OMRON

マシンオートメーションコントローラ

NJ/NXシリーズ

コマンドリファレンスマニュアル
モーション編

形NX701-1□□□
形NX502-1□□□
形NX102-1□□□
形NX102-90□□
形NX1P2-1□□□□□
形NX1P2-9□□□□□
形NJ501-□□□□
形NJ301-1□□□
形NJ101-10□□

SYSMAC
always in control

SBCE-434L

おことわり

(1) 本マニュアルの内容の一部または全部を無断で複写、複製、転載することを禁じます。

(2) 本マニュアルの内容に関しては、改良のため予告なしに仕様などを変更することがあります。
あらかじめご了承ください。

(3) 本マニュアルの内容に関しては、万全を期しておりますが、万一ご不審な点や誤りなど、お気づきのことがありましたら、お手数ですが、弊社支店または営業所までご連絡ください。
その際、巻末記載の Man.No. （マニュアル No.）も併せてお知らせください。

商標

- Sysmac は、オムロン株式会社 FA 機器製品の日本およびその他の国における商標または登録商標です。
- Microsoft、Windows、Excel、Visual Basic 、Microsoft Edgeは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- EtherCAT®は、ドイツ Beckhoff Automation GmbH によりライセンスされた特許取得済み技術であり登録商標です。
- ODVA、CIP、CompoNet、DeviceNet、EtherNet/IP は ODVA の商標です。
- SD、SDHC ロゴは、SD-3C,LLC の商標です。

その他、本文中に掲載している会社名および製品名は、それぞれ各社の商標または登録商標です。

著作権

- スクリーンショットはマイクロソフトの許諾を得て使用しています。
- 本製品にはサードパーティ製のソフトウェアが組み込まれています。このソフトウェアに関連するライセンスと著作権については、http://www.fa.omron.co.jp/nj_info_j/ をご覧ください。

はじめに

このたびは、NJ/NX シリーズ CPU ユニットをお買い上げいただきまして、ありがとうございます。

このマニュアルは、モーション制御命令について詳しく説明した解説書です。機能、取り扱い方法などについて十分ご理解いただき、モーション制御機能モジュール（以降は MC 機能モジュールと呼びます）を正しくご使用ください。
なお、ご使用の際は、NJ/NX シリーズ CPU ユニットのユーザーズマニュアルを併せてご覧ください。
また、お読みになった後も本マニュアルは大切に保管して、いつも手元においてお使いください。

対象となる読者の方々

本マニュアルは、電気の知識（電気工事士あるいは同等の知識）を有する、次の方を対象に記述しています。
- FA 機器の導入を担当される方
- FA システムを設計される方
- FA 機器を設置、接続される方
- FA 現場を管理される方

また、プログラミング言語に関して、国際標準規格 IEC 61131-3 もしくは国内標準規格 JIS B 3503 に規定される内容をご理解いただいている方を対象に記述しています。

対象となる製品

本マニュアルは以下の製品を対象にしています。
- NX シリーズ CPU ユニット
  - 形 NX701-1□□□
  - 形 NX502-1□□□
  - 形 NX102-1□□□
  - 形 NX102-90□□
  - 形 NX1P2-1□□□□□
  - 形 NX1P2-9□□□□□
- NJ シリーズ CPU ユニット
  - 形 NJ501-□□□□
  - 形 NJ301-1□□□
  - 形 NJ101-10□□

各製品の仕様の一部または制約事項が、他のマニュアルに記載されている場合があります。
「分冊構成 (P.2)」および「関連マニュアル (P.23)」をご確認ください。

分冊構成

本製品のマニュアルは、次表のとおりに分冊されています。目的に応じてお読みいただき、本製品を活用してください。
本製品の操作には、おもにオートメーションソフトウェア Sysmac Studio を使用します。Sysmac Studio については、『Sysmac Studio Version 1 オペレーションマニュアル(SBCA-470)』を参照してください。

使用目的	マニュアル																			
	基本情報																			
	NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX701 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編	NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル 内蔵 I/O、オプションボード機能編	NJ/NX シリーズ コマンドリファレンスマニュアル 基本編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	NJ/NX シリーズ コマンドリファレンスマニュアル モーション編	NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT ポート ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP ポート ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル OPC/UA 編	NX シリーズ CPU ユニット ユーザーズマニュアル FINS 機能編	NJ/NX シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	NJ シリーズ SECS/GEM 搭載 CPU ユニット ユーザーズマニュアル	NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル	NJ シリーズ NJ Robotics CPU ユニット ユーザーズマニュアル	NJ/NY シリーズ NC 統合コントローラ ユーザーズマニュアル	NJ/NX シリーズ トラブルシューティングマニュアル
形 NX701 の概要を知る	○																			
形 NX502 の概要を知る		○																		
形 NX102 の概要を知る			○																	
形 NX1P2 の概要を知る				○																
NJ シリーズの概要を知る					○															
装着、設置、ハードウェア設定を行う	○	○	○	○	○	○														
モーション制御を行う場合										○										
EtherCAT を使用する場合												○								
EtherNet/IP を使用する場合													○							
オムロン製ロボットの制御を行う場合																	○			

使用目的	マニュアル																			
	基本情報																			
	NJ/NX シリーズ トラブルシューティングマニュアル	NJ/NY シリーズ NC 統合コントローラ ユーザーズマニュアル	NJ シリーズ NJ Robotics CPU ユニット ユーザーズマニュアル	NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル	NJ シリーズ SECS/GEM 搭載 CPU ユニット ユーザーズマニュアル	NJ/NX シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	NX シリーズ CPU ユニット ユーザーズマニュアル FINS 機能編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル OPC/UA 編	NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP ポートユーザーズマニュアル	NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT ポート ユーザーズマニュアル	NJ/NX シリーズ コマンドリファレンスマニュアル モーション編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	NJ/NX シリーズ コマンドリファレンスマニュアル 基本編	NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル 内蔵 I/O、オプションボード機能編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編
ソフトウェア設定を行う															○					
モーション制御を行う場合												○								
EtherCAT を使用する場合											○									
EtherNet/IP を使用する場合									○											
OPC UA を使用する場合								○												
FINS を使用する場合							○													
データベース接続サービスを使用する場合						○														
GEM サービスを使用する場合					○															
オムロン製ロボットの制御を行う場合				○																
NJ Robotics 機能でロボット制御を行う場合			○																	
数値制御を行う場合		○																		
形 NX1P2 の機能を使用する場合														○						

使用目的	マニュアル																			
	NJ/NX シリーズ トラブルシューティングマニュアル	NJ/NY シリーズ NC 統合コントローラ ユーザーズマニュアル	NJ シリーズ NJ Robotics CPU ユニット ユーザーズマニュアル	NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル	NJ シリーズ SECS/GEM 搭載 CPU ユニット ユーザーズマニュアル	NJ/NX シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	NX シリーズ CPU ユニット ユーザーズマニュアル FINS 機能編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル OPC/UA 編	NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP ポートユーザーズマニュアル	NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT ポート ユーザーズマニュアル	NJ/NX シリーズ コマンドリファレンスマニュアル モーション編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	NJ/NX シリーズ コマンドリファレンスマニュアル 基本編	NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル 内蔵 I/O、オプションボード機能編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編
													基本情報							
ユーザプログラムを作成する															○	○	○	○	○	○
モーション制御を行う場合											○	○								
EtherCAT を使用する場合										○										
EtherNet/IP を使用する場合									○											
OPC UA を使用する場合								○												
FINS を使用する場合							○													
データベース接続サービスを使用する場合						○														
GEM サービスを使用する場合					○															
オムロン製ロボットの制御を行う場合				○																
NJ Robotics 機能でロボット制御を行う場合			○																	
数値制御を行う場合		○																		
異常処理を組み込む場合	○																			
形 NX1P2 の機能を使用する場合														○						

使用目的	マニュアル																				
	基本情報																				
	NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編	NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル 内蔵 I/O、オプションボード機能編	NJ/NX シリーズ コマンドリファレンスマニュアル 基本編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	NJ/NX シリーズ コマンドリファレンスマニュアル モーション編	NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT ポート ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP ポート ユーザーズマニュアル	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル OPC/UA 編	NX シリーズ CPU ユニット ユーザーズマニュアル FINS 機能編	NJ/NX シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	NJ シリーズ SECS/GEM 搭載 CPU ユニット ユーザーズマニュアル	NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル	NJ シリーズ NJ Robotics CPU ユニット ユーザーズマニュアル	NJ/NY シリーズ NC 統合コントローラ ユーザーズマニュアル	NJ/NX シリーズ トラブルシューティングマニュアル
動作確認やデバッグを行う						○															
モーション制御を行う場合						○				○											
EtherCAT を使用する場合						○					○										
EtherNet/IP を使用する場合						○						○									
OPC UA を使用する場合						○							○								
FINS を使用する場合						○								○							
データベース接続サービスを使用する場合						○									○						
GEM サービスを使用する場合						○										○					
オムロン製ロボットの制御を行う場合						○											○				
NJ Robotics 機能でロボット制御を行う場合						○												○			
数値制御を行う場合						○													○		
形 NX1P2 の機能を使用する場合						○	○														
異常管理機能やトラブル発生時の対処方法を知る *1													△	△	△	△	△	△	△	△	○
保守作業について知る	○	○	○	○	○																
モーション制御を行う場合										○											
EtherCAT を使用する場合											○										
EtherNet/IP を使用する場合												○									

*1. 異常管理の考え方と異常項目については、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。ただし、△ マークがついているマニュアルの製品に特有の異常の詳細については、△ マークがついているマニュアルを参照してください。

マニュアルの読み方

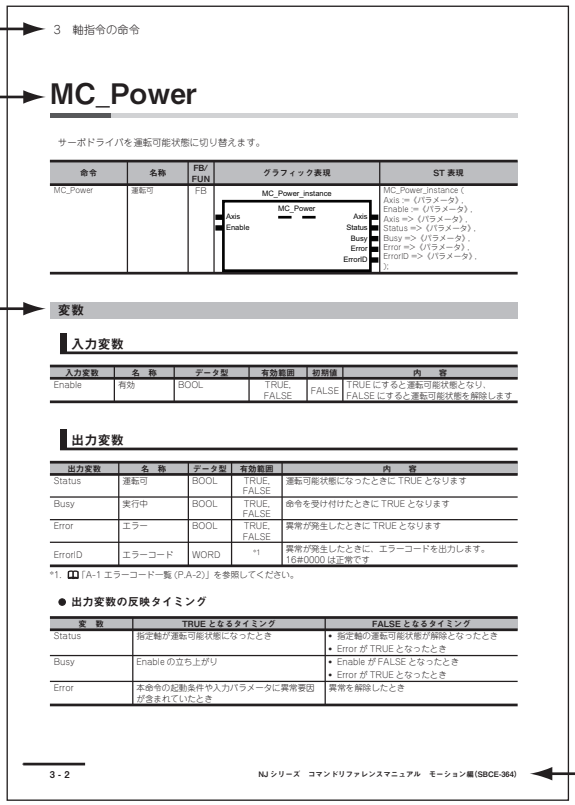
ページ構成

このマニュアルの各ページの構成と記号は以下のようになっています。

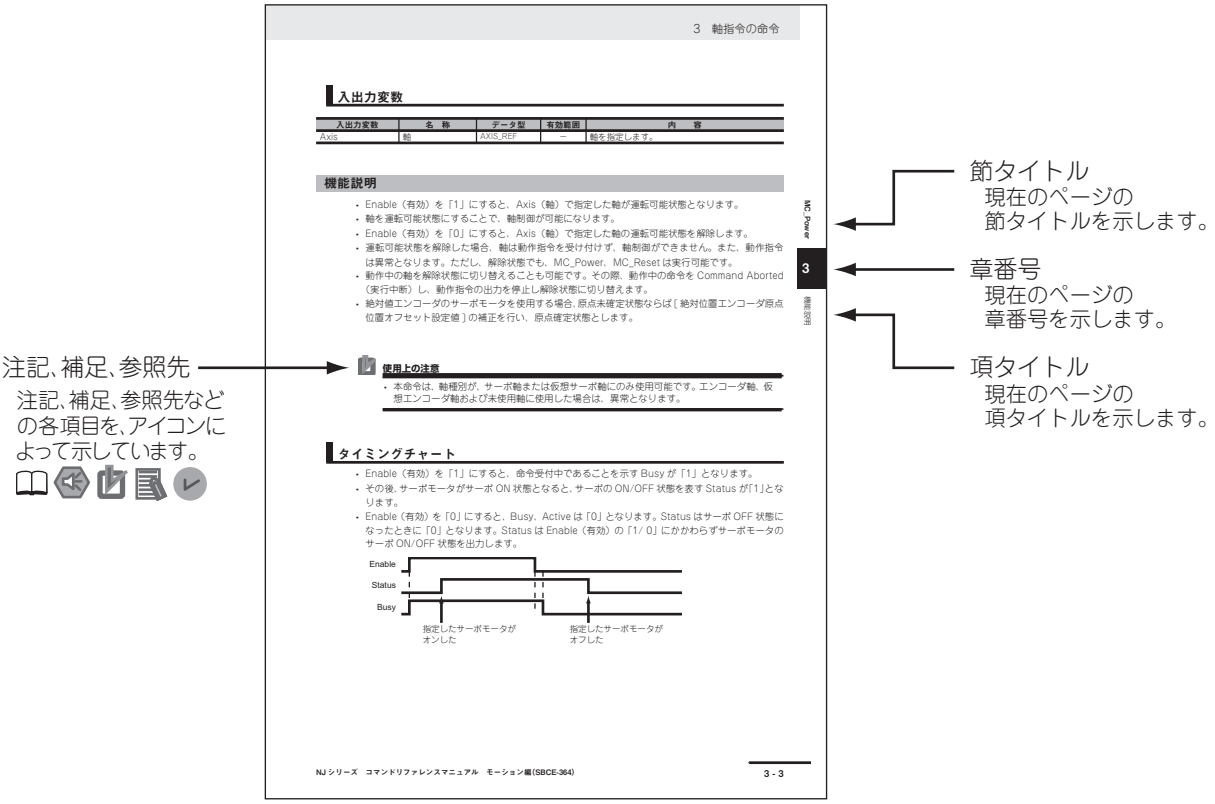
章タイトル

節タイトル

項タイトル



マニュアル名称



節タイトル
現在のページの
節タイトルを示します。

章番号
現在のページの
章番号を示します。

項タイトル
現在のページの
項タイトルを示します。

注記、補足、参照先
注記、補足、参照先などの各項目を、アイコンによって示しています。

（注）各ページの構成は、説明用の見本です。実際の内容とは異なります。

アイコン

このマニュアルで使われているアイコンには、次のような意味があります。

安全上の要点

製品を安全に使用するために実施または回避すべきことを示します。

使用上の注意

製品の動作不能、誤動作、または性能・機能への悪影響を予防するために実施または回避すべきことを示します。

参考

必要に応じて読んでいただきたい項目です。
知っておくと便利な情報や、使用する上で参考となる内容について説明しています。

バージョン関連情報

コントローラや Sysmac Studio の、バージョンによる性能や機能の違いについて説明しています。

表記についての留意事項

本マニュアルでは、Sysmac Studio から実機コントローラにデータを転送することを“ダウンロード”、実機コントローラからデータを Sysmac Studio に転送することを“アップロード”と記述しています。
Sysmac Studio では、“ダウンロード”および“アップロード”は、いずれも“同期”機能を使用します。Sysmac Studio の“同期”機能は、パソコン上のデータと実機コントローラのデータを自動照合し、どちらの方向に転送するかをユーザが選択する機能です。

目次構成

1 モーション制御命令の概要
2 変数と命令
3 軸指令の命令
4 軸グループ指令の命令
5 共通指令の命令
A 付録
I 索引

目次

はじめに **1**
対象となる読者の方々 1
対象となる製品 1

分冊構成 **2**

マニュアルの読み方 **6**
ページ構成 6
アイコン 7
表記についての留意事項 8

目次構成 **9**

ご承諾事項 **16**

安全上の注意 **18**

安全上の要点 **19**

使用上の注意 **20**

法規と規格 **21**

バージョン **22**
CPU ユニットのユニットバージョンと Sysmac Studio のバージョン 22

関連マニュアル **23**

マニュアル改訂履歴 **27**

第 1 章　モーション制御命令の概要

1-1　モーション制御命令 **1-2**
1-1-1　PLCopen®モーション制御用ファンクションブロック 1-2
1-1-2　モーション制御命令の概要 1-2
1-1-3　同期制御の主軸および補助軸の注意事項 1-6

1-2　モーション制御命令の基本知識 **1-8**
1-2-1　モーション制御命令の名称 1-8
1-2-2　モーション制御命令の言語 1-8
1-2-3　モーション制御命令の配置 1-9
1-2-4　モーション制御命令の同時起動 1-16
1-2-5　モーション制御命令のオンラインエディット 1-17
1-2-6　CPU ユニットの動作モード切り替え 1-17

第 2 章　変数と命令

2-1　変数一覧 **2-2**
2-1-1　MC 共通変数 2-3
2-1-2　軸変数 2-3
2-1-3　軸グループ変数 2-8
2-1-4　モーション制御命令の入力変数 2-10

2-1-5　モーション制御命令の出力変数 2-23
2-1-6　モーション制御命令の入出力変数 2-25

2-2　命令一覧 2-28
2-2-1　共通指令 2-28
2-2-2　軸指令 2-29
2-2-3　軸グループ指令 2-31

2-3　PDO マッピング 2-32
2-3-1　必須オブジェクト 2-32
2-3-2　命令別による設定オブジェクト 2-33

第 3 章　軸指令の命令

MC_Power 3-3
変数 3-3
機能説明 3-4

MC_MoveJog 3-8
変数 3-8
機能説明 3-9

MC_Home 3-16
変数 3-16
機能説明 3-17

MC_HomeWithParameter 3-37
変数 3-37
機能説明 3-40

MC_Move 3-42
変数 3-42
機能説明 3-44

MC_MoveAbsolute 3-47
変数 3-47
機能説明 3-49
サンプルプログラム 1 3-55
サンプルプログラム 2 3-63

MC_MoveRelative 3-73
変数 3-73
機能説明 3-75

MC_MoveVelocity 3-81
変数 3-81
機能説明 3-83
サンプルプログラム 3-86

MC_MoveZeroPosition 3-95
変数 3-95
機能説明 3-97

MC_MoveFeed 3-102
変数 3-102
機能説明 3-105
サンプルプログラム 3-117

MC_Stop 3-129
変数 3-129
機能説明 3-130

MC_ImmediateStop 3-138
変数 3-138
機能説明 3-139

MC_SetPosition 3-142
変数 3-142
機能説明 3-143

MC_SetOverride 3-148
変数 3-148
機能説明 3-149

MC_ResetFollowingError 3-153
変数 3-153
機能説明 3-154

MC_CamIn 3-160
変数 3-160
機能説明 3-163
サンプルプログラム 1 3-187
サンプルプログラム 2 3-198

MC_CamOut 3-216
変数 3-216
機能説明 3-217

MC_CamMonitor 3-221
変数 3-221
機能説明 3-224
使用上の注意 3-229

MC_GearIn 3-230
変数 3-230
機能説明 3-232
サンプルプログラム 3-238

MC_GearInPos 3-249
変数 3-249
機能説明 3-251
サンプルプログラム 3-259

MC_GearOut 3-270
変数 3-270
機能説明 3-271

MC_MoveLink 3-275
変数 3-275
機能説明 3-278
サンプルプログラム 3-286

MC_CombineAxes 3-297
変数 3-297
機能説明 3-299

MC_Phasing 3-308
変数 3-308
機能説明 3-310

MC_TorqueControl 3-315
変数 3-315
機能説明 3-317

MC_SetTorqueLimit 3-326
変数 3-326
機能説明 3-327

MC_ZoneSwitch 3-333
変数 3-333
機能説明 3-334

MC_TouchProbe 3-339
変数 3-339
機能説明 3-341
サンプルプログラム 3-353

MC_AbortTrigger 3-360
変数 3-360
機能説明 3-361

MC_AxesObserve 3-364

変数 3-364
機能説明 3-366

MC_SyncMoveVelocity 3-370
変数 3-370
機能説明 3-372

MC_SyncMoveAbsolute 3-379
変数 3-379
機能説明 3-381

MC_Reset 3-386
変数 3-386
機能説明 3-387

MC_ChangeAxisUse 3-389
変数 3-389
機能説明 3-390

MC_DigitalCamSwitch 3-395
変数 3-395
機能説明 3-397
サンプルプログラム 3-406

MC_TimeStampToPos 3-414
変数 3-414
機能説明 3-415
サンプルプログラム 3-418

MC_PeriodicSyncVariables 3-425
変数 3-425
機能説明 3-426
サンプルプログラム 3-429

MC_SyncOffsetPosition 3-433
変数 3-433
機能説明 3-435

MC_OffsetPosition 3-443
変数 3-443
機能説明 3-445

第 4 章　軸グループ指令の命令

MC_GroupEnable 4-2
変数 4-2
機能説明 4-3

MC_GroupDisable 4-6
変数 4-6
機能説明 4-7

MC_MoveLinear 4-11
変数 4-11
機能説明 4-13
サンプルプログラム 4-23

MC_MoveLinearAbsolute 4-38
変数 4-38
機能説明 4-40

MC_MoveLinearRelative 4-41
変数 4-41
機能説明 4-43

MC_MoveCircular2D 4-44
変数 4-44
機能説明 4-46
サンプルプログラム 4-58

MC_GroupStop **4-71**
変数 4-71
機能説明 4-72

MC_GroupImmediateStop **4-79**
変数 4-79
機能説明 4-80

MC_GroupSetOverride **4-83**
変数 4-83
機能説明 4-84

MC_GroupReadPosition **4-87**
変数 4-87
機能説明 4-88

MC_ChangeAxesInGroup **4-91**
変数 4-91
機能説明 4-92

MC_GroupSyncMoveAbsolute **4-95**
変数 4-95
機能説明 4-96

MC_GroupReset **4-101**
変数 4-101
機能説明 4-102

第 5 章　共通指令の命令

MC_SetCamTableProperty **5-2**
変数 5-2
機能説明 5-3

MC_SetCamTableEndPointIndex **5-8**
変数 5-8
機能説明 5-9

MC_SaveCamTable **5-13**
変数 5-13
機能説明 5-14

MC_Write **5-18**
変数 5-18
機能説明 5-21

MC_GenerateCamTable **5-23**
変数 5-23
機能説明 5-25
サンプルプログラム 5-38

MC_WriteAxisParameter **5-50**
変数 5-50
機能説明 5-51

MC_ReadAxisParameter **5-63**
変数 5-63
機能説明 5-64

付録

A-1　多重起動の可否 **A-2**
A-1-1　軸の状態、軸グループの状態 A-2
A-1-2　状態遷移と多重起動の可否 A-4

A-2　バージョン関連情報 **A-10**

A-3　マニュアル No.変更前の改訂履歴 **A-12**

索引

ご承諾事項

「当社商品」について特別の合意がない場合には、お客様のご購入先にかかわらず、本ご承諾事項記載の条件を適用いたします。

定義

本ご承諾事項中の用語の定義は次のとおりです。

- 「当社商品」:「当社」の FA システム機器、汎用制御機器、センシング機器、電子・機構部品
- 「カタログ等」:「当社商品」に関する、ベスト制御機器オムロン、電子・機構部品総合カタログ、その他のカタログ、仕様書、取扱説明書、マニュアル等であって電磁的方法で提供されるものも含みます。
- 「利用条件等」:「カタログ等」に記載の、「当社商品」の利用条件、定格、性能、動作環境、取り扱い方法、利用上の注意、禁止事項その他
- 「お客様用途」:「当社商品」のお客様におけるご利用方法であって、お客様が製造する部品、電子基板、機器、設備またはシステム等への「当社商品」の組み込み、または利用を含みます。
- 「適合性等」:「お客様用途」での「当社商品」の(a)適合性、(b)動作、(c)第三者の知的財産の非侵害、(d)法令の遵守および(e)各種規格の遵守

記載事項のご注意

「カタログ等」の記載内容については次の点をご理解ください。

- 定格値および性能値は、単独試験における各条件のもとで得られた値であり、各定格値および性能値の複合条件のもとで得られる値を保証するものではありません。
- 参考データはご参考として提供するもので、その範囲で常に正常に動作することを保証するものではありません。
- 利用事例はご参考ですので、「当社」は「適合性等」について保証いたしかねます。
- 「当社」は、改善や当社都合等により、「当社商品」の生産を中止し、または「当社商品」の仕様を変更することがあります。

ご利用にあたってのご注意

ご採用およびご利用に際しては次の点をご理解ください。

- 定格・性能ほか「利用条件等」を遵守しご利用ください。
- お客様ご自身にて「適合性等」をご確認いただき、「当社商品」のご利用の可否をご判断ください。「当社」は「適合性等」を一切保証いたしかねます。
- 「当社商品」がお客様のシステム全体の中で意図した用途に対して、適切に配電・設置されていることをお客様ご自身で、必ず事前に確認してください。
- 「当社商品」をご使用の際には、(i)定格および性能に対し余裕のある「当社商品」のご利用、冗長設計などの安全設計、(ii)「当社商品」が故障しても、「お客様用途」の危険を最小にする安全設計、(iii)利用者に危険を知らせるための、安全対策のシステム全体としての構築、(iv)「当社商品」および「お客様用途」の定期的な保守、の各事項を実施してください。
- 「当社」は DDoS 攻撃（分散型 DoS 攻撃）、コンピュータウイルスその他の技術的な有害プログラム、不正アクセスにより、「当社商品」、インストールされたソフトウェア、またはすべてのコンピュータ機器、コンピュータプログラム、ネットワーク、データベースが感染したとしても、そのことにより直接または間接的に生じた損失、損害その他の費用について一切責任を負わないものとします。

お客様ご自身にて、①アンチウイルス保護、②データ入出力、③紛失データの復元、④「当社商品」またはインストールされたソフトウェアに対するコンピュータウイルス感染防止、⑤「当社商品」に対する不正アクセス防止についての十分な措置を講じてください。

- 「当社商品」は、一般工業製品向けの汎用品として設計製造されています。従いまして、次に掲げる用途での使用は意図しておらず、お客様が「当社商品」をこれらの用途に使用される際には、「当社」は「当社商品」に対して一切保証をいたしません。ただし、次に掲げる用途であっても「当社」の意図した特別な商品用途の場合や特別の合意がある場合は除きます。
  a）高い安全性が必要とされる用途（例：原子力制御設備、燃焼設備、航空・宇宙設備、鉄道設備、昇降設備、娯楽設備、医用機器、安全装置、その他生命・身体に危険が及びうる用途）
  b）高い信頼性が必要な用途（例：ガス・水道・電気等の供給システム、24 時間連続運転システム、決済システムほか権利・財産を取扱う用途など）
  c）厳しい条件または環境での用途（例：屋外に設置する設備、化学的汚染を被る設備、電磁的妨害を被る設備、振動・衝撃を受ける設備など）
  d)「カタログ等」に記載のない条件や環境での用途
- 上記の(a)から(d)に記載されている他、「本カタログ等記載の商品」は自動車（二輪車含む。以下同じ）向けではありません。自動車に搭載する用途には利用しないで下さい。自動車搭載用商品については当社営業担当者にご相談ください。

保証条件

「当社商品」の保証条件は次のとおりです。

- 保証期間　ご購入後 1 年間といたします。
  （ただし「カタログ等」に別途記載がある場合を除きます。）
- 保証内容　故障した「当社商品」について、以下のいずれかを「当社」の任意の判断で実施します。
  a）当社保守サービス拠点における故障した「当社商品」の無償修理
  （ただし、電子・機構部品については、修理対応は行いません。）
  b）故障した「当社商品」と同数の代替品の無償提供
- 保証対象外　故障の原因が次のいずれかに該当する場合は、保証いたしません。
  a)「当社商品」本来の使い方以外のご利用
  b)「利用条件等」から外れたご利用
  c）本ご承諾事項「ご利用にあたってのご注意」に反するご利用
  d)「当社」以外による改造、修理による場合
  e)「当社」以外の者によるソフトウェアプログラムによる場合
  f)「当社」からの出荷時の科学・技術の水準では予見できなかった原因
  g）上記のほか「当社」または「当社商品」以外の原因（天災等の不可抗力を含む）

責任の制限

本ご承諾事項に記載の保証が、「当社商品」に関する保証のすべてです。
「当社商品」に関連して生じた損害について、「当社」および「当社商品」の販売店は責任を負いません。

輸出管理

「当社商品」または技術資料を、輸出または非居住者に提供する場合は、安全保障貿易管理に関する日本および関係各国の法令・規制を遵守ください。お客様が法令・規則に違反する場合には、「当社商品」または技術資料をご提供できない場合があります。

安全上の注意

安全上のご注意の内容については、以下のマニュアルを参照してください。
- 『NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-418)』
- 『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編(SBCA-497)』
- 『NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-462)』
- 『NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-448)』
- 『NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-466)』

安全上の要点

安全上の要点の内容については、以下のマニュアルを参照してください。
- 『NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-418)』
- 『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編(SBCA-497)』
- 『NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-462)』
- 『NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-448)』
- 『NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-466)』

使用上の注意

使用上の注意の内容については、以下のマニュアルを参照してください。
- 『NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-418)』
- 『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編(SBCA-497)』
- 『NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-462)』
- 『NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-448)』
- 『NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-466)』

法規と規格

法規と規格の内容については、以下のマニュアルを参照してください。
- 『NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-418)』
- 『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編(SBCA-497)』
- 『NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-462)』
- 『NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-448)』
- 『NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-466)』

バージョン

NJ/NX シリーズの各ユニットおよび各 EtherCAT スレーブは、ハードウェアやソフトウェアが、ハードウェアリビジョンやユニットバージョンという改変番号でバージョン管理されています。ハードウェアまたはソフトウェアに仕様変更があるごとに、ハードウェアリビジョンまたはユニットバージョンが更新されます。したがって、同じ形式のユニットや EtherCAT スレーブでも、ハードウェアリビジョンまたはユニットバージョンが異なると搭載機能や性能が異なります。

バージョンの内容については、以下のマニュアルを参照してください。
- 『NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-418)』
- 『NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編(SBCA-497)』
- 『NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-462)』
- 『NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-448)』
- 『NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編（SBCA-466)』

CPU ユニットのユニットバージョンと Sysmac Studio のバージョン

NJ/NX シリーズ CPU ユニットのユニットバージョンにより、搭載している機能が異なります。バージョンアップで追加された機能を使用するときは、対応しているバージョンの Sysmac Studio を使う必要があります。
CPU ユニットのユニットバージョンの種類と Sysmac Studio のバージョンとの関係、およびユニットバージョンによるサポート機能一覧については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編(SBCA-467)』を参照してください。

関連マニュアル

本マニュアルに関連するマニュアルは、下表のとおりです。あわせてご覧ください。

マニュアル名称	Man.No.	形式	用途	内容
NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	SBCA-418	形 NX701-□□□□	形 NX701 CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。 おもにハードウェアに関する情報。	形 NX701 のシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検
NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編	SBCA-497	形 NX502-□□□□	形 NX502 CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。 おもにハードウェアに関する情報。	形 NX502 のシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検
NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編	SBCA-462	形 NX102-□□□□	形 NX102 CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。 おもにハードウェアに関する情報。	形 NX102 のシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検
NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編	SBCA-448	形 NX1P2-□□□□	形 NX1P2 CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。 おもにハードウェアに関する情報。	形 NX1P2 のシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検
NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	SBCA-466	形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ シリーズ CPU ユニットの概要／設計／取付／保守などの基本的な仕様について知りたいとき。 おもにハードウェアに関する情報。	NJ シリーズのシステム全体概要、および CPU ユニットに関して、以下の内容を説明します。 • 特長やシステム構成 • 概要 • 各部の名称と機能 • 一般仕様 • 設置と配線 • 保守点検
NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	SBCA-467	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズ CPU ユニットのプログラミング／システムの立ち上げについて知りたいとき。 おもにソフトウェアに関する情報。	NJ/NX シリーズ CPU ユニットに関して、以下の内容を説明します。 • CPU ユニットの動作 • CPU ユニットの機能 • 初期設定 • IEC 61131-3 ベースの言語仕様とプログラミング
NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル 内蔵 I/O、オプションボード機能編	SBCA-449	形 NX1P2-□□□□	NX シリーズ 形 NX1P2 CPU ユニットだけに搭載する機能の詳細について、および、NJ/NX シリーズの機能の概要を知りたいとき。	形 NX1P2 CPU ユニットの機能のうち以下の内容を説明します。 • 内蔵 I/O • シリアル通信用オプションボード • アナログ入出力用オプションボード あわせて、NJ/NX シリーズ CPU ユニットの以下機能の概要を説明します。 • モーション制御機能 • EtherNet/IP 通信機能 • EtherCAT 通信機能

マニュアル名称	Man.No.	形式	用途	内容
NJ/NX シリーズ コマンドリファレンス マニュアル 基本編	SBCA-468	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズの基本命令仕様の詳細について知りたいとき。	各命令（IEC 61131-3 仕様）の詳細を説明します。
NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編	SBCE-433	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	モーション制御の設定やプログラミングの考え方について知りたいとき。	モーション制御のための CPU ユニットの設定や動作、プログラミングの考え方について説明します。
NJ/NX シリーズ コマンドリファレンス マニュアル モーション編	SBCE-434	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	モーション命令仕様の詳細について知りたいとき。	各モーション命令の詳細を説明します。
NJ/NX シリーズ CPU ユニット 内蔵 EtherCAT®ポート ユーザーズマニュアル	SBCD-376	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズ CPU ユニットの内蔵 EtherCAT ポートを使用するとき。	内蔵 EtherCAT ポートに関して説明します。 概要、構成、機能、セットアップについて記述しています。
NJ/NX シリーズ CPU ユニット 内蔵 EtherNet/IP™ポート ユーザーズマニュアル	SBCD-377	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズ CPU ユニットの内蔵 EtherNet/IP ポートを使用するとき。	内蔵 EtherNet/IP ポートに関して説明します。 基本設定、タグデータリンク、その他の機能について記述しています。
NJ/NX シリーズ CPU ユニット ユーザーズマニュアル OPC UA 編	SBCD-374	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NJ501-1□00	OPC UA を使用するとき。	OPC UA に関して説明します。
NX シリーズ CPU ユニット ユーザーズマニュアル FINS 機能編	SBCD-375	形 NX701-□□20 形 NX502-□□□□ 形 NX102-□□□□	NX シリーズ CPU ユニットの FINS 機能を使用するとき。	NX シリーズ CPU ユニットの FINS 機能について説明します。
NJ/NX シリーズ データベース接続 CPU ユニット ユーザーズマニュアル	SBCA-411	形 NX701-□□20 形 NX502-□□□□ 形 NX102-□□20 形 NJ501-□□20 形 NJ101-□□20	NJ/NX シリーズでデータベース接続サービス機能を使用するとき。	データベース接続サービス機能について説明します。
NJ シリーズ SECS/GEM 搭載 CPU ユニット ユーザーズマニュアル	SBCA-412	形 NJ501-1340	NJ シリーズで GEM サービス機能を使用するとき。	GEM サービスの機能について説明します。
NJ シリーズ ロボット統合 CPU ユニット ユーザーズマニュアル	SBCE-436	形 NJ501-R□□□	NJ シリーズ ロボット統合 CPU ユニットを使用するとき。	オムロン製ロボットを制御するための CPU ユニットの設定や動作、プログラミングの考え方について説明します。
Sysmac Studio ロボット統合システム構築機能 オペレーションマニュアル ロボット統合 CPU ユニット編	SBCA-464	形 SYSMAC-SE2□□□ 形 SYSMAC-SE200D-64	NJ シリーズ ロボット統合 CPU ユニットを使用してロボット統合システムを構築するとき。	Sysmac Studio で NJ シリーズ ロボット統合 CPU ユニットを使用するための機能の概要、操作方法について説明します。
Sysmac Studio ロボット統合システム構築機能 オペレーションマニュアル IPC Application Controller 編	SBCA-483	形 SYSMAC-SE2□□□ 形 SYSMAC-SE200D-64	IPC Application Controller を使用してロボット統合システムを構築するとき。	Sysmac Studio で IPC Application Controller を使用するための機能の概要、操作方法について説明します。

マニュアル名称	Man.No.	形式	用途	内容
Sysmac Studio 3D シミュレーション機能 オペレーションマニュアル	SBCA-482	形 SYSMAC-SE2□□□ 形 SYSMAC-SA4□□□-64	Sysmac Studio の 3D シミュレーション機能の概要、使用方法について知りたいとき。	Sysmac Studio の 3D シミュレーション機能の概要、実行手順、操作方法について説明します。
NJ シリーズ NJ Robotics CPU ユニット ユーザーズマニュアル	SBCA-421	形 NJ501-4□□□ 形 NJ501-R□□□	NJ シリーズで Robotics 機能を使用するとき。	NJ Robotics 機能について説明します。
NJ/NY シリーズ NC 統合コントローラ ユーザーズマニュアル	SBCE-428	形 NJ501-5300 形 NY532-5400	NJ/NY シリーズで数値制御を行うとき。	数値制御機能について説明します。
NJ/NY シリーズ コマンドリファレンスマニュアル G コード編	SBCE-429	形 NJ501-5300 形 NY532-5400	G コード／M コード命令仕様の詳細について知りたいとき。	G コード／M コード命令の詳細を説明します。
NJ/NX シリーズ トラブルシューティング マニュアル	SBCA-469	形 NX701-□□□□ 形 NX502-□□□□ 形 NX102-□□□□ 形 NX1P2-□□□□ 形 NJ501-□□□□ 形 NJ301-□□□□ 形 NJ101-□□□□	NJ/NX シリーズで検出する異常の詳細について知りたいとき。	NJ/NX シリーズ システムにて検出する異常管理の考え方と各異常項目について説明します。
Sysmac Studio Version 1 オペレーションマニュアル	SBCA-470	形 SYSMAC-SE2□□□	Sysmac Studio の操作方法、機能について知りたいとき。	Sysmac Studio の操作方法について説明します。
CNC Operator オペレーションマニュアル	SBCE-430	形 SYSMAC-RTNC0□□□D	CNC Operator の概要、使用方法について知りたいとき。	CNC Operator の概要、インストール方法、基本操作、接続操作、主な機能の操作方法などについて説明します。
NX シリーズ EtherCAT®カプラユニット ユーザーズマニュアル	SBCD-361	形 NX-ECC□□□	NX シリーズ EtherCAT カプラユニット、および EtherCAT スレーブターミナルの使用方法について知りたいとき。	NX シリーズ EtherCAT カプラユニットと NX ユニットで構成される EtherCAT スレーブターミナルのシステム概要や構成方法、および EtherCAT を介して NX ユニットを設定、制御、モニタするための、EtherCAT カプラユニットのハードウェアや設定方法、機能について説明します。
NX シリーズ データリファレンスマニュアル	SBCA-410	形 NX-□□□□□□	NX シリーズの各ユニットのシステム構成に必要となるデータを一覧で閲覧したいとき。	NX シリーズの各ユニットの「消費電力」、「質量」など、システム構築に必要となるデータを集めて記載しています。
NX シリーズ NX ユニット ユーザーズマニュアル	SBCA-407	形 NX-ID□□□□ 形 NX-IA□□□□ 形 NX-OC□□□□ 形 NX-OD□□□□ 形 NX-MD□□□□	NX ユニットの使用方法について知りたいとき。	NX ユニットのハードウェアや設定方法、機能について説明します。 以下のユニットのマニュアルがあります。 デジタル I/O ユニット、アナログ I/O ユニット、システムユニット、位置インタフェースユニット、コミュニケーションインタフェースユニット、ロードセル入力ユニット、IO-Link マスタユニット
	SBCA-408	形 NX-AD□□□□ 形 NX-DA□□□□		
	SBCA-461	形 NX-HAD□□□		
	SBCA-440	形 NX-TS□□□□ 形 NX-HB□□□□		
	SBCA-409	形 NX-PD1□□□ 形 NX-PF0□□□ 形 NX-PC0□□□ 形 NX-TBX01		
	SBCE-374	形 NX-EC0□□□ 形 NX-ECS□□□ 形 NX-PG0□□□		
	SBCA-422	形 NX-CIF□□□		
	SBCA-439	形 NX-RS□□□□		
	SBCD-370	形 NX-ILM□□□		

マニュアル名称	Man.No.	形式	用途	内容
EtherCAT リモート I/O ターミナル GX シリーズ EtherCAT スレーブ ユーザーズマニュアル	SBCD-350	形 GX-ID□□□□ 形 GX-OD□□□□ 形 GX-OC□□□□ 形 GX-MD□□□□ 形 GX-AD□□□□ 形 GX-DA□□□□ 形 GX-EC□□□□ 形 XWT-ID□□ 形 XWT-OD□□	EtherCAT リモート I/O ターミナルの使用方法について知りたいとき。	EtherCAT リモート I/O ターミナルのハードウェアや設定方法、機能について説明します。
AC サーボモータ／ドライバ 1S シリーズ EtherCAT®通信内蔵タイプ ユーザーズマニュアル	SBCE-377	形 R88M-1□ 形 R88D-1SN□-ECT	EtherCAT 通信内蔵タイプのサーボモータ／ドライバの使用方法について知りたいとき。	EtherCAT 通信内蔵タイプのサーボモータ／ドライバのハードウェアや設定方法、機能について説明します。
	SBCE-438	形 R88M-1AL□/-1AM□ 形 R88D-1SAN□-ECT		
AC サーボモータ／ドライバ G5 シリーズ EtherCAT®通信内蔵タイプ ユーザーズマニュアル	SBCE-365	形 R88M-K□ 形 R88D-KN□-ECT	EtherCAT 通信内蔵タイプのサーボモータ／ドライバの使用方法について知りたいとき。	EtherCAT 通信内蔵タイプのサーボモータ／ドライバのハードウェアや設定方法、機能について説明します。 G5 シリーズには、リニアモータタイプや位置制御限定モデルがあります。
	SBCE-366	形 R88L-EC-□ 形 R88D-KN□-ECT-L		

マニュアル改訂履歴

マニュアル改訂記号は、表表紙・裏表紙に記載されている Man.No.の後尾に付記されます。



改訂記号	改訂年月	改訂理由・改訂ページ
A	2018 年 7 月	• Man.No.の変更 • 誤記訂正
B	2019 年 1 月	誤記訂正による改訂
C	2019 年 4 月	• AC サーボモータ／ドライバ 1S シリーズの機種追加に伴う改訂 • MC_CamMonitor（カムモニタ）命令、および MC_OffsetPosition（位置オフセット補正）命令の追加に伴う改訂
D	2019 年 7 月	Sysmac Studio Ver.1.29 対応による改訂
E	2019 年 10 月	• 形 NX1P2-9B□□□□の追加に伴う改訂 • 誤記訂正
F	2020 年 8 月	形 NJ501-R□□□の追加に伴う改訂
G	2022 年 4 月	誤記修正
H	2023 年 4 月	形 NX502-1□□□の追加に伴う改訂
J	2024 年 4 月	形 NX502-1700、形 NX502-1600 の追加に伴う改訂
K	2024 年 10 月	誤記修正
L	2025 年 7 月	MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令の追加に伴う改訂

1

モーション制御命令の概要

NJ/NX シリーズ CPU ユニットに内蔵されているモーション制御命令の概要について説明します。

1-1　モーション制御命令 1-2
1-1-1　PLCopen®モーション制御用ファンクションブロック 1-2
1-1-2　モーション制御命令の概要 1-2
1-1-3　同期制御の主軸および補助軸の注意事項 1-6
1-2　モーション制御命令の基本知識 1-8
1-2-1　モーション制御命令の名称 1-8
1-2-2　モーション制御命令の言語 1-8
1-2-3　モーション制御命令の配置 1-9
1-2-4　モーション制御命令の同時起動 1-16
1-2-5　モーション制御命令のオンラインエディット 1-17
1-2-6　CPU ユニットの動作モード切り替え 1-17

1-1 モーション制御命令

NJ/NX シリーズ CPU ユニットのユーザプログラムからモーション制御機能を実行するためには、ファンクションブロックとして定義されたモーション制御命令を使用します。

MC 機能モジュールのモーション制御命令は、PLCopen®のモーション制御用ファンクションブロック技術仕様をベースにしています。

モーション制御命令は、PLCopen®で定義された命令と MC 機能モジュール独自の命令の 2 つがあります。

ここでは、PLCopen®モーション制御用ファンクションブロックの概要とモーション制御命令の概要について説明します。

モーション制御命令の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

NX シリーズ 位置インタフェースユニットを使用する場合は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。

バージョン関連情報

NX シリーズ 位置インタフェースユニットを使用する場合は、ユニットバージョン Ver.1.05 以降の CPU ユニットと Sysmac Studio Ver.1.06 以降の組み合わせで使用可能です。

1-1-1 PLCopen®モーション制御用ファンクションブロック

PLCopen®では、モーション制御用ファンクションブロックを標準化することにより、IEC 61131-3(JIS B 3503)規格の言語によるプログラムインタフェースを定義しています。
単軸での位置決めや電子カム機能、および多軸での補間制御などの機能に加え、命令を起動するための基本的な手順についても定義しています。

PLCopen®モーション制御用ファンクションブロックを利用することにより、ハードウェアの依存性を減らし、ユーザプログラムの再利用性を高めることができます。
さらには、トレーニングやサポートにかかるコストを軽減することもできます。

参考

PLCopen®とは

PLCopen®は、欧州に本部を持つ IEC 61131-3 の普及団体であり、ワールド・ワイドな会員組織です。
IEC 61131-3 とは、PLC プログラミングの国際標準規格です。
PLCopen® Japan は、日本市場におけるプロモーションコミッティで日本市場に関心のある会員から構成されています。

- PLCopen® Japan ウェブサイト, http://www.plcopen-japan.jp/
- PLCopen® 欧州本部ウェブサイト, http://www.plcopen.org/

1-1-2 モーション制御命令の概要

ここでは、PLCopen®モーション制御用ファンクションブロック技術仕様で定義されている項目と、MC 機能モジュールでの概要を説明します。

モーション制御命令の種類

つぎの表はモーション制御命令の種別と概要です。

種類	種別	機能項目	概要
共通指令	共通の管理系命令	カムテーブル	MC 機能モジュール共通の状態を制御したり、各種のデータを操作またはモニタしたりする命令です。
		パラメータ	
軸指令	単軸の動作系命令	単軸位置制御	軸を単軸動作させる命令です。
		単軸速度制御	
		単軸トルク制御	
		単軸同期制御	
		単軸手動操作	
	単軸の管理系命令	単軸制御補助	軸の状態を制御・モニタする命令です。
軸グループ指令	多軸の動作系命令	多軸協調制御	軸グループを協調動作させる命令です。
	多軸の管理系命令	多軸協調制御補助	軸グループの状態を制御・モニタする命令です。

状態遷移

軸の状態と軸グループの状態、および命令の起動による状態の遷移が定義されています。

MC 機能モジュールの状態と状態遷移については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション制御命令の起動とステータス

命令を起動する変数や命令の実行状態を示す変数が、命令の共通ルールとして定義されています。
命令を起動するための入力変数には、「Execute」と「Enable」の 2 種類があります。
命令の実行状態を示す出力変数には、「Busy」、「Done」、「CommandAborted」、「Error」などがあります。

MC 機能モジュールの詳細仕様については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

使用上の注意

マニュアルに記載しているタイミングチャートは、Sysmac Studio のデータトレースで表示したタイミングと異なる場合があります。
データトレースの詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編(SBCA-467)』を参照してください。

異常処理

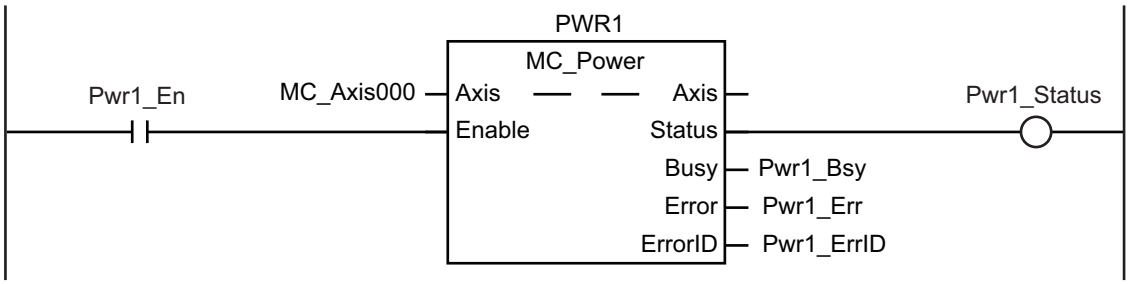
MC 機能モジュールでのモーション制御は、モーション制御命令を使用して実行します。モーション制御命令の実行時に、入力パラメータの誤りや命令処理による異常をチェックしています。
命令に異常があった場合は、命令の出力変数「Error」（エラー）が TRUE になり、「ErrorID」（エラーコード）に異常のコードを出力します。

モーション制御命令では、2 種類の方法で異常処理プログラムを作成することが可能です。

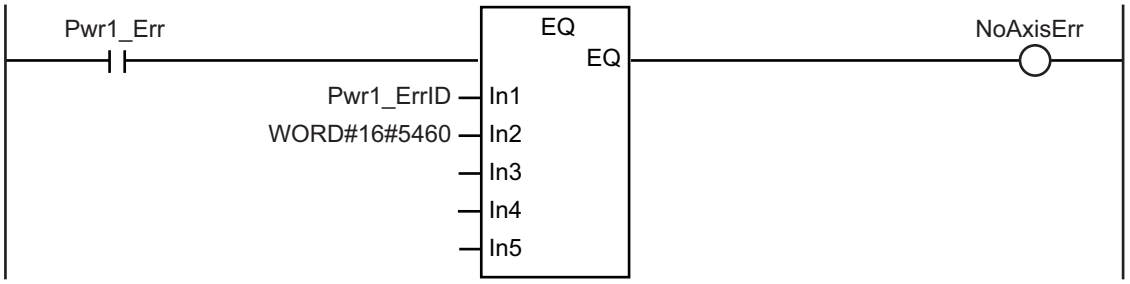
● 命令ごとの異常処理

命令の出力変数「Error」（エラー）または「ErrorID」（エラーコード）を使用することにより、異常が発生した命令ごとに異常処理をおこなうことができます。
下記は、インスタンス名が「PWR1」の命令で「軸指定不正」が発生したことを判別する例です。
「NoAxisErr」の TRUE で異常処理が実行されるようにプログラムしています。

サーボ ON する



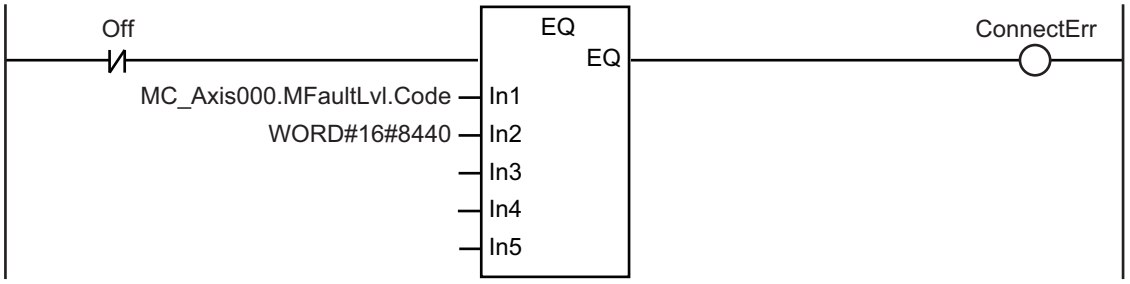
指定した軸が存在しているか確認



● 種別ごとの異常処理

モーション制御システム変数にある異常の状態を使用することにより、異常の種別ごとに異常処理をおこなうことができます。
下記は、「MC_Axis000」の軸で「スレーブ通信異常」が発生したことを判別する例です。
「ConnectErr」の TRUE で、異常処理が実行されるようにプログラムしています。

CPU ユニットとサーボドライバ間の通信異常確認



モーション制御命令実行中の入力変数の変更（命令再起動）

モーション制御命令の実行中に同じ命令インスタンスの入力変数の値を変更して、再度「Execute」を TRUE にすると、変更した値で動作します。

MC 機能モジュールの命令再起動については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

「BufferMode」による命令多重起動

動作中に別の命令インスタンスを実行することができます。
「BufferMode」（バッファモード選択）という入力を設定することにより、動作を開始するタイミングを指定できます。
「BufferMode」（バッファモード選択）には次のモードがあります。

- アボーティング　　：中断(Aborting)
- バッファード　　：待機(Buffered)
- ブレンディングロウ　　：低速度で混合(BlendingLow)
- ブレンディングプレビアス　：前速度で混合(BlendingPrevious)
- ブレンディングネクスト　：次速度で混合(BlendingNext)
- ブレンディングハイ　　：高速度で混合(BlendingHigh)

[アボーティング]（中断）モードでは、ほかの動作を中断してすぐに動作します。
そのほかのバッファモードでは、実行中の命令の出力変数である「Done」や「InVelocity」が TRUE にセットされるまで待機します。
[バッファード]（待機）モードでは、実行中の命令の動作が正常終了(「Done」が TRUE)したのち、次の命令を実行します。
[ブレンディング]（混合）では、2 つの命令の動作を停止することなく連続して実行します。連続するときの速度は 4 種類のバッファモードから選択することができます。

MC 機能モジュールでは「BufferMode」（バッファモード選択）による命令多重起動を、単に命令多重起動と表記して説明しています。

MC 機能モジュールでの命令多重起動については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

MC 機能モジュールでの命令多重起動の可否は、現在の軸の状態または現在の軸グループの状態、および実行する命令の種類によって決まります。それらの関係の詳細は、「A-1 多重起動の可否 (P.A-2)」を参照してください。

モーション制御で使用する構造体

PLCopen®技術資料では、モーション制御に必要な情報を構造体として定義しています。データ型の名称と概要を定義していますが、構造体の内容までは定義していません。
PLCopen®での主なデータ型と MC 機能モジュールで使用しているデータ型は、以下のようになります。

データ型		定義
PLCopen®	MC 機能モジュール	
AXIS_REF	_sAXIS_REF	対応する軸の情報を含んだ構造体です。
AXES_GROUP_REF	_sGROUP_REF	対応する軸グループの情報を含んだ構造体です。
TRIGGER_REF	_sTRIGGER_REF	トリガ入力に関する情報を含んだ構造体です。 • トリガの指定 • 検出パターン情報（正、負、両方、エッジ、レベル、パターン認識など）
INPUT_REF	−	入力仕様に関する情報を含んだ構造体です。仮想データの場合もあります。 MC 機能モジュールでは使用していません。
OUTPUT_REF	_sOUTPUT_REF	物理的な出力に関係する構造体です。 MC 機能モジュールでは、ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで使用します。

MC 機能モジュールでは、上記の表のように、PLCopen®で定義されたデータ型の一部と、MC 機能モジュール独自で定義したデータ型を使用しています。

MC 機能モジュールで扱うデータ型と構造体の定義については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

1-1-3 同期制御の主軸および補助軸の注意事項

同期制御の主軸、および補助軸について、急激な速度変化をする場合の注意事項と、異常となる条件について、以下に記載します。

急激な速度変化をする場合

同期動作開始時、または同期動作中に主軸および補助軸の速度が急激に変化した場合、従軸の動作が急激に変化し、装置に無理な力が加わる可能性があります。
以下の条件の場合に、主軸および補助軸の速度が急激に変化するときがありますので、注意してください。

- 主軸や補助軸に以下の 4 つの命令を実行した場合
  MC_ImmediateStop（即停止）命令
  MC_SetPosition（現在位置変更）命令
  MC_ResetFollowingError（偏差カウンタリセット）命令
  MC_SyncMoveVelocity（サイクリック同期速度制御）命令
  従軸が急激に動作しないように、上記命令の入力パラメータや起動タイミングを適切に設定するか、同期制御を解除した後に、上記命令を起動するかなどの考慮をお願いします。
- 主軸や補助軸で即停止入力信号、限界停止入力信号が TRUE になった場合
- 主軸や補助軸がサーボ ON 状態からサーボ OFF 状態になった場合
  主軸や補助軸が垂直軸などでは、サーボ OFF 状態にした時点で、急激に変化する場合があります。従軸が急激に動作しないように、主軸あるいは補助軸にブレーキをつけるか、同期制御を終了した後に、サーボ OFF 状態にするかなどの対策をお願いします。
- サーボドライバの制御モード切り替え時
  命令起動時の速度変化に注意してください。
  命令の入力パラメータを適切に設定してください。

バージョン関連情報

ユニットバージョン Ver.1.10 以降の CPU ユニットでは、MC_SetPosition（現在位置変更）命令を使用し、同期制御命令の主軸あるいは補助軸の指令現在位置を変更しても、従軸の動作が急激に変化することはありません。

異常となる条件

NJ/NX シリーズ CPU ユニットで共通の条件と、形 NX701 CPU ユニットだけの条件があります。

● NJ/NX シリーズ CPU ユニット共通の条件

同期動作開始時、または同期動作中に主軸および補助軸に以下の 4 つの条件が発生した場合、従軸に「主軸位置読み出し異常」または「補助軸位置読み出し異常」が発生します。
同時に、同期制御命令の出力変数「CommandAborted」（実行中断）が TRUE となります。

- EtherCAT のプロセスデータ通信が確立していない場合
- 「EtherCAT の通信未確立」状態で、「スレーブ通信異常」が発生した場合
- 「絶対値エンコーダ現在位置算出不可」の異常が発生した場合

・ スレーブの離脱をした場合

なお、従軸に対して同期制御命令の多重起動を行った場合は、次のようになります。
・ 多重起動時に主軸や補助軸が上記 4 つの条件のいずれであっても、正常に多重起動を受け付け、バッファ状態となります。
・ 多重起動され、バッファ状態の命令が動作を開始する際に、上記 4 つの条件のいずれかでなければ動作を開始します。



参考

主軸や補助軸に対して MC_Home 命令または MC_HomeWithParameter 命令を起動、または絶対値エンコーダを使用する軸で MC_Power 命令を実行した場合、主軸や補助軸の位置変化を従軸は無視します。
そのため、原点確定時の処理で従軸が急激に動作することはありません。

● 形 NX701 CPU ユニットの条件

主軸および補助軸のタスクと従軸のタスクを異なるタスクに割り付けた場合、従軸に「主軸指定不正（エラーコード: 5462 Hex）」が発生し、同期制御命令の出力変数「Error」（エラー）が TRUE となります。

1-2 モーション制御命令の基本知識

ここでは、NJ/NX シリーズ CPU ユニットに内蔵されている MC 機能モジュールのモーション制御命令を使用してプログラムを作成するための基本仕様と制約事項について説明します。

モーション制御命令の詳細については、「第 3 章 軸指令の命令 (P.3-1)」、「第 4 章 軸グループ指令の命令 (P.4-1)」、「第 5 章 共通指令の命令 (P.5-1)」を参照してください。

