트 내장 EtherCAT® 포트 사용 설명서	SBCD-358	NJ501 -□□□□NJ301 -□□□□	NJ 시리즈 CPU 유니트의 내장 EtherCAT 포트를 사용하는 경우	내장 EtherCAT 포트에 대해 설명합니다. 개요, 구성, 기능, 설치에 대해 기술합니다. 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 CPU 유니트 내장 EtherNet/IP™ 포트 사용 설명서	SBCD-359	NJ501 -□□□□NJ301 -□□□□	NJ 시리즈 CPU 유니트의 내장 EtherNet/IP 포트를 사용하는 경우.	내장 EtherNet/IP 포트에 대해 설명합니다. 기본 설정, 태그 데이터 링크, 기타 기능에 대해 설명합니다. 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 트러블 슈팅 설명서	SBCA-361	NJ501 -□□□□NJ301 -□□□□	NJ 시리즈에서 검출하는 이상에 대해 자세히 알고 싶은 경우.	NJ 시리즈 시스템에서 검출하는 이상 관리의 개념과 각 이상 항목에 대해 설명합니다. 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
NJ 시리즈 데이터베이스 연결 CPU 유니트 사용 설명서	SBCA-411	NJ501 -1□ 20	NJ 시리즈에서 데이터베이스 연결 서비스 기능을 사용하는 경우	데이터베이스 연결 서비스 기능에 대해 설명합니다.

설명서 명칭	설명서 번호	형 식	용 도	내 용
NX 시리즈 통신 커플러 유니트 사용 설명서	SBCD-361	NX-ECC□□□	통신 커플러 유니트 및 슬레이브 터미널의 사용 방법에 대해 알고 싶은 경우	통신 커플러 유니트와 NX 유니트로 구성된 슬레이브 터미널의 시스템 개요 및 구성 방법, 각 유니트의 하드웨어나 설정 방법, 기능에 대해 설명합니다. 다음 유니트에 대해 설명서를 준비합니다. EtherCAT 커플러 유니트
NX 시리즈 NX 유니트 사용 설명서	SBCA-407 SBCA-408 SBCA-409 SBCE-374 SBCA-410	NX-ID□□□□ NX-OC□□□□ NX-OD□□□□ NX-AD□□□□ NX-DA□□□□ NX-TS□□□□ NX-PD1□□□ NX-PF0□□□ NX-TBX□□ NX-EC0□□□ NX-ECS□□□ NX-PG0□□□ NX-□□□□□□	NX 유니트의 사용 방법에 대해 알고 싶은 경우	NX 유니트의 하드웨어나 설정 방법, 기능에 대해 설명합니다. 다음 유니트에 대해 개별적으로 설명서를 준비합니다. 디지털 I/O 유니트, 아날로그 I/O 유니트, 시스템 유니트, 위치 인터페이스 유니트
NX 시리즈 세이프티 컨트롤 유니트 사용 설명서	Z930	NX-SL□□□□ NX-SI□□□□ NX-SO□□□□	NX 시리즈 세이프티 컨트롤 유니트의 사용 방법에 대해 알고 싶은 경우	NX 시리즈 세이프티 컨트롤 유니트의 하드웨어나 설정 방법, 기능에 대해 설명합니다.
NX 시리즈 세이프티 컨트롤 유니트 명령 참조 설명서	Z931	NX-SL□□□□	세이프티 CPU 유니트용 명령의 사양을 자세히 알고 싶은 경우	세이프티 CPU 유니트용 명령에 대해 자세히 설명합니다. 프로그래밍할 때 세이프티 컨트롤 유니트 사용 설명서(SGFM-710)과 함께 사용하십시오.
CJ 시리즈 고기능 유니트 사용 설명서 (NJ 시리즈 연결편)	SBCC-846 SBCC-849 SBCC-847 SDGR-703 SBCC-848 SBCD-354 SBCD-357 SBCD-355 SBCD-353	CJ1W -□□□□	CJ 시리즈 유니트를 NJ 시리즈 CPU 유니트에서 사용하는 방법을 알고 싶은 경우	CJ 시리즈 유니트를 NJ 시리즈 CPU 유니트에서 사용하는 방법(접속 방법, 사용자 프로그램 I/F 등)과 주의점에 대해 설명합니다. 다음 유니트에 대해 개별 설명서가 마련되어 있습니다. 아날로그 I/O 유니트, 절연형 아날로그 유니트, 온도 조절 유니트, ID 센서 유니트, 고속 카운터 유니트, 시리얼 커뮤니케이션 유니트, DeviceNet 유니트, EtherNet / IP 유니트, CompoNet 마스터 유니트 사용 설명서 하드웨어편(SBCA-358)/소프트웨어편(SBCA-359)과 함께 사용하십시오.
Sysmac Studio Version 1 조작 설명서	SBCA-362	SYSMAC -SE2□□□	Sysmac Studio의 조작 방법, 기능에 대해 알고 싶은 경우.	Sysmac Studio의 조작 방법에 대해 설명합니다.
CX-Integrator CS/CJCP/NSJ/NJ 시리즈용 네트워크 컨피그레이션 툴 조작 설명서	SBCA-347		네트워크 구축(데이터 링크, 라우팅 테이블, 통신 유니트 설정 등)에 대해 알고 싶은 경우.	CX-Integrator의 조작 방법에 대해 설명합니다.
CX-Designer 사용 설명서	SBSA-532		프로그래머블 터미널 NS 시리즈의 화면 데이터를 작성하고자 하는 경우.	CX-Designer의 조작 방법에 대해 설명합니다.
CX-Protocol 조작 설명서	SBCA-307		CJ 시리즈 시리얼 커뮤니케이션 유니트와 연결된 범용 외부 기기와의 데이터 송수신 순서(프로토콜)를 작성하고자 하는 경우.	CX-Protocol의 조작 방법에 대해 설명합니다.

설명서 개정 이력

설명서 개정 기호는 앞 표지와 뒷표지 좌측 아래에 기재된 설명서 번호 끝에 알파벳으로 표기하였습니다.

예



개정 기호	개정일	개정 내용 및 개정 페이지
A	2011년 7월	초판
B	2012년 1 월	다음 축 그룹 명령의 추가 • MC_GroupReadPosition(축 그룹 위치 취득) • MC_ChangeAxesInGroup(축 그룹 구성축 쓰기) • MC_GroupSyncMoveAbsolute(축 그룹 주기적 동기 절대 위치 제어) 기재 오류 수정
C	2012년 5월	CPU 유니트 Ver.1.02로 버전 업그레이드에 따른 개정 기재 오류 수정
D	2012년 8월	CPU 유니트 Ver.1.03으로 버전 업그레이드에 따른 개정
E	2013년 2월	CPU 유니트 Ver.1.04로 버전 업그레이드에 따른 개정
F	2013년 4월	CPU 유니트 Ver.1.05로 버전 업그레이드에 따른 개정
G	2013년 6월	CPU 유니트 Ver.1.06으로 버전 업그레이드에 따른 개정

1

모션 제어 명령의 개요

NJ 시리즈의 CPU 유니트에 내장된 모션 제어 명령의 개요에 대해 설명합니다.

1-1 모션 제어 명령 1-2
PLCopen 모션 제어용 펑션 블록 1-2
모션 제어 명령의 개요 1-3
동기 제어의 주축 및 보조축의 주의 사항 1-6
1-2 모션 제어 명령의 기본 지식 1-8
모션 제어 명령의 명칭 1-8
모션 제어 명령 언어 1-8
모션 제어 명령의 배치 1-9
모션 제어 명령의 동시 기동 1-18
모션 제어 명령의 온라인 편집 1-19
CPU 유니트의 동작 모드 전환 1-19

1-1 모션 제어 명령

NJ 시리즈의 사용자 프로그램에서 모션 제어 기능을 실행하려면 펑션 블록으로 정의된 모션 제어 명령을 사용합니다.
MC 기능 모듈의 모션 제어 명령은 PLCopen의 모션 제어용 펑션 블록 기술 사양을 기반으로 합니다.
모션 제어 명령은 PLCopen에서 정의된 명령과 MC 기능 모듈의 자체 명령 두 가지가 있습니다.
여기에서는 PLCopen 모션 제어용 펑션 블록의 개요와 모션 제어 명령의 개요에 대해 설명합니다.

모션 제어 명령에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

CPU 유니트 Ver.1.05 이상과 Sysmac Studio Ver.1.06 이상의 조합에서 NX 시리즈 위치 인터페이스 유니트를 사용할 수 있습니다.
NX 시리즈 위치 인터페이스 유니트를 사용하는 경우는 "NJ 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"를 참조하십시오.

PLCopen 모션 제어용 펑션 블록

PLCopen에서는 모션 제어용 펑션 블록을 표준화해 IEC 61131-3(JIS B 3503) 규격 언어에 따라 프로그램 인터페이스를 정의하고 있습니다.
단축에서의 위치 결정이나 전자 캠 기능 및 다축에서의 보간 제어 등의 기능과 더불어, 명령을 기동하기 위한 기본적인 순서에 대해서도 정의하고 있습니다.
PLCopen 모션 제어용 펑션 블록을 이용하면 하드웨어의 의존성을 낮추고 사용자 프로그램의 재이용성을 높일 수 있습니다.
또한 트레이닝과 지원에 드는 비용을 줄일 수도 있습니다.

참고

PLCopen이란?
PLCopen은 유럽에 본부를 둔 IEC 61131-3의 보급 단체로 세계적인 회원 조직입니다.
IEC 61131-3이란 PLC 프로그래밍의 국제표준규격입니다.
PLCopen Japan은 일본 시장의 홍보 위원회로서 일본 시장에 관심 있는 회원으로 구성됩니다.

- PLCopen Japan 웹사이트: http://www.plcopen-japan.jp/
- PLCopen 유럽 본부 웹사이트: http://www.plcopen.org/

모션 제어 명령의 개요

여기에서는 PLCopen 모션 제어용 펑션 블록 기술 사양에서 정의된 항목과 MC 기능 모듈의 개요를 설명합니다.

모션 제어 명령의 종류

모션 제어 명령의 종류와 개요는 다음 표와 같습니다.

종류	종류	기능 항목	개요
공통 명령	공통 관리 계통 명령	캠 테이블	MC 기능 모듈의 공통 상태를 제어하거나 각종 데이터를 조작 또는 모니터링하는 명령입니다.
		파라메터	
축 명령	단축 동작 계통 명령	단축 위치 제어	축을 단축 동작시키는 명령입니다.
		단축 속도 제어	
		단축 토크 제어	
		단축 동기 제어	
		단축 수동 조작	
	단축 관리 계통 명령	단축 제어 보조	축 상태를 제어 및 모니터링하는 명령입니다.
축 그룹 명령	다축 동작 계통 명령	다축 협조 제어	축 그룹을 협조 동작시키는 명령입니다.
	다축 관리 계통 명령	다축 협조 제어 보조	축 그룹의 상태를 제어 및 모니터링하는 명령입니다.

상태 이동

축 상태와 축 그룹 상태 및 명령 기동에 따른 상태 이동이 정의되어 있습니다.

MC 기능 모듈의 상태와 상태 이동에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

모션 제어 명령 기동과 Status

명령을 기동하는 변수나 명령의 실행 상태를 나타내는 변수가 명령의 공통 규칙으로서 정의되어 있습니다.
명령을 기동하기 위한 입력 변수에는 Execute와 Enable의 2종류가 있습니다.
명령의 실행 상태를 나타내는 출력 변수에는 Busy, Done, CommandAborted, Error 등이 있습니다.

MC 기능 모듈의 자세한 사양에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

사용상의 주의 사항

설명서에 기재된 타이밍 차트는 Sysmac Studio의 데이터 추적에서 표시하는 타이밍과 다른 경우가 있습니다.
데이터 추적에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편 (SBCA-359)"을 참조하십시오.

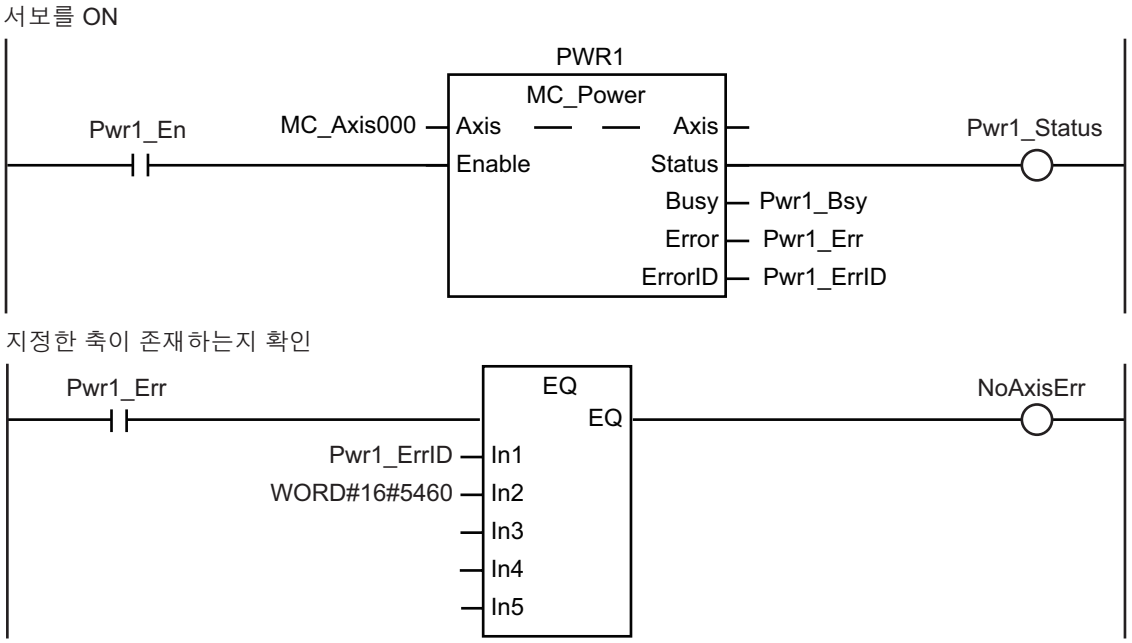
이상 처리

MC 기능 모듈에서 모션 제어는 모션 제어 명령을 사용해서 실행합니다. 모션 제어 명령 실행 시 입력 파라메터의 에러나 명령 처리에 의한 이상을 확인합니다.
명령에 이상이 있으면 명령의 출력 변수 Error(에러)가 TRUE가 되고 ErrorID(에러 코드)에 이상 코드를 출력합니다.

모션 제어 명령에서는 2종류의 방법으로 이상 처리 프로그램을 작성할 수 있습니다.

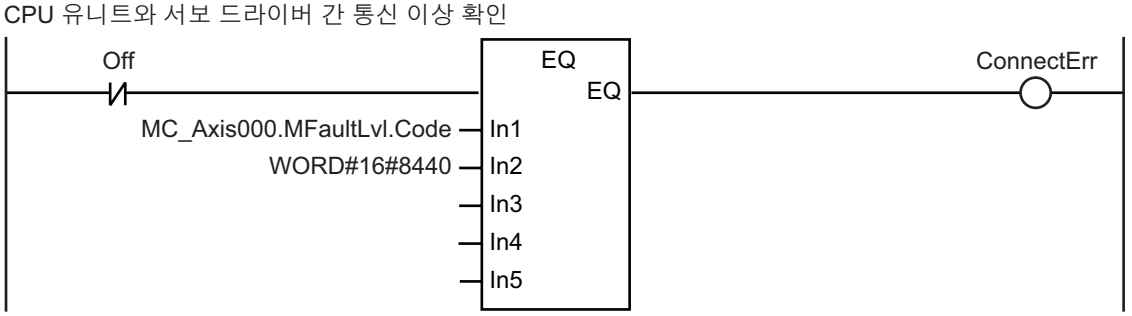
● 명령별 이상 처리

명령의 출력 변수 Error(에러) 또는 ErrorID(에러 코드)를 사용해 이상이 발생한 명령별로 이상 처리를 할 수 있습니다.
다음은 인스턴스명이 'PWR1'인 명령에서 '축 지정 오류'가 발생했음을 판별하는 예입니다. 'NoAxisErr'가 TRUE일 때 이상 처리가 실행되도록 프로그램하고 있습니다.



● 종류별 이상 처리

모션 제어 시스템 변수에 있는 이상 상태를 사용하면 이상 종류별로 이상 처리를 할 수 있습니다.
다음은 'MC_Axis000' 축에서 '슬레이브 통신 이상'이 발생했음을 판별하는 예입니다. 'ConnectErr'가 TRUE일 때 이상 처리가 실행되도록 프로그램하고 있습니다.



모션 제어 명령 실행 중의 입력 변수 변경(명령 재기동)

모션 제어 명령 실행 중에 같은 명령 인스턴스의 입력 변수값을 변경하고, 다시 Execute를 TRUE로 하면 변경한 값으로 동작합니다.

MC 기능 모듈의 명령 재기동에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

버퍼 모드 선택을 통한 명령 다중 기동

동작 중에 다른 명령 인스턴스를 실행할 수 있습니다.
버퍼 모드 선택(BufferMode)이라는 입력을 설정하면 동작을 시작할 타이밍을 지정할 수 있습니다.
버퍼 모드 선택(BufferMode)에는 다음 모드가 있습니다.

- Aborting : 중단(Aborting)
- Buffered : 대기(Buffered)
- BlendingLow : 저속으로 혼재(BlendingLow)
- BlendingPrevious : 이전 속도로 혼재(BlendingPrevious)
- BlendingNext : 다음 속도로 혼재(BlendingNext)
- BlendingHigh : 고속으로 혼재(BlendingHigh)

Aborting(중단) 모드에서는 다른 동작을 중단하고 바로 동작합니다.
그 밖의 버퍼 모드에서는 실행 중인 명령의 출력 변수인 Done이나 InVelocity가 TRUE로 설정될 때까지 대기합니다.
Buffered(대기) 모드에서는 실행 중인 명령의 동작이 정상 종료(Done = TRUE)된 후 다음 명령을 실행합니다.
Blending(혼재)에서는 2개 명령의 동작을 정지하지 않고 연속해서 실행합니다. 연속할 때의 속도는 4종류의 모드에서 선택할 수 있습니다.

MC 기능 모듈에서는 버퍼 모드 선택(BufferMode)에 의한 명령 다중 기동을 줄여서 명령 다중 기동으로 표기해서 설명하고 있습니다.

MC 기능 모듈의 명령 다중 기동에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

MC 기능 모듈에서의 명령 다중 기동의 가능 여부는 현재 축의 상태 또는 현재 축 그룹의 상태 및 실행하는 명령의 종류에 따라 결정됩니다. 이들의 관계에 대한 자세한 내용은 'A-3 다중 기동의 가능 여부 (A-78페이지)'를 참조하십시오.

모션 제어에서 사용하는 구조체

PLCopen 기술 자료에서는 모션 제어에 필요한 정보를 구조체로서 정의하고 있습니다. 데이터 형식명과 개요가 정의되어 있으며, 구조체의 내용까지는 정의되어 있지 않습니다.

PLCopen의 주된 데이터 형식과 MC 기능 모듈에서 사용하는 데이터 형식은 다음과 같습니다.

데이터 형식		정 의
PLCopen	MC 기능 모듈	
AXIS_REF	_sAXIS_REF	대응되는 축 정보를 포함한 구조체입니다.
AXES_GROUP_REF	_sGROUP_REF	대응되는 축 그룹 정보를 포함한 구조체입니다.
TRIGGER_REF	_sTRIGGER_REF	트리거 입력에 관한 정보를 포함한 구조체입니다. • 트리거 지정 • 검출 패턴 정보(정방향, 역방향, 양방향, 에지, 레벨, 패턴 인식 등)
INPUT_REF	–	입력 사양에 관한 정보를 포함한 구조체입니다. 가상 데이터인 경우도 있습니다. MC 기능 모듈에서는 사용하지 않습니다.
OUTPUT_REF	_sOUTPUT_REF	물리적 출력에 관한 구조체입니다. MC 기능 모듈에서는 CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용합니다.

MC 기능 모듈에서는 위의 표와 같이 PLCopen에서 정의된 데이터 형식의 일부와 MC 기능 모듈 자체에서 정의한 데이터 형식을 사용하고 있습니다.

MC 기능 모듈에서 사용하는 데이터 형식과 구조체의 정의에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

동기 제어의 주축 및 보조축의 주의 사항

동기 제어의 주축 및 보조축에 대해서 급격하게 속도가 변화할 경우의 주의사항과 이상이 발생하는 조건에 대해서 기재하면 다음과 같습니다.

급격하게 속도가 변화할 경우

동기 동작 시작 시 또는 동기 동작 중에 주축 및 보조축의 속도가 급격하게 변화하면 종축의 동작도 급격하게 변화해 장치에 지나친 힘이 가해질 수 있습니다.
다음 조건의 경우 주축 및 보조축의 속도가 급격하게 변화하는 경우가 있으므로 주의하십시오.

- 주축이나 보조축에 다음 4개의 명령을 실행한 경우
  MC_ImmediateStop(즉시 정지) 명령
  MC_SetPosition(현재 위치 변경) 명령
  MC_ResetFollowingError(편차 카운터 리셋) 명령
  MC_SyncMoveVelocity(주기적 동기 속도 제어) 명령
  종축이 급격하게 동작하지 않도록 위의 명령 입력 파라메터나 기동 타이밍을 적절하게 설정하거나 동기 제어를 해제한 후 위의 명령을 기동하는 것이 좋습니다.
- 주축이나 보조축에서 즉시 정지 입력 신호, 한계 정지 입력 신호가 TRUE가 되었을 경우
- 주축이나 보조축이 서보 ON 상태에서 서보 OFF 상태가 되었을 경우
  주축이나 보조축이 수직축인 경우 등에는 서보를 OFF 상태로 한 시점에서 급격하게 변화할 수 있습니다.
  종축이 급격하게 동작하지 않도록 주축 또는 보조축에 브레이크를 장착하거나 동기 제어를 종료한 후에 서보 OFF 상태로 하는 등의 대책을 세우십시오.

- 서보 드라이버의 제어 모드 전환 시
  명령 기동 시의 속도 변화에 주의하십시오.
  명령 입력 파라메터를 적절하게 설정하십시오.

이상 발생 조건

동기 동작 시작 시 또는 동기 동작 중에 주축 및 보조축에 다음 4가지 조건이 발생하면 종축에 '주축 위치 읽기 이상' 또는 '보조축 위치 읽기 이상'이 발생합니다.
동시에 동기 제어 명령의 출력 변수 'CommandAborted(실행 중단)'가 TRUE가 됩니다.

- EtherCAT의 프로세스 데이터 통신이 확립되지 않은 경우
- 'EtherCAT 통신 미확립' 상태에서 '슬레이브 통신 이상'이 발생한 경우
- '앱솔루트 엔코더 현재 위치 산출 불가' 이상이 발생한 경우
- 슬레이브가 이탈한 경우

더불어 종축에 대해 동기 제어 명령을 다중 기동한 경우는

- 다중 기동 시에 주축이나 보조축이 위의 4가지 조건 중 어느 것이라도 정상적으로 다중 기동을 접수하고 버퍼 상태가 됩니다.
- 다중 기동되고 버퍼 상태의 명령이 동작하기 시작할 때 위의 4가지 조건에 해당하지 않으면 동작을 시작합니다.



참고

주축이나 보조축에 대해 MC_Home 또는 MC_HomeWithParameter를 기동하거나 또는 앱솔루트 엔코더를 사용하는 축에서 MC_Power를 실행하면 종축은 주축이나 보조축의 위치 변화를 무시합니다.
따라서 원점 확정 시의 처리로 종축이 급격하게 동작하지는 않습니다.

1-2 모션 제어 명령의 기본 지식

여기에서는 CPU 유니트에 내장된 MC 기능 모듈의 모션 제어 명령을 사용해서 프로그램을 작성하기 위한 기본 사양과 제약 사항에 대해서 설명합니다.

모션 제어 명령에 대한 자세한 내용은 '제3장 축 명령', '제4장 축 그룹 명령', '제5장 공통 명령'을 참조하십시오.

모션 제어 명령의 명칭

MC 기능 모듈의 모든 모션 제어 명령에는 앞에 'MC_'가 붙습니다.
PLCopen에서 정의된 명령인지 MC 기능 모듈 자체의 명령인지는 '2-2 명령 목록(2-24페이지)'를 참조하십시오.

모션 제어 명령 언어

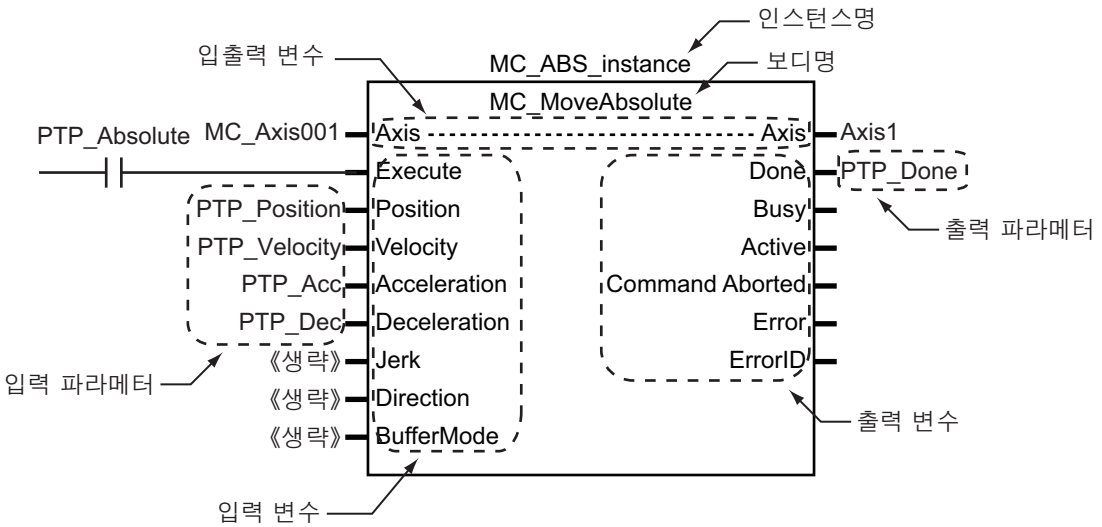
MC 기능 모듈의 모션 제어 명령은 다음 프로그래밍 언어로 사용할 수 있습니다.

- 래더도(LD)
- 스트럭쳐 텍스트(ST)

래더도(LD)

모션 제어 명령의 명령 인스턴스를 래더도에 배치합니다.
명령 인스턴스마다 인스턴스명을 붙일 수 있습니다.

다음은 MC_MoveAbsolute(절대값 위치 결정) 명령의 기술 예입니다.



- 입출력 변수 Axis에는 제어할 서보 드라이버 등의 축 변수명을 지정합니다.
- 각 입력 변수에는 목표 위치나 목표 속도 등의 동작 조건을 지정합니다.
- 각 출력 변수에는 명령 상태나 서보 드라이버 상태가 출력됩니다.
- 각 입력 파라메터를 생략하면 각 입력 변수의 초기값이 됩니다.
- 래더도 언어로 프로그램을 작성할 경우는 입력 변수의 Execute(기동) 또는 Enable(유효)과 좌측 모선 사이에 접점을 두십시오. 접점을 두지 않고 좌측 모선과 직접 연결하면 빌드 실행 시에 이상이 발생합니다.

스트럭쳐 텍스트(ST)

명령 인스턴스명을 지정합니다.
명령 변수는 좌측 위에서 좌측 아래, 우측 위에서 우측 아래의 순으로 기술합니다.
다음은 MC_MoveAbsolute(절대값 위치 결정) 명령의 기술 예입니다.

```
MC_ABS_instance (
    Axis := MC_Axis001 ,
    Execute := PTP_Absolute ,
    Position := PTP_Position ,
    Velocity := PTP_Velocity ,
    Acceleration := PTP_Acc ,
    Deceleration := PTP_Dec ,
    Jerk := PTP_Jerk ,
    Direction := _mcNoDirection ,
    BufferMode := _mcAborting ,
    Axis => MC_Axis001 ,
    Done => PTP_Done
    ) ;
```

모션 제어 명령의 배치

모션 제어 명령을 어느 태스크에 배치할 수 있는지 또한 프로그램의 어느 위치에 배치하면 어떻게 작동할지에 대해서 설명합니다.

태스크 종류

모션 제어 명령은 프라이머리 정주기 태스크 및 정주기 태스크(실행 우선도 16)에서 사용할 수 있습니다.
그 이외의 태스크 내에서 모션 제어 명령을 사용하려고 하면 빌드 시에 에러가 됩니다.

태스크 종류	사용 가능 여부
프라이머리 정주기 태스크	가능
정주기 태스크(실행 우선도 16)	가능
정주기 태스크(실행 우선도 17)	불가능
정주기 태스크(실행 우선도 18)	불가능
이벤트 태스크(실행 우선도 8)	불가능
이벤트 태스크(실행 우선도 48)	불가능

펑션 블록 정의 내

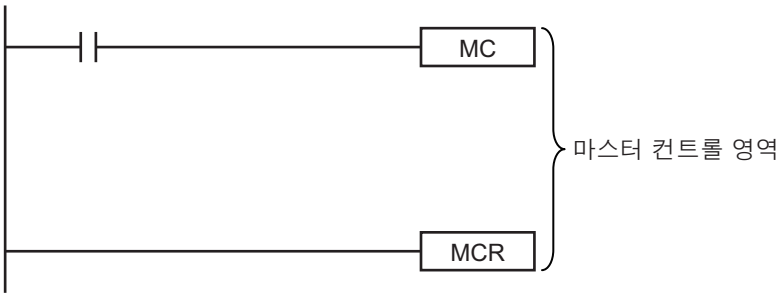
모션 제어 명령은 사용자가 작성하는 펑션 블록 정의 내에서도 사용할 수 있습니다.

참고

여러 동작이 조합된 공정을 하나의 펑션 블록으로 하면 프로그램 구조화에 따른 설계 효율이 향상되고 프로그램의 시인성이 개선됩니다.

마스터 컨트롤 영역

래더도 프로그램 중에서 MC 명령(마스터 컨트롤 시작 명령)에서부터 MCR 명령(마스터 컨트롤 종료 명령)까지의 영역이 마스터 컨트롤 영역입니다.

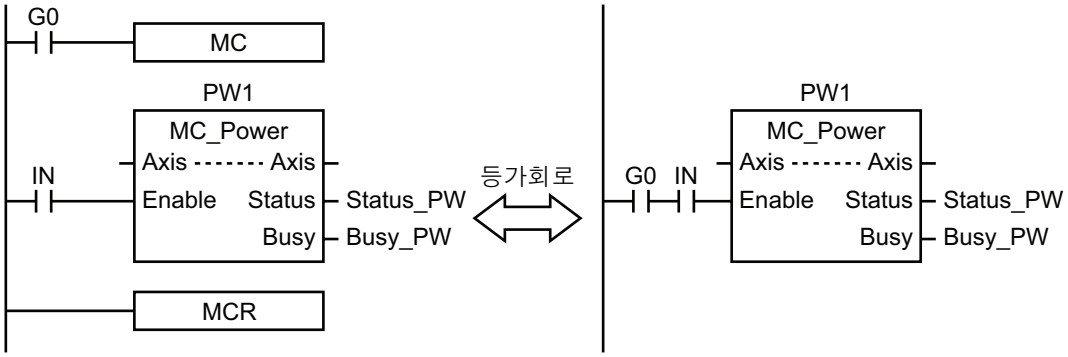


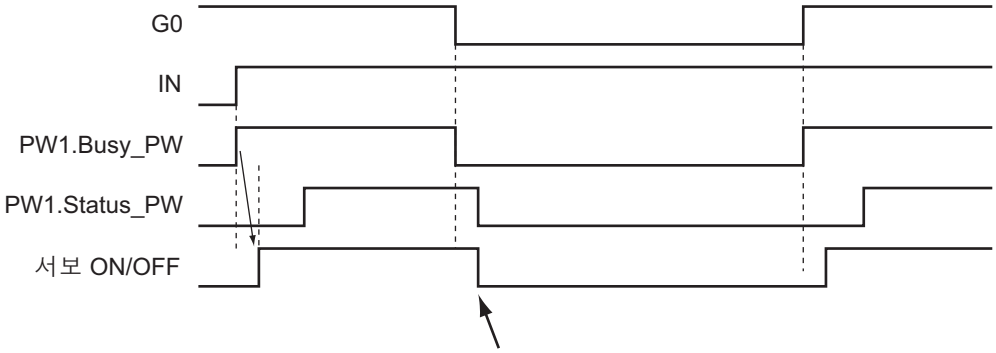
모션 제어 명령을 마스터 컨트롤 영역에 배치한 경우 MC의 입력 조건이 FALSE일 때는 다음과 같이 동작합니다.

- 좌측 모선에서 직접 모션 제어 명령의 입력 변수 Enable 또는 Execute로 입력되는 명령에 대해서는 FALSE로서 동작합니다.
- 인라인 ST 부분은 평소대로 동작합니다.
- 모션 제어 명령의 입력 변수 Enable 또는 Execute가 FALSE로 동작하는 경우라도 출력 파라메터의 값은 평소대로 갱신됩니다.

● Enable 형식 모션 제어 명령의 경우

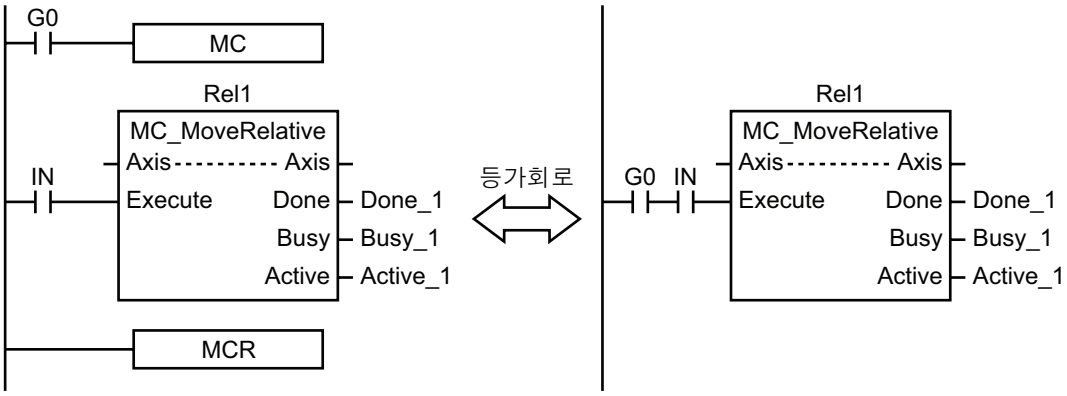
- 마스터 컨트롤 영역에 배치한 명령은 아래 그림의 우측 회로와 등가가 됩니다.
- G0가 TRUE일 때는 평소대로 동작합니다.
- G0가 FALSE일 때 Enable 입력은 FALSE와 같습니다.

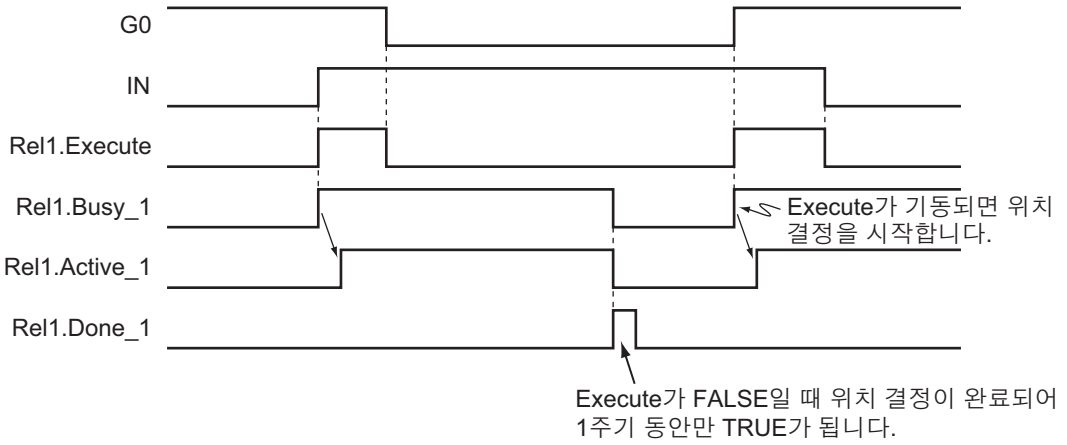




● Execute 형식 모션 제어 명령의 경우

- 마스터 컨트롤 영역에 배치한 명령은 아래 그림의 우측 회로와 등가가 됩니다.
- G0가 TRUE일 때는 평소대로 동작합니다.
- G0가 FALSE일 때 Execute 입력은 FALSE와 같습니다.
- G0가 TRUE일 때 기동한 명령은 동작 중에 GO가 FALSE가 되더라도 완료할 때까지 동작을 계속합니다. 이 때에도 출력 파라메터의 값은 평소대로 갱신됩니다.





사용상의 주의 사항

Execute 형식 모션 제어 명령은 기동하면 동작을 시작하기 때문에 마스터 컨트롤 영역 내에서 사용하는 것은 권장하지 않습니다.
사용할 경우에는 동작에 충분히 주의하십시오.

참고

- ST 프로그램에서는 MC 명령(마스터 컨트롤 시작 명령)과의 기능은 무효입니다. 모든 명령이 평소대로 동작합니다.
  MC 명령/MCR 명령에 대한 자세한 내용은 🕮 "NJ 시리즈 명령 참조 설명서 기본편 (SBCA-360)"을 참조하십시오.

ST 구문 명령 중

IF 명령, CASE 명령, WHILE 명령, REPEAT 명령 등의 ST 구문 명령 안에 모션 제어 명령을 배치한 경우에 대해서 설명합니다.

각 ST 구문 명령의 조건식 평가 결과와 일치하지 않으면 모션 제어 명령은 실행되지 않습니다. 또한 명령의 출력 변수값도 갱신되지 않습니다.
Execute 형식 명령이 실행을 시작한 후에 평가 결과가 불일치로 나오더라도 처리가 완료될 때까지는 계속해서 실행합니다. 이 경우 명령의 출력 변수값은 갱신되지 않습니다.

사용상의 주의 사항

조건식 평가 결과가 불일치로 나온 경우는 실행 중인 Execute 형식 명령의 실행 상태를 알 수 없게 되므로 ST 구문 명령 안에서 사용하는 것은 권장하지 않습니다.
사용할 경우에는 동작에 충분히 주의하십시오.

참고

- 조건식에 따라 실행할 Execute 형식 명령을 전환하려면 Execute 입력 파라메터만 ST 구문 명령 안에 배치합니다. Execute 형식 명령 자체는 ST 구문 명령 밖에 배치해 둡니다.
  ST 구문 명령에 대한 자세한 내용은 "NJ 시리즈 명령 참조 설명서 기본편(SBCA-360)"을 참조하십시오.

실수(REAL 형식, LREAL 형식) 사용 방법

실수(REAL 형식, LREAL 형식)는 부동 소수점 데이터 형식입니다.
표현 방법과 연산 처리에 대해서 설명합니다.

● REAL 형식, LREAL 형식의 데이터 크기

REAL 형식과 LREAL 형식은 데이터 크기가 다릅니다.
REAL 형식의 데이터 크기는 32비트, LREAL 형식은 64비트입니다.

● 부동 소수점 형식의 데이터 형식

부동 소수점 형식이란 실수를 부호, 지수, 가수로 표현하는 형식입니다.
어떤 실수를 아래의 식에 대입해서 나타내면 s값이 부호, e값이 지수, f값이 가수가 됩니다.

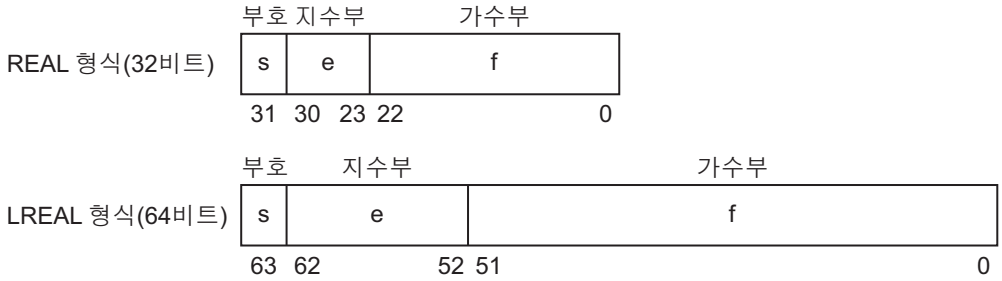
- REAL 형식

  숫자=$(-1)^{s}2^{e-127}(1+f\times 2^{-23})$

- LREAL 형식

  숫자=$(-1)^{s}2^{e-1023}(1+f\times 2^{-52})$

부동 소수점 형식의 데이터 형식은 IEEE754 규격을 준수합니다. 각각을 나타내면 다음과 같습니다.



(예) REAL 형식으로 '-86.625'를 표현할 경우

1 **부호 설정**

음수이므로 s = 1이 됩니다.

2 **2진수로 표현**

'86.625'를 2진수로 표현하면 '1010110.101'이 됩니다.

3 **2진수의 정규화**

위의 값을 정규화하면 '1.010110101×2^{6}'이 됩니다.

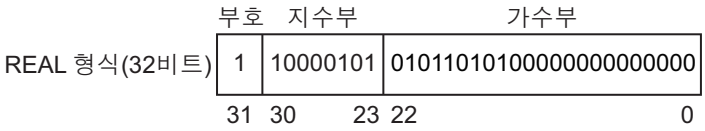
4 **지수 부분의 표현**

앞의 식에 따라 e-127 = 6이 되므로 e = 133이 됩니다.
'133'은 2진수로는 '10000101'이 되고 이것이 지수 부분을 나타냅니다.

5 **가수 부분의 표현**

'1.010110101'의 소수점 이하는 '010110101' 입니다.
이 소수점 이하의 숫자를 23비트로 표현하기 위해서는 자릿수가 부족한 만큼 0을 붙여야 합니다.
23비트로 표현한 것이 f가 됩니다.
따라서 f = 01011010100000000000000입니다.

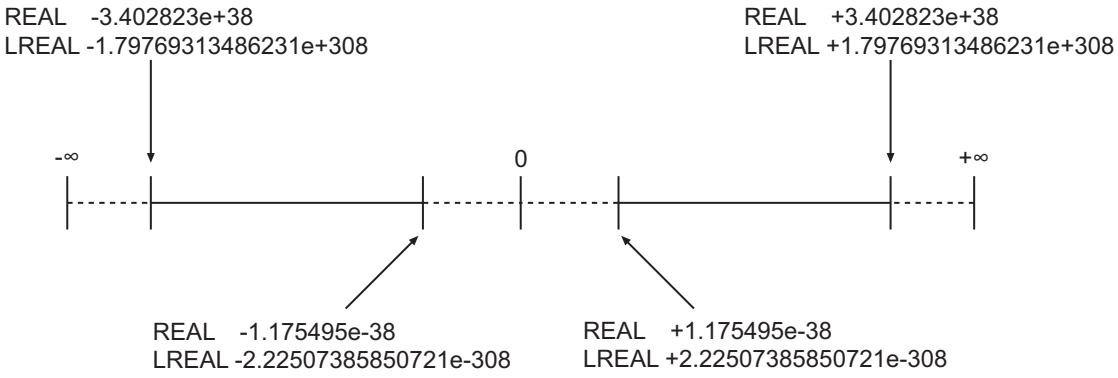
즉, '-86.625'는 다음 그림과 같이 표현할 수 있습니다.



● 유효 범위

REAL 형식, LREAL 형식의 유효 범위는 다음 표와 같습니다.

데이터 형식	-∞	음수	0	양수	+∞
REAL	-∞	-3.402823e+38~-1.175495e-38	0	+1.175495e-38~+3.402823e+38	+∞
LREAL	-∞	-1.79769313486231e+308~ -2.22507385850721e-308	0	+2.22507385850721e-308~ +1.79769313486231e+308	+∞



● 특수값

'+∞', '-∞', '+0', '-0', 비수를 특수값이라고 합니다.

비수란 부동 소수점 수로 볼 수 없는 데이터로서 숫자로 다루어지지 않습니다.

'+0'과 '-0'은 둘 다 수학적으로는 '0'과 같은 의미이지만, 연산 처리 내용에 차이가 있습니다.

자세한 내용은 이후에 기술합니다.

특수값의 부호s, 지수e, 가수f의 값은 다음 표와 같습니다.

데이터 형식	특수값	부호 s	지수 e	가수 f
REAL	+∞	0	255	0
	-∞	1	255	0
	+0	0	0	0
	-0	1	0	0
	비수	-	255	0 이외
LREAL	+∞	0	2047	0
	-∞	1	2047	0
	+0	0	0	0
	-0	1	0	0
	비수	-	2047	0 이외

● 비정규화 수

부동 소수점 형식은 '0'부근(절대값이 매우 작은 값)을 표현할 수 없습니다.
이 때문에 '0' 부근의 유효 범위를 확대하기 위해서 생긴 것이 비정규화 수입니다.
비정규화 수는 일반적인 데이터 형식으로 표현되는 값보다 절대값이 작은 값을 표현할 수 있습니다.

참고

일반적인 데이터 형식으로 표현되는 값을 정규화 수라고 합니다.

지수e=0이며 가수f≠0인 것이 비정규화 수로 간주되며, 값은 아래와 같이 표현됩니다.

- REAL 형식
  숫자=$(-1)^s 2^{-126}(f \times 2^{-23})$
- LREAL 형식
  숫자=$(-1)^s 2^{-1022}(f \times 2^{-52})$

(예) REAL 형식으로 '0.75×2^{-127}'를 표현할 경우

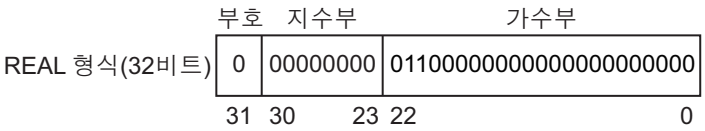
1 부호 설정
양수이므로 s = 0이 됩니다.

2 2진수로 표현
'0.75'를 2진수로 표현하면 '0.11'이 됩니다.

3 가수 부분의 계산
$(0.11)_2 \times 2^{-127} = 2^{-126}(f \times 2^{-23})$에 따라, $f = (0.11)_2 \times 2^{22}$가 됩니다.

4 가수 부분의 표현
앞의 식에 따라 f = 01100000000000000000000가 됩니다.

즉, '0.75×2^{-127}'은 다음 그림과 같이 나타낼 수 있습니다.



비정규화 수는 정규화 수보다 유효 자릿수가 적습니다. 따라서 정규화 수의 연산 결과가 비정규화 수가 되거나 또는 연산 도중 결과가 비정규화 수가 되면, 연산 결과의 유효 자릿수가 정규화 수의 유효 자릿수 미만이 되는 경우가 있습니다.

● 연산 처리

부동 소수점 형식은 근사값을 나타내는 것이므로 실제값과는 오차가 존재합니다. 또한 값의 유효 범위에도 제한이 있습니다. 따라서 연산 시에는 다음과 같이 처리합니다.

반올림

실제값이 가수의 유효 자릿수를 초과할 경우 다음 규칙에 따라 값을 반올림합니다.

- 부동 소수점 형식으로 표현할 수 있는 값 중에서 가장 실제에 가까운 값을 연산 결과로 합니다.
- 부동 소수점 형식으로 표현할 수 있는 값 중에서 가장 실제값에 가까운 것이 2개 있으면, 가수의 최하위 자릿수의 비트값이 '0'인 것을 연산 결과로 나타냅니다.
  가장 실제값에 가까운 것이 2개 있는 경우란 실제값이 2개의 근사값 중앙값인 경우에 해당합니다.

오버 플로우, 언더 플로우

실제값의 절대값을 부동 소수점 형식으로 표현할 수 있는 값보다 커지는 것을 오버 플로우라고 합니다.
반대로 부동 소수점 형식으로 표현할 수 있는 값보다 작아지는 것을 언더 플로우라고 합니다.

- 오버 플로우의 경우 실제값의 부호가 플러스(+)이면 연산 결과를 '+∞'로 나타냅니다.
  실제값의 부호가 마이너스(-)이면 연산 결과를 '-∞'로 나타냅니다.
- 언더 플로우의 경우 실제값의 부호가 플러스(+)이면 연산 결과를 '+0'으로 나타냅니다.
  실제값의 부호가 마이너스(-)이면 연산 결과를 '-0'으로 나타냅니다.

특수값의 연산

특수값의 연산은 다음 규칙에 따릅니다.
특수값에 대해서는 '특수값(1-15페이지)'을 참조하십시오.

- '+∞'과 '-∞'의 덧셈 결과는 비수가 됩니다.
- 부호가 같은 무한대끼리의 뺄셈 결과는 비수가 됩니다.
- '+0'과 무한대, '-0'과 무한대의 곱셈 결과는 비수가 됩니다.
- '+0'끼리의 나눗셈 결과, '-0'끼리의 나눗셈 결과, 무한대끼리의 나눗셈 결과는 비수가 됩니다.
- '+0'과 '-0'을 더한 결과는 '+0'이 됩니다.
- '+0'끼리 뺀 결과, '-0'끼리 뺀 결과는 '+0'이 됩니다.
- 비수를 포함한 사칙연산의 결과는 비수가 됩니다.
- CMP 명령 등의 비교 명령에서는 '+0'과 '-0'은 같은 것으로 판단합니다.
- 비교 명령에서는 비교 대상에 비수를 포함할 경우 항상 같지 않은 것으로 판단합니다.

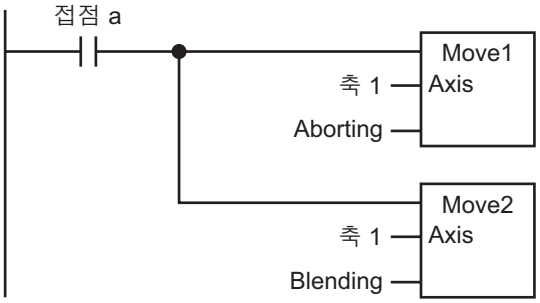
사용상의 주의 사항

MC 기능 모듈에서는 전자 기어 설정값이나 모션 제어 명령의 목표 위치 등에서 부동 소수점(LREAL) 변수를 사용합니다. 따라서 연산 결과에 반올림 오차가 포함됩니다.
예를 들면 MC_MoveRelative(상대값 위치 결정) 명령을 반복 실행하는 것과 같은 경우는 오차가 축적됩니다.
오차 축적이 문제가 될 경우는 지령 단위를 [pulse]로 하거나 MC_MoveAbsolute(절대값 위치 결정) 명령과 같은 절대 위치를 지정하십시오.

모션 제어 명령의 동시 기동

같은 축에 대해서 복수의 모션 제어 명령을 같은 태스크 주기 내에서 기동할 경우에 대해 설명합니다.

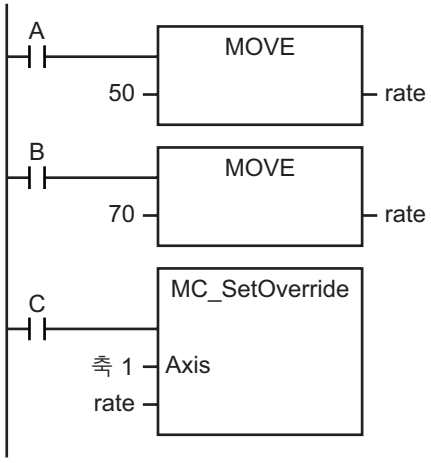
- 다음 그림의 프로그램에서는 접점a가 'ON'이 되면 명령 인스턴스 [Move 1]과 [Move 2]가 같은 태스크 주기 내에서 기동합니다.
- 프로그램에 기술된 명령은 실제로는 위에서부터 순서대로 기동하므로 [Move1]가 먼저 기동하고, [Move1]의 동작이 완료되기 전에 [Move2]가 기동합니다.
- 이것은 모션 제어 명령의 다중 기동에 해당합니다. 이 프로그램에서는 [Move 1]에 대해서 [Move 2]가 Blending으로 다중 기동합니다.



모션 제어 명령의 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

참고

MC_SetOverride(오버라이드값 설정) 명령을 위와 같이 동시 기동하면 아래쪽의 오버라이드값이 유효해집니다. MC_SetOverride(오버라이드값 설정) 명령으로 복수의 오버라이드값을 구분해서 사용하는 경우는 다음과 같이 프로그램하는 것이 좋습니다.



모션 제어 명령의 온라인 편집

Sysmac Studio의 온라인 편집 기능은 모션 제어 명령에 대해서 다음 조작이 가능합니다.

온라인 편집 조작
모션 제어 명령 삭제
모션 제어 명령 추가
모션 제어 명령의 입력, 출력, 입출력 변수 추가
모션 제어 명령의 입력, 출력, 입출력 변수 변경
모션 제어 명령의 입력, 출력, 입출력 변수 삭제

사용상의 주의 사항

축 동작 중에 MC_Stop(강제 정지) 명령이나 MC_GroupStop(축 그룹 강제 정지) 명령 등 축 동작을 정지하는 명령을 삭제하면 사용자 프로그램의 내용에 따라서는 축이 정지하지 않게 될 수 있습니다.
모션 제어 명령을 온라인 편집하려면 안전을 확인한 후 사용하십시오.

CPU 유니트의 동작 모드 전환

NJ 시리즈의 CPU 유니트 동작 모드에는 프로그램 모드와 운전 모드의 2종류가 있습니다.
동작 모드를 전환했을 때의 MC 기능 모듈의 동작에 대해서 설명합니다.

운전 모드에서 프로그램 모드로의 전환

- 실행 중인 모션 제어 명령을 중단합니다. 출력 변수 CommandAborted(실행 중단)는 FALSE로 유지되지만, 축 동작에서는 CommandAborted(실행 중단)가 TRUE일 때와 동일하게 동작합니다.
- 축이 동작 중이면 최대 감속도로 감속 정지합니다. 서보의 ON/OFF 상태는 지속됩니다.
- 캠 테이블 저장 명령으로 캠 테이블 저장을 실행 중인 경우 저장 처리는 계속됩니다.
- 실행 우선도가 16인 정주기 태스크에 배치한 모션 제어 명령은 이 태스크의 엔드 명령 실행 후에 위와 같이 처리합니다.

프로그램 모드에서 운전 모드로의 전환

- 모션 제어 명령의 출력 변수는 클리어됩니다.
- 운전 모드에서 프로그램 모드로 변경하면 축이 감속 정지합니다. 이 감속 정지 중에 운전 모드로 되돌리면 모션 제어 명령의 출력 변수가 클리어되기 때문에, 실행하던 모션 제어 명령의 CommandAborted (실행 중단)는 FALSE인 채로 유지됩니다.



참고

- 동작 모드 변경 후에도 모션 제어 명령의 출력 변수를 참조하고자 하는 경우는 출력 파라메터의 속성이 보유 속성을 갖는 변수를 할당하십시오.
  할당한 출력 파라메터를 참조함으로써 동작 모드 전환 직전의 출력 변수를 참조할 수 있게 됩니다.
- 동작 모드를 전환해도 서보의 ON/OFF 상태는 유지됩니다.

2

변수와 명령

MC 기능 모듈에서 사용하는 변수와 명령 목록표가 실려 있습니다.

2-1 변수 목록 2-2
MC 공통 변수 2-4
축 변수 2-5
축 그룹 변수 2-9
모션 제어 명령의 입력 변수 2-11
모션 제어 명령의 출력 변수 2-21
모션 제어 명령의 입출력 변수 2-23
2-2 명령 목록 2-24
공통 명령 2-24
축 명령 2-25
축 그룹 명령 2-26
2-3 PDO 매핑 2-27
필수 오브젝트 2-27
명령 종류에 따른 설정 오브젝트 2-28

2-1 변수 목록

MC 기능 모듈에서 사용하는 변수는 2가지로 분류됩니다.
하나는 축 상태나 일부 파라메터 설정 내용을 모니터링하기 위한 시스템 정의 변수입니다. MC 기능 모듈에서 사용하는 시스템 정의 변수를 모션 제어 시스템 변수라고 합니다.
다른 하나는 모션 제어 명령이 명령 인수를 입력받거나 명령 실행 상태를 출력하기 위한 변수입니다. 모션 제어 명령의 입력 변수 중 몇 가지는 선택 항목에서 선택하는 열거 형식 변수입니다.
이 절에서는 각종 변수 목록, 모션 제어 명령 입력 변수의 유효 범위 및 열거 형식의 목록을 기재합니다.

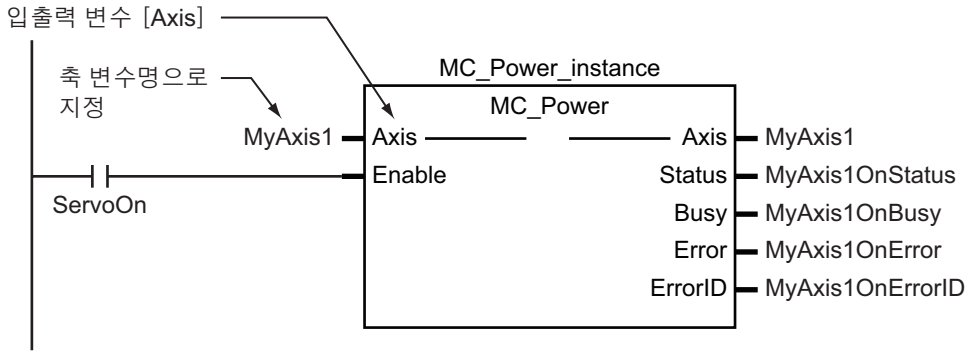
● 모션 제어 시스템 변수

제1계층	제2계층	제3계층	내 용
시스템 정의 변수	모션제어시스템 변수	MC 공통 변수	MC 기능 모듈의 공통 상태를 모니터링할 수 있습니다.
		축 변수	각 축 상태와 일부 축 파라메터의 설정 내용을 모니터링할 수 있습니다.
		축 그룹 변수	각 축 그룹 상태와 일부 축 그룹 파라메터의 설정 내용을 모니터링할 수 있습니다.

모션 제어 시스템 변수에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

참고

축 변수와 축 그룹 변수는 시스템 정의 변수입니다.
이들을 사용자 프로그램에서 사용할 때는 시스템 정의 변수명(_MC_AX[**]나 _MC_GRP[**]] 을 사용합니다. 사용자 프로그램에서 사용할 때는 그 이외에 Sysmac Studio에서 작성한 변수명도 사용할 수 있습니다.
Sysmac Studio에서 작성한 축이나 축 그룹의 변수명은 각각 임의의 이름으로 변경할 수 있습니다.
다음 그림은 시스템 정의 변수의 축 변수명이 '_MC_AX[0]'인 축에 대해 Sysmac Studio에서 추가했을 때의 축 변수명을 'MyAxis1'로 변경한 경우의 예입니다.



● 모션 제어 명령 변수

종 류	개 요
입력 변수	명령 인수
출력 변수	명령 실행 상태 모니터링
입출력 변수	명령 실행 대상 지정



참고

- '_e'로 시작하는 데이터 형식은 열거 형식임을 나타냅니다.
- '_s'로 시작하는 데이터 형식은 구조체 형식임을 나타냅니다.

MC 기능 모듈에서 사용하는 데이터 형식에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

MC 공통 변수

MC 공통 변수는 변수명이 _MC_COM이며 데이터 형식이 _sCOMMON_REF인 구조체 형식의 변수입니다. 여기에서는 MC 공통 변수의 구성과 멤버에 대해 자세히 설명합니다.

변수명	데이터 형식	이 름	기 능
_MC_COM	_sCOMMON_REF	MC 공통 변수	
Status	_sCOMMON_REF_STA	MC 공통 Status	
RunMode	BOOL	MC 운전 중	MC 기능 모듈이 운전 중일 때 TRUE가 됩니다.
TestMode	BOOL	MC 시운전 중	Sysmac Studio를 통해 시운전 중일 때 TRUE가 됩니다.
CamTableBusy	BOOL	캠 테이블 파일 저장 실행 중/대기 중	캠 테이블 파일 저장 실행 중이거나 또는 대기 중일 때 TRUE가 됩니다.
PFaultLvl	_sMC_REF_EVENT	MC 공통 부분 정지 이상	
Active	BOOL	MC 공통 부분 정지 이상 발생 중	MC 공통 정지 이상이 발생하면 TRUE가 됩니다.
Code	WORD	MC 공통 부분 정지 이상 코드	MC 공통 정지 이상 코드를 출력합니다. 이벤트 코드의 상위 4자리와 동일한 값입니다.
MFaultLvl	_sMC_REF_EVENT	MC 공통 가벼운 이상	
Active	BOOL	MC 공통 가벼운 이상 발생 중	MC 공통 가벼운 이상이 발생했을 때 TRUE가 됩니다.
Code	WORD	MC 공통 가벼운 이상 코드	MC 공통 가벼운 이상 코드를 출력합니다. 이벤트 코드의 상위 4자리와 동일한 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 정보 코드를 출력합니다. 이벤트 코드의 상위 4자리와 동일한 값입니다.

축 변수

축 변수는 시스템 정의 변수의 변수명이 _MC_AX[0-63]이며 데이터 형식이 _sAXIS_REF인 구조체 형식의 변수입니다.

여기에서는 축 변수의 구성과 멤버에 대해 자세히 설명합니다.

변수명	데이터 형식	이 름	기 능
_MC_AX[0-63]	_sAXIS_REF	축 변수	
Status	_sAXIS_REF_STA	축 Status	
Ready	BOOL	축 기동 준비 완료	축 기동 준비가 완료되어 정지 중일 때 TRUE가 됩니다. _MC_AX[XX].Status.Standstill가 'TRUE: 정지 중'과 같은 상태를 나타냅니다.
Disabled	BOOL	축 무효	축이 서보 OFF이며 정지한 상태이면 TRUE가 됩니다. 다음 축 Status는 배타적이므로 이 중 하나만 TRUE가 됩니다. Disabled/ Standstill/ Discrete/ Continuous/ Synchronized/ Homing/ Stopping/ ErrorStop/ Coordinated
Standstill	BOOL	정지 중	축이 서보 ON이며 정지한 상태이면 TRUE가 됩니다.
Discrete	BOOL	위치 결정 동작 중	목표 위치를 향해 위치 결정 제어를 실행 중일 때 TRUE가 됩니다. 위치 결정 동작 중에 오버라이드값이 '0'으로 설정되어 속도가 '0'인 상태도 포함됩니다.
Continuous	BOOL	연속 동작 중	목표 위치가 없고 연속 동작 중일 때 TURE가 됩니다. 속도 제어와 토크 제어 시에 이 상태가 됩니다. 목표 속도를 '0'으로 설정해 속도가 '0'인 상태나 연속 동작 중인 오버라이드값을 '0'으로 설정해 속도가 '0'인 상태도 포함됩니다.
Synchronized	BOOL	동기 동작 중	동기 제어를 실행 중일 때 TURE가 됩니다. 동기 제어 명령 전환 후의 동기 대기 상태도 포함됩니다.
Homing	BOOL	원점 복귀 중	MC_Home 명령 또는 MC_HomeWithParameter 명령에 따라 원점 복귀를 실행 중일 때 TURE가 됩니다.
Stopping	BOOL	감속 정지 중	MC_Stop 명령, MC_TouchProbe 명령에 따라 축이 정지할 때까지의 상태일 때 TRUE가 됩니다. MC_Stop 명령으로 축이 정지한 후 Execute가 TRUE일 때도 포함됩니다. 감속 정지 중 상태에서는 축 동작 명령을 기동할 수 없습니다(CommandAborted = TRUE).
ErrorStop	BOOL	에러 감속 정지 중	MC_ImmediateStop 명령 또는 축 가벼운 이상 발생 중(_MC_AX [XX]. MFaultLvl. Active가 'TRUE: 축 가벼운 이상 발생 중')으로 축이 정지할 때까지 및 정지 중인 상태. 이 상태에서는 축 동작 명령을 기동할 수 없습니다(CommandAborted = TRUE).
Coordinated	BOOL	다축 협조 동작 중	다축 협조 명령에 따라 축 그룹 유효인 상태에서 TRUE가 됩니다.

변수명	데이터 형식	이 름	기 능
Details	_sAXIS_REF_DET	축 제어 Status *1	
Idle	BOOL	정지 중	인포지션 대기를 제외하고 명령값을 연산 중이지 않을 때 TRUE가 됩니다. *2 Idle와 InPosWaiting은 배타적이므로 동시에 TRUE가 되지 않습니다.
InPosWaiting	BOOL	인포지션 대기	인포지션 대기일 때 TRUE가 됩니다. 인포지션 확인을 실행하는 위치 결정으로 인포지션을 확인 중인 상태입니다.
Homed	BOOL	원점 확정	원점 확정 상태일 때 TURE가 됩니다. TRUE: 원점 확정 FALSE: 원점 미확정
InHome	BOOL	원점 정지	원점 위치 범위 내에서 TURE가 됩니다. 다음의 AND 조건입니다. • 원점 확정 상태 • 피드백 현재 위치가 원점을 중심으로 '원점 위치 범위'의 폭에 들어갔을 때 명령 상태로 축 동작 중에 원점 위치를 통과하는 경우에도 TRUE가 됩니다.
VelLimit	BOOL	명령 속도 포화	동기 제어 중에 명령 속도가 최고 속도로 제한되고 있을 때 TRUE가 됩니다.
Dir	_sAXIS_REF_DIR	명령 방향 Status *3	
Posi	BOOL	정방향 명령 중	정방향으로 명령 중일 때 TRUE가 됩니다.
Nega	BOOL	역방향 명령 중	역방향으로 명령 중일 때 TRUE가 됩니다.
DrvStatus	_sAXIS_REF_STA_DRV	서보 드라이버 상태*4	
ServoOn	BOOL	서보 ON	서보 모터에 통전 중일 때 TRUE가 됩니다.
Ready	BOOL	서보 Ready	서보 Ready 상태이면 TRUE가 됩니다.
MainPower	BOOL	주회로 전원	서보 드라이버의 주회로 전원이 통전 중일 때 TRUE가 됩니다.
P_OT	BOOL	정방향 한계 입력	정방향 한계 입력이 유효일 때 TRUE가 됩니다.
N_OT	BOOL	역방향 한계 입력	역방향 한계 입력이 유효일 때 TRUE가 됩니다.
HomeSw	BOOL	원점 근방 입력	원점 근방 입력이 유효일 때 TRUE가 됩니다.
Home	BOOL	원점 입력	원점 입력이 유효일 때 TRUE가 됩니다. *5
ImdStop	BOOL	즉시 정지 입력	즉시 정지 입력이 유효일 때 TRUE가 됩니다.
Latch1	BOOL	외부 래치 입력 1	래치 입력 1이 유효일 때 TURE가 됩니다.
Latch2	BOOL	외부 래치 입력 2	래치 입력 2가 유효일 때 TURE가 됩니다.
DrvAlarm	BOOL	드라이버 에러 입력	드라이버 에러가 발생했을 때 TRUE가 됩니다.
DrvWarning	BOOL	드라이버 경고 입력	드라이버 경고가 발생했을 때 TRUE가 됩니다.
ILA	BOOL	드라이버 내부 기능 제한 중	서보 드라이버측의 제한 기능으로 인해 실제로 제한되고 있을 때 TRUE가 됩니다. G5 시리즈에서는 다음 중 하나에 해당합니다. *6 토크 제한/속도 제한/구동 금지 입력/ 소프트웨어 리미트
CSP	BOOL	주기적 동기 위치 (CSP) 모드 중	서보 드라이버측이 서보 ON이며 CSP 모드일 때 TRUE가 됩니다. *7

변수명	데이터 형식	이 름	기 능
CSV	BOOL	주기적 동기 속도 (CSV) 모드 중	서보 드라이버측이 서보 ON이며 CSV 모드일 때 TRUE가 됩니다. *7
CST	BOOL	주기적 동기 토크 (CST) 모드 중	서보 드라이버측이 서보 ON이며 CST 모드일 때 TRUE가 됩니다. *7
Cmd	_sAXIS_REF_CMD_DATA	축 명령값	
Pos	LREAL	명령 현재 위치	명령 위치의 현재값을 출력합니다. (단위: 명령 단위) 서보 OFF 시 및 위치 제어 모드 이외일 때 피드백 현재 위치를 출력합니다.
Vel	LREAL	명령 현재 속도	명령 속도의 현재값을 출력합니다. (단위: 명령 단위/s) 정방향 이동 시에는 플러스 부호, 역방향 이동 시에는 마이너스 부호가 붙습니다. 명령 현재 위치의 차이로 산출한 속도입니다. 서보 OFF 시 및 위치 제어 모드 이외일 때 피드백 현재 위치를 바탕으로 산출한 속도입니다.
AccDec	LREAL	명령 현재 가감속 속도	명령 가감속 속도의 현재값을 출력합니다. (단위: 명령 단위/s^2) 명령 현재 위치의 차이로 산출한 가감속 속도입니다. 가속 시에는 플러스 부호, 감속 시에는 마이너스 부호가 붙습니다. 실행 중인 명령의 명령 가감속 속도가 '0'인 경우는 '0'이 됩니다.
Jerk	LREAL	명령 현재 저크	명령 저크의 현재값을 출력합니다. (명령 단위/s^3) 가감속 속도의 절대값이 증가하면 플러스 부호, 감소하면 마이너스 부호가 붙습니다. 실행 중인 명령의 명령 가감속 속도, 명령 저크가 '0'인 경우는 '0'이 됩니다.
Trq	LREAL	명령 현재 토크	명령 토크의 현재값을 출력합니다. (단위: %) 정방향 이동 시에는 플러스 부호, 역방향 이동 시에는 마이너스 부호가 붙습니다. 토크 제어 모드 이외에서는 피드백 현재 토크와 같은 값을 출력합니다.
Act	_sAXIS_REF_ACT_DATA	축 현재값	
Pos	LREAL	피드백 현재 위치	피드백 위치의 현재값을 출력합니다. (단위: 명령 단위)
Vel	LREAL	피드백 현재 속도	피드백 속도의 현재값을 출력합니다. (단위: 명령 단위/s) 정방향 이동 시에는 플러스 부호, 역방향 이동 시에는 마이너스 부호가 붙습니다.
Trq	LREAL	피드백 현재 토크	피드백 토크의 현재값을 출력합니다. (단위: %) 정방향 이동 시에는 플러스 부호, 역방향 이동 시에는 마이너스 부호가 붙습니다.
TimeStamp *8	ULINT	타임 스탬프	축의 현재 위치를 갱신한 시각을 출력합니다. 변화 시각 취득 기능이 동작한 축에서 유효합니다. (단위: ns)
MFaultLvl	_sMC_REF_EVENT	축 가벼운 이상	
Active	BOOL	축 가벼운 이상 발생 중	축 가벼운 이상이 발생하면 TRUE가 됩니다.
Code	WORD	축 가벼운 이상 코드	축 가벼운 이상 코드를 출력합니다. 이벤트 코드의 상위 4자리와 동일한 값입니다.

변수명		데이터 형식	이 름	기 능
Obsr		_sMC_REF_EVENT	축 감시 정보	
	Active	BOOL	축 감시 정보 발생 중	축 감시 정보가 발생했을 때 TRUE가 됩니다.
	Code	WORD	축 감시 정보 코드	축 감시 정보 코드를 출력합니다. 이벤트 코드의 상위 4자리와 동일한 값입니다.
Cfg		_sAXIS_REF_CFG	축 기본 설정 *9	
	AxNo	UINT	축 번호	축 논리 번호를 나타냅니다. _sAXIS_REF 참조 시에 축 번호를 인식하기 위해 참조합니다.
	AxEnable	_eMC_AXIS_USE	축 사용	축의 사용 여부를 표시합니다. 0: _mcNoneAxis(미작성 축) 1: _mcUnusedAxis(미사용 축) 2: _mcUsedAxis(사용 축)
	AxType	_eMC_AXIS_TYPE	축 종류	축 종류를 표시합니다. 가상 축에서는 입출력 배선이 필요하지 않습니다. 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 표시합니다.*10 16#FFFF는 없음을 나타냅니다.
Scale		_sAXIS_REF_SCALE	단위 변환 *11	
	Num	UDINT	모터 1회전의 펄스 수	명령 위치의 모터 1회전 펄스 수를 표시합니다. 전자 기어비에 따라 명령값에서 펄스량으로 변환합니다.
	Den	LREAL	모터 1회전 이동량	명령 위치 모터 1회전의 워크 이동량을 표시합니다.
	Units	_eMC_UNITS	표시 단위	명령 위치의 표시 단위를 표시합니다. 0: _mcPls(pulse) 1: _mcMm(mm) 2: _mcUm(μm) 3: _mcNm(nm) 4: _mcDeg(degree) 5: _mcInch(inch)

*1 명령의 제어 상태를 나타냅니다.
*2. 속도 '0'에서의 동작, 편차 카운터 리셋 중, 동기 제어 중 및 다축 협조 동작 중도 연산 중인 상태에 포함됩니다.
*3. 명령 중인 이동 방향을 나타냅니다.
*4. 드라이버 등의 상태를 나타냅니다.
*5. Sysmac Studio [축 기본 설정] 화면의 [고급 설정]에서 [디지털 입력]의 [엔코더 Z상 검출]로 설정한 신호 상태입니다. 타사의 드라이버로는 이 신호를 PDO에 매핑할 수 없는 경우가 있습니다. 연결할 드라이버의 설명서를 확인하십시오.
*6. PDO에 매핑된 'Status 워드(6041Hex)'의 비트 11(내부 제한 기능 유효) 상태를 표시합니다. TRUE가 되는 조건은 서보 드라이버의 사양에 따릅니다. 연결한 서보 드라이버의 설명서를 참조하십시오.
*7. PDO에 매핑된 '오퍼레이션 모드 표시(6061Hex)' 값을 바탕으로 표시합니다. CSP, CSV, CST가 TRUE가 되는 조건은 서보 드라이버의 사양에 따릅니다. 연결한 서보 드라이버의 설명서를 참조하십시오.
PDO에 '오퍼레이션 모드 표시(6061Hex)'가 매핑되어 있지 않으면 항상 FALSE가 됩니다.
*8. CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용할 수 있습니다.
*9. 축 기본 설정의 설정 내용을 나타냅니다.
*10.NX 시리즈 위치 인터페이스 유니트의 경우는 위치 인터페이스 유니트를 장착한 EtherCAT 커플러 유니트의 노드 주소를 나타냅니다.
*11.전자 기어비 설정값을 표시합니다.

축 그룹 변수

축 그룹 변수는 시스템 정의 변수의 변수명이 _MC_GRP [0-31]이며 데이터 형식이 _sGROUP_REF인 구조체 형식 변수입니다.
여기에서는 축 그룹 변수의 구성과 멤버에 대해 자세히 설명합니다.

변수명	데이터 형식	이 름	기 능
_MC_GRP[0-31]	_sGROUP_REF	축 그룹 변수	
Status	_sGROUP_REF_STA	축 그룹 Status	
Ready	BOOL	기동 준비 완료	축 그룹이 정지 중이며 기동 준비가 완료되었을 때 TRUE가 됩니다. 기동 준비가 완료되는 AND 조건은 다음과 같습니다. • 구성 축에서 MC_Stop 명령이 실행되지 않는다. • _MC_GRP[XX].Status.Standby(정지 중)가 TRUE • 구성 축이 서보 ON 상태 • 구성 축의 _MC_AX[XX].Details.Homed(원점 확정)가 TRUE
Disabled	BOOL	축 그룹 무효	축 그룹 무효로 정지 중일 때 TRUE가 됩니다. 다음 축 그룹 Status는 배타적이므로 이 중 하나만 TRUE가 됩니다. Disable/ Standby/ Moving/ Stopping/ ErrorStop
Standby	BOOL	정지 중	축 그룹 동작 명령이 정지 중일 때 TRUE가 됩니다. 축 그룹 구성 축의 서보 ON/OFF 상태와는 관련이 없습니다.
Moving	BOOL	동작 중	목표 위치를 향해 축 그룹 동작 명령을 실행 중일 때 TRUE가 됩니다. 인포지션 대기 상태나 오버라이드에 의해 속도가 '0'인 상태도 포함됩니다.
Stopping	BOOL	감속 정지 중	MC_GroupStop 명령으로 축 그룹이 정지할 때까지의 상태일 때 TRUE가 됩니다. MC_GroupStop 명령으로 축이 정지한 후 Execute가 TRUE일 때도 포함됩니다. 이 상태에서는 축 그룹 동작 명령은 기동할 수 없습니다(CommandAborted = TRUE).
ErrorStop	BOOL	에러 감속 정지 중	MC_GroupImmediateStop 명령 또는 축 그룹 가벼운 이상 발생 중(_MC_GRP [XX]. MFaultLvl. Active=TRUE)으로 축 그룹이 정지할 때까지 및 정지하고 있을 때 TRUE가 됩니다. 이 상태에서는 축 그룹 동작 명령은 기동할 수 없습니다(CommandAborted = TRUE).
Details	_sGROUP_REF_DET	축 그룹 제어 Status*1	
Idle	BOOL	정지 중	인포지션 대기를 제외하고 명령값을 연산 중이지 않을 때 TRUE가 됩니다.*2 Idle와 InPosWaiting은 배타적이므로 동시에 TRUE가 되지 않습니다.
InPosWaiting	BOOL	인포지션 대기	구성 축 중 하나가 인포지션 대기일 때 TRUE가 됩니다. 인포지션 확인을 실행하는 위치 결정으로 인포지션을 확인 중인 상태입니다.

변수명	데이터 형식	이 름	기 능
Cmd	_sGROUP_REF_CMD_DATA	축 그룹 명령값	
Vel	LREAL	명령 보간 속도	명령 보간 속도의 현재값을 출력합니다. 보간 명령 현재 위치의 차이로 산출한 보간 속도입니다. 정방향 이동 시에는 플러스 부호, 역방향 이동 시에는 마이너스 부호가 붙습니다. 축 그룹이 무효이면 '0'이 됩니다.
AccDec	LREAL	명령 보간 가감속 속도	명령 보간 가감속 속도의 현재값을 출력합니다. 명령 보간 속도의 차이로 산출한 보간 가감속 속도입니다. 가속 시에는 플러스 부호, 감속 시에는 마이너스 부호가 붙습니다. 축 그룹이 무효인 경우 및 실행 중인 축 그룹 동작 명령의 명령 가감속 속도가 '0'으로 지정된 경우는 '0'이 됩니다.
MFaultLvl	_sMC_REF_EVENT	축 그룹 가벼운 이상	
Active	BOOL	축 그룹 가벼운 이상 발생 중	축 그룹 가벼운 이상이 발생했을 때 TRUE가 됩니다.
Code	UINT	축 그룹 가벼운 이상 코드	축 그룹 가벼운 이상 코드를 출력합니다. 이벤트 코드의 상위 4자리와 동일한 값입니다.
Obsr	_sMC_REF_EVENT	축 그룹 감시 정보	
Active	BOOL	축 그룹 감시 정보 발생 중	축 그룹 감시 정보가 발생했을 때 TRUE가 됩니다.
Code	WORD	축 그룹 감시 정보 코드	축 감시 정보 코드를 출력합니다. 이벤트 코드의 상위 4자리와 동일한 값입니다.
Cfg	_sGROUP_REF_CFG	축 그룹 기본 설정	
GrpNo	UINT	축 그룹 번호	축 그룹의 논리 번호를 나타냅니다. _sGROUP_REF 참조 시에 축 그룹 번호를 인식하기 위해 참조합니다.
GrpEnable	_eMC_GROUP_USE	축 그룹 사용	축 그룹의 사용 여부를 표시합니다. 0:_mcNoneGroup(미작성 축 그룹) 1:_mcUnusedGroup(미사용 축 그룹) 2:_mcUsedGroup(사용 축 그룹)
Kinematics	_sGROUP_REF_KIM	운동학 변환 설정 *3	
GrpType	_eMC_GROUP_TYPE	축 구성	다축 협조 제어의 축 구성을 나타냅니다. 0: _mcXY(2축) 1: _mcXYZ(3축) 2: _mcXYZU(4축)
Axis[0]	UINT	구성 축 선택(축 A0)	축 A0에 할당된 축 번호를 표시합니다.
Axis[1]	UINT	구성 축 선택(축 A1)	축 A1에 할당된 축 번호를 표시합니다.
Axis[2]	UINT	구성 축 선택(축 A2)	축 A2에 할당된 축 번호를 표시합니다.
Axis[3]	UINT	구성 축 선택(축 A3)	축 A3에 할당된 축 번호를 표시합니다.

*1. 명령의 제어 상태를 나타냅니다.
*2. 속도 '0'에서의 동작도 연산 중인 상태에 포함됩니다.
*3. 축 그룹의 운동학 변환을 위한 정의를 나타냅니다.

모션 제어 명령의 입력 변수

모션 제어 명령의 입력 변수 목록과 유효 범위 목록 및 열거 형식의 유효 범위 목록입니다.

입력 변수 목록

입력 변수	명칭	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다. Execute 기동 시에 다른 입력 변수가 전송됩니다. 입력값을 갱신하려면 입력값을 변경한 후 Execute를 다시 기동하면 됩니다. 명령 실행이 완료된 후에도 Execute가 TRUE인 동안 출력 변수는 유효합니다. 그 후 Execute의 종료 에지에서 Error와 ErrorID 이외의 출력 변수는 무효가 됩니다. 명령 실행 완료 전에 Execute를 FALSE로 한 경우에는 출력 변수가 적어도 1주기 동안 유효합니다.
Enable	유효	BOOL	TRUE, FALSE	FALSE	TRUE로 하면 명령 기능이 유효가 되고 FALSE로 하면 무효가 됩니다. Enable이 TRUE인 동안 다른 입력 변수는 주기마다 전송됩니다. Enable을 FALSE로 하면 Error와 ErrorID 이외의 출력 변수는 무효가 됩니다.
Positive Enable	정방향 유효	BOOL	TRUE, FALSE	FALSE	• MC_MoveJog 명령 TRUE로 하면 정방향으로 이동을 시작합니다. FALSE로 하면 이동을 종료합니다. MC_MoveJog 명령의 Velocity, Acceleration, Deceleration는 PositiveEnable 기동 시의 값이 전송됩니다. • MC_SetTorqueLimit 명령 TRUE로 하면 정방향 토크 제한이 유효가 됩니다. FALSE로 하면 정방향 토크 제한이 무효가 됩니다.
Negative Enable	역방향 유효	BOOL	TRUE, FALSE	FALSE	• MC_MoveJog 명령 TRUE로 하면 역방향으로 이동을 시작합니다. FALSE로 하면 이동을 종료합니다. MC_MoveJog 명령의 Velocity, Acceleration, Deceleration는 NegativeEnable 기동 시의 값이 전송됩니다. • MC_SetTorqueLimit 명령 TRUE로 하면 역방향 토크 제한이 유효가 됩니다. FALSE로 하면 역방향 토크 제한이 무효가 됩니다.
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *1	모션 제어 명령 다중 기동 시의 동작을 지정합니다. 0: Aborting 1: Buffered 2: BlendingLow 3: BlendingPrevious 4: BlendingNext 5: BlendingHigh
Velocity	목표 속도	LREAL	양수 *2	0	목표 속도를 지정합니다. *3
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. *4
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. *4
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. *5

입력 변수	명칭	데이터 형식	유효 범위	초기값	내 용
Distance	이동 거리	LREAL	음수, 양수, '0'	0	명령 현재 위치에서의 이동 거리를 지정합니다.
		ARRAY [0..3] OF LREAL	음수, 양수, '0'	0	직선 보간의 목표 위치를 지정합니다. *6
Position	목표 위치	LREAL	음수, 양수, '0'	0	절대 좌표의 목표 위치를 지정합니다. *6
		ARRAY [0..3] OF LREAL	음수, 양수, '0'	0	직선 보간의 목표 위치를 지정합니다. *6
VelFactor	속도 오버라이드값	LREAL	0~500	100	속도 오버라이드값을 지정합니다. 오버라이드값의 유효 범위는 '0.01~500.00'입니다. '500.00 이상'은 '500'으로 다루어지며 '0.01 미만(음수도 포함)'은 '0.01'로 다루어집니다. '0'을 지정한 경우에만 '0'으로 동작합니다. 단위는 [%]입니다.
AccFactor (Reserved)	가감속 속도 오버라이드값	LREAL	0~500	100	(Reserved)
JerkFactor (Reserved)	저크 오버라이드값	LREAL	0~500	100	(Reserved)
Reference Type *7	위치 종류 선택	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *1	주축의 입력 정보를 지정합니다. 0: 명령 위치(직전 프라이머리 주기에서의 계산값) 1: 피드백 위치(동일 프라이머리 주기에서의 취득값) 2: 명령 위치(동일 프라이머리 주기에서의 계산값)
Feed Distance	피딩 거리	LREAL	음수, 양수, '0'	0	인터럽트 입력 후의 이동 거리를 지정합니다.
FeedVelocity	피딩 속도	LREAL	양수	0	인터럽트 입력 후의 이동 목표 속도를 지정합니다.
ErrorDetect	에러 검출 선택	BOOL	TRUE, FALSE	FALSE	인터럽트 미입력 시 이상 검출 여부를 지정합니다. TRUE: 이상 검출 있음 FALSE: 이상 검출 없음
Periodic	반복 모드	BOOL	TRUE, FALSE	FALSE	지정된 캠 테이블을 반복 실행할지 1회만 실행할지를 지정합니다. TRUE: 반복 FALSE: 반복하지 않음
StartMode	시작 위치 방식 지정	_eMC_ START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	0 *1	MasterStartDistance(주축 추종 거리)가 사용할 좌표 계통을 지정합니다. 0: 절대 위치 1: 상대 위치
StartPosition	캠 테이블 시작 위치	LREAL	음수, 양수, '0'	0	캠 테이블의 시작점(위상=0)을 주축의 절대 위치로 지정합니다.
MasterStart Distance	주축 추종 거리	LREAL	음수, 양수, '0'	0	종축이 캠 동작을 시작할 때의 주축 위치를 지정합니다. StartMode(시작 위치 방식 지정)에서 [절대 위치 결정]을 지정한 경우는 주축의 절대 위치를 지정합니다. 또한 상대 위치 결정을 지정한 경우는 StartPosition(캠 테이블 시작 위치)으로부터의 상대량을 지정합니다.
Master Scaling	주축 계수	LREAL	양수값(>0.0)	1.0	지정된 스케일로 주축의 위상을 확대/축소합니다.
SlaveScaling	종축 계수	LREAL	양수값(>0.0)	1.0	지정된 스케일로 종축의 변위를 확대/축소합니다.
MasterOffset	주축 오프셋	LREAL	음수, 양수, '0'	0	지정된 오프셋값으로 주축 위상을 이동합니다.
SlaveOffset	종축 오프셋	LREAL	음수, 양수, '0'	0	지정된 오프셋값으로 종축 변위를 이동합니다.

입력 변수	명칭	데이터 형식	유효 범위	초기값	내 용
Cam Transition (Reserved)	캠 트랜지션 지정	_eMC_CAM_ TRANSITION	0: _mcCTNone	0 *1	(Reserved)
OutMode (Reserved)	동기 해제 모드 선택	_eMC_OUT_ MODE	0: _mcStop	0 *1	(Reserved)
Direction	방향 선택	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	4/0 *1	동작 방향을 지정합니다. 0: 정방향 지정 1: 근거리 회전 지정 2: 역방향 지정 3: 현재 방향 지정 4: 방향 지정 없음
Continuous (Reserved)	계속 방향 선택	BOOL	TRUE, FALSE	FALSE	(Reserved)
Ratio Numerator	기어비 분자	DINT*8	양수 또는 음수*8	10000	주축과 종축 사이의 전자 기어 분자를 지정합니다.
Ratio Denominator	기어비 분모	UDINT*9	양수	10000	주축과 종축 사이의 전자 기어 분모를 지정합니다.
MasterSync Position	주축 동기 위치	LREAL	음수, 양수, '0'	0	절대 좌표의 주축 동기 위치를 지정합니다.
SlaveSync Position	종축 동기 위치	LREAL	음수, 양수, '0'	0	절대 좌표의 종축 동기 위치를 지정합니다.
Slave Distance	종축 이동 거리	LREAL	음수, 양수, '0'	0	종축의 이동 거리를 지정합니다.
Master Distance	주축 이동 거리	LREAL	양수 또는 '0'	0	주축의 이동 거리를 지정합니다.
Master DistanceIn ACC	주축 가속 이동 거리	LREAL	양수 또는 '0'	0	종축 가속 시의 주축 이동 거리를 지정합니다.
Master DistanceIn DEC	주축 감속 이동 거리	LREAL	양수 또는 '0'	0	종축 감속 시의 주축 이동 거리를 지정합니다.
LinkOption	동기 시작 조건 선택	_eMC_ LINKOPTION	0: _mcCommandExecution 1: _mcTriggerDetection 2: _mcMasterReach	0 *1	종축이 주축에 동기할 조건을 지정합니다. 0: 기동 시 1: 트리거 검출 시 2: 주축이 주축 추종 거리에 도달 시
Combine Mode	가감산 방법 선택	_eMC_COMBINE_ MODE	0: _mcAddAxes 1: _mcSubAxes	0 *1	합성 방법을 지정합니다. 0: 가산 1: 감산
Ratio Numerator Master (Reserved)	주축 기어비 분자	DINT*8	양수 또는 음수*8	10000	주축과 종축 사이의 전자 기어 분자를 지정합니다.
Ratio Denominator Master (Reserved)	주축 기어비 분모	UDINT*9	양수	10000	주축과 종축 사이의 전자 기어 분모를 지정합니다.
Ratio Numerator Auxiliary (Reserved)	보조축 기어비 분자	DINT*8	양수 또는 음수*8	10000	보조축과 종축 사이의 전자 기어 분자를 지정합니다.
Ratio Denominator Auxiliary (Reserved)	보조축 기어비 분모	UDINT*9	양수	10000	보조축과 종축 사이의 전자 기어 분모를 지정합니다.
Reference TypeMaster	주축 위치 종류 선택	_eMC_ REFERENCE_ TYPE	1: _mcFeedback 2: _mcLatestCommand	2 *1	주축의 위치 종류를 지정합니다. 1: 피드백 위치(동일 프라이머리 주기에서의 취득값) 2: 명령 위치(동일 프라이머리 주기에서의 계산값)
Reference Type Auxiliary	보조축 위치 종류 선택	_eMC_ REFERENCE_ TYPE	1: _mcFeedback 2: _mcLatestCommand	2 *1	보조축의 위치 종류를 지정합니다. 1: 피드백 위치(동일 프라이머리 주기에서의 취득값) 2: 명령 위치(동일 프라이머리 주기에서의 계산값)
PhaseShift	위상 보정량	LREAL	음수, 양수, '0'	0	주축의 위상 보정량을 지정합니다.*6

입력 변수	명칭	데이터 형식	유효 범위	초기값	내 용
Torque	목표 토크	LREAL	0 ~ 1000.0	300.0	서보 드라이버로 출력할 목표 토크를 0.1% 단위로 지정합니다. 정격 토크를 100%로 했을 때의 비율로 지정합니다. 단위는 '%'입니다.
TorqueRamp	토크 램프	LREAL	양수 또는 '0'	0	현재 토크에서 목표 토크를 출력할 때까지의 토크 변화율을 지정합니다. 단위는 '%/s'입니다.
Positive Value	정방향 토크 제한값	LREAL	0.1 ~ 1000.0, 또는 '0.0'	300.0	정방향에 대한 토크 제한값을 0.1% 단위로 지정합니다. 축 파라메터의 '정방향 토크 제한 상한값'을 넘는 값이 입력되면 [정방향 토크 제한 상한값]이 됩니다. '0' 및 음수를 지정했을 때만 '0'으로 동작합니다.
Negative Value	역방향 토크 제한값	LREAL	0.1 ~ 1000.0, 또는 '0.0'	300.0	역방향에 대한 토크 제한값을 0.1% 단위로 지정합니다. 축 파라메터의 '역방향 토크 제한 상한값'을 넘는 값이 입력되면 [역방향 토크 제한 상한값]이 됩니다. '0' 및 음수를 지정했을 때만 '0'으로 동작합니다.
WindowOnly	창 유효	BOOL	TRUE, FALSE	FALSE	창 마스크의 유효/무효를 지정합니다.
FirstPosition	시작 위치	LREAL	음수, 양수, '0'	0	시작 위치를 지정합니다.
LastPosition	종료 위치	LREAL	음수, 양수, '0'	0	종료 위치를 지정합니다.
StopMode	정지 방법 선택	_eMC_STOP_MODE	1: _mcImmediateStop 2: _mcImmediateStop FEReset 3: _mcFreeRunStop 4: _mcNonStop	4 *1	정지 방법을 지정합니다. 1: 즉시 정지합니다. 2: 즉시 정지하고 편차 카운터 리셋을 실행합니다. 3: 즉시 정지하고 서보를 OFF합니다. 4: 정지하지 않습니다.
Relative (Reserved)	상대 위치 선택	BOOL	TRUE, FALSE	FALSE	(Reserved)
Execution Mode (Reserved)	실행 모드 선택	_eMC_EXECUTION_MODE	0: _mcImmediately	0 *1	(Reserved)
Permitted Deviation	축간 편차 허용값	LREAL	양수 또는 '0'	0	허용 가능한 주축과 종축 위치 편차의 최대값을 지정합니다.
CmdPos Mode	명령 현재 위치 카운터 선택	_eMC_CMDPOS_MODE	0: _mcCount	0 *1	0: 피드백 현재 위치를 사용해 명령 현재 위치를 갱신합니다. 원점 확정 상태가 계속됩니다.
CoordSystem	좌표 계통	_eMC_COORD_SYSTEM	0: _mcACS	0 *1	좌표 계통을 지정합니다. 0: 축 좌표 계통(ACS)
Transition Mode	트랜지션 모드 (이동 모드)	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCorner Superimposed	0 *1	동작 경로를 지정합니다. 0: 트랜지션 무효 10: ConnerSuperimpose
MoveMode	이동 방법 선택	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative	0 *1	이동 방법을 선택합니다. 0: 절대값 위치 결정 1: 상대값 위치 결정
CircAxes	원호축 지정	ARRAY [0,1] OF UINT	0~3	0	원호 보간을 실행할 축을 지정합니다. 0: 축 A0 1: 축 A1 2: 축 A2 3: 축 A3
CircMode	원호 보간 모드	_eMC_CIRC_MODE	0: _mcBorder 1: _mcCenter 2: _mcRadius	0 *1	원호 보간 방법을 지정합니다. 0: 통과점 지정 1: 중심점 지정 2: 반지름 지정
AuxPoint	보조점	ARRAY [0,1] OF LREAL	음수, 양수, '0'	0	통과점 위치/중심 위치/반지름을 지정합니다.
EndPoint	종료점	ARRAY [0,1] OF LREAL	음수, 양수, '0'	0	목표 위치를 지정합니다.
PathChoice	경로 선택	_eMC_CIRC_PATHCHOICE	0: _mcCW 1: _mcCCW	0 *1	경로 방향을 지정합니다. 0: CW 1: CCW

입력 변수	명칭	데이터 형식	유효 범위	초기값	내 용
Parameter Number	파라메터 번호	_eMC_ PARAMETER_ NUMBER	0: _mcChkVel 1: _mcChkAcc 2: _mcChkDec 3: _mcPosiChkTrq 4: _mcNegaChkTrq 5: _mcFELmt 6: _mcChkFELmt 7: _mcSwLmtMode 8: _mcPosiSwLmt 9: _mcNegaSwLmt 10: _mcInPosTime 11:_mcInPosRange*10	0 *1	쓰기 대상 파라메터를 지정합니다. 0: 속도 경고값/보간 속도 경고값 1: 가속도 경고값/보간 가속도 경고값 2: 감속도 경고값/보간 감속도 경고값 3: 정방향 토크 경고값 4: 역방향 토크 경고값 5: 위치 편차 초과값 6: 위치 편차 경고값 7: 소프트웨어 리미트 기능 8: 정방향 소프트웨어 리미트 9: 역방향 소프트웨어 리미트 10: 인포지션 확인 시간 11: 인포지션 폭
HomingMode *11	원점 복귀 동작	_eMC_HOMING_ MODE	0:_mcHomeSwTurnHome SwOff 1:_mcHomeSwTurnHome SwOn 4:_mcHomeSwOff 5:_mcHomeSwOn 8:_mcLimitInputOff 9:_mcHomeSwTurnHome Mask 11:_mcLimitInputOnly 12:_mcHomeSwTurn HoldingTime 13:_mcNoHomeSwHolding HomeInput 14:_mcHomePreset	0 *1	재기입할 수 있는 [원점 복귀 동작]을 지정합니다. 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 마스크 거리 지정 11: 한계 입력만 12: 근방 퇴피 · 유지 시간 지정 13: 원점 근방 입력 없음 · 유지 원점 입력 지정 14: 원점 사전 설정
AxisUse*12	축 사용	_eMC_AXIS_USE	1:_mcUnusedAxis 2:_mcUsedAxis	1 *1	사용 축 또는 미사용 축을 지정합니다. 1: 미사용 축 2: 사용 축
EnableMas*13	트랙 유효	WORD	16#0000~FFFF	0	트랙별로 유효 또는 무효를 지정합니다. 최대 16트랙으로, 0번째 비트의 값이 트랙 번호 0, 15번째 비트의 값이 트랙 번호 15의 유효 또는 무효를 지정합니다. 0: 무효, 1: 유효
ValueSource (Reserved)	입력 정보	_sMC_SOURCE	–	–	(Reserved)
TimeStamp*13	타임 스탬프	ULINT	양수, '0'	0	위치를 산출하는 타임 스탬프를 지정합니다. 타임 스탬프 방식을 지원하는 디지털 입력 유니트나 엔코더 입력 유니트 내의 시각을 기준으로 하는 타임 스탬프를 지정합니다. 단위는 [ns]입니다.

*1. 유효 범위가 열거 형식인 변수는 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*2. MC_MoveJog 명령 및 MC_MoveVelocity 명령 등 '0'을 설정할 수 있는 명령도 있습니다.
*3. 명령 단위/s입니다. 명령 단위는 [mm], [μm], [nm], [degree], [inch] 또는 [pulse]로 표시됩니다.
*4. 명령 단위/s^2입니다.
*5. 명령 단위/s^3입니다.
*6. 명령 단위입니다.
*7. _mcLatestCommand를 사용하는 경우는
Master(주축)에 설정된 모션 제어 시스템 변수의 축 번호 < Slave(종축)에 설정된 모션 제어 시스템 변수의 축 번호
가 되도록 주축과 종축을 사용해야 합니다.
*8. CPU 유니트 Ver1.02 이상과 Sysmac StudioVer1.03 이상의 조합에서 사용할 수 있습니다. 그 이전의 조합에서는 데이터 형식은 'UINT', 유효 범위 '양수'가 됩니다.
*9. CPU 유니트 Ver1.02 이상과 Sysmac StudioVer1.03 이상의 조합에서 사용할 수 있습니다. 그 이전의 조합에서는 데이터 형식은 'UINT'가 됩니다.
*10.CPU 유니트 Ver1.01 이상과 Sysmac StudioVer1.02 이상의 조합에서 사용할 수 있습니다.
*11.CPU 유니트 Ver1.03 이상과 Sysmac StudioVer1.04 이상의 조합에서 사용할 수 있습니다.
*12.CPU 유니트 Ver1.04 이상과 Sysmac StudioVer1.05 이상의 조합에서 사용할 수 있습니다.
*13.CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용할 수 있습니다.

입력 변수의 유효 범위

여기에서는 모션 제어 명령 입력 변수의 유효 범위에 대해 설명합니다. 명령별 유효 범위에 대해서는 각 명령에 대한 설명을 참조하십시오.

● BOOL 형식 입력 변수

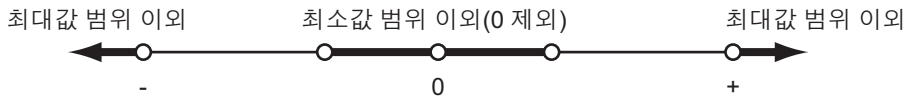
FALSE 이외는 TRUE로 간주합니다. 따라서 범위 이외가 되는 이상은 발생하지 않습니다.

● 열거(ENUM) 형식 입력 변수

유효 범위 이외를 이상으로 간주합니다.

● 유효 범위가 '전체 범위', '양수', '음수'로 기재된 입력 변수

각 입력 변수의 유효 범위와 범위 밖의 숫자가 설정된 경우의 동작은 다음 표와 같습니다.



입력 변수	명칭	유효 범위	최대값 범위 이외	최소값 범위 이외(0 제외)
Velocity	속도	0, 또는 (-1 ≦ 이며 ≦- '최고 속도'), (1 ≦ 이며 ≦ '최고 속도')*1	양수인 경우: '최고 속도' 음수인 경우: -'최고 속도' 로고정됩니다. *2	양수인 경우: 1pulse/s 음수인 경우: -1pulse/s 로 고정됩니다.*2
Acceleration	가속도	0, 또는 (0.004 ≦ 이며 ≦ '최대 가속도')*3	최대 가속도로 고정됩니다. 단, 가속 시간*4 이 125μs 미만이 되면 125μs로 고정됩니다.	양수인 경우: 0.004pulse/s^{2}로 고정됩니다. 단, 가속 시간 *4이 250s를 초과하면 250s로 고정됩니다. 음수인 경우: 이상이 발생합니다.
Deceleration	감속도	0, 또는 (0.004 ≦ 이며 ≦ '최대 감속도')*5	최대 감속도로 고정됩니다. 단, 감속 시간 *4이 125μs 미만이 되면 125μs로 고정됩니다.	양수인 경우: 0.004pulse/s^{2}로 고정됩니다. 단, 감속 시간 *4이 250s를 초과하면 250s로 고정됩니다. 음수인 경우: 이상이 발생합니다.
Jerk	저크	0, 또는 (0.000016 ≦ 이며 25,600,000,000,000,000 pulse/s^3)	25,600,000,000,000,000 pulse/s^{3}로 고정됩니다. 단, 가속 저크 적용 시간 *6, 또는 감속 저크 적용 시간*6 이 125μs 미만이 되면 각각 125μs로 고정됩니다.	0.000016 pulse/s^{3}로 고정됩니다. 단, 가속 저크 적용 시간 *6, 또는 감속 저크 적용 시간 *6 이 250s를 초과하면 각각 250s로 고정됩니다. 음수인 경우는 이상이 발생합니다.
Distance	이동 거리	(0xFFFFFF0000000001) ≦ 이며 ≦(0x000000FFFFFFFFFF)	이상이 발생합니다.	최소 범위 이외는 없습니다.
Position	명령 위치	(0xFFFFFF8000000000) ≦ 이며 <(0x0000007FFFFFFFFF+1)*7	이상이 발생합니다.	최소 범위 이외는 없습니다.
VelFactor	오버 라이드값	0, 또는 0.01 ≦ 이며 ≦ 500.00*8	500.00%를 초과하면 500.00로 고정됩니다.	0.01% 미만이면 0.01%로 고정됩니다.
Velocity	보간 속도	0.000 000 000 1 ≦ 이며 ≦ '최고 보간 속도'*9	'최고 보간 속도'로 고정됩니다.	0.000 000 000 1 pulse/s로 고정됩니다.

입력 변수	명칭	유효 범위	최대값 범위 이외	최소값 범위 이외(0 제외)
Acceleration	보간 가속도	0, 또는 (0.000 000 000 000 4 ≦ 이며 ≦ '최대 보간 가속도')*10	'최대 보간 가속도'로 고정됩니다. 단, 보간 가속 시간 *11 이 125μs 미만이면 125μs로 고정됩니다.	양수인 경우: 0.000 000 000 000 4 pulse/s2로 고정됩니다. 단, 보간 가속 시간 *11이 250s를 초과하면 250s로 고정됩니다. 음수인 경우: 이상이 발생합니다.
Deceleration	보간 감속도	0, 또는 (0.000 000 000 000 000 04 ≦ 이며 ≦ '최대 보간 감속도') *12	'최대 보간 감속도'로 고정됩니다. 단, 보간 감속 시간 *11이 125μs 미만이면 125μs로 고정됩니다.	양수인 경우: 0.000 000 000 000 4 pulse/s2로 고정됩니다. 단, 보간 감속 시간 *11이 250s를 초과하면 250s로 고정됩니다. 음수인 경우: 이상이 발생합니다.
Jerk	보간 저크	0, 또는 (0.000 000 000 000 001 6 ≦ 이며 ≦ 51,200,000,000,000,000 pulse/s^3)	51,200,000,000,000,000 pulse/s^{3}로 고정됩니다. 단, 보간 가속 저크 적용 시간 *13, 또는 보간 감속 저크 적용 시간 *13이 125μs 미만이면 각각 125μs로 고정됩니다.	0.000 000 000 000 001 6 pulse/s3로 고정됩니다. 단, 보간 가속 저크 적용 시간 *13, 또는 보간 감속 저크 적용 시간 *13이 250s를 초과하면 각각 250s로 고정됩니다. 음수인 경우는 이상이 발생합니다.

*1. 축 파라메터 '최고 속도'의 상한은 400,000,000 pulse/s입니다.
*2. 유효 범위에 음수나 '0'이 포함되지 않을 경우 음수나 '0'을 지정하면 이상이 발생합니다.
*3. 축 파라메터 '최대 가속도'의 상한은 3,200,000,000,000 pulse/s^{2}입니다.
*4. '가속 시간=속도/가속도', '감속 시간=속도/감속도', '가감속 시간=가속 시간+감속 시간'으로 계산합니다.
*5. 축 파라메터 '최대 감속도'의 상한은 3,200,000,000,000 pulse/s^{2}입니다.
*6. '가속 저크 적용 시간', '감속 저크 적용 시간'이란 저크가 적용되는 시간입니다.
'가속 저크 적용 시간=가속도/저크', '감속 저크 적용 시간=감속도/저크'로 계산합니다.
*7. 펄스 단위의 절대 위치에서 부호가 있는 40bit 범위 이내여야 합니다.
*8. 단위는 [%]입니다.
*9. 축 파라메터 '최고 보간 속도'의 상한은 800,000,000 pulse/s입니다.
*10.축 파라메터 '최대 보간 가속도'의 상한은 6,400,000,000,000 pulse/s^{2}입니다.
*11.'보간 가속 시간=보간 속도/보간 가속도', '보간 감속 시간=보간 속도/보간 감속도', '가감속 시간=가속 시간+감속 시간'으로 계산합니다.
*12.축 파라메터 '최대 보간 감속도'의 상한은 6,400,000,000,000 pulse/s^{2}입니다.
*13.'보간 가속 저크 적용 시간', '보간 감속 저크 적용 시간'이란 보간 저크가 적용되는 시간입니다.
'보간 가속 저크 적용 시간=가속도/저크', '보간 감속 저크 적용 시간=감속도/저크'로 계산합니다.

열거 형식 목록

모션 제어 명령의 입력 변수에서 사용하는 ENUM 형식 목록입니다.
열거 형식의 입력 변수는 숫자에 따라 설정되는 것이 아니라 열거자에 의해 설정됩니다.

데이터 형식	유효 범위	내 용	적용 변수 (변수 이름)
_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	모션 제어 명령 다중 기동 시의 동작을 지정합니다. 0：Aborting 1：Buffered 2：BlendingLow 3：BlendingPrevious 4：BlendingNext 5：BlendingHigh	BufferMode (버퍼 모드 선택)
_eMC_CIRC_ MODE	0: _mcBorder 1: _mcCenter 2: _mcRadius	원호 보간 방법을 지정합니다. 0 : 통과점 지정 1 : 중심점 지정 2 : 반지름 지정	CircMode (원호 보간 모드)
_eMC_CAM_ TRANSITION	0: _mcCTNone	캠 재기동 시의 종축 명령값 출력 방법을 지정합니다. 0 : 제한・수정 없음	CamTransition (캠 트랜지션 지정)
_eMC_CIRC_ PATHCHOICE	0: _mcCW 1: _mcCCW	경로 방향을 지정합니다. 0 : CW 1 : CCW	PathChoice (경로 선택)
_eMC_COMBINE_ MODE	0: _mcAddAxes 1: _mcSubAxes	합성 방법을 지정합니다. 0 : 가산 1 : 감산	CombineMode (가감산 방법 선택)
_eMC_COORD_ STSTEM	0: _mcACS	좌표 계통을 지정합니다. 0: 축 좌표 계통(ACS)	CoordSystem (좌표 계통)
_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	각종 동작 방향을 지정합니다. 0 : 정방향 지정 1 : 근거리 회전 지정 2 : 역방향 지정 3 : 현재 방향 지정 4 : 방향 지정 없음	Direction(방향 선택) HomingParameter (원점 복귀 파라메터)
eMC EXECUTION_ MODE	0: _mcImmediately	(Reserved)	ExecutionMode (실행 모드 선택)
eMC LINKOPTION	0: _mcCommandExecution 1: _mcTriggerDetection 2: _mcMasterReach	종축이 주축에 동기할 조건을 지정합니다. 0 : 명령 시작 시 1 : 트리거 검출 시 2 : 주축이 주축 추종 거리에 도달 시	LinkOption (동기 시작 조건 선택)
_eMC_MOVE_ MODE	0: _mcAbsolute 1: _mcRelative 2: _mcVelocity	이동 방법을 선택합니다. 0 : 절대값 위치 결정 1 : 상대값 위치 결정 2 : 속도 제어	MoveMode (이동 방법 선택)
_eMC _OUT_MODE	0: _mcStop	동기 명령을 해제할 모드를 지정합니다. 0 : 감속 정지	OutMode (동기 해제 모드 선택) (Reserved)

데이터 형식	유효 범위	내 용	적용 변수 (변수 이름)
eMC PARAMETER_ NUMBER	0: _mcChkVel 1: _mcChkAcc 2: _mcChkDec 3: _mcPosiChkTrq 4: _mcNegaChkTrq 5: _mcFELmt 6: _mcChkFELmt 7: _mcSwLmtMode 8: _mcPosiSwLmt 9: _mcNegaSwLmt 10: _mcInPosTime 11: _mcInPosRange*1	쓰기대상 파라메터를 지정합니다. 0 : 속도 경고값/보간 속도 경고값 1 : 가속도 경고값/보간 가속도 경고값 2 : 감속도 경고값/보간 감속도 경고값 3 : 정방향 토크 경고값 4 : 역방향 토크 경고값 5 : 위치 편차 초과값 6 : 위치 편차 경고값 7 : 소프트웨어 리미트 기능 8 : 정방향 소프트웨어 리미트 9 : 역방향 소프트웨어 리미트 10 : 인포지션 확인 시간 11 : 인포지션 폭	ParameterNumber (파라메터 번호)
eMC SWLMT_MODE	0: _mcNonSwLmt 1: _mcCmdDecelerationStop 2: _mcCmdImmediateStop 3: _mcActDecelerationStop 4: _mcActImmediateStop	소프트웨어 리미트 기능의 유효/무효 및 정지 방법을 지정합니다. 0 : 소프트웨어 리미트 기능 무효 1 : 유효이며 명령 위치에 대해 감속 정지 2 : 유효이며 명령 위치에 대해 즉시 정지 3 : 유효이며 피드백 위치에 대해 감속 정지 4 : 유효이며 피드백 위치에 대해 즉시 정지	SettingValue (설정값)
eMC REFERENCE_ TYPE *2	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	위치 종류를 지정합니다. 0 : 명령 위치(직전 프라이머리 주기에서의 계산값) 1 : 피드백 위치(동일 프라이머리 주기에서의 취득값) 2 : 명령 위치(동일 프라이머리 주기에서의 계산값)	ReferenceType (위치 종류 선택) ReferenceTypeMaster (주축 위치 종류 선택) ReferenceTypeAuxiliary (보조축 위치 종류 선택)
eMC STARTMODE	0: _mcAbsolutePosition 1: _mcRelativePosition	MasterStartDistance(주축 추종 거리)가 사용할 좌표 계통을 지정합니다. 0 : 절대 위치 1 : 상대 위치	StartMode (시작 위치 방식 지정)
_eMC_STOP_ MODE	1: _mcImmediateStop 2: _mcImmediateStopFEReset 3: _mcFreeRunStop 4: _mcNonStop	정지 방법을 지정합니다. 1 : 즉시 정지합니다. 2 : 즉시 정지하고 편차 카운터를 리셋합니다. 3 : 서보를 OFF 합니다. 4 : 정지하지 않습니다.	StopMode (정지 방법 선택)
_eMC_TRIGGER_ LATCH_ID	0: _mcLatch1 1: _mcLatch2	2가지 래치 기능 중 어느 것을 사용할지 지정합니다. 0 : 래치 기능 1 1 : 래치 기능 2	LatchID (래치 ID)
eMC CMDPOS_MODE	0: _mcCount	0 : 피드백 현재 위치를 사용해 명령 현재 위치를 갱신합니다. 원점 확정 상태가 계속됩니다.	CmdPosMode (명령 현재 위치 카운트 선택)
eMC TRANSITION_ MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	동작 경로를 지정합니다. 0 : 트랜지션 무효 10 : ConnerSuperimpose	TransitionMode (트랜지션 모드 (이동 모드))
_eMC_TRIGGER_ MODE	0: _mcDrive 1: _mcController	트리거 모드를 지정합니다. 0 : 드라이버 모드 1 : 컨트롤러 모드	Mode(모드)

데이터 형식	유효 범위	내 용	적용 변수 (변수 이름)
_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	드라이버 모드일 때의 트리거 신호를 지정합니다. 0 : Z상 1 : 외부 입력	InputDrive(드라이브 트리거 입력 신호)
_eMC_HOMING_MODE*3	0:_mcHomeSwTurnHomeSwOff 1:_mcHomeSwTurnHomeSwOn 4:_mcHomeSwOff 5:_mcHomeSwOn 8:_mcLimitInputOff 9:_mcHomeSwTurnHomeMask 11:_mcLimitInputOnly 12:_mcHomeSwTurnHoldingTime 13:_mcNoHomeSwHoldingHomeInput 14:_mcHomePreset	재기입할 수 있는 [원점 복귀 동작]을 지정합니다. 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 마스크 거리 지정 11: 한계 입력만 12: 근방 퇴피 · 유지 시간 지정 13: 원점 근방 입력 없음 · 유지 원점 입력 지정 14: 원점 사전 설정	HomingMode(원점 복귀 동작)
_eMC_HOME_INPUT *3	0:_mcZPhase 1:_mcExternalSignal	원점 입력 신호를 선택합니다. 0 : Z상 입력을 사용 1 : 외부 원점 입력을 사용	HomingParameter(원점 복귀 파라메터)
_eMC_LIMIT_REVERSE_MODE *3	0:_mcErrorStop 1:_mcRevImmediateStop 2:_mcRevDecelerationStop	원점 복귀 중에 양방향 한계 입력이 'ON'한 경우의 정지 방법을 설정합니다. 0: 반전 안 함 · 축 이상으로 정지 (정지 방법은 한계 입력 정지 방법에 따름) 1: 반전 · 즉시 정지 2: 반전 · 속 정지	HomingParameter(원점 복귀 파라메터)

*1. CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서 지정할 수 있습니다.
*2. _mcLatestCommand를 사용하는 경우는 Master(주축)에 설정된 모션 제어 시스템 변수의 축 번호 < 'Slave(종축)에 설정된 모션 제어 시스템 변수의 축 번호가 되도록 주축과 종축을 설정해야 합니다.
*3. CPU 유니트 Ver.1.03 이상과 Sysmac Studio Ver.1.04 이상의 조합에서 지정할 수 있습니다.

모션 제어 명령의 출력 변수

모션 제어 명령의 출력 변수 목록은 다음과 같습니다.

출력 변수	명칭	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다. 이 때 출력 변수 Active, Error, CommandAborted는 FALSE입니다. 명령 완료 시에 입력 변수 Execute가 FALSE이면 Done은 적어도 1주기 동안 TRUE입니다. Execute가 TRUE이면 Done은 Execute가 FALSE가 될 때까지 TRUE를 유지합니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다. 제어 중이란 이 명령이 실제로 축이나 축 그룹을 제어할 때입니다. 이 때 출력 변수 Done, Error, CommandAborted는 FALSE가 됩니다.
Enabled	유효	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령을 기동하지 못했거나 또는 실행 중에 중단되었을 때 TRUE가 됩니다. 대상 축 또는 축 그룹에서 이상이 발생 중일 때는 명령을 기동할 수 없습니다. 마찬가지로 감속 정지 중일 때도 명령을 기동할 수 없습니다. 기타 명령이 기동되거나 또는 자체 명령 이외의 이상이 발생하면 명령은 중단됩니다. 이 때 출력 변수 Done, Active, Error는 FALSE가 됩니다. 입력 변수 Execute가 FALSE일 때 중단되면 CommandAborted는 적어도 1주기 동안 TRUE가 됩니다. Execute 또는 Enable이 TRUE인 경우라면 Execute 또는 Enable이 FALSE가 될 때까지 CommandAborted는 TRUE를 유지합니다.
Error	에러	BOOL	TRUE, FALSE	입력 변수 에러 또는 명령 처리에 의한 이상이 발생하면 TRUE가 됩니다.*1
ErrorID	에러 코드	WORD	*2	이상이 발생하면 에러 코드를 출력합니다. 16#0000는 정상임을 나타냅니다.
Failure	잘못된 종료	BOOL	TRUE, FALSE	명령이 정상 실행되지 않았을 때 TRUE가 됩니다.
Status	운전 가능	BOOL	TRUE, FALSE	운전 가능 상태가 되었을 때 TRUE가 됩니다.
EndOfProfile	캠 사이클 완료	BOOL	TRUE, FALSE	캠 테이블의 종료점이 실행되지 않았을 때 TRUE가 됩니다.
Index	인덱스	UINT	양수 또는 '0'	캠 데이터의 인덱스 번호를 출력합니다.
StartSync	추종 중	BOOL	TRUE, FALSE	동기를 위한 가감속 동작을 시작했을 때 TRUE가 됩니다.
RecordedPosition	래치 위치	LREAL	음수, 양수, '0'	래치한 위치를 출력합니다.*3
Invalid	축간 편차 오버	BOOL	TRUE, FALSE	축간 편차 허용값을 초과했을 때 TRUE가 됩니다.
DeviatedValue	축간 편차값	LREAL	음수, 양수, '0'	지정된 주축과 종축의 차이를 출력합니다.*3
EndPointIndex	종료점 인덱스	UINT	양수 또는 '0'	캠 테이블의 종료점 인덱스를 출력합니다.

출력 변수	명칭	데이터 형식	유효 범위	내 용
MaxDataNumber	최대 캠 데이터 수	UINT	양수	최대 캠 데이터 수를 출력합니다.
InVelocity	목표 속도 도달	BOOL	TRUE, FALSE	목표 속도에 도달했을 때 TRUE가 됩니다.
InSync	동기 중	BOOL	TRUE, FALSE	주축에 동기했거나 또는 종축이 종축 동기 위치에 도달했을 때 TRUE가 됩니다.
InGear	기어비 도달	BOOL	TRUE, FALSE	종축이 목표 속도에 도달했을 때 TRUE가 됩니다.
InCombination	가감산 중	BOOL	TRUE, FALSE	가감산 중에 TRUE가 됩니다.
InCam	캠 동작 중	BOOL	TRUE, FALSE	캠 테이블의 시작점이 실행되었을 때 TRUE가 됩니다.
InTorque	목표 토크 도달	BOOL	TRUE, FALSE	목표 토크에 도달했을 때 TRUE가 됩니다.
InFeed	피딩 중	BOOL	TRUE, FALSE	래치 입력 수신 후 피딩 중이면 TRUE가 됩니다.
InZone	존 내	BOOL	TRUE, FALSE	축 위치가 존 범위 이내일 때 TRUE가 됩니다.
Valid*4	유효	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
Command Position*4	명령 현재 위치	ARRAY [0..3] OF LREAL	음수, 양수, '0'	명령 위치의 현재값을 출력합니다.*3
ActualPosition*4	피드백 현재 위치	ARRAY [0..3] OF LREAL	음수, 양수, '0'	피드백 위치의 현재값을 출력합니다.*3
InPosition*4	인포지션	BOOL	TRUE, FALSE	모든 구성 축의 피드백 현재 위치가 목표 위치의 인포지션 폭 안에 들어갔을 때 TRUE가 됩니다.
InOperation*5	작동 중	BOOL	TRUE, FALSE	명령 작동 중에 TRUE가 됩니다.
CalcPosition*5	산출 위치	LREAL	음수, 양수, '0'	지정한 타임 스탬프의 위치를 출력합니다.*3

*1. Error(에러)는 MC_Reset(축 에러 리셋) 명령, MC_GroupReset(축 그룹 에러 리셋) 명령, ResetMCError(전체 에러 리셋) 명령을 실행할 때까지 0으로 클리어되지 않습니다.
이 사양은 PLCopen과는 다릅니다. PLCopen에서는 Execute(기동) 종료 에지에서 FALSE가 됩니다.
또한 Error가 TRUE인 동안은 모션 제어 명령이 기동하지 않습니다. 이상을 해제했을 때 Execute가 TRUE라도 기동 에지가 아니므로 명령은 기동하지 않습니다. Enable 형식의 모션 제어 명령은 Enable(유효)이 TRUE이면 기동합니다.

*2. 이벤트 코드의 상위 4자리와 동일한 값입니다. 자세한 내용은 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

*3. 명령 단위입니다. 명령 단위는 [mm], [μm], [nm], [degree], [inch] 또는 [pulse]로 표시됩니다.

*4. CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서 사용할 수 있습니다.

*5. CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용할 수 있습니다.

참고

운전 모드 변경 후에도 모션 제어 명령의 출력 변수를 참조하고자 하는 경우는 출력 파라메터의 속성이 보유 속성을 갖는 변수를 할당하십시오.
할당한 출력 파라메터를 참조함으로써 운전 모드 변경 직전의 출력 변수를 참조할 수 있게 됩니다.

모션 제어 명령의 입출력 변수

모션 제어 명령의 입출력 변수 목록은 다음과 같습니다.

입출력 변수	명칭	데이터 형식	유효 범위	내용
Axis	축	_sAXIS_REF	–	축을 지정합니다.
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다.
Auxiliary	보조축	_sAXIS_REF	–	보조축을 지정합니다.
Master	주축	_sAXIS_REF	–	주축을 지정합니다.
Slave	종축	_sAXIS_REF	–	종축을 지정합니다.
CamTable	캠 테이블	ARRAY[0..N] OF _sMC_CAM_REF	–	캠 데이터 구조체 _sMC_CAM_REF 형식의 배열 변수를 캠 테이블로 지정합니다. *1
TriggerInput	트리거 입력 변수	_sTRIGGER_REF	–	트리거 조건을 설정합니다.
TriggerVariable	트리거 변수	BOOL	TRUE, FALSE	트리거 조건에서 컨트롤러 모드를 지정했을 때 트리거가 될 입력 변수를 지정합니다.
Target	쓰기 대상	_sAXIS_REF 또는 _sGROUP_REF	–	쓰기 대상이 되는 축 또는 축 그룹을 지정합니다.
SettingValue	설정값	지정한 변수 형식에 따릅니다.	–	기록할 값을 지정합니다. 유효 범위는 ParameterNumber(파라메터 번호)에서 지정한 모션 제어 파라메터 지정에 따릅니다. *2 초기값은 '0'입니다.
Axes*3	축 그룹 구성축	ARRAY [0..3] OF UINT	–	재기입한 후의 구성축의 축 번호를 지정합니다.
Homing Parameter *4	원점 복귀 파라메터	_sHOMING_REF	–	원점 복귀 파라메터를 설정합니다.
Switches*5	스위치	ARRAY[0..255] OF _sCAMSWITCH_REF	–	스위치 구조체 _sCAMSWITCH_REF 형식의 배열 변수를 스위치의 ON/OFF 패턴 데이터로 지정합니다. 배열 요소 번호는 스위치 번호를 나타냅니다.
Outputs*5	출력 신호	ARRAY[0..15] OF _sOUTPUT_REF	–	스위치의 ON/OFF 패턴 데이터를 바탕으로 산출된 디지털 출력의 ON/OFF 시각의 출력 대상으로 출력 신호 _sOUTPUT_REF 형식의 배열 변수를 지정합니다. 배열 요소 번호는 트랙 번호를 나타냅니다. 이 배열 변수를 NX_AryDOutTimeStamp 명령의 입출력 변수로 지정하면 실제 디지털 출력을 ON/OFF합니다.
TrackOptions*5	트랙 옵션	ARRAY[0..15] OF _sTRACK_REF	–	트랙 옵션 구조체 _sTRACK_REF 형식의 배열 변수를 스위치의 동작 조건으로 지정합니다. 배열 요소 번호는 트랙 번호를 나타냅니다.

*1. 배열 요소 [N]은 Sysmac Studio에 의해 자동 설정됩니다. Sysmac Studio의 캠 편집기로 작성한 캠 데이터 변수를 지정하십시오.

*2. 변수의 데이터 형식에 대해서는 📖 'MC_Write(5-13페이지)'의 '파라메터의 데이터 형식과 유효 범위(5-14페이지)'를 참조하십시오.

*3. CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서 지정할 수 있습니다.

*4. CPU 유니트 Ver.1.03 이상과 Sysmac Studio Ver.1.04 이상의 조합에서 지정할 수 있습니다.

*5. CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 지정할 수 있습니다.

2-2 명령 목록

모션 제어 명령은 다음의 3종류가 있습니다.

종 류	개 요
공통 명령	MC 기능 모듈 공통 명령
축 명령	MC 기능 모듈이 단축 제어를 실행하기 위한 명령
축 그룹 명령	MC 기능 모듈이 다축 제어를 실행하기 위한 명령

공통 명령에 대한 자세한 내용은 '제5장 공통 명령'을, 축 명령에 대한 자세한 내용은 '제3장 축 명령'을, 축 그룹에 대한 자세한 내용은 '제4장 축 그룹 명령'을 참조하십시오.

모션 제어 명령을 NX 시리즈 위치 인터페이스 유니트에 대해 사용하는 경우, 기능에 제약이 있는 명령이나 사용할 수 없는 명령이 있습니다. 자세한 내용은 🕮 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"를 참조하십시오.

공통 명령

MC 기능 모듈의 공통 명령에 대해 설명합니다.
명령 종류란의 '관리'는 비동작 계통 명령이며 '동작'은 동작 계통 명령입니다.

P : PLCopen 기술 사양에서 정의된 명령
O : MC 기능 모듈의 자체 명령
을 나타냅니다.

명 령	명령 이름	개 요	명령 종류	
MC_SetCamTableProperty	캠 테이블 속성 갱신	입력 파라메터에서 지정한 캠 테이블의 종료점 인덱스를 갱신합니다.	관리	O
MC_SaveCamTable	캠 테이블 저장	입력 파라메터에서 지정한 캠 테이블을 비휘발성 메모리에 저장합니다.	관리	O
MC_Write	MC 설정 쓰기	모션 제어 파라메터 설정의 일부를 덮어씁니다.	관리	O

명령 기동에 의한 축 상태 이동에 대해서는 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

축 명령

MC 기능 모듈이 단축 제어를 실행하기 위한 명령에 대해 설명합니다.
명령 종류란의 '관리'는 비동작 계통 명령이며 '동작'은 동작 계통 명령입니다.

P : PLCopen 기술 사양에서 정의된 명령
O : MC 기능 모듈의 자체 명령
을 나타냅니다.

명 령	명령 이름	개 요	명령 종류	
MC_Power	운전 가능	드라이버를 운전 가능 상태로 전환합니다.	관리	P
MC_MoveJog	조그 이송	지정한 목표 속도에 따라 조그를 이송합니다.	동작	O
MC_Home	원점 복귀	모터를 움직여 한계 신호, 원점 근방 신호, 원점 신호를 사용해 기계 원점을 결정합니다.	동작	P
MC_HomeWith Parameter*1	파라메터 지정 원점 복귀	원점 복귀 파라메터를 지정하여 모터를 작동시켜 한계 신호, 원점 근방 신호, 원점 신호를 이용해서 기계 원점을 결정하는 기능	동작	O
MC_Move	위치 결정	절대값 위치 결정 또는 상대값 위치 결정을 실행합니다.	동작	O
MC_MoveAbsolute	절대값 위치 결정	절대 좌표의 목표 위치를 지정하고 위치 결정을 실행합니다.	동작	P
MC_MoveRelative	상대값 위치 결정	명령 현재 위치로부터의 이동 거리를 지정하고 위치 결정을 실행합니다.	동작	P
MC_MoveVelocity	속도 제어	서보 드라이버의 위치 제어 모드를 사용해 의사 속도 제어를 실행합니다.	동작	P
MC_MoveZeroPosition	고속 원점 복귀	절대 좌표의 '0'을 목표 위치로 위치 결정을 실행하고 원점으로 복귀합니다.	동작	O
MC_MoveFeed	인터럽트 피딩 위치 결정	외부 입력에 의해 인터럽트 입력이 발생한 위치에서의 이동 거리를 지정하고 위치 결정을 실행합니다.	동작	O
MC_Stop	강제 정지	축을 감속 정지시킵니다.	동작	P
MC_ImmediateStop	즉시 정지	축 상태에 관계 없이 입력 변수 'StopMode(정지 방법 선택)'의 정지 방법에 따라 정지합니다.	동작	O
MC_SetPosition	현재 위치 변경	축 명령 현재 위치와 피드백 현재 위치를 임의의 값으로 변경합니다.	관리	P
MC_SetOverride	오버라이드값 설정	축의 목표 속도를 변경합니다.	관리	P
MC_ResetFollowingError	편차 카운터 리셋	명령 위치와 피드백 위치의 편차를 리셋합니다.	동작	O
MC_CamIn	캠 동작 시작	지정한 캠 테이블을 사용한 캠 동작을 시작합니다.	동작	P
MC_CamOut	캠 동작 해제	입력 파라메터에서 지정한 축의 캠 동작을 종료합니다.	동작	P
MC_GearIn	기어 동작 시작	주축과 종축 사이의 기어비를 설정하고 기어 동작을 실행합니다.	동작	P
MC_GearInPos	위치 지정 기어 동작	주축과 종축 사이의 기어비를 설정하고 전자 기어 동작을 실행합니다. 동기를 시작할 주축 및 종축 위치를 지정합니다.	동작	P
MC_GearOut	기어 동작 해제	실행 중인 MC_GearIn, MC_GearInPos를 중지합니다.	동작	P
MC_MoveLink	사다리꼴 패턴 캠	지정한 주축에 동기해 위치를 결정합니다.	동작	O
MC_CombineAxes	가감속 위치 결정	2축의 명령 위치를 가산한 값 또는 감산한 값을 명령 위치로서 출력합니다.	동작	O
MC_Phasing	주축 상대값 위상 보정	동기 제어 중인 주축의 위상 보정을 실행합니다.	동작	P
MC_TorqueControl	토크 제어	서보 드라이버의 토크 제어 모드를 사용해 토크를 제어합니다.	동작	P
MC_SetTorqueLimit	토크 제한	서보 드라이버의 토크 제한 기능을 사용해 서보 드라이버 출력 토크를 제한합니다.	관리	O
MC_ZoneSwitch	존 감시	축 명령 위치 또는 피드백 현재 위치가 지정된 범위(존) 내에 존재하는지를 판정합니다.	관리	O
MC_TouchProbe	외부 래치 유효	트리거 신호 발생에 따라 축 위치를 기록합니다.	관리	P
MC_AbortTrigger	외부 래치 무효	실행 중인 래치를 무효로 합니다.	관리	P
MC_AxesObserve	축간 편차 감시	지정된 축의 명령 위치 또는 피드백 위치의 차이가 허용값을 초과하지 않는지 감시합니다.	관리	O
MC_SyncMoveVelocity	주기적 동기 속도 제어	목표 속도로 설정한 값을 프라이머리 주기마다 주기적 동기 속도 모드(csv)로 서보 드라이버에 출력합니다.	동작	O

명 령	명령 이름	개 요	명령 종류	
MC_SyncMoveAbsolute*1	주기적 동기 절대 위치 결정	축에 지정한 목표 위치를 주기적으로 출력합니다.	동작	O
MC_Reset	축 에러 리셋	축 이상을 해제합니다.	관리	P
MC_ChangeAxisUse*2	축 사용 변경	축 파라메터의 [축 사용]을 일시적으로 전환합니다.	관리	P
MC_DigitalCamSwitch*3	디지털 캠 스위치 유효	축의 위치에 따라 디지털 출력을 ON 또는 OFF합니다.	관리	P
MC_TimeStampToPos*3	타임 스탬프→축 위치 산출	지정한 타임 스탬프의 축 위치를 산출합니다.	관리	O

*1. CPU 유니트 Ver1.03 이상과 Sysmac StudioVer1.04 이상의 조합에서 사용할 수 있습니다.
*2. CPU 유니트 Ver1.04 이상과 Sysmac StudioVer1.05 이상의 조합에서 사용할 수 있습니다.
*3. CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용할 수 있습니다.

명령 기동에 의한 축 상태 이동에 대해서는 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

PLCopen 기술 사양 준수 내용에 대해서는 컴플라이언스 리스트를 참조하십시오.
컴플라이언스 리스트는 PLCopen 웹사이트에서 확인할 수 있습니다.

축 그룹 명령

MC 기능 모듈이 다축 제어를 실행하기 위한 명령에 대해 설명합니다.
명령 종류란의 '그룹 관리'는 비동작 계통 명령이며 '그룹 동작'은 동작 계통 명령입니다.

P : PLCopen 기술 사양에서 정의된 명령
O : MC 기능 모듈의 자체 명령
을 나타냅니다.

명 령	명령 이름	개 요	명령 종류	
MC_GroupEnable	축 그룹 유효	축 그룹을 유효로 합니다.	그룹 관리	P
MC_GroupDisable	축 그룹 무효	축 그룹을 무효로 합니다.	그룹 관리	P
MC_MoveLinear	직선 보간	직선 보간을 실행합니다.	그룹 동작	O
MC_MoveLinearAbsolute	절대값 직선 보간	절대 위치를 지정해서 직선 보간을 실행합니다.	그룹 동작	P
MC_MoveLInearRelative	상대값 직선 보간	상대 위치를 지정해서 직선 보간을 실행합니다.	그룹 동작	P
MC_MoveCircular2D	2축 원호 보간	2축 원호 보간을 실행합니다.	그룹 동작	O
MC_GroupStop	축 그룹 강제 정지	보간 동작 중인 모든 축을 감속 정지합니다.	그룹 동작	P
MC_GroupImmediateStop	축 그룹 즉시 정지	축 파라메터에서 선택한 방법으로 보간 동작 중인 모든 축이 즉시 정지합니다.	그룹 동작	O
MC_GroupSetOverride	축 그룹 오버라이드값 설정	보간 동작 중인 합성 목표 속도를 변경합니다.	그룹 관리	P
MC_GroupReadPosition*1	축 그룹 위치 취득	축 그룹의 명령 현재 위치와 피드백 현재 위치를 취득합니다.	그룹 관리	O
MC_ChangeAxesInGroup *1	축 그룹 구성축 쓰기	축 그룹 파라메터의 [구성축]을 일시적으로 재기입합니다.	그룹 관리	O
MC_GroupSyncMove Absolute*1	축 그룹 주기적 동기 절대 위치 제어	각 축에 지정한 목표 위치를 주기적으로 출력합니다.	그룹 관리	O
MC_GroupReset	축 그룹 에러 리셋	축 그룹 및 축의 이상을 해제합니다.	그룹 관리	P

*1. CPU 유니트 Ver1.01 이상과 Sysmac StudioVer1.02 이상의 조합에서 사용할 수 있습니다.

명령 기동에 의한 축 그룹 상태 이동에 대해서는 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

PLCopen 기술 사양 준수 내용에 대해서는 컴플라이언스 리스트를 참조하십시오.
컴플라이언스 리스트는 PLCopen 웹사이트에서 확인할 수 있습니다.

2-3 PDO 매핑

모션 제어의 각 기능을 사용하려면 사용할 기능에 필요한 오브젝트를 프로세스 데이터 통신에 매핑해야 합니다.
미리 필요한 오브젝트를 정리한 것이 PDO 맵입니다.

Sysmac Studio의 기본값에는 다음 PDO 맵이 선정되어 있습니다.

RxPDO (1704h)	컨트롤 워드(6040Hex), 목표 위치(607AHex), 목표 속도(60FFHex), 목표 토크(6071Hex), 조작 모드(6060Hex), 래치 기능(60B8Hex), 최대 프로파일 속도(607FHex), 정회전측 토크 제한(60E0Hex), 역회전측 토크 제한(60E1Hex)
TxPDO (1B02h)	에러 코드(603FHex), Status 워드(6041Hex), 피드백 위치(6064Hex), 피드백 토크(6077Hex), 조작 모드 표시(6061Hex), 래치 Status(60B9Hex), 래치 위치 1(60BAHex), 래치 위치 2(60BCHex), 디지털 입력(60FDHex)

참고

- 오므론의 서보 드라이버 G5 시리즈 R88D-KN□□□-ECT(Ver. 2.1 이상) 또는 R88D-KN□□□-ECT-L(Ver.1.1 이상)은 PDO 맵을 Sysmac Studio의 기본값 그대로 사용할 수 있습니다.
- 오므론 서보 드라이버 G5 시리즈 R88D-KN□□□-ECT에서 풀 클로즈 제어를 수행하는 경우, RxPDO에는 1701h 또는 개체의 합계 크기를 12바이트 이하로 한 1600h(Ver.2.1 이상인 경우)를 선택하십시오.

PDO 매핑 설정 방법에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

NX 시리즈 위치 인터페이스 유니트를 사용하는 경우는 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"의 'I/O 항목 매핑'에 기재된 내용을 참조하십시오.

필수 오브젝트

필수 오브젝트에는 축 종류가 서보 축에 대한 오브젝트와 엔코더 축에 대한 오브젝트가 있습니다.
필수 오브젝트를 하나도 설정하지 않은 경우는 필수 프로세스 데이터 오브젝트 미설정(에러 코드: 3460Hex)이 발생합니다.

서보 축

MC 기능 모듈 명령을 서보 축에 대해 사용하려면 다음 오브젝트를 설정해야 합니다.

입출력	기능명	프로세스 데이터
출력	컨트롤 워드	6040Hex
	목표 위치	607AHex
입력	Status 워드	6041Hex
	피드백 위치	6064Hex

사용상의 주의 사항

오므론 서보 드라이버 G5 시리즈(EtherCAT 통신 내장 타입) 이외를 사용하는 경우는 조작 모드(6060Hex)와 조작 모드 표시(6061Hex)가 필요합니다.

엔코더 축

MC 기능 모듈 명령을 엔코더 축에 대해 사용하려면 다음 오브젝트를 설정해야 합니다.

입출력	기능명	프로세스 데이터
입력	피드백 위치	4010Hex

명령 종류에 따른 설정 오브젝트

명령에 따라서는 설정이 필요한 오브젝트가 있습니다.
서보 축, 엔코더 축을 각각 설정해야 합니다.
또한 각 명령에 필요한 오브젝트가 설정되어 있지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다.

서보 축

서보 축으로 사용하려면 명령에 따라 설정이 필요한 오브젝트가 있습니다.
다음 목록을 참조해 필요한 오브젝트를 설정하십시오.
또한 다음 목록표에 게재되지 않은 명령에 대해서는 필수 오브젝트만 설정하면 서보 축 동작이 가능해집니다.

● 출력 설정

명령 이름	기능명						
	목표 속도 60FFHex	목표 토크 6071Hex	조작 모드 6060Hex*1	래치 기능 60B8Hex	최대 프로파일 속도 607FHex	정회전측 토크 제한 60E0Hex	역회전측 토크 제한 60E1Hex
MC_Home, MC_HomeWith Parameter				△*2			
MC_MoveFeed				△*3			
MC_MoveLink				△*4			
MC_TorqueControl		○	○		△*5		
MC_SetTorqueLimit						○	○
MC_TouchProbe				△*2			
MC_SyncMoveVelocity	○		○				

*1. 조작 모드(6060Hex)를 설정하는 경우는 조작 모드 표시(6061Hex)도 설정하십시오.
둘 중 하나만 설정하면 정상적으로 동작하지 않습니다.
*2 원점 복귀 동작 모드 '11', '12', '14'를 제외하고 설정이 필요합니다.
*3. Mode에서 '드라이버 모드'를 선택한 경우
*4. LinkOption(동기 시작 조건 선택)에서 [_mcTriggerDetection]를 선택하고 Mode에서 [드라이버 모드]를 선택한 경우
*5. 오므론의 서보 드라이버 G5 시리즈(EtherCAT 통신 내장 타입)를 사용 중일 때만 확인합니다.

● 입력 설정

명령 이름	기능명				
	피드백 토크 6077Hex	조작 모드 표시 6061Hex*1	래치 Status 60B9Hex	래치 위치 1 60BAHex	래치 위치 2 60BAHex
MC_Home, MC_HomeWith Parameter			△*2	△*2	
MC_MoveFeed			△*3	△*4	△*5
MC_MoveLink			△*6	△*7	△*8
MC_TorqueControl	○	○			
MC_TouchProbe			△*3	△*4	△*5
MC_SyncMoveVelocity		○			

*1. 조작 모드 표시(6061Hex)를 설정하는 경우는 조작 모드(6060Hex)도 설정하십시오.
둘 중 하나만 설정하면 정상적으로 동작하지 않습니다.
*2. 원점 복귀 동작 모드 '11', '12', '14'를 제외하고 설정이 필요합니다.
*3. Mode에서 [드라이버 모드]를 선택한 경우
*4. Mode에서 [드라이버 모드]를 선택하고 LatchID(래치 ID 선택)에서 [_mcLatch1(래치 기능 1)]을 선택한 경우
*5. Mode에서 [드라이버 모드]를 선택하고 LatchID(래치 ID 선택)에서 [_mcLatch2(래치 기능 2)]를 선택한 경우
*6. LinkOption(동기 시작 조건 선택)에서 [_mcTriggerDetection]를 선택하고 Mode에서 [드라이버 모드]를 선택한 경우
*7. LinkOption(동기 시작 조건 선택)에서 [_mcTriggerDetection]을 선택하고 Mode에서 [드라이버 모드]를 선택한 경우로, LatchID(래치 ID 선택)에서 [_mcLatch1(래치 기능 1)]을 선택한 경우
*8. LinkOption(동기 시작 조건 선택)에서 [_mcTriggerDetection]을 선택하고 Mode에서 [드라이버 모드]를 선택한 경우로, LatchID(래치 ID 선택)에서 [_mcLatch2(래치 기능 2)]을 선택한 경우

엔코더 축

엔코더 축으로 사용하려면 명령에 따라 설정이 필요한 오브젝트가 있습니다.
다음 목록을 참조해 필요한 오브젝트를 설정하십시오.
또한 다음 목록표에 게재되지 않은 명령에 대해서는 필수 오브젝트만 설정하면 엔코더 축 동작이 가능해집니다.

● 출력 설정

명령 이름	기능명	
	래치 기능 4020Hex	엔코더 입력 슬레이브 소프트 스위치 4020Hex
MC_TouchProbe	△*1	△*2

*1. Mode에서 [드라이버 모드]를 선택한 경우
*2. 오므론의 엔코더 슬레이브를 사용해 Mode에서 [드라이버 모드]를 선택한 경우

● 입력 설정

명령 이름	기능명			
	래치 Status 4030Hex	래치 위치 1 4012Hex	래치 위치 2 4013Hex	엔코더 입력 슬레이브 Status 4030Hex
MC_TouchProbe	△ *1	△ *2	△ *3	△ *4

*1. Mode에서 [드라이버 모드]를 선택한 경우
*2. Mode에서 [드라이버 모드]를 선택하고 LatchID(래치 ID 선택)에서 [_mcLatch1(래치 기능 1)]을 선택한 경우
*3. Mode에서 [드라이버 모드]를 선택하고 LatchID(래치 ID 선택)에서 [_mcLatch2(래치 기능 2)]를 선택한 경우
*4. 오므론의 엔코더 슬레이브를 사용해 Mode에서 [드라이버 모드]를 선택한 경우

3

축 명령

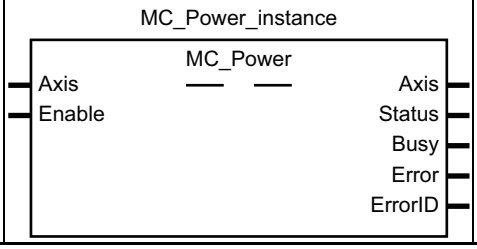
MC 기능 모듈이 단축 제어를 실행하기 위한 명령에 대해 설명합니다.

MC_Power **3-3**
MC_MoveJog **3-8**
MC_Home **3-15**
MC_HomeWithParameter **3-36**
MC_Move **3-41**
MC_MoveAbsolute **3-47**
MC_MoveRelative **3-73**
MC_MoveVelocity **3-80**
MC_MoveZeroPosition **3-95**
MC_MoveFeed **3-102**
MC_Stop **3-129**
MC_ImmediateStop **3-138**
MC_SetPosition **3-143**
MC_SetOverride **3-149**
MC_ResetFollowingError **3-154**
MC_CamIn **3-161**
MC_CamOut **3-210**
MC_GearIn **3-215**
MC_GearInPos **3-235**
MC_GearOut **3-257**
MC_MoveLink **3-262**
MC_CombineAxes **3-283**
MC_Phasing **3-294**
MC_TorqueControl **3-300**
MC_SetTorqueLimit **3-311**
MC_ZoneSwitch **3-318**
MC_TouchProbe **3-324**
MC_AbortTrigger **3-344**
MC_AxesObserve **3-348**
MC_SyncMoveVelocity **3-354**

MC_SyncMoveAbsolute 3-363
MC_Reset 3-369
MC_ChangeAxisUse 3-373
MC_DigitalCamSwitch 3-377
MC_TimeStampToPos 3-393

MC_Power

서보 드라이버를 운전 가능 상태로 전환합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
MC_Power	운전 가능	FB		MC_Power_instance (Axis := 《파라메터》, Enable := 《파라메터》, Status => 《파라메터》, Busy => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Enable	유효	BOOL	TRUE, FALSE	FALSE	TRUE로 하면 운전 가능 상태가 되고 FALSE로 하면 운전 가능 상태가 해제됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Status	운전 가능	BOOL	TRUE, FALSE	운전 가능 상태가 되었을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Status	지정 축이 운전 가능 상태가 되었을 때	• 지정 축의 운전 가능 상태가 해제되었을 때 • Error가 TRUE가 되었을 때
Busy	Enable의 기동	• Enable이 FALSE가 되었을 때 • Error가 TRUE가 되었을 때
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- Enable(유효)을 TRUE로 하면 Axis(축)에서 지정한 축이 운전 가능 상태가 됩니다.
  축을 운전 가능 상태로 하면 축을 제어할 수 있게 됩니다.
- Enable(유효)을 FALSE로 하면 Axis(축)에서 지정한 축의 운전 가능 상태가 해제됩니다.
  운전 가능 상태를 해제하면 축이 동작 명령을 수신하지 않아 축 제어가 불가능합니다. 또한 운전 가능 상태를 해제한 축에 대한 동작 명령은 이상이 됩니다. 단, 해제 상태라도 MC_Power(운전 가능) 명령, MC_Reset(축 에러 리셋) 명령은 실행 가능합니다.
- 동작 중인 축을 해제 상태로 전환할 수도 있습니다. 이 때 동작 중인 명령을 CommandAborted(실행 중단)하고 동작 명령의 출력을 정지해 해제 상태로 전환합니다.
- 앱솔루트 엔코더가 있는 서보 모터를 사용할 경우 축이 운전 가능 상태로 전환될 때 원점 미확정 상태라면 [앱솔루트 엔코더 원점 위치 오프셋] 보정을 실행해 원점 확정 상태로 합니다.
  [앱솔루트 엔코더 원점 위치 오프셋]에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

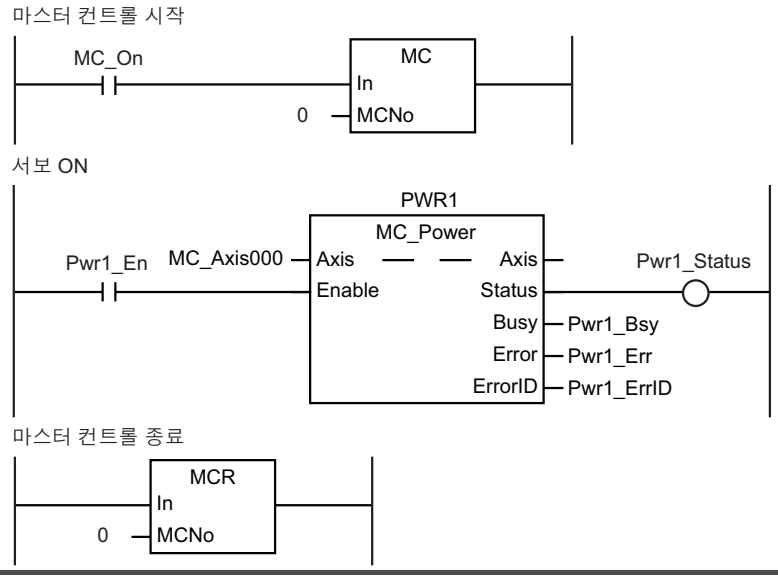
사용상의 주의 사항

- 이 명령은 축 종류가 서보 축 또는 가상 서보 축일 때 사용할 수 있습니다. 엔코더 축, 또는 가상 엔코더 축에 사용하면 이상이 됩니다.
- NX 시리즈 펄스 출력 유니트에 대해 이 명령을 실행하는 경우, 이 명령의 기능이 달라집니다. 자세한 내용은 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"를 참조하십시오.
- **동기 제어의 주축에 대해 이 명령을 실행하는 경우**

  수직축 등에서는 주축을 운전 가능 상태에서 해제한 시점에 주축의 위치가 급격하게 변화하는 경우가 있습니다. 이로 인해 종축의 동작이 급격하게 변화하기도 합니다.
  종축이 급격하게 동작하지 않도록 주축에 브레이크를 걸거나 동기 제어가 종료된 후 주축의 운전 상태를 해제하십시오.

참고

Execute 형식 모션 제어 명령은 Execute 입력에 연결한 접점의 기동으로 동작을 시작하고, 명령에 의한 제어가 완료될 때까지 동작을 계속합니다. 마스터 컨트롤에 의한 리셋 중이라도 리셋 실행 전에 명령이 실행 중인 경우는 명령에 의한 제어가 완료될 때까지 동작을 계속합니다. Execute 형식 모션 제어 명령을 인터록하려면 다음 그림과 같이 MC_Power(운전 가능) 명령을 마스터 컨트롤 영역에 배치하십시오. MC_On을 FALSE로 하면 확실하게 서보 OFF할 수 있습니다.



● CPU 유니트 동작 모드와의 관계

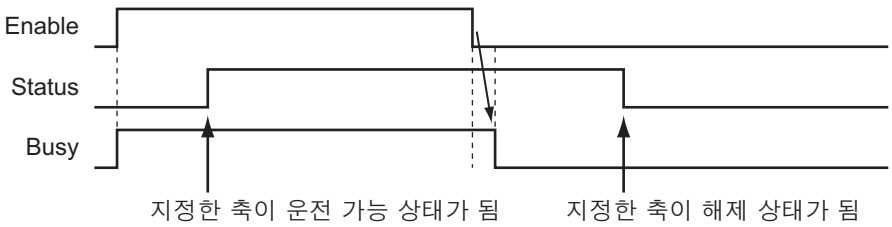
운전 모드 중에 이 명령을 사용해 운전 가능 상태로 한 후 프로그램 모드로 전환해도 운전 상태가 계속됩니다.

● 온라인 편집에 의한 명령 삭제 시

이 명령을 사용해 운전 가능 상태로 한 후 온라인 편집 중에 이 명령을 삭제해도 운전 가능 상태가 계속됩니다.

타이밍 차트

- Enable(유효)을 TRUE로 하면 명령 수신 중임을 나타내는 Busy(실행 중)가 TRUE가 됩니다.
- 그 후 축이 운전 가능 상태가 되면 Status(운전 가능)가 TRUE가 됩니다.
- Enable(유효)을 FALSE로 하면 Busy(실행 중)는 FALSE가 됩니다. Status(운전 가능)는 운전 가능 상태가 해제되었을 때 FALSE가 됩니다. Status(운전 가능)는 Enable(유효)의 TRUE/ FALSE에 관계 없이 축의 운전 가능 상태를 출력합니다.



사용상의 주의 사항

Enable(유효)을 TRUE로 한 후 축에서 처리가 끝나지 않으면 Status(운전 가능)가 TRUE가 되지 않습니다.
축을 운전할 때는 반드시 Status(운전 가능)가 TRUE가 된 것을 확인한 후 동작을 시작하십시오.

안전상의 중요 사항

컨트롤러의 전원을 켠 후 운전을 시작할 때는 EtherCAT 통신이 확립된 것을 확인한 후에 모션 제어 명령이 기동하도록 사용자 프로그램을 작성하십시오. 또한 운전 중에 EtherCAT 통신에 이상이 발생하지 않았는지 감시하는 인터록 기능을 사용자 프로그램에 포함시키십시오.

모션 명령 재기동

입력이 'Enable 형식'인 명령에는 모션 명령 재기동은 없습니다.

모션 명령 다중 기동

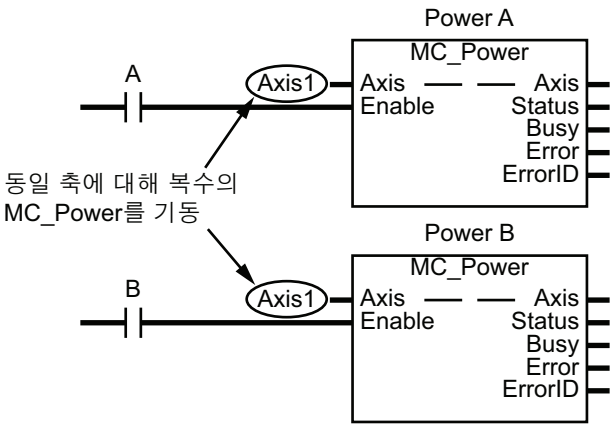
모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

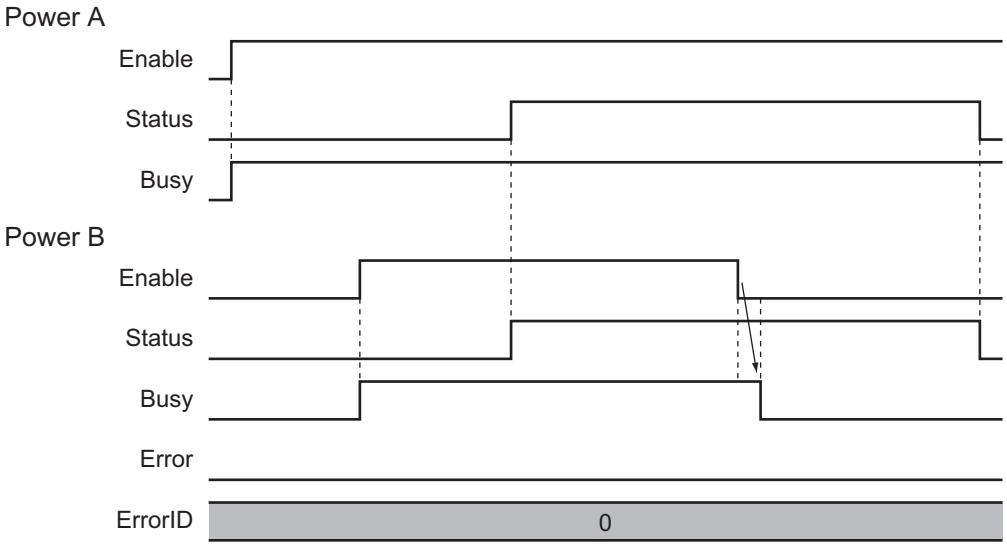
● 이 명령끼리의 다중 기동

사용상의 주의 사항

MC_Power(운전 가능) 명령을 실행 중인 축에 별도 인스턴스의 MC_Power(운전 가능) 명령을 기동하는 프로그램은 작성하지 마십시오. 기본적으로 1축에 대해 MC_Power(운전 가능) 명령은 1개로 하는 것이 좋습니다.

예를 들어 MC_Power(운전 가능) 명령을 실행 중인 축에 별도 인스턴스의 MC_Power(운전 가능) 명령을 기동하면 나중에 실행한 MC_Power(운전 가능) 명령이 우선 실행됩니다.





에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_MoveJog

지정한 목표 속도에 따라 조그를 이송합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
MC_MoveJog	조그 이송	FB	MC_MoveJog_instance MC_MoveJog Axis — — Axis PositiveEnable — Busy NegativeEnable — CommandAborted Velocity — Error Acceleration — ErrorID Deceleration	MC_MoveJog_instance (Axis := 《파라메터》, PositiveEnable := 《파라메터》, NegativeEnable := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
PositiveEnable	정방향 유효	BOOL	TRUE, FALSE	FALSE	TRUE로 하면 정방향으로 이동을 시작합니다. FALSE로 하면 이동을 종료합니다.
NegativeEnable	역방향 유효	BOOL	TRUE, FALSE	FALSE	TRUE로 하면 역방향으로 이동을 시작합니다. FALSE로 하면 이동을 종료합니다.
Velocity	목표 속도	LREAL	양수 또는 '0'	0	목표 속도를 지정합니다. 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Busy	PositiveEnable 또는 NegativeEnable 기동	• 축이 정지했을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동 (Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• PositiveEnable가 TRUE인 경우 PositiveEnable이 FALSE와 동시 • NegativeEnable가 TRUE인 경우 NegativeEnable이 FALSE와 동시 • PositiveEnable 및 NegativeEnable 둘 다 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

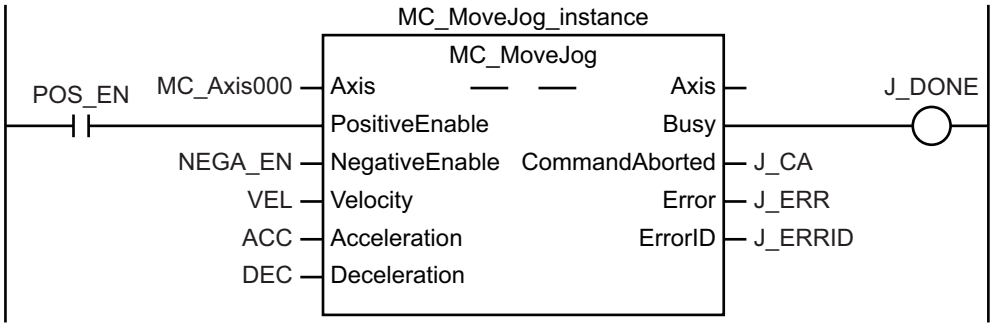
*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

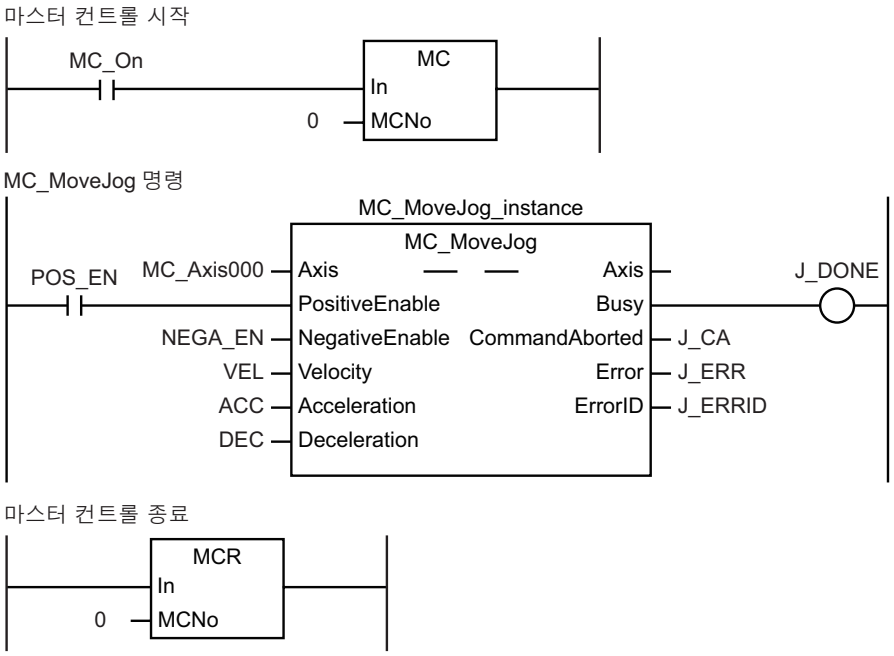
- 지정한 Velocity(목표 속도)에 따라 조그를 이송합니다.
- 정방향으로 조그를 이송하려면 PositiveEnable(정방향 유효)을 TRUE로 하고, 역방향으로 조그를 이송하려면 NegativeEnable(역방향 유효)을 TRUE로 합니다.
- PositiveEnable(정방향 유효)과 NegativeEnable(역방향 유효)을 동시에 TRUE로 하면 PositiveEnable (정방향 유효)을 우선해 정방향으로 조그가 이송됩니다.
- MC_MoveJog(조그 이송) 명령의 속도가 축 파라메터의 [조그 최고 속도]를 초과해서 지정되면 [조그 최고 속도]가 됩니다.

사용상의 주의 사항

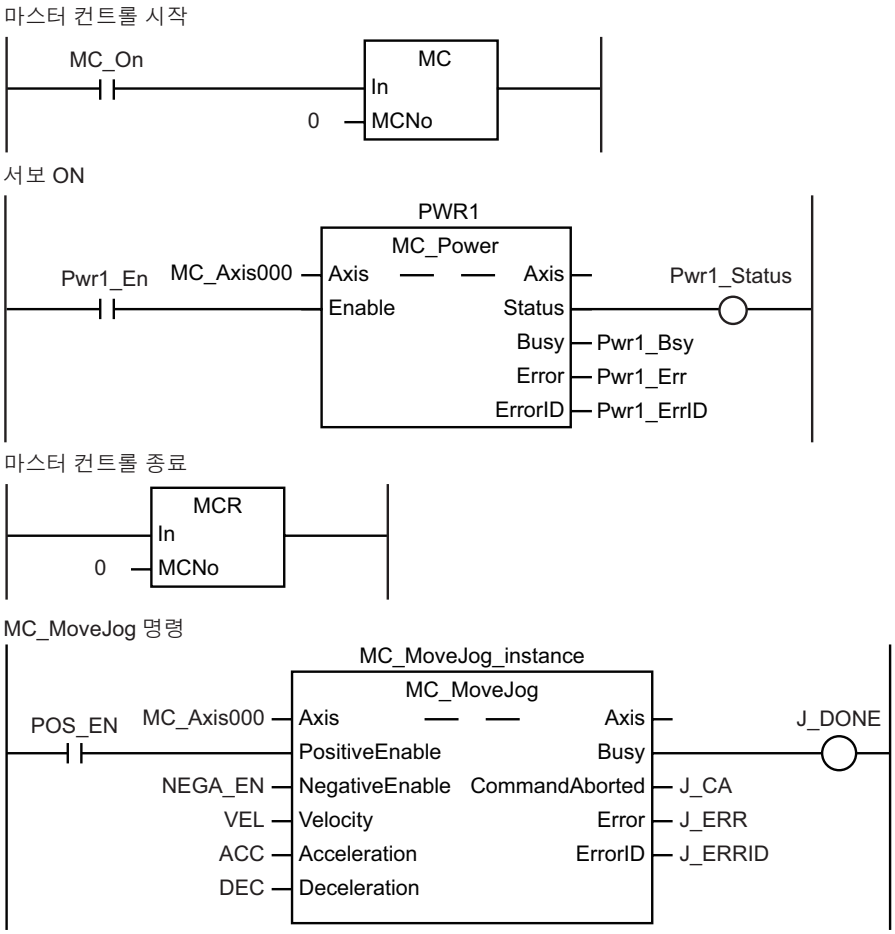
- 래더를 작성할 경우 다음과 같이 입력 변수 'PositiveEnable(정방향 유효)'이 좌측 모선에 연결되고 'NegativeEnable(역방향 유효)'은 변수를 지정해야 합니다.



만일 MC_MoveJog(조그 이송)에 대해 마스터 컨트롤 명령(MC 명령)을 사용하는 경우는 다음과 같이 기술하지 마십시오. 다음과 같이 기술하면 PositiveEnable(정방향 유효)에만 마스터 컨트롤이 기능하고 NegativeEnable(역방향 유효)에는 마스터 컨트롤이 기능하지 않게 됩니다.

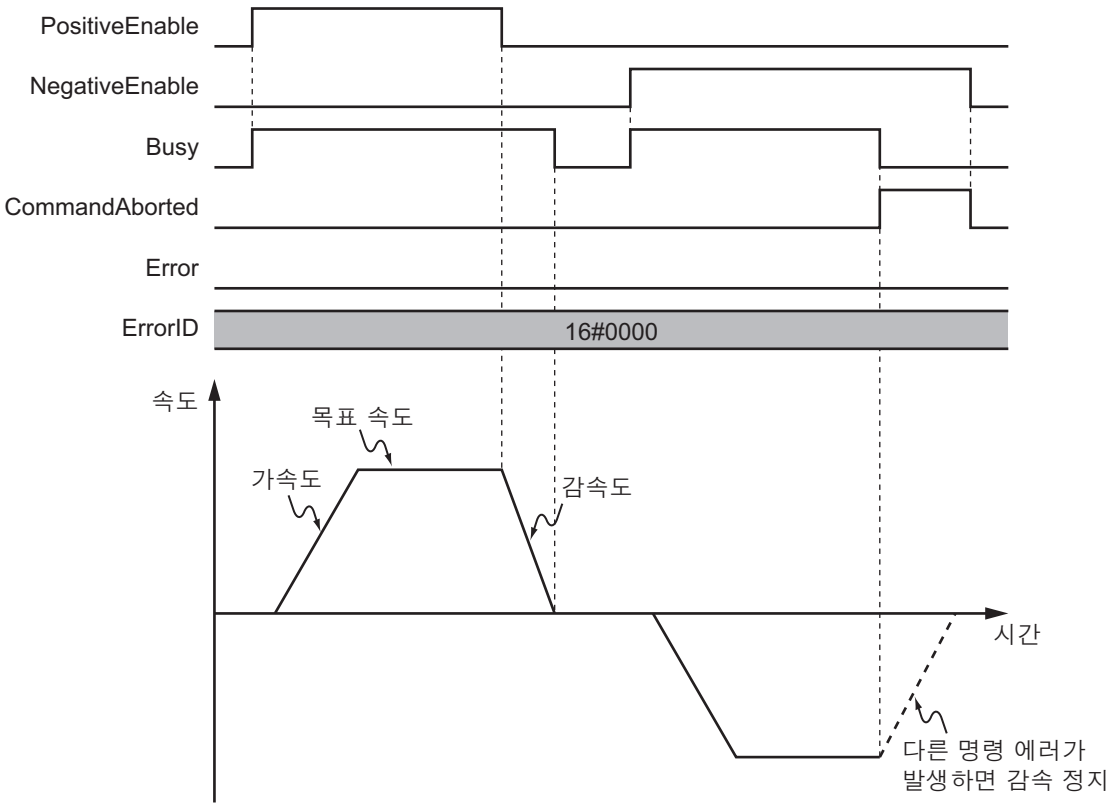


반드시 MC_Power(명령 가능)에 대해 마스터 컨트롤 명령을 사용하십시오.



타이밍 차트

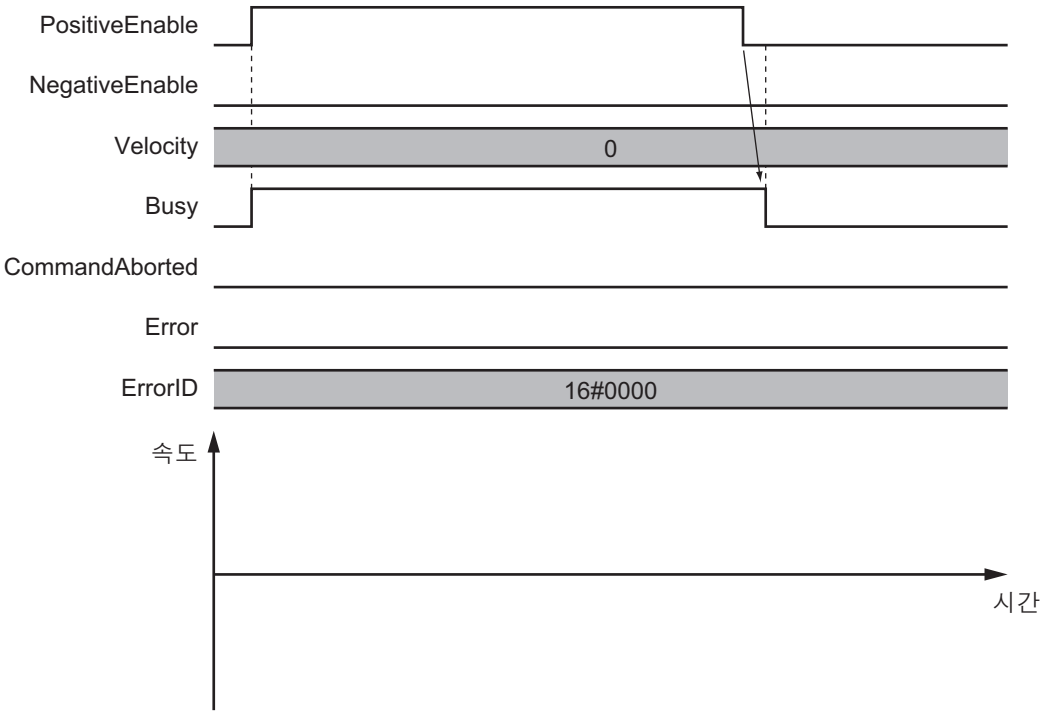
- PositiveEnable(정방향 유효) 또는 NegativeEnable(역방향 유효)의 기동과 동시에 Busy(실행 중)가 TRUE가 됩니다.
- PositiveEnable(정방향 유효) 또는 NegativeEnable(역방향 유효)이 종료되어 감속을 시작해 축이 정지하는 동시에 Busy(실행 중)가 FALSE가 됩니다.
- 다른 명령에 의해 이 명령이 중지되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중)가 FALSE가 됩니다.



입력 변수에 Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)을 지정할 수 있습니다.
입력 변수의 Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)은 PositiveEnable(정방향 유효) 또는 NegativeEnable(역방향 유효) 기동 시에만 동작에 반영됩니다. 따라서 PositiveEnable(정방향 유효) 또는 NegativeEnable(역방향 유효)이 TRUE인 동안 Velocity(목표 속도)를 변경해도 속도는 변경되지 않습니다.

● 목표 속도가 '0'일 때의 타이밍 차트

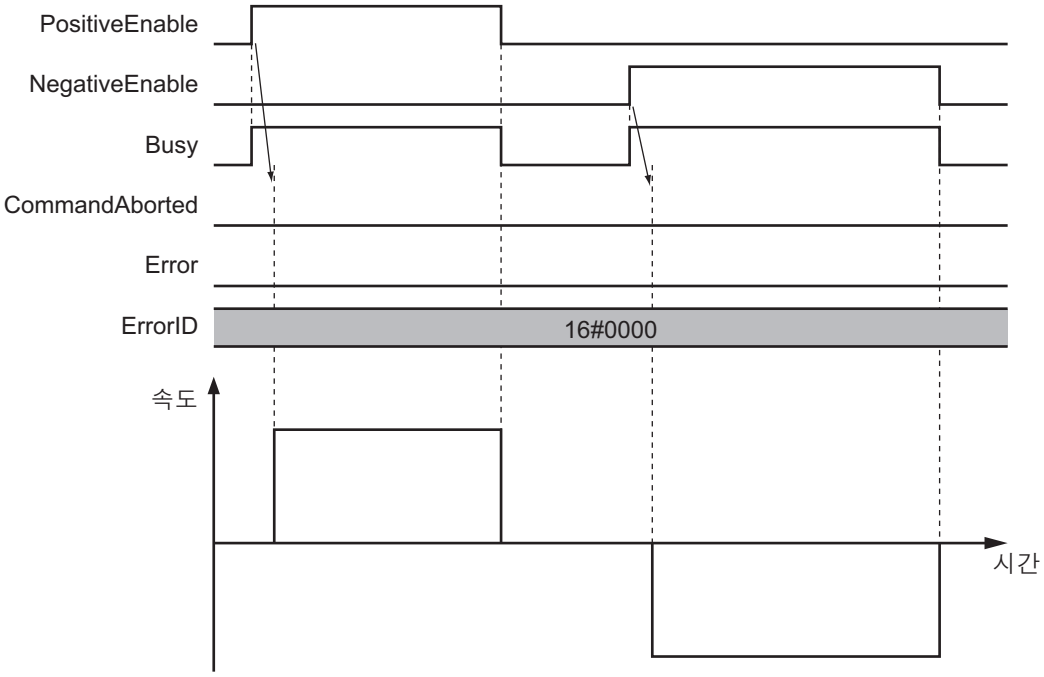
Velocity(목표 속도)를 '0'으로 지정해서 기동하면 축은 이동하지 않지만 연속 동작 중이 됩니다. Velocity (목표 속도)를 '0'으로 해서 기동한 경우의 동작 예는 다음과 같습니다.



● 가감속 속도가 '0'일 때의 타이밍 차트

Acceleration(가속도) 또는 Deceleration(감속도)을 '0'으로 지정해서 기동하면 가감속하지 않고 목표 속도에 도달합니다.

Acceleration(가속도) 및 Deceleration(감속도)이 둘 다 '0'인 경우의 동작 예는 다음과 같습니다.

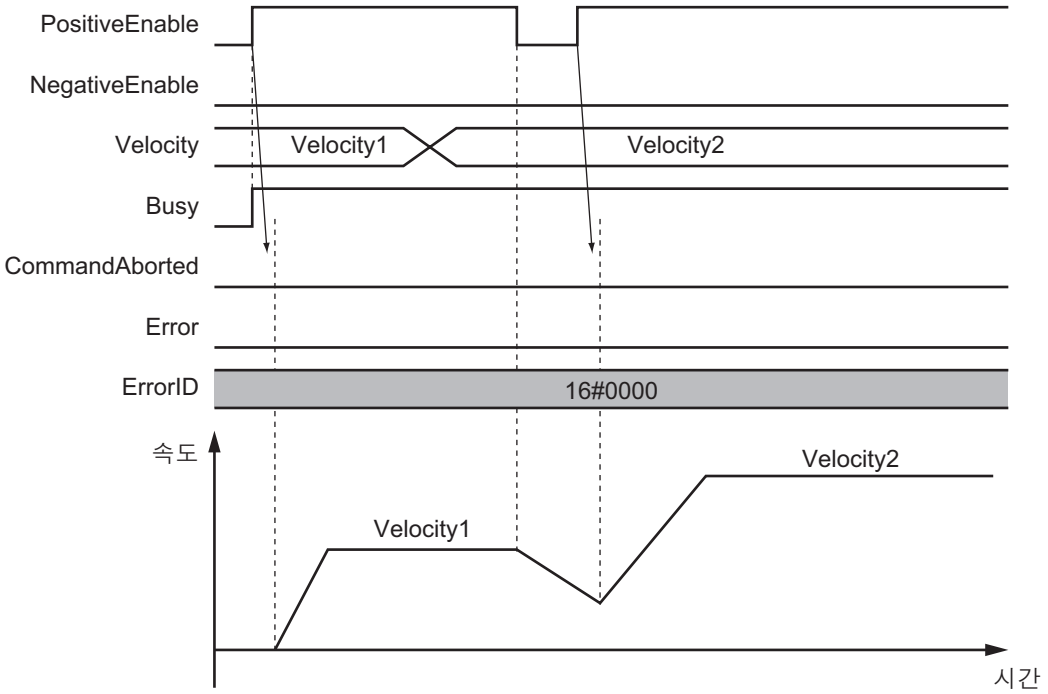


모션 명령 재기동

● 같은 방향의 Enable을 재기동하는 경우

PositiveEnable(정방향 유효) 또는 NegativeEnable(역방향 유효)을 FALSE로 해서 감속하는 도중에 다시 PositiveEnable(정방향 유효) 또는 NegativeEnable(역방향 유효)을 TRUE로 하면 감속 도중부터 가감속하여 목표 속도를 향해 이동합니다.
이 때 Velocity(목표 속도)나 Acceleration(가속도), Deceleration(감속도)이 변경되어 있으면 새로운 입력 변수로 동작합니다.
또한 축이 정지 상태가 아니므로 Busy(실행 중)는 FALSE가 되지 않습니다.

감속 중에 PositiveEnable(정방향 유효)을 TRUE로 한 경우의 동작 예는 다음과 같습니다.



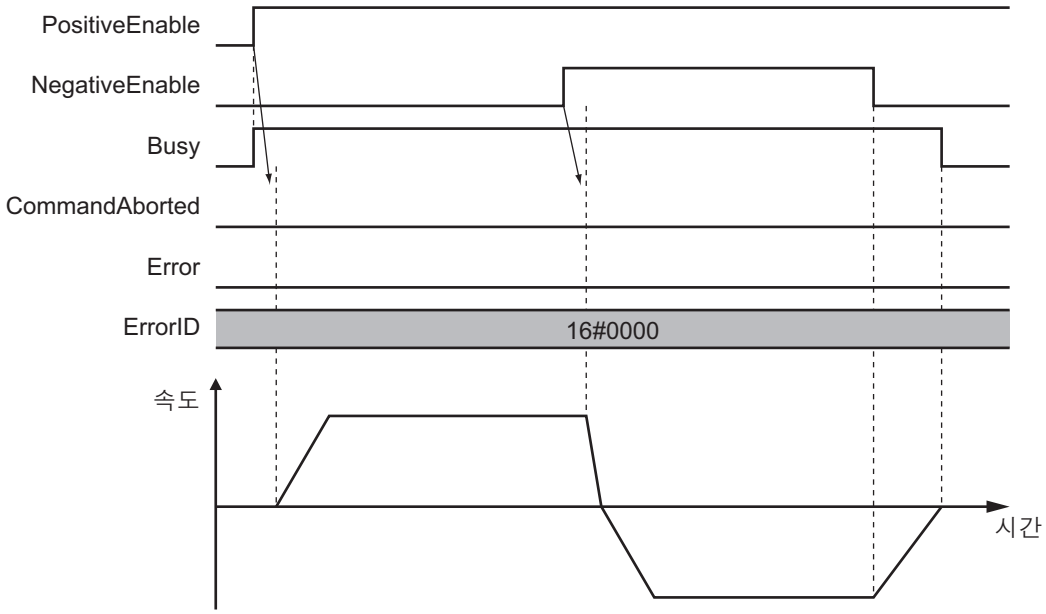
● 다른 방향의 Enable을 재기동하는 경우

PositiveEnable(정방향 유효)을 TRUE로 하고 정방향으로 조그를 이송하는 도중에 NegativeEnable(역방향 유효)을 TRUE로 하면 방향을 반전해 역방향으로 조그를 이송합니다.
이 경우 NegativeEnable(역방향 유효)을 TRUE로 했을 때의 입력 변수로 조그를 이송합니다. 입력 변수에는 Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)이 있습니다.

반전 시의 감속도 및 반전 후의 가속도는 축 파라메터의 [반전 시 동작]에 상관 없이 NegativeEnable(역방향 유효)을 TRUE로 했을 때의 입력 변수에 따릅니다.
NegativeEnable(역방향 유효)을 TRUE로 하고 역방향으로 조그를 이송하는 도중에 PositiveEnable(정방향 유효)을 TRUE로 한 경우에도 동일하게 동작합니다.

PositiveEnable(정방향 유효)을 TRUE로 한 후 NegativeEnable(역방향 유효)을 TRUE로 해서 역방향으로 조그를 이송하는 도중에 PositiveEnable(정방향 유효)이 TRUE인 채로 NegativeEnable(역방향 유효)을 FALSE로 해도 정방향으로 조그가 이송되지 않습니다.
정방향으로 조그를 이송하려면 PositiveEnable(정방향 유효)을 일단 FALSE로 하고 나서 다시 TRUE로 하십시오.
반대의 경우도 동일하게 동작합니다.

PositiveEnable(정방향 유효)을 TRUE로 하고 나서 NegativeEnable(역방향 유효)을 TRUE로 한 경우의 동작 예는 다음과 같습니다.



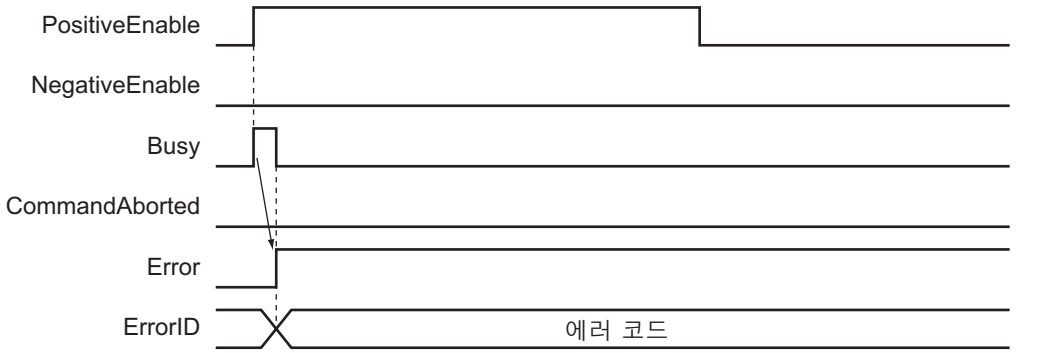
모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

이상

이 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트

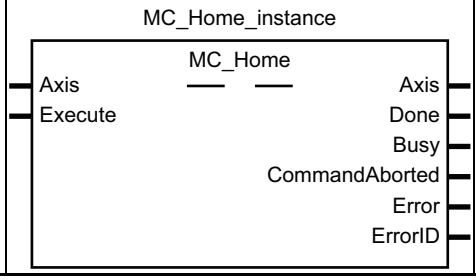


● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_Home

모터를 움직여 한계 신호, 원점 근방 신호, 원점 신호를 사용해 기계 원점을 결정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_Home	원점 복귀	FB		MC_Home_instance (Axis := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- Axis(축)에서 지정된 축에 대해 Execute(기동)가 기동하면 원점 복귀 동작을 시작합니다.
- 원점 복귀 명령에서 사용되는 각종 파라메터는 축 파라메터로 설정합니다.
- 원점 복귀 명령에는 10종류의 원점 복귀 동작 모드가 있습니다.
  Sysmac Studio의 축 파라메터 [원점 복귀 동작]에서 설정하십시오.

사용상의 주의 사항

주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

● 데이터 오브젝트 매핑

MC_Home(원점 복귀) 명령을 사용 할 경우 Sysmac Studio 축 기본 설정의 [고급 설정]에서 다음 오브젝트 데이터를 매핑하십시오.
단, 원점 복귀 동작 모드 '11', '12', '14'를 사용하는 경우는 설정하지 않아도 됩니다.

- 래치 기능(60B8Hex)
- 래치 Status(60B9Hex)
- 래치 위치 1(60BAHex)

이러한 데이터를 설정하지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다.
데이터 오브젝트 매핑에 대해서는 '2-3 PDO 매핑(2-27페이지)', "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.
NX 시리즈 위치 인터페이스 유니트를 사용하는 경우는 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"의 'I/O 항목 매핑'에 기재된 내용을 참조하십시오.

오므론의 서보 드라이버 G5 시리즈 설정

원점 복귀 명령에서 사용하는 원점 근방 신호 등의 입력 신호는 오므론의 서보 드라이버 G5 시리즈로 설정하십시오.

입력 신호의 설정 방법에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '서보 드라이버와의 연결' 및 "G5 시리즈 AC 서보 모터/드라이버 사용 설명서(EtherCAT 통신 내장 타입)(SBCE-365)", "G5 시리즈 리니어 모터/드라이버 사용 설명서(EtherCAT 통신 내장 리니어 모터 타입)(SBCE-366)"의 '시퀀스 입출력 신호'를 참조하십시오.

변수명	이 름	데이터 형식	유효 범위	단위	초기값	내 용
ProximitySignal Reference	근방 신호 선택	Input_REF	–	–	–	원점 근방 입력 신호의 입력대상을 선택합니다.
HomeSignal Reference	원점 신호 선택	Input_REF	–	–	–	원점 입력 신호의 입력대상을 선택합니다.

NX 시리즈 위치 인터페이스 유니트의 설정

NX 시리즈 위치 인터페이스 유니트를 사용하는 경우의 설정은 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"를 참조하십시오.

원점 복귀 동작 모드

기계 원점을 결정하는 동작으로, 10종류의 동작에서 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 · 유지 시간 지정
- 원점 근방 입력 없음 · 유지 원점 입력 지정
- 원점 사전 설정

각 원점 복귀 동작별로 사용하는 원점 복귀 파라메터는 다음 표와 같습니다.

(○: 파라메터 사용, ×: 파라메터 사용 안 함)

	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 유지 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 · 유지 시간 지정	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○
원점 근방 입력 없음 · 유지 원점 입력 지정	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
원점 사전 설정	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×

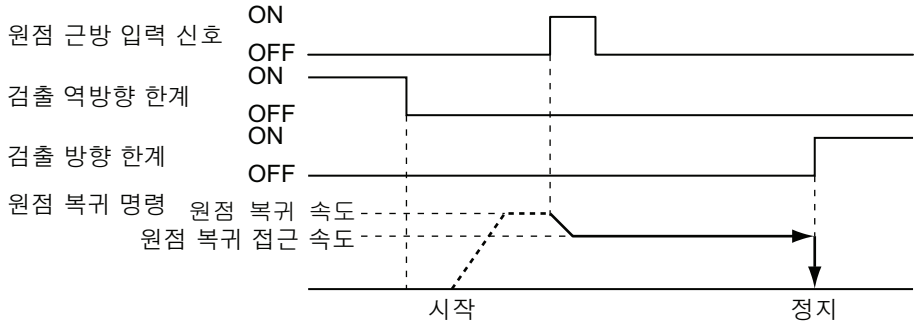
각 원점 복귀 모드의 동작에 대한 자세한 내용은 '원점 확정 동작(3-22페이지)'을 참조하십시오.

참고

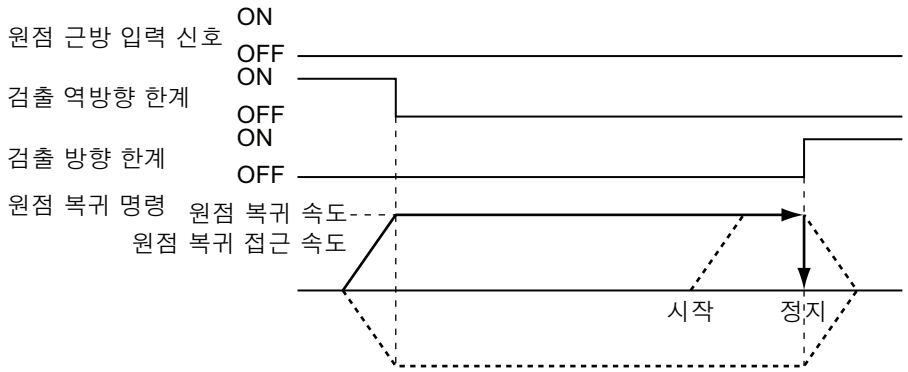
NX 시리즈 위치 인터페이스 유니트를 사용하는 경우, 유지 기능을 사용하는 원점 복귀 동작 모드를 선택할 수 없습니다. 자세한 내용은 "NX 시리즈 위치 인터페이스 유니트 사용 설명서 (SBCE-374)"를 참조하십시오.

정방향 한계 신호 입력 시 동작과 역방향 한계 신호 입력 시 동작

- 원점 복귀 중에 동작 방향의 한계 입력까지 도달했을 때 반전해서 원점 복귀를 계속시킬지, 반전시키지 않고 이상 정지시킬지를 선택합니다. 반전시키는 경우에는 정지 방법도 선택합니다.
- [반전]으로 설정하면 원점 복귀 접근 속도에서 원점 검출 방향의 한계 신호가 'ON'하면 이상 정지합니다.
  단, 근방 신호를 사용하지 않는 동작 모드[13: 원점 근방 입력 없음 · 유지 원점 입력 지정]의 경우는 이상 정지되지 않습니다.



- 또한 양방향 한계 입력 시 동작을 [반전]으로 설정하면 원점 검출 방향에 대해 반대쪽 한계 입력에서 다른 쪽 한계 입력까지 동작하고, 원점을 검출하지 못했을 때는 이상 정지합니다.



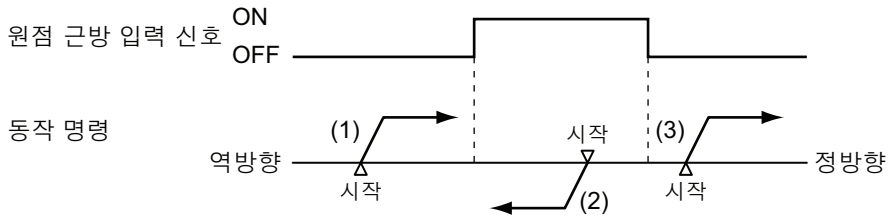
원점 복귀 시작 방향

원점 복귀 기동 시의 동작 시작 방향을 선택합니다.
단, 근방 퇴피에 의한 반전 동작을 포함하는 원점 복귀 동작 모드에서 원점 근방 신호가 'ON'인 상태로 원점 복귀를 기동한 경우는 원점 복귀 시작 방향의 설정에 관계 없이 원점 검출 방향과 반대 방향으로 동작을 시작합니다.

근방 퇴피에 의한 반전 동작을 포함하는 원점 복귀 동작 모드는 다음의 4개입니다.

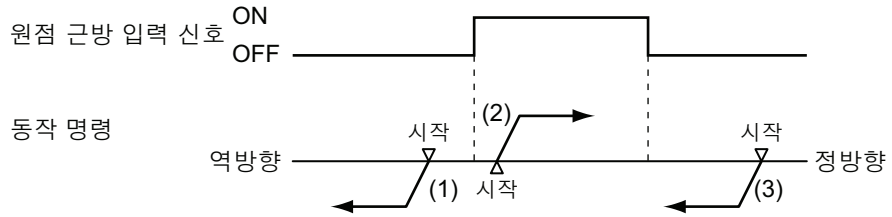
- 0 : 근방 퇴피 · 원점 근방 입력 OFF 지정
- 1 : 근방 퇴피 · 원점 근방 입력 ON 지정
- 9 : 근방 퇴피 · 원점 입력 마스크 거리 지정
- 12 : 근방 퇴피 · 유지 시간 지정

원점 복귀 시작 방향: 정
원점 검출 방향: 정



(1), (3) : 원점 근방 신호 'OFF' 상태에서 시작하기 위해 원점 복귀 시작 방향으로 동작 시작
(2) : 원점 근방 신호 'ON' 상태에서 시작하기 위해 원점 검출 방향과 반대 방향으로 동작 시작

원점 복귀 시작 방향: 역
원점 검출 방향: 역

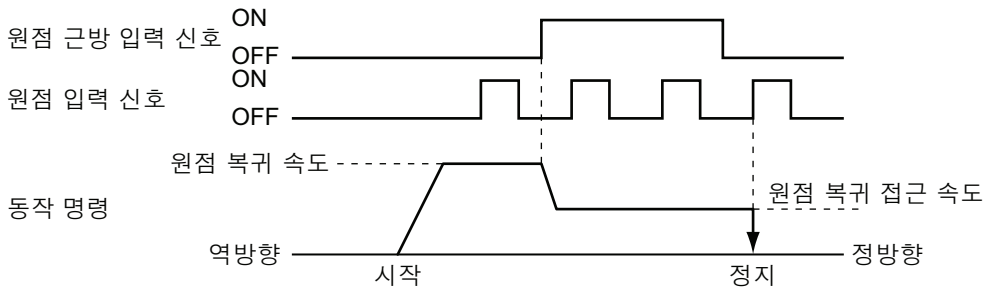


(1), (3) : 원점 근방 신호 'OFF' 상태에서 시작하기 위해 원점 복귀 시작 방향으로 동작 시작
(2) : 원점 근방 신호 'ON' 상태에서 시작하기 위해 원점 검출 방향과 반대 방향으로 동작 시작

원점 검출 방향

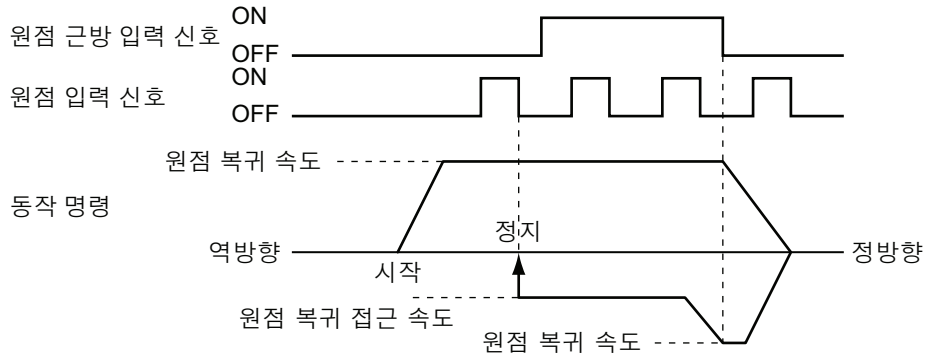
원점 입력 검출 방향을 선택합니다.
원점 검출 방향이 FALSE(정방향)인 경우의 동작은 다음과 같습니다.

동작 예: 근방 퇴피 • 원점 근방 입력 OFF 지정



또한 원점 검출 방향이 TRUE(역방향)인 경우의 동작은 다음과 같습니다.

동작 예: 근방 퇴피 • 원점 근방 입력 OFF 지정



원점 입력 마스크 거리

원점 복귀 동작 모드가 [9: 근방 퇴피・원점 입력 마스크 거리 지정] 설정 시의 피딩 거리를 설정합니다.
동작에 대한 자세한 내용은 '9: 근방 퇴피・원점 입력 마스크 거리 지정 동작(3-25페이지)'을 참조하십시오.

원점 복귀 보정값

원점 검출 후의 원점 복귀 보정량을 설정합니다.
이 때의 이동 속도는 [원점 복귀 보정 속도]에서 설정합니다.
동작에 대한 자세한 내용은 '원점 보정(3-31페이지)'을 참조하십시오.

원점 위치 오프셋

원점 복귀 동작 완료 시 현재 위치의 사전 설정입니다.
[원점 위치 오프셋]에 설정한 값이 현재 위치가 됩니다.

원점 복귀 속도

원점 복귀 시의 고속 속도입니다.

원점 복귀 접근 속도

원점 복귀 시의 근방 속도입니다.

원점 복귀 보정 속도

[원점 복귀 보정값]을 설정한 경우의 이동 속도입니다.
동작에 대한 자세한 내용은 '원점 보정(3-31페이지)'을 참조하십시오.

명령에 대한 자세한 내용

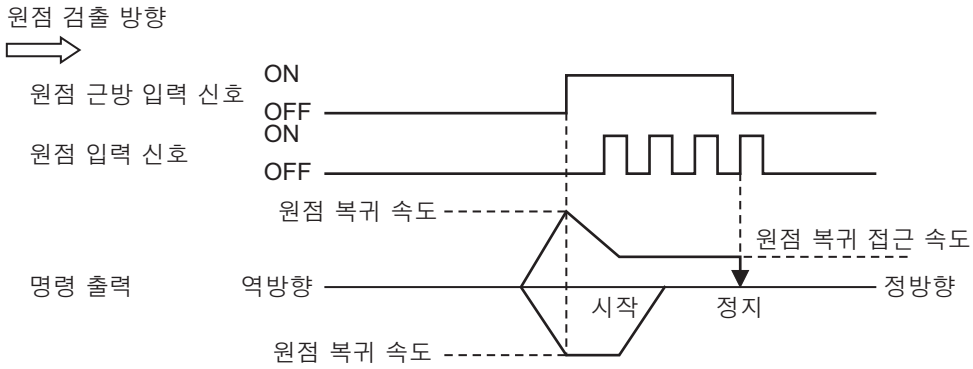
● 원점 확정 동작

여기에서는 10종류의 원점 복귀 동작 모드에 대해 설명합니다.

0: 근방 퇴피 · 원점 근방 입력 OFF 지정 동작

1 **원점 복귀 속도로 기동해 원점 근방 입력 신호의 'OFF' → 'ON'에 의해 원점 복귀 접근 속도로 감속하기 시작합니다.**

2 **원점 복귀 접근 속도에 도달해 원점 근방 입력 신호가 'ON' → 'OFF'된 후 최초의 원점 입력에서 정지하고 기계 원점이 확정됩니다.**

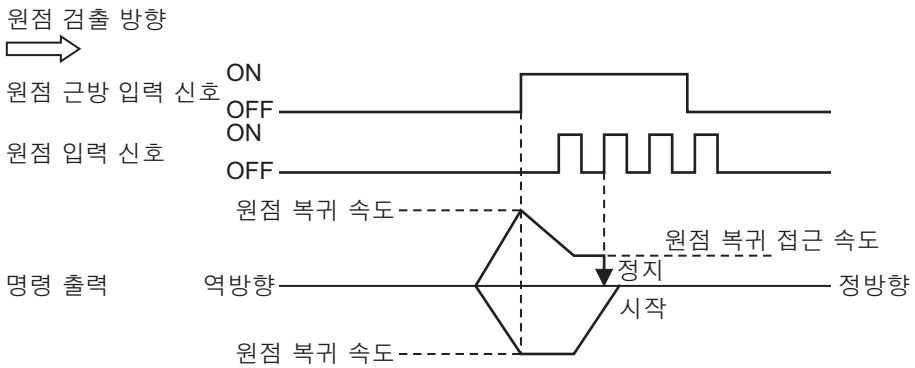


- 원점 근방 입력 신호가 'ON'인 상태에서 원점 복귀를 실행하면 원점 검출 방향과 반대 방향으로 원점 복귀 속도로 동작을 시작합니다. 원점 근방 입력 신호가 'ON'인 상태에서 일단 퇴피한 후에는 원점 검출 방향으로 원점 복귀 속도로 원점 복귀 동작을 시작합니다.
- 원점 복귀 시작 후 원점 복귀 접근 속도 이하인 채로 원점 근방 입력 신호가 'OFF' → 'ON' → 'OFF'한 후에 원점 입력 신호가 'ON'한 경우라도 기계 원점이 확정됩니다.

1: 근방 퇴피 · 원점 근방 입력 ON 지정 동작

1 **원점 복귀 속도로 기동해 원점 근방 입력 신호의 'OFF' → 'ON'에 의해 원점 복귀 접근 속도로 감속하기 시작합니다.**

2 **원점 복귀 접근 속도에 도달한 후의 최초 원점 입력에서 정지하고 기계 원점이 확정됩니다.**
일단 'ON'한 후의 원점 근방 입력 신호의 상태는 동작에 영향을 미치지 않습니다.



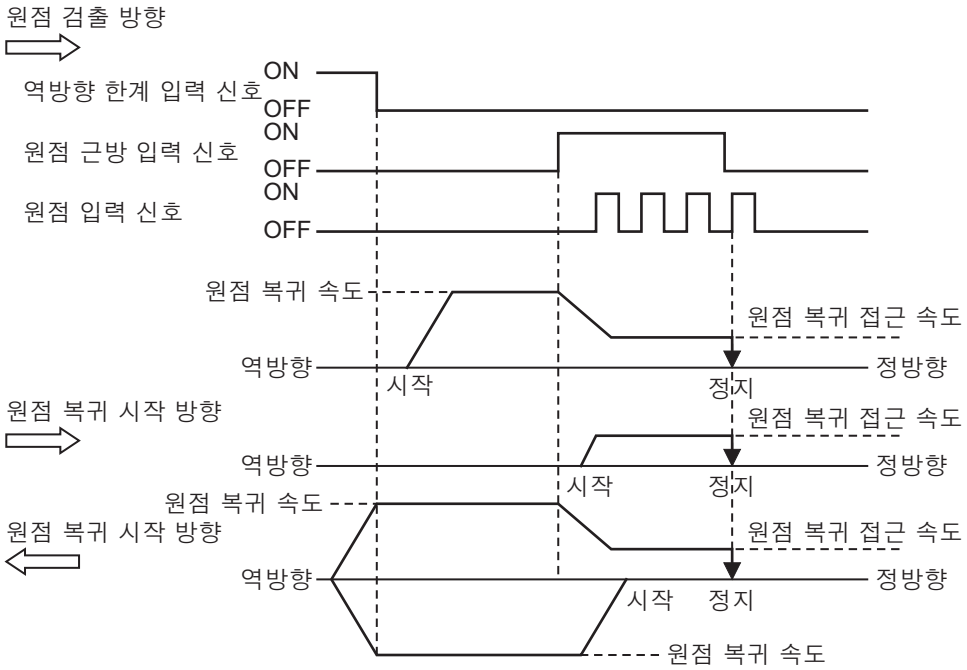
- 원점 근방 입력 신호가 'ON'인 상태에서 원점 복귀를 실행하면 원점 검출 방향과 반대 방향으로 원점 복귀 속도로 동작을 시작합니다. 원점 근방 입력 신호가 'ON'인 상태에서 일단 퇴피한 후에는 원점 검출 방향으로 원점 복귀 속도로 원점 복귀 동작을 시작합니다.

- 원점 복귀 시작 후 원점 복귀 접근 속도 이하인 채로 원점 근방 입력 신호가 'ON'한 후에 원점 입력 신호가 'ON'한 경우라도 기계 원점이 확정됩니다.

4: 원점 근방 입력 OFF 지정 동작

1 원점 복귀 속도로 기동해 원점 근방 입력 신호의 'OFF' → 'ON'에 의해 원점 복귀 접근 속도로 감속하기 시작합니다.

2 원점 복귀 접근 속도에 도달해 원점 근방 입력 신호가 'ON' → 'OFF'된 후 최초의 원점 입력에서 정지하고 기계 원점이 확정됩니다.



- 원점 근방 입력 신호가 'ON'인 상태에서 원점 복귀를 실행하면 원점 복귀 시작 방향의 설정에 따라 다음과 같이 동작합니다.

원점 복귀 시작 방향과 원점 검출 방향이 동일한 경우
퇴피 동작은 하지 않고 원점 검출 방향으로 원점 복귀 접근 속도로 원점 복귀 동작을 시작합니다.

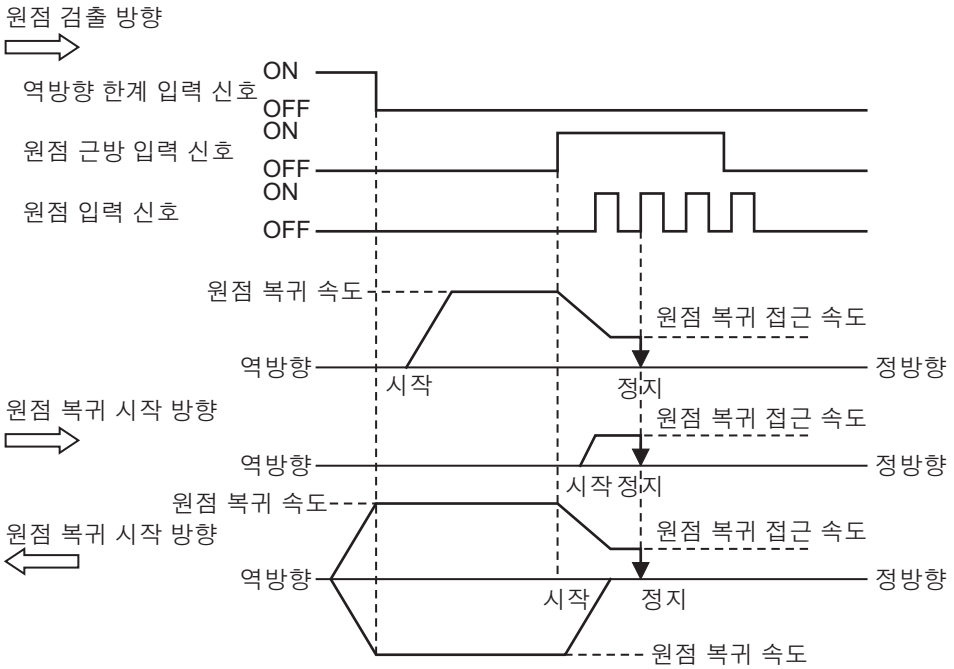
원점 복귀 시작 방향과 원점 검출 방향이 다른 경우
원점 근방 입력 신호의 상태에 상관 없이 원점 복귀 속도로 원점 복귀 시작 방향으로 동작을 시작합니다. 원점 복귀 시작 방향의 한계 입력 시 동작이 '반전'으로 설정된 경우는 한계 입력 검출 시에 반전하고 원점 검출 방향으로 원점 복귀 동작을 실행합니다.

- 원점 복귀 시작 후 원점 복귀 접근 속도 이하인 채로 원점 근방 입력 신호가 'OFF' → 'ON' → 'OFF'한 후에 원점 입력 신호가 'ON'한 경우라도 기계 원점이 확정됩니다.

5: 원점 근방 입력 ON 지정 동작

1 원점 복귀 속도로 기동해 원점 근방 입력 신호의 'OFF' → 'ON'에 의해 원점 복귀 접근 속도로 감속하기 시작합니다.

2 원점 복귀 접근 속도에 도달한 후 최초 원점 입력에서 정지하고 기계 원점이 확정됩니다.
일단 'ON'한 후의 원점 근방 입력 신호 상태는 동작에 영향을 미치지 않습니다.



- 원점 근방 입력 신호가 'ON'인 상태에서 원점 복귀를 실행하면 원점 복귀 시작 방향의 설정에 따라 다음과 같이 동작합니다.

  원점 복귀 시작 방향과 원점 검출 방향이 동일한 경우
  퇴피 동작은 하지 않고 원점 검출 방향으로 원점 복귀 접근 속도로 원점 복귀 동작을 시작합니다.

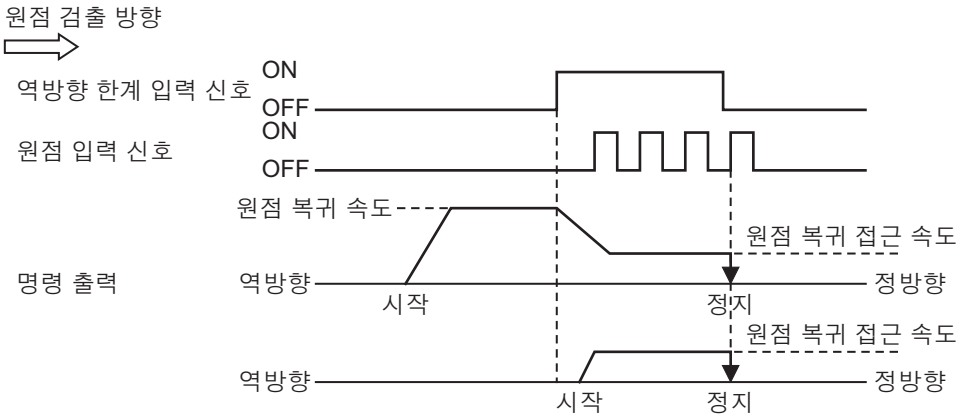
  원점 복귀 시작 방향과 원점 검출 방향이 다른 경우
  원점 근방 입력 신호의 상태에 상관 없이 원점 복귀 속도로 원점 복귀 시작 방향으로 동작을 시작합니다. 원점 복귀 시작 방향의 한계 입력 시 동작이 '반전'으로 설정된 경우는 한계 입력 검출 시에 반전하고 원점 검출 방향으로 원점 복귀 동작을 실행합니다.

- 원점 복귀 시작 후 원점 복귀 접근 속도 이하인 채로 원점 근방 입력 신호가 'OFF' → 'ON'한 후에 원점 입력 신호가 'ON'한 경우라도 기계 원점이 확정됩니다.

8: 한계 입력 OFF 지정 동작

1 **원점 복귀 속도로 기동해 원점 검출 방향과 반대 방향 한계 신호의 'OFF' → 'ON'에 의해 원점 복귀 접근 속도로 감속하기 시작합니다.**

2 **원점 복귀 접근 속도에 도달해 한계 신호가 'ON' → 'OFF'한 후 최초의 원점 입력에서 정지하고 기계 원점이 확정됩니다.**

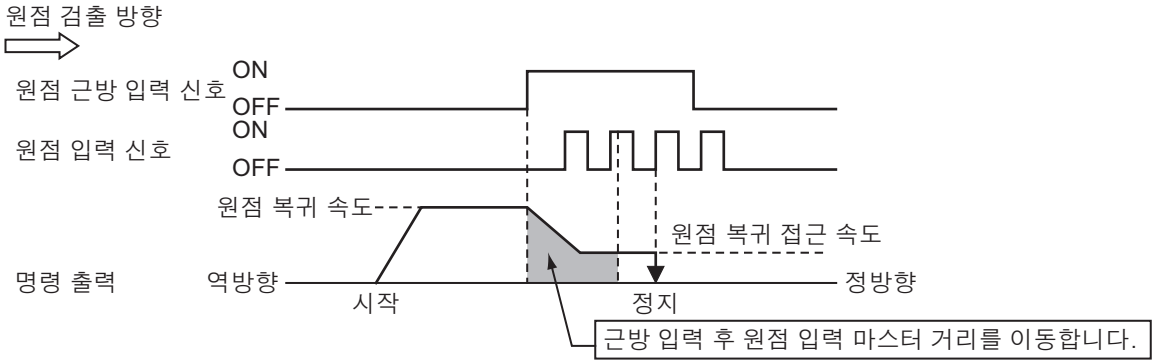


- 원점 검출 방향과 반대쪽 한계 입력이 'ON'인 상태에서 원점 복귀를 실행하면 원점 검출 방향으로 원점 복귀 접근 속도로 원점 복귀 동작을 실행합니다.
- 원점 복귀 시작 후 원점 복귀 접근 속도 이하인 채로 검출 방향과 반대 방향의 한계 신호가 'OFF' → 'ON' → 'OFF'한 후에 원점 입력 신호가 'ON'한 경우에도 기계 원점이 확정됩니다.

9: 근방 퇴피 · 원점 입력 마스크 거리 지정 동작

1 **원점 복귀 속도로 기동해 원점 근방 입력 신호의 'OFF' → 'ON'에 의해 원점 복귀 접근 속도로 감속하기 시작합니다.**

2 **원점 근방 입력 신호가 'OFF' → 'ON'한 후 원점 입력 마스크 거리를 다 이동하고 나서 최초의 원점 입력에서 정지하고 기계 원점이 확정됩니다.**
일단 'ON'한 후의 원점 근방 입력 신호 상태는 동작에 영향을 미치지 않습니다.
이 때 설정된 이동량이 짧아 원점 복귀 접근 속도로 감속하기 전에 이동량에 도달하는 경우는 원점 복귀 기동 시에 '원점 입력 마스크 거리 오류(에러 코드: 742B Hex)'가 됩니다.



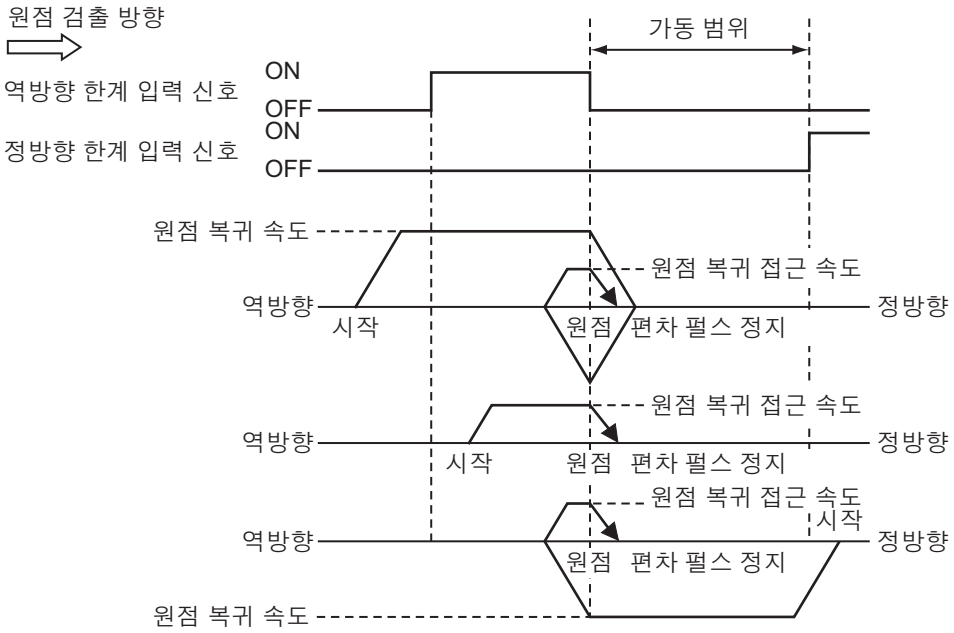
- 원점 근방 입력 신호가 'ON'인 상태에서 원점 복귀를 실행하면 원점 검출 방향과 반대 방향으로 원점 복귀 속도로 동작을 시작합니다. 원점 근방 입력 신호가 'ON'인 상태에서 일단 퇴피한 후에는 원점 검출 방향으로 원점 복귀 접근 속도로 원점 복귀 동작을 시작합니다.
- 원점 복귀 시작 후 원점 복귀 접근 속도 이하인 채로 원점 근방 입력 신호가 'ON'한 후에 원점 입력 마스크 거리를 이동하며, 원점 입력 신호가 'ON'한 경우에도 기계 원점이 확정됩니다.

11: 한계 입력만 동작

1 **원점 복귀 속도로 기동하고 원점 검출 역방향 한계 신호가 'ON' → 'OFF'가 되면 감속 정지합니다.**

2 **정지 후 원점 복귀 속도로 반전하고 한계 신호가 'OFF' → 'ON'이 되면 감속 정지합니다.**

3 **정지 후 원점 복귀 접근 속도로 반전하고 한계 신호가 'ON' → 'OFF'하면 해당 위치를 원점으로 해서 즉시 정지합니다(편차 펄스 정지). 단, 기계 원점 위치로 돌아가는 동작은 실행하지 않습니다.**



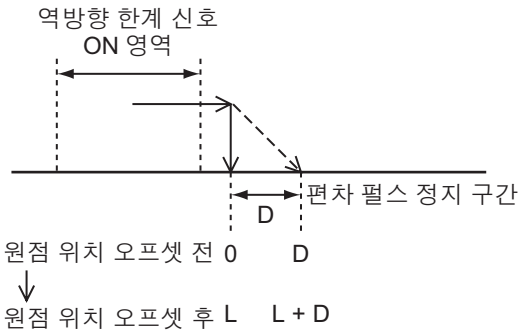
- 한계 신호만 사용해 원점 복귀를 실행하는 경우 원점 검출 방향으로 동작했을 때 한계 신호의 'ON' → 'OFF' 변화점을 원점 검출 위치로 합니다.
- 원점 검출 방향의 한계 신호가 'ON'인 상태에서 원점 복귀를 실행하면 원점 검출 역방향으로 원점 복귀 속도로 동작을 시작합니다. 원점 검출 역방향 한계 신호가 'OFF'→ 'ON'이 되면 감속 정지합니다.
- 한계 신호 검출 후 감속 정지하기 전에 한계 신호가 'OFF'해도 기계 원점은 확정되지 않고 감속을 계속합니다. 이 경우는 이상이 아닙니다.
- 반전 후나 한계 신호 입력 상태에서 기동한 후 원점 복귀 접근 속도에 도달하기 전에 한계 신호가 'ON' → 'OFF'하면 기계 원점이 확정됩니다.

원점 복귀 동작 모드 11과 다른 원점 복귀 동작 모드와의 차이

모드 11은 MC 기능 모듈 처리로 '한계 신호의 ON→OFF'를 검출하고 원점을 확정하는 원점 복귀 동작입니다. 따라서 원점 입력을 외부 래치 신호로 사용하는 원점 복귀 동작 모드와 다음과 같은 점이 다릅니다.

모드 11과 다른 원점 복귀 동작 모드의 차이점을 이해한 후 사용하시기 바랍니다.

- 한계 입력 검출에서는 서보 드라이버 등의 외부 래치 기능을 사용하지 않고 MC 기능 모듈에서 처리해 실행합니다. MC 기능 모듈 처리란 프라이머리 주기 간격에서의 처리를 가리킵니다.
  따라서 원점 확정 정확도는 원점 복귀 접근 속도나 프라이머리 정주기 태스크의 제어 주기에 따릅니다.
- 기계 원점으로 돌아가는 동작은 실행하지 않습니다. 원점 보정량이 '0'인 경우 기계 원점과는 다른 위치에서 정지한 상태로 원점 복귀 처리가 종료됩니다.
- 원점 보정량을 '0'으로 설정한 경우는 원점 보정 동작을 실행하지 않습니다. 원점 보정량이 '0'이면 전술한 대로 기계 원점과 다른 위치에서 원점 복귀 처리가 종료됩니다. 원점 보정량이 '0' 이외인 경우는 다른 원점 복귀 동작 모드와 마찬가지로 원점 보정량을 기계 원점에서부터의 상대량으로 동작합니다.
- 원점 위치 오프셋은 기계 원점 위치를 변경합니다. 다음 그림과 같이 정지 위치가 원점에서 'D'를 벗어난 경우 원점 위치 오프셋이 'L', 원점 보정량이 '0'이면 원점 복귀 처리 종료 시의 위치는 'L+D'가 됩니다.

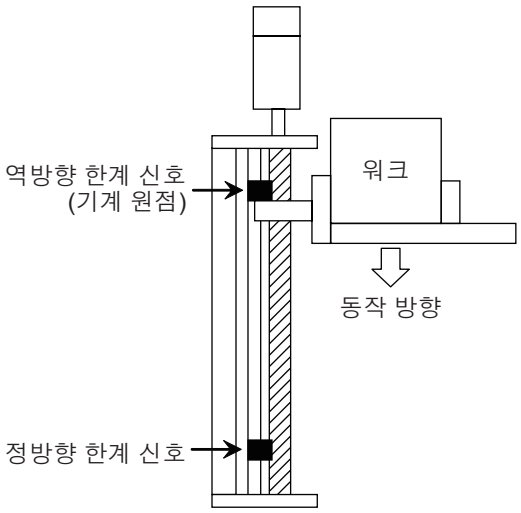


사용상의 주의 사항

- '한계 신호 ON→OFF' 검출 후에 편차 펄스를 정지하고 원점 보정 동작을 시작할 때까지의 동안은 원점 검출 방향과 역방향의 한계 신호를 무시합니다.
- 원점 보정량이 '0'이며 원점을 확정한 직후에 원점 검출 역방향의 한계 입력 신호가 'ON'하는 경우는 한계 신호 입력에서 벗어나는 방향으로 원점 보정량을 설정하십시오.

원점 복귀 동작 모드 11의 사용 예

다음 그림과 같이 '역방향 한계 신호'와 '원점 신호'를 둘 다 설치할 공간이 없으면 '역방향 한계 신호'에 한계 신호와 기계 원점의 역할을 모두 부여해서 사용할 수 있습니다.



12 : 근방 퇴피 · 유지 시간 지정 동작

1 원점 복귀 속도로 기동해 원점 근방 입력 신호의 'OFF' → 'ON'에 의해 원점 복귀 접근 속도로 감속하기 시작합니다.

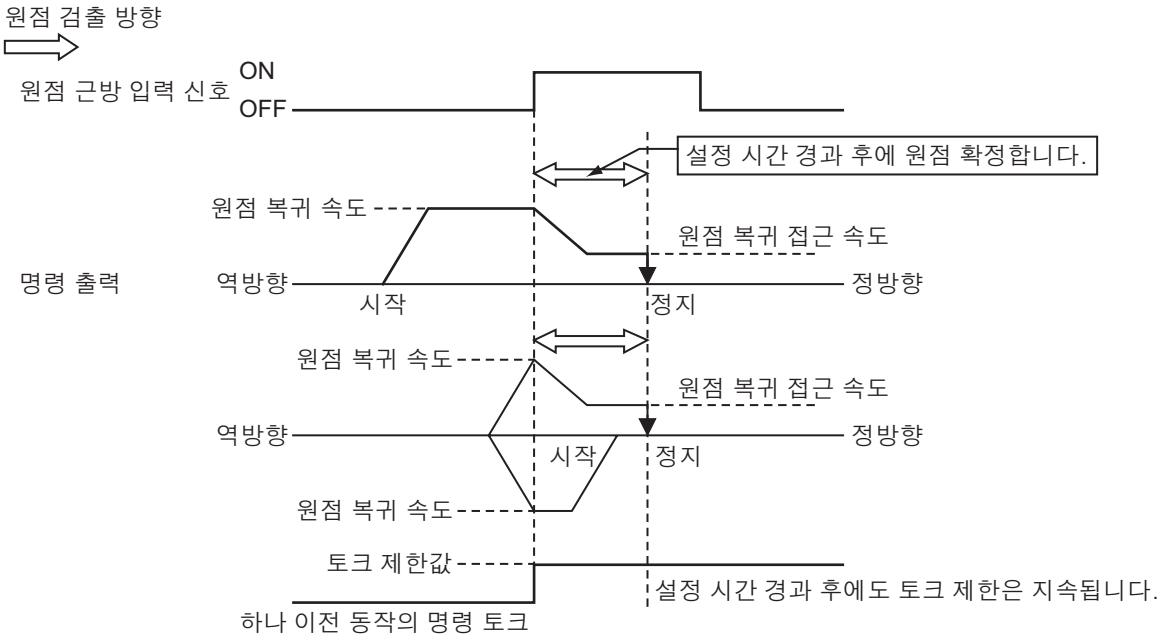
감속 시작과 동시에 시간을 감시하기 시작합니다.
유지 시작 시의 토크 제한은 오므론의 서보 드라이버 G5 시리즈와 그 이외의 서보에서 다음과 같이 다르게 사용됩니다.

G5 서보 : 자동으로 토크 제한을 실행합니다.
G5 서보 이외 : 설정에 따라 토크 제한을 실행합니다.

유지 동작에 대해서는, 🕮 '유지 동작(오므론의 서보 드라이버 G5 시리즈를 사용하는 경우) (3-31 페이지)'을 참조하십시오.

2 설정 시간 경과 후 기계 원점이 확정됩니다.

일단 'ON'한 후의 원점 근방 입력 신호 상태는 동작에 영향을 미치지 않습니다.



- 원점 근방 입력 신호가 'ON'인 상태에서 원점 복귀를 실행하면 원점 검출 방향과 반대 방향으로 원점 복귀 속도로 동작을 시작합니다. 원점 근방 입력 신호가 'ON'인 상태에서 일단 퇴피한 후에는 원점 검출 방향으로 원점 복귀 접근 속도로 원점 복귀 동작을 시작합니다.
- 토크 제한 해제에 대해서도 오므론의 서보 드라이버 G5 시리즈와 그 이외의 서보에서는 다르게 사용됩니다.

  G5 서보 : 원점 복귀 후 원점 복귀와 반대 방향으로 처음 움직일 때 자동으로 'OFF' 됩니다.
  G5 서보 이외 : 토크 제한 사용 시 원점 복귀 후 처음으로 원점 복귀와 반대 방향으로 이동하는 경우 토크 제한을 해제하십시오. 토크 제한을 재기입할 때는 EC_CoESDOWrite(CoE SDO 쓰기) 명령을 사용하십시오.

- 원점 근방 입력 신호 검출 후 원점 복귀 접근 속도에 도달하기 전에 유지 시간이 경과한 경우에도 이상이 발생하지 않고 기계 원점이 확정됩니다.
- 원점 복귀 시작 후 원점 복귀 접근 속도 이하인 채로 원점 근방 입력 신호가 'OFF' → 'ON' 한 후에 유지 시간이 경과해도 기계 원점이 확정됩니다.

13 : 원점 근방 입력 없음 · 유지 원점 입력 지정 동작

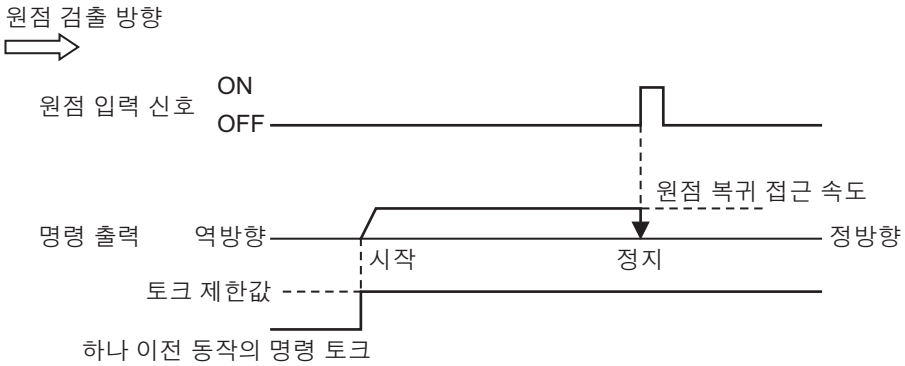
1 원점 복귀 접근 속도로 기동합니다.

유지 시작 시의 토크 제한은 오므론의 서보 드라이버 G5 시리즈와 그 이외의 서보에서 다음과 같이 다르게 사용됩니다.

G5 서보 : 자동으로 토크 제한을 실행합니다.
G5 서보 이외 : 필요에 따라 토크 제한을 실행합니다.

유지 동작에 대해서는, 🕮 '유지 동작(오므론의 서보 드라이버 G5 시리즈를 사용하는 경우) (3-31 페이지)'을 참조하십시오.

2 원점 입력 'OFF' → 'ON'에 의해 기계 원점이 확정됩니다.



- 토크 제한 해제에 대해서도 오므론의 서보 드라이버 G5 시리즈와 그 이외의 서보에서는 다르게 사용됩니다.

  G5 서보 : 원점 복귀 후 원점 복귀와 반대 방향으로 처음 움직일 때 자동으로 'OFF'됩니다.
  G5 서보 이외 : 토크 제한 사용 시 원점 복귀 후 처음으로 원점 복귀와 반대 방향으로 이동하는 경우 토크 제한을 해제하십시오. 토크 제한을 재기입할 때는 EC_CoESDOWrite(CoE SDO 쓰기) 명령을 사용하십시오.

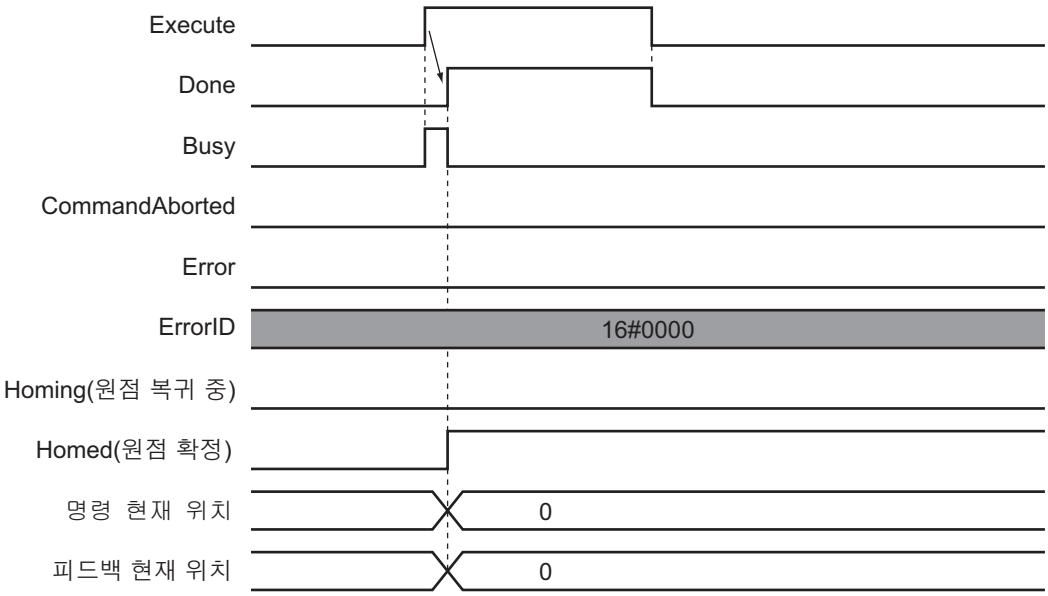
- 원점 복귀 시작 후 원점 복귀 접근 속도에 도달하기 전에 원점 입력 신호가 'ON'한 경우에도 기계 원점이 확정됩니다.

사용상의 주의 사항

오므론의 G5 시리즈 서보 드라이버 리니어 모터 타입은 Z상 입력을 PDO에 매핑할 수 없습니다. 따라서 PDO에 매핑한 Z상 입력을 사용하는 원점 복귀 동작의 '13: 원점 근방 입력 없음 · 유지 원점 입력 지정'을 사용할 때는 원점 입력 신호에 Z상 입력을 선택하지 마십시오.

14 : 원점 사전 설정 동작

명령 현재 위치와 피드백 위치를 강제로 원점 위치 오프셋으로 변경하고 기계 원점을 확정합니다.
원점 위치 오프셋을 '0'으로 설정하면 다음과 같이 동작합니다.

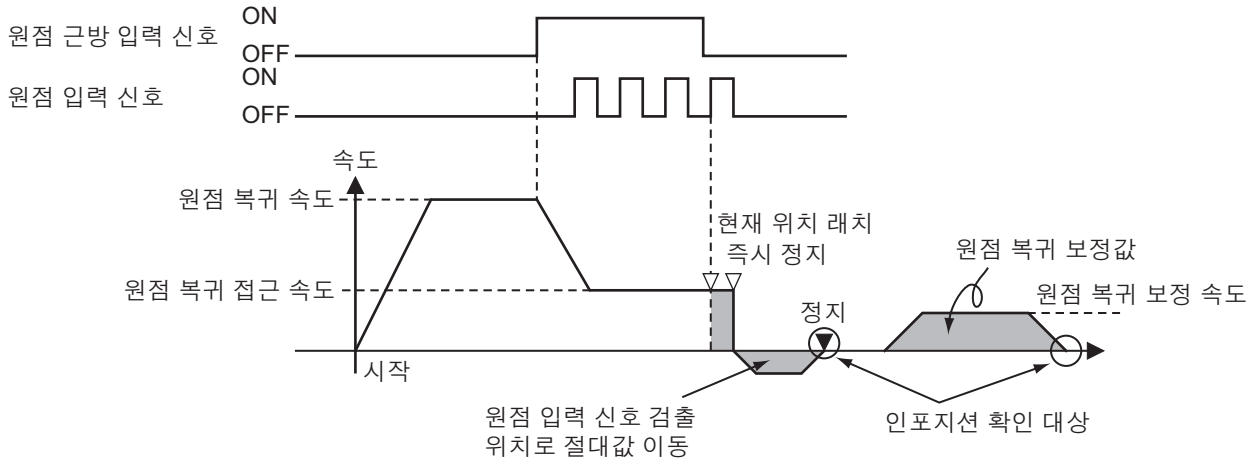


● 원점 확정 시의 정지 방법과 인포지션 확인

원점 확정 시에는 다음과 같은 순서로 정지합니다.

1 **원점 입력 신호 검출 시의 현재 위치를 래치하고 즉시 정지한다.**

2 **즉시 정지 후 1에서 래치한 위치에 절대 이동으로 반전 이동한다.**

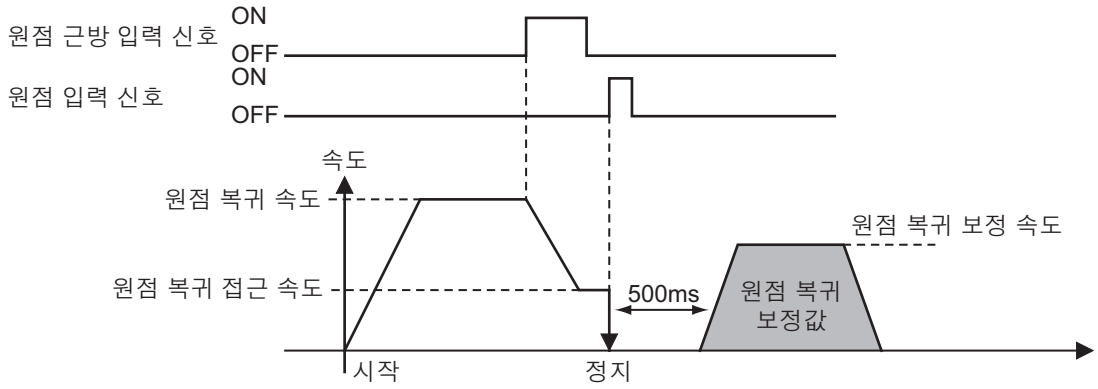


- 이 명령에서는 축 파라메터 '인포지션 확인 시간' 설정에 상관 없이 항상 위치 결정 완료 대기를 실행합니다.
  특히 원점 입력 신호 검출 위치로 절대 이동하는 경우 및 인포지션 확인 시간을 '0'으로 설정한 경우에는 위치 결정이 완료될 때까지 계속 대기하므로 인포지션 폭에 확실하게 들어가도록 설정하십시오.
  원점 복귀 보정값 동작에서는 인포지션 확인 시간을 '0'으로 해도 인포지션 대기를 실행하지 않습니다.
  또한 인포지션 대기 상태인지의 여부는 모션 제어 시스템 변수 'InPosWaiting(인포지션 대기)'에서 확인할 수 있습니다.

● 원점 보정

원점 복귀 보정값을 설정하면 원점 입력 검출 후 원점 복귀 보정값만큼 이동해 원점을 확정합니다.
기계 계통에서 기계 원점이 확정된 뒤에 워크를 조정하기는 어렵기 때문에 미세 조정이 필요한 경우에 최초로 확정한 기계 원점으로부터 어긋난 양을 보정 데이터로 설정합니다.
이 때의 이동 속도는 원점 복귀 보정 속도가 되며, 원점 입력 검출에 의해 정지한 후 500ms가 경과하고 나서 이동하기 시작합니다.

원점 복귀 보정값의 부호는 좌표축에 대한 방향을 나타내며, 원점 복귀 보정값이 음수일 때는 역방향으로 동작합니다.



● 오버라이드

이 명령에 대해 오버라이드는 무효가 됩니다.

● 토크 제한 자동 제어

오므론의 서보 드라이버 G5 시리즈 사용 시에 원점 복귀 동작에서 '12: 근방 퇴피 · 유지 시간 지정', '13: 원점 근방 입력 없음 · 유지 원점 입력 지정' 중 하나를 선택하면 유지 방향으로 토크 제한이 자동으로 시작됩니다.
원점 복귀 시의 토크 제한 자동 제어는 원점 복귀 후에도 토크 제한 상태를 유지하며, 원점 검출 방향과 역방향으로 동작할 때 토크 제한을 무효로 합니다.
또한 다음 경우에도 토크 제한이 자동 해제됩니다.

- 서보 OFF(서보 잠금 해제)
- 주기적 동기 위치 제어(CSP)에서 다른 제어 모드로 전환한 경우

원점 복귀 동작 중에는 원점 검출 방향과는 반대 방향으로 동작할 때 토크 제한이 해제됩니다.
예를 들면 한계 입력 시의 반전 동작 방향이 원점 검출 방향과 반대 방향이면 반전 완료 시에 토크 제한이 해제됩니다.
또한 반전해서 동작 방향이 원점 검출 방향이 되면 다시 토크 제한을 유효로 합니다.

● 유지 동작(오므론의 서보 드라이버 G5 시리즈를 사용하는 경우)

원점 복귀의 [12: 근방 퇴피 · 유지 시간 지정], [13: 원점 근방 입력 없음 · 유지 원점 입력 지정]은 서보 드라이버에 미리 설정된 토크 제한값을 이용해 원점 검출 방향으로 토크 제한을 자동으로 시작합니다.

사용상의 주의 사항

타사의 서보 드라이버를 사용하면 MC_Home(원점 복귀) 명령에서의 자동 토크 제한 기능은 기능하지 않습니다.
MC_SetTorqueLimit(토크 제한) 명령, SDO 통신, 지원 툴 등을 사용해 적절한 값을 설정하십시오.

참고

- 원점 복귀가 정상 완료된 후에도 토크 제한은 계속됩니다.
- 역방향으로 이동하는 명령이 기동되면 토크 제한이 자동으로 해제됩니다.

오므론의 서보 드라이버 G5 시리즈 설정

유지 동작을 사용하려면 서보 드라이버의 지원 소프트웨어를 사용해 G5 서보 드라이버의 토크 리미트 선택(3521 Hex)을 설정해야 합니다.

- 원점 복귀 유지 동작 시에 원점 검출 방향에 대해 토크를 제한하고, 그 밖의 동작은 MC_SetTorqueLimit (토크 제한) 명령의 토크 제한 방향이나 토크 제한값을 사용하는 경우는 '6'을 사용하십시오.
  이 경우 원점 복귀 유지 동작 중에는 MC_SetTorqueLimit(토크 제한) 명령의 입력 변수 내용은 무시됩니다.
- '4'로 설정하면 항상 MC_SetTorqueLimit(토크 제한) 명령의 입력 변수 내용이 적용됩니다. 원점 복귀 유지 동작 시와 그 밖의 경우에 따라 적절한 토크 제한값을 설정해야 합니다.

		토크 리미트 선택(3521 Hex)	
		6(권장)	4
정방향 토크 제한값	원점 복귀 중	제3 토크 리미트(3525 Hex)가 적용됩니다.	MC_SetTorqueLimit(토크 제한) 명령의 PositiveValue(정방향 토크 제한값)와 제1 토크 리미트(3013 Hex) 중 작은 쪽이 적용됩니다.
	원점 복귀 이외	MC_SetTorqueLimit(토크 제한) 명령의 PositiveValue(정방향 토크 제한값)와 제1 토크 리미트(3013 Hex) 중 작은 쪽이 적용됩니다.	
역방향 토크 제한값	원점 복귀 중	제4 토크 리미트(3526 Hex)가 적용됩니다.	MC_SetTorqueLimit(토크 제한) 명령의 NegativeValue(역방향 토크 제한값)와 제2 토크 리미트(3522 Hex) 중 작은 쪽이 적용됩니다.
	원점 복귀 이외	MC_SetTorqueLimit(토크 제한) 명령의 NegativeValue(역방향 토크 제한값)와 제2 토크 리미트(3522 Hex) 중 작은 쪽이 적용됩니다.	

토크 제한에 대해서는 'MC_SetTorqueLimit(3-311페이지)'를 참조하십시오.
또한 G5 서보 드라이버의 설정에 대해서는 "G5 시리즈 AC 서보 모터/드라이버 사용 설명서 (EtherCAT 통신 내장 타입) (SBCE-365)", "G5 시리즈 리니어 모터/드라이버 사용 설명서(EtherCAT 통신 내장 리니어 모터 타입) (SBCE-366)"를 참조하십시오.

● 유지 동작 중인 위치 편차 감시

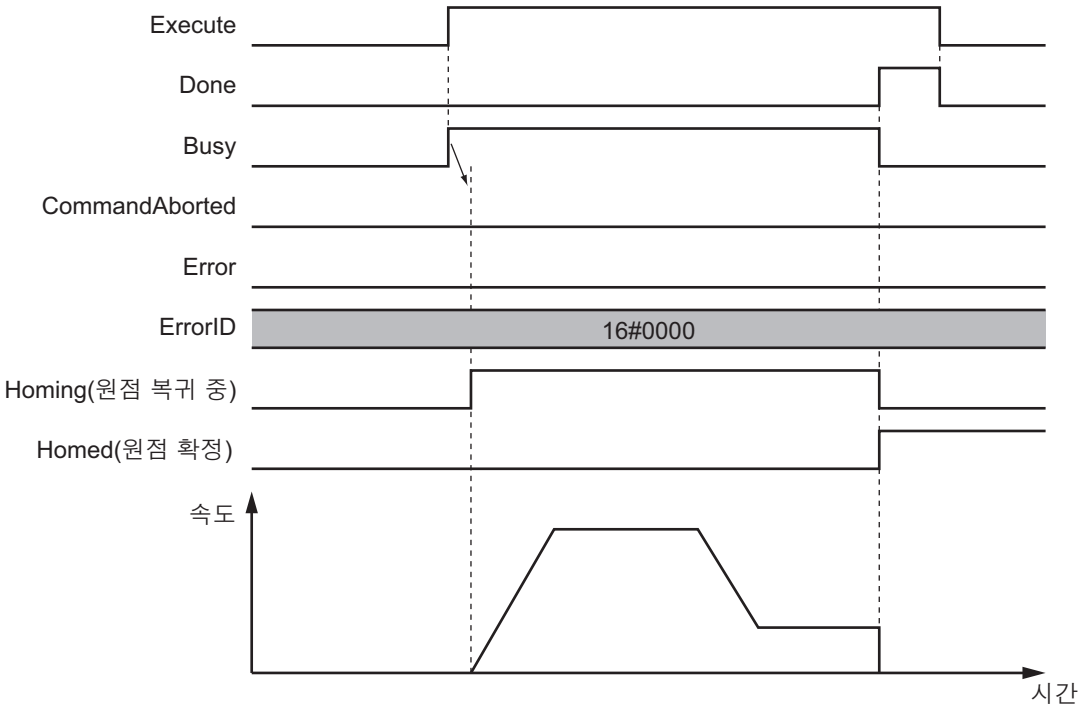
원점 복귀 유지 동작 중에는 위치 편차 감시 기능이 동작하지 않습니다.

위치 편차 감시 기능에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

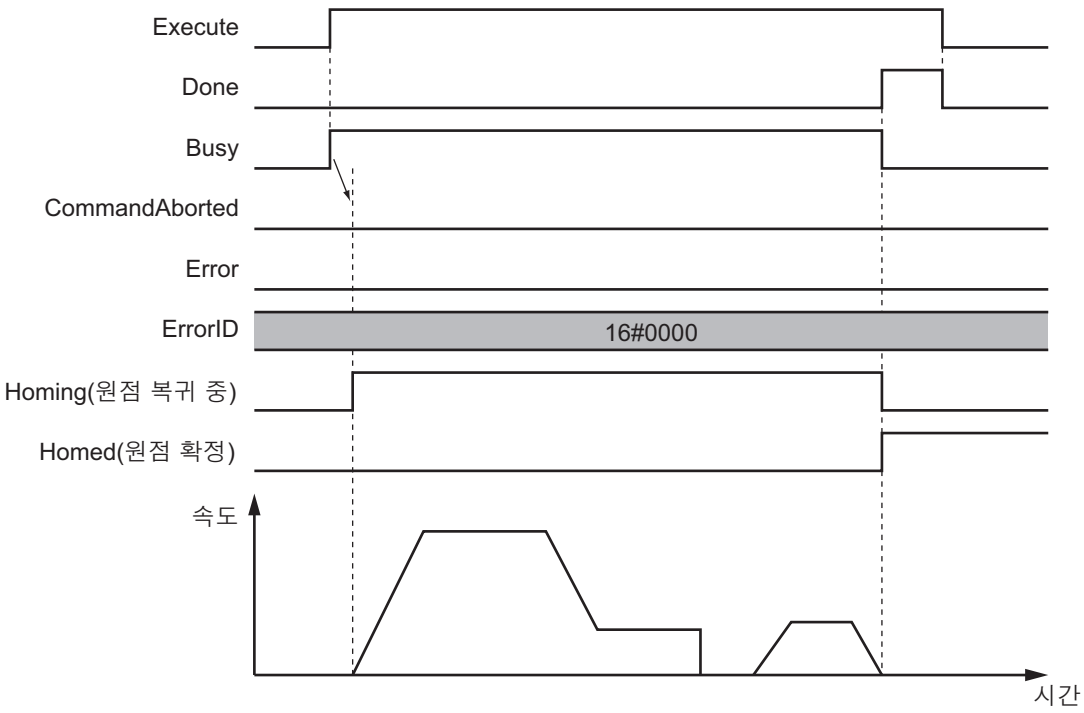
타이밍 차트

원점 복귀를 실행한 경우의 타이밍 차트는 다음과 같습니다.

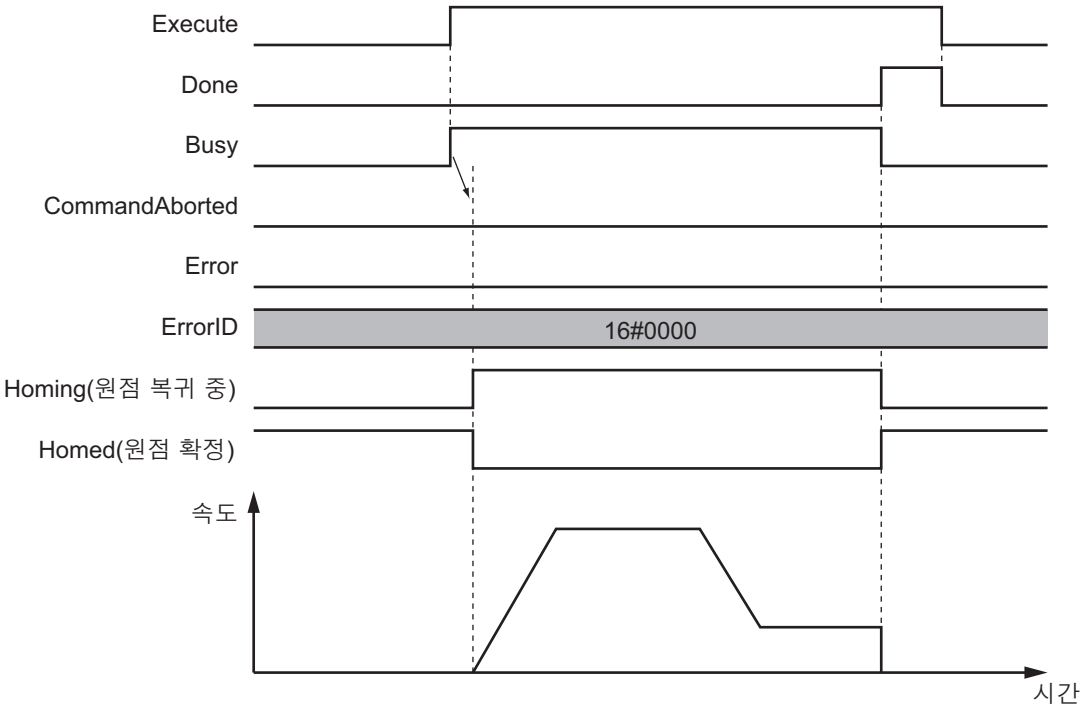
● 원점 보정이 없는 경우



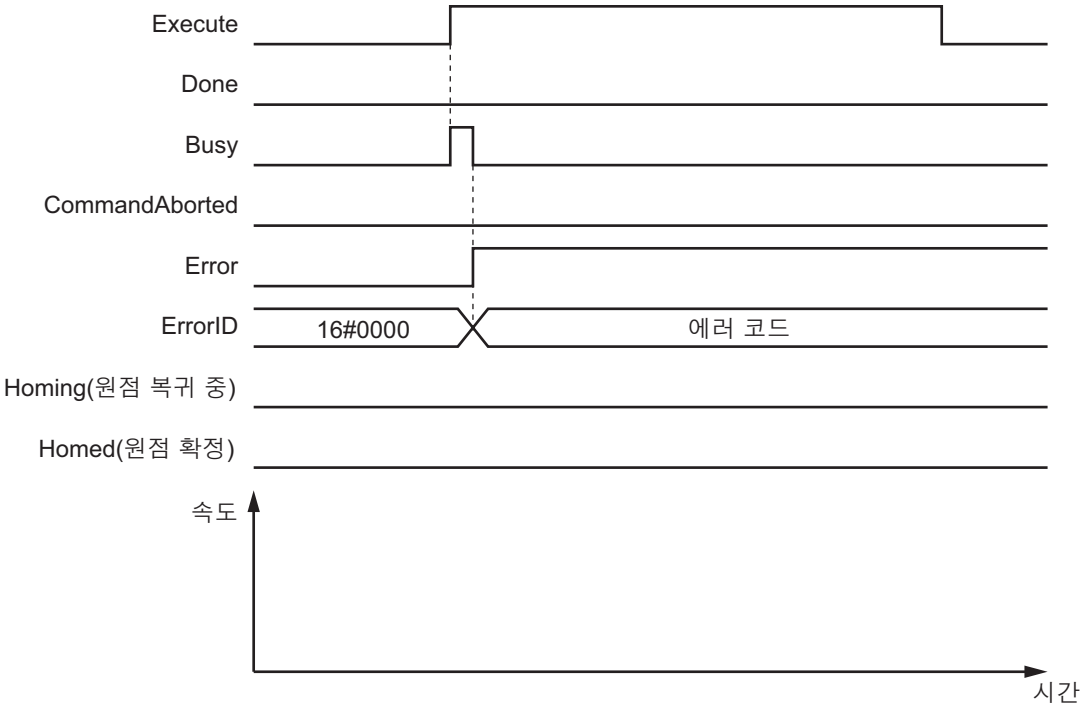
● 원점 보정이 있는 경우



● **원점 확정 상태에서 이 명령을 실행한 경우**



● **파라메터가 잘못된 상태에서 이 명령을 실행한 경우 및 모션 명령 실행이 불가능한 상태에서 이 명령을 실행한 경우**



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_HomeWithParameter

원점 복귀 파라메터를 지정하여 모터를 작동시켜 한계 신호, 원점 근방 신호, 원점 신호를 이용해서 기계 원점을 결정하는 기능

명령	이름	FB/FUN	그래픽 표현	ST 표현
MC_HomeWithParameter	파라메터 지정 원점 복귀	FB	MC_HomeWithParameter_instance MC_HomeWithParameter Axis — — Axis HomingParameter — — HomingParameter Execute — Done Busy CommandAborted Error ErrorID	MC_HomeWithParameter_instance (Axis := 《파라메터》 , HomingParameter := 《파라메터》 , Execute := 《파라메터》 , Done => 《파라메터》 , Busy => 《파라메터》, CommandAborted => 《파라메터》 , Error => 《파라메터》 , ErrorID => 《파라메터》);

버전 관련 정보

이 명령은 CPU 유니트 Ver1.03 이상과 Sysmac StudioVer1.04 이상의 조합에서 사용할 수 있습니다.

변수

입력 변수

입력 변수	이름	데이터 형식	유효 범위	초기값	내용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이름	데이터 형식	유효 범위	내용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이름	데이터 형식	유효 범위	내용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1
HomingParameter	원점 복귀 파라메터	_sHOMING_REF	–	원점 복귀 파라메터를 설정합니다. *2

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.
*2. _sHOMING_REF 형식의 사용자 정의 변수를 정의하십시오.

● _sHOMING_REF

멤버 변수	이름	데이터 형식	유효 범위	기능
Homing Mode	원점 복귀 동작	_eMC_HOMING_MODE	0: _mcHomeSwTurnHomeSwOff 1: _mcHomeSwTurnHomeSwOn 4:_mcHomeSwOff 5:_mcHomeSwOn 8:_mcLimitInputOff 9:_mcHomeSwTurnHomeMask 11:_mcLimitInputOnly 12:_mcHomeSwTurnHoldingTime 13:_mcNoHomeSwHoldingHomeInput 14:_mcHomePreset	재기입할 수 있는 원점 복귀 동작을 지정합니다. 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 마스크 거리 지정 11: 한계 입력만 12: 근방 퇴피 · 유지 시간 지정 13: 원점 근방 입력 없음 · 유지 원점 입력 지정 14: 원점 사전 설정
HomeInput	원점 입력 신호	_eMC_HOME_INPUT	0:_mcZPhase 1:_mcExternalSignal	원점 입력 신호를 선택합니다. 0: Z상 입력을 사용 1: 외부 원점 입력을 사용*1
StartDir	원점 복귀 시작 방향	_eMC_DIRECTION	0:_mcPositiveDirection 2:_mcNegativeDirection	원점 복귀 기동 시의 시작 방향을 설정합니다. 0: 정방향 2: 역방향
HomeDir	원점 검출 방향	_eMC_DIRECTION	0:_mcPositiveDirection 2:_mcNegativeDirection	원점 복귀의 원점 입력 검출 방향을 설정합니다. 0: 정방향 2: 역방향

멤버 변수	이름	데이터 형식	유효 범위	기능
PosiLmt Mode	정방향 한계 입력 시 동작	_eMC_ LIMIT_ REVERSE_ MODE	0:_mcErrorStop 1:_mcRevImmediateStop 2:_mcRevDeceleration Stop	원점 복귀 중에 양방향 한계 입력이 'ON'한 경우의 정지 방법을 설정합니다. 0: 반전 안 함·축 이상으로 정지(정지 방법은 한계 입력 정지 방법에 따름) 1: 반전 · 즉시 정지 2: 반전 · 감속 정지
NegaLmt Mode	역방향 한계 입력 시 동작	_eMC_ LIMIT_ REVERSE_ MODE	0:_mcErrorStop 1:_mcRevImmediateStop 2:_mcRevDeceleration Stop	원점 복귀 중에 역방향 한계 입력이 'ON'한 경우의 정지 방법을 설정합니다 0: 반전 안 함·축 이상으로 정지(정지 방법은 한계 입력 정지 방법에 따름) 1: 반전 · 즉시 정지 2: 반전 · 감속 정지
Vel	원점 복귀 속도	LREAL	양수	원점 복귀 중의 속도를 설정합니다. 최고 속도 이하, 원점 복귀 접근 속도 · 원점 복귀 속도가 되도록 설정하십시오. 단위는 [명령 단위/s]입니다.*2
ApproachVel	원점 복귀 접근 속도	LREAL	양수	원점 근방 입력이 'ON'한 후의 속도를 설정합니다 최고 속도 이하, 원점 복귀 접근 속도 · 원점 복귀 속도가 되도록 설정하십시오. 단위는 [명령 단위/s]입니다. *2
Acc	원점 복귀 가속도	LREAL	양수 또는 '0'	원점 복귀 시의 가속도를 지정합니다. '0'을 설정하면 가속 없음으로 설정됩니다. 단위는 [명령 단위/s2]입니다. *2
Dec	원점 복귀 감속도	LREAL	양수 또는 '0'	원점 복귀 시의 감속도를 지정합니다. '0'을 설정하면 감속 없음으로 설정됩니다. 단위는 [명령 단위/s2]입니다. *2
Jerk	원점 복귀 저크	LREAL	양수 또는 '0'	원점 복귀 시의 저크를 지정합니다. '0'을 설정하면 저크 없음으로 설정됩니다 펄스 단위 40비트 이내로 설정하십시오. 단위는 [명령 단위/s3]입니다. *2
Mask	원점 입력 마스크 거리	LREAL	양수 또는 '0'	원점 복귀 동작 모드가 근방 퇴피 · 원점 입력 마스크 거리 지정인 경우의 원점 입력 마스크 거리를 설정합니다. 단위는 [명령 단위]입니다. *2
Offset	원점 위치 오프셋	LREAL	음수, 양수, '0'	원점 복귀 완료 시 설정한 값으로 현재 위치를 사전 설정합니다. 로터리 모드 시, 링 카운터 하한값≦원점 위치 오프셋＜링 카운터 상한값이 되도록 설정하십시오. 또한 펄스 단위 40비트 이내로 설정하십시오. 단위는 [명령 단위]입니다. *2
PushTime	원점 복귀 유지 시간	UINT	0~10,000	원점 복귀 동작 모드가 근방 퇴피 · 유지 시간 지정인 경우의 유지 시간을 밀리초 단위로 설정합니다. 단위는 [ms]입니다.

멤버 변수	이름	데이터 형식	유효 범위	기능
Compensation	원점 복귀 보정값	LREAL	음수, 양수, '0'	원점 복귀의 원점 확정 후의 원점 복귀 보정량을 설정합니다. 로터리 모드 시, | 원점 복귀 보정값 | < | 링 카운터 상한값 – 링 카운터 하한값 | 이 되도록 설정하십시오. 또한 펄스 단위 40비트 이내로 설정하십시오. 단위는 [명령 단위]입니다. *2
Compensation Vel	원점 복귀 보정 속도	LREAL	양수	원점 복귀 보정 시의 속도를 설정합니다. 최고 속도 이하가 되도록 설정하십시오. 단위는 [명령 단위/s]입니다. *2

*1. 오므론의 G5 시리즈 서보 드라이버를 사용하는 경우에 지정할 수 있습니다. G5 시리즈 서보 드라이버의 출하 시 설정에서는 래치 기능 1에 외부 원점 입력이 할당되어 있습니다. 래치 기능 1은 서보 드라이버의 서보 파라메터 개체를 사용하여 할당 내용을 변경할 수 있습니다. 자세한 내용은 📖 "G5 시리즈 AC 서보 모터/드라이버(EtherCAT 통신 내장 타입) 사용 설명서(SBCE-365)" 및 📖 "G5 시리즈 AC 서보 모터/드라이버(EtherCAT 통신 내장 리니어 모터 타입) 사용 설명서(SBCE-366)"를 참조하십시오.

*2. 명령 단위에 대해서는 📖 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

기능 설명

- Axis(축)에서 지정된 축에 대해 Execute(기동)가 기동하면 원점 복귀 동작을 시작합니다.
- 파라메터 지정 원점 복귀 명령에서 사용되는 각종 파라메터는 입출력 변수 HomingParameter로 설정합니다.
- 이 명령을 실행해서 축 파라메터가 변경되지는 않습니다.
- 이 명령이 MC_Home(원점 복귀) 명령과 다른 것은 각종 파라메터 설정 방법뿐입니다. 이 명령은 각종 파라메터를 입출력 변수 HomingParameter로 설정합니다. MC_Home(원점 복귀) 명령은 각종 파라메터를 Sysmac Studio의 축 파라메터 [원점 복귀 동작]로 설정합니다.

각종 파라메터 설정 방법 이외의 명령 사양은 📖 'MC_Home(3-15페이지)'을 참조하십시오.

사용상의 주의 사항

이 명령에서 지정한 원점 복귀 파라메터는 이 명령 실행 시에만 유효합니다. 이 명령 실행 후 MC_Home(원점 복귀) 명령을 실행한 경우는 축 파라메터로 설정된 원점 복귀 파라메터에 따라 동작합니다.

원점 복귀 동작과 원점 복귀 파라메터의 관계

원점 복귀 파라메터에는 원점 복귀 동작의 설정에 따라 사용하지 않는 것이 있습니다. 사용하지 않는 파라메터는 범위 확인이나 정합 확인 대상에서 제외됩니다. 범위 확인이나 정합 확인은 이 명령 실행 시 실행됩니다.

원점 복귀 동작의 설정에 대해 원점 복귀 파라메터 사용 여부는 다음 표와 같습니다.

(○: 파라메터 사용, ×: 파라메터 사용 안 함)

	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 유지 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 · 유지 시간 지정	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○
원점 근방 입력 없음 · 유지 원점 입력 지정	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○
원점 사전 설정	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×

참고

NX 시리즈 위치 인터페이스 유니트를 사용하는 경우, 유지 기능을 사용하는 원점 복귀 동작 모드를 선택할 수 없습니다. 자세한 내용은 "NX 시리즈 위치 인터페이스 유니트 사용 설명서 (SBCE-374)"를 참조하십시오.

모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동을 수행한 경우 '모션 명령 재기동 불가능(에러 코드: 543B Hex)'이 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_Move

절대값 위치 결정 또는 상대값 위치 결정을 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_Move	위치 결정	FB	MC_Move_instance MC_Move Axis — — Axis Execute — Done Position — Busy Velocity — Active Acceleration — Command Aborted Deceleration — Error Jerk — ErrorID Direction BufferMode MoveMode	MC_Move_instance (Axis := 《파라메터》, Execute := 《파라메터》, Position := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, Direction := 《파라메터》, BufferMode := 《파라메터》, MoveMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Position	목표 위치	LREAL	음수, 양수, '0'	0	이동 방법 선택 시에 절대 위치 결정을 지정한 경우는 절대 좌표의 목표 위치를 지정합니다. 이동 방법 선택 시에 상대 위치 결정을 지정한 경우는 상대 위치를 지정합니다. 단위는 [명령 단위]입니다. *1
Velocity	목표 속도	LREAL	양수	0	목표 속도를 지정합니다. *2 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Direction	방향 선택	_eMC_ DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	0 *3	MoveMode가 [0: 절대값 위치 결정] *4 이고 카운트 모드가 [로터리 모드]인 경우에 회전 방향을 지정합니다. 0 : 정방향 지정 1 : 근거리 회전 지정 2 : 역방향 지정 3 : 현재 방향 지정 4 : 방향 지정 없음
BufferMode	버퍼 모드 선택	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered 2 : BlendingLow 3 : BlendingPrevious 4 : BlendingNext 5 : BlendingHigh
MoveMode	이동 방법 선택	_eMC_MOVE_ MODE	0: _mcAbsolute 1: _mcRelative	0 *3	이동 방법을 선택합니다. 0 : 절대값 위치 결정 1 : 상대값 위치 결정

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 반드시 목표 속도를 설정하십시오. 설정하지 않고 동작하게 하면 목표 속도 설정 범위 외(에러 코드: 5422Hex)가 발생합니다.
*3. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*4. MoveMode가 [1: 상대값 위치 결정]일 때는 position 부호로 이동 방향이 결정됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	위치 결정이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동 (Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 절대값 위치 결정 또는 상대값 위치 결정을 실행할 수 있습니다.
- MoveMode(이동 방법 선택)에서 '절대값 위치 결정'을 지정하면 MC_MoveAbsolute(절대값 위치 결정) 명령과 동일하게 동작합니다. 마찬가지로 '상대값 위치 결정'을 지정하면 MC_MoveRelative(상대값 위치 결정) 명령과 동일하게 동작합니다.
- 상대값 위치 결정 명령으로 동작하는 경우는 Direction(방향 선택)을 사용하지 않습니다.

자세한 내용은 🕮 'MC_MoveAbsolute(3-47페이지)' 또는 🕮 'MC_MoveRelative(3-73페이지)'를 참조하십시오.

명령에 대한 자세한 내용

이 명령에 대해 자세히 설명합니다.

사용상의 주의 사항

절대값 위치 결정을 실행하려면 축 파라메터의 [링 카운터 상한 설정값] 및 [링 카운터 하한 설정값] 범위 이내로 목표 위치를 지정하십시오.
목표 위치를 [링 카운터 상한값 설정]과 [링 카운터 하한값 설정] 범위 밖으로 지정하면 '목표 위치 설정 범위 외(에러 코드: 5478Hex)'가 됩니다.

● 인포지션 확인

이 명령의 인포지션 확인은 축 파라메터의 [인포지션 폭], [인포지션 확인 시간] 설정에 따라 실행됩니다.

모션 명령 재기동

위치 결정 동작 중에 입력 변수를 변경하고 Execute(기동)를 다시 TRUE로 하면 이 명령의 움직임을 변경할 수 있습니다. 모션 명령 재기동으로 변경 가능한 입력 변수는 Position(목표 위치), Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)입니다.

모션 명령 재기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다. 각 축에 1개까지 버퍼링할 수 있습니다.
다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택		설 명
Aborting		현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.
Buffered		현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.
Blending		현재 실행 중인 명령이 목표 위치에 도달했을 때의 속도(중계 속도)를 기동 속도로 버퍼링 중인 이 명령을 연속해서 동작하게 합니다. 중계 속도로 목표 위치에 도달하도록 현재 실행 중인 명령의 동작을 변경합니다. 중계 속도 지정 방법은 다음의 4종류가 있습니다.
	BlendingLow	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 낮은 것을 중계 속도로 합니다.
	BlendingPrevious	현재 실행 중인 목표 속도를 중계 속도로 합니다.
	BlendingNext	버퍼링되고 있는 이 명령의 목표 속도를 중계 속도로 합니다.
	BlendingHigh	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 높은 것을 중계 속도로 합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

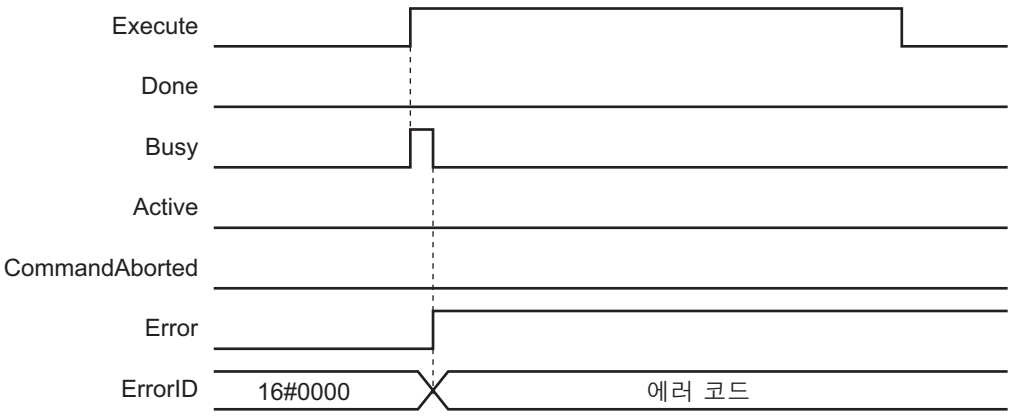
● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하는 경우 Aborting, Buffered, Blending이 가능합니다.

이상

● 이상 발생 시의 타이밍 차트

이 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_MoveAbsolute

절대 좌표의 목표 위치를 지정하고 위치 결정을 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_MoveAbsolute	절대값 위치 결정	FB	MC_MoveAbsolute_instance MC_MoveAbsolute Axis — — Axis Execute — Done Position — Busy Velocity — Active Acceleration — CommandAborted Deceleration — Error Jerk — ErrorID Direction BufferMode	MC_MoveAbsolute_instance (Axis := 《파라메터》, Execute := 《파라메터》, Position := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, Direction := 《파라메터》, BufferMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Position	목표 위치	LREAL	음수, 양수, '0'	0	절대 좌표의 목표 위치를 지정합니다. 단위는 [명령 단위]입니다. *1
Velocity	목표 속도	LREAL	양수	0	목표 속도를 지정합니다. *2 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Direction	방향 선택	_eMC_ DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	0 *3	카운트 모드가 [로터리 모드]인 경우 회전 방식을 지정합니다. 0 : 정방향 지정 1 : 근거리 회전 지정 2 : 역방향 지정 3 : 현재 방향 지정 4 : 방향 지정 없음
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered 2 : BlendingLow 3 : BlendingPrevious 4 : BlendingNext 5 : BlendingHigh

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 반드시 목표 속도를 설정하십시오. 설정하지 않고 동작하게 하면 이상이 발생합니다.
*3. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	위치 결정이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

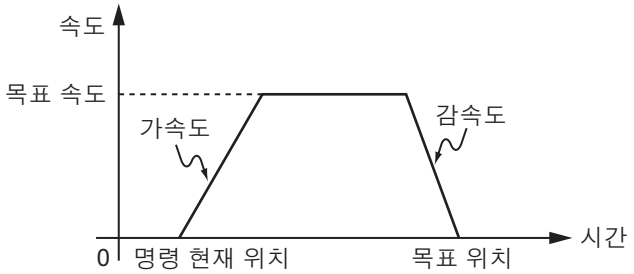
입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

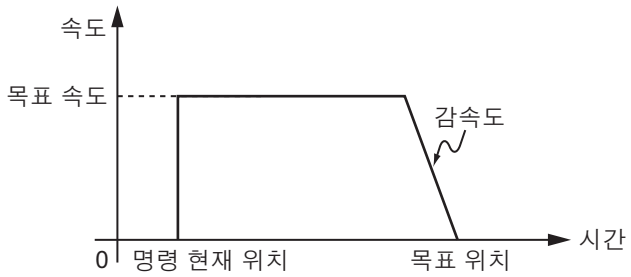
기능 설명

- 절대 좌표의 목표 위치를 지정하고 위치 결정을 실행합니다.
- Execute(기동)가 기동하면 절대 위치 결정 동작을 시작합니다.
- 원점이 확정되지 않아도 이 명령을 실행할 수 있습니다.
- 입력 변수에 Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도), Jerk(저크)를 지정할 수 있습니다.

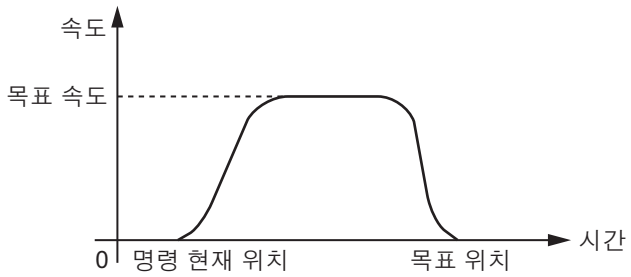
절대 위치 결정의 동작 예는 다음과 같습니다.



가속도 또는 감속도를 '0'으로 지정해서 기동하면 가감속하지 않고 목표 속도에 도달합니다.
가속도가 '0'인 경우의 동작 예는 다음과 같습니다.



완만하게 가감속하게 하려면 Jerk(저크)를 지정합니다.
Jerk(저크)를 지정한 경우의 동작 예는 다음과 같습니다.



Jerk(저크)에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

Position(목표 위치)을 명령 현재 위치와 동일하게 지정해서 기동하면 축은 이동하지 않지만 Done(완료)은 TRUE가 됩니다.

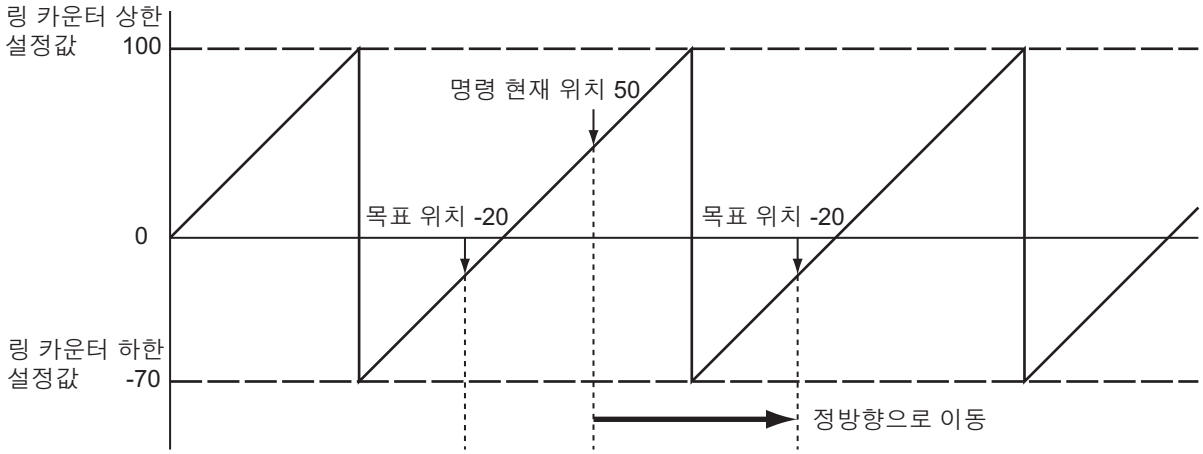
명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

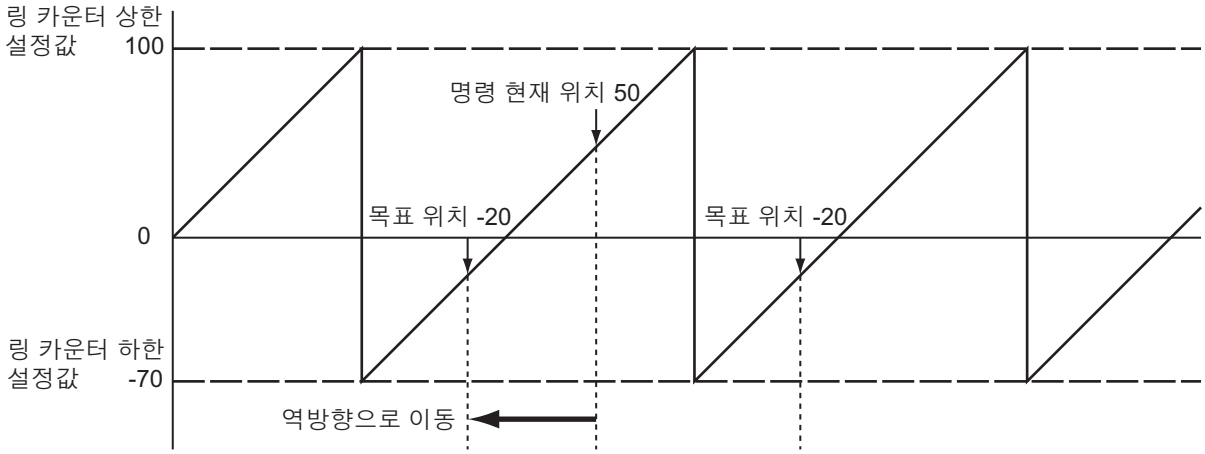
● Direction(방향 선택)

카운트 모드가 [로터리 모드]이면 Direction(방향 선택)에서 위치 결정을 시작할 방향을 지정합니다.
카운트 모드가 [리니어 모드]인 경우 Direction(방향 선택)은 무시됩니다.

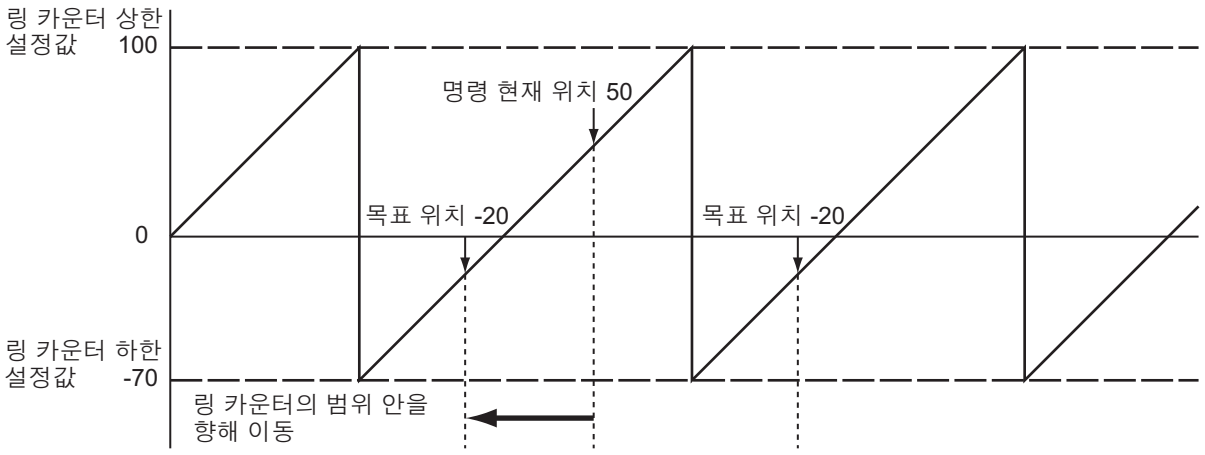
Direction(방향 선택)이 '정방향 지정'인 경우 정방향의 목표 위치를 향해서 위치 결정을 실행합니다.
명령 위치가 '50'에서 '-20'을 향해 위치 결정을 실행하는 경우의 동작 예는 다음과 같습니다.



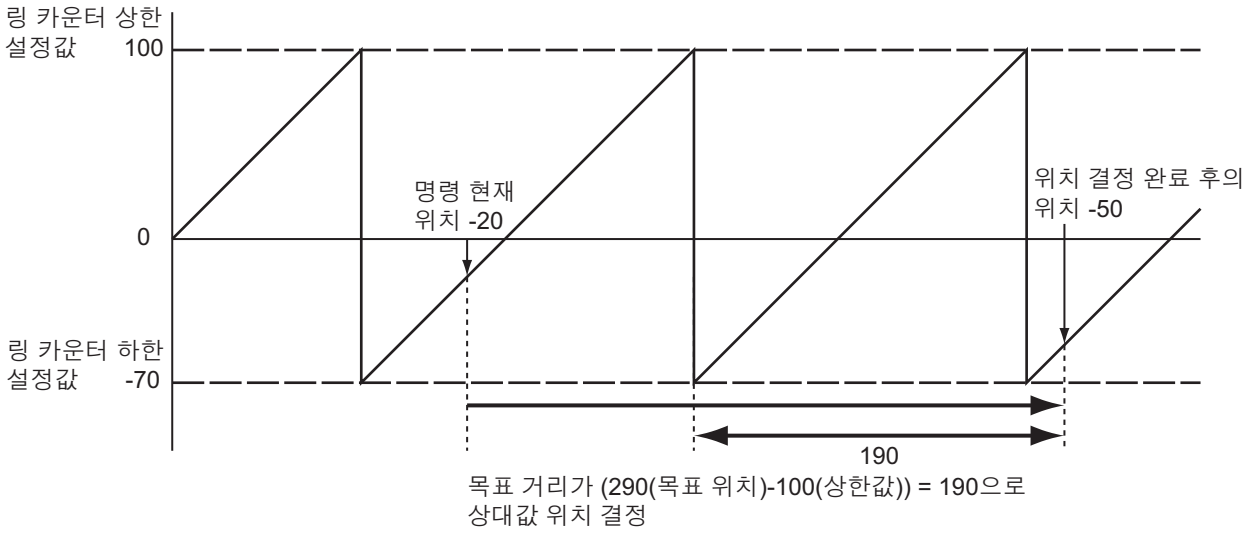
Direction(방향 선택)이 '역방향 지정'인 경우 역방향의 목표 위치를 향해서 위치 결정을 실행합니다.
명령 위치가 '50'에서 '-20'을 향해 위치 결정을 실행하는 경우의 동작 예는 다음과 같습니다.



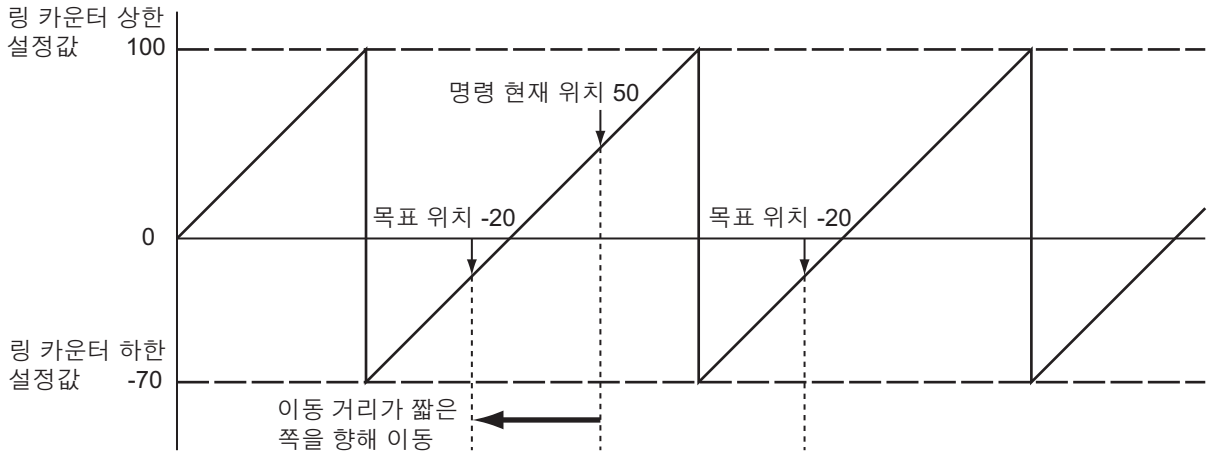
Direction(방향 선택)이 '방향 지정 없음'인 경우 링 카운터 범위 이내의 목표 위치를 향해서 위치 결정을 실행합니다. 따라서 명령 현재 위치와 목표 위치의 대소 관계로 이동 방향이 결정됩니다.
명령 위치가 '50'에서 '-20'을 향해 위치 결정을 실행하는 경우의 동작 예는 다음과 같습니다.



Direction(방향 선택)이 '방향 지정 안 함'인 경우 링 카운터 상하한 설정값 범위를 초과한 Position(목표 위치)을 지정할 수 있습니다.
링 카운터 상한 설정값을 초과한 Position(목표 위치)을 지정하면 링 카운터 상한 설정값을 초과한 이동량만큼 초과한 후의 상대량으로 위치 결정을 실행합니다. 따라서 링 다회전 위치 결정이 가능해집니다.
링 카운터 하한값 설정을 초과한 Position(목표 위치)을 지정한 경우에도 동일하게 위치 결정을 실행합니다.
명령 현재 위치가 '-20'이고 Position(목표 위치)을 '290'으로 지정한 경우의 동작 예는 다음과 같습니다.



Direction(방향 선택)이 '근거리 회전 지정'이면 명령 현재 위치와 목표 위치의 거리가 짧은 방향을 향해서 위치 결정을 실행합니다.
명령 위치가 '50'에서 '-20'을 향해 위치 결정을 실행하는 경우의 동작 예는 다음과 같습니다.



정방향 이동 거리와 역방향 이동 거리가 같은 경우는 '현재 방향 지정' 시와 동일하게 동작합니다.

사용상의 주의 사항

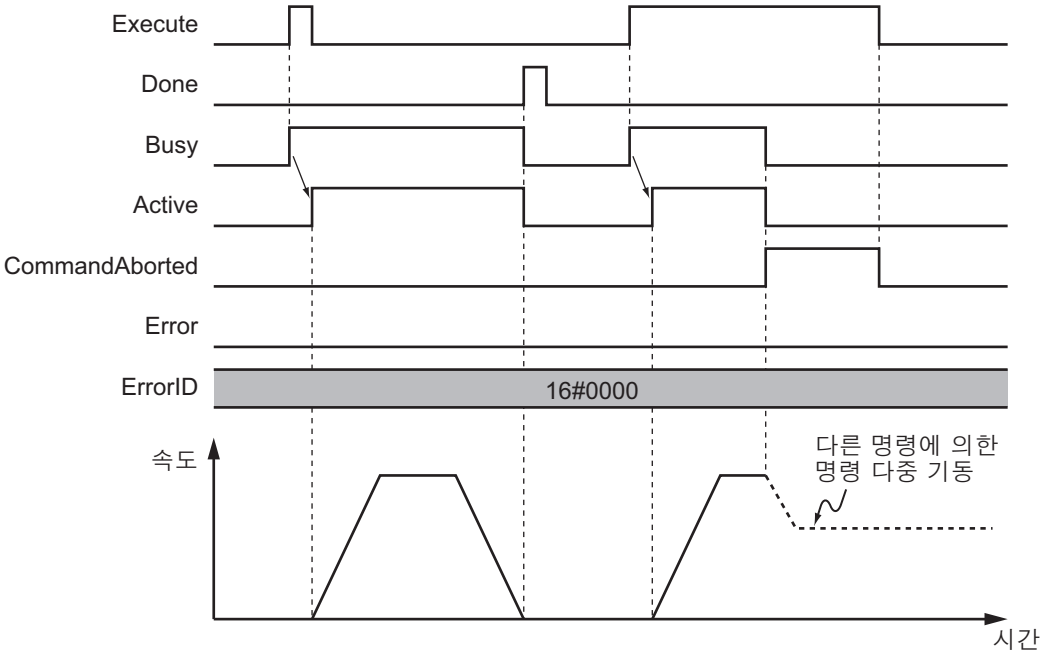
- 절대값 위치 결정을 실행하려면 축 파라메터의 [링 카운터 상한 설정값] 미만 및 [링 카운터 하한 설정값] 이상의 범위 이내로 목표 위치를 지정하십시오.
  목표 위치를 [링 카운터 상한값 설정] 이상 또는 [링 카운터 하한값 설정] 미만으로 지정하면 '목표 위치 설정 범위 외(에러 코드: 5478Hex)'가 됩니다.
  단, Direction(방향 선택)이 '방향 지정 없음'인 경우는 링 카운터 상하한 설정값 범위를 초과한 Position(목표 위치)을 지정할 수 있습니다.
- Direction(방향 선택)에 '3: _mcCurrentDirection(현재 방향 지정)'을 선택하면 이전 동작의 명령 방향을 학습해서 동작합니다. 따라서 명령의 조합에 따라서는 이전 동작의 모션 명령 입력으로 명령한 방향과 일치하지 않는 경우가 있습니다.
  '3: _mcCurrentDirection(현재 방향 지정)'을 사용하는 경우는 축 변수의 Dir. Posi(정방향 명령 중) 및 Dir. Nega(역방향 명령 중)로 현재 방향을 확인하십시오.

● 인포지션 확인

이 명령의 인포지션 확인은 축 파라메터의 [인포지션 폭], [인포지션 확인 시간] 설정에 따라 실행됩니다.

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- Position(목표 위치)에 도달하고 위치 결정이 완료되면 Done(완료)이 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중지되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중)가 FALSE가 됩니다.



모션 명령 재기동

위치 결정 동작 중에 입력 변수를 변경하고 Execute(기동)를 다시 TRUE로 하면 이 명령의 동작을 변경할 수 있습니다.
모션 명령 재기동으로 변경 가능한 입력 변수는 Position(목표 위치), Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)입니다.

모션 명령 재기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다.

각 축에 1개까지 버퍼링할 수 있습니다.

다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택		설 명
Aborting		현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.
Buffered		현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.
Blending		현재 실행 중인 명령이 목표 위치에 도달했을 때의 속도(중계 속도)를 기동 속도로 버퍼링 중인 이 명령을 연속해서 동작하게 합니다. 중계 속도로 목표 위치에 도달하도록 현재 실행 중인 명령의 동작을 변경합니다. 중계 속도 지정 방법은 다음의 4종류가 있습니다.
	BlendingLow	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 낮은 쪽을 중계 속도로 합니다.
	BlendingPrevious	현재 실행 중인 목표 속도를 중계 속도로 합니다.
	BlendingNext	버퍼링되고 있는 이 명령의 목표 속도를 중계 속도로 합니다.
	BlendingHigh	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 높은 쪽을 중계 속도로 합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

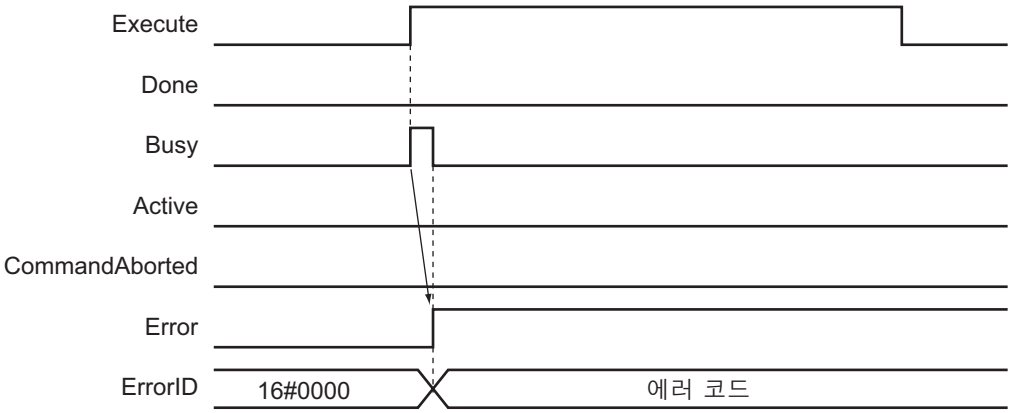
● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하는 경우 Aborting, Buffered, Blending이 가능합니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램 1

여기에는 다중 기동 반복에 의한 위치 결정 방법에 대한 샘플 프로그램이 실려 있습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축

카운터 모드

축	카운터 모드
축 1	리니어 모드

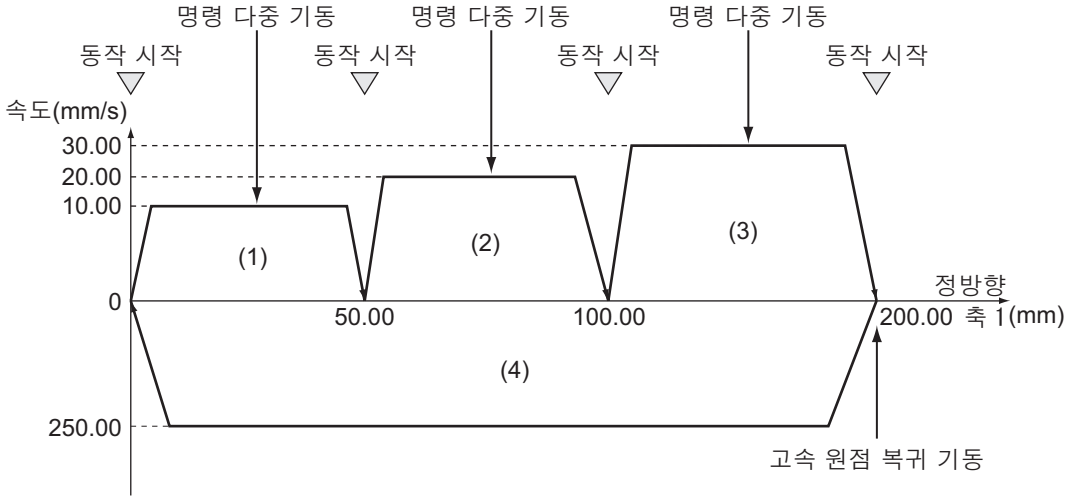
좌표 단위 선택

축	좌표 단위
축 1	mm

동작 예

MC_MoveAbsolute(절대값 위치 결정) 명령에 대해 BufferMode(버퍼 모드 선택)를 [buffered]로 하고 모션 명령 다중 기동을 실행하면서 최종 목표 위치까지 이동시킵니다.
최종 목표 위치에 도달하면 MC_ZeroPosition(고속 원점 복귀) 명령으로 원점으로 되돌립니다.
다중 기동을 실행하는 경우는 직전 명령의 출력 변수 'Active(제어 중)'가 TRUE 상태일 때 실행합니다. 단 축 동작의 명령 다중 기동은 1명령까지 가능합니다.

● 동작 패턴



1 **동작 시작 스위치를 'ON'**

원점에서 동작 시작 스위치를 'ON'하면 축 1이 정방향으로 50.00mm 이동해 위치를 결정합니다.

2 **계속해서 동작 시작 스위치를 'ON'**

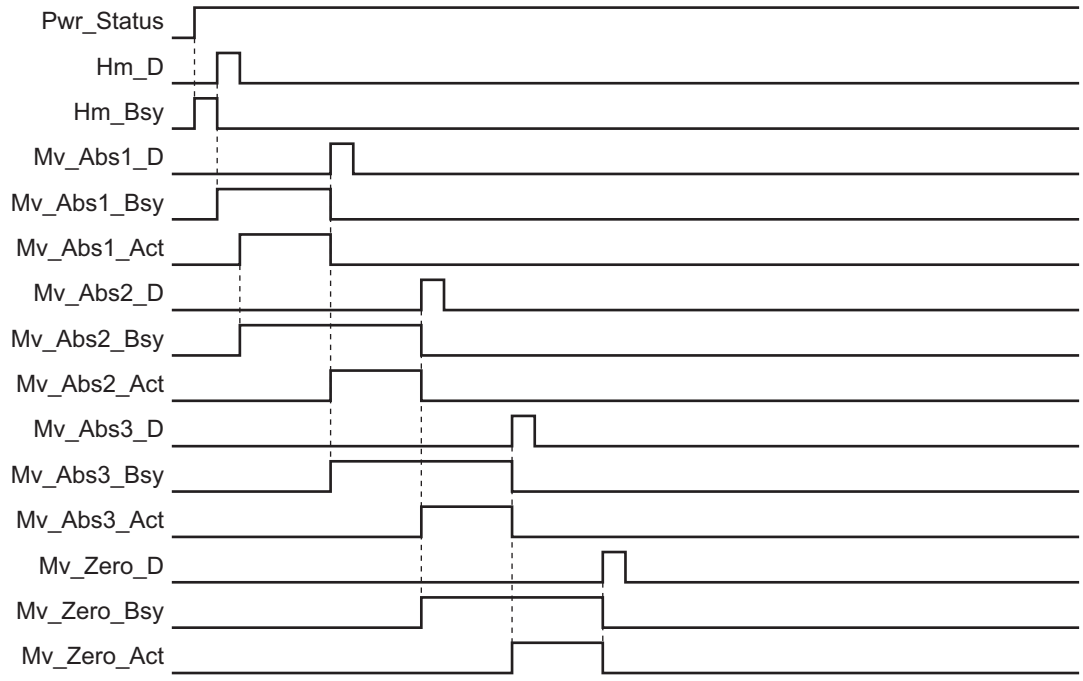
계속해서 동작 시작 스위치를 'ON'할 때마다 축 1이 정방향으로 100.00mm, 200.00mm에 위치를 결정하고 원점으로 돌아가 정지합니다.

래더도

● 주요 변수

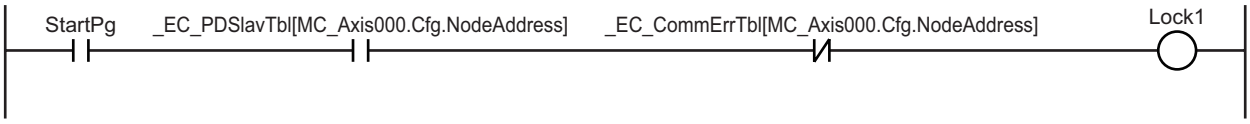
이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.

● 타이밍 차트

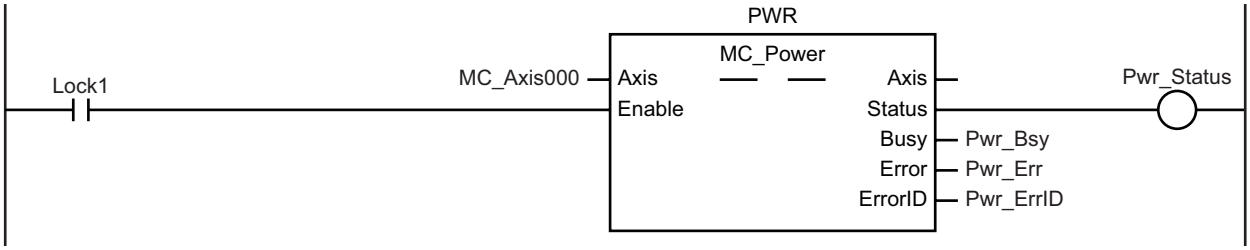


● 샘플 프로그램

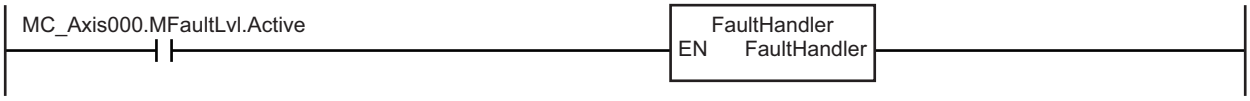
접점 StartPg가 TRUE가 되면 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인



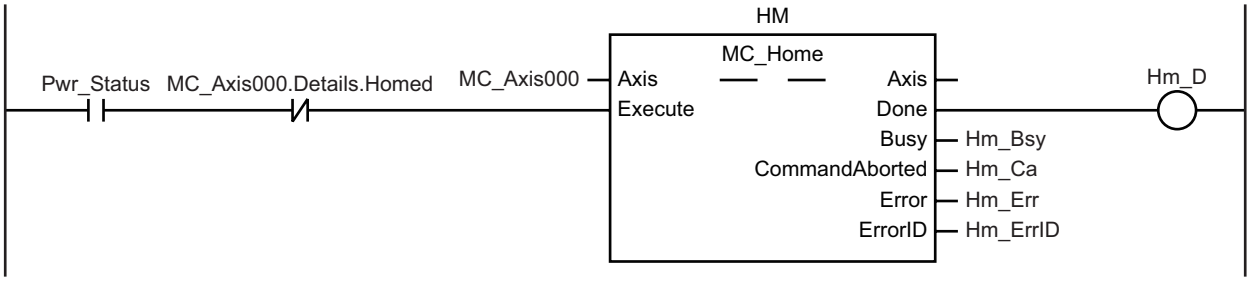
프로세스 데이터 통신을 하고 있으면 서보를 ON



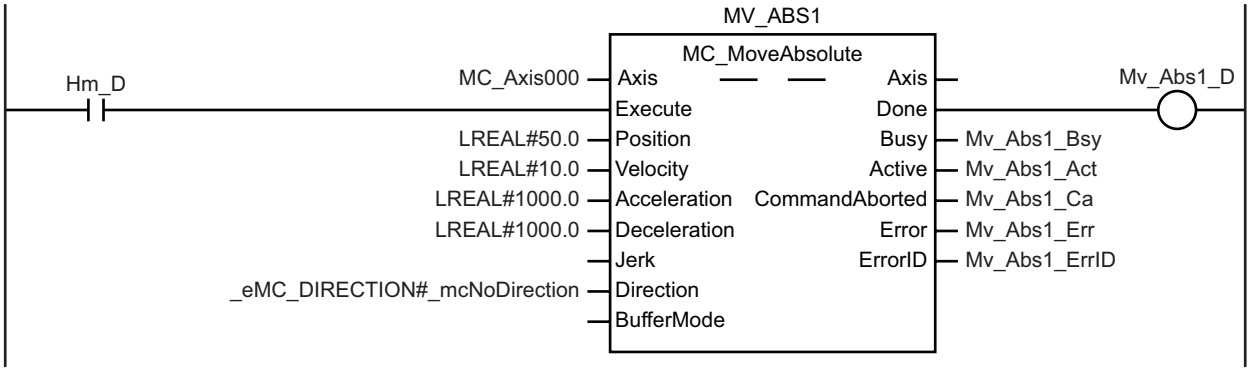
축 1에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



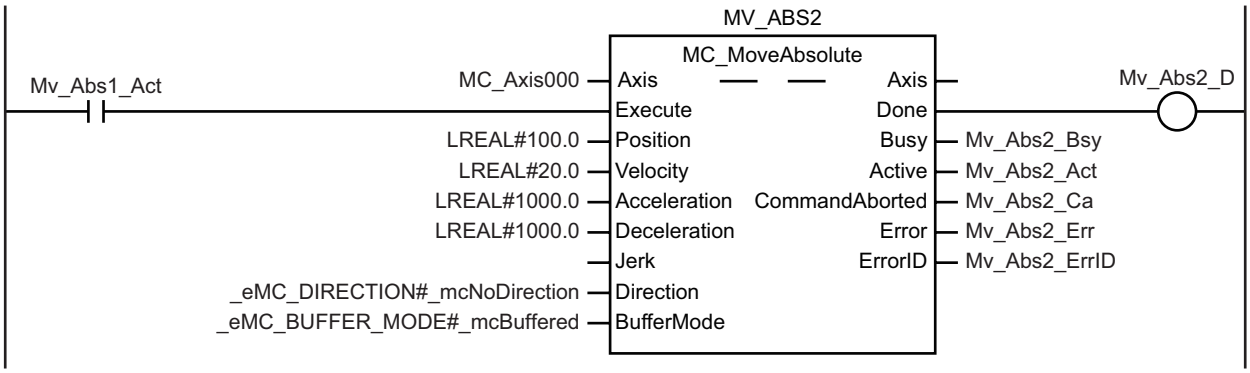
서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행



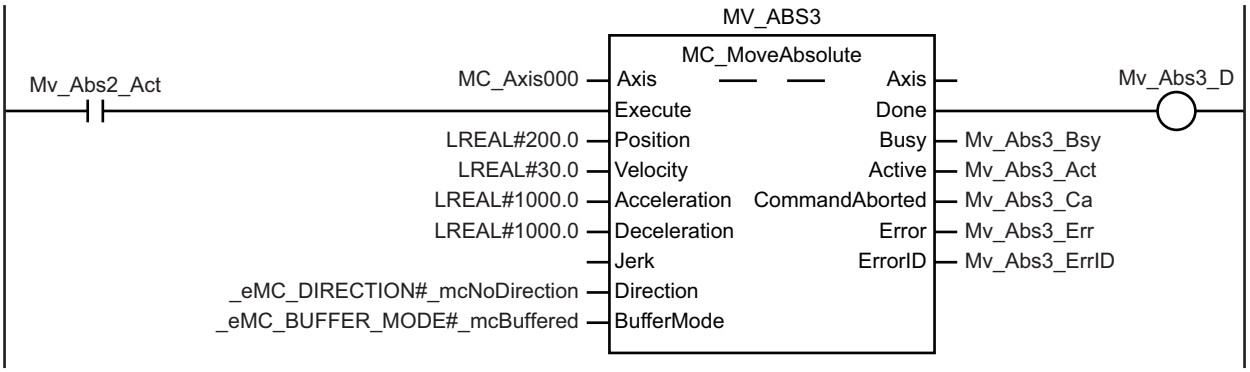
원점 확정 후 절대값 위치 결정(1)을 기동



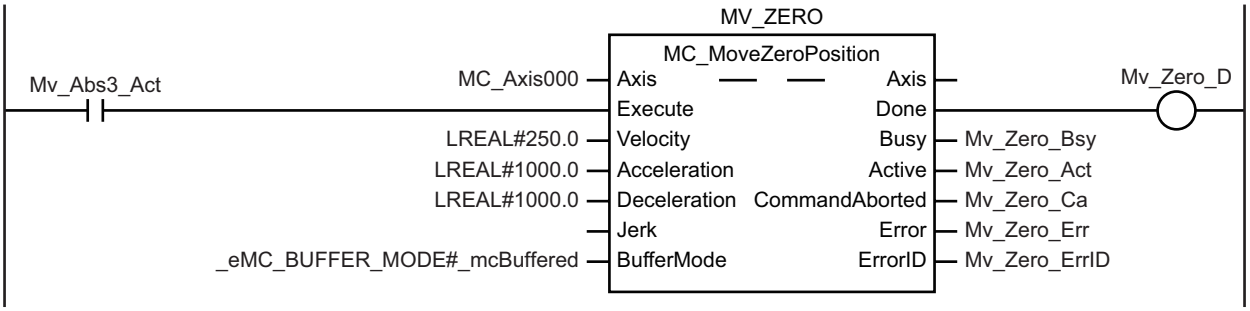
절대값 위치 결정(1)이 시작된 후 절대값 위치 결정(2)을 다중 기동



절대값 위치 결정(2)이 시작된 후 절대값 위치 결정(3)을 다중 기동



절대값 위치 결정(3)이 시작된 후 고속 원점 복귀를 실행



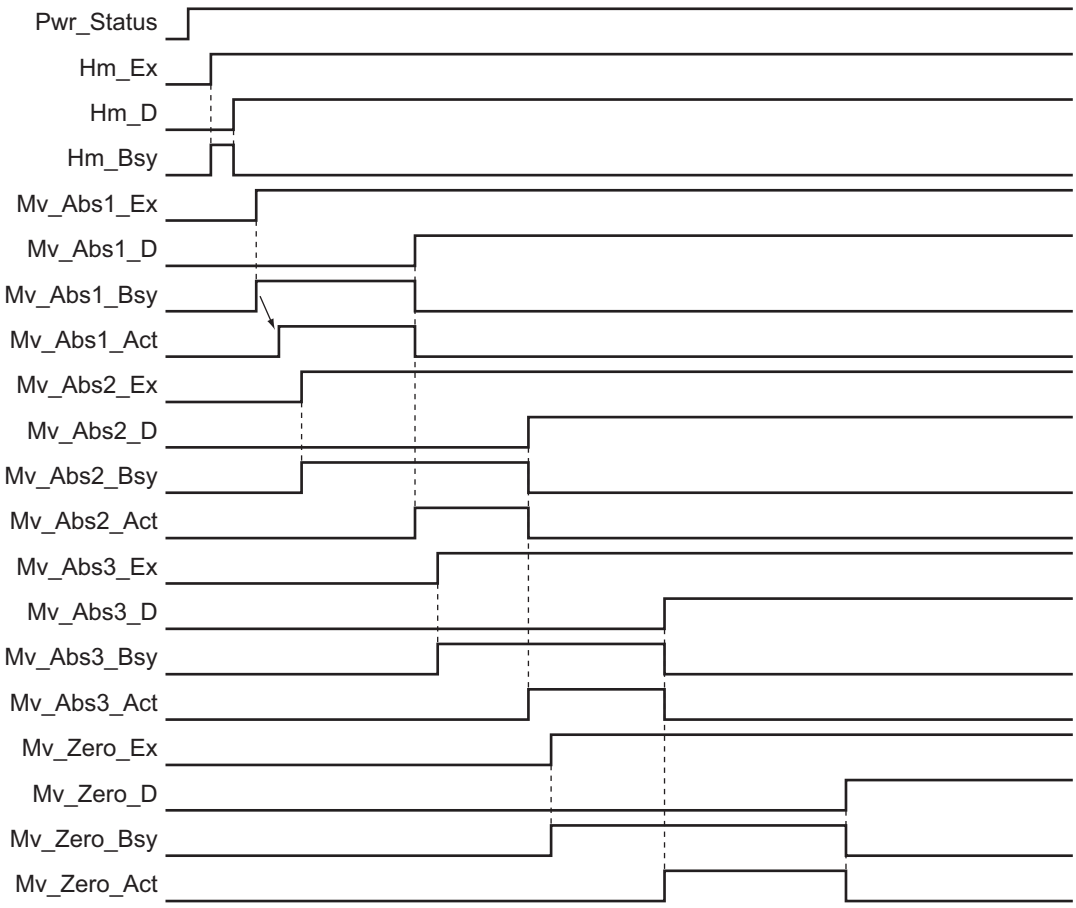
스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Hm_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_Home의 인스턴스 HM을 기동합니다.
Mv_Abs1_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveAbsolute의 인스턴스 MV_ABS1을 기동합니다.
Mv_Abs2_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveAbsolute의 인스턴스 MV_ABS2를 기동합니다.
Mv_Abs3_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveAbsolute의 인스턴스 MV_ABS3을 기동합니다.
Mv_Zero_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveZeroPosition의 인스턴스 MV_ZERO를 기동합니다.

이 름	데이터 형식	초기값	코멘트
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트



● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag = FALSE THEN

    //MV_ABS1 파라메터
    Mv_Abs1_Pos  := LREAL#50.0;
    Mv_Abs1_Vel  := LREAL#10.0;
    Mv_Abs1_Acc  := LREAL#1000.0;
    Mv_Abs1_Dec  := LREAL#1000.0;
    Mv_Abs1_Dir  := _eMC_DIRECTION#_mcNoDirection;

    //MV_ABS2 파라메터
    Mv_Abs2_Pos  := LREAL#100.0;
    Mv_Abs2_Vel  := LREAL#20.0;
    Mv_Abs2_Acc  := LREAL#1000.0;
    Mv_Abs2_Dec  := LREAL#1000.0;
    Mv_Abs2_Dir  := _eMC_DIRECTION#_mcNoDirection;
    Mv_Abs2_Bm   := _eMC_BUFFER_MODE#_mcBuffered;

    //MV_ABS3 파라메터
    Mv_Abs3_Pos  := LREAL#200.0;
    Mv_Abs3_Vel  := LREAL#30.0;
    Mv_Abs3_Acc  := LREAL#1000.0;
```

```
    Mv_Abs3_Dec  := LREAL#1000.0;
    Mv_Abs3_Dir   := _eMC_DIRECTION#_mcNoDirection;
    Mv_Abs3_Bm   := _eMC_BUFFER_MODE#_mcBuffered;

    //MV_ZERO 파라메터
    Mv_Zero_Vel   := LREAL#250;
    Mv_Zero_Acc   := LREAL#1000.0;
    Mv_Zero_Dec   := LREAL#1000.0;
    Mv_Zero_Bm   := _eMC_BUFFER_MODE#_mcBuffered;

    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//가벼운 수준의 이상 발생 시 처리
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;


//서보 ON 상태이고 원점 미확정 상태일 때 원점 복귀를 실행합니다.
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

//원점 확정 후에 MV_ABS1를 실행합니다.
IF Hm_D=TRUE THEN
    Mv_Abs1_Ex:=TRUE;
END_IF;

//MV_ABS1을 시작한 후 MV_ABS2를 다중 기동합니다.
IF Mv_Abs1_Act=TRUE THEN
    Mv_Abs2_Ex:=TRUE;
END_IF;

//MV_ABS2를 시작한 후 MV_ABS3을 다중 기동합니다.
IF Mv_Abs2_Act=TRUE THEN
    Mv_Abs3_Ex:=TRUE;
END_IF;

//MV_ABS3을 시작한 후 MV_ZERO를 다중 기동합니다.
IF Mv_Abs3_Act=TRUE THEN
    Mv_Zero_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_ABS1(
    Axis                := MC_Axis000,
    Execute             := Mv_Abs1_Ex,
    Position            := Mv_Abs1_Pos,
    Velocity            := Mv_Abs1_Vel,
    Acceleration        := Mv_Abs1_Acc,
    Deceleration        := Mv_Abs1_Dec,
    Direction           := Mv_Abs1_Dir,
    Done                => Mv_Abs1_D,
    Busy                => Mv_Abs1_Bsy,
    Active              => Mv_Abs1_Act,
    CommandAborted      => Mv_Abs1_Ca,
    Error               => Mv_Abs1_Err,
    ErrorID             => Mv_Abs1_ErrID
);


MV_ABS2(
    Axis                := MC_Axis000,
    Execute             := Mv_Abs2_Ex,
    Position            := Mv_Abs2_Pos,
    Velocity            := Mv_Abs2_Vel,
    Acceleration        := Mv_Abs2_Acc,
    Deceleration        := Mv_Abs2_Dec,
    Direction           := Mv_Abs2_Dir,
    BufferMode          := Mv_Abs2_Bm,
    Done                => Mv_Abs2_D,
    Busy                => Mv_Abs2_Bsy,
    Active              => Mv_Abs2_Act,
    CommandAborted      => Mv_Abs2_Ca,
    Error               => Mv_Abs2_Err,
    ErrorID             => Mv_Abs2_ErrID
);


MV_ABS3(
    Axis                := MC_Axis000,
    Execute             := Mv_Abs3_Ex,
    Position            := Mv_Abs3_Pos,
    Velocity            := Mv_Abs3_Vel,
    Acceleration        := Mv_Abs3_Acc,
```

```
    Deceleration        := Mv_Abs3_Dec,
    Direction           := Mv_Abs3_Dir,
    BufferMode          := Mv_Abs3_Bm,
    Done                => Mv_Abs3_D,
    Busy                => Mv_Abs3_Bsy,
    Active              => Mv_Abs3_Act,
    CommandAborted      => Mv_Abs3_Ca,
    Error               => Mv_Abs3_Err,
    ErrorID             => Mv_Abs3_ErrID
);

//MC_MoveZeroPosition
MV_ZERO(
    Axis                := MC_Axis000,
    Execute             := Mv_Zero_Ex,
    Velocity            := Mv_Zero_Vel,
    Acceleration        := Mv_Zero_Acc,
    Deceleration        := Mv_Zero_Dec,
    BufferMode          := Mv_Zero_Bm,
    Done                => Mv_Zero_D,
    Busy                => Mv_Zero_Bsy,
    Active              => Mv_Zero_Act,
    CommandAborted      => Mv_Zero_Ca,
    Error               => Mv_Zero_Err,
    ErrorID             => Mv_Zero_ErrID
);
```

샘플 프로그램 2

카운트 모드가 로터리 모드일 때 목표 위치로 위치 결정 시 좌우 어느 쪽으로 회전하는 것이 가까운지를 자동으로 판별해서 위치 결정을 실행합니다.
여기에는 툴 체인저의 근거리 회전 제어에 대한 샘플 프로그램이 실려 있습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축

카운터 모드

축	카운터 모드
축 1	로터리 모드

링 카운터

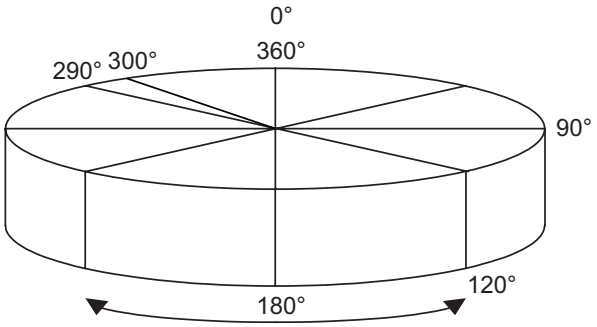
축	상한값	하한값
축 1	360	0

좌표 단위 선택

축	좌표 단위
축 1	degree

동작 예

절대값 위치 결정 명령의 다중 기동을 이용한 0~360° 범위의 동작 예는 다음과 같습니다. 또한 현재 위치는 0~360° 범위를 초과하면 0°로 되돌아옵니다.
다중 기동을 실행하는 경우는 직전 명령의 출력 변수 'Active(제어 중)'가 TRUE 상태일 때 실행합니다. 단축 동작의 명령 다중 기동은 1명령까지 가능합니다.
이 예에서는 BufferMode(버퍼 모드 선택)를 Buffered로 다중 기동하게 합니다.



근거리 회전으로 위치를 결정합니다.

원점(0°), 90°, 120°, 290°의 4가지 임의의 위치를 지정하면 해당 위치로 이동합니다.
이 때의 회전 방식은 가까운 쪽으로 회전합니다. 또한 이동 속도는 250°/s가 됩니다.

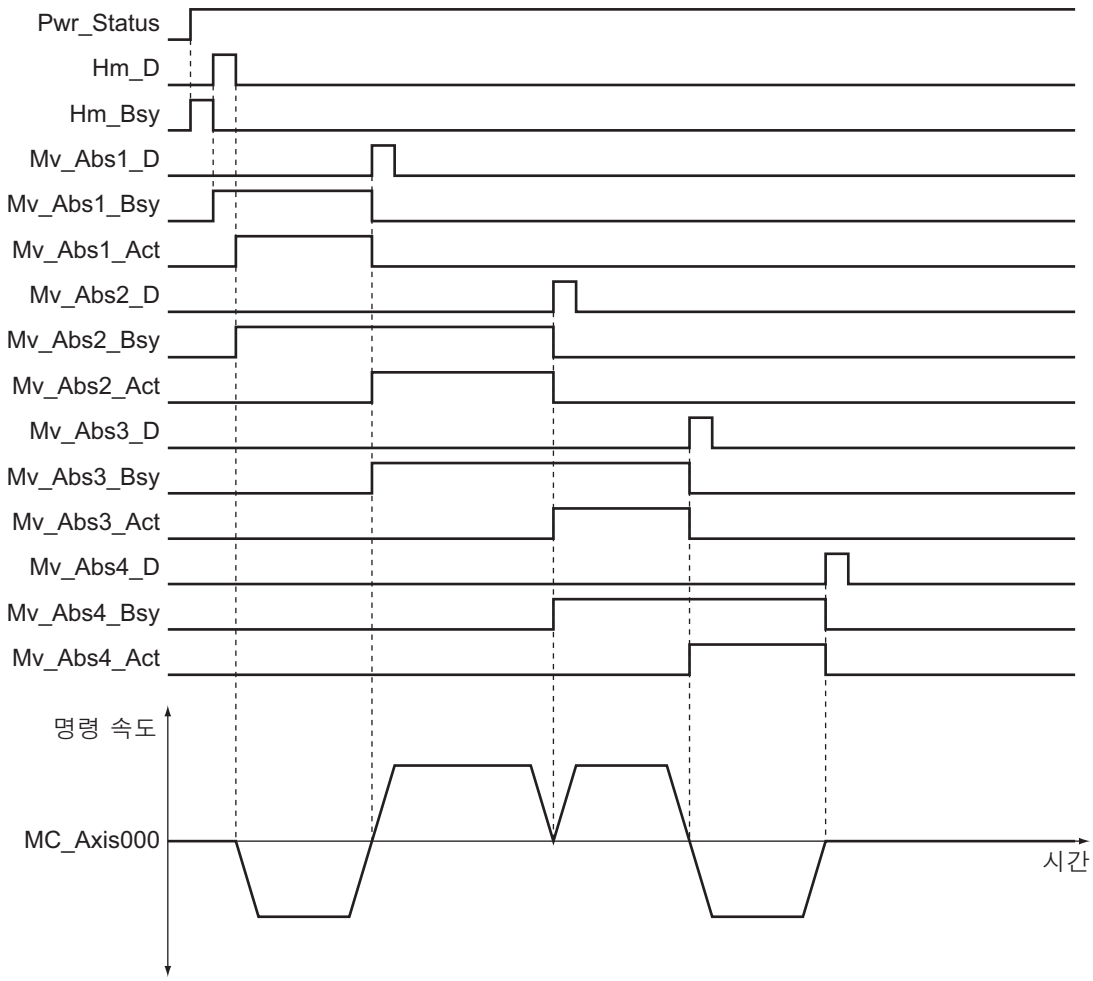
샘플 프로그램에서는 290°→90°→120°→원점(0°)으로 근거리 회전 지정으로 위치를 결정합니다.

래더도

● 주요 변수

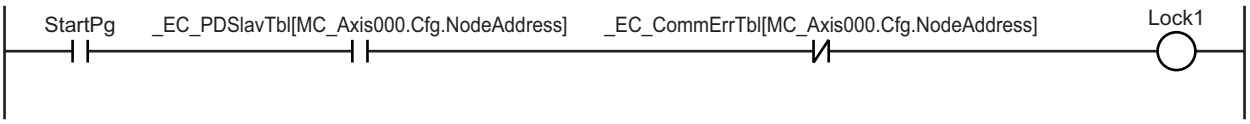
이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.

● 타이밍 차트

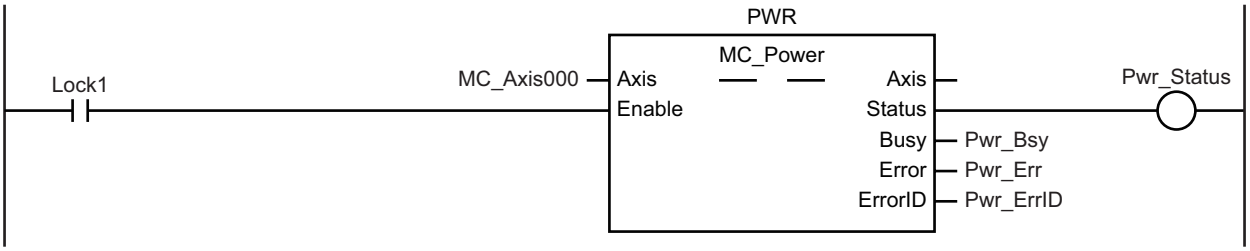


● 샘플 프로그램

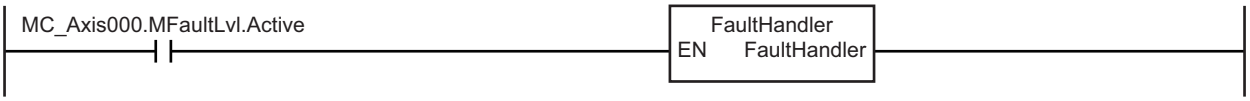
접점 StartPg가 TRUE가 되면 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인



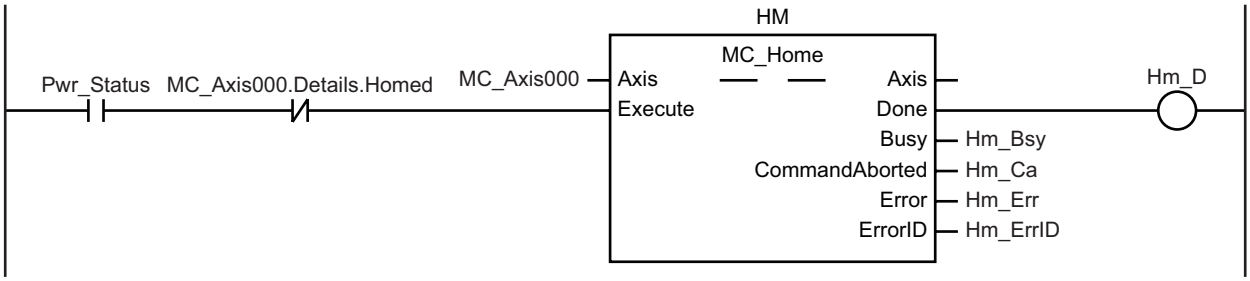
프로세스 데이터 통신을 하고 있으면 서보를 ON



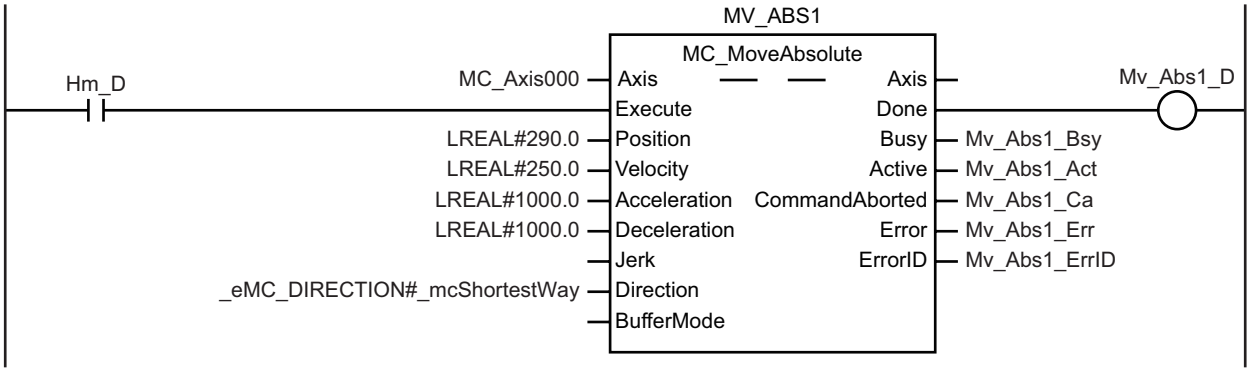
축 1에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



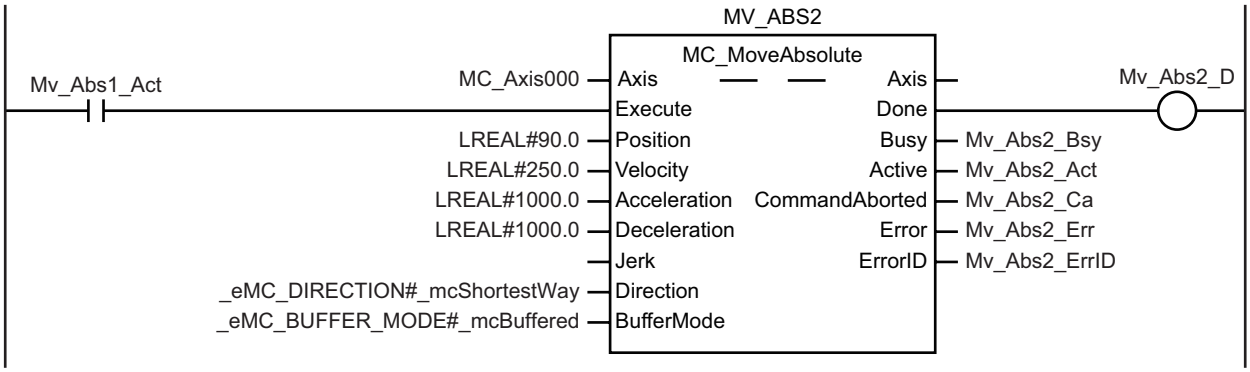
서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행



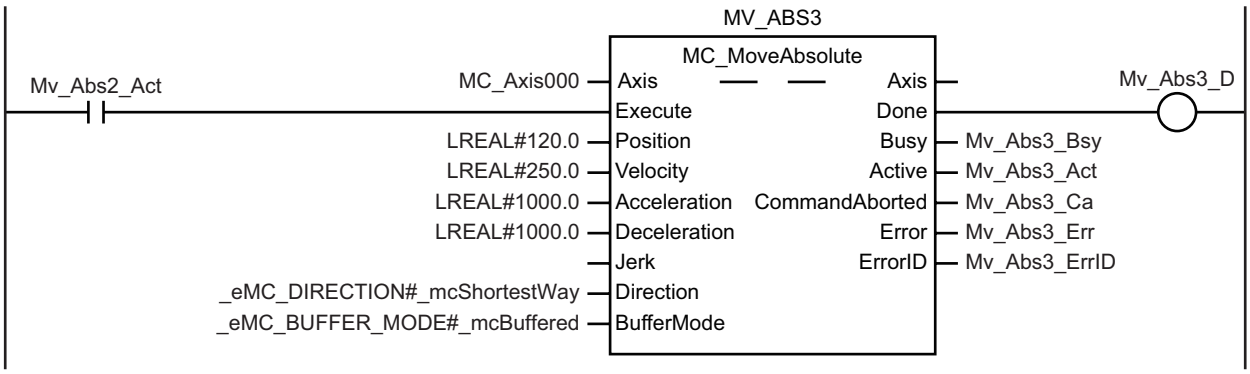
원점 확정 후 절대값 위치 결정(1)을 실행하고 290.0°로 이동. 동작 방향은 근거리 회전 지정



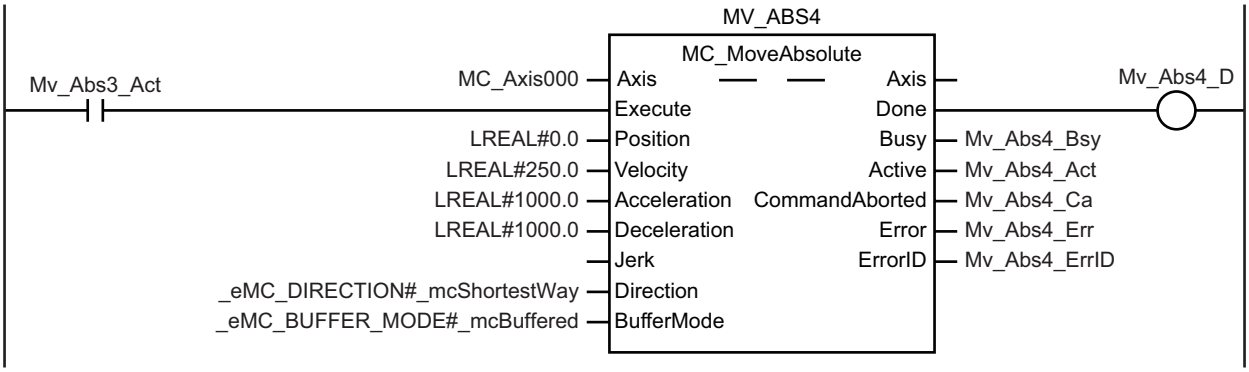
절대값 위치 결정(2)을 다중 기동하고 290.0°에서 90.0°로 이동. 동작 방향은 근거리 회전



절대값 위치 결정(3)을 다중 기동하고 90.0°에서 120.0°로 이동. 동작 방향은 근거리 회전



절대값 위치 결정(4)을 다중 기동하고 120.0°에서 0.0°로 이동. 동작 방향은 근거리 회전

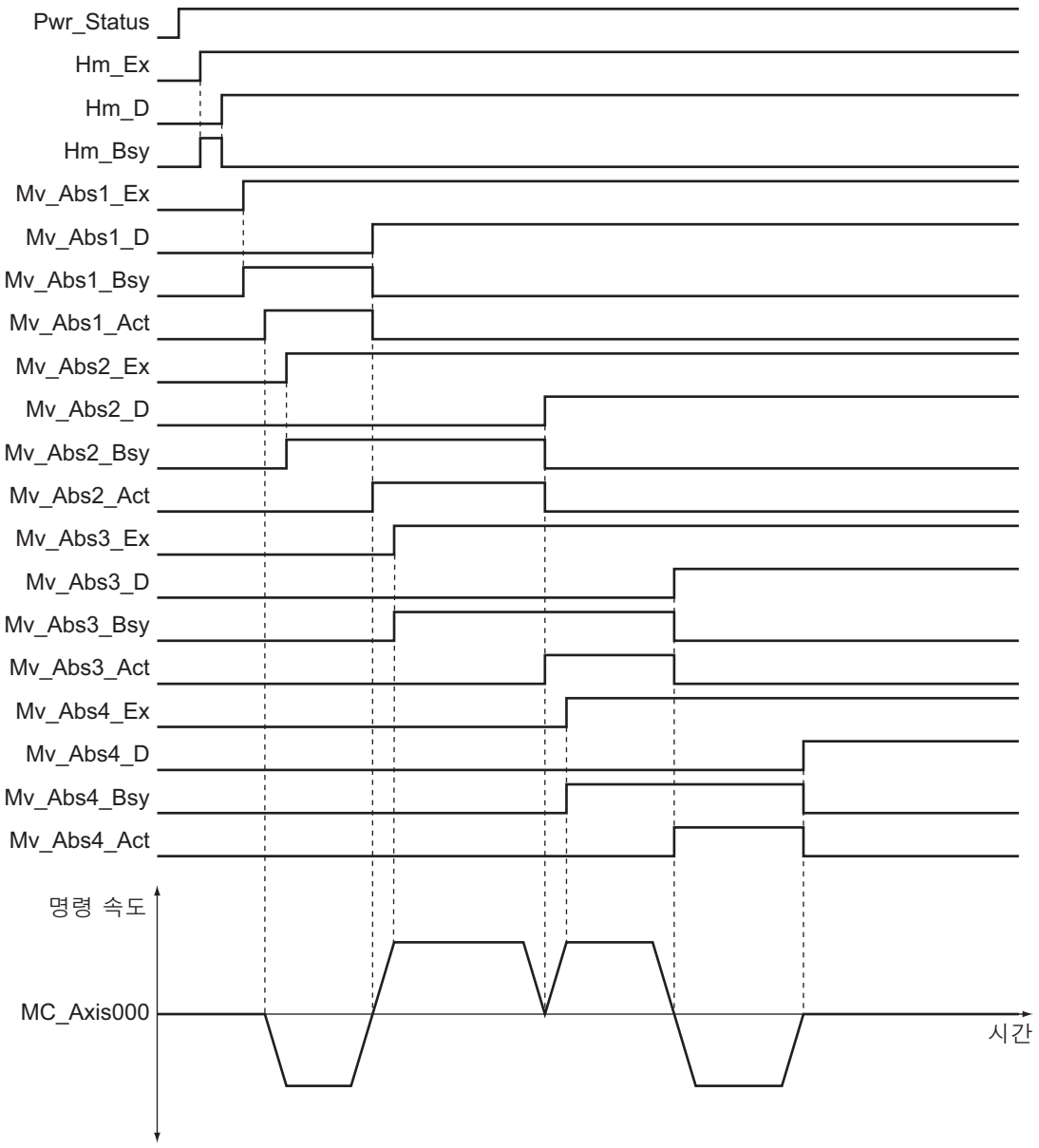


스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Hm_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_Home의 인스턴스 HM을 기동합니다.
Mv_Abs1_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveAbsolute의 인스턴스 MV_ABS1을 기동합니다.
Mv_Abs2_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveAbsolute의 인스턴스 MV_ABS2를 기동합니다.
Mv_Abs3_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveAbsolute의 인스턴스 MV_ABS3을 기동합니다.
Mv_Abs4_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveAbsolute의 인스턴스 MV_ABS4를 기동합니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트



● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag = FALSE THEN

    //MV_ABS1 파라메터
    Mv_Abs1_Pos   := LREAL#290.0;
    Mv_Abs1_Vel   := LREAL#250.0;
    Mv_Abs1_Acc   := LREAL#1000.0;
    Mv_Abs1_Dec   := LREAL#1000.0;
    Mv_Abs1_Dir   := _eMC_DIRECTION#_mcShortestWay;

    //MV_ABS2 파라메터
    Mv_Abs2_Pos   := LREAL#90.0;
    Mv_Abs2_Vel   := LREAL#250.0;
    Mv_Abs2_Acc   := LREAL#1000.0;
    Mv_Abs2_Dec   := LREAL#1000.0;
    Mv_Abs2_Dir   := _eMC_DIRECTION#_mcShortestWay;
    Mv_Abs2_Bm    := _eMC_BUFFER_MODE#_mcBuffered;
```

```
    //MV_ABS3 파라메터
    Mv_Abs3_Pos   := LREAL#120.0;
    Mv_Abs3_Vel   := LREAL#250.0;
    Mv_Abs3_Acc   := LREAL#1000.0;
    Mv_Abs3_Dec   := LREAL#1000.0;
    Mv_Abs3_Dir   := _eMC_DIRECTION#_mcShortestWay;
    Mv_Abs3_Bm    := _eMC_BUFFER_MODE#_mcBuffered;

    //MV_ABS4 파라메터
    Mv_Abs4_Pos   := LREAL#0.0;
    Mv_Abs4_Vel   := LREAL#250.0;
    Mv_Abs4_Acc   := LREAL#1000.0;
    Mv_Abs4_Dec   := LREAL#1000.0;
    Mv_Abs4_Dir   := _eMC_DIRECTION#_mcShortestWay;
    Mv_Abs4_Bm    := _eMC_BUFFER_MODE#_mcBuffered;

    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;


//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//가벼운 수준의 이상 발생 시 처리
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

//서보 ON 상태이고 원점 미확정 상태일 때 원점 복귀를 실행합니다.
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

//원점 확정 후에 MV_ABS1을 실행합니다.
IF Hm_D=TRUE THEN
    Mv_Abs1_Ex:=TRUE;
END_IF;

//MV_ABS1을 시작한 후 MV_ABS2를 다중 기동합니다.
IF Mv_Abs1_Act=TRUE THEN
    Mv_Abs2_Ex:=TRUE;
END_IF;

//MV_ABS2를 시작한 후 MV_ABS3을 다중 기동합니다.
IF Mv_Abs2_Act=TRUE THEN
    Mv_Abs3_Ex:=TRUE;
END_IF;

//MV_ABS3을 시작한 후 MV_ABS4를 다중 기동합니다.
IF Mv_Abs3_Act=TRUE THEN
    Mv_Abs4_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
);

//절대값 위치 결정(1)
MV_ABS1(
    Axis                := MC_Axis000,
    Execute             := Mv_Abs1_Ex,
    Position            := Mv_Abs1_Pos,
    Velocity            := Mv_Abs1_Vel,
    Acceleration        := Mv_Abs1_Acc,
    Deceleration        := Mv_Abs1_Dec,
    Direction           := Mv_Abs1_Dir,
    Done                => Mv_Abs1_D,
    Busy                => Mv_Abs1_Bsy,
    Active              => Mv_Abs1_Act,
    CommandAborted      => Mv_Abs1_Ca,
    Error               => Mv_Abs1_Err,
    ErrorID             => Mv_Abs1_ErrID
);


//절대값 위치 결정(2)
MV_ABS2(
    Axis                := MC_Axis000,
    Execute             := Mv_Abs2_Ex,
    Position            := Mv_Abs2_Pos,
    Velocity            := Mv_Abs2_Vel,
    Acceleration        := Mv_Abs2_Acc,
    Deceleration        := Mv_Abs2_Dec,
    Direction           := Mv_Abs2_Dir,
    BufferMode          := Mv_Abs2_Bm,
    Done                => Mv_Abs2_D,
    Busy                => Mv_Abs2_Bsy,
    Active              => Mv_Abs2_Act,
    CommandAborted      => Mv_Abs2_Ca,
    Error               => Mv_Abs2_Err,
    ErrorID             => Mv_Abs2_ErrID
);


//절대값 위치 결정(3)
MV_ABS3(
    Axis                := MC_Axis000,
    Execute             := Mv_Abs3_Ex,
    Position            := Mv_Abs3_Pos,
    Velocity            := Mv_Abs3_Vel,
    Acceleration        := Mv_Abs3_Acc,
```

```
    Deceleration        := Mv_Abs3_Dec,
    Direction           := Mv_Abs3_Dir,
    BufferMode          := Mv_Abs3_Bm,
    Done                => Mv_Abs3_D,
    Busy                => Mv_Abs3_Bsy,
    Active              => Mv_Abs3_Act,
    CommandAborted      => Mv_Abs3_Ca,
    Error               => Mv_Abs3_Err,
    ErrorID             => Mv_Abs3_ErrID
);

//절대값 위치 결정(4)
MV_ABS4(
    Axis                := MC_Axis000,
    Execute             := Mv_Abs4_Ex,
    Position            := Mv_Abs4_Pos,
    Velocity            := Mv_Abs4_Vel,
    Acceleration        := Mv_Abs4_Acc,
    Deceleration        := Mv_Abs4_Dec,
    Direction           := Mv_Abs4_Dir,
    BufferMode          := Mv_Abs4_Bm,
    Done                => Mv_Abs4_D,
    Busy                => Mv_Abs4_Bsy,
    Active              => Mv_Abs4_Act,
    CommandAborted      => Mv_Abs4_Ca,
    Error               => Mv_Abs4_Err,
    ErrorID             => Mv_Abs4_ErrID
);
```

MC_MoveRelative

명령 현재 위치로부터의 이동 거리를 지정하고 위치 결정을 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_MoveRelative	상대값 위치 결정	FB	MC_MoveRelative_instance MC_MoveRelative Axis — — Axis Execute — Done Distance — Busy Velocity — Active Acceleration — CommandAborted Deceleration — Error Jerk — ErrorID BufferMode	MC_MoveRelative_instance (Axis := 《파라메터》, Execute := 《파라메터》, Distance := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, BufferMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Distance	이동 거리	LREAL	음수, 양수, '0'	0	명령 현재 위치에서의 이동 거리를 지정합니다. 단위는 [명령 단위]입니다. *1
Velocity	목표 속도	LREAL	양수	0	목표 속도를 지정합니다. *2 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0：Aborting 1：Buffered 2：BlendingLow 3：BlendingPrevious 4：BlendingNext 5：BlendingHigh

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 반드시 목표 속도를 설정하십시오. 설정하지 않고 동작하게 하면 이상이 발생합니다.
*3. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	위치 결정이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

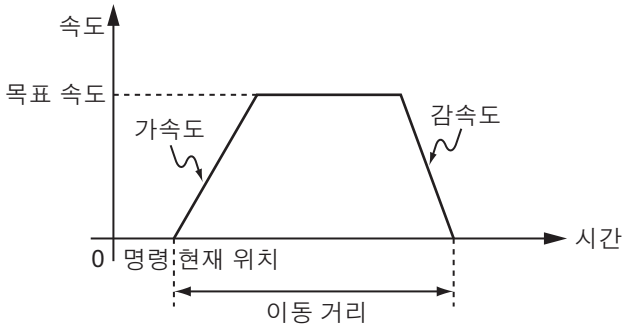
입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

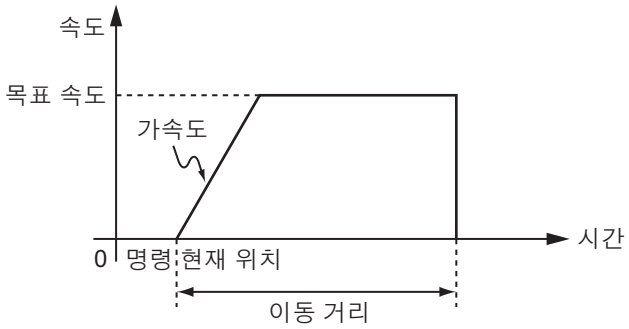
기능 설명

- 명령 현재 위치로부터의 이동 거리를 지정하고 위치 결정을 실행합니다.
- Execute가 기동되면 상대 위치 결정 동작을 시작합니다.
- 입력 변수에 Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도), Jerk(저크)를 지정할 수 있습니다.

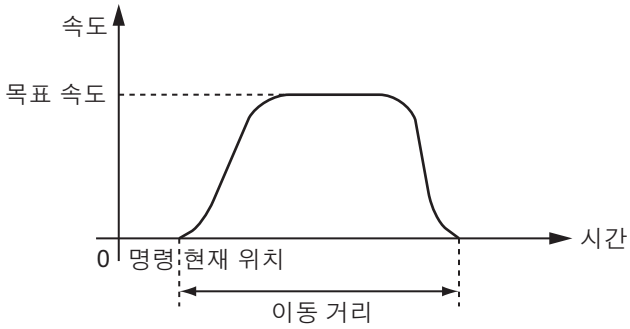
상대 위치 결정의 동작 예는 다음과 같습니다.



가속도 또는 감속도를 '0'으로 지정해서 기동하면 가감속하지 않고 목표 속도에 도달합니다.
감속도가 '0'인 경우의 동작 예는 다음과 같습니다.



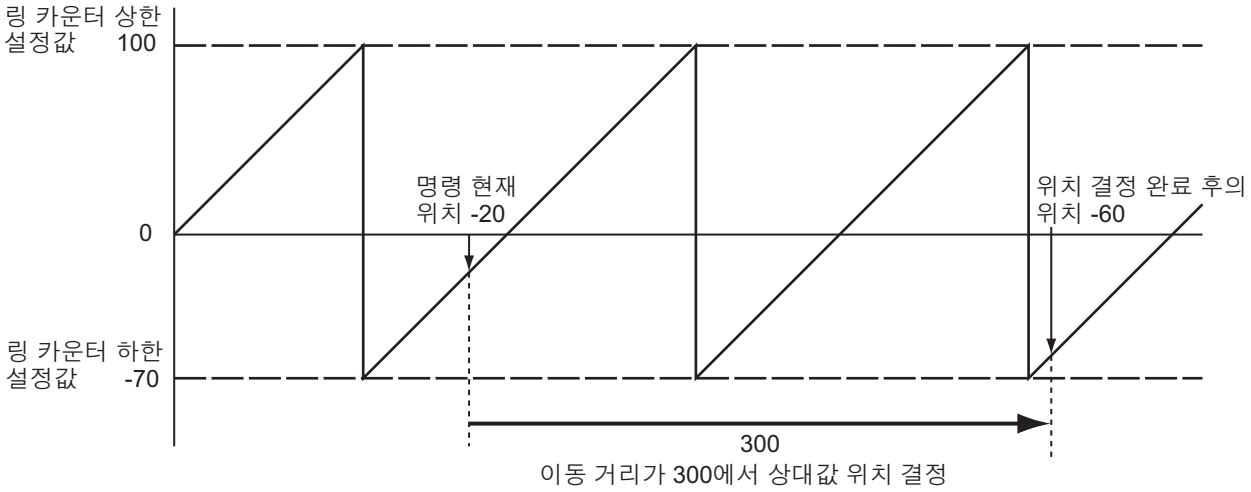
완만하게 가감속하게 하려면 Jerk(저크)를 지정합니다.
Jerk(저크)를 지정한 경우의 동작 예는 다음과 같습니다.



Jerk(저크)에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

명령에 대한 자세한 내용

카운트 모드가 [로터리 모드]인 경우 축 파라메터의 [링 카운터 하한 설정값]으로부터 [링 카운터 상한 설정값]의 상대 거리 범위를 넘는 Distance(이동 거리)를 지정하면 링 다회전 위치 결정이 가능합니다.
명령 현재 위치가 '-20'이고 Distance(이동 거리)를 '300'으로 지정한 경우의 동작 예는 다음과 같습니다.



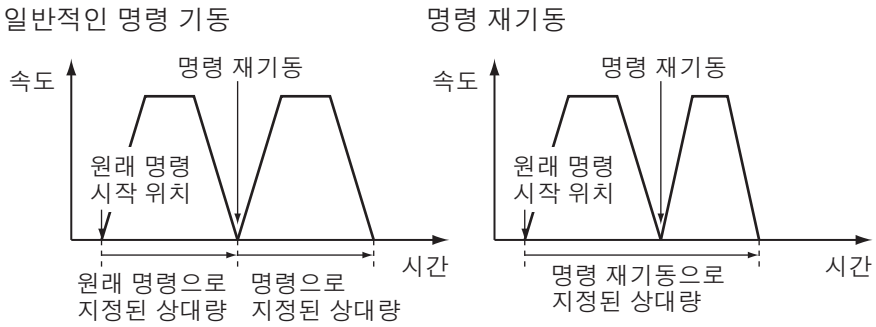
이동 거리가 300에서 상대값 위치 결정

Distance(이동 거리)를 '0'으로 지정해서 기동하면 축은 이동하지 않지만 Done(완료)이 TRUE가 됩니다.

사용상의 주의 사항

위치 결정 완료 직전에 상대값 위치 결정을 재기동하는 경우는 다음 사항에 주의하십시오.
MC 기능 모듈이 재기동할 때까지 위치 결정이 완료되어 있으면 일반적인 명령 기동이 됩니다.

- 일반적인 명령을 기동하는 경우는 명령 기동 시의 위치를 기준으로 한 상대 위치에서 위치 결정을 실행합니다.
- 명령을 재기동하는 경우는 원래 명령 기동 시의 위치를 기준으로 한 상대 위치에서 위치 결정을 실행합니다.

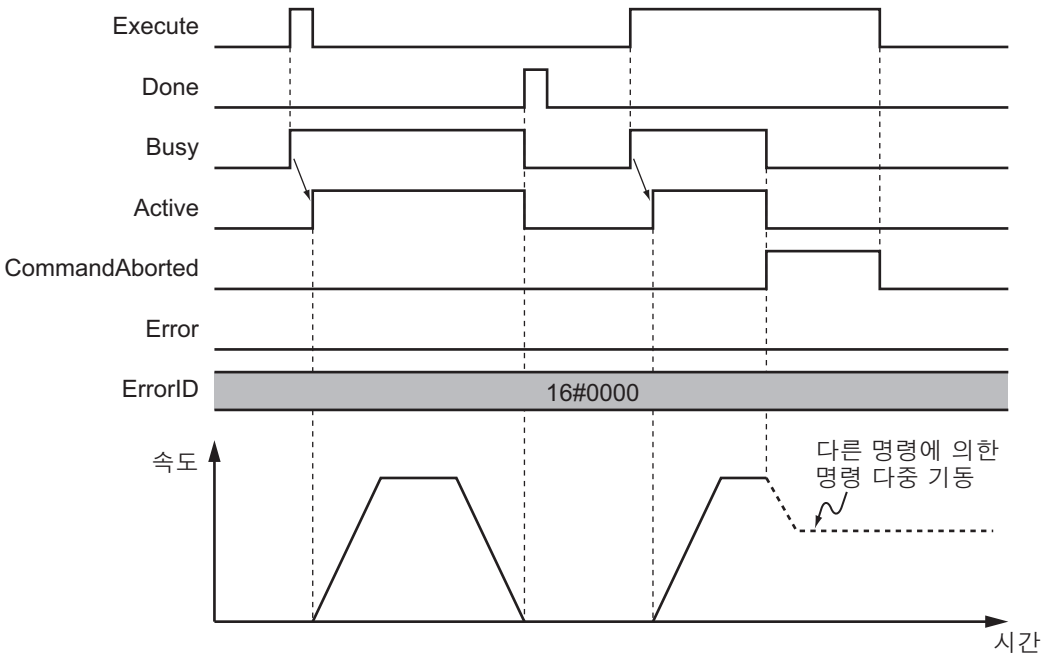


● 인포지션 확인

이 명령의 인포지션 확인은 축 파라메터의 [인포지션 폭], [인포지션 확인 시간] 설정에 따라 실행됩니다.

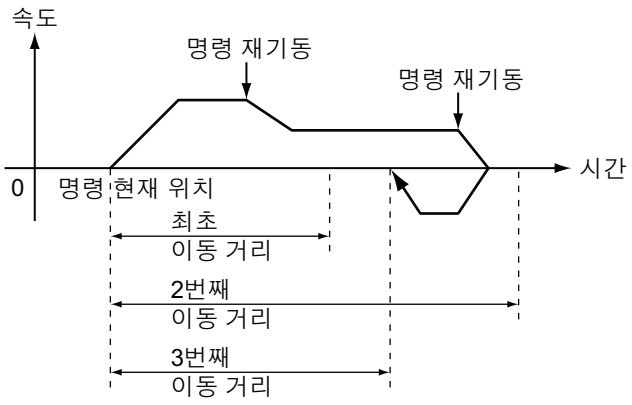
타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- Distance(이동 거리)에 도달하고 위치 결정이 완료되면 Done(완료)이 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중지되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중)가 FALSE가 됩니다.



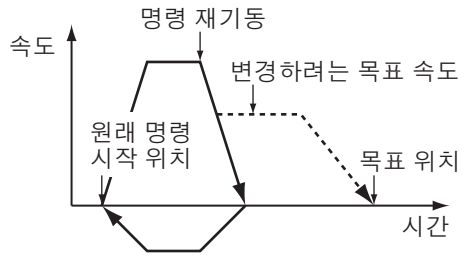
모션 명령 재기동

위치 결정 동작 중에 입력 변수를 변경하고 Execute(기동)를 다시 TRUE로 하면 이 명령의 동작을 변경할 수 있습니다.
모션 명령 재기동으로 변경 가능한 입력 변수는 Distance(이동 거리), Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)입니다.
재기동 시 Distance(이동 거리)의 기점은 재기동 시의 명령 현재 위치가 아니라 최초로 이 명령이 기동된 명령 현재 위치가 됩니다.
Distance(이동 거리)와 Velocity(목표 속도)를 변경하면서 모션 명령을 2회 재기동한 경우의 동작 예는 다음과 같습니다.



사용상의 주의 사항

Distance(이동 거리) 이외의 입력 파라메터를 변경하려는 경우라도 Distance(이동 거리)에는 원래의 명령과 같은 값을 대입해서 명령을 재기동하십시오.
예를 들어 Distance(이동 거리)를 '0'으로 변경하여 명령을 재기동하면 처음 기동할 때부터 Distance(이동 거리)를 '0'으로 해서 명령이 재기동됩니다. 따라서 아래 그림과 같이 원래의 시작 위치로 되돌아갑니다.



모션 명령 재기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다.
각 축에 1개까지 버퍼링할 수 있습니다.
모션 명령 다중 기동 시 Distance(이동 거리)의 기점은 이 명령이 기동되고 Active(제어 중)가 TRUE가 되었을 때의 명령 현재 위치가 됩니다.
다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택		설 명
Aborting		현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.
Buffered		현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.
Blending		현재 실행 중인 명령이 목표 위치에 도달했을 때의 속도(중계 속도)를 기동 속도로 버퍼링 중인 이 명령을 연속해서 동작하게 합니다. 중계 속도로 목표 위치에 도달하도록 현재 실행 중인 명령의 동작을 변경합니다. 중계 속도 지정 방법은 다음의 4종류가 있습니다.
	BlendingLow	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 낮은 쪽을 중계 속도로 합니다.
	BlendingPrevious	현재 실행 중인 목표 속도를 중계 속도로 합니다.
	BlendingNext	버퍼링되고 있는 이 명령의 목표 속도를 중계 속도로 합니다.
	BlendingHigh	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 높은 쪽을 중계 속도로 합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

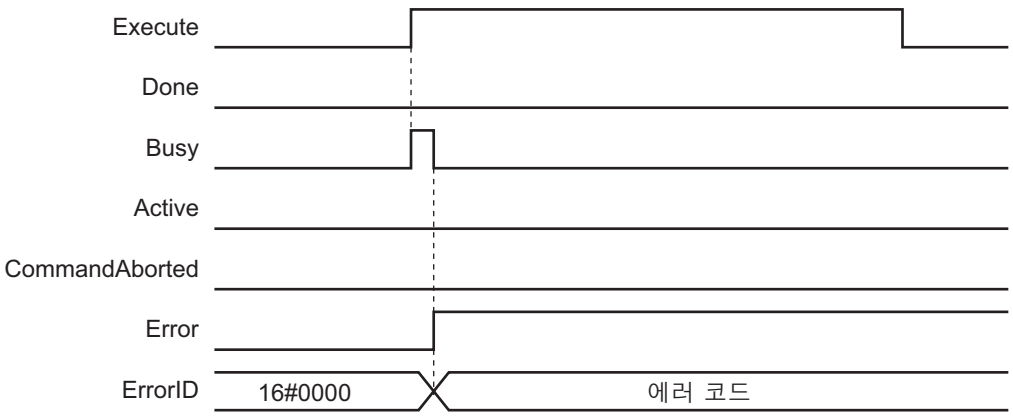
● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하는 경우 Aborting, Buffered, Blending이 가능합니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_MoveVelocity

서보 드라이버의 위치 제어 모드를 사용해 의사 속도 제어를 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_MoveVelocity	속도 제어	FB	MC_MoveVelocity_instance MC_MoveVelocity Axis — — Axis Execute — InVelocity Velocity — Busy Acceleration — Active Deceleration — CommandAborted Jerk — Error Direction — ErrorID Continuous BufferMode	MC_MoveVelocity_instance (Axis := 《파라메터》, Execute := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, Direction := 《파라메터》, Continuous := 《파라메터》, BufferMode := 《파라메터》, InVelocity => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Velocity	목표 속도	LREAL	양수 또는 '0'	0	목표 속도를 지정합니다. 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1
Direction	방향 선택	_eMC_ DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 3: _mcCurrentDirection	0 *2	회전 방향을 지정합니다. 0 : 정방향 지정 2 : 역방향 지정 3 : 현재 방향 지정
BufferMode	버퍼 모드 선택	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 *2	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered
Continuous (Reserved)	계속 방법 선택	BOOL	TRUE, FALSE	FALSE	(Reserved)

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
InVelocity	목표 속도 도달	BOOL	TRUE, FALSE	목표 속도에 도달했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
InVelocity	목표 속도에 도달했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때 • 명령을 재기동해 목표 속도를 변경했을 때
Busy	Execute의 기동	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting, Buffered)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 위치 제어에 의한 의사 속도 제어를 실행합니다.
- Execute(기동)가 기동하면 속도 제어 동작을 시작합니다.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● Direction(방향 선택)

- Direction(방향 선택)으로 이동할 방향을 지정합니다.
- Direction(방향 선택)이 '정방향 지정'인 경우는 정방향으로, '역방향 지정'인 경우는 역방향으로 이동합니다.
- Direction(방향 선택)이 '현재 방향 지정'인 경우 축의 정지 여부에 따라 동작이 달라집니다. 축이 정지하고 있을 경우 이전의 이동 방향으로 축이 이동합니다. 전원을 켜거나 재시작하면 정방향으로 이동합니다.

  모션 명령 다중 기동으로 축이 이동하는 도중에 이 명령을 기동하면 현재 이동하고 있는 방향으로 이동합니다.

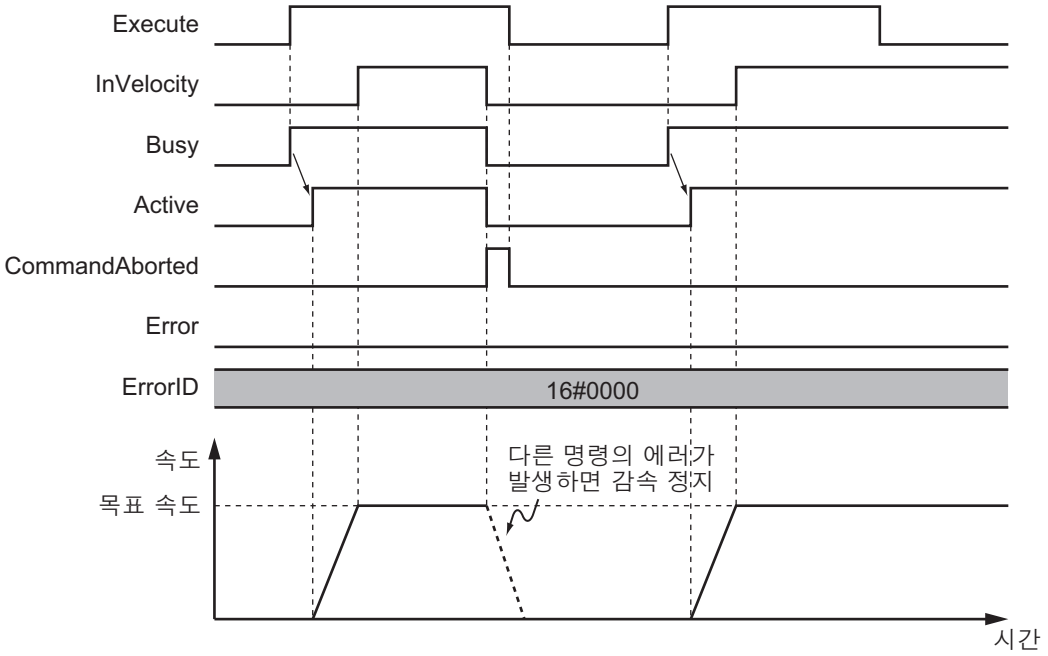


사용상의 주의 사항

Direction(방향 선택)에 '3: _mcCurrentDirection(현재 방향 지정)'을 선택하면 이전 동작의 명령 방향을 학습해서 동작합니다. 따라서 명령의 조합에 따라서는 이전 동작의 모션 명령 입력으로 명령한 방향과 일치하지 않는 경우가 있습니다.
'3: _mcCurrentDirection(현재 방향 지정)'을 사용하는 경우는 축 변수의 Dir. Posi(정방향 명령 중) 및 Dir. Nega(역방향 명령 중)에서 현재 방향을 확인하십시오.

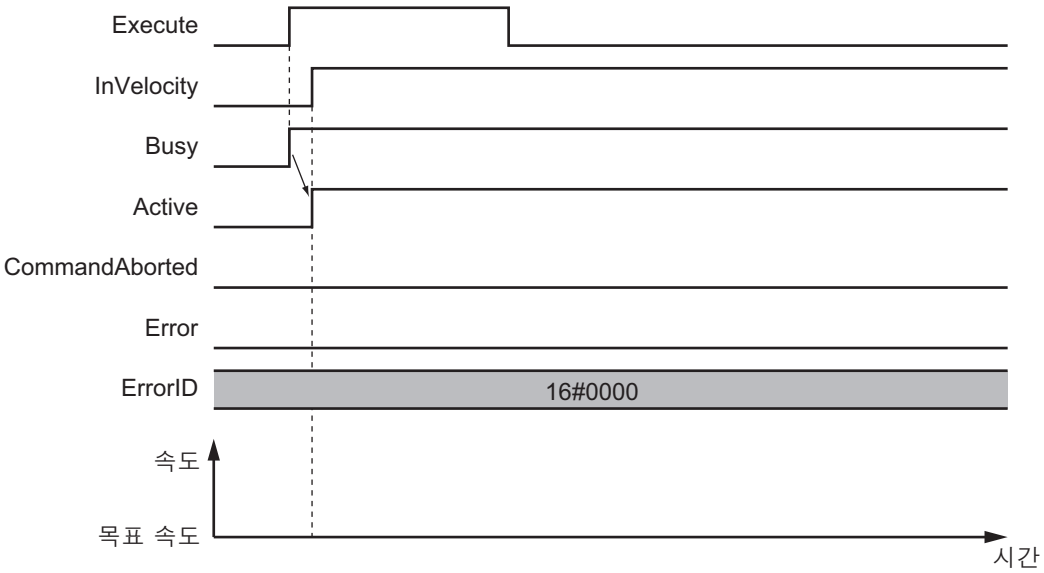
타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- Velocity(목표 속도)에 도달하면 InVelocity(목표 속도 도달)가 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중지되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중), InVelocity(목표 속도 도달)가 FALSE가 됩니다.

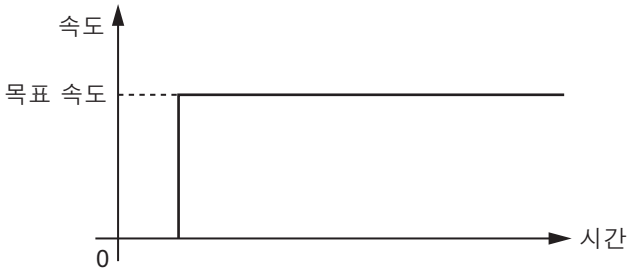


InVelocity(목표 속도 도달)는 이 명령의 기동 및 모션 명령 재기동에 대해 등속으로의 도달을 나타내는 출력입니다. 따라서 InVelocity(목표 속도 도달)가 TRUE가 된 후에 오버라이드를 통해 속도를 변경해도 InVelocity(목표 속도 도달)는 FALSE가 되지 않습니다.
또한 InVelocity(목표 속도 도달)가 TRUE가 되기 전에 오버라이드가 변경된 경우는 변경된 목표 속도에 도달했을 때 InVelocity(목표 속도 도달)가 TRUE가 됩니다.

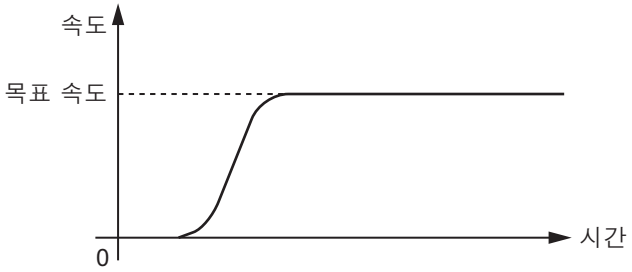
입력 변수에 Acceleration(가속도), Deceleration(감속도), Jerk(저크)를 지정할 수 있습니다.
Velocity(목표 속도)를 '0'으로 지정해서 기동하면 축은 이동하지 않지만 연속 동작 중이 됩니다.
Velocity(목표 속도)를 '0'으로 해서 기동한 경우의 동작 예는 다음과 같습니다.



가속도 또는 감속도를 '0'으로 지정해서 기동하면 가감속하지 않고 목표 속도에 도달합니다.
가속도가 '0'인 경우의 동작 예는 다음과 같습니다.



완만하게 가감속하게 하려면 Jerk(저크)를 지정합니다.
Jerk(저크)를 지정한 경우의 동작 예는 다음과 같습니다.



Jerk(저크)에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

모션 명령 재기동

연속 동작 중에 입력 파라메터를 변경하고 Execute(기동)를 다시 TRUE로 하면 이 명령의 동작을 변경할 수 있습니다.
모션 명령 재기동으로 변경 가능한 입력 변수는 Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)입니다.
모션 명령 재기동을 통해 Velocity(목표 속도)를 변경하는 경우 InVelocity(목표 속도 도달)는 재기동에 의해 설정된 새로운 목표 속도에 대해 동작합니다.

모션 명령 재기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다.
각 축에 1개까지 버퍼링할 수 있습니다.
다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.
Buffered	현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

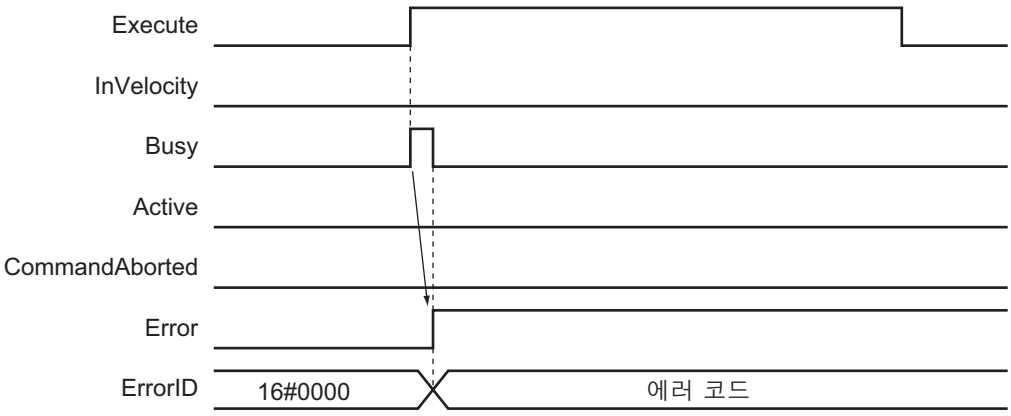
● 이 명령 실행 중의 다른 명령 기동

이 명령 실행 중에 다른 명령에 의한 모션 명령 다중 기동 실행은 다른 명령의 BufferMode(버퍼 모드 선택)에 Aborting과 Buffered를 지정했을 때만 가능합니다.
Bufferd를 지정한 경우 이 명령의 출력 변수 'InVelocity(목표 속도 도달)'가 TRUE가 되었을 때 다중 기동된 명령이 동작하기 시작합니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기에는 스피너나 원심분리기와 같은 속도 제어에 대한 샘플 프로그램이 실려 있습니다.

파라메터 설정

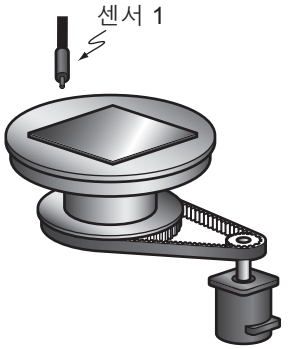
이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

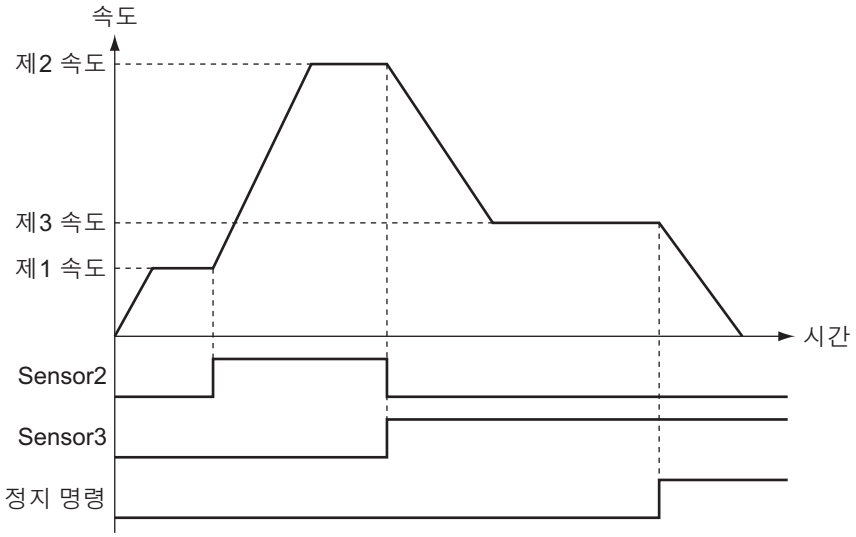
축 종류

축	축 종류
축 1	서보 축

동작 예



● 동작 패턴



1 **속도 제어 시작**
약액 투입을 검출하는 센서 1이 'ON'하면 축 1은 속도 제어를 시작합니다.

2 **제2 속도로 변경**
Sensor2(접점)가 'ON'하면 오버라이드 500%가 설정되고 속도가 변경됩니다.

3 **제3 속도로 변경**
Sensor3(접점)이 'ON'하면 오버라이드 200%가 설정되고 속도가 변경됩니다.
또한 Sensor2와 Sensor3이 동시에 'ON'하면 오버라이드 200%가 됩니다.

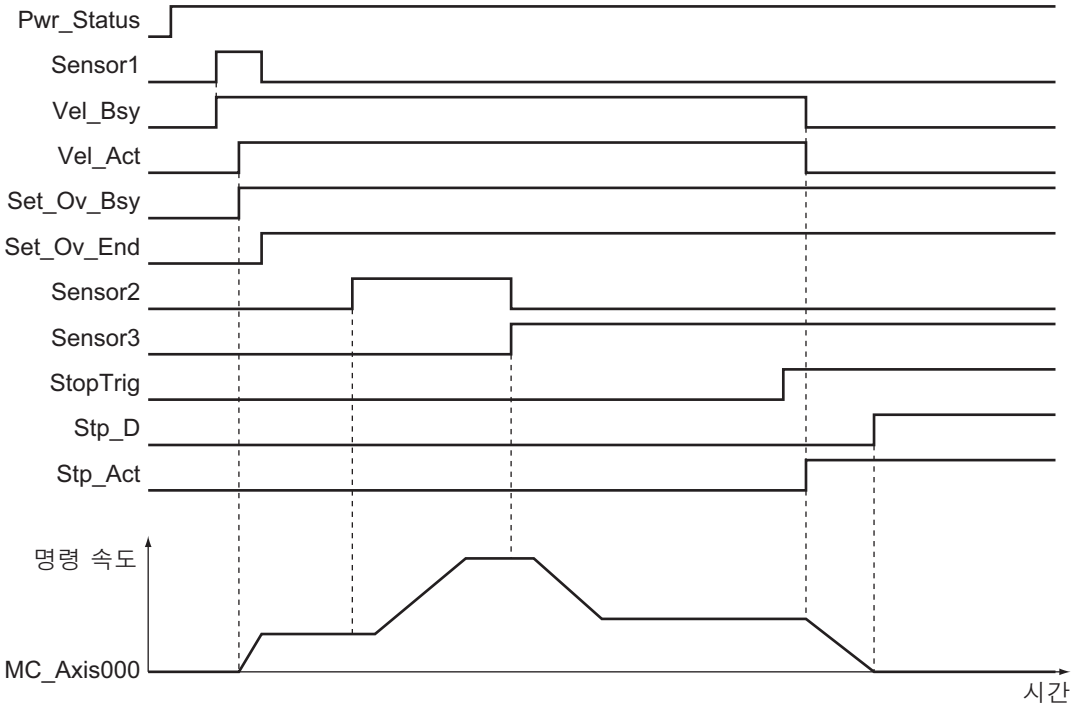
4 **속도 제어의 정지**
정지 명령(StopTrig)이 'ON'하면 감속 정지합니다.

래더도

● 주요 변수

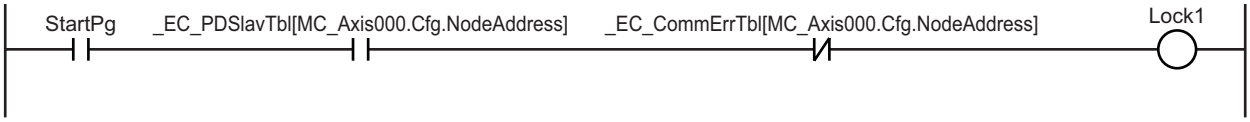
이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Vel_Act	BOOL	FALSE	MC_MoveVelocity 인스턴스 VEL의 출력 Active에 할당한 변수입니다. 인스턴스 VEL이 제어 중이 되면 이 변수가 TRUE가 됩니다.
Set_Ov_Velfct	LREAL	0	오버라이드값입니다.
StopTrig	BOOL	FALSE	이 변수가 TRUE일 때 MC_Stop을 실행합니다.
Sensor1	BOOL	FALSE	약액 투입을 검출하면 TRUE가 됩니다. 축 1이 서보 ON 상태일 때 MC_MoveVelocity(속도 제어)를 기동합니다.
Sensor2	BOOL	FALSE	이 변수가 TRUE일 때 오버라이드값을 500%로 설정합니다. 이 변수는 한 번 TRUE가 되면 상태를 유지하며, Sensor3이 TRUE가 되면 FALSE가 됩니다.
Sensor3	BOOL	FALSE	이 변수가 TRUE일 때 오버라이드값을 200%로 설정합니다. 이 변수는 한 번 TRUE가 되면 상태를 유지합니다.

● 타이밍 차트

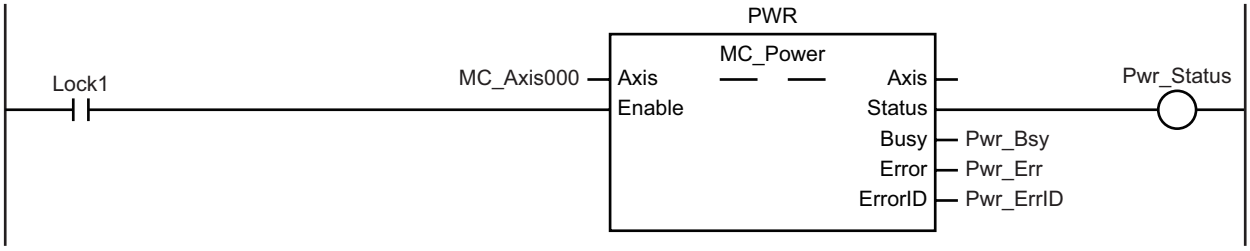


● 샘플 프로그램

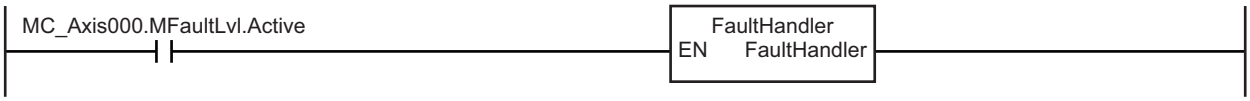
접점 StartPg가 TRUE가 되면 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인



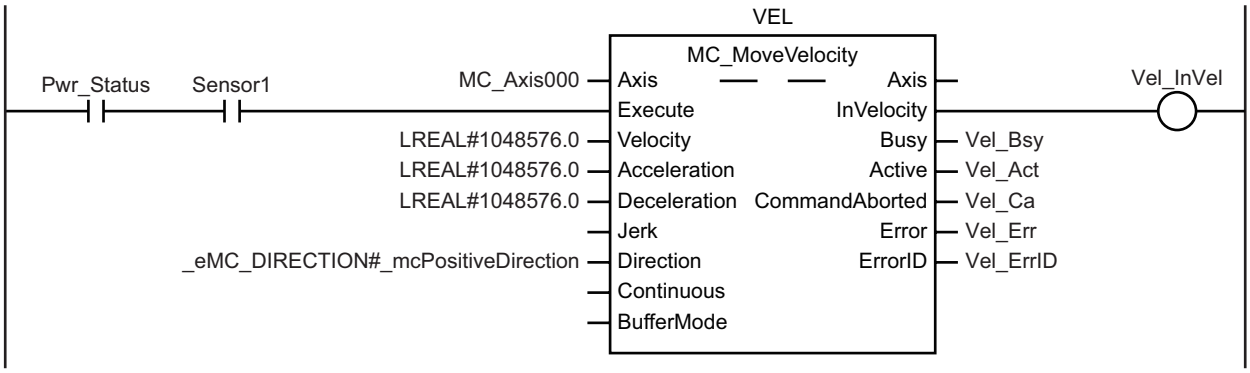
프로세스 데이터 통신을 하고 있으면 서보를 ON



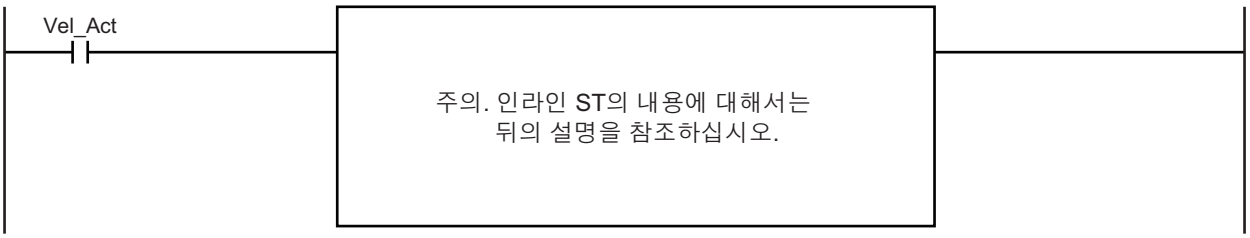
축 1에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



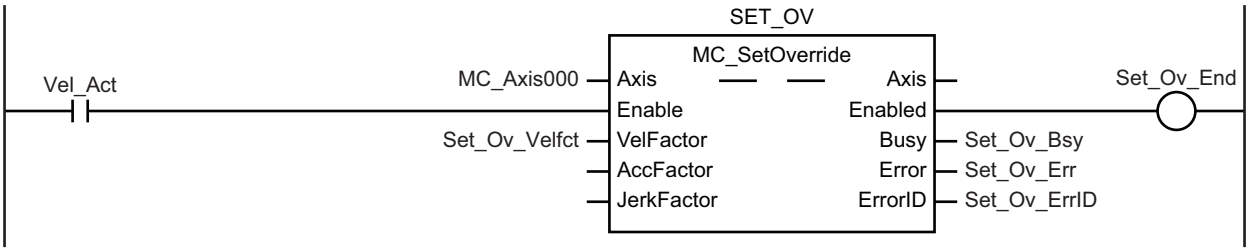
운전 가능 상태에서 Sensor1이 TRUE일 때 MC_MoveVelocity(속도 제어) 명령을 기동



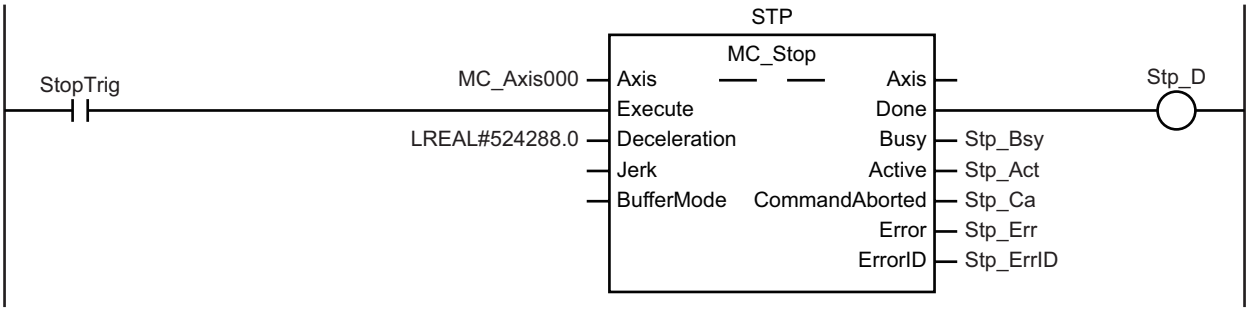
MC_MoveVelocity(속도 제어) 명령 제어 중에 접점 Sensor2와 Sensor3의 ON/OFF에 따라 오버라이드값을 변경



MC_MoveVelocity(속도 제어) 명령 제어 중에는 MC_SetOverride(오버라이드값 설정) 명령을 기동



정지 명령(StopTrig)이 TRUE가 되면 MC_Stop(강제 정지) 명령을 기동



인라인 ST의 내용

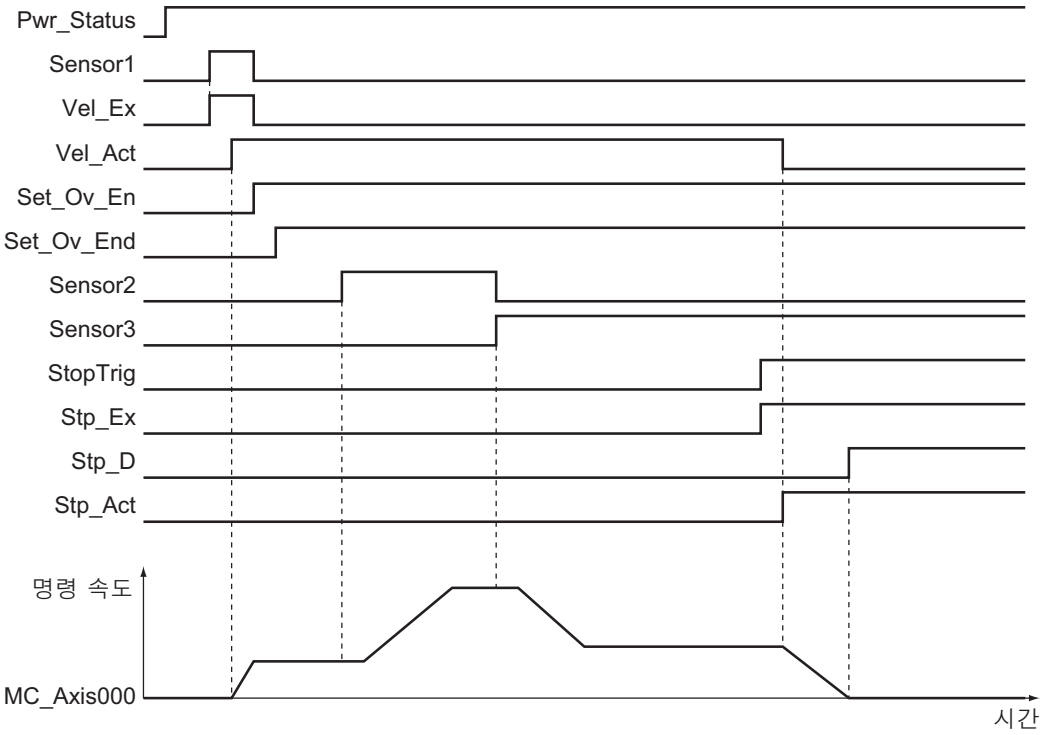
```
IF (Sensor2=FALSE) AND (Sensor3=FALSE) THEN
        Set_Ov_Velfct := LREAL#100.0;
ELSIF (Sensor2=TRUE) AND (Sensor3=FALSE) THEN
        Set_Ov_Velfct := LREAL#500.0;
ELSIF (Sensor2=FALSE) AND (Sensor3=TRUE) THEN
        Set_Ov_Velfct := LREAL#200.0;
ELSE
        Set_Ov_Velfct := LREAL#200.0;
END_IF;
```

스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Vel_Act	BOOL	FALSE	MC_MoveVelocity 인스턴스 VEL의 출력 Active에 할당한 변수입니다. 인스턴스 VEL이 제어 중이 되면 이 변수가 TRUE가 됩니다.
Set_Ov_Velfct	LREAL	0	오버라이드값입니다.
StopTrig	BOOL	FALSE	이 변수가 TRUE일 때 MC_Stop을 실행합니다.
Sensor1	BOOL	FALSE	약액 투입을 검출하면 TRUE가 됩니다. 축 1이 서보 ON 상태일 때 MC_MoveVelocity(속도 제어) 명령을 기동합니다.
Sensor2	BOOL	FALSE	이 변수가 TRUE일 때 오버라이드값을 500%로 설정합니다. 이 변수는 한 번 TRUE가 되면 상태를 유지하며, Sensor3이 TRUE가 되면 FALSE가 됩니다.
Sensor3	BOOL	FALSE	이 변수가 TRUE일 때 오버라이드값을 200%로 설정합니다. 이 변수는 일단 TRUE가 되면 상태를 유지합니다.
Vel_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveVelocity의 인스턴스 VEL을 기동합니다.
Set_Ov_En	BOOL	FALSE	이 변수가 TRUE인 동안 MC_SetOverride의 인스턴스 SET_OV를 실행합니다.
Stp_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_Stop의 인스턴스 STP를 기동합니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트



● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag = FALSE THEN
    //MC_MoveVelocity 파라메터
    Vel_Vel    := LREAL#1048576.0;
    Vel_Acc    := LREAL#1048576.0;
    Vel_Dec    := LREAL#1048576.0;
    Vel_Dir    := _eMC_DIRECTION#_mcPositiveDirection;

    //MC_SetOverride 파라메터
    Set_Ov_Velfct := LREAL#100.0;

    //MC_Stop 파라메터
    Stp_Dec := LREAL#524288.0;


    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;

//접점 StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 1을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//축1 에서 가벼운 이상이 발생하면 이상 시 처리 FaultHandler를 실행합니다.
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
```

```
    FaultHandler();
END_IF;
//축 1이 서보 ON 상태이고 접점 Sensor 1이 TRUE일 때 MC_MoveVelocity를 기동합니다.
IF (Pwr_Status=TRUE) AND (Sensor1=TRUE) THEN
    Vel_Ex := TRUE;
END_IF;

// MC_MoveVelocity 제어 중에 접점 Sensor2와 Sensor3의 ON/OFF에 따라 오버라이드값을 변경합
니다.
IF Vel_Act=TRUE THEN
    IF (Sensor2=FALSE) AND (Sensor3=FALSE) THEN
        Set_Ov_Velfct := LREAL#100.0;
    ELSIF (Sensor2=TRUE) AND (Sensor3=FALSE) THEN
        Set_Ov_Velfct := LREAL#500.0;
    ELSIF (Sensor2=FALSE) AND (Sensor3=TRUE) THEN
        Set_Ov_Velfct := LREAL#200.0;
    ELSE
        Set_Ov_Velfct := LREAL#200.0;
    END_IF;
END_IF;

// MC_MoveVelocity 제어 중에는 MC_SetOverride를 기동합니다.
IF Vel_Act=TRUE THEN
    Set_Ov_En := TRUE;
END_IF;

// StopTrig가 TRUE이면 MC_Stop을 기동합니다.
IF StopTrig=TRUE THEN
    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_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_SetOverride
SET_OV(
    Axis            := MC_Axis000,
    Enable          := Set_Ov_En,
    VelFactor       := Set_Ov_Velfct,
    AccFactor       := Set_Ov_Accfct,
```

```
    JerkFactor      := Set_Ov_Jfct,
    Busy            => Set_Ov_Bsy,
    Enabled         => Set_Ov_End,
    Error           => Set_Ov_Err,
    ErrorID         => Set_Ov_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_MoveZeroPosition

절대 좌표의 '0'을 목표 위치로 위치 결정을 실행하고 원점으로 복귀합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_MoveZeroPosition	고속 원점 복귀	FB	MC_MoveZeroPosition_instance MC_MoveZeroPosition Axis — — Axis Execute — Done Velocity — Busy Acceleration — Active Deceleration — CommandAborted Jerk — Error BufferMode — ErrorID	MC_MoveZeroPosition_instance (Axis := 《파라메터》, Execute := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, BufferMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Velocity	목표 속도	LREAL	양수	0	목표 속도를 지정합니다. *1 단위는 [명령 단위/s]입니다. *2
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *2
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *2
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *2
BufferMode	버퍼 모드 선택	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered

*1. 반드시 목표 속도를 설정하십시오. 설정하지 않고 동작하게 하면 이상이 발생합니다.
*2. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*3. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	위치 결정이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	축이 이동을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 원점 위치로 이동합니다.
- 입력 변수에 Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도), Jerk(저크)를 지정할 수 있습니다.

사용상의 주의 사항

MC_MoveZeroPosition(고속 원점 복귀) 명령은 원점이 확정된 상태에서 실행하십시오. 원점이 확정되지 않으면 '원점 미확정 상태에서의 명령 기동 이상(에러 코드: 5466 Hex)'이 발생합니다.

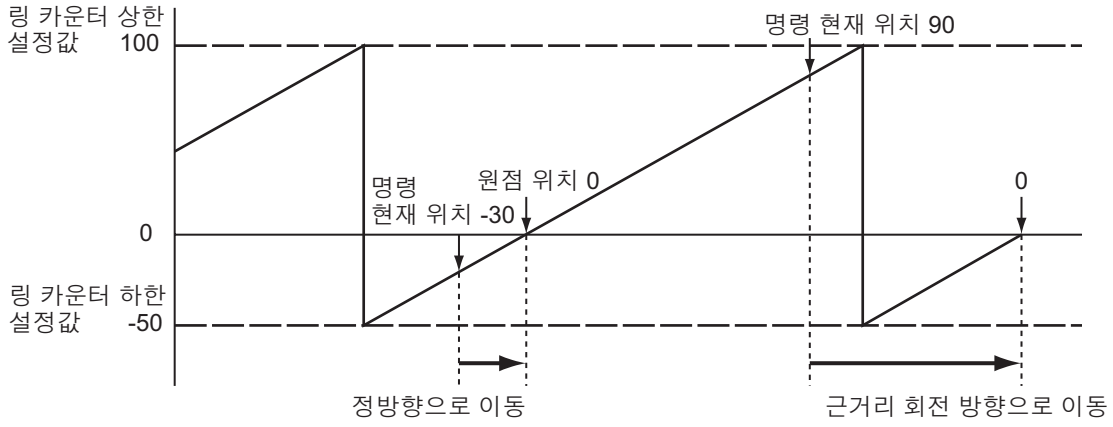
명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● 방향 지정

카운트 모드가 [로터리 모드]이면 원점 위치에 대한 거리가 짧은 방향을 향해 위치가 결정(근거리 회전)됩니다.

명령 현재 위치에 따른 이 명령의 동작 예는 다음과 같습니다.



사용상의 주의 사항

축 파라메터의 [링 카운터 상한 설정값]과 [링 카운터 하한 설정값]의 범위 밖에 원점이 존재하는 상태로 이 명령을 기동하면 '목표 위치 링 카운터 범위 외(에러 코드: 549C Hex)'가 됩니다.

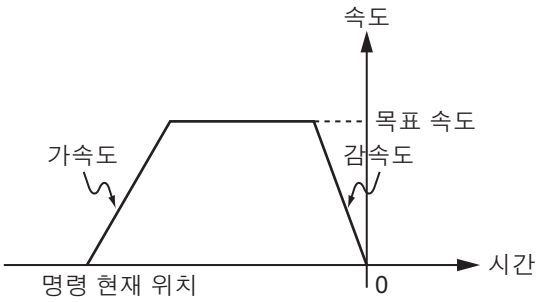
● 오버라이드

이 명령에 대해 오버라이드가 유효합니다.

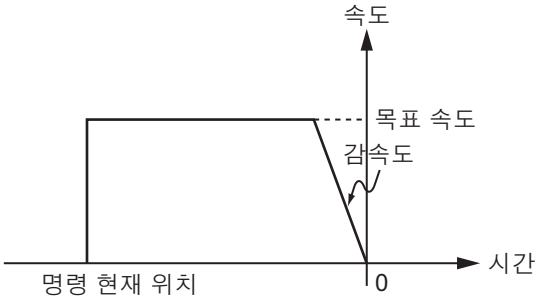
● 인포지션 확인

이 명령의 인포지션 확인은 축 파라메터의 [인포지션 폭], [인포지션 확인 시간] 설정에 따라 실행됩니다.

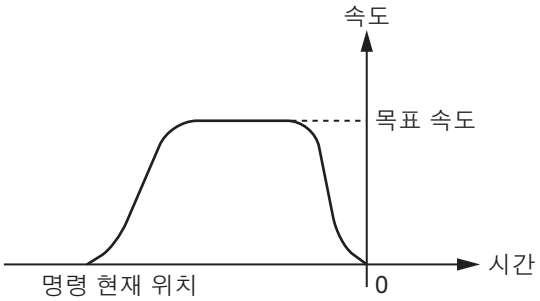
● 동작 예



가속도 또는 감속도를 '0'으로 지정해서 기동하면 가감속하지 않고 목표 속도에 도달합니다.
가속도가 '0'인 경우의 동작 예는 다음과 같습니다.



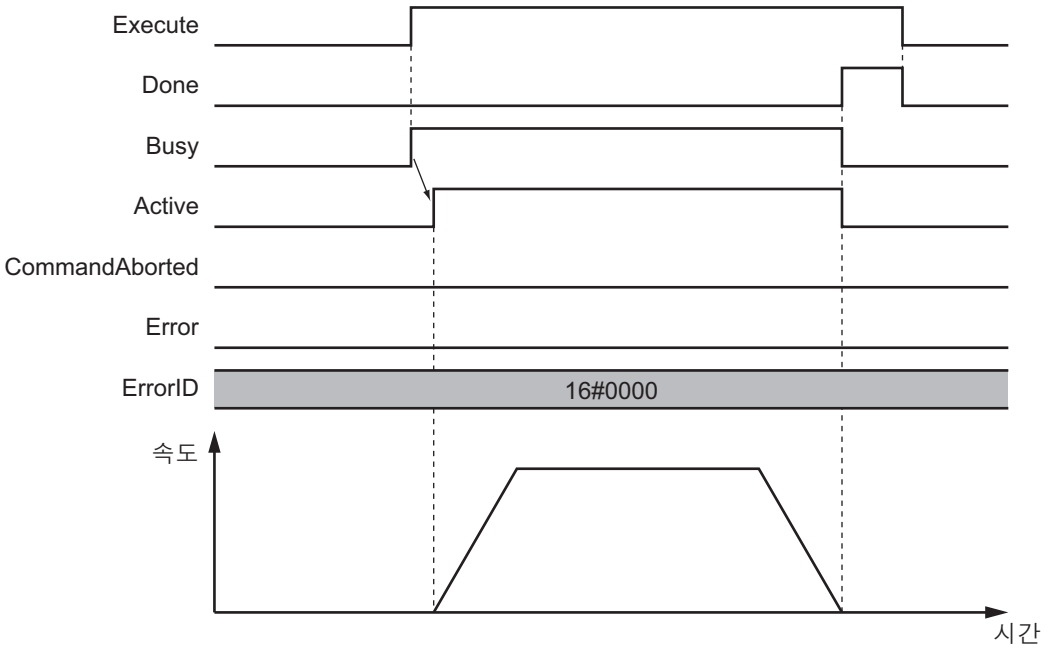
완만하게 가감속하게 하려면 Jerk(저크)를 지정합니다.
Jerk(저크)를 지정한 경우의 동작 예는 다음과 같습니다.



Jerk(저크)에 대한 자세한 내용은 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

타이밍 차트

MC_MoveZeroPosition(고속 원점 복귀) 명령을 실행한 경우의 타이밍 차트는 다음과 같습니다.



명령 중지

이 명령이 중지되고 CommandAborted(실행 중단)가 TRUE가 되어도 원점 미확정 상태가 되지는 않습니다.

모션 명령 재기동

위치 결정 동작 중에 입력 변수를 변경하고 Execute(기동)를 다시 TRUE로 하면 이 명령의 움직임을 변경할 수 있습니다.
모션 명령 재기동으로 변경 가능한 입력 변수는 Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)입니다.

모션 명령 재기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다.
각 축에 1개까지 버퍼링할 수 있습니다.
다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.
Buffered	현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.

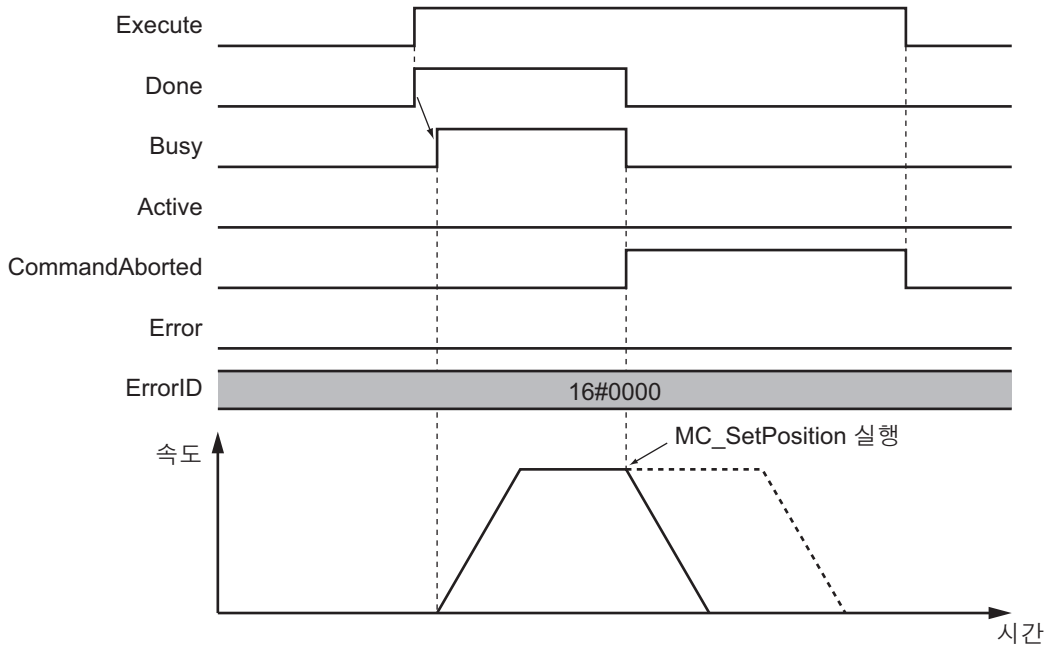
BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하는 경우 Aborting, Buffered가 Blending 가능합니다.
Blending에 의한 명령 다중 기동은 불가능합니다.

이 명령 실행 중에 MC_SetPosition(현재 위치 변경) 명령을 실행한 경우

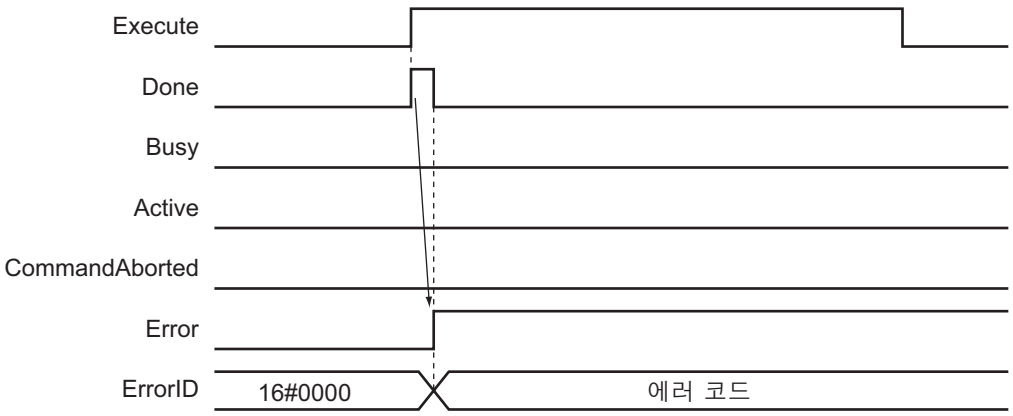
MC_SetPosition(현재 위치 변경) 명령에서는 원점 미확정이 되므로 이 명령 실행 중에는 이상이 되어 실행할 수 없습니다.
MC_SetPosition(현재 위치 변경) 명령 실행에 의해 이상 상태가 되면 이 명령은 감속 정지해 CommandAborted(실행 중단)가 TRUE가 됩니다.
이 때 MC_SetPosition(현재 위치 변경) 명령에서는 출력 변수의 Error(에러)가 TRUE가 됩니다.



이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트

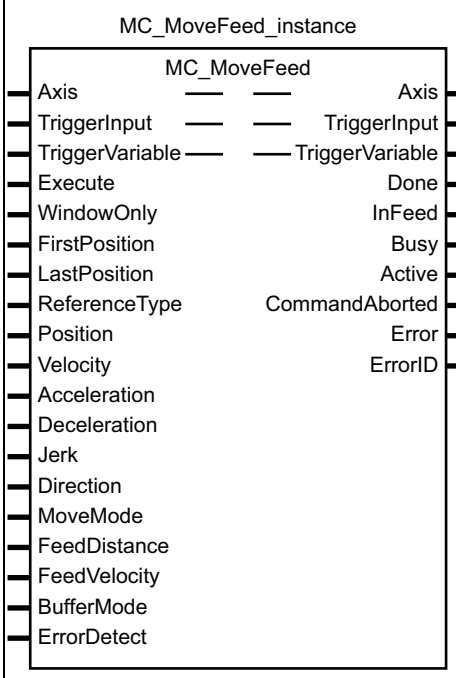


● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_MoveFeed

외부 입력에 의해 인터럽트 입력이 발생한 위치에서의 이동 거리를 지정하고 위치 결정을 실행합니다.
인터럽트 피딩 위치 결정이 가능한 제어는 절대값 위치 결정, 상대값 위치 결정, 속도 제어입니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_MoveFeed	인터럽트 피딩 위치 결정	FB		MC_MoveFeed_instance (Axis := 《파라메터》, TriggerInput := 《파라메터》, TriggerVariable := 《파라메터》, Execute := 《파라메터》, WindowOnly := 《파라메터》, FirstPosition := 《파라메터》, LastPosition := 《파라메터》, ReferenceType := 《파라메터》, Position := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, Direction := 《파라메터》, MoveMode := 《파라메터》, FeedDistance := 《파라메터》, FeedVelocity := 《파라메터》, BufferMode := 《파라메터》, ErrorDetect := 《파라메터》, Done => 《파라메터》, InFeed => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
WindowOnly	창 유효	BOOL	TRUE, FALSE	FALSE	창의 유효/무효를 지정합니다.
FirstPosition	시작 위치	LREAL	음수, 양수, '0'	0	래치가 유효가 되는 위치를 지정합니다. 단위는 [명령 단위]입니다. *1
LastPosition	종료 위치	LREAL	음수, 양수, '0'	0	래치가 무효가 되는 위치를 지정합니다. 단위는 [명령 단위]입니다. *1
Reference Type	위치 종류 선택	_eMC_ REFERENCE_ TYPE	1: _mcFeedback	1 *2	위치 종류를 지정합니다. 1: 피드백 위치(동일 프라이머리 주기에서의 취득값)
Position	목표 위치	LREAL	음수, 양수, '0'	0	절대 좌표의 목표 위치를 지정합니다. 단위는 [명령 단위]입니다.*1
Velocity	목표 속도	LREAL	양수*3	0	목표 속도를 지정합니다. 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1
Direction	방향 선택	_eMC_ DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	0 *2	MoveMode가 [0: 절대값 위치 결정]*4 이고 카운트 모드가 [로터리 모드]인 경우에 회전 방향을 지정합니다. 0 : 정방향 지정 1 : 근거리 회전 지정 2 : 역방향 지정 3 : 현재 방향 지정 4 : 방향 지정 없음
MoveMode	이동 방법 선택	_eMC_MOVE_ MODE	0: _mcAbsolute 1: _mcRelative 2: _mcVelocity	0 *2	이동 방법을 선택합니다. 0 : 절대값 위치 결정 1 : 상대값 위치 결정 2 : 속도 제어

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
FeedDistance	피딩 거리	LREAL	음수, 양수, '0'	0	인터럽트 입력 후의 이동 거리를 지정합니다. 인터럽트 입력 전의 동작 방향과 같은 방향으로 피딩 동작을 하게 하려면 양수를, 역방향으로 동작하게 하려면 음수를 설정합니다. 단위는 [명령 단위]입니다. *1
FeedVelocity	피딩 속도	LREAL	양수	0	인터럽트 입력 후의 이동 목표 속도를 지정합니다. 단위는 [명령 단위/s]입니다. *1
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 *2	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered
ErrorDetect	에러 검출 선택	BOOL	TRUE, FALSE	FALSE	인터럽트 미입력 시 이상 검출 여부를 지정합니다. TRUE : 이상 검출 있음 FALSE : 이상 검출 없음

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*3. MoveMode에 '2: _mcVelocity'가 지정되어 있으면 '0'도 가능합니다.
*4. MoveMode가 [0: 절대값 위치 결정]이고 카운트 모드가 [리니어 모드]인 경우는 지정된 위치로 이동합니다.
MoveMode가 [1: 상대값 위치 결정]인 경우는 Position의 부호에 따라 이동 방향이 결정됩니다.
MoveMode가 [2: 속도 제어]인 경우는 카운터 모드에 상관 없이 '1: _mcShortestWay' 및 '4: _mcNoDirection'은 선택할 수 없습니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
InFeed	피딩 중	BOOL	TRUE, FALSE	래치 입력을 수신하고 피딩 중에 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
InFeed	인터럽트 입력에 의해 피딩 동작을 시작했을 때*1	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동 (Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

*1. 인터럽트 입력이 'ON'하고 나서 InFeed가 TRUE가 될 때까지 몇 프라이머리 주기 늦는 경우가 있습니다.

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1
TriggerInput	트리거 입력 조건	_sTRIGGER_REF	−	트리거 조건을 설정합니다. *2
TriggerVariable	트리거 변수	BOOL	TRUE, FALSE	트리거 조건에서 컨트롤러 모드를 지정했을 때 트리거가 될 입력 변수를 지정합니다.

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.
*2. _sTRIGGER_REF 형식의 사용자 정의 변수를 정의하십시오.

● _sTRIGGER_REF

멤버 변수	이 름	데이터 형식	유효 범위	기 능
Mode	모드	_eMC_ TRIGGER_ MODE	0: _mcDrive 1: _mcController	트리거 모드를 지정합니다. 0 : 드라이버 모드 1 : 컨트롤러 모드
LatchID	래치 ID 선택	_eMC_ TRIGGER_ LATCHID	0: _mcLatch1 1: _mcLatch2	드라이버 모드일 때 2가지 래치 기능 중 어느 것을 사용할지 지정합니다. 0 : 래치 기능 1 1 : 래치 기능 2
InputDrive	드라이브 트리거 입력 신호	_eMC_ TRIGGER_ INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	드라이버 모드일 때 드라이버의 트리거 신호를 지정합니다. 0 : Z상 1 : 외부 입력

기능 설명

- Execute(기동)가 기동하면 MoveMode(이동 방법 선택)의 설정에 따라 절대값 이동 또는 상대값 이동, 속도 제어 중 한 가지 이동 방법으로 이동합니다.
- 절대값 이동에서는 Position(목표 위치)에 목표 위치를 설정하고, 상대값 이동에서는 Position(목표 위치)에 목표 거리를 설정합니다.
  또한 두 이동 방법 다 Velocity(목표 속도)로 이동 동작을 실행합니다.
- 이동 중에 외부 입력(인터럽트 입력)이 기동하면 피드백 위치에서부터 FeedDistance(피딩 거리)에서 지정한 피딩 거리를 FeedVelocity(피딩 속도)로 상대 위치 결정 동작을 실행합니다.
- 절대값 이동 또는 상대값 이동 명령에 따른 인터럽트 피딩으로 목표 위치까지 인터럽트 신호가 입력되지 않으면 처음의 목표 위치에서 정지합니다.
  인터럽트 입력 없이 정지했을 때 ErrorDetect(에러 검출 선택)를 통해 이상 출력 여부를 지정할 수 있습니다. 이상 출력 지정 시에는 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중)가 FALSE가 됩니다.
- 인터럽트 마스크를 이용하는 경우는 WindowOnly(창 유효)를 TRUE로 하고 FirstPositon(시작 위치), LastPositon(종료 위치)을 지정합니다.
  피드백 위치가 FirstPositon(시작 위치)에서부터 LastPosition(종료 위치) 사이에서 발생하는 최초의 인터럽트 신호로 인터럽트 피딩 위치 결정을 실행합니다.