1-2-1 モーション制御命令の名称

MC 機能モジュールのすべてのモーション制御命令は、先頭に“MC_”がつきます。

PLCopen®で定義されている命令か、MC 機能モジュール独自の命令かは、「2-2 命令一覧 (P.2-28)」を参照してください。

1-2-2 モーション制御命令の言語

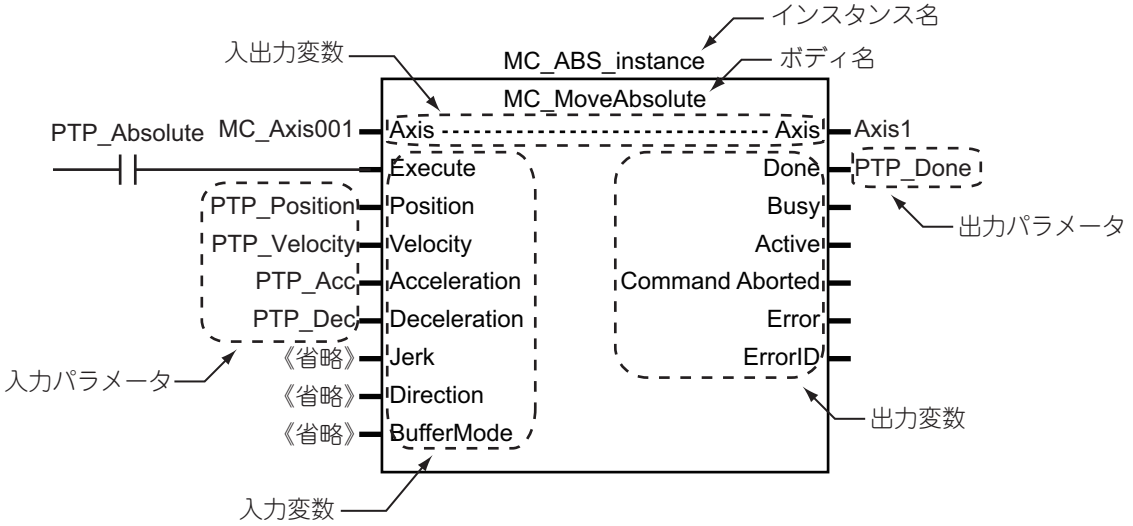
MC 機能モジュールのモーション制御命令は、下記のプログラミング言語で使用することができます。

- ラダー図(LD)
- ストラクチャードテキスト(ST)

ラダー図(LD)

モーション制御命令の命令インスタンスをラダー図に配置します。
命令インスタンスごとにインスタンス名を付けることができます。

下記は MC_MoveAbsolute（絶対値位置決め）命令の記述例です。



- 入出力変数「Axis」には、制御するサーボドライバなどの軸変数名を指定します。
- 各入力変数には、目標位置や目標速度などの動作条件を指定します。
- 各出力変数には、命令の状態やサーボドライバの状態が出力されます。
- 各入力パラメータを省略した場合は、各入力変数の初期値となります。

ストラクチャードテキスト(ST)

命令インスタンス名を指定します。

命令の変数は、左上から左下、右上から右下の順に記述します。
下記は MC_MoveAbsolute（絶対値位置決め）命令の記述例です。

```
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
);
```

1-2-3　モーション制御命令の配置

モーション制御命令をどのタスクに配置できるか、またプログラムのどの位置に配置するとどういう動きとなるかを説明します。

タスク種別

モーション制御命令は、プライマリ定周期タスク、定周期タスク（実行優先度 5）および定周期タスク（実行優先度 16）で使用することができます。それ以外のタスク内でモーション制御命令を使用しようとすると、ビルド実施時に異常となります。

タスク種別	使用可否	備考
プライマリ定周期タスク	可能	• 軸／軸グループを指定しない共通命令 • プライマリ定周期タスクに割り付けられた軸／軸グループに対する命令
定周期タスク（実行優先度 5）	可能 *1	• 軸／軸グループを指定しない共通命令 • 定周期タスク（実行優先度 5）に割り付けられた軸／軸グループに対する命令
定周期タスク（実行優先度 16）	可能 *2	• 軸／軸グループを指定しない共通命令 • プライマリ定周期タスクに割り付けられた軸／軸グループに対する命令
定周期タスク（実行優先度 17）	不可	
定周期タスク（実行優先度 18）	不可	
イベントタスク（実行優先度 8）	不可	
イベントタスク（実行優先度 48）	不可	

*1.　形 NX701 CPU ユニットだけ使用可能です。
*2.　形 NX102 および形 NX1P2 CPU ユニットでは使用できません。

ファンクションブロック定義内

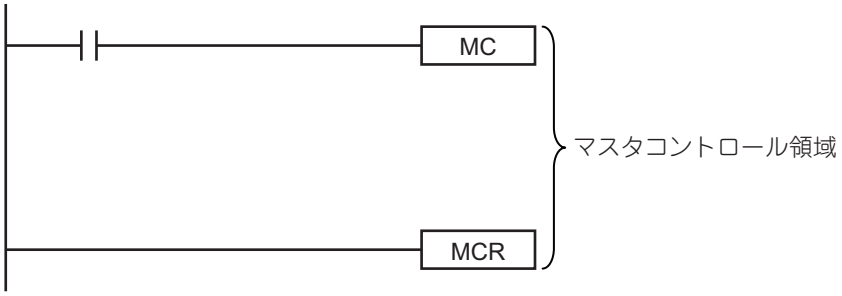
モーション制御命令は、ユーザが作成するファンクションブロック定義内でも使用できます。

参考

複数の動作が組み合わされた工程を、ひとつのファンクションブロックとすることにより、プログラムの構造化による設計効率の向上や、プログラムの視認性が改善されます。

マスタコントロール領域

ラダー図プログラム中で、MC 命令（マスタコントロール開始命令）から MCR 命令（マスタコントロール終了命令）までの領域がマスタコントロール領域です。



モーション制御命令をマスタコントロール領域に配置した場合に、MC の入力条件が FALSE のときは次のような動作となります。

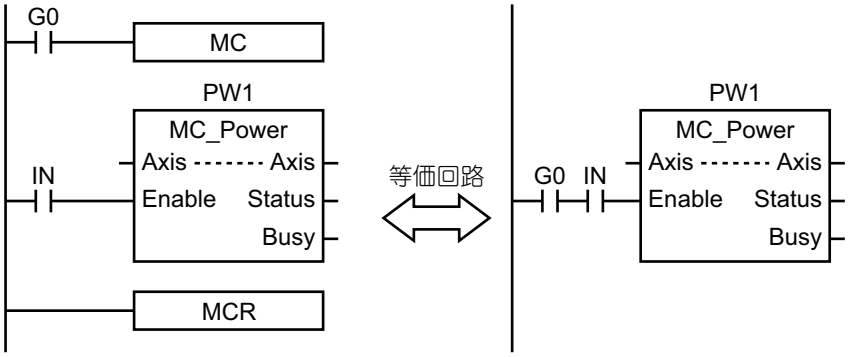
- 左母線から直接、モーション制御命令の入力変数「Enable」、または「Execute」に入力されている命令については、FALSE として動作します。
- モーション制御命令の入力変数「Enable」、または「Execute」が FALSE として動作している場合でも、出力パラメータの値は通常どおり更新しています。

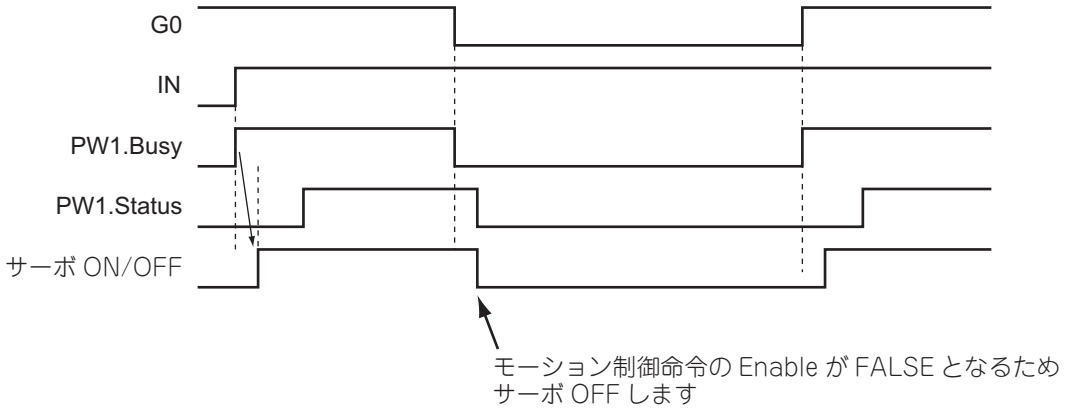
使用上の注意

マスタコントロールは、ST 言語、およびラダー図プログラム中のインライン ST では使用できません。
詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)』の「マスタコントロール」、および『NJ/NX シリーズ コマンドリファレンスマニュアル 基本編(SBCA-468)』の「MC/MCR」命令を参照してください。

● Enable 型のモーション制御命令の場合

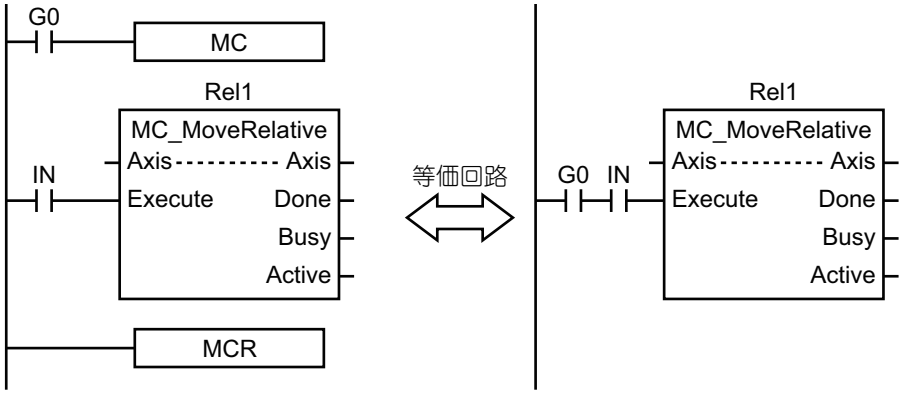
- マスタコントロール領域に配置した命令は、下図の右側の回路と等価となります。
- G0 が TRUE のときは、通常どおり動作します。
- G0 が FALSE のときは、「Enable」入力は FALSE と同じになります。
- G0 が TRUE のときに起動した命令は、動作中に G0 が FALSE になると動作を中断します。

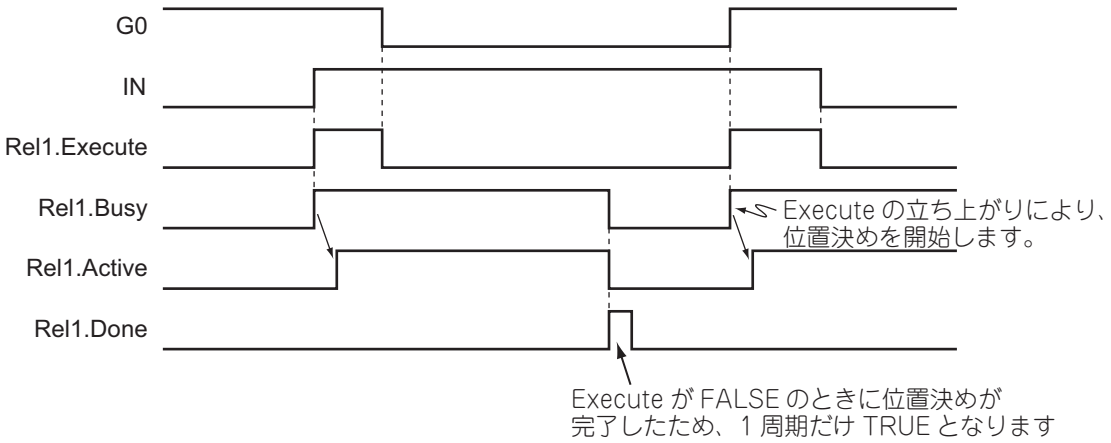




● Execute 型のモーション制御命令の場合

- マスタコントロール領域に配置した命令は、下図の右側の回路と等価となります。
- G0 が TRUE のときは、通常どおり動作します。
- G0 が FALSE のときは、「Execute」入力は FALSE と同じになります。
- G0 が TRUE のときに起動した命令は、動作中に G0 が FALSE になっても完了するまで動作を継続します。このときも、出力パラメータの値は通常どおり更新しています。





使用上の注意

Execute 型のモーション制御命令は立ち上がりで動作を開始するため、マスタコントロール領域内での使用は推奨しません。
使用する場合は、動作に十分注意して使用してください。

ST 構文命令中

IF 命令、CASE 命令、WHILE 命令、REPEAT 命令などの ST 構文命令中に、モーション制御命令を配置した場合について説明します。

各 ST 構文命令の条件式の評価結果に一致しない場合、モーション制御命令は実行されません。また、命令の出力変数の値も更新されません。
Execute 型の命令が実行を開始した後に評価結果が不一致となっても、処理が完了するまでは継続して実行します。その場合、命令の出力変数の値は更新されません。



使用上の注意

条件式の評価結果が不一致となった場合、実行中の Execute 型命令の実行状態が不明となるため、ST 構文命令中での使用は推奨しません。
使用する場合は、動作に十分注意して使用してください。



参考

条件式により実行する Execute 型命令を切り替えるには、「Execute」の入力パラメータだけを ST 構文命令中に配置します。Execute 型命令自体は、ST 構文命令外に配置しておきます。
ST 構文命令の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル 基本編 (SBCA-468)』を参照してください。

実数（REAL 型、LREAL 型）の扱い

実数（REAL 型、LREAL 型）は、浮動小数点型のデータフォーマットを持つデータ型です。
その表現方法や演算処理について説明します。

● REAL 型、LREAL 型のデータサイズ

REAL 型と LREAL 型は、データサイズが異なります。
REAL 型のデータサイズは 32 ビット、LREAL 型は 64 ビットです。

● 浮動小数点形式のデータフォーマット

浮動小数点形式とは、実数を符号、指数、仮数によって表現する形式のことです。
ある実数を、下の式にあてはめて表現した場合の、“s”の値が符号、“e”の値が指数、“f”の値が仮数です。

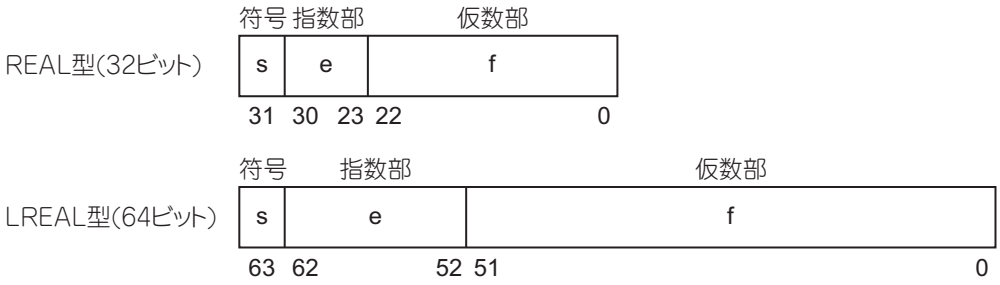
- REAL 型

  数値$=(-1)^{s}2^{e-127}(1+f\times2^{-23})$

- LREAL 型

  数値$=(-1)^{s}2^{e-1023}(1+f\times2^{-52})$

浮動小数点形式のデータフォーマットは、IEEE754 規格に準拠しています。それぞれを、以下に示します。



（例） REAL 型で、「-86.625」を表現する場合

1 **符号の設定**

負数なので、s = 1 になります。

2 **2 進数での表現**

「86.625」を 2 進数表現すると、「1010110.101」になります。

3 **2 進数の正規化**

上の値を正規化すると、「1.010110101×2^6」になります。

4 **指数部の表現**

前式より、e-127 = 6 となりますので、e = 133 になります。
「133」は 2 進数では「10000101」となり、これが指数部の表現となります。

5 **仮数部の表現**

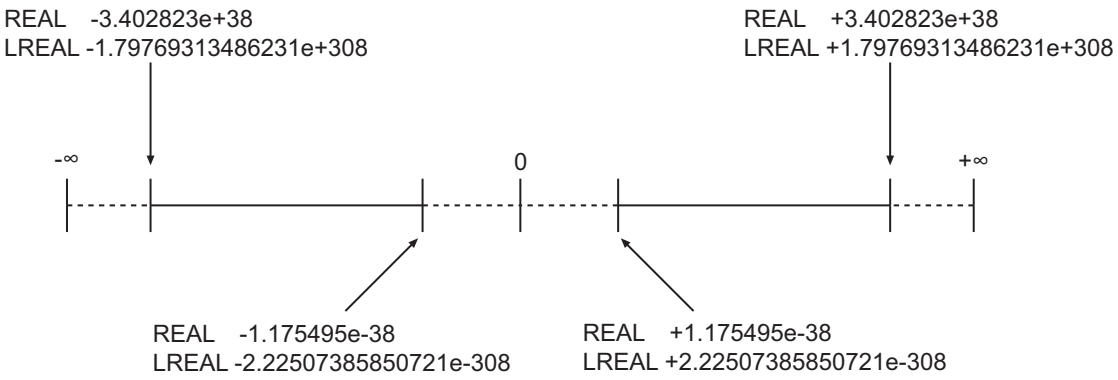
「1.010110101」の小数点以下は「010110101」です。
この小数点以下の数値表現を 23 ビットにするため、けた数が不足している分、0 を付加する必要があります。23 ビットになったものが、f になります。
したがって、f = 01011010100000000000000 です。
すなわち、「-86.625」は、下図のように表現できます。



● 有効範囲

REAL 型、LREAL 型の有効範囲は、下表のとおりです。

データ型	-∞	負数	0	正数	+∞
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	+∞



● 特殊値

「+∞」、「-∞」、「+0」、「-0」、非数を、特殊値と呼びます。
非数とは、浮動小数点数とは見なせないデータのことで、数値として扱われません。
「+0」と「-0」は、どちらも数学的には「0」と同じ意味ですが、演算処理内容に違いがあります。
詳細は、後述します。
特殊値の符号“s”、指数“e”、仮数“f”の値は、下表のとおりです。

データ型	特殊値	符号 s	指数 e	仮数 f
REAL	+∞	0	255	0
	-∞	1	255	0
	+0	0	0	0
	-0	1	0	0
	非数	-	255	0 以外
LREAL	+∞	0	2047	0
	-∞	1	2047	0
	+0	0	0	0
	-0	1	0	0
	非数	-	2047	0 以外

● 非正規化数

浮動小数点形式は、「0」の近傍（絶対値が非常に小さい値）を表現することができません。
そのため、「0」の近傍の有効範囲を拡大するために設けられたのが、非正規化数です。
非正規化数は、通常のデータフォーマットで表現される値よりも、絶対値の小さい値を表現できます。

参考

通常のデータフォーマットで表現される値のことを正規化数と呼びます。

指数 e = 0、かつ仮数 f≠0 のものが、非正規化数と見なされ、値は以下のように表現されます。

- REAL 型
  数値=$(-1)^{s}2^{-126}(f\times2^{-23})$
- LREAL 型
  数値=$(-1)^{s}2^{-1022}(f\times2^{-52})$

（例） REAL 型で、「0.75×2^{-127}」を表現する場合

1 **符号の設定**
正数なので、s = 0 になります。

2 **2 進数での表現**

「0.75」を 2 進数表現すると、「0.11」になります。

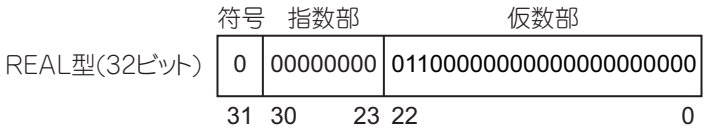
3 **仮数部の計算**

$(0.11)_2 \times 2^{-127} = 2^{-126}(f \times 2^{-23})$より、$f = (0.11)_2 \times 2^{22}$ になります。

4 **仮数部の表現**

前式より、f = 01100000000000000000000 になります。

すなわち、「0.75×2^{-127}」は、下図のように表現できます。



非正規化数は、正規化数よりも有効けた数が少ないです。したがって、正規化数の演算結果が非正規化数になる場合、または演算の途中結果が非正規化数になる場合、演算結果の有効けた数が、正規化数の有効けた数未満になることがあります。

● 演算処理

浮動小数点形式は、近似値表現であるため、真値との間には誤差が存在します。また、値の有効範囲にも制限があります。そのため、演算の際には、つぎのような処理を行います。

丸め

真値が、仮数の有効けた数を超えている場合、以下の規則にしたがって、値の丸めを行います。

- 浮動小数点形式で表現できる値の中で、最も真値に近い値を演算結果とします。
- 浮動小数点形式で表現できる値の中で、最も真値に近い値が 2 つある場合、仮数の最下位けたのビット値が「0」である方を、演算結果とします。
  最も真値に近い値が 2 つある場合とは、真値が 2 つの近似値の中央値の場合があたります。

オーバーフロー、アンダーフロー

真値の絶対値が、浮動小数点形式で表現できる値より大きくなることを、オーバーフローといいます。逆に、浮動小数点形式で表現できる値より小さくなることを、アンダーフローといいます。

- オーバーフローの場合、真値の符号が正なら、演算結果を「+∞」にします。
  真値の符号が負なら、演算結果を「-∞」にします。
- アンダーフローの場合、真値の符号が正なら、演算結果を「+0」にします。
  真値の符号が負なら、演算結果を「-0」にします。

特殊値の演算

特殊値の演算は、つぎの規則に従います。

特殊値については、「特殊値 (P.1-14)」を参照してください。

- 「+∞」と「-∞」の加算結果は、非数になります。
- 符号が同じ無限大どうしの減算結果は、非数になります。
- 「+0」と無限大、「-0」と無限大の乗算結果は、非数になります。
- 「+0」どうしの除算結果、「-0」どうしの除算結果、無限大どうしの除算結果は、非数になります。
- 「+0」と「-0」の加算結果は、「+0」になります。
- 「+0」どうしの減算結果、「-0」どうしの減算結果は、「+0」になります。
- 非数を含む四則演算の結果は、非数になります。
- CMP 命令などの比較命令では、「+0」と「-0」は等しいと判断します。
- 比較命令では、比較対象に非数を含む場合、常に、“等しくない”と判断します。

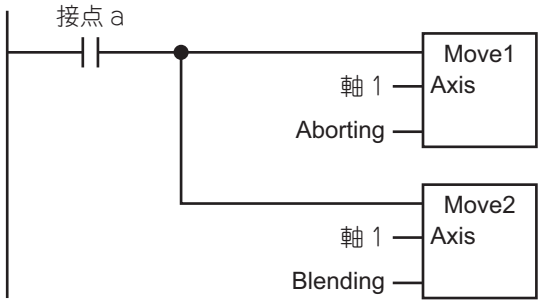
使用上の注意

MC 機能モジュールでは、電子ギアの設定値やモーション制御命令の目標位置などで、浮動小数点(LREAL)の変数を使用しています。そのため、演算結果に丸め誤差が含まれます。
たとえば、MC_MoveRelative（相対値位置決め）命令を繰り返し実行するような場合、誤差が蓄積します。
誤差の蓄積が問題となる場合は、指令単位を［pulse］とするか、MC_MoveAbsolute（絶対値位置決め）命令のような絶対位置を指定してください。

1-2-4 モーション制御命令の同時起動

同じ軸に対して複数のモーション制御命令を、同じタスク周期内で起動する場合について説明します。

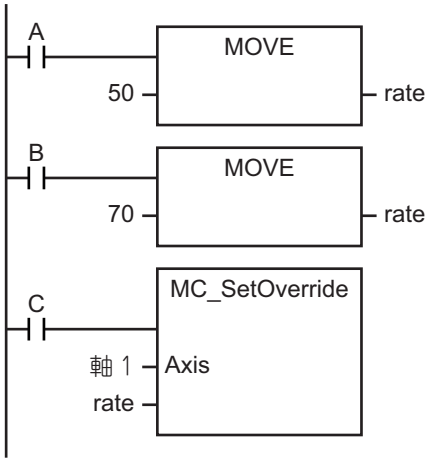
- 下図のプログラムの場合、接点 a が「ON」すると命令インスタンス[Move 1]と[Move 2]が同じタスク周期内で起動されることになります。
- プログラムに記述された命令は、実際には上から順に起動されますので、[Move1]が先に起動され、[Move1]の動作が完了する前に[Move2]が起動されます。
- これはモーション制御命令の多重起動に相当します。このプログラムでは、[Move 1]に対して[Move 2]が［Blending］で多重起動されます。



各モーション制御命令の多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

参考

MC_SetOverride（オーバーライド値設定）命令を上記のように同時起動すると、下側のオーバーライド値が有効になります。MC_SetOverride（オーバーライド値設定）命令で複数のオーバーライド値を使い分ける場合は、下記のようにプログラムすることを推奨します。



1-2-5 モーション制御命令のオンラインエディット

Sysmac Studio のオンラインエディット機能は、モーション制御命令に対して次の操作が可能です。

オンラインエディット操作
モーション制御命令の削除
モーション制御命令の追加
モーション制御命令の入力、出力、入出力変数の追加
モーション制御命令の入力、出力、入出力変数の変更
モーション制御命令の入力、出力、入出力変数の削除

使用上の注意

軸の動作中に MC_Stop（強制停止）命令や MC_GroupStop（軸グループ強制停止）命令など、軸の動作を停止する命令を削除すると、ユーザプログラムの内容によっては、軸が停止しなくなる可能性があります。
モーション制御命令にオンラインエディットをおこなう場合は、安全を確認した上で使用してください。

1-2-6 CPU ユニットの動作モード切り替え

NJ/NX シリーズ CPU ユニットの動作モードには、プログラムモードと運転モードの 2 種類があります。
動作モードを切り替えた場合の MC 機能モジュールの動作について説明します。

運転モードからプログラムモードへの切り替え

- 実行中のモーション制御命令を中断します。出力変数「CommandAborted」（実行中断）は FALSE のままですが、軸の動作としては「CommandAborted」（実行中断）が TRUE のときと同じ動作となります。
- 軸が動作中の場合、最大減速度で減速停止します。サーボ ON/OFF 状態は継続します。
- カムテーブル保存命令でカムテーブル保存を実行中の場合、保存処理は継続します。
- カムテーブル生成命令でカムテーブル生成を実行中の場合、生成処理は継続します。
- 実行優先度が 16 の定周期タスクに配置したモーション制御命令は、このタスクのエンド命令を実行した後に上記の処理を行います。

プログラムモードから運転モードへの切り替え

- モーション制御命令の出力変数はクリアされます。
- 運転モードからプログラムモードに変更すると、軸が減速停止します。この減速停止中に運転モードに戻すと、モーション制御命令の出力変数がクリアされるため、実行していたモーション制御命令の「CommandAborted」（実行中断）は FALSE のままとなります。

参考

- 動作モードの切り替え後もモーション制御命令の出力変数を参照する場合は、出力パラメータの属性が保持属性を持つ変数を割り当ててください。
  割り当てた出力パラメータを参照することにより、動作モードを切り替える直前の出力変数を参照することが可能となります。
- 動作モードを切り替えても、サーボ ON/OFF の状態は保持されます。

2

変数と命令

MC 機能モジュールで扱う変数と命令の一覧表を掲載しています。

2-1　変数一覧 2-2
2-1-1　MC 共通変数 2-3
2-1-2　軸変数 2-3
2-1-3　軸グループ変数 2-8
2-1-4　モーション制御命令の入力変数 2-10
2-1-5　モーション制御命令の出力変数 2-23
2-1-6　モーション制御命令の入出力変数 2-25
2-2　命令一覧 2-28
2-2-1　共通指令 2-28
2-2-2　軸指令 2-29
2-2-3　軸グループ指令 2-31
2-3　PDO マッピング 2-32
2-3-1　必須オブジェクト 2-32
2-3-2　命令別による設定オブジェクト 2-33

2-1 変数一覧

MC 機能モジュールで扱う変数には 2 つの分類があります。
1 つは、軸などの状態や一部のパラメータ設定内容をモニタするためのシステム定義変数です。MC 機能モジュールで扱うシステム定義変数をモーション制御システム変数とよびます。
もう 1 つは、モーション制御命令が指令の引数を入力として受け取ったり、命令の実行状態を出力としたりするための変数です。モーション制御命令の入力変数のいくつかは、選択肢から選択する列挙型の変数があります。
この節では各種の変数の一覧、モーション制御命令の入力変数の有効範囲、および列挙型の一覧を記載しています。

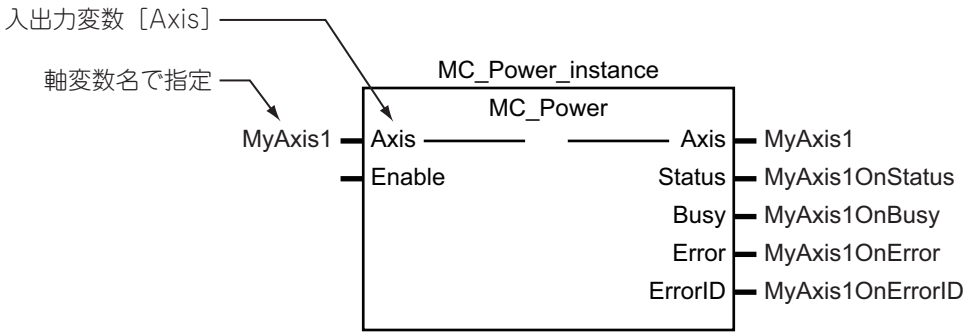
● モーション制御システム変数

第 1 階層	第 2 階層	第 3 階層	内容
システム定義変数	モーション制御システム変数	MC 共通変数	MC 機能モジュール共通の状態をモニタできます。
		軸変数	各軸の状態と一部の軸パラメータの設定内容をモニタできます。
		軸グループ変数	各軸グループの状態と一部の軸グループパラメータの設定内容をモニタできます。

モーション制御システム変数の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

参考

軸変数と軸グループ変数はシステム定義変数です。
これらをユーザプログラムで使用するときは、システム定義変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*], _MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用します。
ユーザプログラムで使用するときにはこのほかに Sysmac Studio で作成した変数名も使用することができます。
Sysmac Studio で作成した軸や軸グループの変数名は、それぞれ任意の名前に変更することができます。
下図はシステム定義変数の軸変数名が「_MC_AX[0]」の軸について、Sysmac Studio で追加したときの軸変数名を「MyAxis1」に変更した場合の例です。



● モーション制御命令の変数

種類	概要
入力変数	命令の引数
出力変数	命令の実行状態のモニタ
入出力変数	命令の実行対象の指定

参考

- 先頭が“_e”のデータ型は、列挙型であることを示します。
- 先頭が“_s”のデータ型は、構造体型であることを示します。

MC 機能モジュールで扱うデータ型については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

2-1-1 MC 共通変数

MC 共通変数は、変数名が_MC_COM で、データ型が_sCOMMON_REF 型の構造体型の変数です。ここでは MC 共通変数の構成とそのメンバの詳細を説明しています。

変数名	データ型	名称	機能
_MC_COM	_sCOMMON_REF	MC 共通変数	
Status	_sCOMMON_REF_STA	MC 共通ステータス	
RunMode	BOOL	MC 運転中	MC 機能モジュールが運転中のときに TRUE となります。
TestMode	BOOL	MC 試運転中	Sysmac Studio による試運転中のときに TRUE となります。
CamTableBusy	BOOL	カムテーブルファイル保存実行中／待機中	カムテーブルファイル保存実行中、または待機中のときに TRUE となります。
GenerateCamBusy *1	BOOL	カムテーブル生成実行中	カムテーブル生成実行中のときに TRUE となります。
PFaultLvl	_sMC_REF_EVENT	MC 共通 部分停止フォールト	
Active	BOOL	MC 共通 部分停止フォールト発生中	MC 共通 停止フォールトが発生したときに TRUE となります。
Code	WORD	MC 共通 部分停止フォールトコード	MC 共通 停止フォールトのコードを出力します。 イベントコードの上 4 けたと同じ値です。
MFaultLvl	_sMC_REF_EVENT	MC 共通 軽度フォールト	
Active	BOOL	MC 共通 軽度フォールト発生中	MC 共通 軽度フォールトが発生したときに TRUE となります。
Code	WORD	MC 共通 軽度フォールトコード	MC 共通 軽度フォールトのコードを出力します。 イベントコードの上 4 けたと同じ値です。
Obsr	_sMC_REF_EVENT	MC 共通 監視情報	
Active	BOOL	MC 共通 監視情報発生中	MC 共通 監視情報が発生したときに TRUE となります。
Code	WORD	MC 共通 監視情報コード	MC 共通 監視情報のコードを出力します。 イベントコードの上 4 けたと同じ値です。

*1. ユニットバージョン Ver.1.08 以降の CPU ユニットと Sysmac Studio Ver.1.09 以降の組み合わせで使用可能です。

2-1-2 軸変数

軸変数は、システム定義変数の変数名が_MC_AX[0-255]、または_MC1_AX[0-255]、_MC2_AX[0-255]で、データ型が_sAXIS_REF 型の構造体型の変数です。
ここでは、_MC_AX[0-255]を例に軸変数の構成とそのメンバの詳細を説明しています。
_MC1_AX[0-255]、_MC2_AX[0-255]についても同様です。

変数名	データ型	名称	機能
_MC_AX[0-255]	_sAXIS_REF	軸変数	
Status	_sAXIS_REF_STA	軸ステータス	
Ready	BOOL	軸起動準備完了	軸の起動準備が整い停止しているときに TRUE となります。 _MC_AX[*].Status.Standstill が「TRUE:停止中」と同じ状態を示します。
Disabled	BOOL	軸無効	軸がサーボ OFF で停止している状態で TRUE となります。 以下の軸ステータスは排他的で、どれか 1 つだけが TRUE になります。 Disabled/ Standstill/ Discrete/ Continuous/ Synchronized/ Homing/ Stopping/ ErrorStop/ Coordinated
Standstill	BOOL	停止中	軸がサーボ ON で停止している状態で TRUE となります。
Discrete	BOOL	位置決め動作中	目標位置に向かって、位置決め制御を実行しているときに TRUE となります。 位置決め動作中にオーバーライド値を「0」に設定したことによる速度「0」の状態も含まれます。
Continuous	BOOL	連続動作中	目標位置を持たずに、連続動作しているときに TRUE となります。 速度制御とトルク制御ときにこの状態になります。 目標速度を「0」にしたことによる速度「0」の状態や、連続動作中にオーバーライド値を「0」に設定したことによる速度「0」の状態も含みます。
Synchronized	BOOL	同期動作中	同期制御を実行しているときに TRUE となります。 同期制御命令切り替え後の同期待ち状態も含みます。
Homing	BOOL	原点復帰中	MC_Home 命令または MC_HomeWithParameter 命令により、原点復帰を実行しているときに TRUE となります。
Stopping	BOOL	減速停止中	MC_Stop 命令、MC_TouchProbe 命令により、軸が停止するまでの状態のとき TRUE となります。 MC_Stop 命令で軸が停止した後に、Execute が TRUE のときも含みます。 減速停止中状態では軸動作命令を起動できません。 (CommandAborted = TRUE)
ErrorStop	BOOL	エラー減速停止中	MC_ImmediateStop 命令または軸 軽度フォールト発生中（_MC_AX[*].MFaultLvl.Active が「TRUE:軸 軽度フォールト発生中」）で、軸が停止するまで、および停止している状態。 この状態では軸動作命令を起動できません。 (CommandAborted = TRUE)
Coordinated	BOOL	多軸協調動作中	多軸協調命令により、軸グループ有効となっている状態で TRUE となります。
Details	_sAXIS_REF_DET	軸制御ステータス *1	
Idle	BOOL	停止中	インポジション待ちを除き、指令値を演算していないときに TRUE となります。*2 「Idle」と「InPosWaiting」は排他的で、同時には TRUE になりません。

変数名		データ型	名称	機能
	InPosWaiting	BOOL	インポジション待ち	インポジション待ちのときに TRUE となります。 インポジションチェックを行う位置決めで、インポジションチェックを行っている状態です。
	Homed	BOOL	原点確定	原点確定状態時に TRUE となります。*3 TRUE: 原点確定 FALSE: 原点未確定
	InHome	BOOL	原点停止	原点位置の範囲内で TRUE となります。 以下の AND 条件です。 • 原点確定状態 • フィードバック現在位置が原点を中心に「原点位置範囲」の幅に入ったとき 指令状態で軸動作中に、原点位置を通過する場合も TRUE となります。
	VelLimit *4	BOOL	指令速度飽和	同期制御中に指令速度が最高速度で制限されているときに TRUE となります。
Dir		_sAXIS_REF_DIR	指令方向ステータス *5	
	Posi	BOOL	正方向指令中	正方向に指令しているときに TRUE となります。
	Nega	BOOL	負方向指令中	負方向に指令しているときに TRUE となります。
DrvStatus		_sAXIS_REF_STA_DRV	サーボドライバ状態 *6	
	ServoOn	BOOL	サーボ ON	サーボモータに通電中 TRUE となります。
	Ready	BOOL	サーボレディ	サーボレディ状態で TRUE となります。
	MainPower	BOOL	主回路電源	サーボドライバの主回路電源が通電中 TRUE となります。
	P_OT	BOOL	正方向限界入力	正方向限界入力が有効なときに TRUE となります。
	N_OT	BOOL	負方向限界入力	負方向限界入力が有効なときに TRUE となります。
	HomeSw	BOOL	原点近傍入力	原点近傍入力が有効なときに TRUE となります。
	Home	BOOL	原点入力	原点入力が有効なときに TRUE となります。*7
	ImdStop	BOOL	即停止入力	即停止入力が有効なときに TRUE となります。
	Latch1	BOOL	外部ラッチ入力 1	ラッチ入力 1 が有効なときに TRUE となります。
	Latch2	BOOL	外部ラッチ入力 2	ラッチ入力 2 が有効なときに TRUE となります。
	DrvAlarm	BOOL	ドライバエラー入力	ドライバエラーが発生しているときに TRUE となります。
	DrvWarning	BOOL	ドライバ ワーニング入力	ドライバワーニングが発生しているときに TRUE となります。
	ILA	BOOL	ドライバ内部機能制限中	サーボドライバ側の制限機能により実際に制限されているときに TRUE となります。*8
	CSP	BOOL	サイクリック同期位置(CSP)モード中	サーボドライバ側がサーボ ON で、CSP モードのときに TRUE となります。*9
	CSV	BOOL	サイクリック同期速度(CSV)モード中	サーボドライバ側がサーボ ON で、CSV モードのときに TRUE となります。*9
	CST	BOOL	サイクリック同期トルク(CST)モード中	サーボドライバ側がサーボ ON で、CST モードのときに TRUE となります。*9
Cmd		_sAXIS_REF_CMD_DATA	軸指令値	

変数名	データ型	名称	機能
Pos	LREAL	指令現在位置	指令位置の現在値を出力します。(単位: 指令単位) サーボ OFF 時および位置制御モード以外のときは、フィードバック現在位置を出力します。*10
Vel	LREAL	指令現在速度	指令速度の現在値を出力します。(単位: 指令単位/s) 正方向移動時はプラス符号、負方向移動時はマイナス符号がつきます。 指令現在位置の差分から算出した速度です。サーボ OFF 時および位置制御モード以外のときは、フィードバック現在位置をもとに算出した速度です。
AccDec	LREAL	指令現在加減速度	指令加減速度の現在値を出力します。(単位: 指令単位/s^2) 指令現在速度の差分から算出した加減速度です。加速時はプラス符号、減速時はマイナス符号がつきます。 実行中の命令の指令加減速度が「0」の場合は、「0」となります。
Jerk	LREAL	指令現在ジャーク	指令ジャークの現在値を出力します。(単位: 指令単位/s^3) 加減速度の絶対値の増加時はプラス符号、減少時はマイナス符号がつきます。 実行中の命令の指令加減速度、指令ジャークが「0」の場合は、「0」となります。
Trq	LREAL	指令現在トルク	指令トルクの現在値を出力します。(単位: %) 正方向移動時はプラス符号、負方向移動時はマイナス符号がつきます。 トルク制御モード以外では、フィードバック現在トルクと同じ値を出力します。
Act	_sAXIS_REF_ACT_DATA	軸現在値	
Pos	LREAL	フィードバック現在位置	フィードバック位置の現在値を出力します。(単位: 指令単位) *10
Vel	LREAL	フィードバック現在速度	フィードバック速度の現在値を出力します。(単位: 指令単位/s) 正方向移動時はプラス符号、負方向移動時はマイナス符号がつきます。
Trq	LREAL	フィードバック現在トルク	フィードバックトルクの現在値を出力します。(単位: %) 正方向移動時はプラス符号、負方向移動時はマイナス符号がつきます。
TimeStamp *11	ULINT	タイムスタンプ	軸の現在位置を更新した時刻を出力します。変化時刻取得機能が動作している軸で有効です。(単位: ns)
MFaultLvl	_sMC_REF_EVENT	軸 軽度フォールト	
Active	BOOL	軸 軽度フォールト発生中	軸 軽度フォールトが発生しているときに TRUE となります。
Code	WORD	軸 軽度フォールトコード	軸 軽度フォールトのコードを出力します。 イベントコードの上 4 けたと同じ値です。
Obsr	_sMC_REF_EVENT	軸 監視情報	
Active	BOOL	軸 監視情報発生中	軸 監視情報が発生しているときに TRUE となります。
Code	WORD	軸 監視情報コード	軸 監視情報のコードを出力します。 イベントコードの上 4 けたと同じ値です。

変数名		データ型	名称	機能
Cfg		_sAXIS_REF_CFG	軸基本設定 *12	
	AxNo	UINT	軸番号	軸の論理番号を表します。 _sAXIS_REF の参照時に軸番号を認識するために参照します。
	AxEnable	_eMC_AXIS_USE	軸使用	軸を使用するかしないかを表します。 0: _mcNoneAxis （未作成軸） 1: _mcUnusedAxis （未使用軸） 2: _mcUsedAxis（使用軸）
	AxType	_eMC_AXIS_TYPE	軸種別	軸種別を表します。 仮想軸では入出力配線が不要となります。 0: _mcServo（サーボ軸） 1: _mcEncdr（エンコーダ軸） 2: _mcVirServo（仮想サーボ軸） 3: _mcVirEncdr（仮想エンコーダ軸）
	NodeAddress	UINT	ノードアドレス	EtherCAT のスレーブアドレスを表します。*13 16#FFFF は、なしを示します。
	ExecID *14	UINT	実行 ID	タスク別の実行 ID を表します。 0: タスク割付なし（未作成軸） 1: プライマリ定周期タスクに割り付け 2: 定周期タスク（実行優先度 5）に割り付け
Scale		_sAXIS_REF_SCALE	単位変換 *15	
	Num	UDINT	モータ 1 回転のパルス数	指令位置のモータ 1 回転のパルス数を表します。 電子ギア比に従い指令値からパルス量への変換を行います。
	Den	LREAL	モータ 1 回転の移動量	指令位置のモータ 1 回転のワークの移動量を表します。*16
	Units	_eMC_UNITS	表示単位	指令位置の表示単位を表します。 0: _mcPls(pulse) 1: _mcMm(mm) 2: _mcUm(μm) 3: _mcNm(nm) 4: _mcDeg(degree) 5: _mcInch(inch)
	CountMode *14	_eMC_COUNT_MODE	カウントモード	カウントモードを表します。 0: _mcCountModeLinear（リニアモード） 1: _mcCountModeRotary（ロータリモード）
	MaxPos *14	LREAL	現在位置上限値	現在位置表示の上限値を表します。*17
	MinPos *14	LREAL	現在位置下限値	現在位置表示の下限値を表します。*18

*1. 指令の制御状態を表します。
*2. 速度「0」での動作、偏差カウンタリセット中、同期制御中、および多軸協調動作中も演算している状態に含まれます。
*3. TRUE でも以下の場合は、原点の再確定が必要になります。
- 位置カウント設定、または単位変換設定を変更した場合。
- サーボドライバ側で、絶対位置情報を喪失する異常発生や操作を行った場合。たとえば、エンコーダケーブルの断線や絶対値エンコーダデータのクリアなど。

*4. 「VelLimit」は、同期制御中の従軸にだけ使用してください。
*5. 指令している移動方向を表します。
*6. ドライバなどの状態を表します。
*7. Sysmac Studio の［軸基本設定］画面の［詳細設定］で、［デジタル入力］の［エンコーダ Z 相検出］に設定した信号の状態です。他社製ドライバでは、この信号を PDO にマッピングできない場合があります。接続するドライバのマニュアルを確認してください。
*8. PDO にマッピングされた「ステータスワード(6041Hex)」のビット 11（内部制限機能有効）の状態を表示しています。TRUE となる条件はサーボドライバの仕様に依存します。接続したサーボドライバのマニュアルを参照してくだ

さい。オムロン製サーボドライバ 1S シリーズおよび G5 シリーズでは、 トルク制限／速度制限／駆動禁止入力／ソフトウェアリミットのいずれかです。

*9. PDO にマッピングされた「オペレーションモード表示(6061Hex)」の値をもとに表示しています。「CSP」、「CSV」、「CST」が TRUE となる条件はサーボドライバの仕様に依存します。接続したサーボドライバのマニュアルを参照してください。
PDO に「オペレーションモード表示 (6061Hex)」がマッピングされていない場合、CPU ユニットのバージョンにより、次のようになります。
- CPU ユニット Ver.1.10 以降では、PDO にマッピングされた「ステータスワード (6041Hex)」の状態が Operation Enabled の場合に TRUE となります。
- CPU ユニット Ver.1.09 以前では、常に FALSE になります。

*10. 軸に割りついている EtherCAT スレーブ、または NX ユニットと CPU ユニット間のプロセスデータ通信が未確立となった場合、CPU ユニットのバージョンにより、次のようになります。
- CPU ユニット Ver.1.10 以降では、軸変数のフィードバック現在位置および指令現在位置は、プロセスデータ通信が未確立となる直前のフィードバック現在位置となります。
- CPU ユニット Ver.1.09 以前では、軸変数のフィードバック現在位置および指令現在位置は、「0」または下限値となります。下限値となる条件は、カウントモードがロータリモードで位置範囲に「0」を含まない場合です。

*11. ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで使用可能です。
*12. 軸基本設定の設定内容を表します。
*13. NX シリーズ 位置インタフェースユニットの場合は、位置インタフェースユニットを装着した EtherCAT カプラユニットのノードアドレスを表します。
*14. ユニットバージョン Ver.1.10 以降の CPU ユニットと Sysmac Studio Ver.1.12 以降の組み合わせで使用可能です。
*15. 電子ギア比の設定値を表します。
*16. ユニットバージョン Ver.1.11 以降の CPU ユニットで追加された単位変換設定の減速機使用を「使用する」に設定している場合、このパラメータは無効となります。代わりに有効なパラメータとなる［ワーク側 1 回転の移動量］、［ワーク側ギア比］、［モータ側ギア比］の値を確認する場合は、MC_ReadAxisParameter（軸パラメータ読出）命令を使用してください。
*17. カウントモードがリニアモードの場合は、オーバーフロー直前の位置を表示します。ロータリモードの場合は、リングカウンタ上限値を表示します。
*18. カウントモードがリニアモードの場合は、アンダーフロー直前の位置を表示します。ロータリモードの場合は、リングカウンタ下限値を表示します。

2-1-3 軸グループ変数

軸グループ変数は、システム定義変数の変数名が_MC_GRP[0-63]や_MC1_GRP[0-63]、_MC2_GRP[0-63]で、データ型が_sGROUP_REF 型の構造体型の変数です。
ここでは、_MC_GRP[0-63]を例に軸グループ変数の構成とそのメンバの詳細を説明しています。
_MC1_GRP[0-63]や_MC2_GRP[0-63]についても同様です。
なお、機能説明で軸変数は、_MC_AX[*]を例に説明していますが、_MC1_AX[*]、_MC2_AX[*]の場合も同様です。

変数名	データ型	名称	機能
_MC_GRP[0-63]	_sGROUP_REF	軸グループ変数	
Status	_sGROUP_REF_STA	軸グループステータス	
Ready	BOOL	起動準備完了	軸グループが停止中で起動準備が整っているときに TRUE となります。 以下の AND 条件が起動準備が整う条件となります。 • 構成軸で MC_Stop 命令が実行されていない。 • _MC_GRP[*].Status.Standby（停止中）が TRUE • 構成軸がサーボ ON 状態 • 構成軸の_MC_AX[*].Details.Homed（原点確定）が TRUE

変数名	データ型	名称	機能
Disabled	BOOL	軸グループ無効	軸グループ無効で停止しているときに TRUE となります。 以下の軸グループステータスは排他的で、どれか 1 つだけが TRUE になります。 Disable/ Standby/ Moving/ Stopping/ ErrorStop
Standby	BOOL	停止中	軸グループ動作命令が停止しているときに TRUE となります。 軸グループを構成する軸のサーボ ON/OFF 状態には関係しません。
Moving	BOOL	動作中	目標位置に向かって、軸グループ動作命令を実行しているときに TRUE となります。 インポジション待ちの状態や、オーバーライドによる、速度「0」の状態も含まれます。
Stopping	BOOL	減速停止中	MC_GroupStop 命令により軸グループが停止するまでの状態のとき TRUE となります。 MC_GroupStop 命令で軸が停止した後に「Execute」が TRUE のときも含みます。 この状態では軸グループ動作命令を起動できません。(CommandAborted=TRUE)
ErrorStop	BOOL	エラー減速停止中	MC_GroupImmediateStop 命令、または軸グループ 軽度フォールト発生中 (_MC_GRP[*].MFaultLvl.Active=TRUE) により、軸グループが停止するまで、および停止しているとき TRUE となります。 この状態では軸グループ動作命令を起動できません。(CommandAborted=TRUE)
Details	_sGROUP_REF_DET	軸グループ制御ステータス *1	
Idle	BOOL	停止中	インポジション待ちを除き、指令値を演算していないときに TRUE となります。*2 「Idle」と「InPosWaiting」は排他的で、同時に TRUE にはなりません。
InPosWaiting	BOOL	インポジション待ち	いずれかの構成軸がインポジション待ちのときに TRUE となります。 インポジションチェックを行う位置決めで、インポジションチェックを行っている状態です。
Cmd	_sGROUP_REF_CMD_DATA	軸グループ指令値	
Vel	LREAL	指令補間速度	指令補間速度の現在値を出力します。 補間指令現在位置の差分から算出した補間速度です。正方向移動時はプラス符号、負方向移動時はマイナス符号がつきます。 軸グループが無効の場合は「0」となります。
AccDec	LREAL	指令補間加減速度	指令補間加減速度の現在値を出力します。 指令補間速度の差分から算出した補間加減速度です。加速時はプラス符号、減速時はマイナス符号がつきます。 軸グループが無効の場合、および実行中の軸グループ動作命令の指令加減速度の指定が「0」の場合は「0」となります。
MFaultLvl	_sMC_REF_EVENT	軸グループ 軽度フォールト	
Active	BOOL	軸グループ 軽度フォールト発生中	軸グループ 軽度フォールトが発生しているときに TRUE となります。

変数名	データ型	名称	機能
Code	UINT	軸グループ 軽度フォールトコード	軸グループ 軽度フォールトのコードを出力します。 イベントコードの上 4 けたと同じ値です。
Obsr	_sMC_REF_EVENT	軸グループ 監視情報	
Active	BOOL	軸グループ 監視情報発生中	軸グループ 監視情報が発生しているときに TRUE となります。
Code	WORD	軸グループ 監視情報コード	軸グループ 監視情報のコードを出力します。 イベントコードの上 4 けたと同じ値です。
Cfg	_sGROUP_REF_CFG	軸グループ基本設定	
GrpNo	UINT	軸グループ番号	軸グループの論理番号を表します。 _sGROUP_REF の参照時に軸グループ番号を認識するために参照します。
GrpEnable	_eMC_GROUP_USE	軸グループ使用	軸グループを使用するかしないかを表します。 0: _mcNoneGroup （未作成軸グループ） 1: _mcUnusedGroup （未使用軸グループ） 2: _mcUsedGroup （使用軸グループ）
ExecID *3	UINT	実行 ID	割り付けしたタスク別の実行 ID を表します。 0: タスク割付なし（未作成グループ） 1: プライマリ定周期タスクに割り付け 2: 定周期タスク（実行優先度 5）に割り付け
Kinematics	_sGROUP_REF_KIM	運動学変換設定 *4	
GrpType	_eMC_GROUP_TYPE	軸構成	多軸協調制御における軸構成を表します。 0: _mcXY（2 軸） 1: _mcXYZ（3 軸） 2: _mcXYZU（4 軸）
Axis[0]	UINT	構成軸選択（軸 A0）	軸 A0 に割り付けられた軸番号を表します。
Axis[1]	UINT	構成軸選択（軸 A1）	軸 A1 に割り付けられた軸番号を表します。
Axis[2]	UINT	構成軸選択（軸 A2）	軸 A2 に割り付けられた軸番号を表します。
Axis[3]	UINT	構成軸選択（軸 A3）	軸 A3 に割り付けられた軸番号を表します。

*1. 指令の制御状態を表します。
*2. 速度「0」での動作も演算している状態に含まれます。
*3. ユニットバージョン Ver.1.10 以降の CPU ユニットと Sysmac Studio Ver.1.12 以降の組み合わせで使用可能です。
*4. 軸グループの運動学変換のための定義を表します。

2-1-4 モーション制御命令の入力変数

モーション制御命令の入力変数の一覧と有効範囲の一覧および列挙型の有効範囲の一覧です。

入力変数一覧

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。 「Execute」の立ち上がり時に、ほかの入力変数が取り込まれます。 入力値を更新したい場合は、入力値を変更し、「Execute」を再度立ち上げたときになります。 命令の実行が完了した後も「Execute」が TRUE の間は、出力変数が有効です。 その後、「Execute」の立ち下がりエッジで、「Error」と「ErrorID」以外の出力変数は無効になります。 命令の実行完了前に「Execute」を FALSE にした場合は、出力変数が少なくとも 1 周期の間は有効です。
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE にすると命令の機能が有効になり、FALSE すると無効になります。 「Enable」が TRUE の間、ほかの入力変数は毎周期取り込まれます。 「Enable」を FALSE にすると「Error」と「ErrorID」以外の出力変数は無効になります。
PositiveEnable	正方向有効	BOOL	TRUE, FALSE	FALSE	• MC_MoveJog 命令 TRUE にすると正方向へ移動を開始します。FALSE にすると移動を終了します。MC_MoveJog 命令の「Velocity」、「Acceleration」、「Deceleration」は、「PositiveEnable」の立ち上がり時の値が取り込まれます。 • MC_SetTorqueLimit 命令 TRUE にすると正方向のトルク制限を有効にします。FALSE にすると正方向のトルク制限を無効にします。
NegativeEnable	負方向有効	BOOL	TRUE, FALSE	FALSE	• MC_MoveJog 命令 TRUE にすると負方向へ移動を開始します。FALSE にすると移動を終了します。MC_MoveJog 命令の「Velocity」、「Acceleration」、「Deceleration」は、「NegativeEnable」の立ち上がり時の値が取り込まれます。 • MC_SetTorqueLimit 命令 TRUE にすると負方向のトルク制限を有効にします。FALSE にすると負方向のトルク制限を無効にします。

入力変数	名称	データ型	有効範囲	初期値	内容
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *1	モーション制御命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2 :ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ
Velocity	目標速度	LREAL	正数 *2	0	目標速度を指定します。*3
Acceleration	加速度	LREAL	正数または「0」	0	加速度を指定します。*4
Deceleration	減速度	LREAL	正数または「0」	0	減速度を指定します。*4
Jerk	ジャーク	LREAL	正数または「0」	0	ジャークを指定します。*5
Distance	移動距離	LREAL	負数、正数、「0」	0	指令現在位置からの移動距離を指定します。
		ARRAY [0..3] OF LREAL	負数、正数、「0」	0	直線補間の目標位置を指定します。*6
Position	目標位置	LREAL	負数、正数、「0」	0	絶対座標の目標位置を指定します。*6
		ARRAY [0..3] OF LREAL	負数、正数、「0」	0	直線補間の目標位置を指定します。*6
VelFactor	速度オーバーライド値	LREAL	0～500	100	速度のオーバーライド値を指定します。 オーバーライド値の有効範囲は「0.01 ～ 500.00」となります。 「500.00 以上」は「500」として扱われ、「0.01 未満（負数も含む）」は「0.01」として扱われます。 「0」を指定したときだけ「0」として動作します。 単位は［%］となります。
AccFactor (Reserved)	加減速度オーバーライド値	LREAL	0～500	100	(Reserved)
JerkFactor (Reserved)	ジャークオーバーライド値	LREAL	0～500	100	(Reserved)
ReferenceType *7	位置種別選択	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *1	主軸の入力情報を指定します。 0: 指令位置（直前タスク周期 *8 での計算値） 1: フィードバック位置（同一タスク周期 *8 での取得値） 2: 指令位置（同一タスク周期 *8 での計算値）
FeedDistance	定寸距離	LREAL	負数、正数、「0」	0	割り込み入力後の移動距離を指定します。
FeedVelocity	定寸速度	LREAL	正数	0	割り込み入力後の移動目標速度を指定します。
ErrorDetect	エラー検出選択	BOOL	TRUE, FALSE	FALSE	割り込み入力が無いときに、異常検出をするかを指定します。 TRUE: 異常検出有 FALSE: 異常検出無

入力変数	名称	データ型	有効範囲	初期値	内容
Periodic	繰り返しモード	BOOL	TRUE, FALSE	FALSE	指定されたカムテーブルを繰り返し実行するか、1 回しか実行しないかを指定します。 TRUE: 繰り返し FALSE: 繰り返しなし
StartMode	開始位置方式指定	_eMC_START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	0 *1	「MasterStartDistance」（主軸追従距離）が使用する座標系を指定します。 0: 絶対位置 1: 相対位置
StartPosition	カムテーブル開始位置	LREAL	負数、正数、「0」	0	カムテーブルの始点（位相=0）を、主軸の絶対位置で指定します。
MasterStartDistance	主軸追従距離	LREAL	負数、正数、「0」	0	従軸がカム動作を開始する際の主軸の位置を指定します。 「StartMode」（開始位置方式指定）で［絶対位置決め］を指定した場合は主軸の絶対位置を指定します。また、［相対位置決め］を指定した場合は「StartPosition」（カムテーブル開始位置）からの相対量を指定します。
MasterScaling	主軸係数	LREAL	正値(>0.0)	1.0	指定されたスケールで、主軸の位相を拡大／縮小します。
SlaveScaling	従軸係数	LREAL	正値(>0.0)	1.0	指定されたスケールで、従軸の変位を拡大／縮小します。
MasterOffset	主軸オフセット	LREAL	負数、正数、「0」	0	指定されたオフセット値で、主軸の位相をずらします。
SlaveOffset	従軸オフセット	LREAL	負数、正数、「0」	0	指定されたオフセット値で、従軸の変位をずらします。
CamTransition (Reserved)	カムトランジション指定	_eMC_CAM_TRANSITION	0: _mcCTNone	0 *1	(Reserved)
OutMode (Reserved)	同期解除モード選択	_eMC_OUT_MODE	0: _mcStop	0 *1	(Reserved)
CamMonitorMode *9	カムモニタモード選択	_eMC_CAM_MONITOR_MODE	0: _mcCalcCamDistanceDiff	0 *1	モニタするカム動作の情報を指定します。 0: 変位偏差計算
Direction	方向選択	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	4/0 *1	動作する方向を指定します。 0: 正方向指定 1: 近回り指定 2: 負方向指定 3: 現在方向指定 4: 方向指定なし
Continuous (Reserved)	継続方法選択	BOOL	TRUE, FALSE	FALSE	(Reserved)
RatioNumerator	ギア比分子	DINT *10	正数または負数 *10	10000	主軸と従軸間の電子ギアの分子を指定します。
RatioDenominator	ギア比分母	UDINT *11	正数	10000	主軸と従軸間の電子ギアの分母を指定します。
MasterSyncPosition	主軸同期位置	LREAL	負数、正数、「0」	0	絶対座標の主軸同期位置を指定します。
SlaveSyncPosition	従軸同期位置	LREAL	負数、正数、「0」	0	絶対座標の従軸同期位置を指定します。
SlaveDistance	従軸移動距離	LREAL	負数、正数、「0」	0	従軸の移動距離を指定します。