사용상의 주의 사항

- 인터럽트 후의 피딩 동작은 FeedDistance(피딩 거리)에서 지정한 거리의 상대 이동을 실행합니다. FeedDistance(피딩 거리)에 양수를 지정하면 인터럽트 입력 전과 같은 방향으로 피딩 동작하며, 음수를 지정하면 역방향으로 피딩 동작합니다.
- 피딩 동작으로 반전할 경우의 가감속 속도는 축 파라메터의 [반전 시 동작] 설정에 따릅니다.
- 인터럽트 피딩 위치 결정 후의 위치가 오버 플로우/언더 플로우가 되면 인터럽트 입력 시 이상이 됩니다. 오버 플로우/언더 플로우 발생 후에 인터럽트 입력이 발생한 경우도 동일하게 축 이상이 됩니다.

참고

절대값 이동에 대해서는 🕮 'MC_MoveAbsolute(3-47페이지)',
상대값 이동에 대해서는 🕮 'MC_MoveRelative(3-73페이지)',
속도 제어에 대해서는 🕮 'MC_MoveVelocity(3-80페이지)'
WindowOnly(창 유효)에 대해서는 🕮 'WindowOnly(창 유효) (3-109페이지)'를 참조하십시오.

● 데이터 오브젝트 매핑

Mode에 [드라이버 모드]를 선택하고 MC_MoveFeed(인터럽트 피딩 위치 결정) 명령을 사용하는 경우는 다음의 오브젝트 데이터를 매핑하십시오.

매핑은 Sysmac Studio 축 기본 설정의 [고급 설정]에서 실행합니다.

- 래치 기능(60B8Hex)
- 래치 Status(60B9Hex)
- 래치 위치 1(60BAHex)
- 래치 위치 2(60BCHex)

이러한 데이터를 설정하지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다.

데이터 오브젝트 매핑에 대해서는 📖 '2-3 PDO 매핑(2-27페이지)', 📖 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

NX 시리즈 위치 인터페이스 유니트를 사용하는 경우는 📖 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"의 'I/O 항목 매핑'에 기재된 내용을 참조하십시오.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● Axis(축) 지정

- Axis(축)에서 래치 대상이 되는 축을 지정합니다.

● 트리거 입력 조건

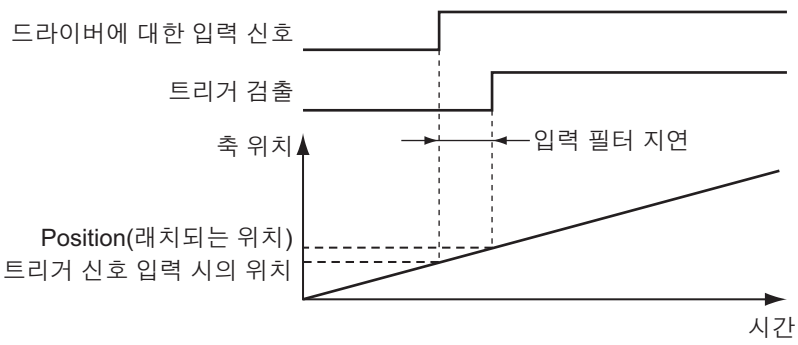
TriggerInput(트리거 입력 조건)의 Mode, LatchID, InputDrive를 지정해 트리거 조건을 선택합니다.

Mode

- Mode 설정에는 트리거에 서보 드라이버 또는 NX 시리즈 펄스 출력 유니트의 신호를 지정하는 '드라이버 모드'와 트리거에 TriggerVariable(트리거 변수)을 지정하는 '컨트롤러 모드'가 있습니다.
- 트리거 신호의 기동을 트리거 발생으로 간주하고, 래치 명령 실행 후 최초의 트리거 발생(FALSE→TRUE) 시 축의 위치를 기록합니다.
- 이 명령이 Busy(동작 중)중에는 Execute(기동)가 FALSE가 되어도 TriggerVariable(트리거 변수)의 변화를 트리거로서 동작합니다.

드라이버 모드

'드라이버 모드'는 서보 드라이버의 기능으로 트리거를 검출하고 현재 위치를 취득하므로 '컨트롤러 모드'에 비해 래치되는 현재 위치가 정확합니다.



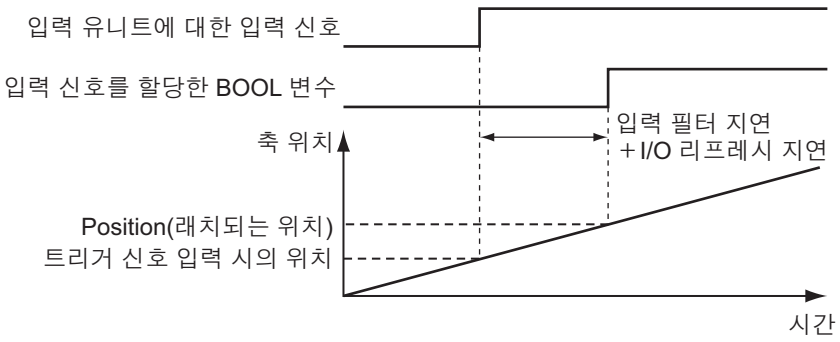
사용상의 주의 사항

- 드라이버 모드를 사용하는 경우는 반드시 사용하는 LatchID(래치 ID 선택)에 래치 신호를 연결하십시오.
- 래치 신호의 폭은 서보 드라이버 또는 NX 시리즈 펄스 출력 유니트의 성능 등에 따릅니다.
- InputDrive에 [드라이버 모드]를 선택하고 MC_MoveFeed(인터럽트 피딩 위치 결정) 명령을 사용하는 경우는 다음의 오브젝트 데이터를 매핑하십시오.

  래치 기능(60B8Hex), 래치 Status(60B9Hex), 래치 위치 1(60BAHex), 래치 위치 2(60BCHex)
  이러한 데이터를 설정하지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다.
  오브젝트 데이터 매핑에 대해서는 '2-3 PDO 매핑(2-27페이지)', "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

컨트롤러 모드

- '컨트롤러 모드'에서는 트리거에 BOOL 변수를 지정할 수 있습니다.
- TriggerVariable(트리거 변수)에 트리거로 하려는 BOOL 변수를 지정합니다.
- '컨트롤러 모드'에서는 트리거 입력 신호가 BOOL 변수에 대해 I/O 리프레시 지연으로 '드라이버 모드'보다도 한참 늦게 반영됩니다.



사용상의 주의 사항

컨트롤러 모드를 사용하면 프라이머리 정주기 태스크의 간격으로 래치 처리를 실행합니다. 따라서 트리거 변수는 프라이머리 정주기 태스크의 1주기 동안 TRUE여야 합니다.
또한 트리거 변수가 TRUE가 된 후 MC 기능 모듈이 처리할 때까지 프라이머리 정주기 태스크 1주기 동안의 시간이 필요합니다.

LatchID

- LatchID(래치 ID 선택)에서 어느 것을 사용할지 지정합니다. 1개의 축에 대해서는 둘 중 하나만 사용할 수 있습니다.
- 서보 드라이버 또는 NX 시리즈 펄스 출력 유니트에서는 LatchID(래치 ID 선택)가 서보 드라이버 또는 NX 시리즈 펄스 출력 유니트의 래치 회로 1(제1ch), 래치 회로 2(제2ch)를 의미합니다.

LatchID에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

InputDrive

- 트리거로 _mcEncoderMark(Z상)와 _mcEXT(외부 입력)를 선택할 수 있습니다.
- _mcEncoderMark(Z상)에서는 서보 드라이버 또는 NX 시리즈 출력 유니트의 Z상을 트리거에 사용합니다.
  _mcEXT(외부 입력)에서는 서보 드라이버 또는 NX 시리즈 출력 유니트에 입력된 외부 트리거 신호를 트리거에 사용합니다.
- 오므론의 서보 드라이버 G5 시리즈에서 _mcEXT(외부입력)은 설정된 Ext1, Ext2, Ext3 중 하나가 됩니다. 설정에는 Sysmac Studio를 사용합니다.
  드라이버에 설정되는 2개의 트리거는 같은 설정이라도 가능합니다.
- NX 시리즈 펄스 출력 유니트를 사용하는 경우는 "NX 시리즈 위치 인터페이스 유니트 사용 설명서 (SBCE-374)"를 참조하십시오.

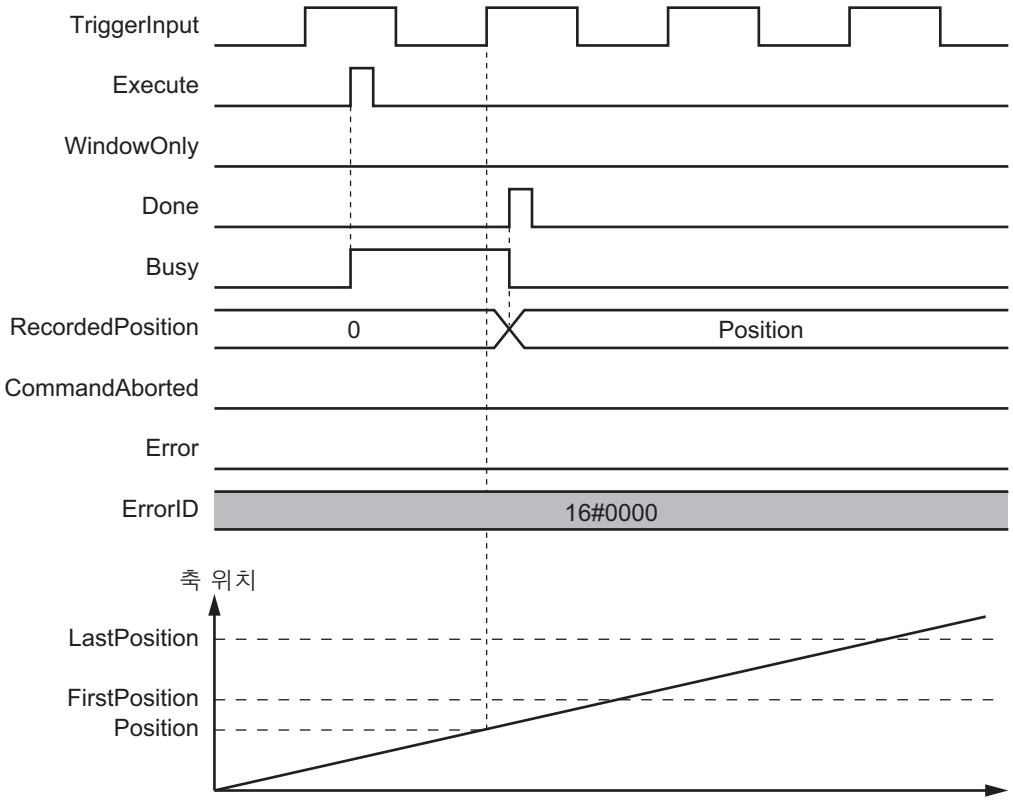
● WindowOnly(창 유효)

- WindowOnly(창 유효)에서는 창의 유효/무효를 지정합니다.
- Disable(무효)을 지정하면 모든 축 위치에서 트리거를 검출합니다.
- Enable(유효)을 지정하면 축 위치가 FirstPosition(시작 위치)과 LastPosition(종료 위치) 범위 내에 있을 때만 트리거를 검출합니다.

WindowOnly(창 유효) 지정에 따른 동작의 차이를 타이밍 차트로 표시합니다.

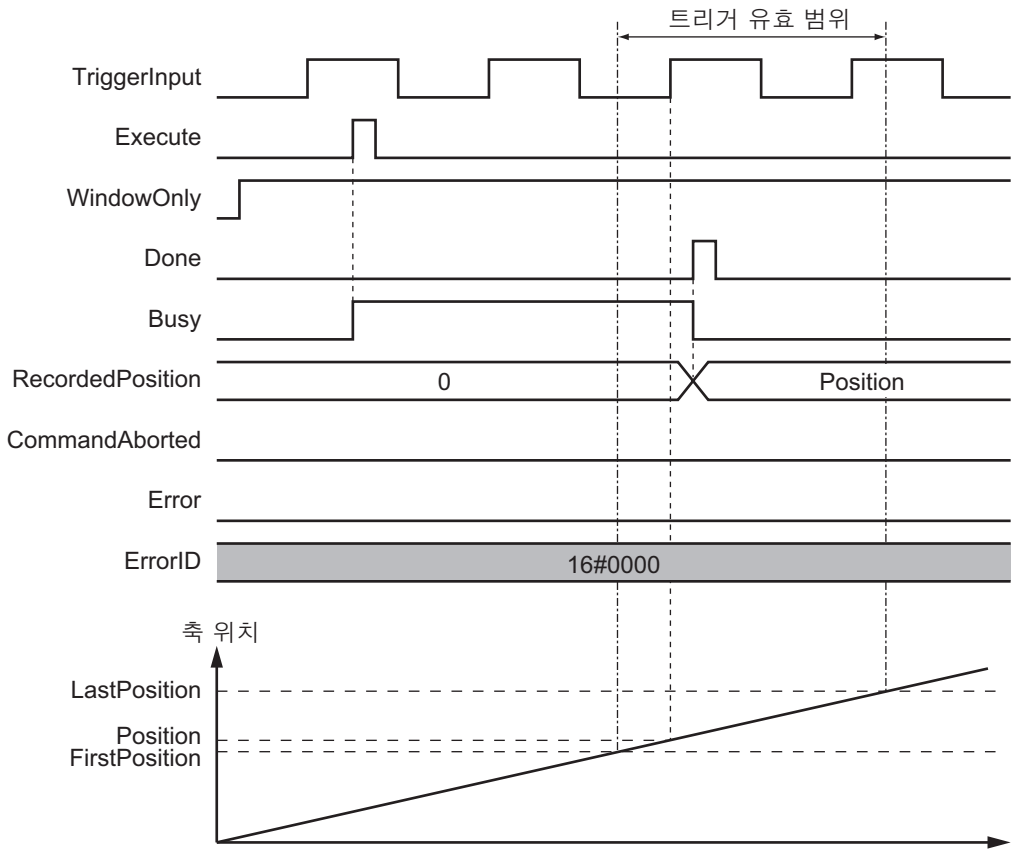
WindowOnly가 Disable인 경우

Execute(기동)가 TRUE가 된 후 최초 트리거 발생 시의 축 위치를 피딩 거리 기준 위치로 합니다.



WindowOnly가 Enable인 경우

창의 범위 내에서만 트리거 입력을 검출해서 축 위치를 취득합니다.



사용상의 주의 사항

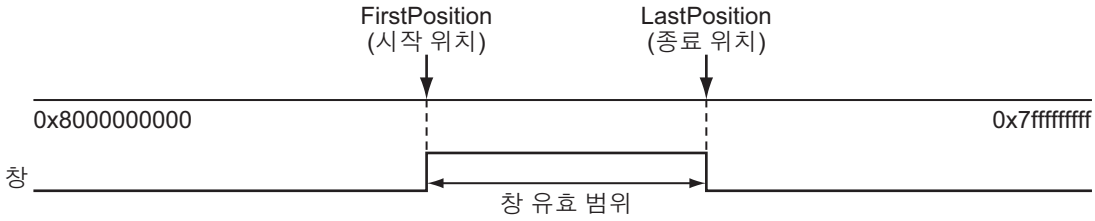
- WindowOnly(창 유효)가 FALSE→TRUE로 변화하는 순간과 래치 기능이 동작할 때까지의 시간은 래치할 수 없습니다.
- 래치 기능이 동작할 때까지 시간이 필요하기 때문에 WindowOnly(창 유효)의 유효 범위가 매우 짧은 경우는 래치할 수 없습니다. 래치 가능한 유효 범위의 한계값은 서보 드라이버, 엔코더 입력 터미널, 위치 인터페이스 등의 성능 및 EtherCAT 통신에 따릅니다.

또한 카운트 모드의 차이에 따라 FirstPosition(시작 위치)과 LastPosition(종료 위치)의 범위는 다음과 같습니다.

리니어 모드

- FirstPosition(시작 위치) ≦ 창 범위 ≦ LastPosition(종료 위치)
- FirstPosition(시작 위치) > LastPosition(종료 위치)으로 지정하면 이상이 됩니다.
- [리니어 모드]의 위치 범위를 넘어서 지정한 경우도 이상이 됩니다.
- 또한 FirstPosition(시작 위치)과 LastPosition(종료 위치)은 LREAL 형식 변수이므로 FirstPosition(시작 위치) = LastPosition(종료 위치)을 지정하지 마십시오.
  LREAL 형식에 대해서는 "실수(REAL 형식, LREAL 형식) 사용 방법(1-14페이지)"을 참조하십시오.

[리니어 모드]일 때 창 유효 범위는 다음 그림과 같습니다.

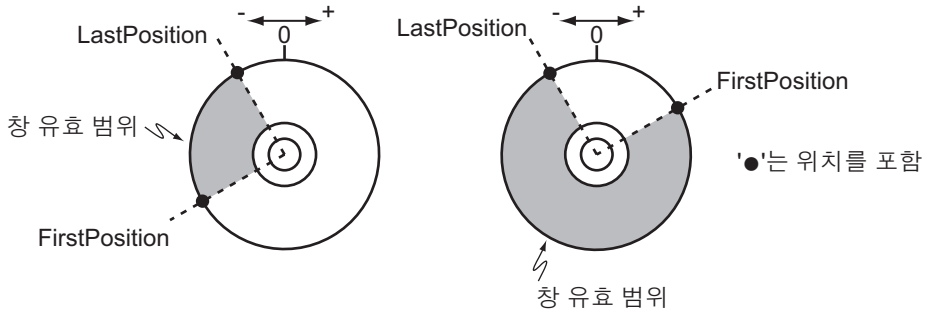


(주의) FirstPosition(시작 위치)과 LastPosition(종료 위치)은 창 유효 범위에 포함됩니다.

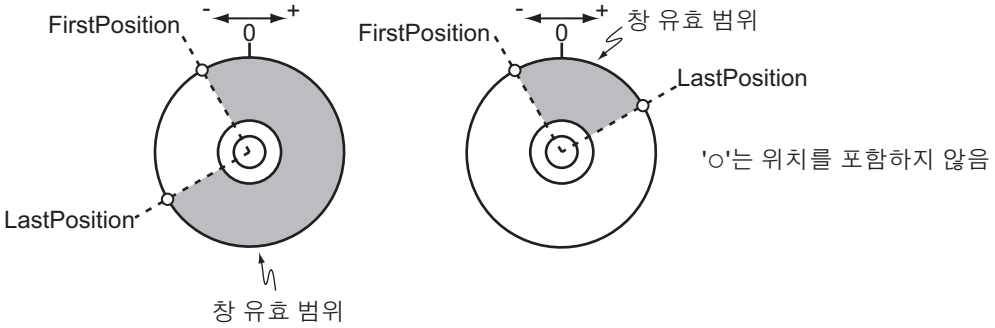
로터리 모드

- FirstPosition(시작 위치) ≦ LastPosition(종료 위치) 및
  FirstPosition(시작 위치) > LastPosition(종료 위치) 둘 다 지정 가능합니다.
- FirstPosition(시작 위치) > LastPosition(종료 위치)인 경우는 링 카운터 상하한 위치에 걸쳐 설정됩니다.
- 링 카운터 상하한 범위를 넘어서 지정하면 이상이 됩니다.

FirstPosition ≦ LastPosition



FirstPosition > LastPosition



● ReferenceType(위치 종류 선택)

위치 종류는 다음과 같습니다.

- _mcFeedback: 동일 프라이머리 주기에서의 취득값
  동일 프라이머리 주기에서 취득한 주축의 피드백 위치를 사용합니다.

● FeedDistance(피딩 거리)

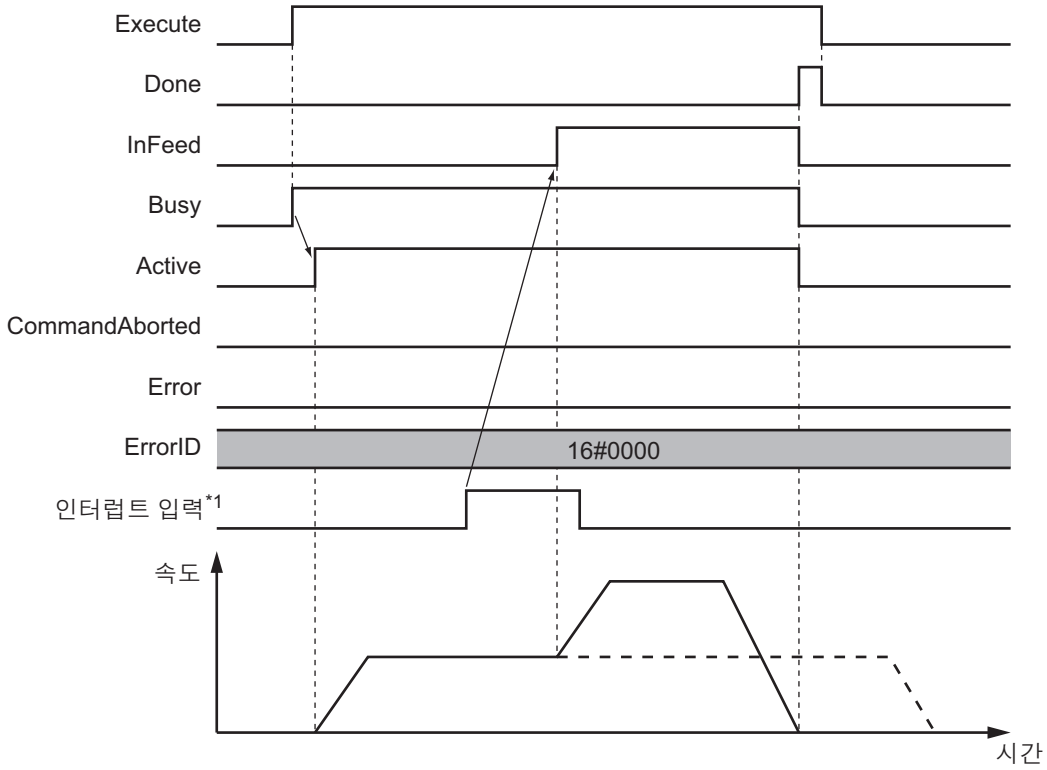
인터럽트 입력 전과 같은 방향으로 피딩 동작을 할 경우 FeedDistance(피딩 거리)에는 양수값을 지정합니다. 인터럽트 입력 전과 반대 방향으로 피딩 동작을 할 경우 FeedDistance(피딩 거리)에는 음수값을 지정합니다.

예를 들면 인터럽트 입력 전 동작이 역방향으로 동작하는 상태에서 FeedDistance(피딩 거리)에 양수값을 지정하면 피딩 동작은 역방향으로 진행됩니다. FeedDistance(피딩 거리)에 음수값을 지정하면 피딩 동작은 정방향으로 진행됩니다.

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- 인터럽트 입력 후 InFeed(피딩 중)가 TRUE가 되어 FeedDistance(피딩 거리)에 도달하고 위치 결정이 완료되면 Done(완료)이 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중단되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중)가 FALSE가 됩니다.

● MoveMode(이동 방법 선택)가 _mcAbsolute, _mcRelative인 경우

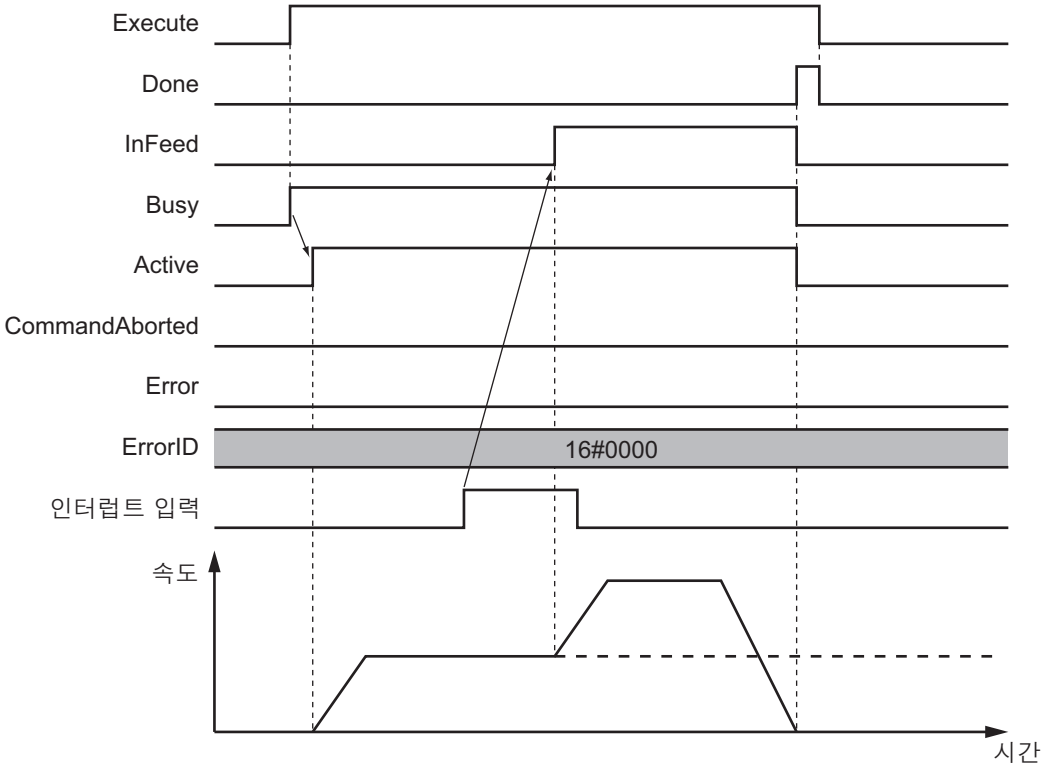


*1 인터럽트 신호가 'ON'이 되고 나서 InFeed가 TRUE가 될 때까지 복수 제어 주기 늦는 경우가 있습니다.

참고

인터럽트 피딩 위치 결정에 대한 가속 및 감속은 입력 변수의 Acceleration(가속도), Deceleration(감속도)에 따릅니다.

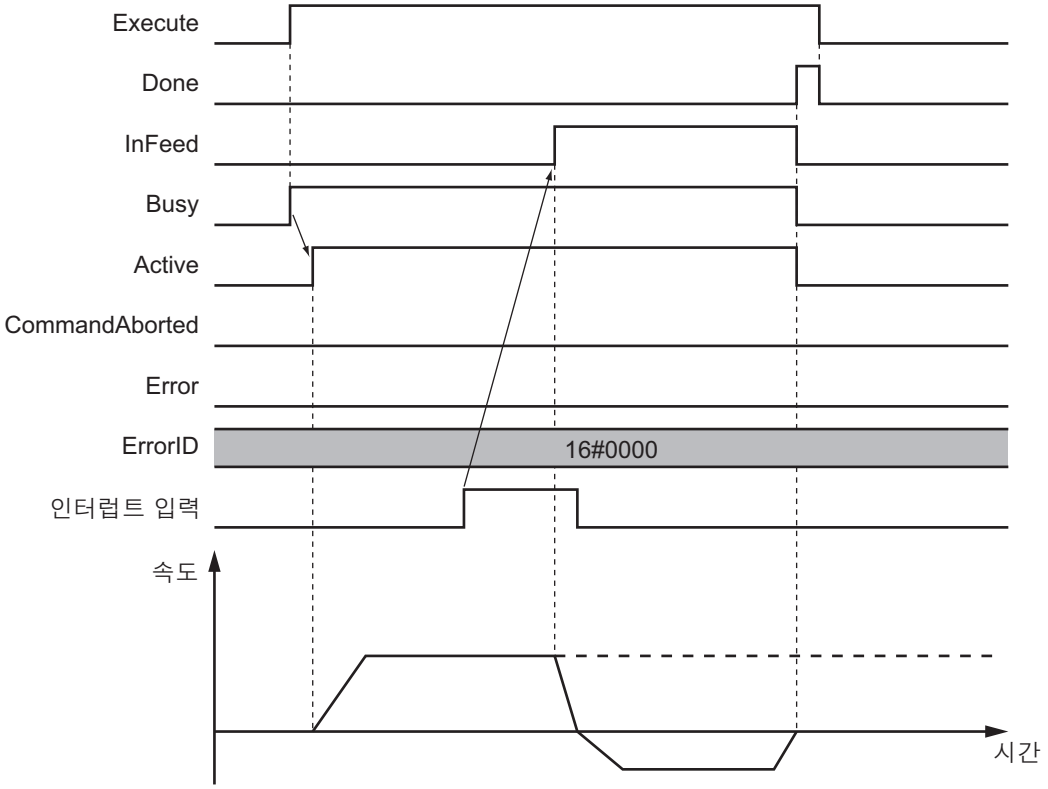
● MoveMode(이동 방법 선택)가 _mcVelocity인 경우



● 축 파라메터의 반전 시 설정

인터럽트 후의 피딩 위치 결정이 반전하는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전합니다.

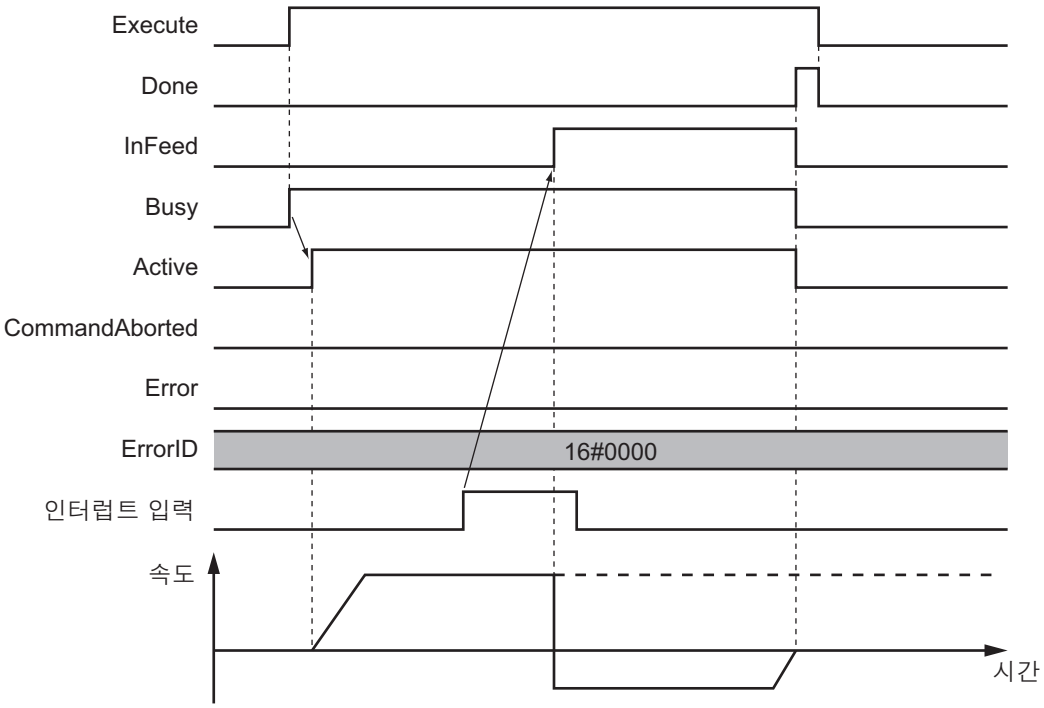
[0: 반전 시 감속 정지로 반전]하는 경우



참고

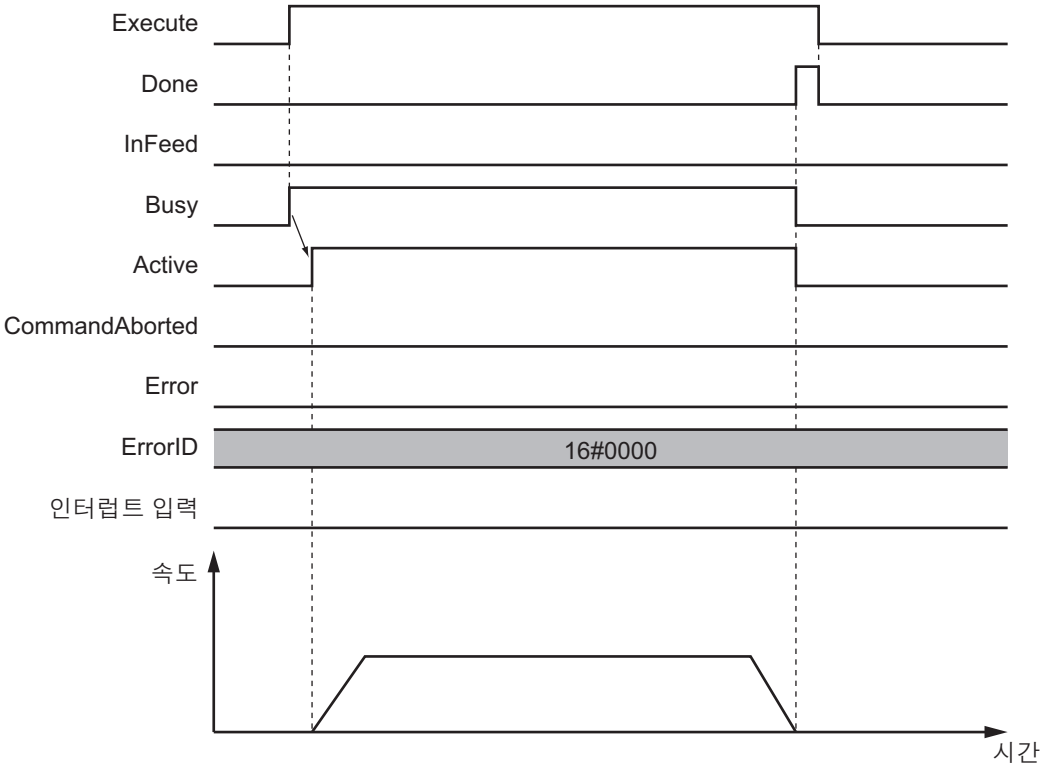
인터럽트 입력 시의 반전 감속도는 입력 변수 Deceleration(감속도)에 따릅니다.

[1: 반전 시 즉시 정지로 반전]하는 경우

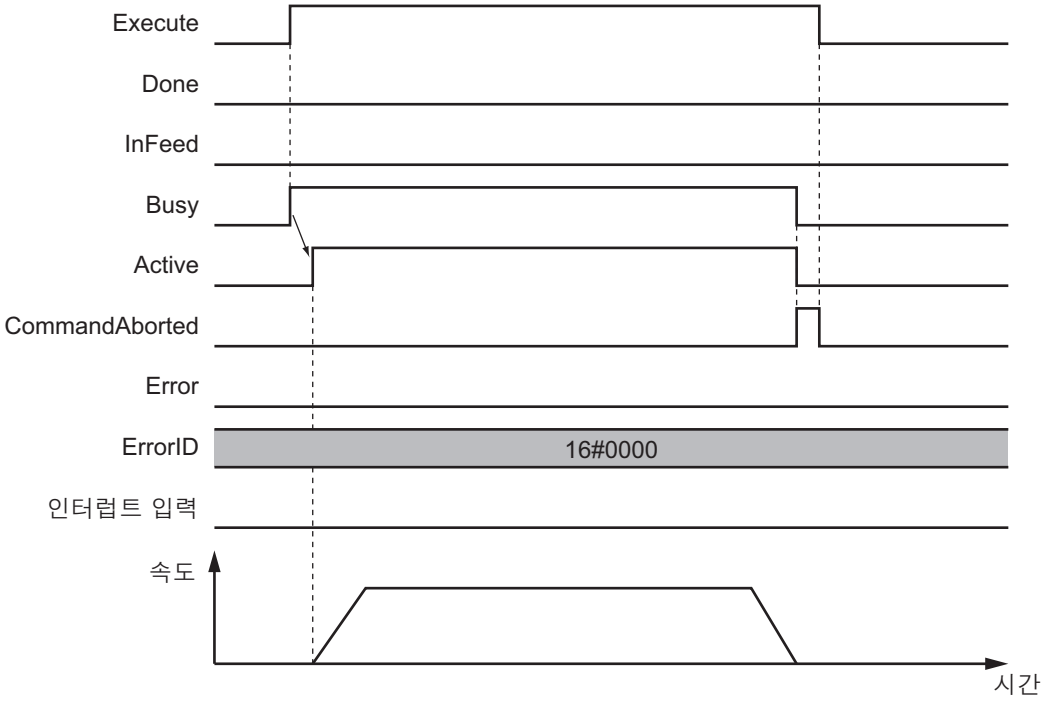


- **MoveMode(이동 방법 선택)가 _mcAbsolute인 경우로 인터럽트 입력이 없는 경우**

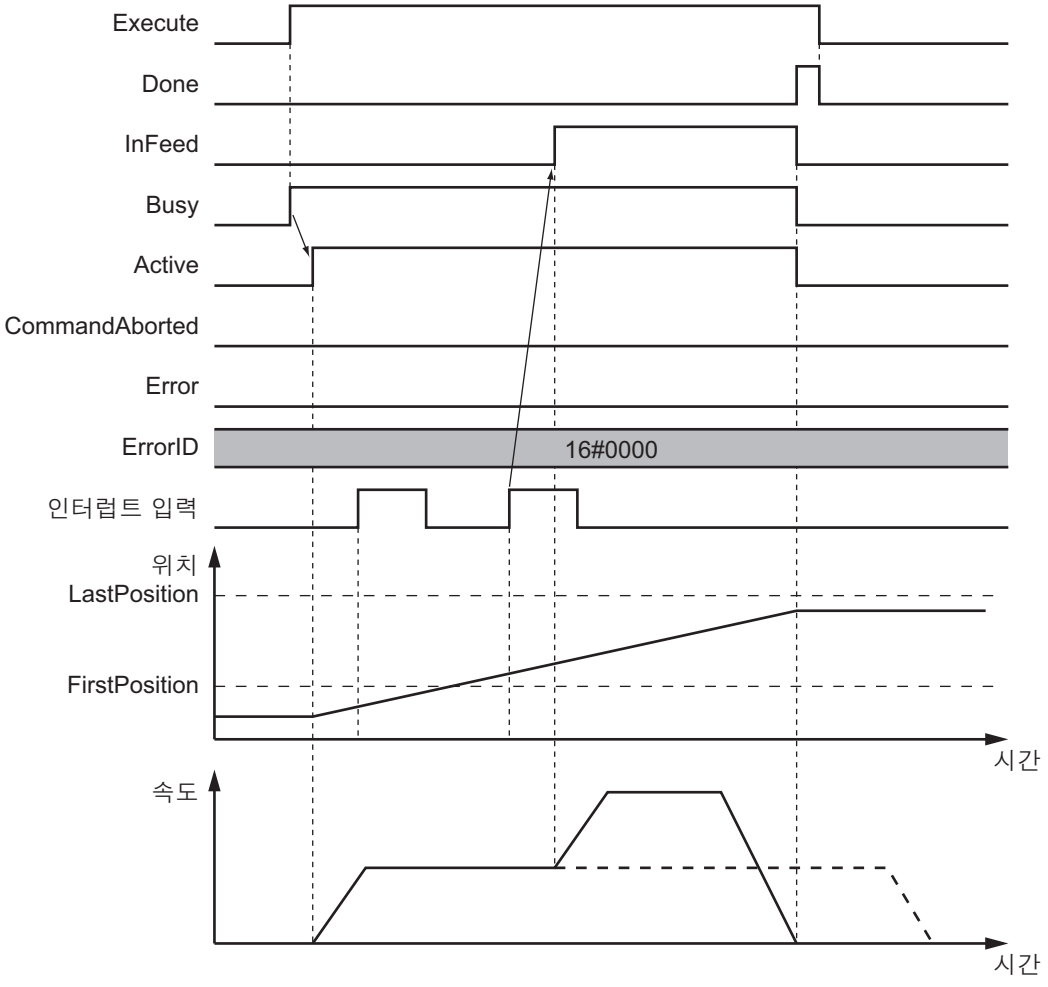
  [ErrorDetect(에러 검출 선택): 이상 검출 없음]에서 인터럽트 입력이 없는 경우



[ErrorDetect(에러 검출 선택): 이상 검출 있음]에서 인터럽트 입력이 없는 경우



- **WindowsOnly(창 유효)가 [유효]인 경우**



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다.
각 축에 1개까지 버퍼링할 수 있습니다.
다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.
Buffered	현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

이 명령이 실행 중에 Aborting에 의한 다중 기동이 가능합니다.

Buffered, Blending에 의한 다중 기동을 실행하면

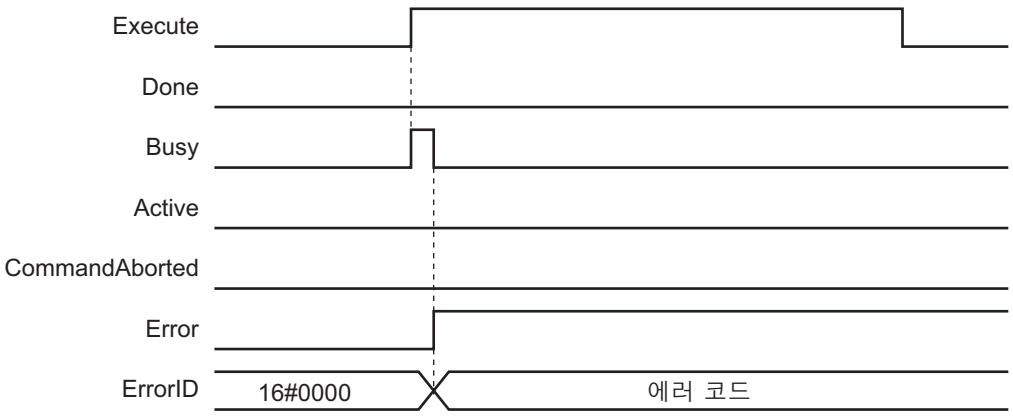
- 다중 기동을 실행한 명령은 Error(에러)가 TRUE가 됩니다. 동시에 ErrorID(에러 코드)에는 '모션 명령 다중 기동 불가(에러 코드: 543C Hex)'가 출력됩니다.
- 이 명령의 처리는 중지되고 CommandAborted(실행 중단)가 TRUE가 됩니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트

이상 발생 시 이 명령에서 사용하는 인터럽트 입력용 래치는 무효가 됩니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기에는 속도 제어에서 인터럽트 피딩 위치 결정으로 변화하는 제어에 대한 샘플 프로그램이 실려 있습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축
축 2	서보 축

카운터 모드

축	카운터 모드
축 1	로터리 모드
축 2	리니어 모드

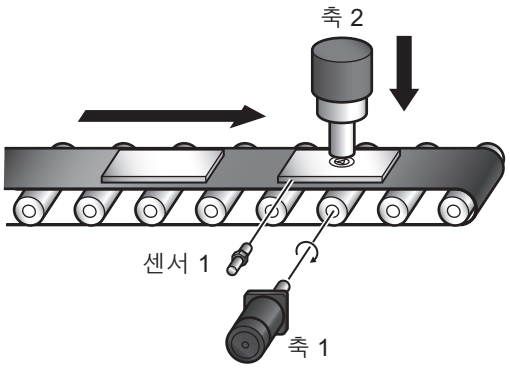
링 카운터

축	상한값	하한값
축 1	360	0

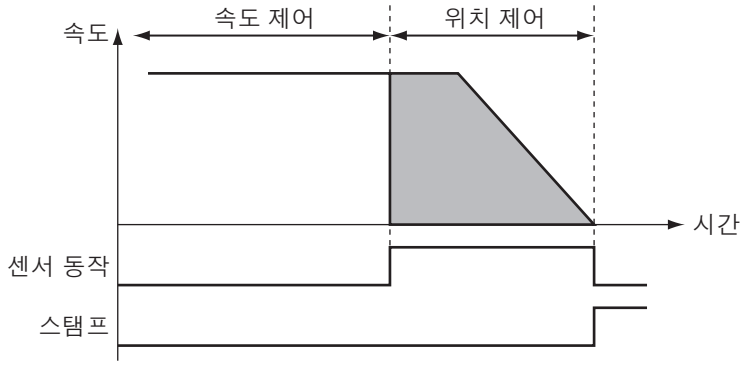
좌표 단위 선택

축	좌표 단위
축 1	degree
축 2	mm

동작 예



● **동작 패턴**



1 **컨베이어 동작**

벨트 컨베이어를 동작시키는 축 1은 인터럽트 피딩 위치 결정 전의 동작으로 속도 제어를 실행합니다.

2 **피딩**

센서 1은 래치 기능 1에 연결되어 있습니다.
센서 1이 'ON'하면 피딩 위치 결정으로 전환되고 소정의 위치에서 정지합니다.

3 **스탬프 날인**

위치 결정이 완료되면 스탬프의 축 2가 절대 위치 결정에 의해 수직으로 내려가 날인합니다.
날인 후 축 2는 원점 위치로 되돌아갑니다.

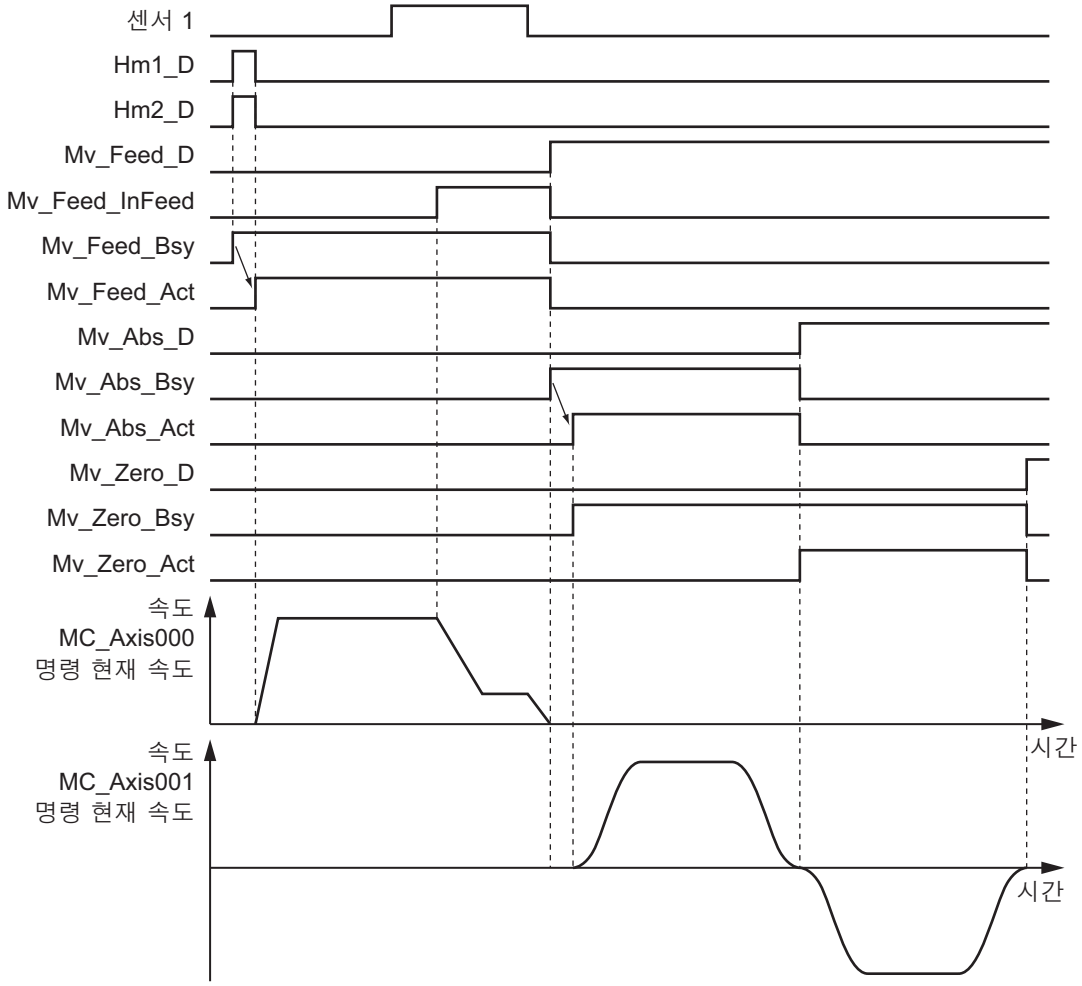
절대 위치 결정 동작이 완료하는 동시에 원점 위치로 복귀시키기 위해 MC_MoveZeroPosition(고속 원점 복귀)의 BufferMode(버퍼 모드 선택)를 Buffered로 기동합니다.
이전 명령의 출력 Active가 TRUE일 때 명령 다중 기동을 실행합니다.

래더도

● **주요 변수**

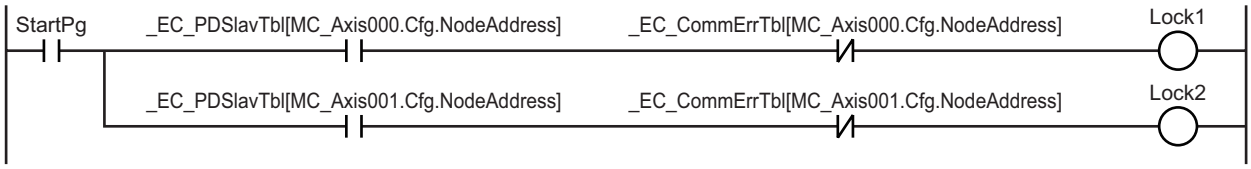
이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
TrigRef	_sTRIGGER_REF	–	인터럽트 입력의 지정 변수입니다. 이 프로그램에서는 서보 드라이버의 래치 기능 1을 사용합니다. 외부 입력(센서 1)의 기동을 검출하고 피딩 이동을 실행합니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트

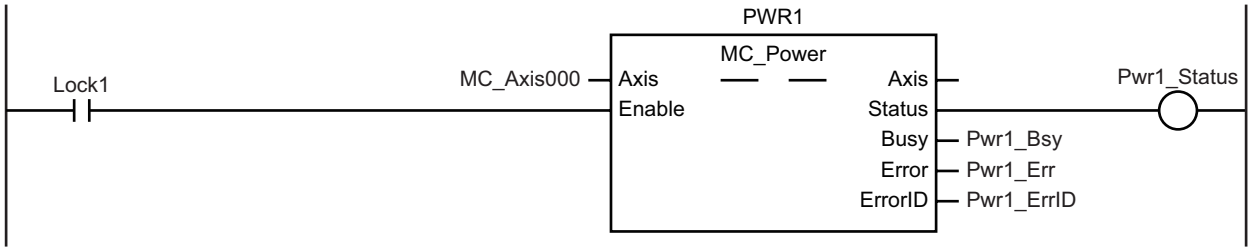


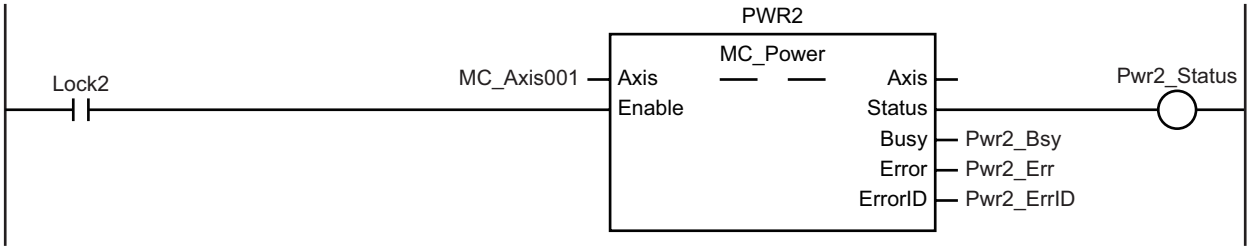
● 샘플 프로그램

접점 StartPg가 TRUE가 되면 각 축의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인

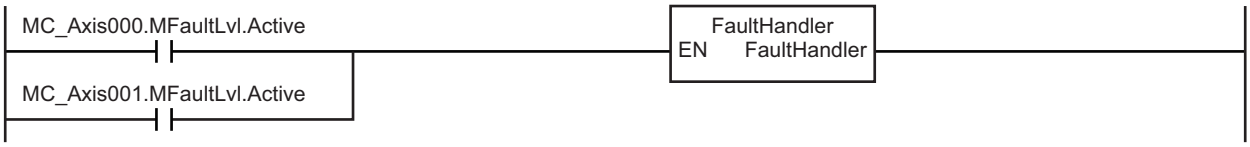


프로세스 데이터 통신을 하고 있으면 각 축의 서보를 ON

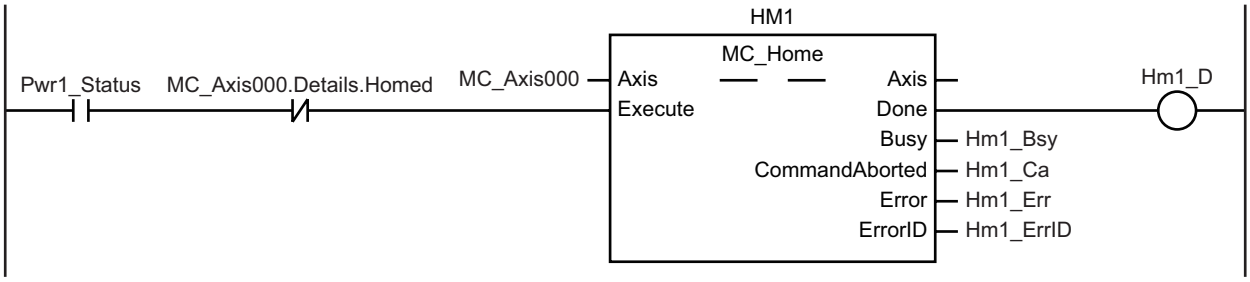




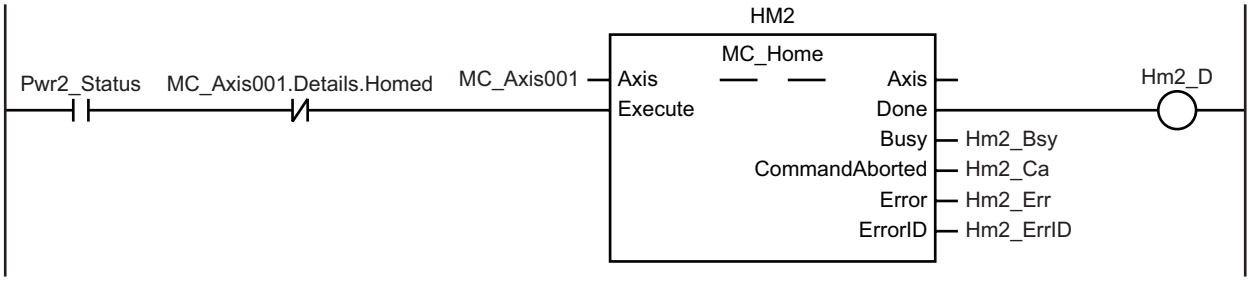
축 구성에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



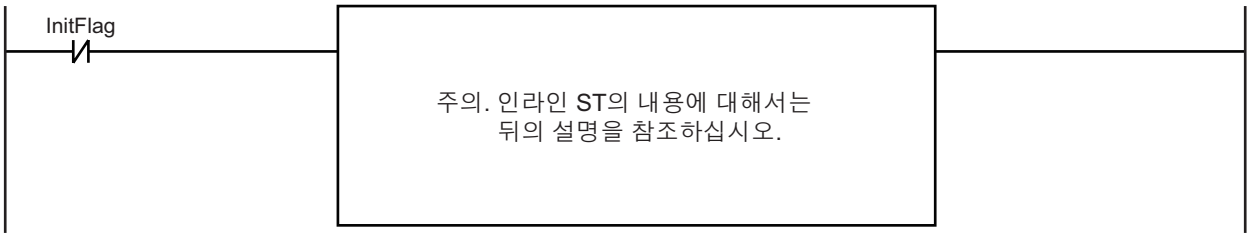
축 1이 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행



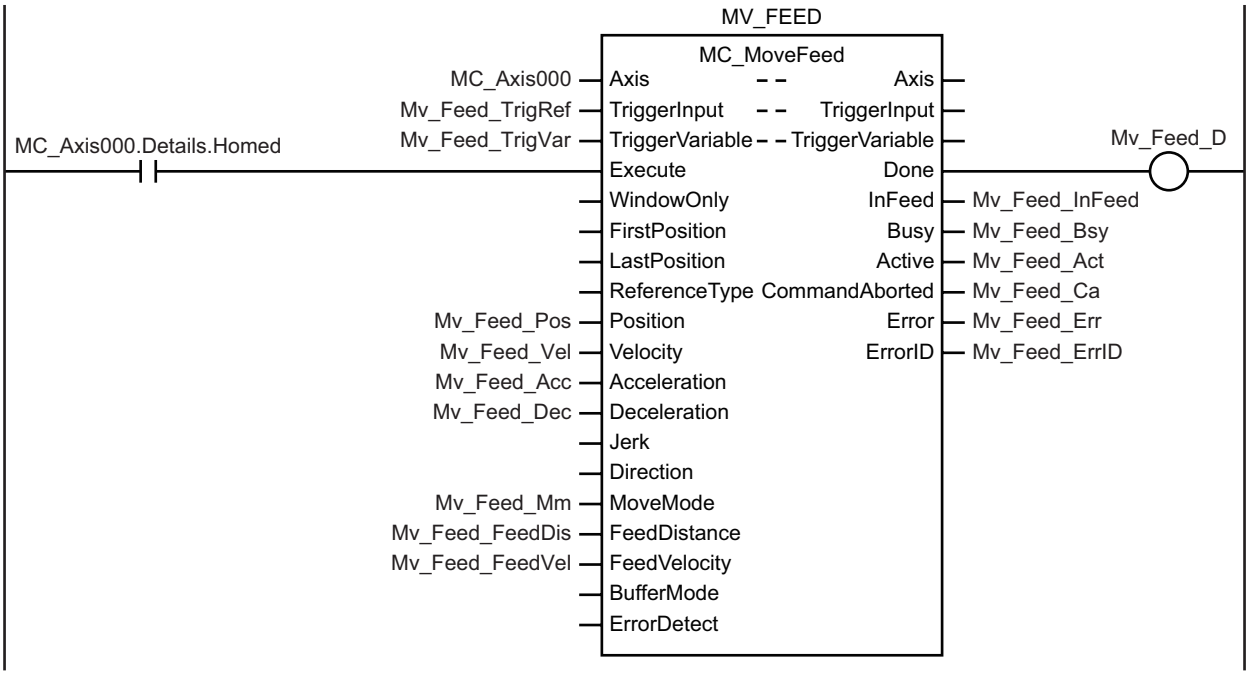
축 2가 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행



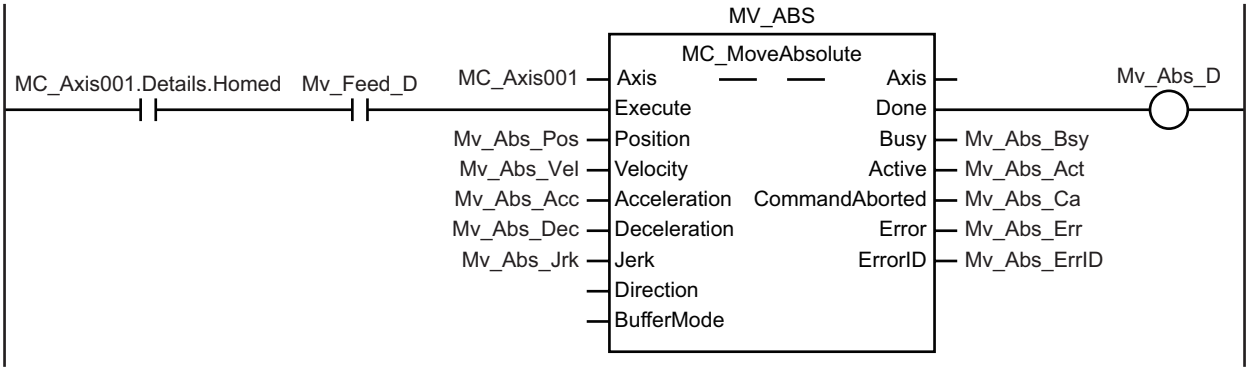
인터럽트 피딩 위치 결정, 절대값 위치 결정, 고속 원점 복귀 파라메터를 설정



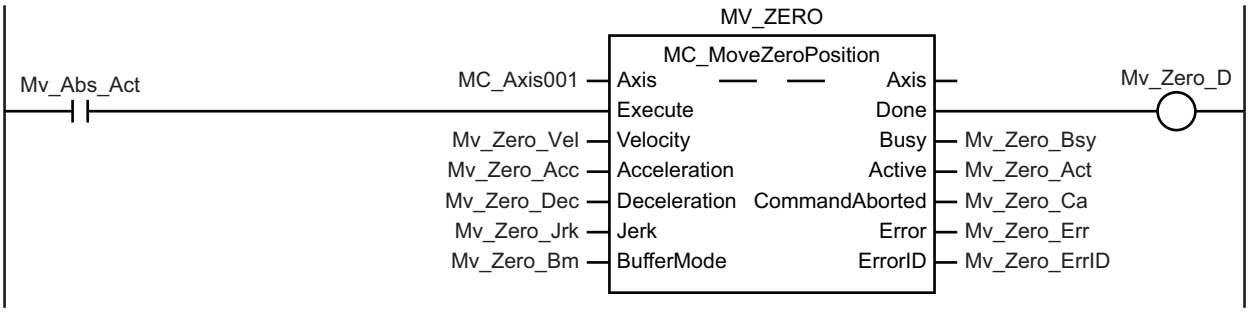
축 1이 원점 확정 상태가 된 것을 확인하고 축 1에서 인터럽트 피딩을 실행



축 2가 원점 확정 상태이고 축 1의 인터럽트 피딩 위치 결정이 완료되면 축 2에서 절대값 위치 결정을 실행



절대값 위치 결정 완료 후 고속 원점 복귀를 실행하고 원점으로 이동



인라인 ST의 내용

```
//MV_FEED 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#_mcEXT;
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_Mm                 := _eMC_MOVE_MODE#_mcVelocity;
Mv_Feed_FeedDis            := LREAL#500.0;
Mv_Feed_FeedVel            := LREAL#500.0;

//MV_ABS 파라메터
Mv_Abs_Pos    := LREAL#1000.0;
Mv_Abs_Vel    := LREAL#500.0;
Mv_Abs_Acc    := LREAL#10000.0;
Mv_Abs_Dec    := LREAL#10000.0;
Mv_Abs_Jrk    := LREAL#10000.0;

//MV_ZERO 파라메터
Mv_Zero_Vel   := LREAL#500.0;
Mv_Zero_Acc   := LREAL#10000.0;
Mv_Zero_Dec   := LREAL#10000.0;
Mv_Zero_Jrk   := LREAL#10000.0;
Mv_Zero_Bm    := _eMC_BUFFER_MODE#_mcBuffered;

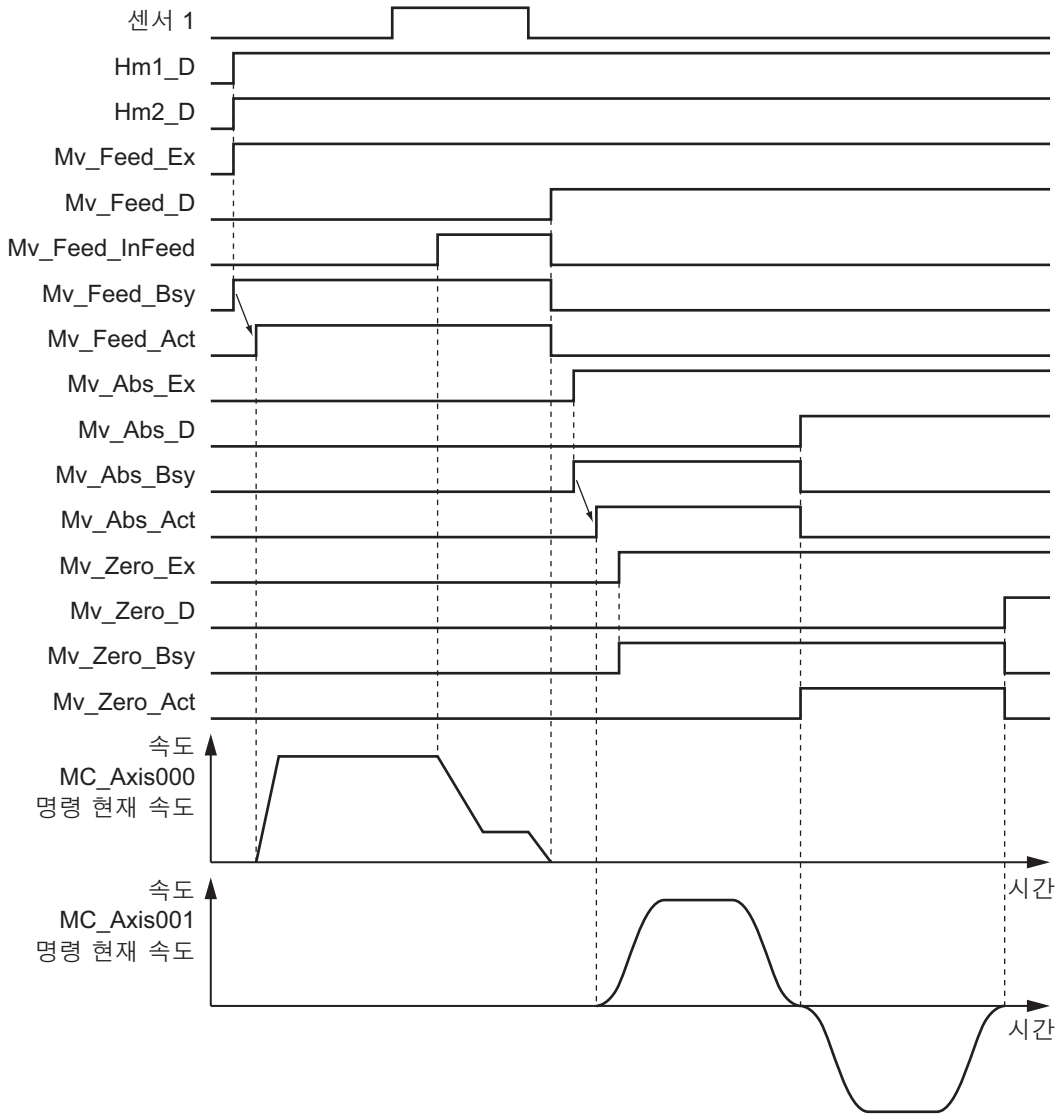
//입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
InitFlag:=TRUE;
```

스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
TrigRef	_sTRIGGER_REF	–	인터럽트 입력의 지정 변수입니다. 이 프로그램에서는 서보 드라이버의 래치 기능 1을 사용합니다. 외부 입력(센서 1)의 기동을 검출하고 피딩 이동을 실행합니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.
Hm1_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_Home의 인스턴스 HM1이 기동합니다.
Hm2_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_Home의 인스턴스 HM2가 기동합니다.
Mv_Feed_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveFeed의 인스턴스 MV_FEED가 기동합니다.
Mv_Abs_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveAbsolute의 인스턴스 MV_ABS가 기동합니다.
Mv_Zero_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveZeroPosition의 인스턴스 MV_ZERO가 기동합니다.

● 타이밍 차트



● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag=FALSE THEN

    //MV_FEED 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#_mcEXT;
    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_Mm                  := _eMC_MOVE_MODE#_mcVelocity;
    Mv_Feed_FeedDis             := LREAL#500.0;
    Mv_Feed_FeedVel             := LREAL#500.0;

    //MV_ABS 파라메터
    Mv_Abs_Pos    := LREAL#1000.0;
    Mv_Abs_Vel    := LREAL#500.0;
    Mv_Abs_Acc    := LREAL#10000.0;
    Mv_Abs_Dec    := LREAL#10000.0;
    Mv_Abs_Jrk    := LREAL#10000.0;
```

```
    //MV_ZERO 파라메터
    Mv_Zero_Vel    := LREAL#500.0;
    Mv_Zero_Acc    := LREAL#10000.0;
    Mv_Zero_Dec    := LREAL#10000.0;
    Mv_Zero_Jrk    := LREAL#10000.0;
    Mv_Zero_Bm     := _eMC_BUFFER_MODE#_mcBuffered;

    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;


//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 1을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr1_En:=TRUE;          //축 1을 서보 ON합니다.
ELSE
    Pwr1_En:=FALSE;         //축 1을 서보 OFF합니다.
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 2를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
    Pwr2_En:=TRUE;          //축 2를 서보 ON합니다.
ELSE
    Pwr2_En:=FALSE;         //축 2를 서보 OFF합니다.
END_IF;

//가벼운 수준의 이상 발생 시 처리
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;


//축 1이 서보 ON 상태이고 원점 미확정일 때 축 1의 원점 복귀를 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//축 2가 서보 ON 상태이고 원점 미확정일 때 축 2의 원점 복귀를 실행합니다.
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//축 1이 원점 확정 후에 MC_MoveFeed를 실행합니다.
IF MC_Axis000.Details.Homed=TRUE THEN
    Mv_Feed_Ex:=TRUE;
END_IF;

//축 2가 원점 확정 상태이고 축 1의 인터럽트 피딩 위치 결정이 완료된 경우 축 2의 절대 위치 결정을
실행합니다.
IF (MC_Axis001.Details.Homed=TRUE) AND (Mv_Feed_D=TRUE) THEN
    Mv_Abs_Ex := TRUE;
END_IF;
```

```
//MC_MoveAbsolute가 시작되면 MC_MoveZeroPosition을 다중 기동합니다.
IF Mv_Abs_Act=TRUE THEN
    Mv_Zero_Ex:=TRUE;
END_IF;


//축 1의 MC_Power
PWR1(

    Axis        := MC_Axis000,
    Enable      := Pwr1_En,
    Status      => Pwr1_Status,
    Busy        => Pwr1_Bsy,
    Error       => Pwr1_Err,
    ErrorID     => Pwr1_ErrID
);

//축 2의 MC_Power
PWR2(
    Axis        := MC_Axis001,
    Enable      := Pwr2_En,
    Status      => Pwr2_Status,
    Busy        => Pwr2_Bsy,
    Error       => Pwr2_Err,
    ErrorID     => Pwr2_ErrID
);

//축 1의 MC_Home
HM1(
    Axis                := MC_Axis000,
    Execute             := Hm1_Ex,
    Done                => Hm_D,
    Busy                => Hm1_Bsy,
    CommandAborted      => Hm1_Ca,
    Error               => Hm1_Err,
    ErrorID             => Hm1_ErrID
);

//축 2의 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_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,
    MoveMode            := Mv_Feed_Mm,
    FeedDistance        := Mv_Feed_FeedDis,
    FeedVelocity        := Mv_Feed_FeedVel,
    Done                => Mv_Feed_D,
    InFeed              => Mv_Feed_InFeed,
    Busy                => Mv_Feed_Bsy,
```

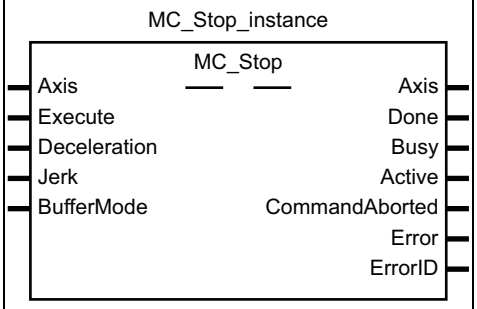
```
    Active              => Mv_Feed_Act,
    CommandAborted      => Mv_Feed_Ca,
    Error               => Mv_Feed_Err,
    ErrorID             => Mv_Feed_ErrID
);

//MC_MoveAbsolute
MV_ABS(
    Axis                := MC_Axis001,
    Execute             := Mv_Abs_Ex,
    Position            := Mv_Abs_Pos,
    Velocity            := Mv_Abs_Vel,
    Acceleration        := Mv_Abs_Acc,
    Deceleration        := Mv_Abs_Dec,
    Jerk                := Mv_Abs_Jrk,
    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_MoveZeroPosition
MV_ZERO(
    Axis                := MC_Axis001,
    Execute             := Mv_Zero_Ex,
    Velocity            := Mv_Zero_Vel,
    Acceleration        := Mv_Zero_Acc,
    Deceleration        := Mv_Zero_Dec,
    Jerk                := Mv_Zero_Jrk,
    BufferMode          := Mv_Zero_Bm,
    Done                => Mv_Zero_D,
    Busy                => Mv_Zero_Bsy,
    Active              => Mv_Zero_Act,
    CommandAborted      => Mv_Zero_Ca,
    Error               => Mv_Zero_Err,
    ErrorID             => Mv_Zero_ErrID
);
```

MC_Stop

축을 감속 정지시킵니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_Stop	강제 정지	FB		MC_Stop_instance (Axis := 《파라메터》, Execute := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, BufferMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1
BufferMode	버퍼 모드 선택	_eMC_BUFFER_ MODE	0: _mcAborting	0 *2	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

*2. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	감속 정지가 완료되고 속도가 '0'에 도달했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 MC_Stop(강제 정지) 명령에 의한 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 축을 현재 속도에서 속도 '0'으로 감속 제어합니다.
- Execute(기동)가 기동하면 감속 정지 동작을 시작합니다.
- MC_Stop(강제 정지) 명령 기동으로 동작 중인 명령은 CommandAborted(실행 중단)합니다.

사용상의 주의 사항

축 상태를 나타내는 Status.ErrorStop(에러 감속 정지 중)이 TRUE일 때 이 명령은 기동할 수 없습니다.
에러 감속 정지 중에 축 동작을 정지시키려면 MC_ImmediateStop(즉시 정지) 명령을 사용하십시오.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

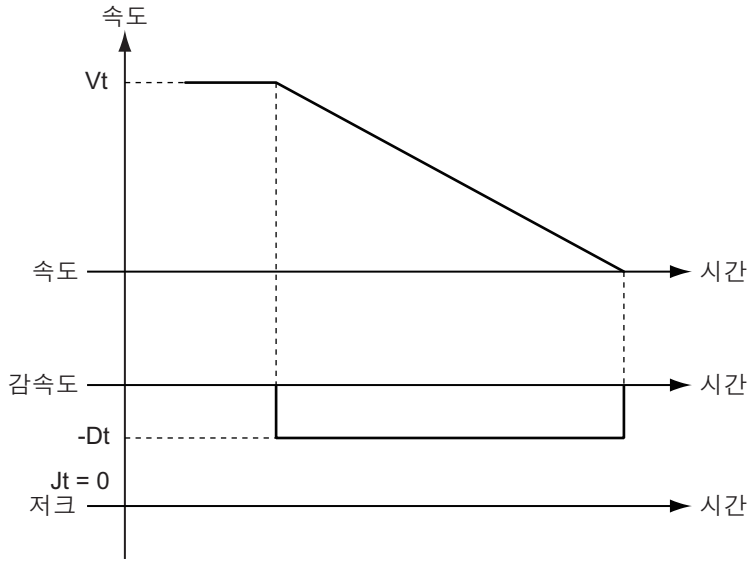
● Deceleration(감속도), Jerk(저크)의 지정

감속 정지 시의 감속도, 저크는 입력 변수의 Deceleration(감속도), Jerk(저크)로 지정합니다.

저크에 '0'을 지정한 경우와 '0' 이외의 것을 지정한 경우의 감속도, 속도의 관계는 다음과 같습니다.

저크가 '0'일 때

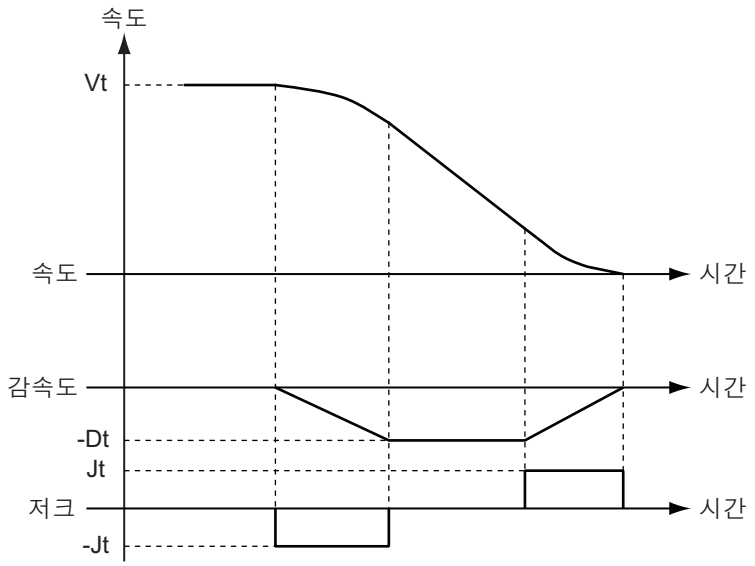
감속도 Dt로 속도의 명령값을 작성합니다.



Vt: 감속 기동 시의 속도, Dt: 감속도의 지정값, Jt: 저크의 지정값

저크가 '0' 이외일 때

감속도의 상한을 Dt로 하고 현재 속도를 기점으로 해서 속도의 명령값을 작성합니다.



Vt : 감속 기동 시의 속도, Dt : 감속도의 지정값, Jt : 저크의 지정값

참고

- 감속도에 '0'을 지정하면 즉시 정지 명령이 됩니다. 감속하지 않고 바로 정지합니다. 감속도에 '0'을 지정한 경우에만 [가감속 초과] 설정에 상관 없이 즉시 정지를 실행합니다.

● BufferMode(버퍼 모드 선택)의 지정

하나 이전의 축 동작과 이번 동작의 연결 방법을 지정합니다.
이 명령의 BufferMode(버퍼 모드 선택)는 향후 확장용 예약 파라메터입니다.
종류는 다음 1가지입니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령을 실행합니다.

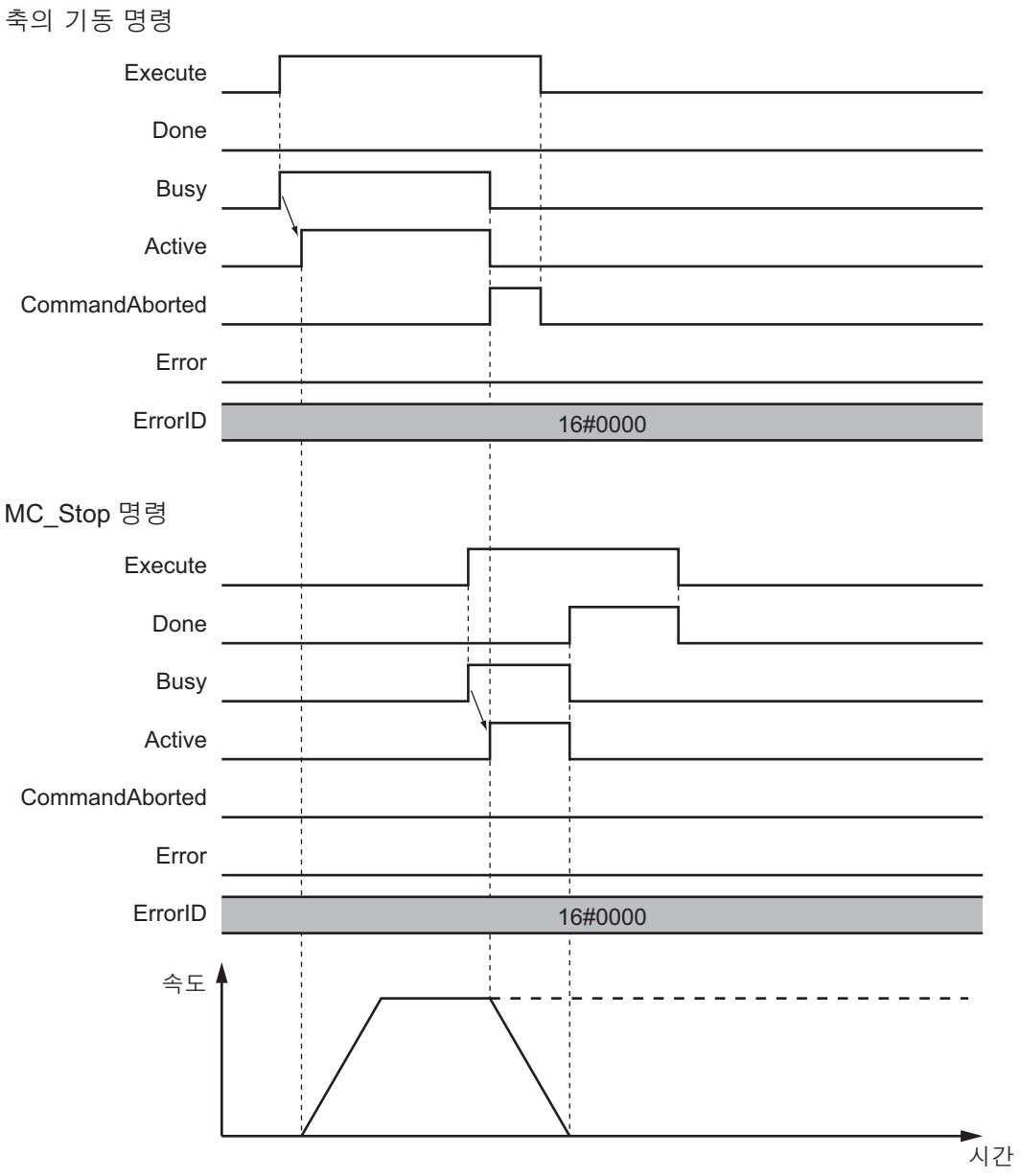
BufferMode(버퍼 모드 선택)에 대해서는 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 인포지션 확인

이 명령으로 정지 시에 인포지션 확인은 실행하지 않습니다.

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- 속도 '0'에 도달하면 Done(완료)이 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중단되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중)가 FALSE가 됩니다.

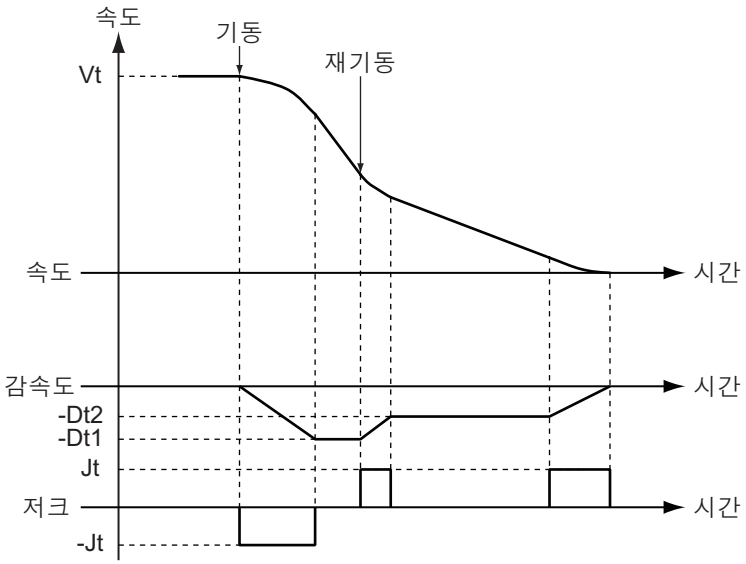


모션 명령 재기동

이 명령 실행 중에 다시 Execute(기동)를 기동하면 Deceleration(감속도)을 변경합니다.
모션 명령 재기동으로는 Jerk(저크)를 변경하지 않습니다.

● 저크가 '0' 이외일 때

현재 속도, 현재 감속도를 기점으로 변경 후 감속도의 상한을 Dt2로서 속도의 명령값을 작성합니다.



Vt : 감속 기동 시의 속도, Dt : 감속도의 지정값, Jt : 저크의 지정값

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

축 그룹 동작 중

지정한 축이 축 그룹 동작 중에 MC_Stop(강제 정지) 명령을 실행하면 축은 이상이 됩니다.
축 그룹도 이상이 되고 축 그룹 동작도 멈춥니다.

축 변수의 Status.ErrorStop(에러 감속 정지 중)이 TRUE일 때

축에서 이상이 발생하는 동안 축 변수의 Status.ErrorStop(에러 감속 정지 중)은 TRUE가 됩니다.
에러 감속 정지 중에 MC_Stop(강제 정지) 명령을 기동하면 정상 기동하지 못하고 CommandAborted(실행 중단)가 TRUE가 됩니다. 이상이 발생 중인 축을 정지하려면 MC_ImmediateStop(즉시 정지) 명령을 사용하십시오.

MC_ResetFollowingError(편차 카운터 리셋) 명령 실행 중

MC_ResetFollowingError(편차 카운터 리셋) 명령 실행 중에 MC_Stop(강제 정지) 명령을 기동하면 MC_ResetFollowingError(편차 카운터 리셋) 명령의 CommandAborted(실행 중단)가 TRUE가 되고 MC_Stop(강제 정지) 명령이 실행됩니다. 단, 이 명령의 Deceleration(감속도)은 적용되지 않고 즉시 정지합니다.

● 이 명령 실행 중의 다른 명령 기동

- MC_Stop(강제 정지) 명령에 의한 축의 감속 정지 외에도 다음 명령을 기동하면 축 변수의 Status.Stopping (감속 정지 중)은 TRUE가 됩니다.
  MC_ResetFollowingError(편차 카운터 리셋) 명령
  MC_TouchProbe(외부 래치 유효) 명령의 즉시 정지 중
  다른 MC_Stop(강제 정지) 명령의 Execute(기동)가 하나라도 TRUE가 되었을 때
- Status.Stopping(감속 정지 중)이 TRUE인 축에 대해서는 다음과 같습니다.
  - 단축 위치 결정, 연속 동작, 동기 동작, 수동 동작을 실행하면 이러한 명령의 CommandAborted(실행 중단)가 TRUE가 됩니다.
  - MC_Stop(강제 정지) 명령 실행 중에 MC_ResetFollowingError(편차 카운터 리셋) 명령을 기동하면 MC_Stop(강제 정지) 명령의 Done(완료)이 TRUE가 되고 MC_ResetFollowingError(편차 카운터 리셋) 명령이 실행됩니다.
  - MC_Stop(강제 정지) 명령끼리의 다중 기동이 가능합니다. 실행 중인 MC_Stop(강제 정지) 명령의 Done(완료)이 TRUE가 됩니다.
- MC_Stop(강제 정지) 명령이 기동한 채로 다음 상태가 되면 MC_Stop(강제 정지) 명령의 Done(완료)이 TRUE가 됩니다.
  - MC_Power(운전 가능)의 입력 변수 'Enable(유효)'을 FALSE로 한 경우(서보를 OFF한 경우)
  - MC_TouchProbe(외부 래치 유효) 명령의 입력 변수 'StopMode(정지 방법 선택)'로 [1: _mcImmediateStop]을 선택한 상태에서 트리거 조건이 성립하고, 오므론의 서보 드라이버 G5 시리즈가 즉시 정지한 경우

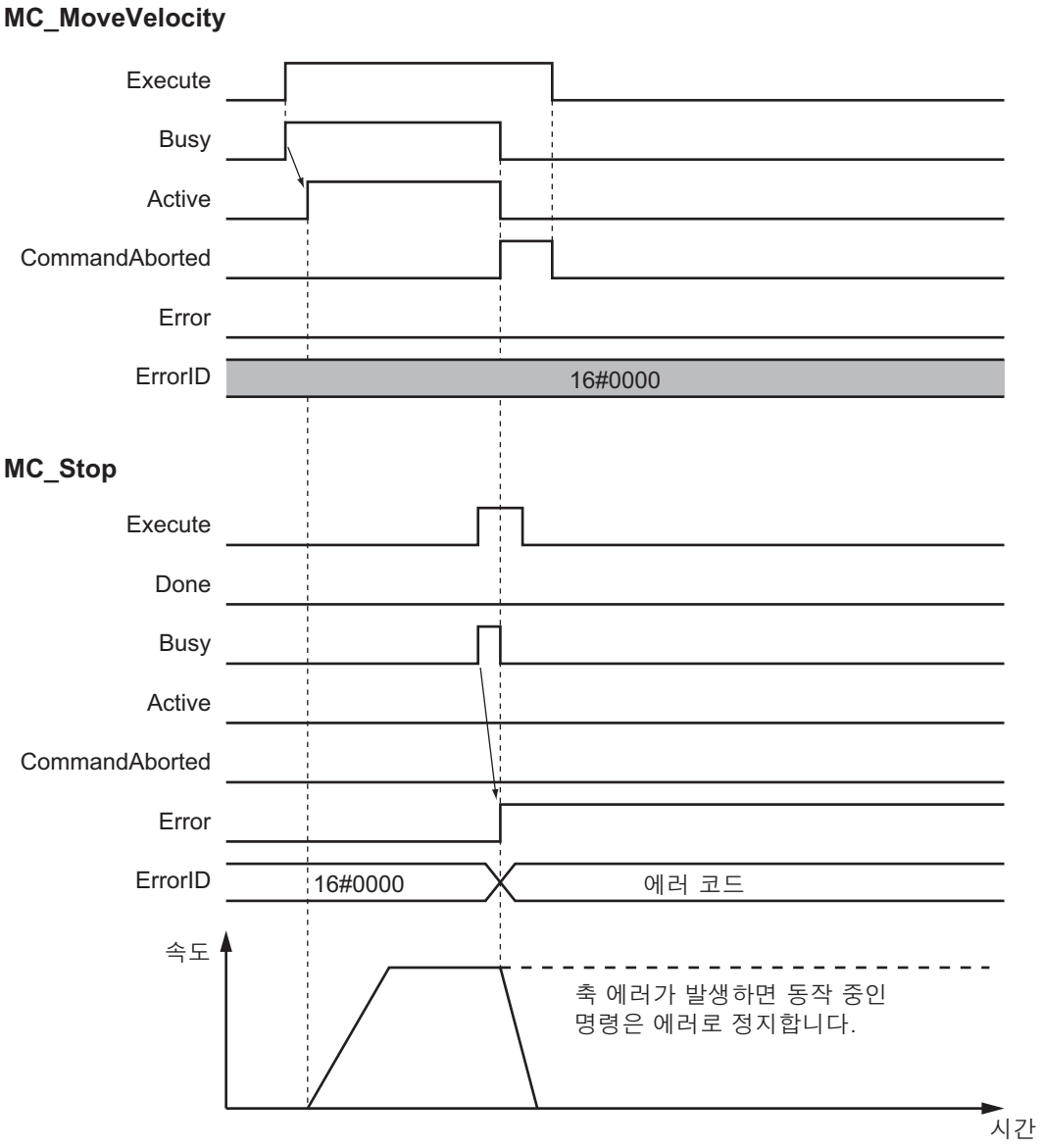
이상

이 명령 실행 중에 축 등에 이상이 발생하면 동작이 정지합니다.
정지 방법은 즉시 정지, 감속 정지, 서보 OFF 중 하나를 축 파라메터로 지정합니다.
정지 방법으로 감속 정지가 지정되어 있으면 감속 정지 동작을 계속합니다.

축 파라메터 정지 방법의 설정에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이상 발생 시의 타이밍 차트

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

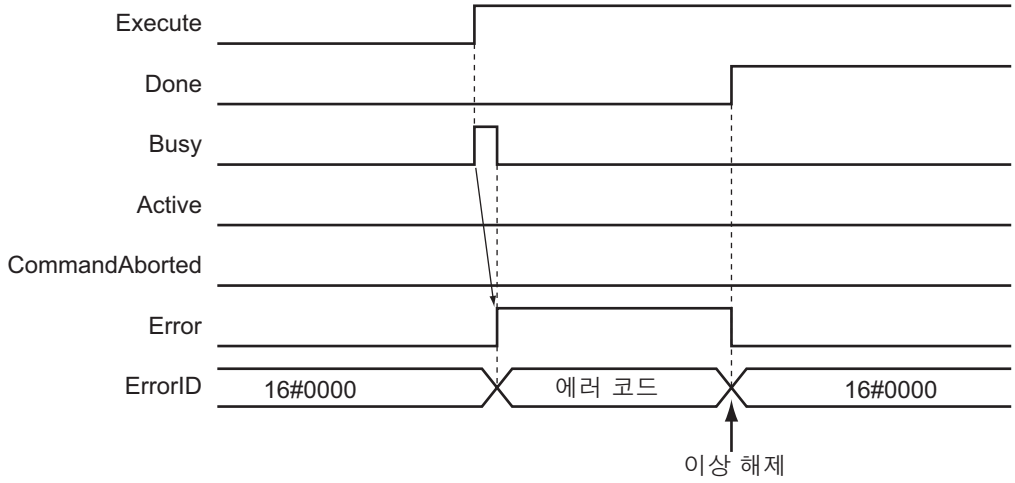


● 이상 해제 시의 동작

이 명령으로 이상이 발생한 상태에서 Execute(기동)가 TRUE인 채로 이상을 해제하면 Error(에러)가 FALSE, Done(완료)이 TRUE가 됩니다. 축 변수의 Status.Stopping(감속 정지 중)은 감속 정지의 정상 실행 시와 동일하게 TRUE가 됩니다.

이상은 축이 정지하고 나서 해제하십시오. 축 동작 중에는 이상을 해제할 수 없습니다.

MC_Stop



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_ImmediateStop

축 상태에 관계 없이 입력 변수 'StopMode(정지 방법 선택)'의 정지 방법에 따라 정지합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_ImmediateStop	즉시 정지	FB	MC_ImmediateStop_instance MC_ImmediateStop Axis — — Axis Execute — Done StopMode — Busy CommandAborted Error ErrorID	MC_ImmediateStop_instance (Axis := 《파라메터》, Execute := 《파라메터》, StopMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
StopMode	정지 방법 선택	_eMC_ STOP_ MODE	1: _mcImmediateStop 2: _mcImmediateStopFEReset 3: _mcFreeRunStop	1 *1	정지 방법을 선택합니다. 1 : 즉시 정지합니다. 2 : 즉시 정지하고 편차 카운터 리셋을 실행합니다. 3 : 즉시 정지하고 서보를 OFF합니다.

*1. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	감속 정지를 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 MC_Stop(강제 정지) 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

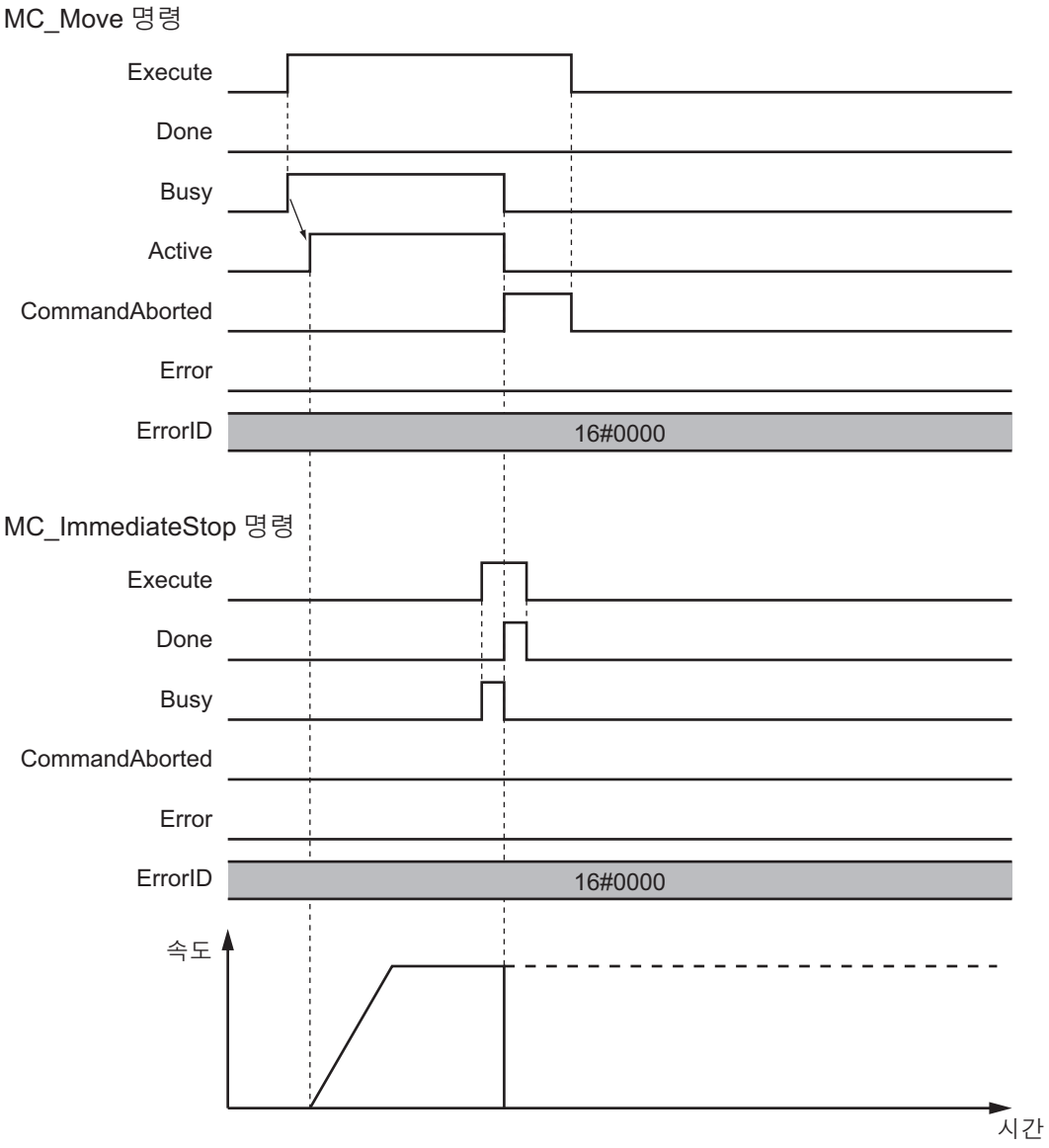
- 이 명령은 어떤 상태에서도 실행이 가능합니다.
  예를 들어 이상으로 감속 정지 중에도 이 명령으로 즉시 정지할 수 있습니다.
  MC_Stop(강제 정지) 명령의 경우는 Status.ErrorStop(에러 감속 정지 중)이 TRUE인 상태에서는 사용할 수 없지만, MC_ImmediateStop(즉시 정지) 명령은 사용 가능합니다.
- 명령이 실행되면 StopMode(정지 방법 선택)에서 지정된 대로 즉시 정지하고, 동작 중인 명령은 CommandAborted(실행 중단)가 됩니다.
- 이 명령을 실행하면 축 상태는 Status.ErrorStop(에러 감속 정지 중)이 TRUE가 되고 '즉시 정지 명령 기동(에러 코드: 5485 Hex)'이 발생합니다.

사용상의 주의 사항

주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 됩니다.
- 즉시 정지 명령 처리가 완료되면 Done(완료)이 TRUE가 됩니다.



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

축 그룹 동작 중

지정한 축이 축 그룹 동작 중에 명령을 실행하면 축은 이상이 되어 즉시 정지합니다.
축 그룹도 이상이 되고 축 그룹 동작도 멈춥니다.

축 변수의 Status.Stopping(감속 정지 중)이 TRUE일 때

다음 상태인 경우 축 변수의 Status.Stopping(감속 정지 중)이 TRUE가 됩니다.

- MC_Stop(강제 정지) 명령에 의한 축 감속 중
- MC_ResetFollowingError(편차 카운터 리셋) 명령 실행 중
- MC_TouchProbe(외부 래치 유효) 명령의 즉시 정지 중
- MC_Stop(강제 정지) 명령의 Execute가 다중 기동에 의해 하나라도 TRUE가 된 동안

Status.Stopping(감속 정지 중)이 TRUE인 축에 대해서도 이 명령을 실행 가능합니다.
이 명령이 기동되면 다음의 실행 중인 명령은 CommandAborted(실행 중단)가 TRUE가 됩니다.

- MC_Stop(강제 정지) 명령
- MC_ResetFollowingError(편차 카운터 리셋) 명령
- 즉시 정지 중인 MC_TouchProbe(외부 래치 유효) 명령

축 변수의 Status.ErrorStop(에러 감속 정지 중)이 TRUE일 때

축에서 이상이 발생하는 동안 축 상태는 Status.ErrorStop(에러 감속 정지 중)이 TRUE가 됩니다.
이상에 의한 감속 정지 중이라도 이 명령은 실행 가능합니다.

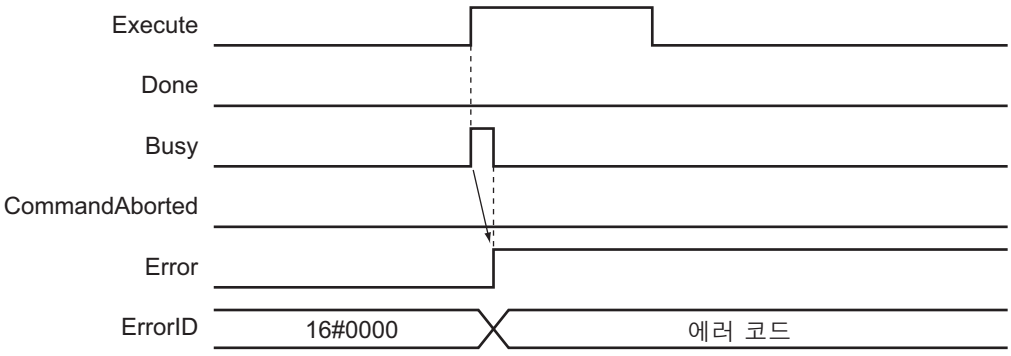
이상

이 명령 실행 중에 축 등에 이상이 발생한 경우라도 그대로 즉시 정지 동작을 계속합니다.

축 파라메터 정지 방법의 설정에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이상 발생 시의 타이밍 차트

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_SetPosition

축 명령 현재 위치와 피드백 현재 위치를 임의의 값으로 변경합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_SetPosition	현재 위치 변경	FB	MC_SetPosition_instance MC_SetPosition Axis — — Axis Execute — Done Position — Busy ReferenceType — CommandAborted Relative — Error ExecutionMode — ErrorID	MC_SetPosition_instance (Axis := 《파라메터》, Execute := 《파라메터》, Position := 《파라메터》, ReferenceType := 《파라메터》, Relative := 《파라메터》, ExcutionMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Position	목표 위치	LREAL	음수, 양수, '0'	0	절대 좌표의 목표 위치를 지정합니다. 단위는 [명령 단위]입니다. *1
ReferenceType	위치 종류 선택	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback	0 *2	위치 종류를 지정합니다. 0 : 명령 위치(서보 축, 가상 서보 축) 1 : 피드백 위치(엔코더 축, 가상 엔코더 축)
Relative (Reserved)	상대 위치 선택	BOOL	TRUE, FALSE	FALSE	(Reserved)
Execution Mode (Reserved)	실행 모드 선택	_eMC_ EXECUTION_ MODE	0: _mcImmediately	0 *2	(Reserved)

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

*2. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령 현재 위치와 피드백 현재 위치의 변경을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때
CommandAborted	• 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 서보 축의 명령 현재 위치를 지정한 목표 위치로 변경합니다.
  또한 엔코더 축에 대해 실행하면 피드백 현재 위치를 지정한 목표 위치로 변경합니다.
- 목표 위치는 절대 좌표 위치에서 지정합니다.
- 명령 현재 위치를 변경하는 시점과 동시에 피드백 현재 위치도 변경되는데, 그 시점의 위치 편차를 유지한 채 변경을 실행합니다. 명령 서보 축에 대해 실행한 경우는 현재 위치와 피드백 현재 위치의 차이를 유지한 채 변경됩니다.
  따라서 이 명령 실행 후 축의 피드백 현재 위치는 다음 계산식에 나타낸 값이 됩니다.

  변경 후의 피드백 현재 위치 = 목표 위치- 변경 전의 위치 편차

- 서보 축에 대해 피드백 위치를 지정한 경우 또는 엔코더 축에 대해 명령 위치를 지정한 경우는 위치 종류 선택 이상이 발생합니다.
- 카운트 모드가 [로터리 모드]이면 목표 위치는 링 카운터 하한값 이상, 링 카운터 상한값 미만의 범위에서 설정합니다. 목표 위치가 이 범위 밖이면 링 카운터 이상이 발생합니다.
- 카운트 모드가 [리니어 모드]이면 목표 위치를 소프트웨어 리미트 범위 밖으로도 설정할 수 있습니다.
- 이 명령은 축이 정지 상태이거나 동작 중에 모두 사용할 수 있습니다.

사용상의 주의 사항

이 명령 완료 후에는 지정한 축이 원점 미확정이 됩니다.
따라서 이 명령 완료 후 다음 기능이나 명령은 사용할 수 없으므로 주의하십시오.

- 소프트웨어 리미트 기능
- MC_MoveZeroPosition(고속 원점 복귀) 명령
- 다축 협조 동작 명령(직선 보간과 원호 보간)

● ReferenceType(위치 종류 선택)

- 서보 축 또는 가상 서보 축을 사용하는 경우는 '0: _mcCommand(명령 위치)'를 선택하십시오.
- 엔코더 축 또는 가상 엔코더 축을 사용하는 경우는 '1: _mcFeedback(피드백 위치)'을 선택하십시오.

● 축 종류와 위치 종류의 관계

감시 가능한 축 종류와 감시하는 위치 종류의 관계는 다음과 같습니다.

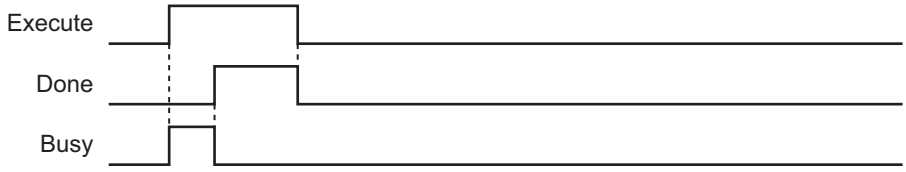
축 종류	ReferenceType	
	_mcCommand	_mcFeedback
서보 축	○	×
엔코더 축	× *1	○
가상 서보 축	○	×
가상 엔코더 축	× *1	○

*1. 명령 기동 시에 '위치 종류 선택 범위 외(에러 코드: 5430 Hex)'가 발생합니다.

타이밍 차트

● 축이 정지 상태일 때 이 명령을 사용하는 경우

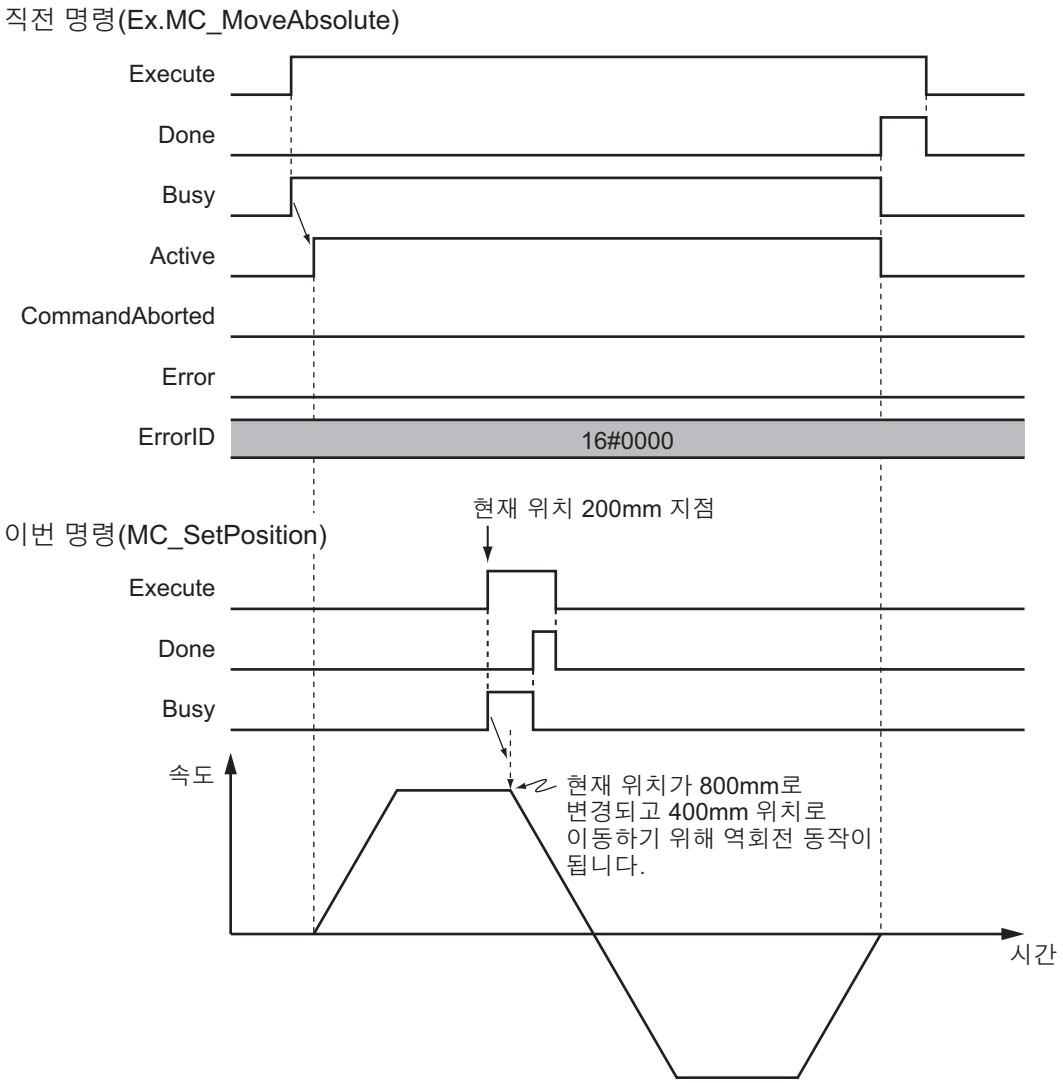
Execute(기동)가 기동하면 현재 위치를 변경하기 시작합니다.
Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 현재 위치의 변경이 완료되면 Done(완료)이 TRUE가 됩니다.



● 축이 동작 중에 이 명령을 사용하는 경우

절대 위치에 대한 위치 결정 중에 이 명령을 실행하면 위치 변경에 따른 동작 목표값이 변경됩니다.

예를 들어 MC_MoveAbsolute(절대값 위치 결정) 명령으로 400(mm)로 이동 중에 현재 위치 200(mm)에서 현재 위치를 800(mm)로 변경한 경우의 축 동작과 타이밍 차트는 다음과 같습니다.
이 경우 현재값이 800(mm)가 되고 목표값이 400(mm)이므로 축은 역전합니다.
아래 그림과 같이 MC_MoveAbsolute(절대값 위치 결정) 명령은 현재 위치가 변경되어도 새로운 현재 위치로부터 MC_MoveAbsolute(절대값 위치 결정) 명령으로 지정된 목표 위치로 위치 결정을 완료한 후 Done(완료)이 TRUE가 됩니다.



참고

- MC_MoveRelative(상대값 위치 결정) 명령 중이나 MC_MoveVelocity(속도 제어) 명령 중에 이 명령을 실행하면 현재 위치는 변경됩니다. 그러나 MC_MoveRelative(상대값 위치 결정) 명령이나 MC_MoveVelocity(속도 제어) 명령의 동작에는 영향을 미치지 않습니다.
- 버퍼링되고 있는 명령이 있으면 버퍼 전환 시에 변경 후의 위치에 대해 위치를 결정합니다.

모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

MC_SetPosition(현재 위치 변경) 명령은 다음 명령을 실행 중인 축에 대해서는 사용할 수 없습니다.
실행하면 다중 기동 이상이 됩니다.

MC_Home(원점 복귀) 명령	MC_MoveLink(사다리꼴 패턴 캠) 명령
MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령	MC_CombineAxes(가감산 위치 결정) 명령
MC_CamIn(캠 동작 시작) 명령	MC_MoveZeroPosition(고속 원점 복귀) 명령
MC_GearIn(기어 동작 시작) 명령	MC_TorqueControl(토크 제어) 명령
MC_GearInPos(위치 지정 기어 동작) 명령	MC_SyncMoveVelocity(주기적 동기 속도 제어) 명령

사용상의 주의 사항

MC_SetPosition(현재 위치 변경) 명령은 MC_GearIn(기어 동작 시작) 명령 등의 동기용 명령 Master(주축)에 대해 사용하지 마십시오.
Master(주축)에 대해 사용하면 Master(주축)의 명령 현재 위치와 피드백 현재 위치가 변화되자마자 Slave(종축)는 Master(주축)가 이동했다고 판단하고 Master(주축)의 이동량에 따라 동작합니다. 이로 인해 Slave(종축)의 움직임이 급격하게 변화하거나 캠 동작이 종료되는 경우가 있습니다.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 사용하려면 Master(주축)와 Slave(종축)의 관계를 일단 해제하고 나서 실행하십시오.
Master(주축)과 Slave(종축)의 관계를 해제하려면 MC_GearOut(기어 동작 해제) 명령 등을 실행하십시오.
주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

MC_SetPosition(현재 위치 변경) 명령 실행 중에 다른 MC_SetPosition(현재 위치 변경) 명령을 실행하면 나중의 MC_SetPosition(현재 위치 변경) 명령이 우선 실행됩니다.
이 경우 실행 중이던 이전의 MC_SetPosition(현재 위치 변경) 명령의 Done(완료)은 TRUE가 되지만, 앞에서 설정한 위치로 변경되지 않은 상태이므로 주의하십시오.

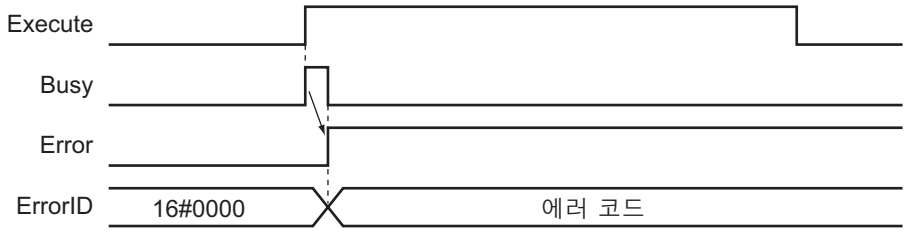
이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

참고

축 동작 중에 이 명령을 실행하면 현재 동작 중인 축의 명령은 재기동으로 간주되고 위치 결정을 위한 데이터를 다시 계산합니다.
이 때 이상이 발생하면 이 명령의 이상이 되지는 않으며, 현재 동작 중인 축 명령이 이상이 됩니다.

● 이상 발생 시의 타이밍 차트

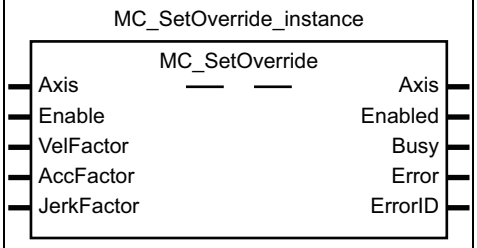


● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_SetOverride

축의 목표 속도를 변경합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_SetOverride	오버라이드값 설정	FB		MC_SetOverride_instance (Axis := 《파라메터》, Enable := 《파라메터》, VelFactor := 《파라메터》, AccFactor := 《파라메터》, JerkFactor := 《파라메터》, Enabled => 《파라메터》, Busy => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Enable	유효	BOOL	TRUE, FALSE	FALSE	TRUE로 하면 오버라이드가 유효가 됩니다. FALSE로 하면 오버라이드는 '100%'로 되돌아갑니다.
VelFactor	속도 오버라이드값	LREAL	0~500	100	속도 오버라이드값을 지정합니다. 오버라이드값의 유효 범위는 '0.01~500.00'입니다. '500.00 이상'은 '500'으로 다루어지며 '0.01 미만(음수도 포함)'은 '0.01'로 다루어집니다. '0'을 지정한 경우에만 '0'으로 동작합니다. 단위는 [%]입니다.
AccFactor (Reserved)	가감속 속도 오버라이드값	LREAL	0~500	100	(Reserved)
JerkFactor (Reserved)	저크 오버라이드값	LREAL	0~500	100	(Reserved)

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Enabled	유효	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Enabled	이 명령을 실행하기 시작했을 때	• Enable이 FALSE인 경우 1주기 후 • Error가 TRUE가 되었을 때
사용 중	Enable의 기동	• Enable이 FALSE가 되었을 때 • Error가 TRUE가 되었을 때
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 축의 목표 속도에 대한 오버라이드를 변경합니다. 오버라이드를 변경해 동작 중인 목표 속도를 변경합니다.
- 오버라이드를 변경할 수 있는 명령은 다음 명령입니다.

MC_Move(위치 결정) 명령	MC_MoveJog(조그 이송) 명령
MC_MoveAbsolute(절대값 위치 결정) 명령	MC_MoveFeed(인터럽트 피딩 위치 결정) 명령
MC_MoveRelative(상대값 위치 결정) 명령	MC_MoveZeroPosition(고속 원점 복귀) 명령
MC_MoveVelocity(속도 제어) 명령	

- 새로운 목표 속도는 다음과 같습니다.

  변경 후의 목표 속도 = 현재 실행 중인 명령의 목표 속도 × 오버라이드

- 오버라이드의 단위는 [%]입니다. '100'으로 '100%'를 나타냅니다.
- 변경 후의 목표 속도가 축 파라메터의 [최고 속도]를 초과하는 경우는 최고 속도가 됩니다.
- 변경한 목표 속도에 대해 축을 가속 또는 감속합니다.
- 오버라이드에 '0'을 지정하면 목표 속도 '0'이 되고, 축 동작은 감속해 속도 '0'에서 동작 중이 됩니다. 동작 상태를 지속하면서 축 동작을 일시 정지하려면 오버라이드를 '0'으로 합니다. 이 때 축 변수의 Status. Discrete, Status. Continuous는 변화하지 않습니다.
- 오버라이드값은 Enable(유효)이 TRUE인 상태에서 명령 실행 시에 항상 반영됩니다.
- 오버라이드값은 동작 명령에 대한 정지 중에서의 기동, 모션 명령 재기동, 모션 명령 다중 기동으로 설정된 새로운 목표 속도에 대해 적용됩니다.
- Enable(유효)이 FALSE이면 오버라이드는 100(=100%)으로 되돌아갑니다.
- MC_SetOverride(오버라이드값 설정) 명령 실행 중에 축 이상이 발생하면 MC_SetOverride(오버라이드값 설정) 명령은 Enabled(유효)가 TRUE인 채로 지속됩니다.

사용상의 주의 사항

이 명령의 Enable(유효)을 FALSE로 하면 이 명령의 Enabled(유효) 및 Busy(실행 중)는 FALSE가 됩니다.
이 때 속도는 오버라이드 '100%'를 향해서 가속 또는 감속합니다.

참고

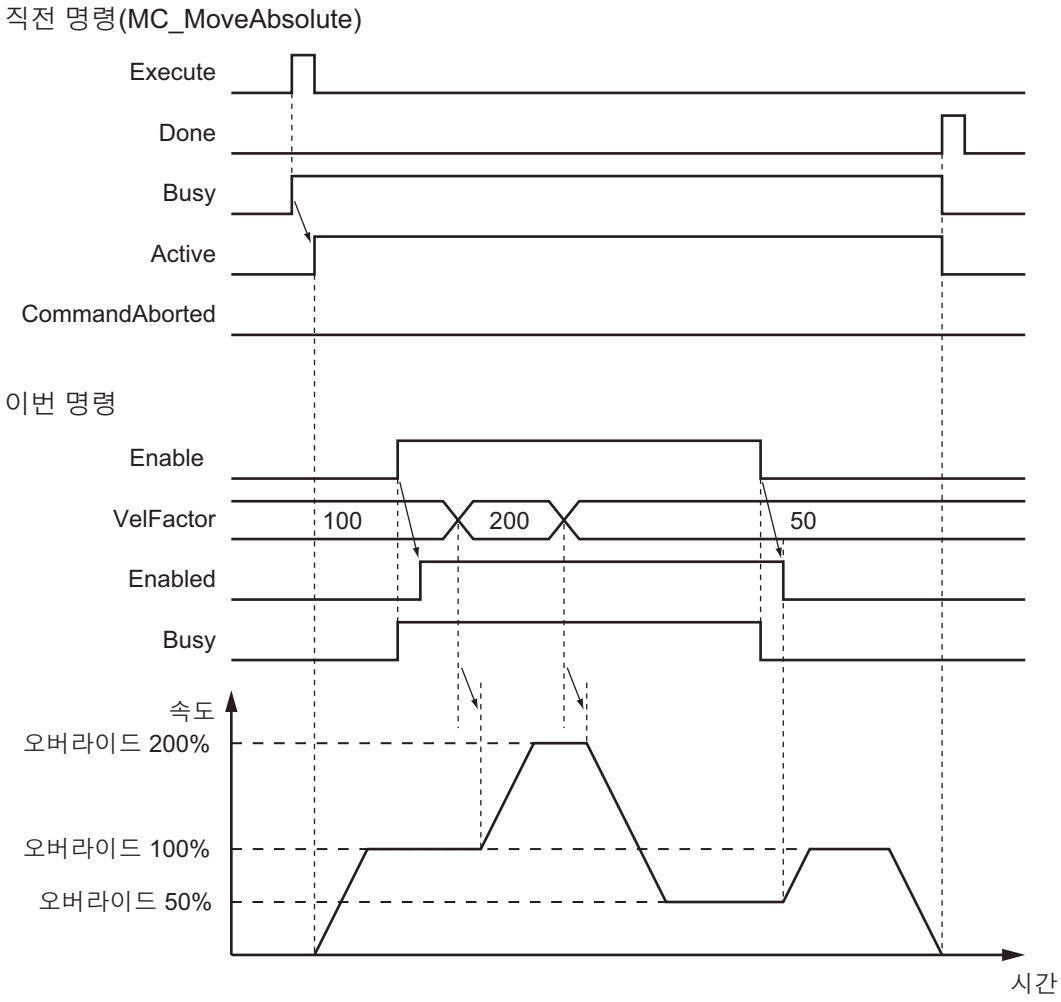
다른 명령에 대한 영향

이 명령은 각 명령의 목표 속도를 일시적으로 변경할 경우에 사용합니다.
따라서 목표 속도가 입력에 없는 명령이나 주기적 동기 속도 제어와 같이 목표 속도를 매주기 변경하는 명령에 대해서는 이 명령이 기능하지 않습니다.
단, 이 명령이 기능하지 않는 명령에 대해 MC_SetOverride(오버라이드값 설정) 명령을 유효로 한 경우에도 Enable(유효)이 TRUE인 채로 지속됩니다.

타이밍 차트

● MC_MoveAbsolute(절대값 위치 결정) 명령에 대한 오버라이드

MC_MoveAbsolute(절대값 위치 결정) 명령에 오버라이드 명령을 사용한 경우의 타이밍 차트 예를 제시합니다.

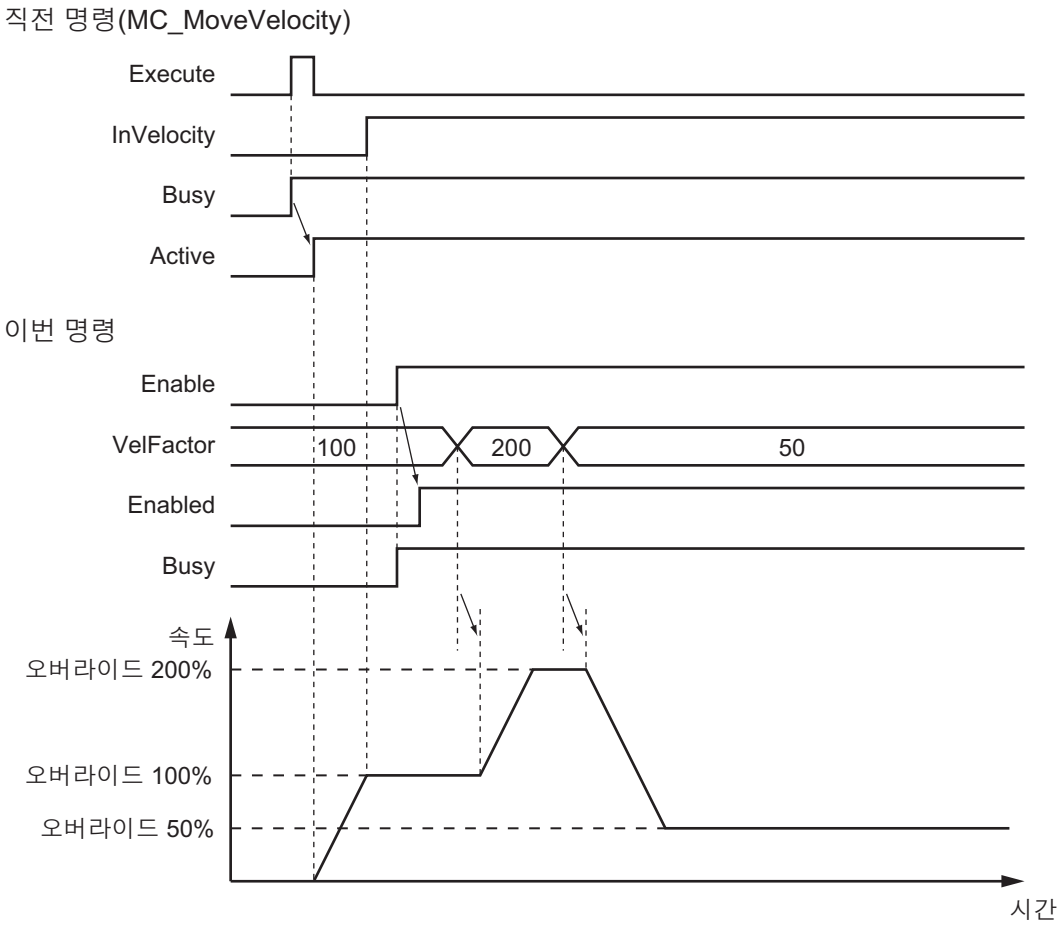


MC_SetOverride(오버라이드값 설정) 명령이 무효가 되면 오버라이드 100%의 목표 속도로 되돌아갑니다.

● MC_MoveVelocity(속도 제어) 명령에 대한 오버라이드

MC_MoveVelocity(속도 제어) 명령에 오버라이드 명령을 사용한 경우의 타이밍 차트 예는 다음과 같습니다.

InVelocity(목표 속도 도달)가 TRUE가 된 후에는 속도를 변경해도 InVelocity(목표 속도 도달)는 TRUE인 상태입니다.



모션 명령 재기동

입력이 'Enable 형식'인 명령에는 모션 명령 재기동은 없습니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령끼리의 다중 기동

MC_SetOverride(오버라이드값 설정) 명령 실행 중인 축에 다른 인스턴스의 MC_SetOverride(오버라이드값 설정) 명령을 기동하면 나중에 실행된 인스턴스가 우선 처리됩니다.

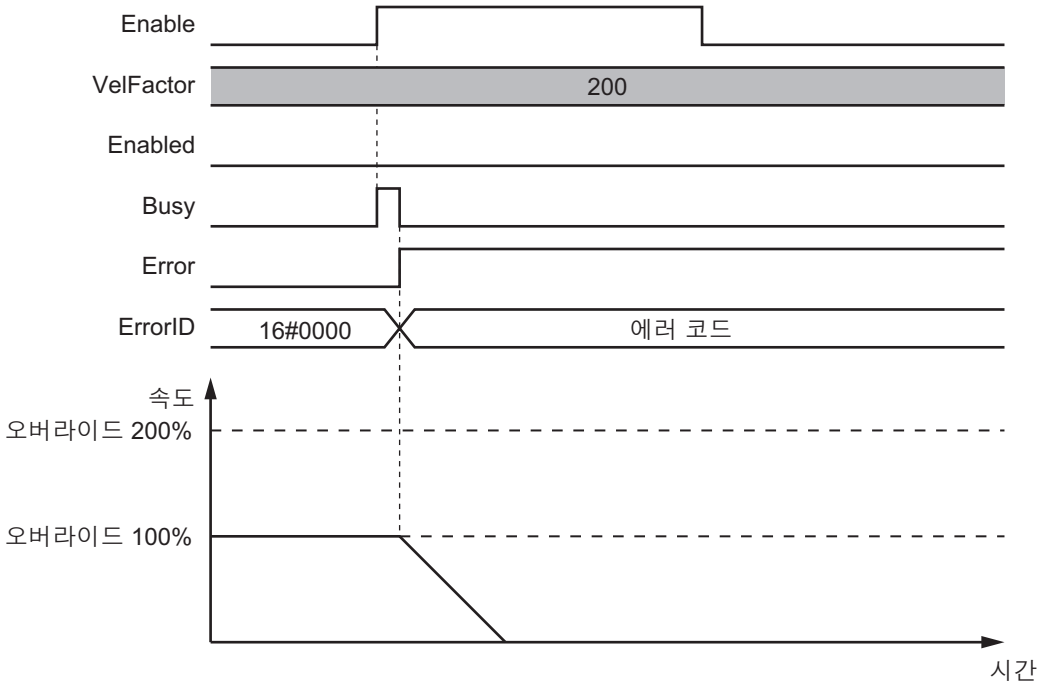
Enabled(유효)는 두 명령 모두 TRUE가 됩니다.

구체적인 동작으로는 나중에 실행된 인스턴스의 오버라이드 값이 유효가 됩니다. 나중에 실행된 인스턴스의 Enable(유효)을 FALSE로 하면 오버라이드는 무효가 됩니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 가벼운 이상이 발생하면 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.
이상 요인을 제거하면 Error(에러)는 FALSE가 됩니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_ResetFollowingError

명령 현재 위치와 피드백 현재 위치의 편차를 리셋합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_ResetFollowingError	편차 카운터 리셋	FB	MC_ResetFollowingError_instance MC_ResetFollowingError Axis — — Axis Execute — Done BufferMode — Busy Active CommandAborted Error ErrorID	MC_ResetFollowingError_instance (Axis := 《파라메터》, Execute := 《파라메터》, BufferMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
BufferMode	버퍼 모드 선택	_eMC_BUFFER_MODE	0: _mcAborting	0 *1	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting

*1. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	편차 카운터 리셋을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 MC_ResetFollowingError 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

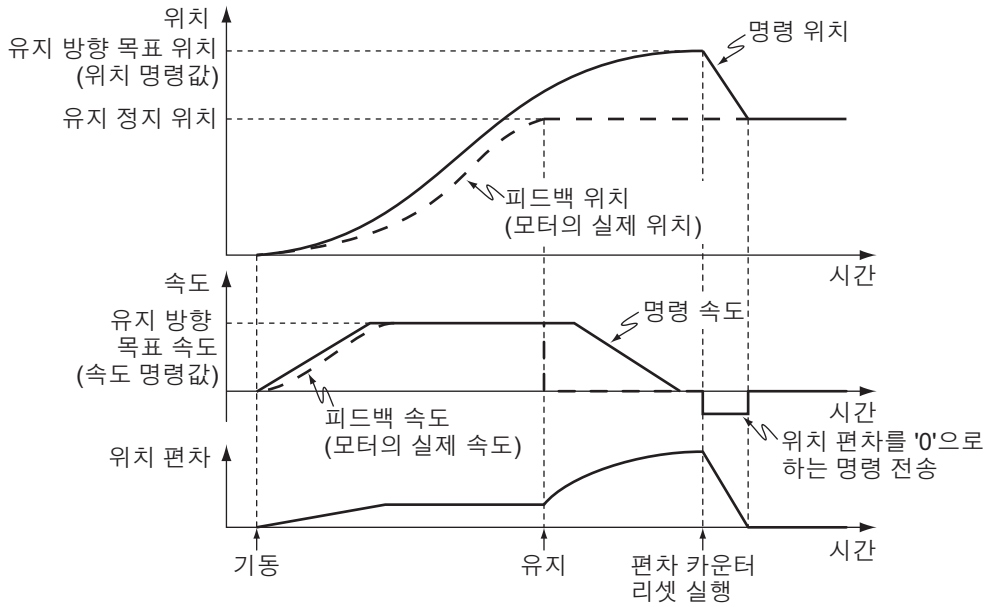
입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 이 명령은 주기적 동기 위치 모드에서 MC 기능 모듈의 명령 현재 위치와 피드백 현재 위치와의 편차를 '0'으로 하는 기능입니다.
- Execute(기동)의 기동을 검출하면 해당 시점에서의 피드백 현재 위치를 새로운 목표 위치로 명령 위치를 전달합니다.
  예를 들어 다음 그림과 같이 위치 편차가 발생하는 유지 동작에서 이 명령을 기동하면 반대 방향에 대한 위치 명령을 전달해서 위치 편차를 '0'으로 합니다.
  위치 편차가 발생한 명령은 CommandAborted(명령 중단)가 TRUE가 되고 명령 실행이 중지됩니다.



- 위치 편차를 '0'으로 하면 축 파라메터에 설정된 최고 속도를 이용해서 위치 명령을 전달합니다. 최대 가속도, 최대 감속도는 적용되지 않습니다.
- 새로운 목표 위치에 대한 명령이 완료되었을 때 출력 변수 'Done(완료)'이 TRUE가 됩니다.
- 이 명령은 위치 편차가 발생한 동작과는 역방향으로 명령 위치를 전달하는데, 축 파라메터의 '반전 시 동작'은 적용되지 않습니다.

사용상의 주의 사항

- 축 속도가 낮은 상태에서 이 명령을 기동하십시오.
  이 명령은 이전 명령(유지 방향)과는 반대 방향으로 명령값을 전달합니다. 따라서 축 속도가 큰 상태로 이 명령을 기동하면 기계에 충격을 주는 경우가 있습니다.
- 수직축 등 항상 일정하게 토크를 걸어야 하는 축에서 이 명령을 사용하는 경우는 이 명령 기동으로 토크 부족이 발생하지 않는 것을 확인한 후 사용하십시오.
- NX 시리즈 펄스 출력 유니트를 사용하는 경우, 펄스 출력 유니트에 연결된 서보 드라이버 내의 편차는 리셋되지 않습니다. 자세한 내용은 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"를 참조하십시오.
- **동기 제어의 주축에 대해 이 명령을 사용하는 경우**

  동기 제어의 주축에서 명령 위치를 동기 데이터로 사용하는 경우 주축에 대해 이 명령을 기동하면 종축은 기어비나 캠 데이터 변수에 따라 반전하므로 주의하십시오.
  주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● 사용 가능 축과 기동 조건

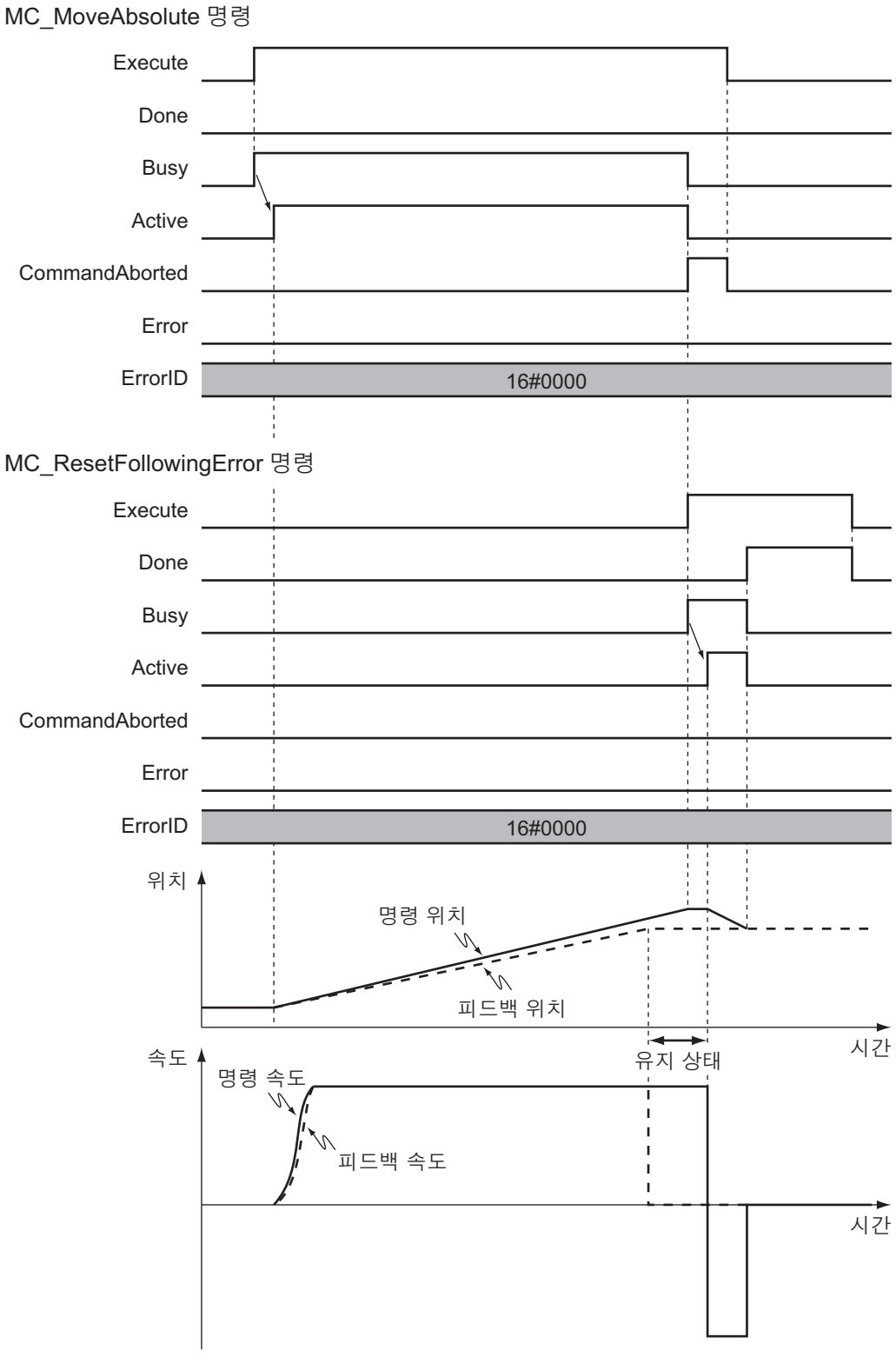
- 서보 축이나 가상 서보 축에서는 다음 경우에 사용 가능합니다.
  단축 위치 제어 중
  MC_MoveVelocity(속도 제어) 명령
  동기 제어 중
- 엔코더 축, 가상 엔코더 축에서는 기동 시 이상이 됩니다.

● 축 변수의 Status

축 변수의 Status는 Status.Stopping(감속 정지 중)이 TRUE가 됩니다.

타이밍 차트

MC_MoveAbsolute(절대값 위치 결정) 명령 기동 후 유지 상태에서 이 명령을 기동한 경우의 타이밍 차트는 다음과 같습니다.



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다.
각 축에 1개까지 버퍼링할 수 있습니다.
다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

MC_Stop(강제 정지) 명령과의 관계

MC_Stop(강제 정지) 명령 실행 중에 MC_ResetFollowingError(편차 카운터 리셋) 명령을 기동하면 MC_Stop(강제 정지) 명령의 Done이 TRUE가 되고 MC_ResetFollowingError(편차 카운터 리셋) 명령이 실행됩니다.

● 이 명령 실행 중의 다른 명령 기동

이 명령 실행 중에는 감속 정지 중이 됩니다.
따라서 감속 정지 중에 다중 기동 가능한 명령을 기동하십시오.
기동할 수 없는 명령에서는 이 명령의 CommandAborted(실행 중단)가 TRUE가 됩니다.

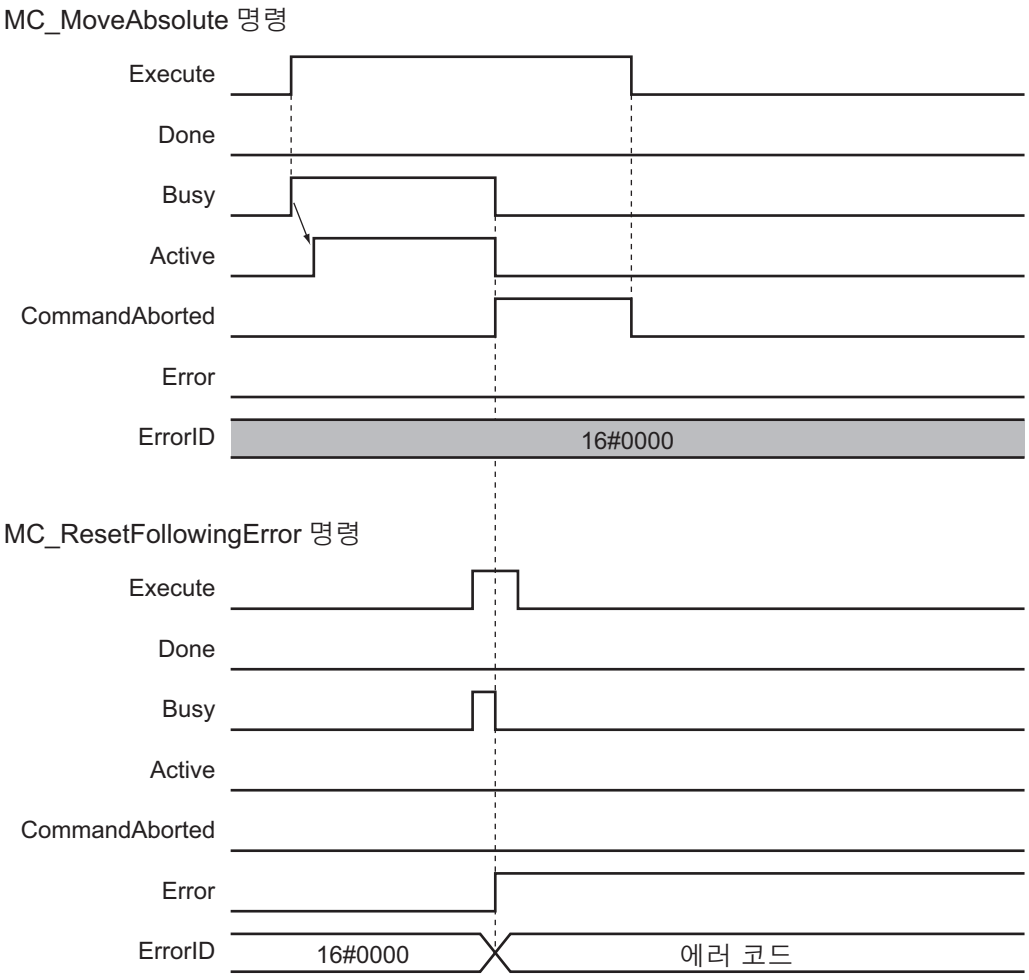
MC_Stop(강제 정지) 명령과의 관계

MC_ResetFollowingError(편차 카운터 리셋) 명령 실행 중에 MC_Stop(강제 정지) 명령을 기동하면 MC_ResetFollowingError(편차 카운터 리셋) 명령의 CommandAborted(실행 중단)가 TRUE가 되고 MC_Stop(강제 정지) 명령이 실행됩니다. 단, MC_Stop(강제 정지) 명령의 감속도는 적용되지 않고 즉시 정지합니다.

이상

이 명령 실행 중에 이상이 발생하면 즉시 정지되고 CommandAborted(실행 중단)가 TRUE가 됩니다. 또한 이 명령 기동 시에 이상을 감지하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

MC_MoveAbsolute(절대값 위치 결정) 명령 기동 후 유지 상태에서 이 명령을 기동한 상태로 이상이 발생한 경우의 타이밍 차트는 다음과 같습니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_CamIn

지정한 캠 테이블을 사용해 캠 동작을 시작합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_CamIn	캠 동작 시작	FB	MC_CamIn_instance MC_CamIn Master — — Master Slave — — Slave CamTable — — CamTable Execute — InCam Periodic — InSync StartMode — EndOfProfile StartPosition — Index MasterStartDistance — Busy MasterScaling — Active SlaveScaling — CommandAborted MasterOffset — Error SlaveOffset — ErrorID ReferenceType Direction CamTransition BufferMode	MC_CamIn_instance (Master := 《파라메터》, Slave := 《파라메터》, CamTable := 《파라메터》, Execute := 《파라메터》, Periodic := 《파라메터》, StartMode := 《파라메터》, StartPosition := 《파라메터》, MasterStartDistance := 《파라메터》, MasterScaling := 《파라메터》, SlaveScaling := 《파라메터》, MasterOffset := 《파라메터》, SlaveOffset := 《파라메터》, ReferenceType := 《파라메터》, Direction := 《파라메터》, CamTransition := 《파라메터》, BufferMode := 《파라메터》, InCam => 《파라메터》, InSync => 《파라메터》, EndOfProfile => 《파라메터》, Index => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Periodic	반복 모드	BOOL	TRUE, FALSE	FALSE	지정된 캠 테이블을 반복 실행할지 1회만 실행할지를 지정합니다. TRUE : 반복 FALSE : 반복 안 함
StartMode	시작 위치 방식 지정	_eMC_ START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	0 *1	MasterStartDistance(주축 추종 거리)가 사용할 좌표 계통을 지정합니다. 0 : 절대 위치 1 : 상대 위치
StartPosition	캠 테이블 시작 위치	LREAL	음수, 양수, '0'	0	캠 테이블의 시작점(위상=0)을 주축의 절대 위치로 지정합니다. 단위는 [명령 단위]입니다. *2

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
MasterStart Distance	주축 추종 거리	LREAL	음수, 양수, '0'	0	종축이 캠 동작을 시작할 때의 주축 위치를 지정합니다. StartMode(시작 위치 방식 지정)에서 [절대 위치 결정]을 지정한 경우는 주축의 절대 위치, [상대 위치 결정]을 지정한 경우는 StartPosition(캠 테이블 시작 위치)으로부터의 상대량으로 지정합니다. 단위는 [명령 단위]입니다. *2
Master Scaling	주축 계수	LREAL	양수값(>0.0)	1.0	지정된 스케일로 주축의 위상을 확대/축소합니다.
SlaveScaling	종축 계수	LREAL	양수값(>0.0)	1.0	지정된 스케일로 종축의 변위를 확대/축소합니다.
MasterOffset	주축 오프셋	LREAL	음수, 양수, '0'	0	지정된 오프셋값으로 주축 위상을 이동합니다.
SlaveOffset	종축 오프셋	LREAL	음수, 양수, '0'	0	지정된 오프셋값으로 종축 변위를 이동합니다.
Reference Type	위치 종류 선택	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *1	주축의 위치 종류를 지정합니다. 0 : 명령 위치(직전 프라이머리 주기에서의 계산값) 1 : 피드백 위치(동일 프라이머리 주기에서의 취득값) 2 : 명령 위치(동일 프라이머리 주기에서의 계산값)
Direction	방향 선택	_eMC_ DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 4: _mcNoDirection	4 *1	지정된 방향으로 주축이 동작하고 있는 경우에만 종축이 캠 동작을 실행합니다. 지정된 방향과 반대 방향으로 주축이 동작하고 있으면 종축은 캠 동작을 정지합니다. 0 : 정방향 지정 2 : 역방향 지정 4 : 방향 지정 없음
Cam Transition (Reserved)	캠 트랜지션 선택	_eMC_CAM_ TRANSITION	0: _mcCTNone	0 *1	(Reserved)
BufferMode	버퍼 모드 선택	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow *3 3: _mcBlendingPrevious *3 4: _mcBlendingNext *3 5: _mcBlendingHigh *3	0 *1	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered 2: Blending low 3: Blending previous 4: Blending next 5: Blending high

*1. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*2. 명령 단위에 대해서는 📖 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*3. CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용할 수 있습니다. 4종류의 Blending 중 어느 것을 지정해도 동일한 동작이 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	단위	내 용
InCam	캠 동작 중	BOOL	TRUE, FALSE	–	캠 테이블의 시작점이 실행되었을 때 TRUE가 됩니다.
InSync	동기 중	BOOL	TRUE, FALSE	–	주축에 동기했을 때 TRUE가 됩니다.
EndOfProfile	캠 사이클 완료	BOOL	TRUE, FALSE	–	캠 테이블의 종료점이 실행되었을 때 TRUE가 됩니다.
Index	인덱스	UINT	0또는 양수값	–	캠 데이터의 인덱스 번호를 출력합니다. *1
Busy	실행 중	BOOL	TRUE, FALSE	–	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	–	제어 중에 TRUE가 됩니다.
Command Aborted	실행 중단	BOOL	TRUE, FALSE	–	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	–	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*2	–	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. InCam(캠 동작 중)이 FALSE인 동안 출력은 '0'이 됩니다.
*2. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
InSync	종축이 캠 동작을 시작했을 때	• Periodic이 FALSE인 경우 EndOfProfile이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
EndOfProfile	캠 테이블 종료점의 위상과 변위가 명령 위치로서 출력된 주기	EndOfProfile이 TRUE가 된 다음 1주기 후
Busy	Execute의 기동	• Periodic이 FALSE인 경우 EndOfProfile이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
InCam	주축이 StartPosition(캠 테이블 시작 위치)을 통과했을 때	• Periodic이 FALSE인 경우 EndOfProfile이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Periodic이 FALSE인 경우 EndOfProfile이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때 • MC_CamOut 명령이 기동되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Master	주축	_sAXIS_REF	–	주축을 지정합니다. *1
Slave	종축	_sAXIS_REF	–	종축을 지정합니다. *1
CamTable	캠 테이블	ARRAY[0..N] OF _sMC_CAM_REF	–	캠 데이터 구조체 _sMC_CAM_REF 형식의 배열 변수를 캠 테이블로 지정합니다. *2

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.
*2. 배열 요소 [N]은 Sysmac Studio에 의해 자동 설정됩니다. Sysmac Studio의 캠 편집기로 작성한 캠 데이터 변수를 지정하십시오.

사용상의 주의 사항

주축과 종축에 같은 축을 지정하면 가벼운 이상 수준의 '주축 종축 동일(에러 코드: 5436 Hex)' 이상이 됩니다.

기능 설명

- 이 명령은 위상(주축)과 변위(종축)가 캠 테이블에 따라 동기하는 캠 동작을 실행합니다.
- 이 명령으로 지정하는 캠 테이블은 미리 캠 편집기를 사용해서 작성하고 컨트롤러 본체에 다운로드해 두어야 합니다.
- Execute(기동)가 기동하면 이 명령이 기동됩니다.

사용상의 주의 사항

캠 테이블은 Sysmac Studio의 캠 편집기로 작성한 캠 데이터 변수를 사용해야 합니다.

참고

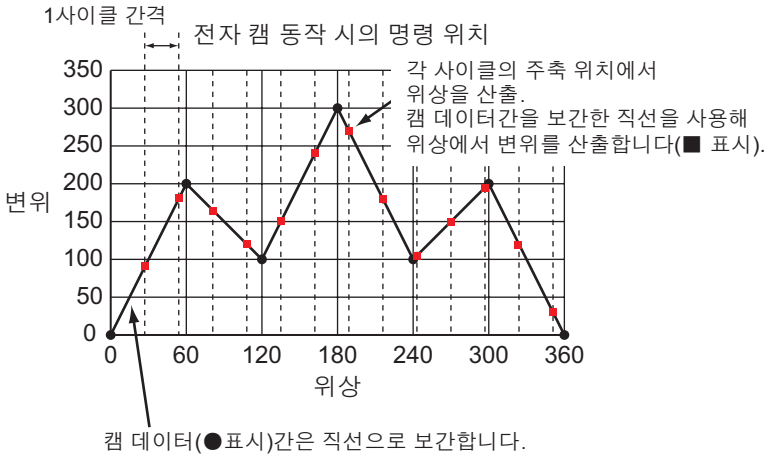
'다운로드'는 Sysmac Studio의 '동기' 기능을 사용합니다.

캠 테이블에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

캠 테이블은 위상과 변위 모두 '0.0'인 시작점으로부터의 상대량으로 전달합니다. 또한 주축, 종축에 대한 명령 위치는 2개의 캠 데이터 사이를 직선으로 보간하고 위상(주축)에 해당하는 변위(종축)를 구합니다. 캠 데이터 수가 적으면 위상의 간격이 커지고 캠 동작의 정확도가 낮아집니다. 반대로 캠 데이터 수가 많으면 위상의 간격이 작아져 캠 동작의 정확도가 높아집니다.

캠 테이블

	위상	변위	
시작점 →	0	0	← Index = 0
캠 데이터 →	60	200	1
	120	100	2
	180	300	3
	240	100	4
	300	200	5
종료점 →	360	0	6



사용상의 주의 사항

- 이 명령에서는 기동 시에 위상이 오름차순으로 나열되어 있는지를 확인합니다. 위상이 오름차순으로 나열되어 있지 않으면 이상을 감지합니다.
  이 명령 기동 후에 캠 데이터를 변경한 경우 위상이 오름차순으로 나열되어 있는지 확인할 때는 MC_SetCamTableProperty(캠 테이블 속성 갱신) 명령을 실행하십시오.
  캠 동작 중에 위상을 변경하려면 변경하기 전에 오름차순으로 되어 있는 것을 확인한 후 변경하시기 바랍니다. 오름차순으로 되어 있지 않으면 캠 동작이 도중에 정지하는 경우가 있습니다.
- 캠 데이터 변수는 글로벌 변수로 복수의 태스크에서 캠 데이터 변수값을 참조하거나 변경할 수 있습니다. 복수의 태스크로부터 캠 데이터 변수값을 변경하려면 복수의 태스크에 의한 재기입 처리로 경쟁이 발생하지 않도록 프로그램을 작성하십시오.
- '글로벌 변수의 태스크 간 배타적 제어' 기능을 이용해서 캠 데이터 변수를 배타적으로 제어하려면 배타 대상인 캠 데이터 변수를 이용해서 모션 명령을 기동하지 마십시오. 가벼운 수준의 이상인 '캠 테이블 지정 오류(에러 코드: 5439 Hex)'가 발생합니다.

안전상의 중요 사항

이 명령을 사용하는 경우는 Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하지 마십시오.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하면 Slave(종축)가 급격하게 추종할 수 있어 위험합니다.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 사용하려면 Master(주축)와 Slave(종축)의 관계를 일단 해제하고 나서 실행하십시오.
주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● 명령 기동 조건

이 명령은 주축이 정지 중이거나 위치 제어 중, 속도 제어 중, 동기 제어 중 가운데 어떤 상태라도 기동할 수 있습니다.
종축에 대해서는 🕮 '모션 명령 재기동(3-181페이지)', 🕮 '모션 명령 다중 기동(3-182페이지)'을 참조하십시오.

● 소프트웨어 리미트

캠 동작 중에 종축이 소프트웨어 리미트를 초과하면 이상이 됩니다.

● 캠 데이터 변수

캠 데이터 변수는 캠 데이터 구조체를 배열 선언으로 사용합니다. 캠 데이터 구조체는 미리 다음과 같이 형식이 선언되어 있습니다.

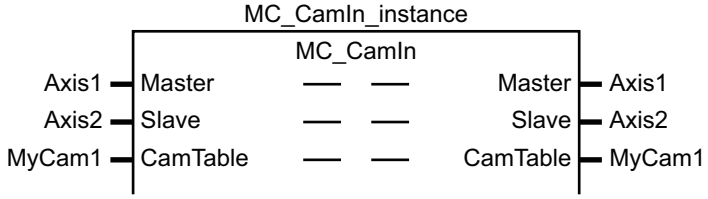
```
TYPE
        (* 캠 데이터 구조체 *)
        _sMC_CAM_REF :
        STRUCT
                Phase       : REAL;         (* 위상 *)
                Distance    : REAL;         (* 변위 *)
        END_STRUCT;
END_TYPE
```

Sysmac Studio로 캠 데이터 변수를 작성합니다.
캠 데이터 변수의 명칭인 캠 테이블명은 임의로 결정할 수 있습니다.

예를 들어 캠 테이블명 'MyCam1', 캠 데이터 수를 1000개로 하면 Sysmac Studio에서 다음과 같이 자동으로 변수가 선언됩니다.

```
VAR
        (* 캠 테이블 *)
        MyCam1      :       ARRAY [0..999] OF _sMC_CAM_REF;
END_VAR
```

이 명령으로 'MyCam1'을 지정하면 다음과 같이 기술합니다. 여기에서는 주축을 Axis1, 종축을 Axis2로 합니다.



지정한 캠 테이블이 컨트롤러에 존재하지 않으면 이상을 감지합니다. 또한 복수 축에서 동일한 캠 테이블을 지정할 수 있습니다.

캠 데이터 변수값은 사용자 프로그램으로 재기입할 수 있습니다. 단, 재기입한 값은 컨트롤러의 전원 OFF, Sysmac Studio로부터의 다운로드 실행 시에 지워집니다.
CPU 유니트 전원을 켠 후 및 다운로드 완료 후에는 Sysmac Studio로부터 다운로드된 값으로 되돌아갑니다.
재기입한 값을 저장하려면 MC_SaveCamTable(캠 테이블 저장) 명령을 사용하십시오.

CPU 유니트 동작 모드의 변경 시 재기입한 캠 데이터 변수값은 유지됩니다.

참고

- 캠 데이터 변수는 네트워크 비공개가 됩니다.
  예를 들어 'MyCam1[10].Phase'나 'MyCam1 [10].Distance'는 Sysmac Studio의 Watch View로 볼 수는 있지만, EIP를 경유해 다른 컨트롤러에서 액세스할 수는 없습니다.
- '다운로드'는 Sysmac Studio의 '동기' 기능을 사용합니다.

● 캠 동작 시작

명령 기동 후, 주축이 StartPosition(캠 테이블 시작 위치)에 도달하는 것을 기다립니다.
주축이 StartPosition(캠 테이블 시작 위치)을 통과하면 캠 테이블의 시작점이 실행되고, 출력 변수 InCam(캠 동작 중)이 TRUE가 됩니다.
캠 테이블의 위상과 변위는 시작점을 제로로 하는 상대량으로 부여되므로 각 위상에서 각 축의 절대 위치는 캠 테이블 시작점에서 각 축의 절대 위치를 기점으로 하는 상대값이 됩니다. 예를 들어 주축의 카운트 모드가 0~360° 인 로터리 모드에서 StartPosition(캠 테이블 시작 위치)=60으로 합니다. 주축의 절대 위치는 다음과 같이 캠 테이블의 위상에 StartPosition을 더한 값이 되며, 종축의 절대 위치는 캠 테이블의 변위에 캠 테이블 시작점에서 종축의 절대 위치를 더한 값이 됩니다.

캠 테이블

	위상	변위
시작점 →	0	0
캠 데이터 →	60	200
	120	100
	180	300
	240	100
	300	200
종료점 →	360	0

StartPosition=60

각 축의 절대 위치

주축	종축
60	0＋캠 테이블 시작점에서의 종축 절대 위치
120	200＋캠 테이블 시작점에서의 종축 절대 위치
180	100＋캠 테이블 시작점에서의 종축 절대 위치
240	300＋캠 테이블 시작점에서의 종축 절대 위치
300	100＋캠 테이블 시작점에서의 종축 절대 위치
360	200＋캠 테이블 시작점에서의 종축 절대 위치
60	0＋캠 테이블 시작점에서의 종축 절대 위치

그리고 MasterStartDistance(주축 추종 거리)를 통과하면 종축의 캠 동작이 시작되고, 출력 변수 InSync(동기 중)가 TRUE가 됩니다.
MasterStartDistance(주축 추종 거리)는 절대 위치 또는 StartPosition(캠 테이블 시작 위치)에서의 상대 거리로 지정합니다. 절대 위치 또는 상대 위치는 StartMode(시작 위치 지정 방식)로 설정합니다.

예 1) MasterStartDistance의 차이에 따른 종축 동작의 차이

예를 들어 동일한 캠 테이블, 동일한 주축을 사용합니다.
캠 테이블 설정은 다음과 같습니다.

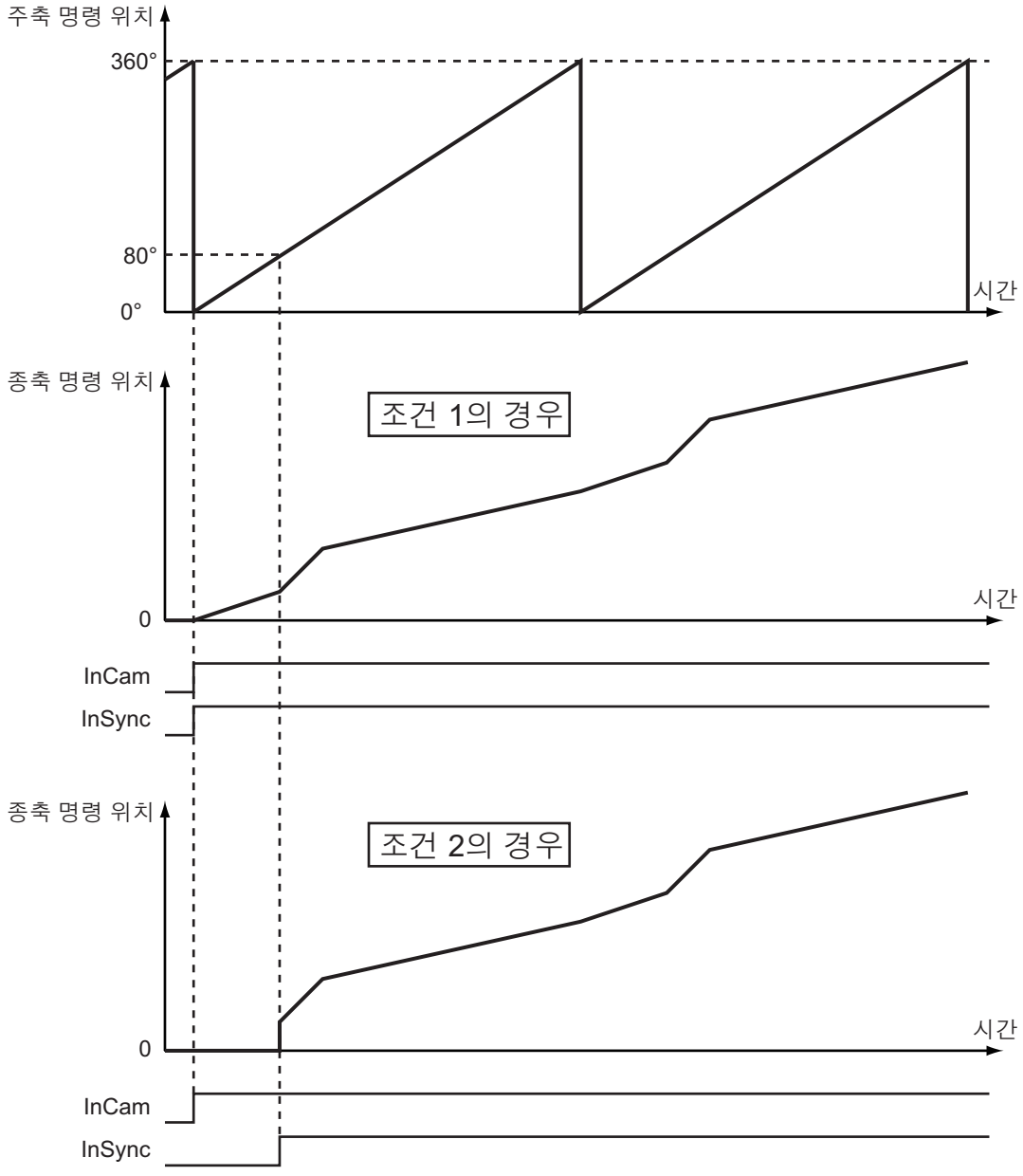
주축	종축	캠 곡선	연결 속도	연결 가속도	위상 간격 폭
0.000	0.000	–	–	–	–
80.000	80.000	직선	360.000	0.000	0.010
120.000	200.000	직선	1080.000	0.000	0.010
360.000	360.000	직선	240.000	0.000	0.010

캠 동작의 시작 조건은 다음과 같습니다.

입력 변수	조건 1	조건 2
Periodic(반복 모드)	TRUE(반복)	TRUE(반복)
StartMode(시작 위치 방식 지정)	_mcRelativePosition(상대 위치)	_mcRelativePosition(상대 위치)
StartPosition(캠 테이블 시작 위치)	0	0
MasterStartDistance(주축 추종 거리)	0	80

조건 1에서는 주축이 0°를 통과하면 출력 변수 InCam(캠 동작 중)과 출력 변수 InSync(동기 중)의 양쪽이 TRUE가 되어 종축이 캠 동작을 시작합니다.
조건 2에서는 주축이 0°를 통과하면 출력 변수 InCam(캠 동작 중)이 TRUE가 되고 다시 80° (주의)를 통과하면 출력 변수 InSync(동기 중)가 TRUE가 되어 종축이 캠 동작을 시작합니다. 조건 2의 경우, 캠 테이블의 중간부터 캠 동작을 시작하므로 종축은 급격하게 가속됩니다.

(주의) StartMode가 _mcRelativePosition이므로 StartPosition + MasterStartDistance=80°가 캠 동작 시작 위치가 됩니다.



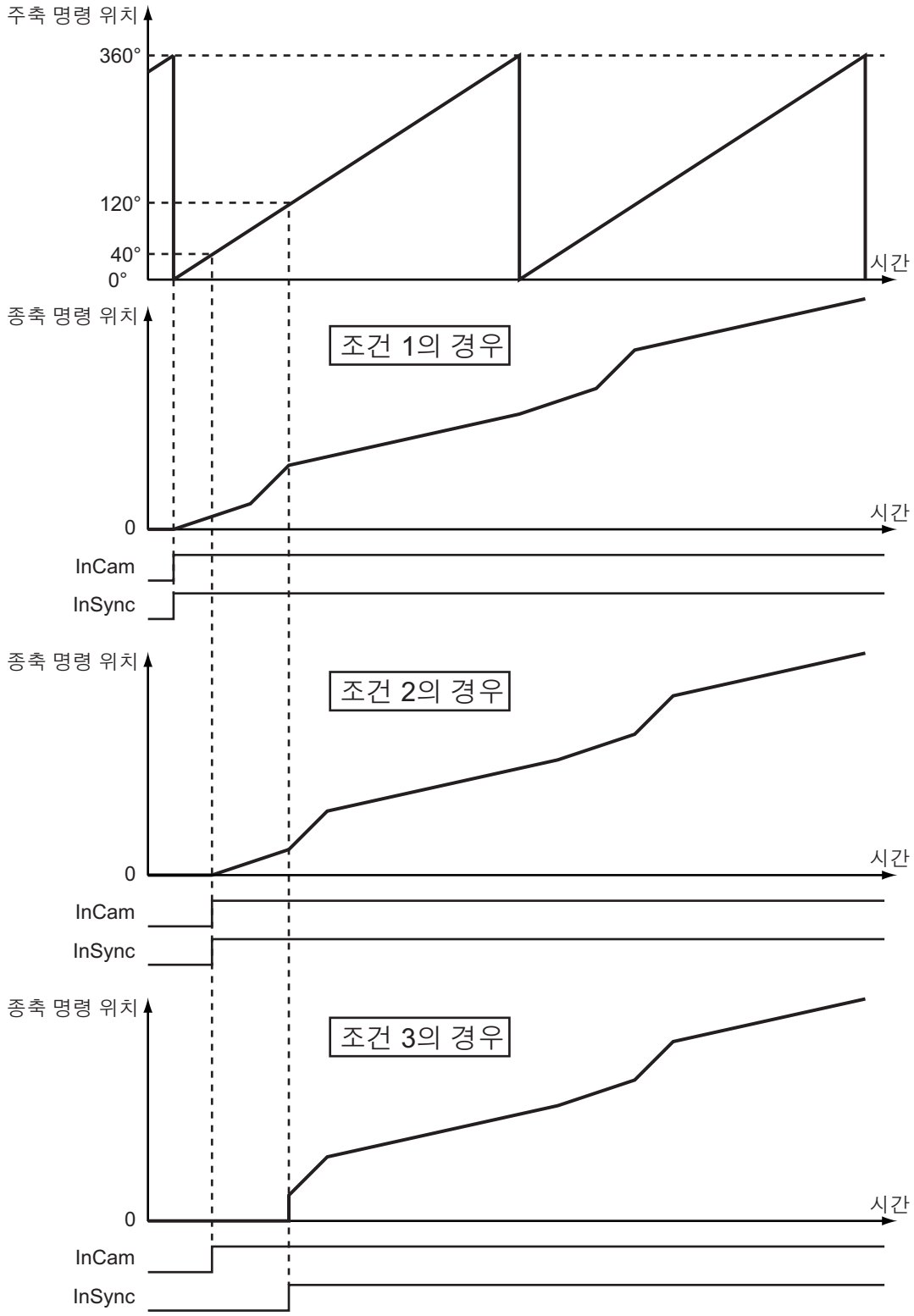
예 2) StartPosition과 MasterStartDistance의 차이에 따른 종축 동작의 차이

캠 테이블의 설정은 앞에서 설명한 예와 동일합니다. 캠 동작의 시작 조건은 다음과 같습니다.

입력 변수	조건 1	조건 2	조건 3
Periodic(반복 모드)	TRUE(반복)	TRUE(반복)	TRUE(반복)
StartMode(시작 위치 방식 지정)	_mcRelativePosition (상대 위치)	_mcRelativePosition (상대 위치)	_mcRelativePosition (상대 위치)
StartPosition(캠 테이블 시작 위치)	0	40	40
MasterStartDistance(주축 추종 거리)	0	0	80

조건 1에서는 주축이 0°를 통과하면 출력 변수 InCam(캠 동작 중)과 출력 변수 InSync(동기 중)의 양쪽이 TRUE가 되어 종축이 캠 동작을 시작합니다.
조건 2에서는 StartPosition(캠 테이블 시작 위치)으로 지정한 40°를 주축이 통과하면 출력 변수 InCam(캠 동작 중)과 출력 변수 InSync(동기 중)의 양쪽이 TRUE가 되고 종축이 캠 동작을 시작합니다.
조건 3에서는 주축이 40°를 통과하면 출력 변수 InCam(캠 동작 중)이 TRUE가 되고 다시 120°(주의)를 통과하면 출력 변수 InSync(동기 중)가 TRUE가 되어 종축이 캠 동작을 시작합니다.

(주의) StartMode가 _mcRelativePosition이므로 StartPosition + MasterStartDistance=120°가 캠 동작 시작 위치가 됩니다.

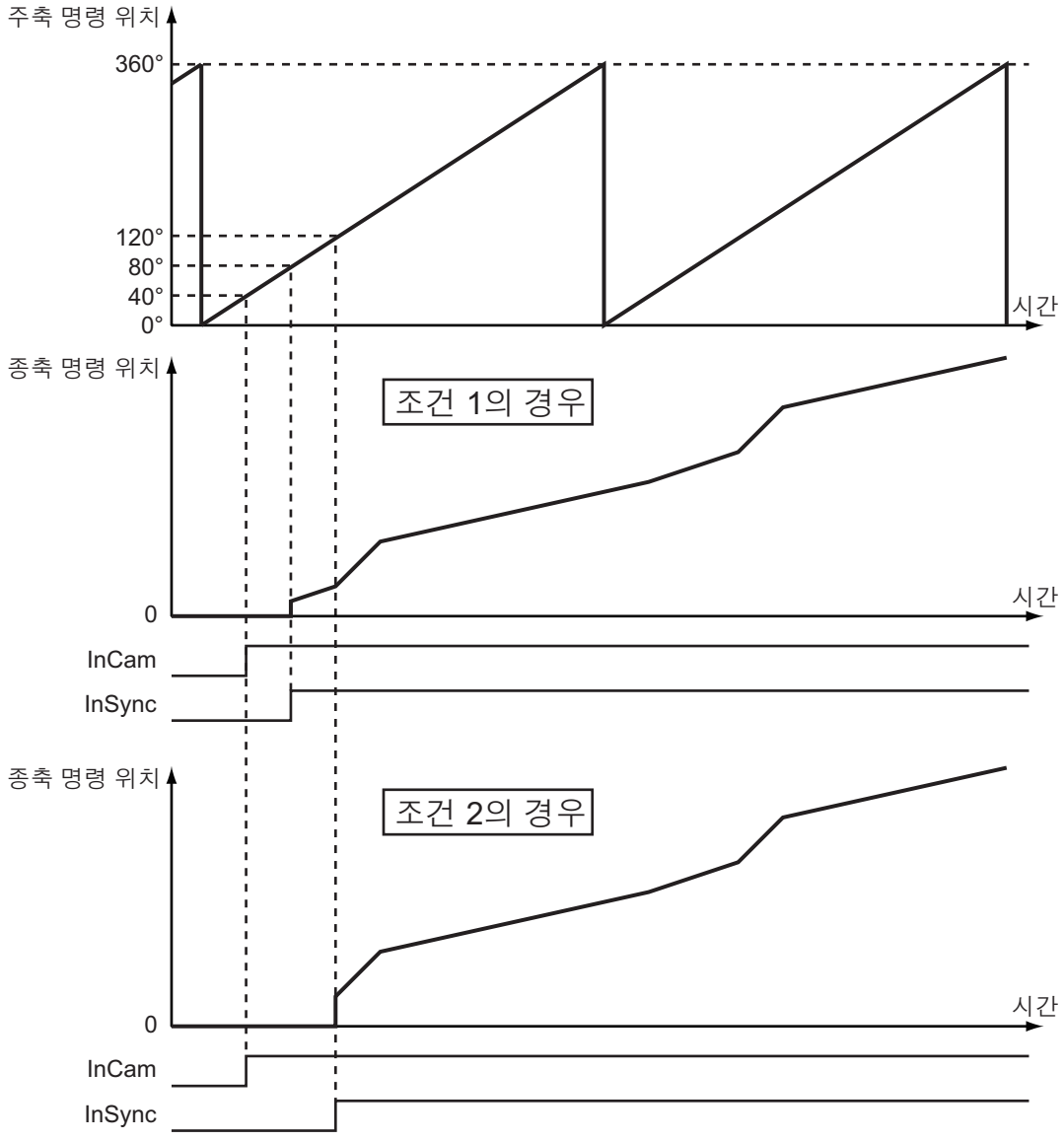


예 3) StartMode(시작 위치 방식 지정)의 차이에 따른 종축의 캠 동작 시작의 차이

StartMode(시작 위치 방식 지정)에 따라 MasterStartDistance(주축 추종 거리)에 지정한 값을 절대 위치로 취급할지, 상대 위치로 취급할지 지정할 수 있습니다.
StartMode(시작 위치 방식 지정)의 차이에 따른 종축의 캠 동작 시작의 차이에 대해 설명합니다. 캠 테이블의 설정은 앞에서 설명한 예와 동일합니다.
캠 동작의 시작 조건은 다음과 같습니다.

입력 변수	조건 1	조건 2
Periodic(반복 모드)	TRUE(반복)	TRUE(반복)
StartMode(시작 위치 방식 지정)	_mcAbsolutePosition(절대 위치)	_mcRelativePosition(상대 위치)
StartPosition(캠 테이블 시작 위치)	40	40
MasterStartDistance(주축 추종 거리)	80	80

조건 1과 조건 2 모두 주축이 40°를 통과하면 출력 변수 InCam(캠 동작 중)이 TRUE가 됩니다. 조건 1에서는 StartMode(시작 위치 방식 지정)가 _mcAbsolutePosition(절대 위치)이므로 주축이 80°를 통과하면 출력 변수 InSync(동기 중)가 TRUE가 되어 종축이 캠 동작을 시작합니다.
조건 2에서는 StartMode(시작 위치 방식 지정)가 _mcRelativePosition(상대 위치)이므로 주축이 120°(=40°+80°)를 통과하면 출력 변수 InSync(동기 중)가 TRUE가 되어 종축이 캠 동작을 시작합니다.



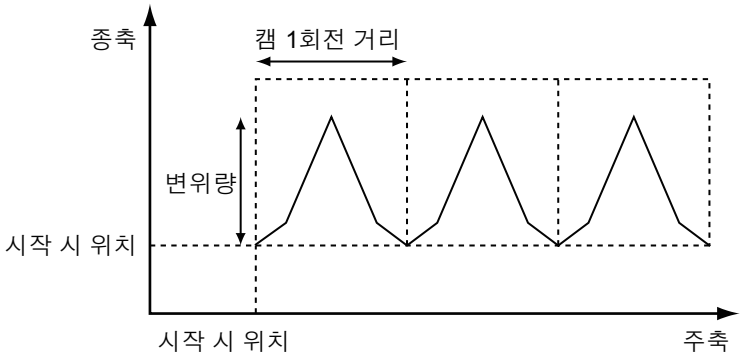
사용상의 주의 사항

캠 동작을 실행하려면 Sysmac Studio의 캠 편집기를 사용해 캠 프로파일을 작성한 후 CPU 유니트에 다운로드해 두어야 합니다.
'다운로드'는 Sysmac Studio의 '동기' 기능을 사용합니다.

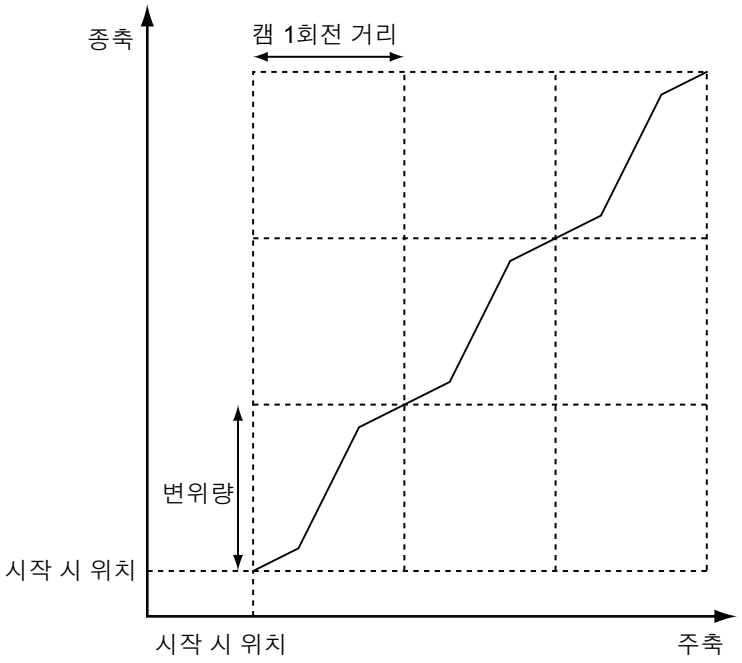
● Periodic(반복 모드)

Periodic(반복 모드)에서 TRUE[반복]를 지정하면 캠 테이블의 시작점부터 종료점까지 캠 동작을 반복 실행합니다.
FALSE[반복 안 함]으로 지정하면 캠 테이블의 종료점까지 실행하고 캠 동작을 종료합니다.
TRUE[반복]일 때 캠 테이블의 시작점과 종료점에서 종축의 스트로크 위치가 동일하면 왕복 캠으로 동작합니다(왕복 캠 동작 참조). 또한 시작점과 종료점에서 종축의 스트로크 위치가 다르면 이송 캠으로 동작합니다(이송 캠 동작 참조).
또한 아래 그림에서는 가로축이 주축의 위치, 세로축이 종축의 위치를 나타냅니다.

왕복 캠 동작



이송 캠 동작



● EndOfProfile(캠 사이클 완료)

캠 테이블의 종료점에서 정의된 위상, 변위를 사용한 캠 동작의 명령값을 출력한 주기로, EndOfProfile(캠 사이클 완료)을 1주기 동안 TRUE로 합니다.

StartPosition(캠 테이블 시작 위치)에는 주축의 절대 위치를 설정하며, 캠 테이블은 해당 위치로부터의 상대 테이블이 됩니다.

EndOfProfile(캠 사이클 완료)은 캠 테이블의 종단을 나타내는 출력입니다.

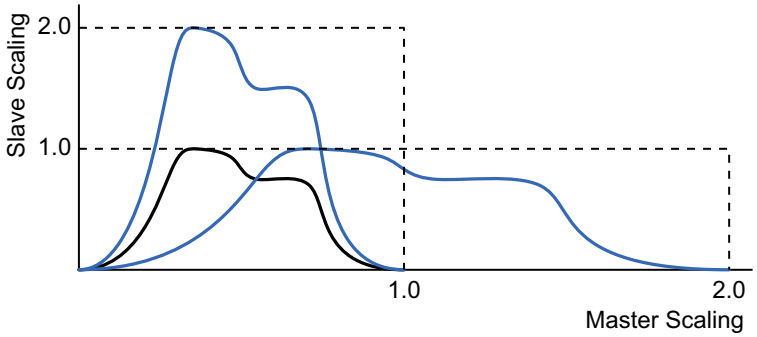
● 캠 동작의 종료

캠 동작을 도중에 종료하려면 MC_CamOut(캠 동작 해제) 명령 또는 MC_Stop(강제 정지) 명령을 사용합니다.

● 계수(스케일링)

지정된 캠 테이블의 주축 위상, 종축 변위 각각에 대해 지정된 스케일로 확대 및 축소할 수 있습니다.

주축과 종축에 각각 다른 배율을 설정할 수 있습니다.

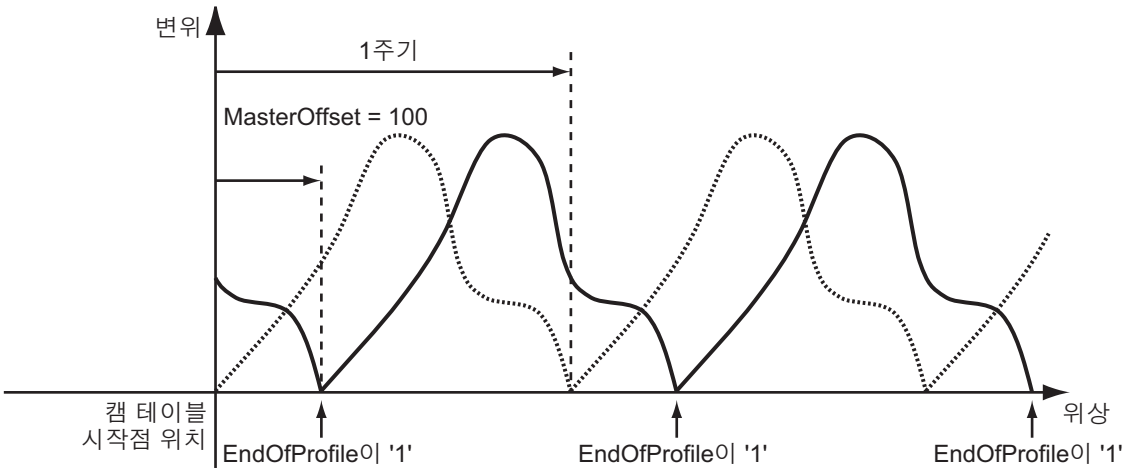


● 오프셋

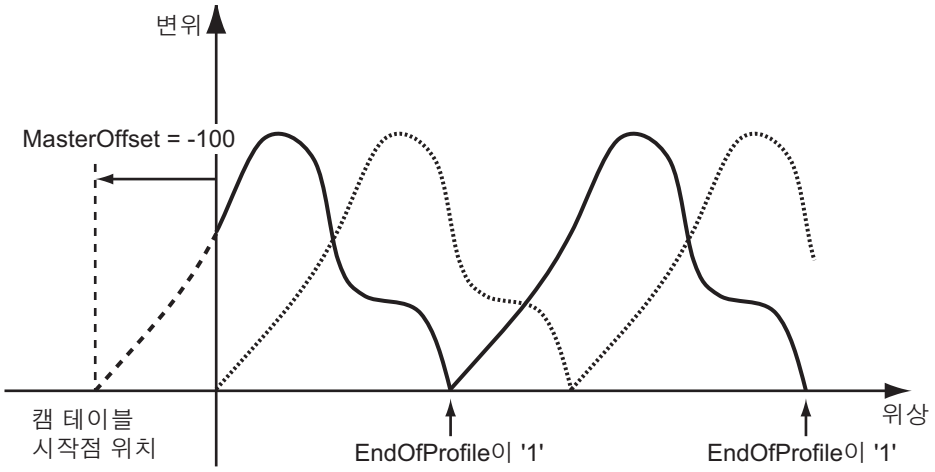
지정된 캠 테이블에 대해 오프셋량만큼 위상과 변위를 이동시키는 기능입니다.

주축 위상, 종축 변위에 대해 각각 다른 오프셋량을 지정할 수 있습니다.

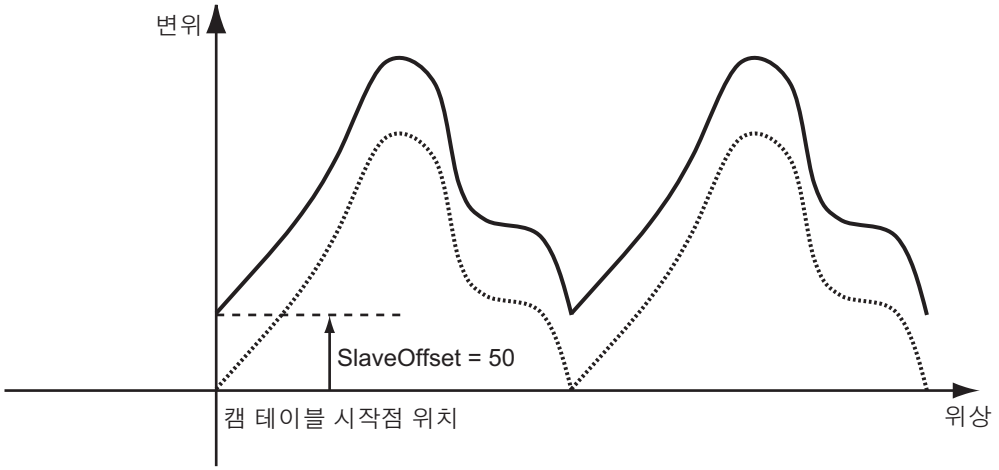
MasterOffset(주축 오프셋) > 0인 경우



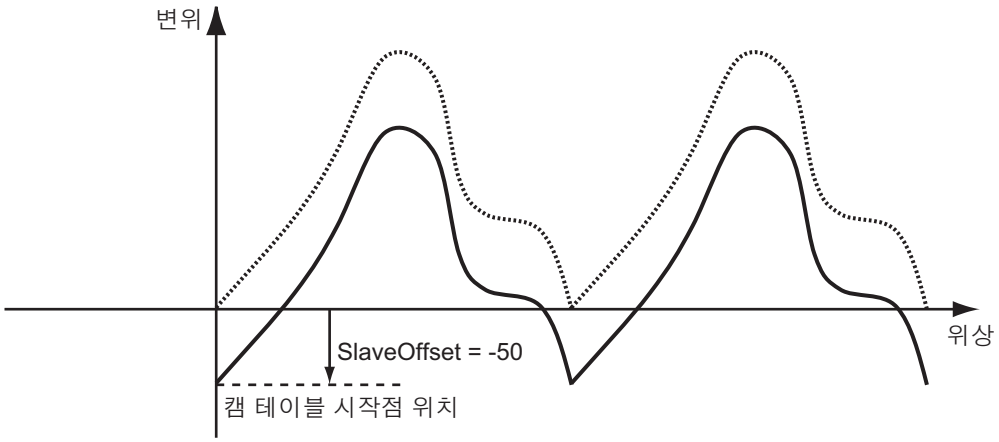
MasterOffset(주축 오프셋) < 0인 경우



SlaveOffset(종축 오프셋) > 0인 경우



SlaveOffset(종축 오프셋) < 0인 경우



● ReferenceType(위치 종류 선택)

종축이 동기하는 주축의 위치 종류를 다음에서 선택할 수 있습니다.

- _mcCommand: 명령 위치(직전 프라이머리 주기에서의 계산값)
  현재 주기에 대해 직전의 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  종축의 명령 위치를 계산하기 직전의 프라이머리 정주기 태스크에서 계산한 주축의 명령 위치를 사용합니다.
- _mcFeedback: 동일 프라이머리 주기에서의 취득값
  동일 프라이머리 주기에서 취득한 주축의 피드백 위치를 사용합니다.
- _mcLatestCommand: 명령 위치(동일 프라이머리 주기에서의 계산값)
  동일 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  _mcCommand보다도 새로운 정보를 사용할 수 있습니다. 단, 주축 번호는 종축 번호보다 빠른 번호를 설정해야 합니다.
  종축이 주축보다 축 번호가 빠르면 이 명령의 Error(에러)가 TRUE가 됩니다. ErrorID(에러 코드)에는 '주축/종축 축 번호 비오름차순(에러 코드: 5438 Hex)'이 출력됩니다.

참고

동일 프라이머리 주기에서 계산한 명령 위치를 선택하는 것이 직전 프라이머리 주기에서의 계산값을 선택하는 것보다 동기 정확도는 향상됩니다.
단, 'Master(주축)에 설정된 모션 제어 시스템 변수의 축 번호 < Slave(종축)에 설정된 모션 제어 시스템 변수의 축 번호'가 되도록 주축과 종축을 사용해야 합니다.

● 축 종류와 위치 종류의 관계

감시 가능한 축 종류와 감시하는 위치 종류의 관계는 다음과 같습니다.

축 종류	ReferecneType	
	_mcCommand 또는 _mcLatestCommand	_mcFeedback
서보 축	○	○
엔코더 축	× *1	○
가상 서보 축	○	○
가상 엔코더 축	× *1	○

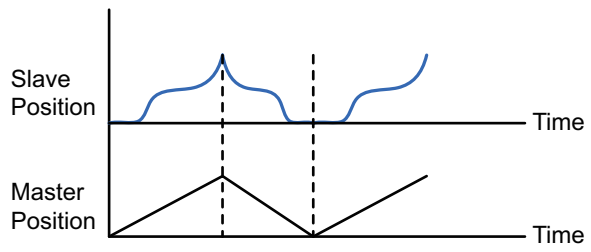
*1. 명령 기동 시에 '위치 종류 선택 범위 외(에러 코드: 5430 Hex)'가 발생합니다.

● Direction(방향 선택)

주축의 이동 방향이 Direction(방향 선택)과 일치하는 경우에만 종축의 캠 동작을 시작할 수 있습니다.
Direction(방향 선택)이 유효가 되는 것은 InSync(동기 중) 중이 됩니다.

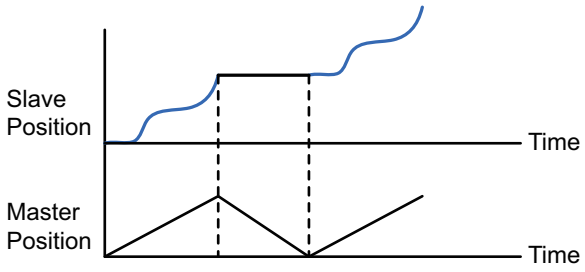
_mcNoDirection(방향 지정 안 함)

주축이 정방향, 역방향 중 어느 쪽으로 이동 중이라도 캠 동작을 시작합니다.



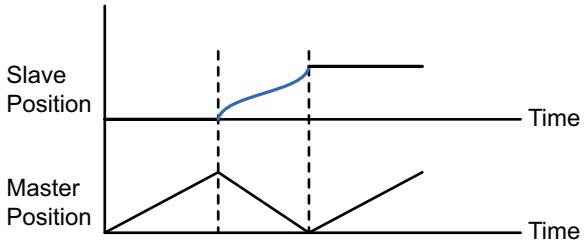
_mcPositiveDirection(정방향 지정)

주축이 정방향으로 이동 중에 캠 동작을 시작합니다.



_mcNegativeDirection(역방향 지정)

주축이 역방향으로 이동 중에 캠 동작을 시작합니다.



● BufferMode(버퍼 모드 선택)

하나 앞의 축 동작과 이번 동작의 연결 방법을 지정합니다.

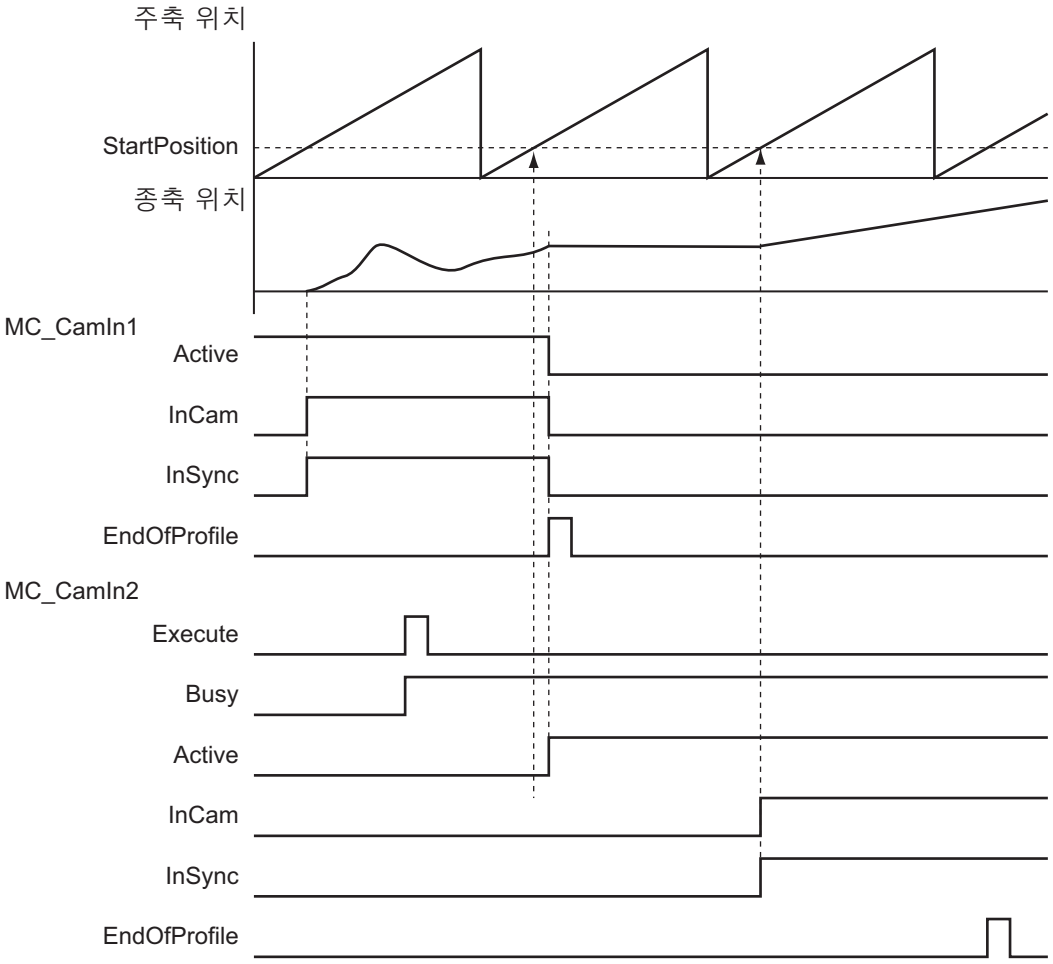
다음 6종류가 있습니다.

버퍼 모드 선택	설명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 주축이 StartPosition(캠 테이블 시작 위치)을 통과한 후 MasterStartDistance(주축 추종 거리)를 통과하면 종축의 캠 동작이 시작되고, 출력 변수 InSync(동기 중)가 TRUE가 됩니다. 버퍼링되고 있는 이 명령의 출력 변수 InSync(동기 중)가 TRUE가 될 때까지 종축은 정지하고 있습니다. 캠 테이블 시작점은 StartPosition(캠 테이블 시작 위치)이 됩니다.
Buffered	현재 실행 중인 명령이 정상 종료된 주기부터, 버퍼링되고 있는 이 명령이 자동으로 기동합니다. 주축이 StartPosition(캠 테이블 시작 위치)을 통과한 후 MasterStartDistance(주축 추종 거리)를 통과하면 종축의 캠 동작이 시작되고, 출력 변수 InSync(동기 중)가 TRUE가 됩니다. 버퍼링되고 있는 이 명령의 출력 변수 InSync(동기 중)가 TRUE가 될 때까지 종축은 정지하고 있습니다. 캠 테이블 시작점은 StartPosition(캠 테이블 시작 위치)이 됩니다.
Blending*1 Blending low Blending previous Blending next Blending high	현재 실행 중인 명령이 정상 종료된 주기부터, 버퍼링되고 있는 이 명령의 출력 변수 InSync(동기 중)가 TRUE가 되며, 종축이 정지하지 않고 캠 동작을 시작합니다. 버퍼링되고 있는 이 명령의 StartPosition(캠 테이블 시작 위치), MasterStartDistance(주축 추종 거리)를 지정한 경우도 지정값에 관계 없이 명령의 기동과 동시에 종축이 캠 동작을 시작합니다. 캠 테이블 시작점은 현재 실행 중인 명령의 종료 위치가 됩니다.

*1. CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용할 수 있습니다. 4종류의 Blending 중 어느 것을 지정해도 동일한 동작이 됩니다.

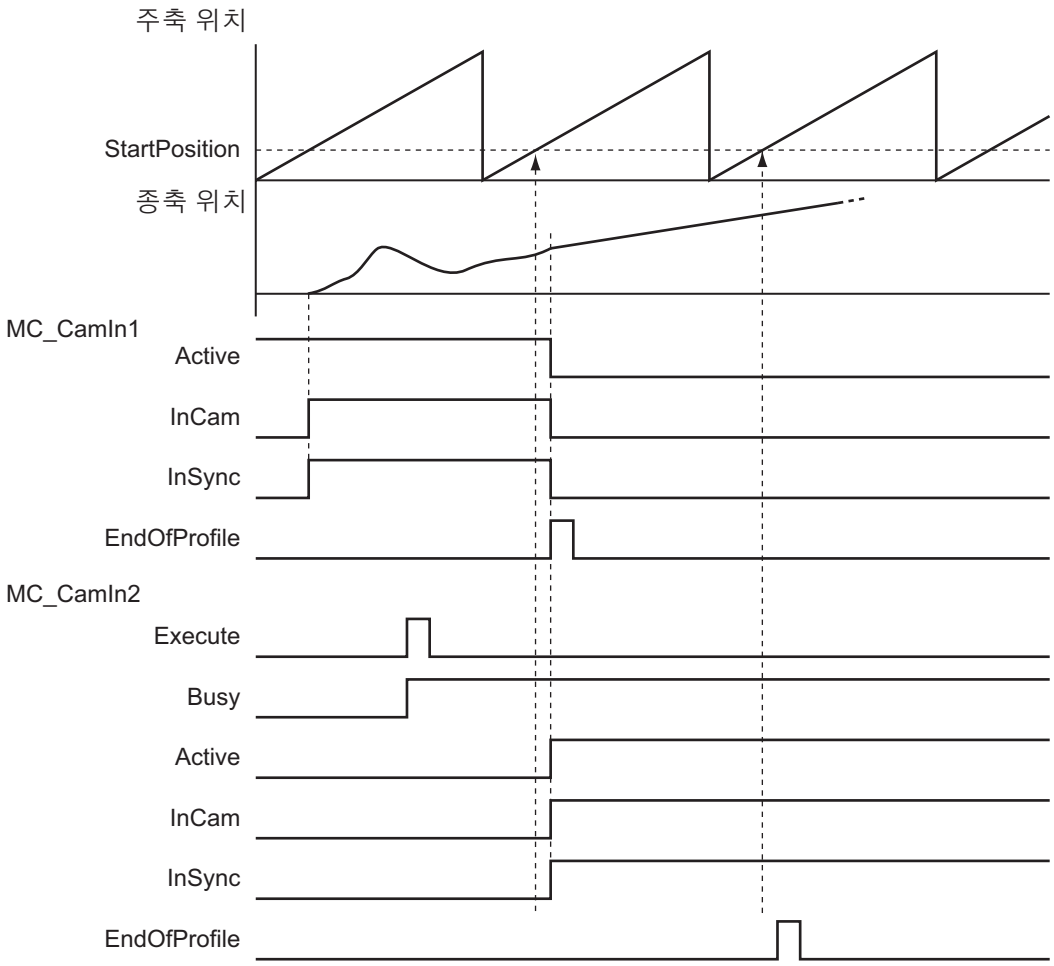
● Buffered를 지정하여 다중 기동하는 경우

MC_CamIn1 실행 중에 Buffered을 지정하여 MC_CamIn2를 다중 기동합니다. MC_CamIn1이 종료되고 MC_CamIn2의 Active(제어 중)가 TRUE가 된 다음의 StartPosition(캠 테이블 시작 위치)에서 InSync(동기 중)가 TRUE가 되고, 캠 동작이 시작됩니다.



● Blending을 지정하여 다중 기동하는 경우

MC_CamIn1 실행 중에 Blending을 지정하여 MC_CamIn2를 다중 기동합니다. MC_CamIn1이 종료된 동일한 주기에 MC_CamIn2의 InSync(동기 중)가 TRUE가 되고, 캠 동작이 시작됩니다.



BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 인덱스

주축, 종축의 명령 위치를 구하기 위해 사용한 캠 데이터 2개 중 캠 데이터 인덱스 번호가 작은 쪽을 출력 변수 'Index(인덱스)'로 출력합니다.

이 값은 캠 편집기 또는 사용자 프로그램으로 캠 데이터를 미세 조정하는 경우 등에 사용합니다.

● 인포지션 확인

이 명령에서는 인포지션 확인을 수행하지 않습니다.

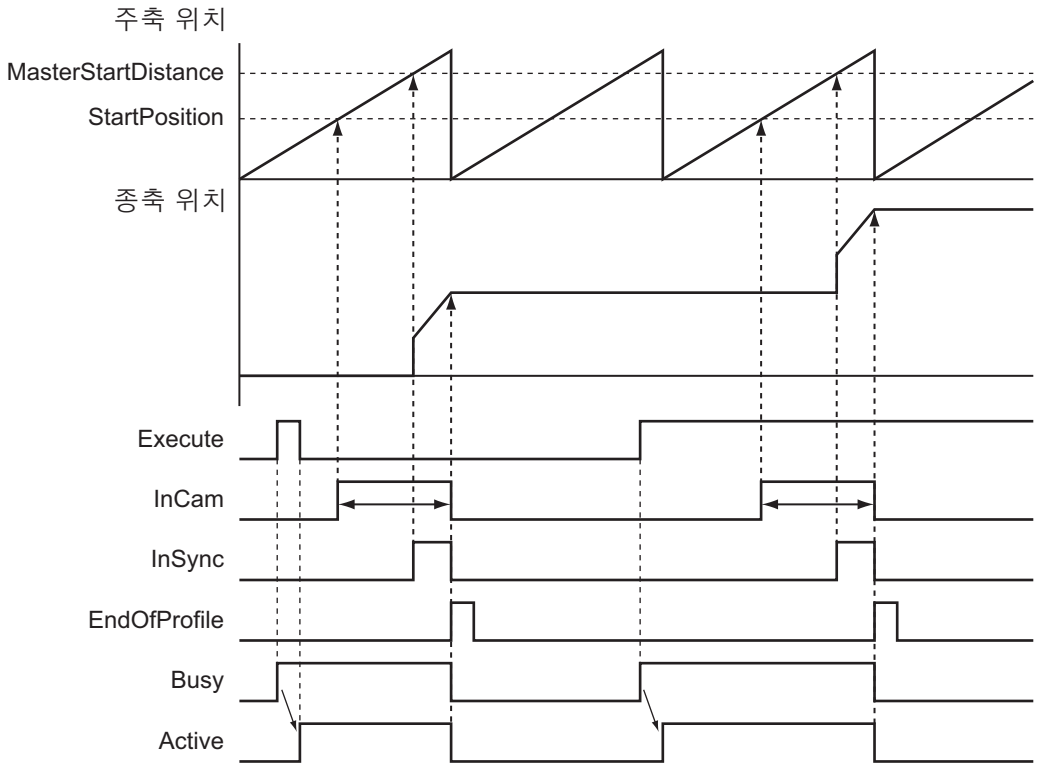
● 오버라이드

이 명령에서는 MC_SetOverride(오버라이드 값 설정) 명령을 통해 오버라이드할 수 없습니다.

타이밍 차트

● 반복 안 함의 경우

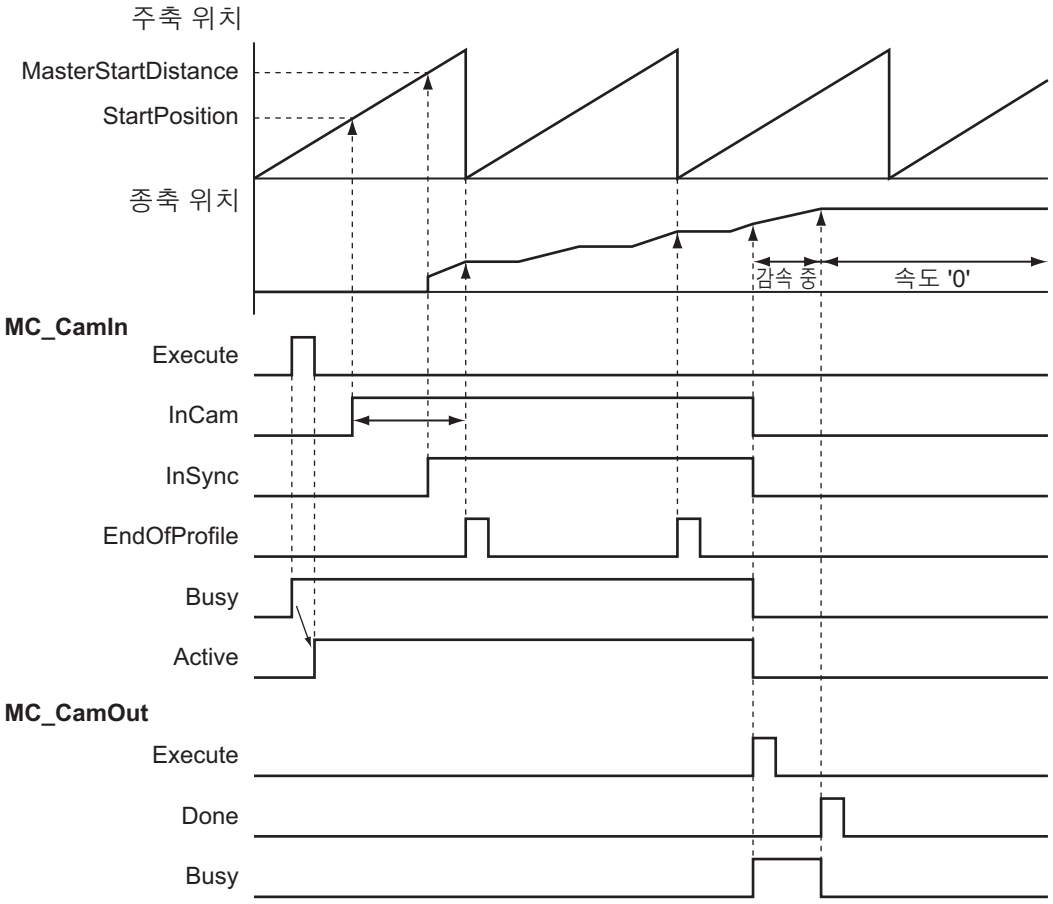
MC_CamIn(캠 동작 시작) 명령의 Periodic(반복 모드)이 FALSE[반복 안 함]인 경우 다음 그림과 같습니다.



● 반복함의 경우

캠 동작이 반복 실행됩니다.
MC_CamOut(캠 동작 해제) 명령을 이용해서 캠을 해제하면 종축은 속도 '0'까지 감속합니다.
주축의 동작에는 영향을 미치지 않습니다.

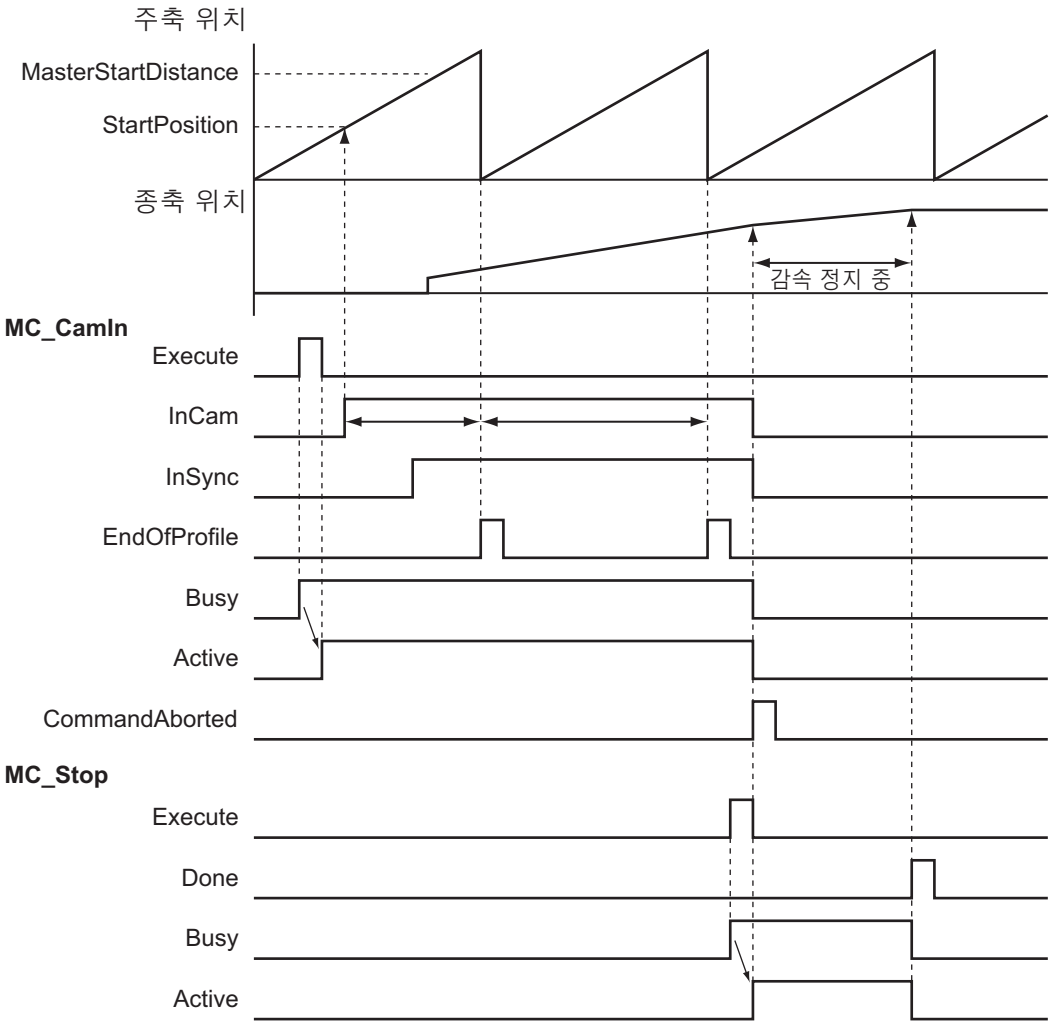
MC_CamIn(캠 동작 시작) 명령의 Periodic(반복 모드)이 TRUE[반복]일 때 MC_CamOut(캠 동작 해제) 명령을 실행하면 다음 그림과 같습니다.



● MC_Stop(강제 정지) 명령의 경우

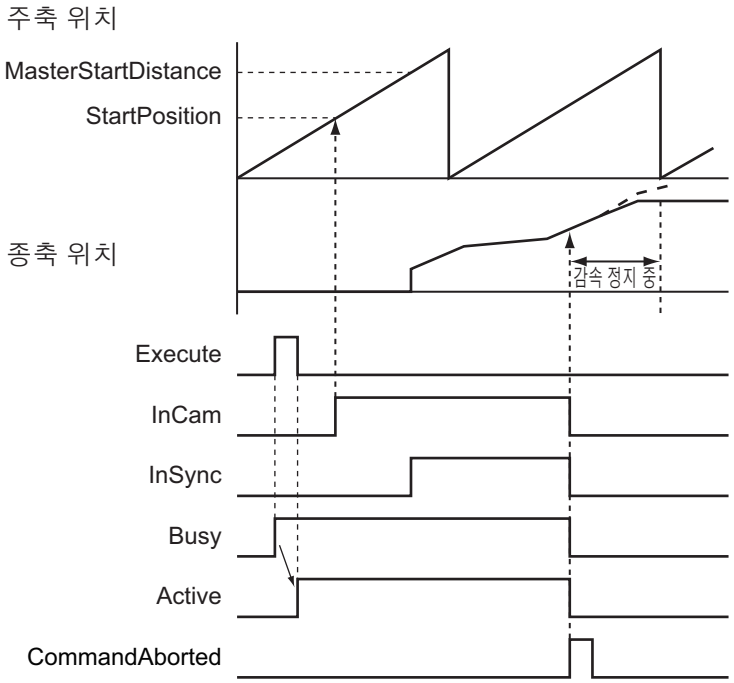
캠 동작 중에 주축에 대해 MC_Stop(강제 정지) 명령을 기동하면 주축과 종축의 동기 관계는 유지됩니다.
캠 동작 중에 종축에 대해 MC_Stop(강제 정지) 명령을 기동하면 주축과 종축의 동기 관계는 종료됩니다.

MC_CamIn(캠 동작 시작) 명령의 Periodic(반복 모드)이 TRUE[반복]일 때 종축에 대해 MC_Stop(강제 정지) 명령을 실행하면 다음 그림과 같습니다.



명령 중지

이 명령 실행 중에 종축에서 축 이상이 발생하면 종축은 축의 최대 감속도로 정지합니다. 종축에 대해 MC_Stop(강제 정지) 명령을 기동시킨 경우, CommandAborted(실행 중단)가 TRUE가 되고 종축은 MC_Stop(강제 정지) 명령으로 지정한 감속도로 정지합니다.
축 이상 발생 요인의 확인 방법에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.
또한 주축에서 축 이상이 발생해도 캠 동작은 계속됩니다.



모션 명령 재기동

동작 중인 캠 테이블을 전환하면 이 명령을 재기동할 수 있습니다.
캠 테이블 전환 이외의 조건을 변경하는 경우에는 이 명령에 대한 명령 다중 기동을 실행하십시오.
명령 재기동 시 InCam(캠 동작 중), InSync(동기 중)는 재기동 전 상태를 유지합니다. InSync(동기 중) 중에 재기동한 동작은 현재 위치에서 구해진 위상보다 큰 위상으로부터 캠 동작을 실행합니다.
현재 위치에서 구해진 위상이란 캠 데이터 간을 직선으로 보간했을 때 구해지는 위상입니다.

또한 재기동 대상은 캠 테이블뿐입니다.

사용상의 주의 사항

캠 동작 중에 명령을 재기동해 캠 테이블을 전환하면 재기동 전후에 종축의 속도나 가속도가 급격하게 변화하는 경우가 있습니다.
기계 기구에 영향을 미칠 수 있으므로 재기동할 경우에는 주의하십시오.

모션 명령 재기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 재기동 방법

재기동 방법에 대해서는 '샘플 프로그램 1(3-183페이지)' 또는 '샘플 프로그램 2(3-194페이지)'를 참조하십시오.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하려면 종축을 지정하십시오.
또한 이 명령 동작 중에 다른 명령을 다중 기동하면 버퍼 모드에 의해 다음의 제약이 있습니다.

- 다른 명령을 Aborting으로 다중 기동하면 그 시점에서 캠 동작을 중단하고 다음 동작을 시작합니다.
- 다른 명령을 Buffered로 다중 기동하면 EndOfProfile(캠 사이클 완료)이 TRUE가 된 타이밍에 다음 동작의 명령값을 출력합니다.

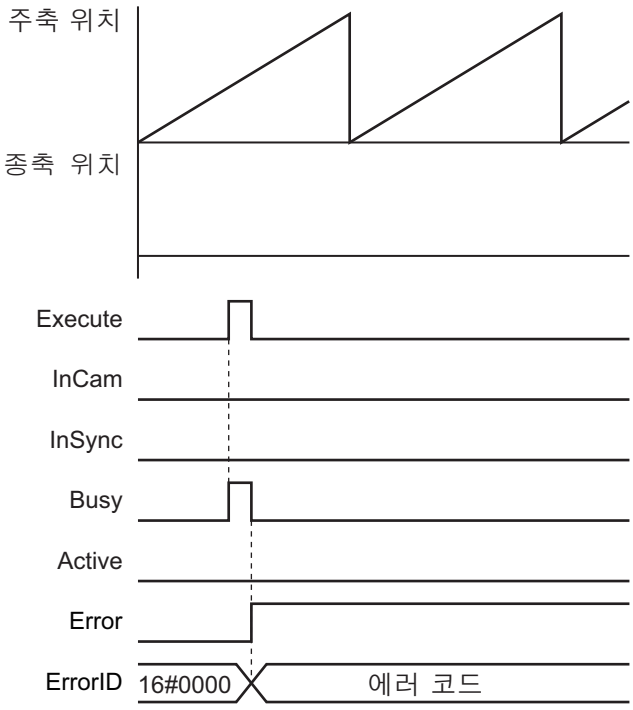
● 이 명령끼리의 Blending에서의 다중 기동

- 이 명령끼리 다중 기동하는 경우에만 Blending을 지정할 수 있습니다. 다른 명령을 실행하는 중에 이 명령을 Blending으로 다중 기동할 수 없습니다.
- Master(주축)와 ReferenceType(위치 종류 선택)은 현재 실행 중인 명령과 각각 같은 값을 지정하십시오. 다른 값을 지정하면 '모션 명령 다중 기동 불가능' 이상이 발생합니다.
- 현재 실행 중인 명령이 정상 종료된 주기부터 버퍼링되고 있는 이 명령의 출력 변수 InSync(동기 중)가 TRUE가 되며, 종축이 정지하지 않고 캠 동작을 시작합니다.
- StartPosition(캠 테이블 시작 위치), MasterStartDistance(주축 추종 거리)를 지정한 경우도 지정값에 관계 없이 명령의 기동과 동시에 종축이 캠 동작을 시작합니다. 캠 테이블 시작점은 현재 실행 중인 명령의 종료 위치가 됩니다. 4종류의 Blending 중 어느 것을 지정해도 동일한 동작이 됩니다.

이상

이 명령 기동 시에 이상을 감지하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램 1

여기에는 반복함의 캠 동작에 대한 샘플 프로그램이 실려 있습니다.
이 예에서는 주축을 축 1, 종축을 축 2로 합니다.



참고

예약(Reserved)으로 취급되는 입력 변수에는 초기값 이외의 값을 지정할 수 없습니다.
이 예에서는 각 명령에서 정의된 초기값을 갖는 변수를 할당하고 있는데, 프로그래밍 시에는 변수나 파라메터를 할당하지 않아도 됩니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축(주축)
축 2	서보 축(종축)

카운터 모드

축	카운터 모드
축 1	로터리 모드
축 2	리니어 모드

링 카운터

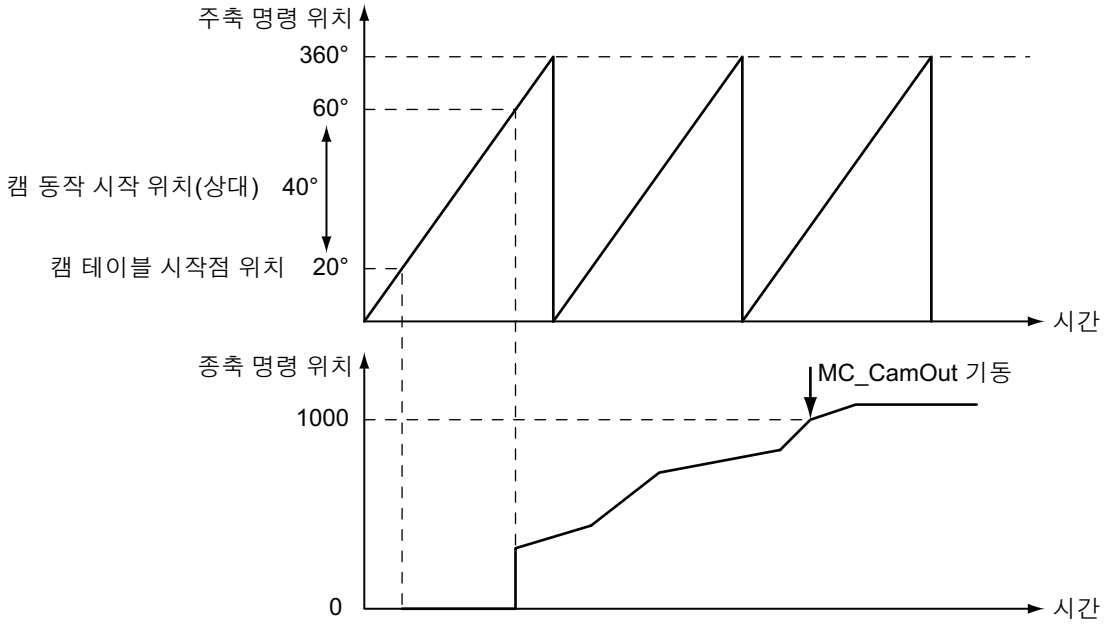
축	상한값	하한값
축 1	360	0

좌표 단위 선택

축	좌표 단위
축 1	degree
축 2	mm

동작 예

● 동작 패턴



1 **캠 동작의 시작**

캠 테이블 시작점 위치(위상 제로의 위치)를 20°로 하고, 이곳에서 상대 각도 40° 떨어진 각도에 주축이 오면 종축은 캠 동작을 시작합니다. 캠 동작은 반복 동작으로 움직입니다.
Periodic(반복 모드)을 TRUE로 하면 반복 동작을 합니다.

2 **캠 동작의 해제**

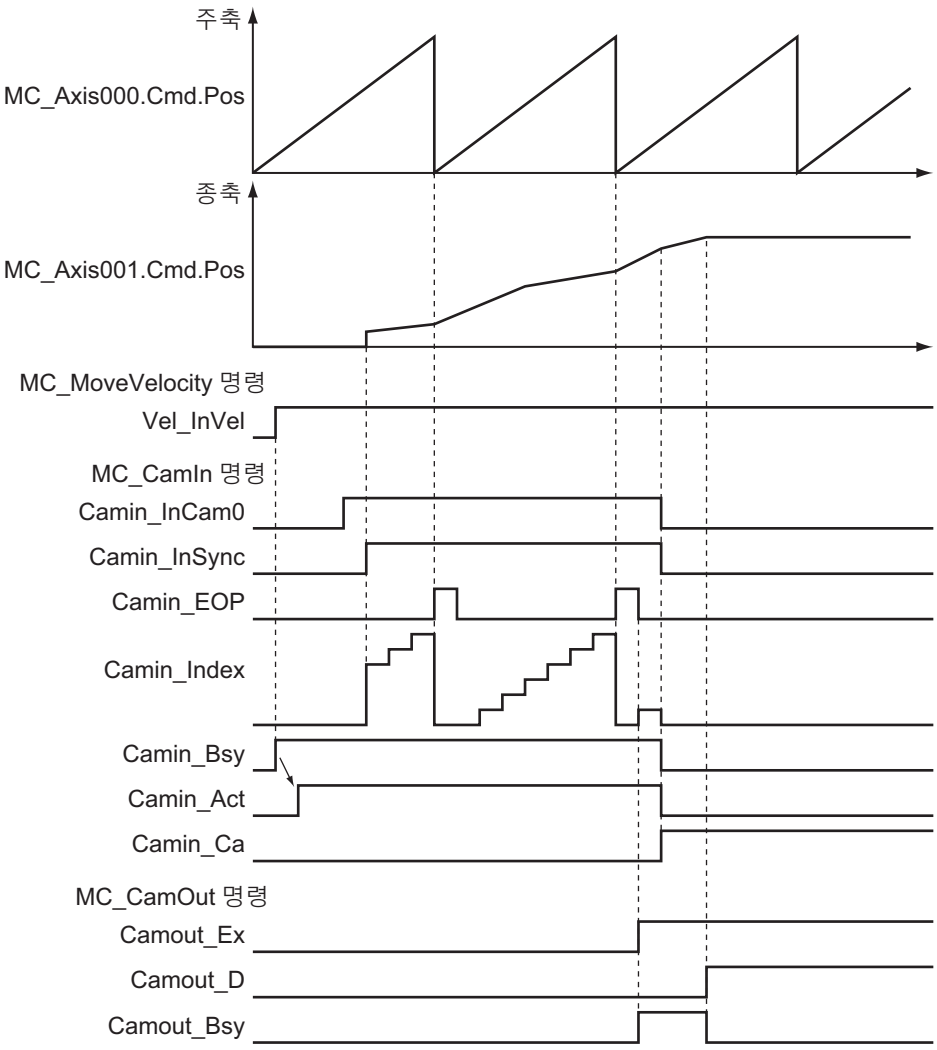
종축의 피드백 위치(MC_Axis001.Act.Pos)가 1000.0 이상이 되면 캠 동작을 해제하고 감속도(DecRate2)로 종축을 정지시킵니다.

래더도

● 주요 변수

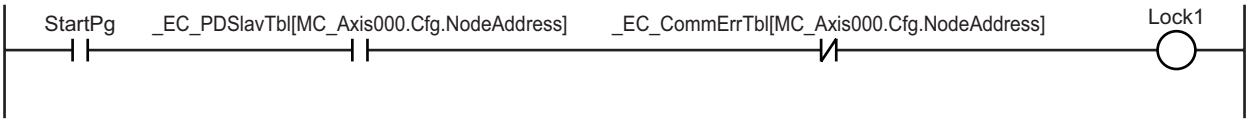
이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
Pwr1_S	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr2_S	BOOL	FALSE	MC_Power 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	–	캠 데이터 변수입니다. 배열 요소 수 ARRAY[0..N]은 캠 편집기로 설정합니다. 이 예에서는 0~360으로 되어 있지만 캠 편집기의 설정에 따라 배열 요소 수는 달라집니다.
DecRate2	LREAL	10000.0	MC_CamOut 실행 시의 감속도를 설정합니다.
Vel_InVel	BOOL	FALSE	MC_MoveVelocity의 인스턴스인 VEL의 출력 InVelocity에 할당한 변수입니다. 목표 속도에 도달하면 이 변수가 TRUE가 됩니다.
Camin_InCam0	BOOL	FALSE	MC_CamIn 인스턴스 CAMIN의 출력 InCam에 할당한 변수입니다. 캠 동작 중일 때 이 변수가 TRUE가 됩니다.
Camout_Ex	BOOL	FALSE	이 변수가 TRUE일 때 MC_CamOut의 인스턴스 CAMOUT이 기동됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.

● 타이밍 차트

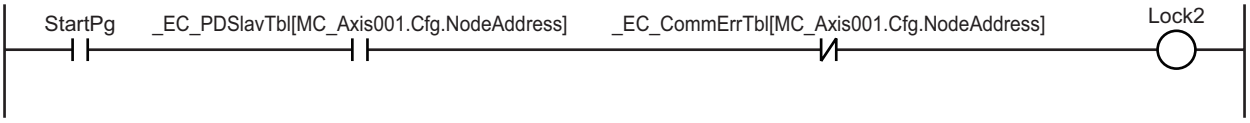


● 샘플 프로그램

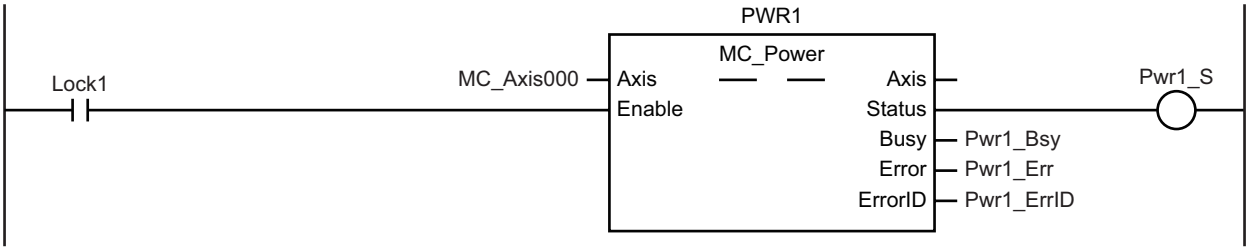
접점 StartPg가 TRUE가 되면 축 1의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인



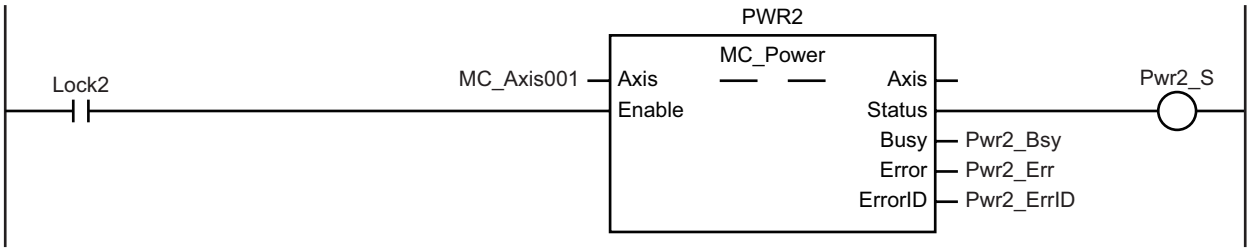
접점 StartPg가 TRUE가 되면 축 2의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인



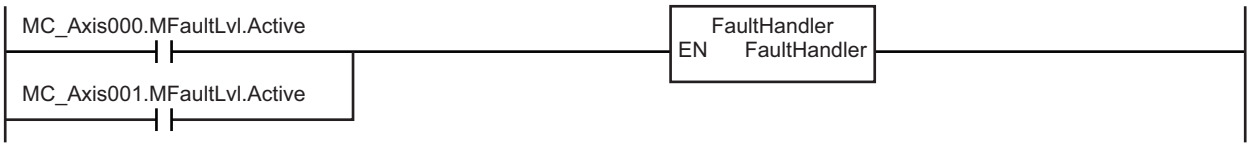
축 1이 프로세스 데이터 통신을 하고 있으면 서보를 ON



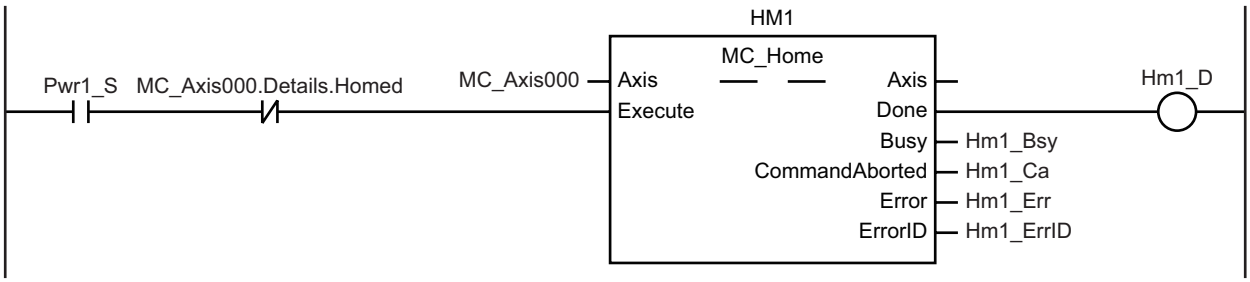
축 2이 프로세스 데이터 통신을 하고 있으면 서보를 ON



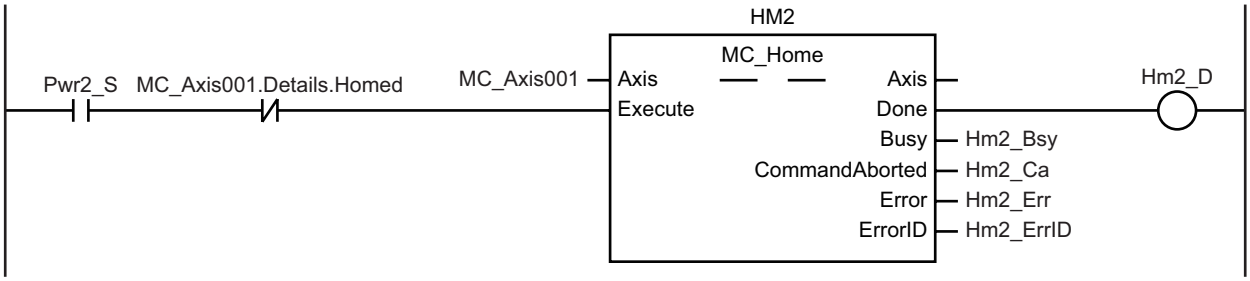
축 구성에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



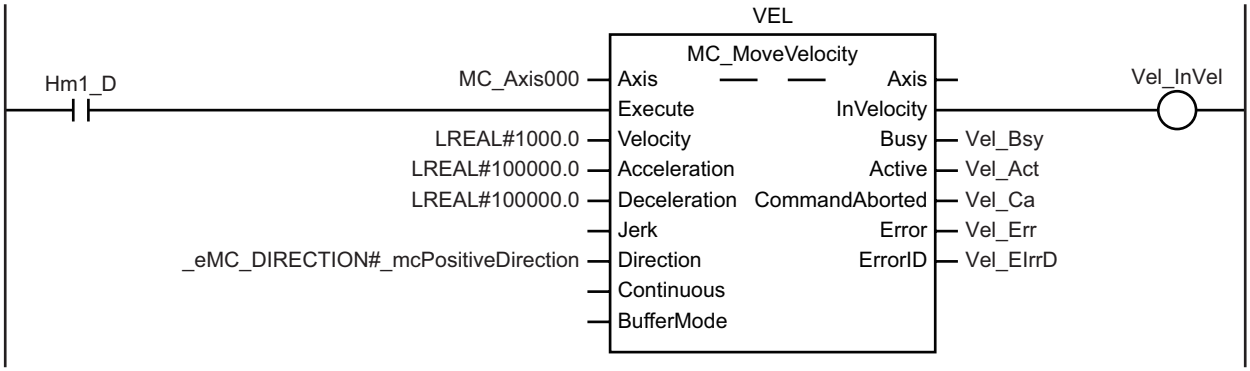
축 1이 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행하고 원점을 확정



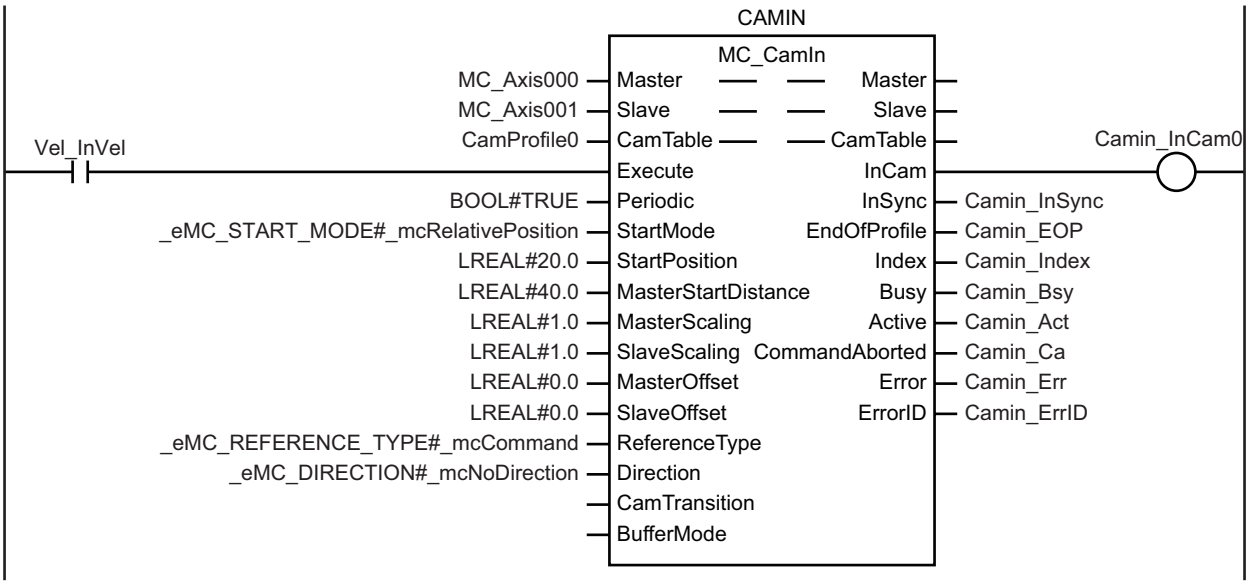
축 2가 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행하고 원점을 확정



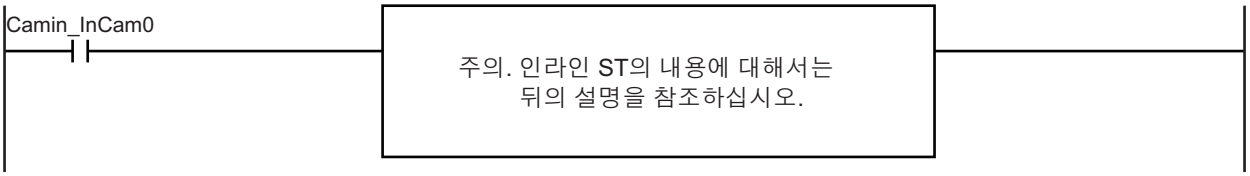
축 1이 원점 복귀 완료 상태가 되면 MC_MoveVelocity(속도 제어) 명령을 실행



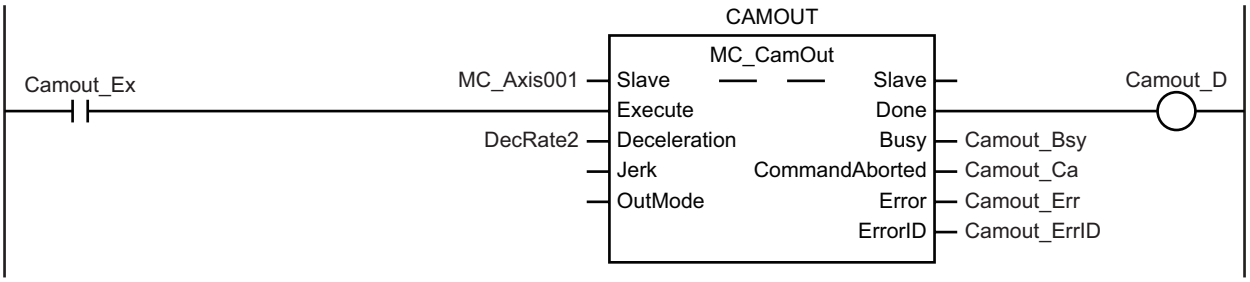
MC_MoveVelocity(속도 제어) 명령의 Vel_InVel이 TRUE가 되면 MC_CamIn(캠 동작 시작) 명령을 기동



Camin_InCam0이 TRUE이고 MC_Axis001.Act.Pos가 1000을 넘으면 CamOut_Ex를 TRUE로 함



Camout_Ex를 TRUE로 해서 MC_CamOut(캠 동작 해제) 명령을 기동
감속도로 DecRate2를 지정



인라인 ST의 내용

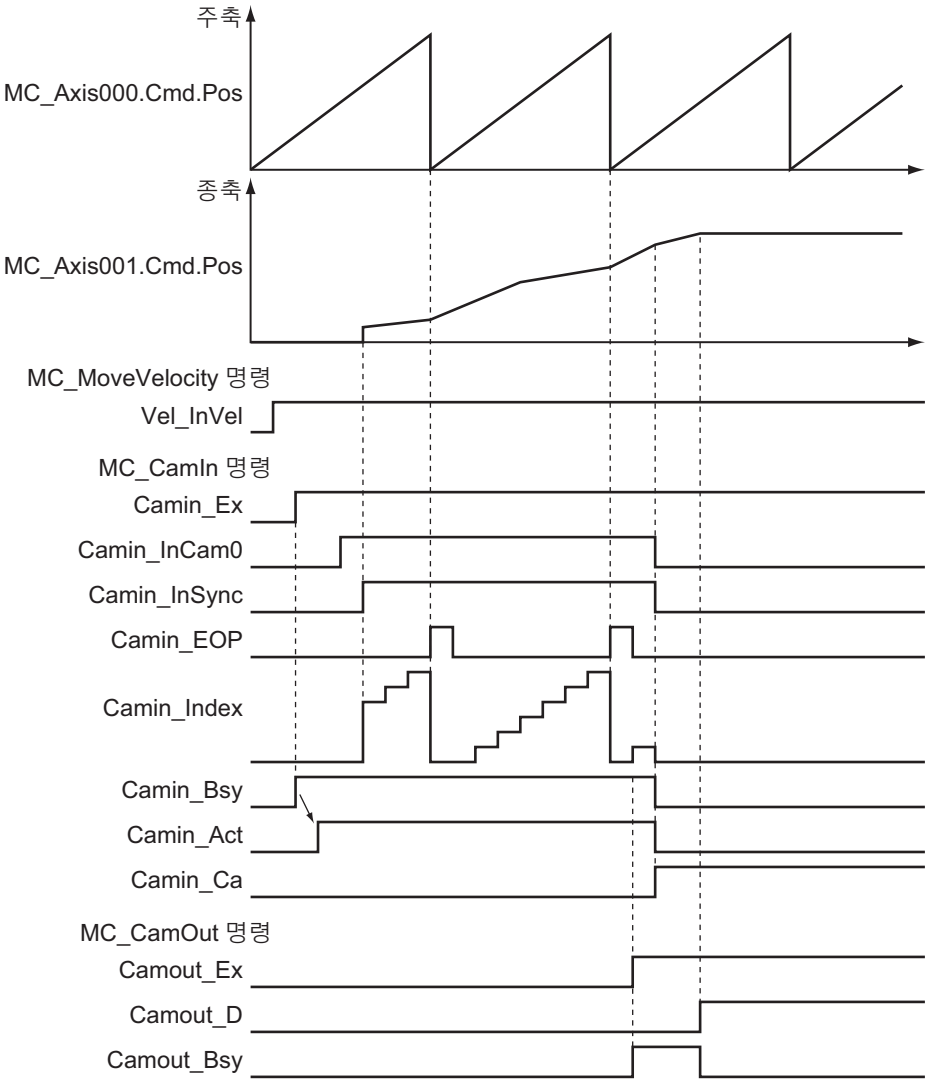
```
IF  MC_Axis001.Act.Pos>LREAL#1000.0 THEN
    Camout_Ex := TRUE;
END_IF;
```

스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
Pwr1_S	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr2_S	BOOL	FALSE	MC_Power 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	–	캠 데이터 변수입니다. 배열 요소 수 ARRAY[0..N]은 캠 편집기로 설정합니다. 이 예에서는 0~360으로 되어 있지만 캠 편집기의 설정에 따라 배열 요소 수는 달라집니다.
DecRate2	LREAL	10000.0	MC_CamOut 실행 시의 감속도를 설정합니다.
Vel_InVel	BOOL	FALSE	MC_MoveVelocity의 인스턴스인 VEL의 출력 InVelocity에 할당한 변수입니다. 목표 속도에 도달하면 이 변수가 TRUE가 됩니다.
Camin_InCam0	BOOL	FALSE	MC_CamIn 인스턴스 CAMIN의 출력 InCam에 할당한 변수입니다. 캠 동작 중일 때 이 변수가 TRUE가 됩니다.
Camout_Ex	BOOL	FALSE	이 변수가 TRUE일 때 MC_CamOut의 인스턴스 CAMOUT이 기동됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Camin_Ex	BOOL	FALSE	Vel_InVel 기동 시에 TRUE로 하면 MC_CamIn 인스턴스 CAMIN의 Execute를 TRUE로 합니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트



● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag=FALSE THEN

    //MC_CamIn 파라메터
    Camin_EM      := TRUE;                  // 반복 모드
    Camin_StMode := _eMC_START_MODE#_mcRelativePosition;
    Camin_StPos   := LREAL#20.0;            // 시작점의 주축 절대 위치
    Camin_MStDis  := LREAL#40.0;            // 캠 동작을 시작하는 주축 위치
    Camin_MSc     := LREAL#1.0;             // 주축 확대
    Camin_SSc     := LREAL#1.0;             // 종축 확대
    Camin_MO      := LREAL#0.0;             // 주축 오프셋
    Camin_SO      := LREAL#0.0;             // 종축 오프셋
    Camin_RT      := _eMC_REFERENCE_TYPE#_mcCommand;     //위치 종류 선택
    Camin_Dir     := _eMC_DIRECTION#_mcNoDirection;      //방향 선택

    //MC_MoveVelocity 파라메터
    Vel_Vel    := LREAL#1000.0;
    Vel_Acc    := LREAL#100000.0;
    Vel_Dec    := LREAL#100000.0;
    Vel_Dir    := _eMC_DIRECTION#_mcPositiveDirection;

    //MC_CamOut 파라메터
    Camout_Dec := DecRate2;
```

```
    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;


//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 1을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 2를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
END_IF;

//축 1~축 2에서 가벼운 이상이 발생하면 이상 시 처리를 실행합니다.
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

//축 1이 서보 ON 상태이고 원점 미확정 상태일 때 축 1의 원점 복귀를 실행합니다.
IF (Pwr1_S=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//축 2가 서보 ON 상태이고 원점 미확정 상태일 때 축 2의 원점 복귀를 실행합니다.
IF (Pwr2_S=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//축 1이 원점 복귀 완료 상태에서 MC_MoveVelocity를 실행합니다.
IF  Hm1_D=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

// MC_MoveVelocity가 InVel=TRUE일 때 CamIn을 기동합니다.
IF  Vel_InVel=TRUE THEN
    Camin_Ex := TRUE;
END_IF;

// Camin_InCam0=TRUE이고 MC_Axis001.Act.Pos>1000일 때 CamOut을 기동합니다.
IF (Camin_InCam0=TRUE) AND (MC_Axis001.Act.Pos>LREAL#1000.0) THEN
    Camout_Ex := TRUE;
END_IF;
```

```
//축 1의 MC_Power
PWR1(
    Axis        := MC_Axis000,
    Enable      := Pwr1_En,
    Status      => Pwr1_S,
    Busy        => Pwr1_Bsy,
    Error       => Pwr1_Err,
    ErrorID     => Pwr1_ErrID
);


//축 2의 MC_Power
PWR2(
    Axis        := MC_Axis001,
    Enable      := Pwr2_En,
    Status      => Pwr2_S,
    Busy        => Pwr2_Bsy,
    Error       => Pwr2_Err,
    ErrorID     => Pwr2_ErrID
);


//축 1의 MC_Home
HM1(
    Axis                := MC_Axis000,
    Execute             := Hm1_Ex,
    Done                => Hm_D,
    Busy                => Hm1_Bsy,
    CommandAborted      => Hm1_Ca,
    Error               => Hm1_Err,
    ErrorID             => Hm1_ErrID
);


//축 2의 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_CamIn
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,
    CamTransition        := Camin_CT,
    BufferMode           := Camin_BM,
    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
);
//MC_CamOut
CAMOUT(
    Slave                := MC_Axis001,
    Execute              := Camout_Ex,
    Deceleration         := Camout_Dec,
    Done                 => Camout_D,
    Busy                 => Camout_Bsy,
    CommandAborted       => Camout_Ca,
    Error                => Camout_Err,
    ErrorID              => Camout_ErrID
);
```

샘플 프로그램 2

여기에는 액체 충전 시의 캠 동작에 대한 샘플 프로그램이 나와 있습니다.

참고

예약(Reserved)으로 취급되는 입력 변수에는 초기값 이외의 값을 지정할 수 없습니다.
이 예에서는 파라메터를 지정하지 않습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축(주축)
축 2	서보 축(종축)
축 3	서보 축(종축)
축 4	서보 축(종축)
축 5	서보 축(종축)

카운터 모드

축	카운터 모드
축 1	로터리 모드
축 2	로터리 모드
축 3	로터리 모드
축 4	로터리 모드
축 5	로터리 모드

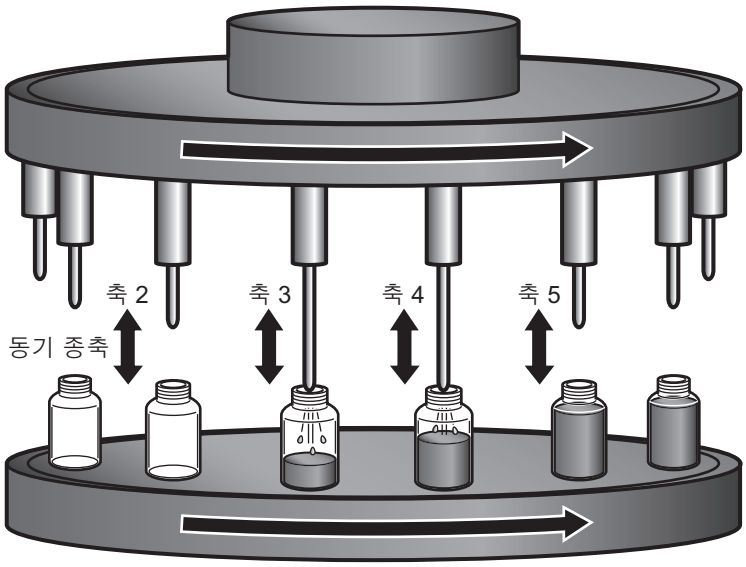
링 카운터

축	상한값	하한값
축 1	360	0
축 2	360	0
축 3	360	0
축 4	360	0
축 5	360	0

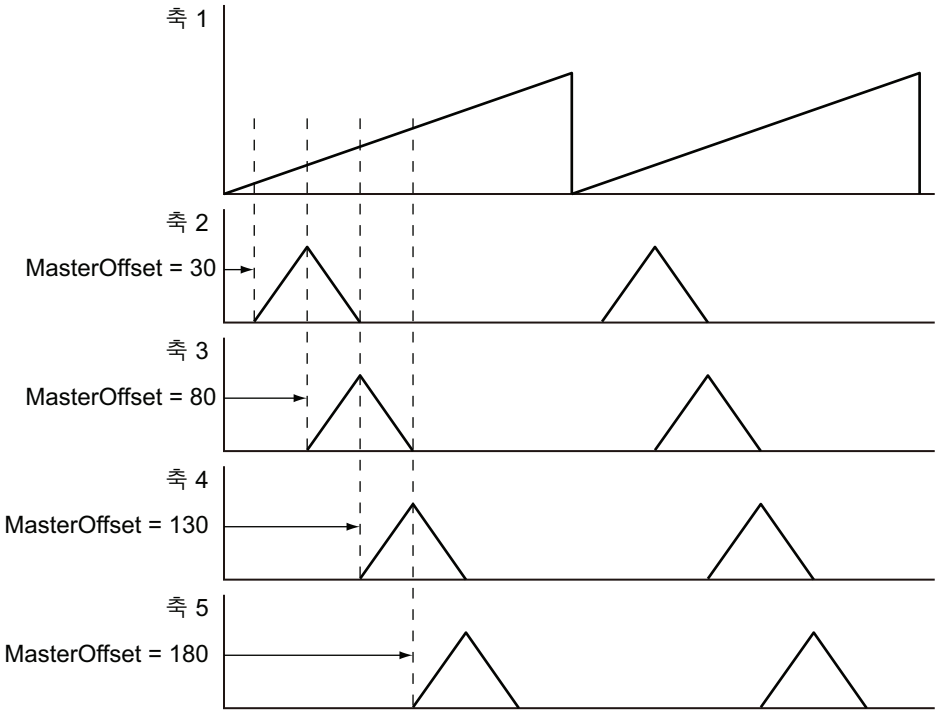
좌표 단위 선택

축	좌표 단위
축 1	degree
축 2	degree
축 3	degree
축 4	degree
축 5	degree

동작 예


축 2
축 3
축 4
축 5
동기 종축
동기 주축
축 1

● 동작 패턴


축 1
축 2
MasterOffset = 30
축 3
MasterOffset = 80
축 4
MasterOffset = 130
축 5
MasterOffset = 180

1 **캠 동작 시작**

주축(축 1)에 동기해서 종축(축 2~축 5)이 캠 동작을 실행합니다. 각 축은 50degree씩 위상을 이동해 캠 동작을 시작합니다.

2 **반복**

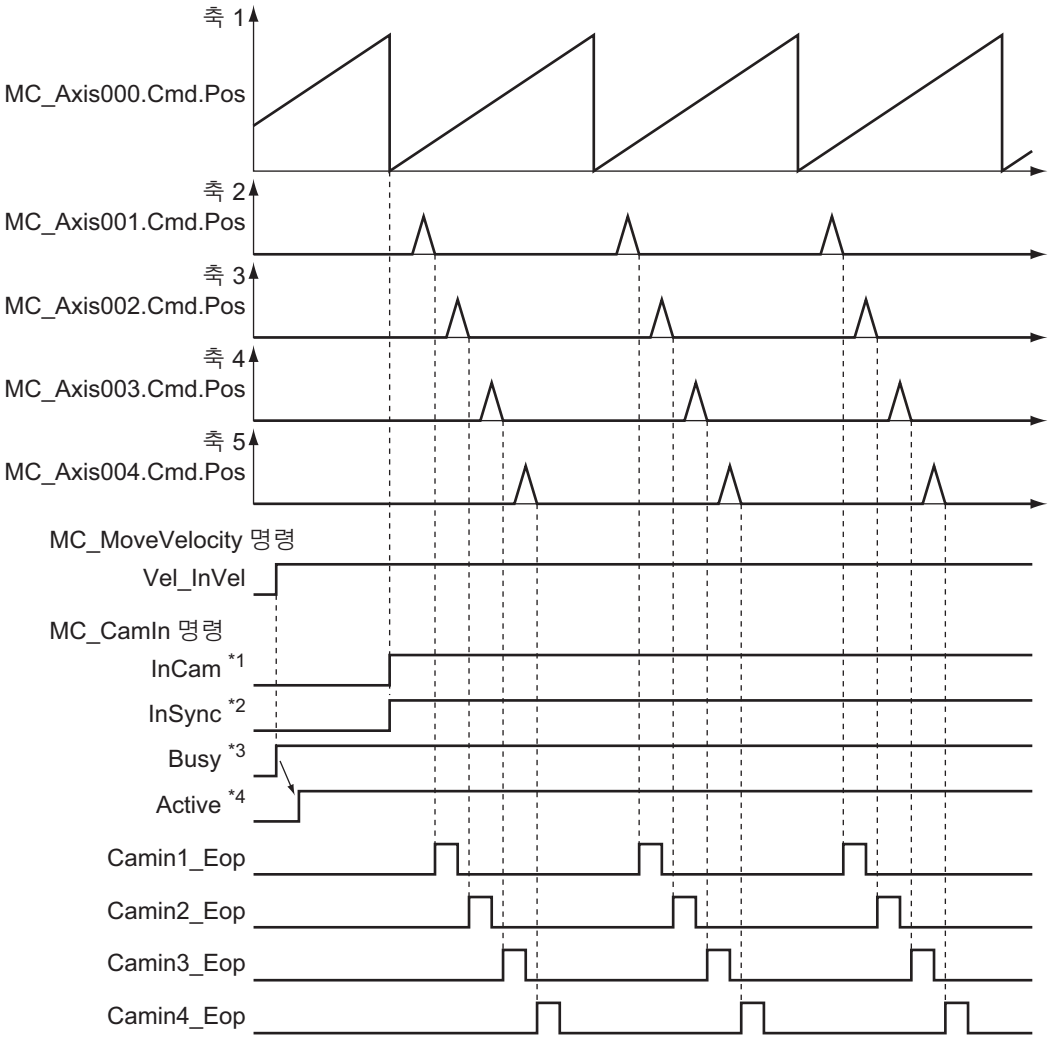
각 축은 지정된 캠 동작을 반복 실행합니다.

래더도

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis002	_sAXIS_REF	–	종축인 축 3의 축 변수입니다.
MC_Axis003	_sAXIS_REF	–	종축인 축 4의 축 변수입니다.
MC_Axis004	_sAXIS_REF	–	종축인 축 5의 축 변수입니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Vel_InVel	BOOL	FALSE	MC_MoveVelocity의 인스턴스인 VEL의 출력 InVelocity에 할당한 변수입니다. 목표 속도에 도달하면 이 변수가 TRUE가 됩니다.
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	–	캠 데이터 변수입니다. MC_CamIn 인스턴스 CAMIN1~CAMIN4의 입력 CamTable에 이 변수를 지정합니다. 배열 요소 수 ARRAY[0..N]은 캠 편집기로 설정합니다.

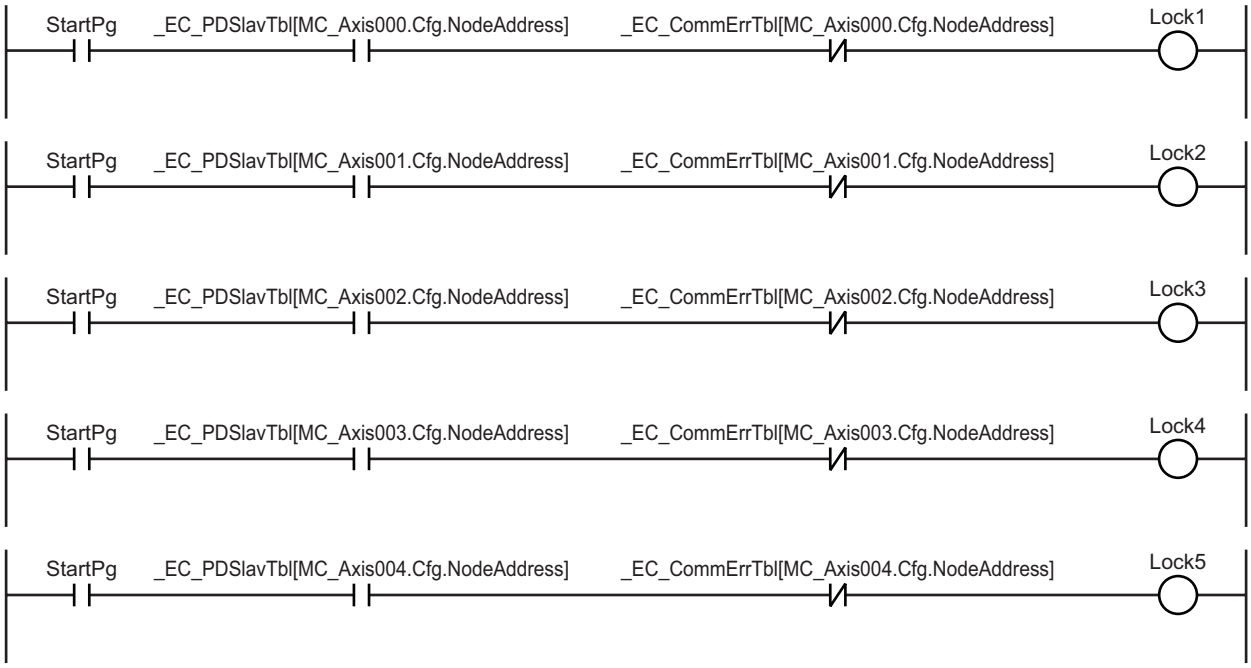
● 타이밍 차트



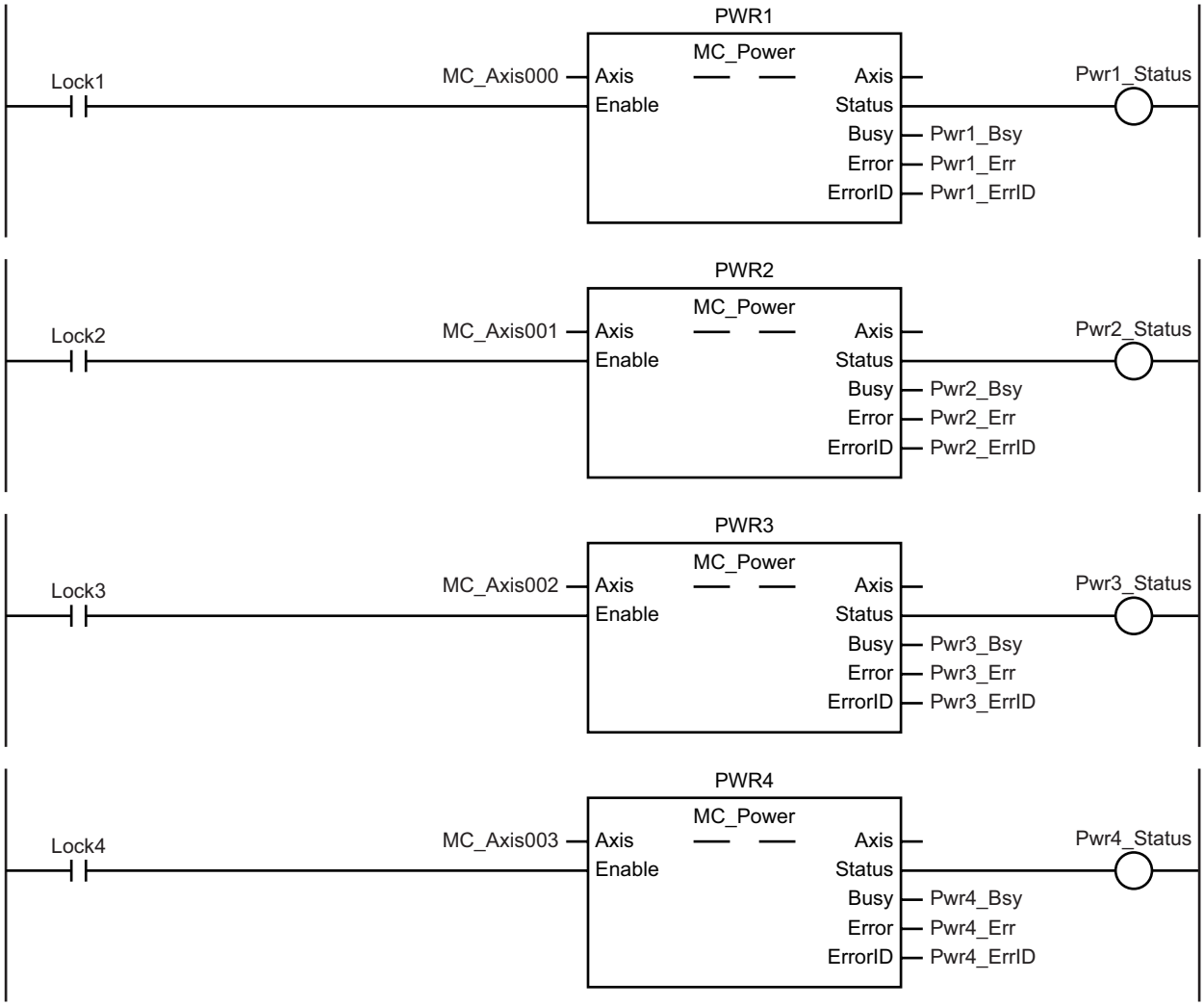
*1 Camin1_InCam, Camin2_InCam, Camin3_InCam, Camin4_InCam이 동일한 타이밍입니다.
*2 Camin1_InSync, Camin2_InSync, Camin3_InSync, Camin4_InSync가 동일한 타이밍입니다.
이 예에서는 MasterStartDistance가 '0'이므로 주축이 '0°'에서부터 InSync가 TRUE가 됩니다.
*3 Camin1_Bsy, Camin2_Bsy, Camin3_Bsy, Camin4_Bsy가 동일한 타이밍입니다.
*4 Camin1_Act, Camin2_Act, Camin3_Act, Camin4_Act가 동일한 타이밍입니다.

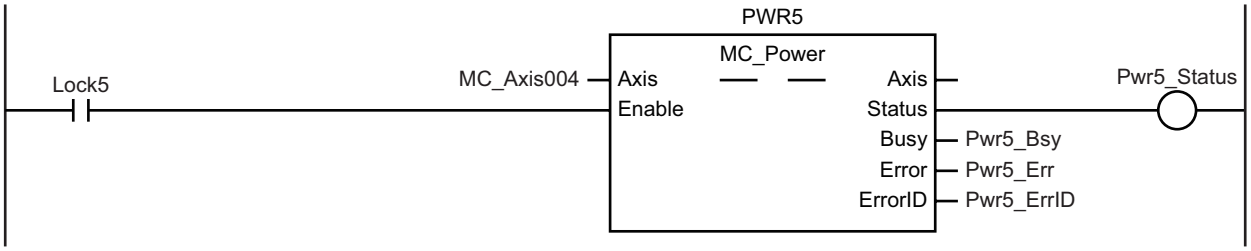
● 샘플 프로그램

접점 StartPg가 TRUE가 되면 각 축의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인

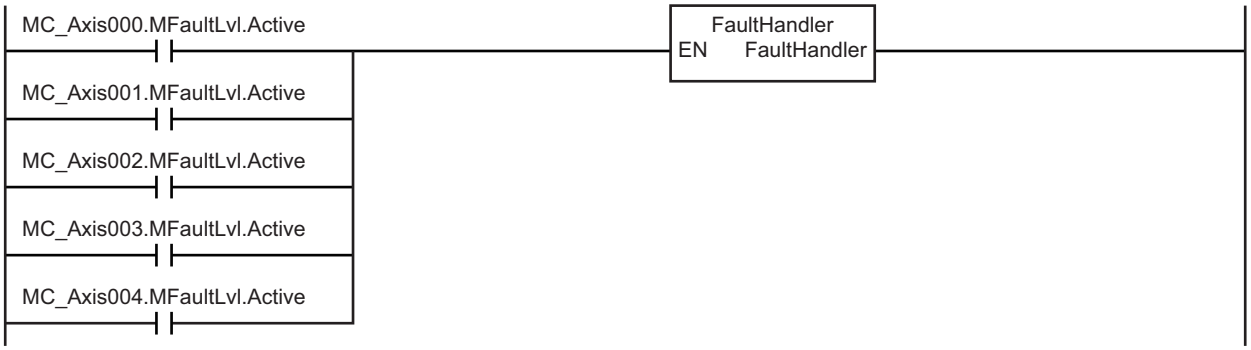


프로세스 데이터 통신을 하고 있으면 각 축의 서보를 ON

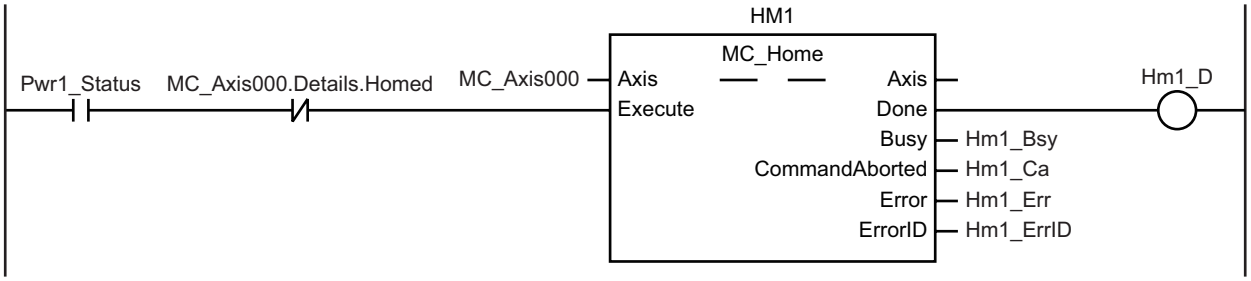


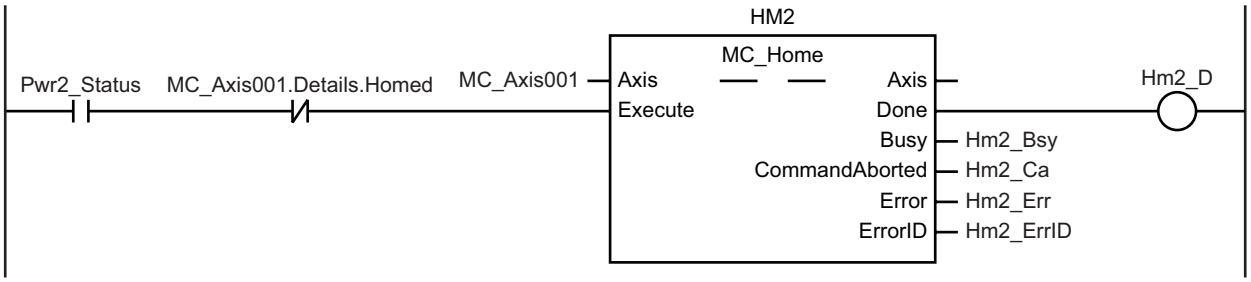


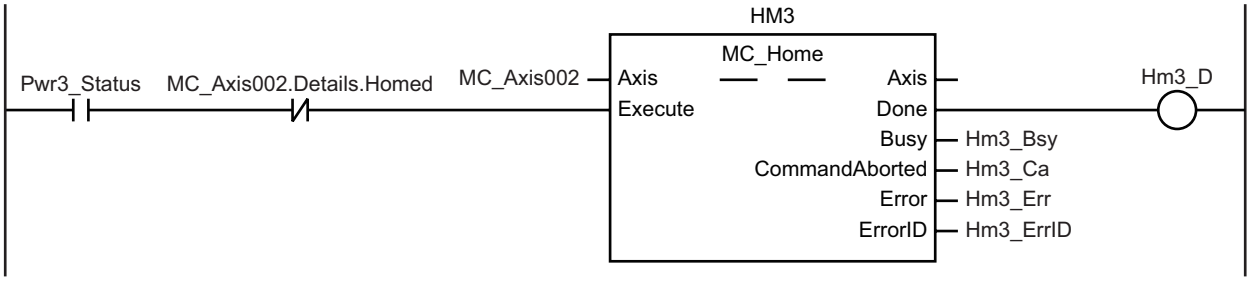
축 구성에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.

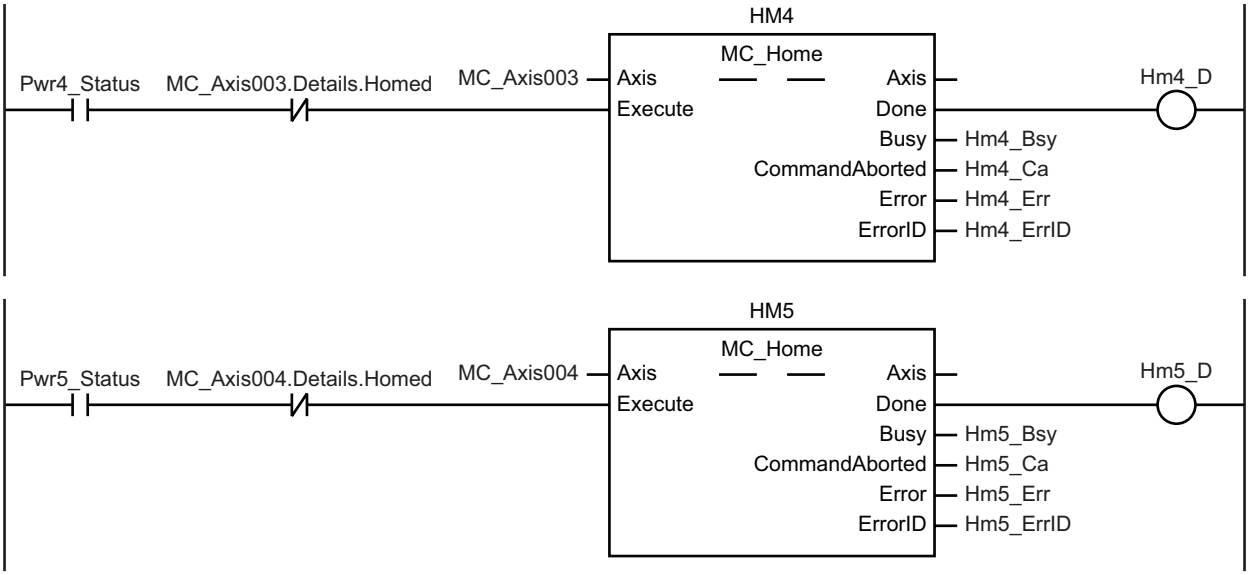


축 1~축 5가 서보가 ON 되고 원점 미확정 상태라면 각 축마다 원점 복귀를 실행

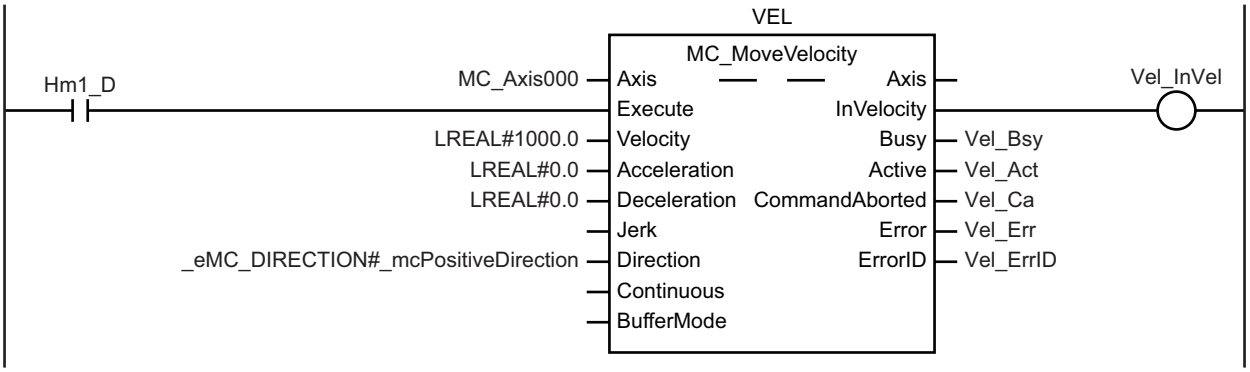




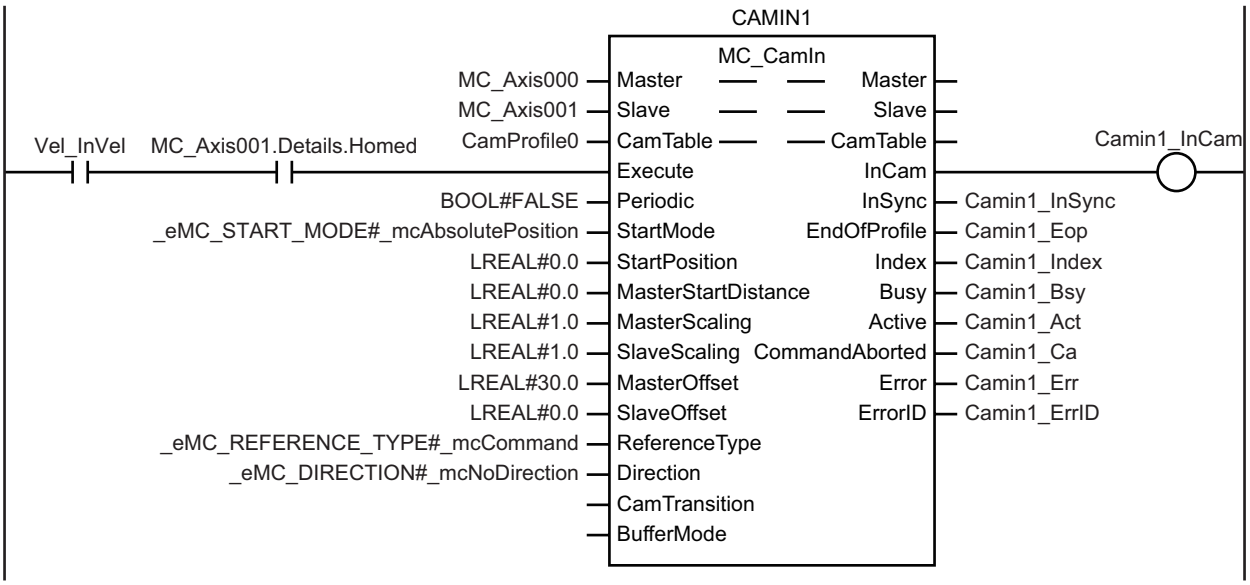




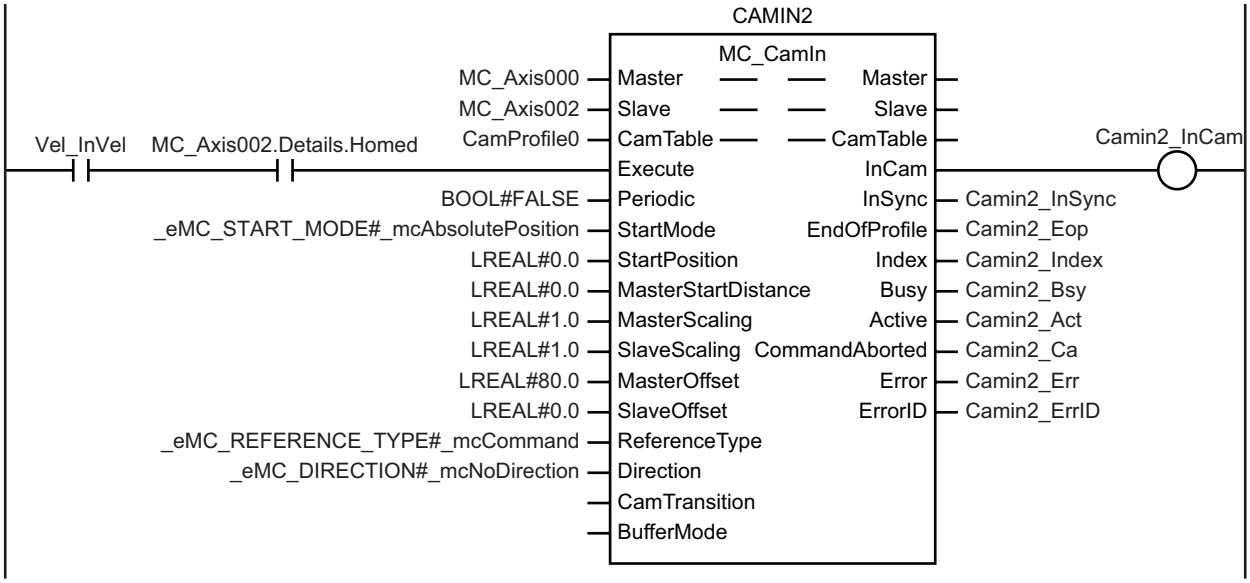
축 1이 원점 복귀 완료 후에 MC_MoveVelocity(속도 제어) 명령을 기동



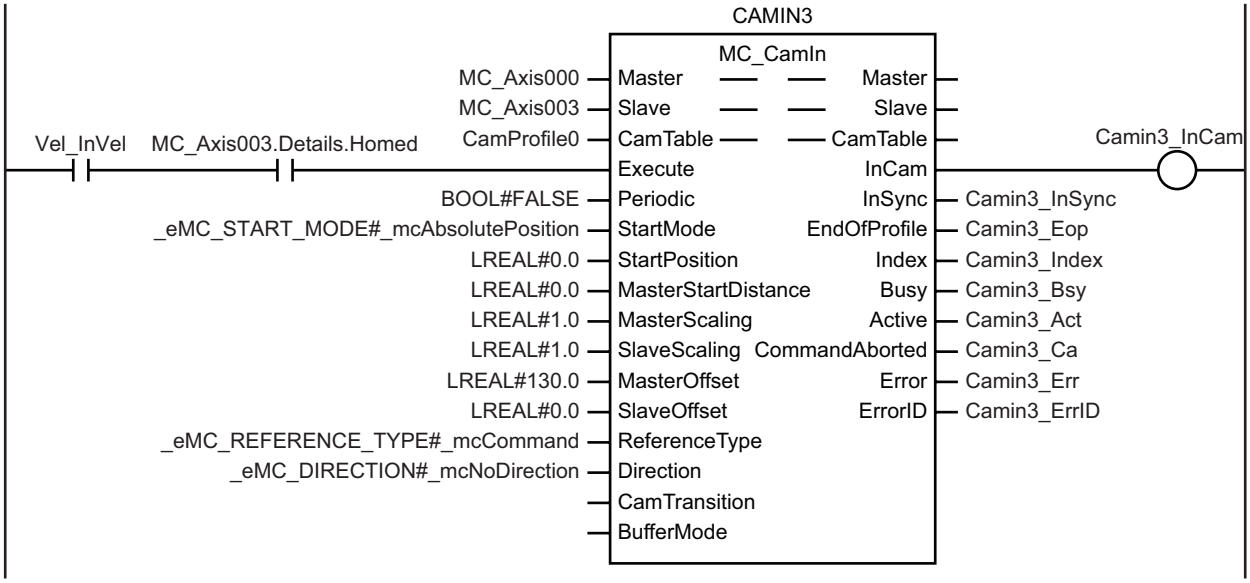
MC_MoveVelocity(속도 제어) 명령의 Vel_InVel이 TRUE이고 축 2의 원점 복귀 완료 후에 종축(축 2)에 설정된 MC_CamIn(캠 동작 시작) 명령을 실행



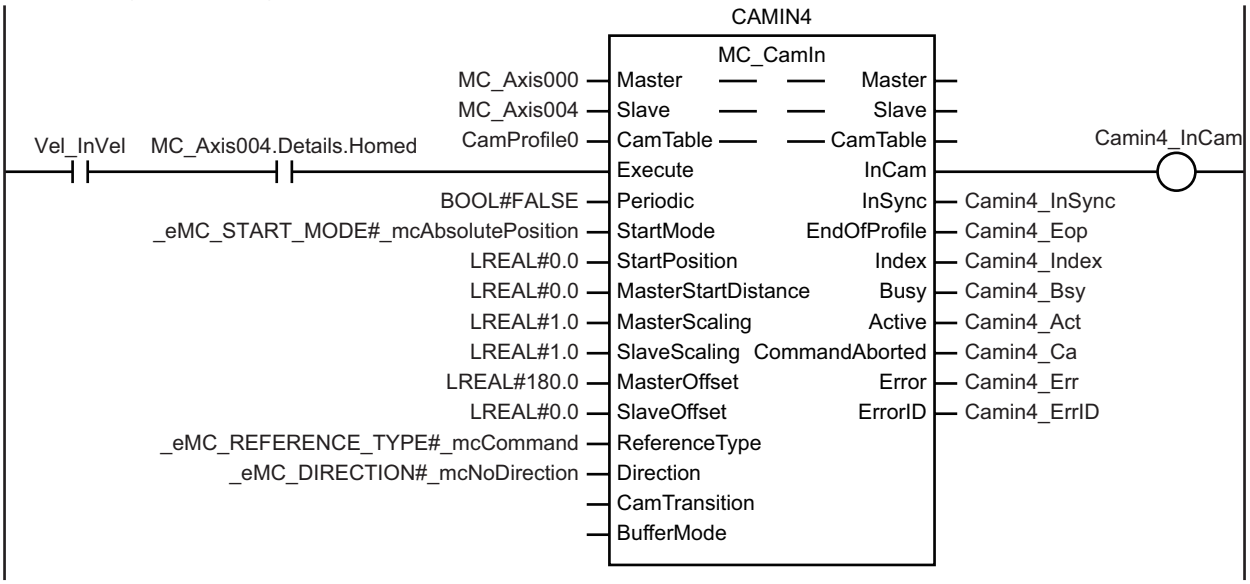
MC_MoveVelocity(속도 제어) 명령의 Vel_InVel이 TRUE이고 축 3의 원점 복귀 완료 후에 종축(축 3)에 설정된 MC_CamIn(캠 동작 시작) 명령을 실행



MC_MoveVelocity(속도 제어) 명령의 Vel_InVel이 TRUE이고 축 4의 원점 복귀 완료 후에 종축(축 4)에 설정된 MC_CamIn(캠 동작 시작) 명령을 실행



MC_MoveVelocity(속도 제어) 명령의 Vel_InVel이 TRUE이고 축 5의 원점 복귀 완료 후에 종축(축 5)에 설정된 MC_CamIn(캠 동작 시작) 명령을 실행

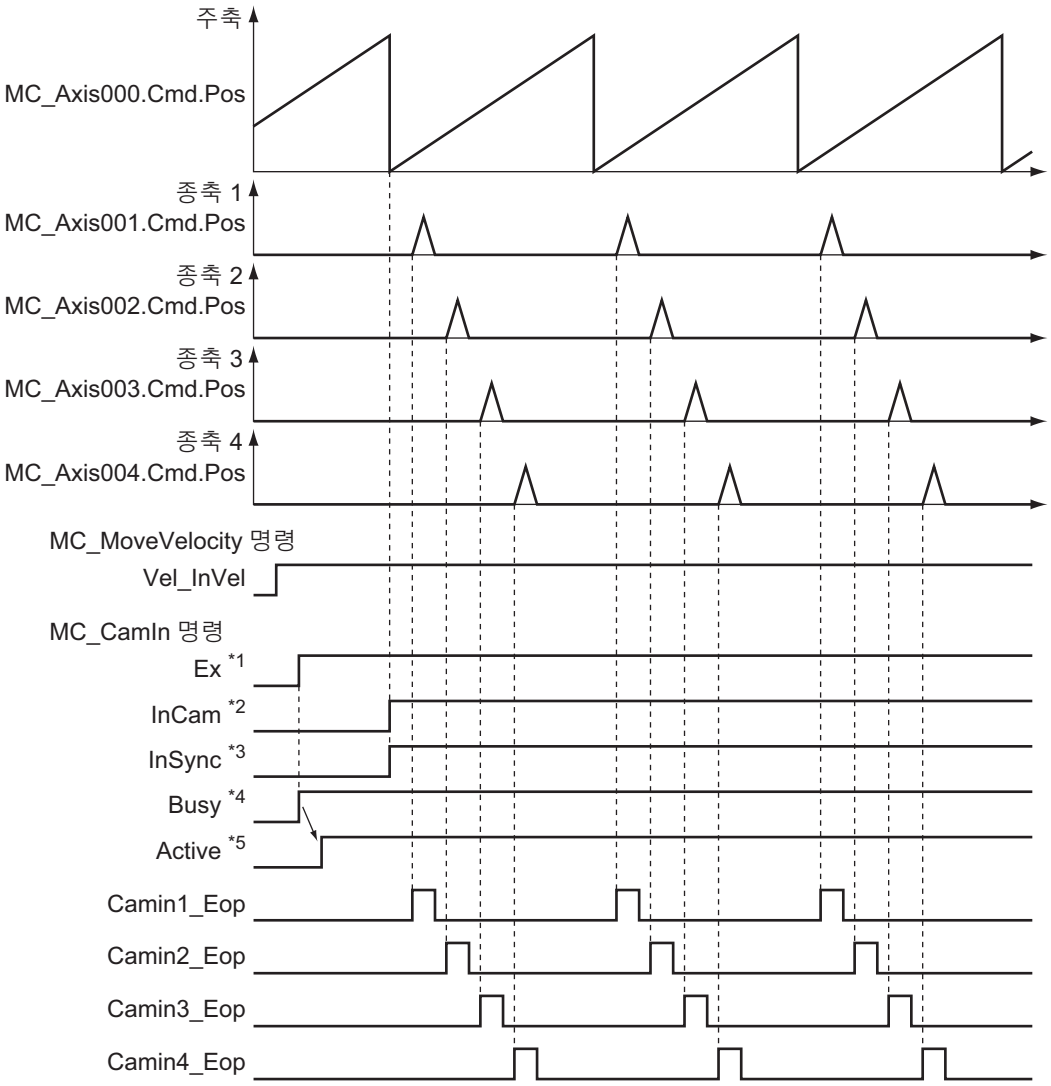


스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis002	_sAXIS_REF	–	종축인 축 3의 축 변수입니다.
MC_Axis003	_sAXIS_REF	–	종축인 축 4의 축 변수입니다.
MC_Axis004	_sAXIS_REF	–	종축인 축 5의 축 변수입니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Vel_InVel	BOOL	FALSE	MC_MoveVelocity의 인스턴스인 VEL의 출력 InVelocity에 할당한 변수입니다. 목표 속도에 도달하면 이 변수가 TRUE가 됩니다.
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	–	캠 데이터 변수입니다. MC_CamIn 인스턴스 CAMIN1~CAMIN4의 입력 CamTable에 이 변수를 지정합니다. 배열 요소 수 ARRAY[0..N]은 캠 편집기로 설정합니다.
Camin1_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_CamIn의 인스턴스 CAMIN1~CAMIN4가 기동합니다.
Camin2_Ex	BOOL	FALSE	
Camin3_Ex	BOOL	FALSE	
Camin4_Ex	BOOL	FALSE	
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트



*1 Camin1_InCam, Camin2_InCam, Camin3_InCam, Camin4_InCam이 동일한 타이밍입니다.

*2 Camin1_InSync, Camin2_InSync, Camin3_InSync, Camin4_InSync가 동일한 타이밍입니다.
이 예에서는 MasterStartDistance가 '0'이므로 주축이 '0° '에서부터 InSync가 TRUE가 됩니다.

*3 Camin1_Bsy, Camin2_Bsy, Camin3_Bsy, Camin4_Bsy가 동일한 타이밍입니다.

*4 Camin1_Act, Camin2_Act, Camin3_Act, Camin4_Act가 동일한 타이밍입니다.

● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag=FALSE THEN

    //MC_MoveVelocity Input Parameter
    Vel_Vel   := LREAL#1000.0;
    Vel_Acc   := LREAL#0.0;
    Vel_Dec   := LREAL#0.0;
    Vel_Dir   := _eMC_DIRECTION#_mcPositiveDirection;

    //MC_CamIn Input Parameter
    Camin1_Em     := TRUE;
    Camin1_Sm     := _eMC_START_MODE#_mcAbsolutePosition;
    Camin1_Sp     := LREAL#0.0;
    Camin1_Msd    := LREAL#0.0;
    Camin1_Ms     := LREAL#1.0;
    Camin1_Ss     := LREAL#1.0;
    Camin1_Mo     := LREAL#30.0;
```

```
    Camin1_So     := LREAL#0.0;
    Camin1_Rt     := _eMC_REFERENCE_TYPE#_mcCommand;
    Camin1_Dir    := _eMC_DIRECTION#_mcNoDirection;

    Camin2_Em     := TRUE;
    Camin2_Sm     := _eMC_START_MODE#_mcAbsolutePosition;
    Camin2_Sp     := LREAL#0.0;
    Camin2_Msd    := LREAL#0.0;
    Camin2_Ms     := LREAL#1.0;
    Camin2_Ss     := LREAL#1.0;
    Camin2_Mo     := LREAL#80.0;
    Camin2_So     := LREAL#0.0;
    Camin2_Rt     := _eMC_REFERENCE_TYPE#_mcCommand;
    Camin2_Dir    := _eMC_DIRECTION#_mcNoDirection;

    Camin3_Em     := TRUE;
    Camin3_Sm     := _eMC_START_MODE#_mcAbsolutePosition;
    Camin3_Sp     := LREAL#0.0;
    Camin3_Msd    := LREAL#0.0;
    Camin3_Ms     := LREAL#1.0;
    Camin3_Ss     := LREAL#1.0;
    Camin3_Mo     := LREAL#130.0;
    Camin3_So     := LREAL#0.0;
    Camin3_Rt     := _eMC_REFERENCE_TYPE#_mcCommand;
    Camin3_Dir    := _eMC_DIRECTION#_mcNoDirection;

    Camin4_Em     := TRUE;
    Camin4_Sm     := _eMC_START_MODE#_mcAbsolutePosition;
    Camin4_Sp     := LREAL#0.0;
    Camin4_Msd    := LREAL#0.0;
    Camin4_Ms     := LREAL#1.0;
    Camin4_Ss     := LREAL#1.0;
    Camin4_Mo     := LREAL#180.0;
    Camin4_So     := LREAL#0.0;
    Camin4_Rt     := _eMC_REFERENCE_TYPE#_mcCommand;
    Camin4_Dir    := _eMC_DIRECTION#_mcNoDirection;


    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;


//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 1을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 2를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
END_IF;
```

```
//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 3을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis002.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis002.Cfg.NodeAddress]=FALSE) THEN
    Pwr3_En:=TRUE;
ELSE
    Pwr3_En:=FALSE;
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 4를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis003.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis003.Cfg.NodeAddress]=FALSE) THEN
    Pwr4_En:=TRUE;
ELSE
    Pwr4_En:=FALSE;
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 5를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis004.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis004.Cfg.NodeAddress]=FALSE) THEN
    Pwr5_En:=TRUE;
ELSE
    Pwr5_En:=FALSE;
END_IF;

//축 1~축 5에서 가벼운 이상이 발생하면 이상 시 처리 FaultHandler를 실행합니다.
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE)
OR (MC_Axis002.MFaultLvl.Active=TRUE)
OR (MC_Axis003.MFaultLvl.Active=TRUE)
OR (MC_Axis004.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

//축 1이 서보 ON 상태이고 원점 미확정 상태일 때 축 1의 원점 복귀를 실행합니다.
IF  (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//축 2가 서보 ON 상태이고 원점 미확정 상태일 때 축 2의 원점 복귀를 실행합니다.
IF  (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//축 3이 서보 ON 상태이고 원점 미확정 상태일 때 축 3의 원점 복귀를 실행합니다.
IF  (Pwr3_Status=TRUE) AND (MC_Axis002.Details.Homed=FALSE) THEN
    Hm3_Ex:=TRUE;
END_IF;

//축 4가 서보 ON 상태이고 원점 미확정 상태일 때 축 4의 원점 복귀를 실행합니다.
IF  (Pwr4_Status=TRUE) AND (MC_Axis003.Details.Homed=FALSE) THEN
    Hm4_Ex:=TRUE;
END_IF;

//축 5가 서보 ON 상태이고 원점 미확정 상태일 때 축 5의 원점 복귀를 실행합니다.
IF  (Pwr5_Status=TRUE) AND (MC_Axis004.Details.Homed=FALSE) THEN
```

```
    Hm5_Ex:=TRUE;
END_IF;

//축 1이 원점 복귀 완료 후에 MC_MoveVelocity를 실행합니다.
IF Hm1_D=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

// 축 2가 원점 확정 상태이고 MC_MoveVelocity의 Vel_InVel가 TRUE일 때
// 종축이 축 2에 설정된 MC_CamIn를 기동합니다.
IF (Vel_InVel=TRUE) AND (MC_Axis001.Details.Homed=TRUE) THEN
    Camin1_Ex := TRUE;
END_IF;

// 축 3이 원점 확정 상태이고 MC_MoveVelocity의 Vel_InVel가 TRUE일 때
// 종축이 축 3에 설정된 MC_CamIn를 기동합니다.
IF (Vel_InVel=TRUE) AND (MC_Axis002.Details.Homed=TRUE) THEN
    Camin2_Ex := TRUE;
END_IF;

// 축 4가 원점 확정 상태이고 MC_MoveVelocity의 Vel_InVel가 TRUE일 때
// 종축이 축 4에 설정된 MC_CamIn를 기동합니다.
IF (Vel_InVel=TRUE) AND (MC_Axis003.Details.Homed=TRUE) THEN
    Camin3_Ex := TRUE;
END_IF;

// 축 5가 원점 확정 상태이고 MC_MoveVelocity의 Vel_InVel가 TRUE일 때
// 종축이 축 5에 설정된 MC_CamIn를 기동합니다.
IF (Vel_InVel=TRUE) AND (MC_Axis004.Details.Homed=TRUE) THEN
    Camin4_Ex := TRUE;
END_IF;


//축 1의 MC_Power
PWR1(
    Axis        := MC_Axis000,
    Enable      := Pwr1_En,
    Status      => Pwr1_Status,
    Busy        => Pwr1_Bsy,
    Error       => Pwr1_Err,
    ErrorID     => Pwr1_ErrID
);

//축 2의 MC_Power
PWR2(
    Axis        := MC_Axis001,
    Enable      := Pwr2_En,
    Status      => Pwr2_Status,
    Busy        => Pwr2_Bsy,
    Error       => Pwr2_Err,
    ErrorID     => Pwr2_ErrID
);

//축 3의 MC_Power
PWR3(
    Axis        := MC_Axis002,
    Enable      := Pwr3_En,
    Status      => Pwr3_Status,
    Busy        => Pwr3_Bsy,
    Error       => Pwr3_Err,
    ErrorID     => Pwr3_ErrID
);

//축 4의 MC_Power
PWR4(
```

```
    Axis        := MC_Axis003,
    Enable      := Pwr4_En,
    Status      => Pwr4_Status,
    Busy        => Pwr4_Bsy,
    Error       => Pwr4_Err,
    ErrorID     => Pwr4_ErrID
);

//축 5의 MC_Power
PWR5(
    Axis        := MC_Axis004,
    Enable      := Pwr5_En,
    Status      => Pwr5_Status,
    Busy        => Pwr5_Bsy,
    Error       => Pwr5_Err,
    ErrorID     => Pwr5_ErrID
);

//축 1의 MC_Home
HM1(
    Axis                := MC_Axis000,
    Execute             := Hm1_Ex,
    Done                => Hm_D,
    Busy                => Hm1_Bsy,
    CommandAborted      => Hm1_Ca,
    Error               => Hm1_Err,
    ErrorID             => Hm1_ErrID
);

//축 2의 MC_Home
HM2(
    Axis                := MC_Axis001,
    Execute             := Hm2_Ex,
    Done                => Hm2_D,
    Busy                => Hm2_Bsy,
    CommandAborted      => Hm2_Ca,
    Error               => Hm2_Err,
    ErrorID             => Hm2_ErrID
);

//축 3의 MC_Home
HM3(
    Axis                := MC_Axis002,
    Execute             := Hm3_Ex,
    Done                => Hm3_D,
    Busy                => Hm3_Bsy,
    CommandAborted      => Hm3_Ca,
    Error               => Hm3_Err,
    ErrorID             => Hm3_ErrID
);

//축 4의 MC_Home
HM4(
    Axis                := MC_Axis003,
    Execute             := Hm4_Ex,
    Done                => Hm4_D,
    Busy                => Hm4_Bsy,
    CommandAborted      => Hm4_Ca,
    Error               => Hm4_Err,
    ErrorID             => Hm4_ErrID
);

//축 5의 MC_Home
HM5(
    Axis                := MC_Axis004,
```

```
    Execute             := Hm5_Ex,
    Done                => Hm5_D,
    Busy                => Hm5_Bsy,
    CommandAborted      => Hm5_Ca,
    Error               => Hm5_Err,
    ErrorID             => Hm5_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
CAMIN1(
    Master                  := MC_Axis000,
    Slave                   := MC_Axis001,
    CamTable                := CamProfile0,
    Execute                 := Camin1_Ex,
    Periodic                := Camin1_Em,
    StartMode               := Camin1_Sm,
    StartPosition           := Camin1_Sp,
    MasterStartDistance     := Camin1_Msd,
    MasterScaling           := Camin1_Ms,
    SlaveScaling            := Camin1_Ss,
    MasterOffset            := Camin1_Mo,
    SlaveOffset             := Camin1_So,
    ReferenceType           := Camin1_Rt,
    Direction               := Camin1_Dir,
    InCam                   => Camin1_InCam,
    InSync                  => Camin1_InSync,
    EndOfProfile            => Camin1_Eop,
    Index                   => Camin1_Index,
    Busy                    => Camin1_Bsy,
    Active                  => Camin1_Act,
    CommandAborted          => Camin1_Ca,
    Error                   => Camin1_Err,
    ErrorID                 => Camin1_ErrID
);

CAMIN2(
    Master                  := MC_Axis000,
    Slave                   := MC_Axis002,
    CamTable                := CamProfile0,
    Execute                 := Camin2_Ex,
    Periodic                := Camin2_Em,
    StartMode               := Camin2_Sm,
    StartPosition           := Camin2_Sp,
    MasterStartDistance     := Camin2_Msd,
    MasterScaling           := Camin2_Ms,
    SlaveScaling            := Camin2_Ss,
    MasterOffset            := Camin2_Mo,
    SlaveOffset             := Camin2_So,
    ReferenceType           := Camin2_Rt,
```

```
    Direction           := Camin2_Dir,
    InCam               => Camin2_InCam,
    InSync              => Camin2_InSync,
    EndOfProfile        => Camin2_Eop,
    Index               => Camin2_Index,
    Busy                => Camin2_Bsy,
    Active              => Camin2_Act,
    CommandAborted      => Camin2_Ca,
    Error               => Camin2_Err,
    ErrorID             => Camin2_ErrID
);

CAMIN3(
    Master              := MC_Axis000,
    Slave               := MC_Axis003,
    CamTable            := CamProfile0,
    Execute             := Camin3_Ex,
    Periodic            := Camin3_Em,
    StartMode           := Camin3_Sm,
    StartPosition       := Camin3_Sp,
    MasterStartDistance := Camin3_Msd,
    MasterScaling       := Camin3_Ms,
    SlaveScaling        := Camin3_Ss,
    MasterOffset        := Camin3_Mo,
    SlaveOffset         := Camin3_So,
    ReferenceType       := Camin3_Rt,
    Direction           := Camin3_Dir,
    InCam               => Camin3_InCam,
    InSync              => Camin3_InSync,
    EndOfProfile        => Camin3_Eop,
    Index               => Camin3_Index,
    Busy                => Camin3_Bsy,
    Active              => Camin3_Act,
    CommandAborted      => Camin3_Ca,
    Error               => Camin3_Err,
    ErrorID             => Camin3_ErrID
);

CAMIN4(
    Master              := MC_Axis000,
    Slave               := MC_Axis004,
    CamTable            := CamProfile0,
    Execute             := Camin4_Ex,
    Periodic            := Camin4_Em,
    StartMode           := Camin4_Sm,
    StartPosition       := Camin4_Sp,
    MasterStartDistance := Camin4_Msd,
    MasterScaling       := Camin4_Ms,
    SlaveScaling        := Camin4_Ss,
    MasterOffset        := Camin4_Mo,
    SlaveOffset         := Camin4_So,
    ReferenceType       := Camin4_Rt,
    Direction           := Camin4_Dir,
    InCam               => Camin4_InCam,
    InSync              => Camin4_InSync,
    EndOfProfile        => Camin4_Eop,
    Index               => Camin4_Index,
    Busy                => Camin4_Bsy,
    Active              => Camin4_Act,
    CommandAborted      => Camin4_Ca,
    Error               => Camin4_Err,
    ErrorID             => Camin4_ErrID
);
```

MC_CamOut

입력 파라메터에서 지정한 축의 캠 동작을 종료합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_CamOut	캠 동작 해제	FB	MC_CamOut_instance MC_CamOut Slave — — Slave Execute — Done Deceleration — Busy Jerk — CommandAborted OutMode — Error — ErrorID	MC_CamOut_instance (Slave := 《파라메터》, Execute := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, OutMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Deceleration	감속도	LREAL	양수값 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk (Reserved)	저크	LREAL	0	0	(Reserved)
OutMode (Reserved)	동기 해제 모드 선택	_eMC_OUT_ MODE	0: _mcStop	0 *2	(Reserved)

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

*2. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	속도 '0'에 도달했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Slave	종축	_sAXIS_REF	−	종축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

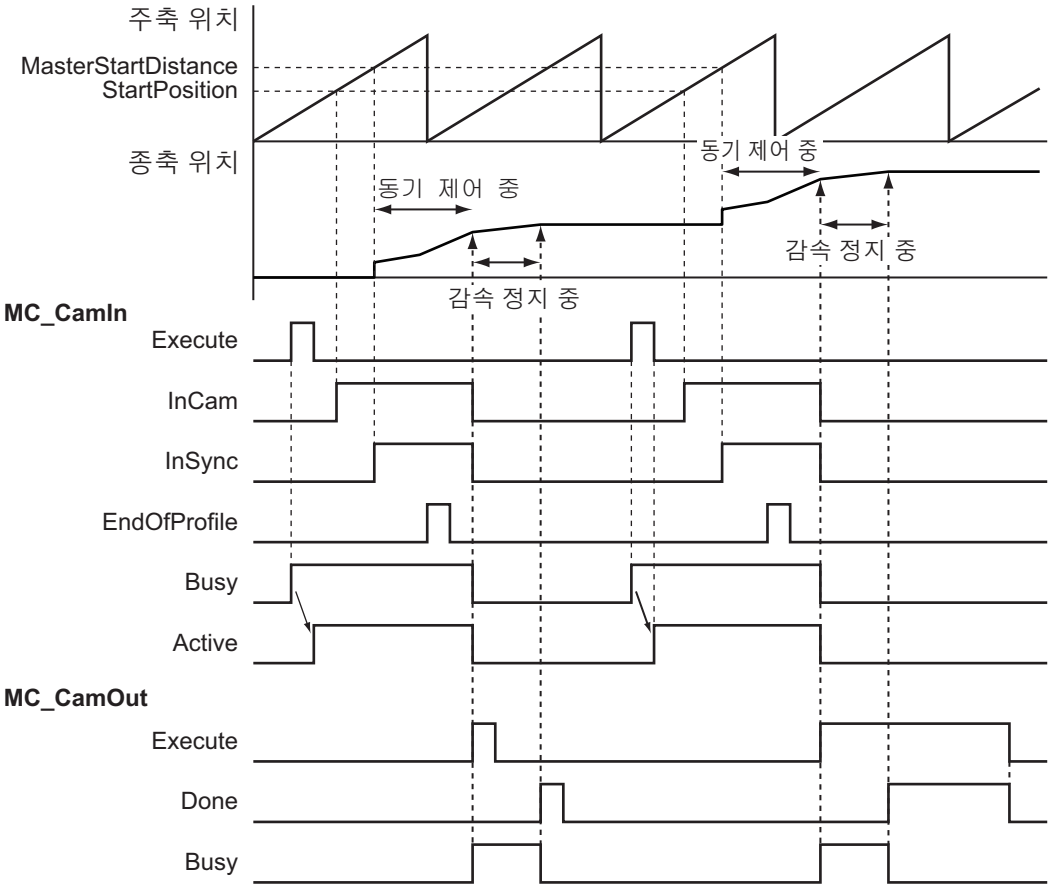
기능 설명

- 이 명령은 종축의 캠 동작을 해제합니다.
- Execute(기동)를 TRUE로 하면 Deceleration(감속도)에 따라 속도 '0'을 향해서 감속합니다.
- 명령 속도가 '0'이 되면 완료됩니다.
- 캠 동작을 하고 있지 않은 축에 대해 이 명령을 기동하면 이상이 됩니다.

사용상의 주의 사항

- 캠 데이터 변수는 글로벌 변수로, 복수의 태스크에서 캠 데이터 변수값을 참조하거나 변경할 수 있습니다. 복수의 태스크로부터 캠 데이터 변수값을 변경하려면 복수의 태스크에 의한 재기입 처리로 경쟁이 발생하지 않도록 프로그램을 작성하십시오.
- '글로벌 변수의 태스크 간 배타적 제어' 기능을 이용해서 캠 데이터 변수를 배타적으로 제어하려면 배타 대상인 캠 데이터 변수를 이용해서 모션 명령을 기동하지 마십시오. 가벼운 수준의 이상인 '캠 테이블 지정 오류(에러 코드: 5439 Hex)'가 발생합니다.

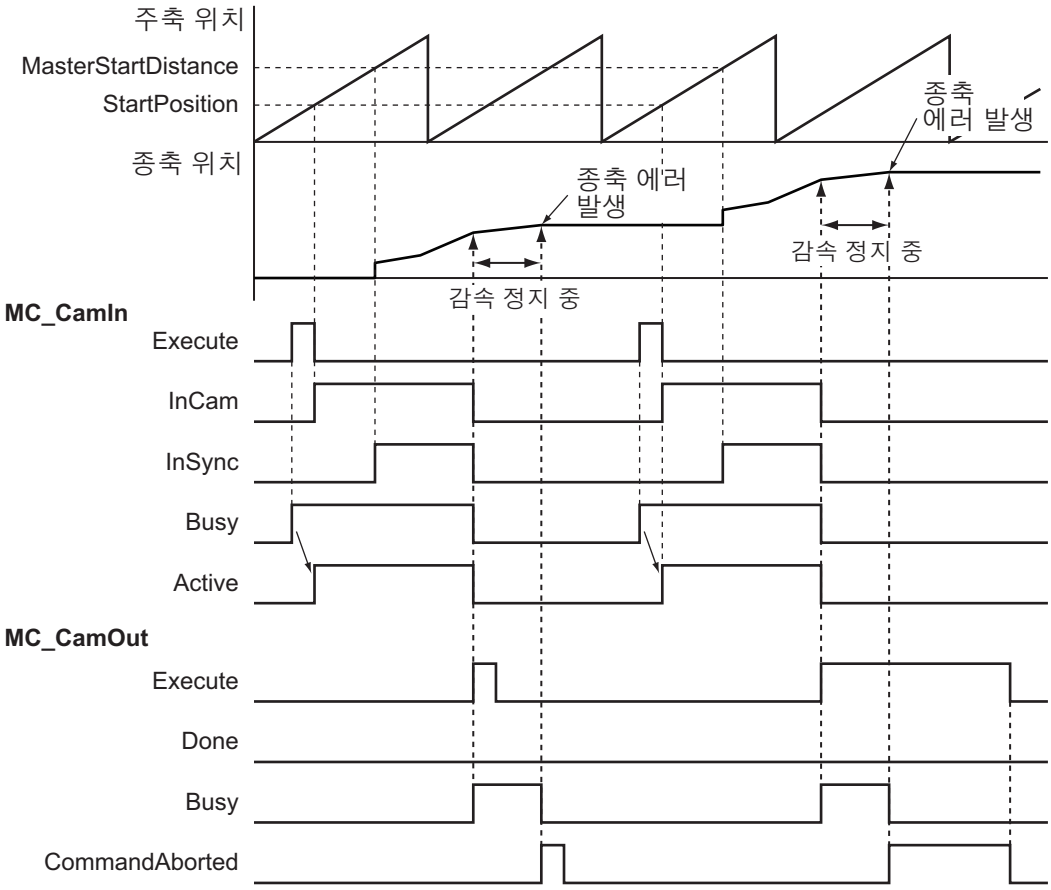
타이밍 차트


주축 위치
MasterStartDistance
StartPosition
종축 위치
동기 제어 중
동기 제어 중
감속 정지 중
감속 정지 중
MC_CamIn
Execute
InCam
InSync
EndOfProfile
Busy
Active
MC_CamOut
Execute
Done
Busy

명령 중지

이 명령 실행 중에 종축에서 축 이상이 발생하면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(동작 중)는 FALSE가 됩니다.
또한 이 명령의 Deceleration(감속도)에 따라 축은 감속합니다.

축 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

● 다른 명령 실행 중의 이 명령 기동

MC_CamIn(캠 동작 시작) 명령 실행 중에 이 명령을 기동하면 MC_CamIn(캠 동작 시작) 명령을 CommandAborted(실행 중단)하고 이 명령은 Busy(실행 중)가 됩니다.
MC_CamIn(캠 동작 시작) 명령 실행 중이 아닐 때 이 명령을 기동하면 Error(에러)가 됩니다.

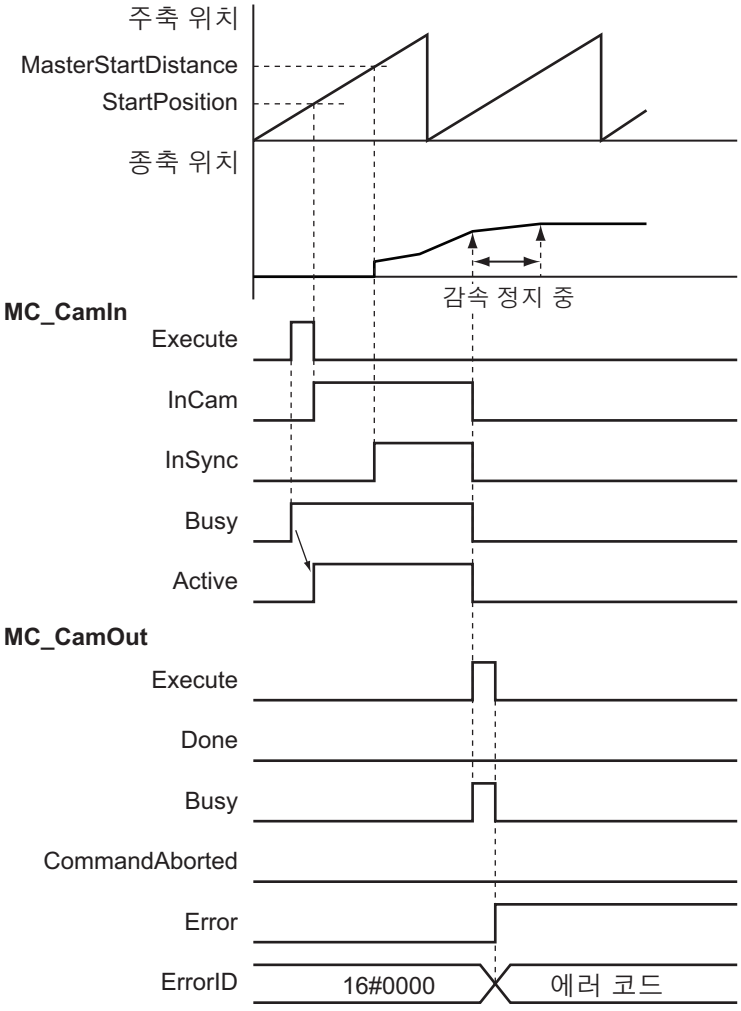
● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하려면 종축을 지정하십시오.
이 명령에 대한 모션 명령 다중 기동을 실행하는 경우 Aborting, Buffered가 Blending 가능합니다.

이상

이 명령 기동 시에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트

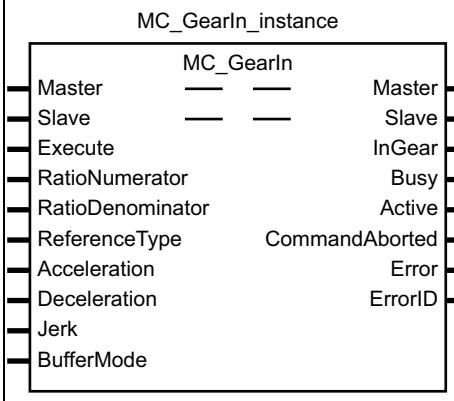


● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_GearIn

주축과 종축 사이의 기어비를 설정하고 기어 동작을 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_GearIn	기어 동작 시작	FB		MC_GearIn_instance (Master := 《파라메터》, Slave := 《파라메터》, Execute := 《파라메터》, RatioNumerator := 《파라메터》, RatioDenominator := 《파라메터》, ReferenceType := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, BufferMode := 《파라메터》, InGear => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Ratio Numerator	기어비 분자	DINT *1	양수 또는 음수 *1	10000	주축과 종축 사이의 전자 기어 분자를 지정합니다.
Ratio Denominator	기어비 분모	UDINT *2	양수	10000	주축과 종축 사이의 전자 기어 분모를 지정합니다.
Reference Type*3	위치 종류 선택	_eMC_ REFERENCE _TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *4	위치 종류를 지정합니다. 0 : 명령 위치(직전 프라이머리 주기에서의 계산값) 1 : 피드백 위치(동일 프라이머리 주기에서의 취득값) 2 : 명령 위치(동일 프라이머리 주기에서의 계산값)
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *5
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *3
Jerk (Reserved)	저크	LREAL	0	0	(Reserved)
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 *2	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered

*1. CPU 유니트 Ver1.02 이상과 Sysmac StudioVer1.03 이상의 조합에서 사용할 수 있습니다. 그 이전의 조합에서는 데이터 형식은 'UINT', 유효 범위 '양수'가 됩니다.
*2. CPU 유니트 Ver1.02 이상과 Sysmac StudioVer1.03 이상의 조합에서 사용할 수 있습니다. 그 이전의 조합에서는 데이터 형식은 'UINT'가 됩니다.
*3. _mcLatestCommand를 사용하는 경우는
Master(주축)에 설정된 모션 제어 시스템 변수의 축 번호 < Slave(종축)에 설정된 모션 제어 시스템 변수의 축 번호
가 되도록 주축과 종축을 사용해야 합니다.
*4. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*5. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
InGear	기어비 도달	BOOL	TRUE, FALSE	종축이 목표 속도에 도달했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
InGear	다음 관계가 성립할 때 • 가속도: 종축 속도≧주축 속도×기어비 • 감속도: 종축 속도≦주축 속도×기어비	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Busy	Execute의 기동	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동 (Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때 • MC_GearOut 명령이 기동되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Master	주축	_sAXIS_REF	−	주축을 지정합니다. *1
Slave	종축	_sAXIS_REF	−	종축을 지정합니다. *1

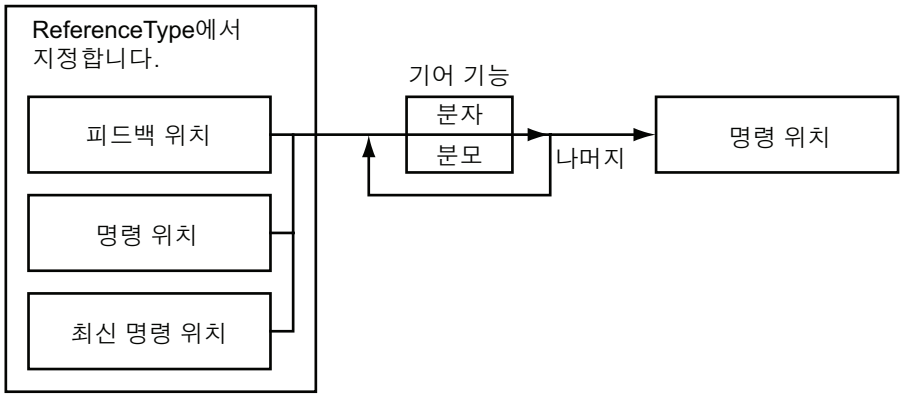
*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

사용상의 주의 사항

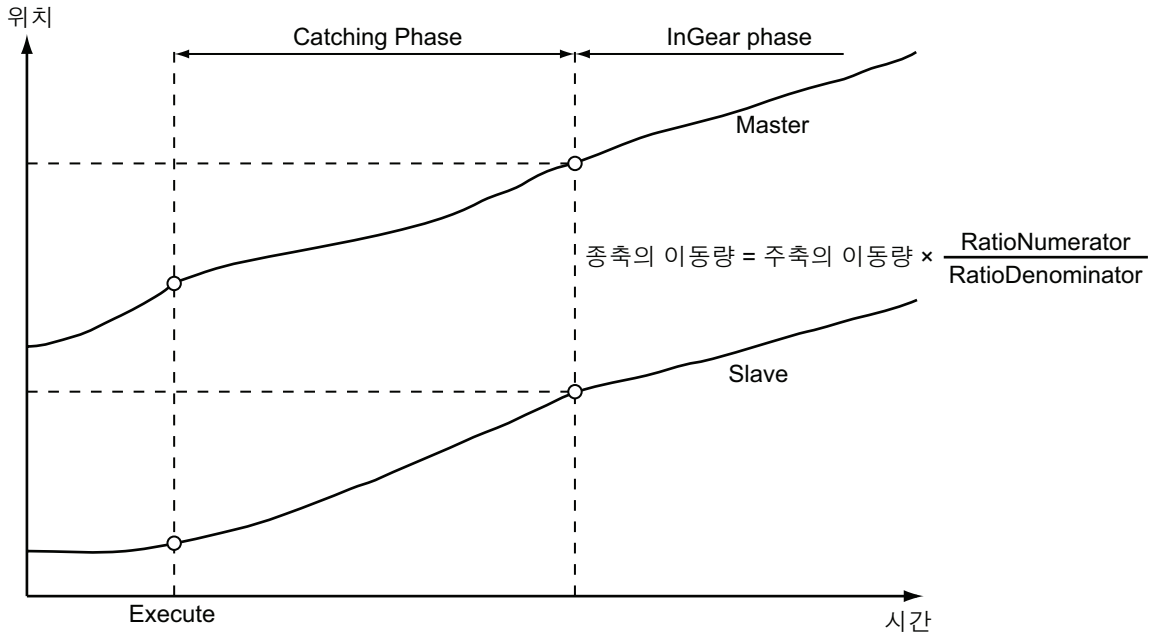
주축과 종축에 같은 축을 지정하면 가벼운 이상 수준의 '주축 종축 동일(에러 코드: 5436 Hex)' 이상이 됩니다.

기능 설명

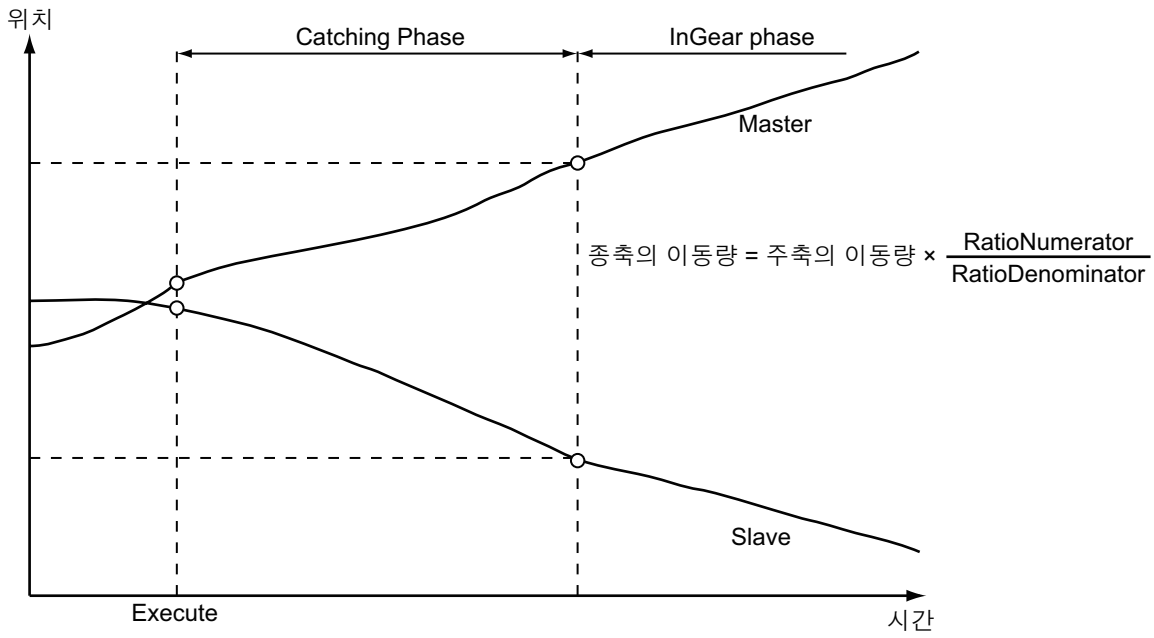
- 동작 대상 축을 Slave(종축)에서 지정하고 RatioNumerator(기어비 분자), RatioDenominator(기어비 분모), ReferenceType(위치 종류 선택), Acceleration(가속도), Deceleration(감속도)을 지정해서 기어를 동작합니다.
- Master(주축)에는 명령 위치와 피드백 위치, 최신 명령 위치를 지정할 수 있습니다.



- 동작 시작 후 Slave(종축)는 Master(주축)의 속도에 기어비를 곱한 속도를 목표 속도로 가감속 동작을 합니다.
- 목표 속도에 도달할 때까지를 Catching phase(추종 중)라고 하며, 도달 후를 InGear phase(기어 동기 중)라고 합니다.
- 기어비가 양수일 때는 Slave(종축)는 Master(주축)와 동일한 방향으로 이동합니다.



- 기어비가 음수일 때는 Slave(종축)는 Master(주축)의 반대 방향으로 이동합니다.



- Execute(기동)가 기동하면 전자 기어의 동작을 시작합니다.

안전상의 중요 사항

이 명령을 사용하는 경우는 Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하지 마십시오.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하면 Slave(종축)가 급격하게 추종하는 경우가 있어 위험합니다.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 사용하려면 Master(주축)와 Slave(종축)의 관계를 일단 해제하고 나서 실행하십시오.
주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

● ReferenceType(위치 종류 선택)

위치 종류를 다음에서 선택할 수 있습니다.

- _mcCommand: 명령 위치(직전 프라이머리 주기에서의 계산값)
  현재 주기에 대해 직전의 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  종축의 명령 위치를 계산하기 직전의 프라이머리 정주기 태스크에서 계산한 주축의 명령 위치를 사용합니다.
- _mcFeedback: 동일 프라이머리 주기에서의 취득값
  동일 프라이머리 주기에서 취득한 주축의 피드백 위치를 사용합니다.
- _mcLatestCommand: 명령 위치(동일 프라이머리 주기에서의 계산값)
  동일 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  _mcCommand보다도 새로운 정보를 사용할 수 있습니다. 단, 주축 번호는 종축 번호보다 빠른 번호를 설정해야 합니다.
  종축이 주축보다 축 번호가 빠르면 이 명령의 Error(에러)가 TRUE가 됩니다. ErrorID(에러 코드)에는 '주축/종축 축 번호 비오름차순(에러 코드: 5438 Hex)'이 출력됩니다.

참고

동일 프라이머리 주기에서 계산한 명령 위치를 선택하는 것이 직전 프라이머리 주기에서의 계산값을 선택하는 것보다 동기 정확도는 향상됩니다.
단, 'Master(주축)에 설정된 모션 제어 시스템 변수의 축 번호 < Slave(종축)에 설정된 모션 제어 시스템 변수의 축 번호'가 되도록 주축과 종축을 사용해야 합니다.

● 축 종류와 위치 종류의 관계

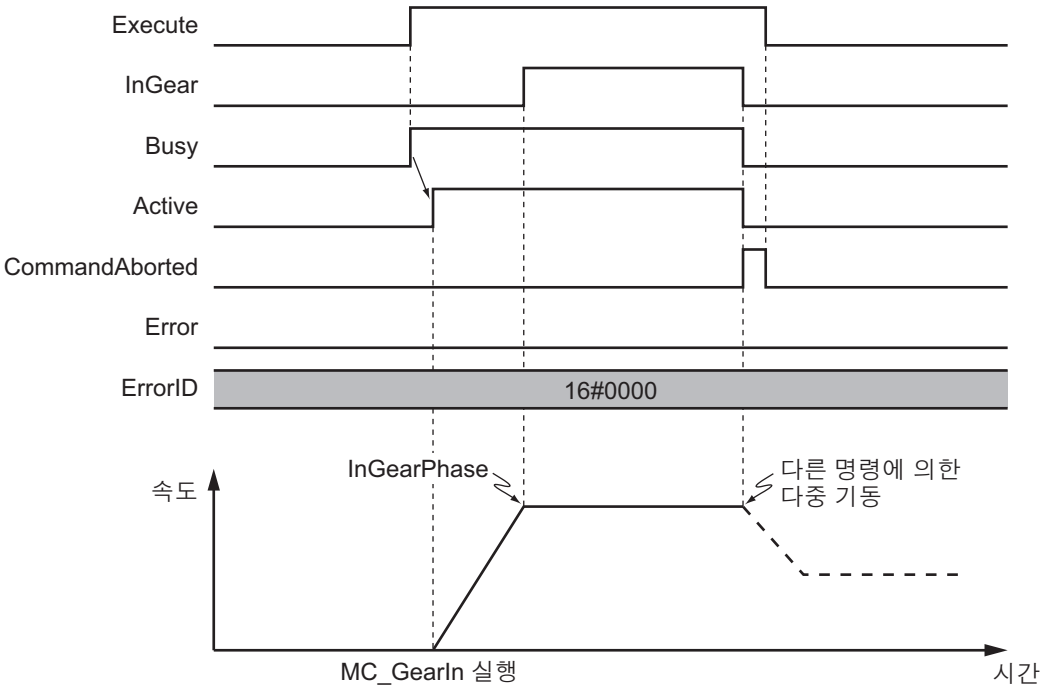
감시 가능한 축 종류와 감시하는 위치 종류의 관계는 다음과 같습니다.

축 종류	ReferenceType	
	_mcCommand, 또는 _mcLatestCommand	_mcFeedback
서보 축	○	○
엔코더 축	× *1	○
가상 서보 축	○	○
가상 엔코더 축	× *1	○

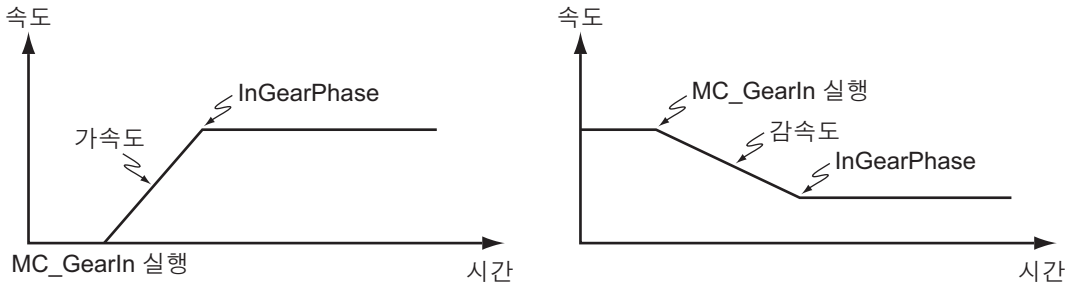
*1. 명령 기동 시에 '위치 종류 선택 범위 외(에러 코드: 5430 Hex)'가 발생합니다.

타이밍 차트

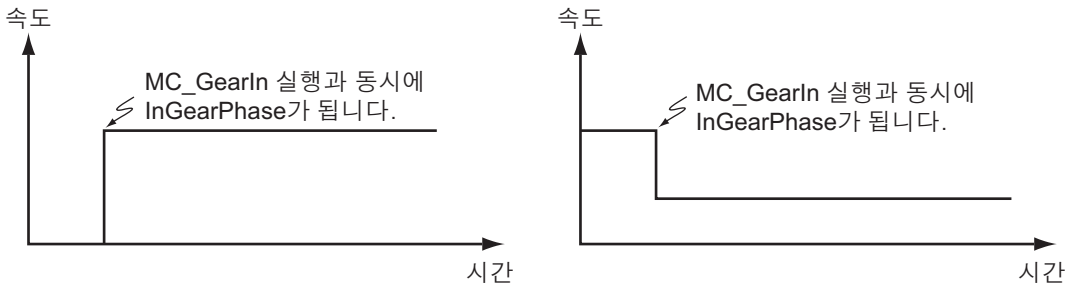
- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- 목표 속도에 도달하면 InGear(기어비 도달)가 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중단되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중), InGear(기어비 도달)가 FALSE가 됩니다.
- 전자 기어 동작을 도중에 종료시키려면 MC_GearOut(기어 동작 해제) 명령 또는 MC_Stop(강제 정지) 명령을 사용합니다.



입력 변수에 Acceleration(가속도), Deceleration(감속도)을 지정할 수 있습니다.



Acceleration(가속도) 또는 Deceleration(감속도)을 '0'으로 지정해서 기동하면 가감속하지 않고 목표 속도에 도달합니다.

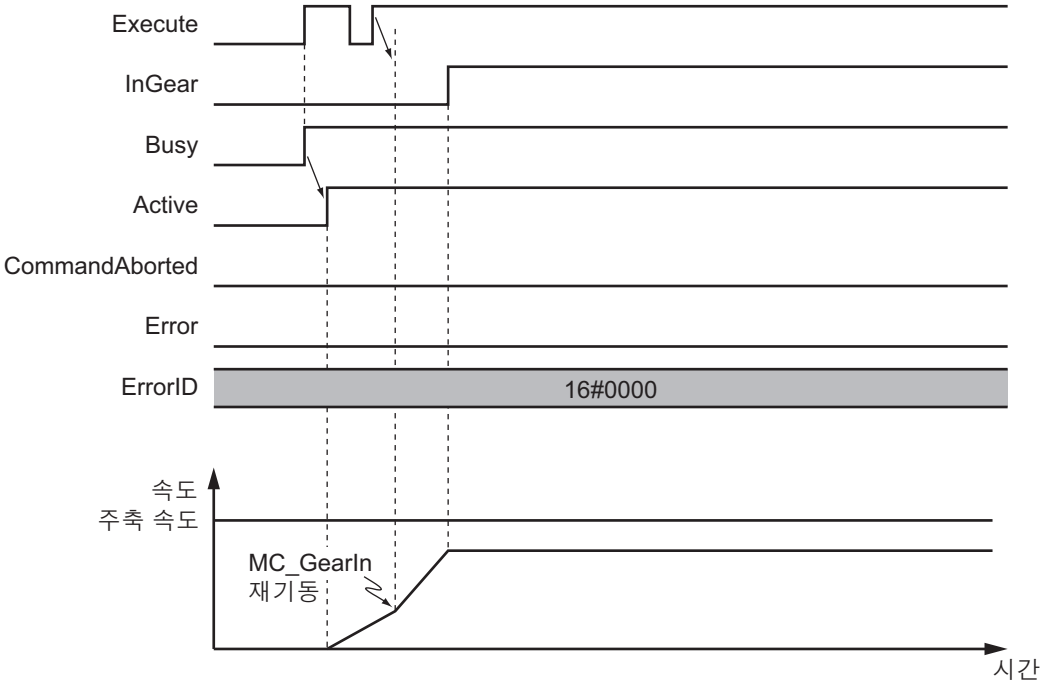


모션 명령 재기동

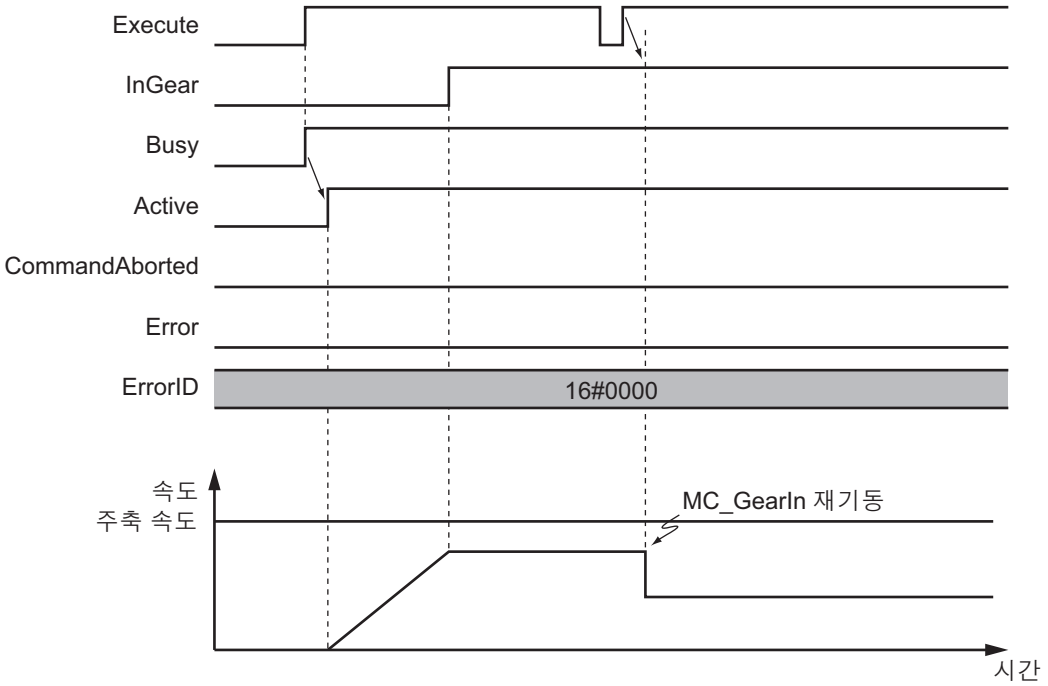
위치 결정 동작 중에 입력 변수를 변경하고 Execute(기동)를 다시 TRUE로 하면 이 명령의 동작을 변경할 수 있습니다.
모션 명령 재기동으로 변경 가능한 입력 변수는 RatioNumerator(기어비 분자), RatioDenominator(기어비 분모), Acceleration(가속도), Deceleration(감속도)입니다.

모션 명령 재기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

참고로 Catching phase(추종 중)로 재기동하고, Acceleration(가속도)을 변경한 경우의 타이밍 차트는 다음과 같습니다.



InGear phase(기어 동기 중)로 재기동하고 RatioNumerator(기어비 분자), RatioDenominator(기어비 분모)를 변경한 경우의 타이밍 차트는 다음과 같습니다.
동작은 Acceleration(가속도)과 Deceleration(감속도)이 모두 '0'으로 설정되어 있는 경우와 동일합니다.



모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다.
각 축에 1개까지 버퍼링할 수 있습니다.
명령 다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.
Buffered	현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

종축에 대해 Aborting으로 다중 명령을 기동할 수 있습니다.
이 경우 기어 동작을 중지하고 다중 기동된 동작을 시작합니다.
Aborting 이외에 의한 명령 다중 기동은 불가능합니다.

이상

이 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.



참고

- 동기 제어 중에는 주축의 이상 상태로부터 영향을 받지 않습니다. 주축의 이상 상태가 해제되고 정상적으로 동작하면 종축은 전자 기어 동작을 계속합니다.
- 이 명령 기동 시 및 이 명령 실행 중에 종축에 이상이 발생해도 주축에 영향을 미치지 않습니다.

● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기에는 기어비에 비례해서 동작하는 샘플 프로그램이 실려 있습니다.



참고

예약(Reserved)으로 취급되는 입력 변수에는 초기값 이외의 값을 지정할 수 없습니다.
이 예에서는 파라메터를 지정하지 않습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축(주축)
축 2	서보 축(종축)
축 3	서보 축(종축)

카운터 모드

축	카운터 모드
축 1	로터리 모드
축 2	로터리 모드
축 3	로터리 모드

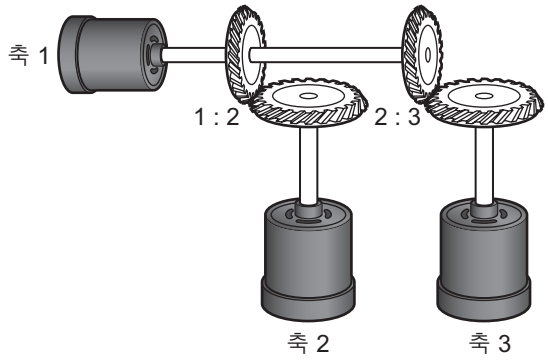
링 카운터

축	상한값	하한값
축 1	360	0
축 2	360	0
축 3	360	0

좌표 단위 선택

축	좌표 단위
축 1	mm
축 2	mm
축 3	mm

동작 예



1 **주축의 시작**

주축(축 1)은 실제 서보 축이며 속도 제어로 동작합니다.

2 **종축의 기동**

주축의 피드백 속도가 목표 속도에 도달하면 주축의 피드백 위치에 대해 종축(축 2)을 기어비 1:2로, 종축(축 3)을 기어비 2:3으로 기어를 동작시킵니다.

3 **종축의 정지**

주축의 피드백 위치(MC_Axis000. Act. Pos)가 1000.0 이상이 되면 축 2의 기어 동작을 해제하고 감속도 DecRate로 축 2를 감속 정지시킵니다. 이 때 축 3은 기어 동작을 계속하게 합니다.

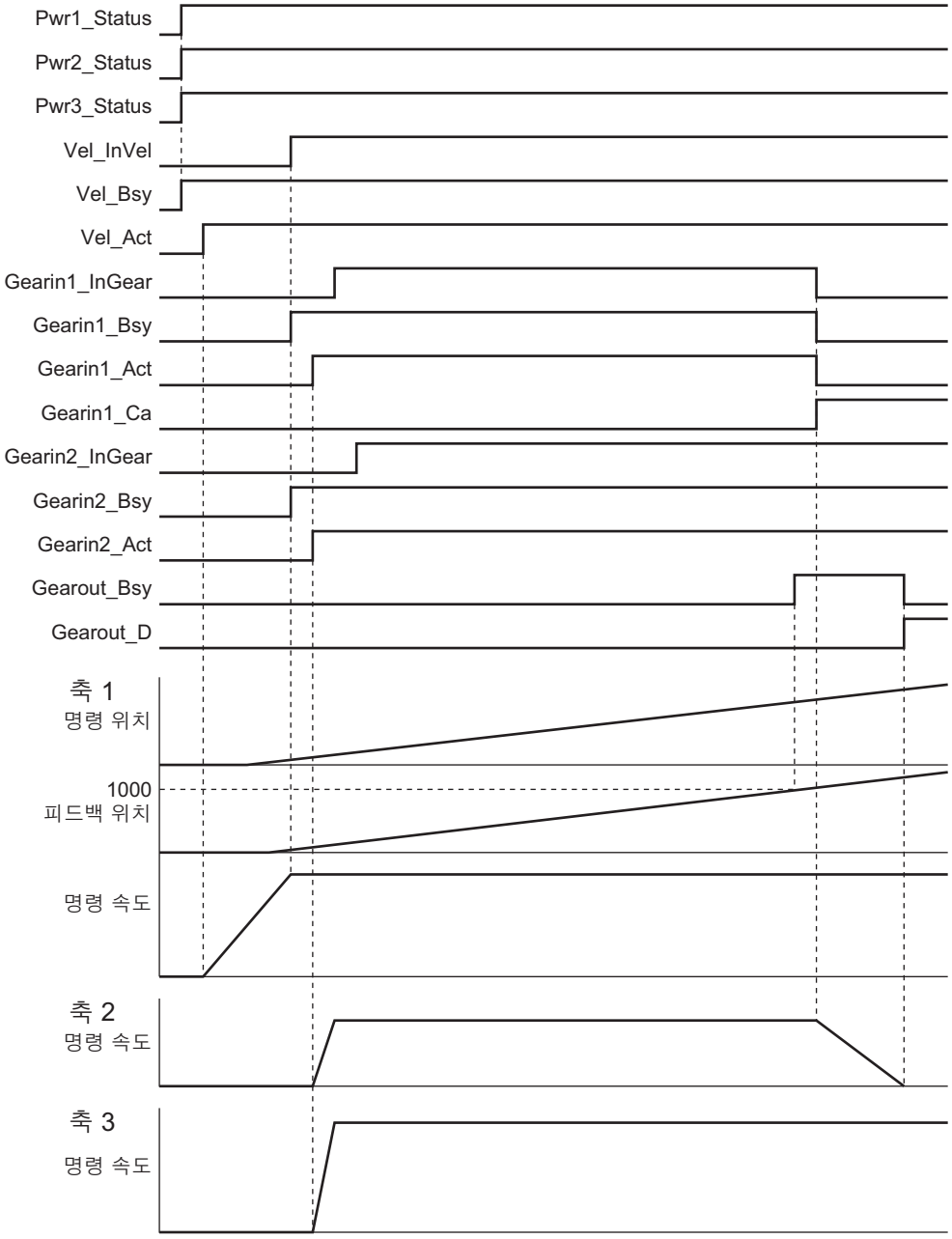
래더도

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis000.Act.Pos	LREAL	–	축 1의 피드백 현재 위치를 표시합니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis002	_sAXIS_REF	–	종축인 축 3의 축 변수입니다.
Pwr1_Status	BOOL	FALSE	MC_Power 인스턴스 PWR1의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 인스턴스 PWR2의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr3_Status	BOOL	FALSE	MC_Power 인스턴스 PWR3의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Vel_InVel	BOOL	FALSE	MC_MoveVelocity 인스턴스 VEL의 출력 InVelocity에 할당한 변수입니다. 목표 속도에 도달하면 이 변수가 TRUE가 됩니다.
Gearin1_Act	BOOL	FALSE	MC_GearIn 인스턴스 GEARIN1의 출력 Active에 할당한 변수입니다. GEARIN1이 제어 중이 되면 이 변수가 TRUE가 됩니다.

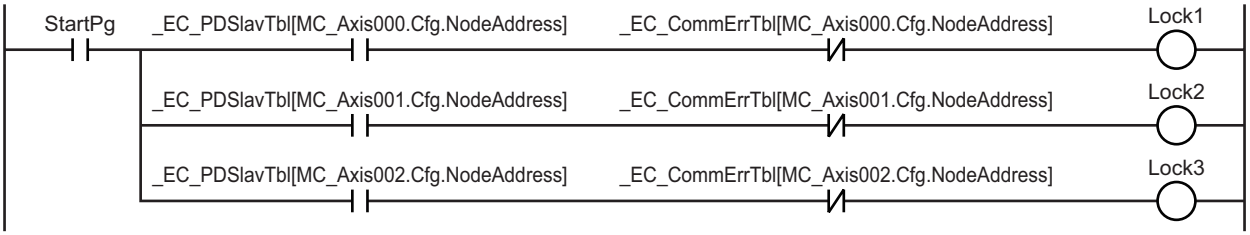
이 름	데이터 형식	초기값	코멘트
Gearout_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_GearOut의 인스턴스 GEAROUT을 기동합니다.

● 타이밍 차트

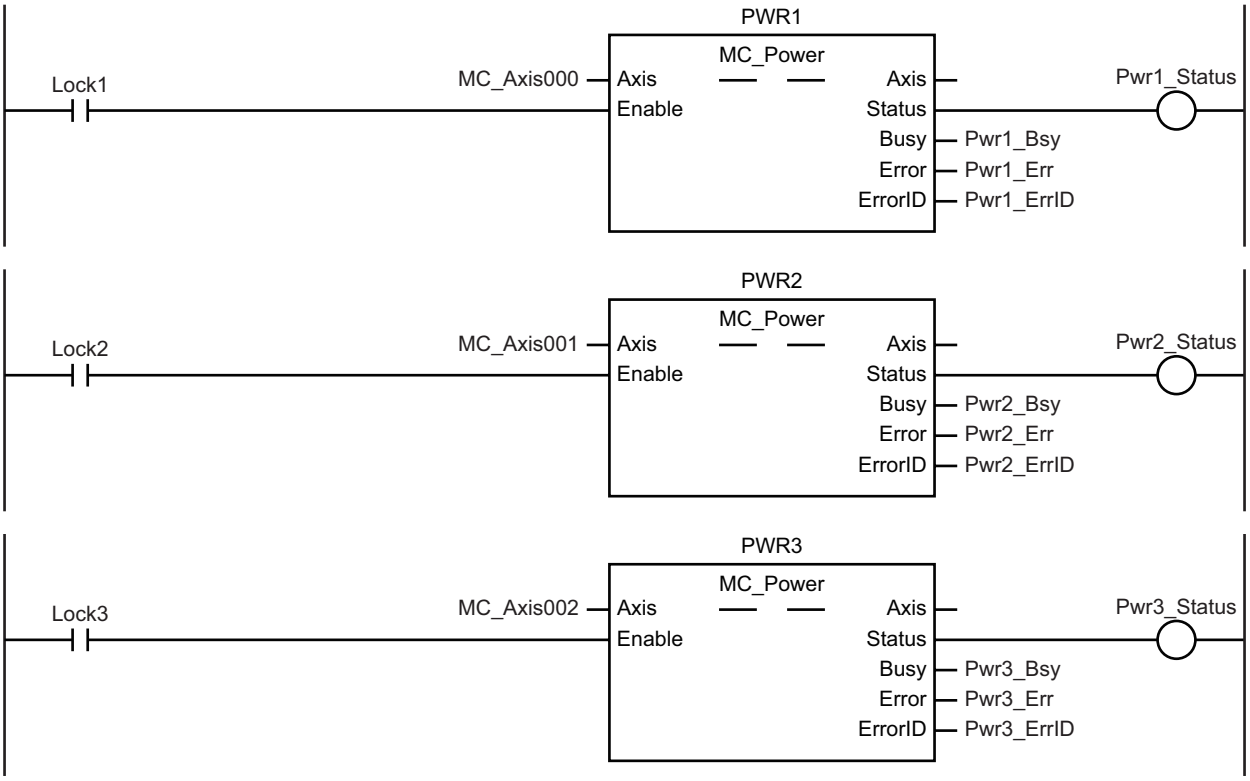


● 샘플 프로그램

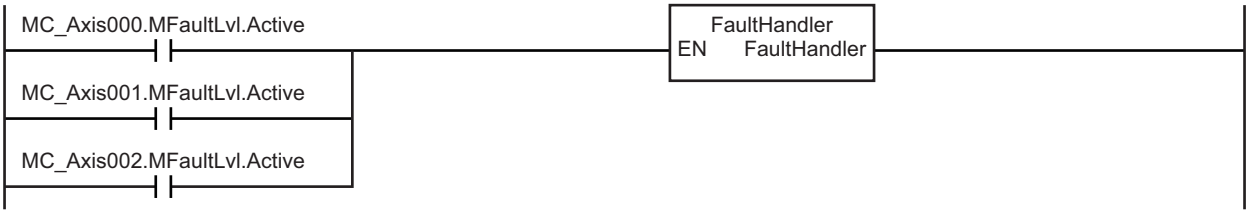
접점 StartPg가 TRUE가 되면 각 축의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인



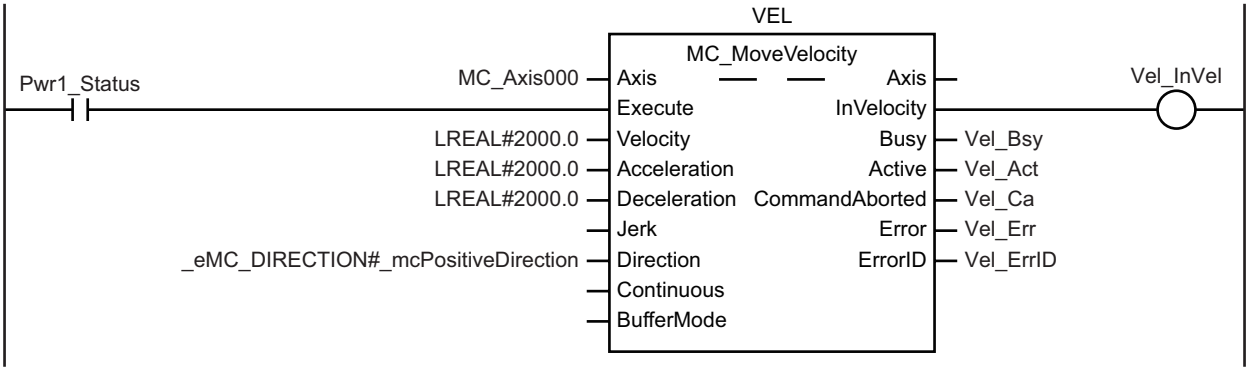
프로세스 데이터 통신을 하고 있으면 각 축의 서보를 ON



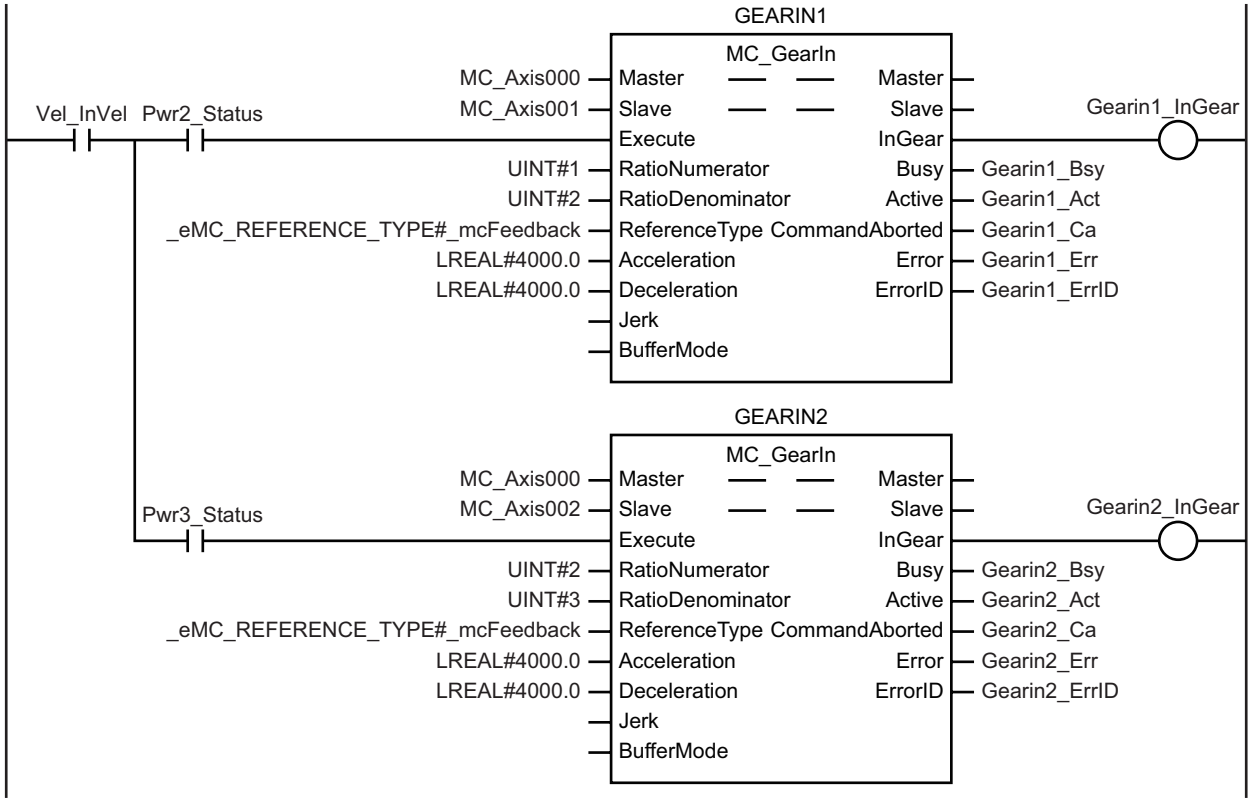
축 구성에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



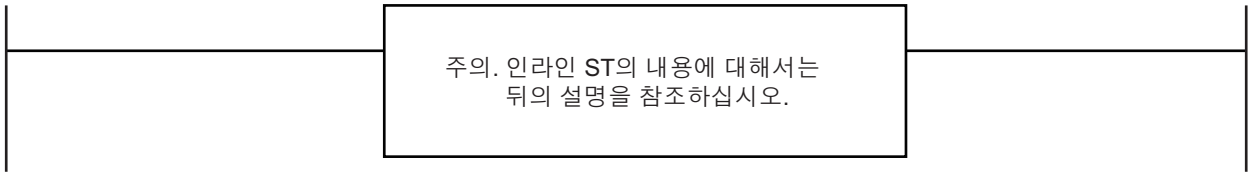
주축(축 1)이 운전 가능 상태가 되면 MC_MoveVelocity(속도 제어) 명령을 기동



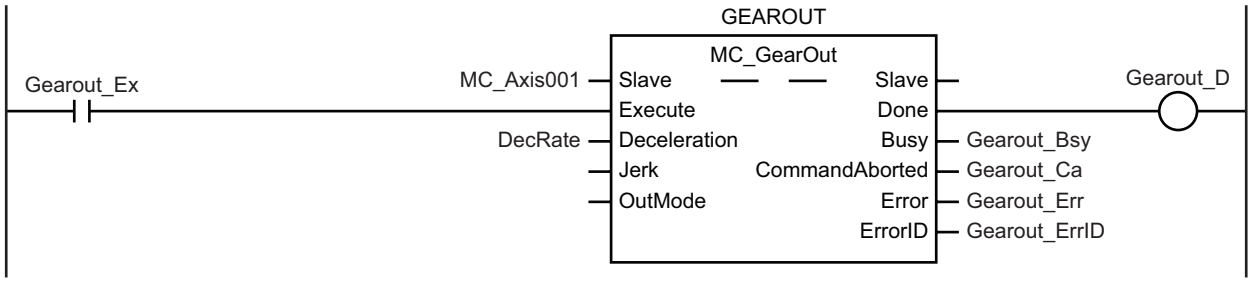
주축(축 1)의 피드백 속도가 목표 속도에 도달했을 때 종축(축 2)을 기어비 1:2로, 종축(축 3)을 기어비 2:3으로 기어 동작을 시작



종축(축 2)이 기어 동작 중이고 주축의 피드백 위치가 1000.0 이상일 때 Gearout_Ex를 TRUE로 함



Gearout_Ex가 TRUE가 되면 종축(축 2)의 기어 동작을 해제하고 감속 정지



인라인 ST의 내용

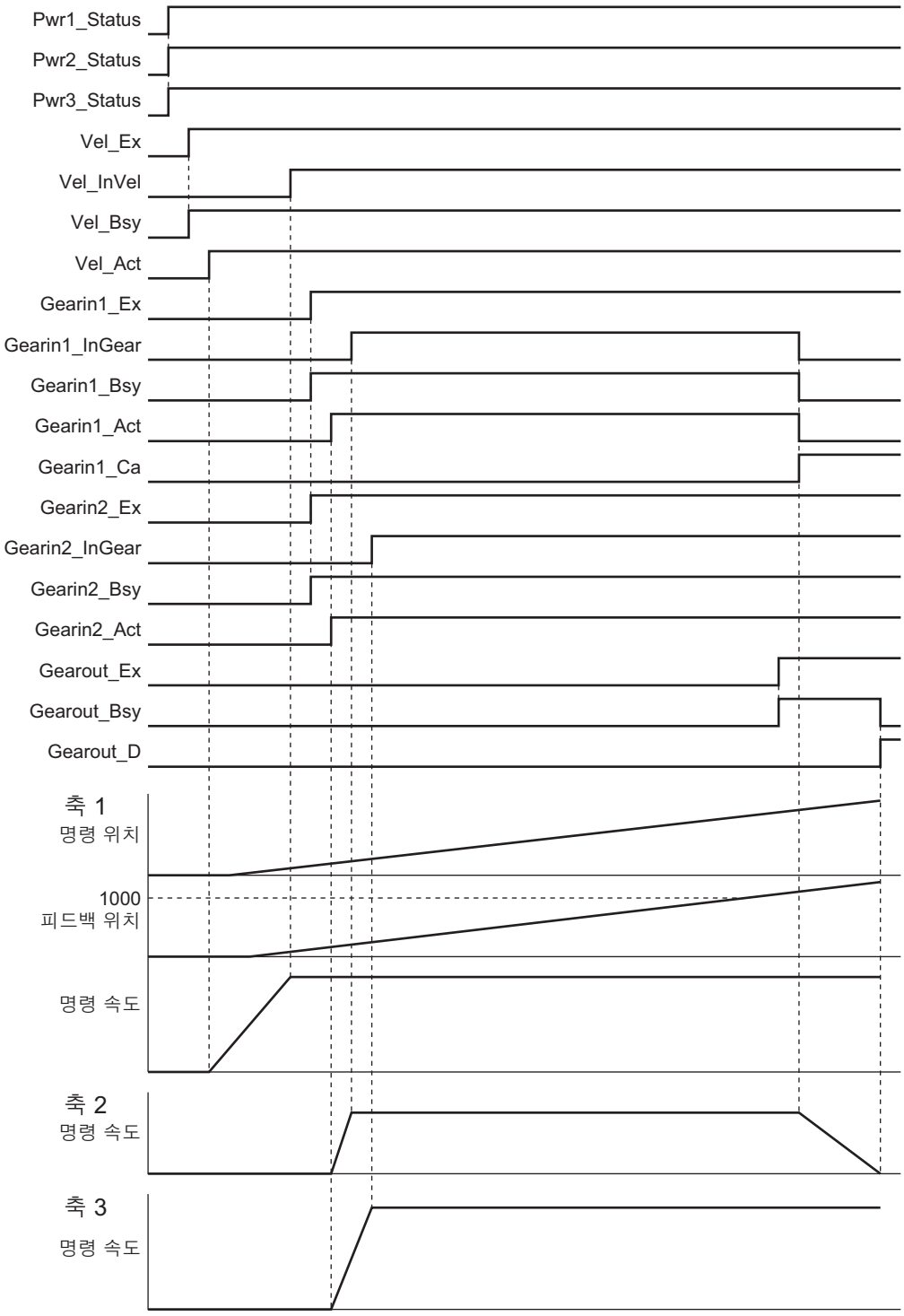
```
IF (Gearin1_Act=TRUE) AND (MC_Axis000.Act.Pos>=LREAL#1000.0) THEN
    Gearout_Ex := TRUE;
END_IF;
```

스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis000.Act.Pos	LREAL	–	축 1의 피드백 현재 위치를 표시합니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis002	_sAXIS_REF	–	종축인 축 3의 축 변수입니다.
Pwr1_Status	BOOL	FALSE	MC_Power 인스턴스 PWR1의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 인스턴스 PWR2의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr3_Status	BOOL	FALSE	MC_Power 인스턴스 PWR3의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Vel_InVel	BOOL	FALSE	MC_MoveVelocity 인스턴스 VEL의 출력 InVelocity에 할당한 변수입니다. 목표 속도에 도달하면 이 변수가 TRUE가 됩니다.
Gearin1_Act	BOOL	FALSE	MC_GearIn 인스턴스 GEARIN1의 출력 Active에 할당한 변수입니다. GEARIN1이 제어 중이 되면 이 변수가 TRUE가 됩니다.
Gearout_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_GearOut의 인스턴스 GEAROUT을 기동합니다.
Vel_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveVelocity의 인스턴스 VEL을 기동합니다.
Gearin1_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_GearIn의 인스턴스 GEARIN1을 기동합니다.
Gearin2_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_GearIn의 인스턴스 GEARIN2를 기동합니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트


Pwr1_Status
Pwr2_Status
Pwr3_Status
Vel_Ex
Vel_InVel
Vel_Bsy
Vel_Act
Gearin1_Ex
Gearin1_InGear
Gearin1_Bsy
Gearin1_Act
Gearin1_Ca
Gearin2_Ex
Gearin2_InGear
Gearin2_Bsy
Gearin2_Act
Gearout_Ex
Gearout_Bsy
Gearout_D
축 1
명령 위치
1000
피드백 위치
명령 속도
축 2
명령 속도
축 3
명령 속도

● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag = FALSE THEN
    //MC_MoveVelocity 파라메터
    Vel_Vel    := LREAL#2000.0;
    Vel_Acc    := LREAL#2000.0;
    Vel_Dec    := LREAL#2000.0;
    Vel_Dir    := _eMC_DIRECTION#_mcPositiveDirection;

    //MC_GearIn1 파라메터
    Gearin1_Rn     := UINT#1;
    Gearin1_Rd     := UINT#2;
    Gearin1_Rt     := _eMC_REFERENCE_TYPE#_mcFeedback;
    Gearin1_Acc    := LREAL#4000.0;
    Gearin1_Dec    := LREAL#4000.0;

    //MC_GearIn2 파라메터
    Gearin2_Rn     := UINT#2;
    Gearin2_Rd     := UINT#3;
    Gearin2_Rt     := _eMC_REFERENCE_TYPE#_mcFeedback;
    Gearin2_Acc    := LREAL#4000.0;
    Gearin2_Dec    := LREAL#4000.0;

    //MC_GearOut 파라메터
    DecRate        := LREAL#200.0;
    Gearout_Dec    := DecRate;

    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 1을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 2를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 3을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis002.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis002.Cfg.NodeAddress]=FALSE) THEN
    Pwr3_En:=TRUE;
ELSE
    Pwr3_En:=FALSE;
END_IF;
```

```
//축 1~축 3에서 가벼운 이상이 발생하면 이상 시 처리 FaultHandler를 실행합니다.
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) OR
(MC_Axis002.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

//축 1이 서보 ON 상태일 때 MC_MoveVelocity를 기동합니다.
IF Pwr1_Status=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

// MC_MoveVelocity의 InVelocity가 TRUE이고 축 2가 서보 ON 상태일 때 주축을 축 1, 종축을 축 2로
설정한 MC_GearIn을 기동합니다.

IF (Vel_InVel=TRUE) AND (Pwr2_Status=TRUE) THEN
    Gearin1_Ex := TRUE;
END_IF;

// MC_MoveVelocity의 InVelocity가 TRUE이고 축 3이 서보 ON 상태일 때 주축을 축 1, 종축을 축 3으
로 설정한 MC_GearIn을 기동합니다.
IF (Vel_InVel=TRUE) AND (Pwr3_Status=TRUE) THEN
    Gearin2_Ex := TRUE;
END_IF;

// 축 2가 기어 동작 중이고 축 1의 피드백 위치가 1000.0 이상일 때
// 종축을 축 2에 설정하고 GearOut을 기동합니다.
IF (Gearin1_Act=TRUE) AND (MC_Axis000.Act.Pos>=LREAL#1000.0) THEN
    Gearout_Ex := TRUE;
END_IF;


//축 1의 MC_Power
PWR1(
    Axis        := MC_Axis000,
    Enable      := Pwr1_En,
    Status      => Pwr1_Status,
    Busy        => Pwr1_Bsy,
    Error       => Pwr1_Err,
    ErrorID     => Pwr1_ErrID
);

//축 2의 MC_Power
PWR2(
    Axis        := MC_Axis001,
    Enable      := Pwr2_En,
    Status      => Pwr2_Status,
    Busy        => Pwr2_Bsy,
    Error       => Pwr2_Err,
    ErrorID     => Pwr2_ErrID
);

//축 3의 MC_Power
PWR3(
    Axis        := MC_Axis002,
    Enable      := Pwr3_En,
    Status      => Pwr3_Status,
    Busy        => Pwr3_Bsy,
    Error       => Pwr3_Err,
    ErrorID     => Pwr3_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
);

//축 1을 주축, 축 2를 종축으로 한 MC_GearIn
GEARIN1(
    Master              := MC_Axis000,
    Slave               := MC_Axis001,
    Execute             := Gearin1_Ex,
    RatioNumerator      := Gearin1_Rn,
    RatioDenominator    := Gearin1_Rd,
    ReferenceType       := Gearin1_Rt,
    Acceleration        := Gearin1_Acc,
    Deceleration        := Gearin1_Dec,
    InGear              => Gearin1_InGear,
    Busy                => Camin1_Bsy,
    Active              => Gearin1_Act,
    CommandAborted      => Gearin1_CA,
    Error               => Gearin1_Err,
    ErrorID             => Gearin1_ErrID
);


//축 1을 주축, 축 3을 종축으로 한 MC_GearIn
GEARIN2(
    Master              := MC_Axis000,
    Slave               := MC_Axis002,
    Execute             := Gearin2_Ex,
    RatioNumerator      := Gearin2_Rn,
    RatioDenominator    := Gearin2_Rd,
    ReferenceType       := Gearin2_Rt,
    Acceleration        := Gearin2_Acc,
    Deceleration        := Gearin2_Dec,
    InGear              => Gearin2_InGear,
    Busy                => Gearin2_Bsy,
    Active              => Gearin2_Act,
    CommandAborted      => Gearin2_CA,
    Error               => Gearin2_Err,
    ErrorID             => Gearin2_ErrID
);

//MC_GearOut
GEAROUT(
    Slave               := MC_Axis001,
    Execute             := Gearout_Ex,
    Deceleration        := Gearout_Dec,
    Done                => Gearout_D,
    Busy                => Gearout_Bsy,
    CommandAborted      => Gearout_CA,
    Error               => Gearout_Err,
    ErrorID             => Gearout_ErrID
);
```

MC_GearInPos

주축과 종축 사이의 기어비를 설정하고 전자 기어 동작을 실행합니다.
동기를 시작할 주축 및 종축 위치를 지정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_GearInPOS	위치 지정 기어 동작	FB	MC_GearInPos_instance MC_GearInPos Master — — Master Slave — — Slave Execute — StartSync RatioNumerator — InSync RatioDenominator — Busy ReferenceType — Active MasterSyncPosition — CommandAborted SlaveSyncPosition — Error Velocity — ErrorID Acceleration Deceleration Jerk BufferMode	MC_GearInPos_instance (Master := 《파라메터》, Slave := 《파라메터》, Execute := 《파라메터》, RatioNumerator := 《파라메터》, RatioDenominator := 《파라메터》, ReferenceType := 《파라메터》, MasterSyncPosition := 《파라메터》, SlaveSyncPosition := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, BufferMode := 《파라메터》, StartSync => 《파라메터》, InSync => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Ratio Numerator	기어비 분자	DINT *1	양수 또는 음수 *1	10000	주축과 종축 사이의 전자 기어 분자를 지정합니다.
Ratio Denominator	기어비 분모	UDINT *2	양수	10000	주축과 종축 사이의 전자 기어 분모를 지정합니다.
Reference Type*3	위치 종류 선택	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *4	위치 종류를 지정합니다. 0 : 명령 위치(직전 프라이머리 주기에서의 계산값) 1 : 피드백 위치(동일 프라이머리 주기에서의 취득값) 2 : 명령 위치(동일 프라이머리 주기에서의 계산값)
MasterSync Position	주축 동기 시작 위치	LREAL	음수, 양수, '0'	0	절대 좌표의 주축 동기 위치를 지정합니다. 단위는 [명령 단위]입니다. *5
SlaveSync Position	종축 동기 시작 위치	LREAL	음수, 양수, '0'	0	절대 좌표의 종축 동기 위치를 지정합니다. 단위는 [명령 단위]입니다. *3

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Velocity	목표 속도	LREAL	양수	0	목표 속도를 지정합니다. 반드시 목표 속도를 설정하십시오. 설정하지 않고 동작하게 하면 이상이 발생합니다. 단위는 [명령 단위/s]입니다. *3
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *3
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *3
Jerk (Reserved)	저크	LREAL	0	0	(Reserved)
BufferMode (Reserved)	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting	0 *2	(Reserved)

*1. CPU 유니트 Ver1.02 이상과 Sysmac StudioVer1.03 이상의 조합에서 사용할 수 있습니다. 그 이전의 조합에서는 데이터 형식은 'UINT', 유효 범위 '양수'가 됩니다.
*2. CPU 유니트 Ver1.02 이상과 Sysmac StudioVer1.03 이상의 조합에서 사용할 수 있습니다. 그 이전의 조합에서는 데이터 형식은 'UINT'가 됩니다.
*3. _mcLatestCommand를 사용하는 경우는
Master(주축)에 설정된 모션 제어 시스템 변수의 축 번호 < Slave(종축)에 설정된 모션 제어 시스템 변수의 축 번호가 되도록 주축과 종축을 사용해야 합니다.
*4. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*5. 명령 단위에 대해서는 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
StartSync	추종 중	BOOL	TRUE, FALSE	동기를 위한 가감속 동작을 시작했을 때 TRUE가 됩니다.
InSync	동기 중	BOOL	TRUE, FALSE	종축이 종축 동기 위치에 도달했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 🕮 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
StartSync	축이 이동을 시작했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
InSync	종축의 위치가 SlaveSyncPosition에 도달했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Busy	Execute의 기동	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 수신했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동 (Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때 • MC_GearOut 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Master	주축	_sAXIS_REF	−	주축을 지정합니다. *1
Slave	종축	_sAXIS_REF	−	종축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

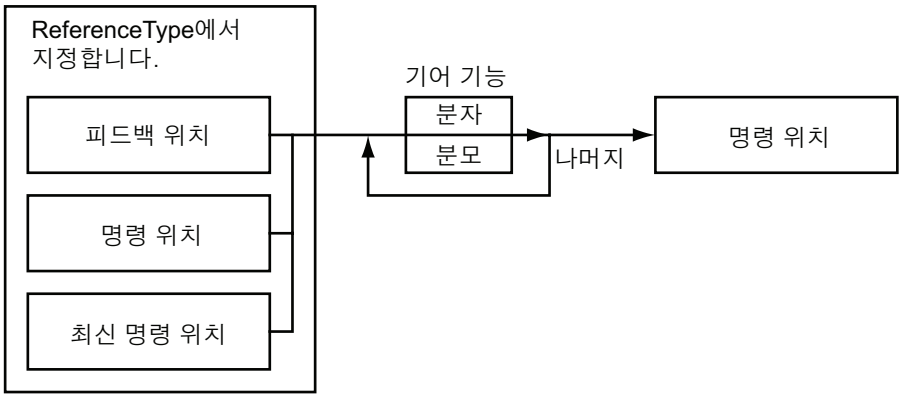


사용상의 주의 사항

주축과 종축에 같은 축을 지정하면 가벼운 이상 수준의 '주축 종축 동일(에러 코드: 5436 Hex)' 이상이 됩니다.

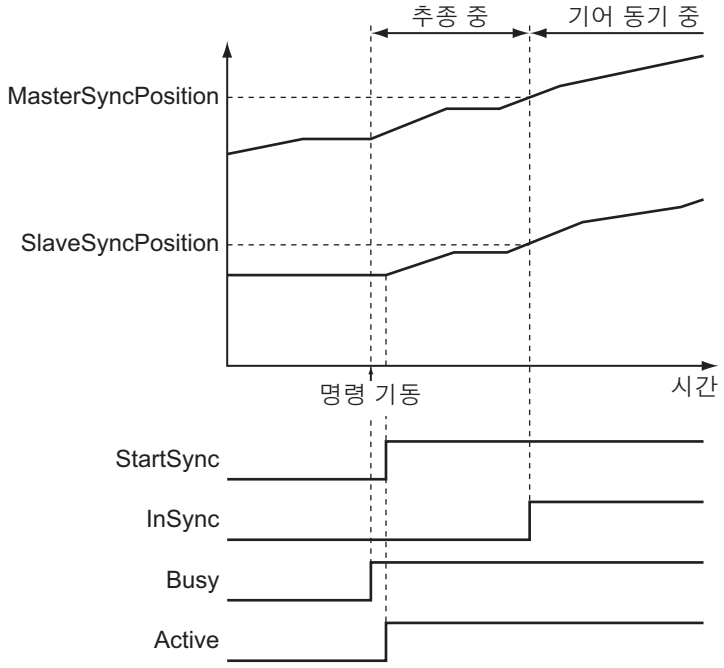
기능 설명

- 동작 대상 축을 Slave(종축)에서 지정하고 RatioNumerator(기어비 분자), RatioDenominator(기어비 분모), ReferenceType(위치 종류 선택), Acceleration(가속도), Deceleration(감속도)을 지정해서 기어를 동작합니다.
- Master(주축)에는 명령 위치와 피드백 위치, 최신 명령 위치를 지정할 수 있습니다.

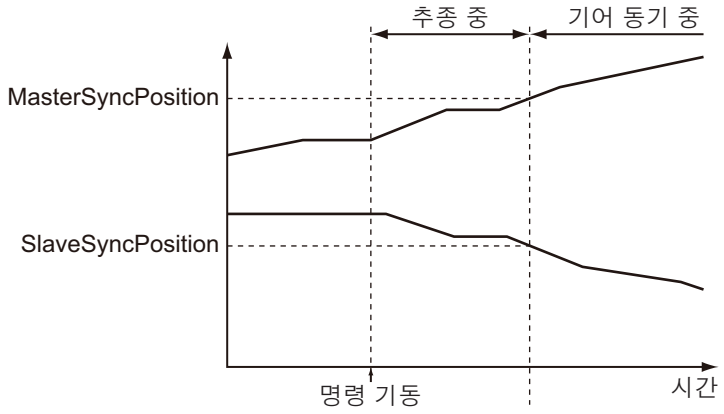


- 동작 시작 후 Slave(종축)는 추종 동작을 통해 Master(주축)에 동기하면서 가감속 동작을 합니다.
- 종축이 종축 동기 위치에 도달할 때까지를 Catching phase(추종 중)라고 하며, 도달 후를 InSync phase(기어 동기 중)라고 합니다. 어느 구간에서도 주축의 위치에 동기합니다.
- 입력 변수의 Velocity(목표 속도)는 Catching phase(추종 중) 시의 목표 속도가 됩니다.
- 종축은 동작 시작 시의 주축과 동일한 방향으로 동작합니다.
  동작 시작 시에 주축 속도가 '0'이었던 경우는 이상이 됩니다.
  또한 주축이 정방향으로 동작하고 SlaveSyncPosition(종축 동기 시작 위치)이 기동 시의 종축 위치보다 작은 경우에는 종축은 반전 동작을 합니다.
  마찬가지로 주축이 역방향으로 동작하고 SlaveSyncPosition(종축 동기 시작 위치)이 기동 시의 종축 위치보다 큰 경우에도 종축은 반전 동작을 합니다.
- 주축의 속도가 주기별로 크게 변동하는 경우 종축의 속도도 일정하지 않습니다.

- 기어비가 양수일 때는 Slave(종축)는 Master(주축)와 동일한 방향으로 이동합니다.



- 기어비가 음수일 때는 Slave(종축)는 Master(주축)의 반대 방향으로 이동합니다.

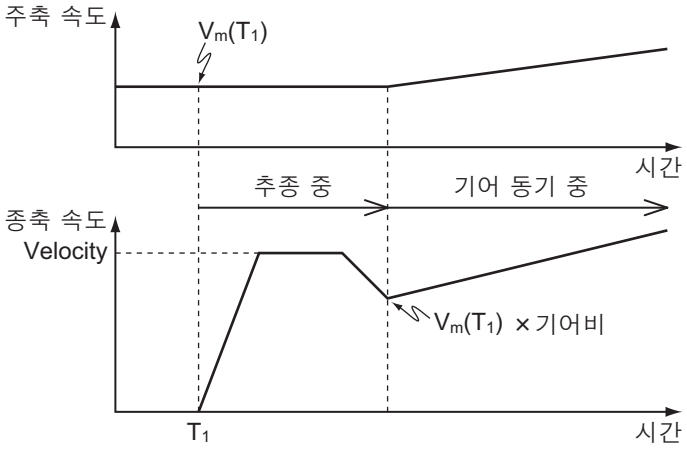


안전상의 중요 사항

이 명령을 사용하는 경우는 Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하지 마십시오.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하면 Slave(종축)가 급격하게 추종하는 경우가 있어 위험합니다.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 사용하려면 Master(주축)와 Slave(종축)의 관계를 일단 해제하고 나서 실행하십시오.

- MC 기능 모듈은 다음의 3가지 속도로 Acceleration(가속도) 및 Deceleration(감속도)을 이용해서 직선으로 가감속하는 속도 프로파일을 계산합니다.
  - 추종 동작 시작 시의 Slave(종축) 속도를 최초 속도
  - 추종 동작 시작 시의 Master(주축) 속도에 기어비를 곱한 속도를 최종 속도
  - Velocity(목표 속도)를 목표 속도

  추종 중의 이동량이 짧은 경우에는 목표 속도에 도달하지 않을 수도 있습니다.



MasterSyncPosition(주축 동기 시작 위치), SlaveSyncPosition(종축 동기 시작 위치)에서 Slave(종축)가 Master(주축)를 따라가기 위해서 Velocity(목표 속도)는 다음 조건을 만족해야 합니다.

$$\text{Velocity} > \frac{\text{MC\_GearInPos 기동 시의 주축 속도} \times \text{기어비 분자}}{\text{기어비 분모}}$$

주축 속도로 사용하는 정보는 위치 종류 ReferenceType의 설정에 따라 다릅니다.

_mcCommand, _mcLatestCommand 설정 시	: 명령 현재 속도를 사용합니다.
_mcFeedback 설정 시	: 피드백 현재 위치를 사용합니다.

MasterSyncPosition(주축 동기 시작 위치), SlaveSyncPosition(종축 동기 시작 위치)에서 Slave(종축)가 Master(주축)를 따라가지 못하면 '위치 지정 기어 동작 목표 속도 부족(에러 코드: 5447 Hex)'이 됩니다.
또한 카운트 모드가 로터리 모드이면 링 카운터가 1주기 이내에 동기 동작을 시작하도록 하십시오.

사용상의 주의 사항

주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

● ReferenceType(위치 종류 선택)

위치 종류를 다음에서 선택할 수 있습니다.

- _mcCommand: 명령 위치(직전 프라이머리 주기에서의 계산값)
  현재 주기에 대해 직전의 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  종축의 명령 위치를 계산하기 직전의 프라이머리 정주기 태스크에서 계산한 주축의 명령 위치를 사용합니다.
- _mcFeedback: 동일 프라이머리 주기에서의 취득값
  동일 프라이머리 주기에서 취득한 주축의 피드백 위치를 사용합니다.
- _mcLatestCommand: 명령 위치(동일 프라이머리 주기에서의 계산값)
  동일 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  _mcCommand보다도 새로운 정보를 사용할 수 있습니다. 단, 주축 번호는 종축 번호보다 빠른 번호를 설정해야 합니다.
  종축이 주축보다 축 번호가 빠르면 이 명령의 Error(에러)가 TRUE가 됩니다. ErrorID(에러 코드)에는 '주축/종축 축 번호 비오름차순(에러 코드: 5438 Hex)'이 출력됩니다.

참고

동일 프라이머리 주기에서 계산한 명령 위치를 선택하는 것이 직전 프라이머리 주기에서의 계산값을 선택하는 것보다 동기 정확도는 향상됩니다.
단, 'Master(주축)에 설정된 모션 제어 시스템 변수의 축 번호 < Slave(종축)에 설정된 모션 제어 시스템 변수의 축 번호'가 되도록 주축과 종축을 사용해야 합니다.

● 축 종류와 위치 종류의 관계

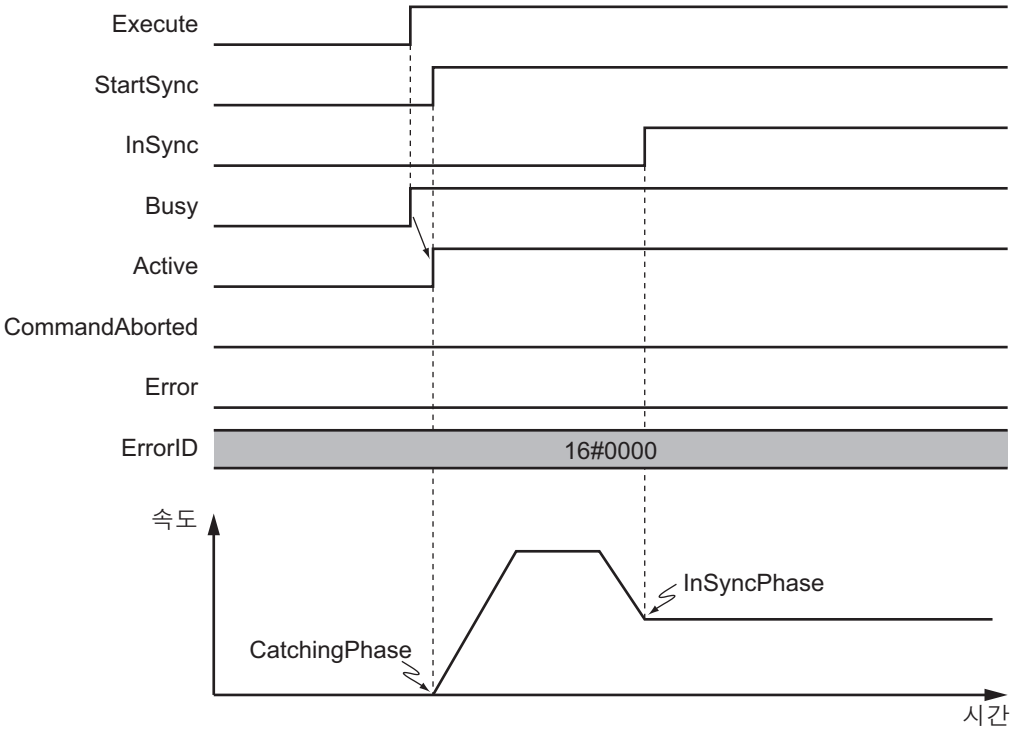
감시 가능한 축 종류와 감시하는 위치 종류의 관계는 다음과 같습니다.

축 종류	ReferenceType	
	_mcCommand, 또는 _mcLatestCommand	_mcFeedback
서보 축	○	○
엔코더 축	× *1	○
가상 서보 축	○	○
가상 엔코더 축	× *1	○

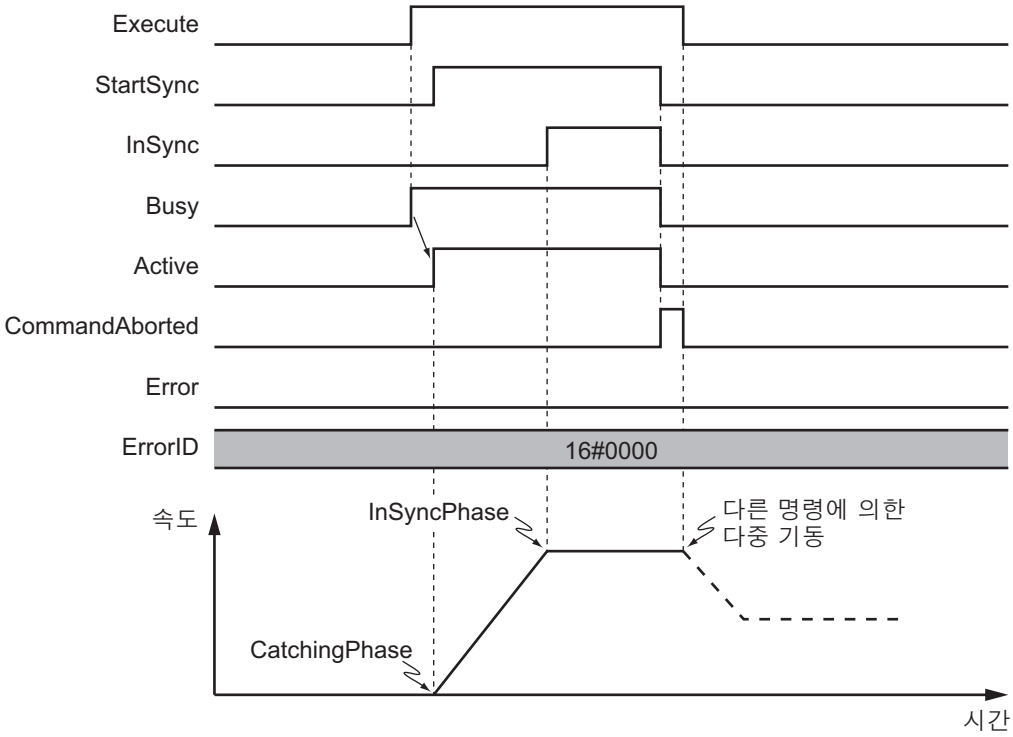
*1. 명령 기동 시에 '위치 종류 선택 범위 외(에러 코드: 5430 Hex)'가 발생합니다.

타이밍 차트

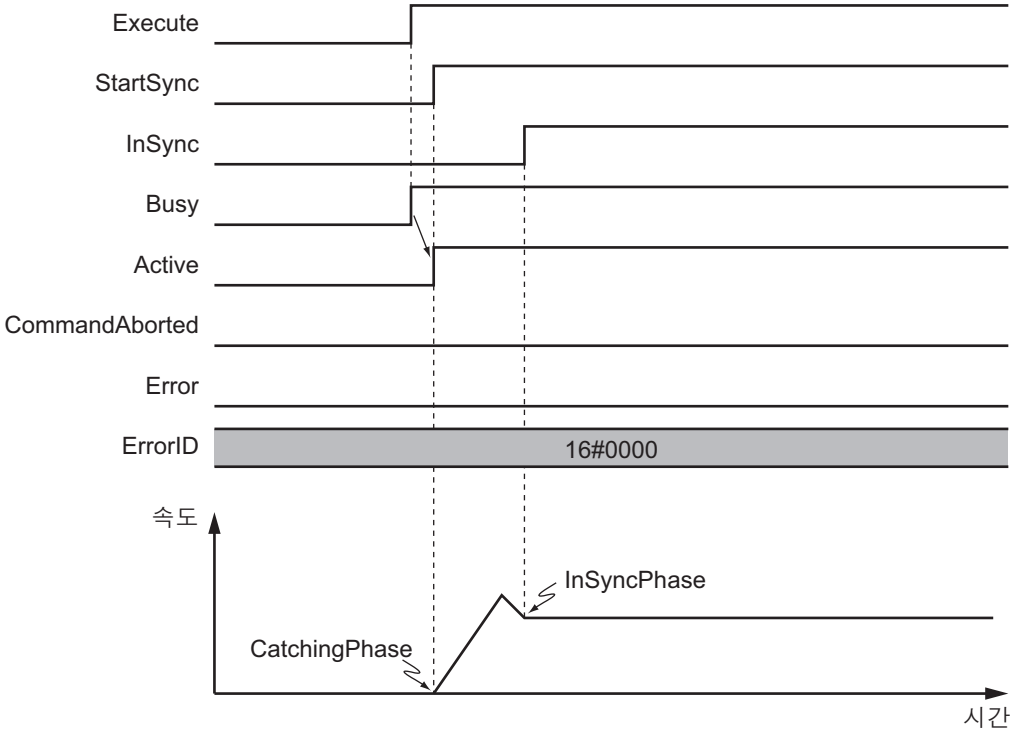
- Execute(기동)가 기동하면 전자 기어의 동작을 시작합니다.
- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 됩니다. 동작 시작 후에 Active(제어 중)와 StartSync(추종 중)가 TRUE가 되고 Slave(종축)가 기어 동작을 시작합니다.
- MasterSyncPosition(주축 동기 시작 위치), SlaveSyncPosition(종축 동기 시작 위치)에 도달하면 InSync (동기 중)가 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중단되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중), StartSync(추종 중), InSync(동기 중)가 FALSE가 됩니다.



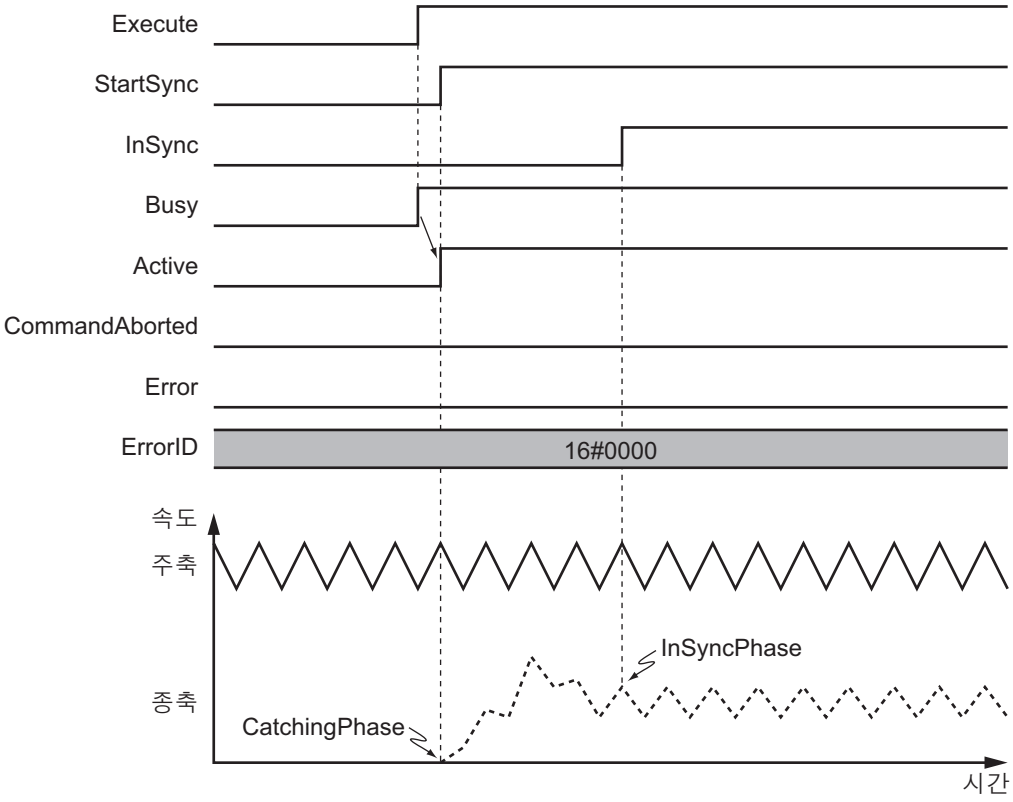
다른 명령에 의해 이 명령이 중단된 경우는 다음과 같습니다.



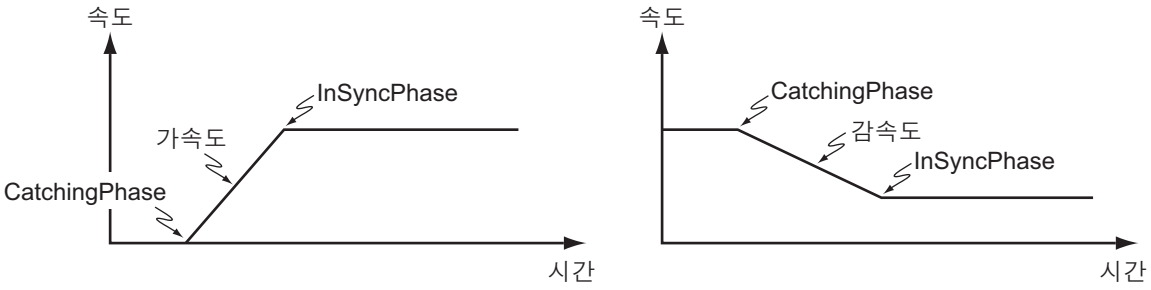
SlaveSyncPosition의 설정에 따라서는 목표 속도에 도달하지 않는 경우가 있습니다. 이 경우의 예는 다음과 같습니다.



또한 InSync phase 도달 전에도 주축 위치에 추종합니다. 이 경우의 예를 다음에 나타냅니다.



입력 변수에 Acceleration(가속도), Deceleration(감속도)을 지정할 수 있습니다. 전자 기어의 동작 예는 다음과 같습니다.



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

이상 정지 중, 감속 정지 중을 포함한 주축의 축 명령 상태는 이 명령 실행에 영향을 미치지 않습니다.
모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

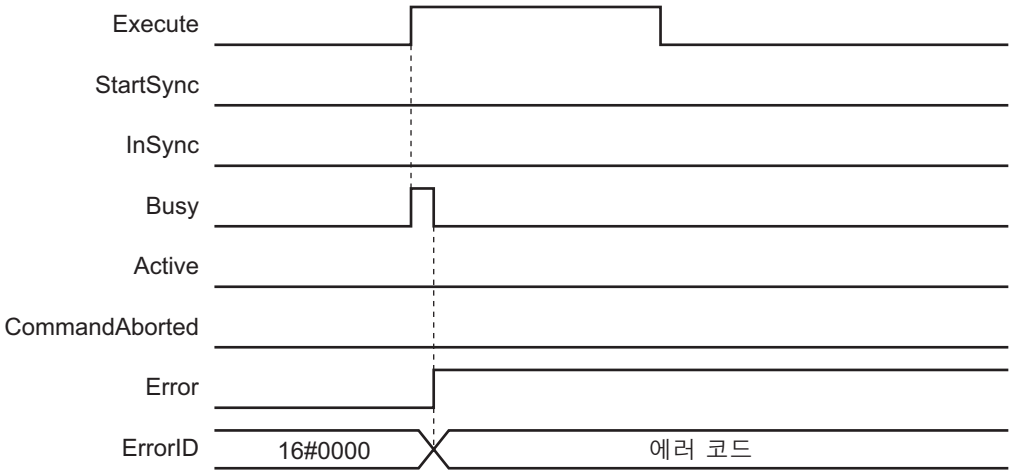
사용상의 주의 사항

다른 명령을 BufferMode(버퍼 모드 선택)의 '0: Aborting' 이외의 모드로 기동하면 다른 명령은 이상이 됩니다.

이상

이 명령 기동 시에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



동기 제어 중에는 주축의 이상 상태로부터 영향을 받지 않습니다. 주축의 이상 상태가 해제되고 동작할 때 종축은 전자 기어 동작을 계속합니다.
이 명령 기동 시 및 이 명령 실행 중에 종축에 이상이 발생해도 주축에 영향을 미치지 않습니다.

● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기에는 동기 위치를 주축은 Pos1, 종축은 Pos2로 동작하는 샘플 프로그램이 실려 있습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축(주축)
축 2	서보 축(종축)

카운터 모드

축	카운터 모드
축 1	로터리 모드
축 2	로터리 모드

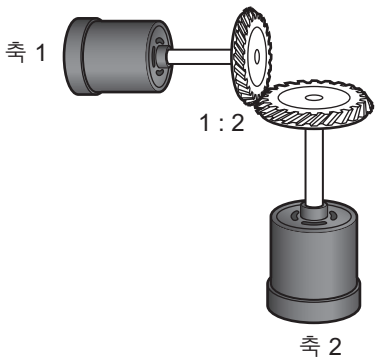
링 카운터

축	상한값	하한값
축 1	360	0
축 2	360	0

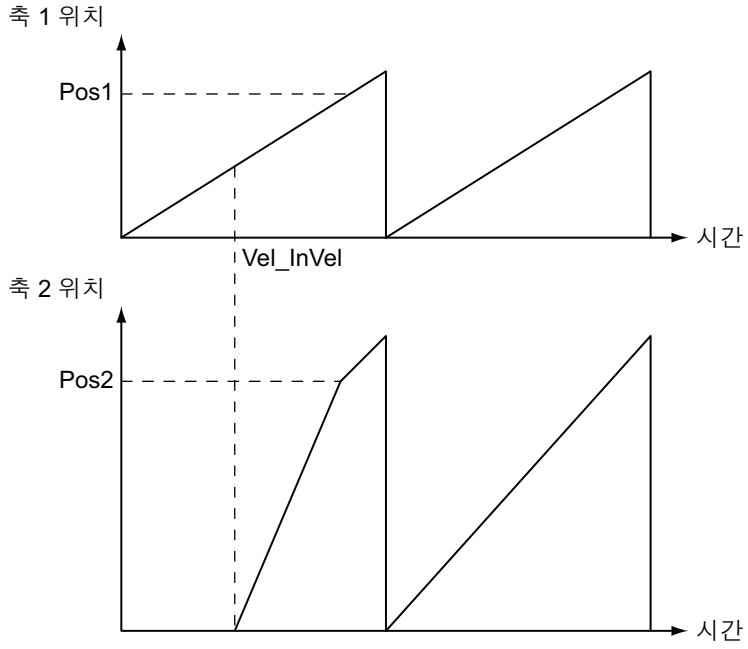
좌표 단위 선택

축	좌표 단위
축 1	degree
축 2	degree

동작 예



● 동작 패턴



1 주축의 시작

주축(축 1)은 실제 서보 축이며 속도 제어로 동작합니다.

2 주축의 목표 속도 도달

주축의 명령 속도가 목표 속도에 도달하면 주축의 InVelocity(목표 속도 도달)가 TRUE가 됩니다.

3 종축의 기동

주축의 InVelocity(목표 속도 도달)가 TRUE가 되면 주축의 피드백 위치에 대해 종축(축 2)을 기어비 1:2로 기어 동작시킵니다.

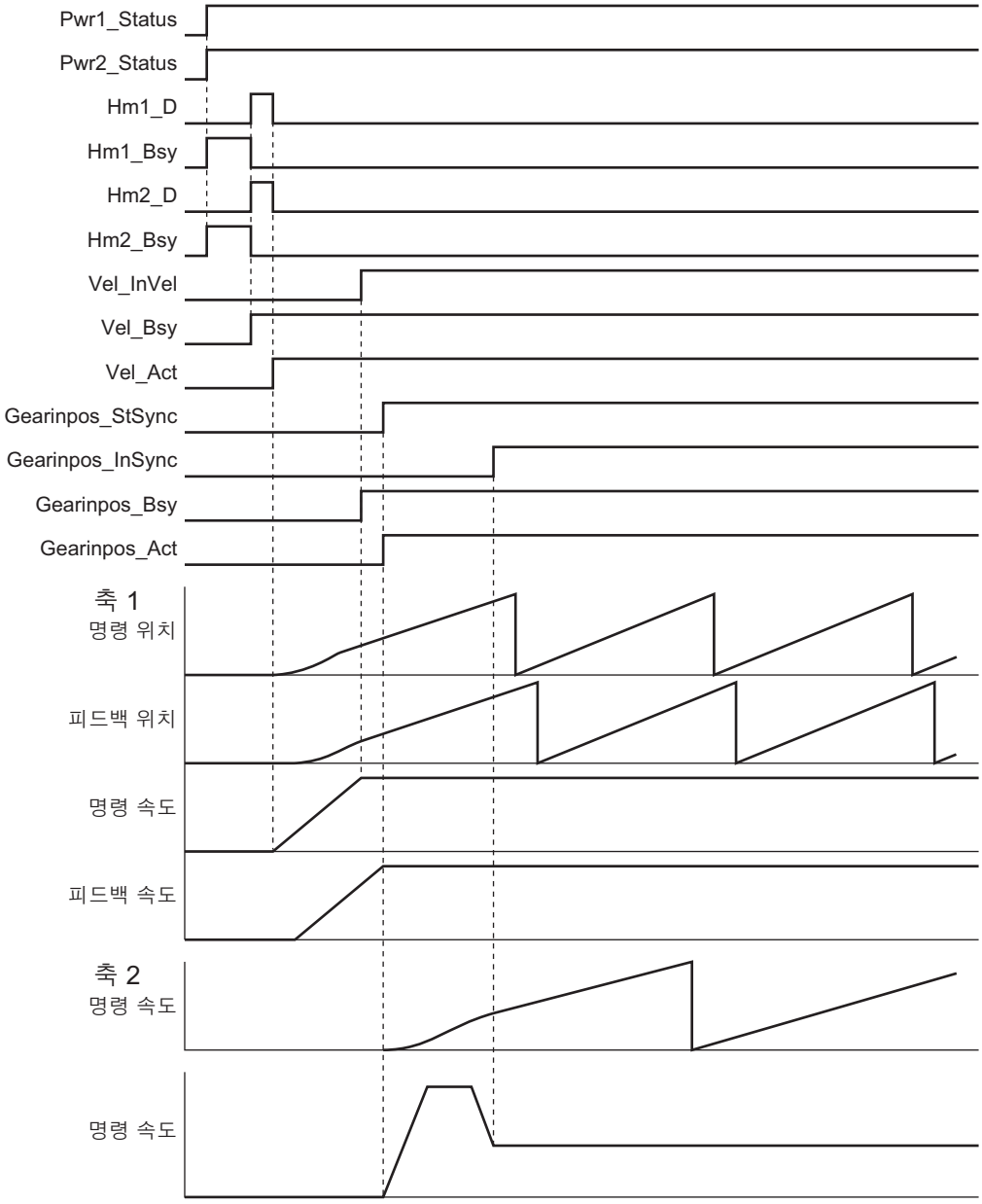
동기할 위치는 주축이 Pos1, 종축이 Pos2입니다.

래더도

● 주요 변수

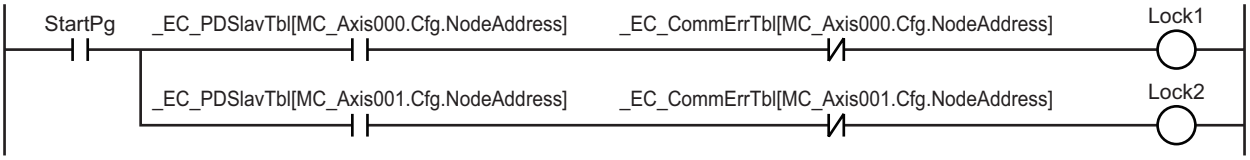
이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
Pwr1_Status	BOOL	FALSE	MC_Power 인스턴스 PWR1의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 인스턴스 PWR2의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Hm1_D	BOOL	FALSE	MC_Home 인스턴스 HM1의 출력 Done에 할당한 변수입니다.
Hm2_D	BOOL	FALSE	MC_Home의 인스턴스 HM2의 출력 Done에 할당한 변수입니다.
Vel_InVel	BOOL	FALSE	MC_MoveVelocity 인스턴스 VEL의 출력 InVelocity에 할당한 변수입니다. 목표 속도에 도달하면 이 변수가 TRUE가 됩니다.
Pos1	LREAL	–	주축의 동기 위치를 나타내는 변수입니다.
Pos2	LREAL	–	종축의 동기 위치를 나타내는 변수입니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.