入力変数	名称	データ型	有効範囲	初期値	内容
MasterDistance	主軸移動距離	LREAL	正数または「0」	0	主軸の移動距離を指定します。
MasterDistanceInACC	主軸加速移動距離	LREAL	正数または「0」	0	従軸加速時の主軸の移動距離を指定します。
MasterDistanceInDEC	主軸減速移動距離	LREAL	正数または「0」	0	従軸減速時の主軸の移動距離を指定します。
LinkOption	同期開始条件選択	_eMC_LINKOPTION	0: _mcCommandExecution 1: _mcTriggerDetection 2: _mcMasterReach	0 *1	従軸が主軸に同期する条件を指定します。 0: 起動時 1: トリガを検出した時 2: 主軸が主軸追従距離に到達した時
CombineMode	加減算方法選択	_eMC_COMBINE_MODE	0: _mcAddAxes 1: _mcSubAxes	0 *1	合成方法を指定します。 0: 加算 1: 減算
RatioNumeratorMaster (Reserved)	主軸ギア比分子	DINT *10	正数または負数 *10	10000	主軸と従軸間の電子ギアの分子を指定します。
RatioDenominatorMaster (Reserved)	主軸ギア比分母	UDINT *11	正数	10000	主軸と従軸間の電子ギアの分母を指定します。
RatioNumeratorAuxiliary (Reserved)	補助軸ギア比分子	DINT *10	正数または負数 *10	10000	補助軸と従軸間の電子ギアの分子を指定します。
RatioDenominatorAuxiliary (Reserved)	補助軸ギア比分母	UDINT *11	正数	10000	補助軸と従軸間の電子ギアの分母を指定します。
ReferenceTypeMaster	主軸位置種別選択	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 *1	主軸の位置種別を指定します。 1: フィードバック位置（同一タスク周期 *8 での取得値） 2: 指令位置（同一タスク周期 *8 での計算値）
ReferenceTypeAuxiliary	補助軸位置種別選択	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 *1	補助軸の位置種別を指定します。 1: フィードバック位置（同一タスク周期 *8 での取得値） 2: 指令位置（同一タスク周期 *8 での計算値）
PhaseShift	位相補正量	LREAL	負数、正数、「0」	0	主軸の位相補正量を指定します。*6
Torque	目標トルク	LREAL	0～1000.0	300.0	サーボドライバへ出力する目標トルクを 0.1%単位で指定します。 定格トルクを 100%としたときの比率で指定します。単位は「%」です。
TorqueRamp	トルクランプ	LREAL	正数または「0」	0	現在のトルクから目標トルクを出力するまでのトルクの変化率を指定します。 単位は「%/s」です。

入力変数	名称	データ型	有効範囲	初期値	内容
PositiveValue	正方向トルク制限値	LREAL	0.1 ～ 1000.0、または「0.0」	300.0	正方向へのトルク制限値を 0.1% 単位で指定します。 軸パラメータの［正方向トルク制限上限値］を超える値が入力された場合、［正方向トルク制限上限値］となります。 「0」および負数を指定したときは、「0」として動作します。
NegativeValue	負方向トルク制限値	LREAL	0.1 ～ 1000.0、または「0.0」	300.0	負方向へのトルク制限値を 0.1% 単位で指定します。 軸パラメータの［負方向トルク制限上限値］を超える値が入力された場合、［負方向トルク制限上限値］となります。 「0」および負数を指定したときは、「0」として動作します。
WindowOnly	ウィンドウ有効	BOOL	TRUE, FALSE	FALSE	ウィンドウマスクの有効／無効を指定します。
FirstPosition	開始位置	LREAL	負数、正数、「0」	0	開始位置を指定します。
LastPosition	終了位置	LREAL	負数、正数、「0」	0	終了位置を指定します。
StopMode	停止方法選択	_eMC_STOP_MODE	1: _mcImmediateStop 2: _mcImmediateStopFEReset 3: _mcFreeRunStop 4: _mcNonStop	4 *1	停止方法を指定します。 1: 即停止します 2: 即停止し偏差カウンタリセットを行います 3: 即停止しサーボ OFF にします 4: 停止しません
Relative (Reserved)	相対位置選択	BOOL	TRUE, FALSE	FALSE	(Reserved)
ExecutionMode (Reserved)	実行モード選択	_eMC_EXECUTION_MODE	0: _mcImmediately	0 *1	(Reserved)
PermittedDeviation	軸間偏差許容値	LREAL	正数または「0」	0	許容可能な主軸と従軸の位置偏差の最大値を指定します。
CmdPosMode	指令現在位置カウント選択	_eMC_CMDPOS_MODE	0: _mcCount	0 *1	0: フィードバック現在位置を使用し、指令現在位置を更新します。 原点確定状態が継続します。
CoordSystem	座標系	_eMC_COORD_SYSTEM	0: _mcACS	0 *1	座標系を指定します。 0: 軸座標系(ACS)
TransitionMode	トランジションモード（遷移モード）	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *1	動作の経路を指定します。 0: トランジション無効 10: コーナースーパーインポーズ
MoveMode	移動方法選択	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative	0 *1	移動方法を選択します。 0: 絶対値位置決め 1: 相対値位置決め
CircAxes	円弧軸指定	ARRAY [0,1] OF UINT	0～3	0	円弧補間を行う軸を指定します。 0: 軸 A0 1: 軸 A1 2: 軸 A2 3: 軸 A3
CircMode	円弧補間モード	_eMC_CIRC_MODE	0: _mcBorder 1: _mcCenter 2: _mcRadius	0 *1	円弧補間の方法を指定します。 0: 通過点指定 1: 中心点指定 2: 半径指定

入力変数	名称	データ型	有効範囲	初期値	内容
AuxPoint	補助点	ARRAY [0,1] OF LREAL	負数、正数、「0」	0	通過点位置／中心位置／半径を指定します。
EndPoint	終点	ARRAY [0,1] OF LREAL	負数、正数、「0」	0	目標位置を指定します。
PathChoice	経路選択	_eMC_CIRC_PATHCHOICE	0: _mcCW 1: _mcCCW	0 *1	経路方向を指定します。 0: CW 1: CCW
ParameterNumber	パラメータ番号	_eMC_PARAMETER_NUMBER	0: _mcChkVel 1: _mcChkAcc 2: _mcChkDec 3: _mcPosiChkTrq *12 4: _mcNegaChkTrq *12 5: _mcFELmt 6: _mcChkFELmt 7: _mcSwLmtMode 8: _mcPosiSwLmt 9: _mcNegaSwLmt 10:_mcInPosTime 11:_mcInPosRange *13 12:_mcStartVel *14	0 *1	書き込み先のパラメータを指定します。 0: 速度警告値／補間速度警告値 1: 加速度警告値／補間加速度警告値 2: 減速度警告値／補間減速度警告値 3: 正方向トルク警告値 4: 負方向トルク警告値 5: 位置偏差オーバー値 6: 位置偏差警告値 7: ソフトウェアリミット機能 8: 正方向ソフトウェアリミット 9: 負方向ソフトウェアリミット 10: インポジションチェック時間 11: インポジション幅 12: 起動速度
HomingMode *15	原点復帰動作	_eMC_HOMING_MODE	0: _mcHomeSwTurnHomeSwOff 1: _mcHomeSwTurnHomeSwOn 4: _mcHomeSwOff 5: _mcHomeSwOn 8: _mcLimitInputOff 9: _mcHomeSwTurnHomeMask 11: _mcLimitInputOnly 12: _mcHomeSwTurnHoldingTime 13: _mcNoHomeSwHoldingHomeInput 14: _mcHomePreset	0 *1	書き換える［原点復帰動作］を指定します。 0: 近傍退避･原点近傍入力 OFF 指定 1: 近傍退避･原点近傍入力 ON 指定 4: 原点近傍入力 OFF 指定 5: 原点近傍入力 ON 指定 8: 限界入力 OFF 指定 9: 近傍退避･原点入力マスク距離指定 11: 限界入力のみ 12: 近傍退避･押し当て時間指定 13: 原点近傍入力なし･押し当て原点入力指定 14: 原点プリセット
AxisUse *16	軸使用	_eMC_AXIS_USE	1: _mcUnusedAxis 2: _mcUsedAxis	1 *1	使用軸または未使用軸を指定します。 1: 未使用軸 2: 使用軸
EnableMask *17	トラック有効	WORD	16#0000～FFFF	0	トラックごとに有効または無効を指定します。最大 16 トラックで、0 ビットめの値がトラック番号 0 の、15 ビットめの値がトラック番号 15 の有効または無効を指定します。 0: 無効、1: 有効
ValueSource (Reserved)	入力情報	_sMC_SOURCE	－	－	(Reserved)

入力変数	名称	データ型	有効範囲	初期値	内容
TimeStamp *17	タイムスタンプ	ULINT	正数、「0」	0	位置を算出するタイムスタンプを指定します。 タイムスタンプ方式対応のデジタル入力ユニットやエンコーダ入力ユニット、オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプ内の時刻を基準とするタイムスタンプを指定します。単位は［ns］です。
ExecID *18	実行 ID	UINT	2	2	変数同期指定先の ID を指定します。 2: 定周期タスク（実行優先度 5）
OffsetPosition *14	位置オフセット	LREAL	負数、正数、「0」	0	指令現在位置に加算する位置オフセットを指定します。*6

*1. 有効範囲が列挙型の変数は、実際の初期値は数値ではなく、列挙子となります。
*2. MC_MoveJog 命令および MC_MoveVelocity 命令など、「0」を設定できる命令もあります。
*3. 指令単位／s となります。指令単位は［mm］、［μm］、［nm］、［degree］、［inch］ または［pulse］で表されます。
*4. 指令単位／s^2 となります。
*5. 指令単位／s^3 となります。
*6. 指令単位となります。
*7. _mcLatestCommand を使用する場合は、次の関係となるように主軸、従軸を使用しなければなりません。
Master（主軸）に設定されているモーション制御システム変数の軸番号 < Slave（従軸）に設定されているモーション制御システム変数の軸番号
*8. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。
*9. 形 NX102 CPU ユニットでは、CPU ユニットのユニットバージョン Ver.1.32 以降の CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。
形 NX1P2 CPU ユニット、NJ シリーズ CPU ユニットでは、ユニットバージョン Ver.1.21 以降の CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。
*10. ユニットバージョン Ver.1.02 以降の CPU ユニットと Sysmac Studio Ver.1.03 以降の組み合わせで使用可能です。
それ以前の組み合わせでは、データ型は「UINT」、有効範囲「正数」となります。
*11. ユニットバージョン Ver.1.02 以降の CPU ユニットと Sysmac Studio Ver.1.03 以降の組み合わせで使用可能です。
それ以前の組み合わせでは、データ型は「UINT」となります。
*12. トルク制御時にだけ有効です。
*13. ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせで使用可能です。
*14. ユニットバージョン Ver.1.10 以降の CPU ユニットと Sysmac Studio Ver.1.12 以降の組み合わせで使用可能です。
*15. ユニットバージョン Ver.1.03 以降の CPU ユニットと Sysmac Studio Ver.1.04 以降の組み合わせで使用可能です。
*16. ユニットバージョン Ver.1.04 以降の CPU ユニットと Sysmac Studio Ver.1.05 以降の組み合わせで使用可能です。
*17. ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで使用可能です。
*18. 形 NX102 CPU ユニット、形 NX1P2 CPU ユニットおよび NJ シリーズ CPU ユニットにはありません。

入力変数の有効範囲

ここではモーション制御命令の入力変数の有効範囲について説明しています。命令ごとの有効範囲については、各命令の説明を参照してください。

● BOOL 型の入力変数

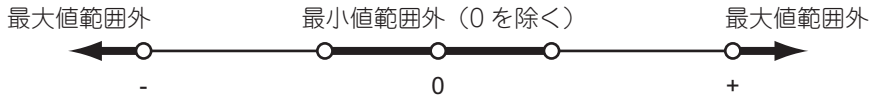
FALSE 以外を TRUE とみなします。そのため、範囲外となる異常は発生しません。

● 列挙(ENUM)型の入力変数

有効範囲外を異常とします。

● 有効範囲が「フルレンジ」、「正数」、「負数」と記載されている入力変数

各入力変数の有効範囲と範囲外の数値が設定された場合の動作は、次の表のとおりです。



入力変数	名称	有効範囲	最大値範囲外	最小値範囲外（0 を除く）
Velocity	速度	0、または（-1≦ かつ ≦-［最高速度］）、（1≦ かつ ≦［最高速度］）*1	正数の場合は［最高速度］で、負数の場合は-［最高速度］で固定となります。*2	正数の場合は 1pulse/s で、負数の場合は-1 pulse/s で、固定となります。*2
Acceleration	加速度	0、または（0.004 ≦ かつ ≦［最大加速度］）*3	［最大加速度］で固定となります。 ただし、［加速時間］*4 が 125μs 未満となる場合は、125μs で固定となります。	正数の場合は、0.004pulse/s^2 で固定となります。 ただし、［加速時間］*4 が 250s を超過する場合は、250s で固定となります。 負数の場合は、異常となります。
Deceleration	減速度	0、または（0.004 ≦ かつ ≦［最大減速度］）*5	［最大減速度］で固定となります。 ただし、［減速時間］*4 が 125μs 未満となる場合は、125μs で固定となります。	正数の場合は、0.004pulse/s^2 で固定となります。 ただし、［減速時間］*4 が 250s を超過する場合は、250s で固定となります。 負数の場合は、異常となります。
Jerk	ジャーク	0、または（0.000016 ≦ かつ 25,600,000,000,000,000 pulse/s^3）	25,600,000,000,000,000 pulse/s^3 で固定となります。 ただし、［加速ジャーク適用時間］*6、または［減速ジャーク適用時間］*6 が 125μs 未満となる場合は、それぞれ 125μs で固定となります。	0.000016pulse/s^3 で固定となります。 ただし、［加速ジャーク適用時間］*6、または［減速ジャーク適用時間］*6 が 250s を超過する場合は、それぞれ 250s で固定となります。 負数の場合は異常となります。
Distance	移動距離	(0xFFFFFF0000000001) ≦ かつ ≦(0x000000FFFFFFFFFF)	異常となります。	最小値範囲外はありません。
Position	指令位置	(0xFFFFFF8000000000) ≦ かつ <(0x0000007FFFFFFFFF+1) *7	異常となります。	最小値範囲外はありません。
VelFactor	オーバーライド値	0、または 0.01≦ かつ ≦500.00 *8	500.00％を超えるものは 500.00%固定となります。	0.01％未満は 0.01％固定となります。
Velocity	補間速度	0.000 000 000 1≦ かつ ≦［最高補間速度］*9	［最高補間速度］で固定となります。	0.000 000 000 1 pulse/s で固定となります。
Acceleration	補間加速度	0、または（0.000 000 000 000 4≦ かつ ≦［最大補間加速度］）*10	［最大補間加速度］で固定となります。 ただし、［補間加速時間］*11 が 125μs 未満となる場合は、125μs で固定となります。	正数の場合は、0.000 000 000 000 4 pulse/s^2 で固定となります。ただし、［補間加速時間］*11 が 250s を超過する場合は、250s で固定となります。 負数の場合は、異常となります。

入力変数	名称	有効範囲	最大値範囲外	最小値範囲外（0 を除く）
Deceleration	補間減速度	0、または（0.000 000 000 000 4≦ かつ ≦［最大補間減速度］）*12	［最大補間減速度］で固定となります。 ただし、［補間減速時間］*11 が 125μs 未満となる場合は、125μs で固定となります。	正数の場合は、0.000 000 000 000 4 pulse/s^2 で固定となります。ただし、［補間減速時間］*11 が 250s を超過する場合は、250s で固定となります。 負数の場合は、異常となります。
Jerk	補間ジャーク	0、または（0.000 000 000 000 0016≦ かつ ≦ 51,200,000,000,000,000 pulse/s^3）	51,200,000,000,000,000 pulse/s^3 で固定となります。 ただし、［補間加速ジャーク適用時間］*13、または［補間減速ジャーク適用時間］*13 が 125μs 未満となる場合は、それぞれ 125μs で固定となります。	0.000 000 000 000 001 6 pulse/s^3 で固定となります。 ただし、［補間加速ジャーク適用時間］*13、または［補間減速ジャーク適用時間］*13 が 250s を超過する場合は、それぞれ 250s で固定となります。 負数の場合は、異常となります。

*1. 設定できる最大値は、パルス単位に変換した値で次のとおりです。
- ユニットバージョン Ver.1.11 以降の CPU ユニットは、2,147,483,647[pulse/s]
- ユニットバージョン Ver.1.03 以降で Ver.1.10 以前の CPU ユニットは、500,000,000[pulse/s]
- ユニットバージョン Ver.1.02 以前の CPU ユニットは、400,000,000[pulse/s]

*2. 有効範囲に負数や「0」を含まない場合に、負数や「0」を指定すると異常となります。
*3. 軸パラメータ［最大加速度］の上限は 3,200,000,000,000 pulse/s^2 となります。
*4. 「加速時間＝速度／加速度」、「減速時間＝速度／減速度」、「加減速時間＝加速時間+減速時間」で計算します。
*5. 軸パラメータ［最大減速度］の上限は 3,200,000,000,000 pulse/s^2 となります。
*6. 「加速ジャーク適用時間」、「減速ジャーク適用時間」とは、ジャークが適用される時間のことです。
「加速ジャーク適用時間=加速度／ジャーク」、「減速ジャーク適用時間=減速度／ジャーク」で計算します。
*7. パルス単位の絶対位置で符号付 40bit の範囲内である必要があります。
*8. 単位は「%」となります。
*9. 軸パラメータ［最高補間速度］の上限は、軸パラメータ［最高速度］の上限の 2 倍となります。
*10. 軸パラメータ［最大補間加速度］の上限は、6,400,000,000,000 pulse/s^2 となります。
*11. 「補間加速時間＝補間速度／補間加速度」、「補間減速時間＝補間速度／補間減速度」、「加減速時間＝加速時間+減速時間」で計算します。
*12. 軸パラメータ［最大補間減速度］の上限は 6,400,000,000,000 pulse/s^2 となります。
*13. 「補間加速ジャーク適用時間」、「補間減速ジャーク適用時間」とは、補間ジャークが適用される時間のことです。
「補間加速ジャーク適用時間=補間加速度／補間ジャーク」、「補間減速ジャーク適用時間=補間減速度／補間ジャーク」で計算します。

列挙型一覧

モーション制御命令の入力変数で扱う ENUM 型の一覧です。
列挙型の入力変数は、数値による設定ではなく、列挙子による設定となります。

データ型	有効範囲	内容	適応する命令の変数（変数の名称）
_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	モーション制御命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2: ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ	BufferMode（バッファモード選択）

データ型	有効範囲	内容	適応する命令の変数（変数の名称）
_eMC_CIRC_MODE	0: _mcBorder 1: _mcCenter 2: _mcRadius	円弧補間の方法を指定します。 0: 通過点指定 1: 中心点指定 2: 半径指定	CircMode （円弧補間モード）
_eMC_CAM_TRANSITION	0: _mcCTNone	カム再起動時における従軸指令値の出力方法を指定します。 0: 制限・修正なし	CamTransition （カムトランジション指定）
_eMC_CIRC_PATHCHOICE	0: _mcCW 1: _mcCCW	経路方向を指定します。 0: CW 1: CCW	PathChoice （経路選択）
_eMC_COMBINE_MODE	0: _mcAddAxes 1: _mcSubAxes	合成方法を指定します。 0: 加算 1: 減算	CombineMode （加減算方法選択）
_eMC_COORD_SYSTEM	0: _mcACS	座標系を指定します。 0: 軸座標系(ACS)	CoordSystem （座標系）
_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	各種動作の方向を指定します。 0: 正方向指定 1: 近回り指定 2: 負方向指定 3: 現在方向指定 4: 方向指定なし	Direction （方向選択）
_eMC_EXECUTION_MODE	0: _mcImmediately	(Reserved)	ExecutionMode （実行モード選択）
_eMC_LINKOPTION	0: _mcCommandExecution 1: _mcTriggerDetection 2: _mcMasterReach	従軸が主軸に同期する条件を指定します。 0: 命令開始時 1: トリガを検出した時 2: 主軸が主軸追従距離に到達した時	LinkOption （同期開始条件選択）
_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative 2: _mcVelocity	移動方法を選択します。 0: 絶対値位置決め 1: 相対値位置決め 2: 速度制御	MoveMode （移動方法選択）
_eMC_OUT_MODE	0: _mcStop	同期制御命令を解除するモードを指定します。 0: 減速停止	OutMode （同期解除モード選択） (Reserved)
_eMC_PARAMETER_NUMBER	0: _mcChkVel 1: _mcChkAcc 2: _mcChkDec 3: _mcPosiChkTrq *1 4: _mcNegaChkTrq *1 5: _mcFELmt 6: _mcChkFELmt 7: _mcSwLmtMode 8: _mcPosiSwLmt 9: _mcNegaSwLmt 10: _mcInPosTime 11: _mcInPosRange *2 12: _mcStartVel *3	書き込み先のパラメータを指定します。 0: 速度警告値／補間速度警告値 1: 加速度警告値／補間加速度警告値 2: 減速度警告値／補間減速度警告値 3: 正方向トルク警告値 4: 負方向トルク警告値 5: 位置偏差オーバー値 6: 位置偏差警告値 7: ソフトウェアリミット機能 8: 正方向ソフトウェアリミット 9: 負方向ソフトウェアリミット 10: インポジションチェック時間 11: インポジション幅 12: 起動速度	ParameterNumber （パラメータ番号）

データ型	有効範囲	内容	適応する命令の変数（変数の名称）
_eMC_SWLMT_MODE	0: _mcNonSwLmt 1: _mcCmdDecelerationStop 2: _mcCmdImmediateStop 3: _mcActDecelerationStop 4: _mcActImmediateStop	ソフトウェアリミット機能の有効／無効、および停止方法を指定します。 0: ソフトウェアリミット機能の無効 1: 有効で、指令位置に対して減速停止 2: 有効で、指令位置に対して即停止 3: 有効で、フィードバック位置に対して減速停止 4: 有効で、フィードバック位置に対して即停止	SettingValue （設定値）
_eMC_REFERENCE_TYPE *4	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	位置の種別を指定します。 0: 指令位置（直前タスク周期*5 での計算値） 1: フィードバック位置（同一タスク周期*5 での取得値） 2: 指令位置（同一タスク周期*5 での計算値）	ReferenceType （位置種別選択） ReferenceTypeMaster （主軸位置種別選択） ReferenceTypeAuxiliary （補助軸位置種別選択）
_eMC_START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	「MasterStartDistance」（主軸追従距離）が使用する座標系を指定します。 0: 絶対位置 1: 相対位置	StartMode （開始位置方式指定）
_eMC_STOP_MODE	0: _mcDeccelerationStop 1: _mcImmediateStop 2: _mcImmediateStopFEReset 3: _mcFreeRunStop 4: _mcNonStop	停止方法を指定します。 0: 指令値を減速停止 1: 即停止します。 2: 即停止し偏差カウンタリセットします。 3: サーボ OFF します。 4: 停止しません。	StopMode （停止方法選択）
_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	2 つのラッチ機能のうちのどちらを使用するかを指定します。 0: ラッチ機能 1 1: ラッチ機能 2	LatchID （ラッチ ID）
_eMC_CMDPOS_MODE	0: _mcCount	0: フィードバック現在位置を使用し、指令現在位置を更新します。原点確定状態が継続します。	CmdPosMode （指令現在位置カウント選択）
_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	動作の経路を指定します。 0: トランジション無効 10: コーナースーパーインポーズ	TransitionMode （トランジションモード（遷移モード））
_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	トリガのモードを指定します。 0: ドライバモード 1: コントローラモード	Mode （モード）
_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	ドライバモードのときのトリガ信号を指定します。 0: Z 相 1: 外部入力	InputDrive （ドライブトリガ入力信号）

データ型	有効範囲	内容	適応する命令の変数（変数の名称）
_eMC_HOMING_MODE *6	0: _mcHomeSwTurnHomeSwOff 1: _mcHomeSwTurnHomeSwOn 4: _mcHomeSwOff 5: _mcHomeSwOn 8: _mcLimitInputOff 9: _mcHomeSwTurnHomeMask 11: _mcLimitInputOnly 12: _mcHomeSwTurnHoldingTime 13: _mcNoHomeSwHoldingHomeInput 14: _mcHomePreset	書き換える［原点復帰動作］を指定します。 0: 近傍退避･原点近傍入力 OFF 指定 1: 近傍退避･原点近傍入力 ON 指定 4: 原点近傍入力 OFF 指定 5: 原点近傍入力 ON 指定 8: 限界入力 OFF 指定 9: 近傍退避･原点入力マスク距離指定 11: 限界入力のみ 12: 近傍退避･押し当て時間指定 13: 原点近傍入力なし･押し当て原点入力指定 14: 原点プリセット	－
_eMC_HOME_INPUT *6	0: _mcZPhase 1: _mcExternalSignal	原点入力信号を選択します。 0: Z 相入力を使用する。 1: 外部原点入力を使用する。	－
_eMC_LIMIT_REVERSE_MODE *6	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDecelerationStop	原点復帰中に限界入力が「ON」した場合の停止方法を設定します。 0: 反転しない・軸の異常で停止（停止方法は限界入力停止方法に従う） 1: 反転する・即停止 2: 反転する・減速停止	－
_eMC_CAM_CURVE *7	0: _mcConstantLine 1: _mcStraightLine 2: _mcParabolic 3: _mcModifiedConstantVel 4: _mcModifiedTrapezoid 5: _mcModifiedSine 6: _mcCycloidal 7: _mcTrapecloid 8: _mcReverseTrapecloid 9: _mcSimpleHarmonic 10: _mcDoubleHarmonic 11: _mcReverseDoubleHarmonic 12: _mcNC2Curve 13: _mcPolynomic3 14: _mcPolynomic5	ノードポイントまでのカム曲線の形状を指定します。 0: 直線（変位維持） 1: 直線 2: 等加速度 3: 変形等速度 4: 変形台形 5: 変形正弦 6: サイクロイド 7: トラペクロイド 8: 逆トラペクロイド 9: 単弦 10: 複弦 11: 逆複弦 12: NC2 曲線 13: 3 次曲線 14: 5 次曲線	Curve（曲線形状）
_eMC_ACCDECOVER *7	0: _mcAccDecOverBuffer 1: _mcAccDecOverRapid 2: _mcAccDecOverErrorStop	軸の加減速制御で、目標位置への停止を優先した結果、減速オーバーとなり最大加減速度を超過してしまう場合の動作を指定します。 0: 加減速度を急激にする（ブレンディングをバッファードに切り替える）。*8 1: 加減速度を急激にする。 2: 異常として停止する。*9	－
_eMC_REVERSE_MODE *7	0: _mcReverseModeDecelerationStop 1: _mcReverseModeImmediateStop	命令多重起動時、命令再起動時、割り込み定寸位置決め時の反転時動作を指定します。 0: 減速停止 1: 即停止	－

データ型	有効範囲	内容	適応する命令の変数（変数の名称）
_eMC_COUNT_MODE *7	0: _mcCountModeLinear 1: _mcCountModeRotary	位置のカウントモードを設定します。 0: リニアモード（有限長） 1: ロータリモード（無限長）	−
_eMC_UNITS *7	0: _mcPls 1: _mcMm 2: _mcUm 3: _mcNm 4: _mcDeg 5: _mcInch	指令位置の単位を設定します。 0: pulse 1: mm 2: μm 3: nm 4: degree 5: inch	−
_eMC_CAM_MONITOR_MODE *10	0: _mcCalcCamDistanceDiff	モニタするカム動作の情報を指定します。 0: 変位偏差計算	−

*1. トルク制御時にだけ有効です。
*2. ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせで指定可能です。
*3. ユニットバージョン Ver.1.10 以降の CPU ユニットと Sysmac Studio Ver.1.12 以降の組み合わせで指定可能です。
*4. _mcLatestCommand を使用する場合は、次の関係となるように主軸、従軸を使用しなければなりません。
Master（主軸）に設定されているモーション制御システム変数の軸番号 < Slave（従軸）に設定されているモーション制御システム変数の軸番号
*5. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。
*6. ユニットバージョン Ver.1.03 以降の CPU ユニットと Sysmac Studio Ver.1.04 以降の組み合わせで指定可能です。
*7. ユニットバージョン Ver.1.08 以降の CPU ユニットと Sysmac Studio Ver.1.09 以降の組み合わせで指定可能です。
*8. ユニットバージョン Ver.1.10 以降の CPU ユニットは、ブレンディングをバッファードに切り替えません。詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。
*9. ユニットバージョン Ver.1.10 以降の CPU ユニットでは、ブレンディング動作の場合、異常として停止しません。詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。
*10. 形 NX102 CPU ユニットでは、ユニットバージョン Ver.1.32 以降の CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。
形 NX1P2 CPU ユニット、NJ シリーズ CPU ユニットでは、ユニットバージョン Ver.1.21 以降の CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。

2-1-5 モーション制御命令の出力変数

モーション制御命令の出力変数の一覧は以下のようになります。

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE になります。 このとき、出力変数「Active」、「Error」、「CommandAborted」は FALSE です。 命令の完了時に入力変数「Execute」が FALSE の場合、「Done」は少なくとも 1 周期の間 TRUE です。 「Execute」が TRUE の場合は、「Done」は「Execute」が FALSE になるまで TRUE を保持します。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE になります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE になります。 制御中とは、この命令が実際に軸や軸グループを制御しているときです。このとき、出力変数「Done」、「Error」、「CommandAborted」は FALSE となります。
Enabled	有効	BOOL	TRUE, FALSE	制御中に TRUE になります。

出力変数	名称	データ型	有効範囲	内容
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が起動できなかったとき、または実行中に中断されたときに TRUE になります。 対象の軸または軸グループで異常が発生しているときには命令を起動することはできません。同様に減速停止中のときも命令を起動することはできません。 ほかの命令が起動されたとき、または自命令以外の異常が発生したときに、命令は中断されます。 このとき、出力変数「Done」、「Active」、「Error」は FALSE になります。入力変数「Execute」が FALSE のときに中断された場合、「CommandAborted」は少なくとも 1 周期の間 TRUE になります。 「Execute」、または「Enable」が TRUE の場合なら、「Execute」、または「Enable」が FALSE になるまで「CommandAborted」は TRUE を保持します。
Error	エラー	BOOL	TRUE, FALSE	入力変数の誤り、または命令処理による異常が発生したときに TRUE になります。*1
ErrorID	エラーコード	WORD	*2	異常が発生したときに、エラーコードを出力します。16#0000 は正常を表します。
Failure	不正終了	BOOL	TRUE, FALSE	命令が正常に実行されなかったときに TRUE になります。
Status	運転可	BOOL	TRUE, FALSE	運転可能状態になったときに TRUE になります。
EndOfProfile	カムサイクル完了	BOOL	TRUE, FALSE	カムテーブルの終点が実行されたときに TRUE になります。
Index	インデックス	UINT	正数または「0」	カムデータのインデックス番号を出力します。
StartSync	追従中	BOOL	TRUE, FALSE	同期のための加減速動作を開始したときに TRUE になります。
RecordedPosition	ラッチ位置	LREAL	負数、正数、「0」	ラッチした位置を出力します。*3
Invalid	軸間偏差オーバー	BOOL	TRUE, FALSE	軸間偏差許容値を超えたときに TRUE になります。
DeviatedValue	軸間偏差値	LREAL	負数、正数、「0」	指定された主軸と従軸の差分を出力します。*3
EndPointIndex	終点インデックス	UINT	正数または「0」	カムテーブルの終点インデックスを出力します。
MaxDataNumber	最大カムデータ数	UINT	正数	最大カムデータ数を出力します。
InVelocity	目標速度到達	BOOL	TRUE, FALSE	目標速度に到達したときに TRUE になります。
InSync	同期中	BOOL	TRUE, FALSE	主軸に同期したとき、または従軸が従軸同期位置に到達したときに TRUE になります。
InGear	ギア比到達	BOOL	TRUE, FALSE	従軸が、目標速度に到達したときに TRUE になります。
InCombination	加減算中	BOOL	TRUE, FALSE	加減算中に TRUE になります。
InCam	カム動作中	BOOL	TRUE, FALSE	カムテーブルの始点が実行されたときに TRUE になります。
InTorque	目標トルク到達	BOOL	TRUE, FALSE	目標トルクに到達したときに TRUE になります。
InFeed	定寸送り中	BOOL	TRUE, FALSE	ラッチ入力受付後、定寸送り中に TRUE になります。
InZone	ゾーン内	BOOL	TRUE, FALSE	軸位置がゾーン範囲内のとき TRUE になります。
Valid *4	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。

出力変数	名称	データ型	有効範囲	内容
CommandPosition *4	指令現在位置	ARRAY [0..3] OF LREAL	負数、正数、「0」	指令位置の現在値を出力します。*3
ActualPosition *4	フィードバック現在位置	ARRAY [0..3] OF LREAL	負数、正数、「0」	フィードバック位置の現在値を出力します。*3
InPosition *4	インポジション	BOOL	TRUE, FALSE	すべての構成軸のフィードバック現在位置が目標位置のインポジション幅内に入ったとき TRUE となります。
InOperation *5	作動中	BOOL	TRUE, FALSE	命令の作動中に TRUE となります。
CalcPosition *5	算出位置	LREAL	負数、正数、「0」	指定したタイムスタンプの位置を出力します。*3
ErrorParameterCode *6	パラメータ詳細コード	WORD	*2	エラーコードによっては、付属情報が存在します。存在する場合は、異常となっているパラメータの詳細コードを出力します。
ErrorNodePointIndex *6	ノードポイント要素番号	UINT	*2	エラーコードによっては、付属情報が存在します。存在する場合は、異常となっているノードポイントの要素番号を出力します。
OutputtedOffsetPosition *7	位置オフセット出力値	LREAL	負数、正数、「0」	指令現在位置に加算された位置オフセットを出力します。 「Active」が TRUE のときに値を更新します。 「CommandAborted」または「Error」が TRUE のときは、更新を停止し値を保持します。

*1. 「Error」（エラー）は、MC_Reset（軸エラーリセット）命令、MC_GroupReset（軸グループエラーリセット）命令、ResetMCError（全エラーリセット）命令を実行するまでゼロクリアされません。
この仕様は PLCopen®とは異なります。PLCopen®では、「Execute」（起動）の立ち下がりエッジで、FALSE となります。
また、「Error」が TRUE の間は、モーション制御命令は起動しません。異常を解除した時に、「Execute」が TRUE でも、立ち上がりエッジではないため、命令は起動しません。Enable 型のモーション制御命令は「Enable」（有効）が TRUE であれば起動します。
*2. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。
*3. 指令単位となります。指令単位は［mm］、［μm］、［nm］、［degree］、［inch］または［pulse］で表されます。
*4. ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせで使用可能です。
*5. ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで使用可能です。
*6. ユニットバージョン Ver.1.08 以降の CPU ユニットと Sysmac Studio Ver.1.09 以降の組み合わせで指定可能です。
*7. ユニットバージョン Ver.1.10 以降の CPU ユニットと Sysmac Studio Ver.1.12 以降の組み合わせで使用可能です。

参考

運転モードの変更後もモーション制御命令の出力変数を参照する場合は、出力パラメータの属性が保持属性を持つ変数を割り当ててください。
割り当てた出力パラメータを参照することにより、運転モードを変更する直前の出力変数を参照することが可能となります。

2-1-6 モーション制御命令の入出力変数

モーション制御命令の入出力変数一覧は以下のようになります。

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。
AxesGroup	軸グループ	_sGROUP_REF	−	軸グループを指定します。
Auxiliary	補助軸	_sAXIS_REF	−	補助軸を指定します。
Master	主軸	_sAXIS_REF	−	主軸を指定します。
Slave	従軸	_sAXIS_REF	−	従軸を指定します。

入出力変数	名称	データ型	有効範囲	内容
CamTable	カムテーブル	ARRAY[0..N] OF _sMC_CAM_REF	－	カムデータ構造体_sMC_CAM_REF 型の配列変数をカムテーブルとして指定します。*1
TriggerInput	トリガ入力条件	_sTRIGGER_REF	－	トリガ条件を設定します。
TriggerVariable	トリガ変数	BOOL	TRUE, FALSE	トリガ条件でコントローラモードを指定したとき、トリガとなる入力変数を指定します。
Target	書込対象	_sAXIS_REF、または _sGROUP_REF	－	書き込み対象となる、軸または、軸グループを指定します。
SettingValue	設定値	指定した変数の型によります。	－	書き込む値を指定します。 有効範囲は ParameterNumber（パラメータ番号）で指定したモーション制御パラメータ指定に従います。*2 初期値は「0」です。
Axes *3	軸グループ構成軸	ARRAY [0..3] OF UINT	－	書き換え後の構成軸の軸番号を指定します。
HomingParameter *4	原点復帰パラメータ	_sHOMING_REF	－	原点復帰パラメータを設定します。
Switches *5	スイッチ	ARRAY[0..255] OF _sCAMSWITCH_REF	－	スイッチ構造体_sCAMSWITCH_REF 型の配列変数を、スイッチの ON/OFF パターンデータとして指定します。 配列要素番号は、スイッチ番号を示します。
Outputs *5	出力信号	ARRAY[0..15] OF _sOUTPUT_REF	－	スイッチの ON/OFF パターンデータをもとに算出された、デジタル出力の ON/OFF 時刻の出力先として、出力信号構造体 _sOUTPUT_REF 型の配列変数を指定します。 配列要素番号は、トラック番号を示します。 この配列変数を NX_AryDOutTimeStamp 命令の入出力変数に指定することにより、実際のデジタル出力を ON/OFF します。
TrackOptions *5	トラックオプション	ARRAY[0..15] OF _sTRACK_REF	－	トラックオプション構造体_sTRACK_REF 型の配列変数を、スイッチの動作条件として指定します。配列要素番号は、トラック番号を示します。
CamProperty *6	カムプロパティ	_sMC_CAM_PROPERTY	－	カムプロパティ構造体 _sMC_CAM_PROPERTY 型の変数を指定します。 _sMC_CAM_PROPERTY 型のユーザ定義変数、または Sysmac studio のカムエディタで作成したカムプロパティ変数を指定してください。
CamNodes *6	カムノード	ARRAY[0..N] OF _sMC_CAM_NODE	－	カムノード構造体_sMC_CAM_NODE 型の配列変数を指定します。 _sMC_CAM_NODE 型のユーザ定義変数*7、または Sysmac studio のカムエディタで作成したカムノード変数を指定してください。
AxisParameter *6	軸パラメータ	_sAXIS_PARAM	－	書き換え時は、軸パラメータを指定します。 読み出し時は、読み出した軸パラメータの格納先として_sAXIS_PARAM 型の変数を指定します。
CamMonitorValue *8	カムモニタ値	_sMC_CAM_MONITOR_DISTANCEDIFF	－	カム動作の情報を出力します。*9

*1. 配列要素「N」は Sysmac Studio により自動的に設定されます。Sysmac Studio のカムエディタで作成したカムデータ変数を指定してください。

*2. 変数のデータ型については、「パラメータ番号のデータ型と有効範囲 (P.5-19)」を参照してください。
*3. ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせで指定可能です。
*4. ユニットバージョン Ver.1.03 以降の CPU ユニットと Sysmac Studio Ver.1.04 以降の組み合わせで指定可能です。
*5. ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで指定可能です。
*6. ユニットバージョン Ver.1.08 以降の CPU ユニットと Sysmac Studio Ver.1.09 以降の組み合わせで指定可能です。
*7. ユーザ定義変数の場合、配列変数の要素の開始番号が 0, 配列要素「N」が 358 以下となるよう作成してください。
*8. 形 NX102 CPU ユニットでは、ユニットバージョン Ver.1.32 以降の CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。
形 NX1P2 CPU ユニット、NJ シリーズ CPU ユニットでは、ユニットバージョン Ver.1.21 以降の CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。
*9. モニタするカム動作の情報は、「CamMonitorMode」（カムモニタモード選択）で指定します。

2-2 命令一覧

モーション制御命令は以下の 3 つの種類があります。

種類	概要
共通指令	MC 機能モジュール共通の命令
軸指令	MC 機能モジュールが単軸制御を行うための命令
軸グループ指令	MC 機能モジュールが多軸協調制御を行うための命令

共通指令の詳細については、「第 5 章 共通指令の命令 (P.5-1)」を、軸指令の詳細については、「第 3 章 軸指令の命令 (P.3-1)」を、軸グループの詳細については、「第 4 章 軸グループ指令の命令 (P.4-1)」を参照してください。

モーション制御命令を NX シリーズ 位置インタフェースユニットに対して使用する場合、機能に制約のある命令や使用できない命令があります。詳細は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。

2-2-1 共通指令

MC 機能モジュール共通の命令について説明します。
“命令種別”欄は、“管理”が非動作系の命令を、“動作”が動作系の命令を表しています。
また、記号は、以下の内容を表しています。

P ：PLCopen®技術仕様で定義されている命令
O ：MC 機能モジュール独自の命令

命令	命令名称	概要	命令種別		単軸 *1
MC_SetCamTableProperty	カムテーブルプロパティ更新	入力パラメータで指定したカムテーブルの終点インデックスを更新します。	管理	O	
MC_SetCamTableEndPointIndex*2	カムテーブル終点インデックス更新	入力パラメータで指定したカムテーブルの終点インデックスを更新します。 カムテーブルに対する操作と得られる結果は、MC_SetCamTableProperty（カムテーブルプロパティ更新）命令と同じですが、命令起動から完了まで必要なタスク周期を調節できます。	管理	O	
MC_SaveCamTable	カムテーブル保存	入力パラメータで指定したカムテーブルを不揮発性メモリに保存します。	管理	O	
MC_Write	MC 設定書込	モーション制御パラメータ設定の一部を書き換えます。	管理	O	O
MC_GenerateCamTable *3	カムテーブル生成	入出力変数で指定したカムプロパティとカムノードをもとにカムテーブルを生成します。	管理	O	
MC_WriteAxisParameter *3	軸パラメータ書込	モーション制御パラメータ設定の軸パラメータ設定を書き換えます。	管理	O	O
MC_ReadAxisParameter *3	軸パラメータ読出	モーション制御パラメータ設定の軸パラメータ設定を読み出します。	管理	O	O

*1. 単軸位置制御軸で使用可能な命令にO印を付けています。
*2. 形 NX701-1□□□では、ユニットバージョン Ver.1.36 以降の CPU ユニットと Sysmac Studio Ver.1.63 以降の組み合わせで使用可能です。
形 NX701-1□□□以外では、ユニットバージョン Ver.1.69 以降の CPU ユニットと Sysmac Studio Ver 1.63 以降の組み合わせで使用可能です。

*3. ユニットバージョン Ver.1.08 以降の CPU ユニットと Sysmac Studio Ver.1.09 以降の組み合わせで使用可能です。

命令起動による軸状態の遷移については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

2-2-2 軸指令

MC 機能モジュールが単軸制御を行うための命令について説明します。
“命令種別”欄は、“管理”が非動作系の命令を、“動作”が動作系の命令を表しています。
また、記号は、以下の内容を表しています。

P ：PLCopen®技術仕様で定義されている命令
O ：MC 機能モジュール独自の命令

命令	命令名称	概要	命令種別		単軸 *1
MC_Power	運転可	ドライバを運転可能状態に切り替えます。	管理	P	O
MC_MoveJog	ジョグ送り	指定した目標速度にしたがって、ジョグ送りを行います。	動作	O	O
MC_Home	原点復帰	モータを動かして、限界信号、原点近傍信号、原点信号を用いて機械原点を決めます。	動作	P	O
MC_HomeWithParameter *2	パラメータ指定原点復帰	原点復帰パラメータを指定し、モータを動かして、限界信号、原点近傍信号、原点信号を用いて機械原点を決めます。	動作	O	O
MC_Move	位置決め	絶対値位置決め、または相対値位置決めを行います。	動作	O	O
MC_MoveAbsolute	絶対値位置決め	絶対座標の目標位置を指定して、位置決めを行います。	動作	P	O
MC_MoveRelative	相対値位置決め	指令現在位置からの移動距離を指定して、位置決めを行います。	動作	P	O
MC_MoveVelocity	速度制御	サーボドライバの位置制御モードを用いた擬似的な速度制御を行います。	動作	P	O
MC_MoveZeroPosition	高速原点復帰	絶対座標の「0」を目標位置として位置決めを行い、原点へ戻ります。	動作	O	O
MC_MoveFeed	割り込み定寸位置決め	外部入力による割り込み入力が発生した位置からの移動距離を指定し、位置決めを行います。	動作	O	O
MC_Stop	強制停止	軸を減速停止させます。	動作	P	O
MC_ImmediateStop	即停止	軸の状態によらず入力変数「StopMode」（停止方法選択）の停止方法に従って停止します。	動作	O	O
MC_SetPosition	現在位置変更	軸の指令現在位置とフィードバック現在位置を任意の値に変更します。	管理	P	O
MC_SetOverride	オーバーライド値設定	軸の目標速度を変更します。	管理	P	O
MC_ResetFollowingError	偏差カウンタリセット	指令位置とフィードバック位置の偏差をリセットします。	動作	O	O
MC_CamIn	カム動作開始	指定したカムテーブルを用いたカム動作を開始します。	動作	P	
MC_CamOut	カム動作解除	入力パラメータで指定した軸のカム動作を終了します。	動作	P	
MC_CamMonitor *3	カムモニタ	カム動作に関する情報のモニタを行います。	動作	O	
MC_GearIn	ギア動作開始	主軸と従軸間のギア比を設定し、ギア動作を行います。	動作	P	
MC_GearInPos	位置指定ギア動作	主軸と従軸間のギア比を設定し、電子ギア動作を行います。同期を開始する主軸および従軸の位置を指定します。	動作	P	

命令	命令名称	概要	命令種別		単軸 *1
MC_GearOut	ギア動作解除	実行中の MC_GearIn、MC_GearInPos を中止します。	動作	P	
MC_MoveLink	台形パターンカム	指定した主軸に同期して位置決めします。	動作	O	
MC_CombineAxes	加減算位置決め	2 軸の指令位置を加算した値、または減算した値を指令位置として出力します。	動作	O	
MC_Phasing	主軸相対値位相補正	同期制御中の主軸の位相補正を行います。	動作	P	
MC_TorqueControl	トルク制御	サーボドライバのトルク制御モードを用いてトルクを制御します。	動作	P	
MC_SetTorqueLimit	トルク制限	サーボドライバのトルク制限機能を使用し、サーボドライバの出力トルクを制限します。	管理	O	○
MC_ZoneSwitch	ゾーン監視	軸の指令位置、もしくはフィードバック現在位置が指定された範囲（ゾーン）内に存在するかを判定します。	管理	O	○
MC_TouchProbe	外部ラッチ有効	トリガ信号の発生により、軸の位置を記録します。	管理	P	○
MC_AbortTrigger	外部ラッチ無効	実行中のラッチを無効にします。	管理	P	○
MC_AxesObserve	軸間偏差監視	指定された軸の指令位置、またはフィードバック位置の差分が許容値を超えることがないかを監視します。	管理	O	○
MC_SyncMoveVelocity	サイクリック同期速度制御	目標速度に設定した値を、タスク周期 *4 ごとにサイクリック同期速度モード(CSV)でサーボドライバへ出力します。	動作	O	
MC_SyncMoveAbsolute *2	サイクリック同期絶対位置決め	軸に指定した目標位置を周期的に出力します。	動作	O	○
MC_Reset	軸エラーリセット	軸の異常を解除します。	管理	P	○
MC_ChangeAxisUse *5	軸使用変更	軸パラメータの［軸使用］を一時的に切り替えます。	管理	P	○
MC_DigitalCamSwitch *6	デジタルカムスイッチ有効	軸の位置に応じてデジタル出力を ON または OFF します。	管理	P	○
MC_TimeStampToPos *6	タイムスタンプ→軸位置算出	指定したタイムスタンプの軸の位置を算出します。	管理	O	○
MC_PeriodicSyncVariables *7	軸変数タスク間定周期同期	軸変数のタスク間同期を定周期に実行します。	管理	O	
MC_SyncOffsetPosition*8	サイクリック同期位置オフセット補正	同期制御中の従軸に対して、指定した位置オフセットを周期的に指令現在位置に加算して出力します。	動作	O	
MC_OffsetPosition *3	位置オフセット補正	同期制御中の従軸に対して、指定した位置オフセットを指令現在位置に加減速カーブをつけて加算して出力します。	動作	O	

*1. 単軸位置制御軸で使用可能な命令に○印を付けています。
*2. ユニットバージョン Ver.1.03 以降の CPU ユニットと Sysmac Studio Ver.1.04 以降の組み合わせで使用可能です。
*3. 形 NX102 CPU ユニットでは、ユニットバージョン Ver.1.32 以降の CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。
形 NX1P2 CPU ユニット、NJ シリーズ CPU ユニットでは、ユニットバージョン Ver.1.21 以降の CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。
*4. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。
*5. ユニットバージョン Ver.1.04 以降の CPU ユニットと Sysmac Studio Ver.1.05 以降の組み合わせで使用可能です。

*6. ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで使用可能です。
*7. 形 NX502 CPU ユニット、形 NX102 CPU ユニット、形 NX1P2 CPU ユニットおよび NJ シリーズ CPU ユニットでは、使用できません。
*8. ユニットバージョン Ver.1.10 以降の CPU ユニットと Sysmac Studio Ver.1.12 以降の組み合わせで使用可能です。

命令起動による軸状態の遷移については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

PLCopen®技術仕様への準拠内容については、コンプライアンスリストを参照してください。
コンプライアンスリストは、PLCopen®の Web で入手可能です。

2-2-3 軸グループ指令

MC 機能モジュールが多軸協調制御を行うための命令について説明します。
“命令種別”欄は、“グループ管理”が非動作系の命令を、“グループ動作”が動作系の命令を表しています。
また、記号は、以下の内容を表しています。

P ：PLCopen®技術仕様で定義されている命令
O ：MC 機能モジュール独自の命令

命令	命令名称	概要	命令種別	
MC_GroupEnable	軸グループ有効	軸グループを有効にします。	グループ管理	P
MC_GroupDisable	軸グループ無効	軸グループを無効にします。	グループ管理	P
MC_MoveLinear	直線補間	直線補間を行います。	グループ動作	O
MC_MoveLinearAbsolute	絶対値直線補間	絶対位置を指定して直線補間を行います。	グループ動作	P
MC_MoveLinearRelative	相対値直線補間	相対位置を指定して直線補間を行います。	グループ動作	P
MC_MoveCircular2D	2 軸円弧補間	2 軸の円弧補間を行います。	グループ動作	O
MC_GroupStop	軸グループ強制停止	補間動作中のすべての軸を減速停止します。	グループ動作	P
MC_GroupImmediateStop	軸グループ即停止	軸パラメータで選択した方法で、補間動作中のすべての軸が即停止します。	グループ動作	O
MC_GroupSetOverride	軸グループオーバーライド値設定	補間動作中の合成目標速度を変更します。	グループ管理	P
MC_GroupReadPosition *1	軸グループ位置取得	軸グループの指令現在位置とフィードバック現在位置を取得します。	グループ管理	O
MC_ChangeAxesInGroup *1	軸グループ構成軸書込	軸グループパラメータの［構成軸］を一時的に書き換えます。	グループ管理	O
MC_GroupSyncMoveAbsolute *1	軸グループサイクリック同期絶対位置制御	各軸に指定した目標位置を周期的に出力します。	グループ動作	O
MC_GroupReset	軸グループエラーリセット	軸グループ、および軸の異常を解除します。	グループ管理	P

*1. ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせで使用可能です。

命令起動による軸グループ状態の遷移については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

PLCopen®技術仕様への準拠内容については、コンプライアンスリストを参照してください。
コンプライアンスリストは、PLCopen®の Web で入手可能です。

2-3 PDO マッピング

モーション制御の各機能を使用するためには、使用する機能に必要なオブジェクトをプロセスデータ通信にマッピングする必要があります。
あらかじめ必要なオブジェクトをまとめたものが PDO マップです。

オムロン製サーボドライバ 1S シリーズ 形 R88D-1SN□□□-ECT、形 R88D-1SAN□□□-ECT、および G5 シリーズ 形 R88D-KN□□□-ECT（Ver.2.1 以降）、形 R88D-KN□□□-ECT-L（Ver.1.1 以降）の場合、PDO マップは、Sysmac Studio のデフォルトのままでも使用することができます。

RxPDO: ［261th 受信 PDO マッピング］(1704Hex)	コントロールワード (6040Hex)、目標位置 (607AHex)、 目標速度 (60FFHex)、目標トルク (6071Hex)、オペレーションモード (6060Hex)、ラッチ機能 (60B8Hex)、最大プロファイル速度 (607FHex)、正方向トルク制限値 (60E0Hex)、負方向トルク制限値 (60E1Hex)
TxPDO: ［259th 送信 PDO マッピング］(1B02Hex)	エラーコード (603FHex)、ステータスワード (6041Hex)、現在位置 (6064Hex)、現在トルク (6077Hex)、オペレーションモード表示 (6061Hex)、ラッチステータス (60B9Hex)、ラッチ 1 位置(60BAHex)、ラッチ 2 位置(60BCHex)、デジタル入力 (60FDHex)

参考

オムロン製サーボドライバ G5 シリーズ 形 R88D-KN□□□-ECT でフルクローズ制御を行う場合、RxPDO には 1701Hex、またはオブジェクトの合計サイズを 12 バイト以下とした 1600Hex（Ver.2.1 以降の場合）を選択してください。

PDO マッピングの設定方法については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

NX シリーズ 位置インタフェースユニットを使用する場合は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』の「I/O エントリマッピング」を参照してください。

2-3-1 必須オブジェクト

必須オブジェクトには、軸種別がサーボ軸に対するオブジェクトとエンコーダ軸に対するオブジェクトがあります。
必須オブジェクトを 1 つでも設定していない場合は、必須プロセスデータオブジェクト未設定（エラーコード: 3460Hex）が発生します。

サーボ軸

MC 機能モジュールの命令をサーボ軸に対して使用するためには、下記オブジェクトの設定が必要となります。

入出力	機能名	プロセスデータ
出力	コントロールワード	6040Hex
	目標位置	607AHex
入力	ステータスワード	6041Hex
	現在位置	6064Hex

バージョン関連情報

- ユニットバージョン Ver.1.10 以降の CPU ユニットの場合、オペレーションモード(6060Hex)およびオペレーションモード表示(6061Hex)のマッピング有無により、次の動作となります。

	オペレーションモード表示 (6061Hex)あり	オペレーションモード表示 (6061Hex)なし
オペレーションモード (6060Hex)あり	• CSP *1、CSV *2、CST *3 を使用する命令が実行可能です。 • CSP、CSV、CST 以外のときは、サーボ OFF となります。	• CSP を使用する命令は実行可能です。その他の制御モードを使用する命令は、実行時にプロセスデータオブジェクト設定不足（エラーコード: 3461Hex）」が発生します。 • MC 機能モジュールは、サーボドライバの制御モードを CSP とみなします。サーボドライバに CSP を指令します。
オペレーションモード (6060Hex)なし	• CSP を使用する命令は実行可能です。その他の制御モードを使用する命令は、実行時にプロセスデータオブジェクト設定不足（エラーコード: 3461Hex）」が発生します。 • CSP 以外のときは、サーボ OFF となります。	• CSP を使用する命令は実行可能です。その他の制御モードを使用する命令は、実行時にプロセスデータオブジェクト設定不足（エラーコード: 3461Hex）」が発生します。

*1. サーボドライバの制御モードのうち、サイクリック同期位置モード(CSP)のことです。
*2. サーボドライバの制御モードのうち、サイクリック同期速度モード(CSV)のことです。
*3. サーボドライバの制御モードのうち、サイクリック同期トルクモード(CST)のことです。

- ユニットバージョン Ver.1.09 以前の CPU ユニットで、オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプ、および G5 シリーズ EtherCAT 通信内蔵タイプ以外を使用する場合は、 オペレーションモード(6060Hex)とオペレーションモード表示(6061Hex)が必要です。

エンコーダ軸

MC 機能モジュールの命令をエンコーダ軸に対して使用するためには、下記オブジェクトの設定が必要となります。

入出力	機能名	プロセスデータ
入力	フィードバック位置	4010Hex

2-3-2 命令別による設定オブジェクト

命令によっては、設定が必要なオブジェクトがあります。
サーボ軸、エンコーダ軸のそれぞれに設定が必要となります。
なお、各命令に必要なオブジェクトが未設定の場合は、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。

サーボ軸

サーボ軸として使用するには、命令によって設定が必要なオブジェクトがあります。
下記、一覧を参照の上、必要なオブジェクトを設定してください。

なお、下記一覧表に掲載されていない命令については、必須オブジェクトだけの設定でサーボ軸の動作が可能です。

● 出力の設定

命令名	機能名						
	目標速度 60FFHex	目標トルク 6071Hex	オペレーションモード 6060Hex *1	ラッチ機能 60B8Hex	最大プロファイル速度 607FHex	正方向トルク制限値 60E0Hex	負方向トルク制限値 60E1Hex
MC_Home				△ *2			
MC_HomeWithParameter				△ *2			
MC_MoveFeed				△ *3			
MC_MoveLink				△ *4			
MC_TorqueControl		○	○		△ *5		
MC_SetTorqueLimit						○	○
MC_TouchProbe				△ *2			
MC_SyncMoveVelocity	○		○				

*1. オペレーションモード (6060Hex)を設定する場合は、オペレーションモード表示 (6061Hex)も設定してください。どちらか 1 つだけを設定すると正常に動作しません。
*2. 原点復帰動作モード「11」、「12」、「14」を除き、設定が必要です。
*3. 「Mode」に［ドライバモード］を選択したとき。
*4. 「LinkOption」（同期開始条件選択）に［_mcTriggerDetection］を選択し、「Mode」に［ドライバモード］を選択したとき。
*5. オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプ、または G5 シリーズ EtherCAT 通信内蔵タイプを使用しているときだけチェックします。

● 入力の設定

命令名	機能名				
	現在トルク 6077Hex	オペレーションモード表示 6061Hex *1	ラッチステータス 60B9Hex	ラッチ 1 位置 60BAHex	ラッチ 2 位置 60BCHex
MC_Home			△ *2	△ *2	
MC_HomeWithParameter			△ *2	△ *2	
MC_MoveFeed			△ *3	△ *4	△ *5
MC_MoveLink			△ *6	△ *7	△ *8
MC_TorqueControl	○	○			
MC_TouchProbe			△ *3	△ *4	△ *5
MC_SyncMoveVelocity		○			

*1. オペレーションモード表示 (6061Hex)を設定する場合は、オペレーションモード (6060Hex)も設定してください。どちらか 1 つだけを設定すると正常に動作しません。
*2. 原点復帰動作モード「11」、「12」、「14」を除き、設定が必要です。
*3. 「Mode」に［ドライバモード］を選択したとき。
*4. 「Mode」に［ドライバモード］を選択し、「LatchID」（ラッチ ID 選択）に［_mcLatch1］（ラッチ機能 1）を選択したとき。
*5. 「Mode」に［ドライバモード］を選択し、「LatchID」（ラッチ ID 選択）に［_mcLatch2］（ラッチ機能 2）を選択したとき。
*6. 「LinkOption」（同期開始条件選択）に［_mcTriggerDetection］を選択し、「Mode」に［ドライバモード］を選択したとき。
*7. 「LinkOption」（同期開始条件選択）に［_mcTriggerDetection］を選択し、「Mode」に［ドライバモード］を選択した場合で、「LatchID」（ラッチ ID 選択）に［_mcLatch1］（ラッチ機能 1）を選択したとき。
*8. 「LinkOption」（同期開始条件選択）に［_mcTriggerDetection］を選択し、「Mode」に［ドライバモード］を選択した場合で、「LatchID」（ラッチ ID 選択）に［_mcLatch2］（ラッチ機能 2）を選択したとき。

エンコーダ軸

エンコーダ軸として使用するには、命令によって設定が必要なオブジェクトがあります。
下記、一覧を参照の上、必要なオブジェクトを設定してください。
なお、下記一覧表に掲載されていない命令については、必須オブジェクトだけの設定でエンコーダ軸の動作が可能です。