● 타이밍 차트

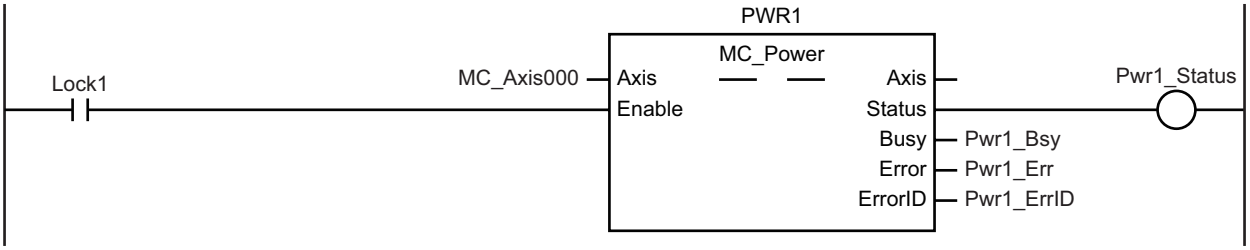

Pwr1_Status
Pwr2_Status
Hm1_D
Hm1_Bsy
Hm2_D
Hm2_Bsy
Vel_InVel
Vel_Bsy
Vel_Act
Gearinpos_StSync
Gearinpos_InSync
Gearinpos_Bsy
Gearinpos_Act
축 1
명령 위치
피드백 위치
명령 속도
피드백 속도
축 2
명령 속도
명령 속도

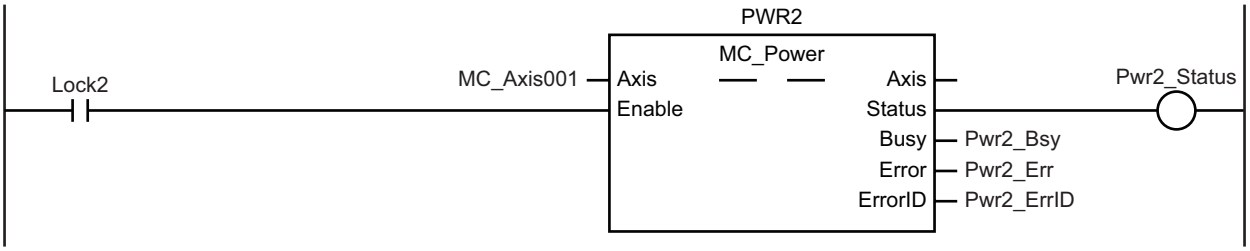
● 샘플 프로그램

접점 StartPg가 TRUE가 되면 각 축의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인

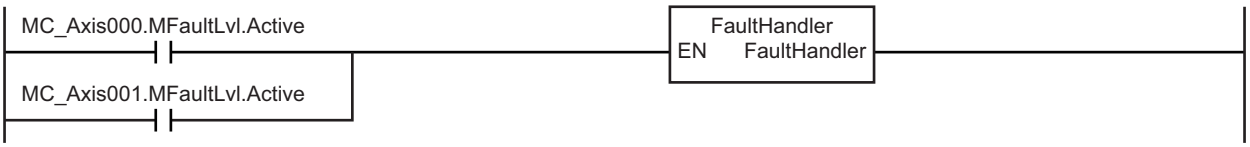


프로세스 데이터 통신을 하고 있으면 각 축의 서보를 ON

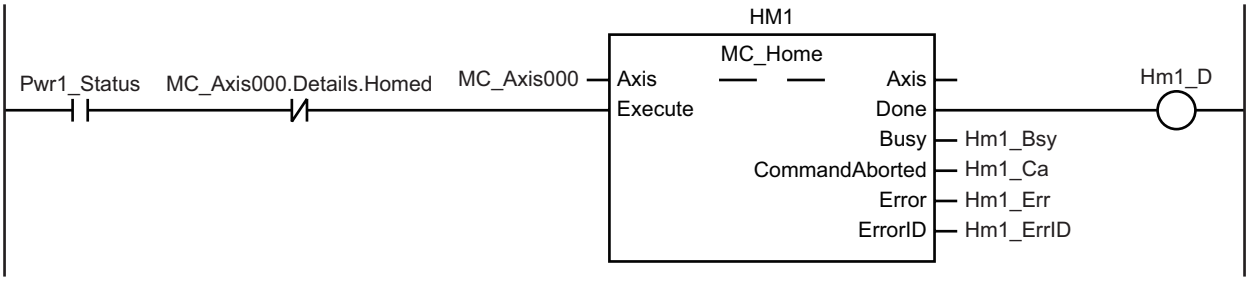




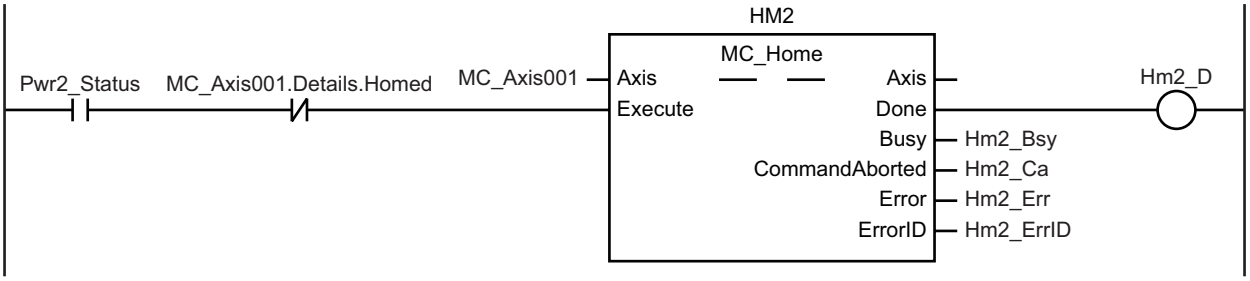
축 구성에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



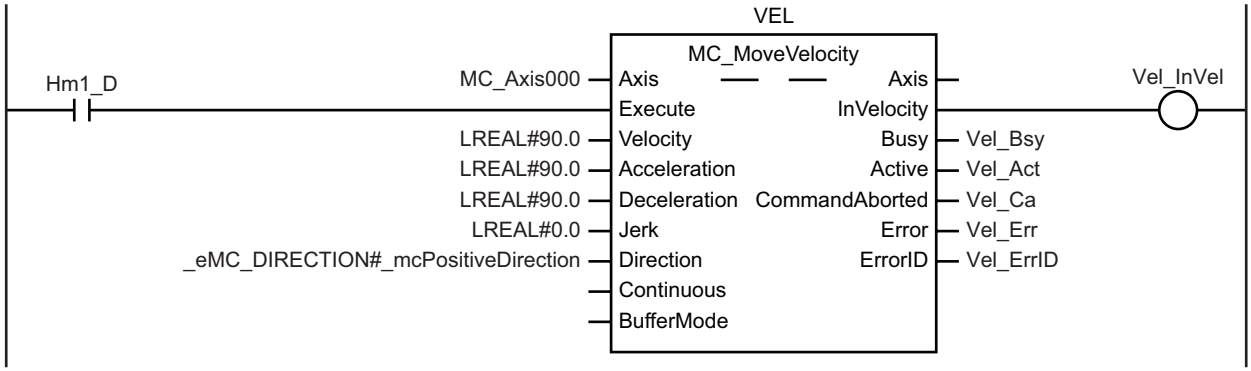
주축(축 1)이 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행하고 원점을 확정



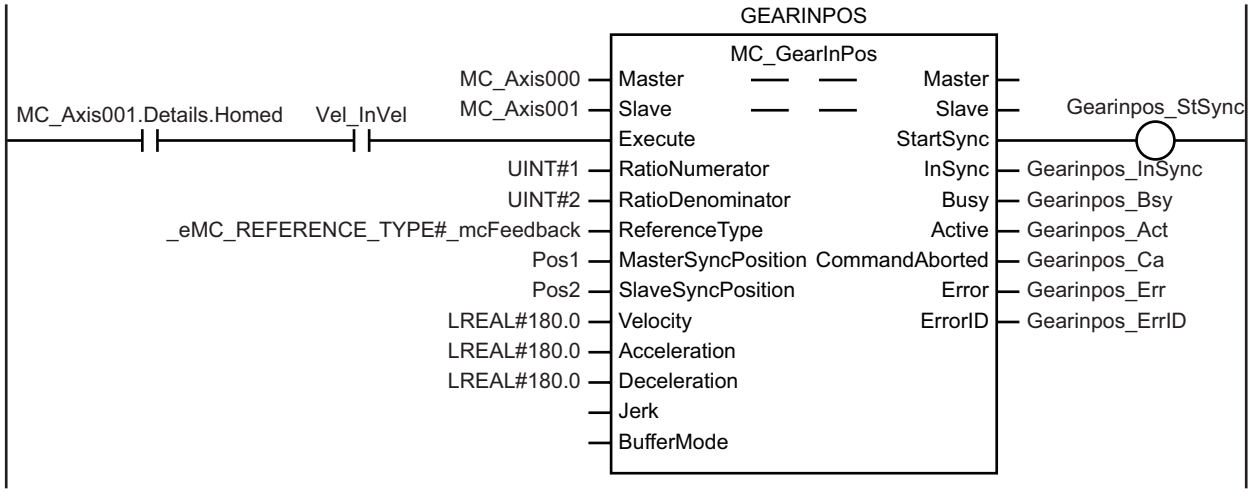
종축(축 2)이 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행하고 원점을 확정



주축(축 1)이 원점 복귀 완료 후 MC_MoveVelocity(속도 제어) 명령을 기동



종축(축 2)이 원점 복귀 완료 후 MC_MoveVelocity의 Vel_InVel이 TRUE이고 MC_GearInPos (위치 지정 기어 동작)를 기동하고 기어 동작을 시작

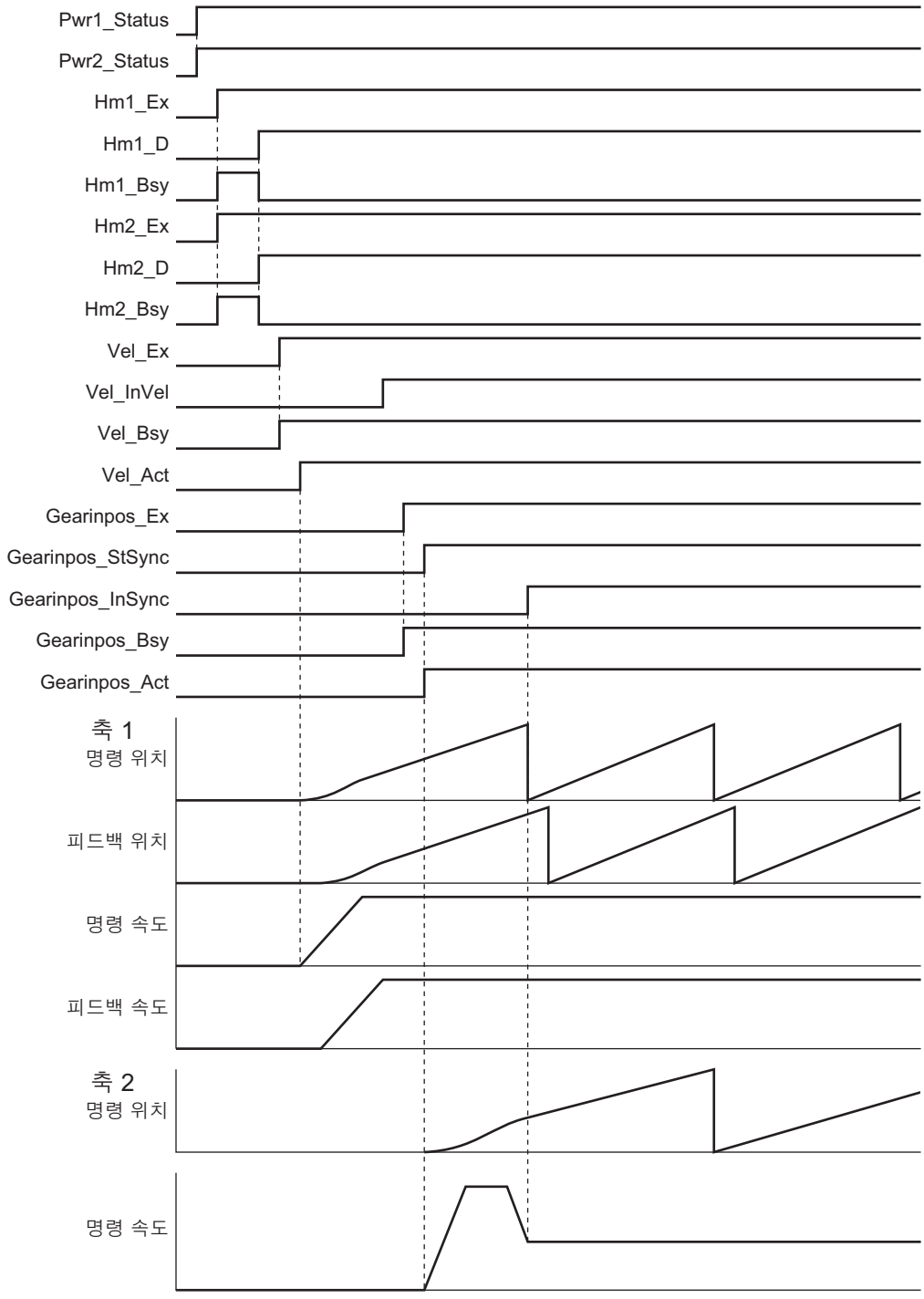


스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
Pwr1_Status	BOOL	FALSE	MC_Power 인스턴스 PWR1의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 인스턴스 PWR2의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Hm1_D	BOOL	FALSE	MC_Home 인스턴스 HM1의 출력 Done에 할당한 변수입니다.
Hm2_D	BOOL	FALSE	MC_Home의 인스턴스 HM2의 출력 Done에 할당한 변수입니다.
Vel_InVel	BOOL	FALSE	MC_MoveVelocity 인스턴스 VEL의 출력 InVelocity에 할당한 변수입니다. 목표 속도에 도달하면 이 변수가 TRUE가 됩니다.
Pos1	LREAL	–	주축의 동기 위치를 나타내는 변수입니다.
Pos2	LREAL	–	종축의 동기 위치를 나타내는 변수입니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Gearinpos_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE가 되었을 때 MC_GearInPos의 인스턴스 GEARIN1을 기동합니다.
Vel_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE가 되었을 때 MC_MoveVelocity의 인스턴스 VEL을 기동합니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트


Pwr1_Status
Pwr2_Status
Hm1_Ex
Hm1_D
Hm1_Bsy
Hm2_Ex
Hm2_D
Hm2_Bsy
Vel_Ex
Vel_InVel
Vel_Bsy
Vel_Act
Gearinpos_Ex
Gearinpos_StSync
Gearinpos_InSync
Gearinpos_Bsy
Gearinpos_Act
축 1
명령 위치
피드백 위치
명령 속도
피드백 속도
축 2
명령 위치
명령 속도

● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag = FALSE THEN
    //MC_MoveVelocity 파라메터
    Vel_Vel    := LREAL#90.0;
    Vel_Acc    := LREAL#90.0;
    Vel_Dec    := LREAL#90.0;
    Vel_Dir    := _eMC_DIRECTION#_mcPositiveDirection;

    // MC_GearInPos 파라메터
    Pos1               := LREAL#300.0;
    Pos2               := LREAL#200.0;
    Gearinpos_Rn       := UINT#1
    Gearinpos_Rd       := UINT#2
    Gearinpos_Rt       := _eMC_REFERENCE_TYPE#_mcFeedback;
    Gearinpos_Mtpos    := Pos1;
    Gearinpos_Svpos    := Pos2;
    Gearinpos_Vel      := LREAL#180.0;
    Gearinpos_Acc      := LREAL#180.0;
    Gearinpos_Dec      := LREAL#180.0;

    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 1을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 2를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
END_IF;


//축 1~축 2에서 가벼운 이상이 발생하면 이상 시 처리 FaultHandler를 실행합니다.
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

// 축 1이 운전 가능 상태이고 원점 미확정 상태일 때 원점 복귀를 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

// 축 2가 운전 가능 상태이고 원점 미확정 상태일 때 원점 복귀를 실행합니다.
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;
```

```
// 축 1이 원점 복귀 완료 후 MC_MoveVelocity을 기동합니다.
IF Hm1_D=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

// 축 2가 원점 복귀 완료 후 MC_MoveVelocity의 Vel_InVel이 TRUE이면 MC_GearInPos를 기동합
니다.
IF (MC_Axis001.Details.Homed=TRUE) AND (Vel_InVel=TRUE) THEN
    Gearinpos_Ex := TRUE;
END_IF;


//축 1의 MC_Power
PWR1(
    Axis        := MC_Axis000,
    Enable      := Pwr1_En,
    Status      => Pwr1_Status,
    Busy        => Pwr1_Bsy,
    Error       => Pwr1_Err,
    ErrorID     => Pwr1_ErrID
);
//축 2의 MC_Power
PWR2(
    Axis        := MC_Axis001,
    Enable      := Pwr2_En,
    Status      => Pwr2_Status,
    Busy        => Pwr2_Bsy,
    Error       => Pwr2_Err,
    ErrorID     => Pwr2_ErrID
);

//축 1의 MC_Home
HM1(
    Axis                := MC_Axis000,
    Execute             := Hm1_Ex,
    Done                => Hm_D,
    Busy                => Hm1_Bsy,
    CommandAborted      => Hm1_Ca,
    Error               => Hm1_Err,
    ErrorID             => Hm1_ErrID
);
//축 2의 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_GearInPos
GEARINPOS(
    Master                := MC_Axis000,
    Slave                 := MC_Axis001,
    Execute               := Gearinpos_Ex,
    RatioNumerator        := Gearinpos_Rn,
    RatioDenominator      := Gearinpos_Rd,
    ReferenceType         := Gearinpos_Rt,
    MasterSyncPosition    := Gearinpos_Mtpos,
    SlaveSyncPosition     := Gearinpos_Svpos,
    Velocity              := Gearinpos_Vel,
    Acceleration          := Gearinpos_Acc,
    Deceleration          := Gearinpos_Dec,
    StartSync             => Gearinpos_StSync,
    InSync                => Gearinpos_InSync,
    Busy                  => Gearinpos_Bsy,
    Active                => Gearinpos_Act,
    CommandAborted        => Gearinpos_Ca,
    Error                 => Gearinpos_Err,
    ErrorID               => Gearinpos_ErrID
);
```

MC_GearOut

실행 중인 MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령을 중지합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
MC_GearOut	기어 동작 해제	FB	MC_GearOut_instance MC_GearOut Slave — — Slave Execute — Done Deceleration — Busy Jerk — CommandAborted OutMode — Error ErrorID	MC_GearOut_instance (Slave := 《파라메터》, Execute := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, OutMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk (Reserved)	저크	LREAL	0	0	(Reserved)
OutMode (Reserved)	동기 해제 모드 선택	_eMC_OUT_MODE	0: _mcStop	0 *2	(Reserved)

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

*2. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Slave	종축	_sAXIS_REF	−	종축을 지정합니다. *1

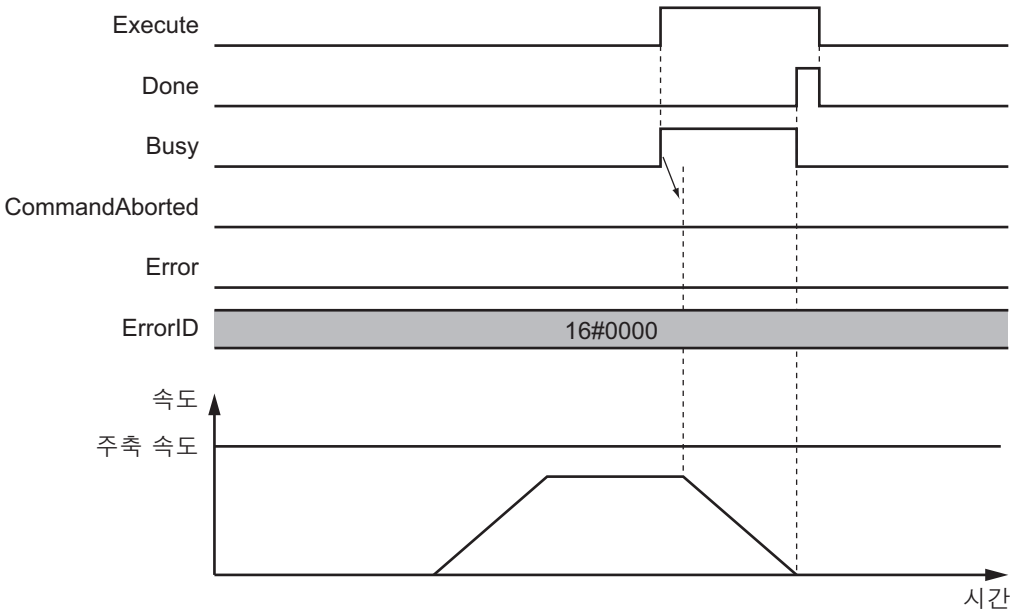
*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

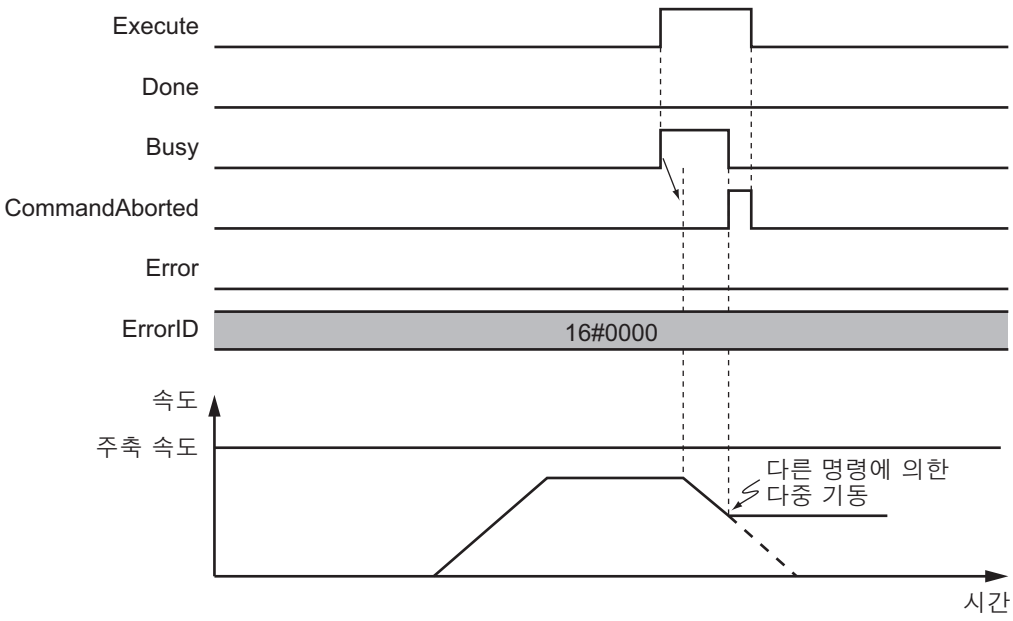
- 동작 대상 축을 Slave(종축)에서 지정하고 Deceleration(감속도)을 지정해 실행 중인 MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령을 중지합니다.
- 이 명령은 MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령의 주축 동작에 영향을 미치지 않습니다.

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 됩니다.
- 목표 속도에 도달하면 Done(완료)이 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중지되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중)가 FALSE가 됩니다.

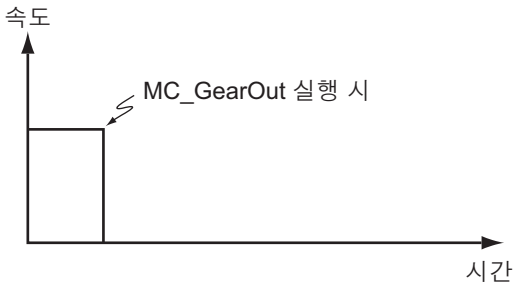


● 다른 명령에 의해 이 명령이 중단된 경우



● 감속도를 '0'으로 지정해서 기동한 경우

감속도를 '0'으로 지정해서 기동하면 감속하지 않고 정지합니다.
감속도가 '0'인 경우의 동작 예는 다음과 같습니다.



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

MC_GearIn(기어 동작 시작) 명령 또는 MC_GearInPos(위치 지정 기어 동작) 명령을 실행 중에 이 명령을 기동하면 MC_GearIn(기어 동작 시작) 명령 또는 MC_GearInPos(위치 지정 기어 동작) 명령을 CommandAborted(실행 중단)하고 이 명령은 Busy(실행 중)가 됩니다.
MC_GearIn(기어 동작 시작) 명령 또는 MC_GearInPos(위치 지정 기어 동작) 명령 이외의 명령을 실행 중에 이 명령을 기동하면 이 명령은 Error(에러)가 됩니다.

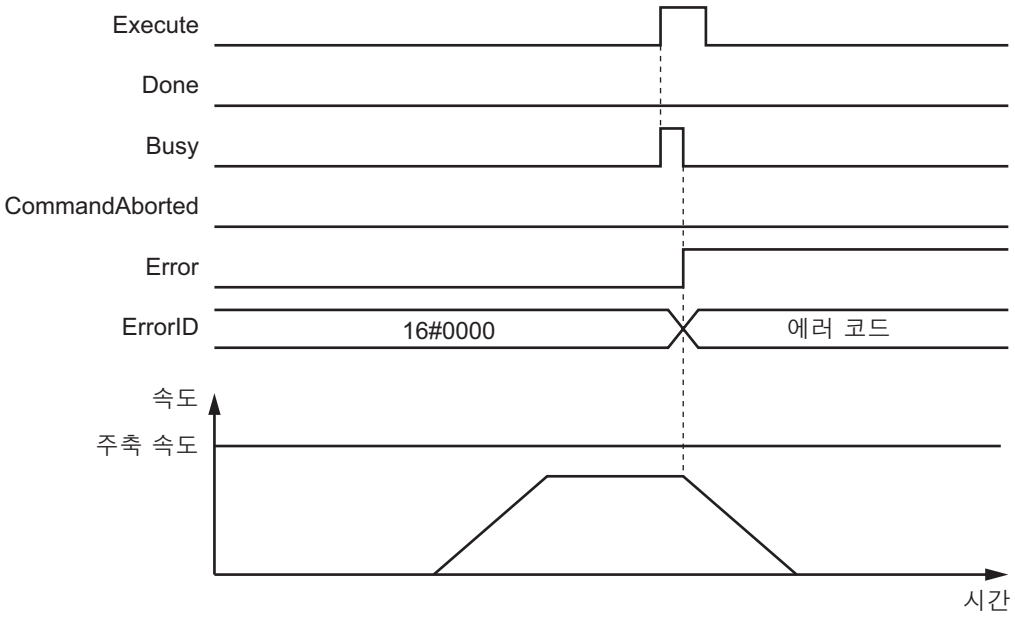
● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하려면 종축을 지정하십시오.
이 명령에 대한 모션 명령 다중 기동을 실행하는 경우 Aborting, Buffered가 Blending 가능합니다.

이상

이 명령 기동 시에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_MoveLink

지정한 주축에 동기해서 위치를 결정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_MoveLink	사다리꼴 패턴 캠	FB	MC_MoveLink_instance MC_MoveLink Master — — Master Slave — — Slave TriggerInput — — TriggerInput TriggerVariable — — TriggerVariable Execute — Done ReferenceType — InSync SlaveDistance — Busy MasterDistance — Active MasterDistanceInACC — CommandAborted MasterDistanceInDEC — Error LinkOption — ErrorID MasterStartDistance BufferMode	MC_MoveVelocity_instance (Master := 《파라메터》, Slave := 《파라메터》, TriggerInput := 《파라메터》, TriggerVariable := 《파라메터》, Execute := 《파라메터》, ReferenceType := 《파라메터》, SlaveDistance := 《파라메터》, MasterDistance := 《파라메터》, MasterDistanceInACC := 《파라메터》, MasterDistanceInDEC := 《파라메터》, LinkOption := 《파라메터》, MasterStartDistance := 《파라메터》, BufferMode := 《파라메터》, Done => 《파라메터》, InSync => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Reference Type *1	위치 종류 선택	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *2	위치 종류를 지정합니다. 0 : 명령 위치(직전 프라이머리 주기에서의 계산값) 1 : 피드백 위치(동일 프라이머리 주기에서의 취득값) 2 : 명령 위치(동일 프라이머리 주기에서의 계산값)
Slave Distance	종축 이동 거리	LREAL	음수, 양수, '0'	0	종축의 이동 거리를 지정합니다. 단위는 [명령 단위]입니다. *3
Master Distance	주축 이동 거리	LREAL	양수 또는 '0'	0	주축의 이동 거리를 부호가 없는 절대값 표현으로 지정합니다. 주축의 이동 방향은 정방향, 역방향 모두 유효합니다. 단위는 [명령 단위]입니다. *3

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Master DistanceIn ACC	주축 가속 이동 거리	LREAL	양수 또는 '0'	0	종축 가속 시 주축의 이동 거리를 부호가 없는 절대값 표현으로 지정합니다. 주축의 이동 방향은 정방향, 역방향 모두 유효합니다. 단위는 [명령 단위]입니다. *3
Master DistanceIn DEC	주축 감속 이동 거리	LREAL	양수 또는 '0'	0	종축 감속 시 주축의 이동 거리를 부호가 없는 절대값 표현으로 지정합니다. 주축의 이동 방향은 정방향, 역방향 모두 유효합니다. 단위는 [명령 단위]입니다. *3
LinkOption	동기 시작 조건 선택	_eMC_ LINKOPTION	0: _mcCommandExecution 1: _mcTriggerDetection 2: _mcMasterReach	0 *2	종축이 주축에 동기할 조건을 지정합니다. 0 : 기동 시 1 : 트리거 검출 시 2 : 주축이 주축 추종 거리에 도달 시
MasterStart Distance	주축 추종 거리	LREAL	음수, 양수, '0'	0	종축이 주축을 추종하기 시작하는 절대 좌표의 주축 위치를 지정합니다. 단위는 [명령 단위]입니다. *3
BufferMode	버퍼 모드 선택	_eMC_BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 *2	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered

*1. _mcLatestCommand를 사용하는 경우는
Master(주축)에 설정된 모션 제어 시스템 변수의 축 번호 < Slave(종축)에 설정된 모션 제어 시스템 변수의 축 번호
가 되도록 주축과 종축을 사용해야 합니다.
*2. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*3. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
InSync	동기 중	BOOL	TRUE, FALSE	동기 동작을 시작했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
InSync	동기 시작 조건이 성립했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 수신했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동 (Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Master	주축	_sAXIS_REF	–	주축을 지정합니다. *1
Slave	종축	_sAXIS_REF	–	종축을 지정합니다. *1
TriggerInput	트리거 입력 조건	_sTRIGGER_REF	–	트리거 조건을 설정합니다. *2
TriggerVariable	트리거 변수	BOOL	TRUE, FALSE	트리거 조건에서 컨트롤러 모드를 지정했을 때 트리거가 될 입력 변수를 지정합니다.

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.
*2. _sTRIGGER_REF 형식의 사용자 정의 변수를 정의하십시오.

● _sTRIGGER_REF

멤버 변수	이 름	데이터 형식	유효 범위	기 능
Mode	모드	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	트리거 모드를 지정합니다. 0 : 드라이버 모드 1 : 컨트롤러 모드
LatchID	래치 ID 선택	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	드라이버 모드일 때 2가지 래치 기능 중 어느 것을 사용할지 지정합니다. 0 : 래치 기능 1 1 : 래치 기능 2
InputDrive	드라이브 트리거 입력 신호	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	드라이버 모드일 때 서보 드라이버의 트리거 신호를 지정합니다. 0 : Z상 1 : 외부 입력

기능 설명

- 지정한 주축에 동기해서 종축이 동작합니다.
- 전자 캠의 일종인데, 주축에 대해 종축은 사다리꼴 곡선으로 동작합니다.
- 이 명령으로 동작 중인 축을 정지시키려면 MC_Stop(강제 정지) 명령을 사용하십시오.

안전상의 중요 사항

이 명령을 사용하는 경우는 Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하지 마십시오.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하면 Slave(종축)가 급격하게 추종하는 경우가 있어 위험합니다.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 사용하려면 Master(주축)와 Slave(종축)의 관계를 일단 해제하고 나서 실행하십시오.
주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

● 데이터 오브젝트 매핑

LinkOption(동기 시작 조건 선택)에서 [_mcTriggerDetection]을 선택하고 Mode에서 [드라이버 모드]를 선택한 상태에서 MC_MoveLink(사다리꼴 패턴 캠) 명령을 사용하려면 다음 오브젝트 데이터를 매핑하십시오.
매핑은 Sysmac Studio 축 기본 설정의 [고급 설정]에서 실행합니다.

- 래치 기능(60B8Hex)
- 래치 Status(60B9Hex)
- 래치 위치 1(60BAHex)
- 래치 위치 2(60BCHex)

이러한 데이터를 설정하지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다.

데이터 오브젝트 매핑에 대해서는 '2-3 PDO 매핑(2-27페이지)', "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.
NX 시리즈 위치 인터페이스 유니트를 사용하는 경우는 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"의 'I/O 항목 매핑'에 기재된 내용을 참조하십시오.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● Master(주축) 지정

Master(주축)에서 주축을 지정합니다.

● Slave(종축) 지정

Slave(종축)에서 종축을 지정합니다.

● TriggerInput(트리거 입력 조건), TriggerVariable(트리거 변수)

동기 시작 조건 지정이 _mcTriggerDetectoin일 때 동기를 시작할 입력 신호를 지정합니다.
입력 신호의 선택, 트리거가 발생하는 타이밍에 대한 자세한 내용은 'MC_TouchProbe(3-324페이지)'를 참조하십시오.

TriggerInput(트리거 입력 조건)의 Mode(모드)에서 드라이버 모드를 지정한 경우는 종축의 드라이버 입력을 사용합니다.
컨트롤러 모드를 지정한 경우는 TriggerVariable(트리거 변수)을 트리거 신호로 사용합니다.

● ReferenceType(위치 종류 선택)

위치 종류를 다음에서 선택할 수 있습니다.

- _mcCommand: 명령 위치(직전 프라이머리 주기에서의 계산값)
  현재 주기에 대해 직전의 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  종축의 명령 위치를 계산하기 직전의 프라이머리 정주기 태스크에서 계산한 주축의 명령 위치를 사용합니다.
- _mcFeedback: 동일 프라이머리 주기에서의 취득값
  동일 프라이머리 주기에서 취득한 주축의 피드백 위치를 사용합니다.
- _mcLatestCommand: 명령 위치(동일 프라이머리 주기에서의 계산값)
  동일 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  _mcCommand보다도 새로운 정보를 사용할 수 있습니다. 단, 주축 번호는 종축 번호보다 빠른 번호를 설정해야 합니다.
  종축이 주축보다 축 번호가 빠르면 이 명령의 Error(에러)가 TRUE가 됩니다. ErrorID(에러 코드)에는 '주축/종축 축 번호 비오름차순(에러 코드: 5438 Hex)'이 출력됩니다.

참고

동일 프라이머리 주기에서 계산한 명령 위치를 선택하는 것이 직전 프라이머리 주기에서의 계산값을 선택하는 것보다 동기 정확도는 향상됩니다.
단, 'Master(주축)에 설정된 모션 제어 시스템 변수의 축 번호 < Slave(종축)에 설정된 모션 제어 시스템 변수의 축 번호'가 되도록 주축과 종축을 사용해야 합니다.

● 축 종류와 위치 종류의 관계

감시 가능한 축 종류와 감시하는 위치 종류의 관계는 다음과 같습니다.

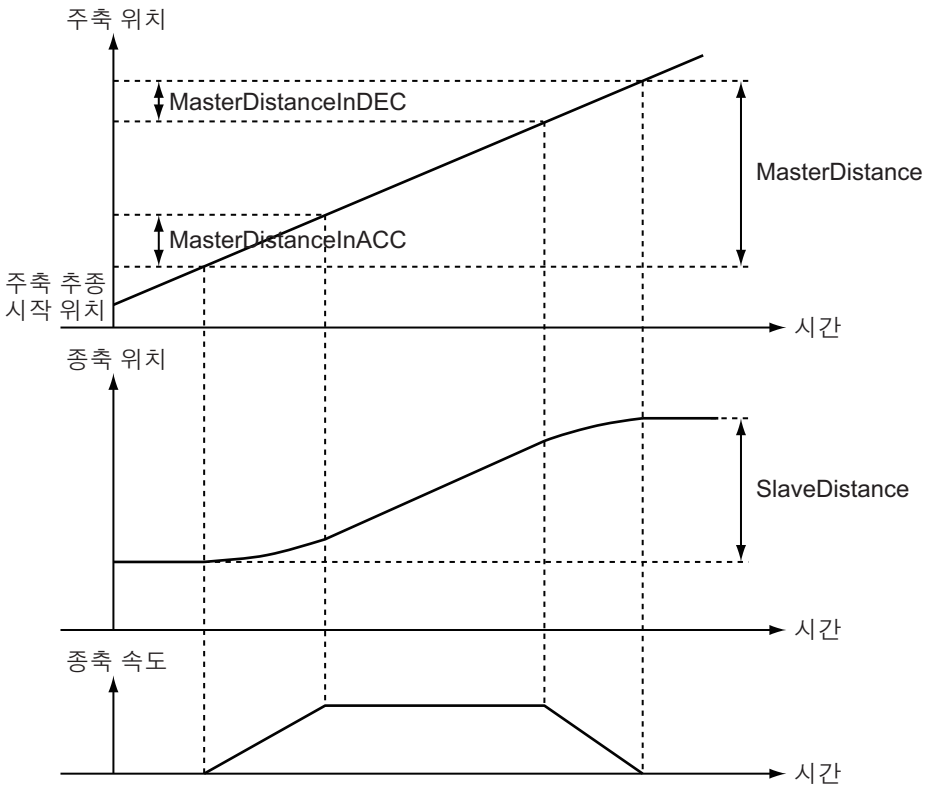
축 종류	ReferenceType	
	_mcCommand 또는 _mcLatestCommand	_mcFeedback
서보 축	○	○
엔코더 축	× *1	○
가상 서보 축	○	○
가상 엔코더 축	× *1	○

*1. 명령 기동 시에 '위치 종류 선택 범위 외(에러 코드: 5430 Hex)'가 발생합니다.

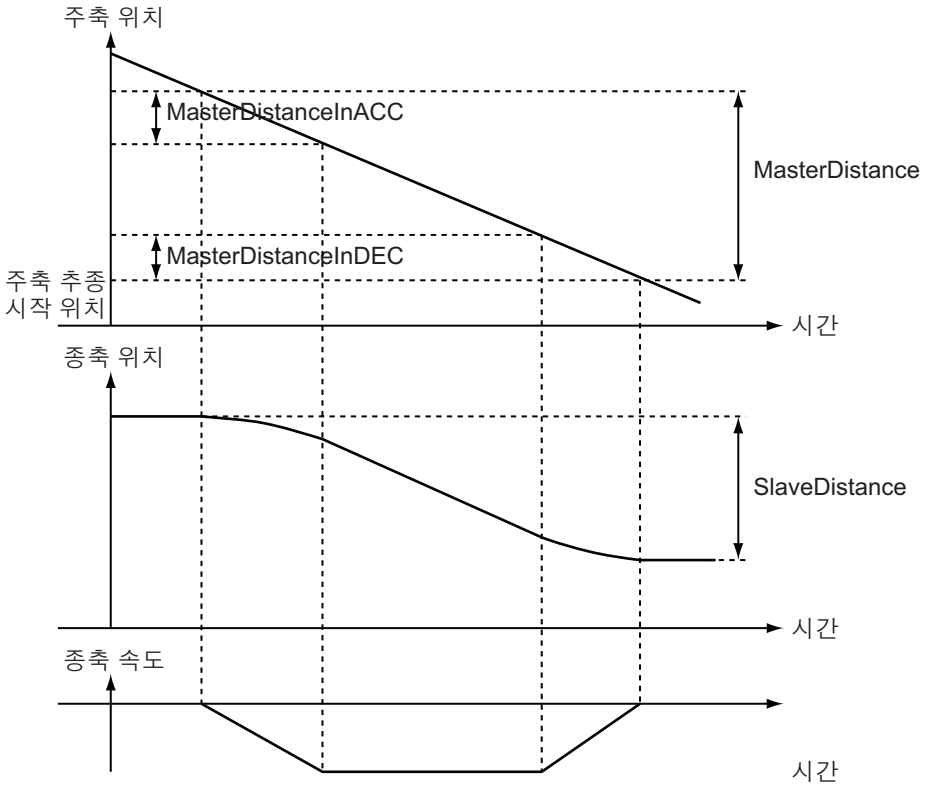
● SlaveDistance(종축 이동 거리), MasterDistance(주축 이동 거리), MasterDistanceInACC(주축 가속 이동 거리), MasterDistanceInDEC(주축 감속 이동 거리)

종축의 속도와 위치는 다음 그림과 같이 주축과 종축의 이동량 비로 결정됩니다.
다음 그림의 주축 추종 거리는 동기 시작 조건이 유효해졌을 때의 위치를 나타냅니다.

주축이 정방향으로 이동하는 경우



주축이 역방향으로 이동하는 경우



주축의 이동량과 종축의 이동량 관계는 다음 표와 같습니다.

구간		주축 이동량과 종축 이동량의 관계
가속	주축	주축 가속 이동 거리
	종축	$\text{종축 이동 거리} \times \frac{\frac{\text{주축 가속 이동 거리}}{2}}{\frac{\text{주축 가속 이동 거리}}{2} + (\text{주축 이동 거리} - \text{주축 가속 이동 거리} - \text{주축 감속 이동 거리}) + \frac{\text{주축 감속 이동 거리}}{2}}$
등속	주축	주축 이동 거리 − 주축 가속 이동 거리 − 주축 감속 이동 거리
	종축	종축 이동 거리 − 상기 가속 시의 종축 이동 거리 − 하기 감속 시의 종축 이동 거리
감속	주축	주축 감속 이동 거리
	종축	$\text{종축 이동 거리} \times \frac{\frac{\text{주축 감속 이동 거리}}{2}}{\frac{\text{주축 가속 이동 거리}}{2} + (\text{주축 이동 거리} - \text{주축 가속 이동 거리} - \text{주축 감속 이동 거리}) + \frac{\text{주축 감속 이동 거리}}{2}}$

주축의 등속 구간이 음수이면 등속 이동량 이상이 발생하고 정지합니다.
또한 종축을 주축과 동일한 속도로 이송하려면 종축의 이동량에는 다음값을 지정하십시오.

$$\text{종축 이동 거리} = \frac{\text{주축 가속 이동 거리}}{2} + (\text{주축 이동 거리} - \text{주축 가속 이동 거리} - \text{주축 감속 이동 거리}) + \frac{\text{주축 감속 이동 거리}}{2}$$

사용상의 주의 사항

주축의 카운터 모드가 로터리 모드인 경우 MasterDistance(주축 이동 거리)에는 주축의 링 카운터 1주기 이내의 값을 지정하십시오.

● LinkOption(동기 시작 조건 선택)

종축이 주축에 동기할 조건을 지정합니다.

- 명령 시작 시
  이 명령 기동 시 다음 주기부터 종축이 주축에 동기해서 위치를 결정합니다.
- 트리거 검출 시
  입력 트리거 지정에서 지정한 입력 신호가 유효해진 다음 주기부터 종축이 주축에 동기해서 위치를 결정합니다.
- 주축이 주축 추종 거리에 도달 시
  이 명령이 기동 중에 주축이 주축 추종 거리에 도달한 시점에서 다음 주기부터 종축이 동기해서 위치를 결정합니다.
  또한 주축이 주축 추종 거리에 정지한 상태로 이 명령이 기동된 시점이라도 다음 주기부터 종축이 동기해서 위치를 결정합니다.

사용상의 주의 사항

LinkOption(동기 시작 조건 선택)에서 [_mcTriggerDetection]을 선택하고 Mode에서 [드라이버 모드]를 선택하면 오브젝트 데이터를 매핑해야 합니다.
다음과 같이 오브젝트를 설정하십시오.

- 래치 기능(60B8Hex)
- 래치 Status(60B9Hex)
- 래치 위치 1(60BAHex)
- 래치 위치 2(60BCHex)

이러한 데이터를 설정하지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다. 오브젝트 데이터 매핑에 대해서는 '2-3 PDO 매핑(2-27페이지)', "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● MasterStartDistance(주축 추종 거리)

종축이 동기를 시작할 주축의 위치를 절대 위치로 지정합니다.

● BufferMode(버퍼 모드 선택)

하나 이전의 축 동작과 이번 동작의 연결 방법을 지정합니다.
종류는 다음 2가지입니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다.
Buffered	현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

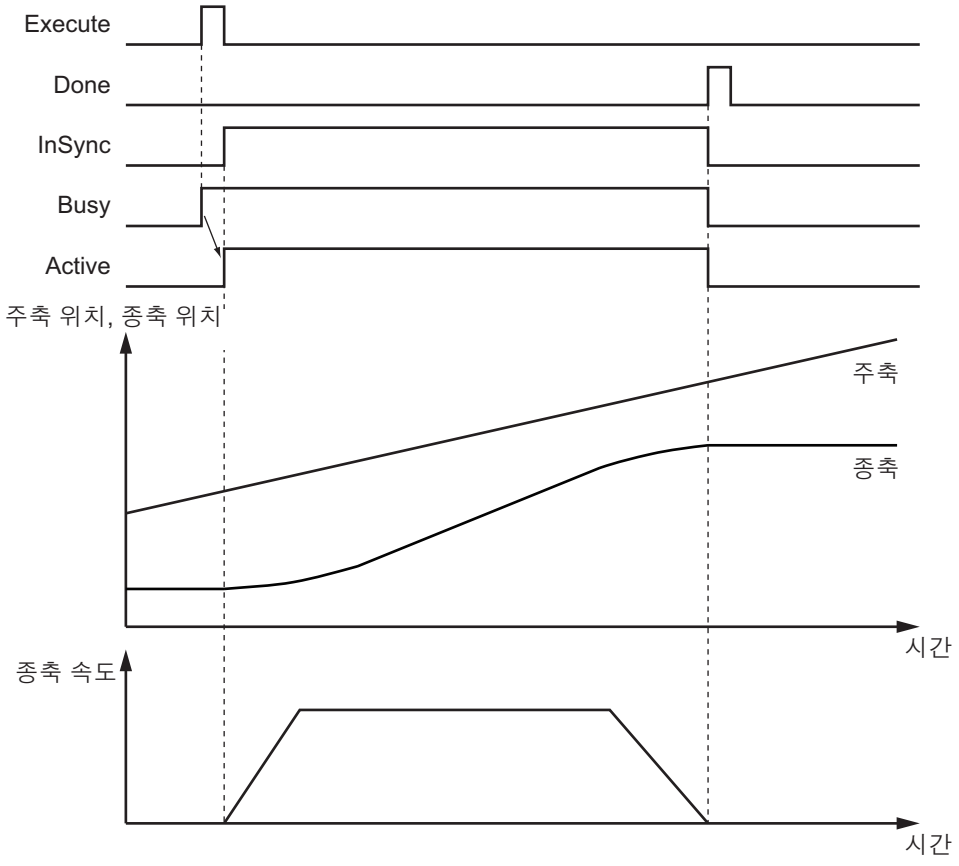
● 인포지션 확인

종축의 동작이 완료된 시점에서 인포지션 확인을 실행합니다.
이 명령의 인포지션 확인은 축 파라메터의 [인포지션 폭], [인포지션 확인 시간] 설정에 따라 실행됩니다.

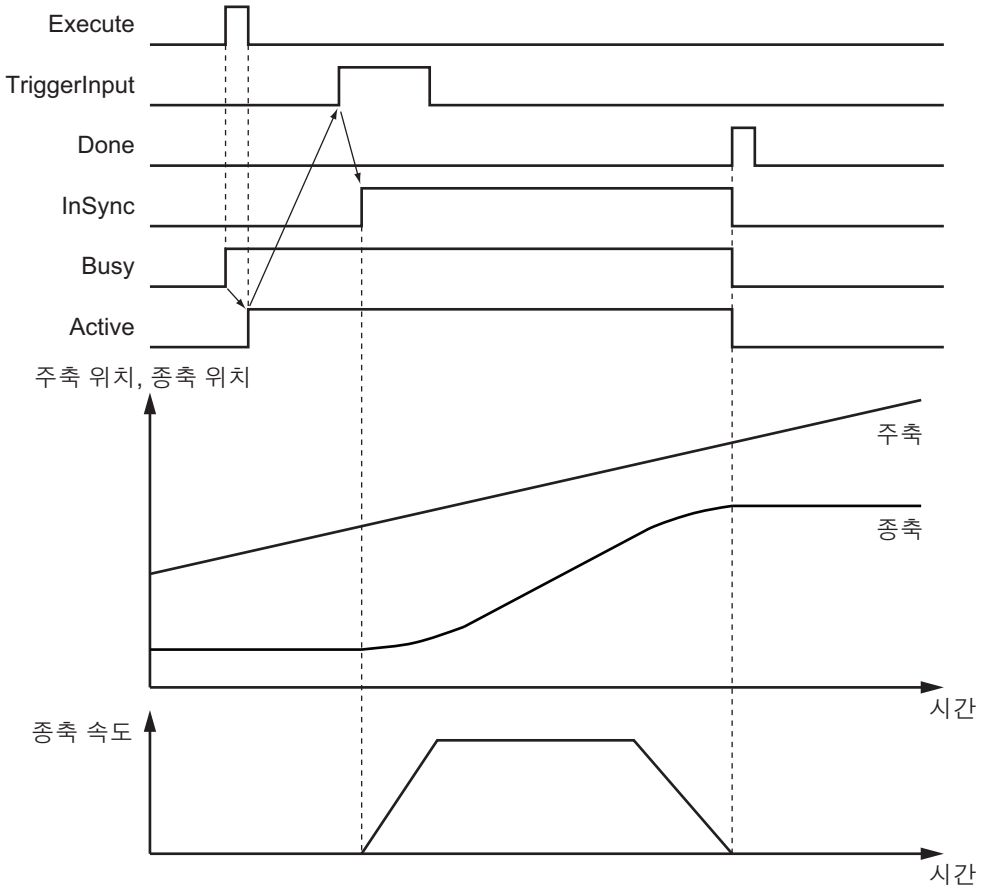
타이밍 차트

● 이 명령을 기동한 경우

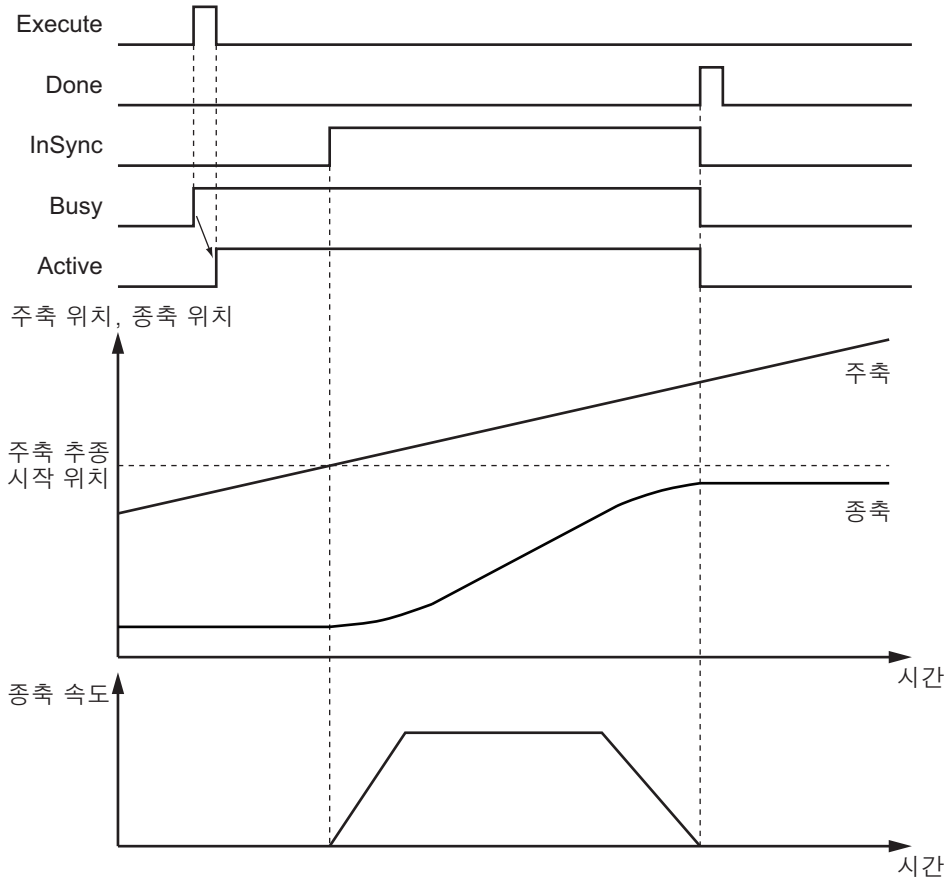
동기 시작 조건 지정=명령 시작 시



동기 시작 조건 지정=트리거 검출 시



동기 시작 조건 지정=주축이 주축 추종 거리에 도달 시

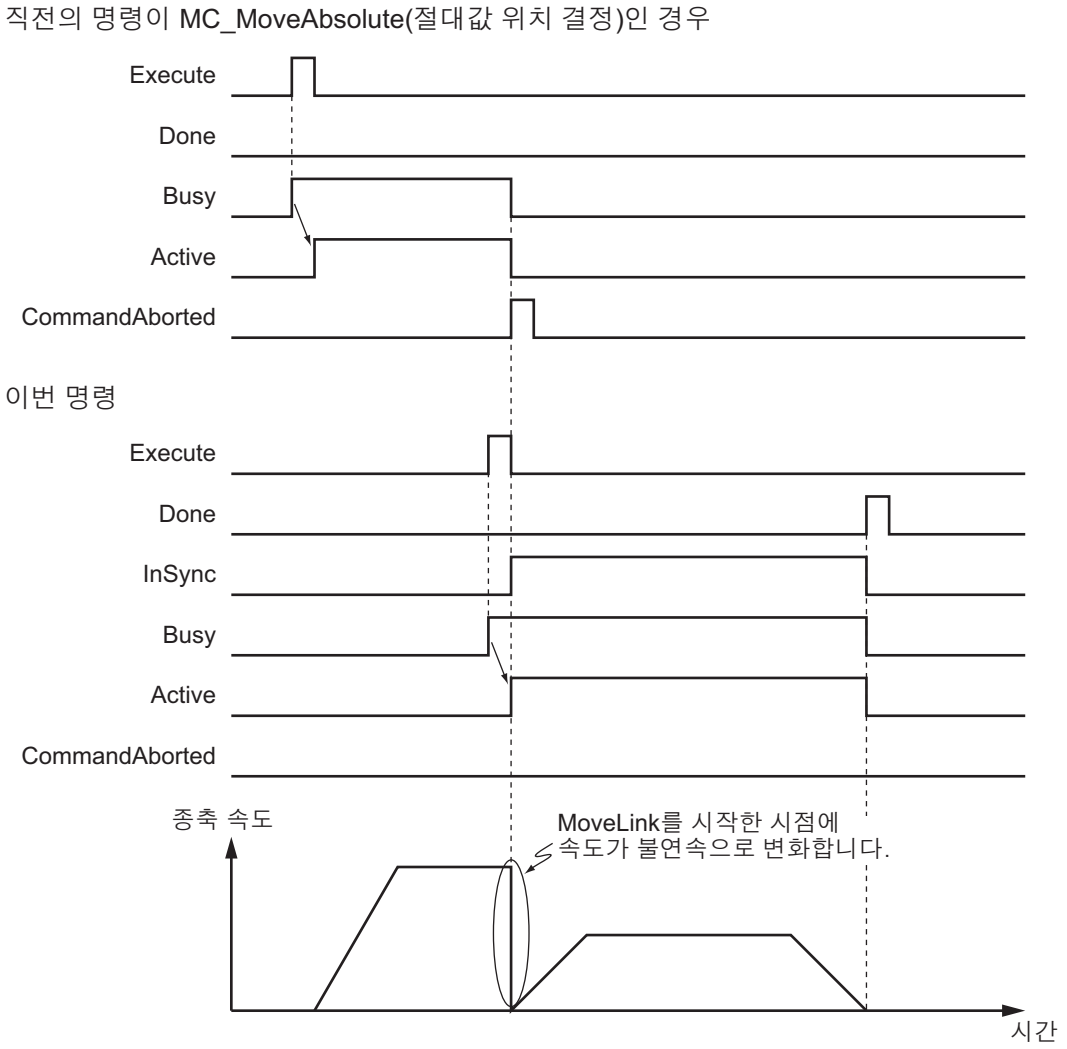


● 하나 이전의 동작 중에 BufferMode(버퍼 모드 선택)를 Aborting으로 기동한 경우

하나 이전의 동작이 MC_MoveAbsolute(절대값 위치 결정) 명령을 실행 중에 동기 시작 조건 지정에서 명령 시작 시를 지정하고 이 명령을 실행한 경우에 대해 설명합니다.

사용상의 주의 사항

다음 그림과 같이 이 명령 시작 시점에 종축의 속도가 불연속이 되므로 주의하십시오.

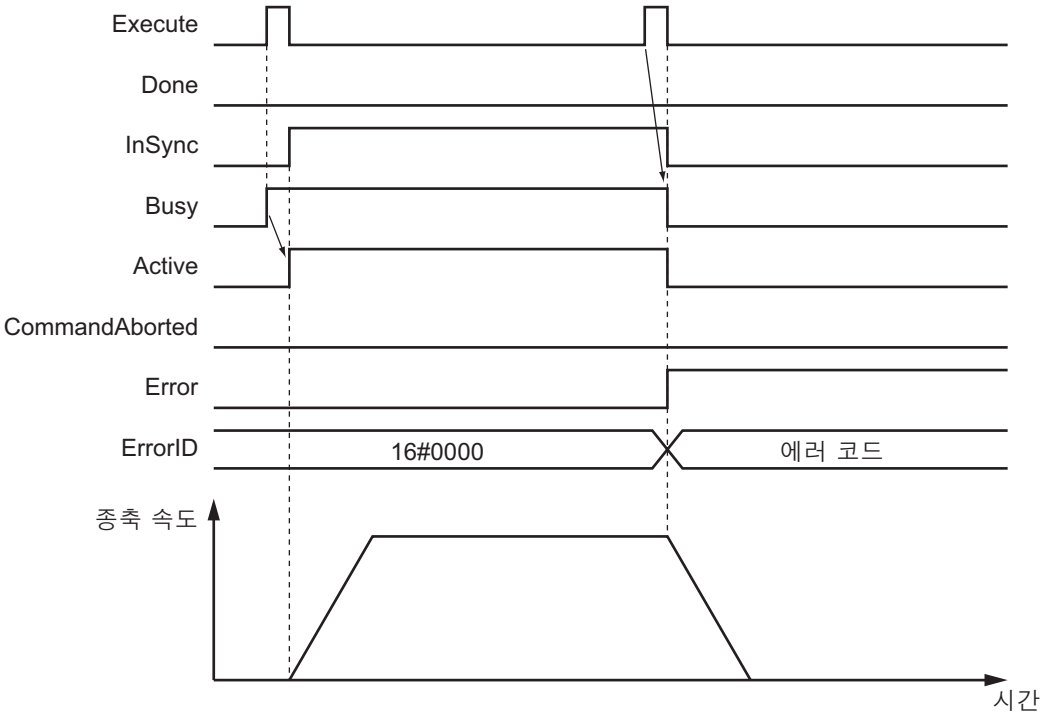


● 하나 이전의 동작 중에 BufferMode(버퍼 모드 선택)를 Buffered로 기동한 경우

하나 이전의 동작이 완료되고 나서 이 명령을 실행합니다.

모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 되고 이 축의 동작이 정지합니다.



모션 명령 다중 기동

명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하는 경우 Aborting, Buffered가 Blending 가능합니다.
Blending에 의한 명령 다중 기동은 불가능합니다.

이상

이 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

이상 발생 후의 타이밍 차트는 '모션 명령 재기동(3-272페이지)'을 참조하십시오.

● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기에는 재단 제어에 대한 샘플 프로그램이 실려 있습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축(주축)
축 2	서보 축(종축)

카운터 모드

축	카운터 모드
축 1	로터리 모드
축 2	리니어 모드

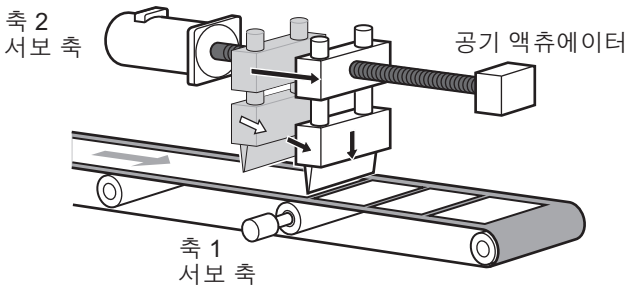
링 카운터

축	상한값	하한값
축 1	360	0

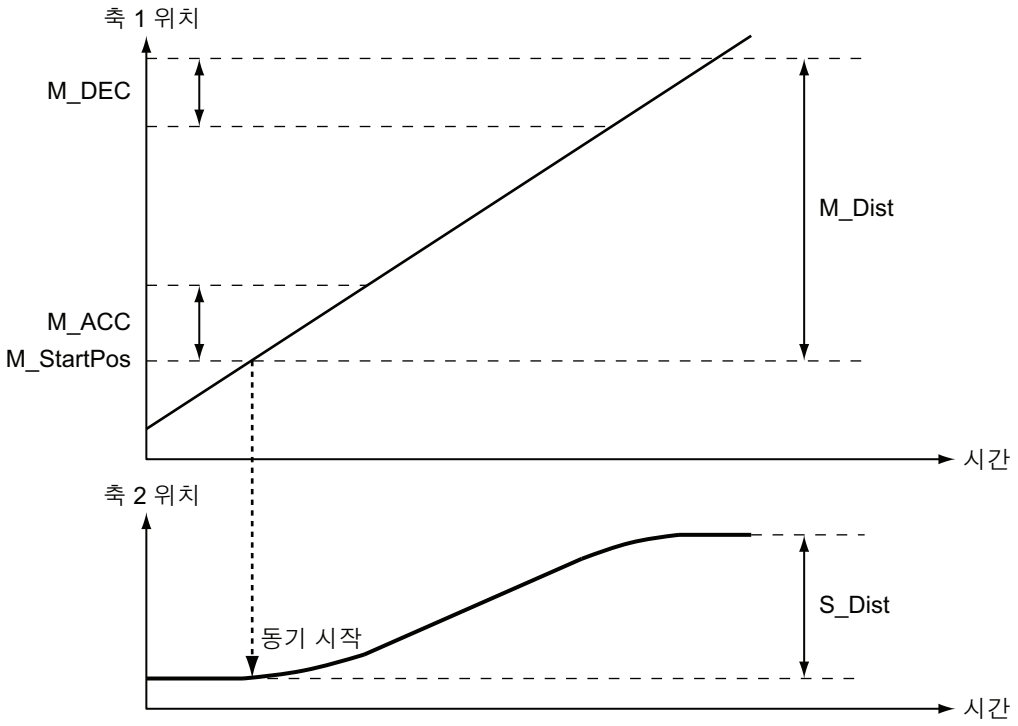
좌표 단위 선택

축	좌표 단위
축 1	degree
축 2	mm

동작 예



● 동작 패턴



1 주축의 시작

벨트 컨베이어에 설치한 축 1을 주축으로 한 위치를 피드백합니다.

2 종축의 기동

수평 방향으로 동작하는 볼 나사에 설치된 축 2가 축 1에 동기해서 동작합니다.

3 커터를 통한 재단

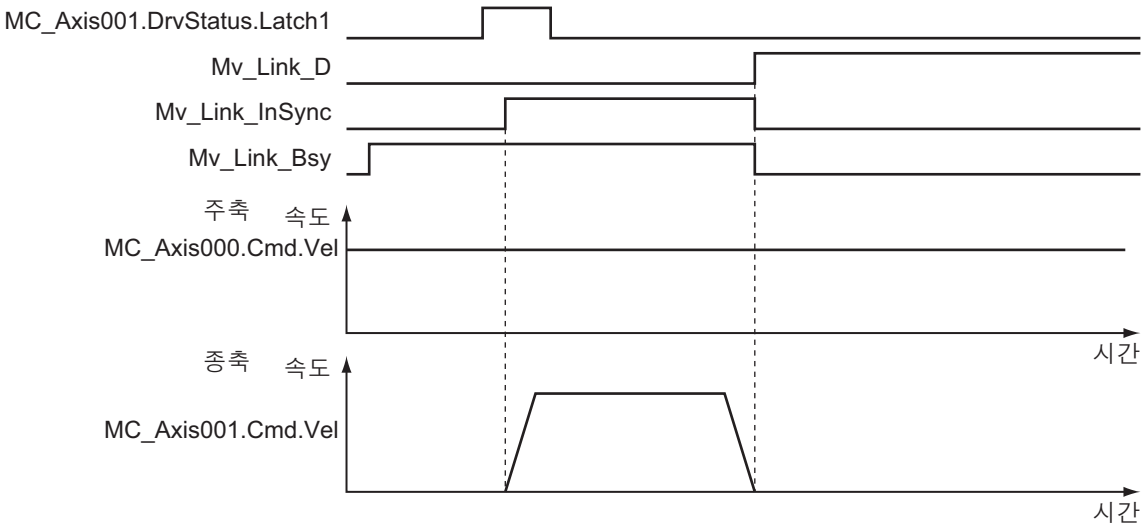
축 2가 동기 중에 공기 액츄에이터가 'ON'하고 공기 액츄에이터에 연결된 커터가 수직 방향으로 내려와 워크를 재단합니다.

래더도

● 주요 변수

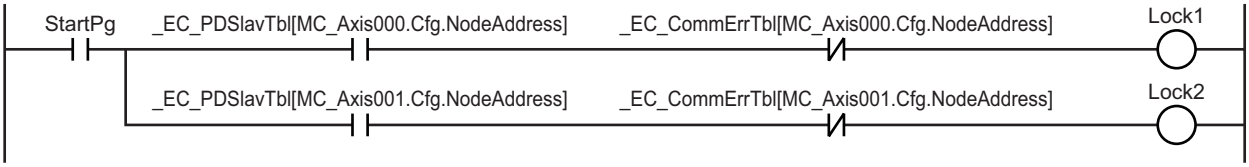
이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
Pwr1_Status	BOOL	FALSE	MC_Power 인스턴스 PWR1의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 인스턴스 PWR2의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Actuator	BOOL	FALSE	축 1과 축 2가 동기하면 TRUE가 됩니다. Actuator가 TRUE일 때 커터가 수직 방향으로 내려옵니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트

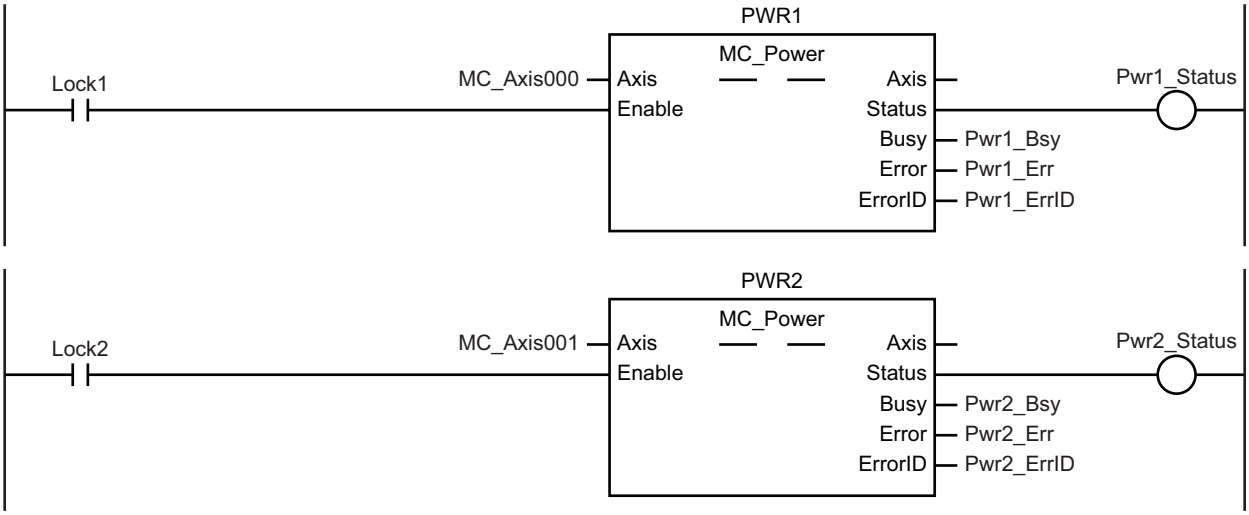


● 샘플 프로그램

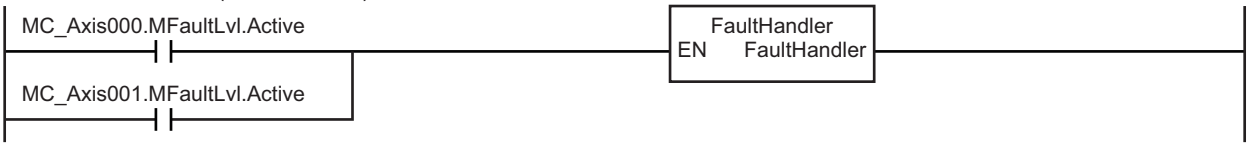
접점 StartPg가 TRUE가 되면 각 축의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인



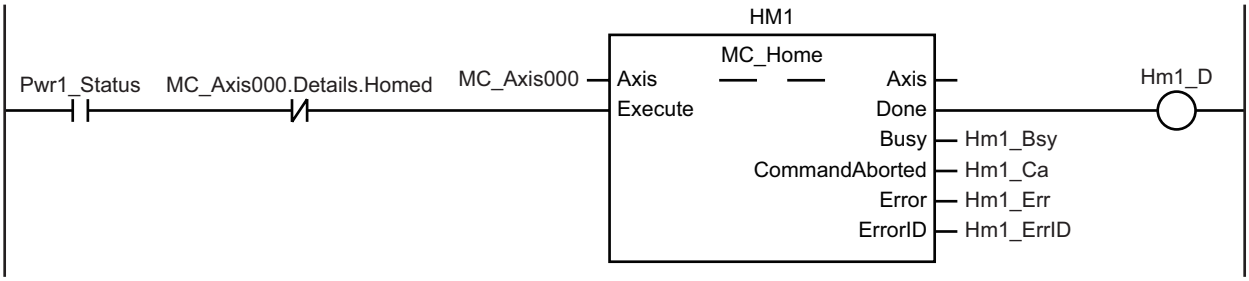
프로세스 데이터 통신을 하고 있으면 각 축의 서보를 ON



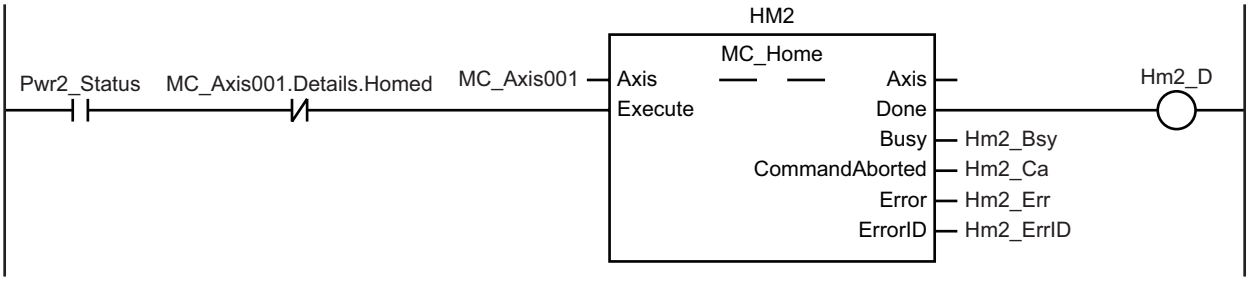
축 구성에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



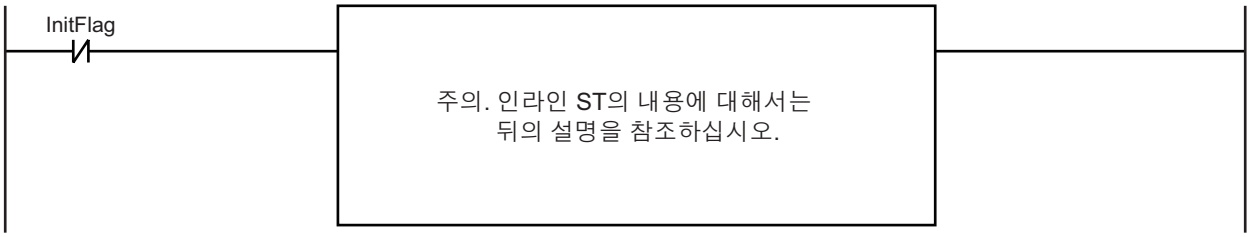
축 1이 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행하고 원점을 확정



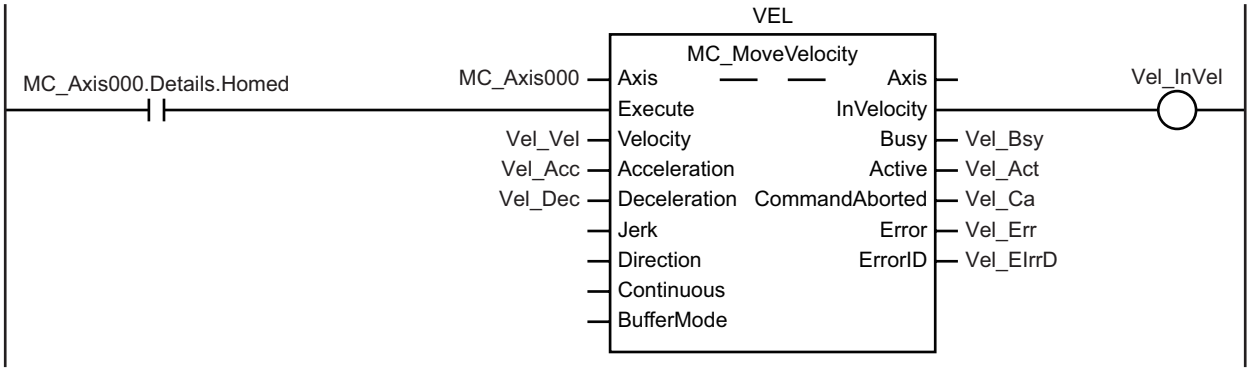
축 2가 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행하고 원점을 확정



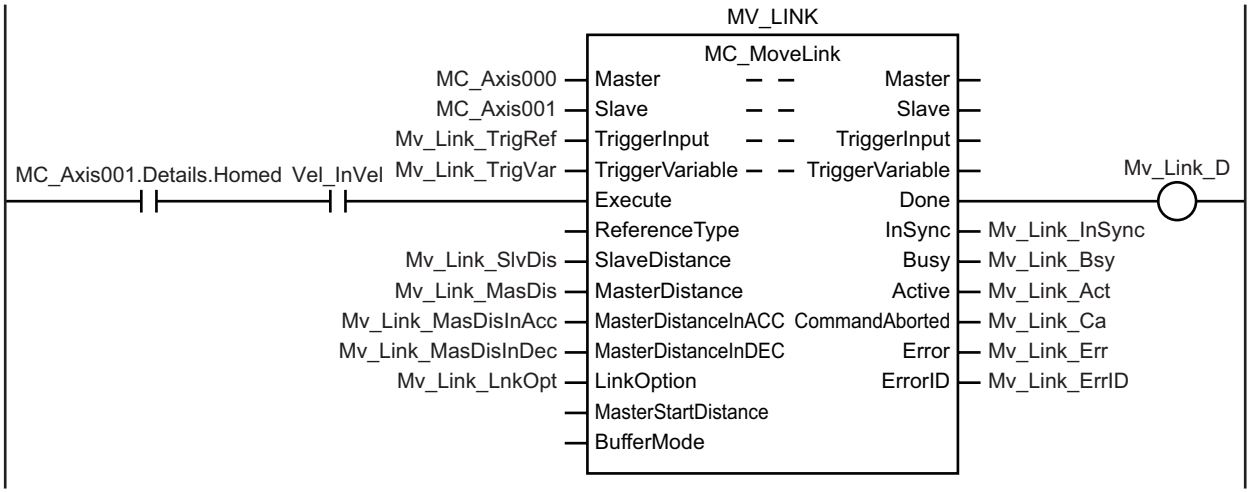
MC_MoveVelocity(속도 제어) 명령과 MC_MoveLink(사다리꼴 패턴 캠) 명령의 파라메터를 설정



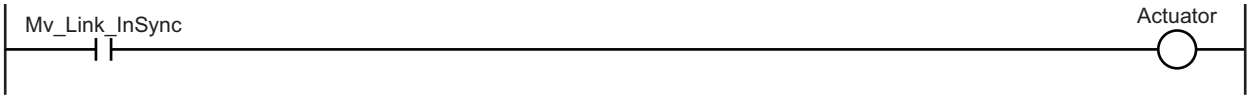
축 1이 원점 확정 상태가 되면 MC_MoveVelocity(속도 제어) 명령을 실행



종축(축 2)이 원점 확정 상태가 되면 MC_MoveLink(사다리꼴 패턴 캠) 명령을 실행



동기 중에는 Actuator를 ON



인라인 ST의 내용

```
//MC_MoveVelocity 파라메터
Vel_Vel    := LREAL#1000.0;
Vel_Acc    := LREAL#0.0;
Vel_Dec    := LREAL#0.0;

//MC_MoveLink 파라메터
Mv_Link_TrigRef.Mode        := _eMC_TRIGGER_MODE#_mcDrive;
Mv_Link_TrigRef.LatchID     := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
Mv_Link_TrigRef.InputDrive  := _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
Mv_Link_TrigVar             := FALSE;
Mv_Link_SlvDis              := LREAL#1000.0;
Mv_Link_MasDis              := LREAL#1000.0;
Mv_Link_MasDisInAcc         := LREAL#100.0;
Mv_Link_MasDisInDec         := LREAL#100.0;
Mv_Link_LnkOpt              := _eMC_LINKOPTION#_mcTriggerDetection;

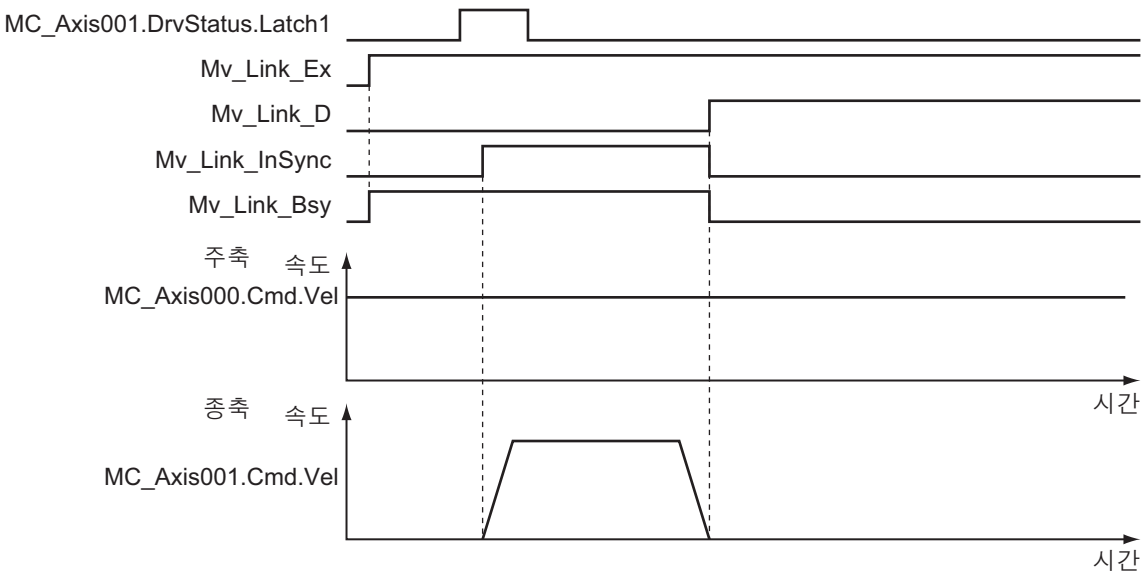
//입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
InitFlag :=TRUE;
```

스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	주축인 축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	종축인 축 2의 축 변수입니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
Pwr1_Status	BOOL	FALSE	MC_Power 인스턴스 PWR1의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power 인스턴스 PWR2의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
Actuator	BOOL	FALSE	축 1과 축 2가 동기하면 TRUE가 됩니다. Actuator가 TRUE일 때 커터가 수직 방향으로 내려옵니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 타이밍 차트



● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag=FALSE THEN

    //MC_MoveVelocity 파라메터
    Vel_Vel   := LREAL#1000.0;
    Vel_Acc   := LREAL#0.0;
    Vel_Dec   := LREAL#0.0;

    //MC_MoveLink 파라메터
    Mv_Link_TrigRef.Mode        := _eMC_TRIGGER_MODE#_mcDrive;
    Mv_Link_TrigRef.LatchID     := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
    Mv_Link_TrigRef.InputDrive  := _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
    Mv_Link_TrigVar             := FALSE;
    Mv_Link_SlvDis              := LREAL#1000.0;
    Mv_Link_MasDis              := LREAL#1000.0;
    Mv_Link_MasDisInAcc         := LREAL#100.0;
    Mv_Link_MasDisInDec         := LREAL#100.0;
    Mv_Link_LnkOpt              := _eMC_LINKOPTION#_mcTriggerDetection;

    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag :=TRUE;

END_IF;


//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 1을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr1_En:=TRUE;      //축 1을 서보 ON합니다.
ELSE
    Pwr1_En:=FALSE;     //축 1을 서보 OFF합니다.
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 2를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
    Pwr2_En:=TRUE;      //축 2를 서보 ON합니다.
ELSE
    Pwr2_En:=FALSE;     //축 2를 서보 OFF합니다.
END_IF;

//가벼운 수준의 이상 발생 시 처리
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

// 축 1이 서보 ON 상태이고 원점 미확정 상태일 때 원점 복귀를 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//축 2가 서보 ON 상태이고 원점 미확정 상태일 때 원점 복귀를 실행합니다.
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;
```

```
//축 1이 원점 확정 후에 축 1에서 MC_MoveVelocity를 실행합니다.
IF MC_Axis000.Details.Homed=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

//축 2가 원점 확정 상태이고 축 1이 목표 속도에 도달한 것을 확인하면 축 2에서 MC_MoveLink를 실
행합니다.
IF (MC_Axis001.Details.Homed=TRUE) AND (Vel_InVel=TRUE) THEN
    Mv_Link_Ex:=TRUE;
END_IF;

//축 1과 축 2가 동기하면 액츄에이터가 ON합니다.
IF Mv_Link_InSync=TRUE THEN
    Actuator:=TRUE;
ELSE
    Actuator:=FALSE;
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                => Hm_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_MoveVelocity
VEL(
    Axis                := MC_Axis000,
    Execute             := Vel_Ex,
```

```
    Velocity            := Vel_Vel,
    Acceleration        := Vel_Acc,
    Deceleration        := Vel_Dec,
    InVelocity          => Vel_InVel,
    Busy                => Vel_Bsy,
    Active              => Vel_Act,
    CommandAborted      => Vel_Ca,
    Error               => Vel_Err,
    ErrorID             => Vel_ErrID
);

// MC_MoveLink
MV_LINK(
    Master                  := MC_Axis000,
    Slave                   := MC_Axis001,
    TriggerInput            := Mv_Link_TrigRef,
    TriggerVariable         := Mv_Link_TrigVar,
    Execute                 := Mv_Link_Ex,
    SlaveDistance           := Mv_Link_SlvDis,
    MasterDistance          := Mv_Link_MasDis,
    MasterDistanceInAcc     := Mv_Link_MasDisInAcc,
    MasterDistanceInDec     := Mv_Link_MasDisInDec,
    LinkOption              := Mv_Link_LnkOpt,
    Done                    => Mv_Link_D,
    InSync                  => Mv_Link_InSync,
    Busy                    => Mv_Link_Bsy,
    Active                  => Mv_Link_Act,
    CommandAborted          => Mv_Link_Ca,
    Error                   => Mv_Link_Err,
    ErrorID                 => Mv_Link_ErrID
);
```

MC_CombineAxes

2축의 위치를 가산한 값 또는 감산한 값을 명령 위치로 출력합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_CombineAxes	가감산 위치 결정	FB	MC_CombineAxes_instance MC_CombineAxes Master — — Master Auxiliary — — Auxiliary Slave — — Slave Execute — InCombination CombineMode — Busy RatioNumeratorMaster — Active RatioDenominatorMaster — CommandAborted RatioNumeratorAuxiliary — Error RatioDenominatorAuxiliary — ErrorID ReferenceTypeMaster ReferenceTypeAuxiliary BufferMode	MC_CombineAxes_instance (Master := 《파라메터》, Auxiliary := 《파라메터》, Slave := 《파라메터》, Execute := 《파라메터》, CombineMode := 《파라메터》, RatioNumeratorMaster := 《파라메터》, RatioDenominatorMaster := 《파라메터》, RatioNumeratorAuxiliary := 《파라메터》, RatioDenominatorAuxiliary := 《파라메터》, ReferenceTypeMaster := 《파라메터》, ReferenceTypeAuxiliary := 《파라메터》, BufferMode := 《파라메터》, InCombination => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
CombineMode	가감산 방법 선택	_eMC _COMBINE_ MODE	0: _mcAddAxes 1: _mcSubAxes	0 *1	합성 방법을 지정합니다. 0 : 가산 1 : 감산
Ratio Numerator Master (Reserved)	주축 기어비 분자	DINT *2	양수 또는 음수 *2	10000	(Reserved)
Ratio Denominator Master (Reserved)	주축 기어비 분모	UDINT *3	양수	10000	(Reserved)
Ratio Numerator Auxiliary (Reserved)	보조축 기어비 분자	DINT *2	양수 또는 음수 *2	10000	(Reserved)
Ratio Denominator Auxiliary (Reserved)	보조축 기어비 분모	UDINT *3	양수	10000	(Reserved)

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Reference TypeMaster	주축 위치 종류 선택	_eMC_ REFERENCE _TYPE	1: _mcFeedback 2: _mcLatestCommand	2 *1	주축의 위치 종류를 지정합니다. 1 : 피드백 위치(동일 프라이머리 주기에서의 취득값) 2 : 명령 위치(동일 프라이머리 주기에서의 계산값)
Reference TypeAuxiliary	보조축 위치 종류 선택	_eMC_ REFERENCE _TYPE	1: _mcFeedback 2: _mcLatestCommand	2 *1	보조축의 위치 종류를 지정합니다. 1 : 피드백 위치(동일 프라이머리 주기에서의 취득값) 2 : 명령 위치(동일 프라이머리 주기에서의 계산값)
BufferMode	버퍼 모드 선택	_eMC _BUFFER_ MODE	0: _mcAborting	0 *1	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting

*1. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*2. CPU 유니트 Ver1.02 이상과 Sysmac StudioVer1.03 이상의 조합에서 사용할 수 있습니다. 그 이전의 조합에서는 데이터 형식은 'UINT', 유효 범위 '양수'가 됩니다.
*3. CPU 유니트 Ver1.02 이상과 Sysmac StudioVer1.03 이상의 조합에서 사용할 수 있습니다. 그 이전의 조합에서는 데이터 형식은 'UINT'가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
InCombination	가감산 중	BOOL	TRUE, FALSE	가감산 중에 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
InCombination	가감산 위치 결정을 시작했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Busy	Execute의 기동	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Master	주축	_sAXIS_REF	–	주축을 지정합니다. *1
Auxiliary	보조축	_sAXIS_REF	–	보조축을 지정합니다. *1
Slave	종축	_sAXIS_REF	–	종축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

사용상의 주의 사항

주축 및 종축, 보조축 중 하나에 동일한 축을 지정하면 다음의 가벼운 수준의 이상이 발생합니다.

- '주축 종축 동일(에러 코드: 5436 Hex)' 이상
- '주축 보조축 동일(에러 코드: 5437 Hex)' 이상
- '보조축 종축 동일(에러 코드: 548E Hex)' 이상

기능 설명

- 이 명령은 Execute(기동)가 기동되면 가감산 위치 결정 동작을 시작합니다.

안전상의 중요 사항

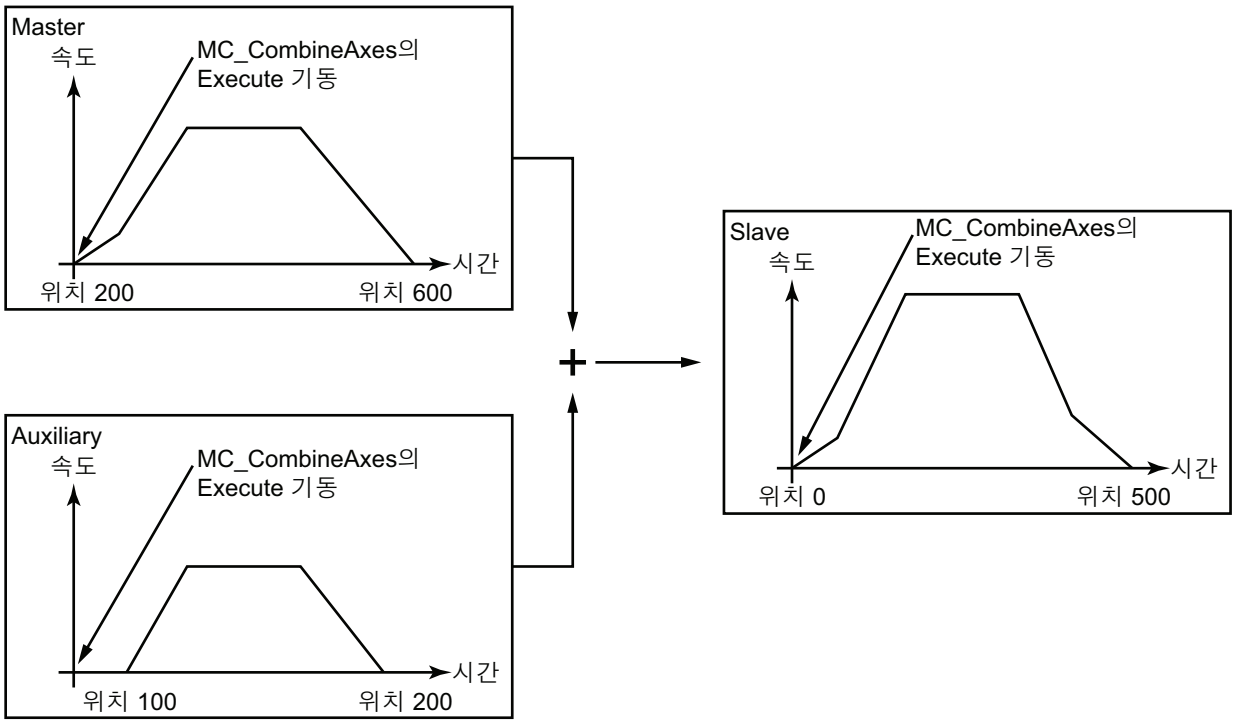
이 명령을 사용하는 경우는 Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하지 마십시오.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 실행하면 Slave(종축)가 급격하게 추종하는 경우가 있어 위험합니다.
Master(주축)에 대해 MC_SetPosition(현재 위치 변경) 명령을 사용하려면 Master(주축)와 Slave(종축)의 관계를 일단 해제하고 나서 실행하십시오.
주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.
또한 Auxiliary(보조축)에 대해서도 Master(주축)와 동일합니다.

명령에 대한 자세한 내용

시작 시점부터의 Master(주축) 위치의 상대값과 Auxiliary(보조축) 위치의 상대값을 가산 또는 감산한 상대값을 Slave(종축)의 명령 위치로 출력합니다.

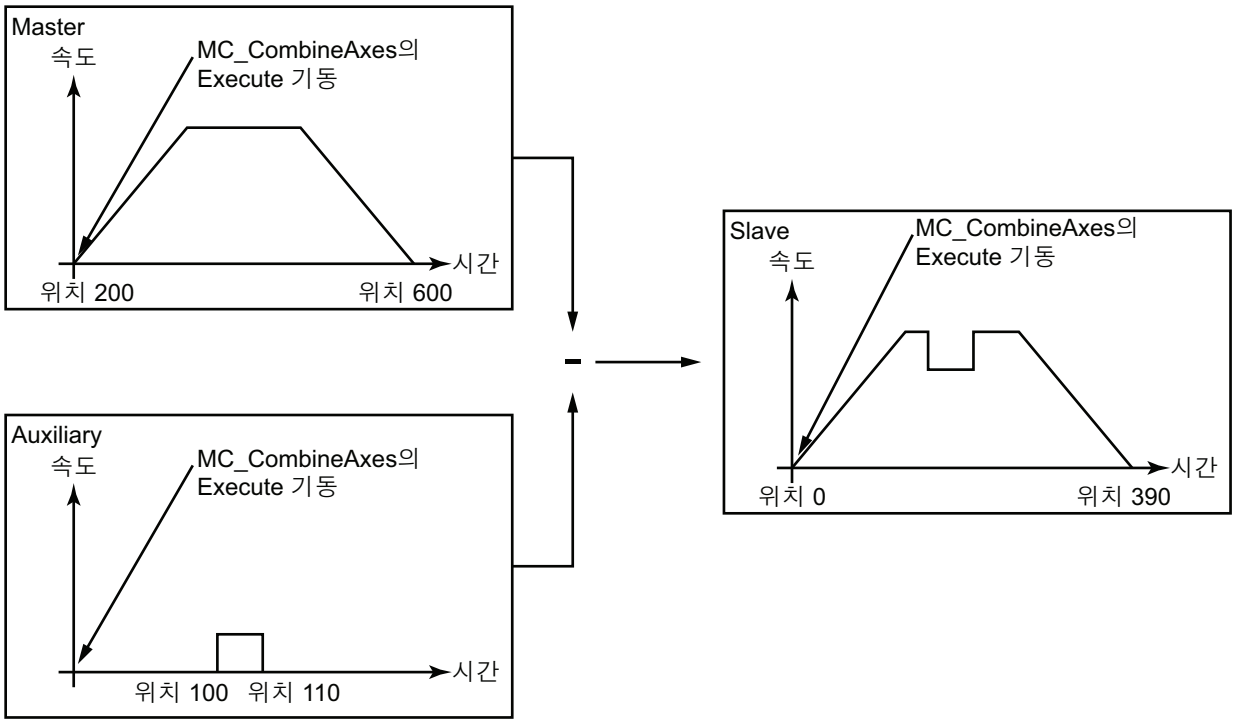
● CombineMode가 _mcAddAxes인 경우

Slave(종축) 위치 = Master(주축) 위치(상대 위치) + Auxiliary(보조축) 위치(상대 위치)



● CombineMode가 _mcSubAxes인 경우

Slave(종축) 위치 = Master(주축) 위치(상대 위치) - Auxiliary(보조축) 위치(상대 위치)



- 위치의 가산 또는 감산에 대해서는 축 파라메터의 축 좌표 단위 선택에 관계 없이 수치로 가산 또는 감산합니다.
- 이 명령을 종료하려면 MC_Stop(강제 정지) 명령을 기동하십시오.



안전상의 중요 사항

Master(주축)와 Auxiliary(보조축)의 값에 따라서는 Slave(종축) 이동량, 속도, 가속도가 급격하게 변화하는 경우가 있습니다.
주의해서 사용하십시오.

● 인포지션 확인

이 명령으로는 인포지션 확인을 실행하지 않습니다.

● 오버라이드

이 명령으로는 MC_SetOverride(오버라이드값 설정) 명령에 의한 오버라이드는 불가능합니다.

● ReferenceType(위치 종류 선택)

위치 종류를 다음에서 선택할 수 있습니다.

- _mcFeedback: 동일 프라이머리 주기에서의 취득값
  동일 프라이머리 주기에서 취득한 주축의 피드백 위치를 사용합니다.
- _mcLatestCommand: 명령 위치(동일 프라이머리 주기에서의 계산값)
  동일 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  _mcLatestCommand를 선택하면 주축 번호와 보조축 번호는 종축 번호보다 빠른 번호를 설정해야 합니다.
  종축이 주축이나 보조축보다 축 번호가 빠르면 이 명령의 Error(에러)가 TRUE가 됩니다. ErrorID(에러 코드)에는 '주축/종축 축 번호 비오름차순(에러 코드: 5438 Hex)'이 출력됩니다.
  또한 주축과 보조축 사이에는 축 번호에 관한 제약은 없습니다.

● 축 종류와 위치 종류의 관계

감시 가능한 축 종류와 감시하는 위치 종류의 관계는 다음과 같습니다.

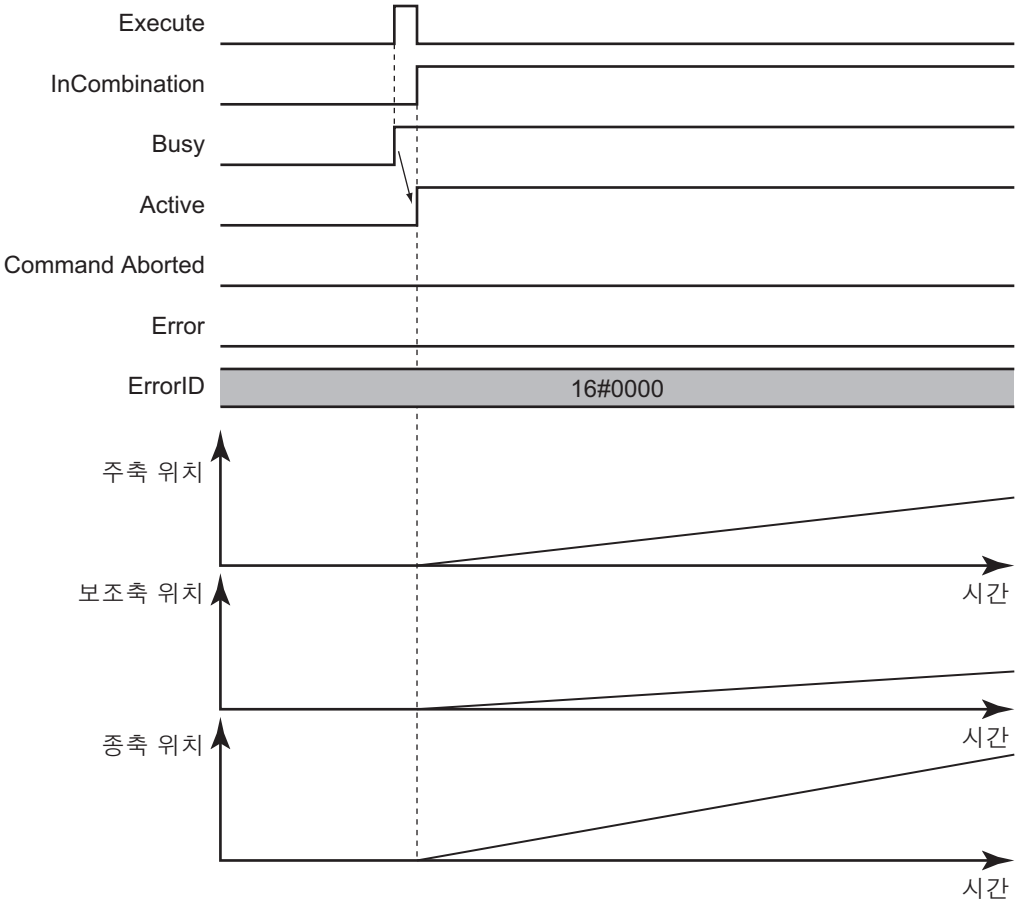
축 종류	ReferenceType	
	_mcFeedback	_mcLatestCommand
서보 축	○	○
엔코더 축	○	× *1
가상 서보 축	○	○
가상 엔코더 축	○	× *1

*1. 명령 기동 시에 '위치 종류 선택 범위 외(에러 코드: 5430 Hex)'가 발생합니다.

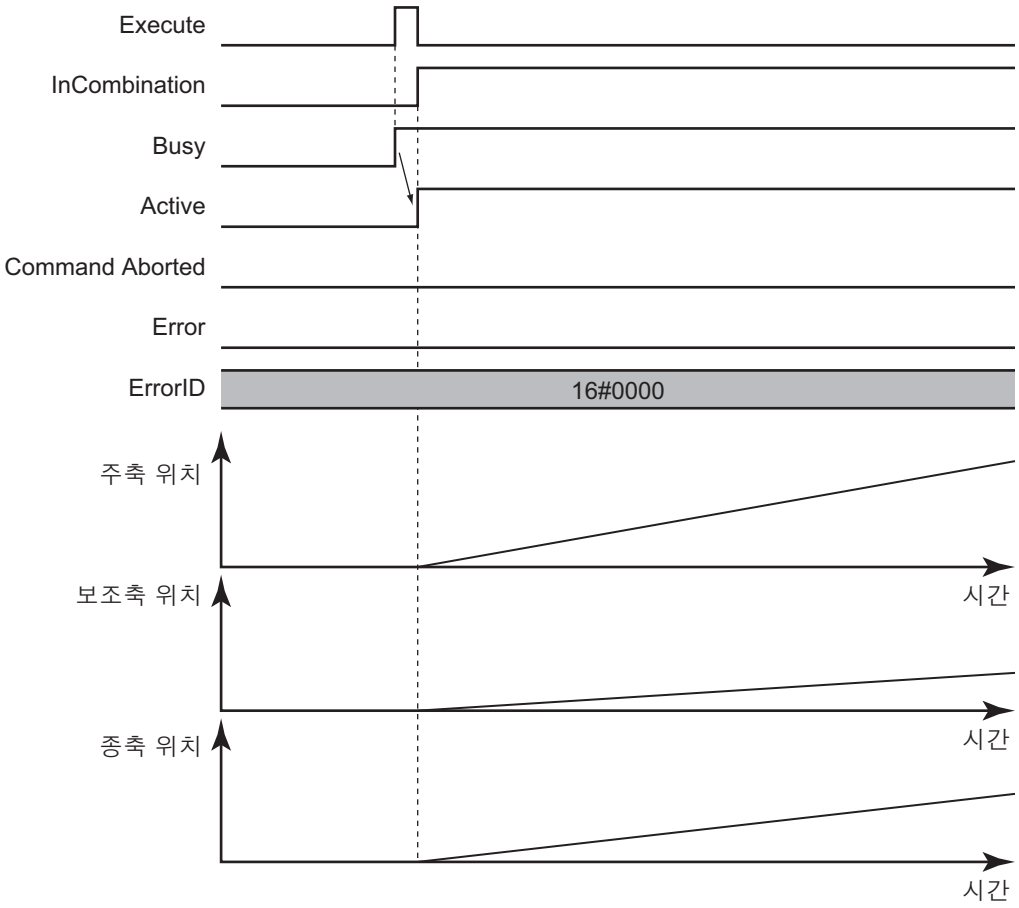
타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- 가감산 출력을 시작한 주기에서 InCombination(가감산 중)이 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중단되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중), InCombination(가감산 중)이 FALSE가 됩니다.

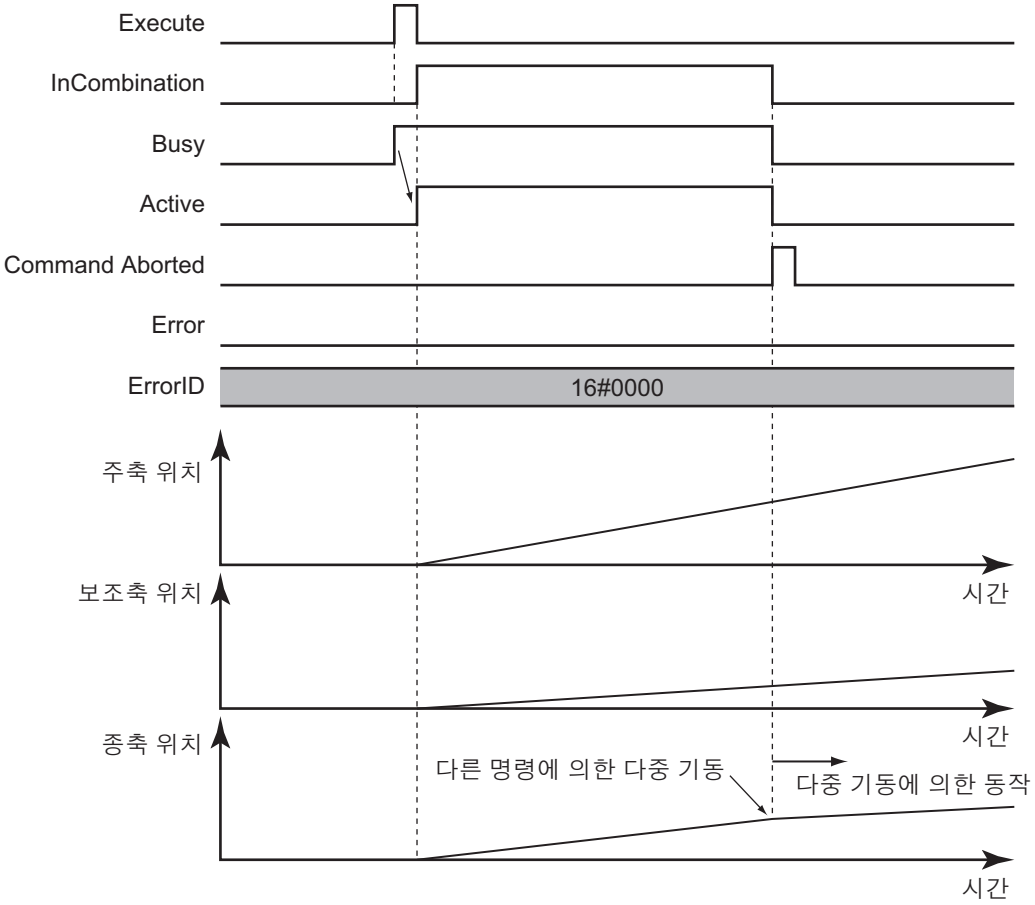
● CombineMode가 _mcAddAxes인 경우



● CombineMode가 _mcSubAxes인 경우



● 중단된 경우



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

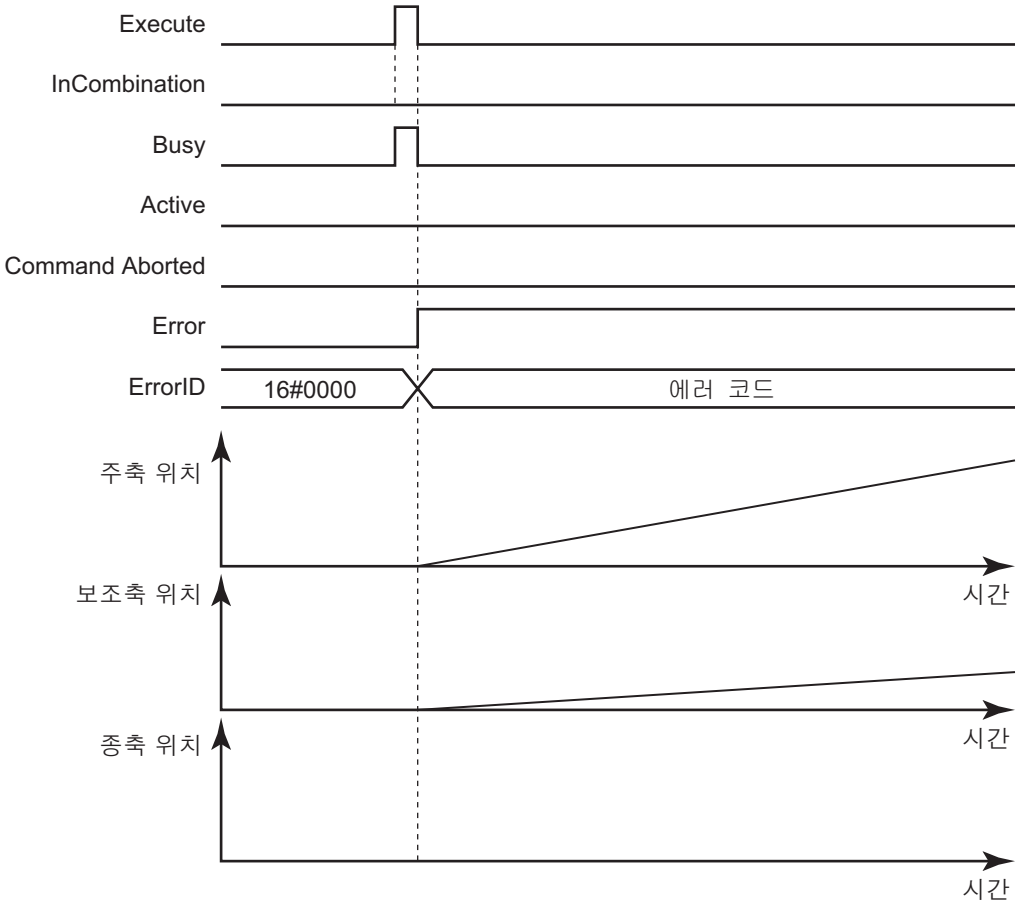
이 명령에 대한 명령 다중 기동을 실행하려면 종축을 지정하십시오.
또한 이 명령 동작 중에 다른 명령을 다중 기동하면 버퍼 모드(버퍼 모드 선택)에 의해 다음 제약이 있습니다.

- 이 명령 실행 중에 다른 명령에 의한 다중 기동은 Aborting으로 가능합니다.
  Buffered, Blending에 의한 명령 다중 기동은 불가능합니다.

이상

● 이상 발생 시의 타이밍 차트

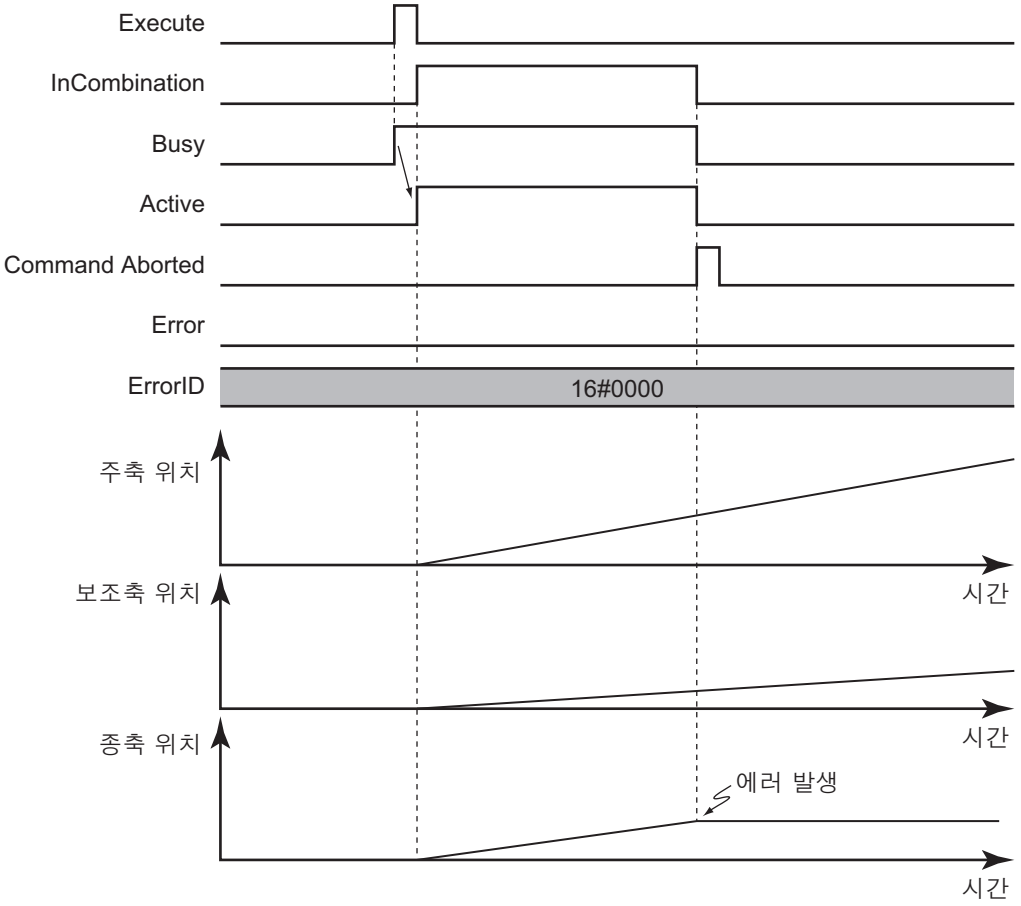
이 명령 기동 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
감속 정지할 때의 감속도는 축 파라메터의 최대 감속도입니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.



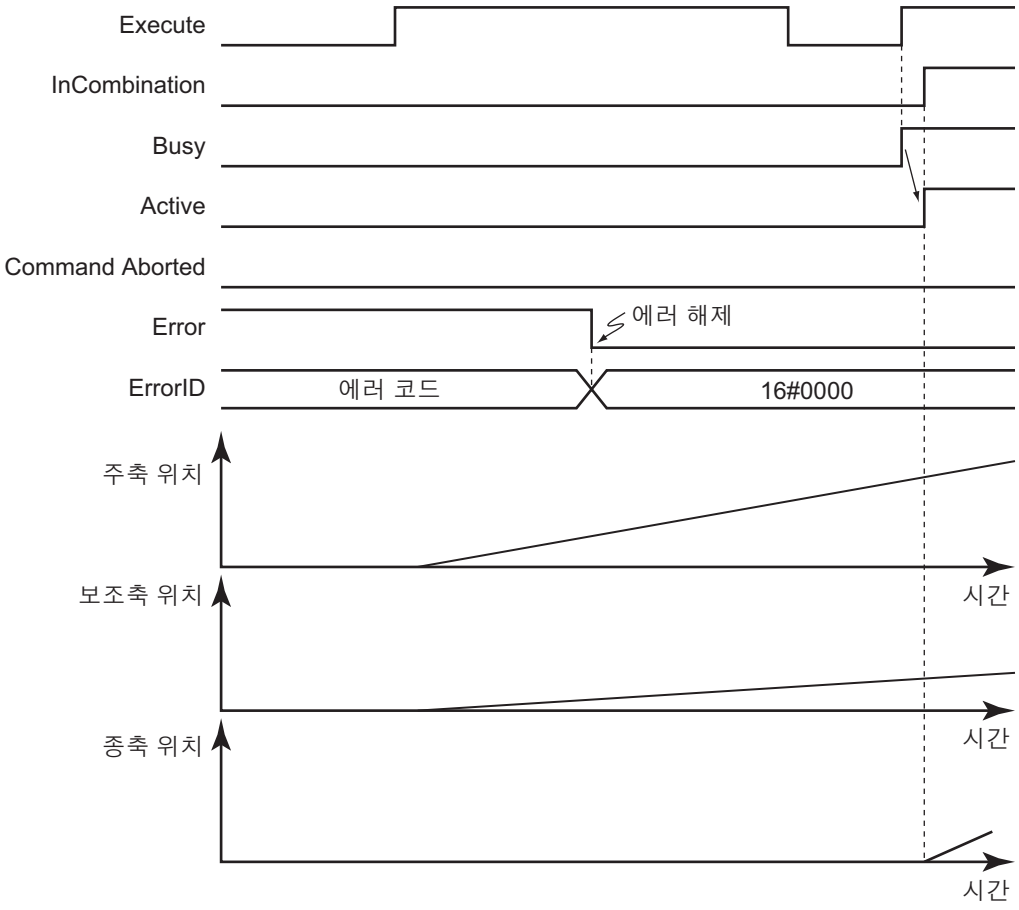
참고

- 이 명령은 Master(주축), Auxiliary(보조축)의 이상 상태에는 영향을 받지 않습니다.
- 이상 상태가 해제되고 Master(주축), Auxiliary(보조축)가 동작할 때 Slave(종축)는 가감산 위치 결정 동작을 계속합니다. 또한 이 명령 기동 시 및 이 명령 실행 중에 종축에 이상이 발생해 이 명령이 중단되어도 Master(주축), Auxiliary(보조축)에는 영향을 미치지 않습니다.

이 명령 실행 중에 가벼운 수준의 이상이 발생하면 CommandAborted(실행 중단)가 TRUE가 되고 축이 정지합니다.
감속 정지할 때의 감속도는 축 파라메터의 최대 감속도입니다.
Slave(종축)의 축 변수 MFalutLvl.Code에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.



이 명령의 이상을 해제하면 다음 Execute(기동)의 기동까지 이 명령은 기동하지 않습니다.

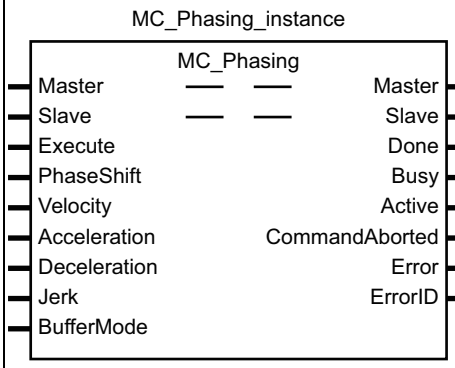


● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_Phasing

동기 제어 중인 주축의 위상 보정을 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_Phasing	주축 상대값 위상 보정	FB		MC_Phasing_instance (Master := 《파라메터》, Slave := 《파라메터》, Execute := 《파라메터》, PhaseShift := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, BufferMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
PhaseShift	위상 보정량	LREAL	음수, 양수, '0'	0	주축의 위상 보정량을 지정합니다. 단위는 [명령 단위]입니다. *1
Velocity	목표 속도	LREAL	양수	0	위상 보정량의 목표 속도(보정 속도)를 지정합니다. *2 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

*2. 목표 속도를 반드시 설정하십시오. 설정하지 않고 동작하게 하면 목표 속도 설정 범위 외(에러 코드: 5422Hex)가 발생합니다.

*3. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	위상 보정을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	위상 보정을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동 (Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때 • 실행 중인 동기 제어 명령이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Master	주축	_sAXIS_REF	–	주축을 지정합니다. *1
Slave	종축	_sAXIS_REF	–	종축을 지정합니다. *1

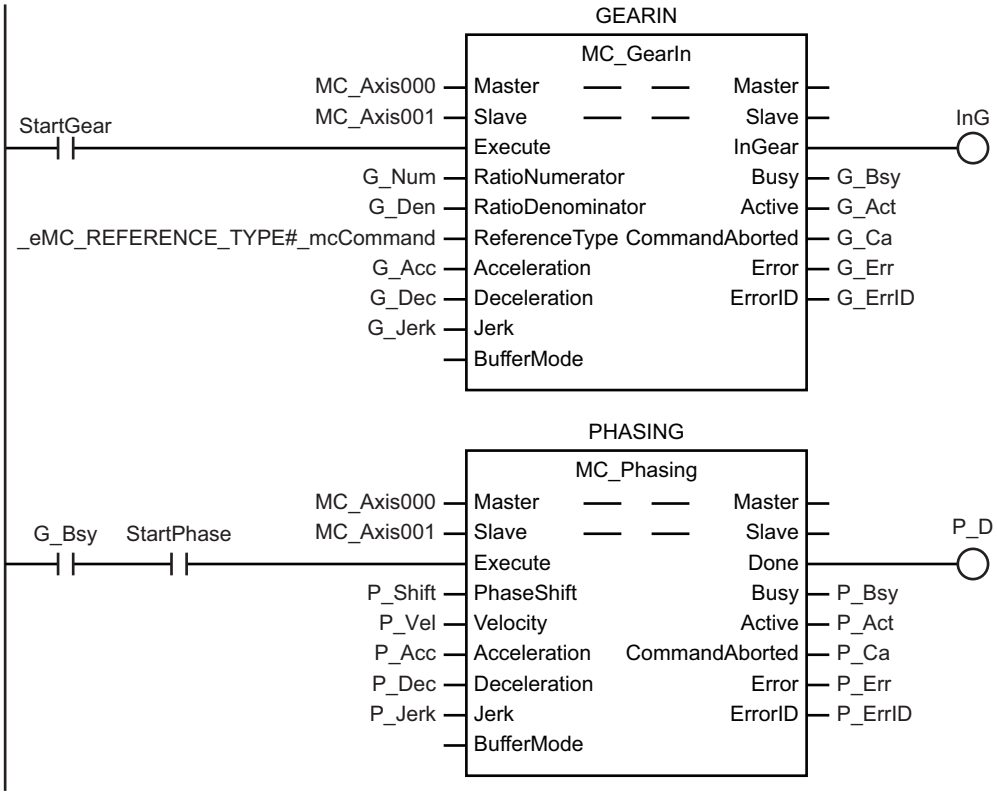
*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

사용상의 주의 사항

주축과 종축에 같은 축을 지정하면 가벼운 이상 수준의 '주축 종축 동일(에러 코드: 5436 Hex)' 이상이 됩니다.

기능 설명

- MC_CombineAxes(가감산 위치 결정) 명령 실행 중을 제외하고 단축 동기 제어 중에 이 명령을 기동하면 설정한 PhaseShift(위상 보정량), Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)에 따라 주축의 위상을 보정합니다.
- 주축의 명령 현재 위치 및 피드백 현재 위치는 변화하지 않으며, 주축의 명령 현재 위치 및 피드백 현재 위치를 상대량으로 보정한 값이 '주축의 위상'이 됩니다.
  종축은 보정된 '주축의 위상'에 대해 동기합니다.
- PhaseShift(위상 보정량)에 도달하면 Done(완료)이 TRUE가 됩니다.
- 실행 중인 동기 제어 명령이 완료되면 보정은 종료됩니다. 동기 제어 명령을 다시 실행한 경우, 이전의 보정량은 영향을 주지 않습니다.
- 주축 위상 보정이 유효가 되는 동기 명령은 MC_CamIn(캠 동작 시작) 명령, MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령의 4가지입니다.
- 이 명령을 사용자 프로그램 안에 기재하려면 다음 그림과 같이 동기 제어 명령 뒤에 MC_Phasing(주축 상대값 위상 보정) 명령을 기술하십시오.



사용상의 주의 사항

주축에 대한 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● Master(주축), Slave(종축)의 지정

Master(주축)와 Slave(종축)는 축 변수명으로 축을 지정합니다.
동기 제어 명령이 실행 중이 아닌 Master(주축), Slave(종축)의 조합을 지정하면 축 지정 이상이 발생합니다.

● PhaseShift(위상 보정량)

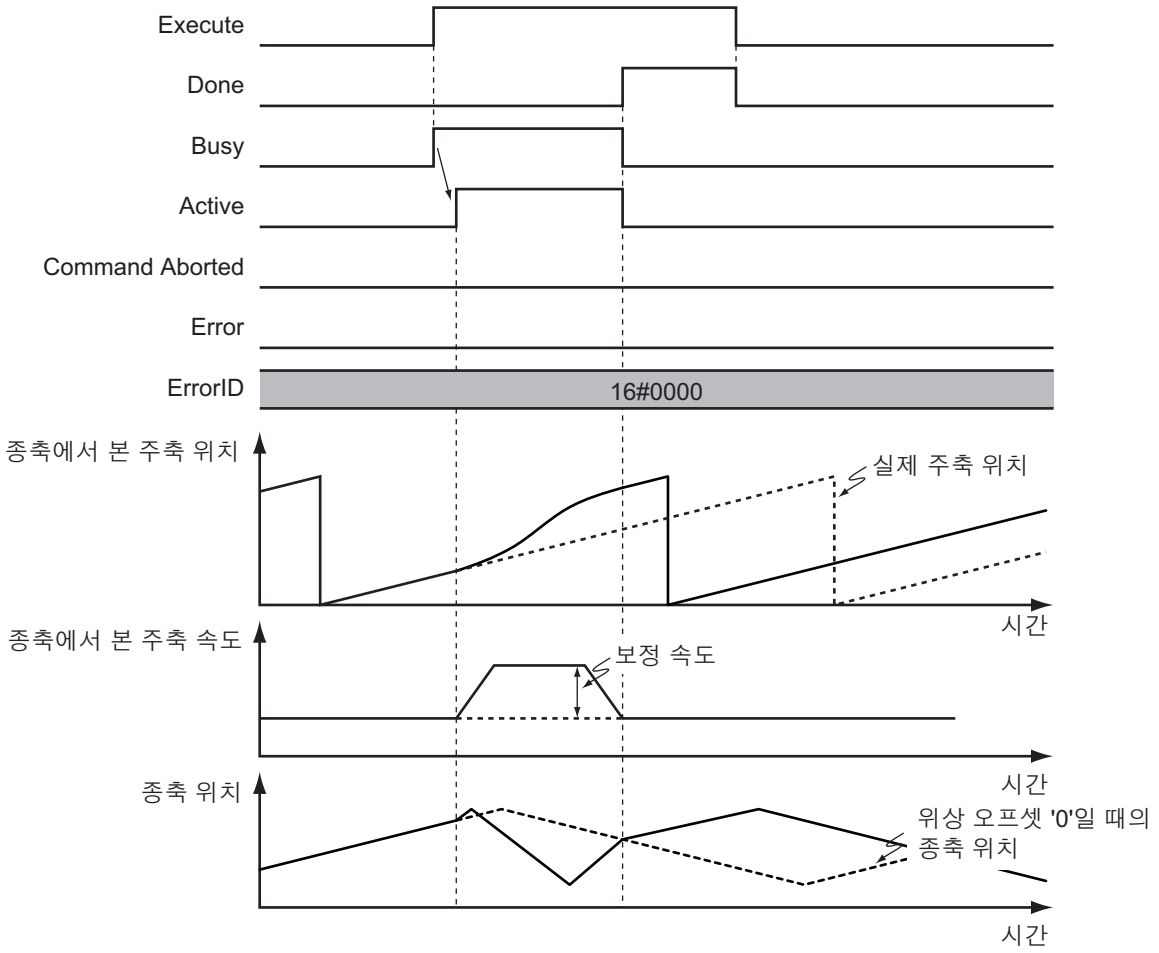
PhaseShift(위상 보정량)에는 Slave(종축)에서 본 Master(주축)의 위상 보정량을 지정합니다.
위상 보정량은 상대량을 지정합니다.

● Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도), Jerk(저크)를 지정할 수 있습니다.

위상 보정량의 목표 속도(보정 속도), 가속도, 감속도, 저크는 Velocity, Acceleration, Deceleration, Jerk로 지정합니다.
Slave(종축)에서 본 Master(주축)의 목표 속도(보정 속도)는 주축 속도에 대한 상대 속도가 됩니다.
Slave(종축)에서 본 Master(주축)의 목표 속도(보정 속도)는 위상 보정량, 가속도, 감속도, 저크에 따라 다음과 같습니다.

예) 전자 캠의 주축 상대값 위상을 보정하는 경우

Slave(종축)에서 본 Master(주축)의 보정 속도는 주축 속도에 대한 상대 속도가 됩니다.



- 위상 보정량에 '0'을 지정하면 Master(주축)의 위상 보정량 '0'으로 정상 종료합니다.
- 목표 속도(보정 속도)에 '0'을 지정하면 범위 외 지정이 되어 Slave(종축) 축 이상이 발생합니다.
- 지정한 목표 속도(보정 속도)와 주축 속도의 가산 속도는 Master(주축)의 최고 속도를 초과해 지정할 수 있습니다.

참고

MC_Phasing(주축 상대값 위상 보정) 명령의 결과는 동기한 Slave(종축)의 동작으로 이상 감지를 실행합니다.
따라서 명령 기동 시에는 이 명령의 Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)의 설정에 대한 이상 감지는 실행되지 않습니다.

● BufferMode(버퍼 모드 선택)

하나 이전의 축 동작과 이번 동작의 연결 방법을 지정합니다.
종류는 다음 1가지입니다.

버퍼 모드 선택	설 명
Aborting	이 명령을 재기동하면 바로 이 명령에 의한 보정을 시작합니다.

다중 기동에 의한 반전 시 동작은 주축에 설정된 [반전 시 동작 선택]에 따릅니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 인포지션 확인

위상 보정 동작이 완료되면 보정값 출력은 종료되고 인포지션 확인을 실행하지 않습니다.

모션 명령 재기동

이 명령 실행 중에 다시 명령 시작을 기동하면 PhaseShift(위상 보정량), Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도)을 변경합니다.
변경 동작은 상대 위치 결정과 동일합니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

Slave(종축)에서 이 명령 실행 중에 다른 명령에 의한 다중 기동은 Aborting으로 가능합니다.
Buffered, Blending에 의한 명령 다중 기동은 불가능합니다.

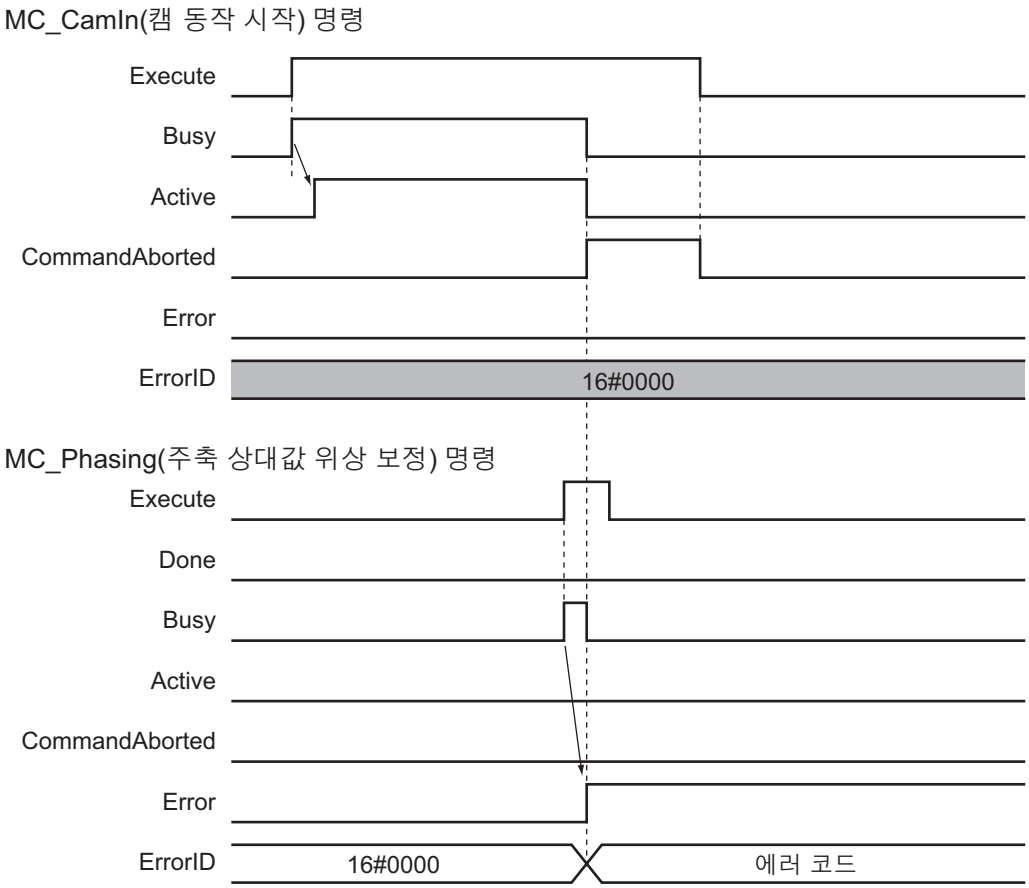
● 이 명령끼리의 다중 기동

지정한 종축이 MC_Phasing(주축 상대값 위상 보정) 명령 동작 중에 MC_Phasing(주축 상대값 위상 보정) 명령을 기동할 수 있습니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 종축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.
Slave(종축)가 이상 발생으로 인해 동기 상태가 해제된 시점에서 MC_Phasing(주축 상대값 위상 보정) 명령 동작은 CommandAborted(실행 중단)가 됩니다.
또한 주축의 이상 상태는 이 명령의 동작에 영향을 미치지 않습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_TorqueControl

서보 드라이버의 토크 제어 모드를 사용해 토크를 제어합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_TorqueControl	토크 제어	FB	MC_TorqueControl_instance MC_TorqueControl Axis — — Axis Execute — InTorque Torque — Busy TorqueRamp — Active Velocity — CommandAborted Direction — Error BufferMode — ErrorID	MC_TorqueControl_instance (Axis := 《파라메터》, Execute := 《파라메터》, Torque := 《파라메터》, TorqueRamp := 《파라메터》, Velocity := 《파라메터》, Direction := 《파라메터》, BufferMode := 《파라메터》, InTorque => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Torque	목표 토크	LREAL	0~1000.0	300.0	서보 드라이버로 출력할 목표 토크를 [0.1%] 단위로 지정합니다. 정격 토크를 '100.0%'로 했을 때의 비율로 지정합니다. *1 단위는 [%]입니다.
TorqueRamp	토크 램프	LREAL	양수 또는 '0'	0	현재 토크에서 목표 토크를 출력할 때까지의 토크 변화율을 지정합니다. 단위는 [%/s]입니다.
Velocity	제한 속도	LREAL	양수 또는 '0'	0	목표 속도를 지정합니다. 단위는 [명령 단위/s]입니다. *2
Direction	방향 선택	_eMC_ DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	0 *3	목표 토크의 방향을 지정합니다. 0 : 정방향 지정 2 : 역방향 지정
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered

*1. '1000.0%'를 넘는 값을 지정하면 '1000.0%'로 다루어집니다. 또한 음수값이 지정되면 '0.0%'로 다루어집니다.
*2. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*3. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
InTorque	목표 토크 도달	BOOL	TRUE, FALSE	목표 토크에 도달했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
InTorque	목표 토크를 출력 중	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때 • 명령을 재기동해 목표 토크를 변경했을 때
Busy	Execute의 기동	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	토크 명령값을 출력하기 시작했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 토크 명령값을 직접 지정해서 서보 모터의 출력 토크를 제어합니다.
- 목표 토크는 [0.1%] 단위로 지정합니다. 소수점 2자리 이하를 지정하면 소수점 2자리에서 반올림합니다.
- 이 명령의 실행을 정지시키려면 MC_Stop(강제 정지) 명령을 사용하십시오.
- 토크 제어는 서보 드라이버의 토크 제어 모드 기능을 사용해서 실행합니다.
- 제어 모드가 전환될 때까지는 바로 이전의 제어 모드를 유지합니다.
  예
  위치 제어→토크 제어로 전환될 때: 서보 드라이버가 토크 제어로 변할 때까지 위치 제어
  토크 제어→위치 제어로 전환될 때: 서보 드라이버가 위치 제어로 변할 때까지 토크 제어

● 데이터 오브젝트 매핑

MC_TorqueControl(토크 제어) 명령을 사용하려면 Sysmac Studio 축 기본 설정의 [고급 설정]에서 다음 오브젝트 데이터를 매핑하십시오.

- 목표 토크(6071Hex)
- 조작 모드(6060Hex)
- 피드백 토크(6077Hex)
- 조작 모드 표시(6061Hex)

이러한 데이터를 설정하지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다.

데이터 오브젝트 매핑에 대해서는 '2-3 PDO 매핑(2-27페이지)', "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)을 참조하십시오.

참고

- NX 시리즈 펄스 출력 유니트에 대해서는 이 명령을 사용할 수 없습니다.

명령에 대한 자세한 내용

다음에 이 명령에 대해 자세히 설명합니다.

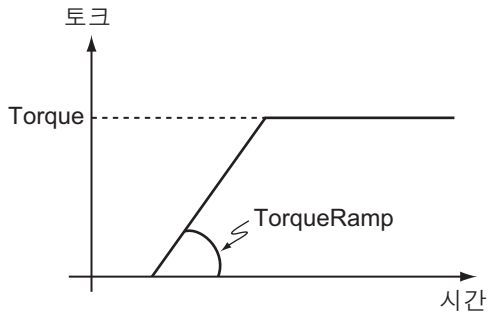
● Axis(축) 지정

Axis(축)에서 토크를 제어할 축을 지정합니다.

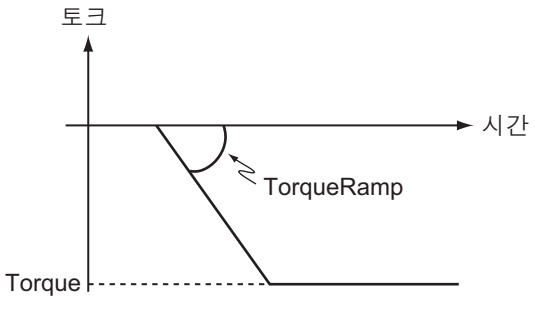
● TorqueRamp(토크 램프)

현재 지정된 명령 토크에서 목표 토크를 출력할 때까지의 경사를 지정합니다.

예1) 방향 지정 = 정방향인 경우



예2) 방향 지정 = 역방향인 경우



사용상의 주의 사항

목표 토크는 모터 최대 토크보다 작은 값이 되도록 설정하십시오.
모터 최대 토크를 초과해서 설정한 경우의 동작은 서보 드라이버에 따릅니다.

● Velocity(제한 속도)

토크 제어 시의 축 최대 속도를 제한합니다.
축 속도가 이 제한 속도값에 이르면 서보 드라이버로 토크를 떨어뜨려서 축의 속도를 낮춥니다.
제한 속도값의 기능은 서보 드라이버의 기능을 사용해서 실행합니다.

자세한 내용은 "각 서보 드라이버 설명서의 토크 제어 기능"을 참조하십시오.

사용상의 주의 사항

- 토크 제어 시에는 축의 속도가 빨라집니다. 안전을 위해 반드시 Velocity(제한 속도)를 설정하십시오.
- 오므론의 서보 드라이버 G5 시리즈를 사용할 경우에는 G5 서보 드라이버의 '속도 제한 선택(3317 Hex)'을 '1: EtherCAT 통신으로부터의 속도 제한'으로 반드시 설정하십시오.
  설정되어 있지 않으면 속도 제한값이 기능하지 않습니다. 또한 한계 입력 신호가 ON이 되어도 축이 정지하지 않습니다.
- 속도 제한값은 프로세스 데이터 607F Hex를 사용합니다.
  오므론의 서보 드라이버 G5 시리즈를 사용할 경우에는 Sysmac Studio 축 파라메터 설정 화면의 '고급 설정'에서 속도 제한값(607F Hex)을 사용하도록 설정하십시오.
  또한 타사의 서보 드라이버로 속도 제한 기능을 사용하려면 각 회사의 서보 드라이버 설명서를 참조하십시오.

● Direction(방향 선택)

목표 토크를 출력할 방향을 지정합니다.
축을 정방향으로 출력하려면 정방향을, 축을 역방향으로 출력하려면 역방향을 지정합니다.

● BufferMode(버퍼 모드 선택)

하나 이전의 축 동작과 이번 동작의 연결 방법을 지정합니다.
종류는 다음 2가지입니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다.
Buffered	현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 토크 제어 중의 축 정지

오므론의 서보 드라이버 G5 시리즈를 사용한 경우 이 명령 실행 중에 MC_Stop(강제 정지) 명령을 기동하면 MC_Stop(강제 정지) 명령의 감속도는 적용되지 않고 즉시 정지합니다.
또한 감속 정지로 다루어지는 이상이 발생하더라도 즉시 정지로 다루어집니다.

● 토크 제어 중의 명령 현재 위치, 피드백 현재 위치

이 명령에 의한 토크 제어 중 모션 제어 시스템 변수로 취득할 수 있는 현재 위치는 다음과 같습니다.

피드백 현재 위치	: 서보 드라이버에서 취득한 값을 기어비로 변환한 값이 됩니다.
명령 현재 위치	: 1주기 전의 피드백 현재 위치와 동일한 값이 됩니다.

● 사용 가능 축과 기동 조건

- 서보 축에서 MC_Power(운전 가능) 명령의 Enable(유효)이 TRUE(서보 ON 상태)일 때 이 명령을 기동할 수 있습니다.
- 가상 서보 축은 항상 수신할 수 있습니다. 단, 서보 드라이버의 제어 모드 전환 처리는 실행하지 않습니다.
- 엔코더 축, 가상 엔코더 축에서는 기동 시 이상이 됩니다.

● 서보 OFF 시의 대응

MC_Power(운전 가능)의 출력 변수 'Status(운전 가능)'가 FALSE가 되면 MC 기능 모듈은 CSP 모드로 전환 처리를 실행합니다.

단, 오므론의 서보 드라이버 G5 시리즈는 서보 OFF 상태에서는 MC 기능 모듈로부터의 제어 모드 전환을 수신하지 않습니다.

● 축 변수의 Status

축 변수의 Status는 Status.Continuous(연속 정지 중)가 TRUE가 됩니다.

또한 축 변수 DrvStatus(서보 드라이버 상태)는 CST(주기적 동기 토크(CST) 모드 중)가 TRUE가 됩니다.

● 원점 상태

원점 확정 상태가 계속됩니다.

● 소프트웨어 리미트

소프트웨어 리미트는 적용됩니다.

축 파라메터에서 [명령 위치에 대해 유효. 감속 정지] 또는 [명령 위치에 대해 유효. 편차 펄스 정지]를 선택한 경우에도 적용됩니다.

● 카운트 모드가 [리니어 모드]인 경우

오버 플로우 또는 언더 플로우한 경우 목표 위치가 없는 동작으로 다루어집니다.

● 반전 시 동작

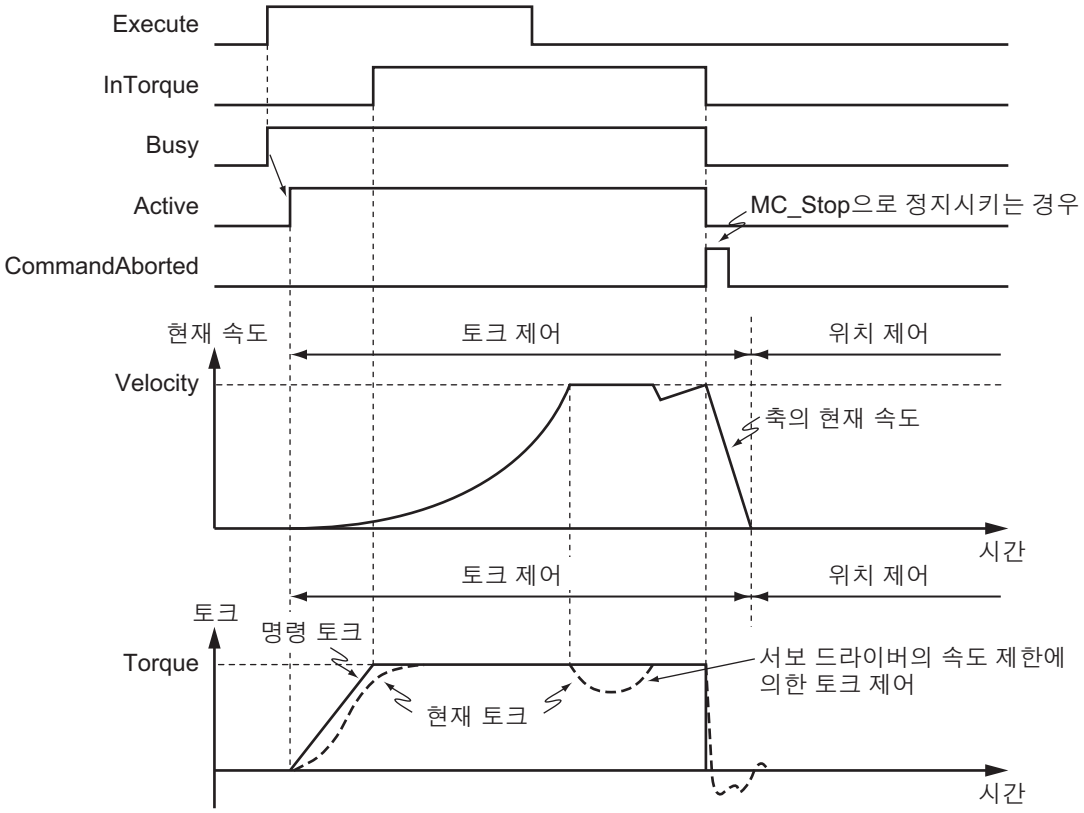
다중 기동을 했을 때에 토크 명령값이 반전하면 축 파라메터의 [반전 시 동작 선택]의 설정 내용에 관계없이 이 명령의 TorqueRamp(토크 램프)로 동작합니다.

또한 다중 기동에 의한 반전 시 동작은 다음과 같습니다.

- 이 명령 동작 중에 CSP를 사용하는 명령의 다중 기동으로 명령 위치가 반전하면 축 파라메터의 [반전 시 동작 선택]에 따라 반전 동작을 실행합니다.
- CSP나 CSV를 사용하는 명령 동작 중에 이 명령의 다중 기동으로 토크 명령값이 반전하면 TorqueRamp(토크 램프)에 따라서 토크 명령을 반전시킵니다.
- 이 명령 동작 중에 이 명령의 다중 기동으로 토크 명령값이 반전하면 TorqueRamp(토크 램프)에 따라서 토크 명령을 반전시킵니다.

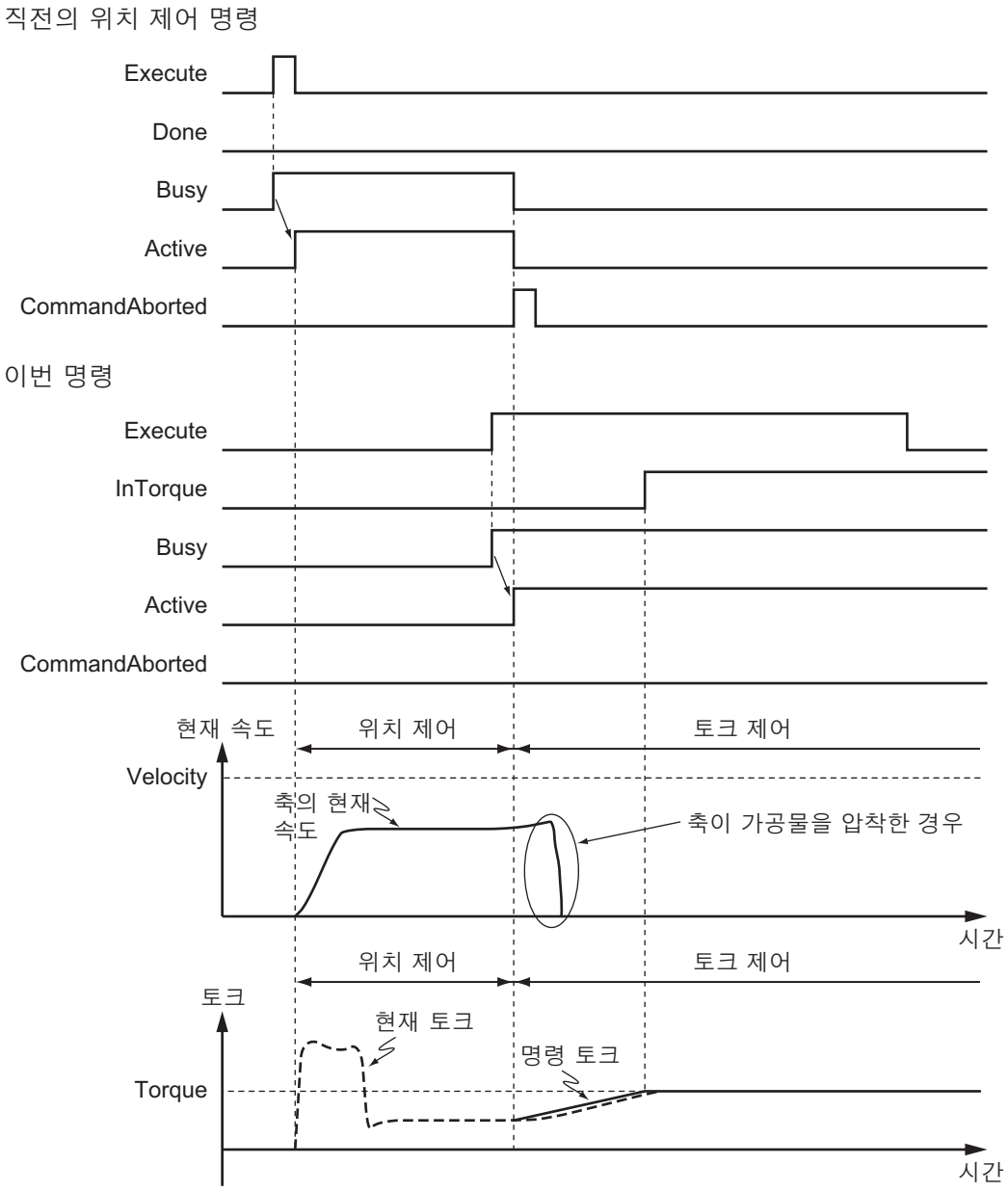
타이밍 차트

이 명령을 기동하고 정지시킨 경우



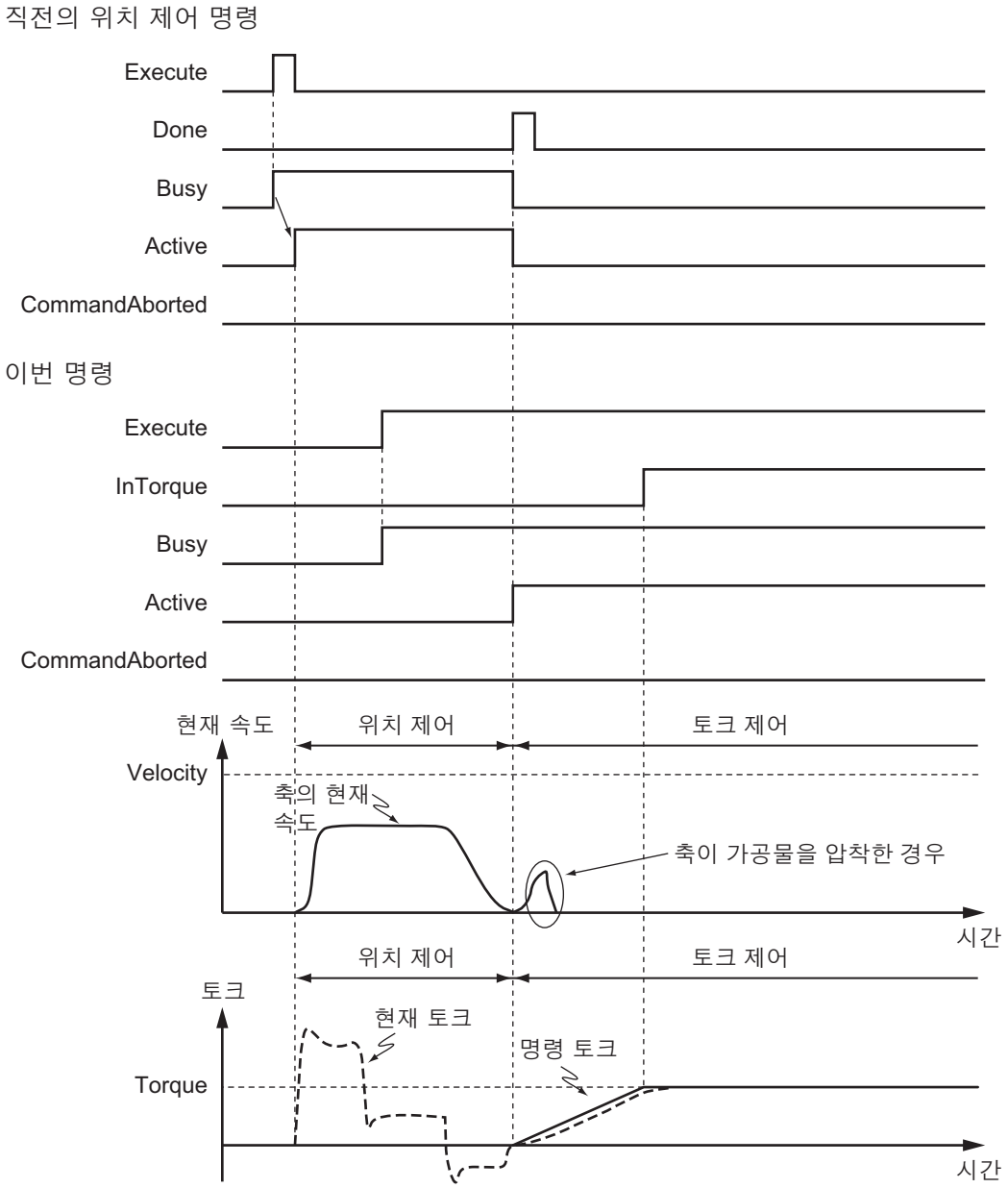
하나 이전의 동작 중에 중지로 기동하는 경우

이 명령 실행 중에 축이 정지하는 프레스 응용 프로그램으로 설명합니다.



하나 이전의 동작 중에 Buffered로 기동하는 경우

이 명령 실행 중에 축이 정지하는 프레스 응용 프로그램으로 설명합니다.



● 제어 모드 전환

- MC_MoveAbsolute(절대값 위치 결정) 명령, MC_MoveRelative(상대값 위치 결정) 명령 등의 위치 제어 명령 실행 중에 MC_TorqueControl(토크 제어) 명령을 실행하면 BufferMode(버퍼 모드 선택)의 설정에 따라 동작이 달라집니다.
  BufferMode(버퍼 모드 선택)가 Aborting일 때는 실행한 시점에서 토크 제어로 전환됩니다. Buffered일 때는 이전 동작이 완료된 시점에서 토크 제어로 전환됩니다.
- MC_TorqueControl(토크 제어) 명령 실행 중에 MC_MoveAbsolute(절대값 위치 결정) 명령 등의 다른 명령으로 중지되거나 축에 이상이 발생하면 그 시점에서 위치 제어로 전환됩니다.
- 명령 실행 타이밍에 따라 Active(제어 중)는 전환되지만, 서보 드라이버 측의 제어 모드가 전환될 때까지는 몇 주기가 필요합니다. 제어 모드가 변환될 때까지의 시간은 서보 드라이버에 따릅니다.

제어 모드 전환 판정

오므론의 서보 드라이버 G5 시리즈에서 축을 정지할 경우 MC 기능 모듈은 속도 제한값(607F Hex)을 '0'으로 합니다. 그 후 프라이머리 정주기 태스크에서 3회 연속 피드백 현재 속도가 다음 판정식을 충족하면 CSP로 전환합니다.

피드백 현재 속도≦(최고 속도)×0.1

타사 서보 드라이버의 경우 서보 드라이버의 제어 모드를 CST에서 CSP로 전환하는 동시에 전환 처리 시의 피드백 현재 위치에서 서보를 잠급니다.

제어 모드 전환 실패

MC 기능 모듈이 서보 드라이버에 대해 제어 모드 전환 명령을 실행한 후 1초 경과해도 서보 드라이버의 전환이 완료되지 않으면 '서보 드라이버의 제어 모드 전환 이상(에러 코드: 7439 Hex)'이 되어 서보를 OFF합니다(프리 런 정지).
'서보 드라이버의 제어 모드 전환 이상(에러 코드: 7439 Hex)'에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

또한 속도 제한값을 '0'으로 한 뒤 10초 경과해도 전환 판정식이 성립하지 않는 경우도 위와 동일하게 서보를 OFF합니다.

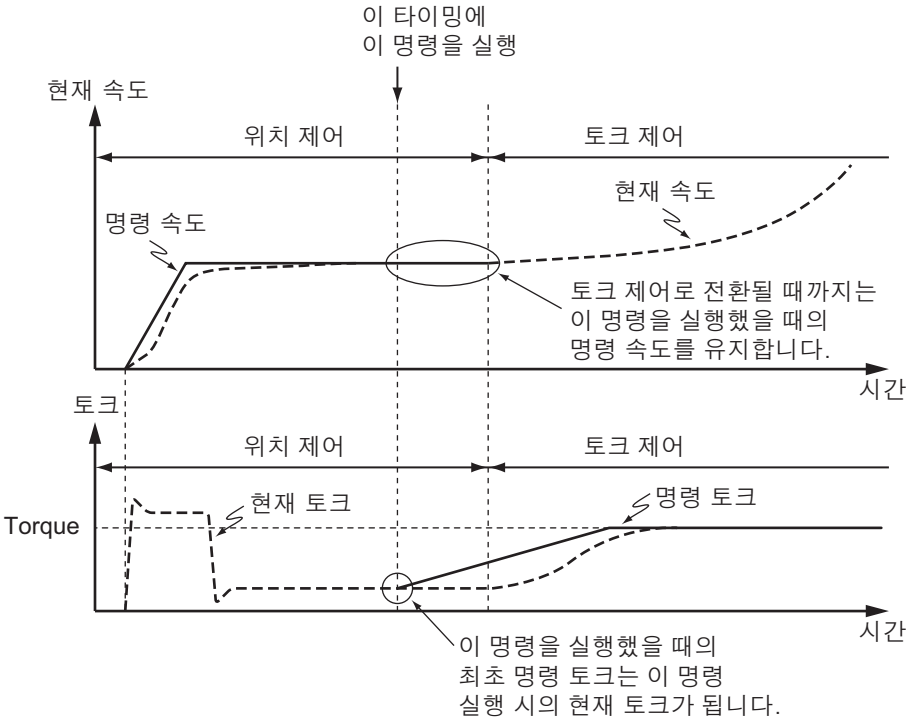
제어 모드 전환 동작의 예

제어 모드가 전환될 때까지의 명령 토크와 명령 속도의 관계를 축 동작 중에 전환한 경우의 예를 이용해서 설명합니다.

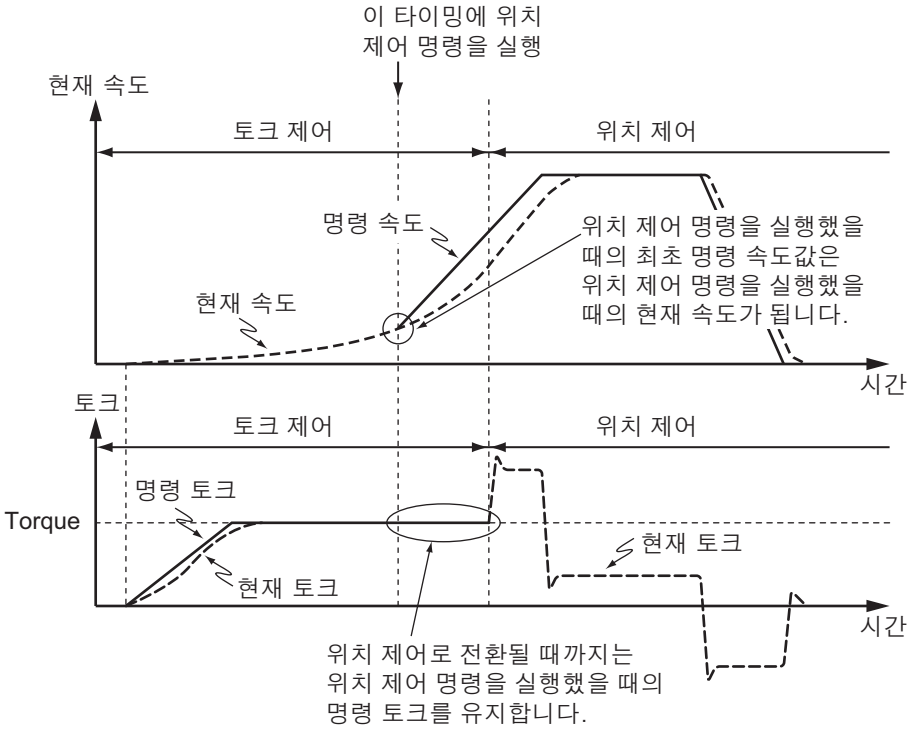
사용상의 주의 사항

축이 동작하고 있을 때 서보 드라이버 측에서 제어 모드를 전환하면 서보 드라이버에 따라서는 이상이 발생하는 경우가 있습니다.

위치 제어에서 토크 제어로 전환할 때



토크 제어에서 위치 제어로 전환할 때



모션 명령 재기동

토크 제어 동작 중에 입력 파라메터를 변경하고 Execute(기동)를 다시 TRUE로 하면 이 명령의 동작을 변경할 수 있습니다.

모션 명령 재기동으로 변경 가능한 입력 변수는 Torque(목표 토크), TorqueRamp(토크 램프), Velocity(제한 속도)입니다.

모션 명령 재기동을 통해 Torque(목표 토크)를 변경하는 경우 InTorque(목표 토크 도달)는 재기동에 의해 설정된 새로운 목표 토크에 대해 동작합니다.

모션 명령 재기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다.
각 축에 1개까지 버퍼링할 수 있습니다.
다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다.
Buffered	현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

이 명령 실행 중에 다른 명령에 의한 다중 기동 실행은 다른 명령의 BufferMode(버퍼 모드 선택)에 Aborting 또는 Buffered를 지정했을 때만 가능합니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_SetTorqueLimit

서보 드라이버의 토크 제한 기능을 사용해 서보 드라이버 출력 토크를 제한합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_SetTorqueLimit	토크 제한	FB	MC_SetTorqueLimit_instance MC_SetTorqueLimit Axis — — Axis Enable — Enabled PositiveEnable — Busy PositiveValue — Error NegativeEnable — ErrorID NegativeValue	MC_SetTorqueLimit_instance (Axis := 《파라메터》, Enable := 《파라메터》, PositiveEnable := 《파라메터》, PositiveValue := 《파라메터》, NegativeEnable := 《파라메터》, NegativeValue := 《파라메터》, Enabled => 《파라메터》, Busy => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Enable	유효	BOOL	TRUE, FALSE	FALSE	TRUE인 동안 명령을 실행합니다.
PositiveEnable	정방향 유효	BOOL	TRUE, FALSE	FALSE	TRUE: 정방향 토크 제한을 유효로 합니다. FALSE: 정방향 토크 제한을 무효로 합니다.
PositiveValue	정방향 토크 제한값	LREAL	0.1~1000.0, 또는 '0.0'	300.0	정방향에 대한 토크 제한값을 0.1% 단위로 지정합니다. 축 파라메터의 '정방향 토크 제한 상한값'을 넘는 값이 입력되면 [정방향 토크 제한 상한값]이 됩니다. '0' 및 음수를 지정하면 '0'으로 동작합니다.
NegativeEnable	역방향 유효	BOOL	TRUE, FALSE	FALSE	TRUE : 역방향 토크 제한을 유효로 합니다. FALSE : 역방향 토크 제한을 무효로 합니다.
NegativeValue	역방향 토크 제한값	LREAL	0.1~1000.0, 또는 '0.0'	300.0	역방향에 대한 토크 제한값을 0.1% 단위로 지정합니다. 축 파라메터의 '역방향 토크 제한 상한값'을 넘는 값이 입력되면 [역방향 토크 제한 상한값]이 됩니다. '0' 및 음수를 지정하면 '0'으로 동작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Enabled	유효	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다. *1
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*2	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. Enabled(유효)는 이 명령의 상태를 나타내는 것으로, 서보 드라이버에 의한 토크 제한 상태를 나타내는 것이 아닙니다.
*2. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Enabled	• Enable이 TRUE가 되었을 때 • MC_Power 명령이 실행 중일 때	• Enable이 FALSE가 된 다음 1주기 후 • Error가 TRUE가 되었을 때 • MC_Power 명령의 Enable이 FALSE가 되었을 때
Busy	Enable이 TRUE가 되었을 때	• Error가 TRUE가 되었을 때 • Enable이 FALSE가 되었을 때
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	−	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 이 명령은 서보 드라이버에 대한 토크 제한값을 설정합니다.
- Enable(유효)이 TRUE인 상태에서
  PositiveEnable(정방향 유효)을 TRUE로 하면 PositiveValue(정방향 토크 제한값)로 제한합니다.
  NegativeEnable(역방향 유효)을 TRUE로 하면 NegativeValue(역방향 토크 제한값)로 제한합니다.
- PositiveEnable(정방향 유효)을 FALSE로 하면 축 파라메터의 [정방향 토크 제한 상한값]을 서보 드라이버에 설정합니다.
  마찬가지로 NegativeEnable(역방향 유효)을 FALSE로 한 경우도 축 파라메터의 [역방향 토크 제한 상한값]을 서보 드라이버에 설정합니다.
- 이 명령의 Enable(유효)을 FALSE로 하면 [정방향 토크 제한 상한값]과 [역방향 토크 제한 상한값]을 서보 드라이버에 설정합니다. 동시에 Busy(실행 중), Enabled(유효)를 FALSE로 합니다.
- 토크 제한값은 모터 토크에 대해 0.1% 단위로 설정할 수 있습니다. 소수점 2자리 이하를 지정하면 소수점 2자리에서 반올림합니다.

사용상의 주의 사항

- 축 파라메터의 [정방향 토크 제한 상한값]과 [역방향 토크 제한 상한값]은 사용하는 서보 드라이버에 정해진 토크 제한의 상한값을 설정하십시오.
- NX 시리즈 펄스 출력 유니트에 대해서는 이 명령을 사용할 수 없습니다.

● 데이터 오브젝트 매핑

MC_SetTorqueLimit(토크 제한) 명령을 사용하는 경우 Sysmac Studio 축 기본 설정의 [고급 설정]에서 다음 오브젝트 데이터를 매핑하십시오.

- 정회전 측 토크 제한(60E0 Hex)
- 역회전 측 토크 제한(60E1 Hex)

이러한 데이터를 설정하지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다.

사용상의 주의 사항

- 타사의 서보 드라이버를 사용하는 경우 정회전 측 토크 제한(60E0 Hex), 역회전 측 토크 제한(60E1 Hex)이 토크 제한값이 되도록 서보 드라이버를 설정하십시오.
  설정 방법에 대해서는 "각 서보 드라이버 설명서"를 참조하십시오.
- 타사 서보 드라이버로 정회전 측 토크 제한, 역회전 측 토크 제한을 PDO에 매핑할 수 없는 경우는 이 명령을 사용할 수 없습니다.
  이 경우 각 회사의 지원 툴이나 SDO 통신을 사용해 토크 제한값을 설정하십시오.

데이터 오브젝트 매핑에 대해서는 '2-3 PDO 매핑(2-27페이지)', "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 입력 파라메터 재기입

Enable(유효)이 TRUE일 때는 다음 입력 변수값을 항상 변경할 수 있습니다.

- PositiveEnable(정방향 유효)
- NegativeEnable(역방향 유효)
- PositiveValue(정방향 토크 제한값)
- NegativeValue(역방향 토크 제한값)

● MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령과 유지 동작의 관계(오므론의 서보 드라이버 G5 시리즈를 사용하는 경우)

원점 복귀의 [12: 근방 퇴피 · 유지 시간 지정], [13: 원점 근방 입력 없음 · 유지 원점 입력 지정]은 서보 드라이버에 미리 설정된 토크 제한값을 이용해 유지 방향으로 토크 제한을 자동으로 시작합니다.



사용상의 주의 사항

타사의 서보 드라이버를 사용하면 MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령에서의 자동 토크 제한 기능은 기능하지 않습니다.
MC_SetTorqueLimit(토크 제한) 명령, SDO 통신, 서보 드라이버의 지원 툴 등을 사용해 적절한 값을 설정하십시오.



참고

- 원점 복귀가 정상 완료된 후에도 토크 제한은 계속됩니다.
- 역방향으로 이동하는 명령이 기동되면 토크 제한이 자동으로 해제됩니다.

원점 복귀에 대한 자세한 내용은 🕮 'MC_Home(3-15페이지)' 또는
🕮 "MC_HomeWithParameter(3-36페이지)"를 참조하십시오.

● 오므론의 서보 드라이버 G5 시리즈 설정

이 명령을 사용하려면 서보 드라이버의 지원 소프트웨어를 사용해 G5 서보 드라이버의 토크 리미트 선택(3521 Hex)을 설정해야 합니다.

- 원점 복귀 유지 동작 시에 원점 검출 방향에 대해 토크를 제한하고, 그 밖의 동작은 이 명령의 토크 제한 방향이나 토크 제한값을 사용할 경우 '6'을 사용하십시오.
  이 경우 원점 복귀 유지 동작 중에는 이 명령의 입력 변수 내용은 무시됩니다.
- '4'로 설정하면 항상 이 명령의 입력 변수 내용이 적용됩니다. 원점 복귀 유지 동작 시와 그 밖의 경우에 따라 적절한 토크 제한값을 설정해야 합니다.

		토크 리미트 선택(3521 Hex)	
		6(권장)	4
정방향 토크 제한값	원점 복귀 중 *1	제3 토크 리미트(3525 Hex)가 적용됩니다.	이 명령의 PositiveValue(정방향 토크 제한값)와 제1 토크 리미트(3013 Hex) 중 작은 것이 적용됩니다.
	원점 복귀 이외	이 명령의 PositiveValue(정방향 토크 제한값)와 제1 토크 리미트(3013 Hex) 중 작은 것이 적용됩니다.	
역방향 토크 제한값	원점 복귀 중 *1	제4 토크 리미트(3526 Hex)가 적용됩니다.	이 명령의 NegativeValue(역방향 토크 제한값)와 제2 토크 리미트(3522 Hex) 중 작은 것이 적용됩니다.
	원점 복귀 이외	이 명령의 NegativeValue(역방향 토크 제한값)와 제2 토크 리미트(3522 Hex) 중 작은 것이 적용됩니다.	

*1. 토크 제한 자동 해제까지

자세한 내용에 대해서는 🕮 "G5 시리즈 AC 서보 모터/드라이버 사용 설명서(EtherCAT 통신 내장 타입)(SBCE-365)", 🕮 "G5 시리즈 리니어 모터/드라이버 사용 설명서(EtherCAT 통신 내장 리니어 모터 타입)(SBCE-366)"를 참조하십시오.

● MC_TorqueControl(토크 제어) 명령과의 관계

MC_SetTorqueLimit(토크 제한) 명령과 MC_TorqueControl(토크 제어) 명령은 조합해서 사용할 수 있습니다.

● 축 그룹에 속하는 축에 대한 사용

유효한 축 그룹에 속한 축에 대해 이 명령을 사용할 수 있습니다.

● CPU 유니트 동작 모드와의 관계

운전 모드 중에 이 명령으로 설정한 내용은 프로그램 모드로 변경되어도 지속됩니다.

● 사용 가능한 축과 기동 조건

- 서보 축에서 이 명령을 사용할 수 있습니다.
  단, 서보 상태에 따라 이 명령의 출력 변수 'Enabled(유효)'의 상태가 다릅니다.

	서보 ON 상태	서보 OFF 상태
Enabled(유효) 상태	TRUE	FALSE *1

*1. 이 명령의 Enabled(유효)가 FALSE이면 서보 드라이버의 토크 제한은 기능하지 않습니다.

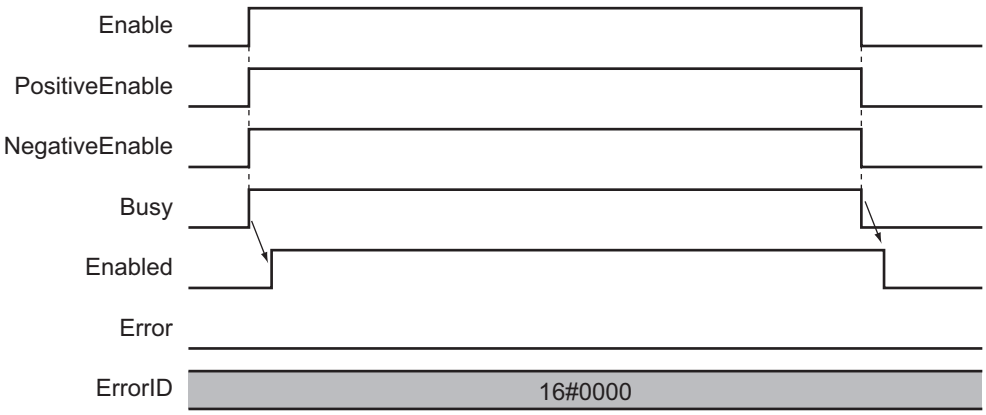
- 가상 서보 축에서도 명령은 수신할 수 있지만 토크 제한은 실행하지 않습니다.
- 엔코더 축, 가상 엔코더 축에서는 기동 시 이상이 됩니다.

● 축 변수의 Status(서보 드라이버 상태)

서보 드라이버의 내부 제한 기능이 작용하면 축 변수의 DrvStatus(서보 드라이버 상태)의 ILA(드라이버 내부 기능 제한 중)가 TRUE가 됩니다.
이 변수는 서보 드라이버의 토크 제한, 속도 제한, 구동 금지 입력, 소프트웨어 리미트 4개의 OR 출력입니다.

타이밍 차트

토크 제한을 실행한 경우의 타이밍 차트는 다음과 같습니다.



모션 명령 재기동

입력이 'Enable 형식'인 명령에는 모션 명령 재기동은 없습니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

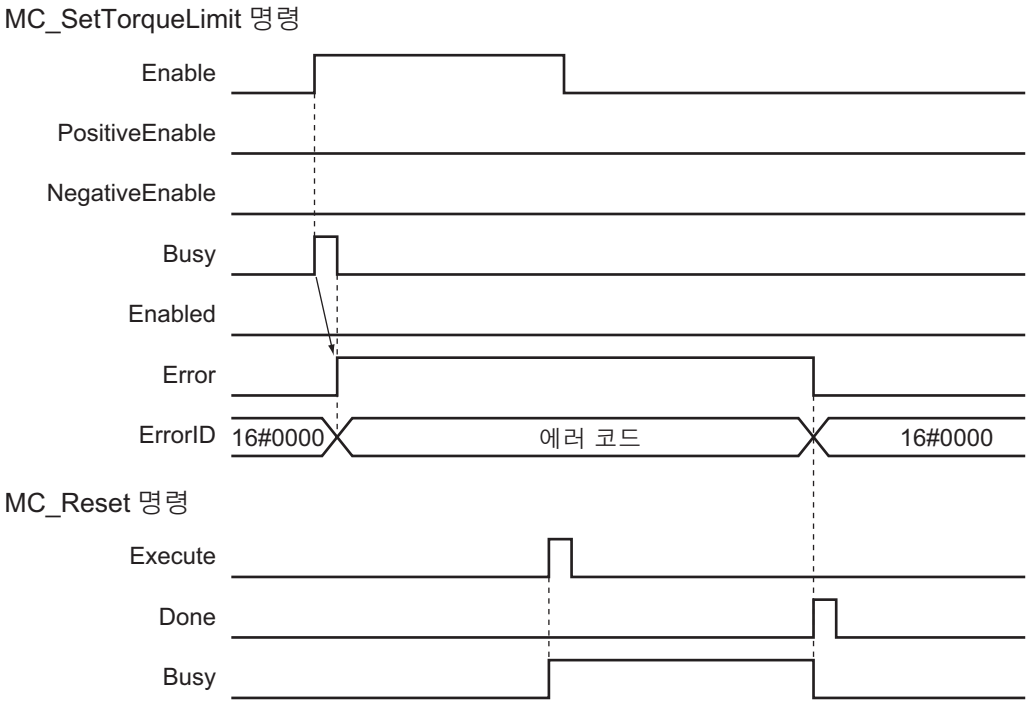
● 이 명령끼리의 다중 기동

이 명령을 실행 중인 축에 이 명령의 다른 인스턴스를 기동하면 나중에 실행된 인스턴스가 우선 처리됩니다.
Enabled(유효)는 둘 다 TRUE가 됩니다.
구체적인 동작으로는 나중에 실행된 인스턴스의 토크 제한값이 유효가 됩니다. 나중에 실행된 인스턴스의 Enable(유효)을 FALSE로 하면 토크 제한은 무효가 됩니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_ZoneSwitch

축의 명령 현재 위치 또는 피드백 현재 위치가 지정된 범위 내에 존재하는지를 판정합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_ZoneSwitch	존 감시	FB	MC_ZoneSwitch_instance MC_ZoneSwitch Axis — — Axis Enable — Enabled FirstPosition — InZone LastPosition — Busy ReferenceType — Error — ErrorID	MC_ZoneSwitch_instance (Axis := 《파라메터》, Enable := 《파라메터》, FirstPosition := 《파라메터》, LastPosition := 《파라메터》, ReferenceType := 《파라메터》, Enabled => 《파라메터》, InZone => 《파라메터》, Busy => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Enable	유효	BOOL	TRUE, FALSE	FALSE	TRUE인 동안 명령을 실행합니다.
FirstPosition	시작 위치	LREAL	음수, 양수, '0'	0	존 범위의 시작 위치를 지정합니다.*1 단위는 [명령 단위]입니다. *2
LastPosition	종료 위치	LREAL	음수, 양수, '0'	0	존 범위의 종료 위치를 지정합니다.*3 단위는 [명령 단위]입니다. *1
ReferenceType	위치 종류 선택	_eMC_ REFERENCE_ TYPE	0: _mcCommand 1: _mcFeedback	0 *4	감시할 축 정보를 지정합니다. 0 : 명령 위치(직전 프라이머리 주기에서의 계산값) 1 : 피드백 위치(동일 프라이머리 주기에서의 취득값)

*1. 종료 위치보다 작은 값을 지정하십시오.
*2. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*3. 시작 위치보다 큰 값을 지정하십시오.
*4. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Enabled	유효	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
InZone	존 내	BOOL	TRUE, FALSE	축 위치가 존 범위 이내일 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Enabled	Enable이 TRUE가 되었을 때	Enable이 FALSE가 되었을 때
InZone	존 내에 들어갔을 때	• 존에서 나왔을 때 • Enable이 FALSE가 되었을 때
Busy	Enable의 기동	• Error가 TRUE가 되었을 때 • Enable이 FALSE가 되었을 때
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 이 명령이 Enable(유효)인 상태에서 축의 명령 위치 또는 피드백 현재 위치가 FirstPosition(시작 위치) 이상, LastPosition(종료 위치) 이하인 존 내에 존재하면 출력 변수 InZone(존 내)이 TRUE가 됩니다. RererenceType(위치 종류 선택)에서 감시 대상의 축 정보를 명령 위치 또는 피드백 위치 중 하나로 설정할 수 있습니다.
- 모든 축 종류에서 존을 감시할 수 있습니다.
- Enable(유효)이 TRUE인 상태에서 FirstPosition(시작 위치)이나 LastPosition(종료 위치)을 재기입하면 재기입한 주기부터 새로운 값이 적용됩니다.
- 1축에 대해 복수의 존을 설정할 수 있으며, 존이 중복되어도 설정 가능합니다. 또한 소프트웨어 리미트 범위 밖에도 존을 설정할 수 있습니다.

사용상의 주의 사항

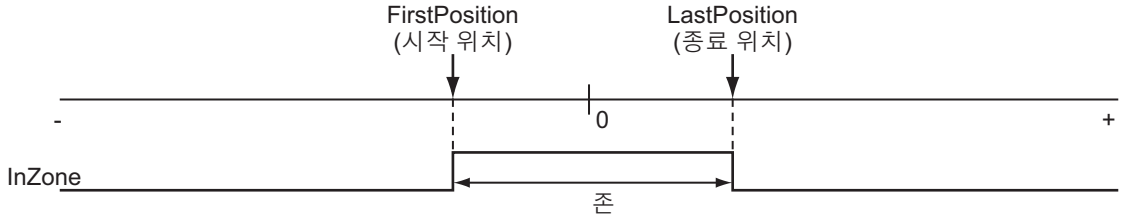
FirstPosition(시작 위치), LastPosition(종료 위치)이 딱 떨어지지 않는 나눗셈 결과 등을 포함하고 있으면 오차로 인해 처리 결과가 기대와 다른 경우가 있습니다.

명령에 대한 자세한 내용

FisrtPosition(시작 위치)과 LastPosition(종료 위치)은 카운트 모드에 따라 다음 관계가 성립하도록 설정하십시오. 관계가 성립하지 않으면 이상이 됩니다.

● 리니어 모드

FirstPosition(시작 위치) ≦ LastPosition(종료 위치)이 되도록 지정하십시오.

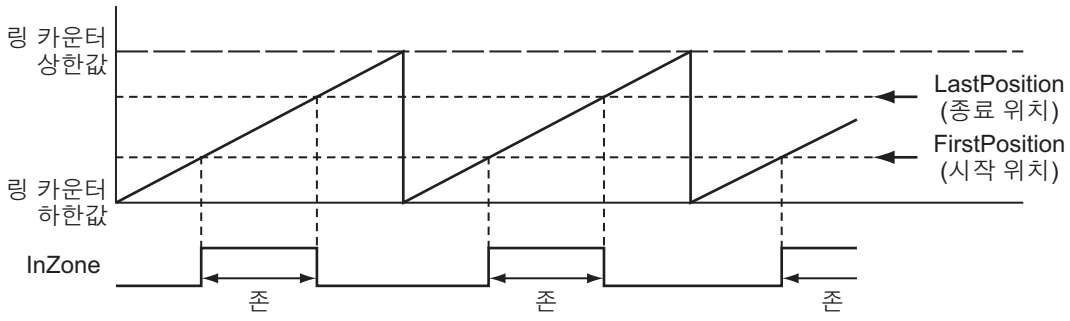


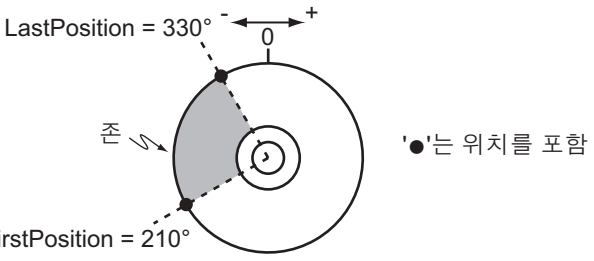
● 로터리 모드

[로터리 모드]인 경우는 링 카운터의 상하한 위치를 포함하는 경우와 포함하지 않는 경우에 다음과 같이 다릅니다.

상하한 위치를 포함하지 않는 경우

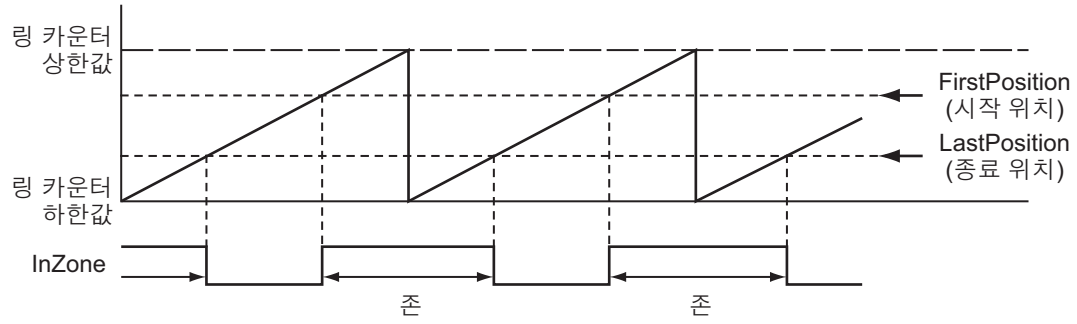
FirstPosition(시작 위치) ≦ LastPosition(종료 위치)이 되도록 지정하십시오.

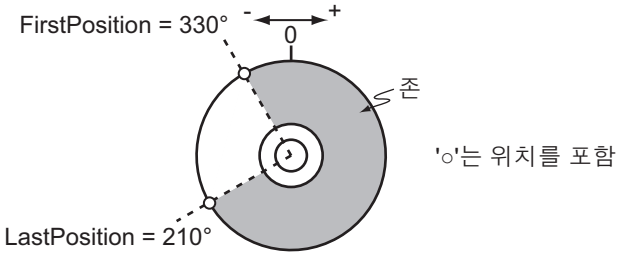




상하한 위치를 포함하는 경우

FirstPosition(시작 위치) > LastPosition(종료 위치)이 되도록 지정하십시오.





● ReferenceType(위치 종류 선택)

위치 종류를 다음에서 선택할 수 있습니다.

- _mcCommand: 명령 위치(직전 프라이머리 주기에서의 계산값)
  현재 주기에 대해 직전의 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  종축의 명령 위치를 계산하기 직전의 프라이머리 정주기 태스크에서 계산한 주축의 명령 위치를 사용합니다.
- _mcFeedback: 동일 프라이머리 주기에서의 취득값
  동일 프라이머리 주기에서 취득한 주축의 피드백 위치를 사용합니다.

● 축 종류와 위치 종류의 관계

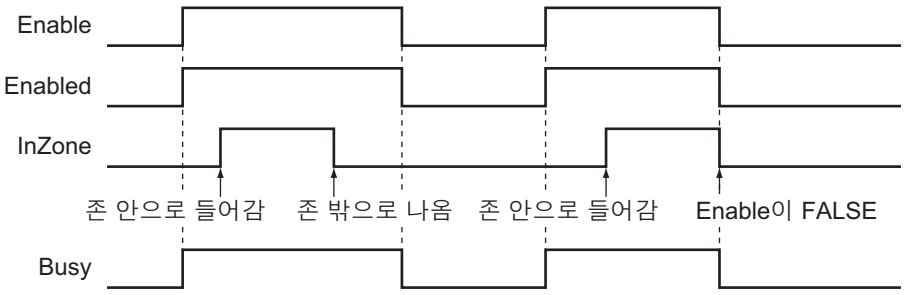
감시 가능한 축 종류와 감시하는 위치 종류의 관계는 다음과 같습니다.

축 종류	ReferenceType	
	_mcCommand	_mcFeedback
서보 축	○	○
엔코더 축	× *1	○
가상 서보 축	○	○
가상 엔코더 축	× *1	○

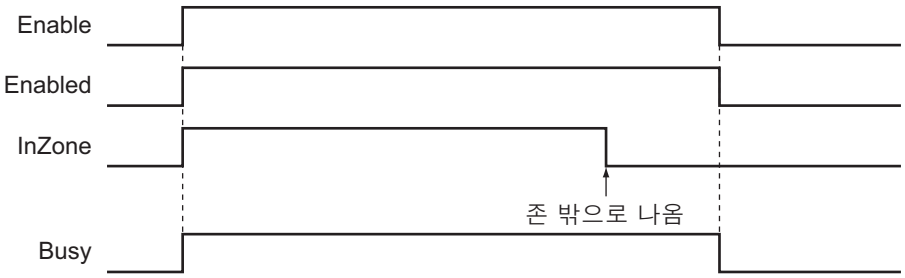
*1. 명령 기동 시에 '위치 종류 선택 범위 외(에러 코드: 5430 Hex)'가 발생합니다.

타이밍 차트

● 동작 중에 존 안에 들어간 경우 및 존 안에서 Enable(유효)이 FALSE가 된 경우



● 명령 기동 전에 존 안에 위치하고 도중에 존에서 나간 경우



모션 명령 재기동

입력이 'Enable 형식'인 명령에는 모션 명령 재기동은 없습니다.

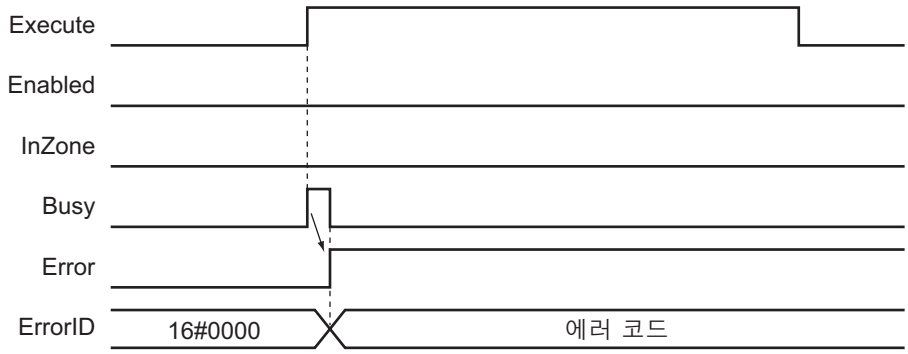
모션 명령 다중 기동

이 명령은 각 명령마다 독립적으로 동작하므로 모션 명령 다중 기동의 제약은 없습니다.
모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

이상

이 명령을 기동하지 못한 경우 이 명령은 이상이 되고 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_TouchProbe

트리거 신호가 발생하면 축 위치를 기록합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_TouchProbe	외부 래치 유효	FB	MC_TouchProbe_instance MC_TouchProbe Axis — — Axis TriggerInput — — TriggerInput TriggerVariable — — TriggerVariable Execute — Done WindowOnly — Busy FirstPosition — RecordedPosition LastPosition — CommandAborted ReferenceType — Error StopMode — ErrorID	MC_TouchProbe_instance (Axis := 《파라메터》, TriggerInput := 《파라메터》, TriggerVariable := 《파라메터》, Execute := 《파라메터》, WindowOnly := 《파라메터》, FirstPosition := 《파라메터》, LastPosition := 《파라메터》, ReferenceType := 《파라메터》, StopMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, RecordedPosition => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
WindowOnly	창 유효	BOOL	TRUE, FALSE	FALSE	창 마스크의 유효/무효를 지정합니다.
FirstPosition	시작 위치	LREAL	음수, 양수, '0'	0	트리거 신호를 시작할 위치를 지정합니다. ReferenceType(위치 종류 선택)에서 지정한 종류를 사용합니다. *1 단위는 [명령 단위]입니다. *2
LastPosition	종료 위치	LREAL	음수, 양수, '0'	0	트리거 수신을 종료할 위치를 지정합니다. ReferenceType(위치 종류 선택)에서 지정한 종류를 사용합니다. *1 단위는 [명령 단위]입니다. *2
ReferenceType (Reserved)	위치 종류 선택	_eMC_ REFERENCE_ TYPE	1: _mcFeedback	1 *3	(Reserved)
StopMode	정지 방법 선택	_eMC_STOP_ MODE	1: _mcImmediateStop 4: _mcNonStop	4 *3	정지 방법을 지정합니다. 1 : 즉시 정지합니다. 4 : 정지하지 않습니다.

*1. 자세한 내용은 📖 'WindowOnly(창 유효) (3-330페이지)'를 참조하십시오.
*2. 명령 단위에 대해서는 📖 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*3. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
RecordedPosition	래치 위치	LREAL	음수, 양수, '0'	래치한 위치를 출력합니다. 단위는 [명령 단위]입니다. *1
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*2	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	• 트리거 신호 발생 후 래치 위치를 기록하고 명령이 완료되었을 때 • 정지 지정 유효 시에는 상기 및 축이 래치 위치에 정지했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • StopMode를 [_mcImmediateStop]으로 실행 중에 CSP 이외의 모드로 변경된 경우 • 슬레이브 이탈 시 • 슬레이브 통신 이상 발생 시(프로세스 데이터 통신 중은 제외) • MC_AbortTrigger 명령이 기동되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1
TriggerInput	트리거 입력 조건	_sTRIGGER_REF	–	트리거 조건을 설정합니다. *2
TriggerVariable	트리거 변수	BOOL	TRUE, FALSE	트리거 조건에서 컨트롤러 모드를 지정했을 때 트리거가 될 입력 변수를 지정합니다.

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.
*2. _sTRIGGER_REF 형식의 사용자 정의 변수를 작성하십시오.

● _sTRIGGER_REF

멤버 변수	이 름	데이터 형식	유효 범위	기 능
Mode	모드	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	트리거 모드를 지정합니다. 0 : 드라이버 모드 1 : 컨트롤러 모드
LatchID	래치 ID 선택	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	드라이버 모드일 때 2가지 래치 기능 중 어느 것을 사용할지 지정합니다. 0 : 래치 기능 1 1 : 래치 기능 2
InputDrive	드라이브 트리거 입력 신호	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	드라이버 모드일 때 드라이버의 트리거 신호를 지정합니다. 0 : Z상 1 : 외부 입력

기능 설명

- 센서 입력 등의 트리거 발생을 기점으로 한 위치 제어 실행을 목적으로 하며, 트리거가 발생하면 축 위치를 기록(래치)합니다.
- 트리거 설정에 따라 Axis(축)에서 지정된 축 위치를 RecordedPosition(래치 위치)에 출력합니다.
  트리거 설정으로 TriggerInput(트리거 입력 조건), WindowOnly(창 유효), FirstPosition(시작 위치), LastPosition(종료 위치), StopMode(정지 방법 선택)를 지정할 수 있습니다.
- RecordedPosition(래치 위치) 출력값은 같은 MC_TouchProbe(외부 래치 유효) 명령의 인스턴스에서 다시 축 위치가 기록될 때까지 유지됩니다.

● 데이터 오브젝트 매핑

Mode에 [드라이버 모드]를 선택하고 MC_TouchProbe(외부 래치 유효) 명령을 사용하는 경우는 다음 오브젝트 데이터를 매핑하십시오.
매핑은 Sysmac Studio 축 기본 설정의 [고급 설정]에서 실행합니다.

축 종류가 서보 축인 경우

- 래치 기능(60B8Hex)
- 래치 Status(60B9Hex)
- 래치 위치 1(60BAHex)
- 래치 위치 2(60BCHex)

축 종류가 엔코더 축인 경우

- 래치 기능(4020Hex)
- 엔코더 입력 슬레이브 소프트 스위치(4020Hex)
- 래치 Status(4030Hex)
- 래치 위치 1(4012Hex)
- 래치 위치 2(4013Hex)
- 엔코더 입력 슬레이브 Status(4030Hex)

이러한 데이터를 설정하지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다.

데이터 오브젝트 매핑에 대해서는 🕮 '2-3 PDO 매핑(2-27페이지)', 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.
NX 시리즈 위치 인터페이스 유니트를 사용하는 경우는 🕮 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"의 'I/O 항목 매핑'에 기재된 내용을 참조하십시오.

명령에 대한 자세한 내용

다음에 이 명령에 대해 자세히 설명합니다.

● Axis(축) 지정

- Axis(축)에서 래치 대상이 되는 축을 지정합니다.
- 지정한 Axis(축)가 MC_GroupEnable(다축 협조 제어) 명령 중에는 MC_TouchProbe(외부 래치 유효) 명령은 이상이 되어 실행할 수 없습니다.
- 축별로 LatchID(래치 ID 선택)를 지정하고 동시에 2개까지 MC_TouchProbe(외부 래치 유효) 명령을 실행할 수 있습니다.
- LatchID(래치 ID 선택)는 MC_AbortTrigger(외부 래치 무효) 명령으로 중지할 래치를 지정하는 경우에도 사용합니다.

참고

- 오므론의 GX 시리즈 EtherCAT 엔코더 입력 슬레이브 GX-EC02□□에 연결한 엔코더 축을 사용해 이 명령에 의한 래치를 실행할 수도 있습니다.
- NX 시리즈 펄스 출력 유니트를 사용하는 경우는 이 명령을 통해 래치를 실행할 수도 있습니다. 자세한 내용은 🕮 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"를 참조하십시오.

● 트리거 입력 조건

TriggerInput(트리거 입력 조건)의 Mode, LatchID, InputDrive를 지정해 트리거 조건을 선택합니다.

Mode

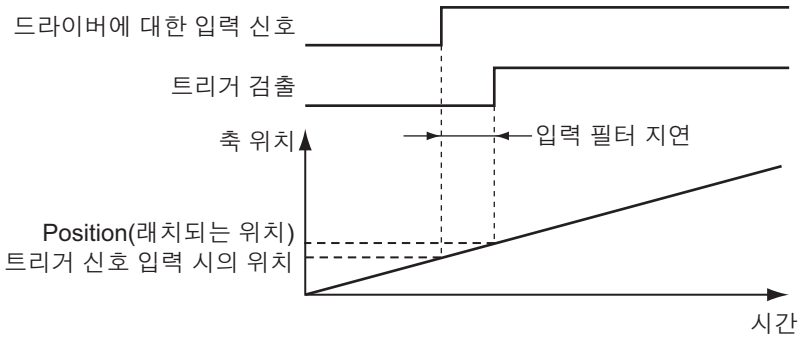
- Mode 설정에는 트리거에 서보 드라이버 등으로부터의 신호를 지정하는 '드라이버 모드'와 트리거에 TriggerVariable(트리거 변수)을 지정하는 '컨트롤러 모드'가 있습니다.
- 트리거 신호의 기동을 트리거 발생으로 간주하고, 래치 명령 실행 후 최초의 트리거 발생(FALSE→TRUE) 시 축의 위치를 기록합니다.
- 이 명령이 Busy(동작 중) 중에는 Execute(기동)가 FALSE가 되어도 TriggerVariable(트리거 변수)의 변화를 트리거로 동작합니다.

참고

오므론의 GX 시리즈 EtherCAT 엔코더 입력 슬레이브 GX-EC02□□를 사용하는 경우는 Mode를 [_mcDrive(드라이브 모드)]로 설정하십시오.

드라이버 모드

'드라이버 모드'는 서보 드라이버 등의 기능으로 트리거를 검출하고 현재 위치를 취득하므로 '컨트롤러 모드'에 비해 래치되는 현재 위치가 정확합니다.



사용상의 주의 사항

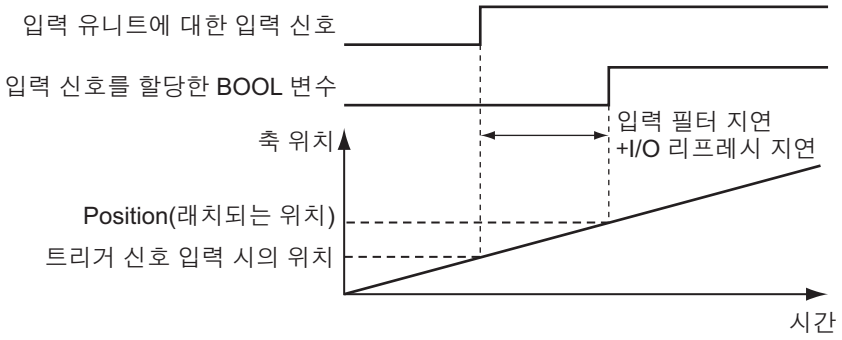
- 드라이버 모드를 사용하는 경우는 반드시 사용하는 LatchID(래치 ID 선택)에 래치 신호를 연결하십시오.
- 래치 신호의 폭은 서보 드라이버 등의 성능 등에 따릅니다.

참고

오므론의 GX 시리즈 EtherCAT 엔코더 입력 슬레이브 GX-EC02□□를 사용하는 경우 InputDrive에는 _mcEXT(외부 입력)를 지정하십시오.
오므론의 GX 시리즈 EtherCAT 엔코더 입력 슬레이브 GX-EC02□□에는 Z상에 의한 래치 기능이 없습니다. 따라서 _mcEncoderMark(Z상)를 지정하면 이 명령 기동 시에 이상이 됩니다. 기동 시에 Error(에러)가 TRUE가 되고 ErrorID(에러 코드)에 '프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461 Hex)'이 출력됩니다.

컨트롤러 모드

- '컨트롤러 모드'에서는 트리거에 BOOL 변수를 지정할 수 있습니다.
- TriggerVariable(트리거 변수)에 트리거로 하려는 BOOL 변수를 지정합니다.
- '컨트롤러 모드'에서는 트리거 입력 신호가 BOOL 변수에 대해 I/O 리프레시 지연으로 '드라이버 모드'보다도 한참 늦게 반영됩니다.



사용상의 주의 사항

컨트롤러 모드를 사용하면 프라이머리 정주기 태스크 간격으로 래치 처리를 실행합니다. 따라서 트리거 변수는 프라이머리 정주기 태스크의 1주기 동안 TRUE여야 합니다.
또한 트리거 변수가 TRUE가 된 후 MC 기능 모듈이 처리할 때까지 프라이머리 정주기 태스크 1주기 동안의 시간이 필요합니다.

LatchID

- 하나의 축에 대해 동시에 2개까지 MC_TouchProbe(외부 래치 유효) 명령을 실행할 수 있습니다. LatchID(래치 ID 선택)에서 어느 것을 사용할지 지정합니다.
- 같은 축에 대해 지정한 래치 ID가 이미 실행 중인 경우는 나중에 실행한 명령이 유효가 됩니다. 그때까지의 명령은 CommandAborted(실행 중단)가 됩니다.
- 서보 드라이버 등에서는 LatchID(래치 ID 선택)가 서보 드라이버의 래치 회로 1(제1 ch), 래치 회로 2(제2 ch)를 의미합니다.

LatchID에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

참고

오므론의 GX 시리즈 EtherCAT 엔코더 입력 슬레이브 GX-EC02□□를 사용하는 경우의 대응은 다음과 같습니다.

열거자	엔코더 입력 터미널의 신호명
_mcLatch1	외부 래치 입력 A
_mcLatch2	외부 래치 입력 B

InputDrive

- 트리거로 _mcEncoderMark(z상)와 _mcEXT(외부 입력)를 선택할 수 있습니다.
- _mcEncoderMark(Z상)에서는 서보 드라이버 등의 Z상을 트리거에 사용합니다.
  _mcEXT(외부 입력)에서는 서보 드라이버에 입력된 외부 트리거 신호를 트리거에 사용합니다.
- 오므론의 서보 드라이버 G5 시리즈에서 _mcEXT(외부 입력)은 설정된 Ext1, Ext2, Ext3 중 하나가 됩니다. 설정에는 Sysmac Studio를 사용합니다.
  드라이버에 설정되는 2개의 트리거는 같은 설정이라도 가능합니다.

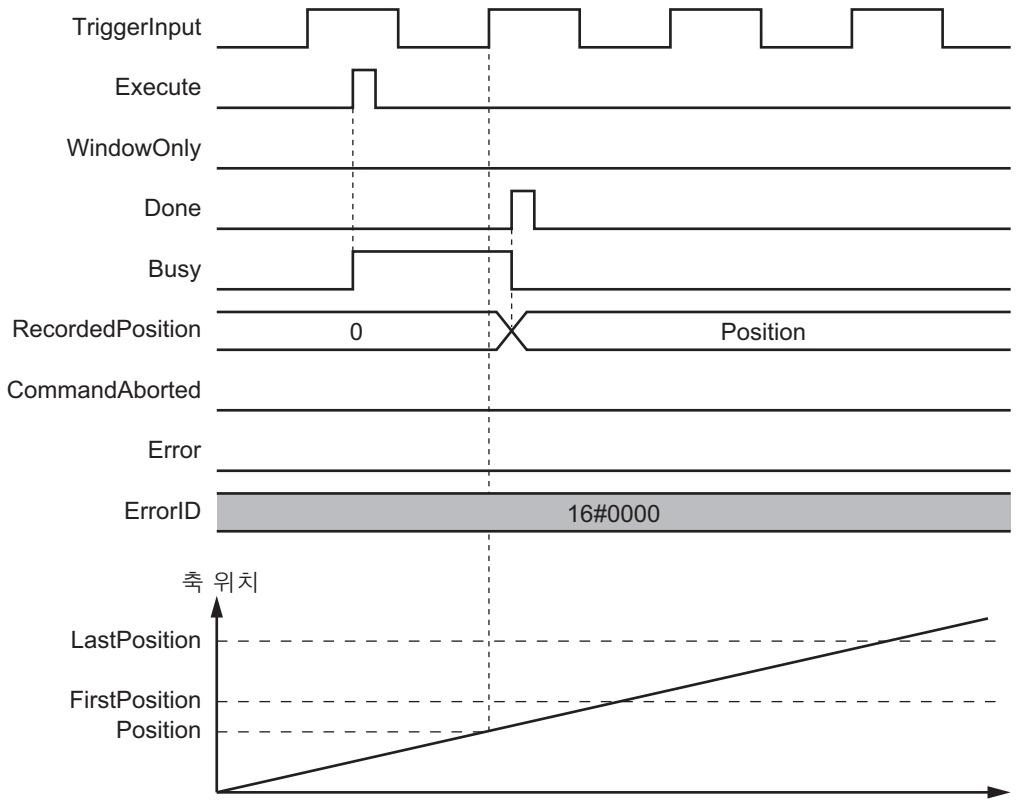
● WindowOnly(창 유효)

- WindowOnly(창 유효)에서는 창의 유효/무효를 지정합니다.
- Disable(무효)을 지정하면 모든 축 위치에서 트리거를 검출합니다.
- Enable(유효)을 지정하면 축 위치가 FirstPosition(시작 위치)과 LastPosition(종료 위치) 범위 내에 있을 때만 트리거를 검출합니다.

WindowOnly(창 유효) 지정에 따른 동작의 차이를 타이밍 차트로 표시합니다.

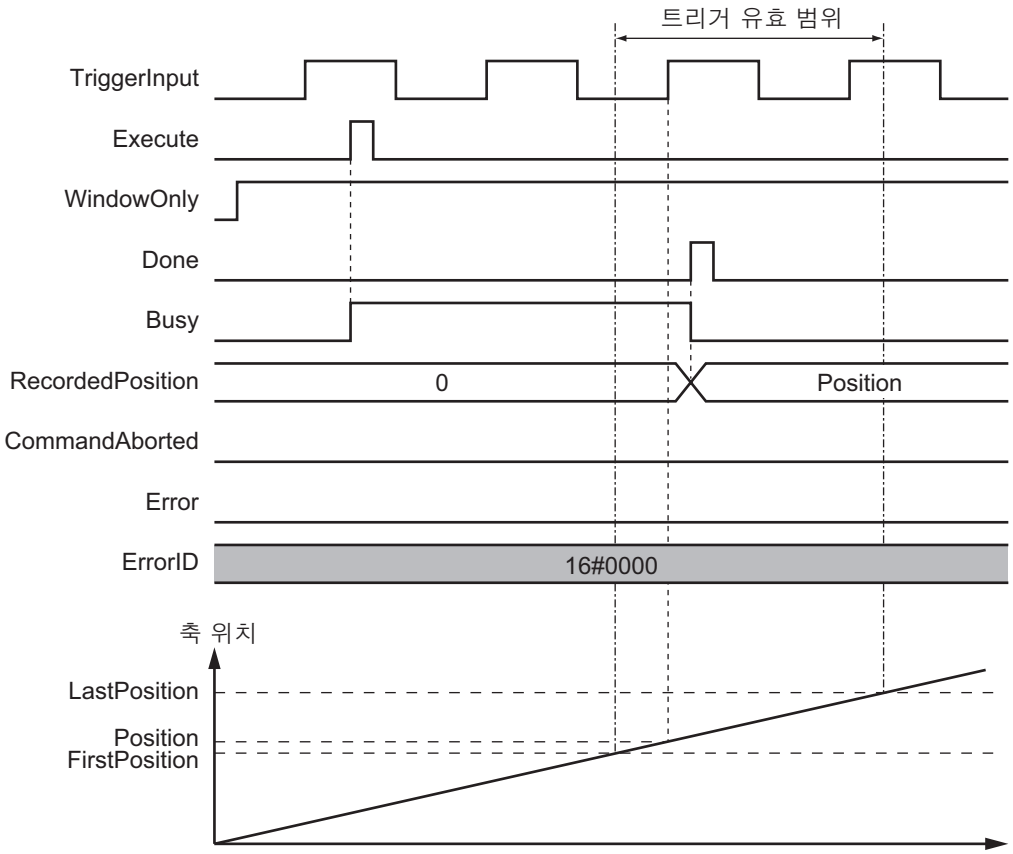
WindowOnly가 Disable인 경우

Execute(기동)가 TRUE가 된 후의 최초 트리거 발생 시의 축 위치를 RecordedPosition(래치 위치)으로 출력합니다.