● 出力の設定

命令名	機能名	
	ラッチ機能 4020Hex	エンコーダ入力スレーブソフトスイッチ 4020Hex
MC_TouchProbe	△ *1	△ *2

*1. 「Mode」に［ドライバモード］を選択したとき。
*2. オムロン製 GX シリーズ エンコーダスレーブを使用し、「Mode」に［ドライバモード］を選択したとき。

● 入力の設定

命令名	機能名			
	ラッチステータス 4030Hex	ラッチ位置 1 4012Hex	ラッチ位置 2 4013Hex	エンコーダ入力スレーブステータス 4030Hex
MC_TouchProbe	△ *1	△ *2	△ *3	△ *4

*1. 「Mode」に［ドライバモード］を選択したとき。
*2. 「Mode」に［ドライバモード］を選択し、「LatchID」（ラッチ ID 選択）に［_mcLatch1］（ラッチ機能 1）を選択したとき。
*3. 「Mode」に［ドライバモード］を選択し、「LatchID」（ラッチ ID 選択）に［_mcLatch2］（ラッチ機能 2）を選択したとき。
*4. オムロン製 GX シリーズ エンコーダスレーブを使用し、「Mode」に［ドライバモード］を選択したとき。

3

軸指令の命令

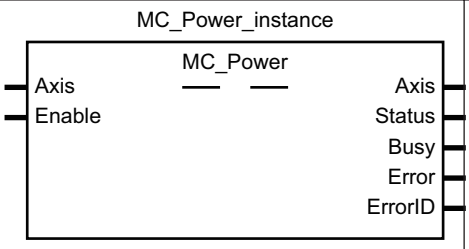
MC 機能モジュールが単軸制御を行うための命令について説明します。

MC_Power 3-3
MC_MoveJog 3-8
MC_Home 3-16
MC_HomeWithParameter 3-37
MC_Move 3-42
MC_MoveAbsolute 3-47
MC_MoveRelative 3-73
MC_MoveVelocity 3-81
MC_MoveZeroPosition 3-95
MC_MoveFeed 3-102
MC_Stop 3-129
MC_ImmediateStop 3-138
MC_SetPosition 3-142
MC_SetOverride 3-148
MC_ResetFollowingError 3-153
MC_CamIn 3-160
MC_CamOut 3-216
MC_CamMonitor 3-221
MC_GearIn 3-230
MC_GearInPos 3-249
MC_GearOut 3-270
MC_MoveLink 3-275
MC_CombineAxes 3-297
MC_Phasing 3-308
MC_TorqueControl 3-315
MC_SetTorqueLimit 3-326

MC_ZoneSwitch 3-333
MC_TouchProbe 3-339
MC_AbortTrigger 3-360
MC_AxesObserve 3-364
MC_SyncMoveVelocity 3-370
MC_SyncMoveAbsolute 3-379
MC_Reset 3-386
MC_ChangeAxisUse 3-389
MC_DigitalCamSwitch 3-395
MC_TimeStampToPos 3-414
MC_PeriodicSyncVariables 3-425
MC_SyncOffsetPosition 3-433
MC_OffsetPosition 3-443

MC_Power

サーボドライバを運転可能状態に切り替えます。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_Power	運転可	FB		MC_Power_instance (Axis :=「パラメータ」, Enable :=「パラメータ」, Status =>「パラメータ」, Busy =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE にすると運転可能状態となり、FALSE にすると運転可能状態を解除します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Status	運転可	BOOL	TRUE, FALSE	運転可能状態になったときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Status	指定軸が運転可能状態になったとき。	• 指定軸の運転可能状態が解除となったとき。 •「Error」が TRUE となったとき。
Busy	「Enable」の立ち上がり	•「Enable」が FALSE となったとき。 •「Error」が TRUE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 「Enable」（有効）を TRUE にすると、「Axis」（軸）で指定した軸が運転可能状態となります。
軸を運転可能状態にすることにより、軸制御が可能になります。
- 「Enable」（有効）を FALSE にすると、「Axis」（軸）で指定した軸の運転可能状態を解除します。
運転可能状態を解除した場合、軸は動作指令を受け付けず、軸制御ができません。また、運転可能状態を解除した軸に対する動作指令は異常となります。ただし、解除状態でも MC_Power（運転可）命令、MC_Reset（軸エラーリセット）命令は実行可能です。
- 動作中の軸を解除状態に切り替えることも可能です。その際､動作中の命令を「CommandAborted」（実行中断）し、動作指令の出力を停止し解除状態に切り替えます。
- 絶対値エンコーダ付きのサーボモータを使用する場合、軸が運転可能状態に切り替わるときに原点未確定状態ならば［絶対値エンコーダ原点位置オフセット］の補正を行い、原点確定状態となります。
ユニットバージョン Ver.1.10 以降の CPU ユニットでは、EtherCAT のプロセスデータ通信が未確立状態から確立状態に遷移した場合も、原点確定状態となります。
［絶対値エンコーダ原点位置オフセット］については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

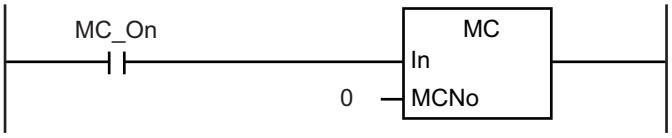
使用上の注意

- 本命令は、軸種別がサーボ軸、または仮想サーボ軸に使用可能です。エンコーダ軸、または仮想エンコーダ軸に使用した場合は、異常となります。
- NX シリーズ パルス出力ユニットに対して本命令を実行する場合、本命令の機能が異なります。詳細は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル (SBCE-374)』を参照してください。
- **同期制御の主軸に対して本命令を実行する場合**
垂直軸などでは、主軸を運転可能状態から解除した時点で、主軸の位置が急激に変化する場合があります。これにより、従軸の動作が急激に変化することもあります。
従軸が急激に動作しないよう主軸にブレーキをつけたり、同期制御が終了した後に主軸の運転状態を解除したりするようにしてください。

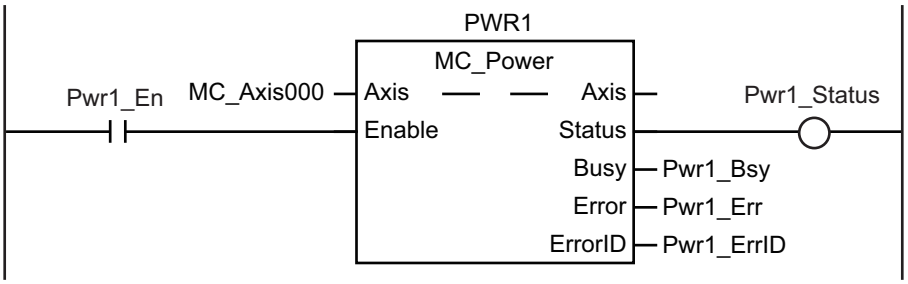
参考

Execute 型のモーション制御命令は、「Execute」入力に接続した接点の立ち上がりで動作を開始し、命令による制御が完了するまで動作を継続します。マスタコントロールによるリセット中であっても、リセット実行前に命令が実行中の場合は、命令による制御が完了するまで動作を継続します。
Execute 型のモーション制御命令をインタロックするには、下図のように MC_Power（運転可）命令をマスタコントロール領域に配置してください。「MC_On」を FALSE にすることにより確実にサーボ OFF することができます。

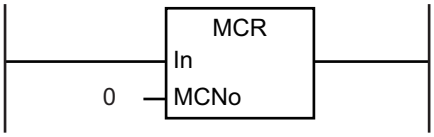
マスタコントロール開始



サーボ ON する



マスタコントロール終了



● CPU ユニットの動作モードとの関係

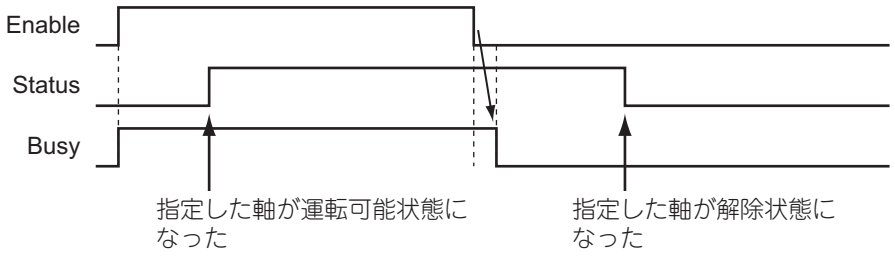
運転モード中に本命令により運転可能状態した後に、プログラムモードに切り替えた場合も運転状態が継続します。

● オンラインエディットによる命令削除時

本命令により運転可能状態にした後で、オンラインエディット中に本命令を削除しても運転可能状態が継続します。

タイミングチャート

- 「Enable」（有効）を TRUE にすると、命令受付中であることを示す「Busy」（実行中）が TRUE となります。
- その後、軸が運転可能状態になると、「Status」（運転可）が TRUE となります。
- 「Enable」（有効）を FALSE にすると、「Busy」（実行中）は FALSE となります。「Status」（運転可）は運転可能状態が解除されたときに FALSE となります。「Status」（運転可）は「Enable」（有効）の TRUE/FALSE にかかわらず軸の運転可能状態を出力します。



使用上の注意

- 「Enable」（有効）を TRUE にしてから、軸側の処理が終わらないと「Status」（運転可）が TRUE とはなりません。軸を運転する場合は、必ず「Status」（運転可）が TRUE になったことを確認してから動作を開始してください。
- コントローラの電源投入後に運転を開始する場合など、EtherCAT 通信が確立したことを確認した後に、モーション制御命令の起動を行うようユーザプログラムを作成してください。また、運転中に EtherCAT 通信に異常が発生していないかを監視するインターロック機能をユーザプログラム中に組み込むようにしてください。

モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

モーション命令多重起動

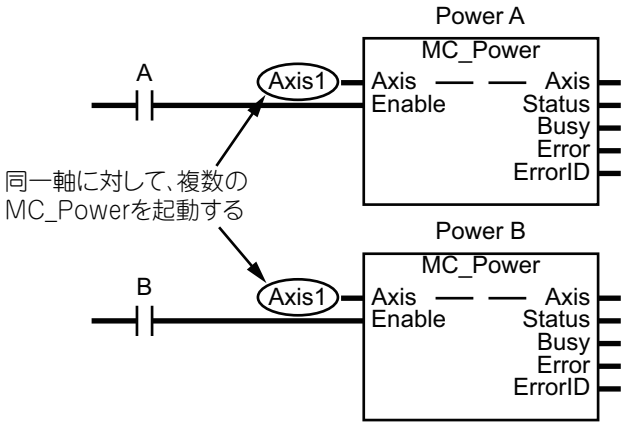
モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

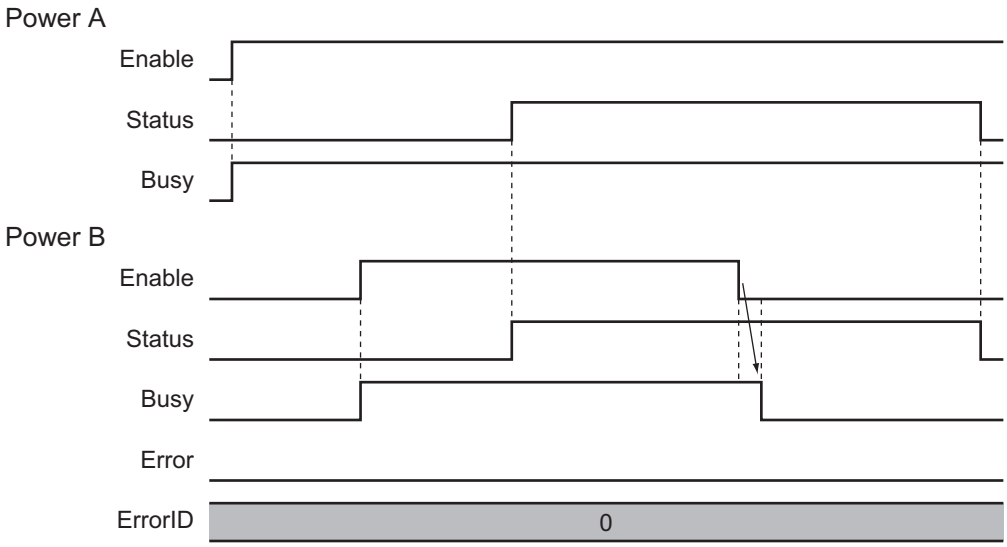
● 本命令同士の多重起動

使用上の注意

MC_Power（運転可）命令実行中の軸に、別インスタンスの MC_Power（運転可）命令を起動するプログラムは作成しないでください。基本的に 1 軸に対して MC_Power（運転可）命令は 1 つとしてください。

仮に MC_Power（運転可）命令実行中の軸に、別インスタンスの MC_Power（運転可）命令を起動した場合は、あとから実行した MC_Power（運転可）命令が優先して実行されます。





エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_MoveJog

指定した目標速度にしたがって、ジョグ送りを行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveJog	ジョグ送り	FB	MC_MoveJog_instance MC_MoveJog Axis — — Axis PositiveEnable — Busy NegativeEnable — CommandAborted Velocity — Error Acceleration — ErrorID Deceleration	MC_MoveJog_instance (Axis :=「パラメータ」, PositiveEnable :=「パラメータ」, NegativeEnable :=「パラメータ」 , Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
PositiveEnable	正方向有効	BOOL	TRUE, FALSE	FALSE	TRUE にすると正方向へ移動を開始します。 FALSE にすると移動を終了します。
NegativeEnable	負方向有効	BOOL	TRUE, FALSE	FALSE	TRUE にすると負方向へ移動を開始します。 FALSE にすると移動を終了します。
Velocity	目標速度	LREAL	正数、または「0」	0	目標速度を指定します。 単位は、[指令単位／s] となります。*1
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は、[指令単位／s^2] となります。*1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は、[指令単位／s^2] となります。*1

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

出力変数

出力変数	名称	データ型	有効範囲	内容
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。

出力変数	名称	データ型	有効範囲	内容
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Busy	「PositiveEnable」、または「NegativeEnable」の立ち上がり	• 軸が停止したとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「PositiveEnable」が TRUE の場合、「PositiveEnable」の FALSE と同時 • 「NegativeEnable」が TRUE の場合、「NegativeEnable」の FALSE と同時 • 「PositiveEnable」、および「NegativeEnable」の両方が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

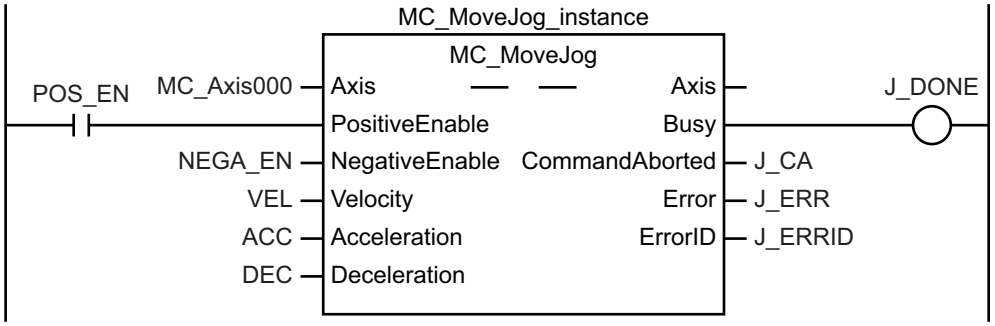
1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 指定した「Velocity」（目標速度）にしたがって、ジョグ送りを行います。
- 正方向にジョグ送りしたい場合は、「PositiveEnable」（正方向有効）を TRUE とし、負方向にジョグ送りしたい場合は「NegativeEnable」（負方向有効）を TRUE とします。
- 「PositiveEnable」（正方向有効）と「NegativeEnable」（負方向有効）を同時に TRUE とした場合は、「PositiveEnable」（正方向有効）優先とし、正方向にジョグ送りとなります。
- MC_MoveJog（ジョグ送り）命令の指令速度が、軸パラメータの［ジョグ最高速度］を超えて指定している場合は、［ジョグ最高速度］となります。

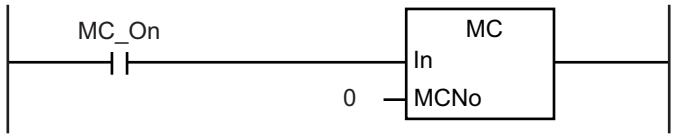
使用上の注意

- ラダー図プログラムを作成する場合、下記のように入力変数「PositiveEnable」（正方向有効）が左母線に接続され、「NegativeEnable」（負方向有効）は変数を指定しなければなりません。

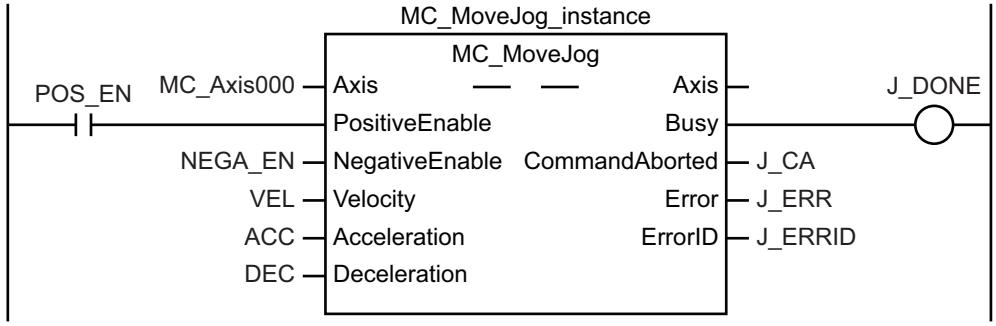


もし、MC_MoveJog（ジョグ送り）に対してマスタコントロール命令（MC 命令）を使用する場合、以下のような記述をしないでください。記述すると、「PositiveEnable」（正方向有効）だけにマスタコントロールが機能し、「NegativeEnable」（負方向有効）にはマスタコントロールが機能しなくなります。

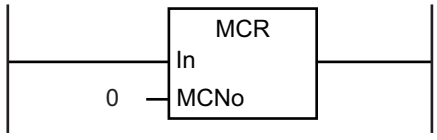
マスタコントロール開始



MC_MoveJog 命令

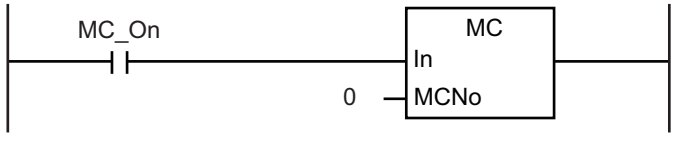


マスタコントロール終了

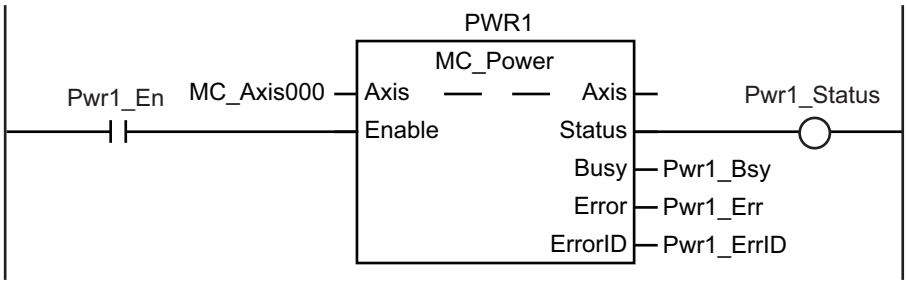


必ず、MC_Power（命令可）に対してマスタコントロール命令を使用してください。

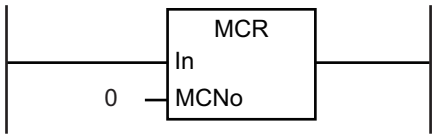
マスタコントロール開始



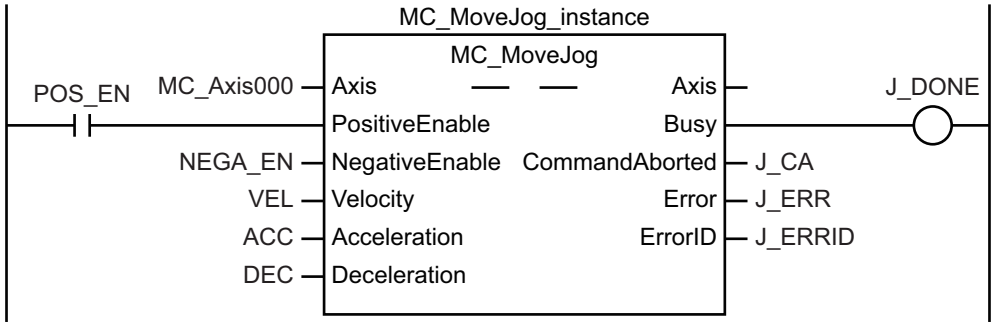
サーボ ON する



マスタコントロール終了

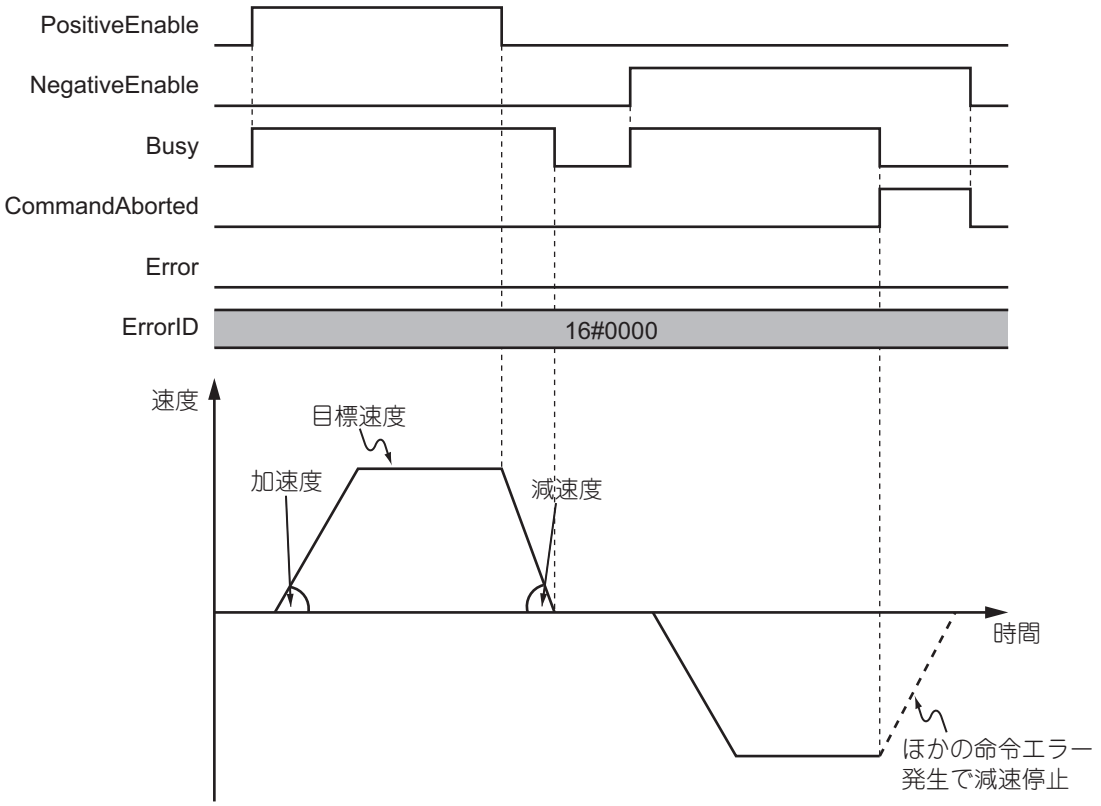


MC_MoveJog 命令



タイミングチャート

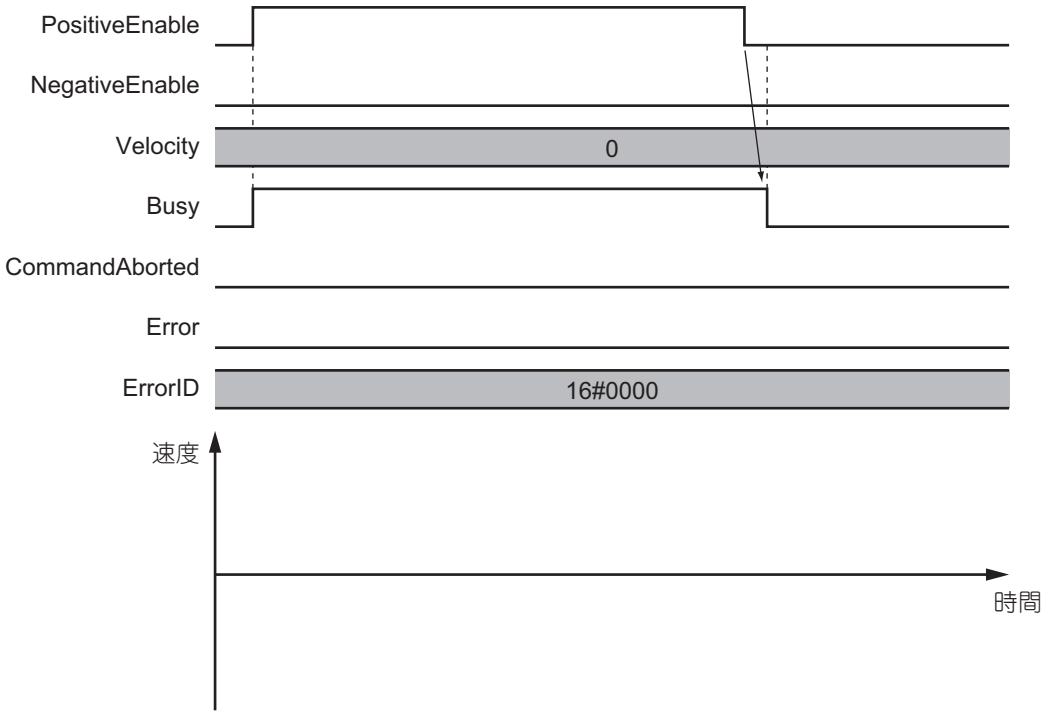
- 「PositiveEnable」（正方向有効）または「NegativeEnable」（負方向有効）の立ち上がりと同時に「Busy」（実行中）が TRUE となります。
- 「PositiveEnable」（正方向有効）または「NegativeEnable」（負方向有効）の立ち下がりで減速を開始し、軸の停止と同時に「Busy」（実行中）が FALSE となります。
- ほかの命令により、本命令が中止された場合は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）が FALSE となります。



入力変数に、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）を指定することができます。
入力変数の「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）は、「PositiveEnable」（正方向有効）、または「NegativeEnable」（負方向有効）の立ち上がり時だけ動作に反映されます。したがって、「PositiveEnable」（正方向有効）、または「NegativeEnable」（負方向有効）が TRUE の間に「Velocity」（目標速度）を変更しても速度は変更されません。

● 目標速度が「0」のときのタイミングチャート

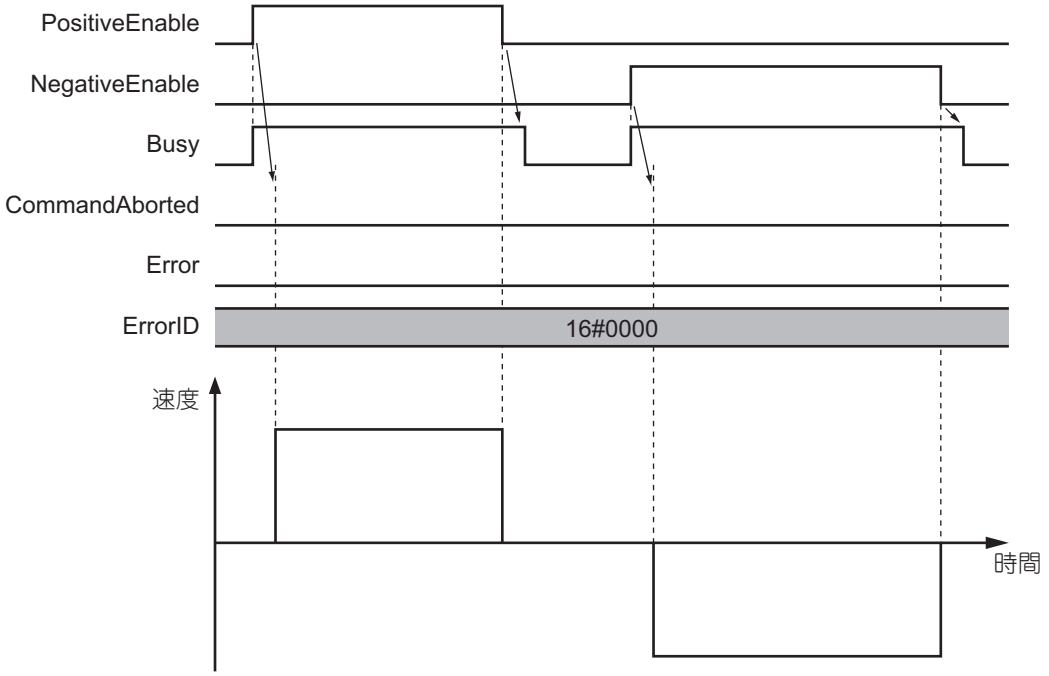
「Velocity」（目標速度）を「0」に指定して起動した場合は、軸は移動しませんが連続動作中となります。「Velocity」（目標速度）を「0」にして起動した場合の動作例を下記に示します。



● 加減速度が「0」のときのタイミングチャート

「Acceleration」（加速度）、または「Deceleration」（減速度）を「0」に指定して起動した場合は、加減速せずに目標速度に到達します。

「Acceleration」（加速度）および「Deceleration」（減速度）の両方が「0」の場合の動作例を下記に示します。

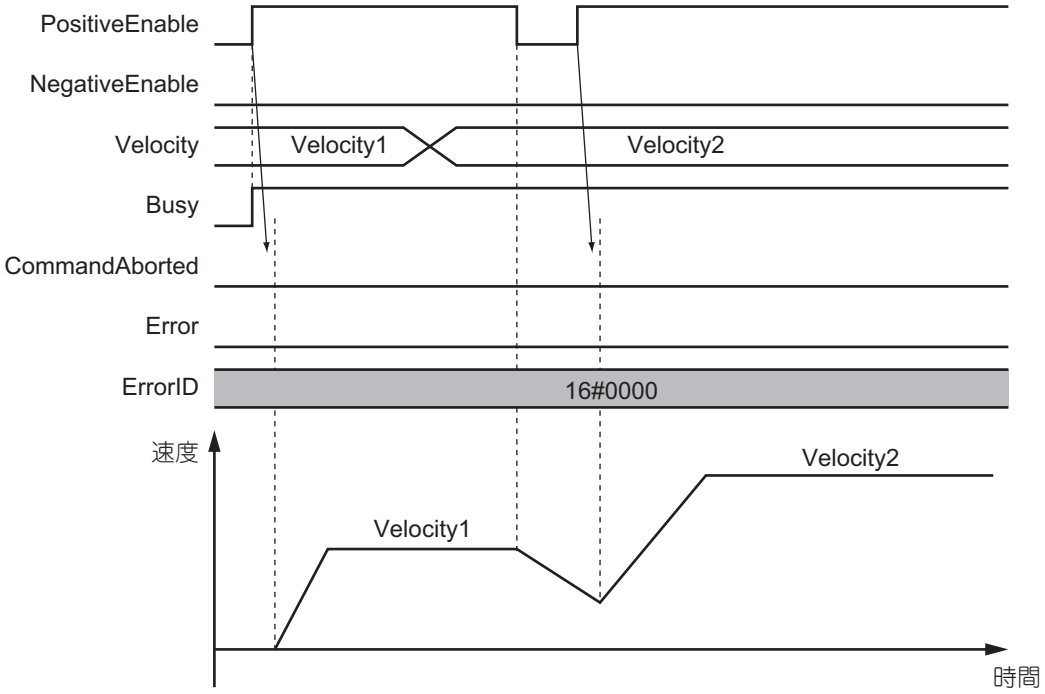


モーション命令再起動

● 同じ方向の「Enable」を再起動する場合

「PositiveEnable」（正方向有効）、または「NegativeEnable」（負方向有効）を FALSE にして減速している最中に、再度、「PositiveEnable」（正方向有効）、または「NegativeEnable」（負方向有効）を TRUE とした場合、減速途中から加減速を行い目標速度に向かって移動します。
このとき、「Velocity」（目標速度）や「Acceleration」（加速度）、「Deceleration」（減速度）を変更していると新たな入力変数で動作します。
また、軸が停止していないため、「Busy」（実行中）は FALSE になりません。

減速中に「PositiveEnable」（正方向有効）を TRUE とした場合の動作例を下記に示します。



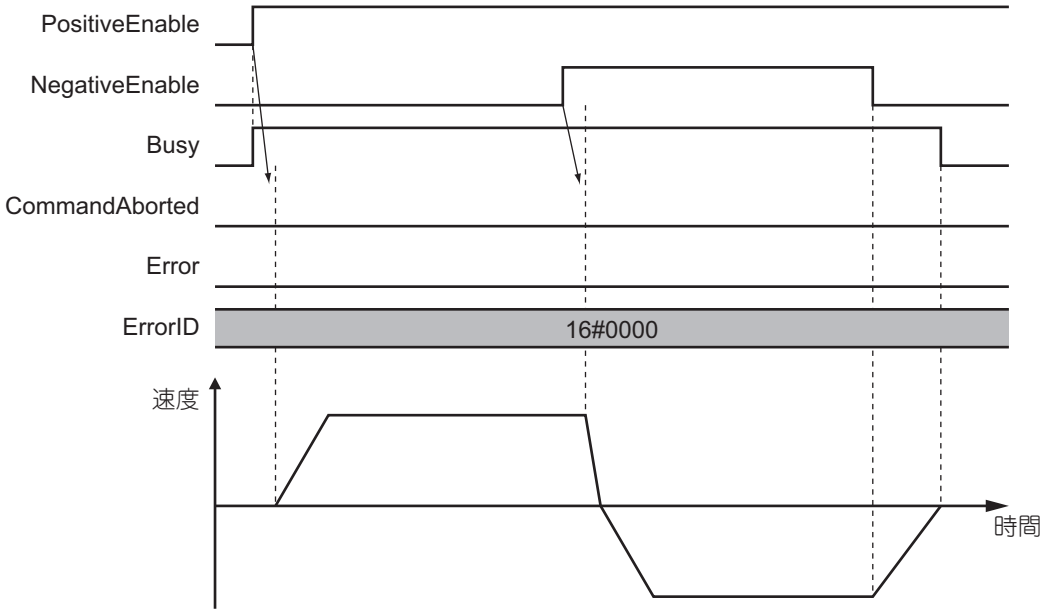
● 異なる方向の「Enable」を再起動する場合

「PositiveEnable」（正方向有効）を TRUE とし、正方向にジョグ送りしている最中に、「NegativeEnable」（負方向有効）を TRUE とすると方向を反転して負方向にジョグ送りします。
この場合、「NegativeEnable」（負方向有効）を TRUE としたときの入力変数でジョグ送りを行います。入力変数には「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）があります。

反転時の減速度および反転後の加速度は、軸パラメータの［反転時動作］にかかわらず「NegativeEnable」（負方向有効）を TRUE としたときの入力変数に従います。
「NegativeEnable」を（負方向有効）TRUE とし、負方向にジョグ送りしている最中に「PositiveEnable」（正方向有効）を TRUE とした場合も同様の動作になります。

「PositiveEnable」（正方向有効）を TRUE とし、その後「NegativeEnable」（負方向有効）を TRUE として負方向にジョグ送りしている最中に、「PositiveEnable」（正方向有効）が TRUE のままの状態で「NegativeEnable」（負方向有効）を FALSE にしても、正方向にジョグ送りしません。
正方向にジョグ送りしたい場合は、「PositiveEnable」（正方向有効）を一度 FALSE にしてから再度 TRUE としてください。
逆のケースも同様の動作になります。

「PositiveEnable」（正方向有効）を TRUE としてから「NegativeEnable」（負方向有効）を TRUE とした場合の動作例を下記に記します。



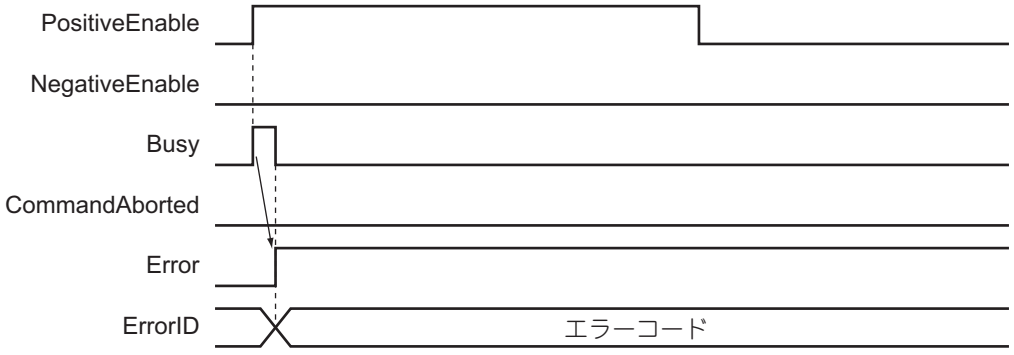
モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート

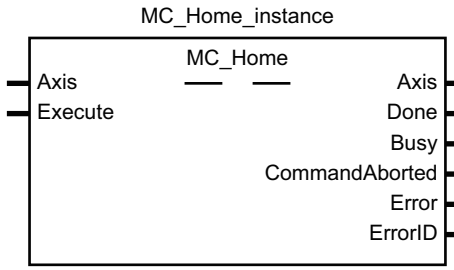


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_Home

モータを動かして、限界信号、原点近傍信号、原点信号を用いて機械原点を決めます。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_Home	原点復帰	FB		MC_Home_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後

変数	TRUE となるタイミング	FALSE となるタイミング
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 「Axis」（軸）で指定された軸に対して、「Execute」（起動）の立ち上がりで原点復帰動作を開始します。
- MC_Home（原点復帰）命令で使用される各種パラメータは軸パラメータで設定します。
- MC_Home（原点復帰）命令には、10 種類の原点復帰動作モードがあります。
  Sysmac Studio の軸パラメータ［原点復帰動作］で設定してください。

使用上の注意

主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。

● データオブジェクトのマッピング

MC_Home（原点復帰）命令を使用する場合、Sysmac Studio の軸基本設定の［詳細設定］で、以下のオブジェクトデータをマッピングしてください。
ただし、原点復帰動作モード「11」、「12」、「14」を使用する場合は、設定の必要はありません。

- ラッチ機能(60B8Hex)
- ラッチステータス(60B9Hex)
- ラッチ 1 位置(60BAHex)

これらを設定していない場合、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。

データオブジェクトのマッピングについては、「2-3 PDO マッピング (P.2-32)」、および『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

NX シリーズ 位置インタフェースユニットを使用する場合は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』の「I/O エントリマッピング」を参照してください。

オムロン製サーボドライバ 1S シリーズの設定

MC_Home（原点復帰）命令で使用する原点近傍信号などの入力信号については、オムロン製サーボドライバ 1S シリーズで設定してください。

入力信号の設定方法については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「サーボドライバとの接続」、および『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-377)』、または『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ 安全機能対応 ユーザーズマニュアル(SBCE-438)』の「汎用入力信号」を参照してください。

オムロン製サーボドライバ G5 シリーズの設定

MC_Home（原点復帰）命令で使用する原点近傍信号などの入力信号については、オムロン製サーボドライバ G5 シリーズで設定してください。

入力信号の設定方法については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「サーボドライバとの接続」、および『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-365)』、『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプ ユーザーズマニュアル(SBCE-366)』の「シーケンス入出力信号」を参照してください。

NX シリーズ 位置インタフェースユニットの設定

NX シリーズ 位置インタフェースユニットを使用する場合の設定は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。

原点復帰動作モード

機械原点を決定する動作として、10 種類の動作から選択できます。

- 近傍退避・原点近傍入力 OFF 指定
- 近傍退避・原点近傍入力 ON 指定
- 原点近傍入力 OFF 指定
- 原点近傍入力 ON 指定
- 限界入力 OFF 指定
- 近傍退避・原点入力マスク距離指定
- 限界入力のみ
- 近傍退避・押し当て時間指定
- 原点近傍入力なし・押し当て原点入力指定
- 原点プリセット

各原点復帰動作ごとに使用する原点復帰パラメータは以下の表のとおりです。

（O: パラメータを使用する、－: パラメータを使用しない）

原点復帰動作	原点復帰パラメータ														
	原点入力信号	原点復帰開始方向	原点検出方向	正方向限界入力時動作	負方向限界入力時動作	原点復帰速度	原点復帰アプローチ速度	原点復帰加速度	原点復帰減速度	原点復帰ジャーク	原点入力マスク距離	原点位置オフセット	原点復帰押し当て時間	原点復帰補正値	原点復帰補正速度
近傍退避･原点近傍入力 OFF 指定	○	○	○	○	○	○	○	○	○	○	－	○	－	○	○
近傍退避･原点近傍入力 ON 指定	○	○	○	○	○	○	○	○	○	○	－	○	－	○	○
原点近傍入力 OFF 指定	○	○	○	○	○	○	○	○	○	○	－	○	－	○	○
原点近傍入力 ON 指定	○	○	○	○	○	○	○	○	○	○	－	○	－	○	○
限界入力 OFF 指定	○	○	○	○	○	○	○	○	○	○	－	○	－	○	○
近傍退避･原点入力マスク距離指定	○	○	○	○	○	○	○	○	○	○	○	○	－	○	○
限界入力のみ	－	○	○	○	○	○	○	○	○	○	－	○	－	○	○
近傍退避･押し当て時間指定	－	○	○	○	○	○	○	○	○	○	－	○	○	○	○
原点近傍入力なし･押し当て原点入力指定	○	○	○	○	○	○	○	○	○	○	－	○	－	○	○
原点プリセット	－	－	－	－	－	－	－	－	－	－	－	○	－	－	－

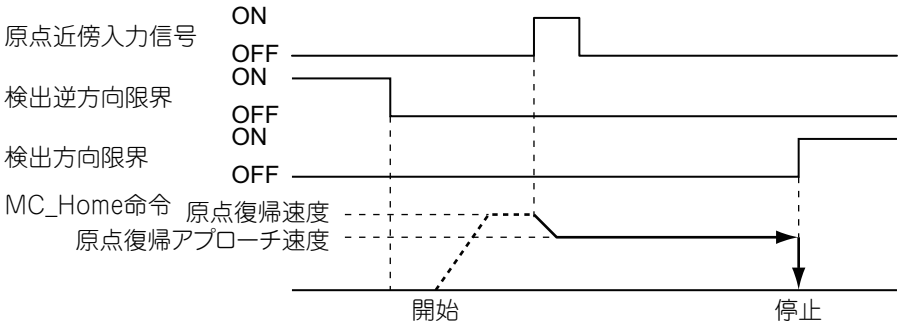
各原点復帰モードの動作の詳細については、「原点確定の動作 (P.3-22)」を参照してください。

参考

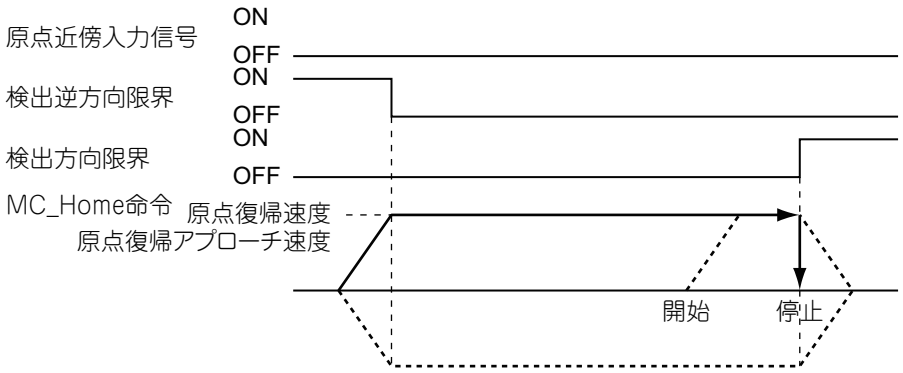
NX シリーズ 位置インタフェースユニットを使用する場合、押し当てを使用する原点復帰動作モードを選択できません。詳細は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。

正方向限界入力時動作と負方向限界入力時動作

- 原点復帰中に動作方向の限界入力まで到達したときに、反転して原点復帰を続けさせるか、反転させないで異常停止させるかを選択します。反転させる場合は、その停止方法も選択します。
- ［反転する］に設定している場合に、原点復帰アプローチ速度で原点検出方向の限界信号が「ON」すると異常停止します。
  ただし、近傍信号を使用しない原点復帰動作モード［13: 原点近傍入力なし・押し当て原点入力指定］の場合は、異常停止となりません。



- また、両方向の限界入力時動作を［反転する］に設定している場合、原点検出方向に対して逆側の限界入力から他方の限界入力まで動作して、原点が検出できなかったときには、異常停止します。



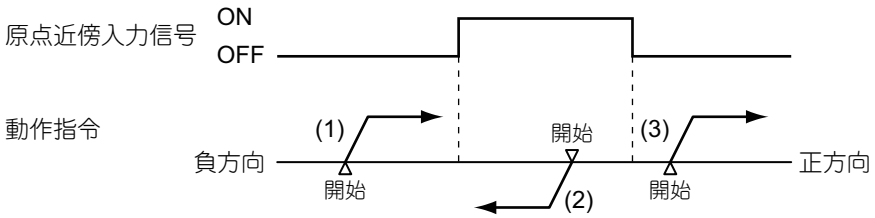
原点復帰開始方向

原点復帰の起動時に動き出す方向を選択します。
ただし、近傍退避による反転動作を含む原点復帰動作モードにおいて、原点近傍信号が「ON」の状態で原点復帰を起動した場合は、原点復帰開始方向の設定にかかわらず、原点検出方向と反対方向に動き出します。

近傍退避による反転動作を含む原点復帰動作モードは以下の 4 つです。

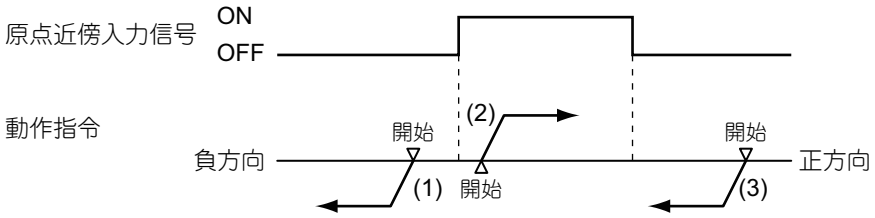
- 0: 近傍退避・原点近傍入力 OFF 指定
- 1: 近傍退避・原点近傍入力 ON 指定
- 9: 近傍退避・原点入力マスク距離指定
- 12: 近傍退避・押し当て時間指定

原点復帰開始方向:正
原点検出方向:正



(1), (3) : 原点近傍信号「OFF」状態から開始のため、原点復帰開始方向に動作開始
(2) : 原点近傍信号「ON」状態から開始のため、原点検出方向と反対方向に動作開始

原点復帰開始方向:負
原点検出方向:負



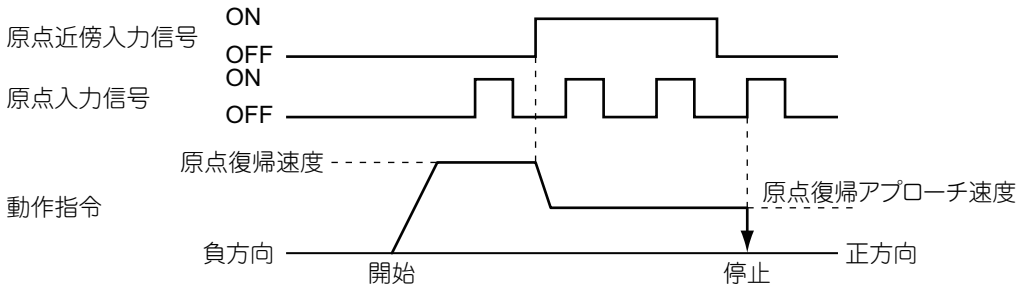
(1), (3) : 原点近傍信号「OFF」状態から開始のため、原点復帰開始方向に動作開始
(2) : 原点近傍信号「ON」状態から開始のため、原点検出方向と反対方向に動作開始

原点検出方向

原点入力検出方向を選択します。

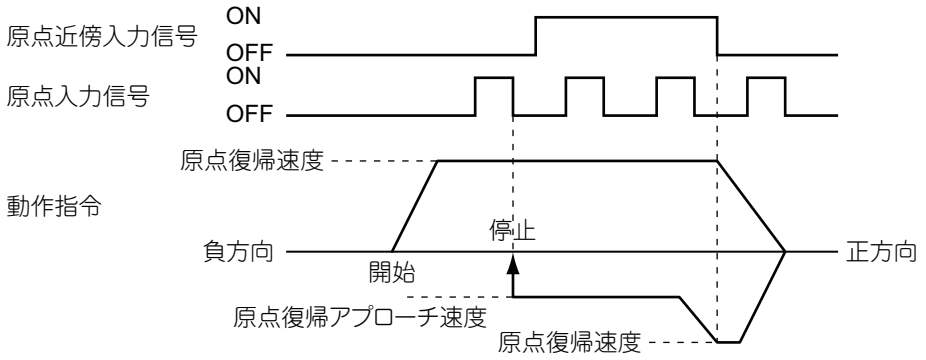
原点検出方向 が FALSE（正方向）の場合の動作は以下のようになります。

動作例:近傍退避･原点近傍入力OFF指定



また、原点検出方向 が TRUE（負方向）の場合の動作は以下のようになります。

動作例:近傍退避･原点近傍入力OFF指定



原点入力マスク距離

原点復帰動作モードが［9: 近傍退避・原点入力マスク距離指定］設定時の定寸距離を設定します。
動作の詳細については、「9: 近傍退避・原点入力マスク距離指定の動作 (P.3-25)」を参照してください。

原点復帰補正値

原点検出後の原点復帰補正量を設定します。
このときの移動速度は、［原点復帰補正速度］で設定します。
動作の詳細については、「原点補正 (P.3-31)」を参照してください。

原点位置オフセット

原点復帰動作完了時の現在位置プリセットとなります。
［原点位置オフセット］に設定した値が、現在位置となります。

原点復帰速度

原点復帰時の高速速度となります。

原点復帰アプローチ速度

原点復帰時の近傍速度となります。

原点復帰補正速度

［原点復帰補正値］を設定した場合の、移動速度となります。
動作の詳細については、「原点補正 (P.3-31)」を参照してください。

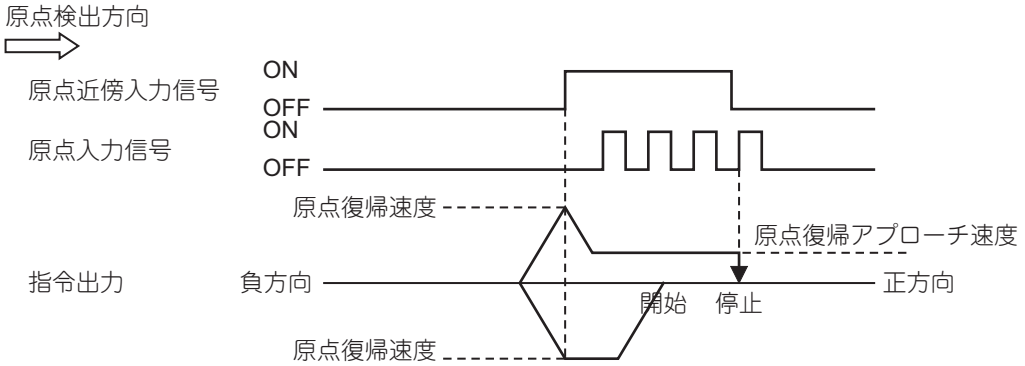
原点確定の動作

ここでは、10 種類の原点復帰動作モードについて説明します。

● 0: 近傍退避・原点近傍入力 OFF 指定の動作

1 **原点復帰速度で起動し、原点近傍入力信号の「OFF」 → 「ON」により、原点復帰アプローチ速度に減速を開始します。**

2 **原点復帰アプローチ速度に到達し、原点近傍入力信号が「ON」 → 「OFF」した後の最初の原点入力で停止し、機械原点が確定します。**

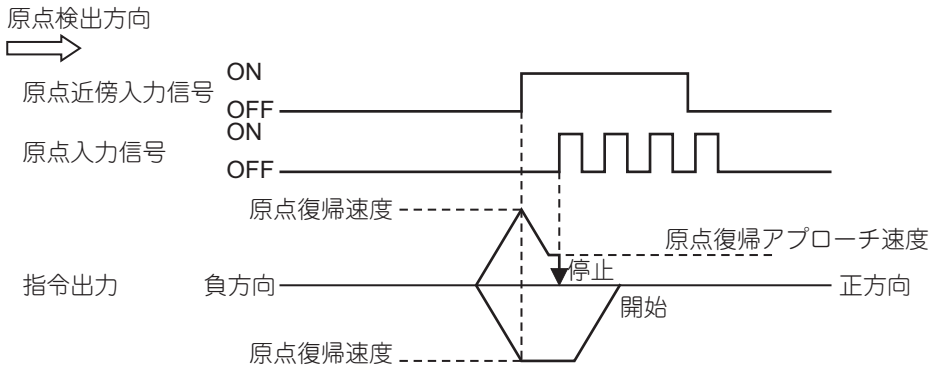


- 原点近傍入力信号が「ON」している状態から原点復帰を行うと、原点検出方向と逆方向に原点復帰速度で動作を開始します。原点近傍入力信号の「ON」状態から、いったん退避した後は、原点検出方向へ原点復帰速度で原点復帰動作を開始します。
- 原点復帰開始後、原点復帰アプローチ速度以下の状態のまま、原点近傍入力信号が「OFF」→「ON」→「OFF」した後に、原点入力信号が「ON」した場合にも機械原点が確定します。

● 1: 近傍退避・原点近傍入力 ON 指定の動作

1 **原点復帰速度で起動し、原点近傍入力信号の「OFF」 → 「ON」により、原点復帰アプローチ速度に減速を開始します。**

2 **原点復帰アプローチ速度に到達した後の最初の原点入力で停止し、機械原点が確定します。**
一度「ON」した後の原点近傍入力信号の状態は動作に影響しません。

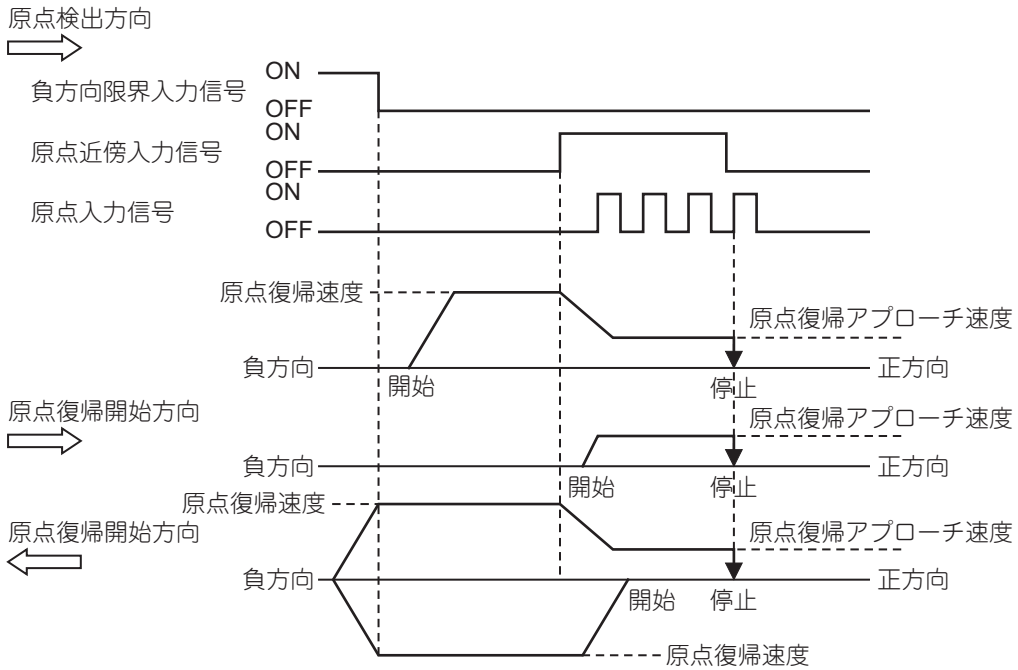


- 原点近傍入力信号が「ON」している状態から原点復帰を行うと、原点検出方向と逆方向に原点復帰速度で動作を開始します。原点近傍入力信号の「ON」状態から、いったん退避した後は、原点検出方向へ原点復帰速度で原点復帰動作を開始します。
- 原点復帰開始後、原点復帰アプローチ速度以下の状態のまま、原点近傍入力信号が「ON」した後に、原点入力信号が「ON」した場合にも機械原点が確定します。

● 4: 原点近傍入力 OFF 指定の動作

1 **原点復帰速度で起動し、原点近傍入力信号の「OFF」 → 「ON」により、原点復帰アプローチ速度に減速を開始します。**

2 **原点復帰アプローチ速度に到達し、原点近傍入力信号が「ON」 → 「OFF」した後の最初の原点入力で停止し、機械原点が確定します。**



- 原点近傍入力信号が「ON」している状態から原点復帰を行うと、原点復帰開始方向の設定により以下のような動作となります。

原点復帰開始方向と原点検出方向が同一の場合
退避動作は行わず、原点検出方向に原点復帰アプローチ速度で原点復帰動作を開始します。

原点復帰開始方向と原点検出方向が異なる場合

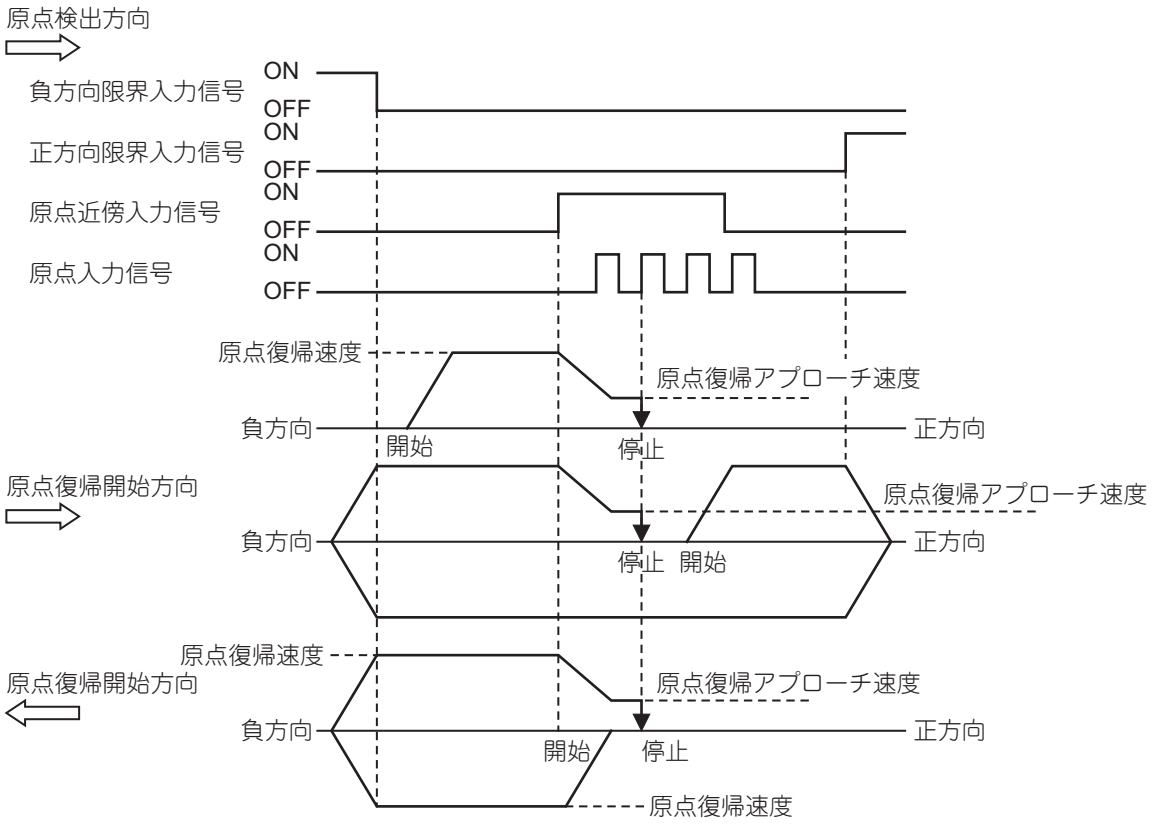
原点近傍入力信号の状態によらず、原点復帰速度で原点復帰開始方向への動作を開始します。原点復帰開始方向の限界入力時動作が［反転する］設定の場合は、限界入力検出時に反転し、原点検出方向への原点復帰動作を行います。

- 原点復帰開始後、原点復帰アプローチ速度以下の状態のまま、原点近傍入力信号が「OFF」→「ON」→「OFF」した後に、原点入力信号が「ON」した場合にも機械原点が確定します。

● 5: 原点近傍入力 ON 指定の動作

1 **原点復帰速度で起動し、原点近傍入力信号の「OFF」 → 「ON」により、原点復帰アプローチ速度に減速を開始します。**

2 **原点復帰アプローチ速度に到達した後の最初の原点入力で停止し、機械原点が確定します。**
一度「ON」した後の原点近傍入力信号の状態は動作に影響しません。



- 原点近傍入力信号が「ON」している状態から原点復帰を行うと、原点復帰開始方向の設定により以下のような動作となります。

原点復帰開始方向と原点検出方向が同一の場合
退避動作は行わず、原点検出方向に原点復帰速度で原点復帰動作を開始します。

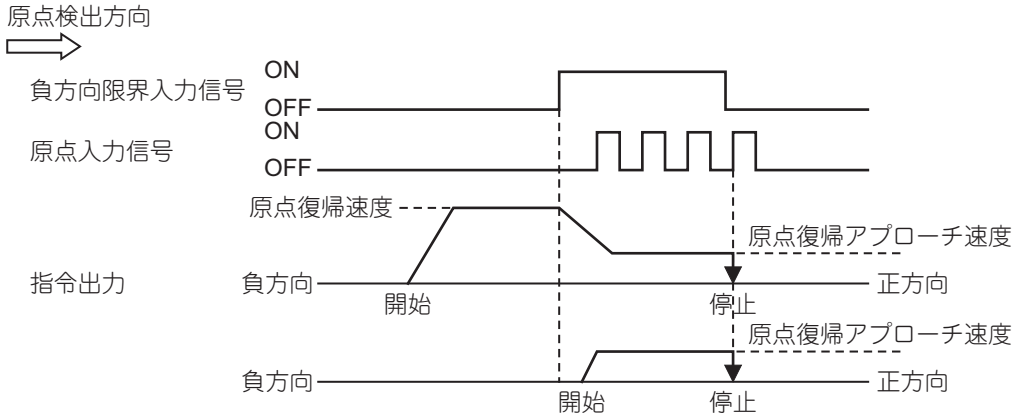
原点復帰開始方向と原点検出方向が異なる場合
原点近傍入力信号の状態によらず、原点復帰速度で原点復帰開始方向への動作を開始します。原点復帰開始方向の限界入力時動作が［反転する］設定の場合は、限界入力検出時に反転し、原点検出方向への原点復帰動作を行います。

- 原点復帰開始後、原点復帰アプローチ速度以下の状態のまま、原点近傍入力信号が「OFF」→「ON」した後に、原点入力信号が「ON」した場合にも機械原点が確定します。

● 8: 限界入力 OFF 指定の動作

1 **原点復帰速度で起動し、原点検出方向と反対方向の限界信号の「OFF」 → 「ON」により、原点復帰アプローチ速度に減速を開始します。**

2 **原点復帰アプローチ速度に到達し、限界信号が「ON」 → 「OFF」した後の最初の原点入力で停止し、機械原点が確定します。**



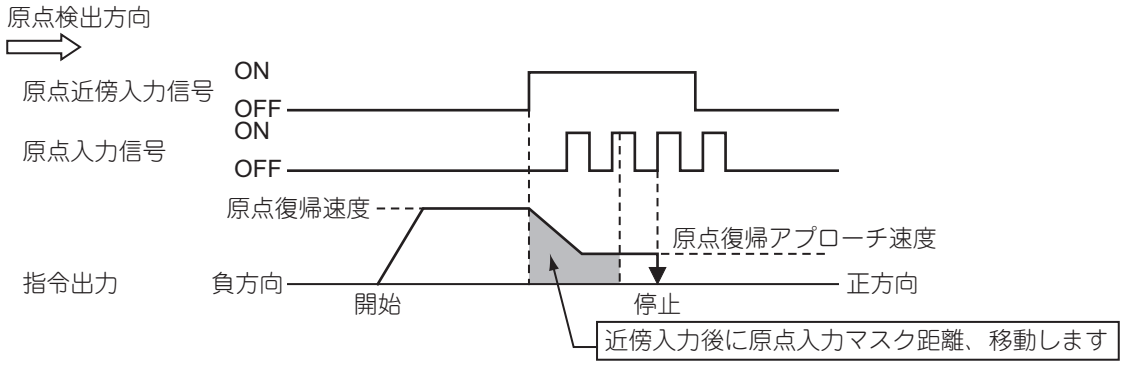
- 原点検出方向と逆側の限界入力が「ON」している状態から原点復帰を行うと、原点検出方向へ原点復帰アプローチ速度で原点復帰動作を行います。
- 原点復帰開始後、原点復帰アプローチ速度以下の状態のまま、検出方向と逆方向の限界信号が「OFF」→「ON」→「OFF」した後に、原点入力信号が「ON」した場合にも機械原点が確定します。

● 9: 近傍退避・原点入力マスク距離指定の動作

1 **原点復帰速度で起動し、原点近傍入力信号の「OFF」 → 「ON」により、原点復帰アプローチ速度に減速を開始します。**

2 **原点近傍入力信号が「OFF」 → 「ON」後、原点入力マスク距離を移動し終わってから最初の原点入力で停止し、機械原点が確定します。**

一度「ON」した後の原点近傍入力信号の状態は動作に影響しません。
その際、設定された移動量が短く、原点復帰アプローチ速度に減速する前に移動量に達するような場合は、原点復帰起動時に「原点入力マスク距離不正（エラーコード: 742B Hex）」となります。



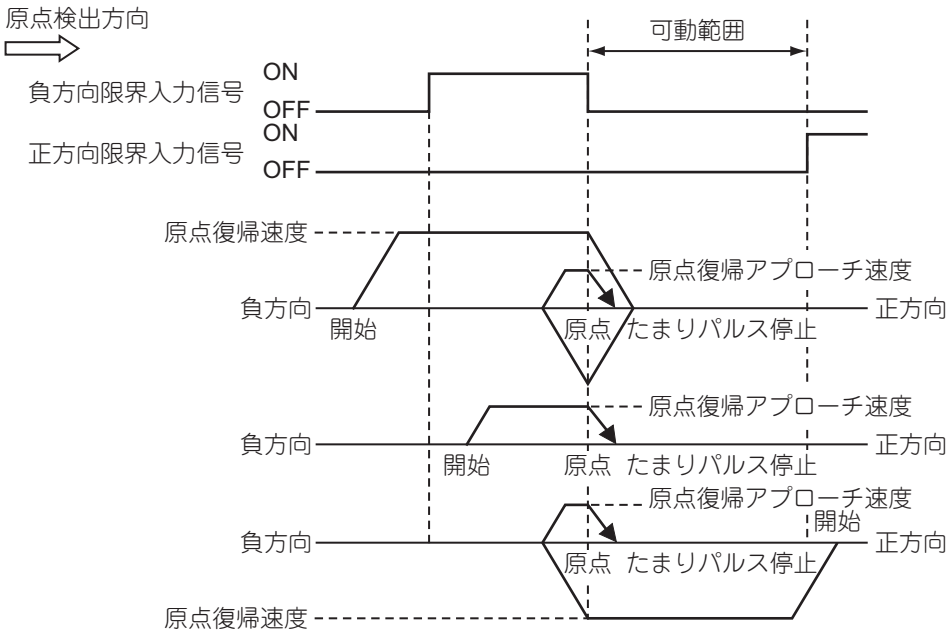
- 原点近傍入力信号が「ON」している状態から原点復帰を行うと、原点検出方向と逆方向に原点復帰速度で動作を開始します。原点近傍入力信号の「ON」状態からいったん退避した後は、原点検出方向へ原点復帰アプローチ速度で原点復帰動作を開始します。
- 原点復帰開始後、原点復帰アプローチ速度以下の状態のまま、原点近傍入力信号が「ON」した後に、原点入力マスク距離を移動し、原点入力信号が「ON」した場合にも機械原点が確定します。

● 11: 限界入力のみの動作

1 **原点復帰速度で起動し、原点検出逆方向の限界信号の「ON」 → 「OFF」により減速停止します。**

2 **停止後、原点復帰速度で反転し、限界信号の「OFF」 → 「ON」で減速停止します。**

3 **停止後、原点復帰アプローチ速度で反転し、限界信号の「ON」 → 「OFF」でその位置を原点とし、即停止します（たまりパルス停止）。ただし、機械原点位置へ戻る動作は行いません。**



- 限界信号だけを使用して原点復帰を行う場合、原点検出方向に動作したときの限界信号の「ON」 → 「OFF」変化点を原点検出位置とします。
- 原点検出方向の限界信号が「ON」している状態から原点復帰を行う場合、原点検出逆方向に原点復帰速度で動作を開始します。原点検出逆方向の限界信号の「OFF」 → 「ON」により減速停止します。
- 限界信号検出後、減速停止する前に限界信号が「OFF」しても機械原点は確定せず、減速を続けます。この場合、異常とはなりません。
- 反転後や限界信号入力状態からの起動後、原点復帰アプローチ速度へ到達する前に限界信号が「ON」→「OFF」した場合、機械原点が確定します。

原点復帰動作モード 11 とほかの原点復帰動作モードとの違い

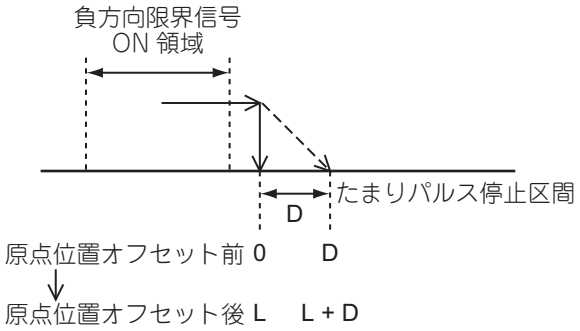
モード 11 は、MC 機能モジュールでの処理で「限界信号の ON→OFF」を検出し、原点を確定させる原点復帰動作です。そのため、原点入力を外部ラッチ信号として用いる原点復帰動作モードと以下の点で異なります。

モード 11 とほかの原点復帰動作モードの違いをご理解いただいた上で、使用してください。

- 限界入力の検出ではサーボドライバ等の外部ラッチ機能を使用せず、MC 機能モジュールの処理で行います。MC 機能モジュールの処理とは、プライマリ定周期タスクの場合はプライマリ周期

間隔で、定周期タスク（実行優先度 5）の場合は定周期タスク（実行優先度 5）のタスク周期での処理を指します。そのため、原点確定の精度は、原点復帰アプローチ速度やプライマリ定周期タスク、定周期タスク（実行優先度 5）の制御周期に依存します。

- 機械原点へ戻る動作は行いません。原点補正量が「0」の場合、機械原点とは異なる位置で停止した状態で原点復帰処理が終了します。
- 原点補正量として「0」を与えた場合、原点補正動作を行いません。原点補正量が「0」の場合、前述のとおり機械原点と異なる位置で原点復帰処理が終了します。原点補正量が「0」以外の場合はほかの原点復帰動作モードと同じく原点補正量を機械原点からの相対量として動作します。
- 原点位置オフセットは、機械原点位置を変更します。下図のように停止位置が原点から「D」ずれていた場合、原点位置オフセットが「L」、原点補正量が「0」なら原点復帰処理終了時の位置は「L + D」となります。

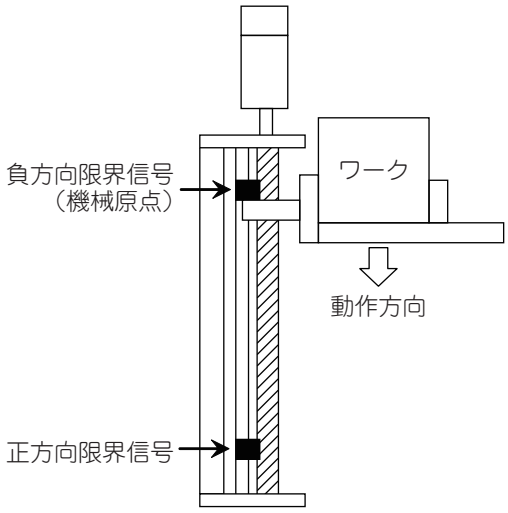


使用上の注意

- 「限界信号 ON→OFF」検出後に、たまりパルス停止をして原点補正動作を開始するまでの間は、原点検出方向と逆方向の限界信号を無視します。
- 原点補正量が「0」で、原点確定させた直後に原点検出逆方向の限界入力信号が「ON」するような場合は、限界信号入力から離れる方向に原点補正量を設定してください。

原点復帰動作モード 11 の使用例

下図のように「負方向限界信号」と「原点信号」の両方を設置するスペースがない場合に、「負方向限界信号」に限界信号と機械原点の両方の役割を持たせて使用することができます。



● 12: 近傍退避・押し当て時間指定の動作

1 **原点復帰速度で起動し、原点近傍入力信号の「OFF」→「ON」により、原点復帰アプローチ速度に減速を開始します。**

減速開始と同時に時間監視を開始します。

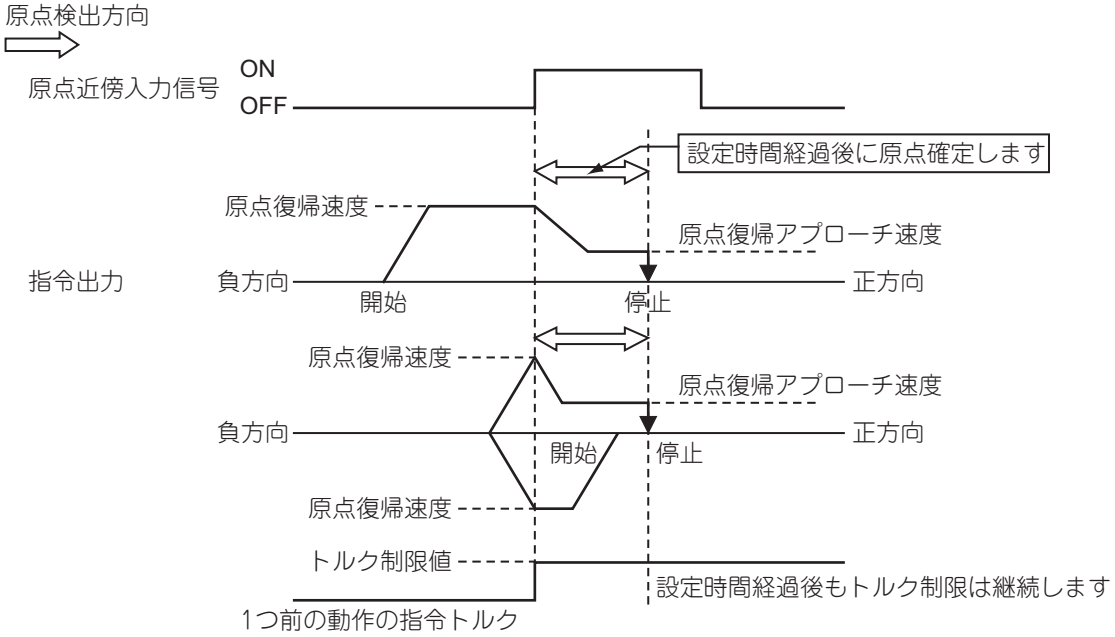
押し当て開始時のトルク制限は、オムロン製サーボドライバ 1S シリーズおよび G5 シリーズとそれ以外のサーボでは扱いが以下のように異なります。

サーボドライバ	動作
1S シリーズ、G5 シリーズ	自動的にトルク制限を実行します。
上記以外のサーボ	設定によりトルク制限を実行します。

押し当て動作については、「押し当て動作（オムロン製サーボドライバ 1S シリーズを使用する場合）（P.3-31）」、または「押し当て動作（オムロン製サーボドライバ G5 シリーズを使用する場合）（P.3-32）」を参照してください。

2 **設定時間が経過後、機械原点が確定します。**

一度「ON」した後の原点近傍入力信号の状態は動作に影響しません。



- 原点近傍入力信号が「ON」している状態から原点復帰を行うと、原点検出方向と逆方向に原点復帰速度で動作を開始します。原点近傍入力信号の「ON」状態からいったん退避した後は、原点検出方向へ原点復帰アプローチ速度で原点復帰動作を開始します。
- トルク制限の解除についても、オムロン製サーボドライバ 1S シリーズおよび G5 シリーズとそれ以外のサーボでは扱いが異なります。

サーボドライバ	動作
1S シリーズ、G5 シリーズ	原点復帰後、原点復帰と逆方向に初めて動くときに自動的に「OFF」します。
上記以外のサーボ	トルク制限使用時、原点復帰後、初めて原点復帰と逆方法に移動される場合に、トルク制限を解除してください。トルク制限の書き替え時は、EC_CoESDOWrite（CoE SDO 書込）命令を使用してください。

- 原点近傍入力信号の検出後、原点復帰アプローチ速度へ到達する前に、押し当て時間が経過した場合にも異常とはならず、機械原点が確定します。
- 原点復帰開始後、原点復帰アプローチ速度以下の状態のまま、原点近傍入力信号が「OFF」→「ON」した後に、押し当て時間が経過した場合にも機械原点が確定します。

● 13: 原点近傍入力なし・押し当て原点入力指定の動作

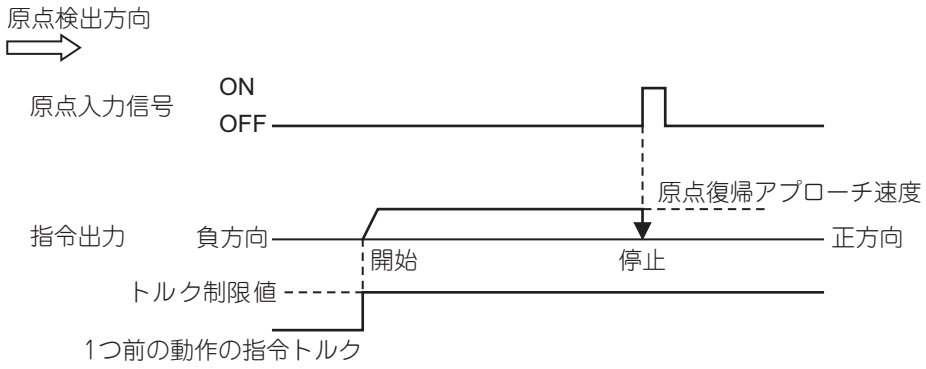
1 原点復帰アプローチ速度で起動します。

押し当て開始時のトルク制限は、オムロン製サーボドライバ 1S シリーズおよび G5 シリーズとそれ以外のサーボでは扱いが以下のように異なります。

サーボドライバ	動作
1S シリーズ、G5 シリーズ	自動的にトルク制限を実行します。
上記以外のサーボ	必要に応じてトルク制限を実行します。

押し当て動作については、「押し当て動作（オムロン製サーボドライバ 1S シリーズを使用する場合）（P.3-31）」、または「押し当て動作（オムロン製サーボドライバ G5 シリーズを使用する場合）（P.3-32）」を参照してください。

2 原点入力の「OFF」 → 「ON」により、機械原点が確定します。



- トルク制限の解除についても、オムロン製サーボドライバ 1S シリーズおよび G5 シリーズとそれ以外のサーボでは扱いが異なります。

サーボドライバ	動作
1S シリーズ、G5 シリーズ	原点復帰後、原点復帰と逆方向に初めて動くときに自動的に「OFF」します。
上記以外のサーボ	トルク制限使用時、原点復帰後、初めて原点復帰と逆方法に移動される場合に、トルク制限を解除してください。トルク制限の書き替え時は、EC_CoESDOWrite（CoE SDO 書込）命令を使用してください。

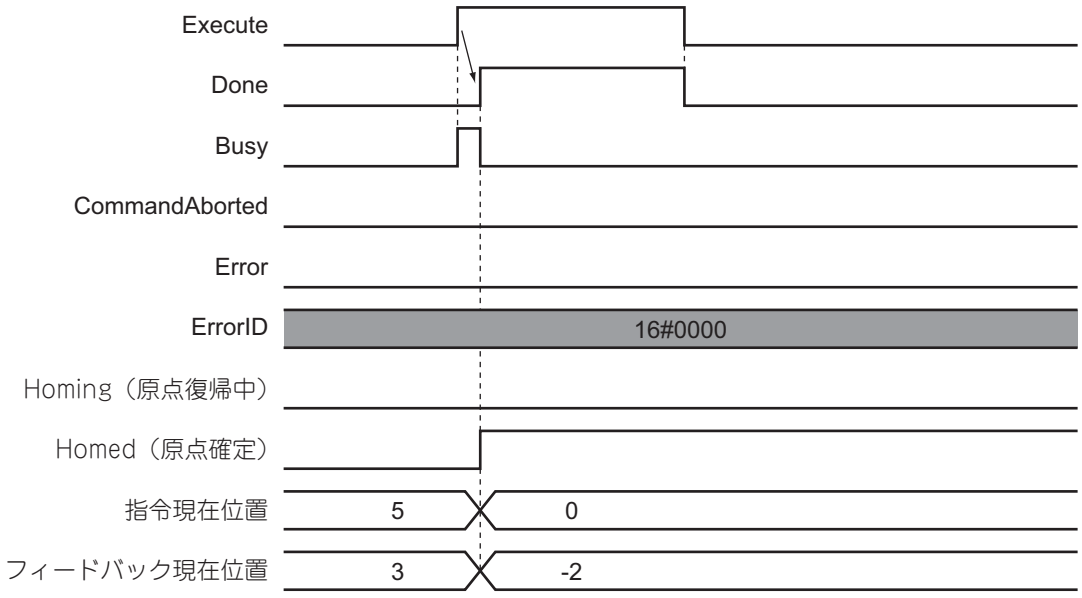
- 原点復帰開始後、原点復帰アプローチ速度へ到達する前に、原点入力信号が「ON」した場合にも機械原点が確定します。

使用上の注意

オムロン製サーボドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプは Z 相入力を PDO にマッピングできません。そのため、PDO にマッピングした Z 相入力を使用する原点復帰動作の［13: 原点近傍入力なし・押し当て原点入力指定］を使用するときは、原点入力信号に Z 相入力を選択しないでください。

● 14: 原点プリセットの動作

指令現在位置を原点位置オフセットに変更して、機械原点を確定します。
また、このときの指令現在位置とフィードバック位置の偏差は維持されます。



命令の詳細

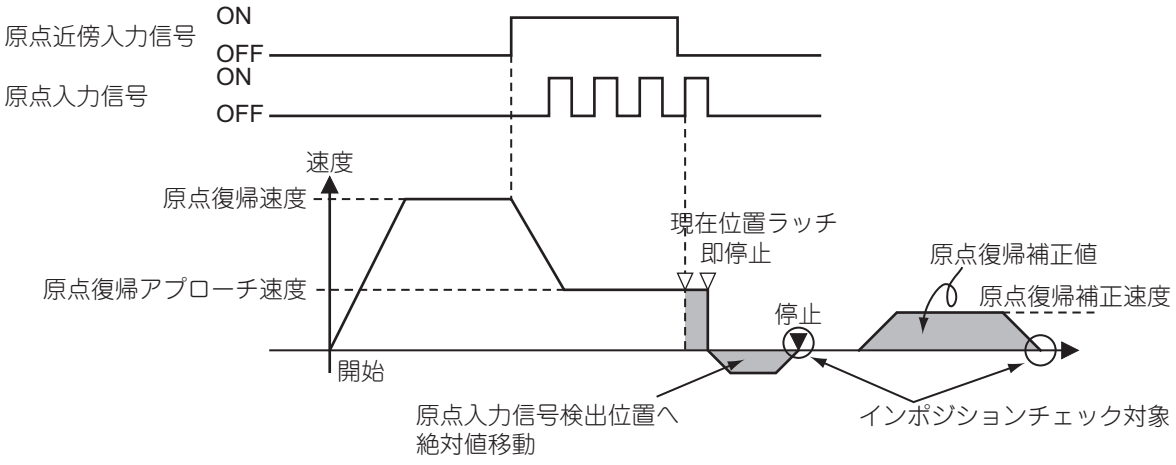
以下に本命令の詳細について説明します。

● 原点確定時の停止方法とインポジションチェック

原点確定時は以下のような手順で停止を行います。

1 **原点入力信号検出時の現在位置をラッチし、即停止を行う。**

2 **即停止後、1 でラッチした位置へ絶対移動で反転移動する。**



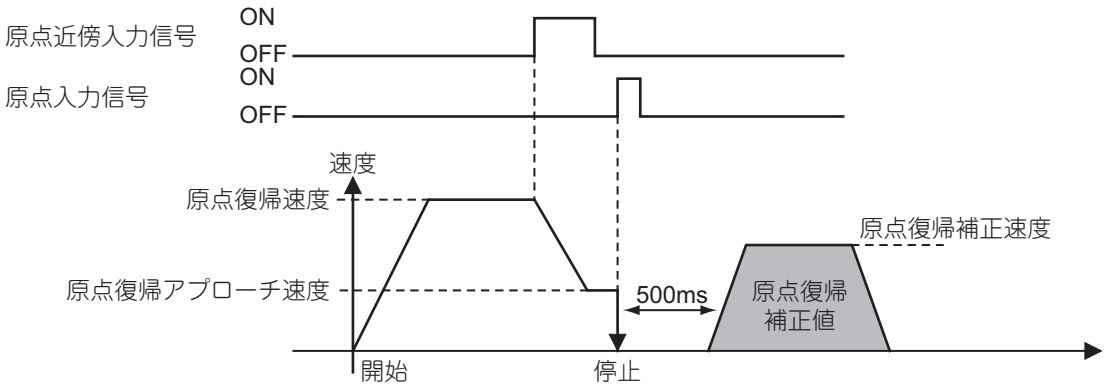
- 本命令では、軸パラメータ［インポジションチェック時間］の設定にかかわらず、常に位置決め完了待ちを行います。
  特に、原点入力信号検出位置へ絶対移動を行う場合、およびインポジションチェック時間を「0」に設定した場合には、位置決め完了を待ち続けますのでインポジション幅に確実に入るように設定してください。
  原点復帰補正値の動作では、インポジションチェック時間を「0」にしてもインポジション待ちを行いません。

なお、インポジション待ち状態かどうかはモーション制御システム変数「InPosWaiting」（インポジション待ち）で確認することができます。

● 原点補正

原点復帰補正値を設定することにより、原点入力を検出後、原点復帰補正値分移動し原点を確定します。
メカ系で機械原点が確定したあとに、ワークの調整を行うことは困難なため、微調整が必要な場合に最初に確定した機械原点からのズレ量を補正データとして設定します。
このときの移動速度は原点復帰補正速度となり、原点入力検出による停止後 500ms 経過してから移動を開始します。

原点復帰補正値の符号は、座標軸に対する方向を表し、原点復帰補正値が負数の時は負方向へ動作します。



● オーバーライド

本命令に対して、オーバーライドは無効となります。

● トルク制限の自動制御

オムロン製サーボドライバ 1S シリーズまたは G5 シリーズ使用時に、原点復帰動作で、［12: 近傍退避・押し当て時間指定］、［13：原点近傍入力なし・押し当て原点入力指定］のいずれかを選択した場合、押し当て方向にトルク制限が自動的に開始されます。
原点復帰時のトルク制限の自動制御は、原点復帰後もトルク制限状態を保持し、原点検出方向と逆方向に動作するときにトルク制限を無効にします。
また、以下の場合にもトルク制限が自動解除されます。

- サーボ OFF（サーボアンロック）
- サイクリック同期位置制御(CSP)から、別の制御モードに切り替えた場合

原点復帰動作中は、原点検出方向とは逆の方向に動作するときにトルク制限は解除されます。
たとえば、限界入力時の反転動作方向が原点検出方向とは逆方向の場合は、反転完了時にトルク制限は解除されます。
さらに反転して動作方向が原点検出方向となった場合は、再度、トルク制限を有効にします。

● 押し当て動作（オムロン製サーボドライバ 1S シリーズを使用する場合）

原点復帰の［12: 近傍退避・押し当て時間指定］、［13: 原点近傍入力なし・押し当て原点入力指定］は、サーボドライバにあらかじめ設定されたトルク制限値を用いて原点検出方向へのトルク制限を自動開始します。

使用上の注意

他社製サーボドライバを使用する場合、MC_Home（原点復帰）命令での自動トルク制限機能は機能しません。
MC_SetTorqueLimit（トルク制限）命令、SDO 通信、サポートツール等を使用して、適切な値を設定してください。

参考

- 原点復帰の正常完了後もトルク制限は継続します。
- 逆方向へ移動する命令が起動された場合には、トルク制限が自動的に解除されます。

オムロン製サーボドライバ 1S シリーズの設定

押し当て動作を使用するには、Sysmac Studio を用いて 1S シリーズ サーボドライバの［トルク制限］の［切り替え選択］(3330Hex-01Hex)を設定する必要があります。

- 原点復帰の押し当て動作時に原点検出方向に対してトルク制限を行い、かつそのほかの動作は MC_SetTorqueLimit（トルク制限）命令のトルク制限方向やトルク制限値を使用する場合には「2」を使用してください。
  この場合、原点復帰の押し当て動作中は MC_SetTorqueLimit（トルク制限）命令の入力変数の内容は無視されます。
- 「0」を設定した場合、常に MC_SetTorqueLimit（トルク制限）命令の入力変数の内容が適用されます。原点復帰の押し当て動作時とそれ以外で適切なトルク制限値を設定していただく必要があります。

		［トルク制限］の［切り替え選択］(3330Hex-01Hex)	
		2	0
正方向トルク制限値	原点復帰中	サーボドライバの［トルク制限］の［正方向トルク制限値 2］(3330Hex-05Hex)が適用されます。	MC_SetTorqueLimit（トルク制限）命令の「PositiveValue」（正方向トルク制限値）が適用されます。
	原点復帰以外	MC_SetTorqueLimit（トルク制限）命令の「PositiveValue」（正方向トルク制限値）が適用されます。	
負方向トルク制限値	原点復帰中	サーボドライバの［トルク制限］の［負方向トルク制限値 2］(3330Hex-06Hex)が適用されます。	MC_SetTorqueLimit（トルク制限）命令の「NegativeValue」（負方向トルク制限値）が適用されます。
	原点復帰以外	MC_SetTorqueLimit（トルク制限）命令の「NegativeValue」（負方向トルク制限値）が適用されます。	

トルク制限については、「MC_SetTorqueLimit (P.3-326)」を参照してください。
また、1S シリーズ サーボドライバの設定については、『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-377)』、または『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ 安全機能対応 ユーザーズマニュアル(SBCE-438)』を参照してください。

● 押し当て動作（オムロン製サーボドライバ G5 シリーズを使用する場合）

原点復帰の［12: 近傍退避・押し当て時間指定］、［13: 原点近傍入力なし・押し当て原点入力指定］は、サーボドライバにあらかじめ設定されたトルク制限値を用いて原点検出方向へのトルク制限を自動開始します。

使用上の注意

他社製サーボドライバを使用する場合、MC_Home（原点復帰）命令での自動トルク制限機能は機能しません。
MC_SetTorqueLimit（トルク制限）命令、SDO 通信、サポートツール等を使用して、適切な値を設定してください。

参考

- 原点復帰の正常完了後もトルク制限は継続します。
- 逆方向へ移動する命令が起動された場合には、トルク制限が自動的に解除されます。

オムロン製サーボドライバ G5 シリーズの設定

押し当て動作を使用するには、サーボドライバのサポートソフトを用いてサーボドライバ G5 シリーズのトルクリミット選択(3521 Hex)を設定する必要があります。

- 原点復帰の押し当て動作時に原点検出方向に対してトルク制限を行い、かつそのほかの動作は MC_SetTorqueLimit（トルク制限）命令のトルク制限方向やトルク制限値を使用する場合には「6」を使用してください。
  この場合、原点復帰の押し当て動作中は MC_SetTorqueLimit（トルク制限）命令の入力変数の内容は無視されます。
- 「4」を設定した場合、常に MC_SetTorqueLimit（トルク制限）命令の入力変数の内容が適用されます。原点復帰の押し当て動作時とそれ以外で適切なトルク制限値を設定していただく必要があります。

		トルクリミット選択(3521 Hex)	
		6（推奨）	4
正方向トルク制限値	原点復帰中	第 3 トルクリミット(3525 Hex)が適用されます。	MC_SetTorqueLimit（トルク制限）命令の「PositiveValue」(正方向トルク制限値）と、第 1 トルクリミット(3013 Hex)の小さい方が適用されます。
	原点復帰以外	MC_SetTorqueLimit（トルク制限）命令の「PositiveValue」(正方向トルク制限値）と、第 1 トルクリミット(3013 Hex)の小さい方が適用されます。	
負方向トルク制限値	原点復帰中	第 4 トルクリミット(3526 Hex)が適用されます。	MC_SetTorqueLimit（トルク制限）命令の「NegativeValue」(負方向トルク制限値）と、第 2 トルクリミット(3522 Hex)の小さい方が適用されます。
	原点復帰以外	MC_SetTorqueLimit（トルク制限）命令の「NegativeValue」(負方向トルク制限値）と、第 2 トルクリミット(3522 Hex)の小さい方が適用されます。	

トルク制限については、「MC_SetTorqueLimit (P.3-326)」を参照してください。
また、サーボドライバ G5 シリーズの設定については、『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-365)』、『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプ ユーザーズマニュアル(SBCE-366)』を参照してください。

● 押し当て動作中の位置偏差監視

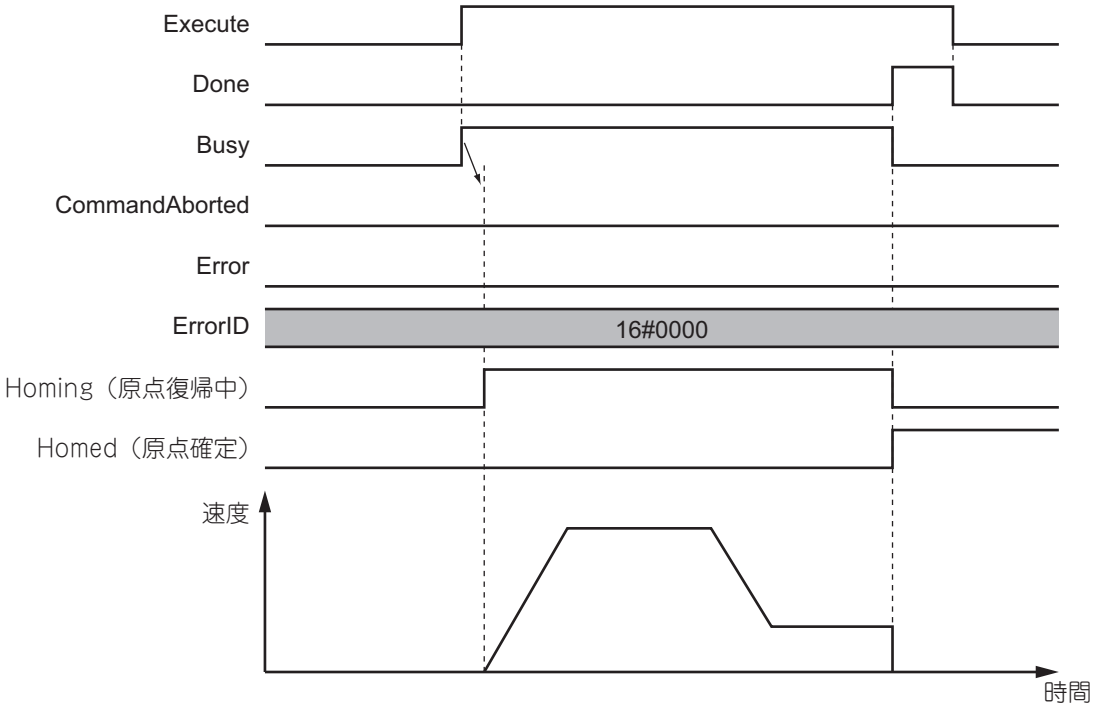
原点復帰の押し当て動作中は、位置偏差監視機能が機能しません。

位置偏差監視機能の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

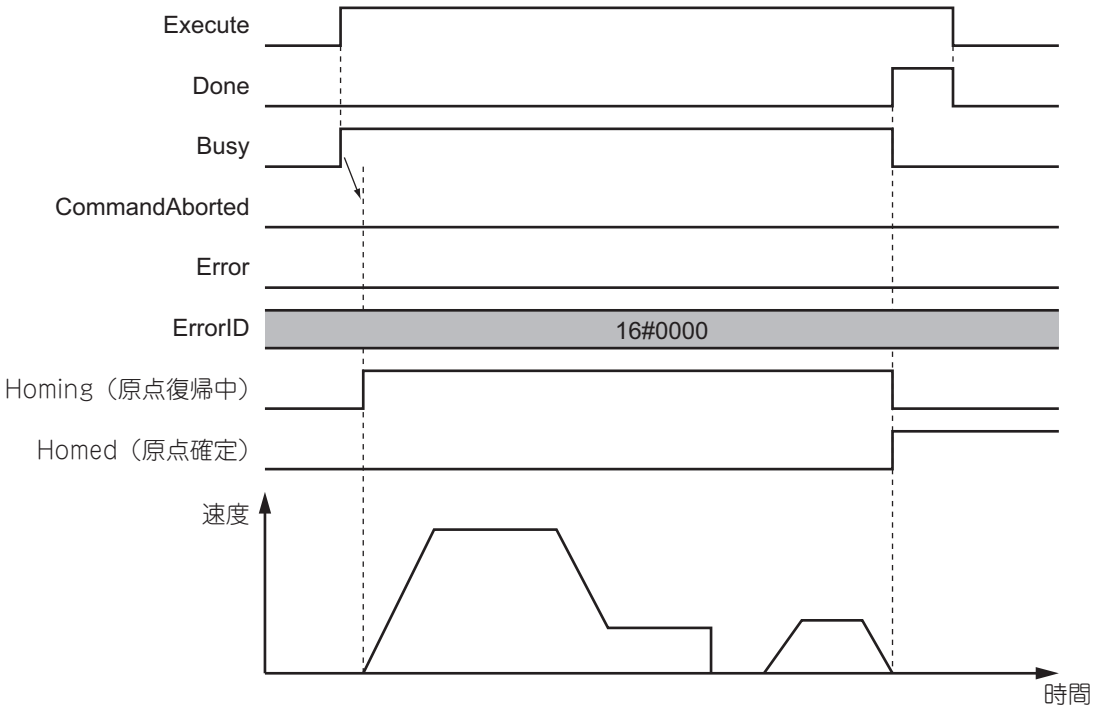
タイミングチャート

原点復帰を実行した場合のタイミングチャートは、次のようになります。

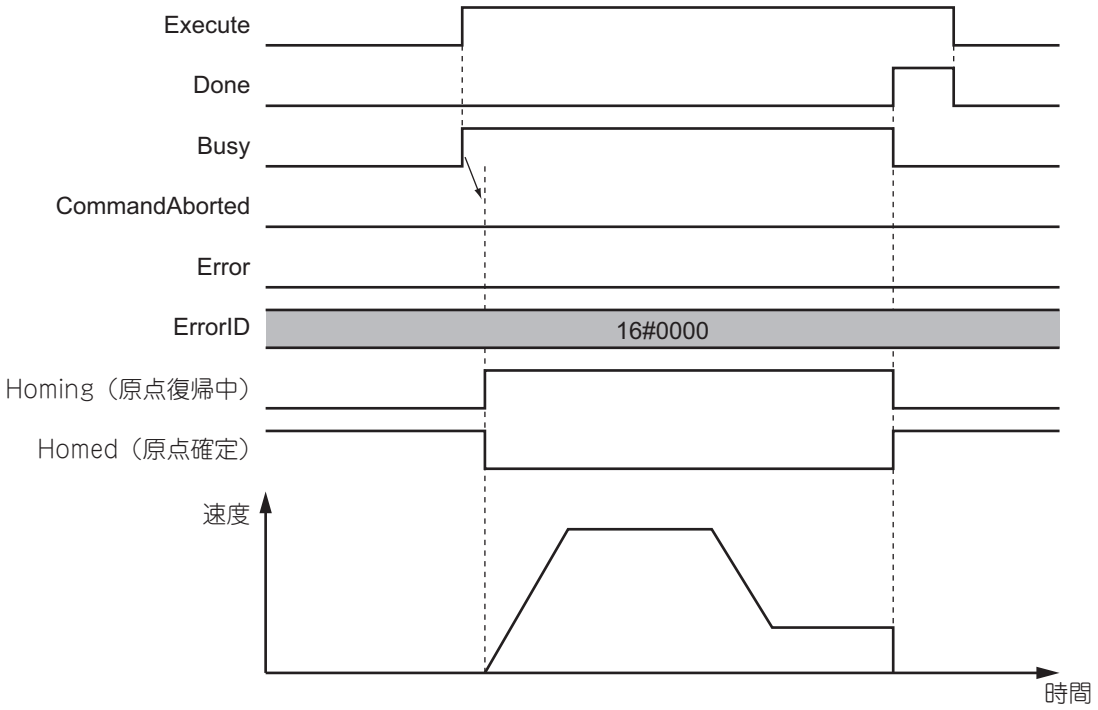
● 原点補正がない場合



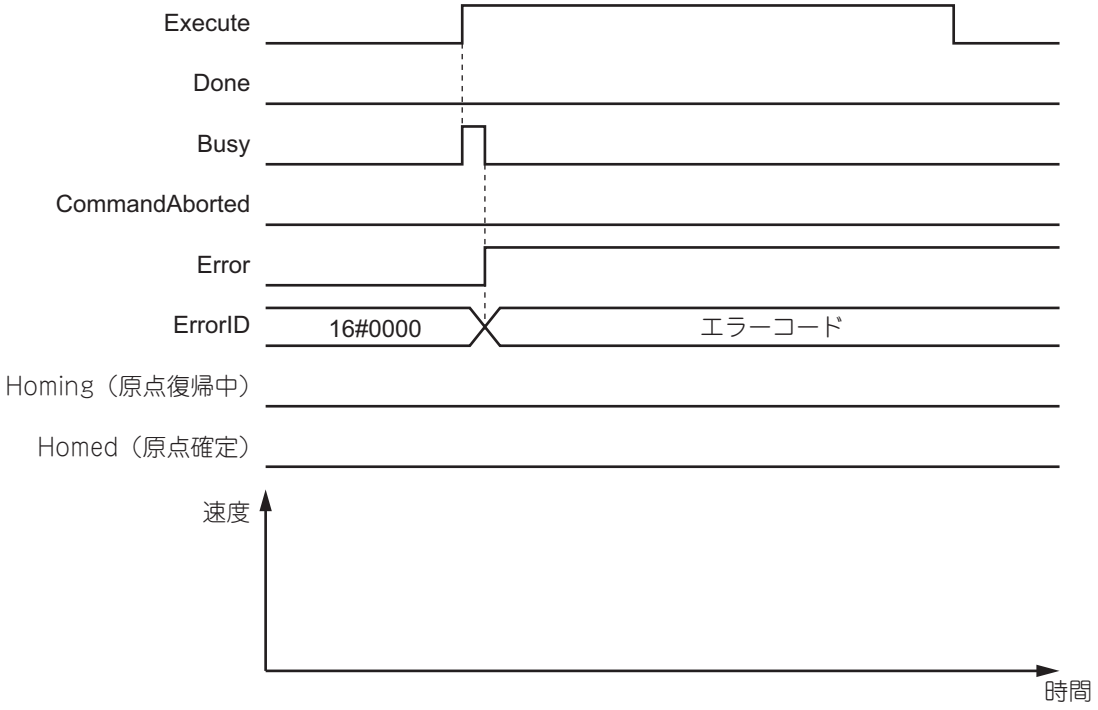
● 原点補正がある場合



● **原点確定状態で本命令を実行した場合**



● **パラメータが不正な状態で本命令を実行した場合、およびモーション命令実行不可の状態で本命令を実行した場合**



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

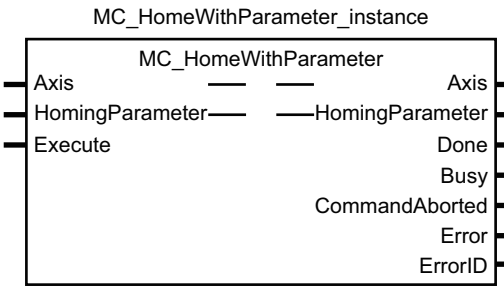
モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_HomeWithParameter

原点復帰パラメータを指定し、モータを動かして、限界信号、原点近傍信号、原点信号を用いて機械原点を決めます。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_HomeWithParameter	パラメータ指定原点復帰	FB		MC_HomeWithParameter_instance (Axis :=「パラメータ」, HomingParameter :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

バージョン関連情報

本命令は、ユニットバージョン Ver.1.03 以降の CPU ユニットと Sysmac Studio Ver.1.04 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1
HomingParameter	原点復帰パラメータ	_sHOMING_REF	−	原点復帰パラメータを設定します。*2

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. _sHOMING_REF 型のユーザ定義変数を定義してください。

● _sHOMING_REF

メンバ変数	名称	データ型	有効範囲	機能
HomingMode	原点復帰動作	_eMC_HOMING_MODE	0: _mcHomeSwTurnHomeSwOff 1: _mcHomeSwTurnHomeSwOn 4: _mcHomeSwOff 5: _mcHomeSwOn 8: _mcLimitInputOff 9: _mcHomeSwTurnHomeMask 11: _mcLimitInputOnly 12: _mcHomeSwTurnHoldingTime 13: _mcNoHomeSwHoldingHomeInput 14: _mcHomePreset	書き換える原点復帰動作を指定します。 0: 近傍退避・原点近傍入力 OFF 指定 1: 近傍退避・原点近傍入力 ON 指定 4: 原点近傍入力 OFF 指定 5: 原点近傍入力 ON 指定 8: 限界入力 OFF 指定 9: 近傍退避・原点入力マスク距離指定 11: 限界入力のみ 12: 近傍退避・押し当て時間指定 13: 原点近傍入力なし・押し当て原点入力指定 14: 原点プリセット

メンバ変数	名称	データ型	有効範囲	機能
HomeInput	原点入力信号	_eMC_HOME_INPUT	0: _mcZPhase 1: _mcExternalSignal	原点入力信号を選択します。 0: Z 相入力を使用する。 1: 外部原点入力を使用する*1
StartDir	原点復帰開始方向	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	原点復帰起動時の開始方向を設定します。 0: 正方向 2: 負方向
HomeDir	原点検出方向	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	原点復帰の原点入力検出方向を設定します。 0: 正方向 2: 負方向
PosiLmtMode	正方向限界入力時動作	_eMC_LIMIT_REVERSE_MODE	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDecelerationStop	原点復帰中に正方向限界入力が「ON」した場合の停止方法を設定します。 0: 反転しない・軸の異常で停止（停止方法は限界入力停止方法に従う） 1: 反転する・即停止 2: 反転する・減速停止
NegaLmtMode	負方向限界入力時動作	_eMC_LIMIT_REVERSE_MODE	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDecelerationStop	原点復帰中に負方向限界入力が「ON」した場合の停止方法を設定します。 0: 反転しない・軸の異常で停止（停止方法は限界入力停止方法に従う） 1: 反転する・即停止 2: 反転する・減速停止
Vel	原点復帰速度	LREAL	正数	原点復帰中の速度を設定します。 最高速度以下、かつ原点復帰アプローチ速度≦原点復帰速度となるように設定してください。 単位は［指令単位／s］です。*2
ApproachVel	原点復帰アプローチ速度	LREAL	正数	原点近傍入力が「ON」した後の速度を設定します。 最高速度以下、かつ原点復帰アプローチ速度≦原点復帰速度となるように設定してください。 単位は［指令単位／s］です。*2
Acc	原点復帰加速度	LREAL	正数、または「0」	原点復帰時の加速度を指定します。 「0」を設定すると加速なしとなります。 単位は［指令単位／s^2］です。*2
Dec	原点復帰減速度	LREAL	正数、または「0」	原点復帰時の減速度を指定します。 「0」を設定すると減速なしとなります。 単位は［指令単位／s^2］です。*2
Jerk	原点復帰ジャーク	LREAL	正数、または「0」	原点復帰時のジャークを指定します。 「0」を設定するとジャークなしとなります。 パルス単位 40 ビット以内で設定してください。 単位は［指令単位／s^3］です。*2
Mask	原点入力マスク距離	LREAL	正数、または「0」	原点復帰動作モードが［近傍退避・原点入力マスク距離指定］の場合の原点入力のマスク距離を設定します。 単位は［指令単位］です。*2
Offset	原点位置オフセット	LREAL	負数、正数、「0」	原点復帰完了時に設定した値で現在位置をプリセットします。 ロータリモード時、リングカウンタ下限値≦原点位置オフセット＜リングカウンタ上限値となるように設定してください。また、パルス単位 40 ビット以内で設定してください。 単位は［指令単位］です。*2

メンバ変数	名称	データ型	有効範囲	機能
PushTime	原点復帰押し当て時間	UINT	0～10,000	原点復帰動作モードが［近傍退避・押し当て時間指定］の場合の押し当て時間をミリ秒単位で設定します。 単位は[ms]です。
Compensation	原点復帰補正値	LREAL	負数、正数、「0」	原点復帰の原点確定後の原点復帰補正量を設定します。 ロータリモード時、｜原点復帰補正値｜＜｜リングカウンタ上限値－リングカウンタ下限値｜となるように設定してください。また、パルス単位 40 ビット以内で設定してください。 単位は［指令単位］です。*2
CompensationVel	原点復帰補正速度	LREAL	正数	原点復帰補正時の速度を設定します。 最高速度以下となるように設定してください。 単位は［指令単位／s］です。*2

*1. オムロン製サーボドライバ 1S シリーズおよび G5 シリーズを使用する場合に指定できます。サーボドライバのラッチ機能 1 に割り付けられた入力を外部原点入力として使用します。サーボドライバ 1S シリーズおよび G5 シリーズの初期状態では、ラッチ機能 1 に外部ラッチ入力 1 が割り付けられています。
詳細については、『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-377)』、『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ 安全機能対応 ユーザーズマニュアル(SBCE-438)』、『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-365)』、および『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプ ユーザーズマニュアル(SBCE-366)』を参照してください。

*2. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

機能説明

- 「Axis」（軸）で指定された軸に対して、「Execute」（起動）の立ち上がりで原点復帰動作を開始します。
- MC_HomeWithParameter（パラメータ指定原点復帰）命令で使用される各種パラメータは、入出力変数「HomingParameter」で設定します。
- 本命令を実行することにより、軸パラメータが変更されることはありません。
- 本命令が、MC_Home（原点復帰）命令と異なるのは、各種パラメータの設定方法だけです。本命令は、各種パラメータを入出力変数「HomingParameter」で設定します。MC_Home（原点復帰）命令は、各種パラメータを Sysmac Studio の軸パラメータ［原点復帰動作］で設定します。

各種パラメータの設定方法以外の命令仕様は、「MC_Home (P.3-16)」を参照してください。

使用上の注意

本命令で指定した原点復帰パラメータは、本命令の実行時だけ有効です。本命令実行後に MC_Home（原点復帰）命令を実行した場合は、軸パラメータで設定された原点復帰パラメータに従って動作します。

原点復帰動作と原点復帰パラメータの関係

原点復帰パラメータには、原点復帰動作の設定によって使用しないものがあります。使用しないパラメータは、範囲チェックや整合チェックの対象外となります。範囲チェックや整合チェックは、本命令実行時に実行されます。

原点復帰動作の設定に対して、各原点復帰パラメータを使用するかしないかは、以下の表のとおりです。

（O: パラメータを使用する、 ー: パラメータを使用しない）

	原点復帰パラメータ														
原点復帰動作	原点入力信号	原点復帰開始方向	原点検出方向	正方向限界入力時動作	負方向限界入力時動作	原点復帰速度	原点復帰アプローチ速度	原点復帰加速度	原点復帰減速度	原点復帰ジャーク	原点入力マスク距離	原点位置オフセット	原点復帰押し当て時間	原点復帰補正値	原点復帰補正速度
近傍退避･原点近傍入力 OFF 指定	O	O	O	O	O	O	O	O	O	O	ー	O	ー	O	O
近傍退避･原点近傍入力 ON 指定	O	O	O	O	O	O	O	O	O	O	ー	O	ー	O	O
原点近傍入力 OFF 指定	O	O	O	O	O	O	O	O	O	O	ー	O	ー	O	O
原点近傍入力 ON 指定	O	O	O	O	O	O	O	O	O	O	ー	O	ー	O	O
限界入力 OFF 指定	O	O	O	O	O	O	O	O	O	O	ー	O	ー	O	O
近傍退避･原点入力マスク距離指定	O	O	O	O	O	O	O	O	O	O	O	O	ー	O	O
限界入力のみ	ー	O	O	O	O	O	O	O	O	O	ー	O	ー	O	O
近傍退避･押し当て時間指定	ー	O	O	O	O	O	O	O	O	O	ー	O	O	O	O
原点近傍入力なし･押し当て原点入力指定	O	O	O	O	O	O	O	O	O	O	ー	O	ー	O	O
原点プリセット	ー	ー	ー	ー	ー	ー	ー	ー	ー	ー	ー	O	ー	ー	ー

参考

NX シリーズ 位置インタフェースユニットを使用する場合、押し当てを使用する原点復帰動作モードを選択できません。詳細は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。

モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex)」となります。

モーション命令多重起動

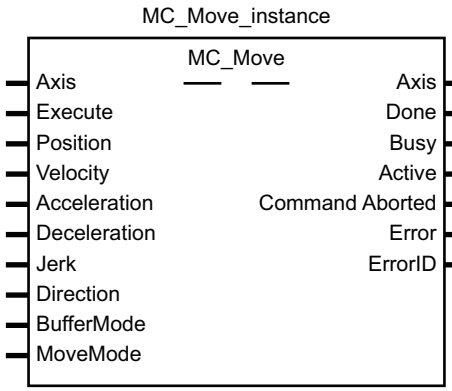
モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_Move

絶対値位置決め、または相対値位置決めを行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_Move	位置決め	FB		MC_Move_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, Position :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, Direction :=「パラメータ」, BufferMode :=「パラメータ」, MoveMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」 , Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Position	目標位置	LREAL	負数、正数、「0」	0	移動方法選択に絶対位置決めを指定する場合は絶対座標の目標位置を指定します。 移動方法選択に相対位置決めを指定する場合は相対位置を指定します。 単位は、［指令単位］となります。*1
Velocity	目標速度	LREAL	正数	0	目標速度を指定します。*2 単位は、［指令単位／s］となります。*1
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は、［指令単位／s^2］となります。*1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は、［指令単位／s^2］となります。*1

入力変数	名称	データ型	有効範囲	初期値	内容
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は、[指令単位／s^3] となります。*1
Direction	方向選択	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	0 *3	「MoveMode」が［0: 絶対値位置決め］*4 で、カウントモードが［ロータリモード］の場合に、回転方向を指定します。 0: 正方向指定 1: 近回り指定 2: 負方向指定 3: 現在方向指定 4: 方向指定なし
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2: ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ
MoveMode	移動方法選択	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative	0 *3	移動方法を選択します。 0: 絶対値位置決め 1: 相対値位置決め

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 目標速度は必ず設定してください。設定せずに動作させると目標速度設定範囲外（エラーコード: 5422Hex）が発生します。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*4. 「MoveMode」が［1: 相対値位置決め］のときは、Position の符号で移動方向が決まります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位置決めが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後

変数	TRUE となるタイミング	FALSE となるタイミング
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 絶対値位置決め、または相対値位置決めを行うことができます。
- 「MoveMode」（移動方法選択）に［絶対値位置決め］を指定した場合は、MC_MoveAbsolute（絶対値位置決め）命令と同じ動作となります。
  同様に［相対値位置決め］を指定した場合は、MC_MoveRelative（相対値位置決め）命令と同じ動作となります。
- 相対値位置決め命令として動作する場合は、「Direction」（方向選択）は使用しません。

詳細については、「MC_MoveAbsolute (P.3-47)」、または「MC_MoveRelative (P.3-73)」を参照してください。

命令の詳細

本命令の詳細について説明します。

使用上の注意

絶対値位置決めを行う場合は、軸パラメータの［リングカウンタ上限設定値］、および［リングカウンタ下限設定値］の範囲内に目標位置を指定してください。
目標位置を［リングカウンタ上限設定値］と［リングカウンタ下限設定値］の範囲外に指定した場合、「目標位置設定範囲外（エラーコード: 5478Hex)」となります。

● インポジションチェック

本命令のインポジションチェックは、軸パラメータの［インポジション幅］、［インポジションチェック時間］の設定に従って行われます。

モーション命令再起動

位置決め動作中に入力変数を変更し、「Execute」（起動）を再度 TRUE にすることにより本命令の動作を変更することができます。
モーション命令再起動で変更可能な入力変数は、「Position」（目標位置）、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）です。

モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して、本命令を起動することにより、本命令への切り替えや本命令へのバッファリングを行うことができます。
各軸に 1 つまでバッファリングすることが可能です。
多重起動時の本命令の動作を、「BufferMode」（バッファモード選択）で指定します。

バッファモード選択		説明
アボーティング		現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。
バッファード		現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。
ブレンディング		現在実行中の命令が目標位置に到達したときの速度（中継速度）を起動速度として、バッファリングされている本命令を連続して動作させます。中継速度で目標位置に到達するように、現在実行中の命令の動作を変更します。中継速度の指定方法は、以下の 4 種類があります。
	ブレンディングロウ	現在実行中の目標速度とバッファリングされている目標速度のどちらか低い方を中継速度とします。
	ブレンディングプレビアス	現在実行中の目標速度を中継速度とします。
	ブレンディングネクスト	バッファリングされている本命令の目標速度を中継速度とします。
	ブレンディングハイ	現在実行中の目標速度とバッファリングされている目標速度のどちらか高い方を中継速度とします。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