WindowOnly가 Enable인 경우

창의 범위 내에서만 트리거 입력을 검출해서 축 위치를 취득합니다.



사용상의 주의 사항

- WindowOnly(창 유효)가 FALSE→TRUE로 변화하는 순간과 래치 기능이 동작할 때까지의 시간은 래치할 수 없습니다.
- 래치 기능이 동작할 때까지 시간이 필요하기 때문에 WindowOnly(창 유효)의 유효 범위가 매우 짧은 경우는 래치할 수 없습니다. 래치 가능한 유효 범위의 한계값은 서보 드라이버, 엔코더 입력 터미널, 위치 인터페이스 유니트 등의 성능 및 EtherCAT 통신에 따릅니다.

또한 카운트 모드의 차이에 따라 FirstPosition(시작 위치)과 LastPosition(종료 위치)의 범위는 다음과 같습니다.

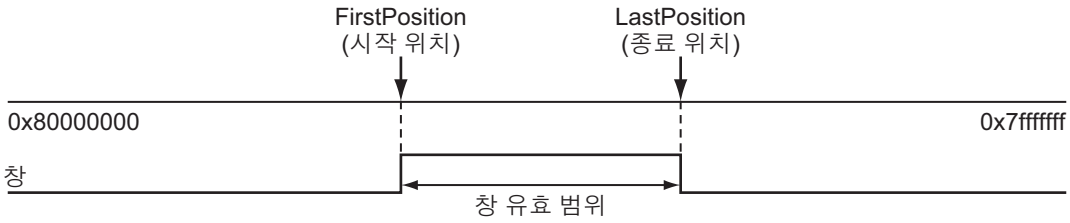
리니어 모드

- 창의 유효 범위는 다음 식으로 표시됩니다.

  FirstPosition(시작 위치) ≦ 창 범위 ≦ LastPosition(종료 위치)

- FirstPosition(시작 위치) > LastPosition(종료 위치)으로 지정하면 이상이 됩니다.
- [리니어 모드] 위치 범위를 넘어서 지정한 경우도 이상이 됩니다.
- 또한 FirstPosition(시작 위치)과 LastPosition(종료 위치)은 LREAL 형식 변수이므로 FirstPosition(시작 위치) = LastPosition(종료 위치)을 지정하지 마십시오.
  LREAL 형식에 대해서는 '실수(REAL 형식, LREAL 형식) 사용 방법(1-14페이지)'을 참조하십시오.

[리니어 모드]일 때 창 유효 범위는 다음 그림과 같습니다.

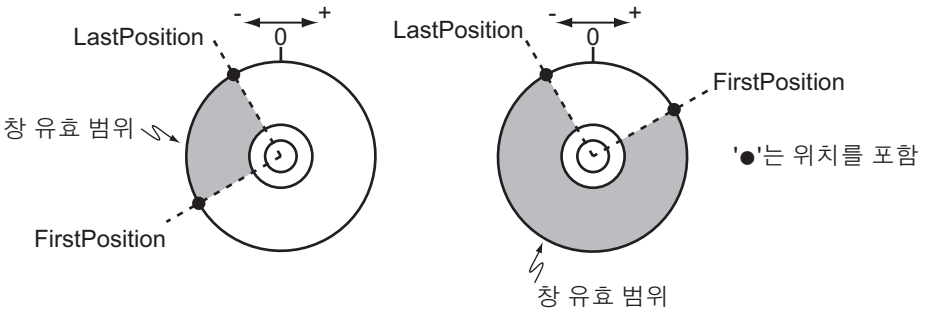


(주의) FirstPosition(시작 위치)과 LastPosition(종료 위치)은 창 유효 범위에 포함됩니다.

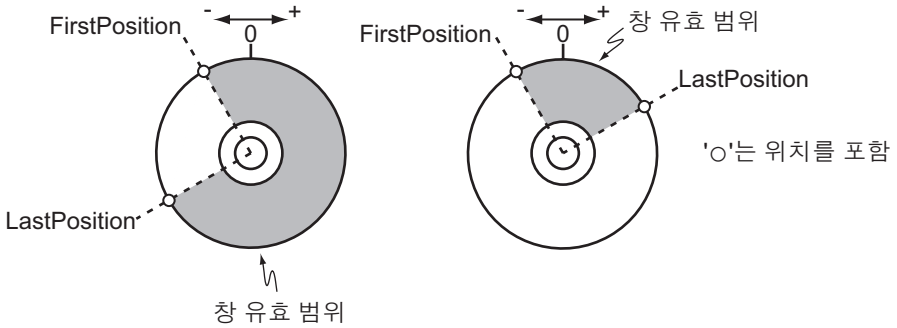
로터리 모드

- FirstPosition(시작 위치) ≦ LastPosition(종료 위치) 및 FirstPosition(시작 위치) > LastPosition(종료 위치) 둘 다 지정 가능합니다.
- FirstPosition(시작 위치) > LastPosition(종료 위치)인 경우는 링 카운터 상하한 위치에 걸쳐 설정됩니다.
- 링 카운터 상하한 범위를 넘어서 지정하면 이상이 됩니다.

FirstPosition ≦ LastPosition

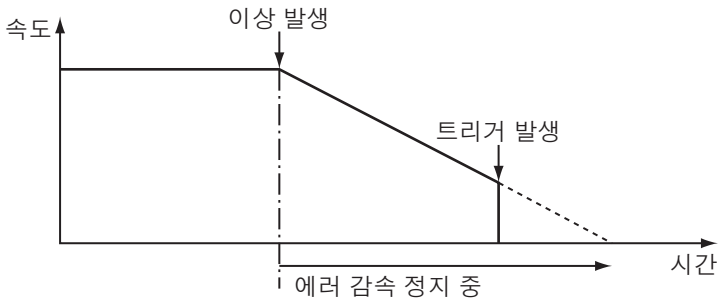


FirstPosition > LastPosition



● StopMode(정지 방법 선택)

- 트리거 발생 시에 지정한 Axis(축)에 대해 StopMode(정지 방법 선택)를 지정할 수 있습니다.
- _mcNonStop(비정지)은 트리거가 발생해도 축을 정지시키지 않습니다.
- _mcImmediateStop(즉시 정지)은 트리거 발생으로 축을 래치한 위치에 즉시 정지합니다. 이 정지로 인해 해당 축을 동작시키던 명령은 CommandAborted(실행 중단)가 됩니다.
- _mcImmediateStop(즉시 정지)은 CSP 모드일 때 기능합니다.
- CSV / CST 모드인 상태에서 _mcImmediateStop(즉시 정지)을 지정해서 기동하면 기동 시 이상이 됩니다. 또한 _mcImmediateStop(즉시 정지)으로 동작 중에 CSV / CST 모드로 변경하면 CommandAborted(실행 중단)가 됩니다.
- mcImmediateStop(즉시 정지)은 축 명령이 정지한 시점에서 Done(완료)이 됩니다. 래치한 위치에 정지할 때까지 Busy(실행 중)가 TRUE를 유지합니다.
- _mcImmediateStop(즉시 정지)에 의한 정지 시 인포지션 확인을 실행하지 않습니다.
- _mcImmediateStop(즉시 정지)을 지정한 축에서 축 이상이 발생해 축 정지가 완료되기 전에 트리거가 발생하면 오므론의 서보 드라이버 G5 시리즈의 기능에 의해 축은 즉시 정지합니다.



트리거 발생 시 오므론의 서보 드라이버 G5 시리즈의 기능에 의해 즉시 정지합니다.
모션 제어 기능 모듈은 오므론의 서보 드라이버 G5 시리즈의 정지를 검출한 후 명령 정지를 즉시 정지시킵니다.
점선은 트리거 미발생의 목표 궤도입니다.

사용상의 주의 사항

- _mcImmediateStop(즉시 정지)은 오므론의 서보 드라이버 G5 시리즈 또는 NX 시리즈 펄스 출력 유니트를 사용하는 경우에는 서보 드라이버 또는 NX 시리즈 펄스 출력 유니트의 기능을 사용해 정지시킵니다.
  타사의 서보 드라이버인 경우에는 MC 기능 모듈이 정지 처리를 합니다.
  어느 경우에도 실제로는 래치 위치를 넘어서 정지하고 컨트롤러의 지시에 따라 래치한 위치까지 되돌아가므로 주의하십시오.
- _mcImmediateStop(즉시 정지)을 사용하는 경우 오므론의 서보 드라이버, 타사의 서보 드라이버 모두 명령 속도가 빠르면 래치한 위치로 되돌아가는 양이 커집니다.
  명령 속도가 빨라지지 않도록 주의해서 설정하십시오.
- 엔코더 축에는 _mcNonStop(비정지)을 지정하십시오. _mcImmediateStop(즉시 정지)을 지정하면 이 명령 기동 시에 Error(에러)가 TRUE가 됩니다. 동시에 ErrorID(에러 코드)에는 '외부 래치 유효 명령 실행 불가(에러 코드: 5492 Hex)'를 출력합니다.
- _mcImmediateStop(즉시 정지)을 사용할 경우, 이하의 2개 처리가 같은 제어 주기로 실행되면 오므론 서보 드라이버 G5 시리즈는 Target Ignore 상태가 됩니다.
  - 래치 신호 ON 시의 서보 드라이버의 카운터 래치 처리
  - MC_TouchProbe(외부 래치 유효) 명령으로 CommandAborted(실행 중단)가 TRUE가 되었을 때의 래치 해제 처리

  래치 신호가 ON하는 타이밍에 이 명령의 CommandAborted(실행 중단)가 TRUE가 되지 않도록 주의하십시오.

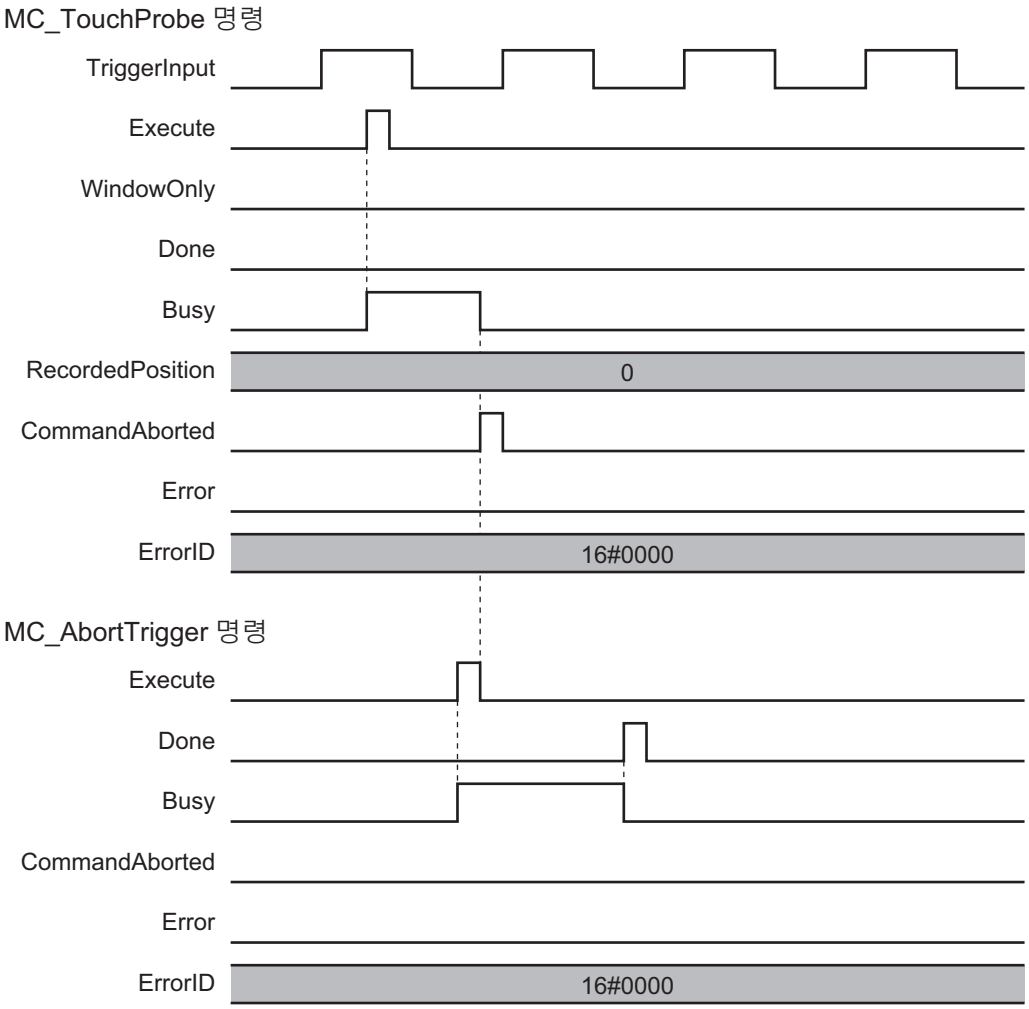
● 축 변수의 Status

StopMode(정지 방법 선택)에 _mcImmediateStop(즉시 정지)을 지정하면 트리거 발생에 의한 정지 처리 중에는 축 변수의 Status.Stopping(감속 정지 중)이 TRUE가 됩니다.

명령 중지

MC_TouchProbe(외부 래치 유효) 명령은 MC_AbortTrigger(외부 래치 무효) 명령으로 종료할 수 있습니다.

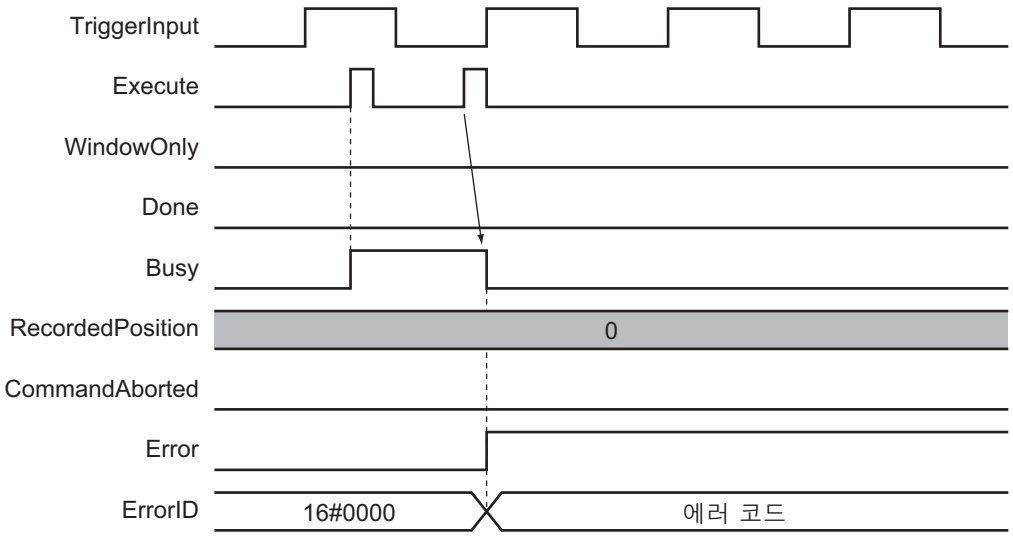
중지시키려는 Axis(축)와 LatchID(래치 ID 선택)를 MC_AbortTrigger(외부 래치 무효) 명령으로 지정해서 명령을 기동하면 종료됩니다.



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

동일한 MC_TouchProbe(외부 래치 유효) 명령의 인스턴스에 대해 RecordedPosition(래치 위치)의 읽기가 완료되기 전에 다시 Execute(기동)를 FALSE → TRUE로 하면 이상이 됩니다.



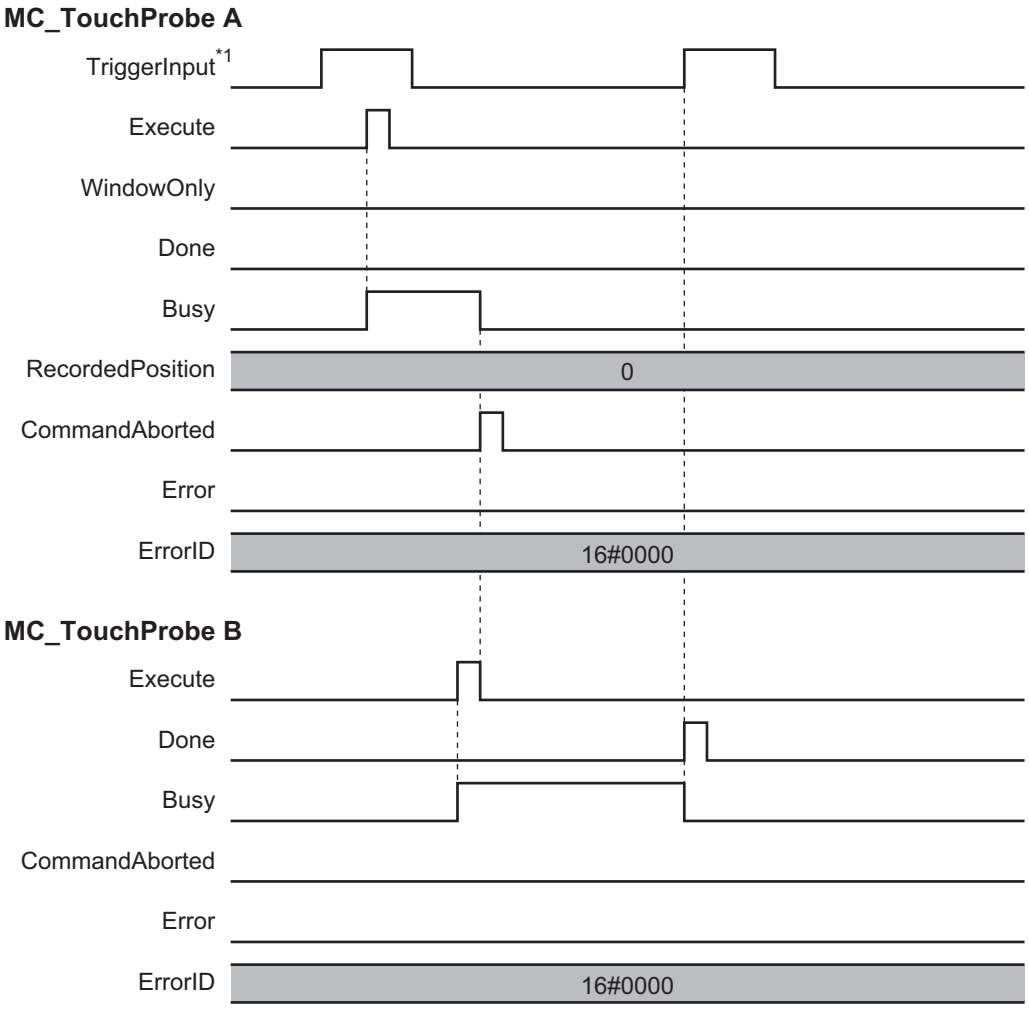
모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

같은 Axis(축)의 1개의 LatchID(래치 ID 선택)에 대해서는 동시에 1개의 트리거 설정만 실행할 수 있습니다.

MC_TouchProbe(외부 래치 유효) 명령 실행 중에 같은 LatchID(래치 ID 선택)에 대해 다른 MC_TouchProbe(외부 래치 유효) 명령의 인스턴스가 실행되면 첫 번째는 CommandAborted(실행 중단)되고 두 번째가 새로 실행됩니다.



*1 서보 드라이버 등의 트리거 입력 신호인 경우입니다.

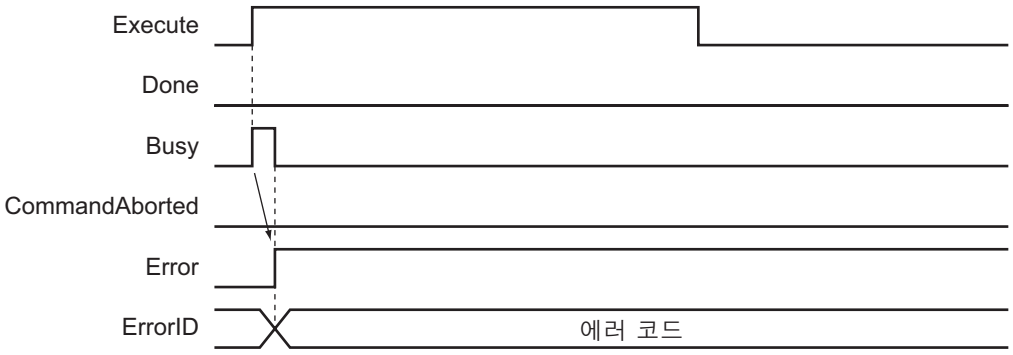
참고

StopMode(정지 방법 선택)에 _mcImmediateStop(즉시 정지)을 지정하면 트리거 발생에 의한 정지 시에 다중 기동된 명령의 CommandAborted(실행 중단)가 TRUE가 됩니다.

이상

MC_TouchProbe(외부 래치 유효) 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 됩니다. ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기에는 외부 센서에 의한 위치 취득 제어에 대한 샘플 프로그램이 실려 있습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터

축 종류

축	축 종류
축 1	서보 축

카운터 모드

축	카운터 모드
축 1	로터리 모드

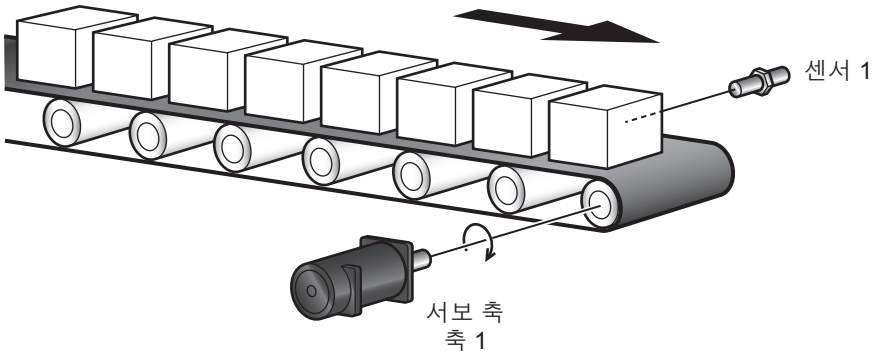
링 카운터

축	상한값	하한값
축 1	360	0

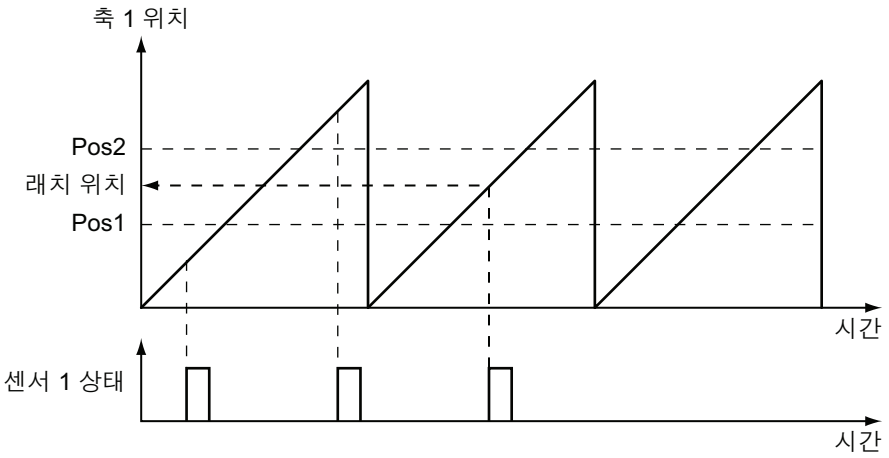
좌표 단위 선택

축	좌표 단위
축 1	degree

동작 예



● 동작 패턴



1 **주축의 시작**

축 1은 속도 제어를 실행합니다.

2 **워크 검출**

센서 1로 워크의 유무를 검출합니다.

3 **위치 취득**

워크가 창의 TRUE 범위 내(Pos1~Pos2)에서 검출되면 이 때 축 1의 래치 위치를 취득합니다.

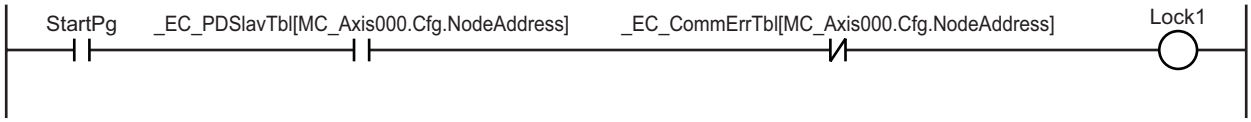
래더도

● 주요 변수

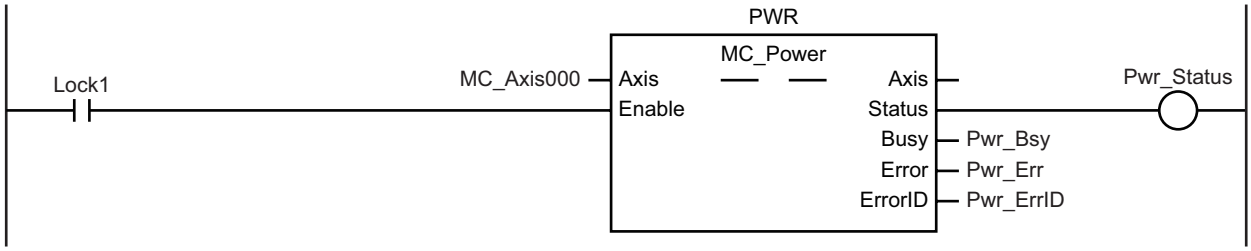
이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pos1	LREAL	–	창 유효의 시작 위치를 표시합니다.
Pos2	LREAL	–	창 유효의 종료 위치를 표시합니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 샘플 프로그램

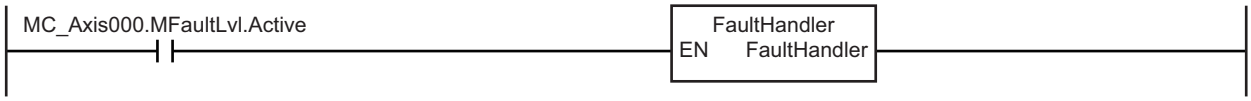
접점 StartPg가 TRUE가 되면 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인



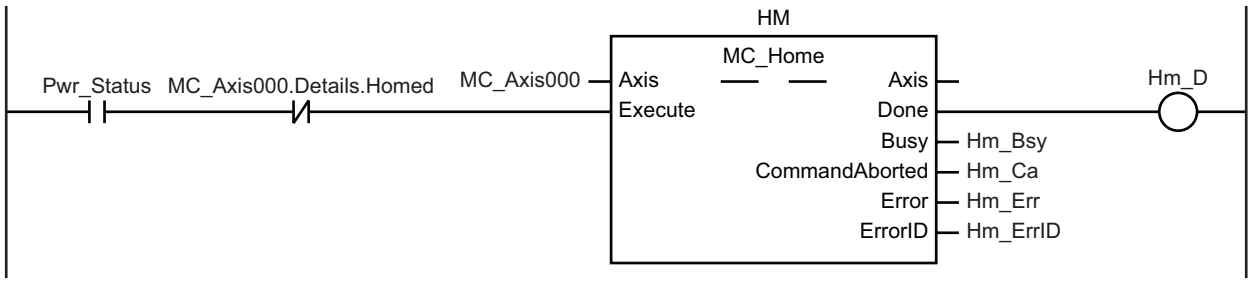
축 1이 프로세스 데이터 통신을 하고 있으면 서보를 ON



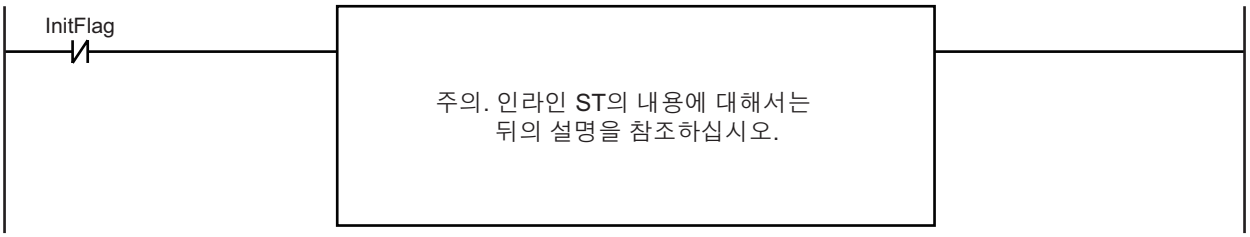
축 1에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



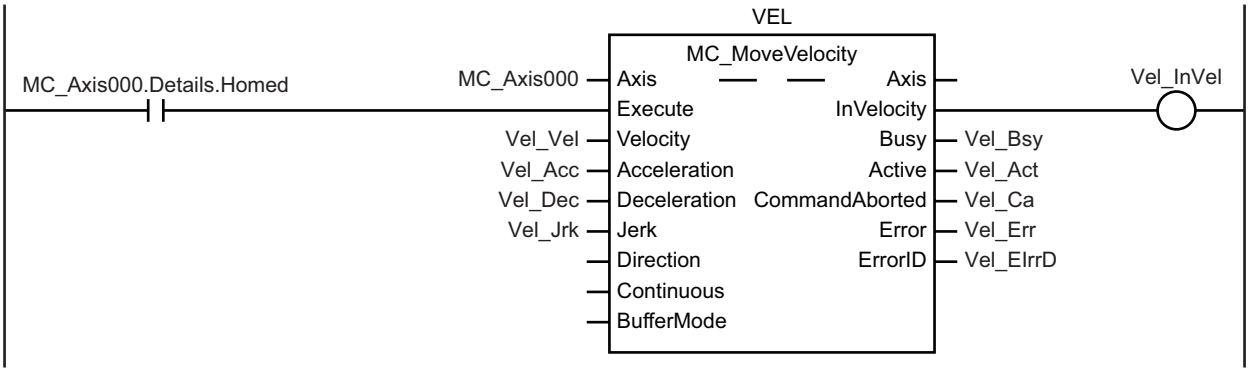
서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행



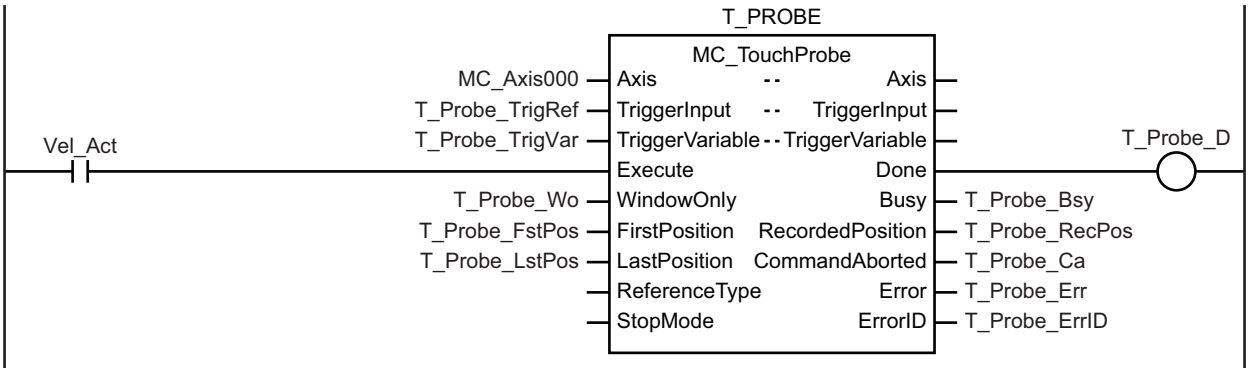
MC_MoveVelocity(속도 제어) 명령과 MC_TouchProbe(외부 래치 유효) 명령의 파라메터를 설정



축 1이 원점 확정 상태가 되면 MC_MoveVelocity(속도 제어) 명령을 실행



MC_MoveVelocity(속도 제어) 명령이 시작되면 래치 처리를 실행



인라인 ST의 내용

```
//MC_MoveVelocity 파라메터
    Vel_Vel    := LREAL#1000.0;
    Vel_Acc    := LREAL#1000.0;
    Vel_Dec    := LREAL#1000.0;
    Vel_Jrk    := LREAL#1000.0;

//MC_TouchProbe 파라메터
    T_Probe_TrigRef.Mode        := _eMC_TRIGGER_MODE#_mcDrive;
    T_Probe_TrigRef.LatchID     := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
    T_Probe_TrigRef.InputDrive  := _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
    T_Probe_TrigVar             := FALSE;
    T_Probe_Wo                  := TRUE;
    T_Probe_FstPos              := LREAL#1000.0;
    T_Probe_LstPos              := LREAL#2000.0;

    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;
```

스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
Pwr_Status	BOOL	FALSE	MC_Power 인스턴스 PWR의 출력 변수 Status에 할당한 변수입니다. 서보 ON 상태가 되면 이 변수가 TRUE가 됩니다.
Pos1	LREAL	–	창 유효의 시작 위치를 표시합니다.
Pos2	LREAL	–	창 유효의 종료 위치를 표시합니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 서보 ON 상태가 됩니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag=FALSE THEN

    //MC_MoveVelocity 파라메터
    Vel_Vel   := LREAL#1000.0;
    Vel_Acc   := LREAL#1000.0;
    Vel_Dec   := LREAL#1000.0;
    Vel_Jrk   := LREAL#1000.0;

    //MC_TouchProbe 파라메터
    T_Probe_TrigRef.Mode        := _eMC_TRIGGER_MODE#_mcDrive;
    T_Probe_TrigRef.LatchID     := _eMC_TRIGGER_LATCH_ID#_mcLatch1;
    T_Probe_TrigRef.InputDrive  := _eMC_TRIGGER_INPUT_DRIVE#_mcEXT;
    T_Probe_TrigVar             := FALSE;
    T_Probe_Wo                  := TRUE;
    T_Probe_FstPos              := LREAL#1000.0;
    T_Probe_LstPos              := LREAL#2000.0;

    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;


//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;
```

```
//가벼운 수준의 이상 발생 시 처리
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;


//서보가 ON이며 원점 미확정 상태면 원점 복귀를 실행합니다.
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

//원점 확정 후에 MC_MoveVelocity를 실행합니다.
IF MC_Axis000.Details.Homed=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

//MC_MoveVelocity가 기동된 후 MC_TouchProbe를 실행합니다.
IF Vel_Act=TRUE THEN
    T_Probe_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_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_TouchProbe
T_PROBE(
    Axis                := MC_Axis000,
    TriggerInput        := T_Probe_TrigRef,
    TriggerVariable     := T_Probe_TrigVar,
    Execute             := T_Probe_Ex,
```

```
    WindowOnly          := T_Probe_Wo,
    FirstPosition       := T_Probe_FstPos,
    LastPosition        := T_Probe_LstPos,
    Done                => T_Probe_D,
    Busy                => T_Probe_Bsy,
    RecordedPosition    => T_Probe_RecPos,
    CommandAborted      => T_Probe_Ca,
    Error               => T_Probe_Err,
    ErrorID             => T_Probe_ErrID
);
```

MC_AbortTrigger

실행 중인 래치를 무효로 합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_AbortTrigger	외부 래치 무효	FB	MC_AbortTrigger_instance MC_AbortTrigger Axis — — Axis TriggerInput — — TriggerInput Execute — Done Busy CommandAborted Error ErrorID	MC_AbortTrigger_instance (Axis := 《파라메터》, TriggerInput := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	중지	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 📖 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	• 래치 정지를 실행했을 때 • 래치 실행 중이 아닌 래치 기능에 대해 이 명령을 실행해서 처리를 종료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	래치 대상이 되는 축을 지정합니다. *1
TriggerInput	트리거 선택	_sTRIGGER_REF	–	트리거 조건을 선택합니다. *2 _sTRIGGER_REF의 자세한 내용은 다음 표를 참조하십시오.

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.
*2. _sTRIGGER_REF 형식의 사용자 정의 변수를 작성하십시오.

● _sTRIGGER_REF

멤버 변수	이 름	데이터 형식	유효 범위	기 능
Mode	모드	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	트리거 모드를 지정합니다. 0 : 드라이버 모드 1 : 컨트롤러 모드
LatchID	래치 ID 선택	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	드라이버 모드일 때 2가지 래치 기능 중 어느 것을 사용할지 지정합니다. 0 : 래치 기능 1 1 : 래치 기능 2
InputDrive	드라이브 트리거 입력 신호	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	드라이버 모드일 때 드라이버의 트리거 신호를 지정합니다. 0 : Z상 1 : 외부 입력

기능 설명

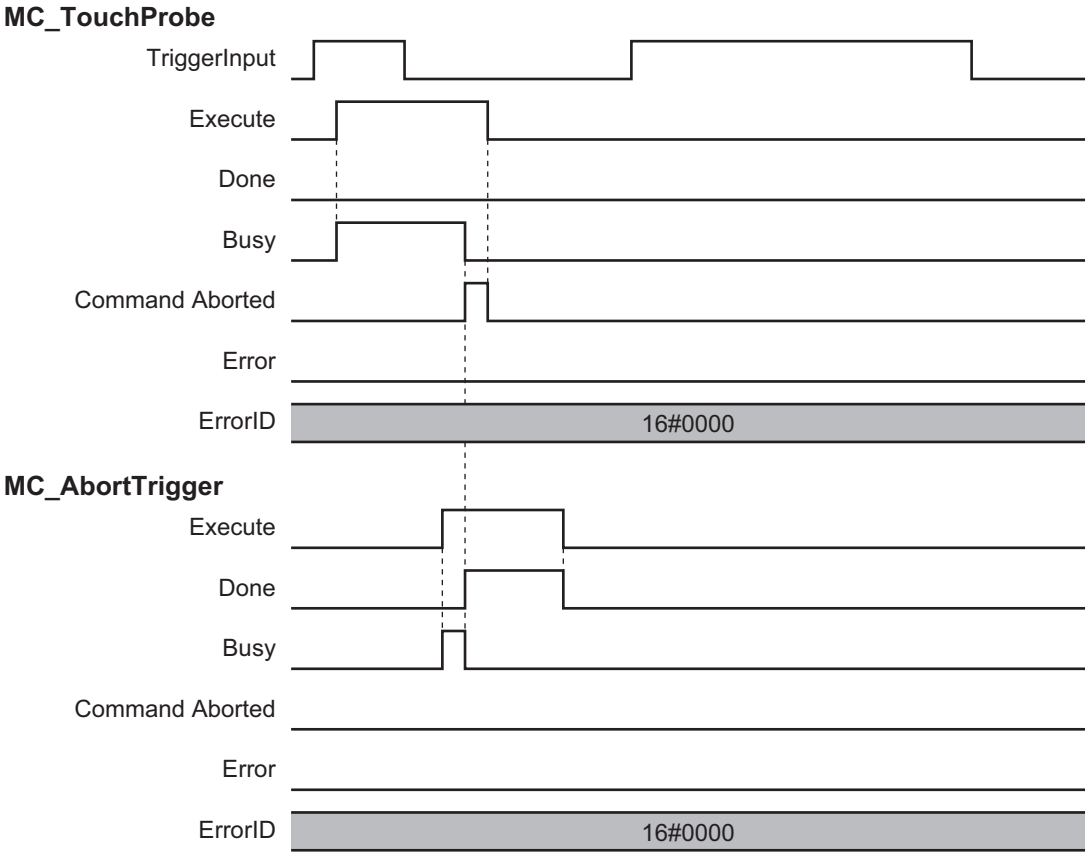
- 실행 중인 래치를 중지합니다.
- MC_AbortTrigger(외부 래치 무효) 명령에서는 Axis(축)와 LatchID(래치 ID 선택)를 지정해 중지할 래치를 지정합니다.
- 래치 요구가 실행되고 있지 않은 트리거에 대해 MC_AbortTrigger(외부 래치 무효) 명령을 실행하면 MC_AbortTrigger(외부 래치 무효) 명령은 아무 것도 하지 않고 정상 종료합니다.
  MC_TouchProbe(외부 래치 유효) 명령의 실행이 Done(완료)이 된 것에 대해 MC_AbortTrigger(외부 래치 무효) 명령을 실행한 경우도 동일합니다.

사용상의 주의 사항

- MC_AbortTrigger(외부 래치 무효) 명령을 지정한 Axis(축)가 MC_GroupEnable(축 그룹 유효) 명령 유효이면 MC_AbortTrigger(외부 래치 무효) 명령이 이상이 되어 실행할 수 없습니다.
- MC_Home(원점 복귀) 명령, MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령, MC_MoveLink (사다리꼴 패턴 캠) 명령 실행 중에 MC_AbortTrigger(외부 래치 무효) 명령을 실행하면 이 명령은 이상이 됩니다.

타이밍 차트

- MC_AbortTrigger(외부 래치 무효) 명령의 Done(완료)은 Execute(기동)의 1주기 후에 TRUE가 됩니다.



모션 명령 재기동

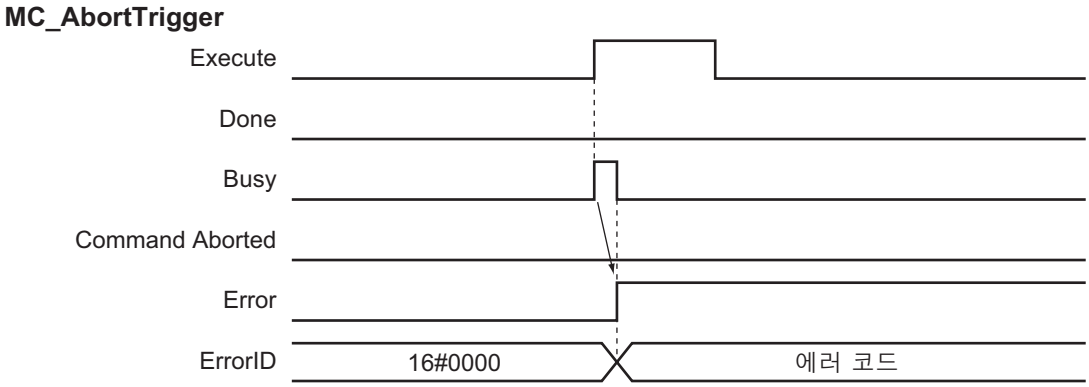
이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

이상

MC_AbortProbe(외부 래치 무효) 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_AxesObserve

지정된 축의 명령 위치 또는 피드백 위치의 차이가 허용값을 초과하지 않는지 감시합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_AxesObserve	축간 편차 감시	FB	MC_AxesObserve_instance MC_AxesObserve Master — — Master Slave — — Slave Enable — Enabled ReferenceType — Invalid PermittedDeviation — Busy — DeviatedValue — Error — ErrorID	MC_AxesObserve_instance (Master := 《파라메터》, Slave := 《파라메터》, Enable := 《파라메터》, ReferenceType := 《파라메터》, PermittedDeviation := 《파라메터》, Enabled => 《파라메터》, Invalid => 《파라메터》, Busy => 《파라메터》, DeviatedValue => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Enable	유효	BOOL	TRUE, FALSE	FALSE	TRUE인 동안 명령을 실행합니다.
ReferenceType	위치 종류 선택	_eMC_ RERERENCE_ TYPE	0: _mcCommand 1: _mcFeedback	0 *1	위치 종류를 지정합니다. 0 : 명령 위치(직전 프라이머리 주기에서의 계산값) 1 : 피드백 위치(동일 프라이머리 주기에서의 취득값)
Permitted Deviation	축간 편차 허용값	LREAL	양수 또는 '0'	0	허용 가능한 주축과 종축 위치 편차의 최대값을 지정합니다. 단위는 [명령 단위]입니다. *2

*1. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*2. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Enabled	유효	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
Invalid	축간 편차 오버	BOOL	TRUE, FALSE	축간 편차 허용값을 초과했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
DeviatedValue	축간 편차값	LREAL	음수, 양수, '0'	지정된 주축과 종축의 차이를 출력합니다. 단위는 [명령 단위]입니다. *1
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*2	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Enabled	Enable이 TRUE일 때	• Error가 TRUE가 되었을 때 • Enable이 FALSE가 되었을 때
Invalid	축간 편차 허용값을 초과했을 때	• Error가 TRUE가 되었을 때 • Enable이 FALSE가 되었을 때 • 축간 편차 허용값을 초과하지 않았을 때
Busy	Enable의 기동	• Error가 TRUE가 되었을 때 • Enable이 FALSE가 되었을 때
DeviatedValue*1	Enable이 TRUE일 때	
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

*1. Deviated Value는 Enable이 FALSE가 되는 타이밍에 FALSE로는 돌아가지 않습니다.

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Master	주축	_sAXIS_REF	–	주축을 지정합니다. *1
Slave	종축	_sAXIS_REF	–	종축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

사용상의 주의 사항

주축과 종축에 같은 축을 지정하면 가벼운 이상 수준의 '주축 종축 동일(에러 코드: 5436 Hex)' 이상이 됩니다.

참고

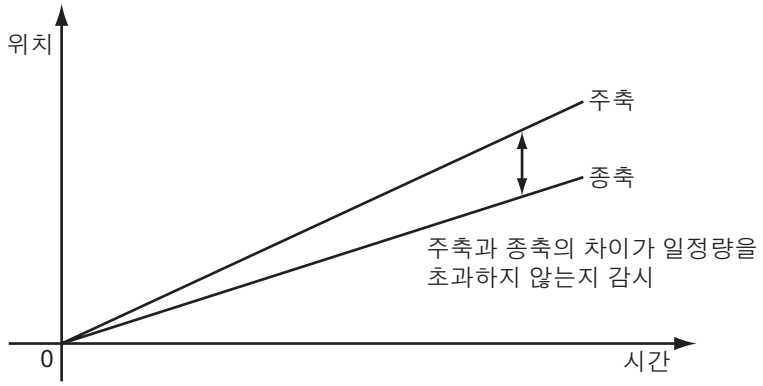
그룹에 속한 축도 설정할 수 있습니다.

기능 설명

- 지정된 Master(주축)와 Slave(종축)의 명령 위치 또는 피드백 위치 차이가 축간 편차 허용값을 넘은 경우에 Invalid(축간 편차 초과)가 TRUE가 됩니다.
  즉, 다음 조건이 성립했을 때 Invalid(축간 편차 초과)가 TRUE가 됩니다.

  | DeviatedValue(축간 편차값) | > PermittedDeviation(축간 편차 허용값)이 성립할 때

- 이 명령은 축의 동작에는 영향을 미치지 않습니다.
- 출력 변수의 Invalid(축간 편차 초과) 상태를 바탕으로 축 정지 등의 처리를 프로그래밍하십시오.



사용상의 주의 사항

- Master(주축)와 Slave(종축)의 카운트 모드는 같은 모드로 하십시오.
  다른 모드의 축을 지정하면 [리니어 모드]로 비교됩니다.
  또한 [로터리 모드]끼리여도 링 카운터의 설정 범위가 다른 경우도 [리니어 모드]로 비교됩니다.
- PermittedDeviation(축간 편차 허용값)이 딱 떨어지지 않는 나눗셈 결과 등을 포함하고 있으면 오차로 인해 처리 결과가 기대와 다른 경우가 있습니다.
- NX 시리즈 펄스 출력 유니트에서는 명령 위치=피드백 현재 위치이므로 이 명령에서 감시하는 것은 실효성이 없습니다.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● ReferenceType(위치 종류 선택)

종축이 동기하는 주축의 위치 종류를 다음에서 선택할 수 있습니다.

- _mcCommand: 명령 위치(직전 프라이머리 주기에서의 계산값)
  현재 주기에 대해 직전의 프라이머리 주기에서 계산한 주축의 명령 위치를 사용합니다.
  종축의 명령 위치를 계산하기 직전의 프라이머리 정주기 태스크에서 계산한 주축의 명령 위치를 사용합니다.
- _mcFeedback: 동일 프라이머리 주기에서의 취득값
  동일 프라이머리 주기에서 취득한 주축의 피드백 위치를 사용합니다.

● 축 종류와 위치 종류의 관계

감시 가능한 축 종류와 감시하는 위치 종류의 관계는 다음과 같습니다.

축 종류	ReferenceType	
	_mcCommand	_mcFeedback
서보 축	○	○
엔코더 축	× *1	○
가상 서보 축	○	○
가상 엔코더 축	× *1	○

*1. 명령 기동 시에 '위치 종류 선택 범위 외(에러 코드: 5430 Hex)'가 발생합니다.

● DeviatedValue(축간 편차값)의 계산 방법

DeviatedValue(축간 편차값)는 다음과 같이 계산합니다.

리니어 모드인 경우

ReferenceType(위치 종류 선택)이 _mcCommand일 때

DeviatedValue(축간 편차값) = Master(주축)의 명령 현재 위치 − Slave(종축)의 명령 현재 위치

ReferenceType(위치 종류 선택)이 _mcFeedback일 때

DeviatedValue(축간 편차값) = Master(주축)의 피드백 현재 위치 − Slave(종축)의 피드백 현재 위치

로터리 모드인 경우

ReferenceType(위치 종류 선택)이 _mcCommand일 때

DeviatedValue(축간 편차값)는 링 카운터의 범위상에서 Master(주축)의 명령 현재 위치와 Slave(종축)의 명령 현재 위치 중 짧은 것의 거리가 됩니다.

이 때 DeviatedValue(축간 편차값)의 부호는 짧은 것의 거리로부터 보아

- Master(주축)의 명령 현재 위치 ≧ Slave(종축)의 명령 현재 위치라면 '+'
- Master(주축)의 명령 현재 위치 < Slave(종축)의 명령 현재 위치라면 '-'가 됩니다.

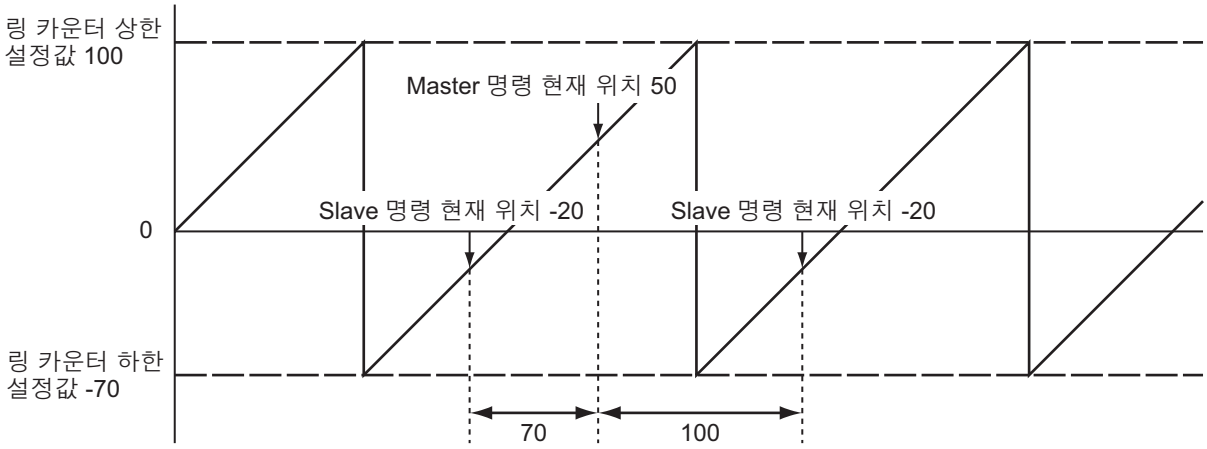
ReferenceType(위치 종류 선택)이 _mcFeedback일 때

DeviatedValue(축간 편차값)는 링 카운터의 범위상에서 Master(주축)의 피드백 현재 위치와 Slave(종축)의 피드백 현재 위치 중 짧은 것의 거리가 됩니다.

이 때 DeviatedValue(축간 편차값)의 부호는 짧은 것의 거리로부터 보아

- Master(주축)의 피드백 현재 위치 ≧ Slave(종축)의 피드백 현재 위치라면 '+'
- Master(주축)의 피드백 현재 위치 < Slave(종축)의 피드백 현재 위치라면 '-'가 됩니다.

[로터리 모드]에서 ReferenceType(위치 종류 선택)이 _mcCommand인 경우의 DeviatedValue(축간 편차 값)의 산출 예

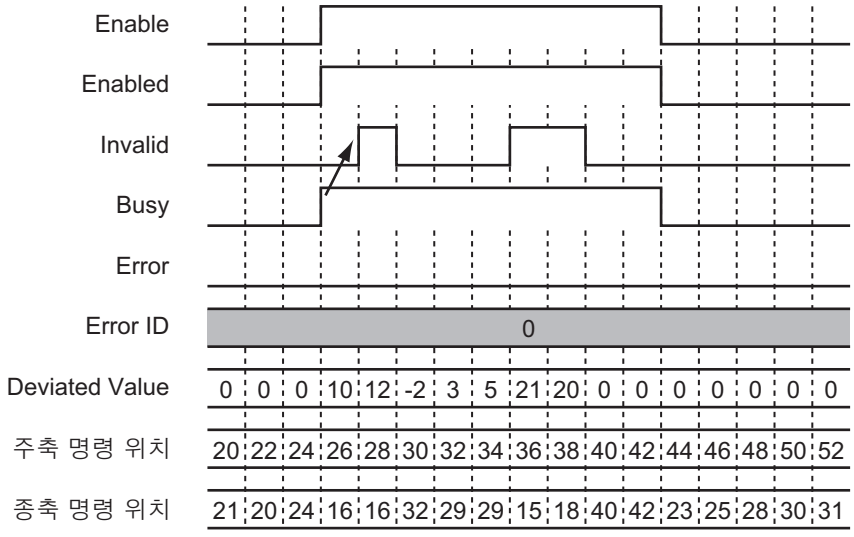


｜DeviatedValue｜는 100＞70이므로 70이 되는 부호는 Master 명령 현재 위치≧Slave 명령 현재 위치에서 '＋'에 따라 DeviatedValue는 '＋70'이 됨

[로터리 모드]에서 ReferenceType(위치 종류 선택)이 _mcFeedback인 경우는 위 그림의 '명령 현재 위치'가 '피드백 현재 위치'가 됩니다.

타이밍 차트

이 명령의 이상은 축 및 그룹의 동작에 영향을 미치지 않습니다.
PermittedDeviation(축간 편차 허용값)을 '10.0'으로 한 경우의 타이밍 차트는 다음과 같습니다.



모션 명령 재기동

입력이 'Enable 형식'인 명령에는 모션 명령 재기동은 없습니다.

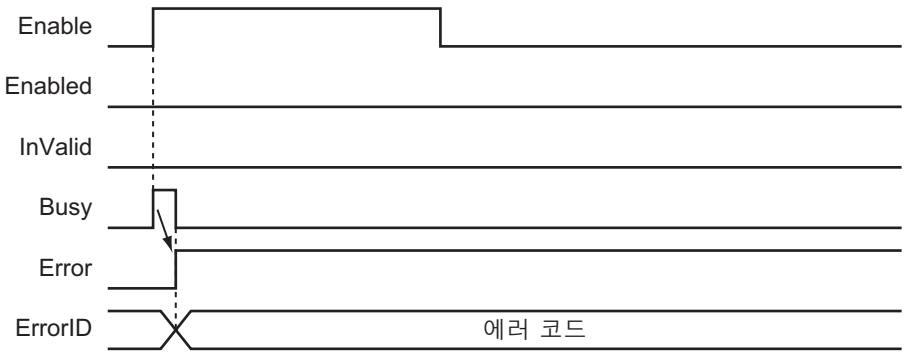
모션 명령 다중 기동

- 다중 기동의 제약은 없습니다.

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

이상

- 이 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
- ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.
- 이 명령의 이상 정보는 MC 공통 모션 변수의 가벼운 이상으로 출력됩니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_SyncMoveVelocity

목표 속도로 설정한 값을 프라이머리 주기마다 주기적 동기 속도 모드(csv)로 서보 드라이버에 출력합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_SyncMoveVelocity	주기적 동기 속도 제어	FB	MC_SyncMoveVelocity_instance MC_SyncMoveVelocity Axis — — Axis Execute — InVelocity Velocity — Busy CmdPosMode — Active BufferMode — CommandAborted — Error — ErrorID	MC_SyncMoveVelocity_instance (Axis := 《파라메터》, Execute := 《파라메터》, Velocity := 《파라메터》, CmdPosMode := 《파라메터》, BufferMode := 《파라메터》, InVelocity => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Velocity	목표 속도	LREAL	음수, 양수, '0'	0	목표 속도를 지정합니다. 0 : 속도 명령값을 '0'으로 합니다. 양수 : 정방향으로 이동합니다. 음수 : 역방향으로 이동합니다. 단위는 [명령 단위/S]입니다.*1
CmdPosMode	명령 현재 위치 카운터 선택	_eMC_CMDPOS_MODE	0: _mcCount	0*2	0 : 피드백 현재 위치를 사용해 명령 현재 위치를 갱신합니다. 원점 확정 상태가 계속됩니다.
BufferMode	버퍼 모드 선택	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0*2	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

*2. 유효 범위가 열거 형식인 변수의 초기값은 실제 초기값이 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
InVelocity	목표 속도 도달	BOOL	TRUE, FALSE	명령 속도가 목표 속도에 도달했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
InVelocity	목표 속도에 도달했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Busy	Execute의 기동	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의해 이 명령이 중지되었을 때 • 다른 명령으로 이상이 발생해 이 명령이 중지되었을 때 • 축 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

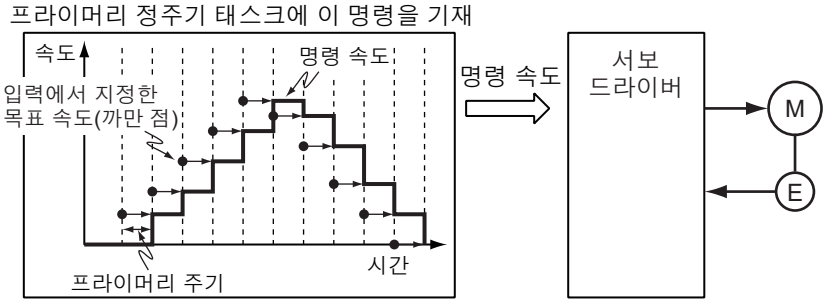
입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

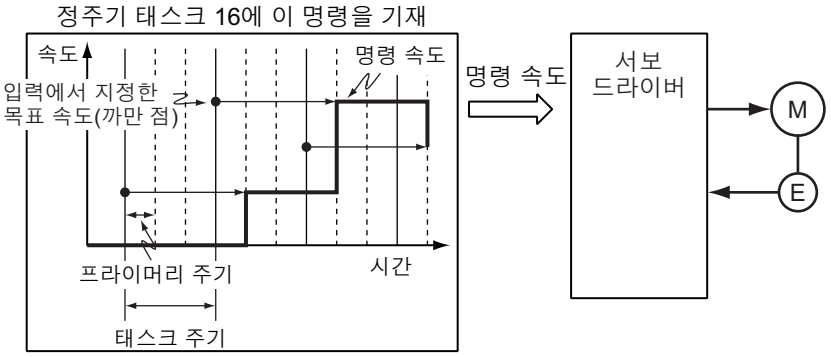
기능 설명

- 이 명령은 사용자 프로그램에서 전달하는 목표 속도를 프라이머리 주기마다 주기적 동기 속도 모드(csv)로 서보 드라이버에 출력합니다.
- Execute(기동)가 기동하면 서보 드라이버의 제어 모드를 전환하는 동시에 명령 속도를 출력합니다.

- 프라이머리 정주기 태스크에 이 명령을 기술하면 다음 주기에서 목표 속도에 도달합니다.



- 정주기 태스크 16에 이 명령을 기술하면 다음 정주기 태스크에서 목표 속도에 도달합니다.



사용상의 주의 사항

이 명령을 동기 제어의 주축으로 사용하는 경우의 주의 사항에 대해서는 '동기 제어의 주축 및 보조축의 주의 사항(1-6페이지)'을 참조하십시오.

참고

- MC_SyncMoveVelocity(주기적 동기 속도 제어) 명령을 사용하면 MC_SetOverride(오버라이드값 설정) 명령은 기능하지 않습니다.
- NX 시리즈 펄스 출력 유니트에 대해서는 이 명령을 사용할 수 없습니다.

● 데이터 오브젝트 매핑

MC_SyncMoveVelocity(주기적 동기 속도 제어) 명령을 사용하려면 Sysmac Studio 축 기본 설정의 [고급 설정]에서 다음 오브젝트 데이터를 매핑하십시오.

- 목표 속도(60FFHex)
- 조작 모드(6060Hex)
- 조작 모드 표시(6061Hex)

이러한 데이터를 설정하지 않으면 프로세스 데이터 오브젝트 설정 부족(에러 코드: 3461Hex)이 발생합니다.

데이터 오브젝트 매핑에 대해서는 📖 '2-3 PDO 매핑(2-27페이지)', 📖 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● Velocity(목표 속도)

입력 변수의 Velocity(목표 속도)는 '0'을 기준으로 한 LREAL 형식으로 설정할 수 있습니다.
양수를 설정하면 정방향으로 이동하고 음수를 설정하면 역방향으로 이동합니다.
'0'을 설정하면 명령 속도가 '0'이 되는데, Status.Continuous(연속 동작 중)는 TRUE인 채로 유지됩니다.
매주기 사용자 프로그램에서 Velocity(목표속도)를 설정할 수 있습니다.
이전 주기와 다른 목표 속도가 지정되면 새로운 목표 속도가 적용됩니다. 이전 주기와 같은 값이면 이전 주기의 목표 속도로 동작합니다.

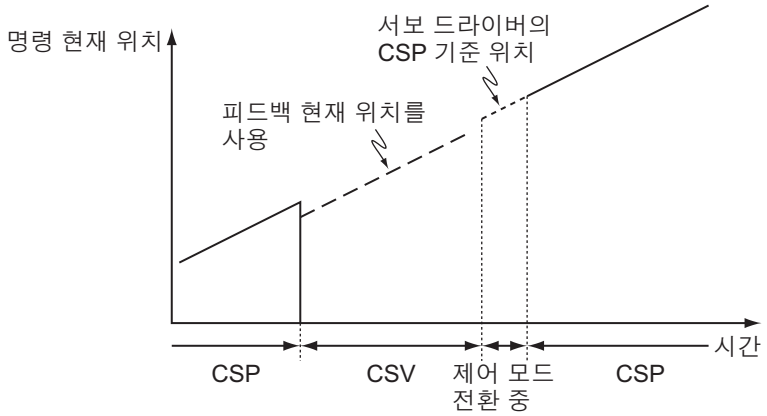
사용상의 주의 사항

- 목표 속도를 설정할 때는 속도 변화로 인해 장치의 기계 기구에 과도한 부담을 주지 않는 값을 지정하십시오.
- 이 명령을 사용하는 축을 동기 제어의 주축으로 사용하면 주축에 대한 목표 속도의 설정에 따라서는 종축이 급격하게 동작할 수 있으므로 주의하십시오.
- 제어 모드 전환 시에 명령 현재 위치가 갑자기 변화하는 경우가 있으므로 주의하십시오.

● 명령 현재 위치

입력 변수의 CmdPosMode(명령 현재 위치 카운트 선택)에서 _mcCount를 선택하면 이 명령 기동 시에 명령 현재 위치는 1주기 전의 피드백 현재 위치가 되고, 이 명령이 종료될 때까지 피드백 현재 위치를 사용합니다.

오므론의 서보 드라이버 G5 시리즈에서는 CSP로의 전환 처리 중에 미리 매핑해 둔 'CSP 기준 위치'가 PDO 송신됩니다. CSV에서 CSP로의 전환 처리가 완료될 때까지 이 기준 위치를 명령 현재 위치로 사용합니다. CSP로의 전환이 완료되면 명령 현재 위치=명령 위치가 됩니다.



● 오므론의 서보 드라이버 G5 시리즈를 사용하는 경우

제어 모드 전환 도중인 서보 드라이버의 'CSP 기준 위치'를 사용하는 경우에는 프로세스 데이터에 CSP용 기준 위치(4020 Hex)를 매핑하십시오.

Sysmac Studio의 PDO 편집 화면에서 CSP용 기준 위치(4020 Hex)를 프로세스 데이터에 매핑합니다. 다음으로 축 기본 설정의 [고급 설정]에서 MC 기능 모듈의 'CSP 변경용 기준 위치'와 CSP용 기준 위치(4020 Hex)를 매핑하십시오.

사용상의 주의 사항

PDO에 'CSP용 기준 위치(4020 Hex)를 매핑한 경우는 프라이머리 주기를 1ms 이상으로 설정하십시오.

프라이머리 주기를 1ms보다 작은 값으로 설정하면 G5 서보 드라이버 측에서 이상이 발생합니다.

자세한 사항은 "G5 시리즈 AC 서보 모터/드라이버 사용 설명서(SBCE-365)"를 참조하십시오.

참고

G5 시리즈 서보 드라이버 중 리니어 모터 타입에서는 CSP용 기준 위치(4020 Hex)에 대응하지 않습니다.

● 오므론의 서보 드라이버 G5 시리즈 이외의 드라이버를 사용하는 경우

이 명령을 기동하는 경우나 이 명령 실행 중에 CSP를 사용하는 명령을 다중 기동하면 서보 드라이버에 따라서는 제어 모드 변경 처리 시에 이상이 발생하는 일이 있습니다.

이 경우는 축을 정지시킨 상태(속도 '0')에서 이 명령을 기동하거나 CSP를 사용하는 명령을 다중 기동하십시오.

● NX 시리즈 위치 인터페이스 유니트를 사용하는 경우

NX 시리즈 위치 인터페이스 유니트에 대해서는 이 명령을 사용할 수 없습니다.

● 정지 처리

축 동작을 정지시키는 경우의 제어 모드나 명령 속도에 대해서 설명합니다.
감속 정지할 경우 이 명령의 목표 속도를 최초 속도로 각 정지 방법이 갖는 감속도를 적용해 감속 정지합니다.

MC_ImmediateStop(즉시 정지) 명령에 의한 정지

명령 속도를 '0'으로 합니다. 다음의 전환 판정식이 성립하면 CSP로 전환합니다.

MC_Stop(강제 정지) 명령에 의한 정지

명령의 감속도에서 명령 속도를 '0'으로 합니다. 다음의 전환 판정식이 성립하면 CSP로 전환합니다.

가벼운 수준의 이상 발생에 의한 정지

각 이상이 정하는 감속도에서 명령 속도를 '0'으로 합니다. 다음의 전환 판정식이 성립하면 CSP로 전환합니다.

전체 정지 수준 및 부분 정지 수준의 이상 발생에 의한 정지

명령 속도를 '0'으로 합니다. 다음의 전환 판정식이 성립하면 CSP로 전환합니다.
단, 이상 수준에 따라서는 제어 모드를 정상적으로 전환하지 못하고 CSV에서 정지하는 경우가 있습니다.

서보 OFF에 의한 정지

소정의 방법으로 명령 속도를 '0'으로 합니다. 제어 모드 전환 처리는 실행하지 않습니다.

CPU 유니트 동작 모드를 프로그램 모드로 전환한 경우의 정지

소정의 방법으로 명령 속도를 '0'으로 합니다. 다음의 전환 판정식이 성립하면 CSP로 전환합니다.

● 전환 판정식

제어 모드의 전환 여부는 서보 드라이버의 사양에 따릅니다.
정지 명령 기동이나 이상 발생에 의한 정지 처리 시에 제어 모드를 확실하게 변경하기 위해서 서보 모터가 충분히 감속한 것을 확인한 후에 CSP로 전환합니다.
명령 속도가 '0'이 된 후 프라이머리 정주기 태스크에서 3회 연속 피드백 현재 속도가 다음의 판정식을 충족하면 CSP로 전환합니다.

$$\text{피드백 현재 속도} \leqq (\text{최고 속도}) \times 0.1$$

● 주기적 동기 위치 모드로의 복귀

정지에 따라서는 CSP로의 변환이 정상적으로 실행되지 않는 경우가 있습니다.
예를 들어 모션 제어 기능 모듈에서 부분 정지 수준의 이상이 발생한 경우 등이 이에 해당합니다.
따라서 MC_Power(운전 가능) 명령의 출력 변수 'Status(운전 가능)'가 FALSE가 되면 CSP 모드로 전환 처리를 실행합니다.

● 제어 모드 전환 실패 시의 동작

서보 드라이버에 대해 제어 모드 전환 명령을 실행한 후 1초 경과해도 서보 드라이버의 전환이 완료되지 않으면 '서보 드라이버의 제어 모드 전환 이상(에러 코드: 7439 Hex)'이 됩니다.

또한 서보 드라이버에 대해 제어 모드 변경을 명령하고 명령 속도가 '0'이 되고 나서 10초 경과해도 전환 판정식이 성립하지 않는 경우도 위와 동일하게 '서보 드라이버의 제어 모드 전환 이상(에러 코드: 7439 Hex)'이 됩니다.

'서보 드라이버의 제어 모드 변경 이상(에러 코드: 7439 Hex)'이 발생하면 명령 속도를 '0'으로 하고 서보를 OFF합니다(프리 런 정지).

'서보 드라이버의 제어 모드 전환 이상(에러 코드: 7439 Hex)'에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 사용 가능 축과 기동 조건

- 서보 축에서 이 명령을 사용할 수 있습니다.
  사용하려면 MC_Power(운전 가능) 명령의 Enable(유효)을 TRUE(서보 ON 상태)로 하십시오.
- 가상 서보 축은 항상 수신할 수 있습니다.
  단, 서보 드라이버의 제어 모드 전환 처리는 실행하지 않습니다.
- 엔코더 축, 가상 엔코더 축에서는 기동 시 이상이 됩니다.

● 축 변수의 Status

축 변수의 Status는 Status.Continuous(연속 정지 중)가 TRUE가 됩니다.

또한 서보 드라이버에 설정된 제어 모드를 확인하려면 축 변수의 DrvStatus(서보 드라이버 상태)를 확인하십시오. 다음에 서보 드라이버 상태를 표시합니다.

변수명	데이터 형식	이 름	설 명
CSP	BOOL	주기적 동기 위치(CSP) 모드 중	서보 ON이고 CSP 모드일 때 TRUE가 됩니다.
CSV	BOOL	주기적 동기 속도(CSP) 모드 중	서보 ON이고 CSV 모드일 때 TRUE가 됩니다.
CST	BOOL	주기적 동기 토크(CST) 모드 중	서보 ON이고 CST 모드일 때 TRUE가 됩니다.

● 원점 상태

CmdPosMode(명령 현재 위치 카운트 선택)가 _mcCount일 때는 원점 확정 상태는 유지되고 변화되지 않습니다.

● 오버라이드

이 명령에 대해 오버라이드는 무효가 됩니다.

● 소프트웨어 리미트

이 명령에 대해 소프트웨어 리미트는 유효가 됩니다.

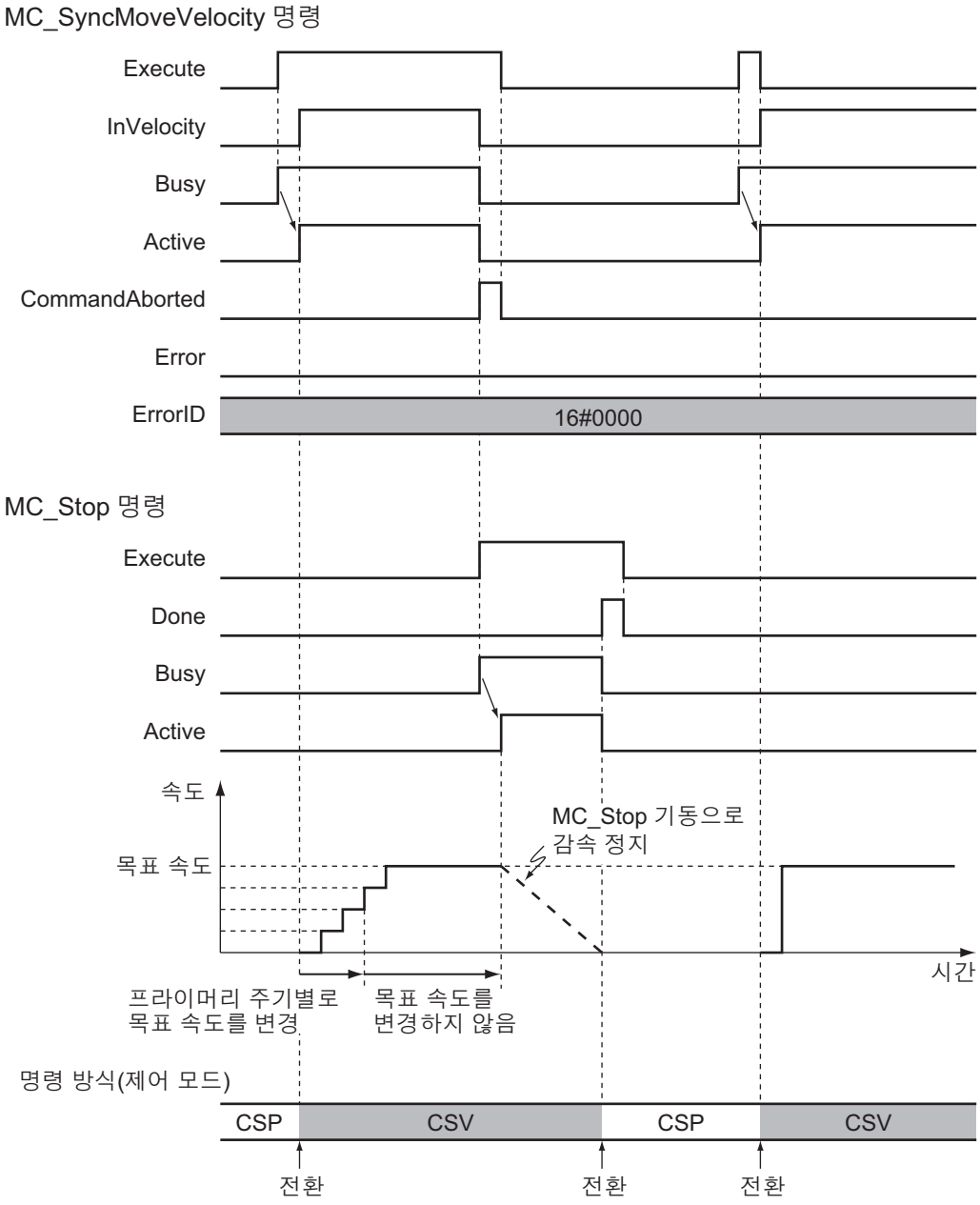
축 파라메터에서 다음 설정을 선택한 경우에도 유효가 됩니다.

- 명령 위치에 대해 유효. 감속 정지
- 명령 위치에 대해 유효. 편차 펄스 정지

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- 명령 속도가 Velocity(목표 속도)에 도달하면 InVelocity(목표 속도 도달)가 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중지되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중), InVelocity(목표 속도 도달)가 FALSE가 됩니다.
- 이 명령을 종료하려면 MC_Stop(강제 정지) 명령을 기동하십시오.

프라이머리 정주기 태스크에 이 명령을 기재한 경우는 다음과 같습니다.



참고

MC 기능 모듈은 서보 드라이버에 대해 위의 타이밍 차트로 제어 모드를 명령합니다. 그러나 서보 드라이버에 반영되는 타이밍은 각 서보 드라이버에 사양에 따릅니다.

모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

MC_MoveVelocity(속도 제어) 명령과 마찬가지로 BufferMode(버퍼 모드 선택)의 Aborting과 Buffered에 대응합니다.
또한 제어 모드의 전환은 다중 기동된 명령의 처리 시작 시에 실행합니다.

BufferMode(버퍼 모드 선택)에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

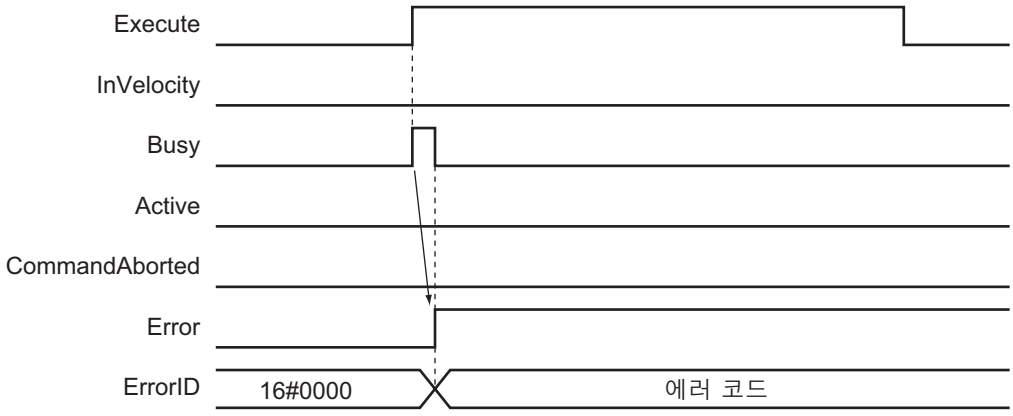
● 이 명령 실행 중의 다른 명령 기동

MC_MoveVelocity(속도 제어) 명령과 마찬가지로 BufferMode(버퍼 모드 선택)의 Aborting과 Buffered에 대응합니다.
Buffered는 InVelocity(목표 속도 도달)가 TRUE가 되었을 때 다른 명령을 기동합니다.
또한 제어 모드의 전환은 다중 기동된 명령의 처리 시작 시에 실행합니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조함으로써 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_SyncMoveAbsolute

축에 지정한 목표 위치를 주기적으로 출력합니다.

명령	이름	FB/FUN	그래픽 표현	ST 표현
MC_SyncMoveAbsolute	주기적 동기 절대 위치 결정	FB	MC_SyncMoveAbsolute_instance MC_SyncMoveAbsolute Axis — Axis Execute — InPosition Position — Busy Direction — Active BufferMode — CommandAborted Error ErrorID	MC_SyncMoveAbsolute_instance (Axis := 《파라메터》 , Execute := 《파라메터》 , Position := 《파라메터》 , Direction := 《파라메터》 , BufferMode := 《파라메터》 , InPosition => 《파라메터》 , Busy => 《파라메터》 , Active => 《파라메터》 , CommandAborted => 《파라메터》 , Error => 《파라메터》 , ErrorID => 《파라메터》);

버전 관련 정보

이 명령은 CPU 유니트 Ver1.03 이상과 Sysmac StudioVer1.04 이상의 조합에서 사용할 수 있습니다.

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Position	목표 위치	LREAL	음수, 양수, 0	0	절대 좌표의 목표 위치를 지정합니다. 단위는 [명령 단위]입니다.*1
Direction	방향 선택	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	1 *2	카운트 모드가 [로터리 모드]인 경우 회전 방식을 지정합니다. *3 0: 정방향 지정 1: 근거리 회전 지정 2: 역방향 지정 3: 현재 방향 지정 4: 방향 지정 안 함
BufferMode	버퍼 모드 선택 *2	_eMC_BUFFER_MODE	0: _mcAborting	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0: Aborting

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.
*3. 회전 방향의 지정 내용에 대해서는 'MC_MoveAbsolute'의 'Direction(방향 선택)(3-51페이지)'을 참조하십시오.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
InPosition	인포지션	BOOL	TRUE, FALSE	피드백 현재 위치가 목표 위치의 인포지션 폭 안에 들어갔을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
InPosition	피드백 현재 위치가 목표 위치의 인포지션 폭 안에 들어갔을 때	• 피드백 현재 위치가 인포지션 폭 밖으로 나왔을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Busy	Execute의 기동	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동으로 이 명령이 중지되었을 때 • 다른 명령으로 이상이 발생해 이 명령이 중지되었을 때 • 축 이상 발생 중에 이 명령을 기동했을 때 • MC_Stop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

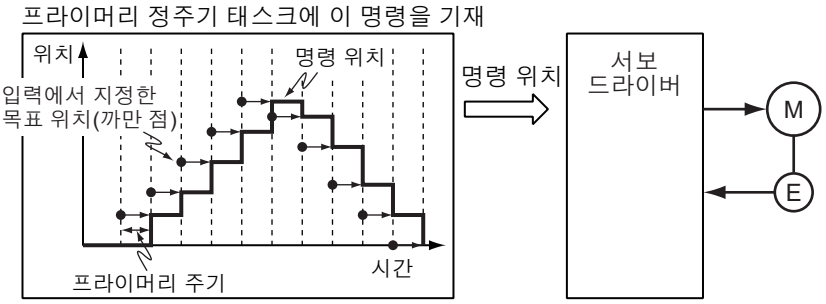
입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

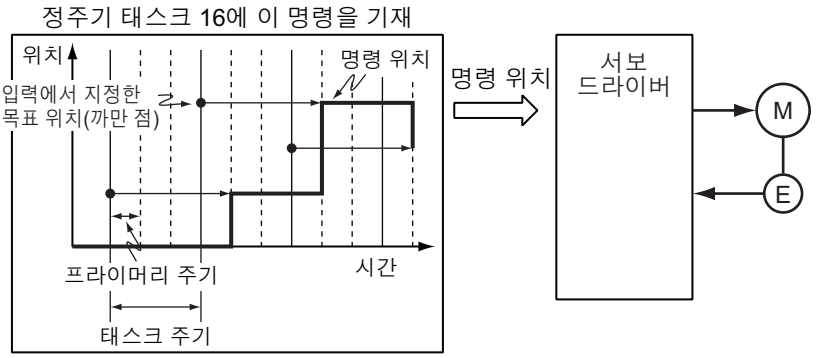
*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- 이 명령은 사용자 프로그램에서 전달하는 목표 위치를 프라이머리 주기마다 주기적 동기 위치 모드(CSP)로 서보 드라이버 등에 출력합니다. 목표 위치는 절대 위치로 전달합니다
- 속도는 축 파라메터의 [최고 속도]를 상한으로 합니다. [최대 가속도]와 [최대 감속도]는 적용되지 않습니다.
- 프라이머리 정주기 태스크에 이 명령을 기술한 경우, 입력에서 지정한 목표 위치는 다음 프라이머리 주기로 서보 드라이버에 출력됩니다.



- 정주기 태스크 16에 이 명령을 기술한 경우, 입력에서 지정한 목표 위치는 다음 정주기 태스크로 서보 드라이버에 출력됩니다.



사용상의 주의 사항

지시한 목표 위치에 의한 이동량이 축 파라메터의 [최고 속도]를 초과하지 않도록 목표 위치를 명령하십시오. [최고 속도]를 초과하는 목표 위치를 지시했을 때는 명령 속도 포화가 되어 [최고 속도]로 제한된 이동량을 출력합니다. 지시한 목표 위치에서 부족한 분량의 이동량 지시는 다음 주기 이후에 출력합니다.
또한 이 때 축 제어 Status의 Details.VelLimit(명령 속도 포화)가 TRUE가 됩니다.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● 인포지션 확인

Position(목표 위치)을 갱신하지 않을 때는 목표 위치와 피드백 위치의 폭이 축 파라메터의 [인포지션 폭] 범위 내가 되었을 때 InPosition(인포지션)이 TRUE가 됩니다.
InPosition(인포지션)이 TRUE일 때 목표 위치를 변경해도 해당 주기 중에는 TRUE인 그대로이며, 다음 주기에 FALSE가 됩니다.
축 파라메터의 [인포지션 확인 시간] 설정은 무효입니다.

● 정지 처리

축 그룹 동작을 정지시킬 때의 동작에 대해 설명합니다.

동작을 정지시키려면 MC_Stop(강제 정지) 명령 또는 MC_ImmediateStop(즉시 정지) 명령을 사용합니다. 이들 명령을 기동하면 이 명령의 CommandAborted(실행 중단)가 TRUE가 됩니다.

MC_Stop(강제 정지) 명령에 의한 정지

즉시 정지합니다.

MC_ImmediateStop(즉시 정지) 명령에 의한 정지

축 파라메터 설정 [즉시 정지 입력 정지 방법]의 지정에 따라 즉시 정지합니다.B

● 이상 발생에 의한 정지

축을 정지시키는 이상이 발생했을 때 설정에 관계 없이 '즉시 정지'합니다.

● 사용 가능 축

- 서보 축에서 이 명령을 사용할 수 있습니다.
  사용하는 경우는 MC_Power(운전 가능) 명령의 Enable(유효)을 TRUE(서보 ON 상태)로 하십시오.
- 가상 서보 축은 항상 수신할 수 있습니다.
- 엔코더 축, 가상 엔코더 축에서는 기동 시 이상이 됩니다.

● 축 변수의 Status

축 변수의 Status는 Status.Discrete(위치 결정 동작 중)이 TRUE가 됩니다.

축 제어 Status에는 영향을 미치지 않습니다.

● 오버라이드

이 명령에 대해 오버라이드는 무효가 됩니다.

● 현재 위치 변경

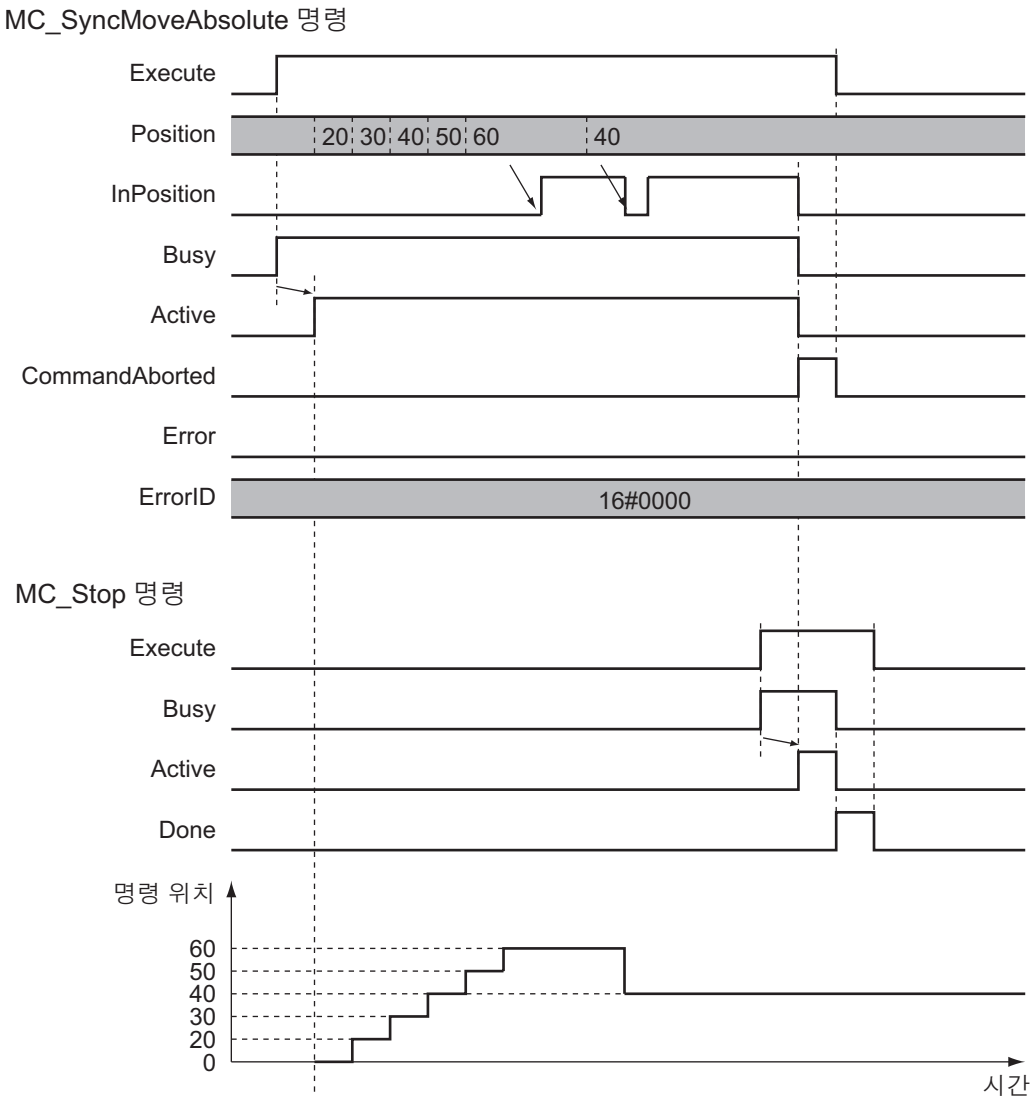
MC_SetPosition(현재 위치 변경) 명령에 의해 현재 위치를 변경했을 때는 변경된 현재 위치에서 Position(목표 위치)에서 지정한 목표 위치로의 이동량을 출력합니다.

현재 위치 변경 전에 InPosition(인포지션)이 TRUE일 때는 현재 위치 변경 후의 다음 주기에 InPosition(인포지션)이 FALSE가 됩니다.

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- 피드백 현재 위치가 Position(목표 위치)의 인포지션 폭 안에 들어갔을 때 InPosition(인포지션)이 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중지되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중), InPosition(인포지션)이 FALSE가 됩니다.
- 이 명령을 종료하려면 MC_Stop(강제 정지) 명령을 기동하십시오.

프라이머리 정주기 태스크에 이 명령을 기재한 경우는 다음과 같습니다.



참고

MC 기능 모듈은 서보 드라이버에 대해 위의 타이밍 차트로 제어 모드를 명령합니다. 그러나 서보 드라이버에 반영되는 타이밍은 각 서보 드라이버에 사양에 따릅니다.

모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동을 수행한 경우 '모션 명령 재기동 불가능(에러 코드: 543B Hex)'이 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

현재 실행 중인 명령에 대해 이 명령을 기동하면 이 명령으로 전환하거나 이 명령에 대한 버퍼링을 실행할 수 있습니다.
각 축에 1개까지 버퍼링할 수 있습니다.
다중 기동 시 이 명령의 동작을 BufferMode(버퍼 모드 선택)에서 지정합니다.

버퍼 모드 선택	설 명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.

BufferMode(버퍼 모드 선택)의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

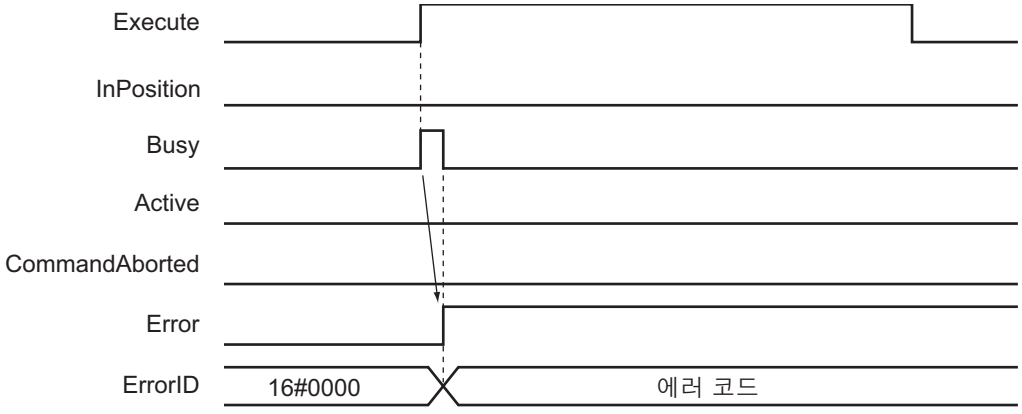
● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하는 경우 Aborting가 가능합니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트

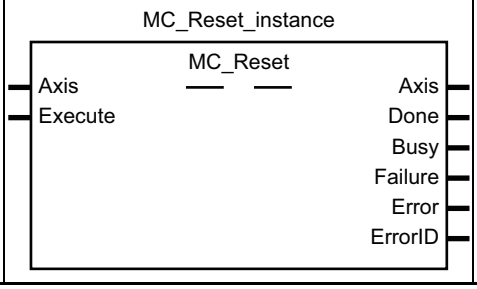


● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_Reset

축 이상을 해제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_Reset	축 에러 리셋	FB		MC_Reset_instance (Axis := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Failure => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Failure	잘못된 종료	BOOL	TRUE, FALSE	명령이 정상 실행되지 않았을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	이상 해제 처리가 정상 종료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • Failure가 TRUE가 되었을 때
Failure	• 이상으로 인한 축의 감속 정지 중에 이상 해제를 실행했을 때 • 축 공통 이상 발생에 의한 축 이상 발생 중에 이상 해제를 실행했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- Execute(기동)가 기동하면 Axis(축)에서 지정한 축의 이상 해제 처리를 시작합니다.
  이상 해제 처리란 축의 이상 해제 및 드라이버 측에서 이상이 발생하는 경우에 드라이버 에러 리셋을 실행하는 것을 말합니다.
  또한 축의 운전 상태에 관계 없이 이상 해제 처리를 실행합니다.
- 이 명령은 모든 축 종류에 대해 사용 가능합니다.
- 이상이 발생한 축에만 이상 해제 처리를 실행합니다.
- 드라이버 에러가 발생한 축에는 드라이버 에러 리셋 처리를 실행한 후 이상 해제 처리를 실행합니다.
  드라이버 에러 리셋 처리는 드라이버 에러가 해제되거나 축 파라메터의 [드라이버 에러 리셋 감시 시간] 동안 계속됩니다.
- 에러 감속 정지 중으로 인한 축 감속 정지 중에 이 명령을 실행하면 실행되지 않고 Failure(잘못된 종료)가 TRUE가 됩니다. 이것은 축이 정지하기 전에 이상을 해제할 수 없도록 하기 위한 것입니다.
  또한 MC 공통 이상 발생에 의한 축 이상도 이 명령으로는 해제할 수 없으므로 Failure(잘못된 종료)가 TRUE가 됩니다.
  MC 공통 이상에는 'MC 공통 부분 정지 이상' 및 'MC 공통 가벼운 이상'이 있습니다.
- 해제되는 이상은 Execute(기동)의 기동 타이밍에서 발생한 이상을 대상으로 합니다. 해제 중에 발생한 이상에 대해서는 해제되지 않습니다.

사용상의 주의 사항

- 이 명령에 의한 이상 해제 처리는 여러 제어 주기에 걸치는 경우가 있습니다.
- 'MC 공통 부분 정지 이상' 및 'MC 공통 가벼운 이상'의 공통 이상이 발생하거나 축이 동작 중인 경우 이 명령의 Failure(잘못된 종료)가 TRUE가 됩니다. 이상 요인을 제거한 뒤 Done(완료)이 될 때까지 재시도 처리하십시오.
- 이상 발생 요인을 제거하려면 축이 완전히 정지한 것을 확인한 후 실행하십시오.
  축이 완전히 정지한 것을 확인하려면 축 변수의 Act. Vel(피드백 현재 속도)이 '0'이 된 것을 확인해야 합니다.
- 오므론 서버 드라이버 G5 시리즈에 대해 이 명령을 사용하는 경우, ResetECError(EtherCAT 이상 해제) 명령이 동시에 실행되지 않도록 배타적 처리를 하십시오.
- NX 시리즈 펄스 출력 유니트에 대해 이 명령을 사용하는 경우, 펄스 출력 유니트에 연결된 드라이버의 이상 상태는 해제되지 않습니다. 자세한 내용은 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"를 참조하십시오.

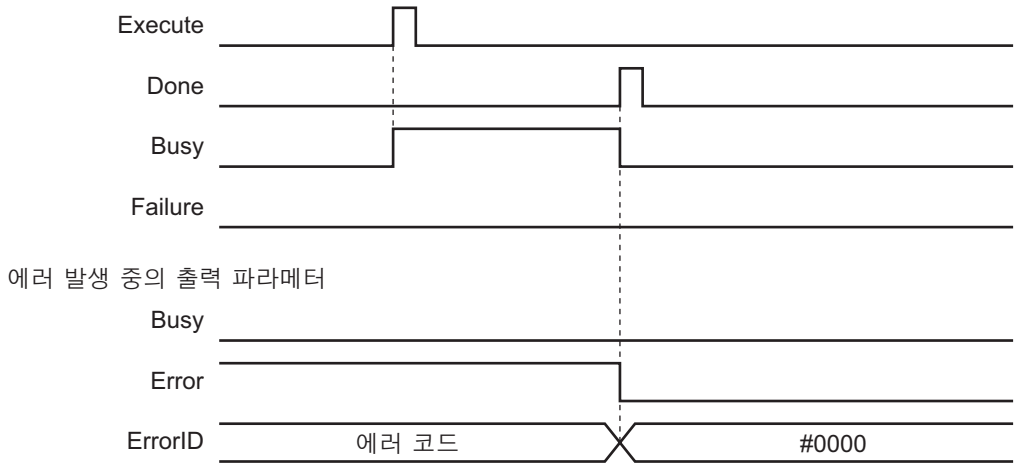
참고

이 명령으로는 다음 이상을 해제할 수 없습니다.

- 축 공통 이상 전체 : ResetMcError(전체 에러 리셋) 명령을 실행하십시오.
- 축 그룹 이상 전체 : MC_GroupReset(축 그룹 에러 리셋) 명령을 실행하십시오.

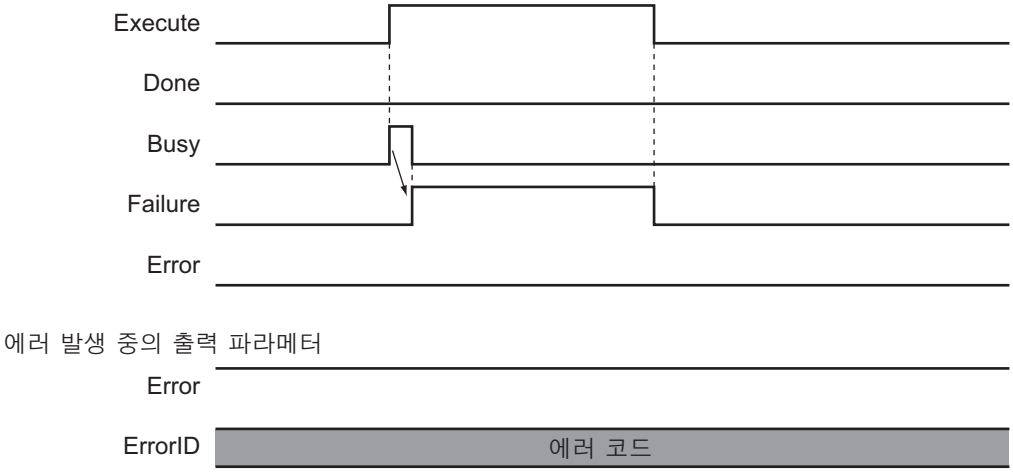
또한 이 명령을 실행해도 슬레이브 통신 이상 등의 네트워크에 관한 이상 요인은 해제할 수 없습니다. ResetECATError 명령을 실행하십시오.

타이밍 차트



명령 중지

이상에 의한 축의 감속 정지 중 또는 축 공통 이상 발생에 의한 축 이상 발생 중의 이상을 해제할 수 없는 경우는 명령을 중단합니다.



에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_ChangeAxisUse

축 파라메터의 [축 사용]을 일시적으로 전환할 수 있습니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_ChangeAxisUse	축 사용 변경	FB	MC_ChangeAxisUse_instance MC_ChangeAxisUse Axis — — Axis Execute — Done AxisUse — Busy CommandAborted Error ErrorID	MC_ChangeAxisUse_instance (Axis := 《파라메터》, Execute := 《파라메터》, AxisUse := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

사용상의 주의 사항

이 명령으로 전환한 값은 CPU 유니트의 비휘발성 메모리에 저장되지 않습니다. 따라서 컨트롤러의 전원 'OFF' 및 다운로드, MC 기능 모듈의 재기동 처리가 시작되면 전환한 값은 지워지고 Sysmac Studio에서 설정한 값으로 되돌아갑니다.
비휘발성 메모리에 저장하려면 Sysmac Studio를 사용해 파라메터를 전송하십시오.

참고

- '다운로드'는 Sysmac Studio의 '동기' 기능을 사용합니다.
- 이 명령을 사용하는 응용 프로그램의 예는 "NJ 시리즈 CPU 유니트 사용 설명서 소프트웨어편(SBCA-359E 이상)"을 참조하십시오.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.04 이상과 Sysmac Studio Ver.1.05 이상의 조합에서 사용할 수 있습니다.

변수

입력 변수

입력 변수	이름	데이터 형식	유효 범위	초기값	내용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시 명령을 시작합니다.
AxisUse	축 사용	_eMC_AXIS_USE	1: _mcUnusedAxis 2: _mcUsedAxis	1*1	사용 축 또는 미사용 축을 지정합니다. 1: 미사용 축 2: 사용 축

*1. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이름	데이터 형식	유효 범위	내용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령이 완료되었을 때	• Execute가 TRUE인 경우, Execute가 FALSE로 변경되는 동시에 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	다른 명령에서 이상이 발생하여 이 명령이 중지되었을 때	• Execute가 TRUE인 경우, Execute가 FALSE로 변경되는 동시에 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이름	데이터 형식	유효범위	내용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.

기능 설명

- Execute(기동)의 기동 시 AxisUse(축 사용)을 통해 지정된 축 사용으로 전환합니다.
- 명령의 실행 완료와 동시에 축 변수인 _MC_AX[ ].Cfg.AxEnable(축 사용)이 지정된 축 사용으로 전환됩니다.
- 전환 가능한 축은 축 파라메터의 [축 사용]이 [미사용 축(사용 축으로 전환 가능)] 또는 [사용 축]으로 설정되어 있는 축입니다. [미사용 축(사용 축으로 전환 불가능)]으로 설정되어 있는 축을 [사용 축]으로 전환하지 마십시오.
- 원점 확정과 원점 미확정 중 어떤 상태에서도 명령을 실행할 수 있습니다. 원점 확정일 때 미사용 축으로 전환하면 원점 미확정이 됩니다.
- 앱솔루트 엔코더를 사용하고 있는 축을 사용 축에서 미사용 축으로 전환한 경우, CPU 유니트의 전원 OFF 시 CPU 유니트 내장 배터리 백업 메모리에 저장되는 [앱솔루트 엔코더 원점 위치 오프셋]은 미사용 축으로 전환되기 직전의 값입니다.

사용상의 주의 사항

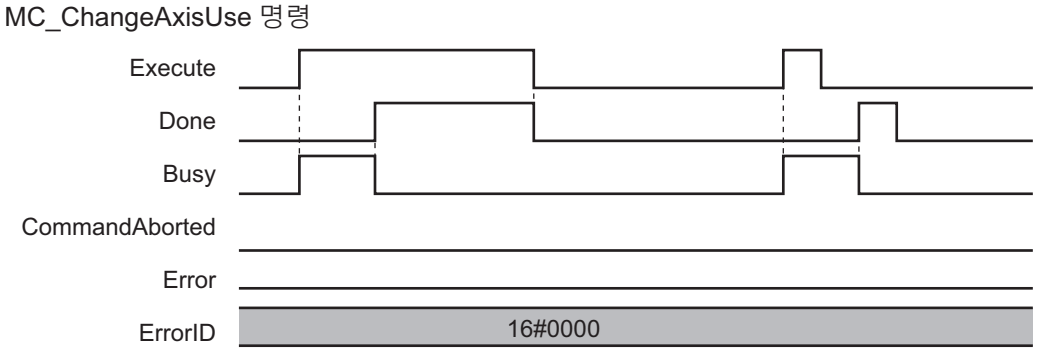
- CPU 유니트의 형식별 제어 축 최대 수를 초과하는 축 번호의 축을 사용 축으로 전환할 수는 없습니다. 또한 사용 축으로 전환 가능한 실제 축의 수는 사용하는 실제 축의 최대 수까지입니다.

사양 항목	NJ501-15□□	NJ501-14□□	NJ501-13□□	NJ301-12□□	NJ301-11□□
사용 축으로 전환 가능한 축 번호	0~63	0~31	0~15	0~14*1	0~14*2
사용하는 실제 축의 최대 수	64축	32축	16축	8축	4축

*1. CPU 유니트 Ver.1.05 이하는 '0~7'입니다.
*2. CPU 유니트 Ver.1.05 이하는 '0~3'입니다.

- 축 변수인 _MC_AX[ ].Status.Disabled(축 무효) 또는 _MC_AX[ ].Status.Standstill(정지 중)이 TRUE일 때만 명령을 실행할 수 있습니다. FALSE일 때 실행하면 이상이 됩니다.
- 축 변수인 _MC_AX[ ].Details.VelLimit(명령 속도 포화)가 TRUE일 때 실행하면 이상이 됩니다.
- 축 사용이 _mcUnusedAxis(미사용 축)로 되어 있는 축에 MC_write(MC 설정 쓰기) 명령으로 축 파라메터를 재기입해도 무효입니다. 축 사용을 _mcUsedAxis(사용 축)로 전환한 후 MC_write(MC 설정 쓰기) 명령을 실행하십시오.
- 이 명령을 통해 미사용 축으로 전환한 축이 포함된 축 그룹에 MC_GroupEnable(축 그룹 유효) 명령을 실행하면 이상이 됩니다.
- 사용 축에 대해 명령을 실행했을 때 이상이 발생한 경우는 축 이상이 되고, 미사용 축에 대해 명령을 실행했을 때 이상이 발생한 경우는 MC 공통 이상이 됩니다.

타이밍 차트



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동을 수행한 경우 '모션 명령 재기동 불가능(에러 코드: 543B Hex)'이 됩니다.

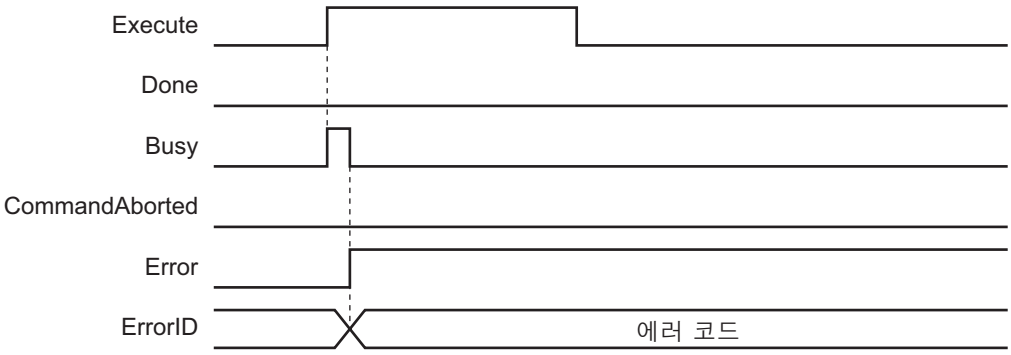
모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

이상

이 명령 기동 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축 사용은 전환되지 않습니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 🕮 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

사용상의 주의 사항

사용 축에 대해 명령을 실행했을 때 이상이 발생한 경우는 축 이상이 되고, 미사용 축에 대해 명령을 실행했을 때 이상이 발생한 경우는 MC 공통 이상이 됩니다.

MC_DigitalCamSwitch

축의 위치에 따라 디지털 출력을 ON 또는 OFF합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
MC_DigitalCamSwitch	디지털 캠 스위치 유효	FB	MC_DigitalCamSwitch_instance MC_DigitalCamSwitch Axis — — Axis Switches — — Switches Outputs — — Outputs TrackOptions — — TrackOptions Enable — InOperation EnableMask — Busy ValueSource — Error ErrorID	MC_DigitalCamSwitch_instance (Axis := 《파라메터》 , Switches := 《파라메터》 , Outputs := 《파라메터》 , TrackOptions := 《파라메터》 , Enable := 《파라메터》 , EnableMask := 《파라메터》 , ValueSource := 《파라메터》 , InOperation => 《파라메터》 , Busy => 《파라메터》 , Error => 《파라메터》 , ErrorID => 《파라메터》);

사용상의 주의 사항

- 이 명령은 NX 시리즈 위치 인터페이스 유니트를 할당한 축에 대해 유효합니다. 사용할 수 있는 NX 유니트는 NX-EC0□□□, NX-ECS□□□입니다.
- 이 명령은 반드시 NX_AryDOutTimeStamp 명령 및 타임 스탬프 방식을 지원하는 디지털 출력 유니트와 조합하여 사용하십시오.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용할 수 있습니다.

변수

입력 변수

입력 변수	이름	데이터 형식	유효 범위	초기값	내용
Enable	유효	BOOL	TRUE, FALSE	FALSE	TRUE인 동안 명령을 실행합니다.FALSE일 때는 'Outputs'의 값을 갱신하지 않습니다.*1
EnableMask	트랙 유효	WORD	16#0000~FFFF	16#0000	트랙별로 유효 또는 무효를 지정합니다. 최대 16트랙으로, 0번째 비트의 값이 트랙 번호 0, 15번째 비트의 값이 트랙 번호 15의 유효 또는 무효를 지정합니다. 0:무효*2 1:유효
ValueSource (Reserved)	입력 정보	_sMC_SOURCE	–	–	(Reserved)

*1. Enable이 FALSE인 경우 Outputs(출력 신호)의 값은 유지됩니다. 또한 NX_AryDOutTimeStamp 명령의 Enable이 FALSE인 경우 디지털 출력 유니트의 디지털 출력은 OFF가 됩니다.

*2. EnableMask의 비트 값이 '0'일 때 해당 트랙 번호의 Outputs(출력 신호)는 OFF가 됩니다.

출력 변수

출력 변수	이름	데이터 형식	유효 범위	내용
InOperation	작동 중	BOOL	TRUE, FALSE	유효한 출력 신호의 출력 중에 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'를 참조하십시오.

● 출력 변수의 반영 타이밍

변수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
InOperation	Enable이 TRUE가 되었을 때	• Enable이 FALSE가 되었을 때 • Error가 TRUE가 되었을 때
Busy	Enable의 기동	• Enable이 FALSE가 되었을 때 • Error가 TRUE가 되었을 때
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이름	데이터 형식	유효 범위	내용
Axis	축	_sAXIS_REF	–	위치를 참조할 축을 지정합니다. *1
Switches	스위치	ARRAY[0..255] OF _sCAMSWITCH_REF *2	–	스위치 구조체 _sCAMSWITCH_REF 형식의 배열 변수를 스위치의 ON/OFF 패턴 데이터로 지정합니다. 배열 요소 번호는 스위치 번호를 나타냅니다.
Outputs	출력 신호	ARRAY[0..15] OF _sOUTPUT_REF*2 *3	–	스위치의 ON/OFF 패턴 데이터를 바탕으로 산출된 디지털 출력의 ON/OFF 시각의 출력 대상으로 출력 신호 _sOUTPUT_REF 형식의 배열 변수를 지정합니다. 배열 요소 번호는 트랙 번호를 나타냅니다. 이 배열 변수를 NX_AryDOutTimeStamp 명령의 입출력 변수로 지정하면 실제 디지털 출력을 ON/OFF합니다.
TrackOptions	트랙 옵션	ARRAY[0..15] OF _sTRACK_REF*2 *3	–	트랙 옵션 구조체 _sTRACK_REF 형식의 배열 변수를 스위치의 동작 조건으로 지정합니다. 배열 요소 번호는 트랙 번호를 나타냅니다.

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.
*2. 배열 요소의 시작 번호가 '0' 이외인 경우는 이상이 됩니다. 또한 2차원 이상의 배열인 경우도 이상이 됩니다.
*3. Outputs의 요소 수와 TrackOptions의 요소 수가 일치하지 않는 경우는 이상이 됩니다.

기능 설명

- 이 명령은 축의 피드백 위치가 'Switches'에 설정된 위치에 도달하는 시각을 디지털 출력이 ON/OFF하기 위한 시각으로 출력합니다.
- 이 명령은 반드시 NX_AryDOutTimeStamp 명령 및 타임 스탬프 방식을 지원하는 디지털 출력 유니트와 조합하여 사용하십시오.

- NX_AryDOutTimeStamp 명령은 지정된 타임 스탬프에서 지정된 디지털 출력을 ON/OFF합니다.타임 스탬프 방식을 사용하면 제어 처리 타이밍의 영향을 받지 않고 임의의 타이밍에 디지털 출력 신호를 ON/OFF할 수 있습니다.

사용상의 주의 사항

- 이 명령만으로는 디지털 출력을 ON/OFF할 수 없습니다.
- 이 명령은 NX 시리즈 엔코더 입력 유니트를 사용하고, 변화 시각 취득 기능이 동작해야 합니다.
  다음과 같은 경우, 변화 시각 취득 기능은 동작하지 않습니다.
  - 변화 시각 취득 기능에 대응하지 않는 엔코더 입력 또는 서보 드라이버를 사용하고 있는 경우
  - NX 시리즈 엔코더 입력 유니트의 개체 6010Hex(타임 스탬프) 를 PDO 에 할당하지 않은 경우
  - EtherCAT 커플러 유니트의 [DC 유효] 를 [무효] 로 설정한 경우
- 타임 스탬프가 갱신되지 않은 경우에도 이 명령은 이상이 발생하지 않습니다. 이 때도 ON/OFF 시각이 산출되지만 의도한 값이 되지는 않습니다.
  Sysmac Studio의 MC 모니터 테이블 또는 감시 창에서 축 변수의 멤버 'TimeStamp(타임 스탬프)'가 갱신되었는지 확인한 후 이 명령을 사용하십시오.
- NX_AryDOutTimeStamp 명령 및 타임 스탬프 방식을 지원하는 디지털 출력 유니트와 조합하여 사용하는 경우, 최소 ON/OFF 폭은 태스크 주기의 값 및 회전 속도의 값에 비례합니다. 예를 들어 로터리 테이블의 1회전이 360°이고 회전 속도가 800r/min, 태스크 주기가 500 μs인 경우, 최소 ON/OFF 폭은 5°가 됩니다. 태스크 주기만 1000 μs로 변경한 경우, 최소 ON/OFF 폭은 10°가 됩니다.
- 스위치 구조체 형식 변수의 FirstOnPosition, LastOnPosition 및 Duration의 값은 디지털 출력의 ON/OFF 폭이 최소 ON/OFF 폭보다 커지도록 지정하십시오. 최소 ON/OFF 폭보다 작아지는 경우는 실제 디지털 출력이 ON/OFF되지 않을 수 있습니다.
- 이 명령은 축의 현재 위치와 현재 속도를 바탕으로 지정된 위치에 도달하는 타임 스탬프를 산출합니다. 산출된 타임 스탬프의 정밀도는 엔코더 분해능 및 축의 회전 속도의 영향을 받습니다. 엔코더 분해능이 낮은 경우나 축의 회전 속도가 낮은 경우에 오차가 커집니다. 최대 오차의 기준은 다음 식으로 계산할 수 있습니다.

  타임 스탬프의 최대 오차(s)
  = 180/[엔코더 분해능(펄스/회전)×회전 속도(r/min)]

  엔코더 분해능과 회전 속도에 대한 타임 스탬프의 최대 오차의 예는 다음과 같습니다.

엔코더 분해능 (펄스/회전)	회전 속도 (r/min)	산출된 타임 스탬프의 최대 오차 (μs)
3600	400	±125.0
	800	±62.5
131072	400	±3.4
	800	±1.7

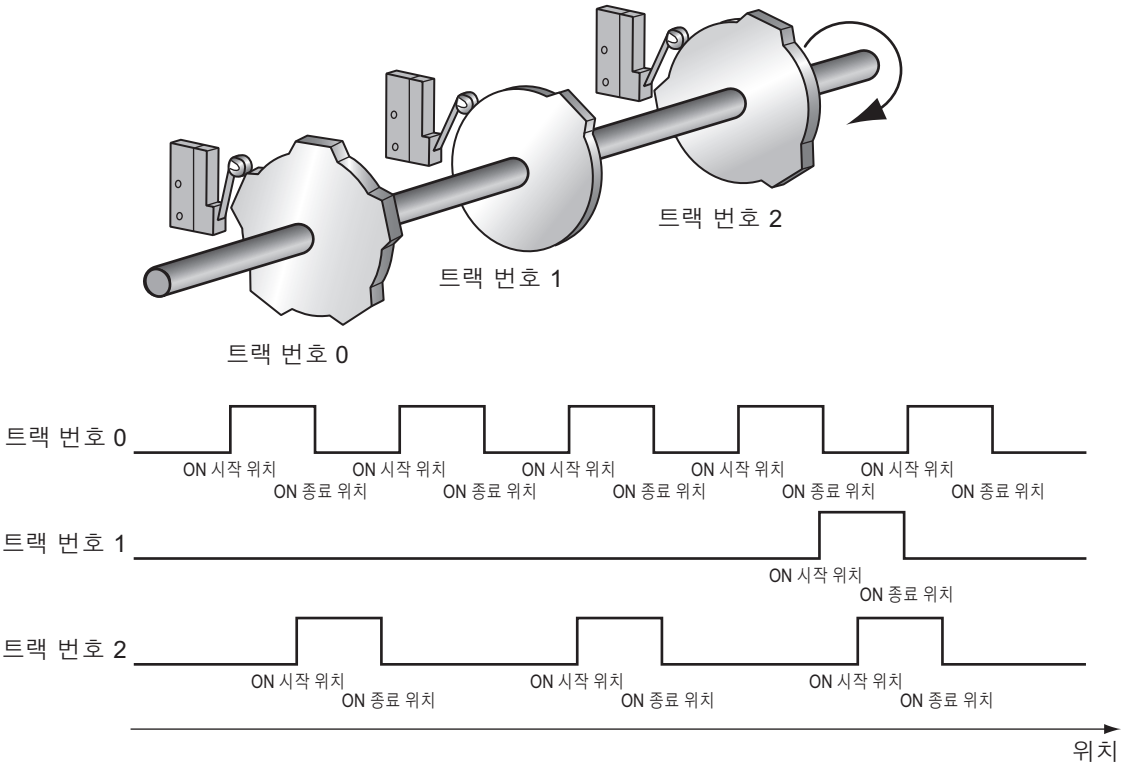
  또한 축이 급격하게 가감속하면 산출 오차가 더 커질 수 있으므로 축이 일정 속도인 상태에서 사용하십시오.
  동작 검증을 충분히 실시하여 안전을 확인하십시오.
- 사용하지 않는 축을 지정한 경우 또는 MC 시운전 중인 경우, Enable이 TRUE일 때 Busy가 TRUE가 되고, InOperation과 Error는 FALSE가 됩니다.
- 이름이 같은 인스턴스를 2개 이상 배치하지 마십시오. 배치하면 의도하지 않은 출력이 되는 경우가 있습니다.

참고

NX_AryDOutTimeStamp 명령에 대해서는 'NJ 시리즈 명령 참조 기본편(SBCA-360)'을 참조하십시오.

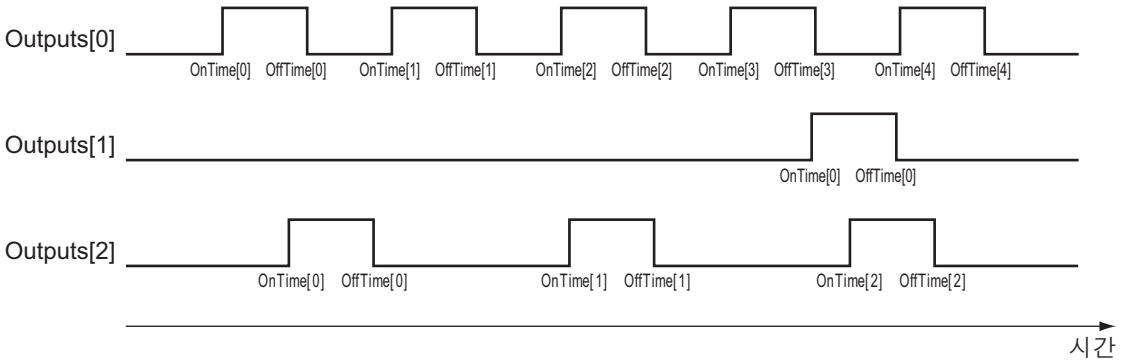
명령에 대한 자세한 내용

이 명령은 캠의 회전 각도에 따라 센서의 출력 신호를 제어하는 기계식 캠 스위치의 기능을 컨트롤러의 프로그램으로 실현합니다. 하나의 트랙은 하나의 캠에 해당합니다.

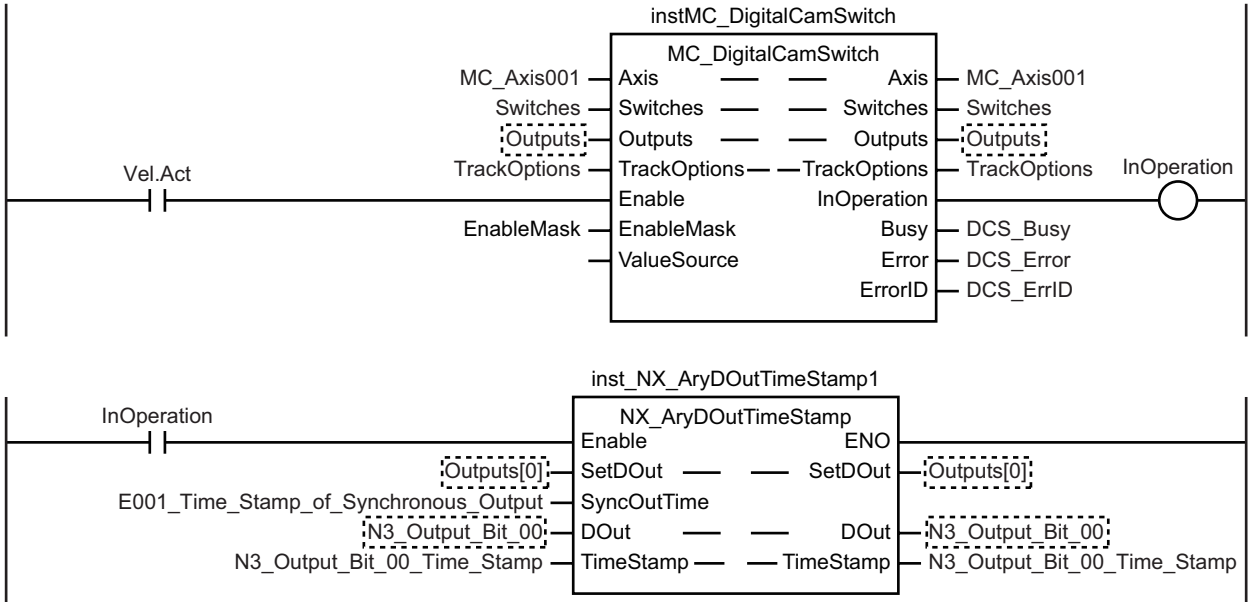


TrackNumber(트랙 번호)는 캠의 번호에 해당합니다. FirstOnPosition(ON 시작 위치), LastOnPosition(ON 종료 위치)은 캠의 형상에 해당합니다.

MC_DigitalCamSwitch 명령은 설정된 ON 시작 위치/ON 종료 위치에 도달하는 OnTime(ON 시각)/OffTime(OFF 시각)을 산출하여 Outputs(출력 신호)로 지정된 파라메터에 저장합니다.



NX_AryDOutTimeStamp 명령은 SetDOut(출력 펄스)으로 지정된 파라메터의 OnTime(ON 시각)/OffTime(OFF 시각)에 따라 실제 디지털 출력을 ON/OFF합니다.
SetDOut(출력 펄스)에는 MC_DigitalCamSwitch 명령의 Outputs(출력 신호)의 파라메터로 지정한 배열 변수의 요소를 지정합니다.
DOut(DOut 유니트 출력 접점)에는 실제 디지털 출력의 지정으로 타임 스탬프 방식을 지원하는 디지털 출력 유니트의 출력 접점에 할당된 디바이스 변수를 지정합니다.



다음으로 명령의 변수에 대해 설명합니다.

● Enable(유효)

- Enable(유효)이 TRUE인 동안 명령을 실행합니다. FALSE일 때는 'Outputs'의 값을 갱신하지 않습니다.

● EnableMask(트랙 유효)

- Enable(유효)이 TRUE일 때, EnableMask(트랙 유효)를 통해 트랙별 유효 또는 무효를 지정할 수 있습니다. 디지털 출력 유니트의 출력을 OFF하는 경우는 대상 트랙을 무효로 하십시오.
- 0번째 비트가 트랙 번호 0을 나타내며, 15번째 비트가 트랙 번호 15를 나타냅니다. 각 비트의 값을 '1'로 하면 대상 트랙이 유효가 되고, '0'으로 하면 무효가 됩니다. 비트의 값을 '1'에서 '0'으로 하면 대상 트랙의 디지털 출력은 OFF가 됩니다.
- EnableMask에 지정된 값은 해당하는 트랙 번호의 EnableOut에 반영됩니다.

● 스위치 구조체(_sCAMSWITCH_REF 형식)

스위치 구조체(_sCAMSWITCH_REF 형식)는 출력 신호의 ON/OFF 패턴을 지정하기 위한 구조체형 데이터 형식입니다. 이 명령에서는 배열 변수로 최대 256개의 ON/OFF 패턴을 지정할 수 있습니다. 1개의 트랙에서는 최대 16개의 ON/OFF 패턴을 지정할 수 있습니다.

다음으로 스위치 구조체의 멤버에 대해 설명합니다.

변수	명칭	데이터 형식	유효 범위	초기값	내용
TrackNumber	트랙 번호	UINT	0~15*1	0	대상이 되는 트랙 번호를 지정합니다.
FirstOnPosition	ON 시작 위치	LREAL	음수, 양수, '0'	0	출력이 ON되는 위치를 지정합니다.*2
LastOnPosition	ON 종료 위치	LREAL	음수, 양수, '0'	0	출력이 OFF되는 위치를 지정합니다.*2 스위치 모드 선택을 위치 기준으로 지정한 경우에 유효가 됩니다.
AxisDirection	축 방향 선택	_eMC_DIRECTION	0:_mcPositiveDirection 2:_mcNegativeDirection 4:_mcNoDirection	0*3	회전 방향을 지정합니다. 0: 정방향 지정 2: 역방향 지정 4: 방향 지정 없음(양방향)
CamSwitchMode	스위치 모드 선택	_eMC_SWITCH_MODE	0:_mcSwitchDisable 1:_mcPositionBased 2:_mcTimeBased	0*3	스위치 모드를 지정합니다. 0: 스위치 무효 1: 위치 기준*4 2: 시간 기준*5
Duration	ON 시간	TIME	양수, T#0s	T#0s	출력이 ON되는 시간을 지정합니다. 스위치 모드 선택을 시간 기준으로 지정한 경우에 유효가 됩니다.

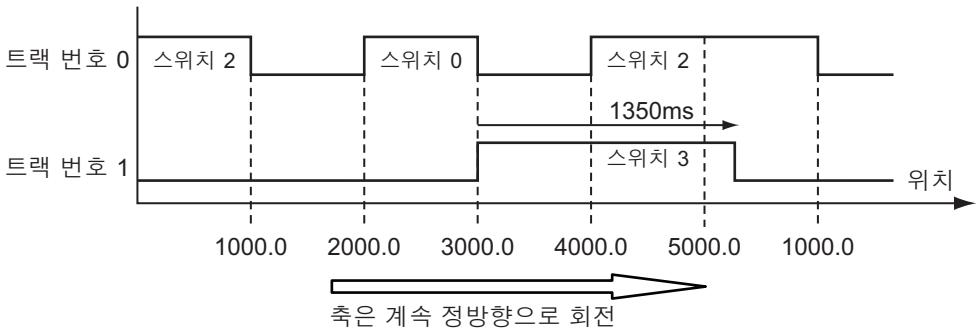
*1. Outputs(출력 신호)에 지정된 변수의 요소 번호까지 지정 가능합니다.
*2. 명령 단위가 사용됩니다. 명령 단위는 [mm], [μm], [nm], [degree], [inch] 또는 [pulse]로 나타냅니다.
카운트 모드가 [리니어 모드]인 경우, 설정 범위는 펄스 단위 변환 후 부호가 있는 정수 형식 40비트(0x8000000000 ~ 7FFFFFFFFF)입니다. 카운트 모드가 [로터리 모드]인 경우, 설정 범위는 링 카운터 하한값 이상, 링 카운터 상한값 미만입니다.
*3. 유효 범위가 열거 형식인 변수는 실제 초기값이 숫자가 아니라 열거자가 됩니다.
*4. 1:_mcPositionBased(위치 기준)를 지정한 경우, FirstOnPosition(ON 시작 위치)과 LastOnPosition(ON 종료 위치)의 값에 따라 동작합니다. Duration(ON 시간)의 값은 관계 없습니다.
*5. 2: _mcTimeBased(시간 기준)을 지정한 경우는 FirstOnPosition(ON 시작 위치)과 Duration(ON 시간)의 값에 따라 동작합니다. LastOnPosition(ON 종료 위치)의 값은 관계 없습니다.

설정 예

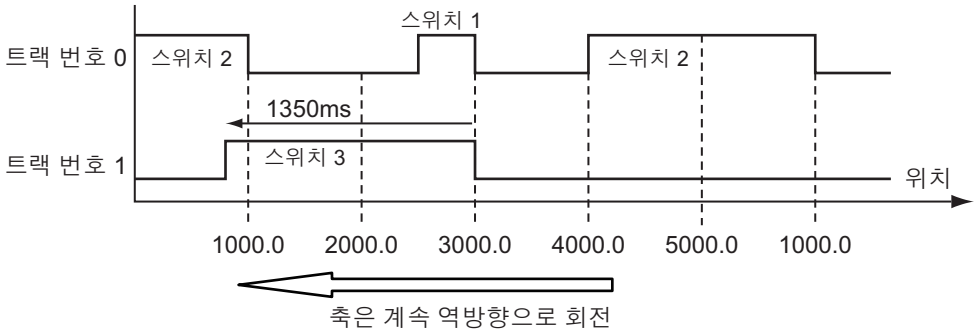
변수	명칭	스위치 0	스위치 1	스위치 2	스위치 3	· · ·	스위치 255
TrackNumber	트랙 번호	0	0	0	1		
FirstOnPosition	ON 시작 위치	2000.0	2500.0	4000.0	3000.0		
LastOnPosition	ON 종료 위치	3000.0	3000.0	1000.0	_*1		
AxisDirection	축 방향 선택	정방향 지정	역방향 지정	방향 지정 없음	방향 지정 없음		
CamSwitchMode	스위치 모드 선택	위치 기준	위치 기준	위치 기준	시간 기준		
Duration	ON 시간	_*2	_*2	_*2	T#1350ms		

*1. 스위치 모드 선택이 시간 기준인 경우, ON 시작 위치와 ON 시간에 동작합니다. ON 종료 위치는 관계 없습니다.
*2. 스위치 모드 선택이 위치 기준인 경우, ON 시작 위치와 ON 종료 위치에서 동작합니다. ON 시간은 관계 없습니다.

축이 계속 정방향으로 회전하고 있는 경우의 동작은 다음과 같습니다. 축 파라메터의 카운트 모드는 로터리 모드이고 링 카운터의 범위는 0~5000입니다.



축이 계속 정/역방향으로 회전하고 있는 경우의 동작은 다음과 같습니다. 축 파라메터의 설정은 위와 동일합니다.



● 출력 신호 구조체(_sOUTPUT_REF 형식)

출력 신호 구조체(_sOUTPUT_REF)는 스위치의 ON/OFF 패턴 데이터를 바탕으로 산출된 디지털 신호의 ON/OFF 시각을 표시하기 위한 구조체형 데이터 형식입니다. 이 명령에서는 Outputs(출력 신호)의 배열 변수로 최대 16개의 요소를 취급합니다.Outputs(출력 신호)의 배열 요소 번호는 트랙 번호를 나타냅니다.

다음으로 출력 신호의 멤버에 대해 설명합니다.

변수	명칭	데이터 형식	유효 범위	내용
EnableOut	출력 유효	BOOL	TRUE, FALSE	해당 트랙 번호의 출력의 유효/무효를 나타냅니다. EnableMask의 동일한 트랙 번호에 해당하는 비트의 값이 반영됩니다.*1 TRUE: 해당 트랙 번호의 출력이 유효 FALSE: 해당 트랙 번호의 출력이 무효
OnTime	ON 시각	ARRAY[0..15] OF ULINT	양수, '0'	디지털 출력을 ON하는 타임 스탬프를 나타냅니다. NX 시리즈 엔코더 입력 유니트 내의 시각을 기준으로 하는 타임 스탬프를 나타냅니다. 태스크 주기마다 값이 갱신됩니다. 단위는 [ns]입니다.
OffTime	OFF 시각	ARRAY[0..15] OF ULINT	양수, '0'	디지털 출력을 OFF하는 타임 스탬프를 나타냅니다. NX 시리즈 엔코더 입력 유니트 내의 시각을 기준으로 하는 타임 스탬프를 나타냅니다. 태스크 주기마다 값이 갱신됩니다. 단위는 [ns]입니다.

*1. EnableMask의 비트의 값이 Outputs[i].EnableOut에 반영됩니다.

● 트랙 옵션 구조체(_sTRACK_REF 형식)

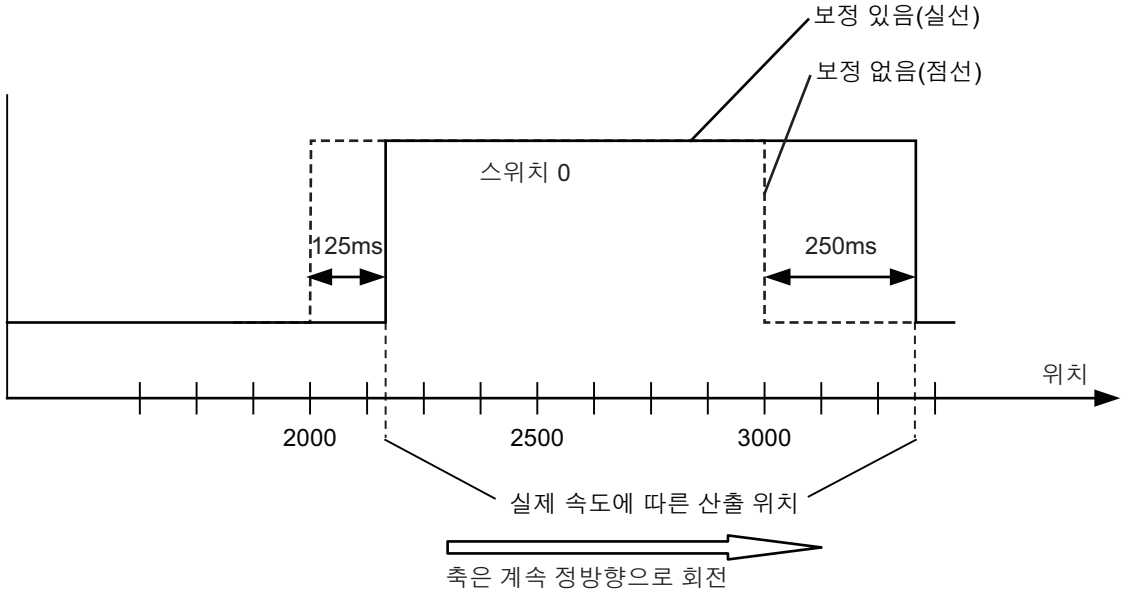
트랙 옵션 구조체(_sTRACK_REF 형식)는 스위치의 동작 조건을 지정하기 위한 구조체형 데이터 형식입니다. 이 명령에서는 배열 변수로 최대 16출력의 조건을 지정할 수 있습니다.

TrackOptions에 지정하는 변수의 요소 수와 Outputs에 지정하는 변수의 요소 수는 동일한 요소 수로 하십시오.

다음으로 트랙 옵션 구조체의 멤버에 대해 설명합니다.

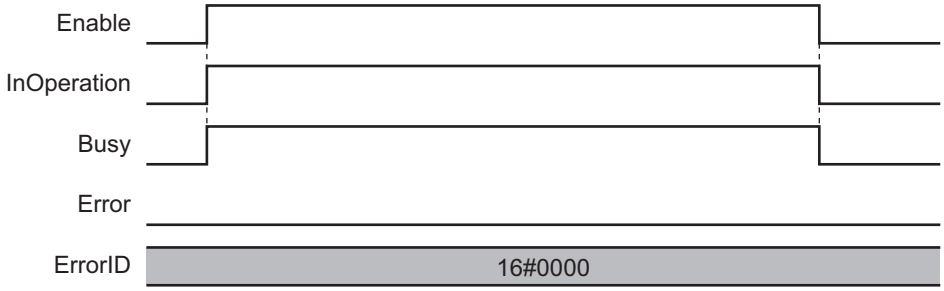
변수	명칭	데이터 형식	유효 범위	초기값	내용
OnCompensation	ON 시간 보정	TIME	T#-1s~T#1s	T#0s	출력이 ON되는 타이밍을 보정합니다. 양수의 경우는 타이밍을 느리게 하고, 음수의 경우는 빠르게 합니다.
OffCompensation	OFF 시간 보정	TIME	T#-1s~T#1s	T#0s	출력이 OFF되는 타이밍을 보정합니다. 양수의 경우는 타이밍을 느리게 하고, 음수의 경우는 빠르게 합니다.

앞에서 설명한 '설정 예(3-382페이지)'에서 다시 OnCompensation(ON 시간 보정)에 T#125ms를, OffCompensation(OFF 시간 보정)에 T#250ms를 설정했을 때의 동작은 다음과 같습니다.

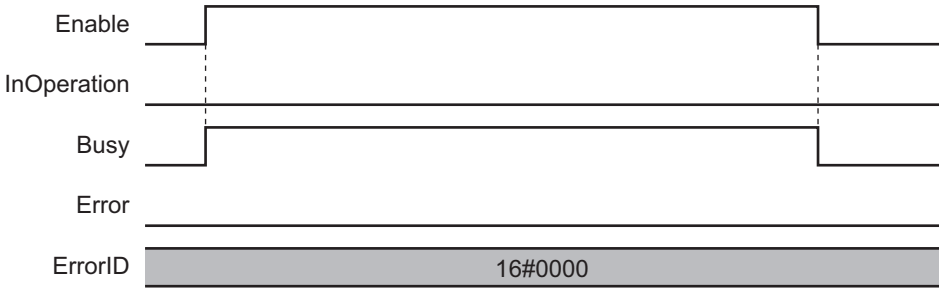


타이밍 차트

이 명령을 실행한 경우의 타이밍 차트는 다음과 같습니다.



사용하지 않는 축을 지정한 경우, 또는 MC 시운전 중인 경우의 타이밍 차트를 나타냅니다.



이때, Outputs의 EnableOut은 FALSE가 됩니다.

모션 명령 재기동

입력이 'Enable 형식'인 명령에는 모션 명령 재기동이 없습니다.

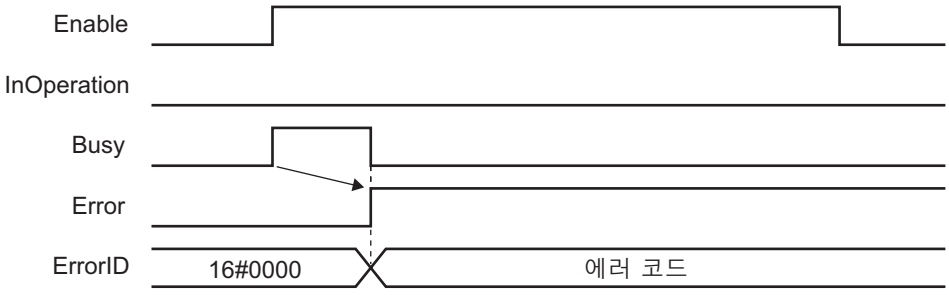
모션 명령 다중 기동

이 명령은 개별 명령 인스턴스마다 독립적으로 동작하므로 모션 명령 다중 기동의 제약이 없습니다.
모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

이상

이 명령을 기동시키지 못한 경우, 이 명령은 이상이 되고 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



이때, Outputs의 EnableOut은 FALSE가 됩니다.

● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기서는 앞에서 설명한 '설정 예(3-382페이지)'에 나온 예의 샘플 프로그램을 소개합니다.

사용 기기

이 샘플 프로그램에서 사용하는 주요 기기는 다음과 같습니다.

기기	형식
EtherCAT 커플러 유니트	NX-ECC201(Ver.1.1)*1
펄스 출력 유니트	NX-PG0122*2
인크리멘탈 엔코더 입력 유니트	NX-EC0122*3
디지털 출력 유니트	NX-OD2154*4

*1. 노드 주소를 '1', 장치명을 'E001'로 설정합니다.
*2. NX 유니트 번호를 '1', 장치명을 'N1'로 설정합니다. 축 1에 할당합니다.
*3. NX 유니트 번호를 '2', 장치명을 'N2'로 설정합니다. 축 2에 할당합니다.
*4. NX 유니트 번호를 '3', 장치명을 'N3'으로 설정합니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 최소한의 필요 설정은 다음과 같습니다.

● 축 파라메터

축 종류

축	축 종류
축 1	서보 축
축 2	엔코더 축

카운터 모드

축	카운터 모드
축 1	로터리 모드
축 2	로터리 모드

링 카운터

축	상한값	하한값
축 1	5000	0
축 2	5000	0

좌표 단위 선택

축	좌표 단위
축 1	pulse
축 2	pulse

래더도

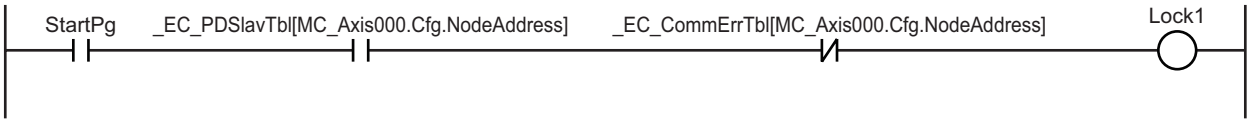
● 주요 변수

이름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
E001_Time_Stamp_of_Synchronous_Output	ULINT	–	디바이스 변수입니다.*1
N3_Output_Bit_00	BOOL	–	디바이스 변수입니다.
N3_Output_Bit_00_Time_Stamp	ULINT	–	디바이스 변수입니다.
N3_Output_Bit_01	BOOL	–	디바이스 변수입니다.
N3_Output_Bit_01_Time_Stamp	ULINT	–	디바이스 변수입니다.
Switches	ARRAY[0..3] OF _sCAMSWITCH_REF	–	입출력 변수 'Switches'의 입력 파라메터입니다. 요소 번호는 스위치 번호에 대응합니다.
Outputs	ARRAY[0..1] OF _sOUTPUT_REF	–	입출력 변수 'Outputs'의 입력 파라메터입니다. 요소 번호는 트랙 번호에 대응합니다.
TrackOptions	ARRAY[0..1] OF _sTRACK_REF	–	입출력 변수 'TrackOptions'의 입력 파라메터입니다. 요소 번호는 트랙 번호에 대응합니다.
EnableMask	WORD	16#0003	입력 변수 'EnableMask'의 입력 파라메터입니다. 트랙 0과 트랙 1이 유효합니다.

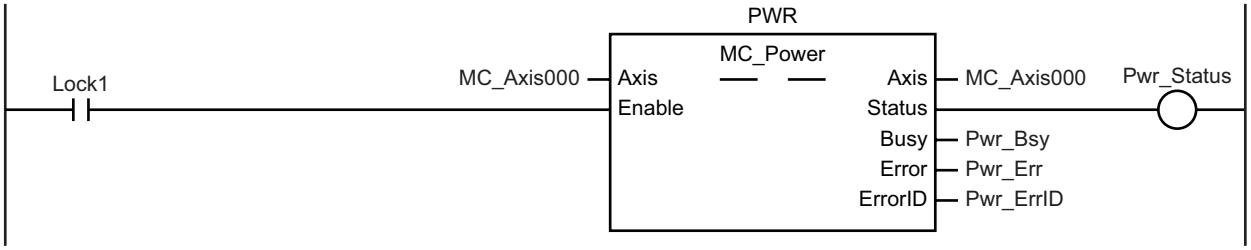
*1. EtherCAT 커플러 유니트의 I/O 항목에 0x200A:02(Time Stamp of Synchronous Output)를 추가해야 합니다.

● 샘플 프로그램

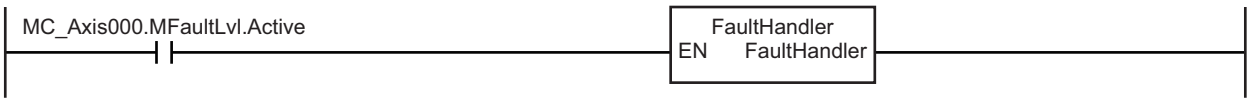
접점 StartPg가 TRUE가 되면 축 1의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 실시하고 있는지 확인한다.



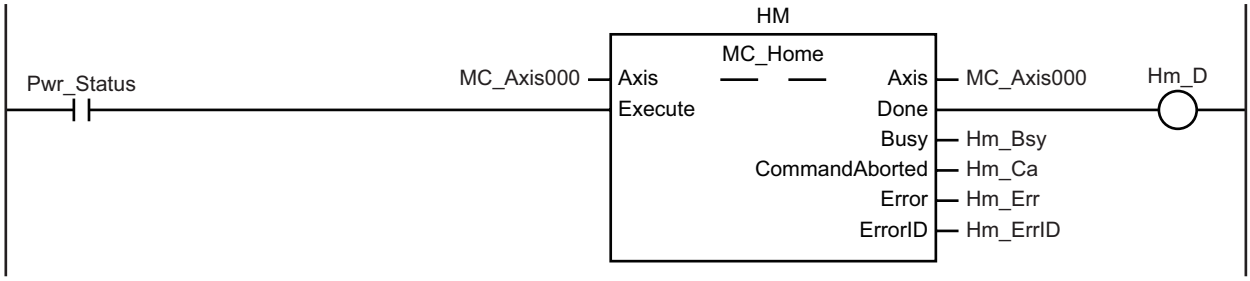
축 1이 프로세스 데이터 통신을 실시하고 있으면 서보 ON 상태가 된다.



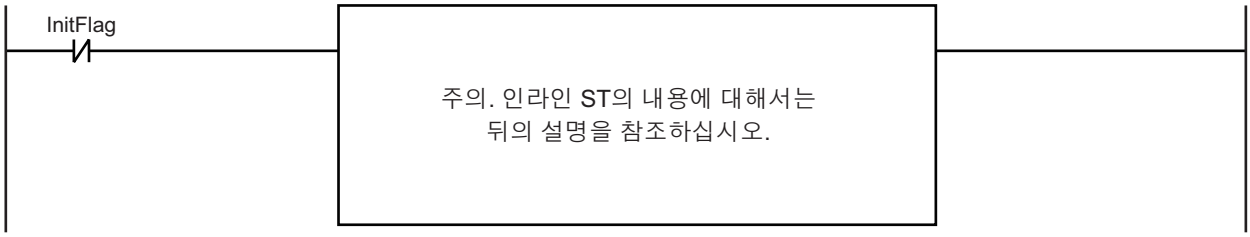
축 구성에서 가벼운 이상 레벨의 이상이 발생하는 경우, 장치에서 이상 발생 시의 처리(FaultHandler)를 실행한다.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰서 고객이 프로그래밍하십시오.



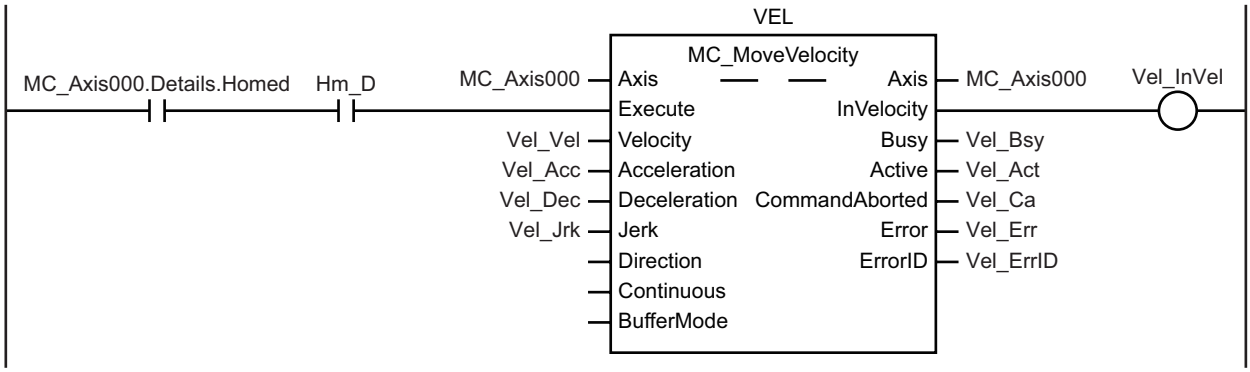
축 1이 서보 ON 상태가 되고 원점 미확정 상태이면 원점 복귀를 실행하여 원점을 확정한다.



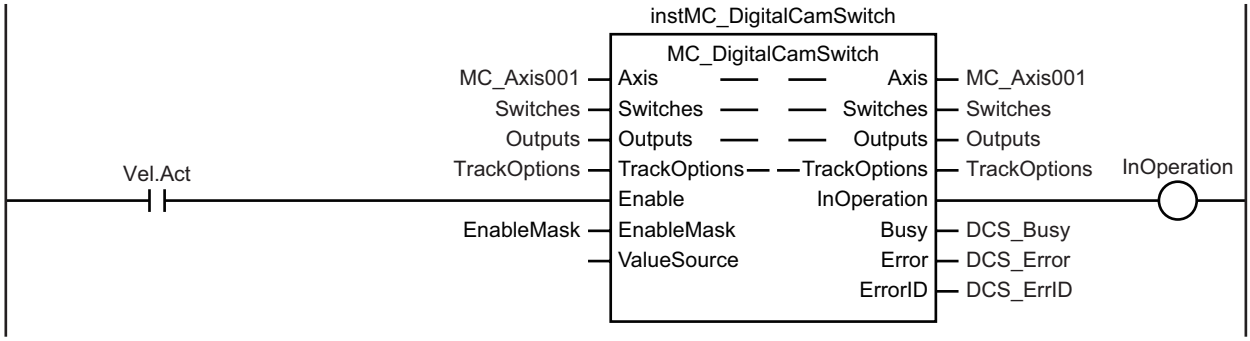
MC_MoveVelocity(속도 제어) 명령의 파라메터와 MC_DigitalCamSwitch(디지털 캠 스위치 유효) 명령의 Switches의 파라메터를 설정한다.



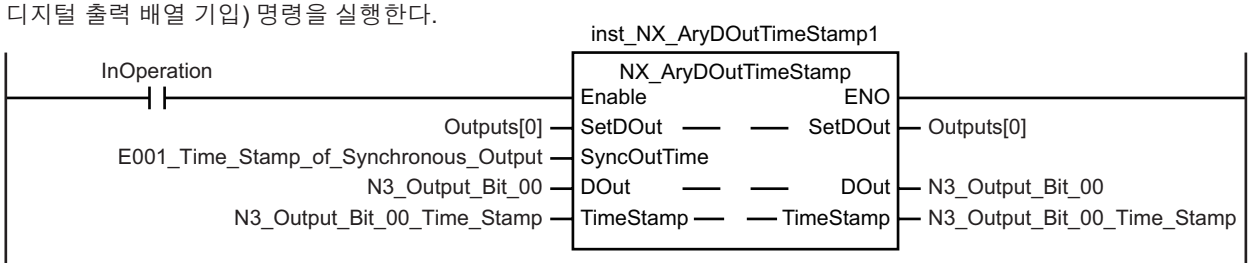
축 1이 원점 확정 상태가 되면 MC_MoveVelocity(속도 제어) 명령을 실행한다.

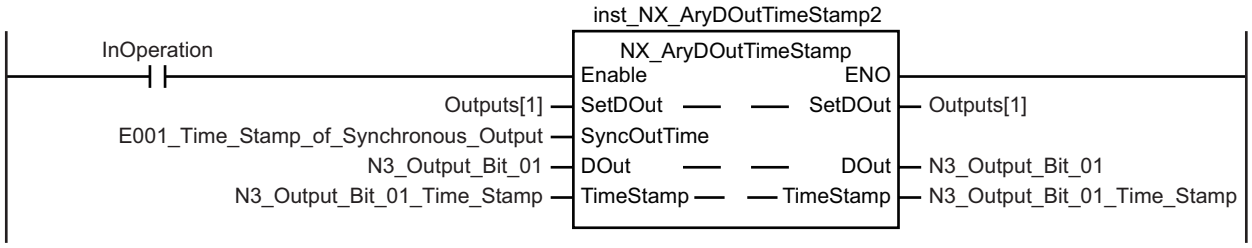


축 1의 MC_MoveVelocity(속도 제어) 명령이 실행되면 축 2의 MC_DigitalCamSwitch(디지털 캠 스위치 유효) 명령을 실행한다.



축 2의 MC_DigitalCamSwitch(디지털 캠 스위치 유효) 명령이 작동되면 NX_AryDOutTimeStamp(타임 스탬프 디지털 출력 배열 기입) 명령을 실행한다.





인라인 ST의 내용

```
//MC_MoveVelocity 파라메터
Vel_Vel         := LREAL#1000.0;
Vel_Acc         := LREAL#0.0;
Vel_Dec         := LREAL#0.0;
Vel_Jrk         := LREAL#1000.0;
InitFlag        := BOOL#TRUE;

//MC_DigitalCamSwitch 파라메터
Switches[0].TrackNumber         := UINT#0;
Switches[0].FirstOnPosition     := LREAL#2000.0;
Switches[0].LastOnPosition      := LREAL#3000.0;
Switches[0].AxisDirection       := _eMC_DIRECTION#_mcPositiveDirection;
Switches[0].CamSwitchMode       := _eMC_SWITCH_MODE#_mcPositionBased;

Switches[1].TrackNumber         := UINT#0;
Switches[1].FirstOnPosition     := LREAL#2500.0;
Switches[1].LastOnPosition      := LREAL#3000.0;
Switches[1].AxisDirection       := _eMC_DIRECTION#_mcNegativeDirection;
Switches[1].CamSwitchMode       := _eMC_SWITCH_MODE#_mcPositionBased;

Switches[2].TrackNumber         := UINT#0;
Switches[2].FirstOnPosition     := LREAL#4000.0;
Switches[2].LastOnPosition      := LREAL#1000.0;
Switches[2].AxisDirection       := _eMC_DIRECTION#_mcNoDirection;
Switches[2].CamSwitchMode       := _eMC_SWITCH_MODE#_mcPositionBased;

Switches[3].TrackNumber         := UINT#1;
Switches[3].FirstOnPosition     := LREAL#3000.0;
Switches[3].Duration            := T#1350ms;
Switches[3].AxisDirection       := _eMC_DIRECTION#_mcNoDirection;
Switches[3].CamSwitchMode       := _eMC_SWITCH_MODE#_mcTimeBased;
```

스트럭쳐 텍스트(ST)

● 주요 변수

이름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
E001_Time_Stamp_of_Synchronous_Output	ULINT	–	디바이스 변수입니다.*1
N3_Output_Bit_00	BOOL	–	디바이스 변수입니다.
N3_Output_Bit_00_Time_Stamp	ULINT	–	디바이스 변수입니다.
N3_Output_Bit_01	BOOL	–	디바이스 변수입니다.
N3_Output_Bit_01_Time_Stamp	ULINT	–	디바이스 변수입니다.
Pwr_En	BOOL	–	MC_Power의 인스턴스 PWR의 입력 변수 Enable에 할당된 변수입니다.
Switches	ARRAY[0..3] OF _sCAMSWITCH_REF	–	입출력 변수 'Switches'의 입력 파라메터입니다. 요소 번호는 스위치 번호에 대응합니다.
Outputs	ARRAY[0..1] OF _sOUTPUT_REF	–	입출력 변수 'Outputs'의 입력 파라메터입니다. 요소 번호는 트랙 번호에 대응합니다.
TrackOptions	ARRAY[0..1] OF _sTRACK_REF	–	입출력 변수 'TrackOptions'의 입력 파라메터입니다. 요소 번호는 트랙 번호에 대응합니다.
EnableMask	WORD	16#0003	입력 변수 'EnableMask'의 입력 파라메터입니다. 트랙 0과 트랙 1이 유효합니다.
Hm_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때, MC_Home의 인스턴스 HM을 기동합니다.
Vel_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때, MC_MoveVelocity의 인스턴스 VEL을 기동합니다.

*1. EtherCAT 커플러 유니트의 I/O 항목에 0x200A:02(Time Stamp of Synchronous Output)를 추가해야 합니다.

● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag=FALSE THEN

    //MC_MoveVelocity 파라메터
    Vel_Vel             := LREAL#1000.0;
    Vel_Acc             := LREAL#0.0;
    Vel_Dec             := LREAL#0.0;
    Vel_Jrk             := LREAL#1000.0;

    //MC_DigitalCamSwitch 파라메터
    Switches[0].TrackNumber                 := UINT#0;
    Switches[0].FirstOnPosition             := LREAL#2000.0;
    Switches[0].LastOnPosition              := LREAL#3000.0;
    Switches[0].AxisDirection               := _eMC_DIRECTION#_mcPositiveDirection;
    Switches[0].CamSwitchMode               := _eMC_SWITCH_MODE#_mcPositionBased;
    Switches[1].TrackNumber                 := UINT#0;
    Switches[1].FirstOnPosition             := LREAL#2500.0;
```

```
    Switches[1].LastOnPosition              := LREAL#3000.0;
    Switches[1].AxisDirection               := _eMC_DIRECTION#_mcNegativeDirection;
    Switches[1].CamSwitchMode               := _eMC_SWITCH_MODE#_mcPositionBased;
    Switches[2].TrackNumber                 := UINT#0;
    Switches[2].FirstOnPosition             := LREAL#4000.0;
    Switches[2].LastOnPosition              := LREAL#1000.0;
    Switches[2].AxisDirection               := _eMC_DIRECTION#_mcNoDirection;
    Switches[2].CamSwitchMode               := _eMC_SWITCH_MODE#_mcPositionBased;
    Switches[3].TrackNumber                 := UINT#1;
    Switches[3].FirstOnPosition             := LREAL#3000.0;
    Switches[3].Duration                    := T#1350ms;
    Switches[3].AxisDirection               := _eMC_DIRECTION#_mcNoDirection;
    Switches[3].CamSwitchMode               := _eMC_SWITCH_MODE#_mcTimeBased;


    //입력 파라메터를 설정한 후 InitFlag를 TRUE로 설정합니다.
    InitFlag:=TRUE;

END_IF;

//접점 StartPg가 TRUE인 경우, EtherCAT 통신이 정상적으로 통신되고 있는지 확인하고,
//축 1을 서보 ON 상태로 합니다.
//EtherCAT 통신이 정상적으로 수행되지 않는 경우는 서보 OFF로 설정합니다.
IF (StartPg=TRUE)
 AND (_EC_RegSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//축 1에서 가벼운 이상이 발생하면 이상 시 처리 FaultHandler를 실행합니다.
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰서 고객이 프로그래밍하십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

//서보 ON 상태이고 원점이 확정되지 않았으면 원점 복귀를 실행합니다.
IF (Pwr_Status=TRUE) THEN
    Hm_Ex:=TRUE;
END_IF;

//원점 확정 후 MC_MoveVelocity를 실행합니다.
IF MC_Axis000.Details.Homed=TRUE  AND Hm_D=TRUE THEN
    Vel_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_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_DigitalCamSwitch
instMC_DigitalCamSwitch(
    Axis            := MC_Axis001,
    Switches        := Switches,
    Outputs         := Outputs,
    TrackOptions    := TrackOptions,
    Enable          := Vel_Act,
    EnableMask      := EnableMask,
    InOperation     => InOperation,
    Busy            => DCS_Busy,
    Error           => DCS_Error,
    ErrorID         => DCS_ErrorID );

inst_NX_AryDOutTimeStamp1(
    Enable          := Vel_Act,
    SetDOut         := Outputs[0],
    SyncOutTime     := E001_Time_Stamp_of_Synchronous_Output,
    DOut            := N3_Output_Bit_00,
    TimeStamp       := N3_Output_Bit_00_Time_Stamp);

inst_NX_AryDOutTimeStamp2(
    Enable          := Vel_Act,
    SetDOut         := Outputs[1],
    SyncOutTime     := E001_Time_Stamp_of_Synchronous_Output,
    DOut            := N3_Output_Bit_01,
    TimeStamp       := N3_Output_Bit_01_Time_Stamp);
```

MC_TimeStampToPos

지정한 타임 스탬프의 축의 위치를 산출합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_TimeStampToPos	타임 스탬프→ 위치 산출	FB	MC_TimeStampToPos_instance MC_TimeStampToPos Axis — — Axis Enable — Enabled TimeStamp — Busy ValueSource — CalcPosition Error ErrorID	MC_TimeStampToPos_instance (Axis := 《파라메터》, Enable := 《파라메터》, TimeStamp := 《파라메터》, ValueSource := 《파라메터》, Enabled => 《파라메터》, Busy => 《파라메터》, CalcPosition => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

사용상의 주의 사항

이 명령은 NX 시리즈 위치 인터페이스 유니트를 할당한 축에 대해 유효합니다.
사용할 수 있는 NX 유니트는 NX-EC0□□□, NX-ECS□□□입니다.

버전 관련 정보

이 명령은 CPU 유니트 Ver.1.06 이상과 Sysmac Studio Ver.1.07 이상의 조합에서 사용할 수 있습니다.

변수

입력 변수

입력 변수	이름	데이터 형식	유효 범위	초기값	내용
Enable	유효	BOOL	TRUE, FALSE	FALSE	TRUE인 동안 명령을 실행합니다.
TimeStamp	타임 스탬프	ULINT	양수 또는 '0'	0	위치를 산출하는 타임 스탬프를 지정합니다. 타임 스탬프 방식을 지원하는 디지털 입력 유니트나 엔코더 입력 유니트 내의 시각을 기준으로 하는 타임 스탬프를 지정합니다. 단위는 [ns]입니다.
ValueSource (Reserved)	입력 정보	_sMC_SOURCE	–	–	(Reserved)

출력 변수

출력 변수	이름	데이터 형식	유효 범위	내용
Enabled	유효	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CalcPosition	산출 위치	LREAL	음수, 양수, '0'	지정한 타임 스탬프의 위치를 출력합니다. 단위는 [명령 단위]입니다.*1
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*2	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 'A-1 에러 코드 목록(A-2페이지)'를 참조하십시오.

● 출력 변수의 반영 타이밍

변수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Enabled	Enable이 TRUE가 되었을 때	• Enable이 FALSE가 되었을 때 • Error가 TRUE가 되었을 때
Busy	Enable의 기동	• Enable이 FALSE가 되었을 때 • Error가 TRUE가 되었을 때
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

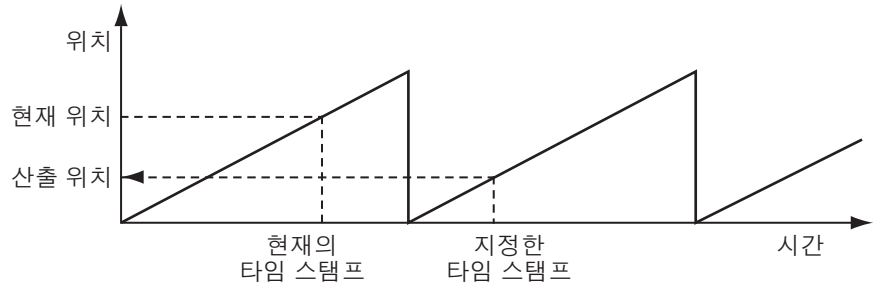
입출력 변수

입출력 변수	이름	데이터 형식	유효 범위	내용
Axis	축	_sAXIS_REF	–	축을 지정합니다. *1 *2

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 시스템 정의 변수의 축 변수명(_MC_AX[**])을 사용하십시오.
*2. NX 시리즈 엔코더 입력 유니트를 할당한 엔코더 축을 지정하십시오.

기능 설명

- 이 명령은 축의 피드백 현재 위치와 타임 스탬프를 바탕으로, 입력 변수로 지정한 타임 스탬프의 피드백 현재 위치를 산출합니다.
- 축에는 NX 시리즈 엔코더 입력 유니트를 지정하십시오.
- 축 파라메터의 카운트 모드가 리니어 모드인 경우, 산출한 위치가 오버 플로우 또는 언더 플로우가 되었을 때, 산출값은 오버 플로우 또는 언더 플로우의 값이 됩니다. 이 경우는 이상이 아닙니다.

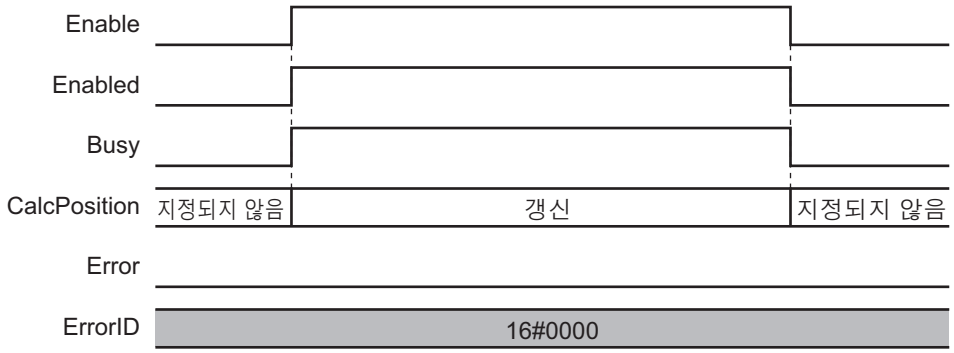


사용상의 주의 사항

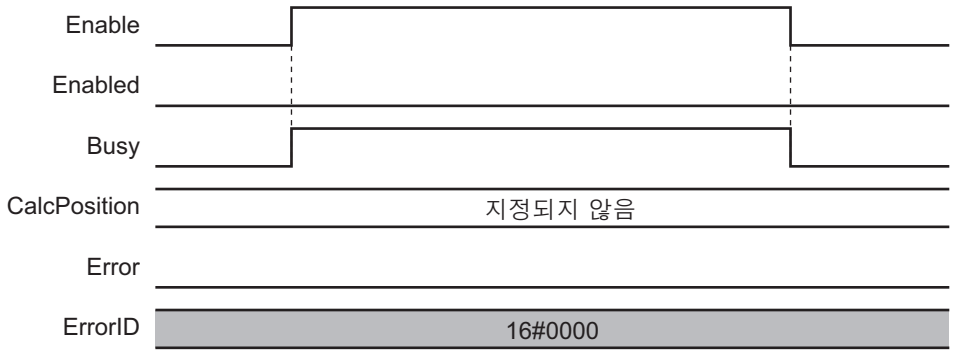
- 이 명령은 NX 시리즈 엔코더 입력 유니트를 사용하고, 변화 시각 취득 기능이 동작해야 합니다. 다음과 같은 경우, 변화 시각 취득 기능은 동작하지 않습니다.
  - 변화 시각 취득 기능에 대응하지 않는 엔코더 입력 또는 서보 드라이버를 사용하고 있는 경우
  - NX 시리즈 엔코더 입력 유니트의 개체 6010Hex(타임 스탬프) 를 PDO 에 할당하지 않은 경우
  - EtherCAT 커플러 유니트의 [DC 유효] 를 [무효] 로 설정한 경우
- 타임 스탬프가 갱신되지 않은 경우에도 이 명령은 이상이 아닙니다. 이 때도 위치가 산출되지만 지정된 타임 스탬프에서의 위치가 되지는 않습니다.
  Sysmac Studio의 MC 모니터 테이블 또는 감시 창에서 축 변수의 멤버 'TimeStamp(타임 스탬프)'가 갱신되었는지 확인한 후 이 명령을 사용하십시오.
- 이 명령은 축의 현재 위치와 현재 속도를 바탕으로 지정된 타임 스탬프에서의 위치를 산출합니다. 축이 급격하게 가감속할 때는 산출 오차가 커질 수 있으므로 축이 일정 속도인 상태에서 사용하십시오.
  동작 검증을 충분히 실시하여 안전을 확인하십시오.
- 사용하지 않는 축을 지정한 경우 또는 MC 시운전 중인 경우, Enable이 TRUE일 때 Busy가 TRUE가 되고, Enabled와 Error는 FALSE가 됩니다.
- 이름이 같은 인스턴스를 2개 이상 배치하지 마십시오. 배치하면 의도하지 않은 출력이 되는 경우가 있습니다.

타이밍 차트

이 명령을 실행한 경우의 타이밍 차트는 다음과 같습니다.



사용하지 않는 축을 지정한 경우, 또는 MC 시운전 중인 경우의 타이밍 차트를 나타냅니다.



모션 명령 재기동

입력이 'Enable 형식'인 명령에는 모션 명령 재기동이 없습니다.

모션 명령 다중 기동

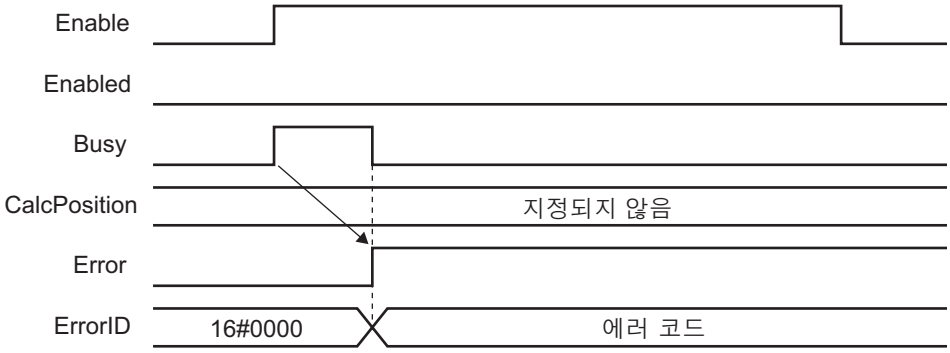
이 명령은 개별 명령마다 독립적으로 동작하므로 모션 명령 다중 기동의 제약이 없습니다.

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

이상

이 명령을 기동시키지 못한 경우, 이 명령은 이상이 되고 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기서는 벨트 컨베이어 위를 이동하는 워크간의 거리를 계측하는 샘플 프로그램을 설명합니다.

사용 기기

이 샘플 프로그램에서 사용하는 주요 기기는 다음과 같습니다.

기기	형식
EtherCAT 커플러 유니트	NX-ECC201(Ver.1.1)*1
펄스 출력 유니트	NX-PG0122*2
인크리멘탈 엔코더 입력 유니트	NX-EC0122*3
디지털 입력 유니트	NX-ID3344*4

*1. 노드 주소를 '1', 장치명을 'E001'로 설정합니다.
*2. NX 유니트 번호를 '1', 장치명을 'N1'로 설정합니다. 축 1에 할당합니다.
*3. NX 유니트 번호를 '2', 장치명을 'N2'로 설정합니다. 축 2에 할당합니다.
*4. NX 유니트 번호를 '3', 장치명을 'N3'으로 설정합니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 최소한의 필요 설정은 다음과 같습니다.

● 축 파라메터

축 종류

축	축 종류
축 1	서보 축
축 2	엔코더 축

카운터 모드

축	카운터 모드
축 1	로터리 모드
축 2	로터리 모드

링 카운터

축	상한값	하한값
축 1	360	0
축 2	360	0

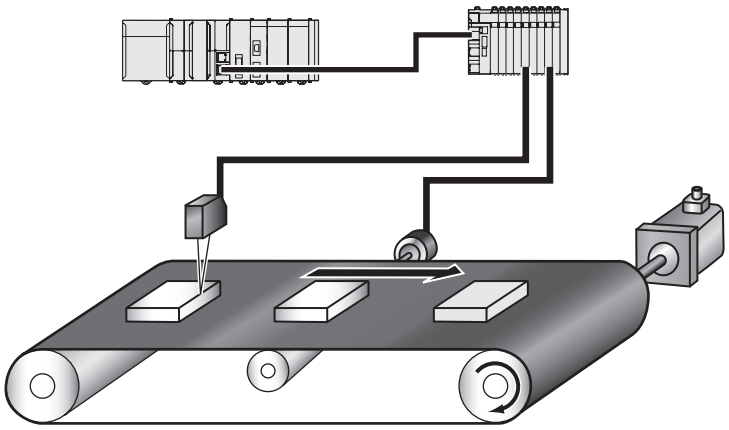
좌표 단위 선택

축	좌표 단위
축 1	degree
축 2	degree

동작 예

센서가 워크를 검출하면 센서 출력은 ON이 되고, 워크가 통과하면 센서 출력은 OFF가 됩니다. 센서가 다음 워크를 검출하면 센서 출력은 다시 ON이 됩니다.

센서 출력이 ON되었을 때의 타임 스탬프를 바탕으로 엔코더 입력의 위치를 계산합니다. 두 위치의 차가 워크간의 거리가 됩니다.



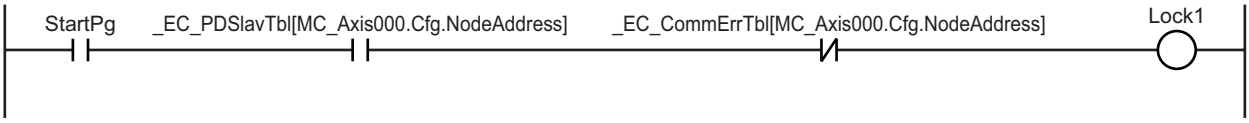
래더도

● 주요 변수

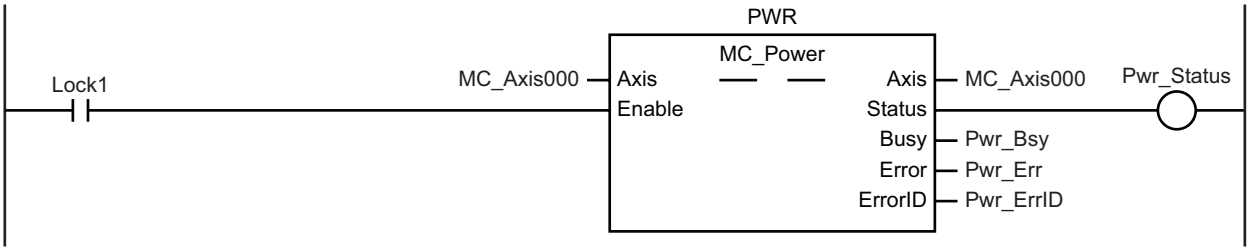
이름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
N3_Input_Bit_00	BOOL	–	디바이스 변수입니다.
N3_Input_Bit_00_Time_Stamp	ULINT	–	디바이스 변수입니다.
Position	ARRAY[0..1] OF LREAL	–	산출 위치를 저장합니다.
Count	ARRAY[0..1] OF ULINT	–	다회전수를 저장합니다.
FirstPoint	UINT	–	처리용 변수입니다.
LastPoint	UINT	–	처리용 변수입니다.
Distance	LREAL	–	워크간의 거리입니다.

● 샘플 프로그램

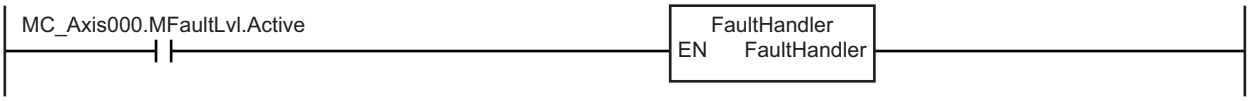
접점 StartPg가 TRUE가 되면 축 1의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 실시하고 있는지 확인한다.



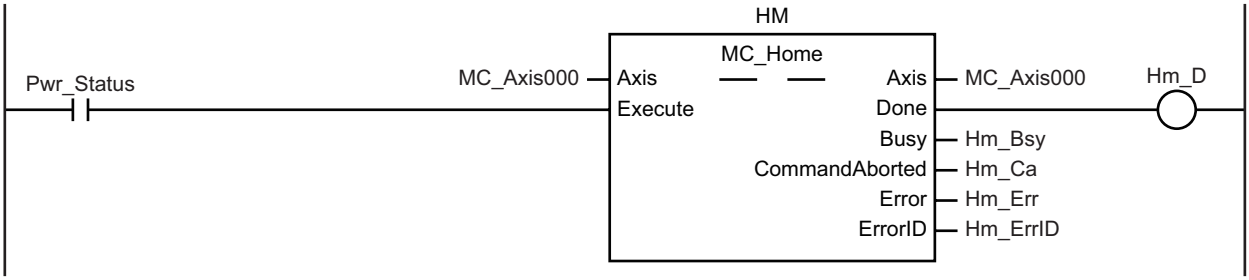
축 1이 프로세스 데이터 통신을 실시하고 있으면 서보 ON 상태가 된다.



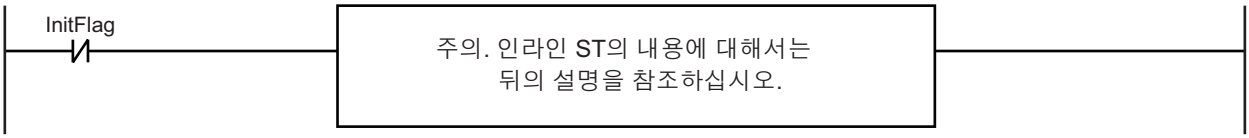
축 구성에서 가벼운 이상 레벨의 이상이 발생하는 경우, 장치에서 이상 발생 시의 처리(FaultHandler)를 실행한다. 이상 발생 시의 처리(FaultHandler)는 장치에 맞춰서 고객이 프로그래밍하십시오.



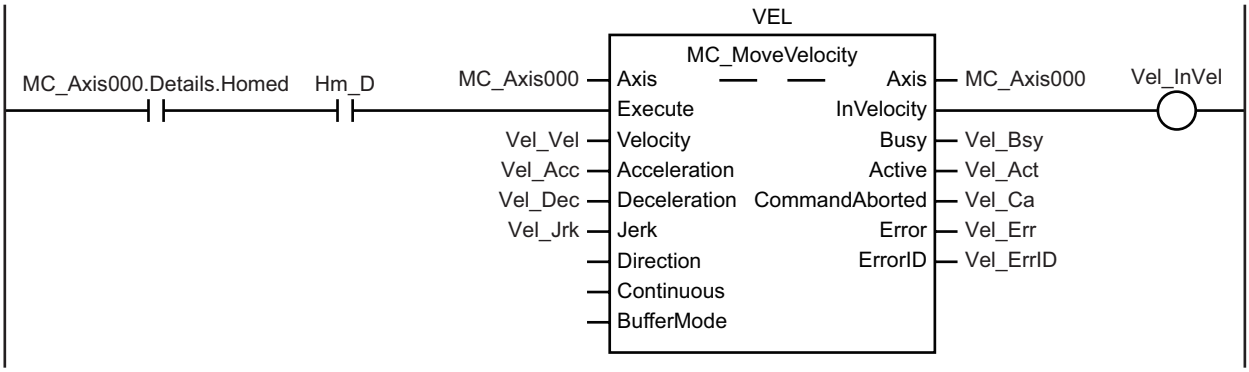
축 1이 서보 ON 상태가 되고 원점 미확정 상태이면 원점 복귀를 실행하여 원점을 확정한다.



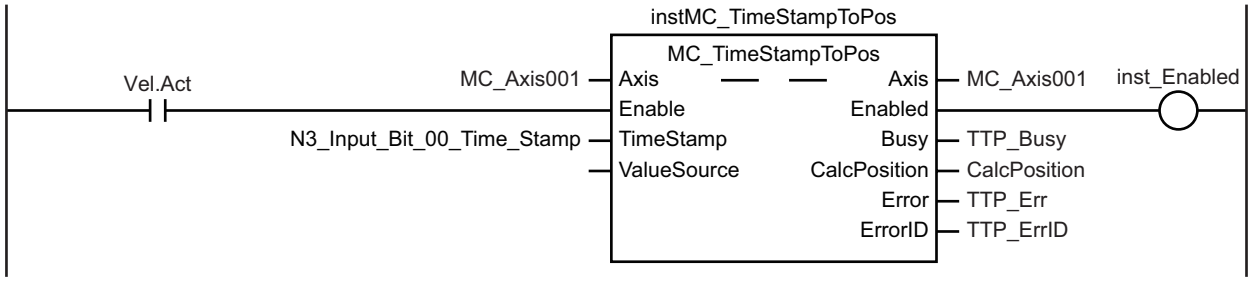
MC_MoveVelocity(속도 제어) 명령의 파라메터를 설정한다.



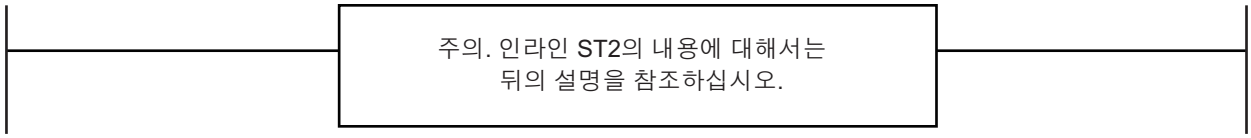
축 1이 원점 확정 상태가 되면 MC_MoveVelocity(속도 제어) 명령을 실행한다.



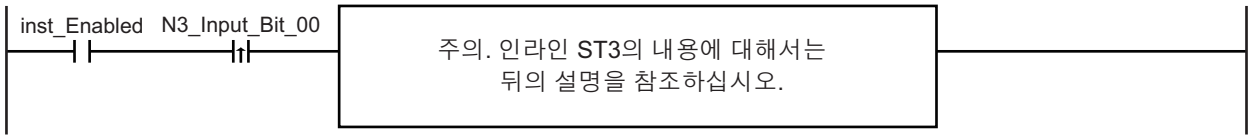
축 1의 MC_MoveVelocity(속도 제어) 명령이 실행되면 축 2의 MC_TimeStampToPos(타임 스탬프→축 위치 산출) 명령을 실행한다.



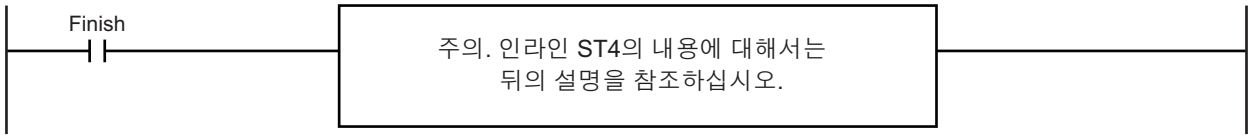
엔코더 축의 다회전수를 카운트한다(현재 값 < 전회 값일 때, 링 카운터의 상한값을 초과했다고 판단).



센서 출력의 ON 기동 시 CalcPosition(산출 위치)을 취득한다(Position[0]과 Position[1]에 교대로 저장)



2 번째 워크의 검출 이후 워크간의 거리를 산출한다.



인라인 ST 1의 내용

```
//MC_MoveVelocity 파라메터
Vel_Vel         := LREAL#1000.0;
Vel_Acc         := LREAL#0.0;
Vel_Dec         := LREAL#0.0;
Vel_Jrk         := LREAL#1000.0;
InitFlag        := BOOL#TRUE;
```

인라인 ST 2의 내용

```
IF MC_Axis001.Act.Pos < PreAxis001ActPos THEN
    Inc(RotaryCount);
END_IF;
PreAxis001ActPos := MC_Axis001.Act.Pos;
```

인라인 ST 3의 내용

```
IF Index < UINT#2 THEN
    Position[Index] := CalcPosition;
    Count[Index] := RotaryCount;
    Index := Index + UINT#1;
END_IF;
IF Index >= UINT#2 THEN
    Finish := BOOL#TRUE;
    Index := UINT#0;
END_IF;
```

인라인 ST 4의 내용

```
//1회째 비교: 워크 1=Position[0], 워크 2=Position[1]
//2회째 비교: 워크 2=Position[1], 워크 3=Position[0]
//3회째 비교: 워크 3=Position[0], 워크 4=Position[1]
//…
FirstPoint := (Index+UINT#1) MOD UINT#2;
LastPoint :=Index;
DiffCount := Count[1] - Count[0];
Distance := (ABS( DiffCount) -LINT#1)* 360.0 +
                                        (360.0 + Position[FirstPoint] - Position[LastPoint]);
```

스트럭쳐 텍스트(ST)

● 주요 변수

이름	데이터 형식	초기값	코멘트
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
N3_Input_Bit_00	BOOL	–	디바이스 변수입니다.
N3_Input_Bit_00_Time_Stamp	ULINT	–	디바이스 변수입니다.
Hm_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때, MC_Home의 인스턴스 HM을 기동합니다.
Vel_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때, MC_MoveVelocity의 인스턴스 VEL을 기동합니다.
PreN3InputBit00	BOOL	–	디바이스 변수 N3_Input_Bit_00의 전회 값입니다.
Position	ARRAY[0..1] OF LREAL	–	산출 위치를 저장합니다.
Count	ARRAY[0..1] OF ULINT	–	다회전수를 저장합니다.
FirstPoint	UINT	–	처리용 변수입니다.
LastPoint	UINT	–	처리용 변수입니다.
Distance	LREAL	–	워크간의 거리입니다.

● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag=FALSE THEN

    //MC_MoveVelocity 파라메터
    Vel_Vel     := LREAL#1000.0;
    Vel_Acc     := LREAL#0.0;
    Vel_Dec     := LREAL#0.0;
    Vel_Jrk     := LREAL#1000.0;
    //입력 파라메터를 설정한 후 InitFlag를 TRUE로 설정합니다.
    InitFlag:=TRUE;

END_IF;

//접점 StartPg가 TRUE인 경우, EtherCAT 통신이 정상적으로 통신되고 있는지 확인하고,
//축 1을 서보 ON 상태로 합니다.
//EtherCAT 통신이 정상적으로 수행되지 않는 경우는 서보 OFF로 설정합니다.
IF (StartPg=TRUE)
 AND (_EC_RegSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//축 1에서 가벼운 이상이 발생하면 이상 시 처리 FaultHandler를 실행합니다.
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰서 고객이 프로그래밍하십시오.
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

//서보 ON 상태이고 원점이 확정되지 않았으면 원점 복귀를 실행합니다.
IF (Pwr_Status=TRUE)  THEN
    Hm_Ex:=TRUE;
END_IF;
```

```
//원점 확정 후 MC_MoveVelocity를 실행합니다.
IF MC_Axis000.Details.Homed=TRUE  AND Hm_D=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

//엔코더 축의 다회전수를 카운트합니다.
//(현재 값<전회 값일 때, 링 카운터의 상한값을 초과했다고 판단)
IF MC_Axis001.Act.Pos<PreAxis001ActPos THEN
    Inc(RotaryCount);
END_IF;
PreAxis001ActPos := MC_Axis001.Act.Pos;


//MC_TimeStampToPos
instMC_TimeStampToPos(
    Axis                := MC_Axis001,
    Enable              := Vel_Ex,
    TimeStamp           := TimeStamp,
    Enabled             => inst_Enabled,
    Busy                => TSTP_Busy,
    CalcPosition        => CalcPosition,
    Error               => TSTP_Error,
    ErrorID             => TSTP_ErrorID );

//센서 출력의 ON 기동 시 CalcPosition(산출 위치)을 취득합니다.
//(Position[0]과 Position[1]에 교대로 저장)
IF inst_Enabled THEN
    IF PreN3InputBit00=FALSE AND N3_Input_Bit_00=TRUE THEN
        IF Index < UINT#2 THEN
                Position[Index] := CalcPosition;
                Count[Index] := RotaryCount;
                Index := Index + UINT#1;
        END_IF;
        IF Index >= UINT#2 THEN
                Finish := BOOL#TRUE;
                Index := UINT#0;
        END_IF;
    END_IF;
END_IF;
PreN3InputBit00 := N3_Input_Bit_00;

//ÇQ번째 워크의 검출 이후 워크간의 거리를 산출합니다.
//1회째 비교: 워크 1=Position[0], 워크 2=Position[1]
//2회째 비교: 워크 2=Position[1], 워크 3=Position[0]
//3회째 비교: 워크 3=Position[0], 워크 4=Position[1]
//Àc
IF Finish THEN
    FirstPoint := (Index+UINT#1) MOD UINT#2;
    LastPoint :=Index;
    DiffCount := Count[1] - Count[0];
    Distance := (ABS( DiffCount) -LINT#1)* 360.0 +
                                (360.0 + Position[FirstPoint] - Position[LastPoint]);
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_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
);
```

4

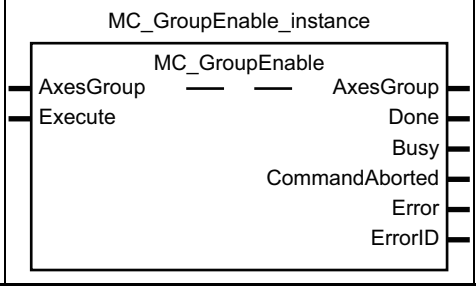
축 그룹 명령

MC 기능 모듈이 다축 협조 제어를 실행하기 위한 명령에 대해 설명합니다.

MC_GroupEnable **4-2**
MC_GroupDisable **4-6**
MC_MoveLinear **4-11**
MC_MoveLinearAbsolute **4-37**
MC_MoveLinearRelative **4-40**
MC_MoveCircular2D **4-43**
MC_GroupStop **4-67**
MC_GroupImmediateStop **4-75**
MC_GroupSetOverride **4-79**
MC_GroupReadPosition **4-83**
MC_ChangeAxesInGroup **4-87**
MC_GroupSyncMoveAbsolute **4-91**
MC_GroupReset **4-97**

MC_GroupEnable

축 그룹을 유효로 합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_GroupEnable	축 그룹 유효	FB		MC_GroupEnable_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	축 그룹 이상 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

이 명령의 기능을 설명하기 전에 축 그룹의 상태에 대해 설명합니다.

• 축 그룹에는 GroupEnable(그룹 유효)과 GroupDisable(그룹 무효)의 두 가지 상태가 있습니다.

상 태	설 명
GroupEnable (그룹 유효)	다축 협조 제어가 유효한 상태입니다. MC_ChangeAxesInGroup(축 그룹 구성축 쓰기) 명령을 제외한 모든 다축 협조 명령을 실행할 수 있습니다.
GroupDisable (그룹 무효)	다축 협조 제어가 무효인 상태입니다. 다축 협조 제어 명령은 다음 명령만 실행할 수 있습니다. • MC_GroupEnable(축 그룹 유효) 명령 • MC_GroupDisable(축 그룹 무효) 명령 • MC_GroupReset(축 그룹 에러 리셋) 명령 • MC_GroupSetOverride(축 그룹 오버라이드값 설정) 명령 • MC_GroupReadPosition(축 그룹 위치 취득) 명령 • MC_ChangeAxesInGroup(축 그룹 구성축 쓰기) 명령

• 다축 협조 제어를 실행하려면 축 그룹의 상태가 GroupEnable(그룹 유효)이어야 합니다.

모션 제어 시스템 변수의 축 그룹 변수를 모니터링하면 축 그룹이 유효인지 무효인지를 확인할 수 있습니다.

변수명	이 름	설 명
_MC_GRP**.Status.Ready	축 그룹 유효	축 그룹이 정지 중이고 축 그룹 명령에 의한 기동 준비가 완료되었을 때 TRUE가 됩니다.
_MC_GRP**.Status.Disabled	축 그룹 무효	축 그룹 무효로 정지 중일 때 TRUE가 됩니다.

(주의) '_MC_GRP**.'의 '**'에는 0～31의 숫자가 들어갑니다.

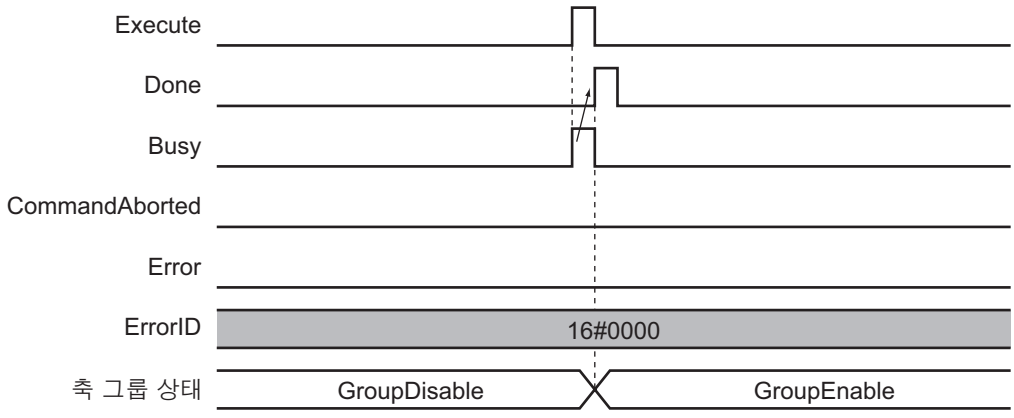
기본 기능

- MC_GroupEnable(축 그룹 유효) 명령은 AxesGroup(축 그룹)에서 지정한 축 그룹의 상태를 GroupEnable(그룹 유효)로 합니다.
- GroupEnable(그룹 유효) 상태로 하면 축 그룹은 모든 다축 협조 명령을 실행할 수 있게 됩니다.
- 축 그룹에 지정할 수 있는 축 종류는 '서보 축', '가상 서보 축'뿐이며, 다른 축 종류를 포함하면 이상이 됩니다.
- 축 그룹에 속하는 모든 축이 정지 중이어야 합니다.
  축이 정지 중인지의 여부는 축 변수의 Status.Disabled(축 무효), 또는 Status.Standstill(정지 중)이 TRUE이면 정지 중임을 알 수 있습니다.
- 이미 다른 축 그룹에 속해 있고 축 그룹 유효로 간주되는 축이 있으면 MC_GroupEnable(축 그룹 유효) 명령이 실행되지 않고 이상이 됩니다.
- 축 그룹이 유효가 되면 축 그룹에 속한 축의 상태는 '다축 협조 동작 중'이 됩니다.
  축 변수의 Status.Coordinated(다축 협조 동작 중)가 TRUE가 됩니다.
- 축 그룹이 무효가 되는 조건은 MC_GroupDisable(축 그룹 무효) 명령의 실행과 프로그램 모드 이행에 의한 운전 정지 및 MC 시운전 시작 시입니다.

사용상의 주의 사항

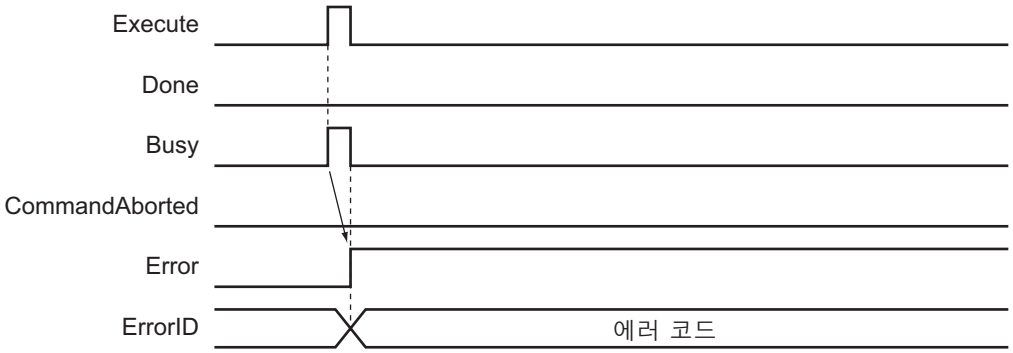
- 축 그룹을 사용하려면 Sysmac Studio에서 축 그룹을 작성한 후 CPU 유니트에 설정을 다운로드해 두어야 합니다. 프로그램 안에서 축 그룹의 축을 변경할 수는 없습니다.
  CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서는 MC_ChangeAxesInGroup(축 그룹 구성축 쓰기) 명령을 사용하여 일시적으로 변경할 수 있습니다.
- '다운로드'는 Sysmac Studio의 '동기' 기능을 사용합니다.

타이밍 차트



이상

MC_GroupEnable(축 그룹 유효) 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.



참고

축 그룹에서 이상이 발생해도 그룹에 포함된 각 축은 이상이 아닙니다.

● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_GroupDisable

축 그룹을 무효로 합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_GroupDisable	축 그룹 무효	FB	MC_GroupDisable_instance MC_GroupDisable AxesGroup — — AxesGroup Execute — Done Busy CommandAborted Error ErrorID	MC_GroupDisable_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	축 그룹 상태가 GroupDisable로 변경되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	TRUE가 되는 조건은 없습니다(Reserved).	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

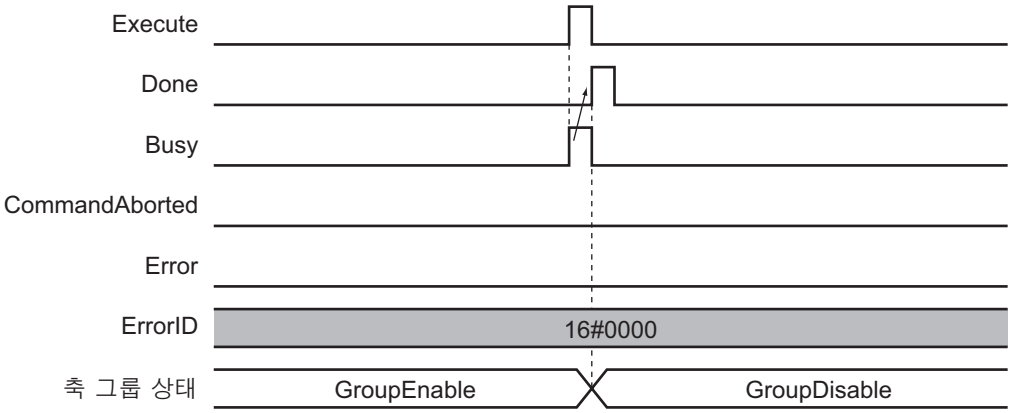
입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

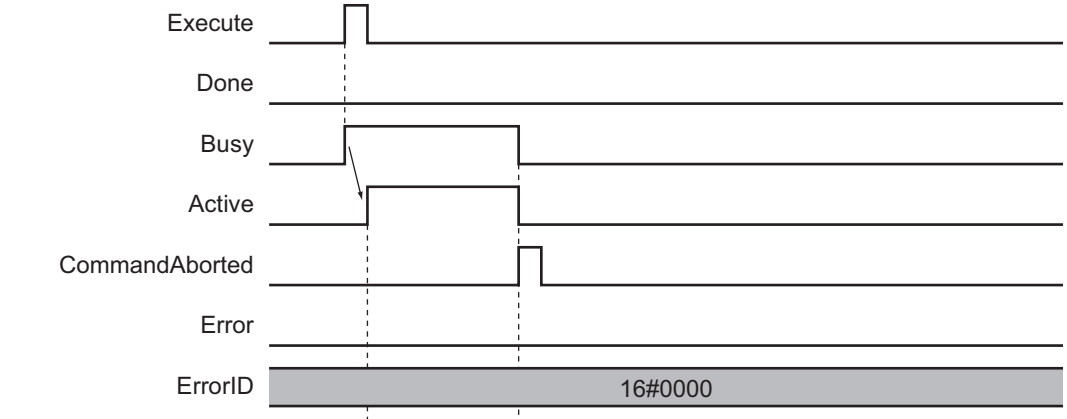
- 이 명령은 축 그룹 상태를 무효로 합니다.
  축 그룹의 상태 설명에 대해서는, '기능 설명(4-3페이지)'을 참조하십시오.
- MC_ GroupDisable(축 그룹 무효) 명령은 AxesGroup(축 그룹)에서 지정한 축 그룹의 상태를 GroupDisable(그룹 무효)로 합니다.
  GroupDisable(그룹 무효) 상태로 하면 축 그룹은 다축 협조 명령을 수신하지 않게 됩니다.
- 축 그룹 상태를 GroupDisable(그룹 무효)로 한 경우 지정한 AxesGroup(축 그룹)의 버퍼링 명령은 클리어됩니다.
- 축 그룹은 운전 정지 시에도 무효가 됩니다.
  축 그룹이 무효가 되면 축 그룹에 속한 축의 상태는 Status.Coordinated(다축 협조 동작 중)가 TRUE가 된 상태에서 각 축의 상태로 변화합니다.
  각 축의 상태를 알려면 축 변수의 Status(축 Status)를 확인하십시오.

타이밍 차트

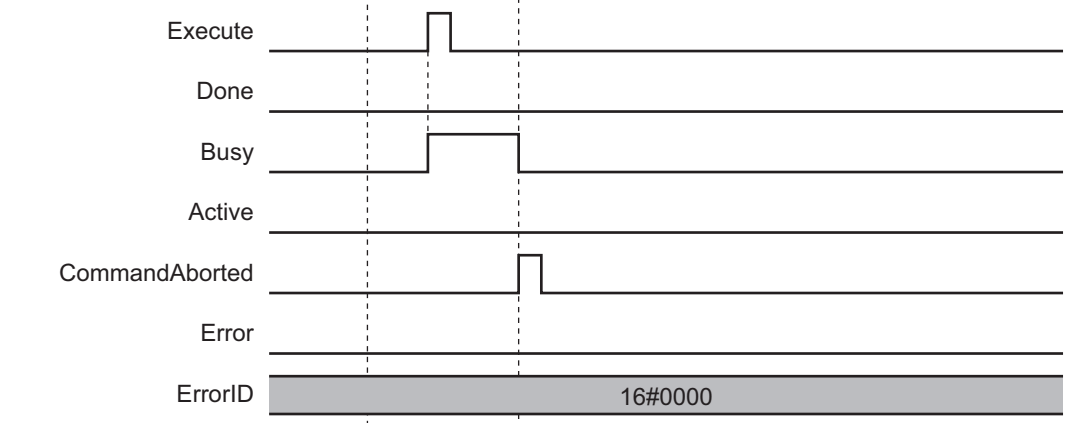


다축 협조 제어 중인 AxesGroup(축 그룹)에 MC_GroupDisable(축 그룹 무효) 명령을 실행한 경우 다축 협조 제어는 CommandAborted(실행 중단)됩니다.
이 경우 버퍼링되고 있는 다축 협조 제어 명령도 동일하게 CommandAborted(실행 중단)됩니다.
동작 중인 축은 각 축의 최대 감속도로 감속 정지합니다.

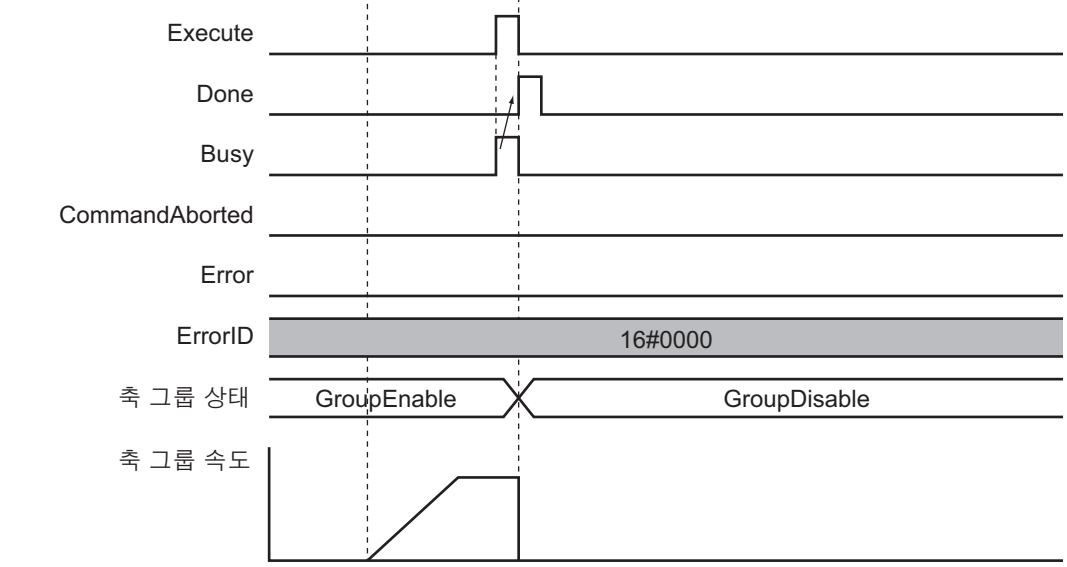
다축 협조 제어 1(Ex.MC_MoveLinearAbsolute)



다축 협조 제어 2(Ex.MC_MoveLinearAbsolute.Bufferd)

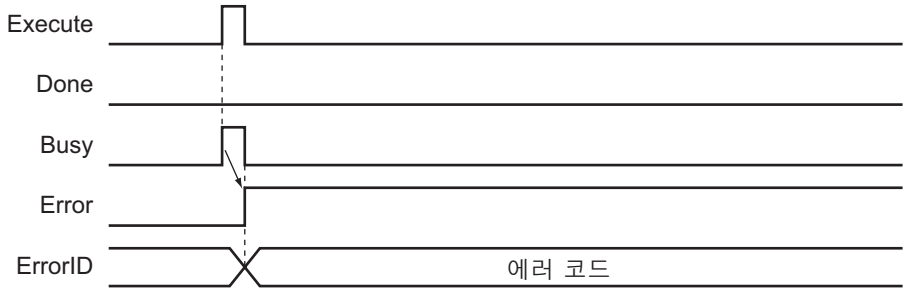


MC_GroupDisable



이상

MC_GroupDisable(축 그룹 무효) 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는, 🕮 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_MoveLinear

직선 보간을 실행합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
MC_MoveLinear	직선 보간	FB	MC_MoveLinear_instance MC_MoveLinear AxesGroup — AxesGroup Execute — Done Position — Busy Velocity — Active Acceleration — CommandAborted Deceleration — Error Jerk — ErrorID CoordSystem BufferMode TransitionMode MoveMode	MC_MoveLinear_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, Position := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, CoordSystem := 《파라메터》, BufferMode := 《파라메터》, TransitionMode := 《파라메터》, MoveMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Position	목표 위치	ARRAY [0..3] OF LREAL	음수, 양수, '0'	0	직선 보간의 목표 위치를 지정합니다. 단위는 [명령 단위]입니다. *1
Velocity*2	목표 속도	LREAL	양수	0	목표 속도를 지정합니다. 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1
CoordSystem	좌표 계통	_eMC_COORD_SYSTEM	0: _mcACS	0 *3	좌표 계통을 지정합니다. 0 : 축 좌표 계통(ACS)

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered 2 : BlendingLow 3 : BlendingPrevious 4 : BlendingNext 5 : BlendingHigh
Transition Mode	트랜지션 모드 (이동 모드)	_eMC_ TRANSITION _MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	동작 경로를 지정합니다. 0 : 트랜지션 무효 10 : ConnerSuper impose
MoveMode	이동 방법 선택	_eMC_MOVE _MODE	0: _mcAbsolute 1: _mcRelative	0 *3	이동 방법을 선택합니다. 0 : 절대값 위치 결정 1 : 상대값 위치 결정

*1. 명령 단위에 대해서는, 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 목표 속도를 반드시 설정하십시오. 설정하지 않고 동작하게 하면 이상이 발생합니다.
*3. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 🕮 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	위치 결정이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	축이 이동했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_GroupStop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- 2축~4축의 직선 보간을 실행합니다.

사용상의 주의 사항

- 축 그룹의 구성 축 중 하나가 원점 미확정 상태이면 '원점 미확정 상태에서의 명령 기동 이상 (에러 코드: 5466 Hex)'이 됩니다.
- 직선 보간을 실행할 경우 해당 축 그룹에 속한 논리축 중 하나의 한계 입력이 'ON'이면 명령을 기동할 수 없습니다.

명령에 대한 자세한 내용

이 명령에 대해 자세히 설명합니다.

● 직선 보간의 순서

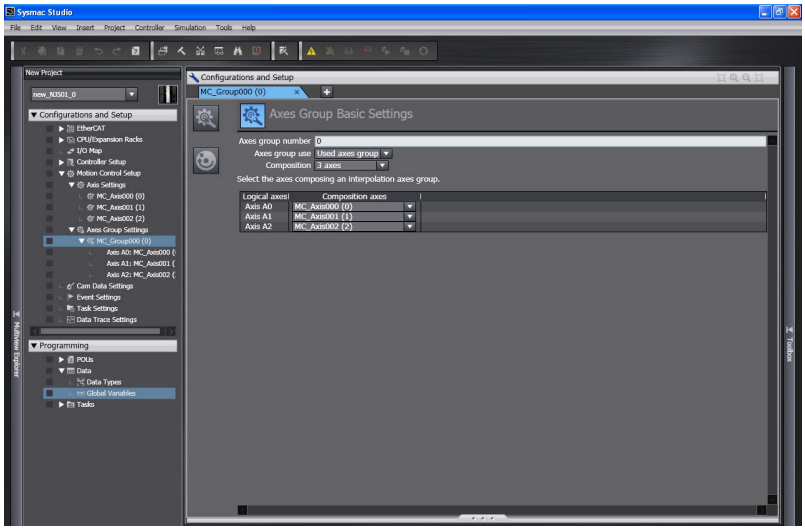
직선 보간은 다음 순서로 실행합니다.

1 보간을 실행할 축 그룹 등록

- 보간을 실행할 축 그룹을 결정합니다.
  축 그룹이란 _MC_GRP[***]입니다.
- 이 축 그룹 변수의 축 구성에서 축 구성을 지정합니다. 지정할 수 있는 축 수는 2~4축입니다.
- 이 축 그룹 변수의 축 선택에서 보간을 실행할 축의 조합을 지정합니다.
- 축은 축 번호(축 0～축 63)가 아니라 논리축(축 A0～축 A3)을 사용합니다.
- 축 선택에서 논리축 A0～A3에 각각 축 번호 0～63을 앞 채우기로 지정합니다.

논리축	축 번호
축 A0	축 0~축 63
축 A1	축 0~축 63
축 A2	축 0~축 63
축 A3	축 0~축 63

- Sysmac Studio의 축 그룹 설정에서 사용할 '축 구성'을 선택하고 논리축에 축 번호를 할당합니다.
  예를 들면 3축이 있는 축 그룹 'MC_Group000'에 대해 MC_Axis000, MC_Axis001, MC_Axis002를 등록하는 예는 다음과 같습니다.



사용상의 주의 사항

축 그룹에 엔코더 축 또는 가상 엔코더 축이 포함되어 있으면 '축 종류 부적합(에러 코드: 543D Hex)'이 되어 동작이 종료됩니다.
반드시 서보 축 또는 가상 서보 축을 선택하십시오.

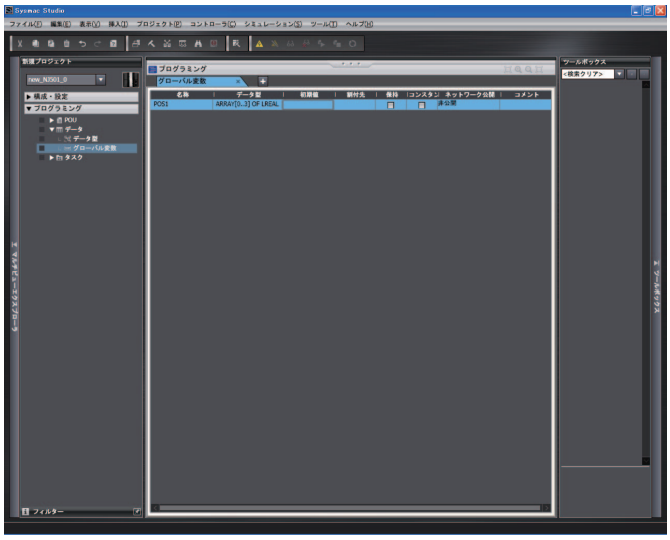
2 축 그룹의 유효화

- 축 그룹의 각 구성축을 서보 ON 상태로 하고, 그리고 이들을 원점 확정 상태로 합니다.
- MC_GroupEnable(축 그룹 유효) 명령을 실행하고 등록한 축 그룹을 유효로 합니다.

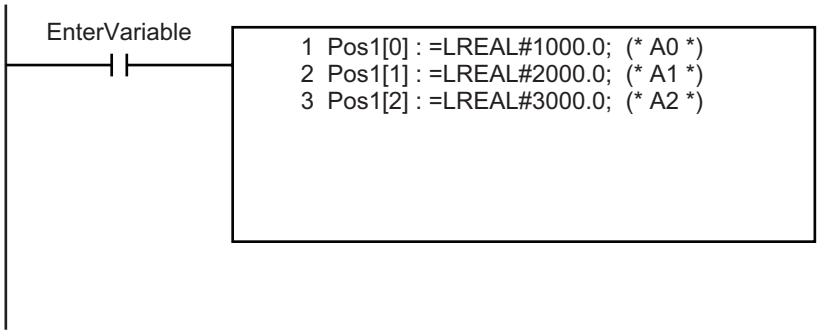
이제 직선 보간 명령을 사용할 수 있게 됩니다.

● Position(목표 위치)

- Position(목표 위치)에는 축 그룹 변수의 축 선택에서 지정한 축 수만큼의 목표 위치를 지정합니다.
- Position(A0, A1, A2, A3)에 대입하는 변수는 Sysmac Studio상에서 1×4의 배열 변수를 작성해야 합니다. 변수명은 임의로 붙일 수 있습니다.
  해당 배열에 각 축의 목표 위치를 대입합니다.
  사용하는 축 그룹에 속하는 축 수가 4축보다 적은 경우에도 반드시 1×4의 배열을 선언하십시오.
  Sysmac Studio에서 배열 변수 'Pos1'을 선언한 경우의 예는 다음과 같습니다.



또한 인라인 ST를 이용해서 'Pos1'에 목표 위치를 대입하는 예는 다음과 같습니다.
그림에서는 A0～A2의 각 축에 목표 위치(1000.0, 2000.0, 3000.0)를 대입했습니다.

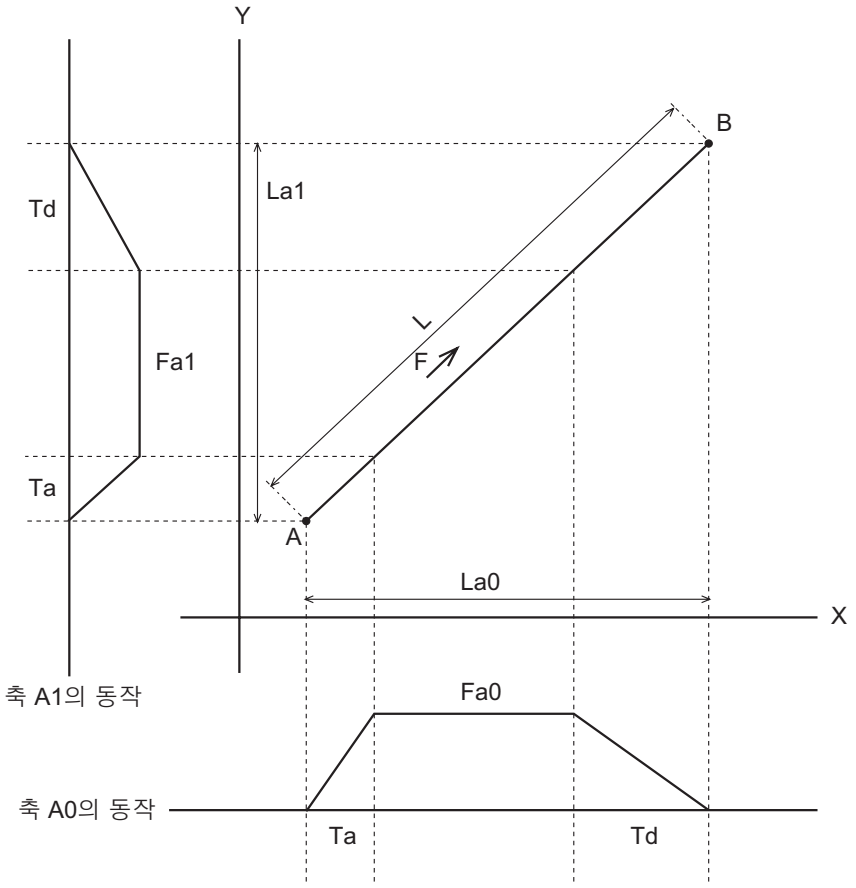


- 카운트 모드 선택에서 [로터리 모드]를 설정한 축을 보간축으로 지정한 경우 절대 위치를 지정하면 목표값은 Direction(방향 선택)이 [방향 지정 안 함]과 동일해집니다.
  자세한 내용은 'Direction(방향 선택) (3-51페이지)'을 참조하십시오.

● Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도), Jerk(저크)

- 직선 보간 시의 보간 속도, 가속도, 감속도, 저크는 Velocity, Acceleration, Deceleration, Jerk에서 지정합니다.
- 직선 보간의 경우 직선을 각 축으로 분해해서 움직입니다.

예를 들어 2축을 사용해 A점에서 B점까지의 직선 보간을 표시하면 다음 그림과 같습니다.



4축 직선 보간의 경우 각 축의 목표 속도는 보간 속도와 각 축의 이동 거리에 따라 다음과 같습니다.

F : 지정된 보간 이송 속도
Fa0 : F를 축 A0으로 전개한 보간 이송 속도
Fa1 : F를 축 A1으로 전개한 보간 이송 속도
Fa2 : F를 축 A2로 전개한 보간 이송 속도
Fa3 : F를 축 A3으로 전개한 보간 이송 속도
Ta : 보간 가속 시간
Td : 보간 감속 시간
L : 지정된 궤적의 이동량
La0, La1, La2, La3: 축 A0, 축 A1, 축 A2, 축 A3의 이동량

이라고 하면, L, Fa0, Fa1, Fa2, Fa3은 다음 식으로 나타낼 수 있습니다.

$$Fa0 = F \times \frac{La0}{L}$$

$$Fa1 = F \times \frac{La1}{L}$$

$$Fa2 = F \times \frac{La2}{L}$$

$$Fa3 = F \times \frac{La3}{L}$$

$$L = \sqrt{La0^2 + La1^2 + La2^2 + La3^2}$$

Velocity(목표 속도)

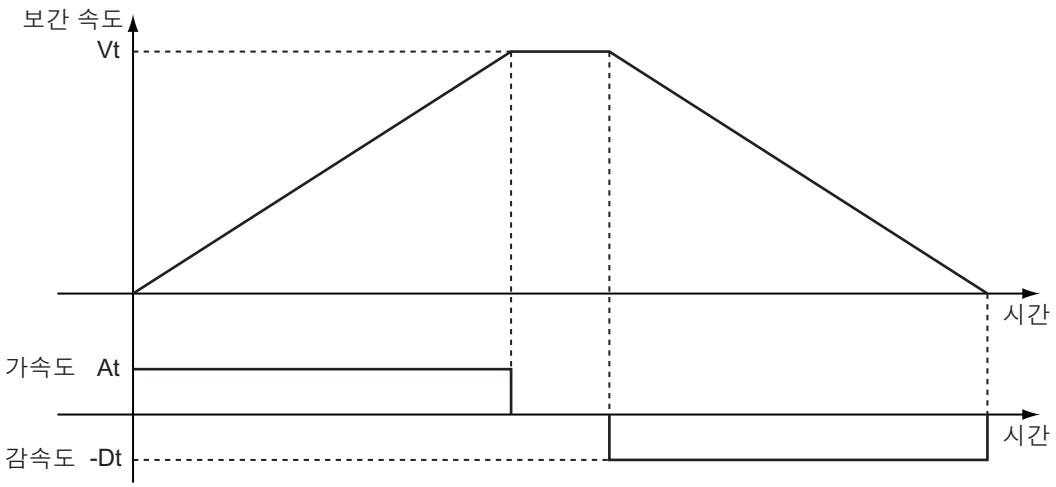
- Velocity(목표 속도)에 '0'을 지정하면 보간 속도 지정 이상이 발생합니다. 지정된 축 그룹의 축이 동작 중이면 전체 축이 정지합니다.
- 산출한 Velocity(목표 속도) Fa0～Fa3 중 하나가 각 축의 최고 속도를 초과하면 그 중 하나의 축이 최고 속도에서 움직이도록 Velocity(목표 속도)를 자동으로 조정합니다.

Jerk(저크)

Jerk(저크)에 '0'을 지정한 경우와 '0' 이외의 것을 지정한 경우의 Acceleration(가속도), Deceleration(감속도), Velocity(목표 속도)의 관계는 다음과 같습니다.

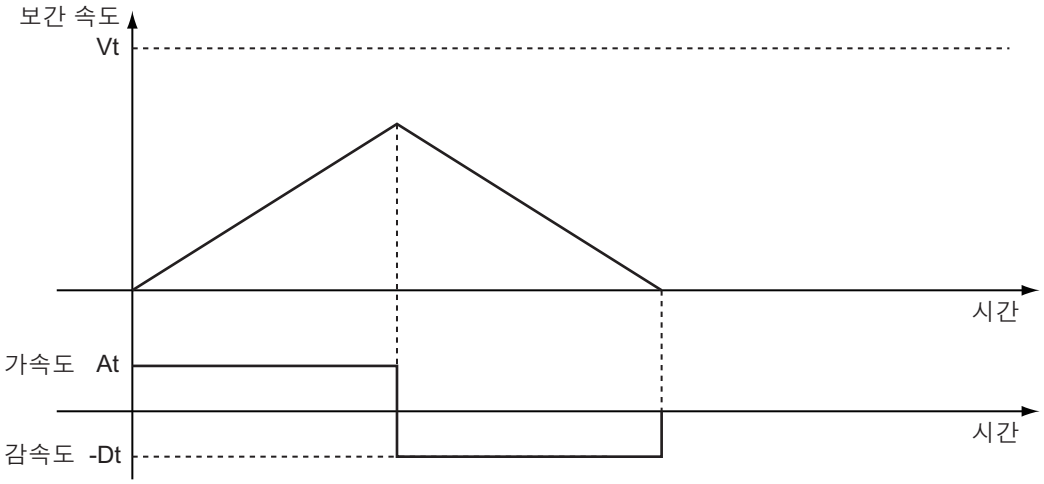
- Jerk(저크)가 '0'인 경우

  가속도 At, 감속도 Dt로 속도의 명령값을 작성합니다.



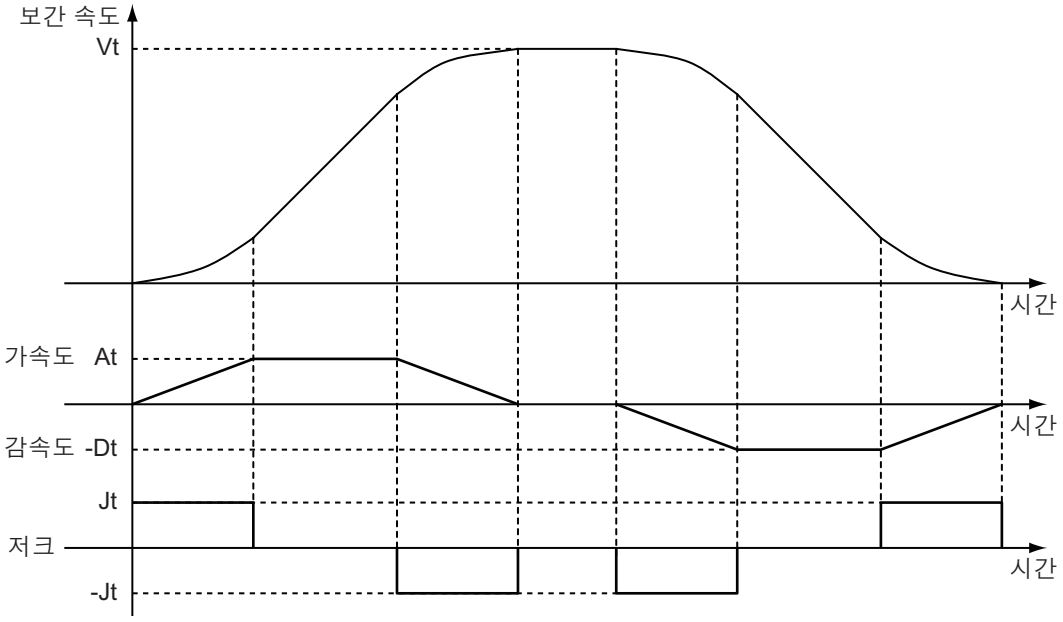
Vt : 보간 속도의 지정값, At : 가속도의 지정값, Dt : 감속도의 지정값

- Jerk(저크)가 '0'이고 이동 거리가 짧은 경우
  보간 속도는 지정값 Vt(목표 속도)까지 도달하지 않습니다.



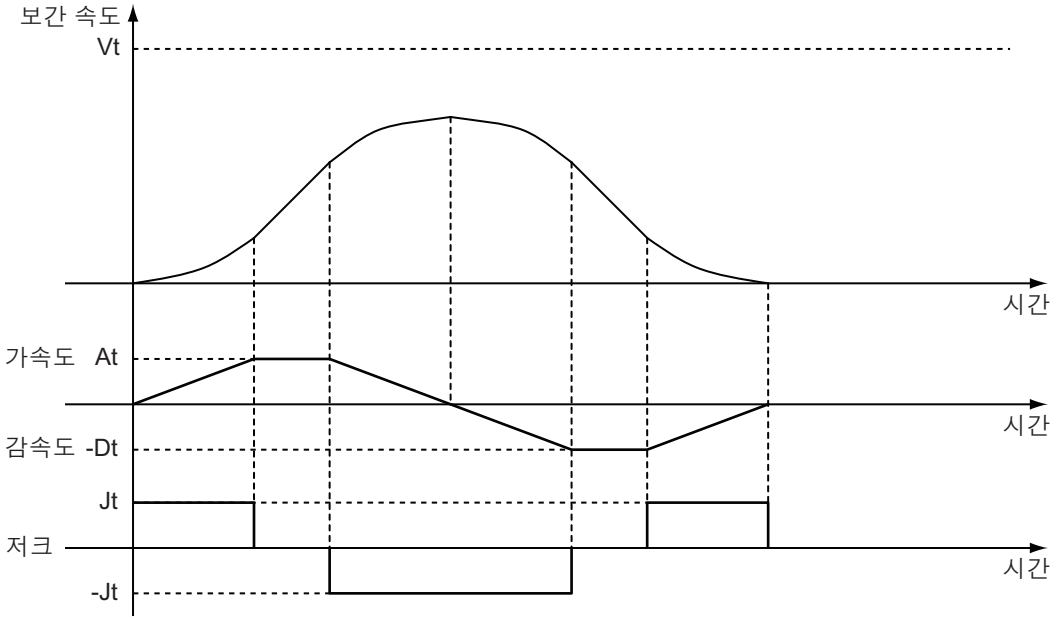
Vt : 보간 속도의 지정값, At : 가속도의 지정값, Dt : 감속도의 지정값

- Jerk(저크)가 '0'이 아닌 경우
  가속도의 상한은 At, 감속도의 상한은 Dt로 속도 명령값을 작성합니다.



Vt : 보간 속도의 지정값, At : 가속도의 지정값, Dt : 감속도의 지정값, Jt : 저크의 지정값

- Jerk(저크)가 '0' 이외이고 이동 거리가 짧은 경우
  보간 속도는 지정값 Vt(목표 속도)까지 도달하지 않습니다.



Vt : 보간 속도의 지정값, At : 가속도의 지정값, Dt : 감속도의 지정값, Jt : 저크의 지정값

참고

- Acceleration(가속도)에 '0'을 지정하면 바로 지정한 Velocity(목표 속도)로 움직입니다.
- Deceleration(감속도)에 '0'을 지정하면 바로 정지합니다. 단, 버퍼 모드로 Blending을 지정한 경우는 정지하지 않고 다음 동작에서 지정한 목표 속도로 이행합니다.
  자세한 내용은 'BufferMode(버퍼 모드 선택) (4-20페이지)'를 참조하십시오.
- Acceleration(가속도) 또는 Deceleration(감속도)에 '0'을 지정하면 저크의 설정은 무효가 됩니다.

● CoordSystem(좌표 계통)

- 원호 보간을 실행할 좌표 계통을 지정합니다.
- 2축 이상으로 구성되는 축 좌표 계통(ACS)에만 대응합니다.

● BufferMode(버퍼 모드 선택)

- 하나 이전의 보간 동작과 이번 보간 동작의 연결 방법을 지정합니다.
- 종류는 다음 6가지입니다.

버퍼 모드 선택		설 명
Aborting		현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.
Buffered		현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.
Blending		현재 실행 중인 명령이 목표 위치에 도달했을 때의 속도(중계 속도)를 기동 속도로 버퍼링 중인 이 명령을 연속해서 동작하게 합니다. 중계 속도로 목표 위치에 도달하도록 현재 실행 중인 명령의 동작을 변경합니다. 중계 속도 지정 방법은 다음의 4종류가 있습니다. 또한 Blending의 옵션으로 트랜지션 지정(뒤에 설명)이 있습니다.
	BlendingLow	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 낮은 것을 중계 속도로 합니다.
	BlendingPrevious	현재 실행 중인 목표 속도를 중계 속도로 합니다.
	BlendingNext	버퍼링되고 있는 이 명령의 목표 속도를 중계 속도로 합니다.
	BlendingHigh	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 높은 것을 중계 속도로 합니다.

BufferMode(버퍼 모드 선택)에 대해서는, 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● TransitionMode(트랜지션 모드)

- 하나 이전의 보간 동작과 이번 보간 동작으로 만들어지는 궤적의 연결 방법을 지정합니다.
- TransitionMode(트랜지션 모드)에서는 _mcTMNone(트랜지션 무효) 또는 _mcTMCornerSuperimposed(코너 수퍼 임포즈) 중 하나를 지정합니다.
- BufferMode(버퍼 모드 선택)에서 Blending을 지정하면 TransitionMode(트랜지션 모드)가 유효가 됩니다.
- Blending을 지정하지 않고 트랜지션 지정을 _mcTMNone(트랜지션 무효) 이외의 것으로 지정하면 이상이 됩니다.

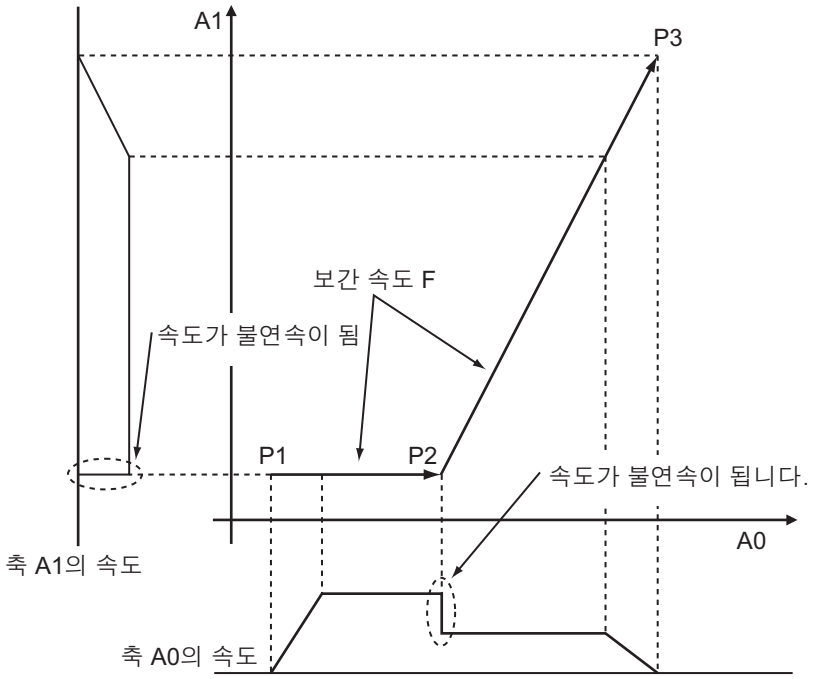
트랜지션 무효

궤적을 우선해서 속도 명령값을 작성하므로 각 축의 속도 명령값은 각 동작이 끊기는 시점에서 급격하게 속도가 변화하는 경우가 있습니다.

동작 예

P1에서 P2, P2에서 P3에 대한 Velocity(목표 속도), BufferMode(버퍼 모드 선택), TransitionMode(트랜지션 모드)는 다음과 같습니다.

- P1에서 P2에 대한 동작: Velocity = F, BufferMode = Aborting, TransitionMode = _mcTMNone(트랜지션 무효)
- P2에서 P3에 대한 동작: Velocity = F, BufferMode = BlendingNext, TransitionMode = _mcTMNone(트랜지션 무효)
- 위치 P1에서 움직이기 시작해 위치 P2를 통과하고 위치 P3으로 직선 보간으로 동작합니다.
- 위치 P2를 통과할 때 직선 보간의 속도 F를 유지합니다. 따라서 위치 P2에서는 그림에서 나타나듯이 속도가 불연속이 됩니다.



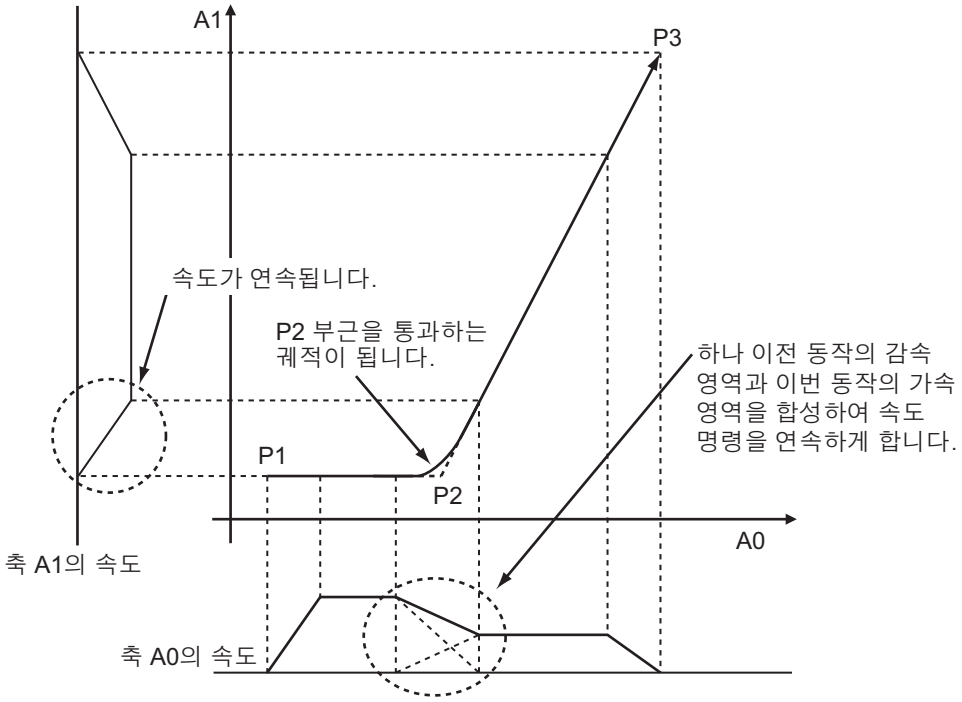
ConnerSuperimpose

각 축의 명령 속도를 연속으로 움직이려는 경우에 사용합니다.

동작 예

P1에서 P2, P2에서 P3에 대한 Velocity(목표 속도), BufferMode(버퍼 모드 선택), TransitionMode(트랜지션 모드)는 다음과 같습니다.

- P1에서 P2에 대한 동작: Velocity = F, BufferMode = Aborting, TransitionMode = _mcTMNone(트랜지션 무효)
- P2에서 P3에 대한 동작: Velocity = F, BufferMode = BlendingNext, TransitionMode = _mcTMCornerSuperimposed (코너 수퍼 임포즈)
- 위치 P1에서 움직이기 시작해 위치 P2 부근을 통과하고 위치 P3으로 직선 보간으로 동작합니다.
- 각 축의 명령 속도가 연속하도록 하려면 하나 이전 동작의 감속 영역과 이번 동작의 가속 영역을 합성하여 명령 속도를 작성합니다. 따라서 이번 동작의 가속 시간은 하나 이전 동작의 감속 시간과 같아집니다.



합성한 궤적은 P2 부근을 통과하는 궤적이 됩니다.

P2에서 이 궤적에 대한 거리는

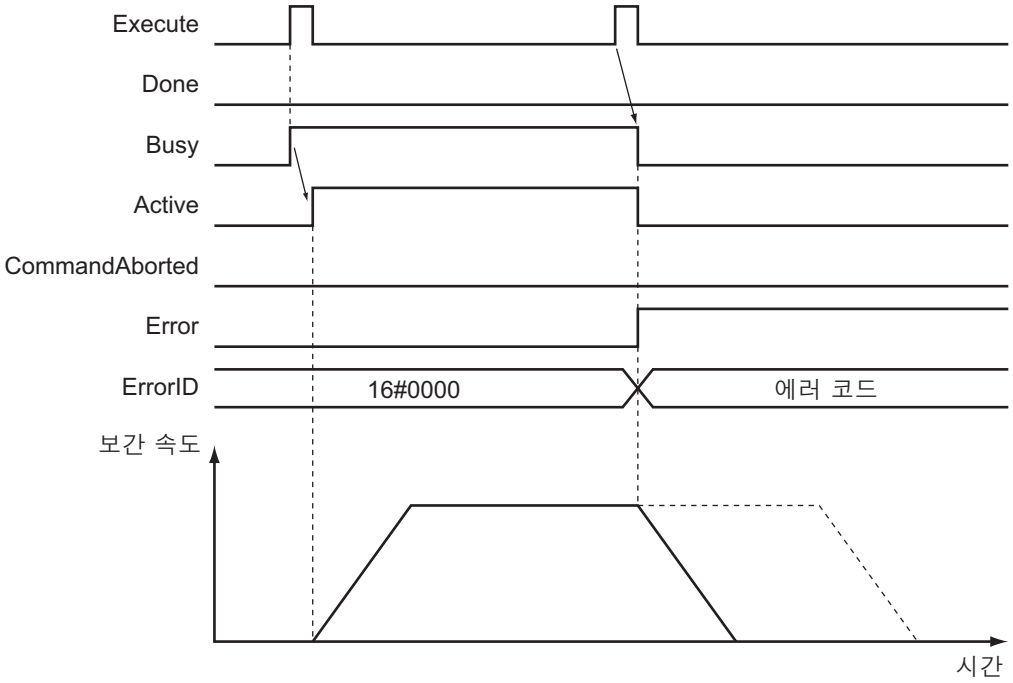
- 보간 속도가 빨라지거나 하나 이전 명령의 감속도가 작아지면 길어집니다.
- 보간 속도가 늦어지거나 하나 이전 명령의 감속도가 커지면 짧아집니다.

참고

ConnerSuperimpose하는 구간에서는 Jerk(저크)의 지정은 무효가 됩니다.

모션 명령 재기동

이 명령은 재기동할 수 없습니다. 재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 되고 직선 보간 중작 중인 전체 축이 정지합니다.

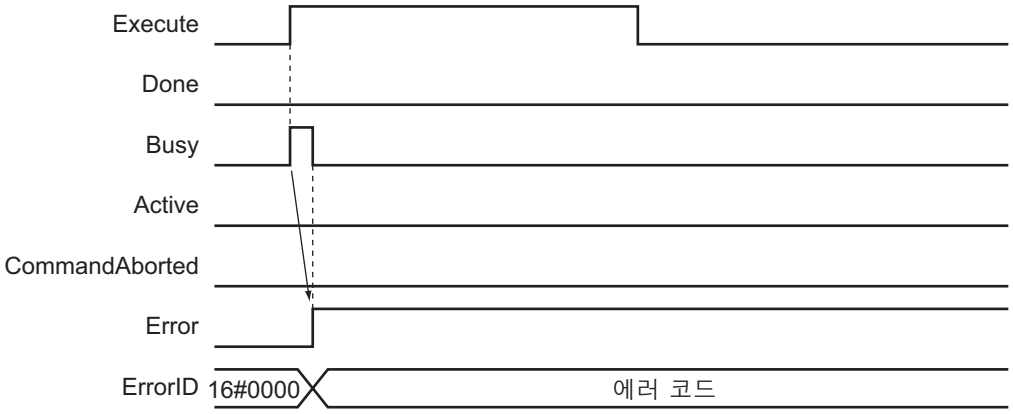


모션 명령 다중 기동

이 명령 실행 중에 사용할 수 있는 명령에는 제한이 있습니다.
모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

이상

이 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는, 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기에는 다중 기동 반복에 의한 직선 보간에 대한 샘플 프로그램이 실려 있습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축
축 2	서보 축

카운터 모드

축	카운터 모드
축 1	리니어 모드
축 2	리니어 모드

좌표 단위 선택

축	좌표 단위
축 1	mm
축 2	mm

● 축 그룹 파라메터 설정

축 구성

[2축]으로 합니다.

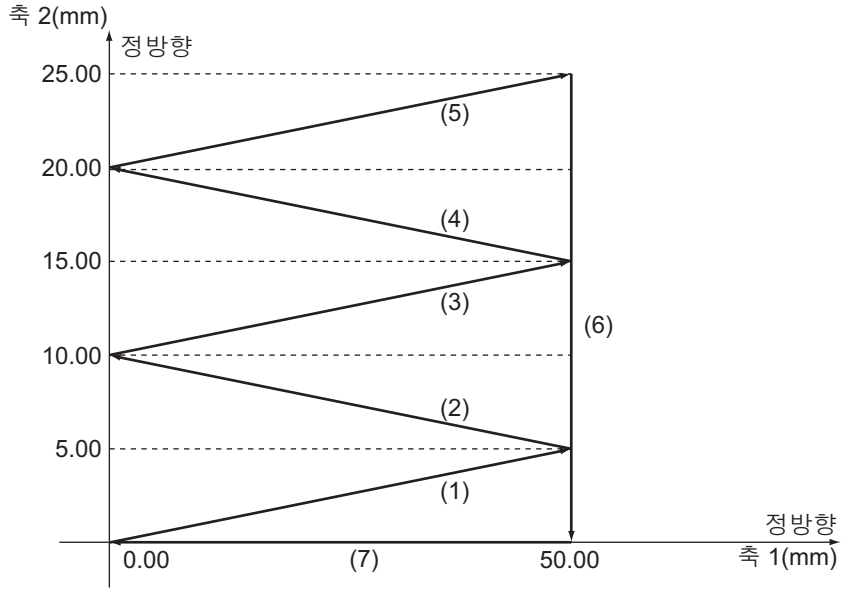
축 선택

[축 1], [축 2]로 합니다.

동작 예

직선 보간 동작을 자동 실행하고 원점으로 돌아온 지점에서 정지하는 동작 예는 다음과 같습니다.
직선 보간(1)이 동작 중에 직선 보간(2)~(7)을 다중 기동합니다. 이 때의 버퍼 모드 선택은 Buffered로 합니다.
이 예에서는 직선 보간(1)의 출력 변수 'Active(제어 중)'가 TRUE가 되었을 때 (2)~(7)의 명령 다중 기동을 실행하고 있습니다. 다축 협조 동작에서는 최대 7명령까지 다중 기동이 가능합니다.

● 동작 패턴



직선 보간을 이용해 (축 1, 축 2) = (50.00mm, 5.00mm) → (0.00mm, 10.00mm) → (50.00mm, 15.00mm) → (0.00mm, 20.00mm) → (50.00mm, 25.00mm) → (50.00mm, 0.00mm) → (0.00mm, 0.00mm)의 순으로 위치 결정을 실행하고 정지합니다.

래더도

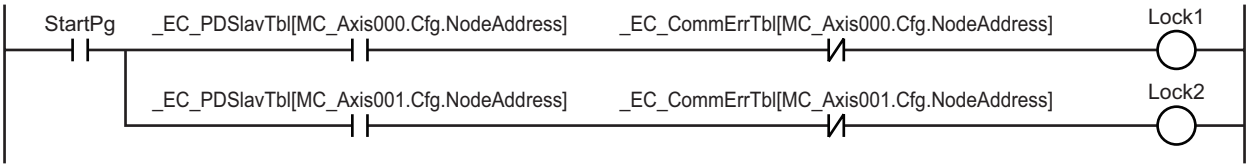
● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Group000	_sGROUP_REF	–	축 그룹 0의 축 그룹 변수입니다.
MC_Group000.MFaultLvl.Active	BOOL	FALSE	축 그룹 0에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Group000.Status.Disabled	BOOL	FALSE	축 그룹 0이 무효인 경우에 TRUE가 됩니다.
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.

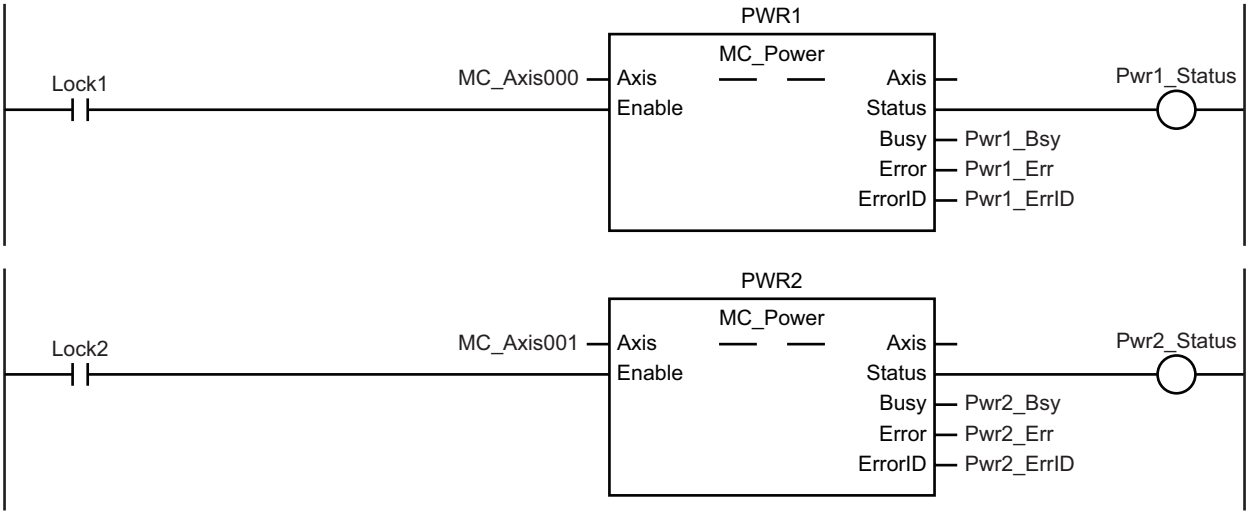
이 름	데이터 형식	초기값	코멘트
Pwr1_Status	BOOL	FALSE	MC_Power의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power의 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 축 그룹에 속하는 각 축을 서보 ON 상태로 합니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 샘플 프로그램

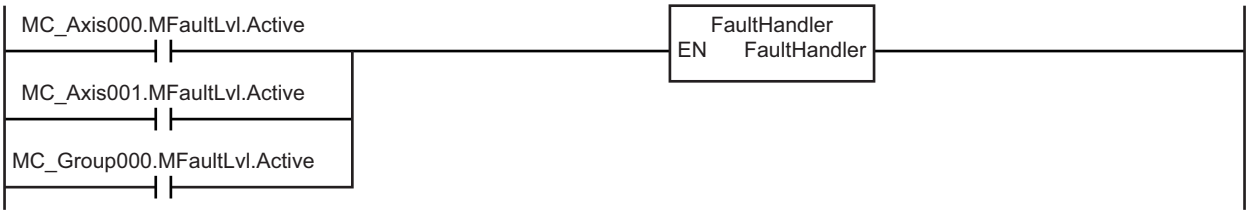
접점 StartPg가 TRUE가 되면 각 축의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인



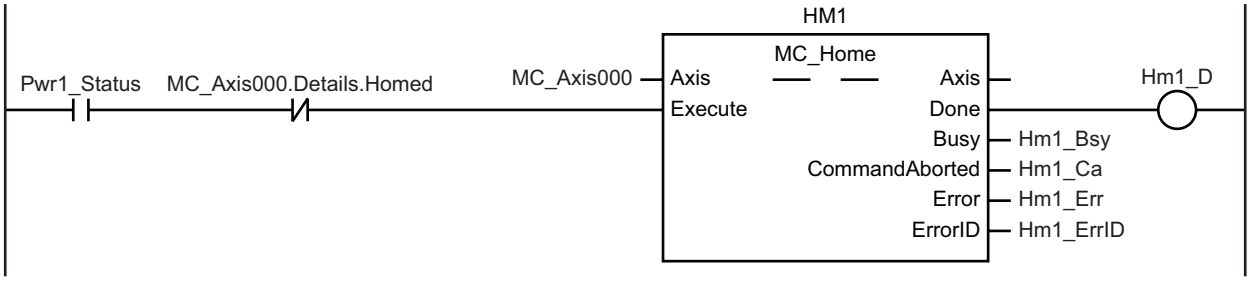
프로세스 데이터 통신을 하고 있으면 각 축의 서보를 ON



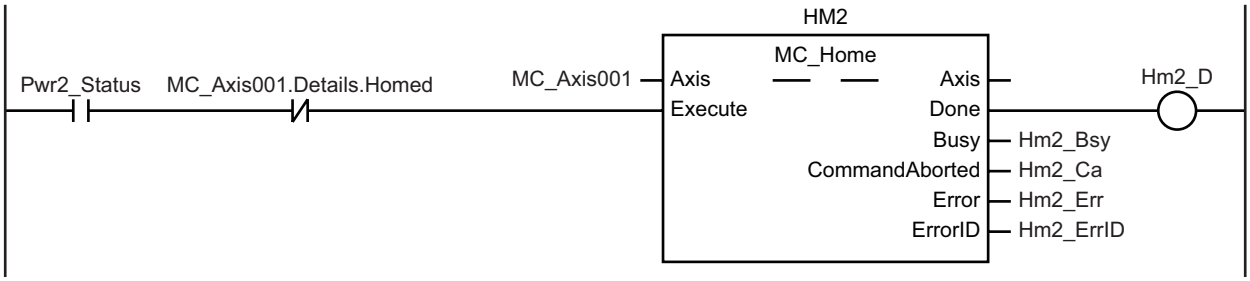
축 구성에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



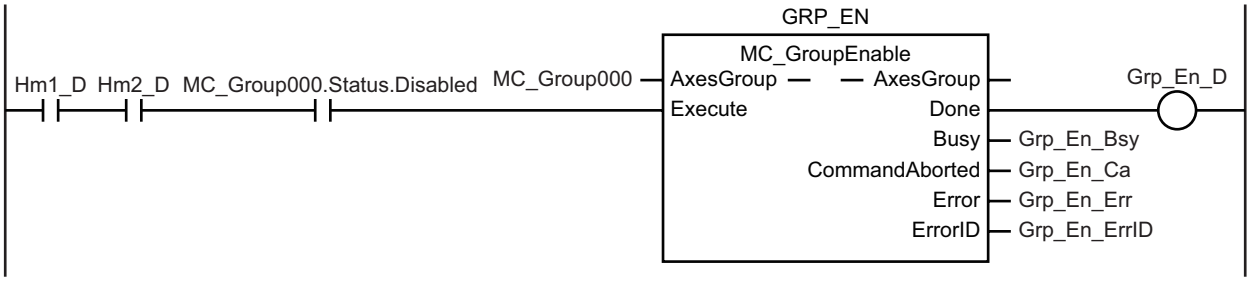
축 1이 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행하고 원점을 확정



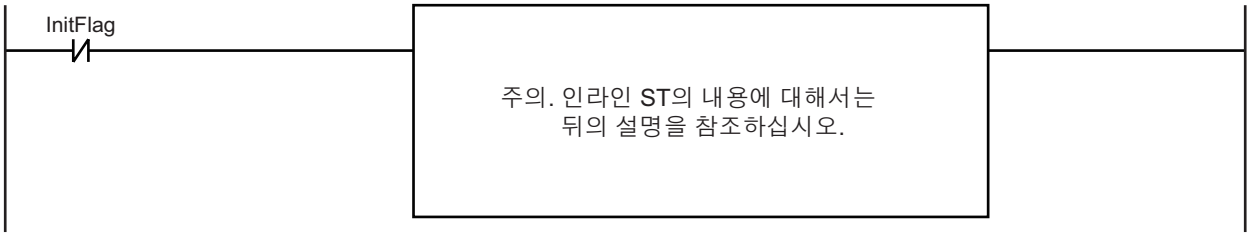
축 2가 서보가 ON 되고 원점 미확정 상태라면 원점 복귀를 실행하고 원점을 확정



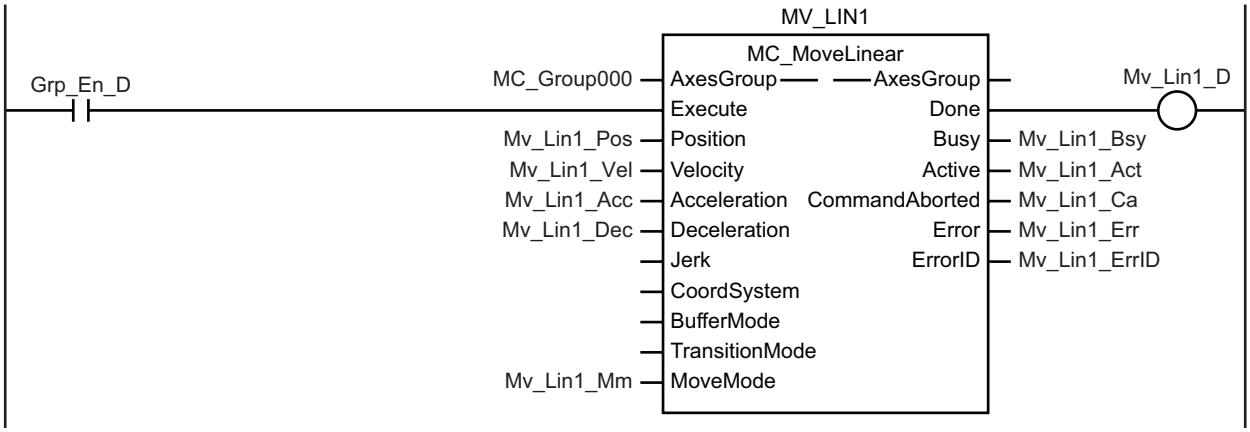
축 1과 축 2의 원점 확정 후 축 그룹을 유효로 함



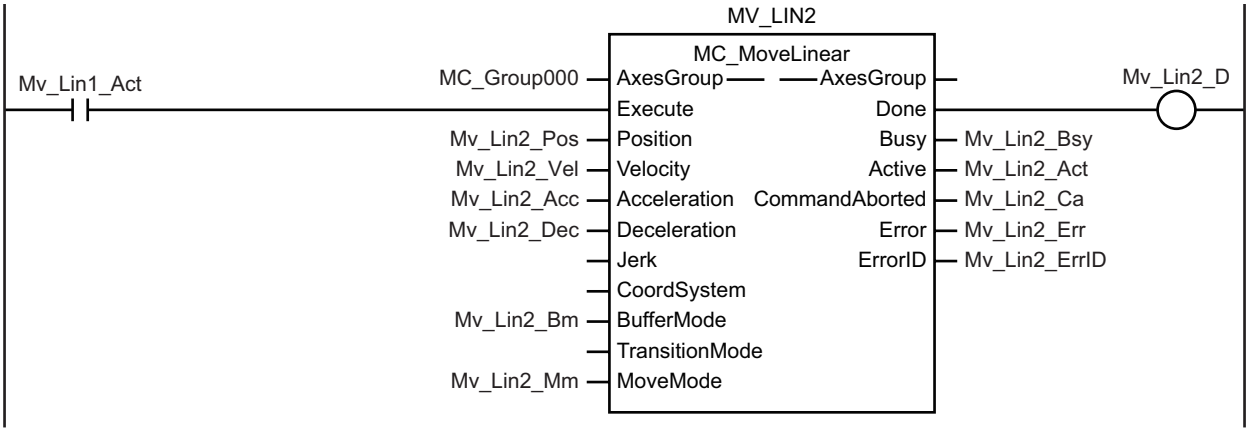
직선 보간의 파라메터를 설정



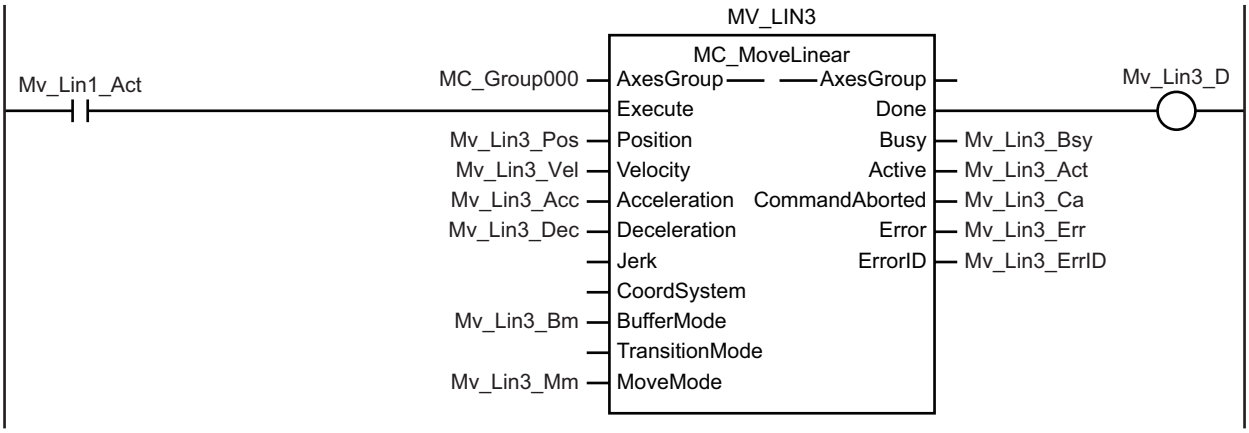
축 그룹이 유효가 된 것을 확인하고 직선 보간(1)을 실행



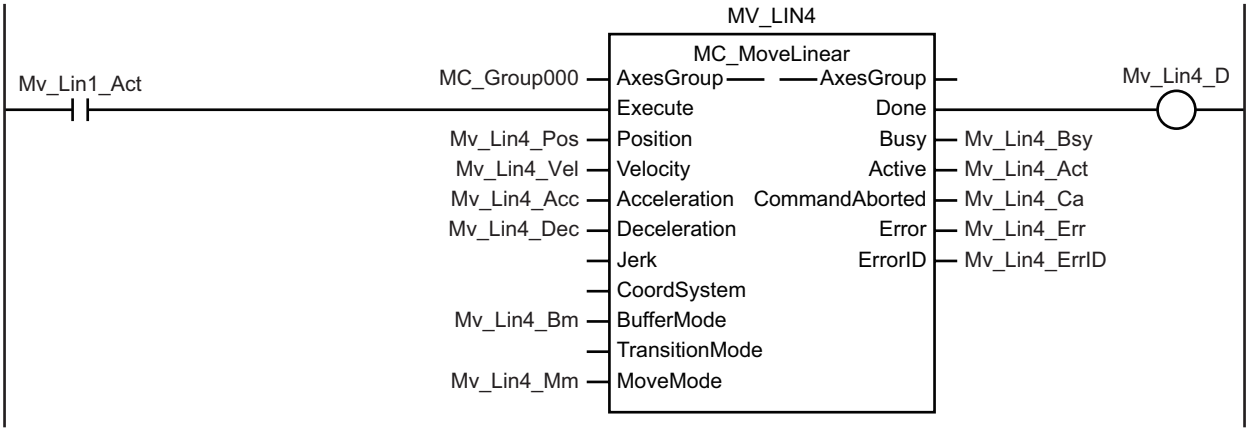
직선 보간(1) 시작 후 직선 보간(2)를 다중 기동



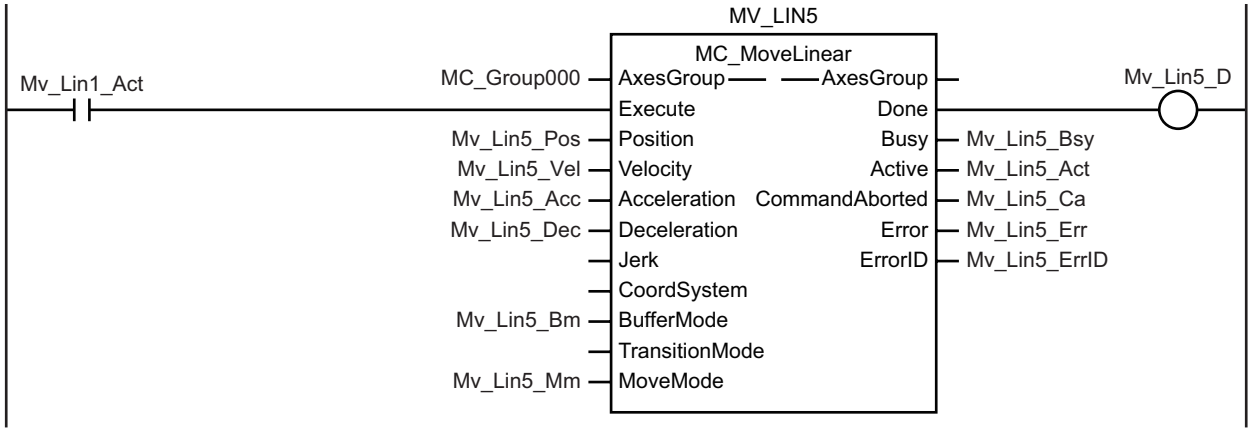
직선 보간(1) 시작 후 직선 보간(3)을 다중 기동



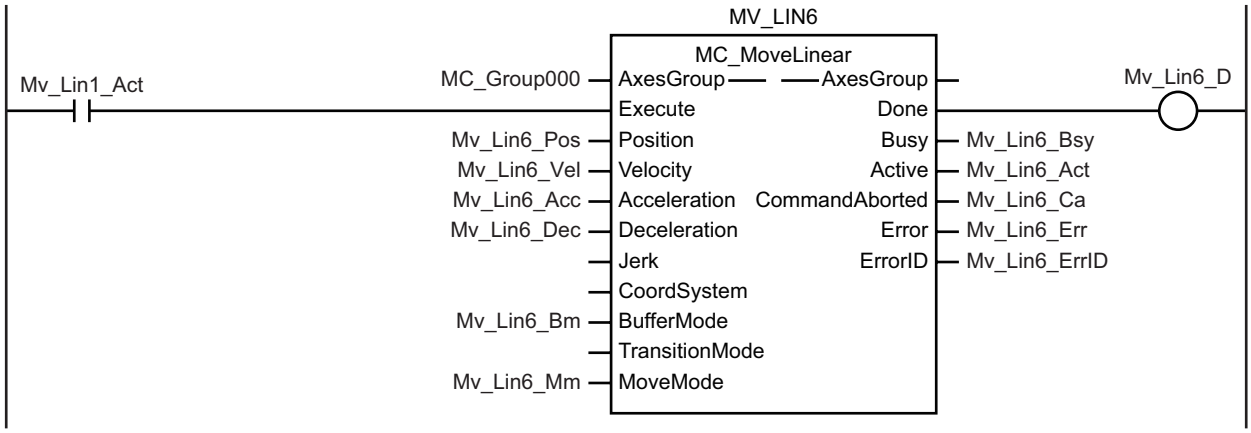
직선 보간(1) 시작 후 직선 보간(4)를 다중 기동



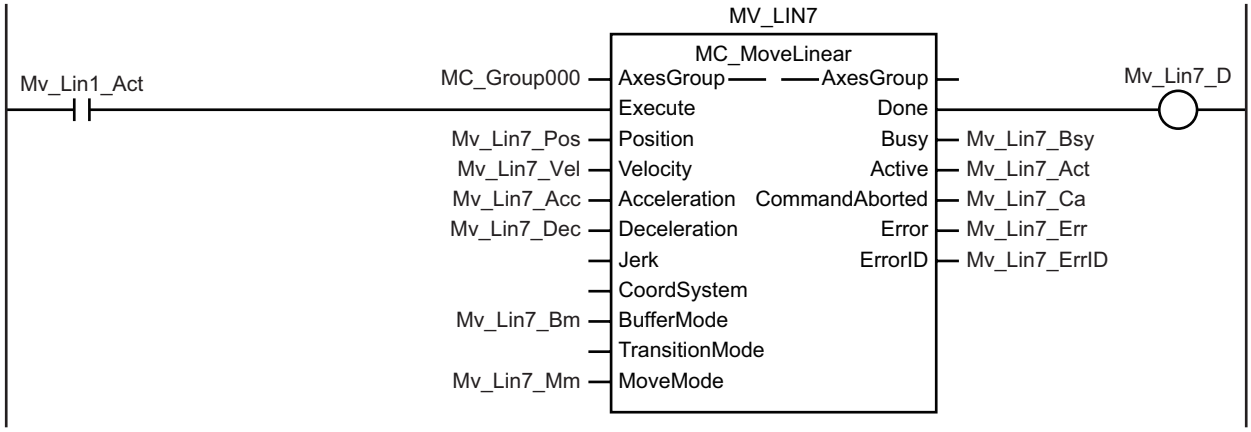
직선 보간(1) 시작 후 직선 보간(5)를 다중 기동



직선 보간(1) 시작 후 직선 보간(6)을 다중 기동



직선 보간(1) 시작 후 직선 보간(7)을 다중 기동



인라인 ST의 내용

```
//MV_LIN1 파라메터
    Mv_Lin1_Pos[0]    := LREAL#50.0;
    Mv_Lin1_Pos[1]    := LREAL#5.0;
    Mv_Lin1_Vel       := LREAL#100.0;
    Mv_Lin1_Acc       := LREAL#100.0;
    Mv_Lin1_Dec       := LREAL#100.0;
    Mv_Lin1_Mm        := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN2 파라메터
    Mv_Lin2_Pos[0]    := LREAL#0.0;
    Mv_Lin2_Pos[1]    := LREAL#10.0;
    Mv_Lin2_Vel       := LREAL#100.0;
    Mv_Lin2_Acc       := LREAL#100.0;
```

```
    Mv_Lin2_Dec      := LREAL#100.0;
    Mv_Lin2_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin2_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN3 파라메터
    Mv_Lin3_Pos[0]   := LREAL#50.0;
    Mv_Lin3_Pos[1]   := LREAL#15.0;
    Mv_Lin3_Vel      := LREAL#100.0;
    Mv_Lin3_Acc      := LREAL#100.0;
    Mv_Lin3_Dec      := LREAL#100.0;
    Mv_Lin3_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin3_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN4 파라메터
    Mv_Lin4_Pos[0]   := LREAL#0.0;
    Mv_Lin4_Pos[1]   := LREAL#20.0;
    Mv_Lin4_Vel      := LREAL#100.0;
    Mv_Lin4_Acc      := LREAL#100.0;
    Mv_Lin4_Dec      := LREAL#100.0;
    Mv_Lin4_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin4_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN5 파라메터
    Mv_Lin5_Pos[0]   := LREAL#50.0;
    Mv_Lin5_Pos[1]   := LREAL#25.0;
    Mv_Lin5_Vel      := LREAL#100.0;
    Mv_Lin5_Acc      := LREAL#100.0;
    Mv_Lin5_Dec      := LREAL#100.0;
    Mv_Lin5_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin5_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN6 파라메터
    Mv_Lin6_Pos[0]   := LREAL#50.0;
    Mv_Lin6_Pos[1]   := LREAL#0.0;
    Mv_Lin6_Vel      := LREAL#100.0;
    Mv_Lin6_Acc      := LREAL#100.0;
    Mv_Lin6_Dec      := LREAL#100.0;
    Mv_Lin6_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin6_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN7 파라메터
    Mv_Lin7_Pos[0]   := LREAL#0.0;
    Mv_Lin7_Pos[1]   := LREAL#0.0;
    Mv_Lin7_Vel      := LREAL#100.0;
    Mv_Lin7_Acc      := LREAL#100.0;
    Mv_Lin7_Dec      := LREAL#100.0;
    Mv_Lin7_Bm       := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin7_Mm       := _eMC_MOVE_MODE#_mcAbsolute;

//입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag :=TRUE;
```

스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Group000	_sGROUP _REF	–	축 그룹 0의 축 그룹 변수입니다.
MC_Group000.MFaultLvl.Active	BOOL	FALSE	축 그룹 0에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Group000.Status.Disabled	BOOL	FALSE	축 그룹 0이 무효인 경우에 TRUE가 됩니다.
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
Pwr1_Status	BOOL	FALSE	MC_Power의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power의 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 축 그룹에 속하는 각 축을 서보 ON 상태로 합니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag=FALSE THEN

    //MV_LIN1 파라메터
    Mv_Lin1_Pos[0]    := LREAL#50.0;
    Mv_Lin1_Pos[1]    := LREAL#5.0;
    Mv_Lin1_Vel       := LREAL#100.0;
    Mv_Lin1_Acc       := LREAL#100.0;
    Mv_Lin1_Dec       := LREAL#100.0;
    Mv_Lin1_Mm        := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN2 파라메터
    Mv_Lin2_Pos[0]    := LREAL#0.0;
    Mv_Lin2_Pos[1]    := LREAL#10.0;
    Mv_Lin2_Vel       := LREAL#100.0;
    Mv_Lin2_Acc       := LREAL#100.0;
    Mv_Lin2_Dec       := LREAL#100.0;
    Mv_Lin2_Bm        := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin2_Mm        := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN3 파라메터
    Mv_Lin3_Pos[0]    := LREAL#50.0;
    Mv_Lin3_Pos[1]    := LREAL#15.0;
```

```
    Mv_Lin3_Vel         := LREAL#100.0;
    Mv_Lin3_Acc         := LREAL#100.0;
    Mv_Lin3_Dec         := LREAL#100.0;
    Mv_Lin3_Bm          := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin3_Mm          := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN4 파라메터
    Mv_Lin4_Pos[0]      := LREAL#0.0;
    Mv_Lin4_Pos[1]      := LREAL#20.0;
    Mv_Lin4_Vel         := LREAL#100.0;
    Mv_Lin4_Acc         := LREAL#100.0;
    Mv_Lin4_Dec         := LREAL#100.0;
    Mv_Lin4_Bm          := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin4_Mm          := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN5 파라메터
    Mv_Lin5_Pos[0]      := LREAL#50.0;
    Mv_Lin5_Pos[1]      := LREAL#25.0;
    Mv_Lin5_Vel         := LREAL#100.0;
    Mv_Lin5_Acc         := LREAL#100.0;
    Mv_Lin5_Dec         := LREAL#100.0;
    Mv_Lin5_Bm          := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin5_Mm          := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN6 파라메터
    Mv_Lin6_Pos[0]      := LREAL#50.0;
    Mv_Lin6_Pos[1]      := LREAL#0.0;
    Mv_Lin6_Vel         := LREAL#100.0;
    Mv_Lin6_Acc         := LREAL#100.0;
    Mv_Lin6_Dec         := LREAL#100.0;
    Mv_Lin6_Bm          := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin6_Mm          := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN7 파라메터
    Mv_Lin7_Pos[0]      := LREAL#0.0;
    Mv_Lin7_Pos[1]      := LREAL#0.0;
    Mv_Lin7_Vel         := LREAL#100.0;
    Mv_Lin7_Acc         := LREAL#100.0;
    Mv_Lin7_Dec         := LREAL#100.0;
    Mv_Lin7_Bm          := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin7_Mm          := _eMC_MOVE_MODE#_mcAbsolute;


    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag:=TRUE;

END_IF;


//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 1을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr1_En:=TRUE;          //축 1을 서보 ON합니다.
ELSE
    Pwr1_En:=FALSE;         //축 1을 서보 OFF합니다.
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 2를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
```

```
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
    Pwr2_En:=TRUE;       //축 2를 서보 ON합니다.
ELSE
    Pwr2_En:=FALSE;      //축 2를 서보 OFF합니다.
END_IF;

//가벼운 수준의 이상 발생 시 처리
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) OR
(MC_Group000.MFaultLvl.Active=TRUE)THEN
    FaultHandler();
END_IF;

//축 1이 서보 ON 상태이고 원점 미확정 상태이면 원점 확정을 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//축 2가 서보 ON 상태이고 원점 미확정 상태이면 원점 확정을 실행합니다.
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//축 1과 축 2가 원점 확정 상태이고 축 그룹 0이 무효일 때 축 그룹 0을 유효로 합니다.
IF (Hm1_D=TRUE) AND (Hm2_D=TRUE) AND (MC_Group000.Status.Disabled=TRUE) THEN
        Grp_En_Ex:= TRUE;
END_IF;

//축 그룹 유효 MC_GroupEnable이 완료되면 직선 보간(1)을 실행합니다.
IF Grp_En_D=TRUE THEN
    Mv_Lin1_Ex:=TRUE;
END_IF;

//직선 보간(1)의 출력 Active가 TRUE가 되면 직선 보간(2)~(7)을 다중 기동합니다.
IF Mv_Lin1_Act=TRUE THEN
    Mv_Lin2_Ex:=TRUE;
    Mv_Lin3_Ex:=TRUE;
    Mv_Lin4_Ex:=TRUE;
    Mv_Lin5_Ex:=TRUE;
    Mv_Lin6_Ex:=TRUE;
    Mv_Lin7_Ex:=TRUE;
END_IF;


//축 1의 MC_Power
PWR1(
    Axis        := MC_Axis000,
    Enable      := Pwr1_En,
    Status      => Pwr1_Status,
    Busy        => Pwr1_Bsy,
    Error       => Pwr1_Err,
    ErrorID     => Pwr1_ErrID
);

//축 2의 MC_Power
PWR2(
    Axis        := MC_Axis001,
    Enable      := Pwr2_En,
    Status      => Pwr2_Status,
    Busy        => Pwr2_Bsy,
    Error       => Pwr2_Err,
    ErrorID     => Pwr2_ErrID
);
```

```
//축 1의 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 2의 MC_Home
HM2(
    Axis                := MC_Axis001,
    Execute             := Hm2_Ex,
    Done                => Hm2_D,
    Busy                => Hm2_Bsy,
    CommandAborted      => Hm2_Ca,
    Error               => Hm2_Err,
    ErrorID             => Hm2_ErrID
);

//축 그룹 0을 유효로 함
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
);


//직선 보간(1)
MV_LIN1(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Lin1_Ex,
    Position            := Mv_Lin1_Pos,
    Velocity            := Mv_Lin1_Vel,
    Acceleration        := Mv_Lin1_Acc,
    Deceleration        := Mv_Lin1_Dec,
    MoveMode            := Mv_Lin1_Mm,
    Done                => Mv_Lin1_D,
    Busy                => Mv_Lin1_Bsy,
    Active              => Mv_Lin1_Act,
    CommandAborted      => Mv_Lin1_Ca,
    Error               => Mv_Lin1_Err,
    ErrorID             => Mv_Lin1_ErrID
);

//직선 보간(2)
MV_LIN2(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Lin2_Ex,
    Position            := Mv_Lin2_Pos,
    Velocity            := Mv_Lin2_Vel,
    Acceleration        := Mv_Lin2_Acc,
    Deceleration        := Mv_Lin2_Dec,
    BufferMode          := Mv_Lin2_Bm,
    MoveMode            := Mv_Lin2_Mm,
    Done                => Mv_Lin2_D,
    Busy                => Mv_Lin2_Bsy,
    Active              => Mv_Lin2_Act,
    CommandAborted      => Mv_Lin2_Ca,
```

```
    Error               => Mv_Lin2_Err,
    ErrorID             => Mv_Lin2_ErrID
);

//직선 보간(3)
MV_LIN3(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Lin3_Ex,
    Position            := Mv_Lin3_Pos,
    Velocity            := Mv_Lin3_Vel,
    Acceleration        := Mv_Lin3_Acc,
    Deceleration        := Mv_Lin3_Dec,
    BufferMode          := Mv_Lin3_Bm,
    MoveMode            := Mv_Lin3_Mm,
    Done                => Mv_Lin3_D,
    Busy                => Mv_Lin3_Bsy,
    Active              => Mv_Lin3_Act,
    CommandAborted      => Mv_Lin3_Ca,
    Error               => Mv_Lin3_Err,
    ErrorID             => Mv_Lin3_ErrID
);

//직선 보간(4)
MV_LIN4(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Lin4_Ex,
    Position            := Mv_Lin4_Pos,
    Velocity            := Mv_Lin4_Vel,
    Acceleration        := Mv_Lin4_Acc,
    Deceleration        := Mv_Lin4_Dec,
    BufferMode          := Mv_Lin4_Bm,
    MoveMode            := Mv_Lin4_Mm,
    Done                => Mv_Lin4_D,
    Busy                => Mv_Lin4_Bsy,
    Active              => Mv_Lin4_Act,
    CommandAborted      => Mv_Lin4_Ca,
    Error               => Mv_Lin4_Err,
    ErrorID             => Mv_Lin4_ErrID
);

//직선 보간(5)
MV_LIN5(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Lin5_Ex,
    Position            := Mv_Lin5_Pos,
    Velocity            := Mv_Lin5_Vel,
    Acceleration        := Mv_Lin5_Acc,
    Deceleration        := Mv_Lin5_Dec,
    BufferMode          := Mv_Lin5_Bm,
    MoveMode            := Mv_Lin5_Mm,
    Done                => Mv_Lin5_D,
    Busy                => Mv_Lin5_Bsy,
    Active              => Mv_Lin5_Act,
    CommandAborted      => Mv_Lin5_Ca,
    Error               => Mv_Lin5_Err,
    ErrorID             => Mv_Lin5_ErrID
);

//직선 보간(6)
MV_LIN6(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Lin6_Ex,
    Position            := Mv_Lin6_Pos,
    Velocity            := Mv_Lin6_Vel,
    Acceleration        := Mv_Lin6_Acc,
```

```
    Deceleration        := Mv_Lin6_Dec,
    BufferMode          := Mv_Lin6_Bm,
    MoveMode            := Mv_Lin6_Mm,
    Done                => Mv_Lin6_D,
    Busy                => Mv_Lin6_Bsy,
    Active              => Mv_Lin6_Act,
    CommandAborted      => Mv_Lin6_Ca,
    Error               => Mv_Lin6_Err,
    ErrorID             => Mv_Lin6_ErrID
);
//직선 보간(7)
MV_LIN7(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Lin7_Ex,
    Position            := Mv_Lin7_Pos,
    Velocity            := Mv_Lin7_Vel,
    Acceleration        := Mv_Lin7_Acc,
    Deceleration        := Mv_Lin7_Dec,
    BufferMode          := Mv_Lin7_Bm,
    MoveMode            := Mv_Lin7_Mm,
    Done                => Mv_Lin7_D,
    Busy                => Mv_Lin7_Bsy,
    Active              => Mv_Lin7_Act,
    CommandAborted      => Mv_Lin7_Ca,
    Error               => Mv_Lin7_Err,
    ErrorID             => Mv_Lin7_ErrID
);
```

MC_MoveLinearAbsolute

절대 위치를 지정해서 직선 보간을 실행합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
MC_MoveLinearAbsolute	절대값 직선 보간	FB	MC_MoveLinearAbsolute_instance MC_MoveLinearAbsolute AxesGroup — AxesGroup Execute — Done Position — Busy Velocity — Active Acceleration — CommandAborted Deceleration — Error Jerk — ErrorID CoordSystem BufferMode TransitionMode	MC_MoveLinearAbsolute_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, Position := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, CoordSystem := 《파라메터》, BufferMode := 《파라메터》, TransitionMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Position	목표 위치	ARRAY [0..3] OF LREAL	음수, 양수, '0'	0	직선 보간의 목표 위치를 지정합니다. 단위는 [명령 단위]입니다. *1
Velocity*2	목표 속도	LREAL	양수	0	목표 속도를 지정합니다. 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1
Coord System	좌표 계통	_eMC_COORD_SYSTEM	0: _mcACS	0 *3	좌표 계통을 지정합니다. 0: 축 좌표 계통(ACS)

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Buffer Mode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered 2 : BlendingLow 3 : BlendingPrevious 4 : BlendingNext 5 : BlendingHigh
Transition Mode	트랜지션 모드 (이동 모드)	_eMC_ TRANSITION_ MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	동작 경로를 지정합니다. 0 : 트랜지션 무효 10 : ConnerSuper impose

*1. 명령 단위에 대해서는, 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 목표 속도를 반드시 설정하십시오. 설정하지 않고 동작하게 하면 이상이 발생합니다.
*3. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 🕮 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	위치 결정이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	축이 이동을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_GroupStop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- 2축~4축의 직선 보간을 실행합니다.
- 목표 위치는 절대 위치로 지정합니다.

그 밖에는 MC_MoveLinear(직선 보간) 명령과 사양이 동일합니다.

자세한 내용은 '기능 설명(4-14페이지)'을 참조하십시오.

사용상의 주의 사항

- 축 그룹의 구성 축 중 하나가 원점 미확정인 경우는 '원점 미확정 상태에서 명령 기동 이상(에러 코드: 5466 Hex)'이 됩니다.
- 직선 보간을 실행할 경우 해당 축 그룹에 속하는 논리축 중 하나의 한계 입력이 'ON'이면 명령을 기동할 수 없습니다.

MC_MoveLinearRelative

상대 위치를 지정해서 직선 보간을 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_MoveLinearRelative	상대값 직선 보간	FB	MC_MoveLinearRelative_instance MC_MoveLinearRelative AxesGroup — — AxesGroup Execute — Done Distance — Busy Velocity — Active Acceleration — CommandAborted Deceleration — Error Jerk — ErrorID CoordSystem BufferMode TransitionMode	MC_MoveLinearRelative_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, Distance := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, CoordSystem := 《파라메터》, BufferMode := 《파라메터》, TransitionMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Distance	이동 거리	ARRAY [0..3] OF LREAL	음수, 양수, '0'	0	직선 보간의 목표 위치를 지정합니다. 단위는 [명령 단위]입니다. *1
Velocity*2	목표 속도	LREAL	양수	0	목표 속도를 지정합니다. 단위는 [명령 단위/s]입니다. *1
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1
CoordSystem	좌표 계통	_eMC_ COORD_ SYSTEM	0: _mcACS	0 *3	좌표 계통을 지정합니다. 0 : 축 좌표 계통(ACS)

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered 2 : BlendingLow 3 : BlendingPrevious 4 : BlendingNext 5 : BlendingHigh
Transition Mode	트랜지션 모드 (이동 모드)	_eMC_ TRANSITION _MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	동작 경로를 지정합니다. 0 : 트랜지션 무효 10 : ConnerSuper impose

*1. 명령 단위에 대해서는, 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*2. 목표 속도를 반드시 설정하십시오. 설정하지 않고 동작하게 하면 이상이 발생합니다.
*3. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 🕮 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	위치 결정이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	축이 이동을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_GroupStop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- 2축~4축의 직선 보간을 실행합니다.
- 목표 위치는 상대 위치로 지정합니다.

그 밖에는 MC_MoveLinear(직선 보간) 명령과 사양이 동일합니다.

자세한 내용은 📖 '기능 설명(4-14페이지)'을 참조하십시오.

사용상의 주의 사항

- 축 그룹의 구성 축 중 하나가 원점 미확정인 경우는 '원점 미확정 상태에서 명령 기동 이상(에러 코드: 5466 Hex)'이 됩니다.
- 직선 보간을 실행할 경우 해당 축 그룹에 속하는 논리축 중 하나의 한계 입력이 'ON'이면 명령을 기동할 수 없습니다.

MC_MoveCircular2D

2축의 원호 보간을 실행합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_MoveCircular2D	2축 원호 보간	FB	MC_MoveCircular2D_instance MC_MoveCircular2D AxesGroup — — AxesGroup Execute — Done CircAxes — Busy CircMode — Active AuxPoint — CommandAborted EndPoint — Error PathChoice — ErrorID Velocity Acceleration Deceleration Jerk CoordSystem BufferMode TransitionMode MoveMode	MC_MoveCircular2D_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, CircAxes := 《파라메터》, CircMode := 《파라메터》, AuxPoint := 《파라메터》, EndPoint := 《파라메터》, PathChoice := 《파라메터》, Velocity := 《파라메터》, Acceleration := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, CoordSystem := 《파라메터》, BufferMode := 《파라메터》, TransitionMode := 《파라메터》, MoveMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
CircAxes	원호축 지정	ARRAY [0,1] OF UINT	0~3	0	원호 보간을 실행할 축을 지정합니다. 0 : 축 A0 1 : 축 A1 2 : 축 A2 3 : 축 A3
CircMode	원호 보간 모드	_eMC_CIRC_MODE	0: _mcBorder 1: _mcCenter 2: _mcRadius	0 *1	원호 보간 방법을 지정합니다. 0 : 통과점 지정 1 : 중심점 지정 2 : 반지름 지정
AuxPoint	보조점	ARRAY [0,1] OF LREAL	음수, 양수, '0'	0	통과점 위치/중심 위치/반지름을 지정합니다. 단위는 [명령 단위]입니다. *2
EndPoint	종료점	ARRAY [0,1] OF LREAL	음수, 양수, '0'	0	목표 위치를 지정합니다. 단위는 [명령 단위]입니다. *2

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
PathChoice	경로 선택	_eMC_CIRC_ PATHCHOICE	0: _mcCW 1: _mcCCW	0 *1	경로 방향을 지정합니다. 0 : CW 1 : CCW
Velocity *3	목표 속도	LREAL	양수	0	목표 속도를 지정합니다. 단위는 [명령 단위/s]입니다. *2
Acceleration	가속도	LREAL	양수 또는 '0'	0	가속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *2
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *2
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *2
Coord System	좌표 계통	_eMC_ COORD_ SYSTEM	0: _mcACS	0 *1	좌표 계통을 지정합니다. 0 : 축 좌표 계통(ACS)
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *1	모션 명령 다중 기동 시의 동작을 지정합니다. 0 : Aborting 1 : Buffered 2 : BlendingLow 3 : BlendingPrevious 4 : BlendingNext 5 : BlendingHigh
Transition Mode	트랜지션 모드 (이동 모드)	_eMC_ TRANSITION_ MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *1	동작 경로를 지정합니다. 0 : 트랜지션 무효 10 : ConnerSuper impose
MoveMode	이동 방법 선택	_eMC_MOVE_ MODE	0: _mcAbsolute 1: _mcRelative	0 *1	이동 방법을 선택합니다. 0 : 절대값 위치 결정 1 : 상대값 위치 결정

*1. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.
*2. 명령 단위에 대해서는, 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.
*3. 목표 속도를 반드시 설정하십시오. 설정하지 않고 동작하게 하면 이상이 발생합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	위치 결정이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	축이 이동을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때 • MC_GroupStop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- 2차원 평면에서 2축의 원호 보간을 실행합니다.

사용상의 주의 사항

- 축 그룹의 구성 축 중 하나가 원점 미확정인 경우는 '원점 미확정 상태에서 명령 기동 이상(에러 코드: 5466 Hex)'이 됩니다.
- 2축 원호 보간을 실행할 경우 축 그룹에 속한 논리축 중 하나의 한계 입력이 'ON'이면 명령을 기동할 수 없습니다.

명령에 대한 자세한 내용

이 명령에 대해 자세히 설명합니다.

● 원호 보간의 순서

원호 보간은 다음 순서로 실행합니다.

1 보간을 실행할 축 그룹 등록

- 보간을 실행할 축 그룹을 결정합니다.
  축 그룹이란 _MC_GRP***입니다.
- 이 축 그룹 변수의 축 구성에서 축 구성을 지정합니다.
- 이 축 그룹 변수의 축 선택에서 보간을 실행할 축의 조합을 지정합니다.
- 축은 축 번호(축 0～축 63)가 아니라 논리축(축 A0～축 A3)을 사용합니다.
- 축 선택에서 논리축 A0～A3에 각각 축 번호 0～63을 앞 채우기로 지정합니다.

논리축	축 번호
축 A0	축 0~축 63
축 A1	축 0~축 63
축 A2	축 0~축 63
축 A3	축 0~축 63

예: 축 구성이 2축이고 축 번호 0~1을 축 A0～A1에 지정하려면 다음과 같이 지정합니다.

논리축	축 번호	설명
축 A0	축 0	축 A0부터 축 번호가 작은 순으로 지정합니다.
축 A1	축 1	

사용상의 주의 사항

축 그룹에 엔코더 축 또는 가상 엔코더 축이 포함되어 있으면 '축 종류 부적합(에러 코드: 543D Hex)'이 되어 동작이 종료됩니다.
반드시 서보 축 또는 가상 서보 축을 선택하십시오.

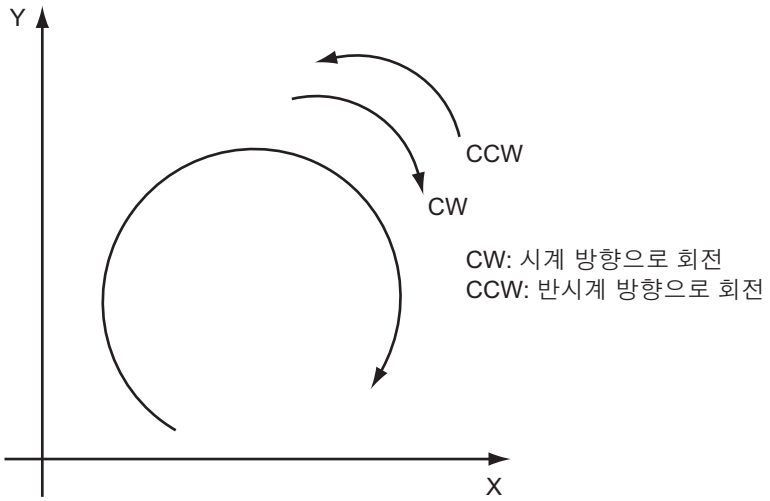
2 축 그룹의 유효화

- 축 그룹의 각 구성축을 서보 ON 상태로 하고, 그리고 이들을 원점 확정 상태로 합니다.
- MC_GroupEnable(축 그룹 유효) 명령을 실행하여 등록한 축 그룹을 유효로 합니다.

이제 원호 보간 명령을 사용할 수 있게 됩니다.

● CircAxes(원호 축 지정)

원호 보간으로 사용하는 축은 X축, Y축입니다.



CircAxes(원호 축 지정)에서 X축, Y축에 할당할 축을 지정합니다.
지정할 축은 축 번호(축 0～축 63)가 아니라 논리축(축 A0～축 A3)으로 지정합니다.

사용상의 주의 사항

X축 또는 Y축에 지정할 축의 카운트 모드는 [리니어 모드]로 하십시오.
[로터리 모드]로 기동하면 '카운트 모드 설정에 의한 원호 보간 명령 기동 이상(에러 코드: 544A Hex)'이 발생합니다.

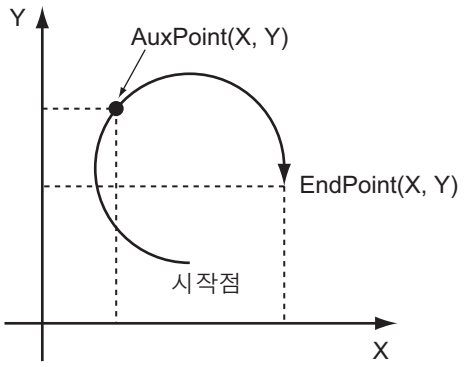
● CircMode(원호 보간 모드)

원호 보간 방식에는 '통과점 지정/중심점 지정/반경 지정'의 3종류가 있으며, CircMode(원호 보간 모드)에서 지정합니다.
이러한 방식으로 지정하는 위치는 '절대값 위치 결정'과 '상대값 위치 결정'이 있으며 MoveMode(이동 방법 선택)에서 지정합니다.

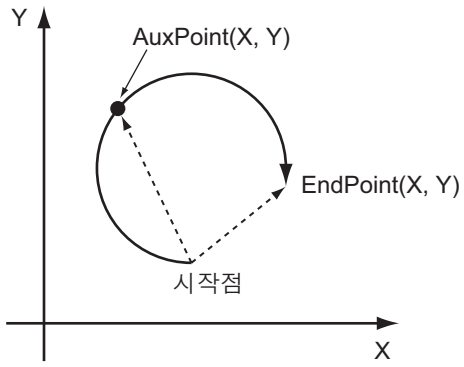
MoveMode	설명
절대값 위치 결정	통과점 지정에서 지정하는 통과점 위치, 중심점 지정에서 지정하는 중심점 위치와 종료점은 축 좌표 계통 원점에서의 절대 위치로 지정합니다.
상대값 위치 결정	통과점 지정에서 지정하는 통과점 위치, 중심점 지정에서 지정하는 중심점 위치와 종료점은 시작점에서의 상대 위치로 지정합니다.

통과점 지정을 예로 [절대값 위치 결정]과 [상대값 위치 결정]의 차이를 나타내면 다음과 같습니다.

원호 방식: 통과점 지정<절대값 위치 결정>



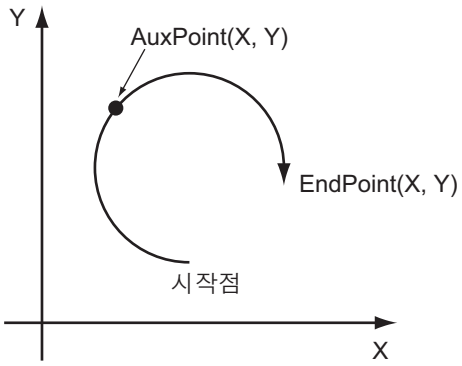
원호 방식: 통과점 지정<상대값 위치 결정>



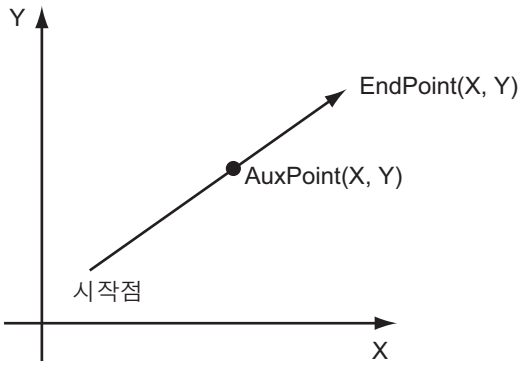
이후의 설명은 MoveMode(이동 방식 선택)에서 '절대값 위치 결정'을 지정한 것을 전제로 합니다.

통과점 지정

현재 위치를 시작점으로 통과점 위치 AuxPoint(X, Y)와 종료점 EndPoint(X, Y)를 잇는 원호 보간을 실행합니다.



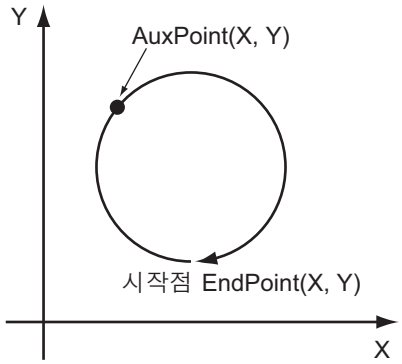
시작점, 통과점 위치와 종료점이 동일 직선상인 경우 또는 통과점 위치와 종료점이 동일점인 경우, 시작점과 통과점 위치가 동일점인 경우는 시작점부터 종료점까지 직선 보간을 실행합니다.



사용상의 주의 사항

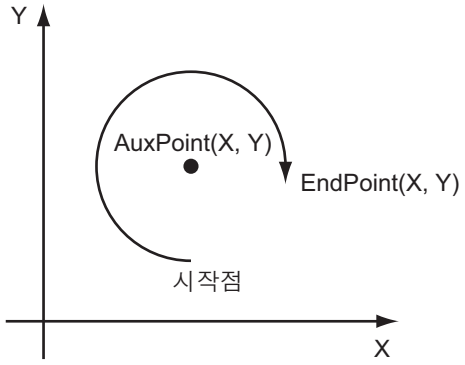
- 동일 직선상에 있다는 것은 시작점과 종료점을 잇는 직선과 통과점 위치의 거리가 X좌표, Y 좌표 모두 1펄스 미만인 경우 동일 직선상에 있다고 판단합니다.
- 시작점, 통과점 위치, 종료점이 동일점인 경우는 이상이 됩니다.
  시작점, 통과점 위치, 종료점이 동일한 점에 있다는 것은 명령 단위의 지정 위치가 동일한 경우를 말합니다.
  명령 단위에서의 명령 위치가 다르고 펄스 환산 위치가 동일한 경우는 동일점으로 간주되지 않으며 이상이 되지 않습니다.

시작점과 종료점이 동일점인 경우 시작점과 통과점을 지름으로 하는 진원을 그립니다. 이 경우 원호의 회전 방향을 PathChoice(경로 선택)에서 지정합니다.



중심점 지정

현재 위치를 시작점으로 중심 위치 AuxPoint(X, Y)에서 지정된 원호에서 종료점 EndPoint(X, Y)를 잇는 원호 보간을 실행합니다.
원호의 회전 방향은 PathChoice(경로 선택)에서 지정합니다.
시작점과 종료점이 동일점이면 진원을 그립니다.
지정한 중심 위치에서 시작점까지의 반지름과 종료점까지의 반지름이 다르면 2개 반지름의 평균값을 사용해서 원호 보간을 실행합니다. 이 경우 반지름 지정과 동일하게 중심 위치를 산출해 해당 반지름과 중심 위치를 사용합니다.



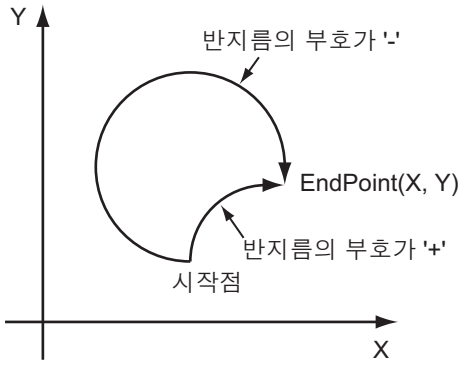
사용상의 주의 사항

축 그룹 파라메터의 [중심점 보정 허용률]에 '0' 이외가 설정된 경우, 지정한 중심점 위치가 다음 식의 원을 초과하는 경우, '원호 보간 중심점 지정 위치 범위 밖(에러 코드: 5449 Hex)'이 발생합니다.

보정한 중심점에서 산출한 반지름을 '100%'로 하여 '반지름×중심점 지정 방식 확인 방법 [%]/100'의 원을 초과하는 경우

반지름 지정

현재 위치를 시작점으로 반지름 AuxPoint(X, Y)에서 지정한 원호에서 종료점 EndPoint(X, Y)를 잇는 원호 보간을 실행합니다.
반지름은 AuxPoint(X, Y)의 최초 요소로 지정합니다. 두 번째 요소는 사용하지 않습니다.
예를 들어 반지름을 100으로 하려면 AuxPoint(X, Y)에 AuxPoint(100, 0)로 지정합니다.
반지름의 부호가 '-'이면 원호 길이가 긴 원호를, 반지름의 부호가 '+'이면 원호 길이가 짧은 원호를 그립니다.



원호의 회전 방향은 PathChoice(경로 선택)에서 지정합니다.

사용상의 주의 사항

- 시작점과 종료점이 동일점이면 원호 보간 시작점 종료점 동일 이상이 발생하고, 이 그룹 전체 축의 동작이 정지합니다.
- 지정한 반경이 시작점과 종료점 거리의 1/2 이하인 경우는 원호를 작성할 수 없으므로 이상이 됩니다.

● Velocity(목표 속도), Acceleration(가속도), Deceleration(감속도), Jerk(저크)

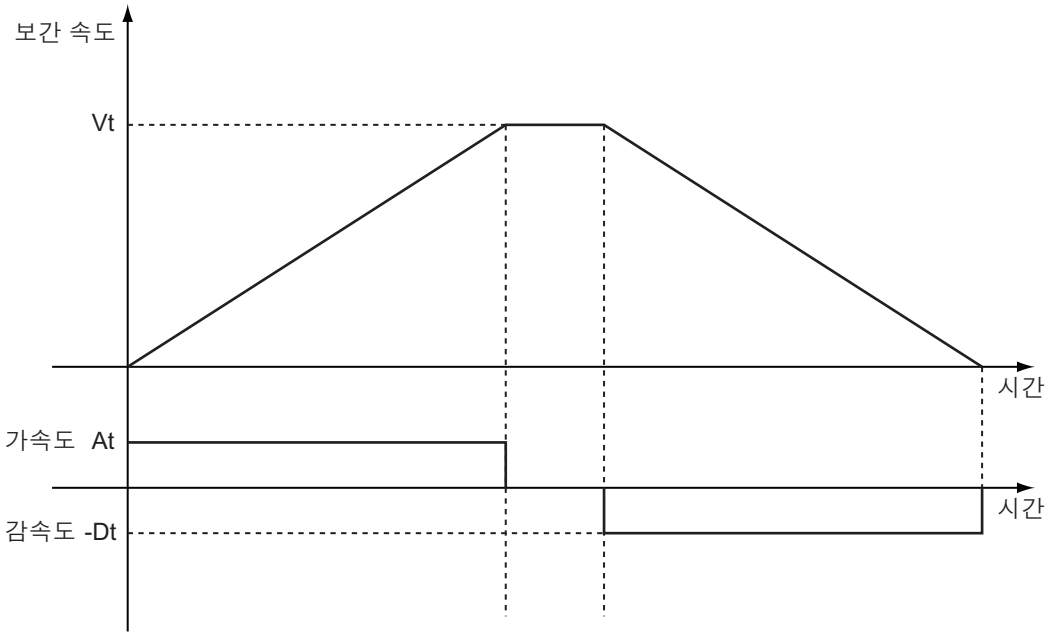
- 원호 보간 시의 보간 속도, 가속도, 감속도, 저크는 Velocity, Acceleration, Deceleration, Jerk에서 지정합니다.
- 원호 보간의 보간 속도에 '0'을 지정하면 Velocity 지정 이상이 발생하고, 이 그룹 전체 축의 동작이 정지합니다.
- 지정한 보간 속도가 축 중 하나의 최고 속도를 초과하면 다음과 같이 동작합니다.

1축만 초과하는 경우	: 이 축의 최고 속도로 동작하도록 보간 속도를 자동 조정합니다.
2축 모두 초과하는 경우	: 2축의 최고 속도 중 늦는 것의 최고 속도로 동작하도록 보간 속도를 자동 조정합니다.

Jerk(저크)

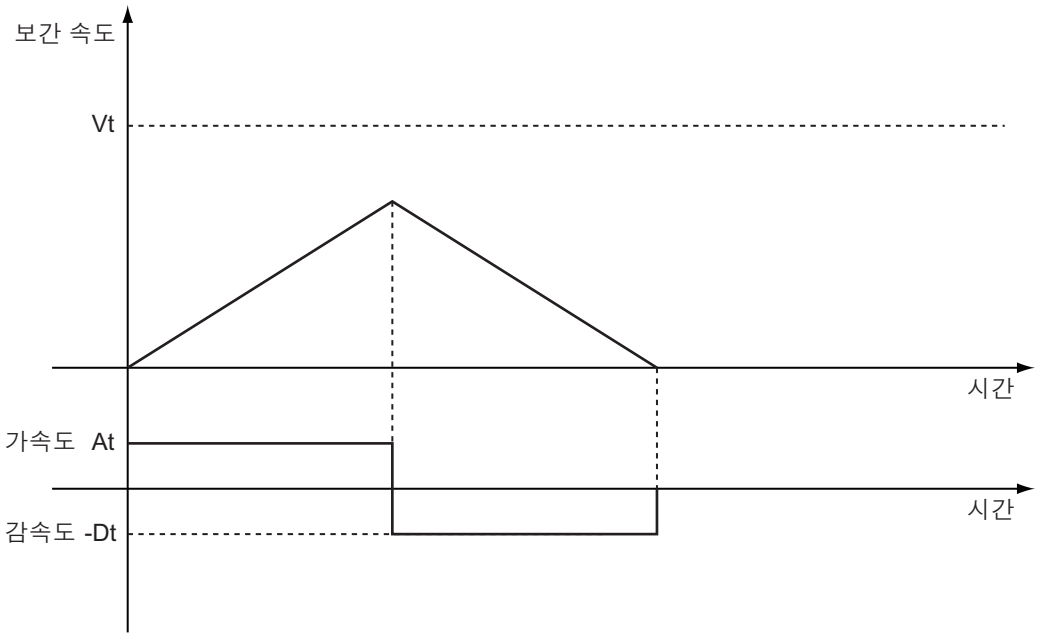
Jerk(저크)에 '0'을 지정한 경우와 '0' 이외의 것을 지정한 경우의 Acceleration(가속도), Deceleration(감속도), Velocity(목표 속도)의 관계는 다음과 같습니다.

- Jerk(저크)가 '0'인 경우
  가속도 At, 감속도 Dt로 속도의 명령값을 작성합니다.



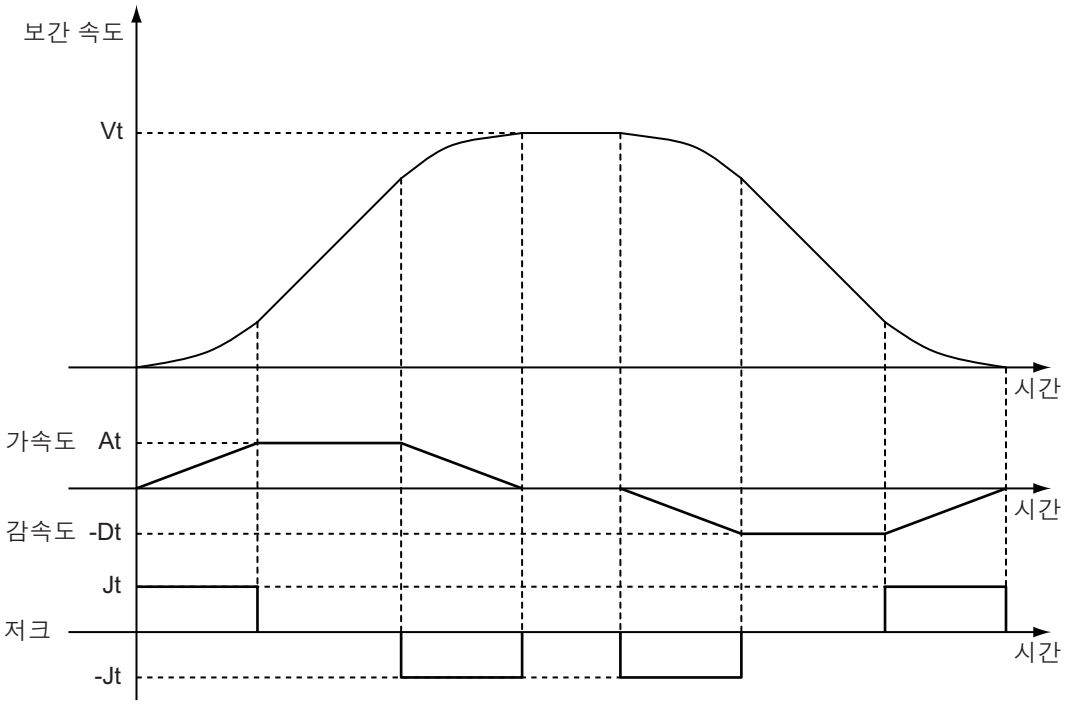
Vt : 보간 속도의 지정값, At : 가속도의 지정값, Dt : 감속도의 지정값

- Jerk(저크)가 '0'이고 이동 거리가 짧은 경우
  보간 속도는 지정값 Vt(목표 속도)까지 도달하지 않습니다.



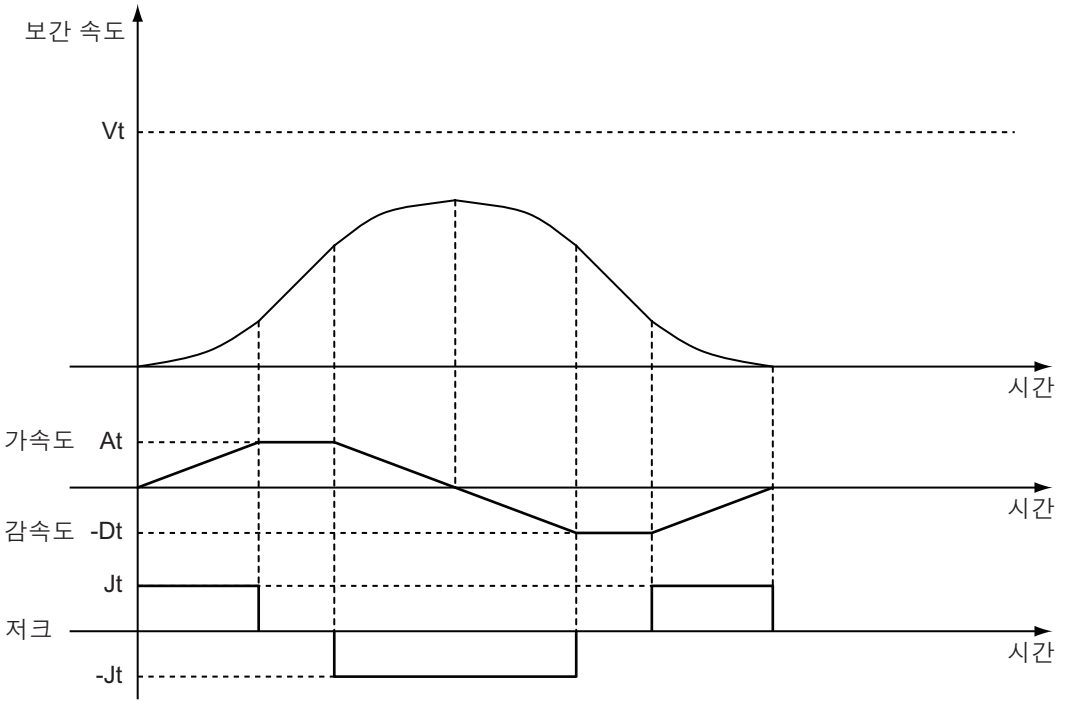
Vt : 보간 속도의 지정값, At : 가속도의 지정값, Dt : 감속도의 지정값

- Jerk(저크)가 '0' 이외인 경우
  가속도의 상한은 At, 감속도의 상한은 Dt로 속도 명령값을 작성합니다.



Vt : 보간 속도의 지정값, At : 가속도의 지정값, Dt : 감속도의 지정값, Jt : 저크의 지정값

- Jerk(저크)가 '0' 이외이고 이동 거리가 짧은 경우
  보간 속도는 지정값 Vt(목표 속도)까지 도달하지 않습니다.



Vt : 보간 속도의 지정값, At : 가속도의 지정값, Dt : 감속도의 지정값, Jt : 저크의 지정값

참고

- Acceleration(가속도)에 '0'을 지정하면 바로 지정한 보간 속도로 움직입니다.
- Deceleration(감속도)에 '0'을 지정하면 바로 정지합니다. 단, 버퍼 모드에서 Blending을 지정한 경우는 정지하지 않고 다음 동작에서 지정한 보간 속도로 바로 이행합니다. 자세한 내용은, 🕮 'BufferMode(버퍼 모드 선택) (4-53페이지)'를 참조하십시오.
- Acceleration(가속도) 또는 Deceleration(감속도)에 '0'을 지정하면 Jerk(저크)의 설정은 무효가 됩니다.

● CoordSystem(좌표 계통)

- 원호 보간을 실행할 좌표 계통을 지정합니다.
- 2축 이상으로 구성되는 축 좌표 계통(ACS)에만 대응합니다.

● BufferMode(버퍼 모드 선택)

- 하나 이전의 보간 동작과 이번 보간 동작의 연결 방법을 지정합니다.
- 종류는 다음 6가지입니다.

버퍼 모드 선택		설 명
Aborting		현재 실행 중인 명령을 즉시 중지하고 이 명령으로 전환합니다. 명령이 전환되어 축 동작 방향이 반전되는 경우는 축 파라메터의 [반전 시 동작]에 따라 반전 동작합니다.
Buffered		현재 실행 중인 명령이 정상 완료한 뒤 버퍼링되고 있는 이 명령이 자동으로 기동합니다.
Blending		현재 실행 중인 명령이 목표 위치에 도달했을 때의 속도(중계 속도)를 기동 속도로 버퍼링 중인 이 명령을 연속해서 동작하게 합니다. 중계 속도로 목표 위치에 도달하도록 현재 실행 중인 명령의 동작을 변경합니다. 중계 속도 지정 방법은 다음의 4종류가 있습니다. 또한 Blending의 옵션으로 트랜지션 지정(뒤에 설명)이 있습니다.
	BlendingLow	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 낮은 것을 중계 속도로 합니다.
	BlendingPrevious	현재 실행 중인 목표 속도를 중계 속도로 합니다.
	BlendingNext	버퍼링되고 있는 이 명령의 목표 속도를 중계 속도로 합니다.
	BlendingHigh	현재 실행 중인 목표 속도와 버퍼링되고 있는 목표 속도 중 높은 것을 중계 속도로 합니다.

BufferMode(버퍼 모드 선택)에 대해서는, 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

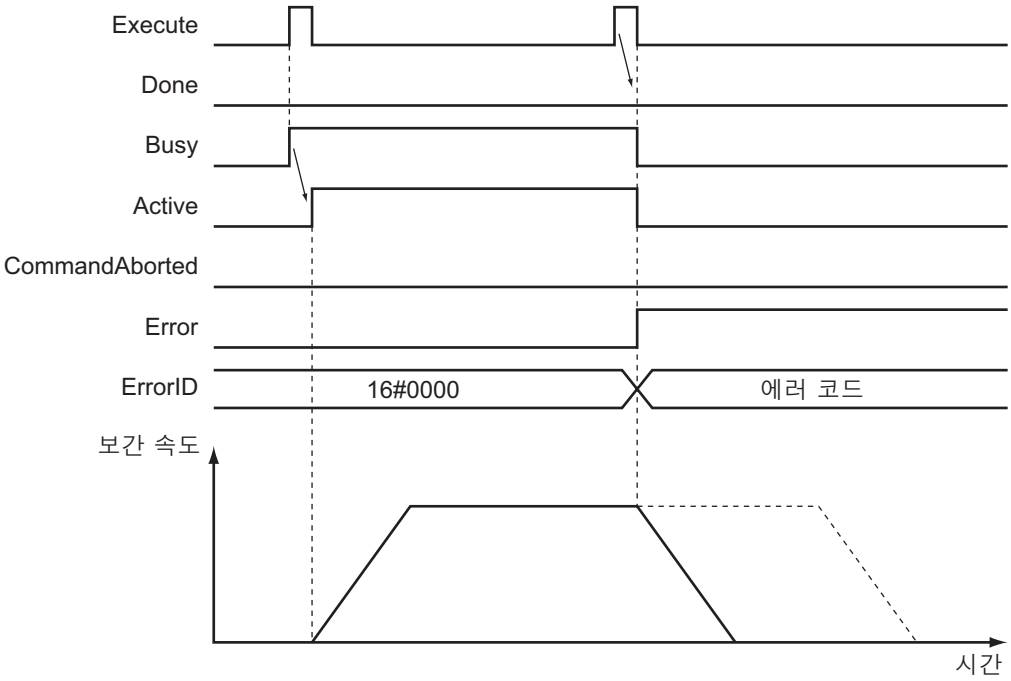
● TransitionMode(트랜지션 모드)

- 하나 이전의 보간 동작과 이번 보간 동작으로 만들어지는 궤적의 연결 방법을 지정합니다.
- BufferMode(버퍼 모드 선택)에서 Blending을 지정하면 TransitionMode(트랜지션 모드)가 유효가 됩니다.
- Blending을 지정하지 않고 TransitionMode(트랜지션 모드)를 _mcTMNone(트랜지션 무효) 이외의 것으로 지정하면 이상이 됩니다.

자세한 내용은 🕮 'TransitionMode(트랜지션 모드) (4-20페이지)'를 참조하십시오.

모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 되고 원호 보간 중작 중인 전체 축이 정지합니다.



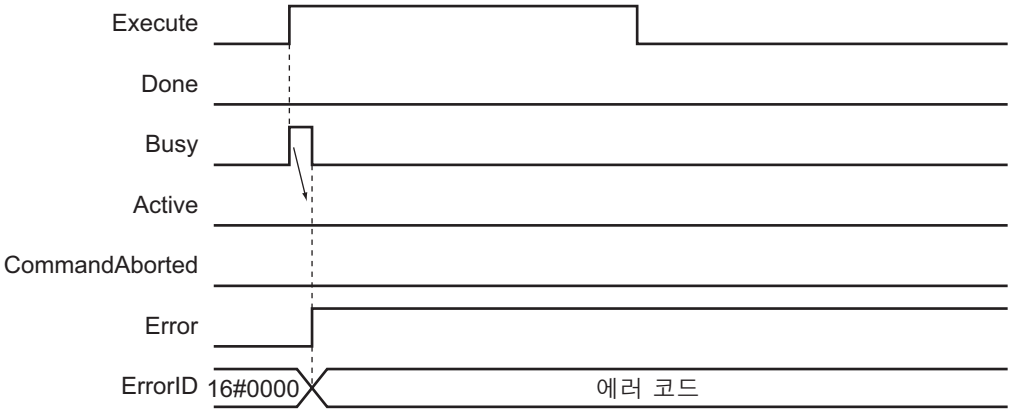
모션 명령 다중 기동

이 명령 실행 중에 사용할 수 있는 명령에는 제한이 있습니다.

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

이상

이 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는, 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

샘플 프로그램

여기에는 다중 기동에 의한 원호 보간에 대한 샘플 프로그램이 실려 있습니다.

파라메터 설정

이 샘플 프로그램에서 사용하는 필요 최소한의 설정은 다음과 같습니다.

● 축 파라메터 설정

축 종류

축	축 종류
축 1	서보 축
축 2	서보 축

카운터 모드

축	카운터 모드
축 1	리니어 모드
축 2	리니어 모드

좌표 단위 선택

축	좌표 단위
축 1	mm
축 2	mm

● 축 그룹 파라메터 설정

축 구성

[2축]으로 합니다.

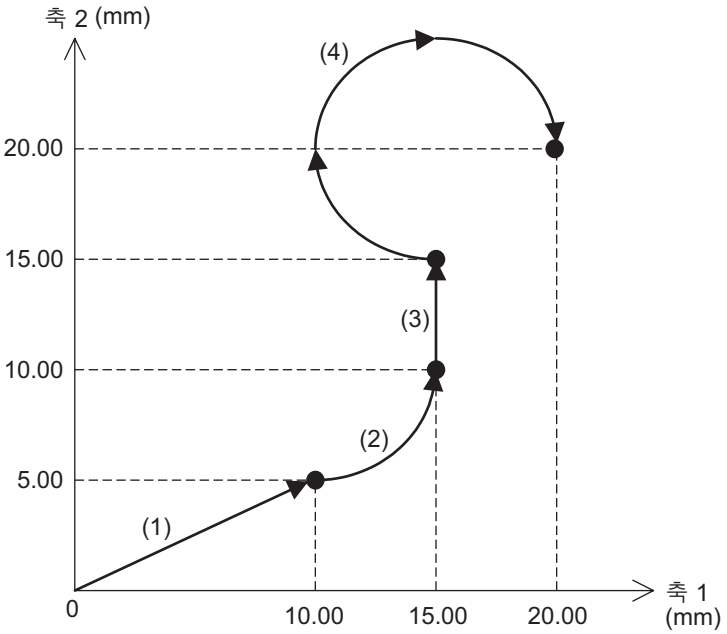
축 선택

[축 1], [축 2]로 합니다.

동작 예

직선 보간 동작에서 원호 보간을 자동으로 위치 결정하는 동작 예는 다음과 같습니다.
직선 보간과 원호 보간을 이용해 최종 목표 위치(20.00mm, 20.00mm)로 이동합니다.
버퍼 모드 선택을 Buffered로 지정하고 다중 기동을 통해 명령합니다.
이 예에서는 직선 보간(1)의 출력 Active가 TRUE가 되었을 때 (2)～(4)의 명령 다중 기동을 실행하고 있습니다. 다축 협조 동작에서는 최대 7명령까지 다중 기동이 가능합니다.

● 동작 패턴



1 **기동**

원점에서 동작 시작 스위치를 'ON'하면 직선 보간 동작으로 (10.00mm, 5.00mm)로 이동합니다.

2 **연속 이동**

그대로 원호 보간 동작으로 (15.00mm, 10.00mm)로 이동해 직선 보간 동작으로 (15.00mm, 15.00mm), 또 원호 보간 동작으로 (20.00mm, 20.00mm)로 위치 결정하고 정지합니다.
이 때의 속도는 10.00mm/s입니다.

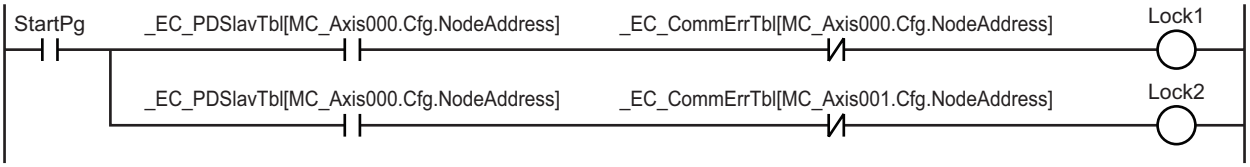
래더도

● 주요 변수

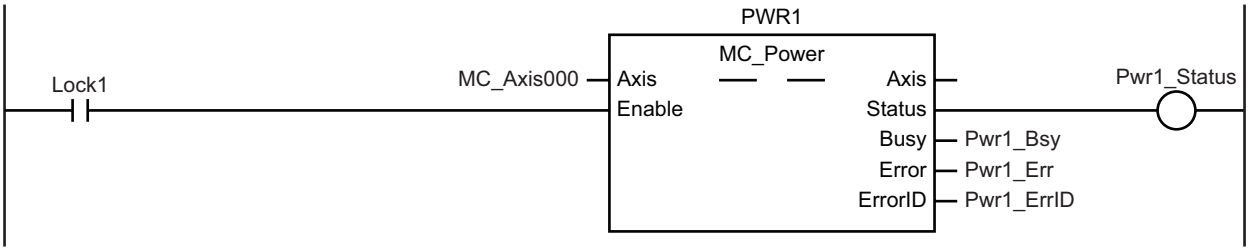
이 름	데이터 형식	초기값	코멘트
MC_Group000	_sGROUP_REF	–	축 그룹 0의 축 그룹 변수입니다.
MC_Group000.MFaultLvl.Active	BOOL	FALSE	축 그룹 0에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Group000.Status.Disabled	BOOL	FALSE	축 그룹 0이 무효인 경우에 TRUE가 됩니다.
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
Pwr1_Status	BOOL	FALSE	MC_Power의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power의 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 축 그룹에 속하는 각 축을 서보 ON 상태로 합니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.

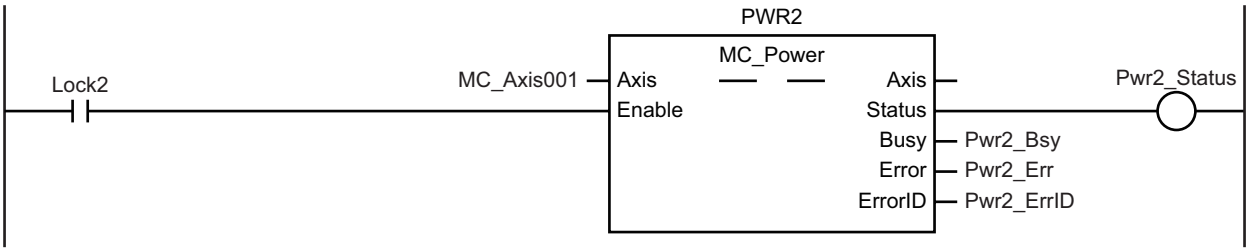
● 샘플 프로그램

접점 StartPg가 TRUE가 되면 각 축의 EtherCAT 통신이 정상적으로 프로세스 데이터 통신을 하고 있는지 확인

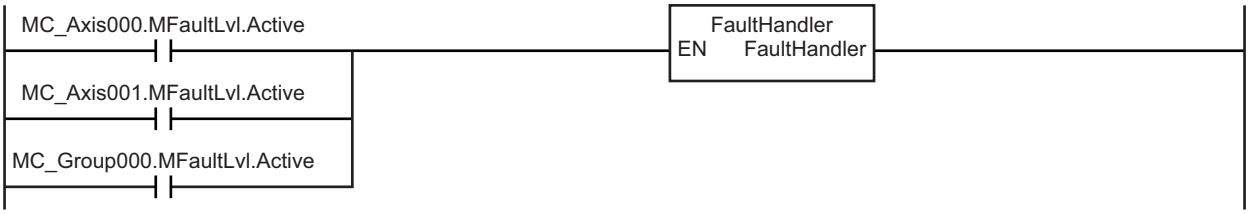


프로세스 데이터 통신을 하고 있으면 서보를 ON

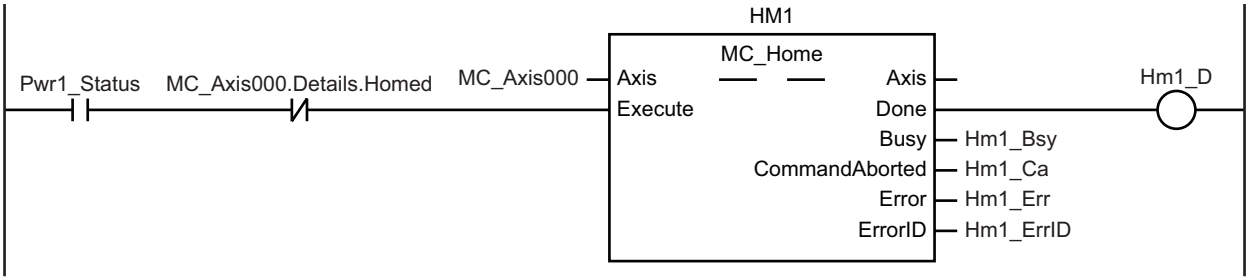




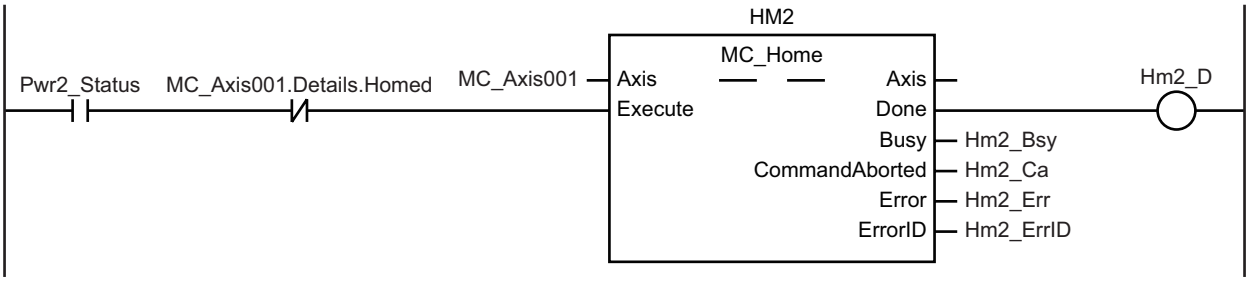
축 구성에서 가벼운 수준의 이상이 발생하면 장치에서 이상 발생 시의 처리(FaultHandler)를 실행.
이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.



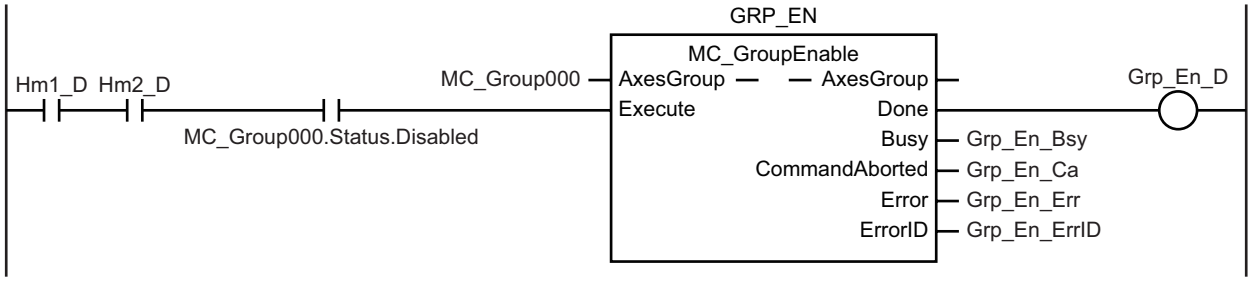
축 1이 서보가 ON되고 원점 미확정 상태라면 원점 복귀를 실행



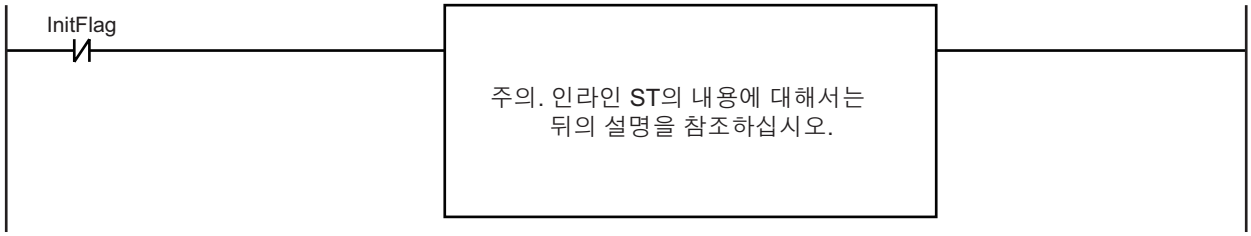
축 2가 서보가 ON되고 원점 미확정 상태라면 원점 복귀를 실행



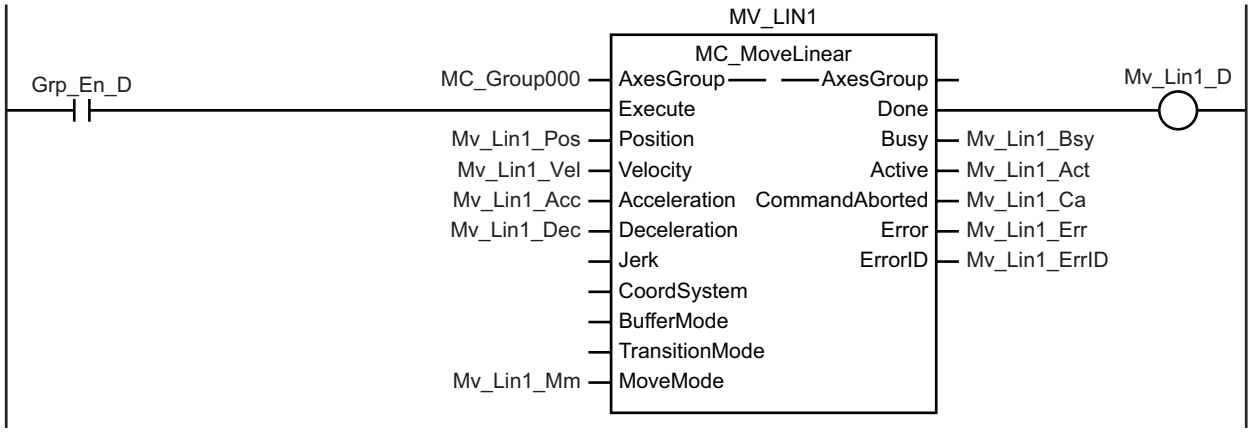
축 1과 축 2의 원점 확정 후 축 그룹을 유효로 함



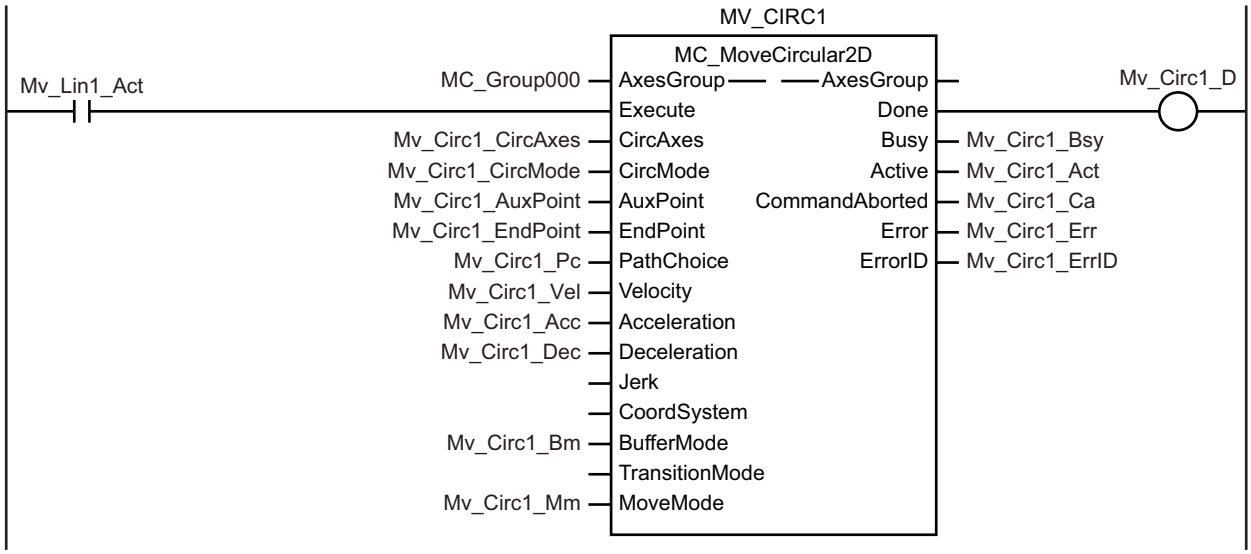
직선 보간과 원호 보간의 파라메터를 설정



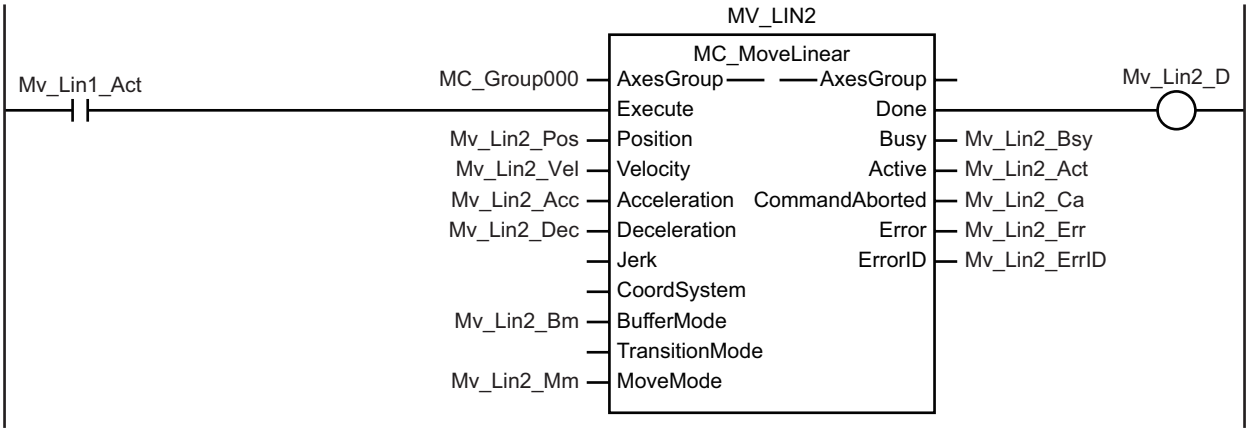
축 그룹이 유효가 된 것을 확인하고 직선 보간(1)을 실행



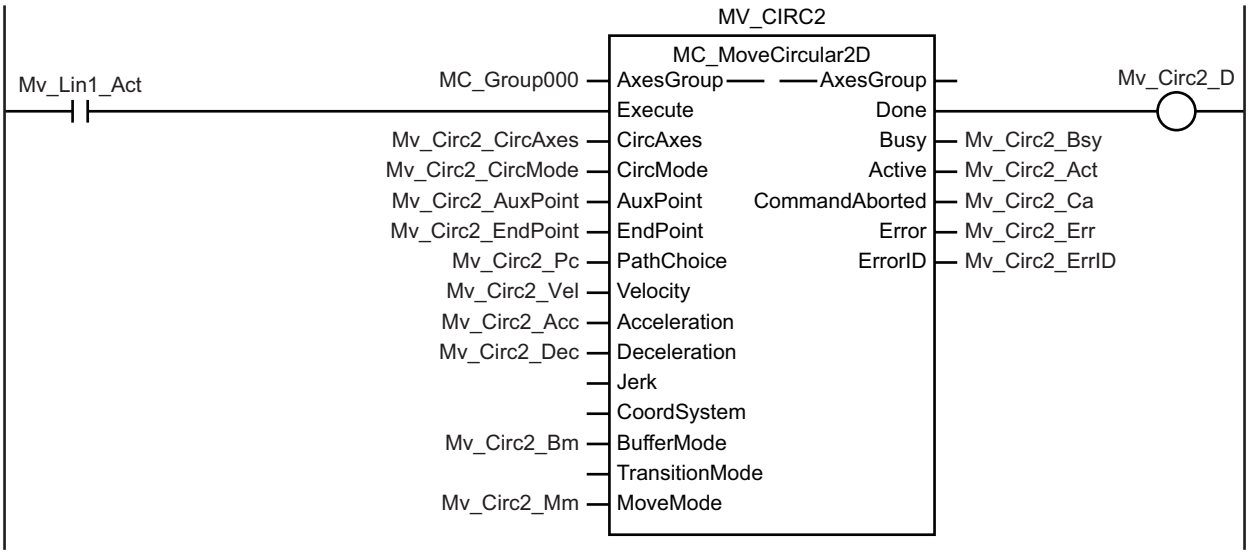
직선 보간(1)이 기동 후 원호 보간(2)를 다중 기동



직선 보간(1)이 기동 후 원호 보간(3)을 다중 기동



직선 보간(1)이 기동 후 원호 보간(4)를 다중 기동



인라인 ST의 내용

```
//MV_CIRC1 파라메터
    Mv_Circ1_CircAxes[0]    := UINT#0;
    Mv_Circ1_CircAxes[1]    := UINT#1;
    Mv_Circ1_CircMode       := _eMC_CIRC_MODE#_mcRadius;
    Mv_Circ1_AuxPoint[0]    := LREAL#5.0;
    Mv_Circ1_AuxPoint[1]    := LREAL#0.0;
    Mv_Circ1_EndPoint[0]    := LREAL#15.0;
    Mv_Circ1_EndPoint[1]    := LREAL#10.0;
    Mv_Circ1_Pc             := _eMC_CIRC_PATHCHOICE#_mcCCW;
    Mv_Circ1_Vel            := LREAL#100.0;
    Mv_Circ1_Acc            := LREAL#20.0;
    Mv_Circ1_Dec            := LREAL#20.0;
    Mv_Circ1_Bm             := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Circ1_Mm             := _eMC_MOVE_MODE#_mcAbsolute;

//MV_CIRC2 파라메터
    Mv_Circ2_CircAxes[0]    := UINT#0;
    Mv_Circ2_CircAxes[1]    := UINT#1;
    Mv_Circ2_CircMode       := _eMC_CIRC_MODE#_mcCenter;
    Mv_Circ2_AuxPoint[0]    := LREAL#15.0;
    Mv_Circ2_AuxPoint[1]    := LREAL#20.0;
    Mv_Circ2_EndPoint[0]    := LREAL#20.0;
    Mv_Circ2_EndPoint[1]    := LREAL#20.0;
    Mv_Circ2_Pc             := _eMC_CIRC_PATHCHOICE#_mcCW;
    Mv_Circ2_Vel            := LREAL#100.0;
    Mv_Circ2_Acc            := LREAL#20.0;
```

```
    Mv_Circ2_Dec        := LREAL#20.0;
    Mv_Circ2_Bm         := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Circ2_Mm         := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN1 파라메터
    Mv_Lin1_Pos[0]      := LREAL#10.0;
    Mv_Lin1_Pos[1]      := LREAL#5.0;
    Mv_Lin1_Vel         := LREAL#100.0;
    Mv_Lin1_Acc         := LREAL#20.0;
    Mv_Lin1_Dec         := LREAL#20.0;
    Mv_Lin1_Mm          := _eMC_MOVE_MODE#_mcAbsolute;

//MV_LIN2 파라메터
    Mv_Lin2_Pos[0]      := LREAL#15.0;
    Mv_Lin2_Pos[1]      := LREAL#15.0;
    Mv_Lin2_Vel         := LREAL#100.0;
    Mv_Lin2_Acc         := LREAL#20.0;
    Mv_Lin2_Dec         := LREAL#20.0;
    Mv_Lin2_Bm          := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin2_Mm          := _eMC_MOVE_MODE#_mcAbsolute;

//입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag :=TRUE;
```

스트럭쳐 텍스트(ST)

● 주요 변수

이 름	데이터 형식	초기값	코멘트
MC_Group000	_sGROUP_REF	–	축 그룹 0의 축 그룹 변수입니다.
MC_Group000.MFaultLvl.Active	BOOL	FALSE	축 그룹 0에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Group000.Status.Disabled	BOOL	FALSE	축 그룹 0이 무효인 경우에 TRUE가 됩니다.
MC_Axis000	_sAXIS_REF	–	축 1의 축 변수입니다.
MC_Axis000.Details.Homed	BOOL	FALSE	축 1이 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	축 1에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
MC_Axis001	_sAXIS_REF	–	축 2의 축 변수입니다.
MC_Axis001.Details.Homed	BOOL	FALSE	축 2가 원점 확정 상태일 때 TRUE가 됩니다.
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	축 2에서 가벼운 수준의 이상이 발생하면 TRUE가 됩니다.
Pwr1_Status	BOOL	FALSE	MC_Power의 인스턴스 PWR1의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 TRUE가 됩니다.
Pwr2_Status	BOOL	FALSE	MC_Power의 인스턴스 PWR2의 출력 Status에 할당한 변수입니다. 서보 ON 상태가 되면 TRUE가 됩니다.
StartPg	BOOL	FALSE	이 변수가 TRUE이고 EtherCAT의 프로세스 데이터 통신이 확립된 상태이면 축 그룹에 속하는 각 축을 서보 ON 상태로 합니다.
InitFlag	BOOL	FALSE	입력 파라메터의 설정이 완료되었음을 나타냅니다. FALSE일 때 입력 파라메터를 설정합니다. 입력 파라메터의 설정이 완료되면 TRUE가 됩니다.
Hm1_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_Home의 인스턴스 HM1을 기동합니다.
Hm2_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_Home의 인스턴스 HM2를 기동합니다.
Grp_En_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_GroupEnable의 인스턴스 GRP_EN을 기동합니다.
Mv_Lin1_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveLinear의 인스턴스 MV_LIN1을 기동합니다.
Mv_Lin2_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveLinear의 인스턴스 MV_LIN2를 기동합니다.
Mv_Circ1_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveCircular의 인스턴스 MV_CIRC1을 기동합니다.
Mv_Circ2_Ex	BOOL	FALSE	이 변수가 FALSE→TRUE일 때 MC_MoveCircular의 인스턴스 MV_CIRC2를 기동합니다.

● 샘플 프로그램

```
//입력 파라메터가 설정되지 않은 경우의 처리
IF InitFlag=FALSE THEN

    //MV_CIRC1 파라메터
    Mv_Circ1_CircAxes[0]     := UINT#0;
    Mv_Circ1_CircAxes[1]     := UINT#1;
    Mv_Circ1_CircMode        := _eMC_CIRC_MODE#_mcRadius;
    Mv_Circ1_AuxPoint[0]     := LREAL#5.0;
    Mv_Circ1_AuxPoint[1]     := LREAL#0.0;
    Mv_Circ1_EndPoint[0]     := LREAL#15.0;
    Mv_Circ1_EndPoint[1]     := LREAL#10.0;
    Mv_Circ1_Pc              := _eMC_CIRC_PATHCHOICE#_mcCCW;
    Mv_Circ1_Vel             := LREAL#100.0;
    Mv_Circ1_Acc             := LREAL#20.0;
    Mv_Circ1_Dec             := LREAL#20.0;
    Mv_Circ1_Bm              := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Circ1_Mm              := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_CIRC2 파라메터
    Mv_Circ2_CircAxes[0]     := UINT#0;
    Mv_Circ2_CircAxes[1]     := UINT#1;
    Mv_Circ2_CircMode        := _eMC_CIRC_MODE#_mcCenter;
    Mv_Circ2_AuxPoint[0]     := LREAL#15.0;
    Mv_Circ2_AuxPoint[1]     := LREAL#20.0;
    Mv_Circ2_EndPoint[0]     := LREAL#20.0;
    Mv_Circ2_EndPoint[1]     := LREAL#20.0;
    Mv_Circ2_Pc              := _eMC_CIRC_PATHCHOICE#_mcCW;
    Mv_Circ2_Vel             := LREAL#100.0;
    Mv_Circ2_Acc             := LREAL#20.0;
    Mv_Circ2_Dec             := LREAL#20.0;
    Mv_Circ2_Bm              := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Circ2_Mm              := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN1 파라메터
    Mv_Lin1_Pos[0]       := LREAL#10.0;
    Mv_Lin1_Pos[1]       := LREAL#5.0;
    Mv_Lin1_Vel          := LREAL#100.0;
    Mv_Lin1_Acc          := LREAL#20.0;
    Mv_Lin1_Dec          := LREAL#20.0;
    Mv_Lin1_Mm           := _eMC_MOVE_MODE#_mcAbsolute;

    //MV_LIN2 파라메터
    Mv_Lin2_Pos[0]       := LREAL#15.0;
    Mv_Lin2_Pos[1]       := LREAL#15.0;
    Mv_Lin2_Vel          := LREAL#100.0;
    Mv_Lin2_Acc          := LREAL#20.0;
    Mv_Lin2_Dec          := LREAL#20.0;
    Mv_Lin2_Bm           := _eMC_BUFFER_MODE#_mcBuffered;
    Mv_Lin2_Mm           := _eMC_MOVE_MODE#_mcAbsolute;

    //입력 파라메터 설정 후 InitFlag를 TRUE로 합니다.
    InitFlag := TRUE;

END_IF;


//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 1을 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis000.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis000.Cfg.NodeAddress]=FALSE) THEN
    Pwr1_En:=TRUE;       //축 1을 서보 ON합니다.
```

```
ELSE
    Pwr1_En:=FALSE;         //축 1을 서보 OFF합니다.
END_IF;

//StartPg가 TRUE인 경우 EtherCAT이 정상적으로 통신하고 있는지 확인하고
//축 2를 서보 ON 상태로 합니다.
//EtherCAT이 정상적으로 통신하지 않으면 서보를 OFF합니다.
IF (StartPg=TRUE)
 AND (_EC_PDSlavTbl[MC_Axis001.Cfg.NodeAddress]=TRUE)
 AND (_EC_CommErrTbl[MC_Axis001.Cfg.NodeAddress]=FALSE) THEN
    Pwr2_En:=TRUE;          //축 2를 서보 ON합니다.
ELSE
    Pwr2_En:=FALSE;         //축 2를 서보 OFF합니다.
END_IF;

//가벼운 수준의 이상 발생 시 처리
//이상 발생 시의 처리(FaultHandler)는 장치에 맞춰 직접 프로그래밍하십시오.
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE) OR
(MC_Group000.MFaultLvl.Active=TRUE)THEN
    FaultHandler();
END_IF;

//축 1이 서보 잠금 상태이고 원점 미확정 상태일 때 원점 복귀를 실행합니다.
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//축 2가 서보 잠금 상태이고 원점 미확정 상태일 때 원점 복귀를 실행합니다.
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//축 1과 축 2가 원점 복귀를 완료하고 축 그룹 0이 무효일 때 축 그룹 0을 유효로 합니다.
IF (Hm1_D=TRUE) AND (Hm2_D=TRUE) AND (MC_Group000.Status.Disabled=TRUE) THEN
    Grp_En_Ex:= TRUE;
END_IF;

//축 그룹 0이 유효가 되면 직선 보간(1)을 기동합니다.
IF Grp_En_D=TRUE THEN
    Mv_Lin1_Ex:=TRUE;
END_IF;

//직선 보간(1)의 출력 Active가 TRUE가 되면 이후의 명령을 다중 기동합니다.
IF Mv_Lin1_Act=TRUE THEN
    Mv_Circ1_Ex:=TRUE;
    Mv_Lin2_Ex:=TRUE;
    Mv_Circ2_Ex:=TRUE;
END_IF;


//축 1의 MC_Power
PWR1(
    Axis        := MC_Axis000,
    Enable      := Pwr1_En,
    Status      => Pwr1_Status,
    Busy        => Pwr1_Bsy,
    Error       => Pwr1_Err,
    ErrorID     => Pwr1_ErrID
);

//축 2의 MC_Power
PWR2(
    Axis        := MC_Axis001,
    Enable      := Pwr2_En,
    Status      => Pwr2_Status,
```

```
    Busy        => Pwr2_Bsy,
    Error       => Pwr2_Err,
    ErrorID     => Pwr2_ErrID
);

//축 1의 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 2의 MC_Home
HM2(
    Axis                := MC_Axis001,
    Execute             := Hm2_Ex,
    Done                => Hm2_D,
    Busy                => Hm2_Bsy,
    CommandAborted      => Hm2_Ca,
    Error               => Hm2_Err,
    ErrorID             => Hm2_ErrID
);


//축 그룹 0 유효
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
);

//직선 보간(1)
MV_LIN1(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Lin1_Ex,
    Position            := Mv_Lin1_Pos,
    Velocity            := Mv_Lin1_Vel,
    Acceleration        := Mv_Lin1_Acc,
    Deceleration        := Mv_Lin1_Dec,
    MoveMode            := Mv_Lin1_Mm,
    Done                => Mv_Lin1_D,
    Busy                => Mv_Lin1_Bsy,
    Active              => Mv_Lin1_Act,
    CommandAborted      => Mv_Lin1_Ca,
    Error               => Mv_Lin1_Err,
    ErrorID             => Mv_Lin1_ErrID
);

//원호 보간(2)
MV_CIRC1(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Circ1_Ex,
    CircAxes            := Mv_Circ1_CircAxes,
    CircMode            := Mv_Circ1_CircMode,
    AuxPoint            := Mv_Circ1_AuxPoint,
    EndPoint            := Mv_Circ1_EndPoint,
    PathChoice          := Mv_Circ1_Pc,
```

```
    Velocity            := Mv_Circ1_Vel,
    Acceleration        := Mv_Circ1_Acc,
    Deceleration        := Mv_Circ1_Dec,
    BufferMode          := Mv_Circ1_Bm,
    MoveMode            := Mv_Circ1_Mm,
    Done                => Mv_Circ1_D,
    Busy                => Mv_Circ1_Bsy,
    Active              => Mv_Circ1_Act,
    CommandAborted      => Mv_Circ1_Ca,
    Error               => Mv_Circ1_Err,
    ErrorID             => Mv_Circ1_ErrID
);

//직선 보간(3)
MV_LIN2(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Lin2_Ex,
    Position            := Mv_Lin2_Pos,
    Velocity            := Mv_Lin2_Vel,
    Acceleration        := Mv_Lin2_Acc,
    Deceleration        := Mv_Lin2_Dec,
    BufferMode          := Mv_Lin2_Bm,
    MoveMode            := Mv_Lin2_Mm,
    Done                => Mv_Lin2_D,
    Busy                => Mv_Lin2_Bsy,
    Active              => Mv_Lin2_Act,
    CommandAborted      => Mv_Lin2_Ca,
    Error               => Mv_Lin2_Err,
    ErrorID             => Mv_Lin2_ErrID
);

//원호 보간(4)
MV_CIRC2(
    AxesGroup           := MC_Group000,
    Execute             := Mv_Circ2_Ex,
    CircAxes            := Mv_Circ2_CircAxes,
    CircMode            := Mv_Circ2_CircMode,
    AuxPoint            := Mv_Circ2_AuxPoint,
    EndPoint            := Mv_Circ2_EndPoint,
    PathChoice          := Mv_Circ2_Pc,
    Velocity            := Mv_Circ2_Vel,
    Acceleration        := Mv_Circ2_Acc,
    Deceleration        := Mv_Circ2_Dec,
    BufferMode          := Mv_Circ2_Bm,
    MoveMode            := Mv_Circ2_Mm,
    Done                => Mv_Circ2_D,
    Busy                => Mv_Circ2_Bsy,
    Active              => Mv_Circ2_Act,
    CommandAborted      => Mv_Circ2_Ca,
    Error               => Mv_Circ2_Err,
    ErrorID             => Mv_Circ2_ErrID
);
```

MC_GroupStop

보간 동작 중인 모든 축을 감속 정지합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_GroupStop	축 그룹 강제 정지	FB	MC_GroupStop_instance MC_GroupStop AxesGroup — AxesGroup Execute — Done Deceleration — Busy Jerk — Active BufferMode — CommandAborted Error ErrorID	MC_GroupStop_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, Deceleration := 《파라메터》, Jerk := 《파라메터》, BufferMode := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Deceleration	감속도	LREAL	양수 또는 '0'	0	감속도를 지정합니다. 단위는 [명령 단위/s^2]입니다. *1
Jerk	저크	LREAL	양수 또는 '0'	0	저크를 지정합니다. 단위는 [명령 단위/s^3]입니다. *1
BufferMode	버퍼 모드 선택	_eMC_ BUFFER_ MODE	0: _mcAborting	0 *2	모션 명령 다중 기동 시의 동작을 지정합니다. 0: Aborting

*1. 명령 단위에 대해서는, 📖 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

*2. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동 (Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때 • 이상 발생 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	−	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 MC_Group***) 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- 이 명령은 축 그룹 명령으로 동작 중인 AxesGroup(축 그룹)에서 지정한 축 그룹에 속하는 모든 축을 정지시키고 축 그룹 명령을 무효 상태로 합니다.
  축을 정지할 때 Deceleration(감속도)로 감속 정지하는 명령은 MC_MoveLinear(직선 보간) 명령, MC_MoveLinearAbsolute(절대값 직선 보간) 명령, MC_MoveLinearRelative(상대값 직선 보간) 명령, MC_MoveCircular2D(2축 원호 보간) 명령입니다.
  축을 정지할 때 Deceleration(감속도)의 설정값에 관계 없이 즉시 정지하는 명령은 MC_GroupSyncMoveAbsolute(축 그룹 주기적 동기 위치 제어) 명령입니다.
- 이 명령이 기동하면 보간 동작 중인 명령은 CommandAborted(실행 중단)됩니다.
- 보간 명령을 실행 중인 경우에 이 명령을 실행하면 직선 보간 또는 원호 보간의 궤적 위에 감속 정지합니다.
- Execute(기동)가 기동하면 감속 정지 동작을 시작합니다.

사용상의 주의 사항

축 그룹 변수의 Status.ErrorStop(축 그룹 에러 감속 정지 중)이 TRUE이면 이 명령은 기동할 수 없습니다.
에러 감속 정지 중에 축 그룹 동작을 정지시키려면 MC_GroupImmediateStop(즉시 정지) 명령을 사용하십시오.

명령에 대한 자세한 내용

이 명령에 대해 자세히 설명합니다.

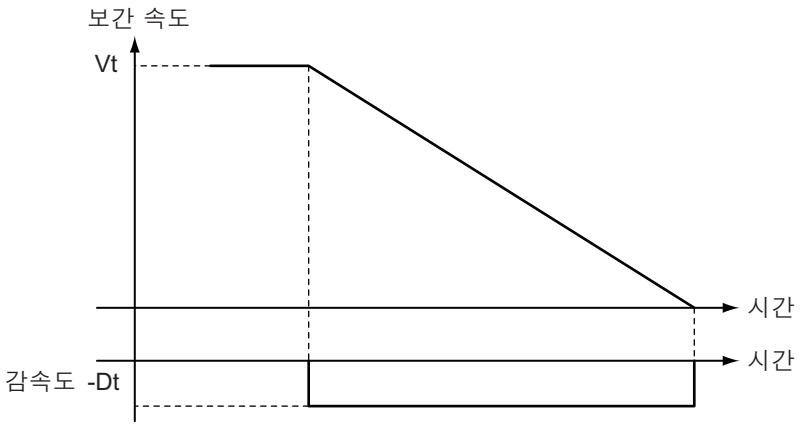
● Deceleration(감속도), Jerk(저크)

감속 정지 시의 감속도, 저크는 입력 변수의 Deceleration(감속도), Jerk(저크)로 지정합니다.
이 명령을 실행하면 보간 속도에 대해 여기에서 지정한 Deceleration(감속도), Jerk(저크)를 사용합니다.

Jerk(저크)

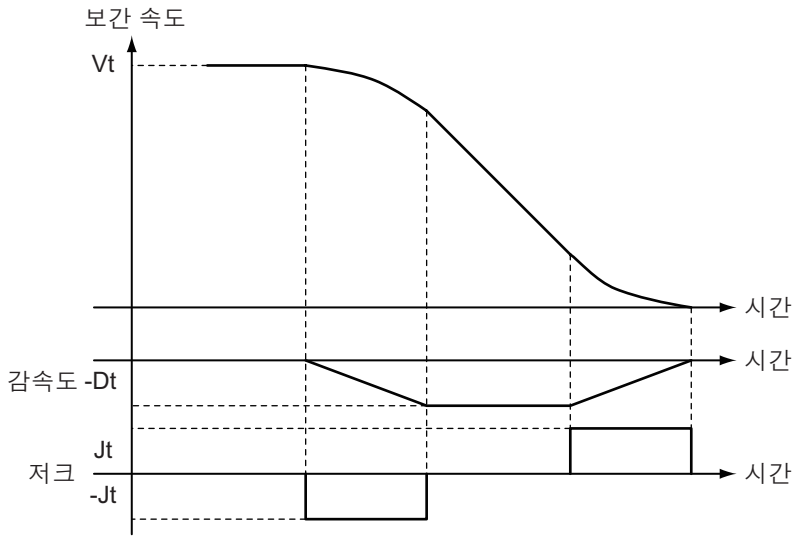
Jerk(저크)에 '0'을 지정한 경우와 '0' 이외의 것을 지정한 경우의 감속도, 보간 속도의 관계는 다음과 같습니다.

- Jerk(저크)가 '0'인 경우
  감속도 Dt로 속도의 명령값을 작성합니다.



Vt: 감속 기동 시의 속도, Dt: 감속도의 지정값

- Jerk(저크)가 '0' 이외인 경우
  감속도의 상한을 Dt로 하고 현재 속도를 기점으로 해서 속도의 명령값을 작성합니다.



Vt: 감속 기동 시의 속도, Dt: 감속도의 지정값, Jt: 저크의 지정값

참고

- 감속도에 '0'을 지정하면 즉시 정지 명령이 되어 바로 정지합니다.
- 감속도에 '0'을 지정한 경우에만 축 파라메터의 [가감속 초과] 설정에 상관 없이 즉시 정지를 실행합니다.

● BufferMode(버퍼 모드 선택)

하나 이전의 보간 동작과 이번 보간 동작의 연결 방법을 지정합니다.
종류는 다음 1가지입니다.

버퍼 모드 선택	설명
Aborting	현재 실행 중인 명령을 즉시 중지하고 이 명령을 실행합니다.

BufferMode(버퍼 모드 선택)에 대해서는, 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

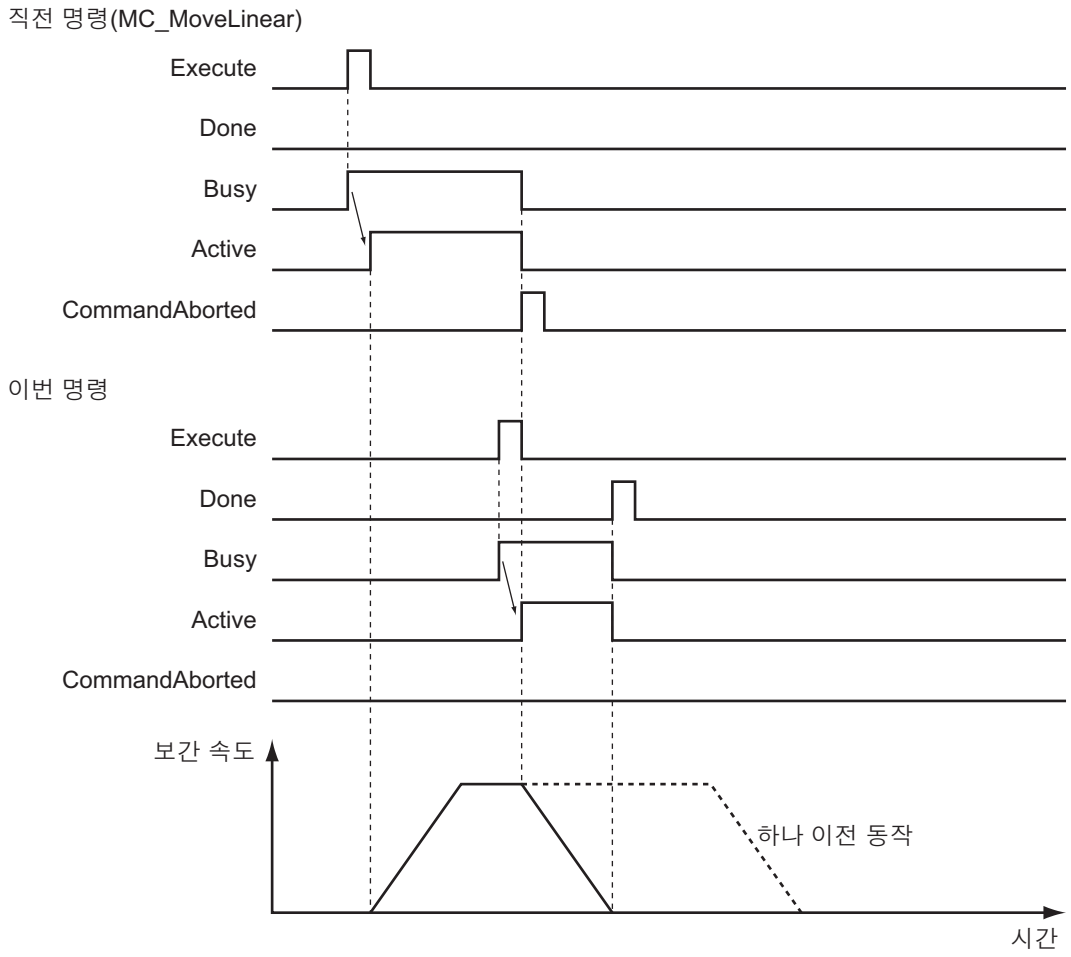
● 인포지션 확인

이 명령으로 정지 시에 인포지션 확인은 실행하지 않습니다.

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- 속도 '0'에 도달하면 Done(완료)이 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중단되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중)가 FALSE가 됩니다.

직선 보간 중인 감속 정지의 타이밍 차트는 다음과 같습니다.
이 명령이 기동하면 보간 동작 중인 명령은 CommandAborted(실행 중단)됩니다.

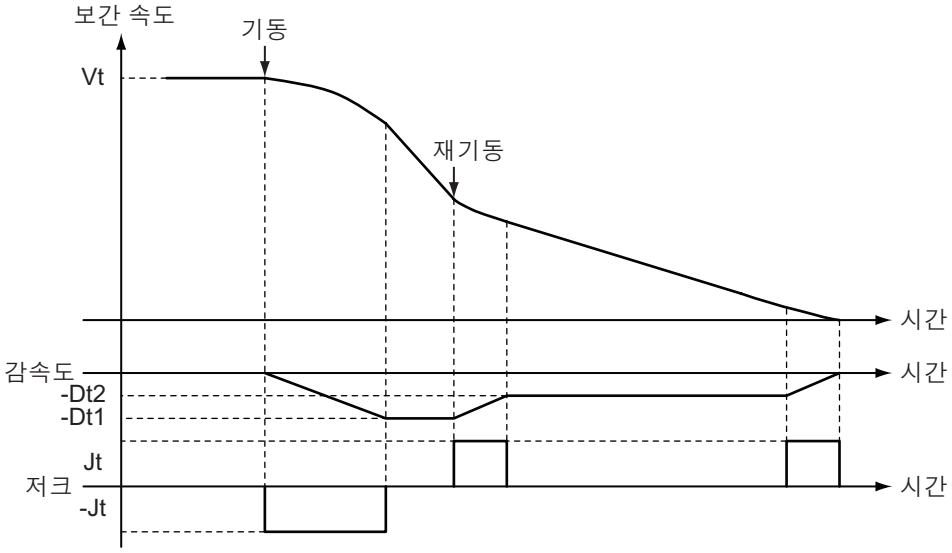


모션 명령 재기동

이 명령 실행 중에 다시 Execute(기동)를 기동하면 감속도를 변경합니다.
모션 명령 재기동으로는 Jerk(저크)를 변경하지 않습니다.

● Jerk(저크)가 '0' 이외인 경우

현재 속도, 현재 감속도를 기점으로 변경 후 감속도의 상한을 Dt2로 속도의 명령값을 작성합니다.



Vt: 감속 기동 시의 속도, Dt: 감속도의 지정값, Jt: 저크의 지정값

모션 명령 다중 기동

이 명령의 실행 및 이 명령 실행 중에 사용할 수 있는 상태와 명령에는 제한이 있습니다.

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

축 그룹 무효 중

무효인 축 그룹에 MC_GroupStop(축 그룹 강제 정지) 명령을 실행하면 축 그룹은 이상이 됩니다. 축에는 영향을 미치지 않습니다.

축 그룹 변수의 Status.Stopping(축 그룹 감속 정지 중)이 TRUE일 때

다음 상태인 경우 축 변수의 Status.Stopping(감속 정지 중)이 TRUE가 됩니다.

- MC_GroupStop(축 그룹 강제 정지) 명령에 의한 축 그룹 감속 중
- MC_GroupStop(축 그룹 강제 정지) 명령의 Execute가 다중 기동에 의해 하나라도 TRUE가 된 동안

감속 정지 중인 상태의 축 그룹에 대해 다음 명령을 기동하면 기동한 명령의 CommandAborted(실행 중단)가 TRUE가 됩니다.

- MC_MoveLinear(직선 보간) 명령
- MC_MoveLinearAbsolute(절대값 직선 보간) 명령
- MC_MoveLinearRelative(상대값 직선 보간) 명령
- MC_MoveCircular2D(2축 원호 보간) 명령

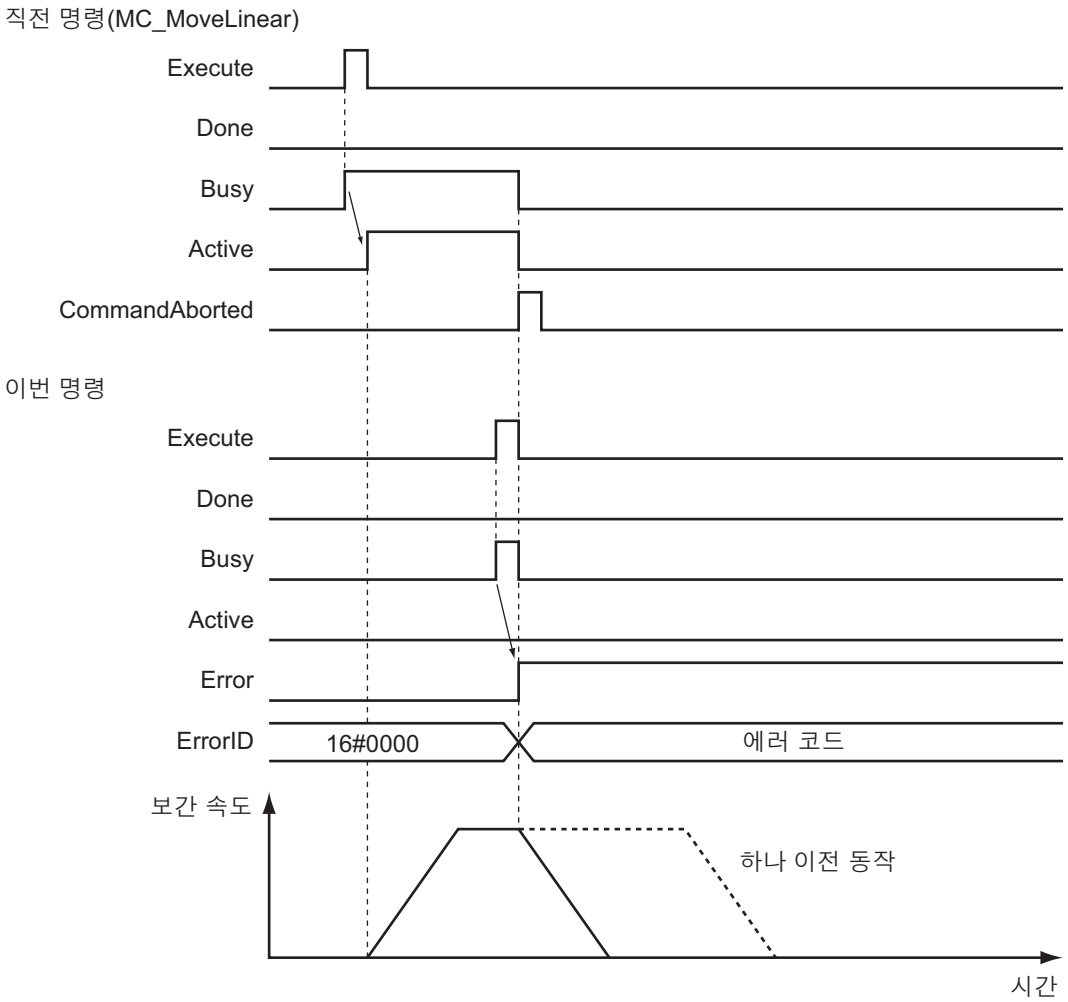
MC_GroupStop(축 그룹 강제 정지) 명령을 실행 중일 때 MC_GroupStop(축 그룹 강제 정지) 명령을 다중 기동할 수 있습니다. 실행 중인 MC_GroupStop(축 그룹 강제 정지) 명령은 Done(완료)이 TRUE가 됩니다.

축 그룹 변수 Status.ErrorStop(축 그룹 에러 감속 정지 중)이 TRUE일 때

축 그룹에서 이상이 발생하는 동안 축 그룹은 Status.ErrorStop(에러 감속 정지 중)이 TRUE가 됩니다.
Status.ErrorStop(에러 감속 정지 중)이 TRUE인 경우에 MC_GroupStop(축 그룹 강제 정지) 명령을 기동하면 CommandAborted(실행 중단)가 TRUE가 됩니다.
이 경우 MC_GroupImmediateStop(축 그룹 즉시 정지) 명령을 사용하십시오.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

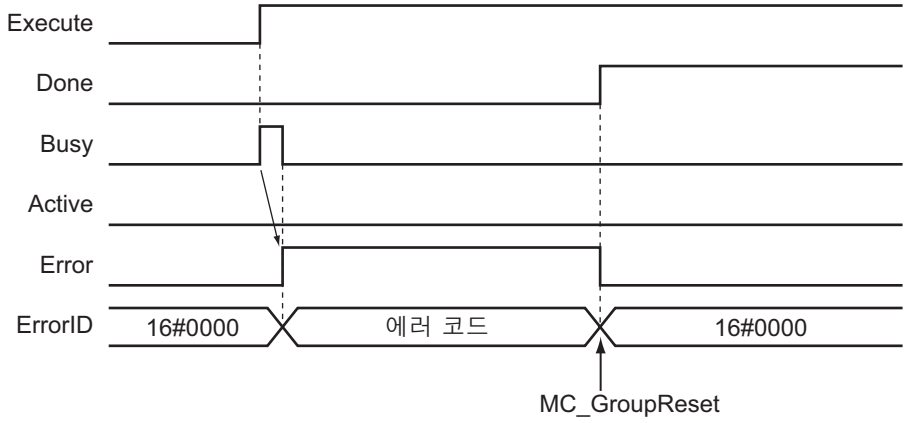


● 이상 해제 시의 동작

이 명령으로 이상이 발생한 상태에서 Execute(기동)가 TRUE인 채로 이상을 해제하면 Error(에러)가 FALSE, Done(완료)이 TRUE가 되고, 감속 정지의 정상 실행 시와 동일하게 축 상태는 Status.Stopping (감속 정지 중)이 TRUE가 됩니다.

이상 해제는 축 그룹을 정지하고 나서 실행하십시오.

축 그룹 동작 중에는 이상을 해제할 수 없습니다.

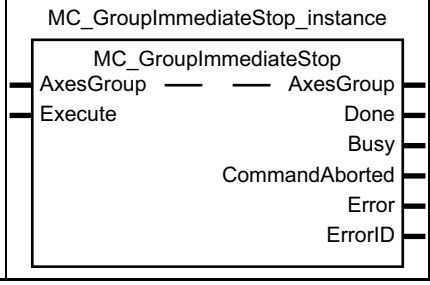


● 에러 코드

이 명령에서 발생하는 이상에 대해서는, 🕮 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_GroupImmediateStop

보간 동작 중인 모든 축이 즉시 정지합니다.
대상 축 그룹이 유효 상태이면 축 상태에 상관 없이 축 파라메터의 [즉시 정지 입력 정지 방법]의 정지 방법에 따라 구성 축의 모든 축이 정지합니다.

명령	명칭	FB/FUN	그래픽 표현	ST 표현
MC_GroupImmediateStop	축 그룹 즉시 정지	FB		MC_GroupImmediateStop_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	즉시 정지를 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 MC_GroupStop(축 그룹 강제 정지) 명령에 의한 모션 명령 다중 기동(Aborting)으로 이 명령이 중지되었을 때 • 이상이 발생해 이 명령이 중지되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

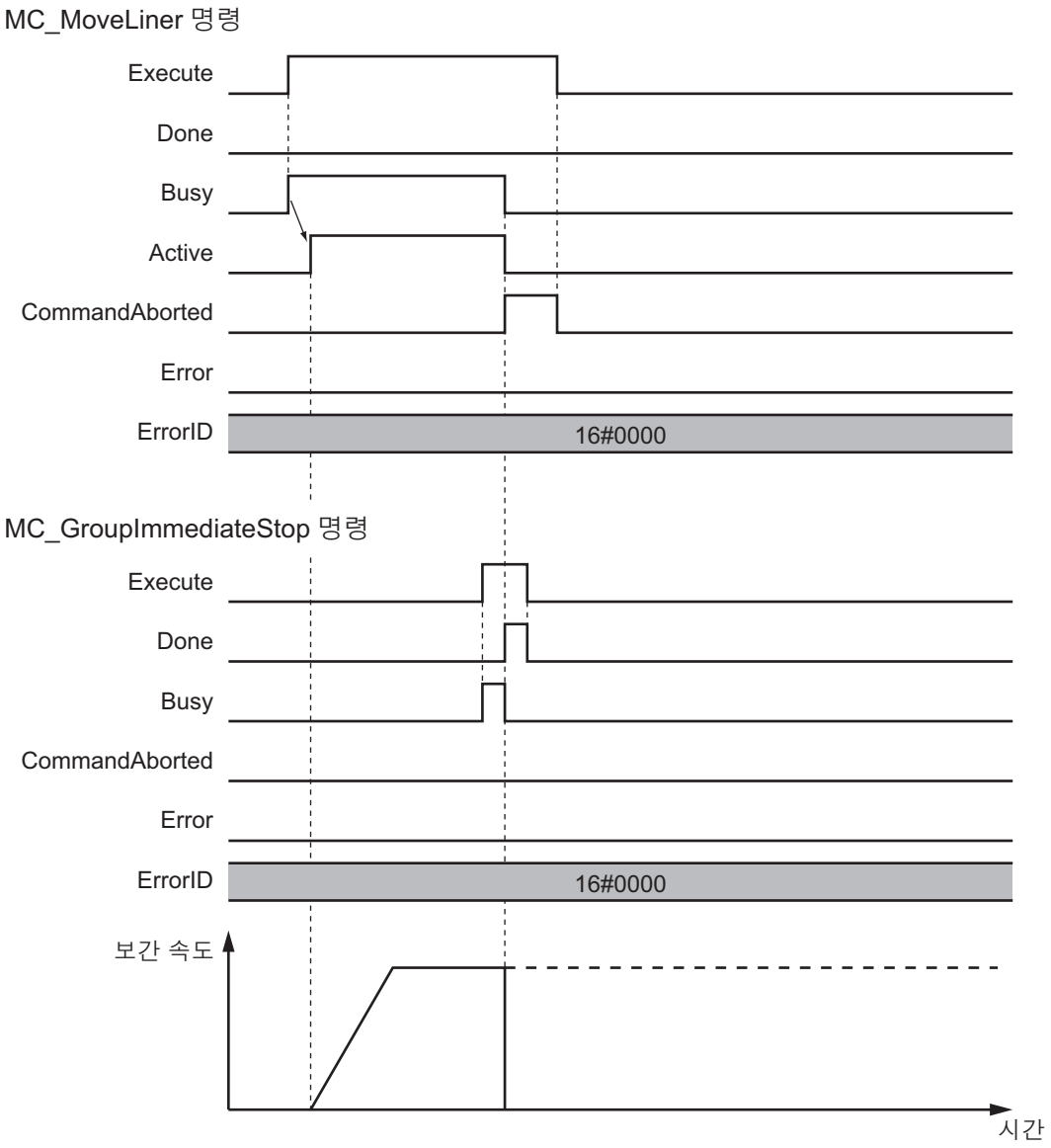
*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- 이 명령은 축 그룹 명령으로 동작 중인 AxesGroup(축 그룹)에서 지정한 축 그룹에 속하는 모든 축 및 MC_GroupStop(축 그룹 강제 정지) 명령 또는 이상 발생으로 인해 정지 상태인 축 그룹에 대해 실행할 수 있습니다.
- 명령이 실행되면 각 축의 파라메터 설정[즉시 정지 입력 정지 방법]에서 지정된 대로 즉시 정지하고, 동작 중인 명령은 CommandAborted(실행 중단)가 됩니다.
- 이 명령을 실행하면 축 그룹 상태는 Status.ErrorStop(에러 감속 정지 중)이 TRUE가 되고 '축 그룹 즉시 정지 명령 기동(에러 코드: 5486 Hex)'이 발생합니다.
  '축 그룹 즉시 정지 명령 기동(에러 코드: 5486 Hex)'에 대해서는, 📖 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 됩니다.
- 이 명령의 처리가 완료되면 Done(완료)이 TRUE가 됩니다.



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 되고 이 축의 동작이 정지합니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

축 그룹 무효 중

무효인 축 그룹에 이 명령을 실행하면 축 그룹은 이상이 됩니다. 축에는 영향을 미치지 않습니다.

축 그룹 변수 Status.Stopping(감속 정지 중)이 TRUE일 때

다음 상태이면 축 그룹 변수 Status.Stopping(감속 정지 중)이 TRUE가 됩니다.

- MC_GroupStop(축 그룹 강제 정지) 명령에 의한 축 그룹 감속 중
- MC_GroupStop(축 그룹 강제 정지) 명령의 Execute가 다중 기동에 의해 하나라도 TRUE가 된 동안

감속 정지 중 상태인 축 그룹에 대해서도 이 명령에 의한 다중 기동이 가능합니다.

이 명령이 기동되면 실행 중인 MC_GroupStop(축 그룹 강제 정지) 명령의 CommandAborted(실행 중단)가 TRUE가 됩니다.

축 그룹 변수 Status.ErrorStop(에러 감속 정지 중)이 TRUE일 때

축 그룹에서 이상이 발생하는 동안 축 그룹의 상태는 Status.ErrorStop(에러 감속 정지 중)이 TRUE가 됩니다.

축 그룹 에러 감속 정지 중이라도 이 명령은 실행 가능합니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.

ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

● 에러 코드

이 명령에서 발생하는 이상에 대해서는, 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_GroupSetOverride

보간 동작 중인 합성 목표 속도를 변경합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_GroupSetOverride	축 그룹 오버라이드값 설정	FB	MC_GroupSetOverride_instance MC_GroupSetOverride AxesGroup — AxesGroup Enable — Enabled VelFactor — Busy AccFactor — Error JerkFactor — ErrorID	MC_GroupSetOverride_instance (AxesGroup := 《파라메터》, Enable := 《파라메터》, VelFactor := 《파라메터》, AccFactor := 《파라메터》, JerkFactor := 《파라메터》, Enabled => 《파라메터》, Busy => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Enable	유효	BOOL	TRUE, FALSE	FALSE	TRUE로 하면 오버라이드가 유효가 됩니다. FALSE로 하면 오버라이드는 '100%'로 되돌아갑니다.
VelFactor	속도 오버라이드값	LREAL	0~500	100	속도 오버라이드값을 지정합니다. 오버라이드값의 유효 범위는 '0.01~500.00'입니다. '500.00 이상'은 '500'으로 다루어지며 '0.01 미만(음수도 포함)'은 '0.01'로 다루어집니다. '0'을 지정한 경우에만 '0'으로 동작합니다. 단위는 [%]입니다.
AccFactor (Reserved)	가감속 속도 오버라이드값	LREAL	0~500	100	(Reserved)
JerkFactor (Reserved)	저크 오버라이드값	LREAL	0~500	100	(Reserved)

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Enabled	유효	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Enabled	이 명령을 실행하기 시작했을 때	• Enable이 FALSE인 경우 1주기 후 • Error가 TRUE가 되었을 때
Busy	Enable의 기동	• Error가 TRUE가 되었을 때
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- 다음 명령으로 움직이는 그룹의 보간 목표 속도에 대한 오버라이드를 변경합니다.
  오버라이드를 변경해 동작 중인 목표 속도를 변경합니다.
  대상이 되는 명령은 다음 명령입니다.

MC_MoveLinear(직선 보간) 명령	MC_MoveLinearRelative(상대값 직선 보간) 명령
MC_MoveLinearAbsolute(절대값 직선 보간) 명령	MC_MoveCircular2D(2축 원호 보간) 명령

- 새로운 목표 속도는 다음과 같습니다.
  변경 후의 목표 속도 = 현재 실행 중인 명령의 보간 속도 × 오버라이드[%]
- 오버라이드의 단위는 [%]입니다. '100'으로 '100%'를 나타냅니다.
- 변경 후의 보간 속도가 축 그룹 변수의 최고 보간 속도를 초과하는 경우는 최고 보간 속도가 됩니다.
- 변경한 보간 속도에 대해 축을 가속 또는 감속합니다.
- 오버라이드에 '0'을 지정하면 목표 속도 '0'이 되고 그룹 동작은 감속 정지하며 속도 '0'에서 동작 중이 됩니다. 동작 중인 축 그룹 변수 Status.Moving(동작 중)이 TRUE가 됩니다.
  동작 상태를 지속하면서 축 동작을 일시 정지하려면 오버라이드를 '0'으로 합니다.

- Enable(유효)이 FALSE이면 오버라이드는 100(=100%)으로 되돌아갑니다.
- MC_GroupSetOverride(축 그룹 오버라이드값 설정) 명령 실행 중에 그룹 이상이 발생하면 MC_GroupSetOverride(축 그룹 오버라이드값 설정) 명령은 Enabled(유효)를 지속합니다.

사용상의 주의 사항

이 명령의 Enable(유효)을 FALSE로 하면 이 명령의 Enabled(유효) 및 Busy(실행 중)는 FALSE가 됩니다.
이 때 속도는 오버라이드 '100%'를 향해서 가속 또는 감속합니다.

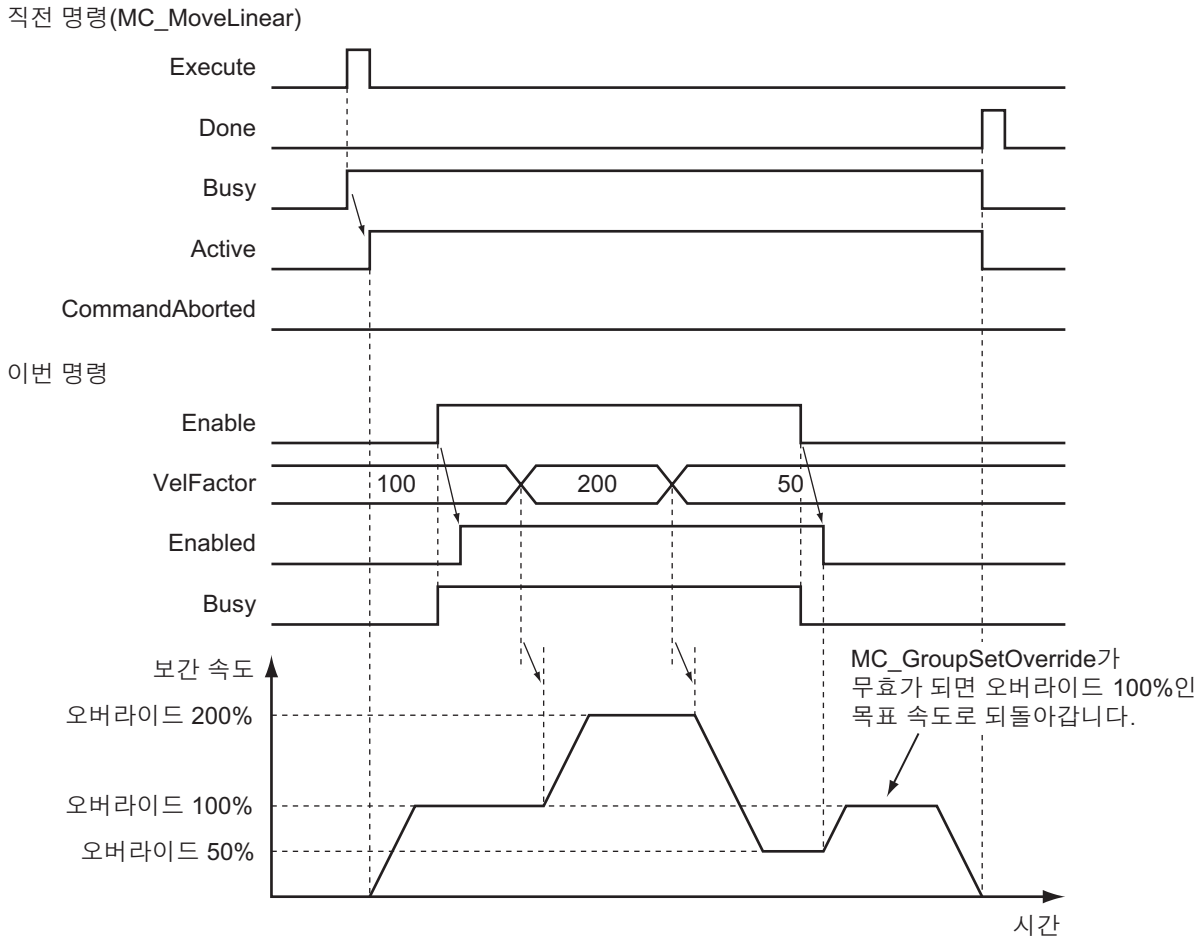
참고

다른 명령에 대한 영향

이 명령은 각 명령의 목표 속도를 일시적으로 변경할 경우에 사용합니다.
따라서 목표 속도가 입력에 없는 명령에 대해서는 이 명령이 기능하지 않습니다.
단, 이 명령이 기능하지 않는 명령에 대해 MC_GroupSetOverride(축 그룹 오버라이드값 설정) 명령을 유효로 한 경우에도 Enable(유효)이 TRUE인 채로 지속됩니다.

타이밍 차트

● MC_MoveLinear(직선 보간) 명령에 이 명령을 사용한 경우



모션 명령 재기동

입력이 'Enable 형식'인 명령에는 모션 명령 재기동은 없습니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

MC_GroupSetOverride(축 그룹 오버라이드값 설정) 명령 실행 중에 같은 축 그룹에 대해 다른 인스턴스의 MC_GroupSetOverride(축 그룹 오버라이드값 설정) 명령을 기동하면 나중에 실행된 인스턴스가 우선 처리됩니다.

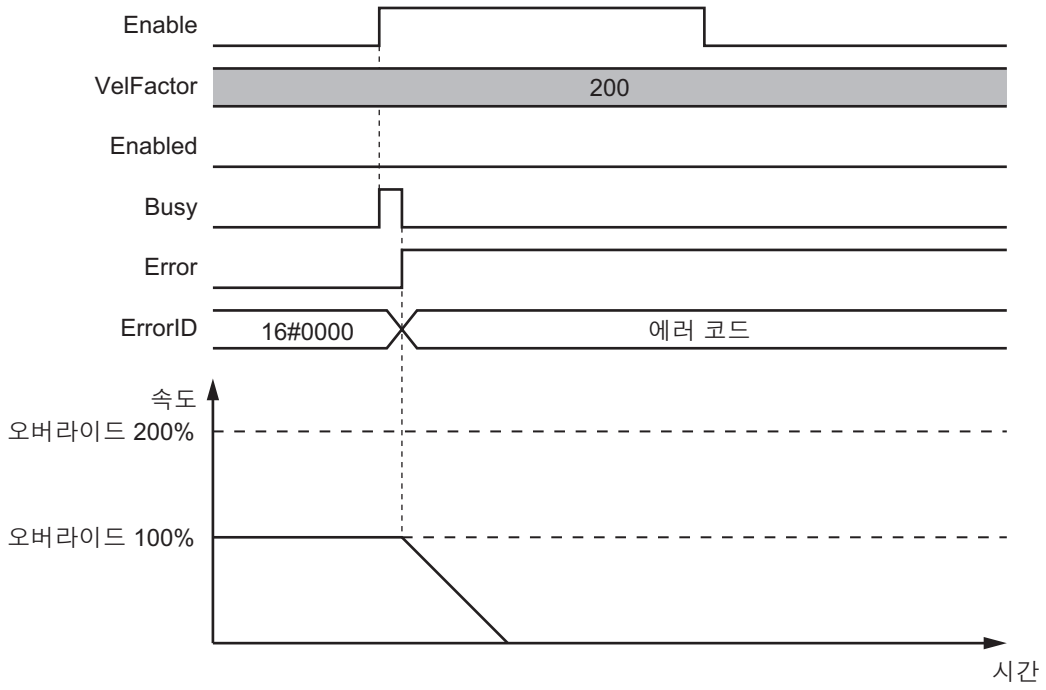
Enabled(유효)는 둘 다 TRUE가 됩니다.

구체적인 동작으로는 나중에 실행된 인스턴스의 오버라이드 값이 유효가 됩니다. 나중에 실행된 인스턴스의 Enable(유효)을 FALSE로 하면 오버라이드는 무효가 됩니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고, 축 그룹 가벼운 이상이 발생하면 축 그룹이 정지합니다.

ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.



● 에러 코드

명령에서 발생하는 이상에 대해서는, 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_GroupReadPosition

축 그룹의 명령 현재 위치와 피드백 현재 위치를 취득합니다.

명령	이름	FB / FUN	그래픽 표현	ST 표현
MC_GroupReadPosition	축 그룹 위치 취득	FB	MC_GroupReadPosition_instance MC_GroupReadPosition AxesGroup — — AxesGroup Enable — Valid CoordSystem — Busy Error ErrorID CommandPosition ActualPosition	MC_GroupReadPosition_instance (AxesGroup :=《파라메터》, Enable :=《파라메터》, CoordSystem :=《파라메터》, Valid =>《파라메터》, Busy =>《파라메터》, Error =>《파라메터》, ErrorID =>《파라메터》, CommandPosition =>《파라메터》 ActualPosition =>《파라메터》);

버전 관련 정보

이 명령은 CPU 유니트 Ver1.01 이상과 Sysmac StudioVer1.02 이상의 조합에서 사용할 수 있습니다.

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Enable	유효	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
CoordSystem	좌표 계통	_eMC_COORD_SYSTEM	0: _mcACS	0 *1	좌표 계통을 지정합니다. 0: 축 좌표 계통(ACS)

*1. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Valid	유효	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.
CommandPosition	명령 현재 위치	ARRAY [0..3] OF LREAL	음수, 양수, '0'	명령 위치의 현재값을 출력합니다. 단위는 [명령 단위]입니다.*2
ActualPosition	피드백 현재 위치	ARRAY [0..3] OF LREAL	음수, 양수, '0'	피드백 위치의 현재값을 출력합니다. 단위는 [명령 단위]입니다.*1

*1 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

*2 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

(주의) 1 그룹의 축 구성이 2축 또는 3축 구성일 때 빈 구성축의 CommandPosition(명령 현재 위치)이나 ActualPosition(피드백 현재 위치)의 값은 잘못된 값이 됩니다.

2 Enable(유효)이 FALSE가 되면 CommandPosition(명령 현재 위치)이나 ActualPosition(피드백 현재 위치)의 값은 잘못된 값이 됩니다.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Valid	Enable의 기동	• Enable이 FALSE가 되었을 때 • Error가 TRUE가 되었을 때
Busy	Enable의 기동	Error가 TRUE가 되었을 때
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- Valid(유효)가 TRUE일 때, 축 그룹 구성축의 명령 현재 위치와 피드백 현재 위치를 제어 주기마다 취득합니다.
- 축 종류별 사용 가능한 현재 위치의 종류는 다음과 같습니다.

축 종류	현재 위치의 종류	
	명령 현재 위치	피드백 현재 위치
서보 축	사용 가능	사용 가능
가상 서보 축	사용 가능	사용 가능*1
엔코더 축	사용 불가능	사용 가능
가상 엔코더 축	사용 불가능	사용 가능

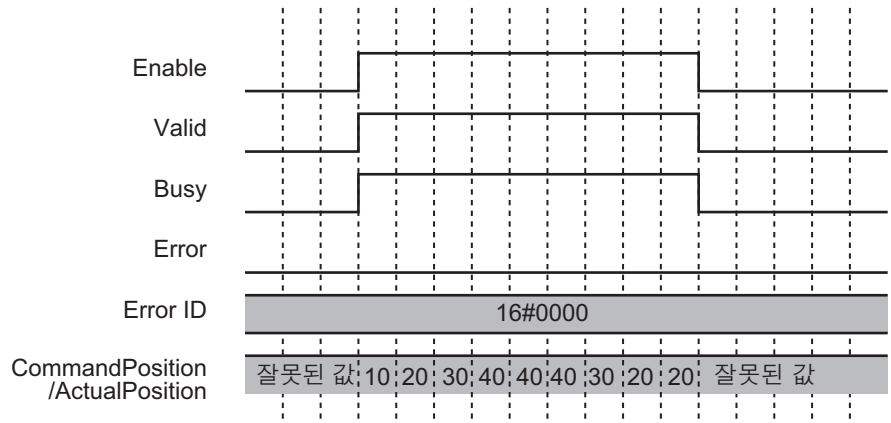
*1. 가상 서보 축에서는 '피드백 위치=명령 위치'가 됩니다.
MC 기능 모듈 내부에서는 배정도 실수 형식으로 처리하고 있으므로 연산 오차를 포함하는 경우가 있습니다.

- 원점이 확정되지 않아도 이 명령을 실행할 수 있습니다.
- 축 그룹 유효, 무효에 관계 없이 이 명령을 사용할 수 있습니다.

타이밍 차트

- Enable(유효)을 TRUE로 하면 Busy(실행 중)와 Valid(유효)가 TRUE가 됩니다.
- Enable(유효)을 FALSE로 하면 Busy(실행 중)와 Valid(유효)는 FALSE가 됩니다.

MC_GroupReadPosition



모션 명령 재기동

입력이 'Enable 형식'인 명령에는 모션 명령 재기동은 없습니다.

모션 명령 다중 기동

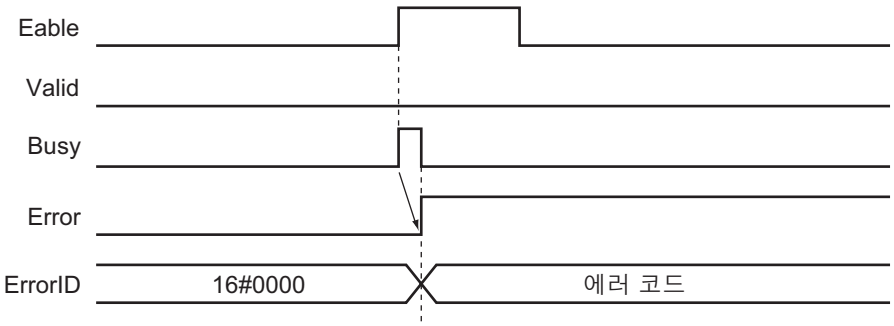
이 명령은 각 명령마다 독립적으로 동작하므로 다중 기동의 제약은 없습니다.

모션 명령 다중 기동의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

MC_GroupReadPosition



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_ChangeAxesInGroup

축 그룹 파라메터의 [구성축]을 일시적으로 재기입합니다.

명령	이름	FB / FUN	그래픽 표현	ST 표현
MC_ChangeAxesInGroup	축 그룹 구성축 쓰기	FB	MC_ChangeAxesInGroup_instance MC_ChangeAxesInGroup AxesGroup — — AxesGroup Axes — — Axes Execute — Done Busy CommandAborted Error ErrorID	MC_ChangeAxesInGroup_instance (AxesGroup := 《파라메터》, Axes := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

사용상의 주의 사항

이 명령으로 재기입한 값은 CPU 유니트의 비휘발성 메모리에 저장되지 않습니다. 따라서 컨트롤러의 전원 'OFF' 및 다운로드, MC 기능 모듈의 재기동 처리가 시작되면 재기입한 값은 사라지고 Sysmac Studio에서 설정한 값으로 되돌아갑니다.
비휘발성 메모리에 저장하려면 Sysmac Studio를 사용해 파라메터를 전송하십시오.

참고

'다운로드'는 Sysmac Studio의 '동기' 기능을 사용합니다.

버전 관련 정보

이 명령은 CPU 유니트 Ver1.01 이상과 Sysmac StudioVer1.02 이상의 조합에서 사용할 수 있습니다.

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*1	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

*1. 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	다른 명령에서 이상이 발생해 이 명령이 중지되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

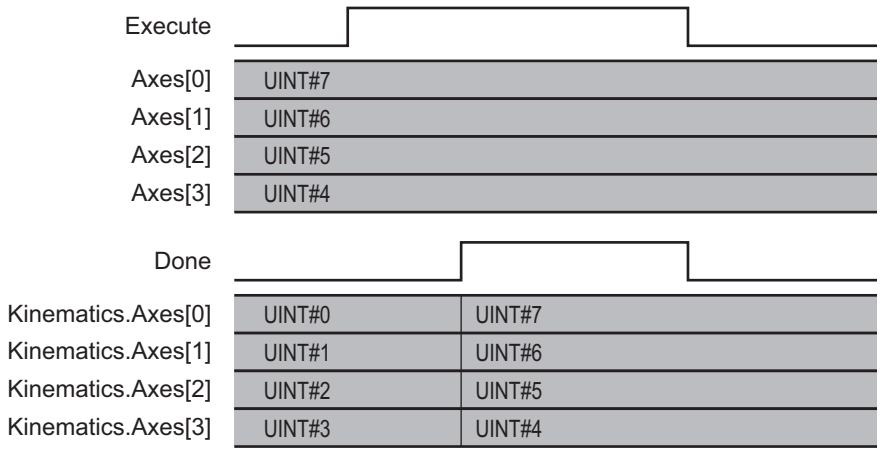
입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	재기입 대상 축 그룹을 지정합니다. *1
Axes	축 그룹 구성축	ARRAY [0..3] OF UINT	–	재기입한 후의 구성축의 축 번호를 지정합니다.*2

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

*2. 구성축에는 서보 축 또는 가상 서보 축을 설정하십시오. 또한 재기입한 후의 구성축 수는 재기입 전의 구성축 수와 동일하게 하십시오.

기능 설명

- Execute(기동)가 기동하면 AxesGroup(축 그룹)에서 지정한 축 그룹의 축 그룹 파라메터에 Axes(축 그룹 구성축)에서 지정한 구성축을 기입합니다.
- 명령의 실행이 완료되면 축 그룹 변수인 Kinematics(운동학 변환 설정)의 Axis[ ]가 지정한 구성축으로 재기입됩니다.
- 이 명령은 축 구성(구성축의 수)을 재기입할 수 없습니다. 예를 들어 구성축이 3축인 축 그룹에 이 명령을 실행해도 구성축은 3축 그대로입니다.
- 구성축의 축 번호를 '0, 1, 2, 3'에서 '7, 6, 5, 4'로 변경할 때의 동작은 다음과 같아집니다.



- 원점의 확정, 미확정에 관계 없이 이 명령을 실행할 수 있습니다.
- 축 그룹이 무효일 때만 이 명령을 실행할 수 있습니다.

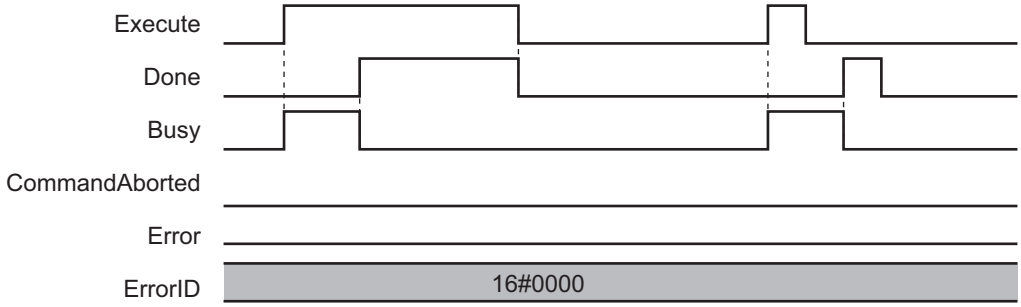
사용상의 주의 사항

이 명령으로 재기입한 값은 CPU 유니트의 비휘발성 메모리에 저장되지 않습니다. 따라서 컨트롤러의 전원 'OFF' 및 다운로드, MC 기능 모듈의 재기동 처리가 시작되면 재기입한 값은 사라지고 Sysmac Studio에서 설정한 값으로 되돌아갑니다.
비휘발성 메모리에 저장하려면 Sysmac Studio를 사용해 파라메터를 전송하십시오.

타이밍 차트

- Execute(기동)를 TRUE로 하면 Busy(실행 중)가 TRUE가 됩니다.
- 구성축 재기입이 완료되면 Done(완료)이 TRUE가 됩니다.

MC_ChangeAxesInGroup 명령



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동을 수행한 경우 '모션 명령 재기동 불가능(에러 코드: 543B Hex)'이 됩니다.

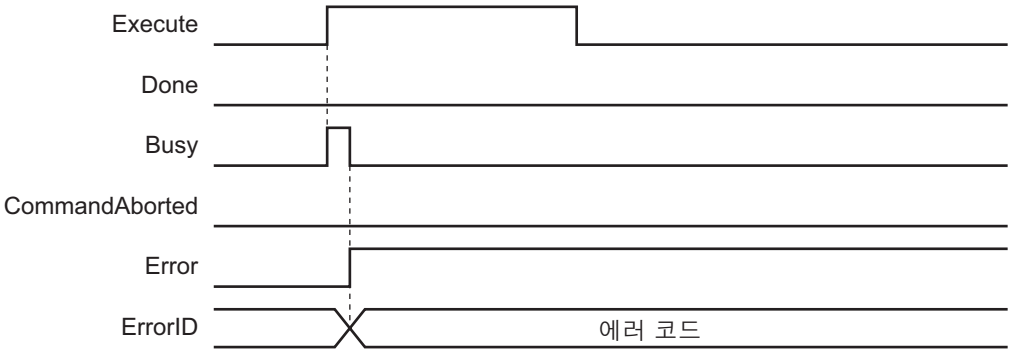
모션 명령다중기동

모션 명령 다중 기동의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

이상

이 명령 실행 시 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
이 때, 파라메터 재기입이 수행되지 않고 명령 실행 전의 값이 유지됩니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_GroupSyncMoveAbsolute

각 축에 지정한 목표 위치를 주기적으로 출력합니다.

명령	이름	FB / FUN	그래픽 표현	ST 표현
MC_GroupSyncMoveAbsolute	축 그룹 주기적 동기 절대 위치 제어	FB	MC_GroupSyncMoveAbsolute_instance MC_GroupSyncMoveAbsolute AxesGroup — AxesGroup Execute — InPosition Position — Busy CoordSystem — Active BufferMode — CommandAborted Error ErrorID	MC_GroupSyncMoveAbsolute_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, Position := 《파라메터》, CoordSystem := 《파라메터》, BufferMode := 《파라메터》, InPosition => 《파라메터》, Busy => 《파라메터》, Active => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

버전 관련 정보

이 명령은 CPU 유니트 Ver1.01 이상과 Sysmac StudioVer1.02 이상의 조합에서 사용할 수 있습니다.