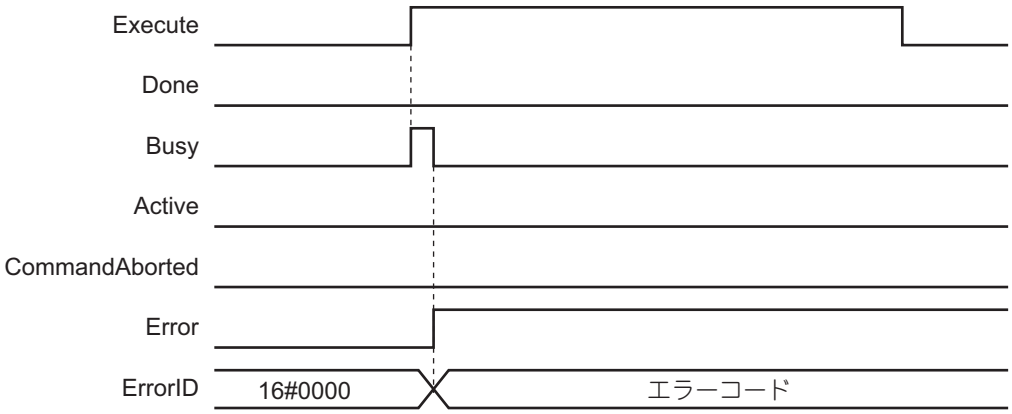
● 本命令実行中の他命令起動

本命令に対するモーション命令多重起動を行う場合、［アボーティング］、［バッファード］、［ブレンディング］が可能です。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_MoveAbsolute

絶対座標の目標位置を指定して、位置決めを行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveAbsolute	絶対値位置決め	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 :=「パラメータ」, Execute :=「パラメータ」, Position :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, Direction :=「パラメータ」, BufferMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Position	目標位置	LREAL	負数、正数、「0」	0	絶対座標の目標位置を指定します。 単位は、[指令単位] となります。*1
Velocity	目標速度	LREAL	正数	0	目標速度を指定します。*2 単位は、[指令単位／s] となります。*1
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は、[指令単位／s^2] となります。*1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は、[指令単位／s^2] となります。*1
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は、[指令単位／s^3] となります。*1

入力変数	名称	データ型	有効範囲	初期値	内容
Direction	方向選択	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	0 *3	カウントモードが［ロータリモード］の場合に、回転方向を指定します。 0: 正方向指定 1: 近回り指定 2: 負方向指定 3: 現在方向指定 4: 方向指定なし
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2: ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 目標速度は必ず設定してください。設定せずに動作させると異常が発生します。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位置決めが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

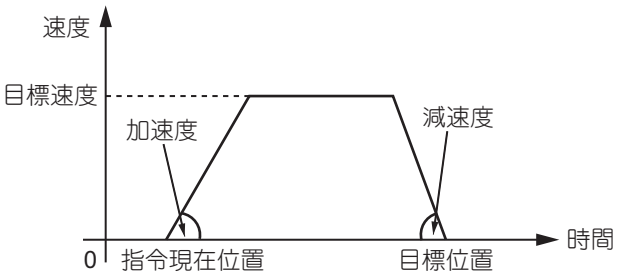
入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

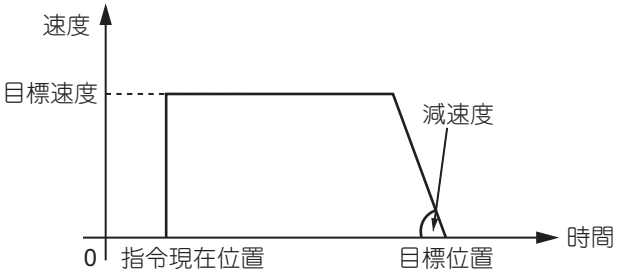
機能説明

- 絶対座標の目標位置を指定して、位置決めを行います。
- 「Execute」（起動）の立ち上がりで、絶対位置決めの動作を開始します。
- 原点未確定でも本命令を実行することができます。
- 入力変数に、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）、「Jerk」（ジャーク）を指定することができます。

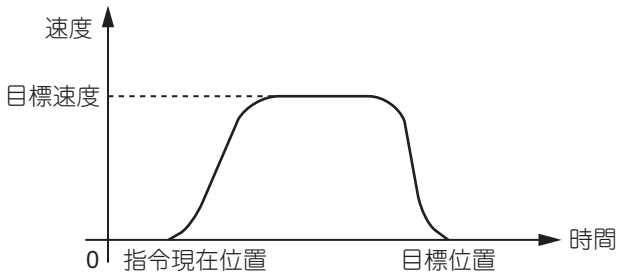
絶対位置決めの動作例を下記に示します。



「Acceleration」（加速度）、または「Deceleration」（減速度）を「0」に指定して起動した場合は、加減速せずに目標速度に到達します。
「Acceleration」（加速度）が「0」の場合の動作例を下記に示します。



なめらかに加減速させたい場合に「Jerk」（ジャーク）を指定します。
「Jerk」（ジャーク）を指定した場合の動作例を下記に示します。



「Jerk」（ジャーク）の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

「Position」（目標位置）を指令現在位置と同じに指定して起動した場合は、軸は移動しませんが「Done」（完了）は TRUE となります。

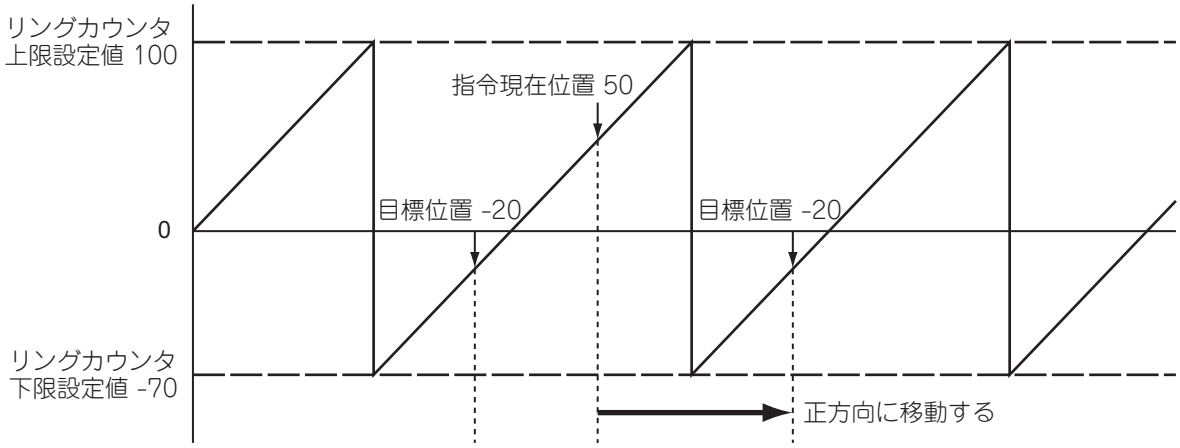
命令の詳細

以下に本命令の詳細について説明します。

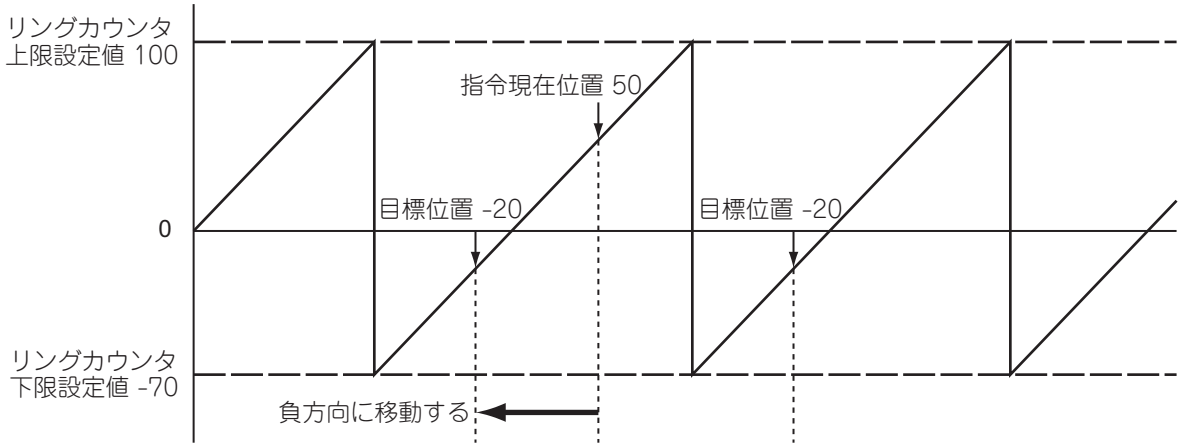
● Direction（方向選択）

カウントモードが［ロータリモード］の場合、「Direction」（方向選択）で位置決めを開始する方向を指定します。
カウントモードが［リニアモード］の場合は、「Direction」（方向選択）は無視されます。

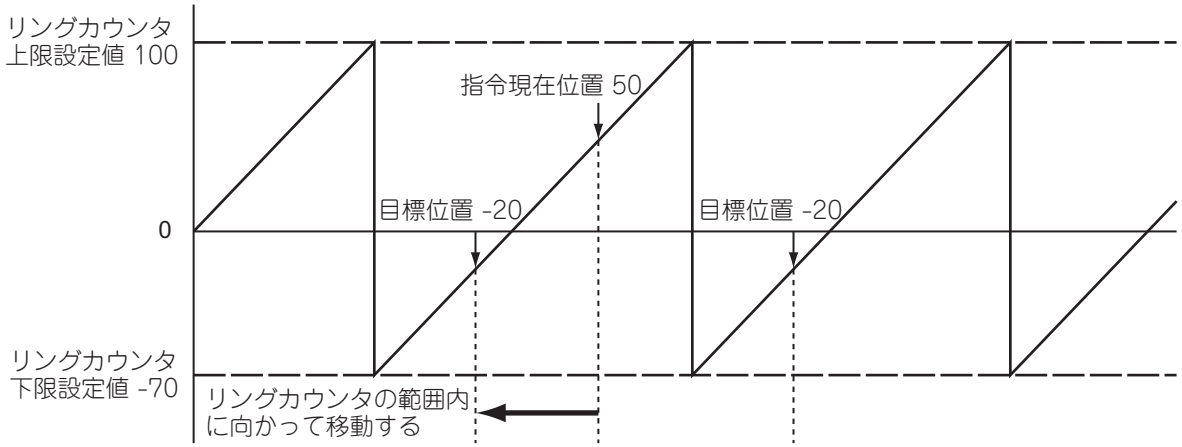
「Direction」（方向選択）が［0: _mcPositiveDirection］（正方向指定）の場合、正方向の目標位置に向かって位置決めを行います。
指令位置が「50」から「-20」に向かって位置決めを行う場合の動作例を下記に示します。



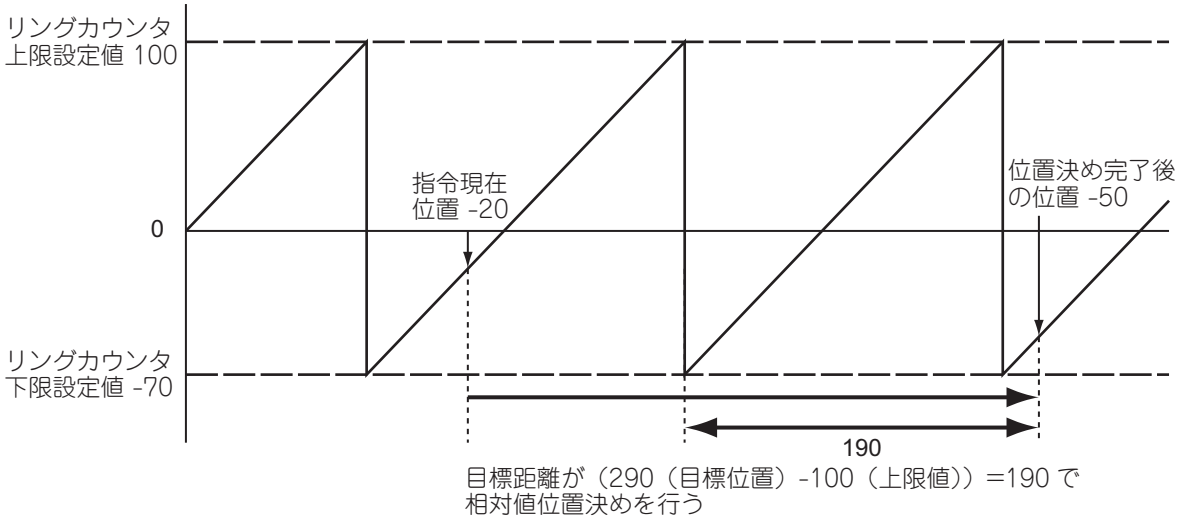
「Direction」（方向選択）が［2: _mcNegativeDirection］（負方向指定）の場合、負方向の目標位置に向かって位置決めを行います。
指令位置が「50」から「-20」に向かって位置決めを行う場合の動作例を下記に示します。



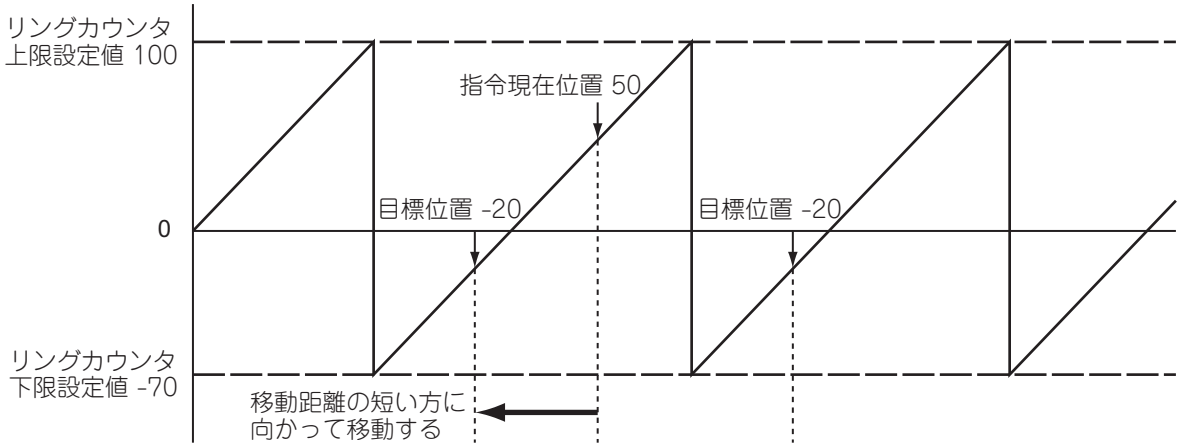
「Direction」（方向選択）が［4: _mcNoDirection］（方向指定なし）の場合、リングカウンタの範囲内の目標位置に向かって位置決めを行います。したがって、指令現在位置と目標位置の大小関係で移動方向が決まります。
指令位置が「50」から「-20」に向かって位置決めを行う場合の動作例を下記に示します。



「Direction」（方向選択）が［4: _mcNoDirection］（方向指定なし）の場合、リングカウンタ上下限設定値の範囲を超えた「Position」（目標位置）の指定が可能です。
リングカウンタ上限設定値を超えた「Position」（目標位置）を指定した場合、リングカウンタ上限設定値を超えた移動量分を超えた後の相対量として位置決めを行います。したがって、リング多回転の位置決めが可能となります。
リングカウンタ下限値設定を超えた「Position」（目標位置）を指定した場合も、同様に位置決めします。
指令現在位置が「-20」で「Position」（目標位置）を「290」に指定した場合の動作例を下記に示します。



「Direction」（方向選択）が［1: _mcShortestWay］（近回り指定）の場合、指令現在位置と目標位置の距離が短い方向に向かって位置決めを行います。
指令位置が「50」から「-20」に向かって位置決めを行う場合の動作例を下記に示します。



正方向の移動距離も負方向の移動距離も同じ場合は、［3: _mcCurrentDirection］（現在方向指定）時と同じ動作になります。

使用上の注意

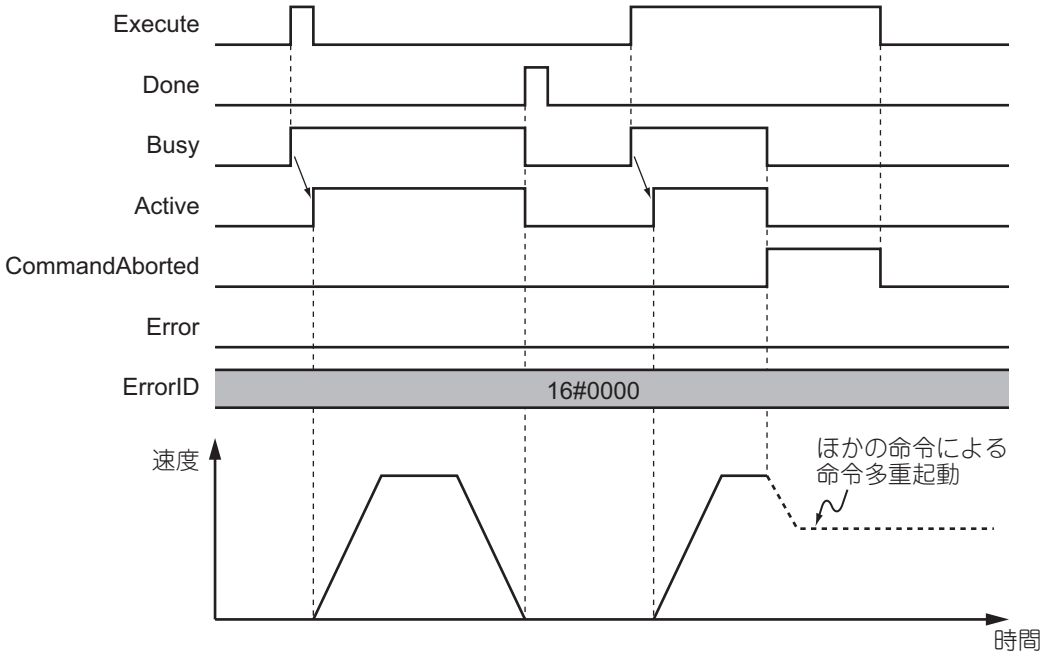
- 絶対値位置決めを行う場合は、軸パラメータの［リングカウンタ上限設定値］未満、および［リングカウンタ下限設定値］以上の範囲内に目標位置を指定してください。
  目標位置を［リングカウンタ上限設定値］以上、または［リングカウンタ下限設定値］未満に指定した場合、「目標位置設定範囲外（エラーコード: 5478Hex）」となります。
  ただし、「Direction」（方向選択）が［4: _mcNoDirection］（方向指定なし）の場合は、リングカウンタ上下限設定値の範囲を超えた「Position」（目標位置）の指定が可能です。
- 「Direction」（方向選択）に［3: _mcCurrentDirection］（現在方向指定）を選択した場合、前の動作における指令方向を踏襲して動作します。そのため命令の組み合わせによっては、前動作のモーション命令の入力で指令した方向と一致しない場合があります。
  ［3: _mcCurrentDirection］（現在方向指定）を使用する場合は、軸変数の「Dir.Posi」（正方向指令中）、および「Dir.Nega」（負方向指令中）で現在の方向を確認してください。

● インポジションチェック

本命令のインポジションチェックは、軸パラメータの［インポジション幅］、［インポジションチェック時間］の設定に従って行われます。

タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- 「Position」（目標位置）に到達し、位置決めが完了すると「Done」（完了）が TRUE となります。
- ほかの命令により本命令が中止された場合は、「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）が FALSE になります。



モーション命令再起動

位置決め動作中に入力変数を変更し、「Execute」（起動）を再度 TRUE にすることにより本命令の動作を変更することができます。
モーション命令再起動で変更可能な入力変数は、「Position」（目標位置）、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）です。

モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して、本命令を起動することにより、本命令への切り替えや本命令へのバッファリングを行うことができます。
各軸に 1 つまでバッファリングすることが可能です。
多重起動時の本命令の動作を、「BufferMode」（バッファモード選択）で指定します。

バッファモード選択		説明
アボーティング		現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。
バッファード		現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。
ブレンディング		現在実行中の命令が目標位置に到達したときの速度（中継速度）を起動速度として、バッファリングされている本命令を連続して動作させます。中継速度で目標位置に到達するように、現在実行中の命令の動作を変更します。中継速度の指定方法は、以下の 4 種類があります。
	ブレンディングロウ	現在実行中の目標速度とバッファリングされている目標速度のどちらか低い方を中継速度とします。
	ブレンディングプレビアス	現在実行中の目標速度を中継速度とします。
	ブレンディングネクスト	バッファリングされている本命令の目標速度を中継速度とします。
	ブレンディングハイ	現在実行中の目標速度とバッファリングされている目標速度のどちらか高い方を中継速度とします。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

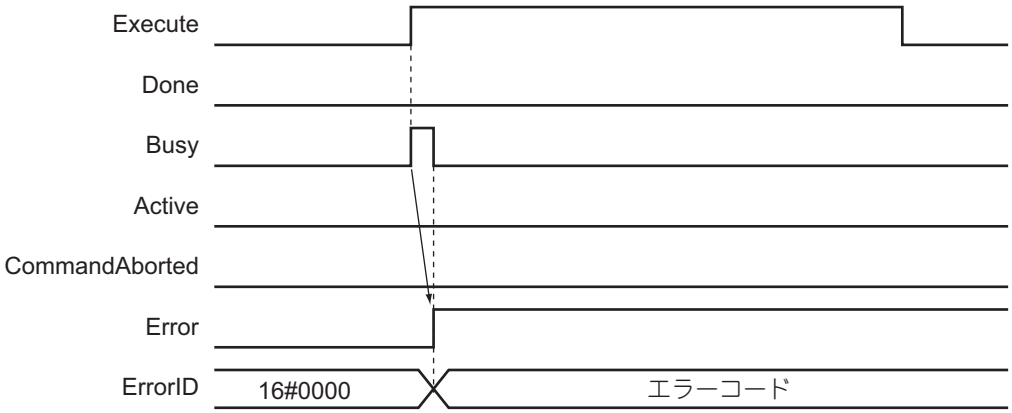
● 本命令実行中の他命令起動

本命令に対するモーション命令多重起動を行う場合、［アボーティング］、［バッファード］、［ブレンディング］が可能です。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム 1

ここでは、多重起動の繰り返しによる位置決め方法についてサンプルプログラムを掲載します。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

軸種別

軸	軸種別
軸 1	サーボ軸

カウンタモード

軸	カウンタモード
軸 1	リニアモード

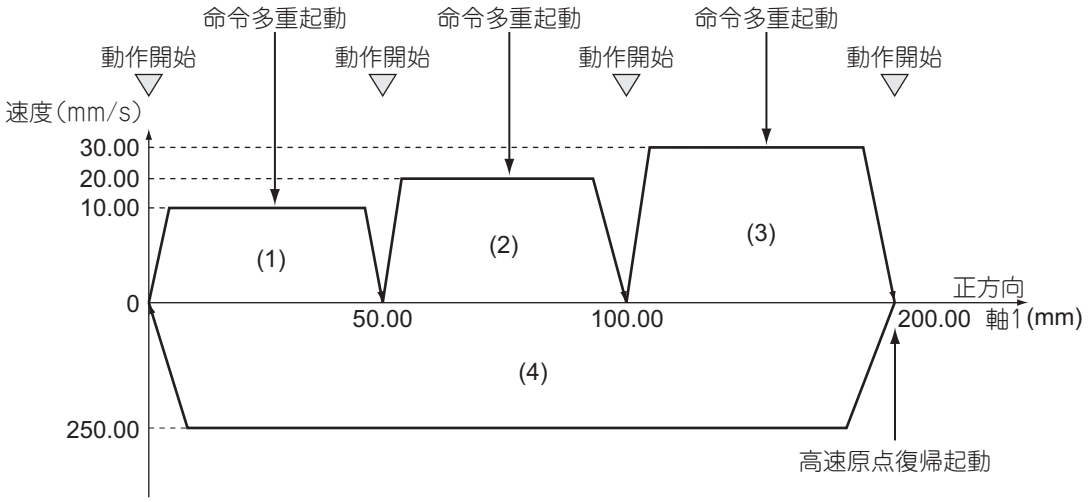
座標単位選択

軸	座標単位
軸 1	mm

動作例

MC_MoveAbsolute（絶対値位置決め）命令を「BufferMode」（バッファモード選択）を［バッファード］にし、モーション命令多重起動しながら、最終目標位置まで移動させます。
最終目標位置に到達したら、MC_ZeroPosition（高速原点復帰）命令で原点に戻します。
多重起動を行う場合は、直前の命令の出力変数「Active」（制御中）が TRUE の状態のときに行います。単軸動作の命令多重起動は 1 命令まで可能です。

● 動作パターン



1 **動作開始スイッチの「ON」**

原点で動作開始スイッチを「ON」 すると、軸 1 が正方向に 50.00mm 位置決めします。

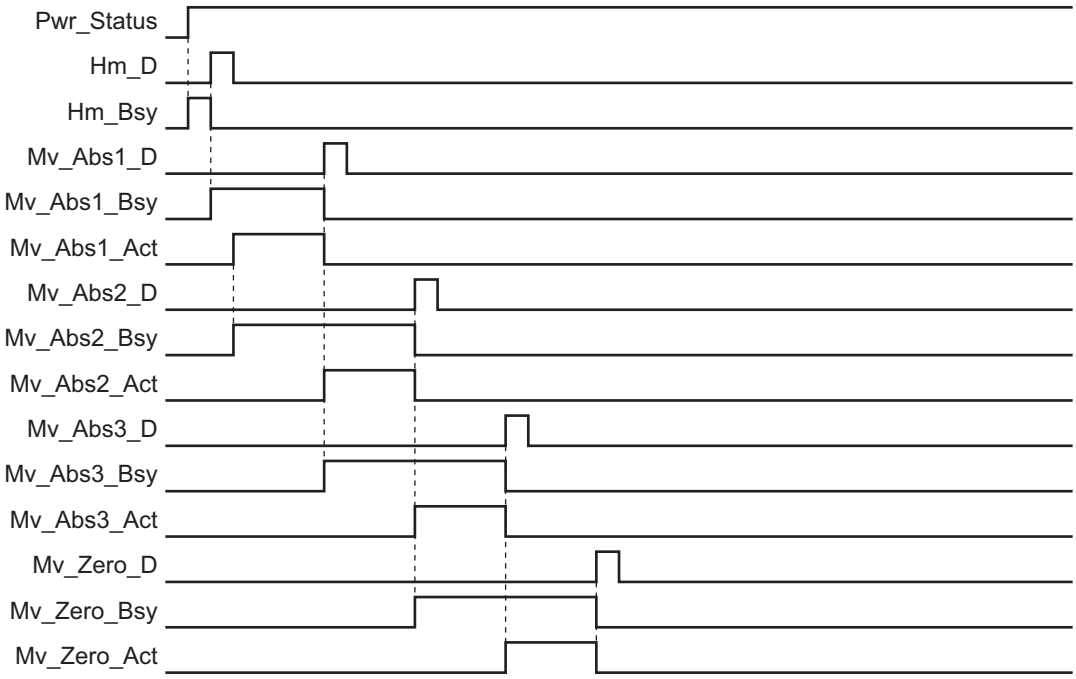
2 **続いて動作開始スイッチの「ON」**
続いて動作開始スイッチを「ON」 するごとに、軸 1 が正方向に 100.00mm、200.00mm に位置決めし、原点に戻って停止します。

ラダー図

● 主な変数

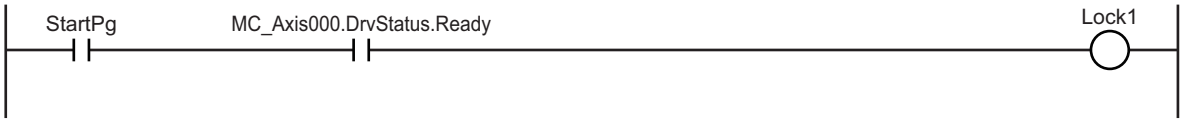
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power のインスタンス PWR の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。

● タイミングチャート

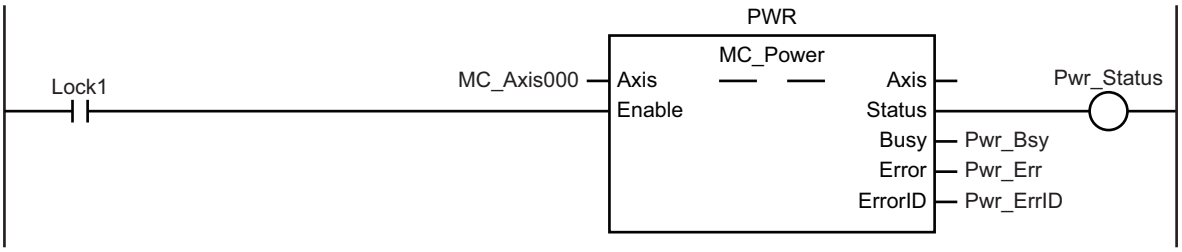


● サンプルプログラム

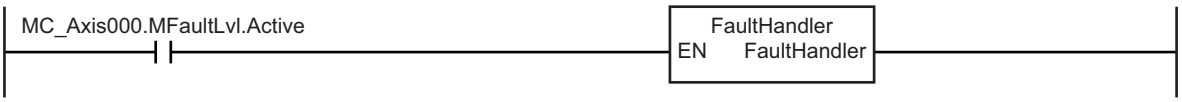
接点 StartPg が TRUE になったら、サーボドライバがサーボレディ状態になっていることを確認する



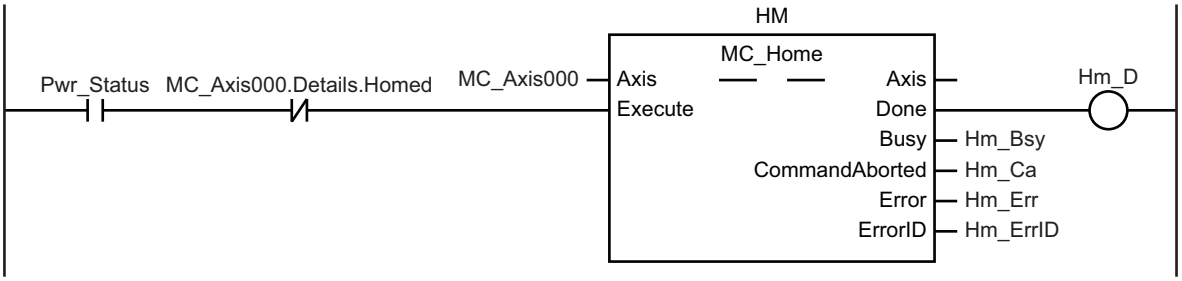
サーボレディ状態になっていれば、サーボ ON 状態にする



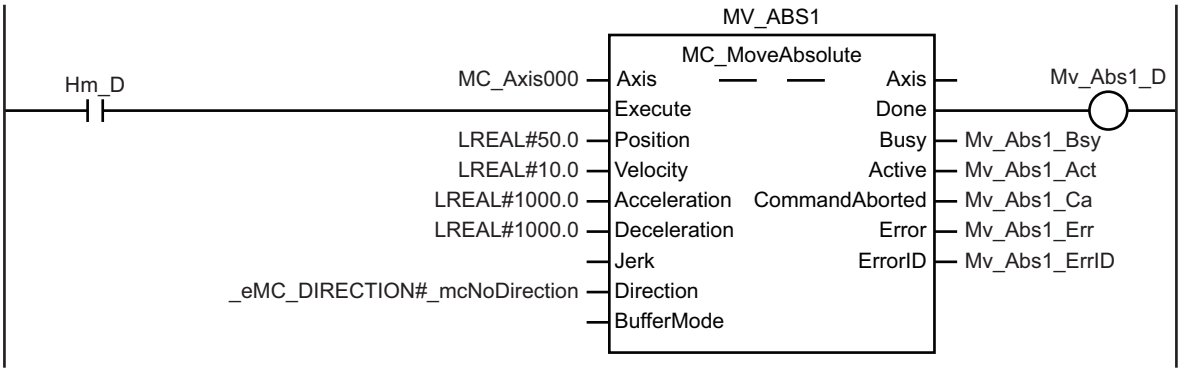
軸 1 で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください。



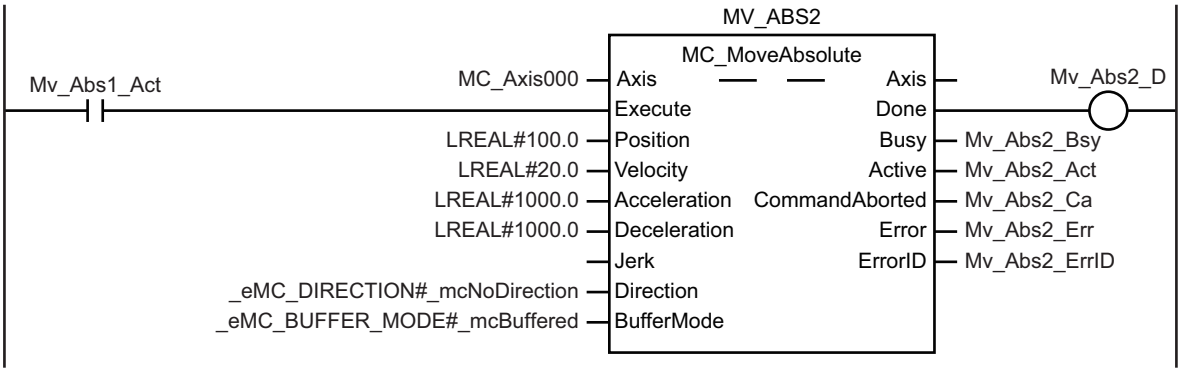
サーボ ON 状態になり原点未確定状態なら、原点復帰を実行する



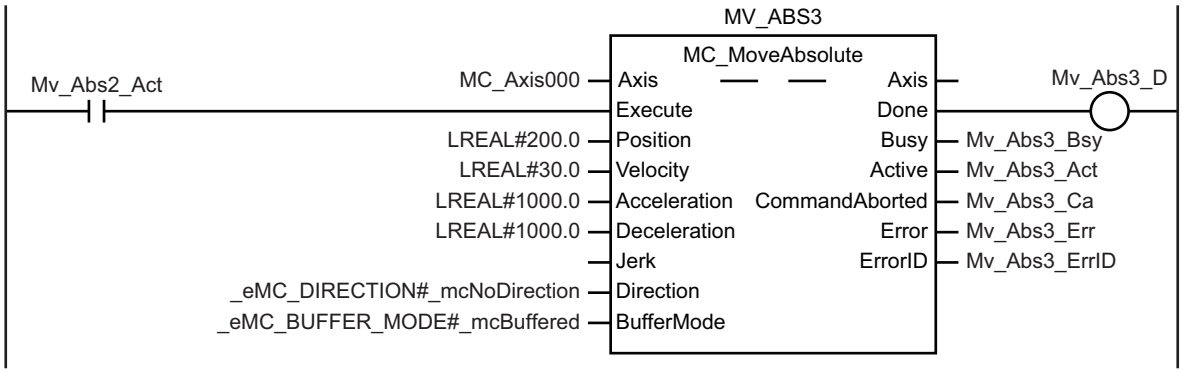
原点確定後、絶対値位置決め (1) を起動する



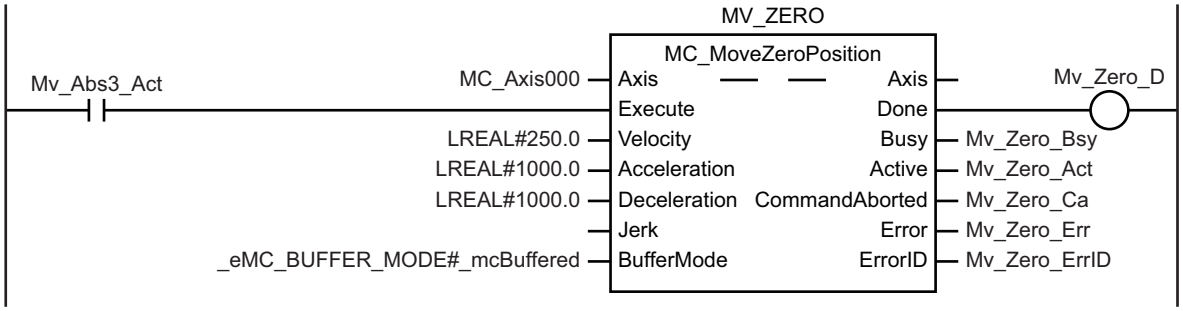
絶対値位置決め (1) が開始された後、絶対値位置決め (2) を多重起動する



絶対値位置決め (2) が開始された後、絶対値位置決め (3) を多重起動する



絶対値位置決め (3) が開始された後、高速原点復帰を実行する



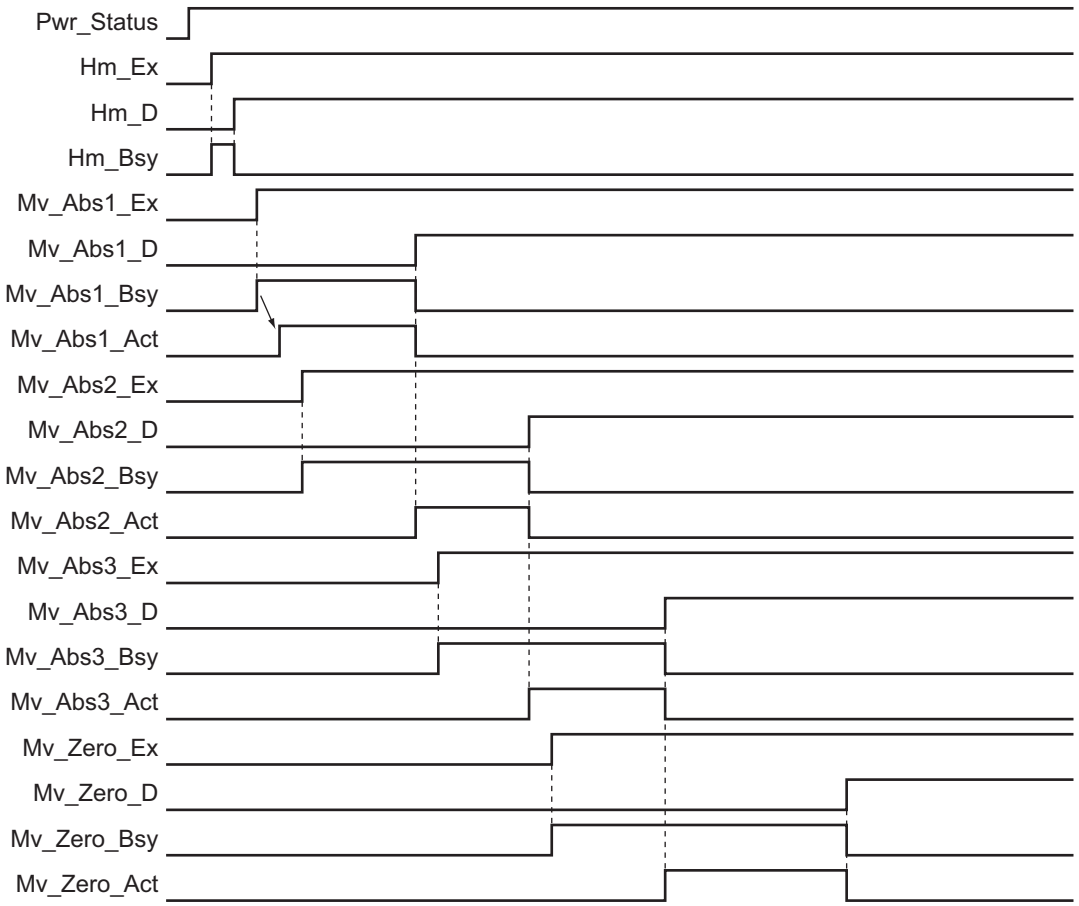
ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power のインスタンス PWR の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Hm_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_Home のインスタンス HM を起動します。
Mv_Abs1_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveAbsolute のインスタンス MV_ABS1 を起動します。
Mv_Abs2_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveAbsolute のインスタンス MV_ABS2 を起動します。
Mv_Abs3_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveAbsolute のインスタンス MV_ABS3 を起動します。
Mv_Zero_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、MC_MoveZeroPosition のインスタンス MV_ZERO を起動します。

名称	データ型	初期値	コメント
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート



● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag = FALSE THEN

    //MV_ABS1 パラメータ
    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 パラメータ
    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 パラメータ
    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 パラメータ
    Mv_Zero_Vel := LREAL#250;
    Mv_Zero_Acc := LREAL#1000.0;
    Mv_Zero_Dec := LREAL#1000.0;
    Mv_Zero_Bm :=_eMC_BUFFER_MODE#_mcBuffered;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag:=TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、サーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//軽度フォールトレベルの異常発生時の処理
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

//サーボ ON 状態で、かつ原点未確定のとき、原点復帰を行います
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

//原点確定後に、MV_ABS1 を実行します
IF Hm_D=TRUE THEN
    Mv_Abs1_Ex:=TRUE;
END_IF;

//MV_ABS1 を開始した後、MV_ABS2 を多重起動します
IF Mv_Abs1_Act=TRUE THEN
    Mv_Abs2_Ex:=TRUE;
END_IF;
```

```
//MV_ABS2 を開始した後、MV_ABS3 を多重起動します
IF Mv_Abs2_Act=TRUE THEN
    Mv_Abs3_Ex:=TRUE;
END_IF;

//MV_ABS3 を開始した後、MV_ZERO を多重起動します
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
);
```

サンプルプログラム 2

カウントモードがロータリモードのとき、目標位置への位置決めを行うのに、左右どちら回りが近いかを自動で判別して位置決めを行います。
ここでは、ツールチェンジャーの近回り制御についてサンプルプログラムを掲載します。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

軸種別

軸	軸種別
軸 1	サーボ軸

カウンタモード

軸	カウンタモード
軸 1	ロータリモード

リングカウンタ

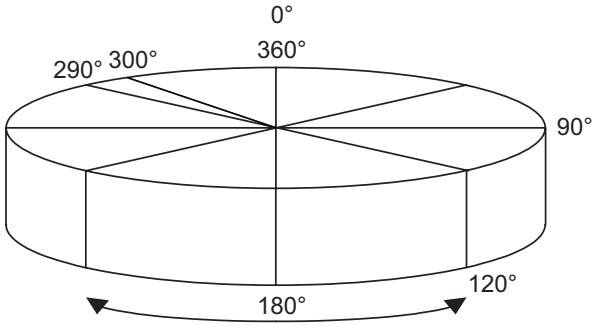
軸	上限値	下限値
軸 1	360	0

座標単位選択

軸	座標単位
軸 1	degree

動作例

絶対値位置決め命令の多重起動を用いて、0～360° の範囲での動作例を示します。なお、現在位置は 0 ～ 360 ° の範囲を超えると 0° に戻ります。
多重起動を行う場合は、直前の命令の出力変数「Active」（制御中） が TRUE の状態のときに行います。単軸動作の命令多重起動は 1 命令まで可能です。
この例では、「BufferMode」（バッファモード選択）を［バッファード］として多重起動させます。



近回りで位置決めします

原点（0° ）、90° 、120° 、290° の 4 つの任意の位置を指定すると、その位置に移動します。
その際の回転方向は、近い方に回転します。なお、移動速度は、250° /s とします。

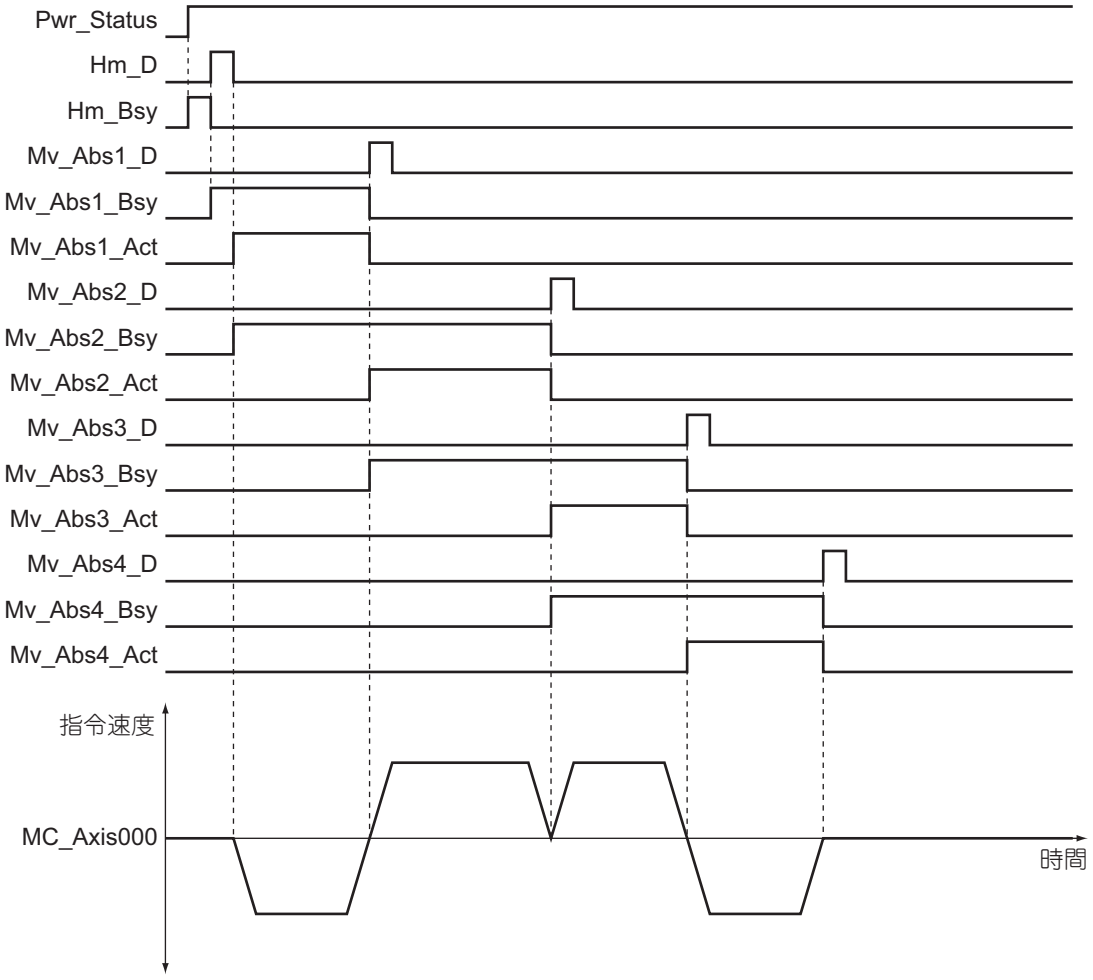
サンプルプログラムでは、290°→90°→120°→原点（0°）に近回り指定で位置決めします。

ラダー図

● 主な変数

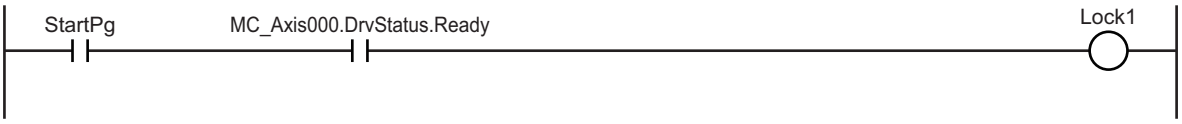
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power のインスタンス PWR の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。

● タイミングチャート

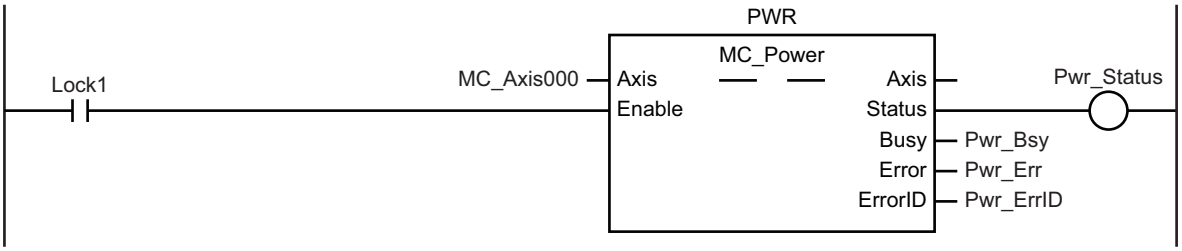


● サンプルプログラム

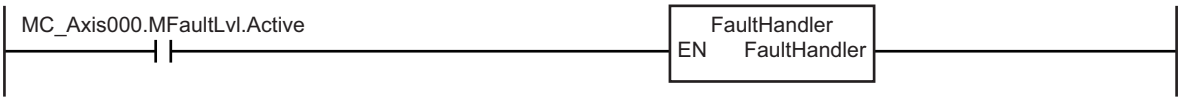
接点 StartPg が TRUE になったら、サーボドライバがサーボレディ状態になっていることを確認する



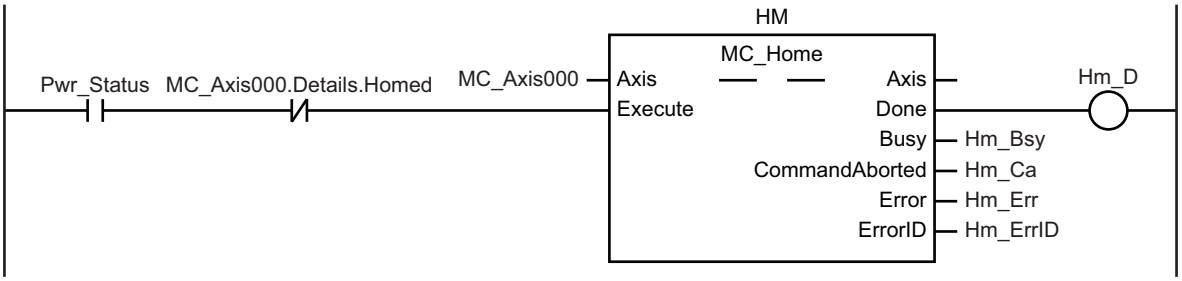
サーボレディ状態になっていれば、サーボ ON 状態にする



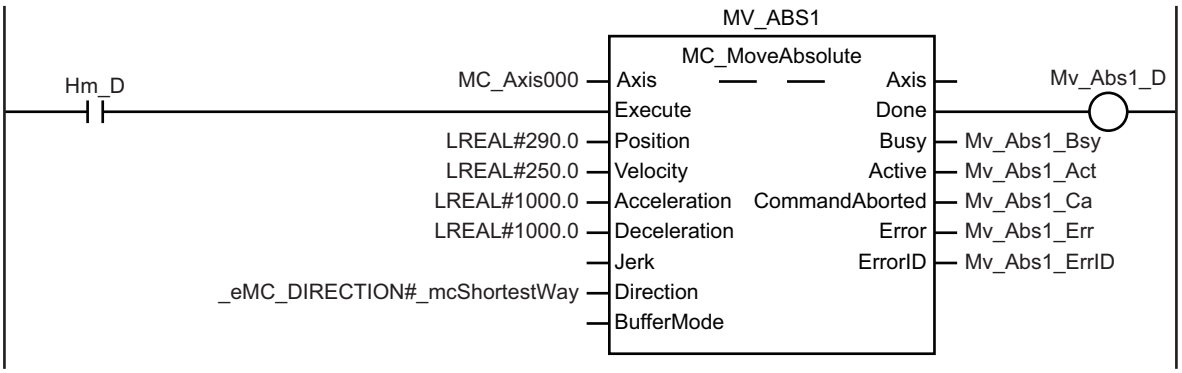
軸 1 で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください。



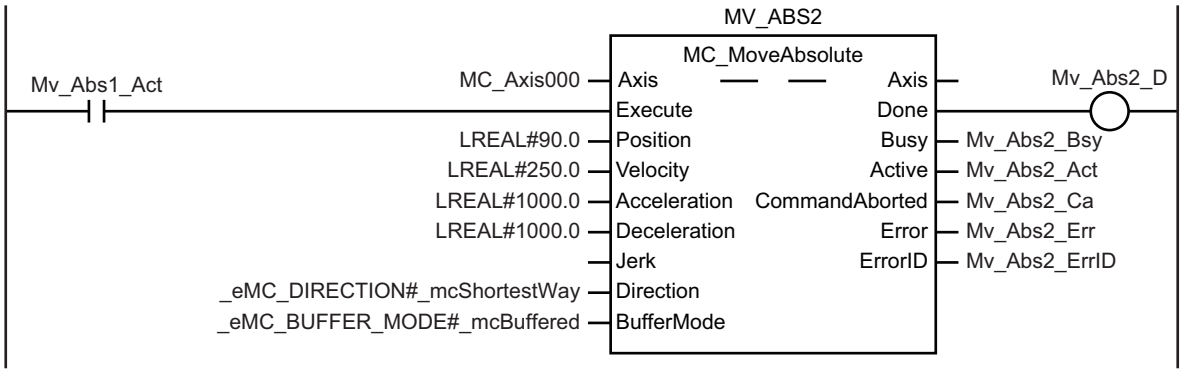
サーボ ON 状態になり原点未確定状態なら、原点復帰を実行する



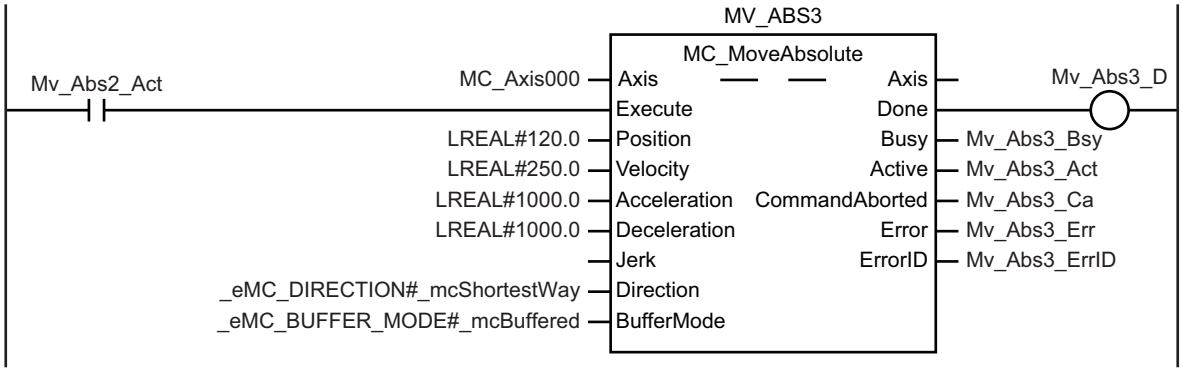
原点確定後、絶対値位置決め (1) を実行し、290.0° に移動する。動作方向は、近回り指定とする



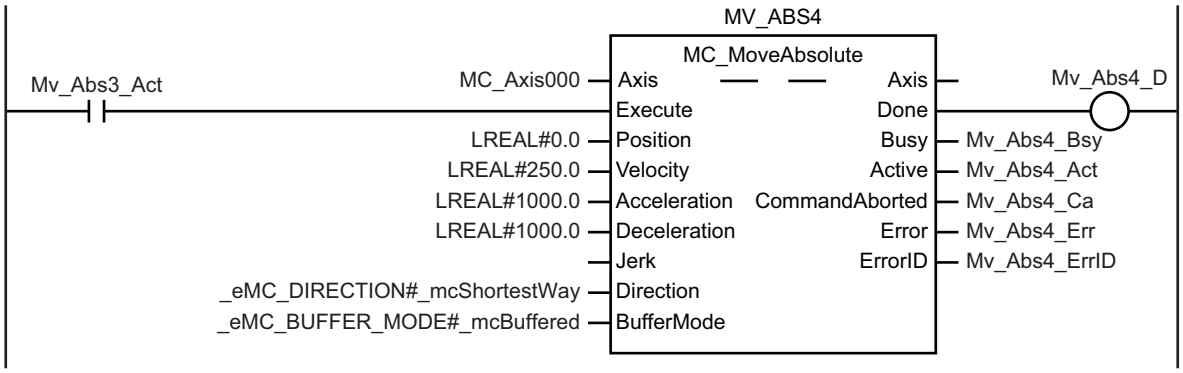
絶対値位置決め (2) を多重起動し、290.0° から 90.0° に移動する。動作方向は近回りとする



絶対値位置決め (3) を多重起動し、90.0° から 120.0° に移動する。動作方向は近回りとする



絶対値位置決め (4) を多重起動し、120.0° から 0.0° に移動する。動作方向は近回りとする



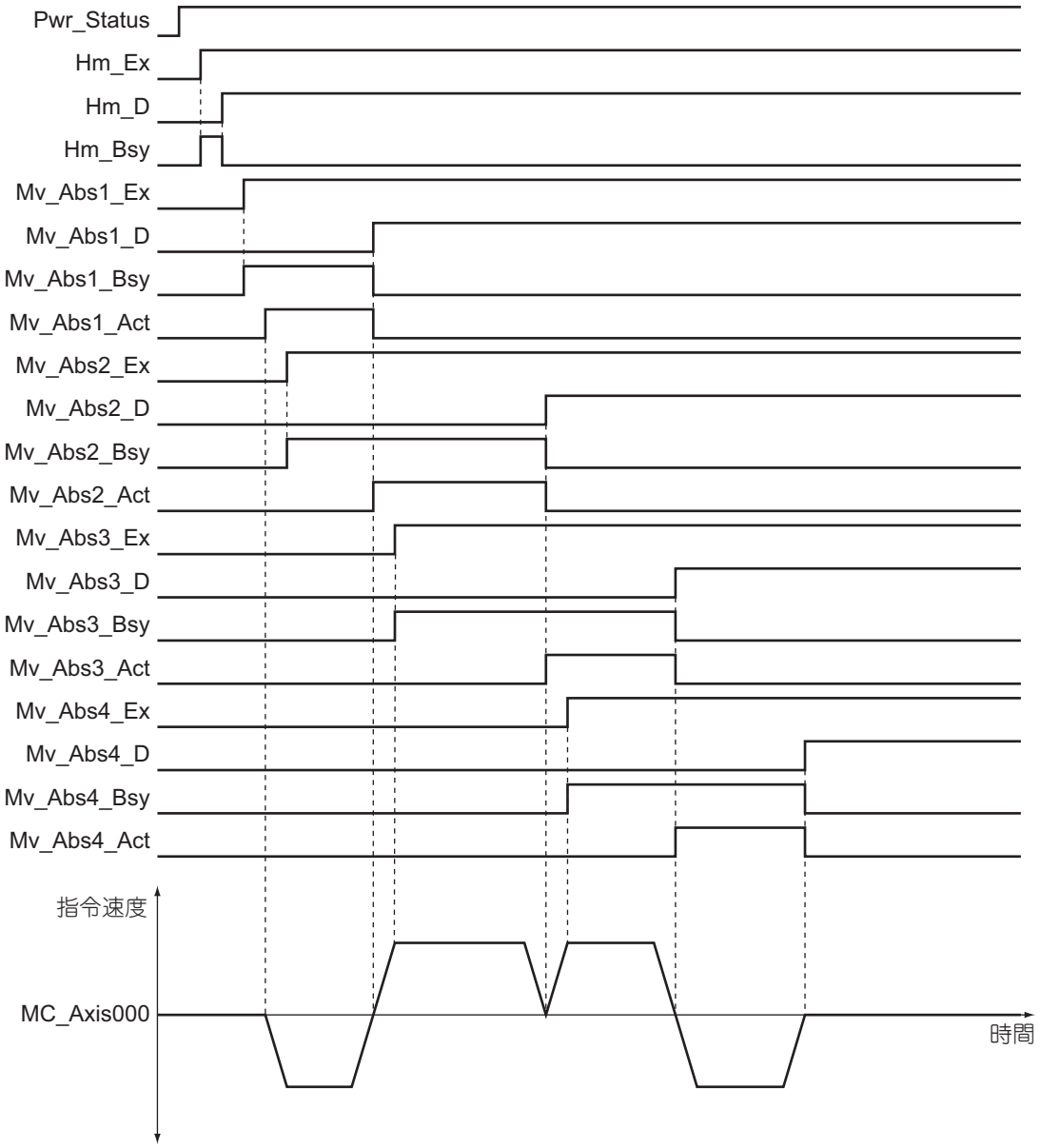
ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power のインスタンス PWR の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。

名称	データ型	初期値	コメント
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Hm_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_Home のインスタンス HM を起動します。
Mv_Abs1_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveAbsolute のインスタンス MV_ABS1 を起動します。
Mv_Abs2_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveAbsolute のインスタンス MV_ABS2 を起動します。
Mv_Abs3_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveAbsolute のインスタンス MV_ABS3 を起動します。
Mv_Abs4_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveAbsolute のインスタンス MV_ABS4 を起動します。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート



● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag = FALSE THEN

    //MV_ABS1 パラメータ
    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 パラメータ
    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 パラメータ
    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 パラメータ
    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;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag := TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、サーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//軽度フォールトレベルの異常発生時の処理
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

//サーボ ON 状態で、かつ原点未確定のとき、原点復帰を行います
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

//原点確定後に、MV_ABS1 を実行します
IF Hm_D=TRUE THEN
    Mv_Abs1_Ex:=TRUE;
END_IF;
```

```
//MV_ABS1 を開始した後、MV_ABS2 を多重起動します
IF Mv_Abs1_Act=TRUE THEN
    Mv_Abs2_Ex:=TRUE;
END_IF;

//MV_ABS2 を開始した後、MV_ABS3 を多重起動します
IF Mv_Abs2_Act=TRUE THEN
    Mv_Abs3_Ex:=TRUE;
END_IF;

//MV_ABS3 を開始した後、MV_ABS4 を多重起動します
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
);

//絶対値位置決め(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
```

```
);

//絶対値位置決め(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
);

//絶対値位置決め(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
);

//絶対値位置決め(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

指令現在位置からの移動距離を指定して、位置決めを行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveRelative	相対値位置決め	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 :=「パラメータ」, Execute :=「パラメータ」, Distance :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, BufferMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Distance	移動距離	LREAL	負数、正数、「0」	0	指令現在位置からの移動距離を指定します。 単位は、［指令単位］となります。*1
Velocity	目標速度	LREAL	正数	0	目標速度を指定します。*2 単位は、［指令単位／s］となります。*1
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は、［指令単位／s^2］となります。*1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は、［指令単位／s^2］となります。*1
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は、［指令単位／s^3］となります。*1

入力変数	名称	データ型	有効範囲	初期値	内容
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2: ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 目標速度は必ず設定してください。設定せずに動作させると異常が発生します。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位置決めが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

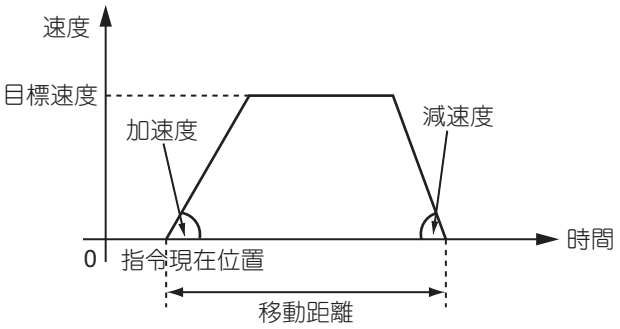
入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

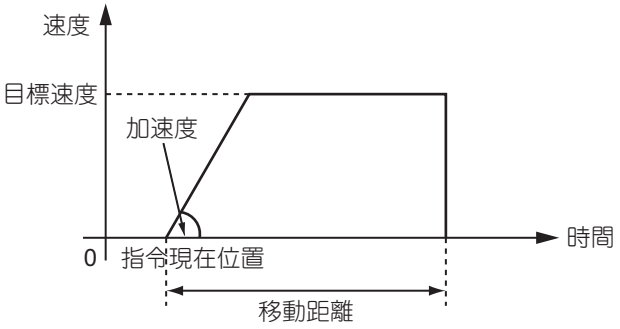
機能説明

- 指令現在位置からの移動距離を指定して、位置決めを行います。
- 「Execute」（起動）の立ち上がりで、相対位置決めの動作を開始します。
- 入力変数に、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）、「Jerk」（ジャーク）を指定することができます。

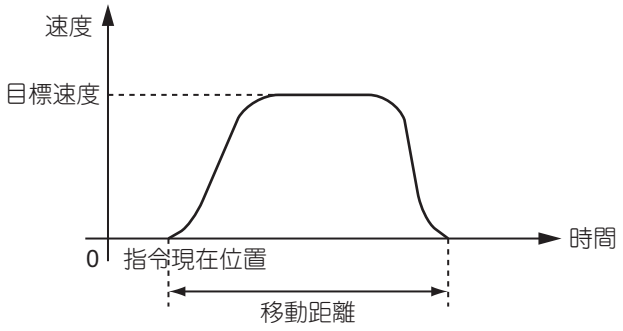
相対位置決めの動作例を下記に示します。



「Acceleration」（加速度）、または「Deceleration」（減速度）を「0」に指定して起動した場合は、加減速せずに目標速度に到達します。
「Deceleration」（減速度）が「0」の場合の動作例を下記に示します。



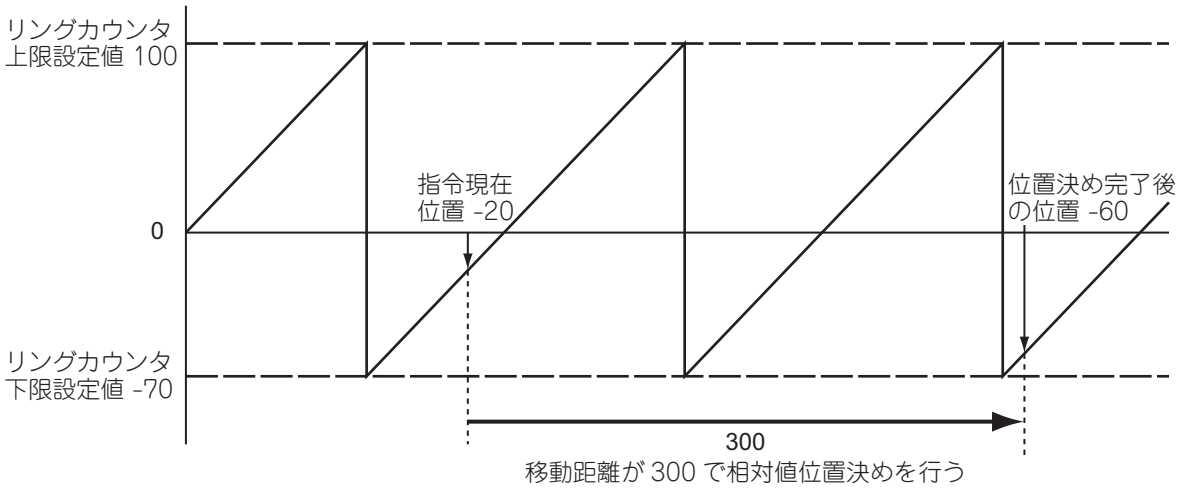
なめらかに加減速させたい場合に「Jerk」（ジャーク）を指定します。
「Jerk」（ジャーク）を指定した場合の動作例を下記に示します。



「Jerk」（ジャーク）の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

命令の詳細

カウントモードが［ロータリモード］の場合、軸パラメータの［リングカウンタ下限設定値］から［リングカウンタ上限設定値］の相対距離範囲を超えた「Distance」（移動距離）の指定をすることにより、リング多回転の位置決めが可能です。
指令現在位置が「-20」で「Distance」（移動距離）を「300」に指定した場合の動作例を下記に示します。

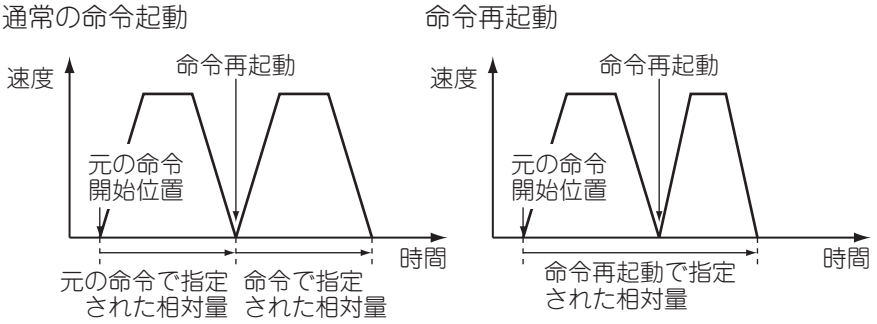


「Distance」（移動距離）を「0」に指定して起動した場合は、軸は移動しませんが「Done」（完了）が TRUE となります。

使用上の注意

位置決めの完了直前に相対値位置決めを再起動する場合は以下の点に注意してください。
MC 機能モジュールが再起動をするまでに位置決めが完了していたら、通常の命令起動となります。

- 通常の命令起動の場合は、命令起動時の位置を基準にした相対位置で位置決めを行います。
- 命令再起動の場合は、元の命令起動時の位置を基準にした相対位置で位置決めを行います。

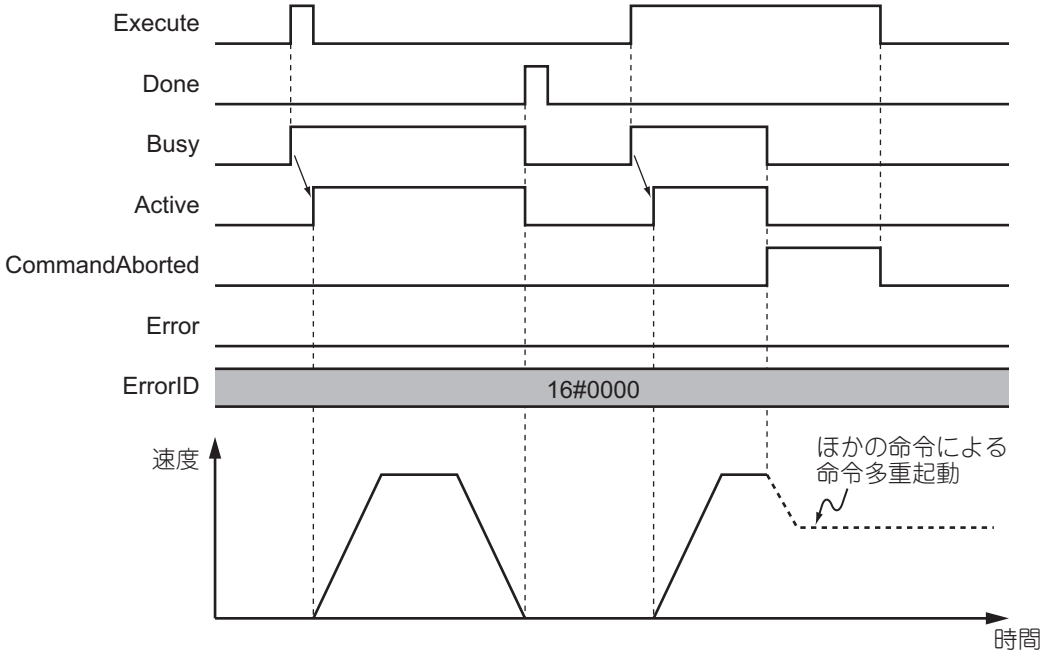


● インポジションチェック

本命令のインポジションチェックは、軸パラメータの［インポジション幅］、［インポジションチェック時間］の設定に従って行われます。

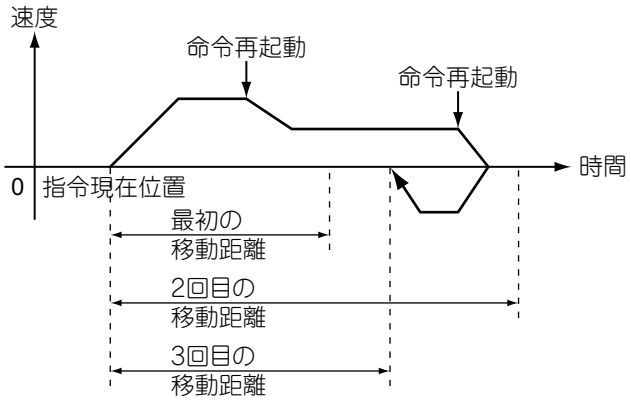
タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- 「Distance」（移動距離）に到達し、位置決めが完了すると「Done」（完了）が TRUE となります。
- ほかの命令により本命令が中止された場合は、「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）が FALSE になります。



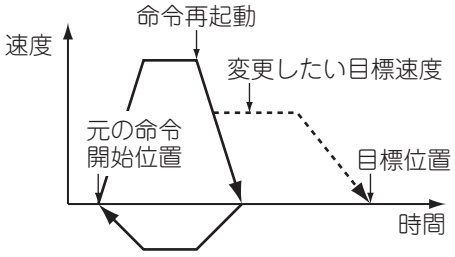
モーション命令再起動

位置決め動作中に入力変数を変更し、「Execute」（起動）を再度 TRUE にすることにより、本命令の動作を変更することができます。
モーション命令再起動で変更可能な入力変数は、「Distance」（移動距離）、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）です。
再起動時の「Distance」（移動距離）の起点は、再起動時の指令現在位置ではなく、最初に本命令が起動された指令現在位置となります。
「Distance」（移動距離）と「Velocity」（目標速度）を変更しながらモーション命令再起動を 2 回行った場合の動作例を以下に記します。



使用上の注意

「Distance」（移動距離）以外の入力パラメータを変更する場合でも、「Distance」（移動距離）には元の命令と同じ値を代入して命令再起動を行ってください。
たとえば、「Distance」（移動距離）の値を「0」に変更して命令再起動すると、初回起動時からの、「Distance」（移動距離）を「0」として命令再起動されます。そのため、下図のように元の開始位置に戻ります。



モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して本命令を起動することにより、本命令への切り替えや本命令のバッファリングを行うことができます。

各軸に 1 つまでバッファリングすることが可能です。
モーション命令多重起動時の「Distance」（移動距離）の起点は、本命令が起動され、「Active」（制御中）が TRUE となったときの指令現在位置となります。
多重起動時の本命令の動作を「BufferMode」（バッファモード選択）で指定します。

バッファモード選択		説明
アボーティング		現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。
バッファード		現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。
ブレンディング		現在実行中の命令が目標位置に到達したときの速度（中継速度）を起動速度として、バッファリングされている本命令を連続して動作させます。中継速度で目標位置に到達するように、現在実行中の命令の動作を変更します。中継速度の指定方法は、以下の 4 種類があります。
	ブレンディングロウ	現在実行中の目標速度とバッファリングされている目標速度のどちらか低い方を中継速度とします。
	ブレンディングプレビアス	現在実行中の目標速度を中継速度とします。
	ブレンディングネクスト	バッファリングされている本命令の目標速度を中継速度とします。
	ブレンディングハイ	現在実行中の目標速度とバッファリングされている目標速度のどちらか高い方を中継速度とします。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

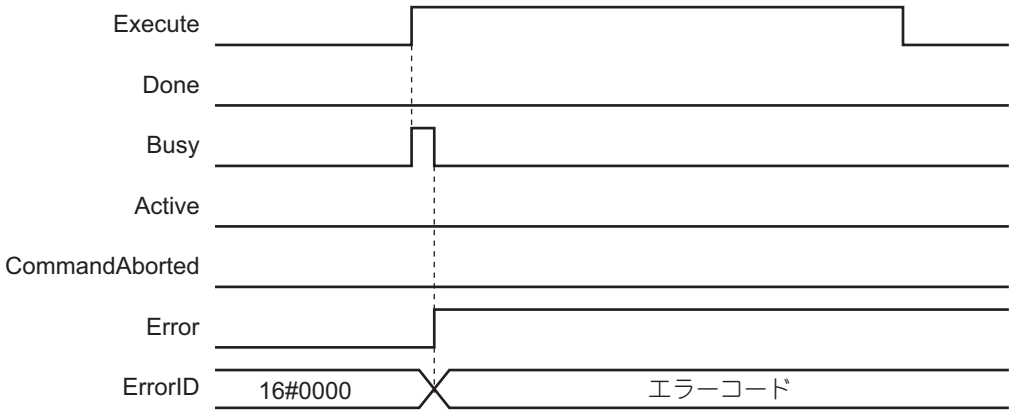
● 本命令実行中の他命令起動

本命令に対するモーション命令多重起動を行う場合、［アボーティング］、［バッファード］、［ブレンディング］が可能です。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_MoveVelocity

サーボドライバの位置制御モードを用いた擬似的な速度制御を行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveVelocity	速度制御	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 :=「パラメータ」, Execute :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, Direction :=「パラメータ」, Continuous :=「パラメータ」, BufferMode :=「パラメータ」, InVelocity =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Velocity	目標速度	LREAL	正数、または「0」	0	目標速度を指定します。 単位は、[指令単位／s] となります。*1
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は、[指令単位／s^2] となります。*1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は、[指令単位／s^2] となります。*1
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は、[指令単位／s^3] となります。*1
Direction	方向選択	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 3: _mcCurrentDirection	0 *2	回転方向を指定します。 0: 正方向指定 2: 負方向指定 3: 現在方向指定

入力変数	名称	データ型	有効範囲	初期値	内容
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *2	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード
Continuous (Reserved)	継続方法選択	BOOL	TRUE, FALSE	FALSE	(Reserved)

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
InVelocity	目標速度到達	BOOL	TRUE, FALSE	目標速度に到達したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
InVelocity	目標速度に到達したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。 • 命令再起動で目標速度を変更したとき。
Busy	「Execute」の立ち上がり	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]、[バッファード]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 位置制御による擬似的な速度制御を行います。
- 「Execute」（起動）の立ち上がりで、速度制御の動作を開始します。

命令の詳細

以下に本命令の詳細について説明します。

● Direction（方向選択）

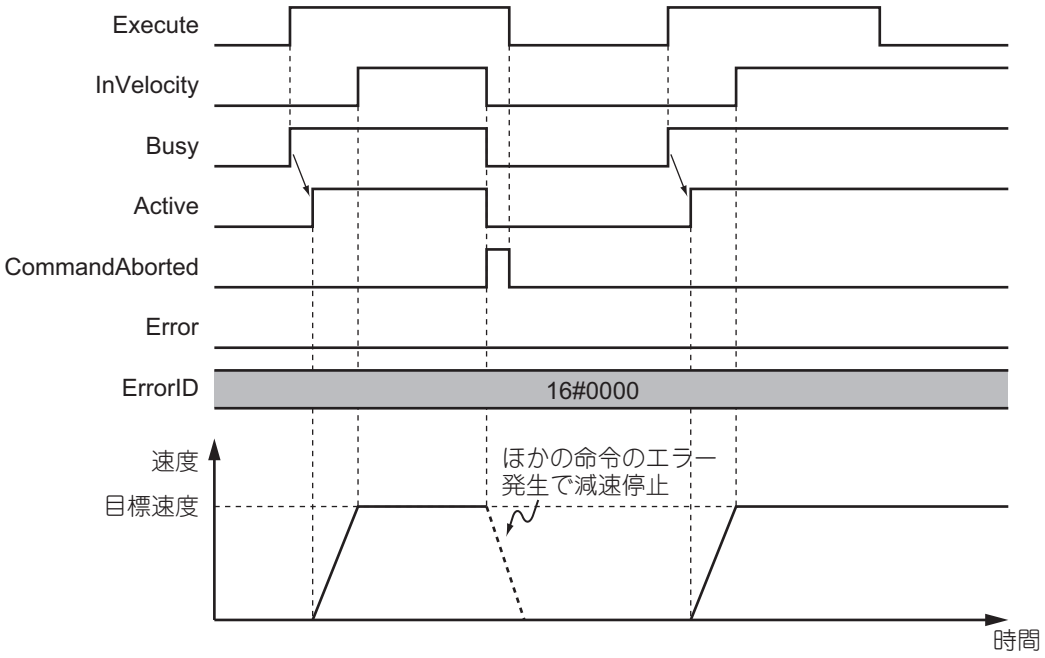
- 「Direction」（方向選択）で移動する方向を指定します。
- 「Direction」（方向選択）が［0: _mcPositiveDirection］（正方向指定）の場合は正方向に、［2: _mcNegativeDirection］（負方向指定）の場合は負方向に移動します。
- 「Direction」（方向選択）が［3: _mcCurrentDirection］（現在方向指定）の場合、軸が停止しているかどうかで動作が異なります。軸が停止している場合、前回の移動方向に軸が移動します。電源投入時、またはリスタート時は正方向に移動します。
  モーション命令多重起動で軸が移動している最中に本命令を起動した場合は、現在移動している方向に移動します。

使用上の注意

「Direction」（方向選択）に［3: _mcCurrentDirection］（現在方向指定）を選択した場合、前の動作における指令方向を踏襲して動作します。そのため命令の組み合わせによっては、前動作のモーション命令の入力で指令した方向と一致しない場合があります。
［3: _mcCurrentDirection］（現在方向指定）を使用する場合は、軸変数の「Dir.Posi」（正方向指令中）、および「Dir.Nega」（負方向指令中）で現在の方向を確認してください。

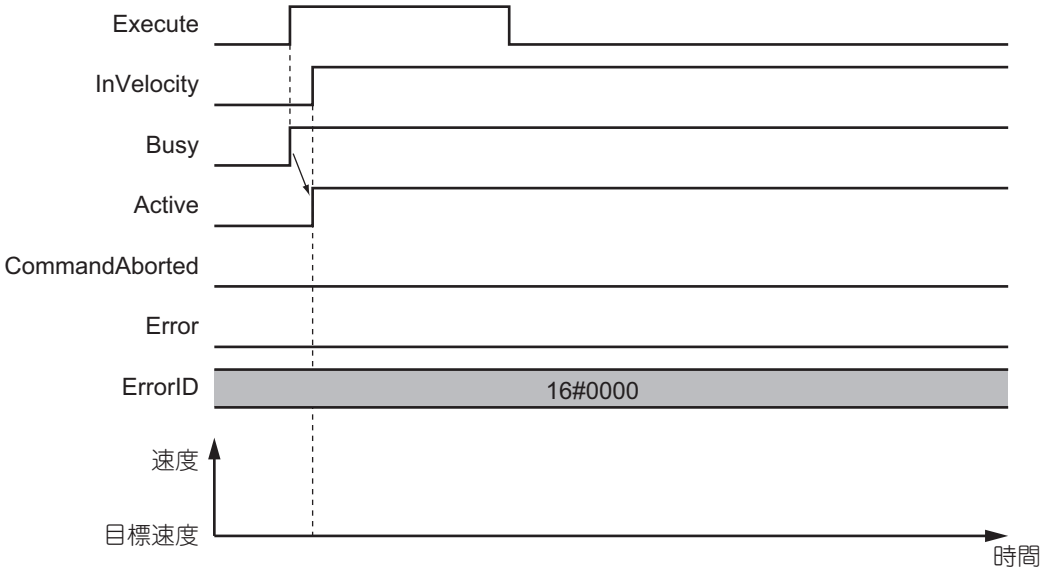
タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- 「Velocity」（目標速度）に到達すると、「InVelocity」（目標速度到達）が TRUE となります。
- ほかの命令により本命令が中止された場合は、「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）、「InVelocity」（目標速度到達）が FALSE になります。

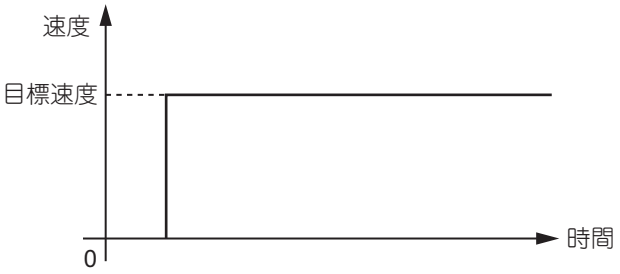


「InVelocity」（目標速度到達）は、本命令の起動およびモーション命令再起動に対する等速への到達を示す出力です。したがって「InVelocity」（目標速度到達）が TRUE となった後に、オーバーライドによって速度を変更しても「InVelocity」（目標速度到達）は FALSE になりません。
また、「InVelocity」（目標速度到達）が TRUE になる前にオーバーライドが変更された場合は、変更された目標速度に到達したときに「InVelocity」（目標速度到達）が TRUE となります。

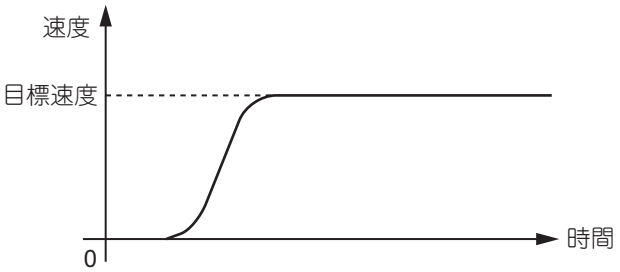
入力変数に、「Acceleration」（加速度）、「Deceleration」（減速度）、「Jerk」（ジャーク）を指定することができます。
「Velocity」（目標速度）を「0」に指定して起動した場合は、軸は移動しませんが連続動作中となります。
「Velocity」（目標速度）を「0」にして起動した場合の動作例を下記に示します。



「Acceleration」（加速度）、または「Deceleration」（減速度）を「0」に指定して起動した場合は、加減速せずに目標速度に到達します。
「Acceleration」（加速度）が「0」の場合の動作例を下記に示します。



なめらかに加減速させたい場合に「Jerk」（ジャーク）を指定します。
「Jerk」（ジャーク）を指定した場合の動作例を下記に示します。



「Jerk」（ジャーク）の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション命令再起動

連続動作中に入力パラメータを変更し、「Execute」（起動）を再度 TRUE にすることにより、本命令の動作を変更することができます。
モーション命令再起動で変更可能な入力変数は、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）です。
モーション命令再起動によって「Velocity」（目標速度）を変更する場合、「InVelocity」（目標速度到達）は、再起動によって設定された新しい目標速度に対して動作します。

モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して、本命令を起動することにより、本命令への切り替えや本命令のバッファリングを行うことができます。
各軸に 1 つまでバッファリングすることが可能です。
多重起動時の本命令の動作を、「BufferMode」（バッファモード選択）で指定します。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。

バッファモード選択	説明
バッファード	現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

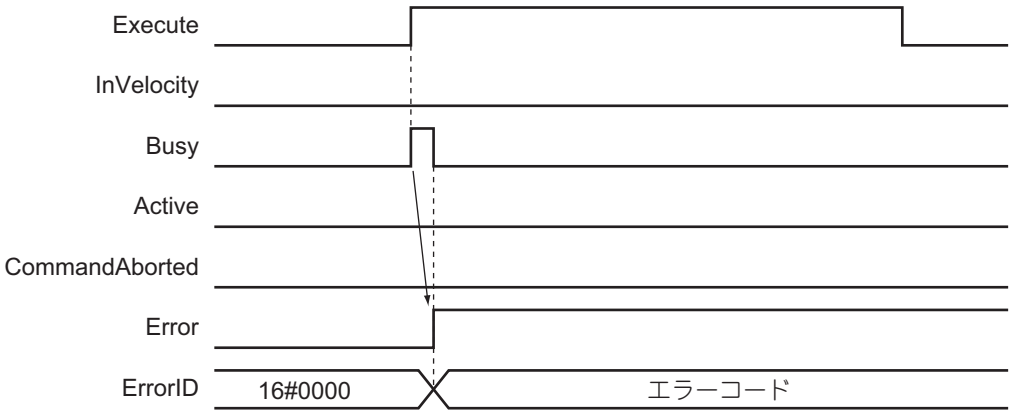
● 本命令実行中の他命令起動

本命令実行中のほかの命令によるモーション命令多重起動の実行は、ほかの命令の「BufferMode」（バッファモード選択）に［アボーティング］と［バッファード］を指定した時だけ可能です。
［バッファード］を指定した場合、本命令の出力変数「InVelocity」（目標速度到達）が TRUE になったときに多重起動された命令の動作が開始します。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、スピナーや遠心分離器のような速度制御についてサンプルプログラムを掲載します。

パラメータ設定

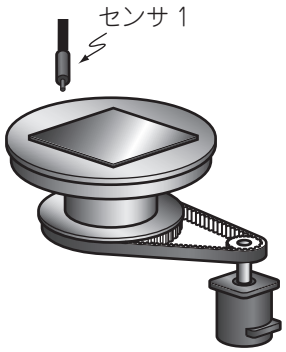
このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

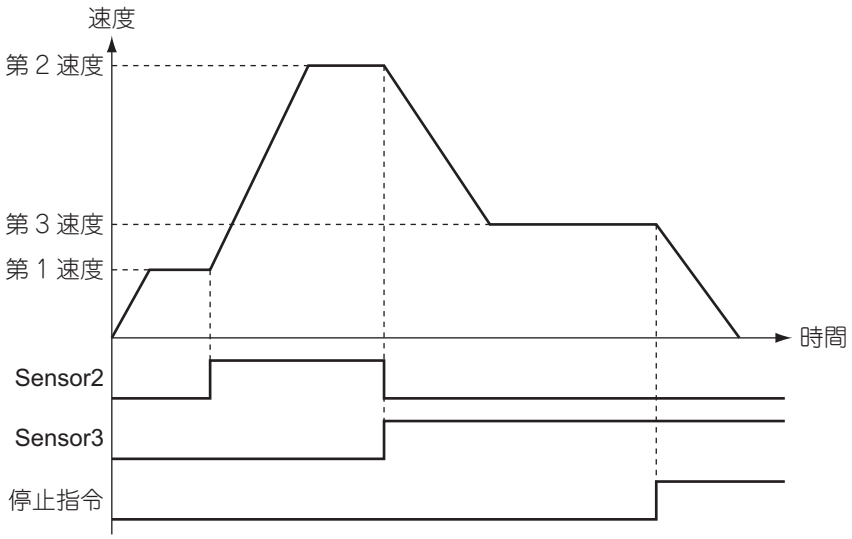
軸種別

軸	軸種別
軸 1	サーボ軸

動作例



● 動作パターン



1 **速度制御の開始**
薬液投入を検出するセンサ 1 が「ON」すると、軸 1 は速度制御を開始します。

2 **第 2 速度への変更**
Sensor2（接点）が「ON」するとオーバーライド 500％が設定され、速度が変更されます。

3 **第 3 速度への変更**
Sensor3（接点）が「ON」するとオーバーライド 200％が設定され、速度が変更されます。
なお、Sensor2 と Sensor3 が同時に「ON」した場合は、オーバーライド 200%となります。

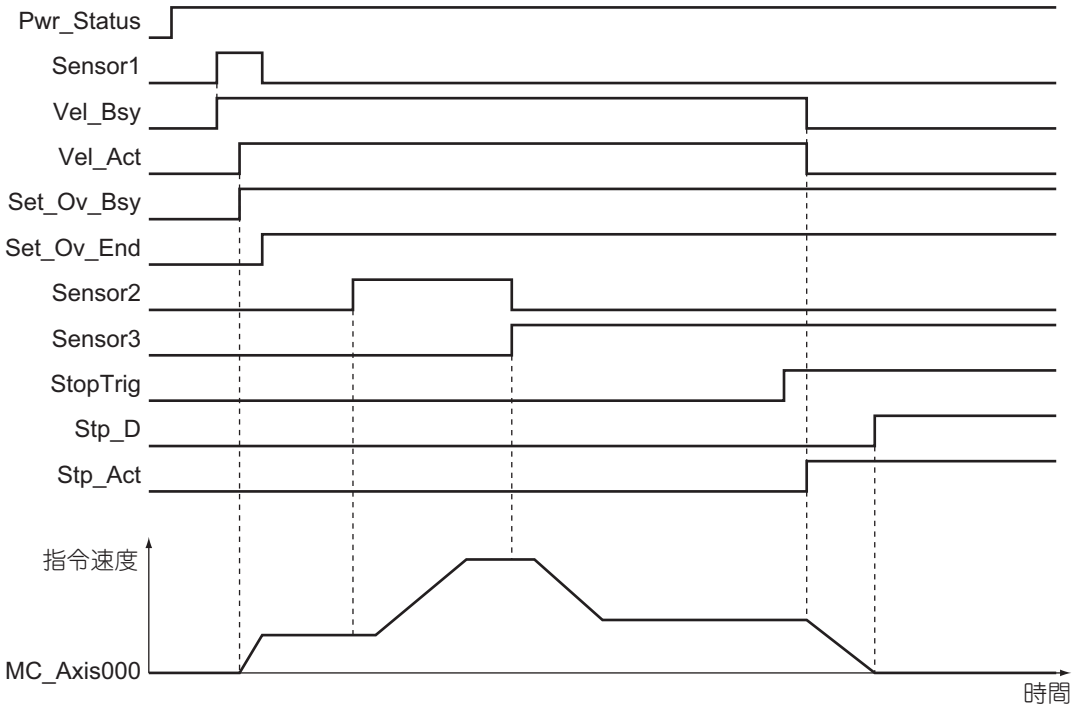
4 **速度制御の停止**
停止指令(StopTrig)が「ON」すると減速停止します。

ラダー図

● 主な変数

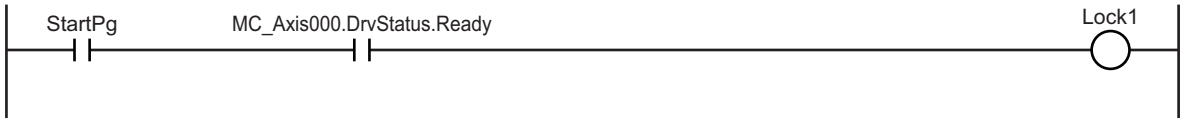
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power のインスタンス PWR の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Vel_Act	BOOL	FALSE	MC_MoveVelocity のインスタンス VEL の出力「Active」に割り付けた変数です。インスタンス VEL が制御中になると、この変数が TRUE となります。
Set_Ov_Velfct	LREAL	0	オーバーライド値です。
StopTrig	BOOL	FALSE	この変数が TRUE の時に、MC_Stop を実行します。
Sensor1	BOOL	FALSE	薬液投入を検出すると TRUE になります。 軸 1 がサーボ ON 状態のとき、 MC_MoveVelocity（速度制御）を起動します。
Sensor2	BOOL	FALSE	この変数が TRUE の時に、オーバーライド値を 500%に設定します。この変数は 1 度 TRUE になると状態を保持し、Sensor3 が TRUE になると、FALSE となります。
Sensor3	BOOL	FALSE	この変数が TRUE の時に、オーバーライド値を 200%に設定します。この変数は 1 度 TRUE になると状態を保持します。

● タイミングチャート

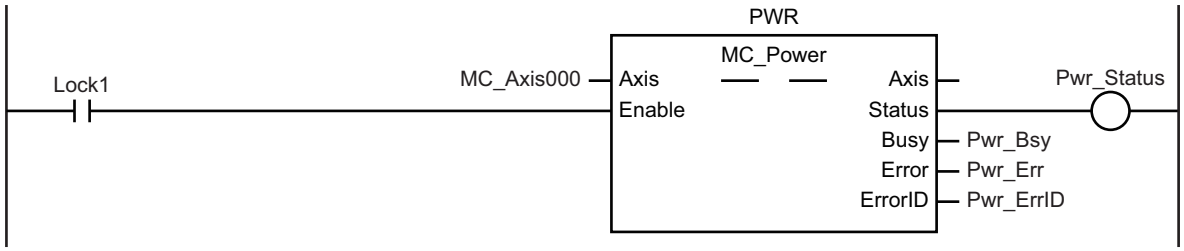


● サンプルプログラム

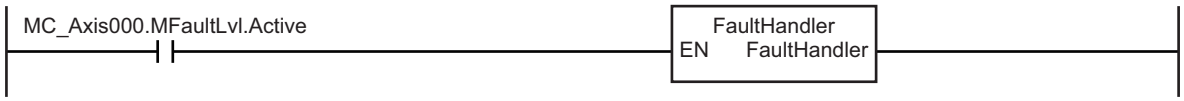
接点 StartPg が TRUE になったら、サーボドライバがサーボレディ状態になっていることを確認する



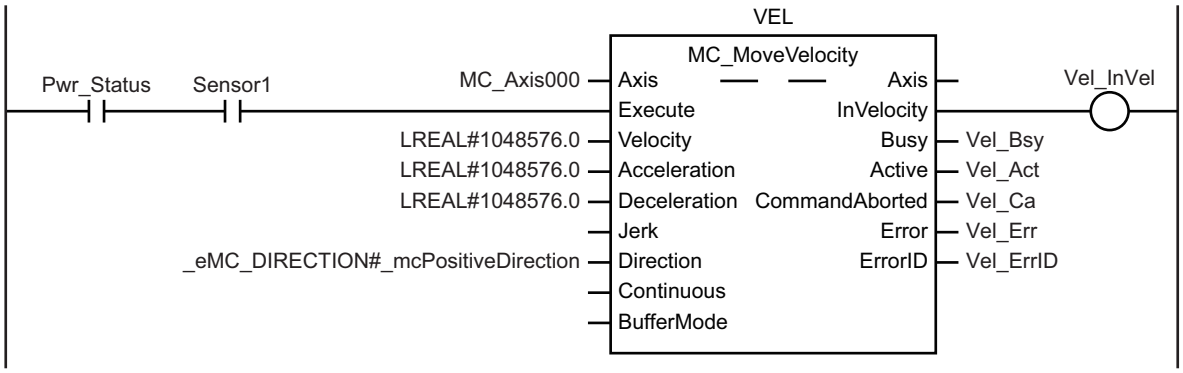
サーボレディ状態になっていれば、サーボ ON 状態にする



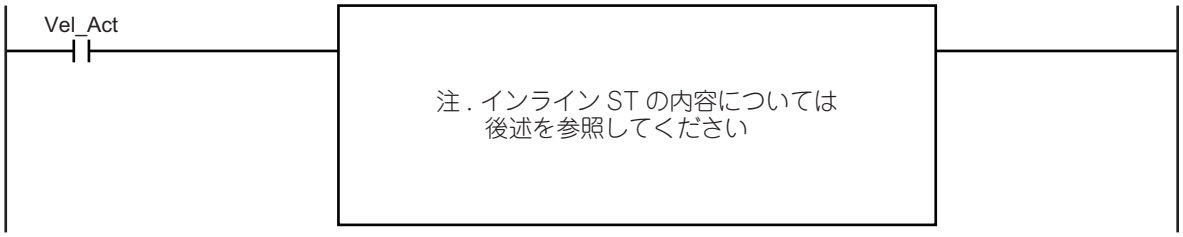
軸 1 で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



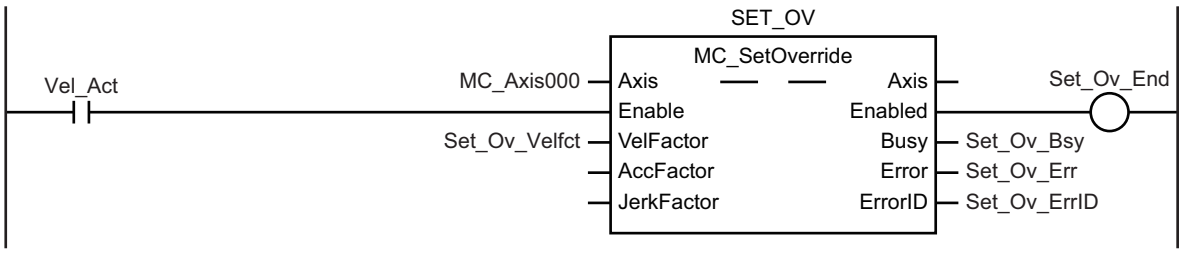
運転可能状態で、Sensor1 が TRUE のときに MC_MoveVelocity（速度制御）命令を起動する



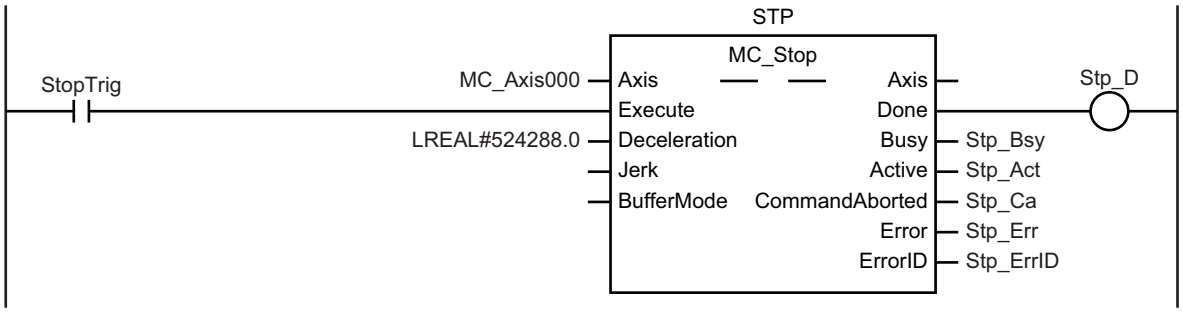
MC_MoveVelocity（速度制御）命令制御中に、接点 Sensor2 と接点 Sensor3 の ON/ OFF に伴い、オーバーライド値を変更する



MC_MoveVelocity（速度制御）命令の制御中は、MC_SetOverride（オーバーライド値設定）命令を起動する



停止指令（StopTrig）が TRUE になると、MC_Stop（強制停止）命令を起動する



インライン ST の内容

```
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
```

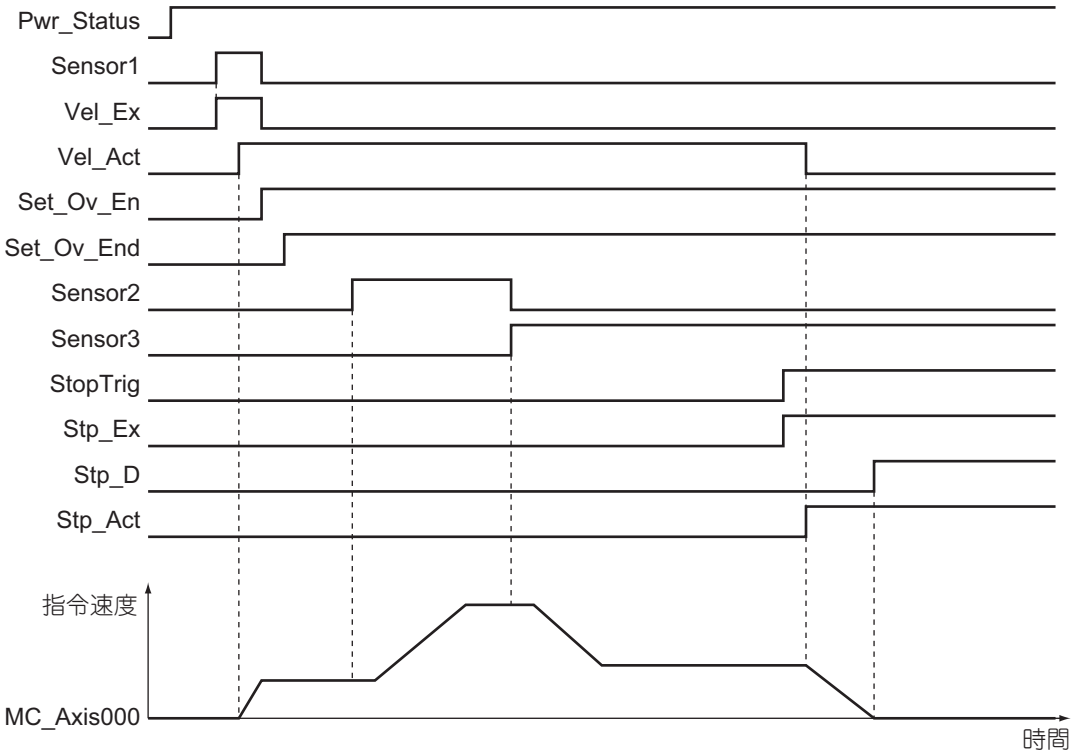
```
    Set_Ov_Velfct := LREAL#200.0;
END_IF;
```

ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power のインスタンス PWR の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Vel_Act	BOOL	FALSE	MC_MoveVelocity のインスタンス VEL の出力「Active」に割り付けた変数です。インスタンス VEL が制御中になると、この変数が TRUE となります。
Set_Ov_Velfct	LREAL	0	オーバーライド値です。
StopTrig	BOOL	FALSE	この変数が TRUE の時に、MC_Stop を実行します。
Sensor1	BOOL	FALSE	薬液投入を検出すると TRUE になります。 軸 1 がサーボ ON 状態のとき、 MC_MoveVelocity（速度制御）命令を起動します。
Sensor2	BOOL	FALSE	この変数が TRUE の時に、オーバーライド値を 500%に設定します。この変数は 1 度 TRUE になると状態を保持し、Sensor3 が TRUE になると、FALSE となります。
Sensor3	BOOL	FALSE	この変数が TRUE の時に、オーバーライド値を 200%に設定します。この変数は 1 度 TRUE になると状態を保持します。
Vel_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、 MC_MoveVelocity のインスタンス VEL を起動します。
Set_Ov_En	BOOL	FALSE	この変数が TRUE の間、MC_SetOverride のインスタンス SET_OV を実行します。
Stp_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_Stop のインスタンス STP を起動します。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート



● サンプルプログラム

```
// 入力パラメータが設定されていない場合の処理
IF InitFlag = FALSE THEN

    //MC_MoveVelocity パラメータ
    Vel_Vel := LREAL#1048576.0;
    Vel_Acc := LREAL#1048576.0;
    Vel_Dec := LREAL#1048576.0;
    Vel_Dir :=_eMC_DIRECTION#_mcPositiveDirection;

    //MC_SetOverride パラメータ
    Set_Ov_Velfct := LREAL#100.0;

    //MC_Stop パラメータ
    Stp_Dec := LREAL#524288.0;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag:=TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
```

```
ELSE
    Pwr_En:=FALSE;
END_IF;

// 軸 1 で軽度フォールトが発生したら、異常時処理 FaultHandler を実行します
// 異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

// 軸 1 がサーボ ON 状態で、かつ接点 Sensor1 が TRUE のとき、MC_MoveVelocity を起動します
IF (Pwr_Status=TRUE) AND (Sensor1=TRUE) THEN
    Vel_Ex := TRUE;
END_IF;

// MC_MoveVelocity 制御中に、接点 Sensor2 と接点 Sensor3 の ON/OFF に伴い、オーバーライド値を変更します
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 制御中は MC_SetOverride を起動します
IF Vel_Act=TRUE THEN
    Set_Ov_En := TRUE;
END_IF;

// StopTrig が TRUE で MC_Stop を起動します
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

絶対座標の「0」を目標位置として位置決めを行い、原点へ戻ります。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveZeroPosition	高速原点復帰	FB	MC_MoveZeroPosition_instance MC_MoveZeroPosition Axis — Axis Execute — Done Velocity — Busy Acceleration — Active Deceleration — CommandAborted Jerk — Error BufferMode — ErrorID	MC_MoveZeroPosition_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, BufferMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Velocity	目標速度	LREAL	正数	0	目標速度を指定します。*1 単位は、[指令単位／s] となります。*2
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は、[指令単位／s^2] となります。*2
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は、[指令単位／s^2] となります。*2
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は、[指令単位／s^3] となります。*2
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *3	モーション命令多重起動時の動作指定します。 0: アボーティング 1: バッファード

*1. 目標速度は必ず設定してください。設定せずに動作させると異常が発生します。
*2. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位置決めが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 原点位置へ移動します。
- 入力変数に、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）、「Jerk」（ジャーク）を指定することができます。

使用上の注意

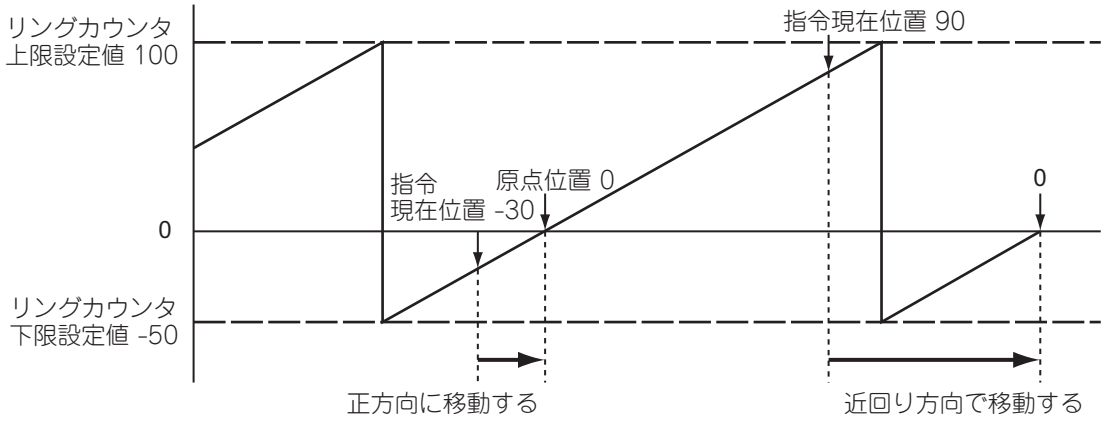
MC_MoveZeroPosition（高速原点復帰）命令は原点が確定した状態で実行してください。原点が確定していないと「原点未確定状態での命令起動異常（エラーコード: 5466 Hex）」が発生します。

命令の詳細

以下に本命令の詳細について説明します。

● 方向指定

カウントモードが［ロータリモード］の場合、原点位置への距離が短い方向に向かっての位置決め（近回り）となります。
指令現在位置に応じた本命令の動作例を下記に示します。



使用上の注意

軸パラメータの［リングカウンタ上限設定値］と［リングカウンタ下限設定値］の範囲外に原点が存在した状態で本命令を起動した場合、「目標位置リングカウンタ範囲外（エラーコード: 549C Hex）」となります。

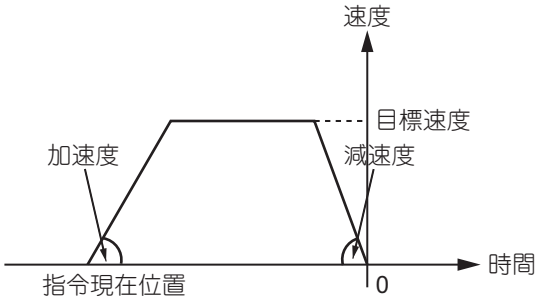
● オーバーライド

本命令に対して、オーバーライドが有効です。

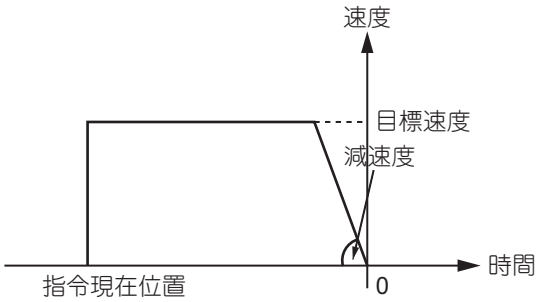
● インポジションチェック

本命令のインポジションチェックは、軸パラメータの［インポジション幅］、［インポジションチェック時間］の設定に従って行われます。

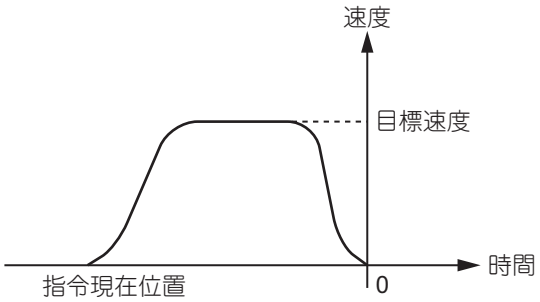
● **動作例**



「Acceleration」（加速度）、または「Deceleration」（減速度）を「0」に指定して起動した場合は、加減速せずに目標速度に到達します。
「Acceleration」（加速度）が「0」の場合の動作例を下記に示します。



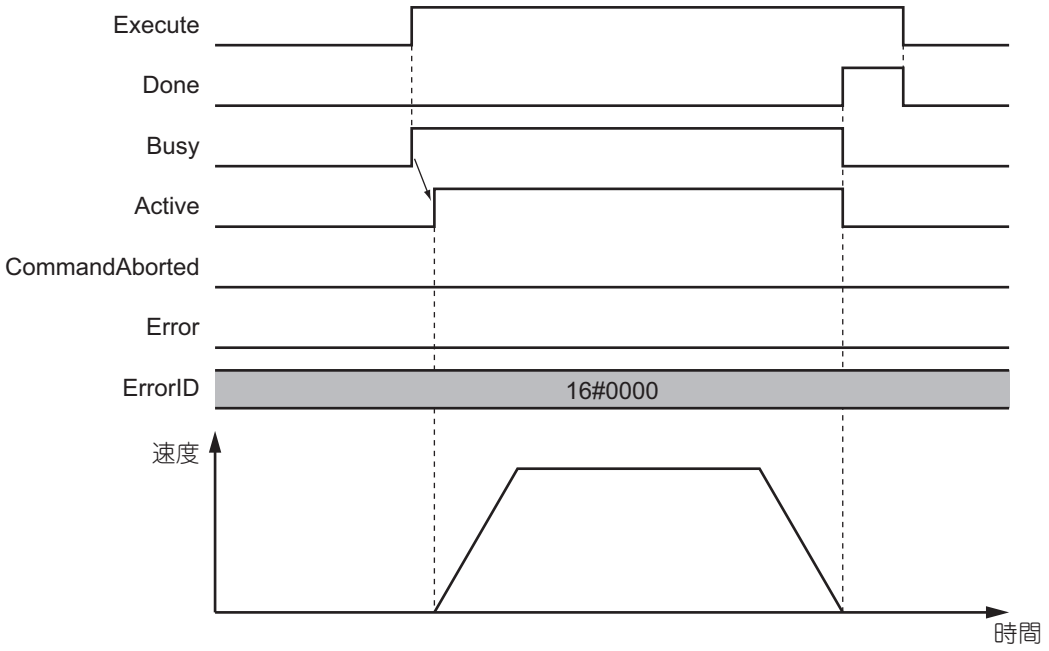
なめらかに加減速させたい場合に「Jerk」（ジャーク）を指定します。
「Jerk」（ジャーク）を指定した場合の動作例を下記に示します。



「Jerk」（ジャーク）の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

タイミングチャート

MC_MoveZeroPosition（高速原点復帰）命令を実行した場合のタイミングチャートを下記に示します。



命令の中止

本命令が中止され「CommandAborted」（実行中断）が TRUE となった場合でも、原点未確定状態になることはありません。

モーション命令再起動

位置決め動作中に入力変数を変更し、「Execute」（起動）を再度 TRUE にすることにより、本命令の動きを変更することができます。
モーション命令再起動で変更可能な入力変数は、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）です。

モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して、本命令を起動することにより、本命令への切り替えや本命令のバッファリングを行うことができます。
各軸に 1 つまでバッファリングすることが可能です。
多重起動時の本命令の動作を、「BufferMode」（バッファモード選択）で指定します。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。

バッファモード選択	説明
バッファード	現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。

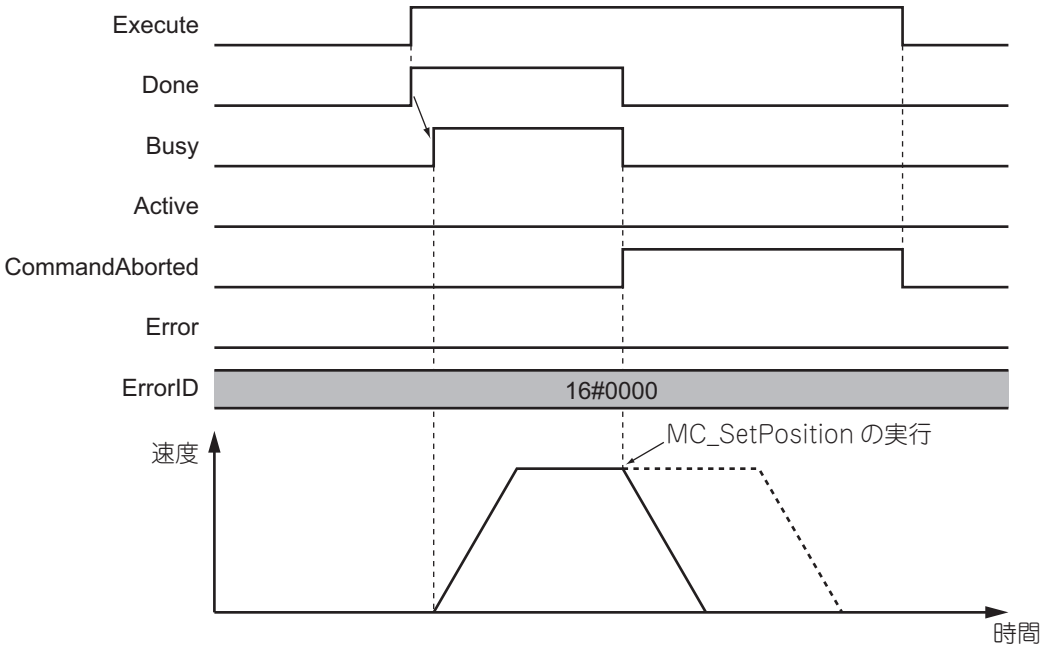
「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

本命令に対するモーション命令多重起動を行う場合、［アボーティング］、［バッファード］が可能です。
［ブレンディング］による命令多重起動はできません。

本命令実行中に MC_SetPosition（現在位置変更）命令を実行した場合

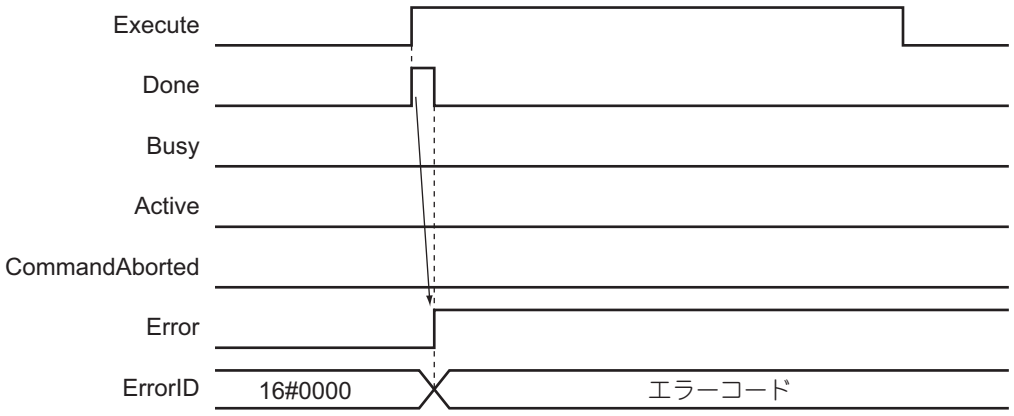
MC_SetPosition（現在位置変更）命令では原点未確定となるため、本命令実行中には異常となり実行できません。
MC_SetPosition（現在位置変更）命令実行により異常状態になった場合、本命令は減速停止し「CommandAborted」（実行中断）が TRUE となります。
この際、MC_SetPosition（現在位置変更）命令では出力変数の「Error」（エラー）が TRUE となります。



異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート

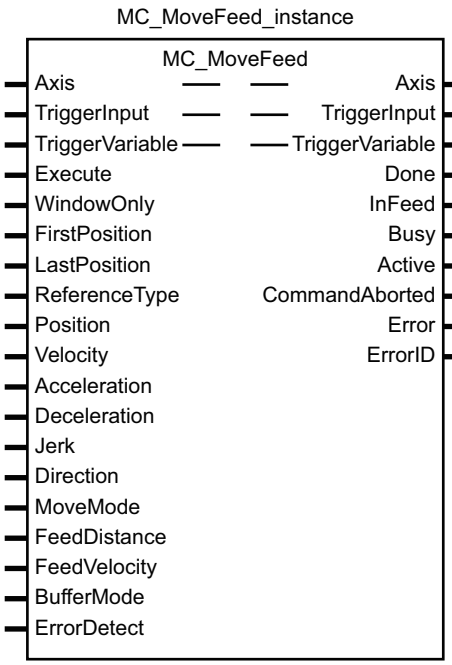


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_MoveFeed

外部入力による割り込み入力が発生した位置からの移動距離を指定し、位置決めを行います。
割り込み定寸位置決めができる制御は、絶対値位置決め、相対値位置決め、速度制御です。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveFeed	割り込み定寸位置決め	FB		MC_MoveFeed_instance (Axis :=「パラメータ」, TriggerInput :=「パラメータ」, TriggerVariable :=「パラメータ」, Execute :=「パラメータ」, WindowOnly :=「パラメータ」, FirstPosition :=「パラメータ」, LastPosition :=「パラメータ」, ReferenceType :=「パラメータ」, Position :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, Direction :=「パラメータ」, MoveMode :=「パラメータ」, FeedDistance :=「パラメータ」, FeedVelocity :=「パラメータ」, BufferMode :=「パラメータ」, ErrorDetect :=「パラメータ」, Done =>「パラメータ」, InFeed =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
WindowOnly	ウィンドウ有効	BOOL	TRUE, FALSE	FALSE	ウィンドウの有効／無効を指定します。
FirstPosition	開始位置	LREAL	負数、正数、「0」	0	ラッチが有効となる位置を指定します。 単位は、[指令単位] となります。 *1
LastPosition	終了位置	LREAL	負数、正数、「0」	0	ラッチが無効となる位置を指定します。 単位は、[指令単位] となります。 *1