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Position	목표 위치	ARRAY [0..3] OF LREAL	음수, 양수, '0'	0	절대 좌표의 목표 위치를 지정합니다. 단위는 [명령 단위]입니다.*1
CoordSystem	좌표 계통	_eMC_COORD_SYSTEM	0: _mcACS	0 *2	좌표 계통을 지정합니다. 0: 축 좌표 계통(ACS)
BufferMode	버퍼 모드 선택	_eMC_BUFFER_MODE	0: _mcAborting	0	모션 명령 다중 기동 시의 동작을 지정합니다. 0: Aborting

*1. 명령 단위에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

*2. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
InPosition	인포지션	BOOL	TRUE, FALSE	모든 구성 축의 피드백 현재 위치가 목표 위치의 인포지션 폭 안에 들어갔을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Active	제어 중	BOOL	TRUE, FALSE	제어 중에 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
InPosition	모든 구성 축의 피드백 현재 위치가 목표 위치의 인포지션 폭 안에 들어갔을 때	• 피드백 현재 위치가 인포지션 폭 밖으로 나왔을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Busy	Execute의 기동	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
Active	명령을 시작했을 때	• Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	• 다른 명령에 의한 모션 명령 다중 기동으로 이 명령이 중지되었을 때 • 다른 명령으로 이상이 발생해 이 명령이 중지되었을 때 • 축 이상 발생 중에 이 명령을 기동했을 때 • MC_GroupStop 명령 실행 중에 이 명령을 기동했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

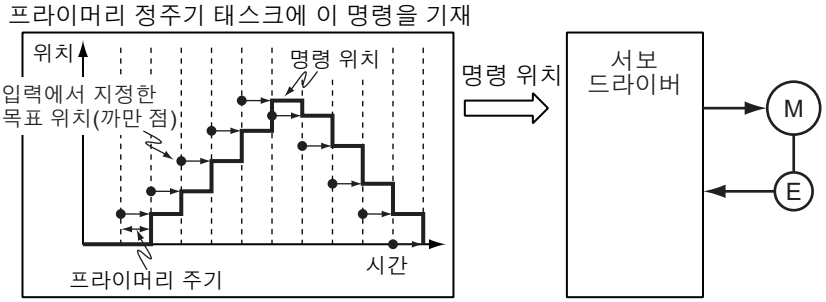
입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

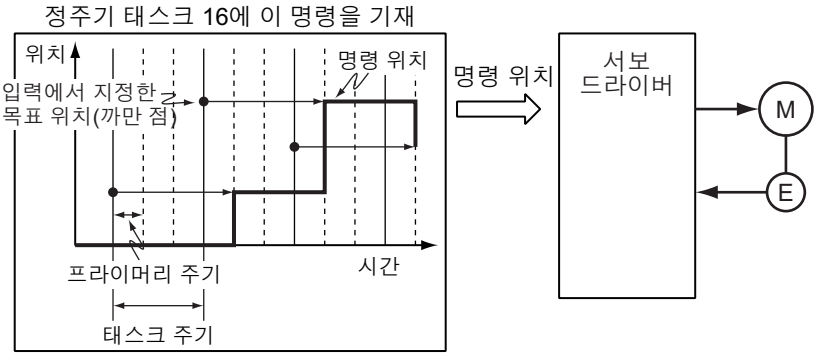
*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- 이 명령은 사용자 프로그램에서 전달하는 목표 위치를 프라이머리 주기마다 주기적 동기 위치 모드(CSP)로 서보 드라이버 등에 출력합니다. 목표 위치는 절대 위치로 전달합니다.
- 속도는 축 파라메터의 [최고 속도]를 상한으로 합니다. 최대 가속도와 최대 감속도는 적용되지 않습니다.
- 프라이머리 정주기 태스크에 이 명령을 기술한 경우, 입력에서 지정한 목표 위치는 다음 프라이머리 주기로 서보 드라이버에 출력됩니다.



- 정주기 태스크 16에 이 명령을 기술한 경우, 입력에서 지정한 목표 위치는 다음 정주기 태스크로 서보 드라이버에 출력됩니다.



사용상의 주의 사항

- 축 그룹의 구성축 중 하나가 원점 미확정인 경우는 '원점 미확정 상태에서 명령 기동 이상(에러 코드: 5466 Hex)'이 됩니다.
- 지시한 목표 위치에 의한 이동량이 축 파라메터의 [최고 속도]를 초과하지 않도록 목표 위치를 명령하십시오. [최고 속도]를 초과하는 목표 위치를 지시했을 때는 명령 속도 포화가 되어 [최고 속도]로 제한된 이동량을 출력합니다. 지시한 목표 위치에서 부족한 분량의 이동량 지시는 다음 주기 이후에 출력합니다.
  또한 이 때 축 제어 Status의 Details.VelLimit(명령 속도 포화)가 TRUE가 됩니다.

명령에 대한 자세한 내용

명령에 대해 자세히 설명합니다.

● 인포지션 확인

Position(목표 위치)을 갱신하지 않을 때는 목표 위치와 피드백 위치의 폭이 축 파라메터의 [인포지션 폭] 범위 내가 되었을 때 InPosition(인포지션)이 TRUE가 됩니다.
InPosition(인포지션)이 TRUE일 때 목표 위치를 변경해도 해당 주기 중에는 TRUE인 그대로이며, 다음 주기에 FALSE가 됩니다.
축 파라메터의 [인포지션 확인 시간] 설정은 무효입니다.

● 정지 처리

축 그룹 동작을 정지시킬 때의 동작에 대해 설명합니다.
동작을 정지시키려면 MC_GroupStop(축 그룹 강제 정지) 명령 또는 MC_GroupImmediateStop(축 그룹 즉시 정지) 명령을 사용합니다. 이 명령들을 기동하면 이 명령의 CommandAborted(실행 중단)가 TRUE가 됩니다.

MC_GroupStop(축 그룹 강제 정지) 명령에 의한 정지

즉시 정지합니다.

MC_GroupImmediateStop(축 그룹 즉시 정지) 명령에 의한 정지

각 축의 파라메터 설정 [즉시 정지 입력 정지 방법]의 지정에 따라 즉시 정지합니다.

● 이상 발생에 의한 정지

축을 정지시키는 이상이 발생했을 때 설정에 관계 없이 '즉시 정지'합니다.

● 사용 가능 축

- 서보 축에서 이 명령을 사용할 수 있습니다.
  사용하는 경우는 MC_Power(운전 가능) 명령의 Enable(유효)을 TRUE(서보 ON 상태)로 하십시오.
- 가상 서보 축은 항상 수신할 수 있습니다.

● 기동 조건

- 축 파라메터의 [카운트 모드]를 리니어 모드로 하십시오.
- 모든 구성축을 원점 확정 상태로 하십시오.
- MC_GroupEnable(축 그룹 유효) 명령으로 축 그룹을 유효로 하십시오.

● 축 변수의 Status

축 그룹 변수의 축 그룹 Status는 Status.Moving(동작 중)이 TRUE가 됩니다.
축 그룹 제어 Status에는 영향을 미치지 않습니다.

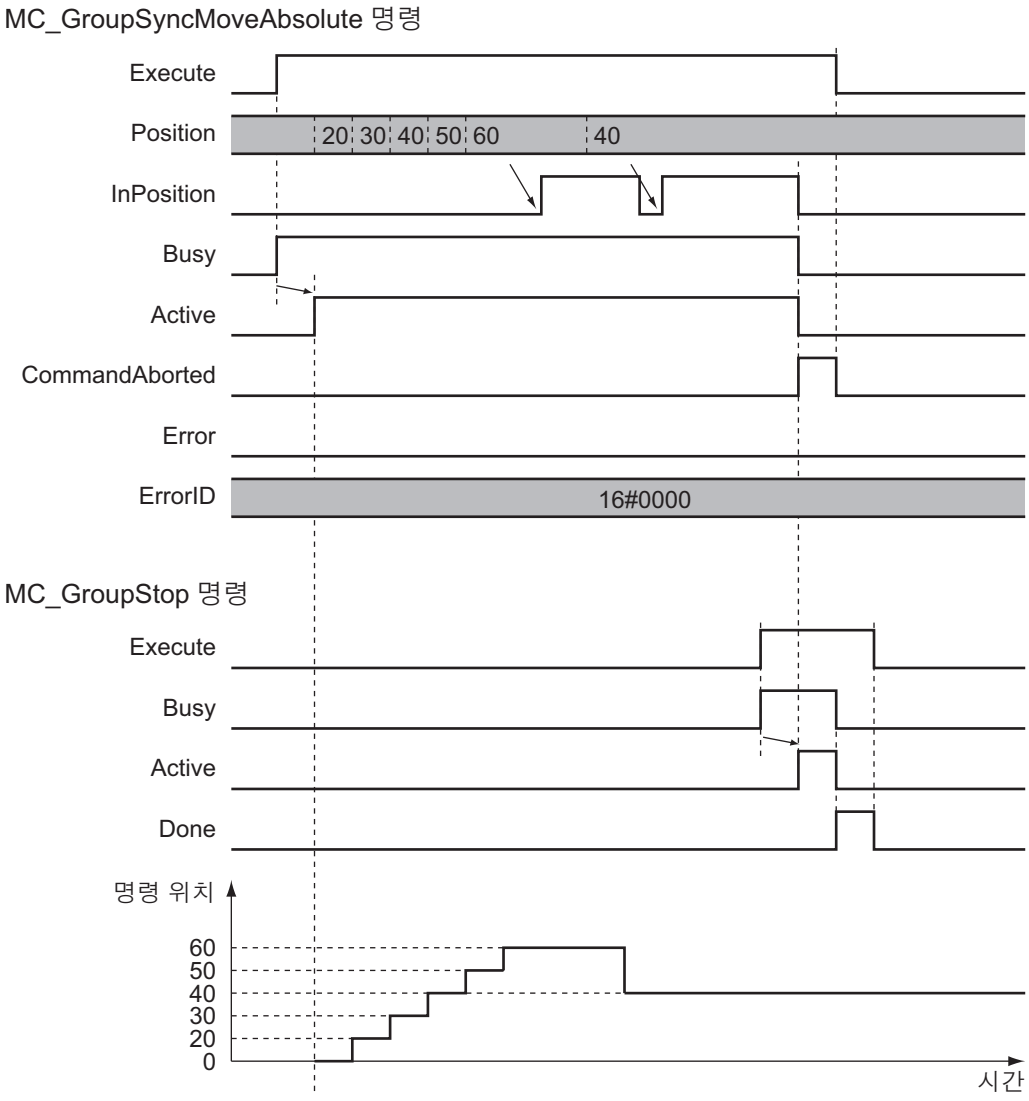
● 오버라이드

이 명령에 대해 오버라이드는 무효가 됩니다.

타이밍 차트

- Execute(기동)가 기동하는 동시에 Busy(실행 중)가 TRUE가 되고, 다음 주기에서 Active(제어 중)가 TRUE가 됩니다.
- 모든 구성축의 피드백 현재 위치가 Position(목표 위치)의 인포지션 폭 안에 들어갔을 때 InPosition(인포지션)이 TRUE가 됩니다.
- 다른 명령에 의해 이 명령이 중지되면 CommandAborted(실행 중단)가 TRUE가 되고 Busy(실행 중), Active(제어 중), InPosition(인포지션)이 FALSE가 됩니다.
- 이 명령을 종료하려면 MC_GroupStop(축 그룹 강제 정지) 명령을 기동하십시오.

프라이머리 정주기 태스크에 이 명령을 기재한 경우는 다음과 같습니다.



참고

MC 기능 모듈은 서보 드라이버에 대해 위의 타이밍 차트로 제어 모드를 명령합니다. 그러나 서보 드라이버에 반영되는 타이밍은 각 서보 드라이버에 사양에 따릅니다.

모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동을 수행한 경우 '모션 명령 재기동 불가능(에러 코드: 543B Hex)'이 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

이 명령의 BufferMode(버퍼 모드 선택)는 Aborting에만 대응합니다.

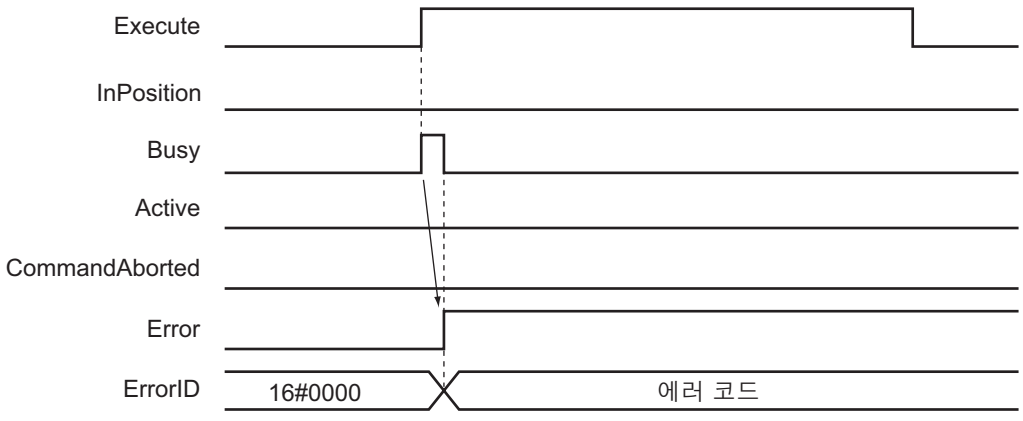
● 이 명령 실행 중의 다른 명령 기동

이 명령에 대한 모션 명령 다중 기동을 실행하는 경우 Aborting만 가능합니다.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 축이 정지합니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_GroupReset

축 그룹 및 축의 이상을 해제합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_GroupReset	축 그룹 에러 리셋	FB	MC_GroupReset_instance MC_GroupReset AxesGroup — — AxesGroup Execute — Done Busy Failure Error ErrorID	MC_GroupReset_instance (AxesGroup := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, Failure => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
Failure	잘못된 종료	BOOL	TRUE, FALSE	명령이 정상 실행되지 않았을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	이상 해제 처리가 정상 종료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • Failure가 TRUE가 되었을 때
Failure	• 이상에 의한 축, 축 그룹의 감속 정지 중에 명령을 실행했을 때 • 축 공통 이상 발생에 의한 축 그룹 이상 발생 중에 명령을 실행했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
AxesGroup	축 그룹	_sGROUP_REF	–	축 그룹을 지정합니다. *1

*1. Sysmac Studio의 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***') 또는 시스템 정의 변수의 축 그룹 변수명(_MC_GRP[**])을 사용하십시오.

기능 설명

- Execute(기동)가 기동하면 GroupEnable(그룹 유효) 상태인 AxesGroup(축 그룹)에서 지정한 축 그룹의 이상 및 축 그룹에 속하는 축 이상의 해제 처리를 실행합니다.
  해제가 가능한 것은 축이나 축 그룹에서 발생하는 '가벼운 이상' 수준, '감시 정보' 수준의 이상 해제 및 드라이버 에러 리셋입니다.
- 축이 서보의 ON/OFF 상태에 상관 없이 이상 해제 처리를 실행합니다.
- 드라이버 에러가 발생한 축에는 드라이버 에러 리셋 처리를 실행한 후 이상 해제 처리를 실행합니다.
- 드라이버 에러 리셋 처리는 드라이버 에러가 해제되거나 축 파라메터의 [드라이버 에러 리셋 감시 시간] 동안 계속됩니다. 드라이버 에러 리셋 처리는 축 그룹에 속하는 축에 대해 동시에 이루어집니다.
- 해제되는 이상은 Execute(기동)의 기동 타이밍에서 발생한 이상을 대상으로 합니다. 해제 중에 발생한 이상은 해제되지 않습니다.
- 축 그룹 에러 감속 정지 중에 명령을 실행하면 실행되지 않고 Failure(잘못된 종료)가 TRUE가 됩니다. 이것은 축이 정지하기 전에 이상을 해제할 수 없도록 하기 위한 것입니다.
  또한 축 공통 이상 발생에 의한 축 이상도 이 명령으로는 해제할 수 없으므로 Failure(잘못된 종료)가 TRUE가 됩니다.

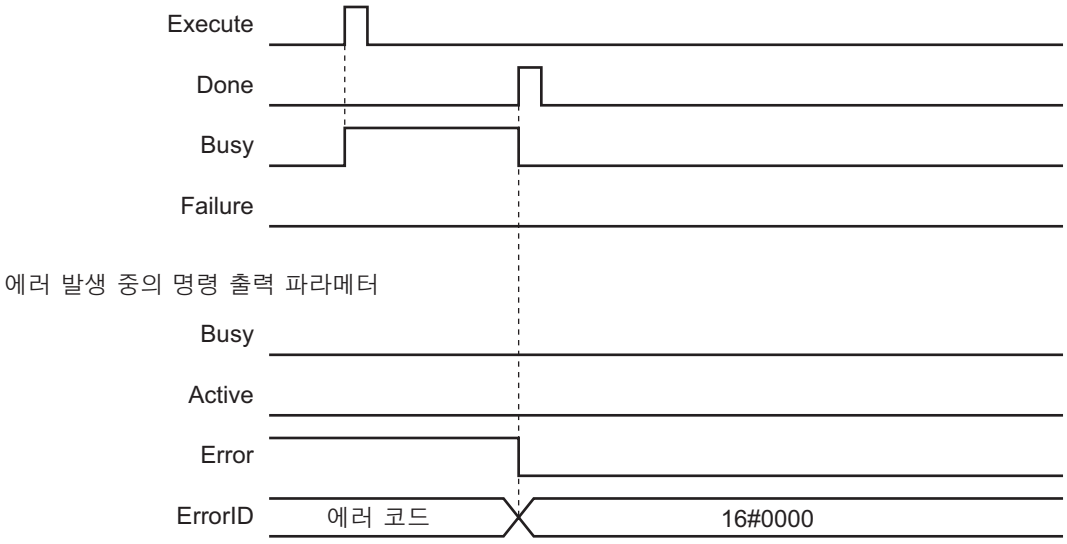
사용상의 주의 사항

- 이 명령에 의한 이상 해제 처리는 여러 제어 주기를 포함하는 경우가 있습니다.
- 이 명령을 기동해도 축이 동작 중이면 이 명령의 출력 변수 'Failure(잘못된 종료)'가 TRUE가 됩니다.
  이상 요인을 제거한 후 Done(완료)이 TRUE가 될 때까지 재시도 처리를 실행하십시오.
- 이상 발생 요인을 제거하는 경우는 각 축이 완전히 정지한 것을 확인한 후 실행하십시오.
- 오므론 서버 드라이버 G5 시리즈에 대해 이 명령을 사용하는 경우, ResetECError(EtherCAT 이상 해제) 명령이 동시에 실행되지 않도록 배타적 처리를 하십시오.
- NX 시리즈 펄스 출력 유니트에 대해 이 명령을 사용하는 경우, 펄스 출력 유니트에 연결된 드라이버의 이상 상태는 해제되지 않습니다. 자세한 내용은 "NX 시리즈 위치 인터페이스 유니트 사용 설명서(SBCE-374)"를 참조하십시오.

참고

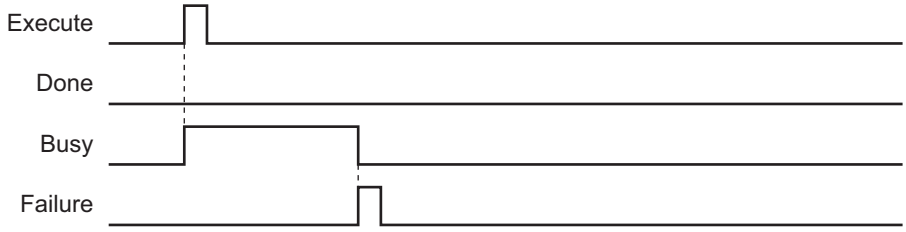
- 축 이상은 축 그룹이 유효인 경우만 해제할 수 있습니다.
- 이 명령으로는 다음 이상을 해제할 수 없습니다.
  축 공통 이상 전체
  이상을 해제하려면 ResetMCError(전체 에러 리셋) 명령을 별도로 실행하십시오.
- 이상이 발생하지 않은 축 그룹에 대해 이 명령을 실행하면 명령은 정상 종료되고 상태가 지속됩니다.

타이밍 차트



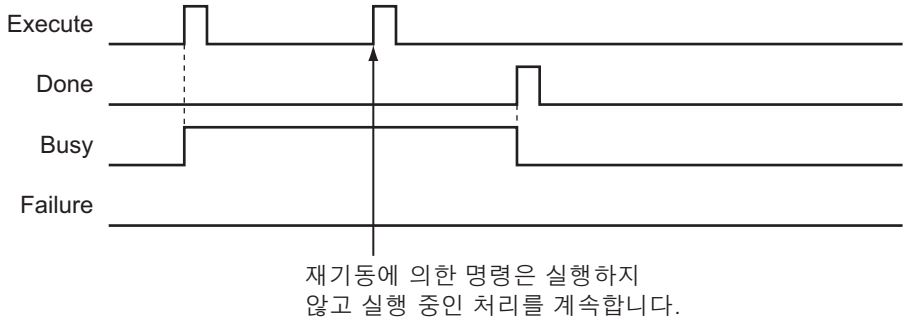
명령 중지

이상에 의한 축, 축 그룹의 감속 정지 중 또는 축 공통 이상 발생에 의한 축 그룹 이상 발생 중에 이상을 해제할 수 없는 경우는 명령을 중단합니다.



모션 명령 재기동

이 명령 실행 중에 다시 Execute(기동)를 기동하면 재기동 명령은 무시하고 이상 해제 처리를 계속합니다.



모션 명령 다중 기동

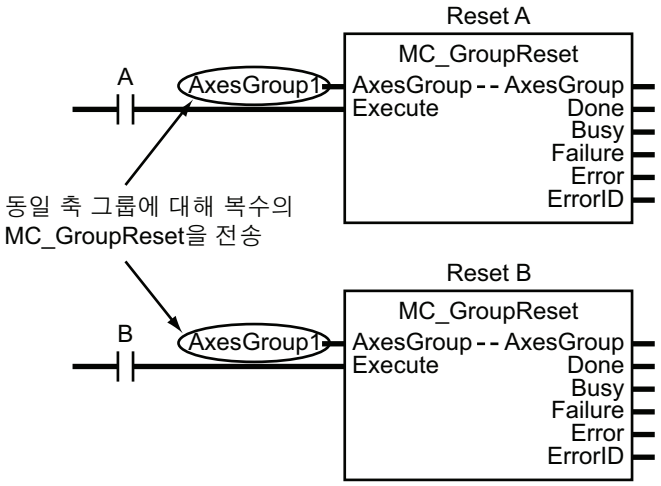
모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 이 명령 실행 중의 다른 명령 기동

동일한 축 그룹에 대해 다른 인스턴스의 MC_GroupReset(축 그룹 에러 리셋) 명령이 동시에 실행되면 두 명령을 각각 실행합니다.

슬레이브 계통 이상 발생 중에는 축 설정의 '드라이버 에러 리셋 감시 시간'까지 대기하는 경우가 있습니다. 경과 시간도 명령 인스턴스별로 카운트합니다.

이 명령 실행 중에 지정한 축 그룹에 속한 축에 대해 MC_Reset(축 에러 리셋) 명령이 실행된 경우에도 마찬가지로 두 명령을 각각 실행합니다.



에러 코드

명령에서 발생하는 이상에 대해서는, 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

5

공통 명령

축 및 축 그룹에 공통된 명령에 대해 설명합니다.

MC_SetCamTableProperty **5-2**
MC_SaveCamTable **5-8**
MC_Write **5-13**

MC_SetCamTableProperty

입출력 파라메터에서 지정한 캠 테이블의 종료점 인덱스를 갱신합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_SetCamTableProperty	캠 테이블 속성 갱신	FB	MC_SetCamTableProperty_instance MC_SetCamTableProperty CamTable — — CamTable Execute — Done EndPointIndex MaxDataNumber Busy CommandAborted Error ErrorID	MC_SetCamTableProperty_instance (CamTable := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, EndPointIndex => 《파라메터》, MaxDataNumber => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이름	데이터 형식	유효 범위	초기값	내용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행이 완료되었을 때 TRUE가 됩니다.
EndPointIndex	종료점 인덱스	UINT	양수 또는 '0'	캠 테이블의 종료점 인덱스를 출력합니다.
MaxDataNumber	최대 캠 데이터 수	UINT	양수	최대 캠 데이터 수를 출력합니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	캠 테이블의 재기입이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	이 명령을 중지할 요인이 발생했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
CamTable	캠 테이블	ARRAY[0..N] OF _sMC_CAM_REF	–	캠 데이터 구조체 _sMC_CAM_REF 형식의 배열 변수를 캠 테이블로 지정합니다. *1

*1. 배열 요소 [N]은 Sysmac Studio에 의해 자동 설정됩니다. Sysmac Studio의 캠 편집기로 작성한 캠 데이터 변수를 지정하십시오.

기능 설명

- 이 명령은 입출력 파라메터에서 지정한 캠 테이블의 종료점 인덱스를 갱신합니다.
- 종료점이란 캠 테이블을 시작점부터 순서대로 검색했을 때 위상이 최초로 '0'이 되는 캠 데이터보다 하나 이전에 배치된 데이터입니다.
- 종료점의 배열 번호를 EndPointIndex(종료점 인덱스)에 출력합니다.
- 위상 '0'을 검출하면 그 이후는 무효 캠 데이터로 다루어지므로 위상/변위의 값은 무시됩니다.
- 최대 캠 데이터 수란 캠 데이터 구조체 배열의 최대 요소 수를 나타냅니다.
  최대 캠 데이터 수란 Sysmac Studio에서 구조체 배열을 선언할 때 지정한 값입니다.
- 사용자 프로그램에서 캠 데이터의 종료점 인덱스를 변경한 경우 종료점을 갱신해야 합니다. 이 경우 이 명령을 사용해서 유효 캠 데이터 수를 갱신하십시오.

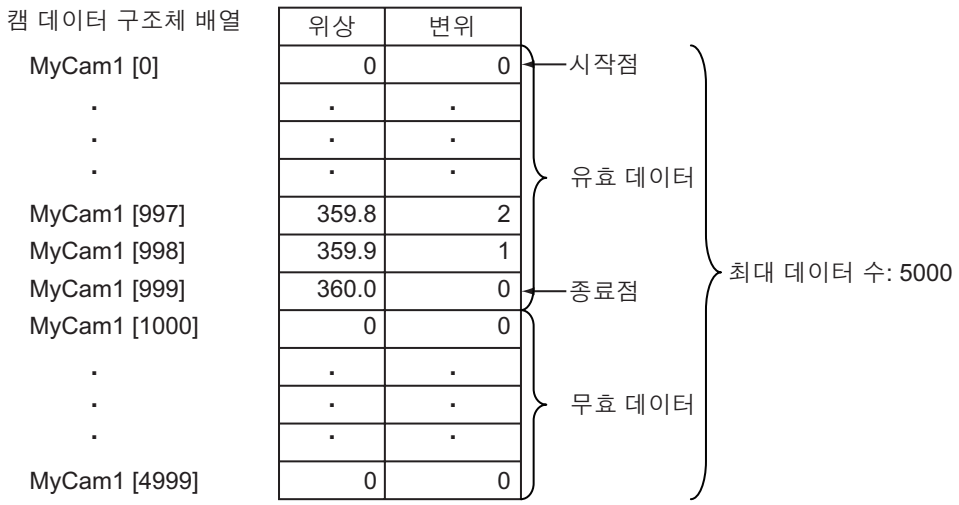
캠 테이블에 대해서는 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

사용상의 주의 사항

- 캠 테이블 검색 중에 위상 '0'을 발견하기 전에 위상이 오름차순으로 되어 있지 않은 것을 검출하면 이상이 됩니다.
- 최대 캠 데이터 수는 사용자 프로그램에서 변경할 수 없습니다.
- 캠 데이터의 종료점을 변경하거나 캠 데이터의 값을 재기입한 후에는 이 명령을 실행하십시오. 종료점 인덱스를 변경하거나 위상이 오름차순이 아니면 캠 동작이나 MC_CamIn(캠 동작 시작) 명령의 EndOfProfile(캠 사이클 완료)이 의도한 대로 동작하지 않습니다.
- 이 명령 처리 중에 캠 테이블 데이터를 변경하면 갱신 처리가 제대로 실행되지 않습니다. 프로그램에서 캠 데이터를 변경하려면 이 명령을 완료한 후 실행하십시오.
- 캠 데이터 변수는 글로벌 변수로 복수의 태스크에서 캠 데이터 변수값을 참조하거나 변경할 수 있습니다. 복수의 태스크로부터 캠 데이터 변수값을 변경하려면 복수의 태스크에 의한 재기입 처리로 경쟁이 발생하지 않도록 프로그램을 작성하십시오.
- '글로벌 변수의 태스크 간 배타적 제어' 기능을 이용해서 캠 데이터 변수를 배타적으로 제어하려면 배타 대상인 캠 데이터 변수를 이용해서 모션 명령을 기동하지 마십시오. 가벼운 수준의 이상인 '캠 테이블 지정 오류(에러 코드: 5439 Hex)'가 발생합니다.

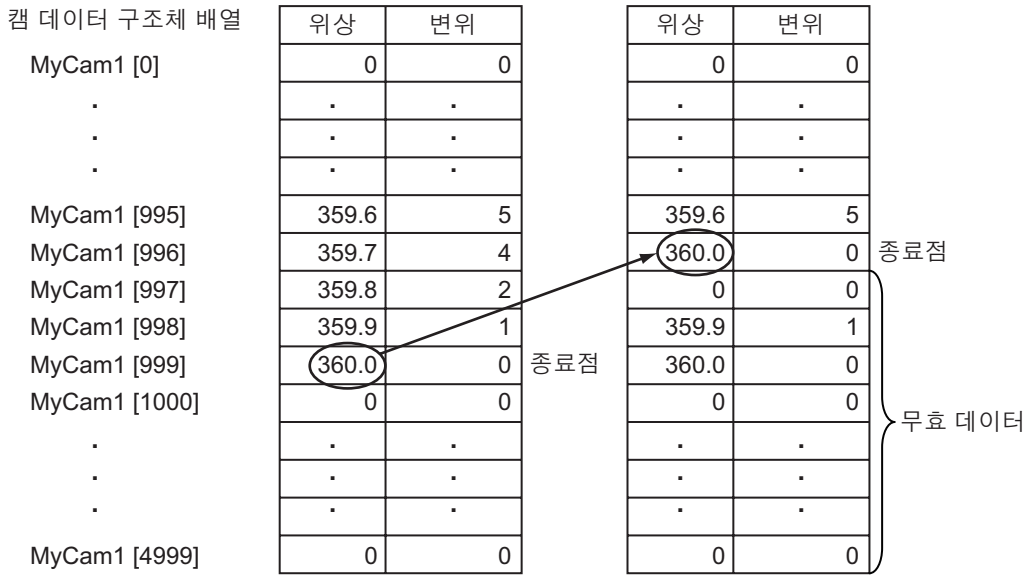
명령에 대한 자세한 내용

예를 들어 다음과 같은 캠 테이블의 경우 EndPointIndex(종료점 인덱스)는 '999', MaxDataNumber(최대 캠 데이터 수)는 '5000'이 됩니다.

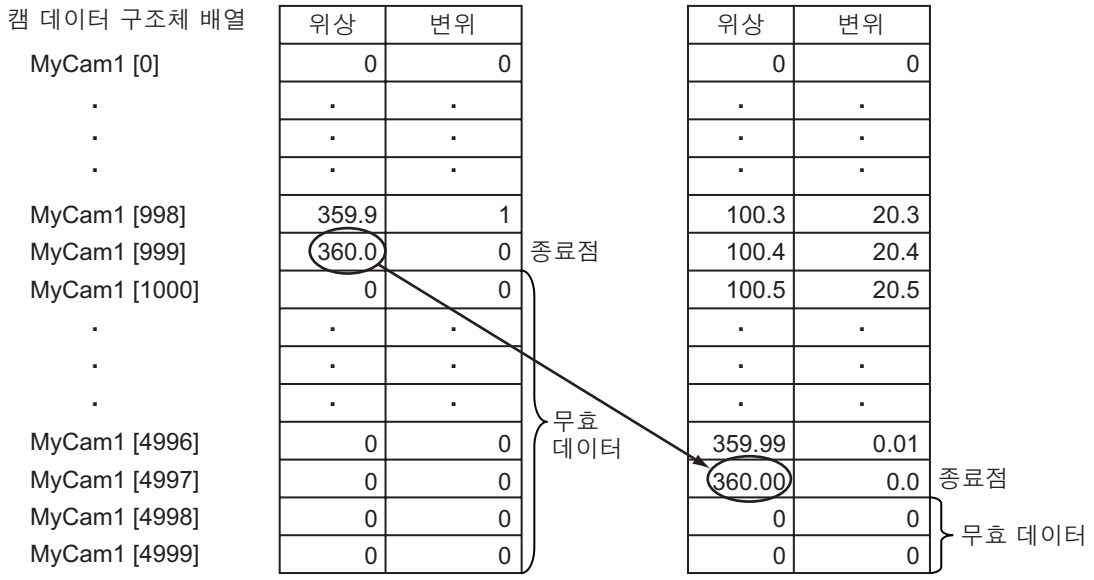


캠 데이터 재기입과 EndPointIndex(종료점 인덱스)의 관계를 나타내면 다음과 같습니다.

MyCam1[1000] 이후의 위상이 '0'인 캠 테이블에서 이 명령을 기동하면 EndPointIndex(종료점 인덱스)가 '999'가 됩니다.
MyCam1[997]의 위상을 '0'으로 재기입한 후 이 명령을 기동하면 EndPointIndex(종료점 인덱스)가 '996'이 됩니다.



또한 MyCam1[1000]에서 MyCam1[4997]까지의 위상을 '0' 이외의 숫자로 재기입한 후 이 명령을 기동하면 EndPointIndex(종료점 인덱스)가 '4997'이 됩니다.



MyCam1 [1000]에서 MyCam1 [4997]까지의 위상을 '0' 이외의 숫자로 재기입했는데도 이 명령을 기동하지 않으면 캠 동작은 MyCam1[0]부터 MyCam1[999] 사이에서 실행됩니다.
캠 테이블을 재기입해도 EndPointIndex(종료점 인덱스)는 바뀌지 않습니다.

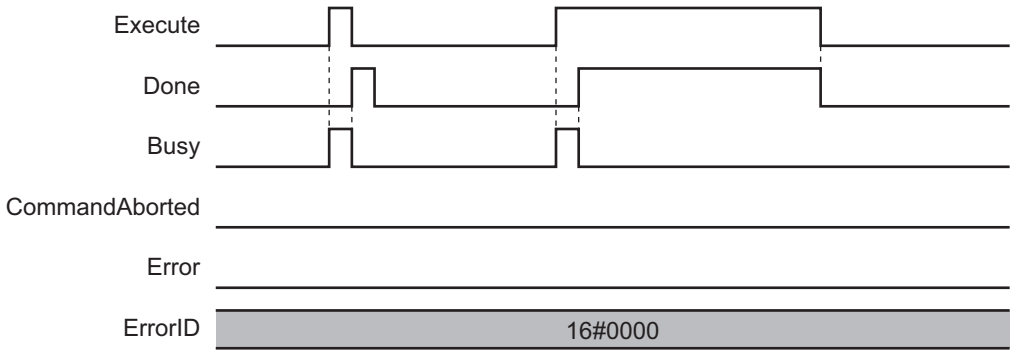
캠 테이블의 데이터 구조에 대한 자세한 내용은 'MC_CamIn(3-161페이지)'을 참조하십시오.

타이밍 차트

이 명령의 기동 방법 2가지는 다음과 같습니다. 동작은 두 기동 방법의 경우 모두 정상 종료를 나타냅니다.

첫 번째 기동은 Execute(기동)를 TRUE로 하고 완료 전에 Execute(기동)를 False로 한 경우입니다.

두 번째 기동은 Execute(기동)를 유지한 경우를 나타냅니다.



모션 명령 재기동

이 명령은 재기동할 수 없습니다.

재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

● 다른 명령 실행 중의 이 명령 기동

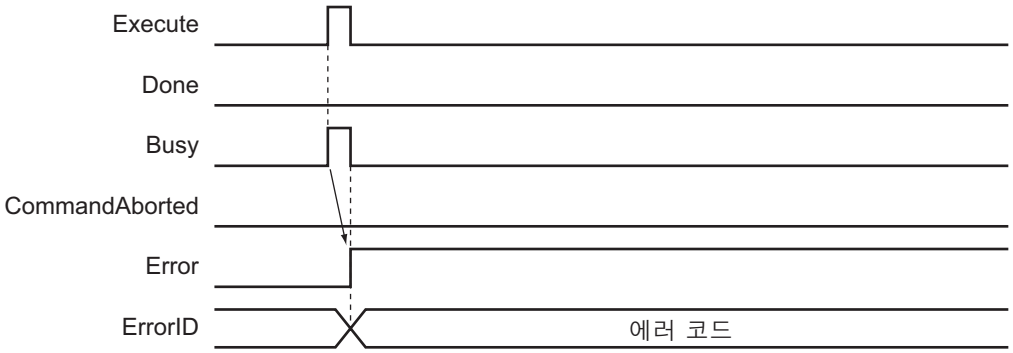
이 명령과 동일한 CamTable(캠 테이블)을 사용하는 MC_SaveCamTable(캠 테이블 저장) 명령이나 MC_SetCamTableProperty(캠 테이블 속성 갱신) 명령이 이미 기동되어 있으면 이 명령은 다중 기동할 수 없습니다.

● 이 명령 실행 중의 다른 명령 기동

이 명령 실행 중에 동일한 CamTable(캠 테이블)을 사용하는 MC_SaveCamTable(캠 테이블 저장) 명령이나 MC_SetCamTableProperty(캠 테이블 속성 갱신) 명령을 다중 기동할 수 없습니다.

이상

이 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.



● 에러 코드

이 명령에서 발생하는 이상에 대해서는, 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_SaveCamTable

입력 파라메터에서 지정한 캠 테이블을 비휘발성 메모리에 저장합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_SaveCamTable	캠 테이블 저장	FB	MC_SaveCamTable_instance MC_SaveCamTable CamTable — CamTable Execute — Done Busy CommandAborted Error ErrorID	MC_SaveCamTable_instance (CamTable := 《파라메터》, Execute := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	캠 테이블의 저장이 완료되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	이 명령을 중지할 요인이 발생했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
CamTable	캠 테이블	ARRAY[0..N] OF _sMC_CAM_REF	–	캠 데이터 구조체 _sMC_CAM_REF 형식의 배열 변수를 캠 테이블로 지정합니다. *1

*1. 배열 요소 [N]은 Sysmac Studio에 의해 자동 설정됩니다. Sysmac Studio의 캠 편집기로 작성한 캠 데이터 변수를 지정하십시오.

기능 설명

- 이 명령은 입출력 변수에서 지정한 캠 테이블을 비휘발성 메모리에 저장합니다.
- 캠 테이블의 저장 처리 중에는 모션 제어 시스템 변수 _MC_COM.Status.CamTableBusy(캠 테이블 파일 저장 실행 중/대기 중)가 TRUE가 됩니다.

캠 테이블의 데이터 사양에 대해서는, 'MC_CamIn(3-161페이지)'을 참조하십시오.

사용상의 주의 사항

- 캠 데이터를 재기입한 후에는 컨트롤러의 전원을 'OFF'하기 전에 이 명령으로 저장하십시오. 저장하지 않고 컨트롤러 전원을 'OFF'하면 재기입한 데이터는 사라집니다.
- 이 명령 처리 중에 캠 테이블 데이터를 변경하면 갱신 처리가 제대로 실행되지 않습니다. 프로그램에서 캠 테이블을 변경하는 경우는 이 명령 처리 중에 캠 테이블에 기록하지 마십시오.
- 이 명령은 다른 명령에 비해 현저하게 처리 시간이 길어집니다. 또한 CPU 유니트의 연산 부하량에 따라 처리 시간이 크게 다른 경우가 있습니다. 따라서 이 명령의 완료를 트리거로 다음 명령을 기동할 경우는 다음 명령의 기동 타이밍에 주의하십시오.
- 이 명령 처리 중에 컨트롤러의 전원을 'OFF'하지 마십시오. 'OFF'하면 제대로 데이터가 저장되지 않습니다. 또한 비휘발성 메모리에 저장되어 있는 캠 테이블이 파손될 가능성이 있습니다.
- 이 명령 처리 중에는 캠 데이터의 다운로드나 업로드, 온라인 시작 조작, 온라인 편집, 데이터 트레이스 시작 조작 등을 사용할 수 없습니다.
  또한 캠 데이터의 업로드나 다운로드, 온라인 편집 등을 실행하고 있는 경우에 이 명령을 실행하면 캠 테이블 저장 명령 실행 불가(에러 코드: 743CHex)가 됩니다. 캠 데이터가 정확하게 저장될 때까지 재시도 처리를 실행하십시오.
  또한 '다운로드'나 '업로드'는 Sysmac Studio의 '동기' 기능을 사용합니다.
- 비휘발성 메모리의 기록 횟수에는 제한이 있습니다. 이 명령을 자주 실행하면 비휘발성 메모리의 수명이 짧아지므로 필요할 때만 이 명령을 사용하십시오.
- 캠 데이터 변수는 글로벌 변수로 복수의 태스크에서 캠 데이터 변수값을 참조하거나 변경할 수 있습니다. 복수의 태스크로부터 캠 데이터 변수값을 변경하려면 복수의 태스크에 의한 재기입 처리로 경쟁이 발생하지 않도록 프로그램을 작성하십시오.
- '글로벌 변수의 태스크 간 배타적 제어' 기능을 이용해서 캠 데이터 변수를 배타적으로 제어하려면 배타 대상인 캠 데이터 변수를 이용해서 모션 명령을 기동하지 마십시오. 가벼운 수준의 이상인 '캠 테이블 지정 오류(에러 코드: 5439 Hex)'가 발생합니다.
- 온라인 편집 반영 중(*)에 이 명령을 실행하지 마십시오. 실행하면 온라인 편집에 의한 변경이 정확하게 반영되지 않을 가능성이 있습니다.

* 반영 확인 창에서 [Yes]를 선택한 후 반영 확인 창이 닫힐 때까지의 시간을 말하며, CPU 유니트 Ver.1.04 이상과 Sysmac Studio Ver.1.05 이상의 조합에서는 내장 비휘발성 메모리 백업 중 창이 닫힐 때까지의 시간을 말합니다.

● CPU 유니트 동작 모드와의 관계

이 명령에 따라 캠 데이터를 저장 중이거나 CPU 유니트 동작 모드를 프로그램 모드로 전환한 후에도 캠 데이터의 저장 처리는 계속됩니다.

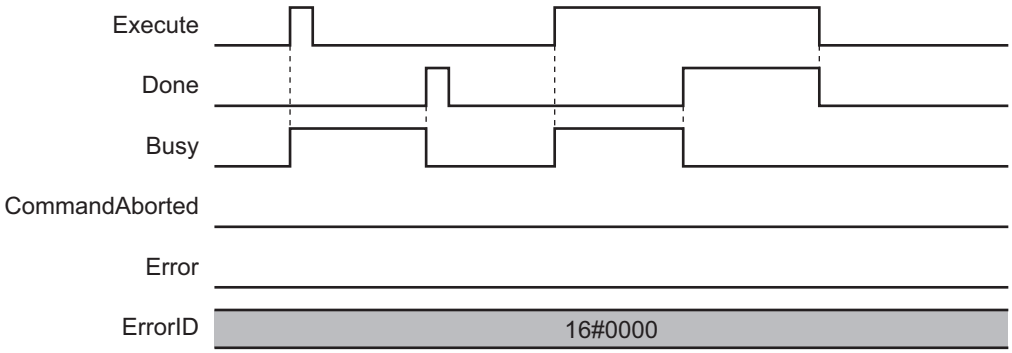
프로그램 모드 중에 캠 데이터 저장 처리가 실행되고 있는지 확인하려면 Sysmac Studio를 온라인으로 해서 모션 제어 시스템 변수 _MC_COM.Status.CamTableBusy를 모니터링하십시오.

● 온라인 편집에 의한 명령 삭제 시

이 명령을 통해 캠 데이터를 저장 중에 온라인 편집을 통해 이 명령을 삭제해도 캠 데이터의 저장 처리는 계속됩니다.

타이밍 차트

이 명령의 기동 방법 2가지는 다음과 같습니다. 동작은 두 기동 방법의 경우 모두 정상 종료를 나타냅니다.
첫 번째 기동은 Execute(기동)를 TRUE로 하고 완료 전에 Execute(기동)를 False로 한 경우입니다.
두 번째 기동은 Execute(기동)를 유지한 경우를 나타냅니다.



모션 명령 재기동

이 명령은 재기동할 수 없습니다.
재기동하면 '모션 명령 재기동 불가(에러 코드: 543B Hex)'가 됩니다.

모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

● 다른 명령 실행 중의 이 명령 기동

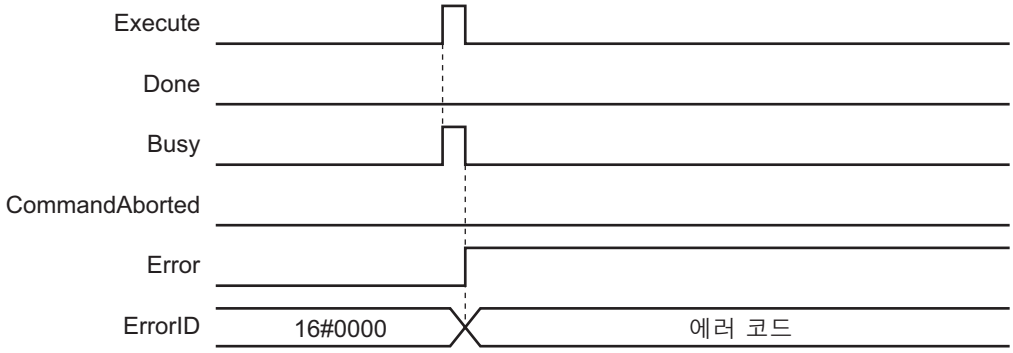
이 명령과 동일한 CamTable(캠 테이블)을 사용하는 MC_SetCamTableProperty(캠 테이블 속성 갱신) 명령이 이미 기동되어 있으면 이 명령은 다중 기동할 수 없습니다.
또한 MC_SaveCamTable(캠 테이블 저장) 실행 중에는 이 명령을 다중 기동할 수 없습니다.

● 이 명령 실행 중의 다른 명령 기동

이 명령 실행 중에 동일한 CamTable(캠 테이블)을 사용하는 다른 명령의 MC_SetCamTableProperty(캠 테이블 속성 갱신) 명령을 다중 기동할 수 없습니다.
또한 이 명령 실행 중에 MC_SaveCamTable(캠 테이블 저장) 명령을 다중 기동할 수 없습니다.

이상

이 명령 실행 중에 이상이 발생하면 Error(에러)가 TRUE가 됩니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.



● 에러 코드

이 명령에서 발생하는 이상에 대해서는, 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

MC_Write

MC 기능 모듈의 모션 제어 파라메터 설정의 일부를 재기입합니다.

명령	명칭	FB/ FUN	그래픽 표현	ST 표현
MC_Write	MC 설정 쓰기	FB	MC_Write_instance MC_Write Target — — Target SettingValue — — SettingValue Execute — Done ParameterNumber — Busy — CommandAborted — Error — ErrorID	MC_Write_instance (Target := 《파라메터》, SettingValue := 《파라메터》, Execute := 《파라메터》, ParameterNumber := 《파라메터》, Done => 《파라메터》, Busy => 《파라메터》, CommandAborted => 《파라메터》, Error => 《파라메터》, ErrorID => 《파라메터》);

사용상의 주의 사항

이 명령으로 재기입한 값은 CPU 유니트의 비휘발성 메모리에 저장되지 않습니다. 따라서 컨트롤러의 전원 'OFF' 및 다운로드, MC 기능 모듈의 재기동 처리가 시작되면 재기입한 값은 사라지고 Sysmac Studio에서 설정한 값으로 되돌아갑니다.
비휘발성 메모리에 저장하려면 Sysmac Studio를 사용해 파라메터를 전송하십시오.

참고

'다운로드'는 Sysmac Studio의 '동기' 기능을 사용합니다.

변수

입력 변수

입력 변수	이 름	데이터 형식	유효 범위	초기값	내 용
Execute	기동	BOOL	TRUE, FALSE	FALSE	기동 시에 명령을 시작합니다.
Parameter Number	파라메터 번호	_eMC_ PARAMETER_ NUMBER	0: _mcChkVel 1: _mcChkAcc 2: _mcChkDec 3: _mcPosiChkTrq 4: _mcNegaChkTrq 5: _mcFELmt 6: _mcChkFELmt 7: _mcSwLmtMode 8: _mcPosiSwLmt 9: _mcNegaSwLmt 10: _mcInPosTime 11:_mcInPosRange*1	0 *2	쓰기대상 파라메터를 지정합니다. 0 : 속도 경고값/보간 속도 경고값 1 : 가속도 경고값/보간 가속도 경고값 2 : 감속도 경고값/보간 감속도 경고값 3 : 정방향 토크 경고값 4 : 역방향 토크 경고값 5 : 위치 편차 초과값 6 : 위치 편차 경고값 7 : 소프트웨어 리미트 기능 8 : 정방향 소프트웨어 리미트 9 : 역방향 소프트웨어 리미트 10 : 인포지션 확인 시간 11 : 인포지션 폭

*1. CPU 유니트 Ver.1.01 이상과 Sysmac Studio Ver.1.02 이상의 조합에서 지정할 수 있습니다.
*2. 유효 범위가 열거 형식인 변수의 초기값은 실제로는 숫자가 아니라 열거자가 됩니다.

● 파라메터의 데이터 형식과 유효 범위

파라메터	데이터 형식	유효 범위	비 고
0 : 속도 경고값/ 보간 속도 경고 값	UINT	0~100	단위는 [%]입니다.
1 : 가속도 경고값/ 보간 가속도 경고값	UINT	0~100	단위는 [%]입니다.
2 : 감속도 경고값/ 보간 감속도 경고값	UINT	0~100	단위는 [%]입니다.
3 : 정방향 토크 경고값	UINT	0~1000	단위는 [%]입니다.
4 : 역방향 토크 경고값	UINT	0~1000	단위는 [%]입니다.
5 : 위치 편차 초과값	LREAL	양수	단위는 [명령 단위]입니다. *1
6 : 위치 편차 경고값	LREAL	양수	위치 편차 초과값 이하로 합니다. 단위는 [명령 단위]입니다. *1
7 : 소프트웨어 리미트 기능	_eMC_ SWLMT_ MODE	0: _mcNonSwLmt 1: _mcCmdDecelerationStop 2: _mcCmdImmediateStop 3: _mcActDecelerationStop 4: _mcActImmediateStop	0: 소프트웨어 리미트 기능의 무효 1: 명령 위치에 대해 유효일 때 감속 정지 2: 명령 위치에 대해 유효일 때 즉시 정지(편차 펄스 정지) 3: 피드백 위치에 대해 유효일 때 감속 정지 4: 피드백 위치에 대해 유효일 때 즉시 정지(편차 펄스 정지)
8 : 정방향 소프트웨어 리미트	LREAL	음수, 양수, '0'	단위는 [명령 단위]입니다. *1
9 : 역방향 소프트웨어 리미트	LREAL	음수, 양수, '0'	단위는 [명령 단위]입니다. *1
10 : 인포지션 확인 시간	UINT	0~10000	단위는 [ms]입니다.
11 : 인포지션 폭	LREAL	양수, '0'	단위는 [명령 단위]입니다. *1

*1. 명령 단위에 대해서는, 🕮 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"의 '단위 변환 설정'을 참조하십시오.

출력 변수

출력 변수	이 름	데이터 형식	유효 범위	내 용
Done	완료	BOOL	TRUE, FALSE	명령 실행을 완료했을 때 TRUE가 됩니다.
Busy	실행 중	BOOL	TRUE, FALSE	명령을 수신했을 때 TRUE가 됩니다.
CommandAborted	실행 중단	BOOL	TRUE, FALSE	명령이 중지되었을 때 TRUE가 됩니다.
Error	에러	BOOL	TRUE, FALSE	이상이 발생했을 때 TRUE가 됩니다.
ErrorID	에러 코드	WORD	*	이상이 발생했을 때 에러 코드를 출력합니다. 16#0000은 정상입니다.

* 🕮 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

● 출력 변수의 반영 타이밍

변 수	TRUE가 되는 타이밍	FALSE가 되는 타이밍
Done	명령을 완료했을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Busy	Execute의 기동	• Done이 TRUE가 되었을 때 • Error가 TRUE가 되었을 때 • CommandAborted가 TRUE가 되었을 때
CommandAborted	다른 명령에서 이상이 발생해 이 명령이 중지되었을 때	• Execute가 TRUE인 경우 Execute가 FALSE와 동시 • Execute가 FALSE인 경우 1주기 후
Error	이 명령의 기동 조건이나 입력 파라메터에 이상 요인이 포함되어 있을 때	이상을 해제했을 때

입출력 변수

입출력 변수	이 름	데이터 형식	유효 범위	내 용
Target	쓰기 대상	_sAXIS_REF 또는 _sGROUP_REF	–	쓰기 대상이 되는 축 또는 축 그룹을 지정합니다. *1
SettingValue	설정값	지정한 변수 형식에 따릅니다. *2	–	기록할 값을 지정합니다. 유효 범위는 ParameterNumber(파라메터 번호)에서 지정한 모션 제어 파라메터 지정에 따릅니다. 초기값은 '0'입니다.

*1. Sysmac Studio의 축 기본 설정 화면에서 작성한 사용자 정의 변수의 축 변수명(기본값 'MC_Axis***') 또는 축 그룹 기본 설정 화면에서 작성한 사용자 정의 변수의 축 그룹 변수명(기본값 'MC_Group***')을 사용하십시오. 어느 경우든 시스템 정의 변수의 변수명(_MC_AX[**]나 _MC_GRP[**])도 사용할 수 있습니다.

*2. 변수의 데이터 형식에 대해서는, 📖 '파라메터의 데이터 형식과 유효 범위(5-14페이지)'를 참조하십시오.

● 입출력 변수의 반영 타이밍

입출력 변수	쓰기 타이밍
SettingValue	Done의 기동 시

기능 설명

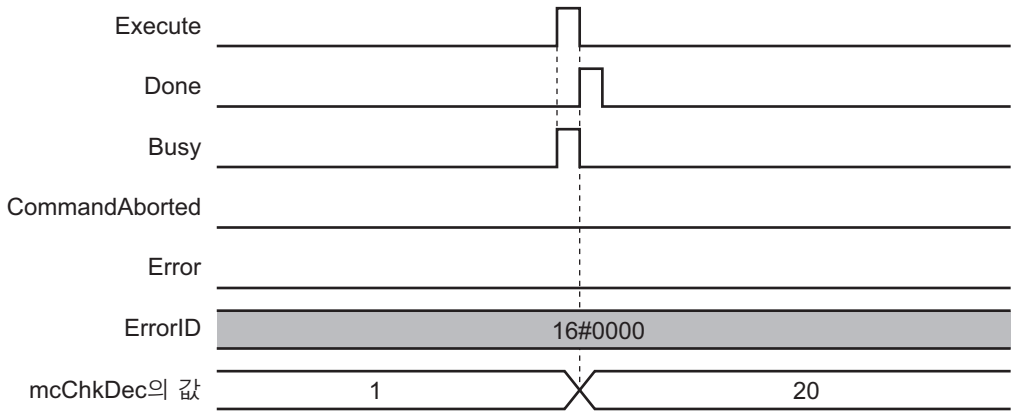
- Execute(기동)의 기동으로 Target(쓰기 대상) 및 ParameterNumber(파라메터 번호)에서 지정한 모션 제어 시스템 변수에 SettingValue(설정값)를 기록합니다.
- 입력 변수에서 지정된 파라메터는 축 변수 Status.Standstill(정지 중)이 TRUE일 때 또는 축 그룹 변수 Status.Standby(정지 중)가 TRUE일 때의 동작 시작 시와, Aborting이 지정된 경우의 다중 기동 시에 적용됩니다.
  따라서 재기동 및 Aborting이 지정된 이외의 경우에 다중 기동으로 동작을 계속시키면 적용되지 않습니다.

사용상의 주의 사항

이 명령으로 재기입한 값은 CPU 유니트의 비휘발성 메모리에 저장되지 않습니다. 따라서 컨트롤러의 전원 'OFF' 및 다운로드, MC 기능 모듈의 재기동 처리가 시작되면 재기입한 값은 사라지고 Sysmac Studio에서 설정한 값으로 되돌아갑니다.
비휘발성 메모리에 저장하려면 Sysmac Studio를 사용해 파라메터를 전송하십시오.

타이밍 차트

축 파라메터 설정의 _mcChkDec(감속도 경고값)에 데이터 '20'을 기록한 경우의 타이밍 차트는 다음과 같습니다.



모션 명령 재기동

이 명령의 Busy(실행 중)가 TRUE일 때) 동일한 인스턴스의 Execute(기동)의 기동이 있는 경우, 명령이 재기동됩니다. 이 때, 다음 Execute(기동)의 기동 시점의 Target(쓰기 대상)과 ParameterNumber(파라메터 번호), SettingValue(설정값)으로 재기입합니다.

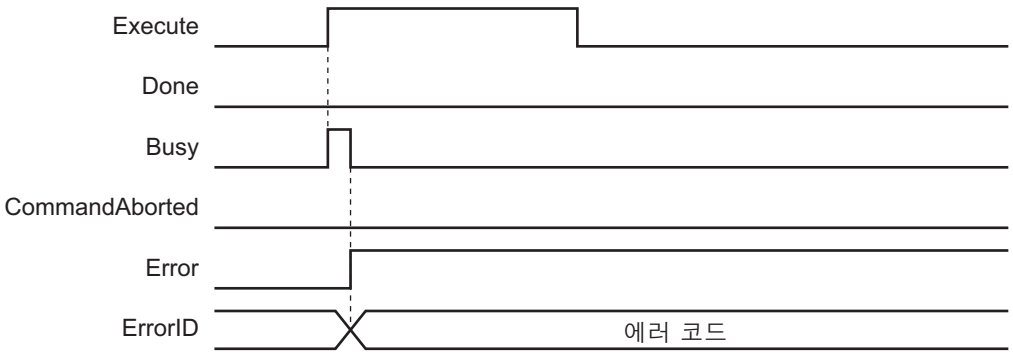
모션 명령 다중 기동

모션 명령 다중 기동에 대한 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

이상

이 명령 실행 시에 이상이 발생하면 Error(에러)가 TRUE가 되고 파라메터를 기록하지 않습니다. 명령 실행 전의 값이 저장됩니다.
ErrorID(에러 코드)에 출력된 값을 참조하면 이상의 원인을 알 수 있습니다.

● 이상 발생 시의 타이밍 차트



● 에러 코드

명령에서 발생하는 이상에 대해서는, 🕮 'A-1 에러 코드 목록(A-2페이지)'을 참조하십시오.

A

부록

명령에서 발생하는 에러 코드에 대해 설명합니다.

A-1 에러 코드 목록 A-2
A-2 에러 코드에 대한 자세한 내용 A-20
A-3 다중 기동의 가능 여부 A-78
A-4 버전 관련 정보 A-86

A-1 에러 코드 목록

여기에서는 명령어를 실행한 경우 등에 표시되는 ErrorID(에러 코드)의 목록표를 제공합니다.
다음 이벤트 코드의 상위 4자리가 ErrorID(에러 코드)입니다.
에러 코드에 대한 자세한 내용은 'A-2 에러 코드에 대한 자세한 내용(A-20페이지)'을 참조하십시오.

중요도 란에는 다음과 같은 약칭이 사용됩니다.

약칭	이름
전체	전체 정지 이상
부분	부분 정지 이상
가벼움	가벼운 이상
감시	감시 정보
일반	일반 정보

이벤트 코드란의 괄호 안에는 해당 이벤트 코드가 추가된 CPU 유니트의 유니트 버전이 표시됩니다.
NJ 시리즈의 모든 이벤트 코드에 대해서는 "NJ 시리즈 트러블 슈팅 설명서(SBCA-361)"를 참조하십시오.

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
34610000Hex	프로세스 데이터 오브젝트 설정 부족	PDO 매핑이 올바르지 않습니다.	• 모션 명령에서 필요한 PDO가 매핑되어 있지 않음 • 해당 명령에 적합한 개체가 없는 대상 기기에 대해 해당 명령을 실행함 • 오므론의 EtherCAT 엔코더 슬레이브 GX-EC02□□를 매핑한 축에 대해 트리거 조건으로 Z상(_mcEncoderMark)을 지정해서 모션 명령을 기동함			○			A-21 페이지
54200000Hex	전자 기어 분자 설정 범위 외	모션 제어 명령의 입력 변수 'RatioNumerator'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-21 페이지
54210000Hex	전자 기어 분모 설정 범위 외	모션 제어 명령의 입력 변수 'RatioDenominator'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-22 페이지
54220000Hex	목표 속도 설정 범위 외	모션 제어 명령의 입력 변수 'Velocity'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-22 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
54230000Hex	가속도 설정 범위 외	모션 제어 명령의 입력 변수 'Acceleration'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-22 페이지
54240000Hex	감속도 설정 범위 외	모션 제어 명령의 입력 변수 'Deceleration'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-23 페이지
54250000Hex	저크 설정 범위 외	모션 제어 명령의 입력 변수 'Jerk'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-23 페이지
54270000Hex	토크 경사 설정 범위 외	모션 제어 명령의 입력 변수 'TorqueRamp'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-23 페이지
54280000Hex	주축 계수 설정 범위 외	모션 제어 명령의 입력 변수 'MasterScaling'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-24 페이지
54290000Hex	종축 계수 설정 범위 외	모션 제어 명령의 입력 변수 'SlaveScaling'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-24 페이지
542A0000Hex	피딩 속도 설정 범위 외	모션 제어 명령의 입력 변수 'FeedVelocity'에 지정된 파라메터가 범위 밖입니다.	• 피딩 속도(입력 변수 'FeedVelocity')가 초기값(0) 그대로임			○			A-24 페이지
542B0000Hex	버퍼 모드 선택 범위 외	모션 제어 명령의 입력 변수 'BufferMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-25 페이지
542C0000Hex	좌표 계통 선택 범위 외	모션 제어 명령의 입력 변수 'CoordSystem'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-25 페이지
542D0000Hex	원호 보간 모드 선택 범위 외	모션 제어 명령의 입력 변수 'CircMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-25 페이지
542E0000Hex	방향 선택 범위 외	모션 제어 명령의 입력 변수 'Direction'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-26 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
542F0000Hex	경로 선택 범위 외	모션 제어 명령의 입력 변수 'PathChoice'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-26 페이지
54300000Hex	위치 종류 선택 범위 외	모션 제어 명령의 입력 변수 'ReferenceType'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-26 페이지
54310000Hex	이동 방법 선택 범위 외	모션 제어 명령의 입력 변수 'MoveMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-27 페이지
54320000Hex	트랜지션 모드 선택 범위 외	모션 제어 명령의 입력 변수 'TransitionMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외 • 'BufferMode'에서 '_mcAborting', '_mcBuffered'를 지정하고 'TransitionMode'를 '_mcTMCornerSuperimposed'로 지정			○			A-27 페이지
54330000Hex	계속 방법 선택 범위 외	모션 제어 명령의 입력 변수 'Continuous(Reserved)'의 값이 변경되었습니다.	• 입력 변수 'Continuous(Reserved)'의 값이 변경됨			○			A-28 페이지
54340000Hex	가감산 방법 선택 범위 외	모션 제어 명령의 입력 변수 'CombineMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-28 페이지
54350000Hex	동기 시작 조건 지정 범위 외	모션 제어 명령의 입력 변수 'LinkOption'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-28 페이지
54360000Hex	주축 종축 동일	모션 제어 명령의 입력 변수 'Master'와 'Slave'에 지정된 축이 동일합니다.	• 명령의 입력 변수 'Master'와 'Slave'의 파라메터가 동일			○			A-29 페이지
54370000Hex	주축 보조축 동일	모션 제어 명령의 입력 변수 'Master'와 'Auxiliary'에 지정된 축이 동일합니다.	• 명령의 입력 변수 'Master'와 'Auxiliary'의 파라메터가 동일			○			A-29 페이지
54380000Hex	주축/종축 축 번호 비오름차순	모션 제어 명령의 입력 변수 'Master'와 'Slave'에 지정된 축 번호가 오름차순이 아닙니다.	• 명령의 입력 변수 'ReferenceType'에서 '_mcLatestCommand' 지정 시 명령의 입력 변수 'Master'와 'Slave'의 파라메터가 비오름차순			○			A-29 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
54390000Hex	캠 테이블 지정 오류	모션 제어 명령의 입력 변수 'CamTable'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 변수 'CamTable'에 캠 데이터 변수 이외의 것을 지정			○			A-30 페이지
543A0000Hex	동기 정지 중	모션 제어의 동기 제어 명령이 실행되었지만 실행 가능한 조건이 아닙니다.	• MC_CamOut(캠 동작 해제) 명령이 실행되었지만 MC_CamIn(캠 동작 시작) 명령 실행 중이 아님 • MC_GearOut(기어 동작 해제) 명령이 실행되었지만 MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령 실행 중이 아님 • MC_Phasing(주축 상대값 위상 보정) 명령이 실행되었지만 MC_CamIn(캠 동작 시작) 명령, MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령이 실행 중이 아님			○			A-30 페이지
543B0000Hex	모션 명령 재기동 불가	재기동할 수 없는 모션 제어 명령의 재기동을 실행했습니다.	• 재기동할 수 없는 모션 제어 명령의 재기동			○			A-31 페이지
543C0000Hex	모션 명령 다중 기동 불가	동일한 대상(MC 공통/축/축 그룹)에 대해 동시에 실행할 수 없는 기능을 둘 이상 실행했습니다.	• 동일한 대상(MC 공통/축)에 대해 동시에 실행할 수 없는 기능을 둘 이상 실행			○			A-31 페이지
543D0000Hex	축 종류 부적합	엔코더 축에 대해 동작 명령을 실행했습니다.	• 엔코더 축에 대해 동작 명령을 실행			○			A-32 페이지
543E0000Hex	다축 협조 동작 중인 명령 기동 불가	다축 협조 동작 중인 축 또는 축 그룹에 대해 동작 명령을 실행했습니다.	• 다축 협조 동작 중인 축 또는 축 그룹에 대해 동작 명령을 실행			○			A-32 페이지
543F0000Hex	축 그룹 무효 상태에서 다축 협조 명령 기동	축 그룹 무효 상태인 축 그룹에 대해 다축 협조 명령을 기동했습니다.	• 축 그룹 무효 상태인 축 그룹에 대해 다축 협조 명령을 기동			○			A-32 페이지
54400000Hex	축 그룹 유효 불가	MC_GroupEnable(축 그룹 유효)명령을 실행하지 못했습니다.	• MC_GroupEnable(축 그룹 유효) 명령 실행 시 구성 축 중에 정지 중이 아닌 축이 존재함 • MC_GroupEnable(축 그룹 유효) 명령 실행 시 구성 축 중에 MC_TouchProbe(외부 래치 유효) 명령 실행 중인 축이 존재함			○			A-33 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
54410000Hex	운전 불가(서보 OFF) 축 동작 지시	서보 OFF 중인 축에 대해 동작 명령을 실행했습니다.	• 서보 OFF 중인 축에 대해 동작 명령을 실행 • MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령에 의한 원점 사전 설정을 EtherCAT의 프로세스 데이터 통신이 확립되지 않은 축에 대해 실행			○			A-33 페이지
54420000Hex	구성 축 강제 정지 중 에러	구성 축에 대해 MC_Stop(강제 정지) 명령이 실행되고 있는 축 그룹에 동작 명령을 실행했습니다.	• 구성 축에 대해 MC_Stop(강제 정지) 명령이 실행되고 있는 축 그룹에 동작 명령을 실행			○			A-34 페이지
54430000Hex	모션 명령 다중 기동 수 초과	버퍼 모드 Buffered, Blending에 의해 버퍼링되는 모션 제어 명령의 버퍼링 수가 한계를 초과했습니다.	• 축 명령에서 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '2'를 초과함 • 축 그룹 명령에서 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '8'을 초과함			○			A-34 페이지
54440000Hex	이동량 부족	위치 결정 명령의 다중 기동/재기동 시에 지정된 감속도 또는 가속도대로 동작을 실행할 수 없습니다.	• '가감속 초과'를 '이상 시 정지'로 설정했을 때 위치 결정 명령의 다중 기동/재기동 시에 지정된 감속도/가속도로 목표 위치에서 정지할 수 없음			○			A-35 페이지
54450000Hex	Blending 중계 속도에 도달하기 위한 이동량 부족	중계 속도로 가감속하기 위한 이동량이 부족합니다.	• '가감속 초과'에 '이상 시 정지'가 설정되어 있을 때 현재 명령을 중계 속도로 가감속하기 위한 이동량이 부족			○			A-35 페이지
54460000Hex	사다리꼴 패턴 캠 등속 이동량 부족	주축의 등속 이동량이 '0' 미만입니다.	• MC_MoveLink(사다리꼴 패턴 캠) 명령에서 주축의 등속 이동량이 '0' 미만			○			A-36 페이지
54470000Hex	위치 지정 기어 동작 목표 속도 부족	MC_GearInPos(위치 지정 기어 동작) 명령에서 종축의 '목표 속도'가 낮아 필요한 속도에 도달할 수 없습니다.	• MC_GearInPos(위치 지정 기어 동작) 명령에서 입력 변수 'Velocity(목표 속도)'의 값이 (명령 기동 시의 주축 속도×기어비)보다 작음			○			A-36 페이지
54480000Hex	원호 보간 시작점 종료점 동일	MC_MoveCircular2D(2축 원호 보간) 명령에서 반지름 지정 방식을 시작점과 종료점이 동일한 위치로 지정했습니다. 또는 통과점 지정 방식을 시작점과 종료점과 통과점이 동일한 위치로 지정했습니다.	• MC_MoveCircular2D(2축 원호 보간) 명령에서 반지름 지정 방식이 지정되고 시작점과 종료점의 위치가 동일 • MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식이 지정되고 시작점과 종료점이 동일한 위치로 지정			○			A-36 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
54490000Hex	원호 보간 중심점 지정 위치 범위 외	MC_MoveCircular2D(2축 원호 보간) 명령에서 중심점 지정 방식 지정 시에 중심점의 위치를 허용 범위를 넘어서 지정했습니다.	• MC_MoveCircular2D(2축 원호 보간) 명령에서 중심점 지정 방식이 지정되고, 시작점과 중심점 거리와 종료점과 중심점 거리의 차이가 축 그룹 설정의 '중심점 보정 허용률'에서 지정한 허용 범위를 초과함			○			A-37 페이지
544A0000Hex	카운트 모드 설정에 의한 명령 기동 이상	카운트 모드가 로터리 모드로 설정된 축에 대해 로터리 모드에서 사용할 수 없는 명령을 실행했습니다.	• 카운트 모드가 로터리 모드로 설정된 축을 로터리 모드에서 사용할 수 없는 명령에서 사용			○			A-37 페이지
544C0000Hex	파라메터 선택 범위 외	모션 제어 명령의 입력 변수 'ParameterNumber'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-37 페이지
544D0000Hex	정지 방법 선택 범위 외	모션 제어 명령의 입력 변수 'StopMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-38 페이지
544E0000Hex	트리거 입력 조건의 래치 ID 선택 범위 외	모션 제어 명령의 입력 변수 'TriggerInput::LatchID'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-38 페이지
544F0000Hex	MC 설정 쓰기의 설정 범위 외	모션 제어 명령의 입력 변수 'SettingValue'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외 • 파라메터 지정과 설정값의 데이터 형식이 일치하지 않음			○			A-38 페이지
54500000Hex	트리거 입력 조건의 모드 선택 범위 외	모션 제어 명령의 입력 변수 'TriggerInput::Mode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-39 페이지
54510000Hex	트리거 입력 조건의 드라이브 트리거 입력 신호 선택 범위 외	모션 제어 명령의 입력 변수 'TriggerInput::InputDrive'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-39 페이지
54530000Hex	모션 명령 재기동 불가(축 지정)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Axis'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-39 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
54540000Hex	모션 명령 재기동 불가(버퍼 모드 선택)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'BufferMode'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-40 페이지
54550000Hex	모션 명령 재기동 불가(방향 선택)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Direction'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-40 페이지
54560000Hex	모션 명령 재기동 불가(반복 모드)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Periodic'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-41 페이지
54570000Hex	모션 명령 재기동 불가(축 그룹 지정)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'AxesGroup'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-41 페이지
54580000Hex	모션 명령 재기동 불가(약동 설정)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Jerk'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-42 페이지
54590000Hex	모션 명령 재기동 불가(주축)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Master'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-42 페이지
545A0000Hex	모션 명령 재기동 불가(Master Offset)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'MasterOffset'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-43 페이지
545B0000Hex	모션 명령 재기동 불가(Master Scaling)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'MasterScaling'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-43 페이지
545C0000Hex	모션 명령 재기동 불가 (Master Start Distance)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'MasterStart Distance'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-43 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
545D0000Hex	모션 명령 재기동 불가 (Continuous)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Continuous'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-44 페이지
545E0000Hex	모션 명령 재기동 불가(Move Mode)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'MoveMode'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-44 페이지
545F0000Hex	보조축 지정 오류	모션 제어 명령의 입력 변수 'Auxiliary'에 지정된 축이 존재하지 않습니다.	• 명령의 입력 변수 'Auxiliary'에 지정된 축이 존재하지 않는 변수			○			A-44 페이지
54600000Hex	축 지정 오류	모션 제어 명령의 입력 변수 'Axis'에 지정된 축이 존재하지 않습니다.	• 명령의 입력 변수 'Axis'에 지정된 축이 존재하지 않는 변수			○			A-45 페이지
54610000Hex	축 그룹 지정 오류	모션 제어 명령의 입력 변수 'AxesGroup'에 지정된 축 그룹이 존재하지 않거나 사용 축 그룹이 아닙니다.	• 명령의 'AxesGroup'에 지정된 축 그룹이 존재하지 않는 변수 • 명령의 'AxesGroup'에 지정된 축 그룹이 사용 축 그룹으로 설정되지 않음			○			A-45 페이지
54620000Hex	주축 지정 오류	모션 제어 명령의 입력 변수 'Master'에 지정된 축이 존재하지 않거나 동기 주축이 아닙니다.	• 명령의 입력 변수 'Master'에 지정된 축이 존재하지 않는 변수 • MC_Phasing(주축 상대값 위상 보정) 명령에서 입력 변수 'Master'에 지정된 축이 동기 주축이 아님			○			A-45 페이지
54630000Hex	모션 명령 재기동 불가(Slave Offset)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'SlaveOffset'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-46 페이지
54640000Hex	모션 명령 재기동 불가(Slave Scaling)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'SlaveScaling'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-46 페이지
54650000Hex	모션 명령 재기동 불가(Start Position)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'StartPosition'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-46 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
54660000Hex	원점 미확정 상태에서의 명령 기동 이상	원점 미확정 상태에서 고속 원점 복귀 또는 보간 명령이 실행되었습니다.	• 원점 미확정 상태에서 고속 원점 복귀가 실행됨 • 원점 미확정 상태를 포함한 구성 축에 대해 보간 명령이 실행됨			○			A-47 페이지
54670000Hex	모션 명령 재기동 불가(위치 종류)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'ReferenceType'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-47 페이지
54680000Hex	미사용 축 지정(주축)	모션 제어 명령에서 지정된 주축이 사용하지 않는 축입니다.	• 모션 제어 명령에서 지정된 주축이 사용하지 않는 축			○			A-48 페이지
54690000Hex	시작 위치 설정 범위 외	모션 제어 명령의 입력 변수 'FirstPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-48 페이지
546A0000Hex	종료 위치 설정 범위 외	모션 제어 명령의 입력 변수 'LastPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-48 페이지
546B0000Hex	시작 위치/종료 위치 · 대소 관계 오류 (리니어 모드)	모션 제어 명령의 입력 변수 'LastPosition'에 지정된 파라메터가 입력 변수 'FirstPosition'에 지정된 파라메터보다 작은 값입니다.	• 카운트 모드가 리니어 모드일 때 명령의 입력 파라메터에서 'LastPosition'의 값이 'FirstPosition'의 값보다 작음			○			A-49 페이지
546C0000Hex	주축 동기 위치 설정 범위 외	모션 제어 명령의 입력 변수 'MasterSyncPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-49 페이지
546D0000Hex	종축 동기 위치 설정 범위 외	모션 제어 명령의 입력 변수 'SlaveSyncPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-49 페이지
546E0000Hex	트리거 입력 조건의 래치 ID 중복	모션 제어 명령에서 지정된 래치 ID가 둘 이상의 명령에서 중복됩니다.	• MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령에서 동일한 래치 ID가 동시에 사용됨 • MC_TouchProbe(외부 래치 유효) 이외의 명령에서 사용 중인 래치를 MC_AbortTrigger(외부 래치 무효) 명령에서 중지하려고 함			○			A-50 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
546F0000Hex	저크 오버라이드값 범위 외	모션 제어 명령의 입력 변수 'JerkFactor'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-50 페이지
54700000Hex	가감속 속도 오버라이드값 범위 외	모션 제어 명령의 입력 변수 'AccFactor'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-51 페이지
54710000Hex	시작 위치 방식 지정 범위 외	모션 제어 명령의 입력 변수 'StartMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-51 페이지
54720000Hex	모션 명령 재기동 불가(시작 위치 방식)	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'StartMode'의 파라메터를 변경했습니다.	• 명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함			○			A-51 페이지
54740000Hex	미사용 축 지정 (보조축)	모션 제어 명령의 입력 변수 'Auxiliary'에 지정된 축이 사용하지 않는 축입니다.	• 명령의 'Auxiliary'에 지정된 축이 사용하지 않는 축			○			A-52 페이지
54750000Hex	위치 지정 기어 지정값 이상	모션 제어 명령에 입력된 속도/가속도/감속도로는 동기 동작할 수 없습니다.	• 명령에 입력된 속도/가속도/감속도로는 지정된 동기 동작을 실행할 수 없음			○			A-52 페이지
54760000Hex	위치 지정 기어 주축 속도 제로	모션 제어 명령 기동 시에 주축의 속도가 '0'입니다.	• 명령 기동 시에 주축의 속도가 '0'			○			A-52 페이지
54780000Hex	목표 위치 설정 범위 외	모션 제어 명령의 입력 변수 'Position'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외 • 로터리 모드인 축에서 목표 위치가 링 설정 범위 외			○			A-53 페이지
54790000Hex	이동 거리 범위 외	모션 제어 명령의 입력 변수 'Distance'에 지정된 파라메터가 범위 밖이거나 'Distance'의 값을 가산한 목표 위치가 범위 밖입니다.	• 명령 입력 파라메터의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함 • 리니어 모드인 축에서 이동 거리를 가산한 목표 위치를 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함			○			A-53 페이지
547A0000Hex	캠 테이블 시작점 위치 설정 범위 외	모션 제어 명령의 입력 변수 'StartPosition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-54 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
547B0000Hex	캠 동작(주축 추종) 시작 위치 설정 범위 외	모션 제어 명령의 입력 변수 'MasterStart Distance'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-54 페이지
547C0000Hex	원호 보간 반지름 지정 이상	MC_MoveCircular2D(2축 원호 보간) 명령에서 반지름 지정 방식 지정 시에 지정된 반지름으로 원호의 궤적을 작성할 수 없습니다.	• MC_MoveCircular2D(2축 원호 보간) 명령에서 반지름 지정 방식이 지정되고 지정된 반지름으로는 원호의 궤적을 작성할 수 없음			○			A-54 페이지
547D0000Hex	원호 보간 반지름 오버 플로우	MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식/중심점 지정 방식일 때 원호의 반지름이 최대값을 초과했습니다.	• MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식/중심점 지정 방식인 경우 원호의 반지름을 펄스 단위로 변환했을 때 40비트의 범위를 초과함			○			A-55 페이지
547E0000Hex	원호축 지정 범위 외	모션 제어 명령의 입력 변수 'CircAxes'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외 • 'CircAxes'에 지정된 축이 축 그룹 설정의 구성 축에 포함되지 않음 • 'CircAxes'의 2축에 동일 축이 지정됨			○			A-55 페이지
547F0000Hex	보조축/종축 · 축 번호 비오름차순	모션 제어 명령에서 지정된 입력 변수 'Auxiliary'와 'Slave'의 파라메터 값이 오름차순으로 나열되어 있지 않습니다.	• 명령의 입력 변수 'Auxiliary'와 'Slave'의 파라메터가 오름차순이 아님			○			A-56 페이지
54800000Hex	캠 테이블 속성 갱신 중 데이터 오름차순 이상	유효 데이터 수 계산 중에 위상이 오름차순이 아님이 확인되었습니다. 또는 계산 결과 유효 데이터 수가 '0'임이 확인되었습니다.	• 유효 데이터 수 계산 중에 위상이 오름차순이 아님이 확인됨 • 계산 결과 유효 데이터 수가 '0'임이 확인됨			○			A-56 페이지
54810000Hex	MC 설정 쓰기의 대상 범위 외	모션 제어 명령의 입력 변수 'Target'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-56 페이지
54820000Hex	주축 이동 거리 지정 범위 외	모션 제어 명령의 입력 변수 'MasterDistance'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-57 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
54830000Hex	주축 가속 이동 거리 지정 범위 외	모션 제어 명령의 입력 변수 'MasterDistanceInACC'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-57 페이지
54840000Hex	주축 감속 이동 거리 지정 범위 외	모션 제어 명령의 입력 변수 'MasterDistanceInDEC'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-57 페이지
54870000Hex	실행 모드 선택 범위 외	모션 제어 명령의 입력 변수 'ExecutionMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-58 페이지
54880000Hex	축간 편차 허용값 범위 외	모션 제어 명령의 입력 변수 'PermittedDeviation'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-58 페이지
54890000Hex	통과점 위치/중심 위치/반지름 지정 범위 외	모션 제어 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 범위 밖입니다.	• 통과점 또는 중심으로 지정된 경우 'AuxPoint'의 값을 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함 • 반지름으로 지정된 경우 'AuxPoint[0]'의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함			○			A-58 페이지
548A0000Hex	종료점 지정 범위 외	모션 제어 명령의 입력 변수 'EndPoint'에 지정된 파라메터가 범위 밖입니다.	• 명령 입력 파라메터를 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함			○			A-59 페이지
548B0000Hex	종축 이동 거리 지정 범위 외	모션 제어 명령의 입력 변수 'SlaveDistance'에 지정된 파라메터가 범위 밖입니다.	• 명령 입력 파라메터를 펄스 단위로 변환했을 때 40비트의 범위를 초과함			○			A-59 페이지
548C0000Hex	위상 보정량 범위 외	모션 제어 명령의 입력 변수 'PhaseShift'에 지정된 파라메터가 범위 밖입니다.	• 명령 입력 파라메터의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함			○			A-59 페이지
548D0000Hex	피딩 거리 범위 외	모션 제어 명령의 입력 변수 'FeedDistance'에 지정된 파라메터가 범위 밖입니다.	• 명령 입력 파라메터의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함			○			A-60 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
548E0000Hex	보조축/종축 동일	모션 제어 명령의 입력 변수 'Auxiliary'와 'Slave'에 지정된 축이 동일합니다.	• 명령의 입력 변수 'Auxiliary'와 'Slave'의 파라메터가 동일			○			A-60 페이지
548F0000Hex	상대 위치 선택 범위 외	모션 제어 명령의 입력 변수 'Relative'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-60 페이지
54900000Hex	캠 트랜지션 지정 선택 범위 외	모션 제어 명령의 입력 변수 'CamTransition'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-61 페이지
54910000Hex	동기 제어 해제 모드 선택 범위 외	모션 제어 명령의 입력 변수 'OutMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-61 페이지
54920000Hex	외부 래치 유효 명령 실행 불가	CSP 이외의 제어 모드일 때 입력 변수 'StopMode'에 '_mcImmediateStop(즉시 정지)'을 지정해서 드라이버 모드의 MC_TouchProbe(외부 래치 유효) 명령을 실행했습니다.	• CSP 이외의 제어 모드이거나 서보 OFF 시에 입력 변수 'StopMode'에 '_mcImmediateStop(즉시 정지)'을 지정해서 드라이버 모드의 MC_TouchProbe(외부 래치 유효) 명령을 실행함			○			A-61 페이지
54930000Hex	주축 오프셋 설정 범위 외	모션 제어 명령의 입력 변수 'MasterOffset'에 지정된 파라메터가 범위 밖입니다.	• 명령 입력 파라메터를 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함			○			A-62 페이지
54940000Hex	종축 오프셋 설정 범위 외	모션 제어 명령의 입력 변수 'SlaveOffset'에 지정된 파라메터가 범위 밖입니다.	• 명령 입력 파라메터를 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함			○			A-62 페이지
54950000Hex	명령 현재 위치 카운트 선택 범위 외	모션 제어 명령의 입력 변수 'CmdPosMode'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-62 페이지
54960000Hex	주축 기어비 분자 범위 외	모션 제어 명령의 입력 변수 'RatioNumeratorMaster'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-63 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
54970000Hex	주축 기어비 분모 범위 외	모션 제어 명령의 입력 변수 'RatioDenominatorMaster'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-63 페이지
54980000Hex	보조축 기어비 분자 범위 외	모션 제어 명령의 입력 변수 'RatioNumeratorAuxiliary'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-63 페이지
54990000Hex	보조축 기어비 분모 범위 외	모션 제어 명령의 입력 변수 'RatioDenominatorAuxiliary'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-64 페이지
549A0000Hex	주축 위치 종류 선택 범위 외	모션 제어 명령의 입력 변수 'ReferenceTypeMaster'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-64 페이지
549B0000Hex	보조축 위치 종류 선택 범위 외	모션 제어 명령의 입력 변수 'ReferenceTypeAuxiliary'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수의 범위 외			○			A-64 페이지
549C0000Hex	목표 위치 링 카운터 범위 외	실행된 명령의 목표 위치가 링 카운터의 범위 밖이므로 동작할 수 없습니다.	• 링 카운터 범위에 0이 포함되지 않도록 설정되었을 때 고속 원점 복귀를 실행함			○			A-65 페이지
549D0000Hex (Ver.1.01)	축 그룹 구성축 설정 범위 외	모션 제어 명령의 입력 변수 'Axes'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외			○			A-65 페이지
549E0000Hex (Ver.1.04)	축 사용 설정 범위 외	모션 제어 명령의 입력 변수 'AxisUse'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 밖			○			A-66 페이지
57000000Hex (Ver.1.03)	원점 복귀 파라메터 설정 범위 외	모션 제어 명령의 입력 변수 'HomingParameter'에 지정된 파라메터가 범위 밖입니다.	• 명령의 입력 파라메터가 입력 변수 범위 외			○			A-66 페이지
57020000Hex (Ver.1.04)	축 미사용 전환 이상	정지 중이 아니거나 명령 속도 포화 상태인 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령을 실행했습니다.	• 정지 중이 아니거나 명령 속도 포화 상태인 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령을 실행함			○			A-67 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
57030000Hex (Ver.1.06)	축 사용 전환 불가	사용 실제 축 최대 수를 초과하도록 MC_ChangeAxisUse(축 사용 변경) 명령을 실행했습니다.	• 사용 실제 축 최대 수를 초과하도록 MC_ChangeAxisUse(축 사용 변경) 명령을 실행함			○			A-67 페이지
57200000Hex (Ver.1.04)	축 사용 전환 시 모션 제어 파라메터 설정 이상	사용 축으로 전환한 축의 모션 제어 파라메터 설정이 잘못되었습니다.	• MC_ChangeAxisUse(축 사용 변경) 명령을 사용하여 미사용 축에서 사용 축으로 전환한 축의 모션 제어 파라메터 설정이 잘못됨 • 모션 제어 파라메터 설정 다운로드 중에 전원 OFF됨 • 비휘발성 메모리의 고장 또는 비휘발성 메모리의 수명이 다함			○			A-68 페이지
57210000Hex (Ver.1.04)	축 사용 전환 시 필수 프로세스 데이터 개체 미설정	사용 축으로 전환한 축의 축 종류에 대한 필수 개체가 설정되어 있지 않습니다.	• 사용 축으로 전환한 축의 축 종류에 대한 필수 개체가 PDO 매핑 설정에 설정되어 있지 않음 • 모션 제어 파라메터 설정 다운로드 중에 전원 OFF됨 • 비휘발성 메모리의 고장 또는 비휘발성 메모리의 수명이 다함 • [축 사용]이 [사용하지 않는 축(사용 축으로 전환 불가)]으로 설정되어 있는 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령이 실행됨			○			A-69 페이지
572F0000Hex (Ver.1.06)	모션 명령 다중 기동 불가능 (주축)	명령 다중 기동 시 변경할 수 없는 입출력 변수 'Master'를 변경했습니다.	• '명령 다중 기동 시 변경할 수 없는 입출력 변수 'Master'를 변경			○			A-69 페이지
57300000Hex (Ver.1.06)	모션 명령 다중 기동 불가능 (위치 종류 선택)	명령 다중 기동 시 변경할 수 없는 입출력 변수 'ReferenceType'을 변경했습니다.	• 명령 다중 기동 시 변경할 수 없는 입출력 변수 'ReferenceType'을 변경			○			A-70 페이지
64400000Hex	목표 위치 정방향 소프트웨어 리미트 초과	지정한 위치가 정방향 소프트웨어 리미트의 범위 밖입니다.	• 명령의 입력 변수 'Position'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 외 • 시작 위치가 정방향 소프트웨어 리미트의 범위 밖이고 소프트웨어 리미트의 범위와는 반대 방향으로 동작을 지정해 명령을 실행 • 통과점이 지정된 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 외			○			A-70 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
64410000Hex	목표 위치 역방향 소프트웨어 리미트 초과	지정한 위치가 역방향 소프트웨어 리미트의 범위 밖입니다.	• 명령의 입력 변수 'Position'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 외 • 시작 위치가 역방향 소프트웨어 리미트의 범위 밖이고 소프트웨어 리미트의 범위와는 반대 방향으로 동작을 지정해 명령을 실행 • 통과점이 지정된 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 외			○			A-71 페이지
64420000Hex	명령 위치 오버 플로우/언더 플로우 발생 중	명령 위치 오버 플로우/언더 플로우 발생 중에 위치 결정, 오버 플로우/언더 플로우 방향에 대한 명령 또는 방향을 확정할 수 없는 명령을 실행했습니다.	• 명령 위치 오버 플로우/언더 플로우 발생 중에 다음을 실행 • 위치 결정 명령을 실행 • 오버 플로우/언더 플로우 방향으로 연속 제어 명령을 실행 • 방향을 확정할 수 없는 명령(동기 기능, 토크 제어)을 실행			○			A-71 페이지
64430000Hex	정방향 한계 입력 중	정방향 한계 입력이 'ON'인 상태에서 정방향으로 동작하는 명령을 실행했습니다.	• 정방향 한계 입력이 'ON'인 상태에서 정방향으로 동작하는 명령을 실행했거나 정방향 한계 입력이 'ON'인 상태에서 동작 방향을 지정하지 않는 명령을 실행. 정방향 한계 입력이 'ON'인 상태에서 축 그룹 동작 명령을 실행			○			A-72 페이지
64440000Hex	역방향 한계 입력 중	역방향 한계 입력이 'ON'인 상태에서 역방향으로 동작하는 명령을 실행했습니다.	• 역방향 한계 입력이 'ON'인 상태에서 역방향으로 동작하는 명령을 실행했거나 역방향 한계 입력이 'ON'인 상태에서 동작 방향을 지정하지 않는 명령을 실행. 역방향 한계 입력이 'ON'인 상태에서 축 그룹 동작 명령을 실행			○			A-72 페이지
74220000Hex	서보 주회로 전원 OFF 상태	서보 드라이버의 주회로 전원이 'OFF'인 상태에서 서보 ON을 실행했습니다.	• 서보 드라이버의 주회로 전원이 'OFF'인 상태에서 서보 ON을 실행			○			A-73 페이지
57220000Hex (Ver.1.06)	피드백 위치 오버 플로우/언더 플로우 발생 중	피드백 위치 오버 플로우/언더 플로우 발생 중에 실행할 수 없는 명령을 기동했습니다.	• 피드백 위치 오버 플로우 또는 언더 플로우 발생 중에 실행할 수 없는 명령을 기동함				○		A-73 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
57230000Hex (Ver.1.06)	스위치 구조체의 트랙 번호 설정 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'TrackNumber'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		A-73 페이지
57240000Hex (Ver.1.06)	스위치 구조체의 ON 시작 위치 설정 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'FirstOnPosition'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		A-74 페이지
57250000Hex (Ver.1.06)	스위치 구조체의 ON 종료 위치 설정 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'LastOnPosition'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		A-74 페이지
57260000Hex (Ver.1.06)	스위치 구조체의 방향 선택 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'AxisDirection'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		A-74 페이지
57270000Hex (Ver.1.06)	스위치 구조체의 스위치 모드 선택 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'CamSwitchMode'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		A-75 페이지
57280000Hex (Ver.1.06)	스위치 구조체의 ON 시간 설정 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'Duration'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		A-75 페이지
57290000Hex (Ver.1.06)	트랙 옵션 구조체의 ON 시각 보정 설정 범위 밖	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 'OnCompensation'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		A-75 페이지
572A0000Hex (Ver.1.06)	트랙 옵션 구조체의 OFF 시각 보정 설정 범위 밖	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 'OffCompensation'의 값이 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		A-76 페이지
572B0000Hex (Ver.1.06)	스위치 구조체 형식 변수의 배열 요소 수 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖				○		A-76 페이지

이벤트 코드	이벤트명	내용	발생 요인(추정 원인)	중요도					참조 대상
				전체	부분	가벼움	감시	일반	
572C0000Hex (Ver.1.06)	출력 신호 구조체 형식 변수의 배열 요소 수 범위 밖	모션 제어 명령의 입출력 변수 'Outputs'에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위 밖				○		A-76 페이지
572D0000Hex (Ver.1.06)	트랙 옵션 구조체 형식 변수의 배열 요소 수 범위 밖	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다.	• 명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위 밖				○		A-77 페이지
572E0000Hex (Ver.1.06)	출력 신호와 트랙 옵션의 배열 요소 수 불일치	모션 제어 명령의 입출력 변수 'Outputs'와 'TrackOptions'에 지정된 구조체 형식 변수의 배열 요소 수가 일치하지 않습니다.	• 명령의 입출력 변수로 지정한 출력 신호 구조체 형식 변수와 트랙 옵션 구조체 형식 변수의 배열 요소 수가 일치하지 않음				○		A-77 페이지
57310000Hex (Ver.1.06)	스위치 구조체의 동일 트랙 지정 수 범위 밖	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'TrackNumber'의 동일 트랙 번호 수가 범위 밖입니다.	• 명령의 입출력 변수로 지정한 스위치 구조체 형식 변수의 'TrackNumber'에 지정된 동일 트랙 번호의 수가 1트랙에 지정 가능한 개수의 범위 밖				○		A-77 페이지

A-2 에러 코드에 대한 자세한 내용

컨트롤러 이상에 대한 설명 보는 방법

각 이상에 대한 설명에서 사용하는 표의 각 항목의 의미를 [ ]안에 표시합니다.

<table>
<tr><td>이벤트명</td><td colspan="2">[이상(이벤트)의 명칭입니다.]</td><td>이벤트 코드</td><td colspan="2">[이상(이벤트)의 코드입니다.]</td></tr>
<tr><td>내용</td><td colspan="5">[이상(이벤트)의 내용입니다.]</td></tr>
<tr><td>발생원</td><td colspan="2">[이상(이벤트)이 발생한 부분입니다.]</td><td>발생원 상세 정보</td><td>[발생원에 대한 자세한 내용입니다.]</td><td>검출 타이밍</td><td>[이상 검출의 타이밍입니다.]</td></tr>
<tr><td rowspan="2">이상의 속성</td><td>중요도</td><td>[제어에 대한 영향의 정도입니다.] *1</td><td>복귀 방법</td><td>[복귀 방법입니다.] *2</td><td>로그 카테고리</td><td>[저장되는 로그의 종류입니다.] *3</td></tr>
<tr><td>발생의 영향</td><td>사용자 프로그램</td><td>[사용자 프로그램의 실행 상태입니다.] *4</td><td>동작</td><td colspan="2">[이상(이벤트) 발생 시의 동작에 관한 특기 사항입니다.]</td></tr>
<tr><td>LED</td><td colspan="6">[내장 EtherNet/IP 포트용 LED, 내장 EtherCAT 포트용 LED의 표시 상태입니다. 발생원이 EtherCAT 마스터 기능 모듈, EtherNet/IP 기능 모듈일 때만 기재합니다.]</td></tr>
<tr><td rowspan="2">시스템 정의 변수</td><td colspan="2">변수명</td><td colspan="2">데이터 형식</td><td colspan="2">명칭</td></tr>
<tr><td colspan="6">[이상을 감지하는 시스템 정의 변수, 이상의 영향을 받는 시스템 정의 변수, 이상의 원인이 되는 시스템 정의 변수의 변수명과 데이터 형식, 명칭입니다.]</td></tr>
<tr><td rowspan="2">발생 요인과 처리 방법</td><td colspan="2">발생 요인(추정 원인)</td><td colspan="2">처리 대책</td><td colspan="2">재발 방지</td></tr>
<tr><td colspan="6">[이상(이벤트) 발생 요인과 처리 대책, 재발 방지 방법입니다.]</td></tr>
<tr><td>부속 정보</td><td colspan="6">[Sysmac Studio/NS 시리즈 표시기에서 표시되는 부속 정보 내용입니다.]</td></tr>
<tr><td>주의 사항 /비고</td><td colspan="6">[그 밖의 주의 사항, 제약 사항, 보충 설명 등입니다.]</td></tr>
</table>

*1. 다음 중 하나에 해당합니다.
전체 정지 이상: 전체 정지 이상 수준
부분 정지 이상: 부분 정지 이상 수준
가벼운 이상: 가벼운 이상 수준
감시 정보
일반 정보

*2. 다음 중 하나에 해당합니다.
자동 복귀: 요인이 제거되었을 때 자동으로 정상 복귀합니다.
이상 해제: 요인을 제거한 후 이상 해제를 실행하면 정상으로 돌아갑니다.
전원 다시 켜기: 요인을 제거한 후 전원을 다시 켜면 정상으로 돌아갑니다.
컨트롤러 리셋: 요인을 제거한 후 컨트롤러를 리셋하면 정상으로 돌아갑니다.
발생 요인에 따름: 발생 요인에 따릅니다.

*3. 다음 중 하나에 해당합니다.
시스템 로그: 시스템 이벤트 로그
액세스 로그: 액세스 이벤트 로그

*4. 다음 중 하나에 해당합니다.
계속: 사용자 프로그램의 실행을 계속합니다.
정지: 사용자 프로그램의 실행을 정지합니다.
시작: 사용자 프로그램의 실행을 시작합니다.

이상 설명

명령어를 실행한 경우 등에 표시되는 ErrorID(에러 코드)의 자세한 내용을 제공합니다.
다음 이벤트 코드의 상위 4자리가 ErrorID(에러 코드)입니다.

이벤트명	프로세스 데이터 오브젝트 설정 부족			**이벤트 코드**	34610000Hex	
내용	PDO 매핑이 올바르지 않습니다.					
발생원	모션 제어 기능 모듈		**발생원 상세 정보**	축	**검출 타이밍**	명령 기동 시
이상의 속성	**중요도**	가벼운 이상	**복귀 방법**	이상 해제	**로그 카테고리**	시스템
발생의 영향	**사용자 프로그램**	계속	**동작**	해당 축 동작 불가		
시스템 정의 변수	**변수명**		**데이터 형식**		**명칭**	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	**발생 요인(추정 원인)**		**처리 대책**		**재발 방지**	
	모션 명령에서 필요한 PDO가 매핑되어 있지 않다.		해당 명령에서 필요한 PDO를 매핑하십시오. 필요한 PDO 매핑에 대해서는 해당 명령의 '기능 설명'을 참조하십시오.		사용하는 명령에서 필요한 PDO를 매핑하십시오. 각 명령에서 필요한 PDO 매핑(서보 드라이버의 설정)에 대해서는 "NJ 시리즈 사용 설명서 모션 제어편"을 참조하십시오.	
	해당 명령에 적합한 개체가 없는 대상 기기에 대해 해당 명령을 실행함		해당 명령에 적합한 기능이 없는 대상 기기가 있습니다. 대상 기기의 설명서를 참조하여 해당 명령에 적합한지 확인하고 적합하지 않은 명령을 실행하지 않도록 프로그램을 수정하십시오.		대상 기기의 설명서를 참조하여 적합하지 않은 명령을 실행하지 않도록 프로그램을 작성하십시오.	
	오므론의 EtherCAT 엔코더 슬레이브 GX-EC02□□를 매핑한 축에 대해 트리거 조건으로 Z상(_mcEncoderMark)을 지정해서 모션 명령을 기동했다.		오므론의 EtherCAT 엔코더 슬레이브 GX-EC02□□를 매핑한 축에서는 트리거 조건에 외부 입력(_mcEXT)을 사용하십시오.		오므론의 EtherCAT 엔코더 슬레이브 GX-EC02□□를 매핑한 축에서는 트리거 조건에 외부 입력(_mcEXT)을 사용하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	전자 기어 분자 설정 범위 외			**이벤트 코드**	54200000Hex	
내용	모션 제어 명령의 입력 변수 'RatioNumerator'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		**발생원 상세 정보**	축	**검출 타이밍**	명령 기동 시
이상의 속성	**중요도**	가벼운 이상	**복귀 방법**	이상 해제	**로그 카테고리**	시스템
발생의 영향	**사용자프로그램**	계속	**동작**	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	**변수명**		**데이터 형식**		**명칭**	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	**발생 요인(추정 원인)**		**처리 대책**		**재발 방지**	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	전자 기어 분모 설정 범위 외		이벤트 코드	54210000Hex	
내용	모션 제어 명령의 입력 변수 'RatioDenominator'에 지정된 파라메터가 범위 밖입니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축의 축 동작 중에는 감속 정지	

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	명령의 입력 파라메터가 입력 변수의 범위 외	해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.

부속 정보	없음
주의 사항/비고	없음

이벤트명	목표 속도 설정 범위 외		이벤트 코드	54220000Hex	
내용	모션 제어 명령의 입력 변수 'Velocity'에 지정된 파라메터가 범위 밖입니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지	

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	명령의 입력 파라메터가 입력 변수 범위 외	해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.

부속 정보	없음
주의 사항/비고	없음

이벤트명	가속도 설정 범위 외		이벤트 코드	54230000Hex	
내용	모션 제어 명령의 입력 변수 'Acceleration'에 지정된 파라메터가 범위 밖입니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지	

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	명령의 입력 파라메터가 입력 변수의 범위 외	해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.

부속 정보	없음
주의 사항/비고	없음

이벤트명	감속도 설정 범위 외			이벤트 코드	54240000Hex	
내용	모션 제어 명령의 입력 변수 'Deceleration'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	저크 설정 범위 외			이벤트 코드	54250000Hex	
내용	모션 제어 명령의 입력 변수 'Jerk'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	토크 경사 설정 범위 외			이벤트 코드	54270000Hex	
내용	모션 제어 명령의 입력 변수 'TorqueRamp'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 계수 설정 범위 외		이벤트 코드		54280000Hex	
내용	모션 제어 명령의 입력 변수 'MasterScaling'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	종축 계수 설정 범위 외		이벤트 코드		54290000Hex	
내용	모션 제어 명령의 입력 변수 'SlaveScaling'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	피딩 속도 설정 범위 외		이벤트 코드		542A0000Hex	
내용	모션 제어 명령의 입력 변수 'FeedVelocity'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	피딩 속도(입력 변수 'FeedVelocity')가 초기값(0) 그대로임		피딩 속도(입력 변수 'FeedVelocity')에 양수값을 지정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	버퍼 모드 선택 범위 외			이벤트 코드	542B0000Hex	
내용	모션 제어 명령의 입력 변수 'BufferMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	좌표 계통 선택 범위 외			이벤트 코드	542C0000Hex	
내용	모션 제어 명령의 입력 변수 'CoordSystem'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	원호 보간 모드 선택 범위 외			이벤트 코드	542D0000Hex	
내용	모션 제어 명령의 입력 변수 'CircMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	방향 선택 범위 외			이벤트 코드	542E0000Hex	
내용	모션 제어 명령의 입력 변수 'Direction'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	경로 선택 범위 외			이벤트 코드	542F0000Hex	
내용	모션 제어 명령의 입력 변수 'PathChoice'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	위치 종류 선택 범위 외			이벤트 코드	54300000Hex	
내용	모션 제어 명령의 입력 변수 'ReferenceType'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통/축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	이동 방법 선택 범위 외			이벤트 코드	54310000Hex	
내용	모션 제어 명령의 입력 변수 'MoveMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	트랜지션 모드 선택 범위 외			이벤트 코드	54320000Hex	
내용	모션 제어 명령의 입력 변수 'TransitionMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
	'BufferMode'에서 '_mcAborting', '_mcBuffered'를 지정하고 'TransitionMode'를 '_mcTMCornerSuperimposed'로 지정		'BufferMode'에서 '_mcAborting', '_mcBuffered'를 지정한 경우는 'TransitionMode'를 '_mcTMNone'로 지정하고, 'TransitionMode'를 '_mcTMCornerSuperimposed'로 지정한 경우는 'BufferMode'에서 '_mcBlendingLow', '_mcBlendingPrevious', '_mcBlendingNext', '_mcBlendingHigh'를 지정하십시오.		'BufferMode'에서 '_mcAborting', '_mcBuffered'를 지정한 경우는 'TransitionMode'를 '_mcTMNone'로 지정하고, 'TransitionMode'를 '_mcTMCornerSuperimposed'로 지정한 경우는 'BufferMode'에서 '_mcBlendingLow', '_mcBlendingPrevious', '_mcBlendingNext', '_mcBlendingHigh'를 지정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	계속 방법 선택 범위 외		이벤트 코드	54330000Hex		
내용	모션 제어 명령의 입력 변수 'Continuous(Reserved)'의 값이 변경되었습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	입력 변수 'Continuous(Reserved)'의 값이 변경됨		입력 변수 'Continuous(Reserved)'의 값을 변경하지 않도록 프로그램을 수정하십시오.		입력 변수 'Continuous(Reserved)'의 값을 변경하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	가감산 방법 선택 범위 외		이벤트 코드	54340000Hex		
내용	모션 제어 명령의 입력 변수 'CombineMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	동기 시작 조건 지정 범위 외		이벤트 코드	54350000Hex		
내용	모션 제어 명령의 입력 변수 'LinkOption'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 종축 동일			이벤트 코드	54360000Hex	
내용	모션 제어 명령의 입력 변수 'Master'와 'Slave'에 지정된 축이 동일합니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통/축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Master'와 'Slave'의 파라메터가 동일		해당 명령의 입력 변수 'Master'와 'Slave'에 지정되는 축이 다르도록 파라메터를 수정하십시오.		명령의 입력 변수 'Master'와 'Slave'에는 다른 축을 지정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 보조축 동일			이벤트 코드	54370000Hex	
내용	모션 제어 명령의 입력 변수 'Master'와 'Auxiliary'에 지정된 축이 동일합니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Master'와 'Auxiliary'의 파라메터가 동일		해당 명령의 입력 변수 'Master'와 'Auxiliary'에 지정되는 축이 다르도록 파라메터를 수정하십시오.		명령의 입력 변수 'Master'와 'Auxiliary'에는 다른 축을 지정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축/종축 축 번호 비오름차순			이벤트 코드	54380000Hex	
내용	모션 제어 명령의 입력 변수 'Master'와 'Slave'에 지정된 축 번호가 오름차순이 아닙니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'ReferenceType'에서 '_mcLatestCommand' 지정 시 명령의 입력 변수 'Master'와 'Slave'의 파라메터가 비오름차순		해당 명령의 입력 변수 'ReferenceType'에서 '_mcLatestCommand'를 지정하는 경우에는 해당 명령의 입력 변수 'Master'와 'Slave'에 지정되는 축 번호가 오름차순이 되도록 파라메터를 수정하십시오. 또한 주축 위치 종류 선택에서 '_mcCommand'를 지정하십시오.		변수 'ReferenceType'에서 '_mcLatestCommand'를 지정하는 경우에는 입력 변수의 주축과 종축이 오름차순이 되도록 지정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	캠 테이블 지정 오류			이벤트 코드	54390000Hex	
내용	모션 제어 명령의 입력 변수 'CamTable'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통/축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 'MC 공통'인 경우: 영향 없음 발생원 상세 정보가 '축'인 경우: 해당 종축의 동작 불가		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_COM.MFaultLvl.Active	BOOL	MC 공통 가벼운 이상 발생 중
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	명령의 입력 변수 'CamTable'에 캠 데이터 변수 이외의 것을 지정	해당 명령의 입력 변수 'CamTable'에 지정되는 파라메터가 캠 데이터 변수가 되도록 파라메터를 수정하십시오.	명령의 입력 변수 'CamTable'에는 캠 데이터 변수를 지정하십시오.

부속 정보	없음
주의 사항/비고	없음

이벤트명	동기 정지 중			이벤트 코드	543A0000Hex	
내용	모션 제어의 동기 제어 명령이 실행되었지만 실행 가능한 조건이 아닙니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	• MC_CamOut(캠 동작 해제) 명령이 실행되었지만 MC_CamIn(캠 동작 시작) 명령 실행 중이 아님 • MC_GearOut(기어 동작 해제) 명령이 실행되었지만 MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령 실행 중이 아님 • MC_Phasing(주축 상대값 위상 보정) 명령이 실행되었지만 MC_CamIn(캠 동작 시작) 명령, MC_GearIn(기어 동작 시작) 명령, MC_GearInPos(위치 지정 기어 동작) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령이 실행 중이 아님	해당 명령 실행 시에 실행 가능한 조건이 되도록 프로그램을 수정하십시오.	동기 명령 실행 시에는 실행 가능한 조건이 되도록 하십시오.

부속 정보	없음
주의 사항/비고	없음

이벤트명	모션 명령 재기동 불가			이벤트 코드	543B0000Hex	
내용	재기동할 수 없는 모션 제어 명령의 재기동을 실행했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통/축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 'MC 공통'인 경우: 영향 없음 발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_COM.MFaultLvl.Active	BOOL	MC 공통 가벼운 이상 발생 중
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	재기동할 수 없는 모션 제어 명령의 재기동	해당 명령의 출력 변수 'Busy'가 FALSE가 될 때까지 입력 변수 'Execute'가 기동되지 않도록 프로그램을 수정하십시오.	재기동할 수 없는 명령을 사용하는 경우는 입력 변수 'Execute'의 기동 조건에 해당 명령의 출력 변수 'Busy'가 FALSE인 것을 포함하도록 하십시오. 또는 다시 명령을 기동하기 전에 일단 명령을 정지하도록 하십시오.

부속 정보	없음
주의 사항/ 비고	없음

이벤트명	모션 명령 다중 기동 불가			이벤트 코드	543C0000Hex	
내용	동일한 대상(MC 공통/축/축 그룹)에 대해 동시에 실행할 수 없는 기능을 둘 이상 실행했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통/축/축 그룹	검출 타이밍	명령 다중 기동 시작 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 'MC 공통'인 경우: 영향 없음 발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹의 동작 불가능. 해당 축 그룹이 동작 중일 때는 감속 정지		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_COM.MFaultLvl.Active	BOOL	MC 공통 가벼운 이상 발생 중
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	동일한 대상(MC 공통/축/축 그룹)에 대해 동시에 실행할 수 없는 기능을 둘 이상 실행	해당 명령의 다중 기동 사양을 확인하고 동시에 실행할 수 없는 명령을 동시에 실행하지 않도록 프로그램을 수정하십시오.	사용하는 명령의 다중 기동 사양을 확인하고 동시에 실행할 수 없는 명령을 동시에 실행하지 않도록 하십시오.

부속 정보	없음
주의 사항/ 비고	없음

이벤트명	축 종류 부적합		이벤트 코드	543D0000Hex	
내용	엔코더 축에 대해 동작 명령을 실행했습니다.				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	엔코더 축에 대해 동작 명령을 실행		해당 명령의 축 종류 설정에 서보 축 또는 가상 서보 축을 지정하거나 엔코더 축에 대해 해당 동작 명령을 실행하지 않도록 프로그램을 수정하십시오.		동작 명령은 서보 축 또는 가상 서보 축에 대해 실행하도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	다축 협조 동작 중인 명령 기동 불가		이벤트 코드	543E0000Hex	
내용	다축 협조 동작 중인 축 또는 축 그룹에 대해 동작 명령을 실행했습니다.				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 다중 기동 시작 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	축 그룹의 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	다축 협조 동작 중인 축 또는 축 그룹에 대해 동작 명령을 실행		해당 축이나 해당 축 그룹이 다축 협조 동작 중이 아닐 때 해당 명령을 실행하도록 프로그램을 수정하십시오.		다축 협조 동작 중이 아닌 축 또는 축 그룹에서 축 동작 명령을 실행하도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	축 그룹 무효 상태에서 다축 협조 명령 기동		이벤트 코드	543F0000Hex	
내용	축 그룹 무효 상태인 축 그룹에 대해 다축 협조 명령을 기동했습니다.				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	축 그룹 무효 상태인 축 그룹에 대해 다축 협조 명령을 기동		축 그룹을 유효 상태로 한 후에 해당 명령을 실행하도록 프로그램을 수정하십시오. 축 그룹을 유효 상태로 하려면 MC_GroupEnable(축 그룹 유효) 명령을 실행하십시오.		축 그룹을 유효하게 한 후 다축 협조 동작 명령을 실행하십시오. 축 그룹을 유효 상태로 하려면 MC_GroupEnable(축 그룹 유효) 명령을 실행하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	축 그룹 유효 불가		이벤트 코드	54400000Hex	
내용	MC_GroupEnable(축 그룹 유효) 명령 실행에 실패했습니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가. 구성 축의 동작은 계속		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	MC_GroupEnable(축 그룹 유효) 명령 실행 시 구성 축 중에 정지 중이 아닌 축이 존재함	모든 구성 축이 정지 중일 때 MC_GroupEnable(축 그룹 유효) 명령을 실행하도록 프로그램을 수정하십시오. 축 변수 'Status.Disabled' 또는 'Status.Standstill'이 TRUE이면 해당 축은 정지 중입니다.	모든 구성 축이 정지 중일 때 MC_GroupEnable(축 그룹 유효) 명령을 실행하도록 프로그램을 작성하십시오. 축 변수 'Status.Disabled' 또는 'Status.Standstill'이 TRUE이면 해당 축은 정지 중입니다.
	MC_GroupEnable(축 그룹 유효) 명령 실행 시 구성 축 중에 MC_TouchProbe(외부 래치 유효) 명령 실행 중인 축이 존재함	모든 구성 축에서 MC_TouchProbe(외부 래치 유효) 명령이 실행되지 않는 상태일 때 MC_GroupEnable(축 그룹 유효) 명령을 실행하도록 프로그램을 수정하십시오.	모든 구성 축에서 MC_TouchProbe(외부 래치 유효) 명령이 실행되지 않는 상태일 때 MC_GroupEnable(축 그룹 유효) 명령을 실행하도록 프로그램을 작성하십시오.

부속 정보	없음
주의 사항/ 비고	없음

이벤트명	운전 불가(서보 OFF) 축 동작 지시		이벤트 코드	54410000Hex	
내용	서보 OFF 중인 축에 대해 동작 명령을 실행했습니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	동작 명령은 기동하지 않음		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	서보 OFF 중인 축에 대해 동작 명령을 실행	서보가 ON한 후에 해당 명령을 실행하도록 프로그램을 수정하십시오.	서보가 ON한 후에 축 동작 명령을 실행하도록 하십시오.
	MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령에 의한 원점 사전 설정을 EtherCAT의 프로세스 데이터 통신이 확립되지 않은 축에 대해 실행	EtherCAT 마스터의 시스템 정의 변수 '_EC_PDSlavTbl'(프로세스 데이터 통신 중 슬레이브 테이블)이 FALSE가 되면 해당 요인을 제거하고, '_EC_PDSlavTbl'이 TRUE가 된 후에 MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령을 통해 원점 사전 설정을 실행하십시오.	컨트롤러의 전원 ON이나 다운로드 직후 또는 슬레이브의 통신 이상 해제, 이탈, 재가입, 무효, 유효 직후에 MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령으로 원점 사전 설정을 실행하려면 EtherCAT 마스터의 시스템 정의 변수 '_EC_PDSlavTbl'(프로세스 데이터 통신 중 슬레이브 테이블)이 TRUE가 된 것을 확인한 후 실행하도록 프로그램을 작성하십시오.

부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 이상이 발생한 논리축 번호
주의 사항/ 비고	없음

이벤트명	구성 축 강제 정지 중 에러			이벤트 코드		54420000Hex
내용	구성 축에 대해 MC_Stop(강제 정지) 명령이 실행되고 있는 축 그룹에 동작 명령을 실행했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	구성 축에 대해 MC_Stop(강제 정지) 명령이 실행되고 있는 축 그룹에 동작 명령을 실행		구성 축의 강제 정지 명령 입력 변수 'Execute'를 FALSE로 하여 실행을 멈추고, 이상 해제를 실행한 후에 동작 명령을 전송하도록 하십시오.		구성 축의 강제 정지 명령 입력 변수 'Execute'를 FALSE로 해서 종료시킨 후 축 그룹의 동작 명령을 실행하도록 하십시오.	
부속 정보	부속 정보 1: 강제 정지된 논리축 번호					
주의 사항/ 비고	없음					

이벤트명	모션 명령 다중 기동 수 초과			이벤트 코드		54430000Hex
내용	버퍼 모드 Buffered, Blending에 의해 버퍼링되는 모션 제어 명령의 버퍼링 수가 한계를 초과했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 다중 기동 시작 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	축 명령으로 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '2'를 초과함		해당 명령 실행 시에 버퍼링 수가 상한을 넘지 않도록 프로그램을 수정하십시오.		축 명령인 경우는 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '2'를 초과하지 않도록 하십시오.	
	축 그룹 명령에서 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '8'을 초과함				축 그룹 명령인 경우는 실행 중인 명령과 버퍼링 중인 명령 수의 합계가 '8'을 초과하지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	이동량 부족		이벤트 코드	54440000Hex		
내용	위치 결정 명령의 다중 기동/재기동 시에 지정된 감속도 또는 가속도대로 동작을 실행하지 못했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	'가감속 초과'를 '이상 시 정지'로 설정했을 때 위치 결정 명령의 다중 기동/재기동 시에 지정된 감속도/가속도로 목표 위치에서 정지할 수 없음		해당 명령의 동작 사양을 바탕으로 위치 결정 동작이 다중 기동/재기동하는 타이밍에 지정한 감속도 또는 가속도로 목표 위치를 초과하지 않도록 프로그램을 수정하십시오. 또는 '가감속 초과'를 '이상 시 정지' 이외의 설정으로 변경하십시오.		사용하는 명령의 동작 사양을 확인해 두고 이러한 이상이 발생하지 않도록 프로그램을 작성하십시오. 또는 '가감속 초과'를 '이상 시 정지' 이외의 것으로 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	Blending 중계 속도에 도달하기 위한 이동량 부족		이벤트 코드	54450000Hex		
내용	중계 속도로 가감속하기 위한 이동량이 부족합니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 다중 기동 시작 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	'가감속 초과'에 '이상 시 정지'가 설정되어 있을 때 현재 명령을 중계 속도로 가감속하기 위한 이동량이 부족		해당 명령의 동작 사양을 바탕으로 이동량이 부족하지 않도록 프로그램을 수정하십시오. 또는 '가감속 초과'를 '이상 시 정지' 이외의 설정으로 변경하십시오.		사용하는 명령의 동작 사양을 확인해 두고 이러한 이상이 발생하지 않도록 프로그램을 작성하십시오. 또는 '가감속 초과'를 '이상 시 정지' 이외의 것으로 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	사다리꼴 패턴 캠 등속 이동량 부족		이벤트 코드	54460000Hex	
내용	주축의 등속 이동량이 '0' 미만입니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	MC_MoveLink(사다리꼴 패턴 캠) 명령에서 주축의 등속 이동량이 '0' 미만		(주축 이동 거리≧주축 가속 이동 거리+주축 감속 이동 거리)가 되도록 프로그램을 수정하십시오.	사용하는 명령의 동작 사양을 확인해 두고 이러한 이상이 발생하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음				
주의 사항/ 비고	없음				

이벤트명	위치 지정 기어 동작 목표 속도 부족		이벤트 코드	54470000Hex	
내용	MC_GearInPos(위치 지정 기어 동작) 명령에서 종축의 '목표 속도'가 낮아 필요한 속도에 도달하지 못했습니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	MC_GearInPos(위치 지정 기어 동작) 명령에서 입력 변수 'Velocity(목표 속도)'의 값이 (명령 기동 시의 주축 속도×기어비)보다 작음		해당 명령의 동작 사양을 바탕으로 (명령 기동 시의 주축 속도×기어비)보다 커지도록 입력 변수 'Velocity(목표 속도)'의 값을 설정하십시오.	사용하는 명령의 동작 사양을 확인해 두고 이러한 이상이 발생하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음				
주의 사항/ 비고	없음				

이벤트명	원호 보간 시작점 종료점 동일		이벤트 코드	54480000Hex	
내용	MC_MoveCircular2D(2축 원호 보간) 명령에서 반지름 지정 방식을 시작점과 종료점이 동일한 위치로 지정했습니다. 또는 통과점 지정 방식에서 시작점과 종료점과 통과점을 동일한 위치로 지정했습니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL	축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 반지름 지정 방식이 지정되고 시작점과 종료점의 위치가 동일		해당 명령의 시작점과 종료점의 위치가 동일한 상태에서 반지름 지정 방식을 지정하지 않도록 프로그램을 수정하십시오.	반지름 지정 방식의 원호 보간을 실행하려면 시작점과 종료점이 동일하지 않도록 설정하십시오.	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식을 시작점과 종료점이 동일한 위치로 지정		해당 명령의 시작점과 종료점과 통과점의 위치가 동일한 상태에서 통과점 지정 방식을 지정하지 않도록 프로그램을 수정하십시오.	통과점 지정 방식의 원호 보간을 실행하려면 시작점과 종료점과 통과점이 동일하지 않도록 설정하십시오.	
부속 정보	없음				
주의 사항/ 비고	없음				

이벤트명	원호 보간 중심점 지정 위치 범위 외			이벤트 코드	54490000Hex	
내용	MC_MoveCircular2D(2축 원호 보간) 명령에서 중심점 지정 방식을 중심점의 위치가 허용 범위를 넘도록 지정했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 중심점 지정 방식이 지정되고, 시작점과 중심점 거리와 종료점과 중심점 거리의 차이가 축 그룹 설정의 '중심점 보정 허용률'에서 지정한 허용 범위를 초과함		시작점과 중심점 거리와 종료점과 중심점 거리의 차이가 축 그룹 설정의 '중심점 보정 허용률'에서 지정한 허용 범위보다 작은 중심점이 설정되도록 수정하십시오.		시작점과 중심점 거리와 종료점과 중심점 거리의 차이가 축 그룹 설정의 '중심점 보정 허용률'에서 지정한 허용 범위 이하가 되도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	카운트 모드 설정에 의한 명령 기동 이상			이벤트 코드	544A0000Hex	
내용	카운트 모드가 로터리 모드로 설정된 축에 대해 로터리 모드에서 사용할 수 없는 명령을 실행했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	카운트 모드가 로터리 모드로 설정된 축을 로터리 모드에서 사용할 수 없는 명령에서 사용		해당 명령에서 지정된 해당 축의 카운트 모드 설정을 리니어 모드로 변경하십시오.		사용하는 명령이 실행할 수 있는 카운트 모드를 확인해서 축의 카운트 모드를 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	파라메터 선택 범위 외			이벤트 코드	544C0000Hex	
내용	모션 제어 명령의 입력 변수 'ParameterNumber'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령 비실행		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	정지 방법 선택 범위 외		이벤트 코드	544D0000Hex	
내용	모션 제어 명령의 입력 변수 'StopMode'에 지정된 파라메터가 범위 밖입니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음				
주의 사항/ 비고	없음				

이벤트명	트리거 입력 조건의 래치 ID 선택 범위 외		이벤트 코드	544E0000Hex	
내용	모션 제어 명령의 입력 변수 'TriggerInput::LatchID'에 지정된 파라메터가 범위 밖입니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음				
주의 사항/ 비고	없음				

이벤트명	MC 설정 쓰기의 설정 범위 외		이벤트 코드	544F0000Hex	
내용	모션 제어 명령의 입력 변수 'SettingValue'에 지정된 파라메터가 범위 밖입니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	MC 공통	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령 비실행	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_COM.MFaultLvl.Active		BOOL	MC 공통 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
	파라메터 지정과 설정값의 데이터 형식이 일치하지 않음		파라메터 지정과 설정값의 데이터 형식이 일치하도록 수정하십시오.	파라메터 지정과 설정값의 데이터 형식을 일치시키십시오.	
부속 정보	없음				
주의 사항/ 비고	없음				

이벤트명	트리거 입력 조건의 모드 선택 범위 외			이벤트 코드	54500000Hex	
내용	모션 제어 명령의 입력 변수 'TriggerInput::Mode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	트리거 입력 조건의 드라이브 트리거 입력 신호 선택 범위 외			이벤트 코드	54510000Hex	
내용	모션 제어 명령의 입력 변수 'TriggerInput::InputDrive'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(축 지정)			이벤트 코드	54530000Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Axis'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(버퍼 모드 선택)		이벤트 코드	54540000Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'BufferMode'의 파라메터를 변경했습니다.				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(방향 선택)		이벤트 코드	54550000Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Direction'의 파라메터를 변경했습니다.				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(반복 모드)		이벤트 코드	54560000Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Periodic'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(축 그룹 지정)		이벤트 코드	54570000Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'AxesGroup'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(약동 설정)		이벤트 코드	54580000Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Jerk'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(주축)		이벤트 코드	54590000Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Master'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(MasterOffset)		이벤트 코드	545A0000Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'MasterOffset'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(MasterScaling)		이벤트 코드	545B0000Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'MasterScaling'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(MasterStartDistance)		이벤트 코드	545C0000Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'MasterStartDistance'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(Continuous)			이벤트 코드	545D0000Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'Continuous'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(MoveMode)			이벤트 코드	545E0000Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'MoveMode'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	보조축 지정 오류			이벤트 코드	545F0000Hex	
내용	모션 제어 명령의 입력 변수 'Auxiliary'에 지정된 축이 존재하지 않습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Auxiliary'에 지정된 축이 존재하지 않는 변수		해당 명령에서 지정된 축을 존재하는 변수로 수정하십시오.		명령에 대한 입력 파라메터에 변수를 지정하려면 존재하는 변수를 지정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	축 지정 오류			이벤트 코드	54600000Hex	
내용	모션 제어 명령의 입력 변수 'Axis'에 지정된 축이 존재하지 않습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령 비실행		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Axis'에 지정된 축이 존재하지 않는 변수		해당 명령에서 지정된 축을 존재하는 변수로 수정하십시오.		명령에 대한 입력 파라메터에 변수를 지정하려면 존재하는 변수를 지정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	축 그룹 지정 오류			이벤트 코드	54610000Hex	
내용	모션 제어 명령의 입력 변수 'AxesGroup'에 지정된 축 그룹이 존재하지 않거나 사용 축 그룹이 아닙니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령 비실행		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'AxesGroup'에 지정된 축 그룹이 존재하지 않는 변수		해당 명령에서 지정된 축 그룹을 존재하는 축 그룹 변수로 수정하십시오.		명령에 대한 입력 파라메터에 변수를 지정하려면 존재하는 변수를 지정하십시오.	
	명령의 입력 변수 'AxesGroup'에 지정된 축 그룹이 사용 축 그룹으로 설정되지 않음		해당 명령에서 지정된 축 그룹을 사용 축 그룹으로 설정하십시오.		입력 변수 'AxesGroup'에 지정하는 축 그룹은 사용 축 그룹으로 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 지정 오류			이벤트 코드	54620000Hex	
내용	모션 제어 명령의 입력 변수 'Master'에 지정된 축이 존재하지 않거나 동기 주축이 아닙니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통/축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Master'에 지정된 축이 존재하지 않는 변수		해당 명령에서 지정된 축을 존재하는 변수로 수정하십시오.		명령에 대한 입력 파라메터에 변수를 지정하려면 존재하는 변수를 지정하십시오.	
	MC_Phasing(주축 상대값 위상 보정) 명령에서 입력 변수 'Master'에 지정된 축이 동기 주축이 아님		MC_Phasing(주축 상대값 위상 보정) 명령의 입력 변수 'Master'에 입력된 변수를 동기 명령의 주축에 지정된 축으로 수정하십시오.		MC_Phasing(주축 상대값 위상 보정) 명령의 입력 변수 'Master'에 입력된 변수를 동기 명령의 주축에 지정된 축으로 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(SlaveOffset)		이벤트 코드	54630000Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'SlaveOffset'을 변경했습니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.	사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음				
주의 사항/ 비고	없음				

이벤트명	모션 명령 재기동 불가(SlaveScaling)		이벤트 코드	54640000Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'SlaveScaling'을 변경했습니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.	사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음				
주의 사항/ 비고	없음				

이벤트명	모션 명령 재기동 불가(StartPosition)		이벤트 코드	54650000Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'StartPosition'을 변경했습니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.	사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음				
주의 사항/ 비고	없음				

이벤트명	원점 미확정 상태에서의 명령 기동 이상		이벤트 코드	54660000Hex		
내용	원점 미확정 상태에서 고속 원점 복귀 또는 보간 명령이 실행되었습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	원점 미확정 상태에서 고속 원점 복귀가 실행됨		원점 복귀를 통해 원점을 확정하고 나서 해당 고속 원점 복귀를 실행하십시오.		원점 복귀를 통해 원점 확정 상태로 한 후 고속 원점 복귀 명령을 실행하십시오.	
	원점 미확정 상태를 포함한 구성 축에 대해 보간 명령이 실행됨		원점 복귀를 통해 모든 구성 축의 원점을 확정하고 나서 해당 보간 명령을 실행하십시오.		원점 복귀를 통해 모든 구성 축의 원점을 확정하고 나서 보간 명령을 실행하십시오.	
부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 논리축 번호					
주의 사항/ 비고	원점 복귀 실행 후에도 '현재 위치 변경' 명령을 실행하면 원점 미확정 상태가 되므로 다시 한번 원점 복귀를 실행해 원점 확정 상태로 해야 합니다.					

이벤트명	모션 명령 재기동 불가(위치 종류)		이벤트 코드	54670000Hex		
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'ReferenceType'을 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	미사용 축 지정(주축)			이벤트 코드	54680000Hex	
내용	모션 제어 명령에서 지정된 주축이 사용하지 않는 축입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	모션 제어 명령에서 지정된 주축이 사용하지 않는 축		해당 명령에서 지정된 주축을 사용 축으로 수정하십시오.		모션 제어 명령에서 지정하는 주축에는 사용 축을 지정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	시작 위치 설정 범위 외			이벤트 코드	54690000Hex	
내용	모션 제어 명령의 입력 변수 'FirstPosition'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	종료 위치 설정 범위 외			이벤트 코드	546A0000Hex	
내용	모션 제어 명령의 입력 변수 'LastPosition'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	시작 위치/종료 위치 · 대소 관계 오류(리니어 모드)		이벤트 코드		546B0000Hex	
내용	모션 제어 명령의 입력 변수 'LastPosition'에 지정된 파라메터가 입력 변수 'FirstPosition'에 지정된 파라메터보다 작은 값입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	카운트 모드가 리니어 모드일 때 명령의 입력 파라메터에서 'LastPosition'의 값이 'FirstPosition'의 값보다 작음		해당 명령에서 지정된 'LastPosition'의 값이 'FirstPosition'의 값 이상이 되도록 수정하십시오. 또는 카운트 모드를 로터리 모드로 변경하십시오.		해당 명령에서 지정된 'LastPosition'의 값이 'FirstPosition'의 값 이상이 되도록 프로그램을 작성하십시오. 또는 해당 축의 카운트 모드가 로터리 모드로 되어 있는지 확인하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 동기 위치 설정 범위 외		이벤트 코드		546C0000Hex	
내용	모션 제어 명령의 입력 변수 'MasterSyncPosition'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	종축 동기 위치 설정 범위 외		이벤트 코드		546D0000Hex	
내용	모션 제어 명령의 입력 변수 'SlaveSyncPosition'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	트리거 입력 조건의 래치 ID 중복		이벤트 코드	546E0000Hex	
내용	모션 제어 명령에서 지정된 래치 ID가 복수의 명령에서 중복됩니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령에서 동일한 래치 ID가 동시에 사용됨		같은 래치 ID가 해당 명령과 다른 명령에서 동시에 사용되지 않도록 프로그램을 수정하십시오. 다른 래치 ID를 사용하도록 하거나 같은 래치 ID를 사용하는 명령을 동시에 실행하지 않도록 하십시오. 또한 MC_Home(원점 복귀) 명령 또는 MC_HomeWithParameter(파라메터 지정 원점 복귀) 명령 실행 중에는 래치 기능 1/2도 사용 중으로 간주됩니다.	MC_TouchProbe(외부 래치 유효) 명령, MC_MoveLink(사다리꼴 패턴 캠) 명령, MC_MoveFeed(인터럽트 피딩 위치 결정) 명령에서 동시에 동일한 래치 ID를 사용하지 않도록 하십시오.	
	MC_TouchProbe(외부 래치 유효) 이외의 명령에서 사용 중인 래치를 MC_AbortTrigger(외부 래치 무효) 명령에서 중지하려고 함		외부 래치 유효 이외의 명령에서 사용 중인 래치를 외부 래치 무효 명령으로 중지하지 않도록 하십시오.	외부 래치 유효 이외의 명령에서 사용 중인 래치는 외부 래치 무효 명령을 실행하지 않도록 하십시오.	
부속 정보	없음				
주의 사항/ 비고	래치 ID를 변경하려면 다른 명령에서 해당 래치 ID가 동시에 사용 중이 아님을 확인하십시오.				

이벤트명	저크 오버라이드값 범위 외		이벤트 코드	546F0000Hex	
내용	모션 제어 명령의 입력 변수 'JerkFactor'에 지정된 파라메터가 범위 밖입니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지	
시스템 정의 변수	변수명		데이터 형식	명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL	축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책	재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 논리축 번호				
주의 사항/ 비고	없음				

이벤트명	가감속 속도 오버라이드값 범위 외			이벤트 코드	54700000Hex	
내용	모션 제어 명령의 입력 변수 'AccFactor'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 논리축 번호					
주의 사항/ 비고	없음					

이벤트명	시작 위치 방식 지정 범위 외			이벤트 코드	54710000Hex	
내용	모션 제어 명령의 입력 변수 'StartMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	모션 명령 재기동 불가(시작 위치 방식)			이벤트 코드	54720000Hex	
내용	모션 제어 명령 재기동 시에 변경할 수 없는 입력 변수 'StartMode'의 파라메터를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 재기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 재기동 시에 변경할 수 없는 입력 변수의 파라메터를 변경함		해당 명령 재기동 시에 해당 입력 변수의 파라메터가 변하지 않도록 프로그램을 수정하십시오.		사용하는 모션 제어 명령의 각 입력 변수가 재기동 시에 변경 가능한지를 설명서에서 확인하고, 변경 불가능한 입력 변수에서는 재기동 시에 입력 파라메터가 변화하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	미사용 축 지정(보조축)			이벤트 코드	54740000Hex	
내용	모션 제어 명령의 입력 변수 'Auxiliary'에 지정된 축이 사용하지 않는 축입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Auxiliary'에 지정된 축이 사용하지 않는 축		해당 명령에서 지정된 축을 사용 축으로 설정하십시오. 또는 사용 축을 지정하도록 파라메터를 수정하십시오.		명령에서 지정하는 축은 사용 축으로 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	위치 지정 기어 지정값 이상			이벤트 코드	54750000Hex	
내용	모션 제어 명령에 입력된 속도/가속도/감속도로는 동기 동작할 수 없습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령에 입력된 속도/가속도/감속도로는 지정된 동기 동작을 실행할 수 없음		MC_GearInPos(위치 지정 기어 동작) 명령의 동작 사양을 바탕으로 동기 동작이 가능한 설정이 되도록 프로그램을 수정하십시오.		사용하는 명령의 처리를 확인하고 동기 동작이 가능하도록 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	위치 지정 기어 주축 속도 제로			이벤트 코드	54760000Hex	
내용	모션 제어 명령 기동 시에 주축의 속도가 '0'입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 기동 시에 주축의 속도가 '0'		해당 명령 기동 시에 주축의 속도가 '0'이 되지 않도록 프로그램을 수정하십시오.		명령 실행 시에 주축의 속도가 '0'이 되지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	목표 위치 설정 범위 외			이벤트 코드	54780000Hex	
내용	모션 제어 명령의 입력 변수 'Position'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	명령의 입력 파라메터가 입력 변수의 범위 외	해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.	명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.
	로터리 모드인 축에서 목표 위치가 링 설정 범위 외	로터리 모드인 축에서는 목표 위치를 링 설정의 범위 이내가 되도록 수정하십시오.	로터리 모드인 축에서는 목표 위치를 링 설정의 범위 이내가 되도록 하십시오.

부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 논리축 번호
주의 사항/ 비고	없음

이벤트명	이동 거리 범위 외			이벤트 코드	54790000Hex	
내용	모션 제어 명령의 입력 변수 'Distance'에 지정된 파라메터가 범위 밖이거나 'Distance'의 값을 가산한 목표 위치가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	명령 입력 파라메터의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함	해당 명령의 입력 변수 'Distance'에서 지정되는 입력 파라메터를 이동 거리나 목표 위치가 범위 밖이 되지 않도록 수정하십시오.	명령에서 이동 거리나 목표 위치가 범위 밖이 되지 않도록 프로그램을 작성하십시오.
	리니어 모드인 축에서 이동 거리를 가산한 목표 위치를 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함		

부속 정보	없음
주의 사항/ 비고	없음

이벤트명	캠 테이블 시작점 위치 설정 범위 외		이벤트 코드		547A0000Hex	
내용	모션 제어 명령의 입력 변수 'StartPosition'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	캠 동작(주축 추종) 시작 위치 설정 범위 외		이벤트 코드		547B0000Hex	
내용	모션 제어 명령의 입력 변수 'MasterStartDistance'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	원호 보간 반지름 지정 이상		이벤트 코드		547C0000Hex	
내용	MC_MoveCircular2D(2축 원호 보간) 명령에서 반지름 지정 방식에서 지정된 반지름으로 원호의 궤적을 작성할 수 없습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 반지름 지정 방식이 지정되고 지정된 반지름으로는 원호의 궤적을 작성할 수 없음		원호의 궤적을 작성할 수 있도록 반지름을 수정하십시오.		사용하는 명령의 처리를 확인하여 원호의 궤적을 작성할 수 있도록 반지름을 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	원호 보간 반지름 오버 플로우		이벤트 코드	547D0000Hex		
내용	MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식/중심점 지정 방식일 때 원호의 반지름이 최대값을 초과했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_MoveCircular2D(2축 원호 보간) 명령에서 통과점 지정 방식/중심점 지정 방식인 경우 원호의 반지름을 펄스 단위로 변환했을 때 40비트의 범위를 초과함		해당 명령을 동작 사양에 기초하여 원호의 반지름을 펄스 단위로 변환했을 때 40비트의 범위를 넘지 않도록 입력 파라메터를 수정하십시오. 통과점을 지정한 경우: 시작점, 통과점, 종료점 중심점을 지정한 경우: 시작점, 종료점, 중심점		사용하는 명령의 처리를 확인하고 원호의 반지름을 펄스 단위로 변환했을 때 40비트의 범위를 넘지 않도록 입력 파라메터를 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	반지름 지정 방식인 경우에 반지름이 최대값을 넘으면 '통과점 위치/중심 위치/반지름 지정 범위 외' 이상이 발생합니다.					

이벤트명	원호축 지정 범위 외		이벤트 코드	547E0000Hex		
내용	모션 제어 명령의 입력 변수 'CircAxes'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		해당 명령에 입력된 파라메터를 입력 변수의 범위를 넘지 않도록 설정하십시오.	
	'CircAxes'에 지정된 축이 축 그룹 설정의 구성 축에 포함되지 않음		'CircAxes'에 지정하는 축을 축 그룹 구성에 포함하도록 설정을 변경하십시오.		'CircAxes'에 지정하는 축을 축 그룹 구성에 포함하십시오.	
	'CircAxes'의 2축에 동일 축이 지정됨		'CircAxes'에 지정하는 2축이 다른 축이 되도록 수정하십시오.		'CircAxes'에 지정하는 2축이 다르도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	보조축/종축 · 축 번호 비오름차순		이벤트 코드		547F0000Hex	
내용	모션 제어 명령에서 지정된 입력 변수 'Auxiliary'와 'Slave'의 파라메터 값이 오름차순으로 나열되어 있지 않습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Auxiliary'와 'Slave'의 파라메터가 오름차순이 아님		해당 명령의 'Auxiliary'와 'Slave'의 입력 파라메터에 지정하는 축 번호가 오름차순이 되도록 수정하십시오.		'Auxiliary'와 'Slave'에 지정하는 축 번호가 오름차순이 되도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	캠 테이블 속성 갱신 중 데이터 오름차순 이상		이벤트 코드		54800000Hex	
내용	유효 데이터 수 계산 중에 위상이 오름차순이 아님이 확인되었습니다. 또는 계산 결과 유효 데이터 수가 '0'임이 확인되었습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통	검출 타이밍	명령 실행 중
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	유효 데이터 수 계산 중에 위상이 오름차순이 아님이 확인됨		캠 데이터를 위상이 오름차순이 되도록 수정하십시오.		캠 테이블 데이터의 위상이 오름차순이 되도록 하십시오.	
	계산 결과 유효 데이터 수가 '0'임이 확인됨		캠 테이블의 데이터를 '0' 이외의 위상도 포함하도록 수정하십시오.		캠 테이블의 데이터를 '0' 이외의 위상도 포함하도록 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	MC 설정 쓰기의 대상 범위 외		이벤트 코드		54810000Hex	
내용	모션 제어 명령의 입력 변수 'Target'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 이동 거리 지정 범위 외			이벤트 코드	54820000Hex	
내용	모션 제어 명령의 입력 변수 'MasterDistance'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 가속 이동 거리 지정 범위 외			이벤트 코드	54830000Hex	
내용	모션 제어 명령의 입력 변수 'MasterDistanceInACC'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 감속 이동 거리 지정 범위 외			이벤트 코드	54840000Hex	
내용	모션 제어 명령의 입력 변수 'MasterDistanceInDEC'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	실행 모드 선택 범위 외			이벤트 코드	54870000Hex	
내용	모션 제어 명령의 입력 변수 'ExecutionMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	축간 편차 허용값 범위 외			이벤트 코드	54880000Hex	
내용	모션 제어 명령의 입력 변수 'PermittedDeviation'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 명령 비실행		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	통과점 위치/중심 위치/반지름 지정 범위 외			이벤트 코드	54890000Hex	
내용	모션 제어 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	통과점 또는 중심으로 지정된 경우 'AuxPoint'의 값을 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
	반지름으로 지정된 경우 'AuxPoint[0]'의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함					
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	종료점 지정 범위 외			이벤트 코드	548A0000Hex	
내용	모션 제어 명령의 입력 변수 'EndPoint'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 입력 파라메터를 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	종축 이동 거리 지정 범위 외			이벤트 코드	548B0000Hex	
내용	모션 제어 명령의 입력 변수 'SlaveDistance'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 입력 파라메터를 펄스 단위로 변환했을 때 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	위상 보정량 범위 외			이벤트 코드	548C0000Hex	
내용	모션 제어 명령의 입력 변수 'PhaseShift'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 입력 파라메터의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	피딩 거리 범위 외			이벤트 코드	548D0000Hex	
내용	모션 제어 명령의 입력 변수 'FeedDistance'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 입력 파라메터의 절대값을 펄스 단위로 변환했을 때 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	보조축/종축 동일			이벤트 코드	548E0000Hex	
내용	모션 제어 명령의 입력 변수 'Auxiliary'와 'Slave'에 지정된 축이 동일합니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Auxiliary'와 'Slave'의 파라메터가 동일		해당 명령의 입력 변수 'Auxiliary'와 'Slave'에 지정되는 축이 다르도록 파라메터를 수정하십시오.		명령 입력 변수의 보조축과 종축에는 다른 축을 지정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	상대 위치 선택 범위 외			이벤트 코드	548F0000Hex	
내용	모션 제어 명령의 입력 변수 'Relative'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	캠 트랜지션 지정 선택 범위 외			이벤트 코드		54900000Hex
내용	모션 제어 명령의 입력 변수 'CamTransition'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	동기 제어 해제 모드 선택 범위 외			이벤트 코드		54910000Hex
내용	모션 제어 명령의 입력 변수 'OutMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	외부 래치 유효 명령 실행 불가			이벤트 코드		54920000Hex
내용	CSP 이외의 제어 모드일 때 입력 변수 'StopMode'에 '_mcImmediateStop(즉시 정지)'을 지정해서 드라이버 모드의 MC_TouchProbe(외부 래치 유효) 명령을 실행했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	CSP 이외의 제어 모드이거나 서보 OFF 시에 입력 변수 'StopMode'에 '_mcImmediateStop(즉시 정지)'을 지정해서 드라이버 모드의 MC_TouchProbe(외부 래치 유효) 명령을 실행함		'_mcImmediateStop(즉시 정지)'을 지정하고 드라이버 모드의 외부 래치 유효 명령을 실행하는 경우는 제어 모드를 CSP 모드로 하십시오.		'_mcImmediateStop(즉시 정지)'을 지정하고 드라이버 모드의 외부 래치 유효 명령을 실행하는 경우는 제어 모드를 CSP 모드로 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 오프셋 설정 범위 외			이벤트 코드	54930000Hex	
내용	모션 제어 명령의 입력 변수 'MasterOffset'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 입력 파라메터를 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	종축 오프셋 설정 범위 외			이벤트 코드	54940000Hex	
내용	모션 제어 명령의 입력 변수 'SlaveOffset'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 입력 파라메터를 펄스 단위로 변환했을 때 부호가 있는 40비트의 범위를 초과함		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	명령 현재 위치 카운트 선택 범위 외			이벤트 코드	54950000Hex	
내용	모션 제어 명령의 입력 변수 'CmdPosMode'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 기어비 분자 범위 외			이벤트 코드	54960000Hex	
내용	모션 제어 명령의 입력 변수 'RatioNumeratorMaster'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가. 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 기어비 분모 범위 외			이벤트 코드	54970000Hex	
내용	모션 제어 명령의 입력 변수 'RatioDenominatorMaster'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가. 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	보조축 기어비 분자 범위 외			이벤트 코드	54980000Hex	
내용	모션 제어 명령의 입력 변수 'RatioNumeratorAuxiliary'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가. 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	보조축 기어비 분모 범위 외			이벤트 코드	54990000Hex	
내용	모션 제어 명령의 입력 변수 'RatioDenominatorAuxiliary'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가. 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	주축 위치 종류 선택 범위 외			이벤트 코드	549A0000Hex	
내용	모션 제어 명령의 입력 변수 'ReferenceTypeMaster'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가. 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	보조축 위치 종류 선택 범위 외			이벤트 코드	549B0000Hex	
내용	모션 제어 명령의 입력 변수 'ReferenceTypeAuxiliary'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 종축의 동작 불가. 해당 종축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수의 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	목표 위치 링 카운터 범위 외			이벤트 코드	549C0000Hex	
내용	실행된 명령의 목표 위치가 링 카운터의 범위 밖이므로 동작할 수 없습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	링 카운터 범위에 0이 포함되지 않도록 설정되었을 때 고속 원점 복귀를 실행함		링 카운터 범위에 0이 포함되지 않도록 설정되었을 때는 고속 원점 복귀를 실행할 수 없습니다. 고속 원점 복귀를 실행하지 않도록 프로그램을 수정하십시오. 또는 해당 축의 링 카운터 범위에 0이 포함되도록 설정을 변경하십시오.		링 카운터 범위에 0이 포함되지 않도록 설정되었을 때는 고속 원점 복귀를 실행할 수 없습니다. 고속 원점 복귀를 실행하지 않도록 프로그램을 작성하십시오. 또는 해당 축의 링 카운터 범위에 0이 포함되도록 설정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	축 그룹 구성축 설정 범위 외			이벤트 코드	549D0000Hex *1	
내용	모션 제어 명령의 입력 변수 'Axes'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 그룹 동작 불가. 해당 축 그룹이 축 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		이름	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 세부 사항 01Hex: 형식 지정 이상 02Hex: 배열의 요소 수가 구성축 수보다 작음 03Hex: 지정한 축 지정한 축 번호의 축 종류가 대상 외, 지정한 축 번호가 범위 외					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.01에서 추가된 이벤트입니다.

이벤트명	축 사용 설정 범위 밖			이벤트 코드	549E0000 Hex *1	
내용	모션 제어 명령의 입력 변수 'AxisUse'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통/축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		이름	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 밖		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	사용 축에 대해 명령을 실행했을 때 이상이 발생한 경우는 축 이상이 되고, 미사용 축에 대해 명령을 실행했을 때 이상이 발생한 경우는 MC 공통 이상이 됩니다.					

*1. CPU 유니트의 유니트 버전 Ver.1.04에서 추가된 이벤트입니다.

이벤트명	원점 복귀 파라메터 설정 범위 외			이벤트 코드	57000000 Hex *1	
내용	모션 제어 명령의 입력 변수 'HomingParameter'에 지정된 파라메터가 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		이름	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 파라메터가 입력 변수 범위 외		해당 명령에서 입력 변수의 범위를 넘지 않도록 파라메터를 수정하십시오.		명령에 대한 입력 파라메터가 입력 변수의 범위를 넘지 않도록 하십시오.	
부속 정보	부속 정보 1: 이상의 세부 사항 1: 원점 복귀 동작 범위 외, 2: 원점 입력 신호 범위 외, 3: 원점 복귀 시작 방향 범위 외, 4: 원점 검출 방향 범위 외, 5: 정방향 한계 입력 시 동작 범위 외, 6: 역방향 한계 입력 시 동작 범위 외, 7: 원점 복귀 속도 범위 외, 8: 원점 복귀 접근 속도 범위 외, 9: 원점 복귀 가속도 범위 외, 10: 원점 복귀 감속도 범위 외, 11: 원점 복귀 저크 범위 외, 12: 원점 입력 마스크 범위 외, 13: 원점 위치 오프셋 범위 외, 14: 원점 복귀 유지 시간 범위 외, 15: 원점 복귀 보정값 범위 외, 16: 원점 보정 속도 범위 외, 100: 원점 입력 마스크 거리가 펄스 단위 변환 후 40비트를 초과, 101: 원점 입력 마스크 거리가 링 길이를 초과, 102: 원점 복귀 보정값이 펄스 단위 변환 후 40비트를 초과, 103: 원점 복귀 보정값이 링 길이를 초과, 104: 원점 위치 오프셋이 펄스 단위 변환 후 40비트를 초과, 105: 원점 위치 오프셋이 링 범위 외, 106: 원점 복귀 속도가 최고 속도를 초과, 107: 원점 복귀 접근 속도가 최고 속도를 초과, 108: 원점 복귀 접근 속도≦원점 복귀 속도가 아님, 109: 원점 보정 속도≦원점 복귀 속도가 아님, 110: 원점 복귀 가속도가 최고 가속도를 초과, 111: 원점 복귀 감속도가 최고 감속도를 초과					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.03에서 추가된 이벤트입니다.

이벤트명	축 미사용 전환 이상			이벤트 코드	57020000Hex *1	
내용	정지 중이 아니거나 명령 속도 포화 상태인 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령을 실행했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복구 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	정지 중이 아니거나 명령 속도 포화 상태인 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령을 실행함		이상을 해제하고 축이 정지 중일 때, 명령 속도 포화 상태가 아닐 때 MC_ChangeAxisUse(축 사용 변경) 명령을 실행하십시오. 축 변수 'Status.Disabled' 또는 'Status.Standstill'이 TRUE이면 해당 축은 정지 중입니다. 축 변수 'Details.VelLimit'가 TRUE이면 해당 축은 명령 속도 포화 상태입니다.		축이 정지 중일 때, 명령 속도 포화 상태가 아닐 때 MC_ChangeAxisUse(축 사용 변경) 명령을 실행하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.04에서 추가된 이벤트입니다.

이벤트명	축 사용 전환 불가			이벤트 코드	57030000Hex *1	
내용	사용 실제 축 최대 수를 초과하도록 MC_ChangeAxisUse(축 사용 변경) 명령을 실행했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복구 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	사용 실제 축 최대 수를 초과하도록 MC_ChangeAxisUse(축 사용 변경) 명령을 실행했습니다.		사용하는 CPU 유니트의 사용 실제 축 최대 수를 넘지 않도록 프로그램을 수정하십시오.		사용하는 CPU 유니트의 사용 실제 축 최대 수를 넘지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	축 사용 전환 시 모션 제어 파라메터 설정 이상		이벤트 코드	57200000Hex *1		
내용	사용 축으로 전환한 축의 모션 제어 파라메터 설정이 잘못되었습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복구 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	MC_ChangeAxisUse(축 사용 변경) 명령을 사용하여 미사용 축에서 사용 축으로 전환한 축의 모션 제어 파라메터 설정이 잘못됨		Sysmac Studio를 사용하여 이상이 발생한 축의 [축 사용]을 [사용 축]으로 변경하고, 이상 위치를 확인한 후 수정하십시오. 이상이 발생하지 않으면 [미사용 축]으로 다시 변경하고 다운로드하십시오.		[사용 축] 설정에서 정상적으로 동작하는지 확인한 후 [미사용 축]으로 설정하고 다운로드하십시오.	
	모션 제어 파라메터 설정 다운로드 중에 전원 OFF됨		Sysmac Studio에서 모션 제어 파라메터 설정을 다운로드하십시오.		파라메터 설정 저장 중에 전원을 OFF하지 마십시오.	
	비휘발성 메모리의 고장 또는 비휘발성 메모리의 수명이 다함		상기의 처리를 실시해도 이 이상이 해제되지 않는 경우는 CPU 유니트를 교환하십시오.		없음	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.04에서 추가된 이벤트입니다.

이벤트명	축 사용 전환 시 필수 프로세스 데이터 개체 미설정		이벤트 코드	57210000Hex *1		
내용	사용 축으로 전환한 축의 축 종류에 대한 필수 개체가 설정되어 있지 않습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	MC 공통	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복구 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		MC 공통 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	사용 축으로 전환한 축의 축 종류에 대한 필수 개체가 PDO 매핑 설정에 설정되어 있지 않음		Sysmac Studio의 [PDO 매핑 설정 편집]에서 이상이 발생한 축의 필수 개체를 설정하십시오. 필수 개체에 대해서는 '2-3 PDO 매핑(2-27페이지)'을 참조하십시오.		[사용 축] 설정에서 정상적으로 동작하는지 확인한 후 [미사용 축]으로 설정하고 다운로드하십시오.	
	모션 제어 파라메터 설정 다운로드 중에 전원 OFF됨		Sysmac Studio에서 모션 제어 파라메터 설정을 다운로드하십시오.		파라메터 설정 저장 중에 전원을 OFF하지 마십시오.	
	비휘발성 메모리의 고장 또는 비휘발성 메모리의 수명이 다함		상기의 처리를 실시해도 이 이상이 해제되지 않는 경우는 CPU 유니트를 교환하십시오.		없음	
	[축 사용]이 [사용하지 않는 축(사용 축으로 전환 불가)]으로 설정되어 있는 축에 대해 MC_ChangeAxisUse(축 사용 변경) 명령이 실행됨		[축 사용]이 [사용하지 않는 축(사용 축으로 전환 불가)]으로 설정되어 있는 축에 대해서 MC_ChangeAxisUse(축 사용 변경) 명령을 실행하지 않도록 프로그램을 수정하십시오.		[축 사용]이 [사용하지 않는 축(사용 축으로 전환 불가)]으로 설정되어 있는 축에 대해서 MC_ChangeAxisUse(축 사용 변경) 명령을 실행하지 않도록 프로그램을 수정하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.04에서 추가된 이벤트입니다.

이벤트명	모션 명령 다중 기동 불가능(주축)		이벤트 코드	572F0000Hex *1		
내용	명령 다중 기동 시 변경할 수 없는 입출력 변수 'Master'를 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 다중 기동 시
이상의 속성	중요도	가벼운 이상	복구 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_COM.MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 다중 기동 시 변경할 수 없는 입출력 변수 'Master'를 변경		해당 명령의 다중 기동 시 입출력 변수 'Master'의 값을 변경하지 않도록 프로그램을 수정하십시오.		해당 명령의 다중 기동 시 입출력 변수 'Master'의 값을 변경하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	모션 명령 다중 기동 불가능(위치 종류 선택)		이벤트 코드	57300000Hex *1		
내용	명령 다중 기동 시 변경할 수 없는 입출력 변수 'ReferenceType'을 변경했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 다중 기동 시
이상의 속성	중요도	가벼운 이상	복구 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령 다중 기동 시 변경할 수 없는 입출력 변수 'ReferenceType'을 변경		해당 명령의 다중 기동 시 입출력 변수 'ReferenceType'의 값을 변경하지 않도록 프로그램을 수정하십시오.		해당 명령의 다중 기동 시 입출력 변수 'ReferenceType'의 값을 변경하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	목표 위치 정방향 소프트웨어 리미트 초과		이벤트 코드	64400000Hex		
내용	지정한 위치가 정방향 소프트웨어 리미트의 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
	_MC_GRP[*].MFaultLvl.Active		BOOL		축 그룹 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입력 변수 'Position'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 외		해당 명령의 입력 변수 'Position'에서 지정된 파라메터를 정방향 소프트웨어 리미트의 범위 이내가 되도록 수정하십시오.		명령의 입력 변수 'Position'에 지정된 입력 파라메터가 정방향 소프트웨어 리미트의 범위 이내가 되도록 하십시오.	
	시작 위치가 정방향 소프트웨어 리미트의 범위 밖이고 소프트웨어 리미트의 범위와는 반대 방향으로 동작을 지정해 명령을 실행		해당 명령에서의 이동 방향이 정방향 소프트웨어 리미트 범위 이내를 향하는 방향이 되도록 프로그램을 수정하십시오.		시작 위치가 정방향 소프트웨어 리미트 범위 밖이면 이동 방향이 정방향 소프트웨어 리미트 범위 이내를 향하는 방향이 되도록 프로그램을 작성하십시오.	
	통과점이 지정된 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 외		해당 명령의 입력 변수 'AuxPoint'에서 지정된 파라메터를 정방향 소프트웨어 리미트의 범위 이내가 되도록 수정하십시오.		통과점이 지정된 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 정방향 소프트웨어 리미트의 범위 이내가 되도록 하십시오.	
부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 논리축 번호					
주의 사항/ 비고	없음					

이벤트명	목표 위치 역방향 소프트웨어 리미트 초과		이벤트 코드	64410000Hex	
내용	지정한 위치가 역방향 소프트웨어 리미트의 범위 밖입니다.				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가. 동작 중에는 감속 정지 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가. 해당 축 그룹이 동작 중에는 감속 정지		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	명령의 입력 변수 'Position'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 외	해당 명령의 입력 변수 'Position'에서 지정된 파라메터를 역방향 소프트웨어 리미트의 범위 이내가 되도록 수정하십시오.	명령의 입력 변수 'Position'에 지정된 입력 파라메터가 역방향 소프트웨어 리미트의 범위 이내가 되도록 설정하십시오.
	시작 위치가 역방향 소프트웨어 리미트의 범위 밖이고 소프트웨어 리미트의 범위와는 반대 방향으로 동작을 지정해 명령을 실행	해당 명령에서의 이동 방향이 역방향 소프트웨어 리미트 범위 이내를 향하는 방향이 되도록 프로그램을 수정하십시오.	시작 위치가 역방향 소프트웨어 리미트 범위 밖이면 이동 방향이 역방향 소프트웨어 리미트 범위 이내를 향하는 방향이 되도록 프로그램을 작성하십시오.
	통과점이 지정된 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 외	해당 명령의 입력 변수 'AuxPoint'에서 지정된 파라메터를 역방향 소프트웨어 리미트의 범위 이내가 되도록 수정하십시오.	통과점이 지정된 MC_MoveCircular2D(2축 원호 보간) 명령의 입력 변수 'AuxPoint'에 지정된 파라메터가 역방향 소프트웨어 리미트의 범위 이내가 되도록 하십시오.

부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 논리축 번호
주의 사항/ 비고	없음

이벤트명	명령 위치 오버 플로우/언더 플로우 발생 중		이벤트 코드	64420000Hex	
내용	명령 위치 오버 플로우/언더 플로우 발생 중에 위치 결정, 오버 플로우/언더 플로우 방향에 대한 명령 또는 방향을 확정할 수 없는 명령을 실행했습니다.				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축 동작 불가. 해당 축이 동작 중에는 감속 정지		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	명령 위치 오버 플로우/언더 플로우 발생 중에 다음을 실행 • 위치 결정 명령을 실행 • 오버 플로우/언더 플로우 방향으로 연속 제어 명령을 실행 • 방향을 확정할 수 없는 명령(동기 기능, 토크 제어)을 실행	이상 해제를 실행하고 원점 복귀 또는 현재 위치 사전 설정을 통해 오버 플로우/언더 플로우 상태를 해제하십시오.	오버 플로우/언더 플로우가 발생하지 않도록 설정하십시오.

부속 정보	없음
주의 사항/ 비고	없음

이벤트명	정방향 한계 입력 중		이벤트 코드	64430000Hex	
내용	정방향 한계 입력이 'ON'인 상태에서 정방향으로 동작하는 명령을 실행했습니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹 동작 불가.		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	정방향 한계 입력이 'ON'인 상태에서 정방향으로 동작하는 명령을 실행했거나 정방향 한계 입력이 'ON'인 상태에서 동작 방향을 지정하지 않는 명령을 실행. 정방향 한계 입력이 'ON'인 상태에서 축 그룹 동작 명령을 실행	이상을 해제하고 역방향으로 복귀 동작을 실행하십시오. 축 그룹 동작 명령 실행 중에 이상이 발생한 경우는 이상이 발생한 축 그룹을 무효로 한 후 상기 복귀 동작을 수행하십시오. 다시 이 이상이 발생하면 정방향 한계 신호의 연결 상태, 정방향 한계 입력의 논리 설정, 기동 명령의 실행 조건을 확인하고 오류를 수정하십시오. 논리 설정은 축 파라메터와 슬레이브측의 설정을 모두 확인하십시오.	정방향 한계 신호의 연결 상태, 정방향 한계 입력의 논리 설정, 기동 명령의 실행 조건에 오류가 없는 것을 확인하십시오. 논리 설정은 축 파라메터와 슬레이브측의 설정을 모두 확인하십시오.

부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 논리축 번호
주의 사항/ 비고	없음

이벤트명	역방향 한계 입력 중		이벤트 코드	64440000Hex	
내용	역방향 한계 입력이 'ON'인 상태에서 역방향으로 동작하는 명령을 실행했습니다.				
발생원	모션 제어 기능 모듈	발생원 상세 정보	축/축 그룹	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	발생원 상세 정보가 '축'인 경우: 해당 축 동작 불가 발생원 상세 정보가 '축 그룹'인 경우: 해당 축 그룹의 동작 불가		

시스템 정의 변수	변수명	데이터 형식	명칭
	_MC_AX[*].MFaultLvl.Active	BOOL	축 가벼운 이상 발생 중
	_MC_GRP[*].MFaultLvl.Active	BOOL	축 그룹 가벼운 이상 발생 중

발생 요인과 처리 방법	발생 요인(추정 원인)	처리 대책	재발 방지
	역방향 한계 입력이 'ON'인 상태에서 역방향으로 동작하는 명령을 실행했거나 역방향 한계 입력이 'ON'인 상태에서 동작 방향을 지정하지 않는 명령을 실행. 역방향 한계 입력이 'ON'인 상태에서 축 그룹 동작 명령을 실행	이상을 해제하고 정방향으로 복귀 동작을 실행하십시오. 축 그룹 동작 명령 실행 중에 이상이 발생한 경우는 이상이 발생한 축 그룹을 무효로 한 후 상기 복귀 동작을 수행하십시오. 다시 이 이상이 발생하면 역방향 한계 신호의 연결 상태, 역방향 한계 입력의 논리 설정, 기동 명령의 실행 조건을 확인하고 오류를 수정하십시오. 논리 설정은 축 파라메터와 슬레이브측의 설정을 모두 확인하십시오.	역방향 한계 신호의 연결 상태, 역방향 한계 입력의 논리 설정, 기동 명령의 실행 조건에 오류가 없는 것을 확인하십시오. 논리 설정은 축 파라메터와 슬레이브측의 설정을 모두 확인하십시오.

부속 정보	부속 정보 1: 발생원 상세 정보에 따라 다름 축인 경우: '0' 축 그룹인 경우: 논리축 번호
주의 사항/ 비고	없음

이벤트명	서보 주회로 전원 OFF 상태			이벤트 코드	74220000Hex	
내용	서보 드라이버의 주회로 전원이 'OFF'인 상태에서 서보 ON을 실행했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	가벼운 이상	복귀 방법	이상 해제	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	해당 축의 서보 'OFF'		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].MFaultLvl.Active		BOOL		축 가벼운 이상 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	서보 드라이버의 주회로 전원이 'OFF'인 상태에서 서보 ON을 실행		이 이상이 발생한 축의 서보 드라이버 주회로 전원을 'ON'한 후 서보 ON을 실행하십시오.		서보 드라이버의 주회로 전원을 'ON'한 후 서보 ON을 실행하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

이벤트명	피드백 위치 오버 플로우/언더 플로우 발생 중			이벤트 코드	57220000Hex *1	
내용	피드백 위치 오버 플로우/언더 플로우 발생 중에 실행할 수 없는 명령을 기동했습니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	피드백 위치 오버 플로우 또는 언더 플로우 발생 중에 실행할 수 없는 명령을 기동함		이상 해제를 실행하고 '현재 위치 변경' 또는 '원점 복귀'를 통해 오버 플로우 또는 언더 플로우 상태를 해제하십시오.		오버 플로우 또는 언더 플로우가 발생하지 않도록 프로그램을 작성하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	스위치 구조체의 트랙 번호 설정 범위 밖			이벤트 코드	57230000Hex *1	
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'TrackNumber'의 값이 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	스위치 구조체의 ON 시작 위치 설정 범위 밖		이벤트 코드	57240000 Hex *1		
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'FirstOnPosition'의 값이 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	스위치 구조체의 ON 종료 위치 설정 범위 밖		이벤트 코드	57250000 Hex *1		
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'LastOnPosition'의 값이 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	스위치 구조체의 방향 선택 범위 밖		이벤트 코드	57260000 Hex *1		
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'AxisDirection'의 값이 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	−	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	스위치 구조체의 스위치 모드 선택 범위 밖		이벤트 코드	57270000Hex *1		
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'CamSwitchMode'의 값이 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	스위치 구조체의 ON 시간 설정 범위 밖		이벤트 코드	57280000Hex *1		
내용	모션 제어 명령의 입출력 변수 'Switches'에 지정된 'Duration'의 값이 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	트랙 옵션 구조체의 ON 시각 보정 설정 범위 밖		이벤트 코드	57290000Hex *1		
내용	모션 제어 명령의 입출력 변수 'TrackOptions'에 지정된 'OnCompensation'의 값이 범위 밖입니다.					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오.		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오.	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	트랙 옵션 구조체의 OFF 시각 보정 설정 범위 밖		이벤트 코드	572A0000 Hex *1	
내용	모션 제어 명령의 입출력 변수 'TrackOptions' 에 지정된 'OffCompensation' 의 값이 범위 밖입니다 .				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버 값이 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 멤버가 유효 범위를 넘지 않도록 값을 수정하십시오 .		명령의 입출력 변수로 지정하는 구조체 형식 변수의 멤버 값이 유효 범위를 넘지 않도록 하십시오 .	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	스위치 구조체 형식 변수의 배열 요소 수 범위 밖		이벤트 코드	572B0000 Hex *1	
내용	모션 제어 명령의 입출력 변수 'Switches' 에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다 .				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 수정하십시오 .		명령의 입출력 변수로 지정하는 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 하십시오 .	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	출력 신호 구조체 형식 변수의 배열 요소 수 범위 밖		이벤트 코드	572C0000 Hex *1	
내용	모션 제어 명령의 입출력 변수 'Outputs' 에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다 .				
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 수정하십시오 .		명령의 입출력 변수로 지정하는 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 하십시오 .	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	트랙 옵션 구조체 형식 변수의 배열 요소 수 범위 밖		이벤트 코드		572D0000Hex *1	
내용	모션 제어 명령의 입출력 변수 'TrackOptions' 에 지정된 구조체 형식 변수의 배열 요소 수가 범위 밖입니다 .					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위 밖		해당 명령의 입출력 변수로 지정한 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 수정하십시오 .		명령의 입출력 변수로 지정하는 구조체 형식 변수의 배열 요소 수가 범위를 넘지 않도록 하십시오 .	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	출력 신호와 트랙 옵션의 배열 요소 수 불일치		이벤트 코드		572E0000Hex *1	
내용	모션 제어 명령의 입출력 변수 'Outputs' 와 'TrackOptions' 에 지정된 구조체 형식 변수의 배열 요소 수가 일치하지 않습니다 .					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 출력 신호 구조체 형식 변수와 트랙 옵션 구조체 형식 변수의 배열 요소 수가 일치하지 않음		해당 명령의 입출력 변수로 지정한 출력 신호 구조체 형식 변수와 트랙 옵션 구조체 형식 변수의 배열 요소 수가 일치하도록 수정하십시오 .		명령의 입출력 변수로 지정하는 출력 신호 구조체 형식 변수와 트랙 옵션 구조체 형식 변수의 배열 요소 수가 일치하도록 하십시오 .	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

이벤트명	스위치 구조체의 동일 트랙 지정 수 범위 밖		이벤트 코드		57310000Hex *1	
내용	모션 제어 명령의 입출력 변수 'Switches' 에 지정된 'TrackNumber' 의 동일 트랙 번호 수가 범위 밖입니다 .					
발생원	모션 제어 기능 모듈		발생원 상세 정보	축	검출 타이밍	명령 기동 시
이상의 속성	중요도	감시 정보	복귀 방법	–	로그 카테고리	시스템
발생의 영향	사용자 프로그램	계속	동작	영향 없음		
시스템 정의 변수	변수명		데이터 형식		명칭	
	_MC_AX[*].Obsr.Active		BOOL		축 감시 정보 발생 중	
발생 요인과 처리 방법	발생 요인(추정 원인)		처리 대책		재발 방지	
	명령의 입출력 변수로 지정한 스위치 구조체 형식 변수의 'TrackNumber' 에 지정된 동일 트랙 번호의 수가 1 트랙에 지정 가능한 개수의 범위 밖		'TrackNumber' 에 지정된 동일 트랙 번호의 수가 1 트랙에 지정할 수 있는 개수를 넘지 않도록 값을 수정하십시오 .		'TrackNumber' 에 지정된 동일 트랙 번호의 수가 1 트랙에 지정할 수 있는 개수를 넘지 않도록 하십시오	
부속 정보	없음					
주의 사항/ 비고	없음					

*1. CPU 유니트의 유니트 버전 Ver.1.06에서 추가된 이벤트입니다.

A-3 다중 기동의 가능 여부

모션 명령의 다중 기동 가능 여부는 현재 축의 상태 또는 현재 축 그룹의 상태 및 실행하는 명령의 종류에 따라 결정됩니다. 여기서는 이들의 관계에 대해 설명합니다.
또한 모션 명령의 다중 기동의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편 (SBCE-363)"을 참조하십시오.

사용상의 주의 사항

- 단축 하나에서 버퍼할 수 있는 다중 기동은 하나뿐입니다. 2개 이상의 명령을 다중 기동하려고 하면 명령에서 이상이 발생합니다.
- 축 그룹 하나에서 버퍼할 수 있는 다중 기동은 7개까지입니다. 8개 이상의 명령을 다중 기동하려고 하면 명령에서 이상이 발생합니다.
- 축 명령의 명령에 의해 동작하는 축에 대해 축 그룹 명령을 다중 기동할 수 없습니다. 또한 축 그룹 명령의 명령에 의해 동작하는 축에 대해 축 명령을 다중 기동할 수 없습니다. 이들 다중 기동을 수행하려고 한 경우 명령에서 이상이 발생합니다.

A-3-1 축의 상태, 축 그룹의 상태

다중 기동의 가능 여부는 현재 축의 상태 또는 현재 축 그룹의 상태에 따라 다릅니다. 축의 상태, 축 그룹의 상태는 해당하는 축의 축 변수 및 축 그룹 변수의 값으로 알 수 있습니다.
또한 축의 상태, 축 그룹의 상태, 축 변수, 축 그룹 변수의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

축 상태를 나타내는 축 변수

축 상태를 나타내는 축 변수에는 다음 9종류가 있습니다. 이들은 배타적이며, 항상 이 중 하나만 TRUE가 됩니다.

변수명	데이터 형식	이름	기능
_MC_AX[0-63]	_sAXIS_REF	축 변수	
Status	_sAXIS_REF_STA	축 Status	
Disabled	BOOL	축 무효	축이 서보 OFF이며 정지한 상태이면 TRUE가 됩니다.
Standstill	BOOL	정지 중	축이 서보 ON이며 정지한 상태이면 TRUE가 됩니다.
Descrete	BOOL	위치 결정 동작 중	목표 위치를 향해 위치 결정 제어를 실행 중일 때 TRUE가 됩니다. 위치 결정 동작 중에 오버라이드값이 '0'으로 설정되어 속도가 '0'인 상태도 포함됩니다.
Continuous	BOOL	연속 동작 중	목표 위치가 없고 연속 동작 중일 때 TURE가 됩니다. 속도 제어 또는 토크 제어일 때 이 상태가 됩니다. 목표 속도를 '0'으로 설정해 속도가 '0'인 상태나 연속 동작 중인 오버라이드값을 '0'으로 설정해 속도가 '0'인 상태도 포함됩니다.
Synchronized	BOOL	동기 동작 중	동기 제어를 실행 중일 때 TURE가 됩니다. 동기 제어 명령 전환 후의 동기 대기 상태도 포함됩니다.
Homing	BOOL	원점 복귀 중	MC_Home 명령, MC_HomeWithParameter 명령에 의해 원점 복귀를 실행했을 때 TRUE가 됩니다.
Stopping	BOOL	감속 정지 중	MC_Stop 명령, MC_TouchProbe 명령에 따라 축이 정지할 때까지의 상태일 때 TRUE가 됩니다. MC_Stop 명령으로 축이 정지한 후 Execute가 TRUE일 때도 포함됩니다. 이 상태에서는 축 동작 명령을 기동할 수 없습니다 (CommandAborted=TRUE).
ErrorStop	BOOL	에러 감속 정지 중	MC_ImmediateStop 명령 또는 축 가벼운 이상 발생 중 (_MC_AX[XX].MFaultLvl.Active=TRUE)으로 축이 정지할 때까지 및 정지하고 있을 때 TRUE가 됩니다. 이 상태에서는 축 동작 명령을 기동할 수 없습니다 (CommandAborted=TRUE).
Coordinated	BOOL	다축 협조 동작 중	다축 협조 명령에 의해 축 그룹 유효가 되었을 때 TRUE가 됩니다.

축 그룹의 상태를 나타내는 축 그룹 변수

축 그룹의 상태를 나타내는 축 그룹 변수에는 다음 5종류가 있습니다. 이들은 배타적이며, 항상 이 중 하나만 TRUE가 됩니다.

변수명	데이터 형식	이름	기능
_MC_GRP[0-31]	_sGROUP_REF	축 그룹 변수	
Status	_sGROUP_REF_STA	축 그룹 Status	
Disabled	BOOL	축 그룹 무효	축 그룹 무효로 정지 중일 때 TRUE가 됩니다.
Standby	BOOL	정지 중	축 그룹 동작 명령이 정지 중일 때 TRUE가 됩니다. 축 그룹 구성축의 서보 ON/OFF 상태와 상관 없습니다.
Moving	BOOL	동작 중	목표 위치를 향해 축 그룹 동작 명령을 실행 중일 때 TRUE가 됩니다. 인포지션 대기 상태나 오버라이드에 의해 속도가 '0'인 상태도 포함됩니다.
Stopping	BOOL	감속 정지 중	MC_GroupStop 명령에 따라 축 그룹이 정지할 때까지의 상태일 때 TRUE가 됩니다. MC_GroupStop 명령으로 축 그룹이 정지한 후 Execute가 TRUE일 때도 포함됩니다. 이 상태에서는 축 동작 명령을 기동할 수 없습니다 (CommandAborted=TRUE).
ErrorStop	BOOL	에러 감속 정지 중	MC_GroupImmediateStop 명령 또는 축 그룹 가벼운 이상 발생(_MC_GRP [XX]. MFaultLvl. Active=TRUE)으로 축 그룹이 정지할 때까지 및 정지하고 있을 때 TRUE가 됩니다. 이 상태에서는 축 그룹 동작 명령은 기동할 수 없습니다(CommandAborted=TRUE).

A-3-2 상태 이동과 다중 기동의 가능 여부

각 축의 상태 및 각 축 그룹의 상태에서의 다중 기동의 가능 여부를 표시합니다. 그와 함께 각 축의 상태 및 각 축 그룹의 상태의 이동 대상도 표시합니다.
또한 모션 제어 기능 모듈의 상태 이동과 상태 이동에 관계된 모션 제어 명령의 자세한 내용은 "NJ 시리즈 CPU 유니트 사용 설명서 모션 제어편(SBCE-363)"을 참조하십시오.

서보 축, 가상 서보 축에 대한 명령의 다중 기동

명령의 대상이 서보 축 및 서보 축 그룹인 경우의 다중 기동의 가능 여부와 상태 이동 대상이 다음 표에 나와 있습니다.
다중 기동의 가능 여부는 표의 셀의 색상으로 표시합니다.

백색 : 다중 기동 가능
회색 : 다중 기동 불가능. 이상이 발생합니다.
황색 : 다중 기동 불가능. CommandAborted(실행 중단)가 TRUE가 되고 명령 실행이 무효가 됩니다.

또한 표 안의 기호는 다음과 같이 이동대상의 상태를 나타냅니다.
축의 상태

A : 축 무효(Disabled)
B : 정지 중(Standstill)
C : 위치 결정 동작 중(Discrete)
D : 연속 동작 중(Continuous)
E : 동기 동작 중(Synchronized)
F : 원점 복귀 중(Homing)
G : 감속 동작 중(Stopping)
H : 에러 감속 정지 중(ErrorStop)
I : 다축 협조 동작 중(Coordinated)

축 그룹의 상태

a : 축 그룹 무효(Disabled)
b : 정지 중(Standby)
c : 동작 중(Moving)
d : 감속 동작 중(Stopping)
e : 에러 감속 정지 중(ErrorStop)

● 축 명령

명령	명령 실행 전의 축 상태									명령 실행 전의 축 그룹 상태				
	A	B	C	D	E	F	G	H	I	a	b	c	d	e
	축 무효	정지 중	위치 결정 동작 중	연속 동작 중	동기 동작 중	원점 복귀 중	감속 정지 중	에러 감속 정지 중	다축 협조 동작 중	축 그룹 무효	정지 중	동작 중	감속 정지 중	에러 감속 정지 중
MC_Power(Enable=TRUE)	*1	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_Power(Enable=FALSE)	A	A	A	A	A	A	A	H	I	a	b	*2	d	e
MC_MoveJog	H	D	H	H	H	H	G	H	I	a	e	e	e	e
MC_Home	H	F	H	H	H	H	G	H	I	a	e	e	e	e
MC_HomeWithParameter	H	F	H	H	H	H	G	H	I	a	e	e	e	e
MC_Move	H	C	C	C	C	H	G	H	I	a	e	e	e	e
MC_MoveAbsolute	H	C	C	C	C	H	G	H	I	a	e	e	e	e
MC_MoveRelative	H	C	C	C	C	H	G	H	I	a	e	e	e	e
MC_MoveVelocity	H	D	D	D	D	H	G	H	I	a	e	e	e	e
MC_MoveZeroPosition	H	C	C	C	C	H	G	H	I	a	e	e	e	e
MC_MoveFeed	H	*3	*3	*3	*3	H	G	H	I	a	e	e	e	e
MC_Stop	A	G	G	G	G	G	G	H	I	a	e	e	e	e
MC_ImmediateStop	H	H	H	H	H	H	H	H	I	a	e	e	e	e
MC_SetPosition	A	B	F*4	D*5	H	H	G	H	I	a	e	e	e	e
MC_SetOverride	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_ResetFollowingError	H	B	G	G	G	H	G	H	I	a	e	e	e	e
MC_CamIn	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_CamOut	H	H	H	H	C*6	H	G	H	I	a	e	e	e	e
MC_GearIn	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_GearInPos	H	E	H	H	H	H	G	H	I	a	e	e	e	e
MC_GearOut	H	H	H	H	C*7	H	G	H	I	a	e	e	e	e
MC_MoveLink	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_CombineAxes	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_Phasing	H	H	H	H	E*8	H	G	H	I	a	e	e	e	e
MC_TorqueControl	H	D	D	D	D	H	G	H	I	a	e	e	e	e
MC_SetTorqueLimit	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_ZoneSwitch	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_TouchProbe	A	B	C	D	E	H	G	H	I	a	e	e	e	e
MC_AbortTrigger	A	B	C*9	D*5	E	H	G	H	I	a	e	e	e	e
MC_AxesObserve	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_SyncMoveVelocity	H	D	D	D	D	H	G	H	I	a	e	e	e	e
MC_SyncMoveAbsolute	H	C	C	C	C	H	G	H	I	a	e	e	e	e
MC_Reset	A	B	C	D	E	F	G	*10	I	a	b	c	d	e
MC_ChangeAxisUse (미사용 축→사용 축)	A	–	–	–	–	–	–	–	–	–	–	–	–	–
MC_ChangeAxisUse (사용 축→미사용 축)	–	*11	H	H	H	H	H	H	H	a	b	c	d	e
MC_DigitalCamSwitch	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_TimeStampToPos	A	B	C	D	E	F	G	H	I	a	b	c	d	e

*1. MC_Stop의 Execute가 FALSE이면 B: 정지 중에 이동합니다. TRUE라면 G: 감속 정지 중에 이동합니다.
*2. 최대 감속도에 따라 b: 정지 중 또는 d: 감속 정지 중에 이동합니다.

*3. 목표 위치를 설정하고 있는 경우는 C: 위치 결정 동작 중에 이동합니다. 그렇지 않은 경우는 D: 연속 동작 중에 이동합니다.
*4. MC_MoveZeroPosition이 실행 중인 경우는 이상 상태가 되며, 상태 이동대상은 H: 에러 감속 정지 중이 됩니다.
*5. MC_TorqueControl 또는 MC_SyncMoveVelocity가 실행 중인 경우는 이상 상태가 되며, 상태 이동대상은 H: 에러 감속 정지 중이 됩니다.
*6. MC_CamIn이 실행 중인 경우는 이상 상태가 되며, 상태 이동대상은 H: 에러 감속 정지 중이 됩니다.
*7. MC_GearIn 또는 MC_GearInPos가 실행 중인 경우는 이상 상태가 되며, 상태 이동대상은 H: 에러 감속 정지 중이 됩니다.
*8. MC_CombineAxes가 실행 중인 경우는 이상 상태가 되며, 상태 이동대상은 H: 에러 감속 정지 중이 됩니다.
*9. MC_AbortTrigger의 LatchID 가 실행 중인 MC_MoveFeed의 LatchID 와 중복된 경우는 이상 상태가 되며, 상태 이동대상은 H: 에러 감속 정지 중이 됩니다.
*10. 조건에 따라 상태 이동대상이 다음과 같이 달라집니다.
서보 OFF에서 이상 해제에 성공한 경우 또는 서보 OFF에서 MC_Stop의 Execute가 TRUE인 경우 A: 축 무효로 이동합니다.
서보 ON에서 이상 해제에 성공한 경우 B: 정지 중에 이동합니다.
서보 ON에서 MC_Stop의 Execute가 TRUE인 경우 G: 감속 정지 중에 이동합니다.
서보 ON/OFF에 관계 없이 이상 해제에 실패한 경우 H: 에러 감속 정지 중에 이동합니다.
*11. 축 변수인 _MC_AX[ ].Details.VelLimit(명령 속도 포화)가 TRUE인 경우, H: 에러 감속 정지 중으로 변경됩니다. FALSE인 경우, 미사용 축으로 전환되므로 −: 축의 상태 없음이 됩니다.

● 축 그룹 명령

명령	명령 실행 전의 축 상태									명령 실행 전의 축 그룹 상태				
	A	B	C	D	E	F	G	H	I	a	b	c	d	e
	축 무효	정지 중	위치 결정 동작 중	연속 동작 중	동기 동작 중	원점 복귀 중	감속 정지 중	에러 감속 정지 중	다축 협조 동작 중	축 그룹 무효	정지 중	동작 중	감속 정지 중	에러 감속 정지 중
MC_GroupEnable	I	I	C	D	E	F	G	H	I	b	b	c	d	e*1
MC_GroupDisable	A	B	C	D	E	F	G	H	*2	a	a	a	a	e
MC_MoveLinear	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_MoveLinearAblolute	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_MoveLinearRelative	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_MoveCircular2D	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_GroupStop	A	B	C	D	E	F	G	H	I	e	d	d	d	e
MC_GroupImmediateStop	A	B	C	D	E	F	G	H	I	e	e	e	e	e
MC_GroupSetOverride	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_GroupReadPosition	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_ChangeAxesInGroup	A	B	C	D	E	F	G	H	I	a*3	e	e	e	e*4
MC_GroupSyncMoveAbsolute	A	B	C	D	E	F	G	H	I	e	c	c	d	e
MC_GroupReset	A	B	C	D	E	F	G	H	I	a	b	c	d	*5

*1. 서보 OFF의 경우 CommandAborted(실행 중단)가 TRUE가 됩니다.
*2. 조건에 따라 상태 이동대상이 다음과 같이 달라집니다.
서보 OFF에서 이상이 발생하지 않은 경우 A: 축 무효로 이동합니다.
서보 OFF에서 MC_GroupStop의 Execute가 FALSE인 경우 B: 정지 중에 이동합니다.
서보 ON에서 MC_GroupStop의 Execute가 TRUE인 경우 G: 감속 정지 중에 이동합니다.
서보 ON/OFF에 관계 없이 이상이 발생한 경우 H: 에러 감속 정지 중에 이동합니다.
*3. MC_GroupEnable이 실행 중인 경우는 이상 상태가 되며, 상태 이동대상은 e: 에러 감속 정지 중이 됩니다.
*4. 서보 ON인 경우 이상 상태가 됩니다.

*5. 조건에 따라 상태 이동대상이 다음과 같이 달라집니다.
서보 OFF에서 이상 해제에 성공한 경우 a: 축 그룹 무효로 이동합니다.
서보 ON에서 이상 해제에 성공한 경우 b: 정지 중에 이동합니다.
서보 ON에서 MC_GroupStop의 Execute가 TRUE인 경우 d: 감속 정지 중에 이동합니다.

● 공통 명령

공통 명령은 축의 상태 또는 현재 축 그룹의 상태에 관계 없이 항상 다중 기동이 가능합니다.
또한 공통 명령을 실행해서 축의 상태, 현재 축 그룹의 상태가 다른 상태로 이동되는 일은 없습니다. 현재 상태가 유지됩니다.

엔코더 축, 가상 엔코더 축에 대한 명령의 다중 기동

명령의 대상이 엔코더 축 및 엔코더 축 그룹인 경우의 상태 이동과 다중 기동의 가능 여부가 다음 표에 나와 있습니다.

다중 기동의 가능 여부는 표의 셀의 색상으로 표시합니다.

백색 : 다중 기동 가능
회색 : 다중 기동 불가능. 이상이 발생합니다.
황색 : 다중 기동 불가능. CommandAborted(실행 중단)가 TRUE가 되고 명령 실행이 무효가 됩니다.

또한 표 안의 기호는 다음과 같이 이동대상의 상태를 나타냅니다.
축의 상태

A : 축 무효(Disabled)
H : 에러 감속 정지 중(ErrorStop)

● 축 명령

명령	명령 실행 전의 축 상태	
	A	H
	축 무효	에러 감속 정지 중
MC_Power	H	H
MC_MoveJog	H	H
MC_Home	H	H
MC_HomeWithParameter	H	H
MC_Move	H	H
MC_MoveAbsolute	H	H
MC_MoveRelative	H	H
MC_MoveVelocity	H	H
MC_MoveZeroPosition	H	H
MC_MoveFeed	H	H
MC_Stop	H	H
MC_ImmediateStop	H	H
MC_SetPosition	A	H
MC_SetOverride	H	H
MC_ResetFollowingError	H	H
MC_CamIn	H	H
MC_CamOut	H	H
MC_GearIn	H	H
MC_GearInPos	H	H
MC_GearOut	H	H
MC_MoveLink	H	H
MC_CombineAxes	H	H
MC_Phasing	H	H
MC_TorqueControl	H	H
MC_SetTorqueLimit	H	H
MC_ZoneSwitch	A	H
MC_TouchProbe	A	H
MC_AbortTrigger	A	H
MC_AxesObserve	A	H
MC_SyncMoveVelocity	H	H
MC_SyncMoveAbsolute	H	H
MC_Reset	A	A
MC_ChangeAxisUse (미사용 축→사용 축)	A	H
MC_ChangeAxisUse (사용 축→미사용 축)	–	H
MC_DigitalCamSwitch	A	H
MC_TimeStampToPos	A	H

● 축 그룹 명령

엔코더 축, 가상 엔코더 축은 축 그룹 구성축으로 설정할 수 없습니다. 따라서 엔코더 축, 가상 엔코더 축에 대해 축 그룹 명령을 다중 기동하려고 하면 모두 이상 상태가 됩니다.

● 공통 명령

공통 명령은 축의 상태에 관계 없이 항상 다중 기동이 가능합니다.
또한 공통 명령을 실행해서 축의 상태가 다른 상태로 이동되는 일은 없습니다. 현재 상태가 유지됩니다.

A-4 버전 관련 정보

여기에는 CPU 유니트와 Sysmac Studio의 각 버전 갱신에 관련된 정보가 나와 있습니다.

버전 갱신에 의해 사양이 변경된 명령 및 신규 명령 목록

명령에는 CPU 유니트, Sysmac Studio의 각 버전에 따라 대응 상황이나 사양이 다른 경우가 있습니다. 해당 내용은 다음과 같습니다.
○: 해당 버전에서 새로 추가
△: 해당 버전에서 사양 변경

종류	명령	이름	버전		페이지
			CPU 유니트	Sysmac Studio	
공통 명령	MC_Write	MC 설정 쓰기	△ Ver.1.01	△ Ver.1.02	5-13
축 그룹 명령	MC_GroupReadPosition	축 그룹 위치 취득	○ Ver.1.01	○ Ver.1.02	4-83
	MC_ChangeAxesInGroup	축 그룹 구성축 쓰기			4-87
	MC_GroupSyncMoveAbsolute	축 그룹 주기적 동기 절대 위치 제어			4-91
축 명령	MC_GearIn	기어 동작 시작	△ Ver.1.02	△ Ver.1.03	3-215
	MC_GearInPos	위치 지정 기어 동작			3-235
	MC_HomeWithParameter	파라메터 지정 원점 복귀	○ Ver.1.03	○ Ver.1.04	3-36
	MC_SyncMoveAbsolute	주기적 동기 절대 위치 결정			3-363
	MC_ChangeAxisUse	축 사용 변경	○ Ver.1.04	○ Ver.1.05	3-373
	MC_DigitalCamSwitch	디지털 캠 스위치 유효	○ Ver.1.06	○ Ver.1.07	3-377
	MC_TimeStampToPos	타임 스탬프→축 위치 산출	○ Ver.1.06	○ Ver.1.07	3-393

NX 시리즈 위치 인터페이스 유니트를 사용하는 경우

NX 시리즈 위치 인터페이스 유니트는 CPU 유니트 Ver.1.05 이상과 Sysmac Studio Ver.1.06 이상의 조합에서 사용할 수 있습니다.
모션 제어 명령을 NX 시리즈 위치 인터페이스 유니트에 대해 사용하는 경우, 기능에 제약이 있는 명령이나 사용할 수 없는 명령이 있습니다. 자세한 내용은 "NX 시리즈 위치 인터페이스 유니트 사용 설명서 (SBCE-374)"를 참조하십시오.

I

색인

색인

번호

2축 원호 보간 4-43

가

가감산 위치 결정 3-283
강제 정지 3-129, 4-67
고속 원점 복귀 3-95
기어 동작 3-235
기어 동작 시작 3-215
기어 동작 해제 3-257

다

디지털 캠 스위치 유효 3-377

사

사다리꼴 패턴 캠 3-262
상대값 위치 결정 3-73
상대값 직선 보간 4-40
설정 쓰기 5-13
속도 제어 3-80

아

에러 리셋 3-369, 4-97
에러 코드
3461Hex A-2, A-21
5420Hex A-2, A-21
5421Hex A-2, A-22
5422Hex A-2, A-22
5423Hex A-3, A-22
5424Hex A-3, A-23
5425Hex A-3, A-23
5427Hex A-3, A-23
5428Hex A-3, A-24
5429Hex A-3, A-24
542AHex A-3, A-24
542BHex A-3, A-25
542CHex A-3, A-25
542DHex A-3, A-25
542EHex A-3, A-26
542FHex A-4, A-26
5430Hex A-4, A-26
5431Hex A-4, A-27
5432Hex A-4, A-27
5433Hex A-4, A-28
5434Hex A-4, A-28
5435Hex A-4, A-28
5436Hex A-4, A-29
5437Hex A-4, A-29
5438Hex A-4, A-29
5439Hex A-5, A-30
543AHex A-5, A-30
543BHex A-5, A-31
543CHex A-5, A-31
543DHex A-5, A-32
543EHex A-5, A-32
543FHex A-5, A-32
5440Hex A-5, A-33
5441Hex A-6, A-33
5442Hex A-6, A-34
5443Hex A-6, A-34
5444Hex A-6, A-35
5445Hex A-6, A-35
5446Hex A-6, A-36
5447Hex A-6, A-36
5448Hex A-6, A-36
5449Hex A-7, A-37
544AHex A-7, A-37
544CHex A-7, A-37
544DHex A-7, A-38
544EHex A-7, A-38
544FHex A-7, A-38
5450Hex A-7, A-39
5451Hex A-7, A-39
5453Hex A-7, A-39
5454Hex A-8, A-40
5455Hex A-8, A-40
5456Hex A-8, A-41
5457Hex A-8, A-41
5458Hex A-8, A-42
5459Hex A-8, A-42
545AHex A-8, A-43
545BHex A-8, A-43
545CHex A-8, A-43
545DHex A-9, A-44
545EHex A-9, A-44
545FHex A-9, A-44
5460Hex A-9, A-45
5461Hex A-9, A-45
5462Hex A-9, A-45
5463Hex A-9, A-46
5464Hex A-9, A-46
5465Hex A-9, A-46
5466Hex A-10, A-47
5467Hex A-10, A-47
5468Hex A-10, A-48
5469Hex A-10, A-48
546AHex A-10, A-48
546BHex A-10, A-49
546CHex A-10, A-49
546DHex A-10, A-49
546EHex A-10, A-50
546FHex A-11, A-50
5470Hex A-11, A-51
5471Hex A-11, A-51
5472Hex A-11, A-51
5474Hex A-11, A-52
5475Hex A-11, A-52
5476Hex A-11, A-52
5478Hex A-11, A-53
5479Hex A-11, A-53
547AHex A-11, A-54
547BHex A-12, A-54
547CHex A-12, A-54
547DHex A-12, A-55
547EHex A-12, A-55
547FHex A-12, A-56
5480Hex A-12, A-56
5481Hex A-12, A-56
5482Hex A-12, A-57
5483Hex A-13, A-57

5484Hex A-13, A-57
5487Hex A-13, A-58
5488Hex A-13, A-58
5489Hex A-13, A-58
548AHex A-13, A-59
548BHex A-13, A-59
548CHex A-13, A-59
548DHex A-13, A-60
548EHex A-14, A-60
548FHex A-14, A-60
5490Hex A-14, A-61
5491Hex A-14, A-61
5492Hex A-14, A-61
5493Hex A-14, A-62
5494Hex A-14, A-62
5495Hex A-14, A-62
5702Hex A-67
5720Hex A-68
5721Hex A-69
6440Hex A-16, A-70
6441Hex A-17, A-71
6442Hex A-17, A-71
6443Hex A-17, A-72
6444Hex A-17, A-72
7422Hex A-17, A-73
오버라이드값 설정 3-149, 4-79
외부 래치 무효 3-344
외부 래치 유효 3-324
운전 가능 3-3
원점 복귀 3-15
원호 보간 4-43
위상 보정 3-294
위치 결정 3-41, 3-47, 3-73, 3-102
위치 지정 기어 동작 3-235
이벤트 코드
549DHex A-65
549EHex A-66
인터럽트 피딩 위치 결정 3-102

자

전자 기어 3-235
절대값 위치 결정 3-47
절대값 직선 보간 4-37
조그 이송 3-8
존 감시 3-318
주기적 동기 속도 제어 3-354
주축 상대값 위상 보정 3-294
즉시 정지 3-138
직선 보간 4-11, 4-37, 4-40

차

축 그룹 강제 정지 4-67
축 그룹 구성축 쓰기 4-87
축 그룹 무효 4-6
축 그룹 에러 리셋 4-97
축 그룹 오버라이드값 설정 4-79
축 그룹 위치 취득 4-83
축 그룹 유효 4-2
축 그룹 주기적 동기 절대 위치 제어 4-91
축 그룹 즉시 정지 4-75
축 사용 변경 3-373
축 에러 리셋 3-369
축간 편차 감시 3-348

카

캠 동작 시작 3-161
캠 동작 해제 3-210
캠 테이블 속성 갱신 5-2
캠 테이블 저장 5-8

타

타임 스탬프→위치 산출 3-393
토크 제어 3-300
토크 제한 3-311

파

파라메터 지정 원점 복귀 3-36
편차 카운터 리셋 3-154

하

현재 위치 변경 3-143

M

MC 설정 쓰기 5-13
MC_AbortTrigger(외부 래치 무효) 3-344
MC_AxesObserve(축간 편차 감시) 3-348
MC_CamIn(캠 동작 시작) 3-161
MC_CamOut(캠 동작 해제) 3-210
MC_ChangeAxesInGroup(축 그룹 구성축 쓰기) 4-87
MC_ChangeAxisUse(축 사용 변경) 3-373
MC_CombineAxes(가감산 위치 결정) 3-283
MC_DigitalCamSwitch 3-377
MC_GearIn(기어 동작 시작) 3-215
MC_GearInPos(위치 지정 기어 동작) 3-235
MC_GearOut(기어 동작 해제) 3-257
MC_GroupDisable(축 그룹 무효) 4-6
MC_GroupEnable(축 그룹 유효) 4-2
MC_GroupImmediateStop(축 그룹 즉시 정지) 4-75
MC_GroupReset(축 그룹 에러 리셋) 4-97
MC_GroupSetOverride(축 그룹 오버라이드값 설정) ... 4-79
MC_GroupStop(축 그룹 강제 정지) 4-67
MC_GroupSyncMoveAbsolute(축 그룹 주기적 동기 절대 위치 제어) 4-91
MC_Home(원점 복귀) 3-15
MC_HomeWithParameter(파라메터 지정 원점 복귀) .. 3-36
MC_ImmediateStop(즉시 정지) 3-138
MC_Move(위치 결정) 3-41
MC_MoveAbsolute(절대값 위치 결정) 3-47
MC_MoveCircular2D(2축 원호 보간) 4-43
MC_MoveFeed(인터럽트 피딩 위치 결정) 3-102
MC_MoveJog(조그 이송) 3-8
MC_MoveLinear(직선 보간) 4-11
MC_MoveLinearAbsolute(절대값 직선 보간) 4-37
MC_MoveLinearRelative(상대값 직선 보간) 4-40
MC_MoveLink(사다리꼴 패턴) 3-262
MC_MoveRelative(상대값 위치 결정) 3-73

MC_MoveVelocity(속도 제어)..... 3-80
MC_MoveZero(고속 원점 복귀)..... 3-95
MC_Phasing(주축 상대값 위상 보정)..... 3-294
MC_Power(운전 가능)..... 3-3
MC_Reset(축 에러 리셋)..... 3-369
MC_ResetFollowingError(편차 카운터 리셋)..... 3-154
MC_SaveCamTable(캠 테이블 저장) 5-8
MC_SetCamTableProperty(캠 테이블 속성 갱신) 5-2
MC_SetOverride(오버라이드값 설정) 3-149
MC_SetPosition(현재 위치 변경)..... 3-143
MC_SetTorqueLimit(토크 제한)..... 3-311
MC_Stop(강제 정지)..... 3-129
MC_SyncMoveAbsolute(주기적 동기 절대 위치 3-363
MC_SyncMoveVelocity(주기적 동기 속도 제어)..... 3-354
MC_TimeStampToPos..... 3-393
MC_TorqueControl(토크 제어) 3-300
MC_TouchProbe(외부 래치 유효)..... 3-324
MC_Write(MC 설정 쓰기)..... 5-13
MC_ZoneSwitch(존 감시)..... 3-318

Z

エラ?コ?ド
5703Hex A-16, A-67
5722Hex A-17, A-73
5723Hex A-18, A-73
5724Hex A-18, A-74
5725Hex A-18, A-74
5726Hex A-18, A-74
5727Hex A-18, A-75
5728Hex A-18, A-75
5729Hex A-18, A-75
572AHex..... A-18, A-76
572BHex..... A-18, A-76
572CHex..... A-19, A-76
572DHex..... A-19, A-77
572EHex..... A-19, A-77
572FHex A-16, A-69
5730Hex A-16, A-70
5731Hex A-19, A-77

- 이 사용 설명서에 기재되지 않은 조건이나 환경에서 사용하거나 원자력 제어, 철도, 항공, 차량, 연소 장치, 의료 기기, 오락 기계, 안전 기기, 그 밖에 인명이나 재산에 큰 영향을 미치는 등 안전성이 각별히 요구되는 용도로 사용할 것을 검토하는 경우는 정격이나 성능에 여유가 있는 사용 방법이나 장애 시 안전 대책 등을 마련하는 한편, 당사 영업 담당자에게 상담하여 사양서 등을 확인하시기 바랍니다.
- 본 제품 중 외환 및 해외 무역법의 규정에 따라 수출허가, 승인 대상 화물(또는 기술)에 해당하는 것을 수출(또는 비거주자에게 제공)하는 경우, 동법에 규정된 수출 허가, 승인(또는 직무 거래 허가)이 필요합니다.

Industrial Web ▶ http://www.ia.omron.co.kr

한국 오므론 제어기기 주식회사

137-920 서울시 서초구 서초동 1303-22 교보타워 B동 21층 TEL : 02-3483-7789 FAX : 02-3483-7788

카탈로그 번호 SBCE-364G-K6 2014년 01월 현재

© OMRON Corporation 2011 All Rights Reserved.
예고없이 사양 등을 변경하는 경우가 있으므로 양해 부탁드립니다.