入力変数	名称	データ型	有効範囲	初期値	内容
ReferenceType	位置種別選択	_eMC_REFERENCE_TYPE	1: _mcFeedback	1 *2	位置の種別を指定します。 1: フィードバック位置（同一タスク周期 *3 での取得値）
Position	目標位置	LREAL	負数、正数、「0」	0	「MoveMode」 が［0: 絶対値位置決め］の場合、絶対座標の目標位置を指定します。 「MoveMode」 が［1: 相対値位置決め］の場合、移動距離を指定します。 「MoveMode」 が［2: 速度制御］の場合、指定は不要です。 単位は、［指令単位］となります。 *1
Velocity	目標速度	LREAL	正数 *4	0	目標速度を指定します。 単位は、［指令単位／s］となります。 *1
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は、［指令単位／s^2］となります。 *1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は、［指令単位／s^2］となります。 *1
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は、［指令単位／s^3］となります。 *1
Direction	方向選択	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	0 *2	「MoveMode」が［0: 絶対値位置決め］ *5 で、カウントモードが［ロータリモード］の場合に、回転方向を指定します。 0: 正方向指定 1: 近回り指定 2: 負方向指定 3: 現在方向指定 4: 方向指定なし
MoveMode	移動方法選択	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative 2: _mcVelocity	0 *2	移動方法を選択します。 0: 絶対値位置決め 1: 相対値位置決め 2: 速度制御
FeedDistance	定寸距離	LREAL	負数、正数、「0」	0	割り込み入力後の移動距離を指定します。 割り込み入力前の動作方向と同じ方向に定寸動作をさせる場合には正数を、逆方向に動作させるには負数を設定します。 単位は、［指令単位］となります。 *1
FeedVelocity	定寸速度	LREAL	正数	0	割り込み入力後の移動目標速度を指定します。 単位は、［指令単位／s］となります。 *1
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *2	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード

入力変数	名称	データ型	有効範囲	初期値	内容
ErrorDetect	エラー検出選択	BOOL	TRUE, FALSE	FALSE	割り込み入力が無いときに、異常検出をするかを指定します。 TRUE: 異常検出有 FALSE: 異常検出無

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*3. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。
*4. 「MoveMode」に［2: 速度制御］が指定されているときは、「0」も可能となります。
*5. 「MoveMode」が［0: 絶対値位置決め］ で、カウントモードが［リニアモード］の場合は、指定された位置に移動します。
「MoveMode」が［1: 相対値位置決め］の場合は、Position の符号により移動方向が決まります。
「MoveMode」が［2: 速度制御］の場合は、カウンタモードに関係なく［1: 近回り指定］、および［4: 方向指定なし］は選択できません。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
InFeed	定寸送り中	BOOL	TRUE, FALSE	ラッチ入力を受け付け、定寸送り中に TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	•「Execute」が TRUE の場合、「Execute」の FALSE と同時 •「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	•「Done」が TRUE となったとき。 •「Error」が TRUE となったとき。 •「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	•「Done」が TRUE となったとき。 •「Error」が TRUE となったとき。 •「CommandAborted」が TRUE となったとき。
InFeed	割り込み入力により、定寸動作を開始したとき。*1	•「Done」が TRUE となったとき。 •「Error」が TRUE となったとき。 •「CommandAborted」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

*1. 割り込み入力が「ON」してから「InFeed」が TRUE となるまでに数タスク周期遅れる場合があります。
タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1
TriggerInput	トリガ入力条件	_sTRIGGER_REF	−	トリガ条件を設定します。*2
TriggerVariable	トリガ変数	BOOL	TRUE, FALSE	トリガ条件でコントローラモードを指定したとき、トリガとなる入力変数を指定します。

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. _sTRIGGER_REF 型のユーザ定義変数を定義してください。

● _sTRIGGER_REF

メンバ変数	名称	データ型	有効範囲	機能
Mode	モード	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	トリガのモードを指定します。 0: ドライバモード 1: コントローラモード
LatchID	ラッチ ID 選択	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	[ドライバモード] のときに 2 つのラッチ機能のうちのどちらを使用するかを指定します。 0: ラッチ機能 1 1: ラッチ機能 2
InputDrive	ドライブトリガ入力信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	[ドライバモード] のときにサーボドライバのトリガ信号を指定します。 0: Z 相 1: 外部入力

機能説明

- 「Execute」（起動）の立ち上がりで、「MoveMode」（移動方法選択）の設定に従い、絶対値移動、または相対値移動、速度制御のいずれかの移動方法で移動します。
- 絶対値移動では「Position」（目標位置）に目標位置を設定し、相対値移動では「Position」（目標位置）に目標距離を設定します。
  また、いずれの移動方法も「Velocity」（目標速度）での移動動作を行います。
- 移動中に外部入力（割込入力）の立ち上がりで、フィードバック位置から「FeedDistance」（定寸距離）で指定した定寸距離を「FeedVelocity」（定寸速度）で相対位置決め動作を行います。

- 絶対値移動、または相対値移動指令による割り込み定寸送りで、目標位置までに割込信号の入力がない場合は、当初の目標位置で停止します。
  割り込み入力がなく停止した際に、「ErrorDetect」（エラー検出選択）により、異常出力の有無が指定可能です。異常出力指定時は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）が FALSE となります。
- 割り込みマスクを用いる場合は、「WindowOnly」（ウィンドウ有効）を TRUE にし、「FirstPosition」（開始位置）、「LastPosition」（終了位置）を指定します。
  フィードバック位置が「FirstPosition」（開始位置）から「LastPosition」（終了位置）の間で発生する最初の割込信号にて割り込み定寸位置決めを実行します。

使用上の注意

- 割り込み後の定寸動作は、「FeedDistance」（定寸距離）で指定した距離の相対移動を行います。「FeedDistance」（定寸距離）に正数を指定した場合は、割り込み入力前と同じ方向に定寸動作し、負数を指定した場合は、逆方向に定寸動作します。
- 定寸動作で反転する場合の加減速は、軸パラメータ［反転時動作］の設定に従います。
- 割り込み定寸位置決め後の位置がオーバーフロー／アンダーフローとなる場合は、割込入力時異常となります。オーバーフロー／アンダーフロー発生後に割込入力が入った場合も同様に軸異常となります。

参考

絶対値移動については、「MC_MoveAbsolute (P.3-47)」を、相対値移動については、「MC_MoveRelative (P.3-73)」を、速度制御については、「MC_MoveVelocity (P.3-81)」を、「WindowOnly」（ウィンドウ有効）については、「WindowOnly（ウィンドウ有効）(P.3-108)」を参照してください。

● データオブジェクトのマッピング

「Mode」に［ドライバモード］を選択して、MC_MoveFeed（割り込み定寸位置決め）命令を使用する場合は、以下のオブジェクトデータをマッピングしてください。
マッピングは、Sysmac Studio の軸基本設定の［詳細設定］で行います。

- ラッチ機能(60B8Hex)
- ラッチステータス(60B9Hex)
- ラッチ 1 位置(60BAHex)
- ラッチ 2 位置(60BCHex)

これらを設定していない場合、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。

データオブジェクトのマッピングについては、「2-3 PDO マッピング (P.2-32)」、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。
NX シリーズ 位置インタフェースユニットを使用する場合は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』の「I/O エントリマッピング」を参照してください。

命令の詳細

以下に本命令の詳細について説明します。

● Axis（軸）指定

- 「Axis」（軸）にてラッチの対象とする軸を指定します。

● トリガ入力条件

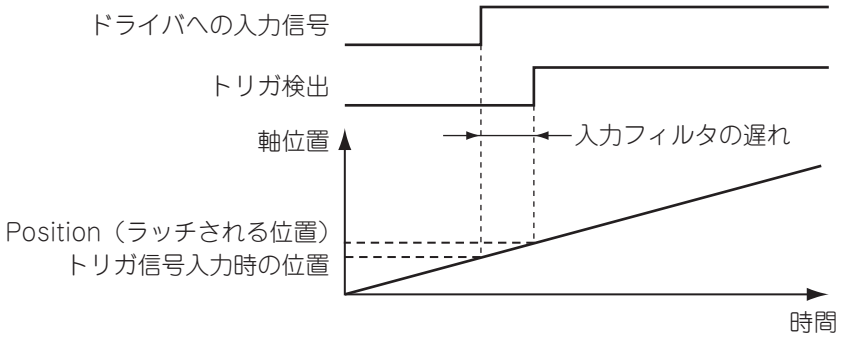
「TriggerInput」（トリガ入力条件）の「Mode」、「LatchID」、「InputDrive」の指定によりトリガの条件を選択します。

Mode

- 「Mode」の設定として、トリガにサーボドライバまたは NX シリーズ パルス出力ユニットの信号を指定する［ドライバモード］と、トリガに「TriggerVariable」（トリガ変数）を指定する［コントローラモード］があります。
- トリガ信号の立ち上がりをトリガ発生とみなし、ラッチ命令実行後の最初のトリガ発生（FALSE→TRUE）時の軸の位置を記録します。
- 本命令が「Busy」（動作中）中は「Execute」（起動）が FALSE となっても、「TriggerVariable」（トリガ変数）の変化をトリガとして動作します。

ドライバモード

［ドライバモード］はサーボドライバの機能でトリガ検出と現在位置の取得を行うため［コントローラモード］に比べてラッチされる現在位置が正確です。

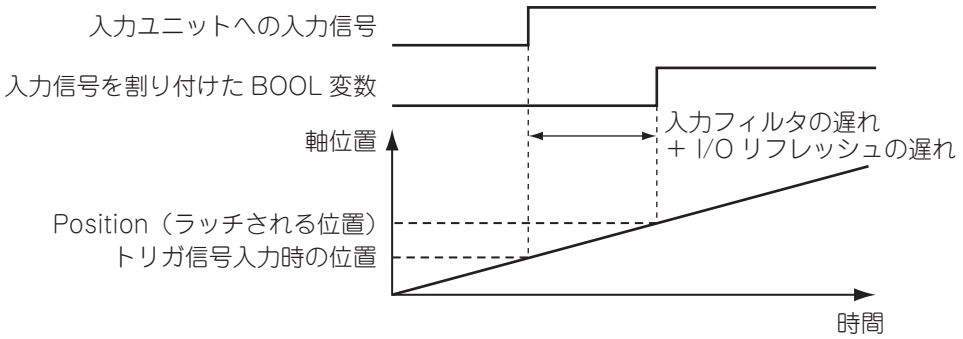


使用上の注意

- ［ドライバモード］を使用する場合は、必ず使用する「LatchID」（ラッチ ID 選択）に、ラッチ信号を接続してください。
- ラッチ信号の幅は、サーボドライバまたは NX シリーズ パルス出力ユニットの性能などに依存します。
- 「InputDrive」に［ドライバモード］を選択して、MC_MoveFeed（割り込み定寸位置決め）命令を使用する場合は、以下のオブジェクトデータをマッピングしてください。
  ラッチ機能(60B8Hex)、ラッチステータス(60B9Hex)、ラッチ 1 位置(60BAHex)、ラッチ 2 位置(60BCHex)
  これらを設定していない場合、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。
  オブジェクトデータのマッピングについては、「2-3 PDO マッピング (P.2-32)」、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

コントローラモード

- ［コントローラモード］では、トリガに BOOL 変数を指定することができます。
- 「TriggerVariable」（トリガ変数）にトリガとしたい BOOL 変数を指定します。
- ［コントローラモード］では、トリガ入力信号の BOOL 変数への反映が I/O リフレッシュの遅れにより［ドライバモード］よりも大きな遅れが生じます。



使用上の注意

［コントローラモード］を使用する場合、タスク周期の間隔でラッチ処理を行います。そのため、トリガ変数はタスク周期の 1 周期の間、TRUE でなければなりません。
また、トリガ変数が TRUE になったあと、MC 機能モジュールが処理するまでタスク周期の 1 周期分の時間が必要です。
ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。

LatchID

- 「LatchID」（ラッチ ID 選択）で、どちらを使用するかを指定します。1 つの軸に対しては、どちらか一方だけしか使用することはできません。
- サーボドライバまたは NX シリーズ パルス出力ユニットでは、「LatchID」（ラッチ ID 選択）がサーボドライバまたは NX シリーズ パルス出力ユニットのラッチ回路 1（第 1ch）、ラッチ回路 2（第 2ch）を意味します。

「LatchID」については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編 (SBCE-433)』を参照してください。

InputDrive

- トリガとして_mcEncoderMark（Z 相）と_mcEXT（外部入力）の選択が可能です。
- _mcEncoderMark（Z 相）では、サーボドライバまたは NX シリーズ パルス出力ユニットの Z 相をトリガに使用します。
  _mcEXT（外部入力）では、サーボドライバまたは NX シリーズ パルス出力ユニットに入力された外部トリガ信号をトリガに使用します。
- _mcEXT（外部入力）は、オムロン製サーボドライバ 1S シリーズでは、設定されている Ext1, Ext2 のどちらかになります。G5 シリーズでは、設定されている Ext1, Ext2, Ext3 のうちの 1 つになります。設定は Sysmac Studio により行います。
  ドライバに設定される 2 つのトリガは同じ設定でも可能です。
- NX シリーズ パルス出力ユニットを使用する場合は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。

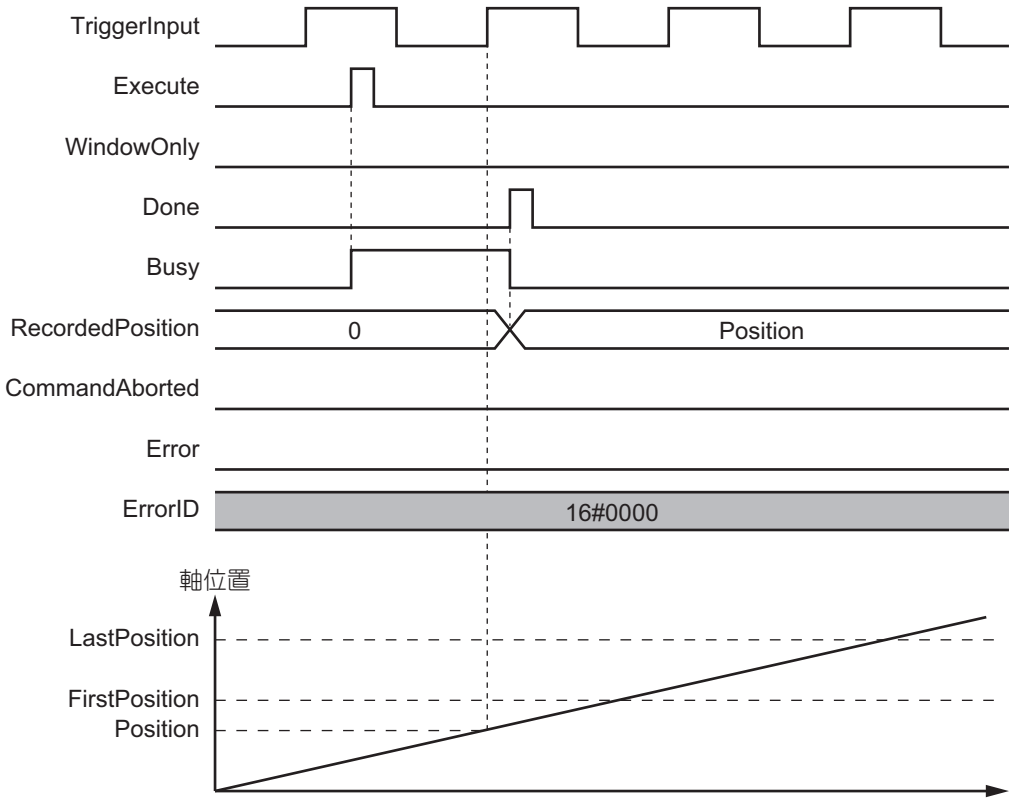
● WindowOnly（ウィンドウ有効）

- 「WindowOnly」（ウィンドウ有効）ではウィンドウの有効／無効を指定します。
- 「Disable」（無効）を指定した場合、すべての軸位置においてトリガを検出します。
- 「Enable」（有効）を指定した場合、軸位置が「FirstPosition」（開始位置）と「LastPosition」（終了位置）の範囲内にあるときだけトリガを検出します。

「WindowOnly」（ウィンドウ有効）の指定による動作の違いをタイミングチャートで示します。

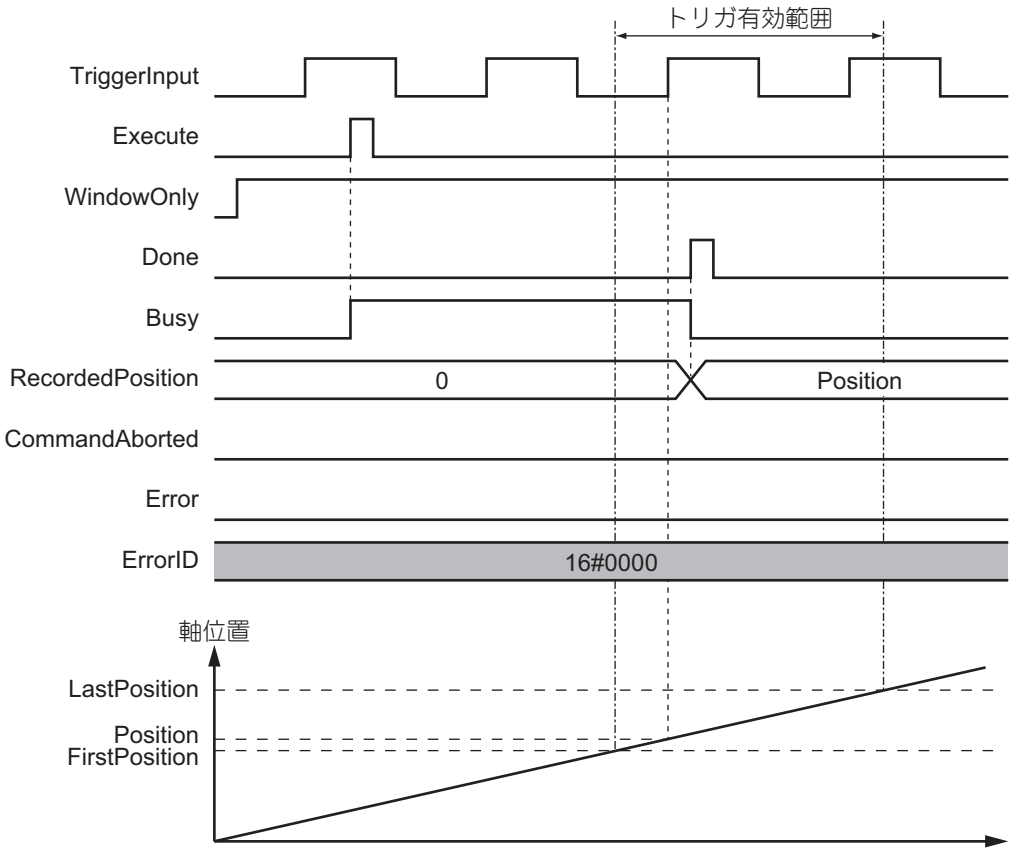
「WindowOnly」が Disable の場合

「Execute」（起動）が TRUE となった後の最初のトリガ発生時の軸位置を定寸距離の基準位置とします。



「WindowOnly」が Enable の場合

ウィンドウの範囲内だけ、トリガ入力を検出して軸位置を取得します。



使用上の注意

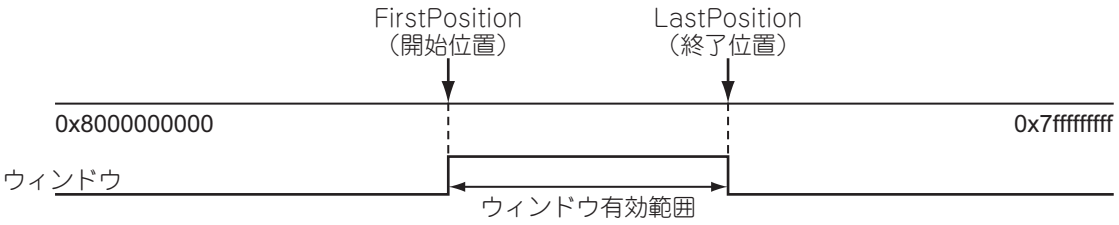
- 「WindowOnly」（ウィンドウ有効）が FALSE→TRUE に変化する瞬間とラッチ機能が動作するまでの時間はラッチできません。
- ラッチ機能が動作するまでに時間が必要なため、「WindowOnly」（ウィンドウ有効）の有効範囲がきわめて短い場合はラッチすることができません。ラッチ可能な有効範囲の限界値は、サーボドライバ、エンコーダ入力ターミナル、位置インタフェースユニットなどの性能および EtherCAT の通信に依存します。

また、カウントモードの違いにより「FirstPosition」（開始位置）と「LastPosition」（終了位置）の範囲は以下のようになります。

リニアモード

- 「FirstPosition」（開始位置）≦ ウィンドウ範囲 ≦「LastPosition」（終了位置）
- 「FirstPosition」（開始位置）＞「LastPosition」（終了位置）と指定した場合は異常となります。
- ［リニアモード］の位置範囲を超えて指定した場合も異常となります。
- また「FirstPosition」（開始位置）と「LastPosition」（終了位置）は LREAL 型の変数のため、「FirstPosition」（開始位置）＝「LastPosition」（終了位置）を指定しないでください。
  LREAL 型については、「実数（REAL 型、LREAL 型）の扱い (P.1-12)」を参照してください。

［リニアモード］のときの、ウィンドウ有効範囲は下図のようになります。

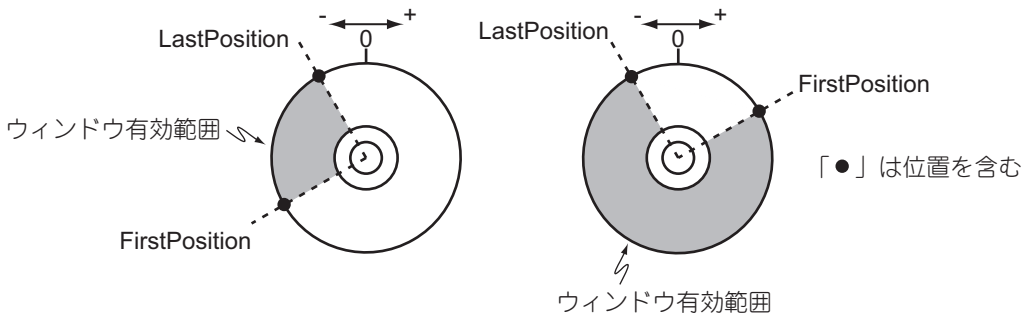


（注）「FirstPosition」（開始位置）と「LastPosition」（終了位置）は、ウィンドウ有効範囲に含まれます。

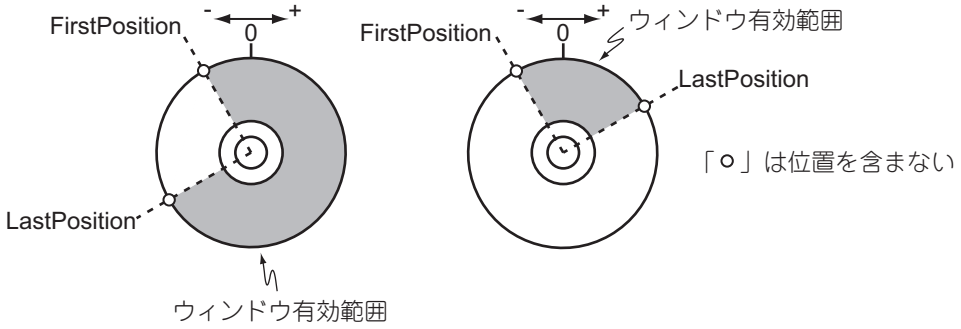
ロータリモード

- 「FirstPosition」（開始位置）≦「LastPosition」（終了位置）、および「FirstPosition」（開始位置）＞「LastPosition」（終了位置）の両方が指定可能です。
- 「FirstPosition」（開始位置）＞「LastPosition」（終了位置）の場合はリングカウンタの上下限位置をまたぐ設定となります。
- リングカウンタ上下限の範囲を超えて指定した場合は異常となります。

FirstPosition ≦ LastPosition



FirstPosition > LastPosition



● ReferenceType（位置種別選択）

位置の種類は以下のようになります。

- _mcFeedback: 同一タスク周期での取得値
  同一タスク周期で取得した主軸のフィードバック位置を使用します。
  ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。

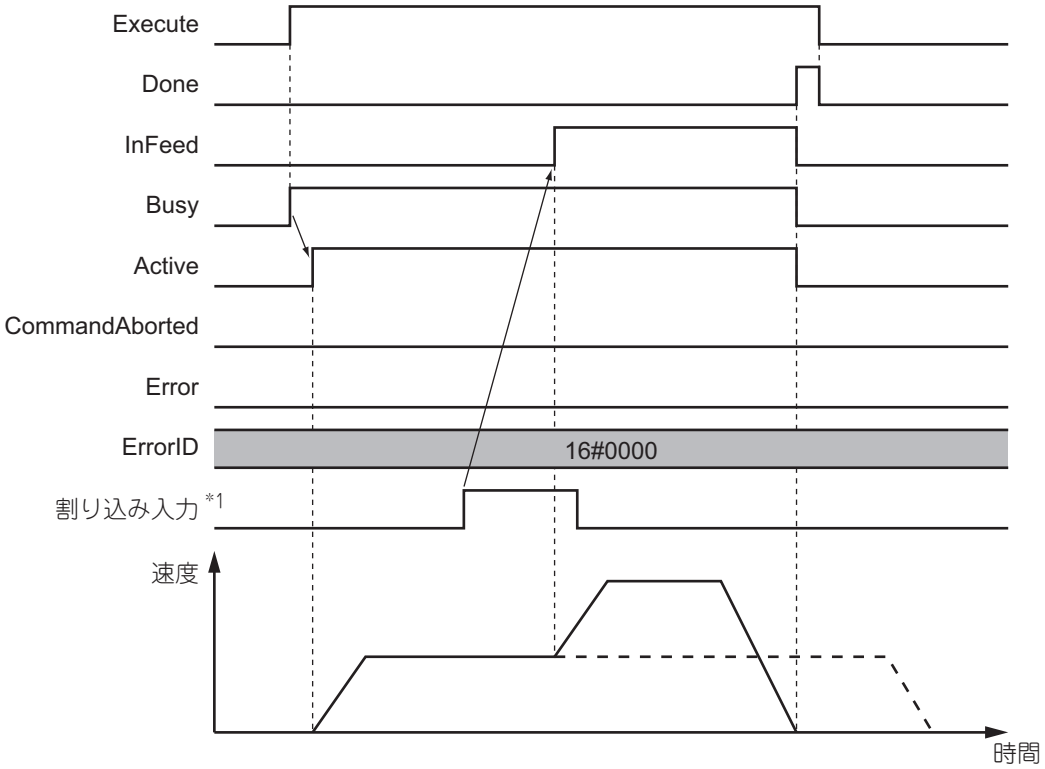
● FeedDistance（定寸距離）

割り込み入力前と同じ方向に定寸動作を行う場合、「FeedDistance」（定寸距離） には正値を指定します。割り込み入力前と逆方向に定寸動作を行い場合には、「FeedDistance」（定寸距離）に負値を指定します。
例えば、割り込み入力前動作が負方向に動作している状態で 「FeedDistance」（定寸距離） に正値を指定すると、定寸動作は負方向に動作します。「FeedDistance」（定寸距離）に負値を指定すると、定寸動作は正方向に動作します。

タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- 割り込み入力後、「InFeed」（定寸送り中）が TRUE となり「FeedDistance」（定寸距離）に到達し、位置決めが完了すると「Done」（完了）が TRUE となります。
- ほかの命令により本命令が中断された場合は、「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）が FALSE となります。

●「MoveMode」（移動方法選択）が_mcAbsolute、_mcRelative の場合

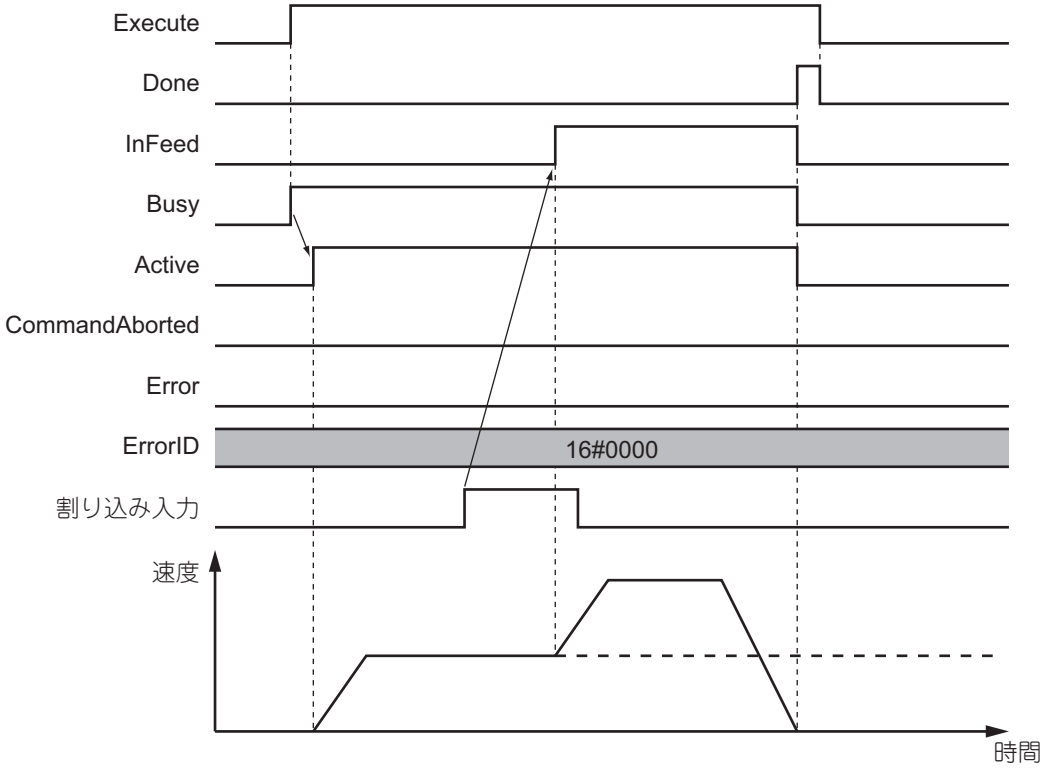


*1. 割込信号が「ON」になってから、「InFeed」が TRUE となるまでに複数制御周期遅れる場合があります。

参考

割り込み定寸位置決めへの加速および減速は入力変数の「Acceleration」（加速度）、「Deceleration」（減速度）に従います。

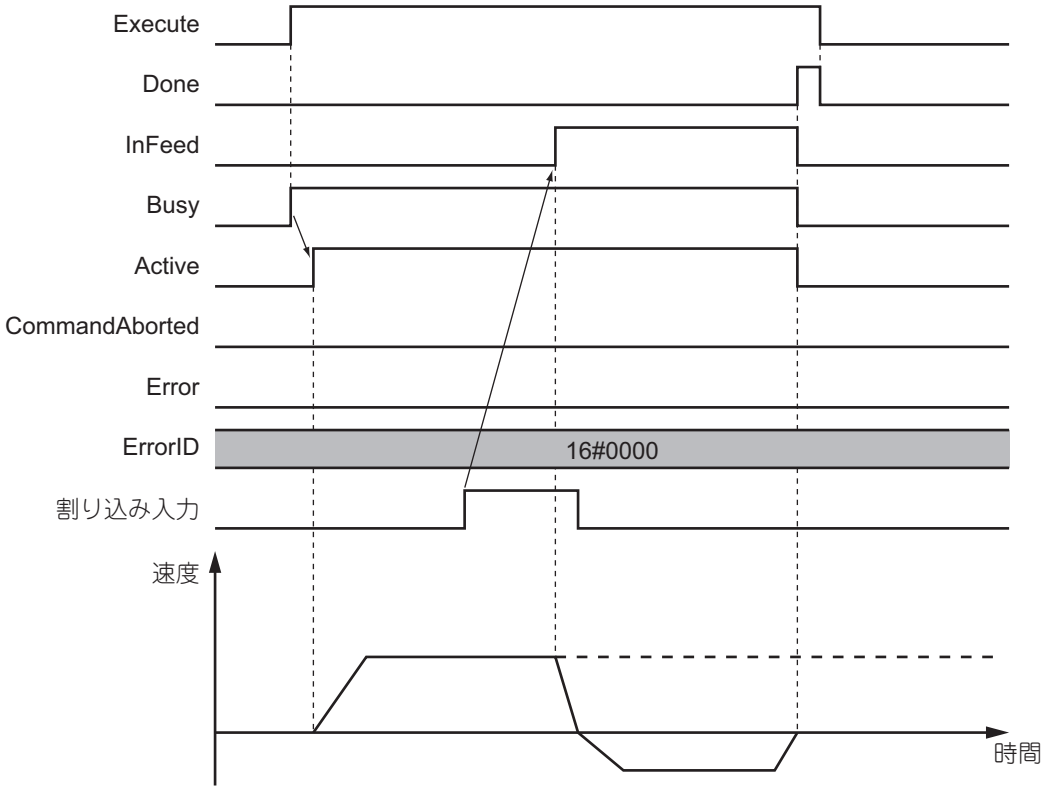
● **「MoveMode」（移動方法選択）が_mcVelocity の場合**



● **軸パラメータの反転時動作**

割り込み後の定寸位置決めが反転する場合は、軸パラメータの［反転時動作］に従って反転します。

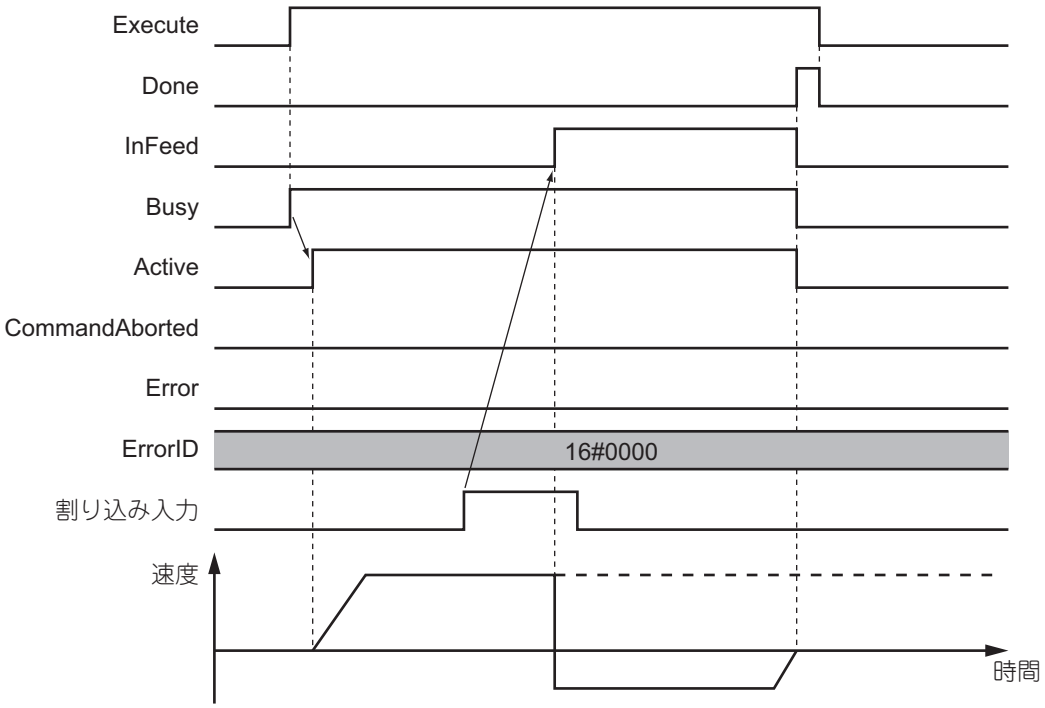
［0: 反転時減速停止で反転する］場合



参考

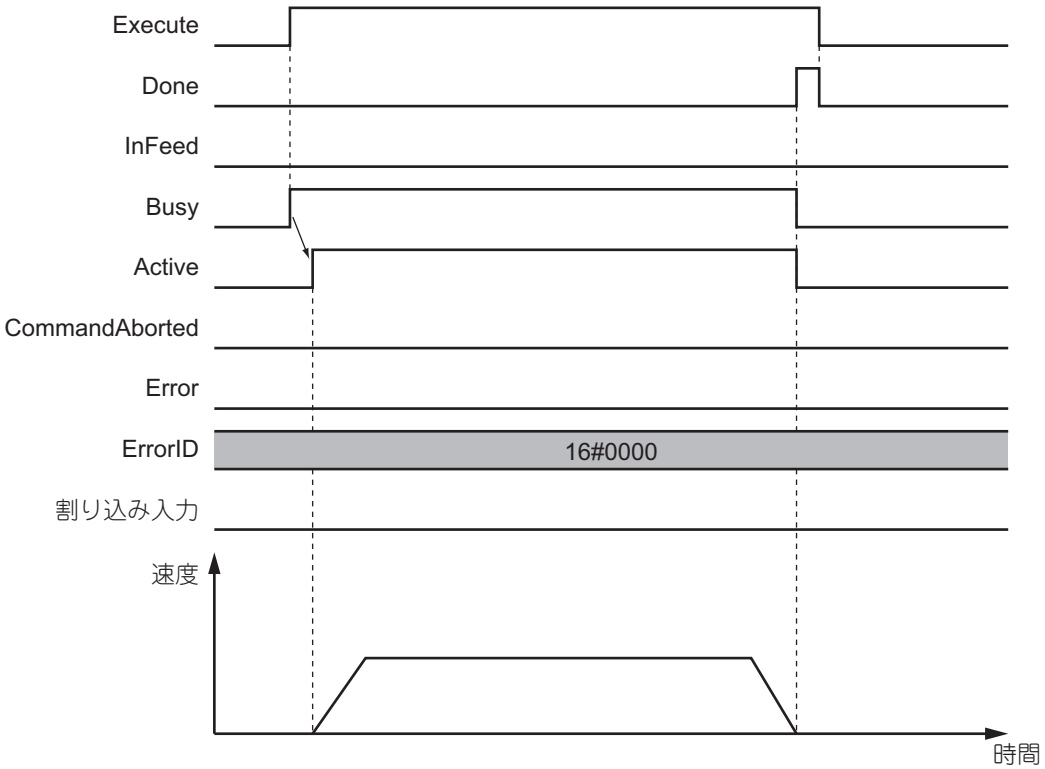
割り込み入力時の反転の減速度は入力変数の「Deceleration」（減速度）に従います。

［1: 反転時即停止で反転する］場合

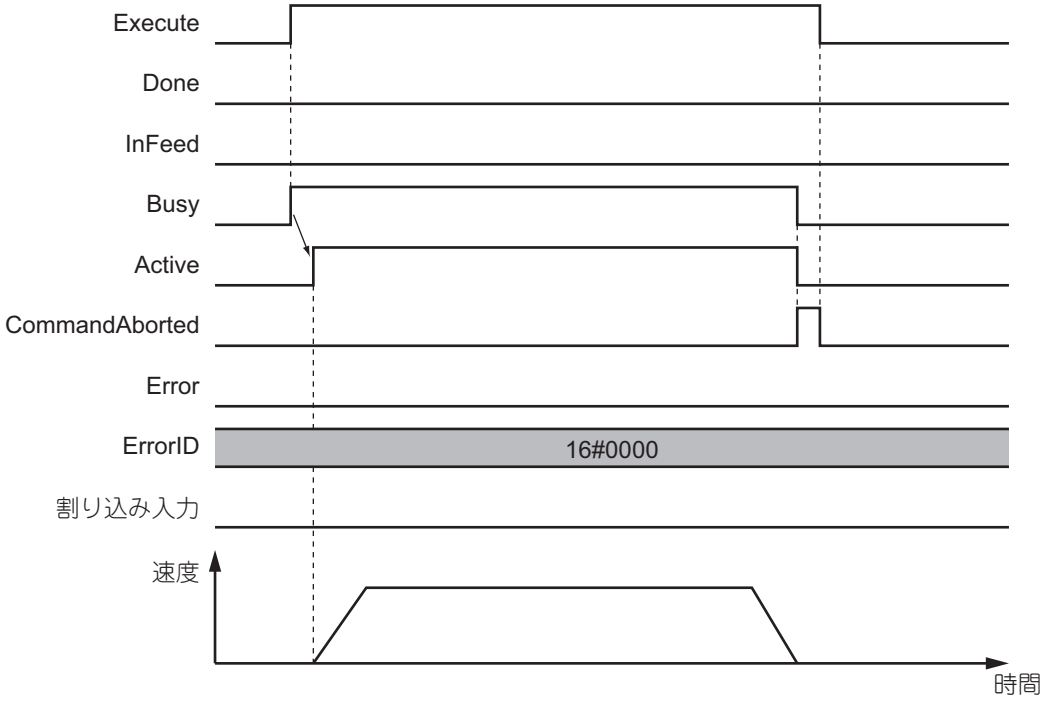


●「MoveMode」（移動方法選択）が_mcAbsolute の場合で、割り込み入力がない場合

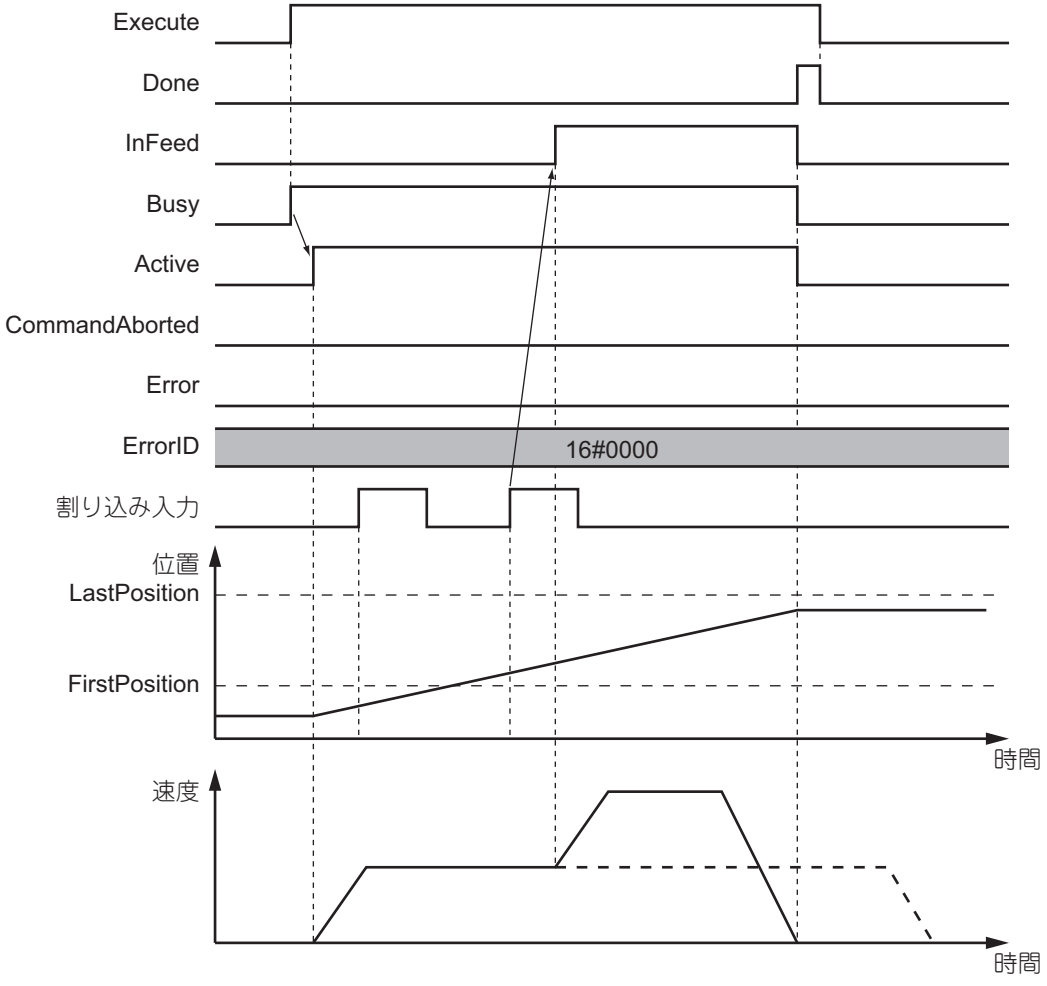
「ErrorDetect」（エラー検出選択）が FALSE で割り込み入力がない場合



「ErrorDetect」（エラー検出選択）が TRUE で割り込み入力がない場合



● **「WindowsOnly」（ウィンドウ有効）が TRUE の場合**



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して、本命令を起動することにより、本命令への切り替えや本命令のバッファリングを行うことができます。
各軸に 1 つまでバッファリングすることが可能です。
多重起動時の本命令の動作を、「BufferMode」（バッファモード選択）で指定します。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。
バッファード	現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

本命令が実行中に、［アボーティング］による多重起動が可能です。

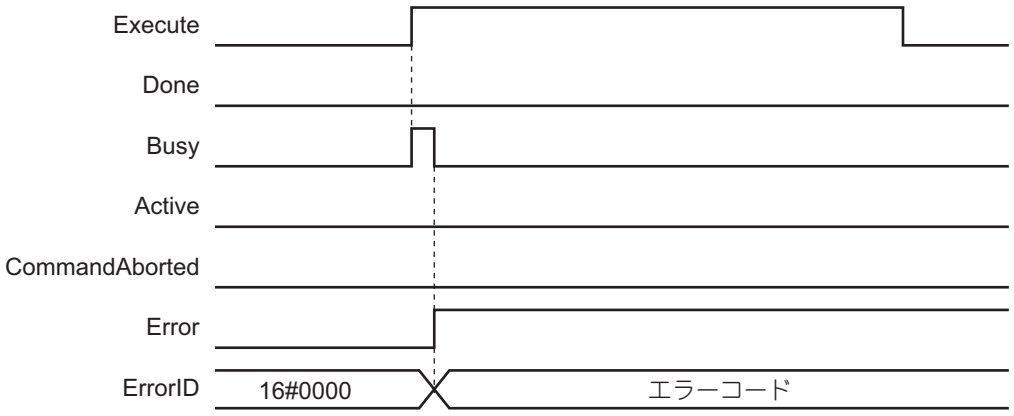
［バッファード］、［ブレンディング］による多重起動を行った場合、次のような動作となります。
- 多重起動を行った命令は、「Error」（エラー）が TRUE となります。同時に「ErrorID」（エラーコード）には「モーション命令多重起動不可（エラーコード: 543C Hex）」が出力されます。
- 本命令の処理は中止され、「CommandAborted」（実行中断）が TRUE となります。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート

異常発生時、本命令で使用している割り込み入力用のラッチは、無効になります。



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、速度制御から割り込み定寸位置決めに変わる制御についてサンプルプログラムを掲載します。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

軸種別

軸	軸種別
軸 1	サーボ軸
軸 2	サーボ軸

カウンタモード

軸	カウンタモード
軸 1	ロータリモード
軸 2	リニアモード

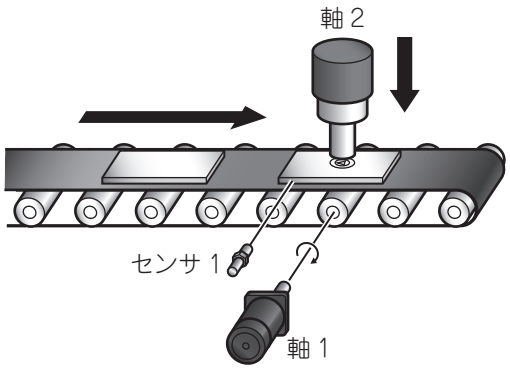
リングカウンタ

軸	上限値	下限値
軸 1	360	0

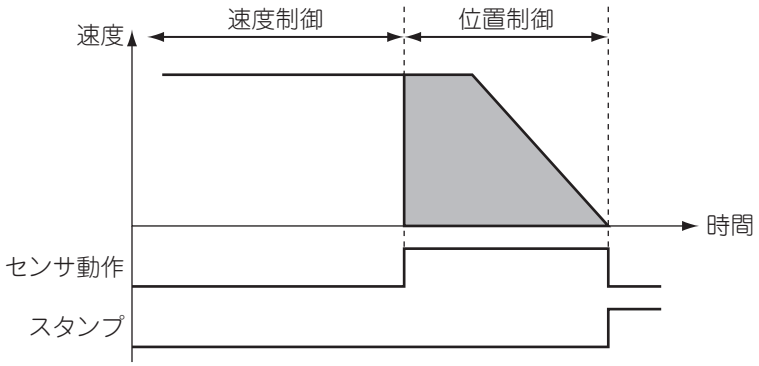
座標単位選択

軸	座標単位
軸 1	degree
軸 2	mm

動作例



● 動作パターン



1 **コンベア動作**

ベルトコンベアを動作させる軸 1 は、割り込み定寸位置決め前の動作で速度制御を行っています。

2 **定寸送り**

センサ 1 はラッチ機能 1 に接続しています。
センサ 1 が「ON」すると定寸位置決めに切り替わり、所定の位置で停止します。

3 **スタンプの押印**

位置決めが完了したら、スタンプの軸 2 が絶対位置決めにより垂直に降下し、押印を行います。
押印後、軸 2 は原点位置に戻ります。

絶対位置決めの動作完了と同時に原点位置に復帰させるため、MC_MoveZeroPosition（高速原点復帰）の「BufferMode」（バッファモード選択）をバッファードで起動します。
前の命令の出力「Active」が TRUE のときに、命令多重起動を行います。

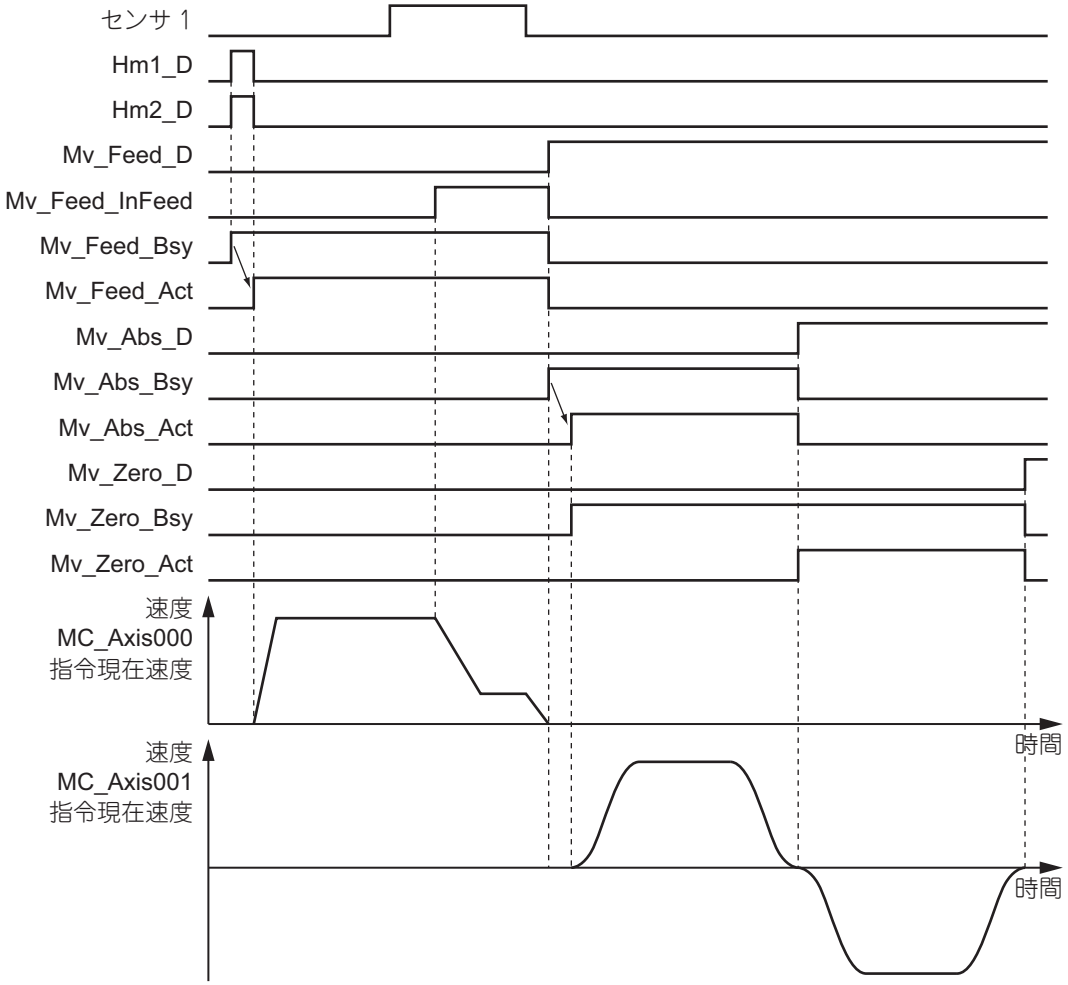
ラダー図

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。

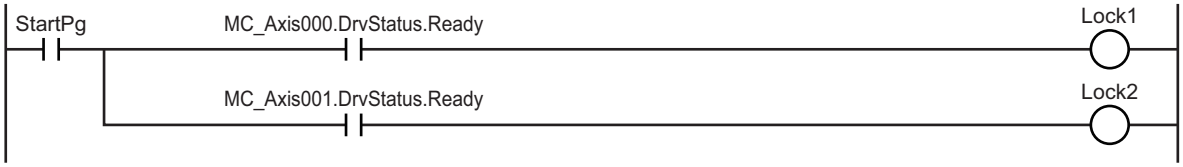
名称	データ型	初期値	コメント
MC_Axis001	_sAXIS_REF	−	軸 2 の軸変数です。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態のとき、TRUE となります。
TrigRef	_sTRIGGER_REF	−	割り込み入力の指定変数です。 このプログラムでは、サーボドライバのラッチ機能 1 を使用します。外部入力（センサ 1）の立ち上がりを検出し、定寸移動を実行します。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート

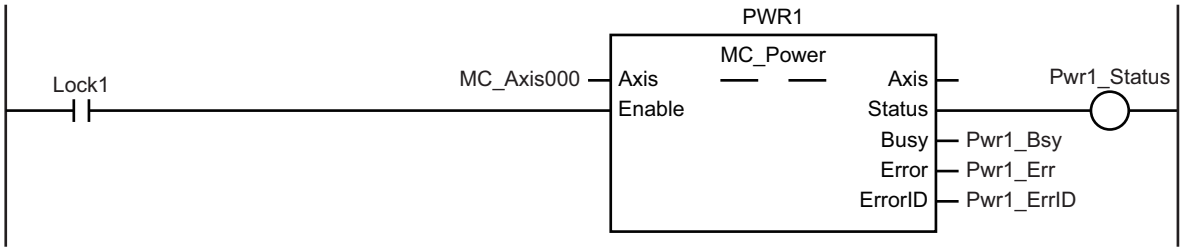


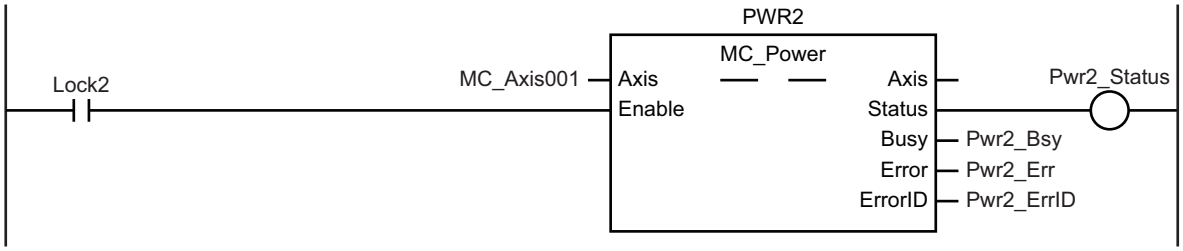
● サンプルプログラム

接点 StartPg が TRUE になったら、各軸のサーボドライバがサーボレディ状態になっていることを確認する

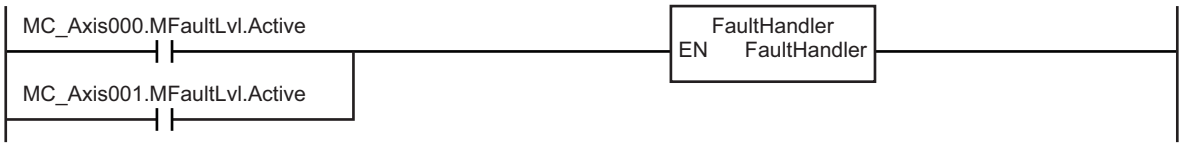


サーボレディ状態になっていれば、各軸をサーボ ON 状態にする

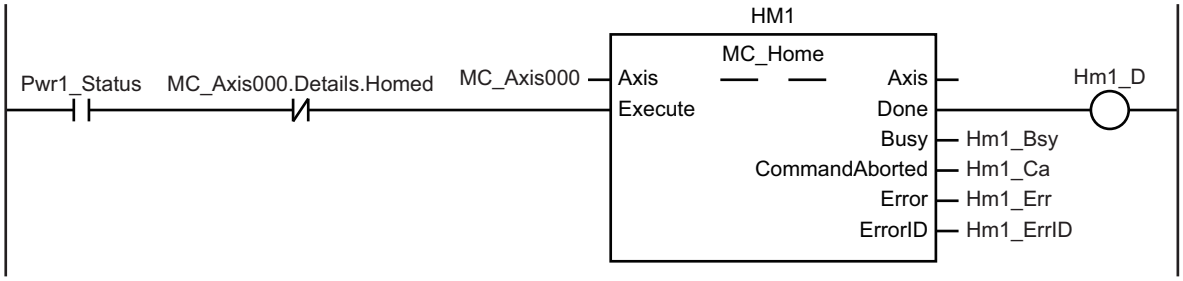




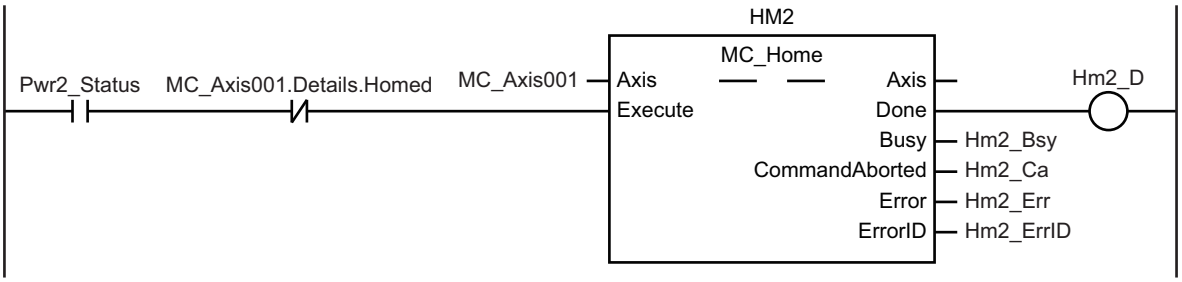
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



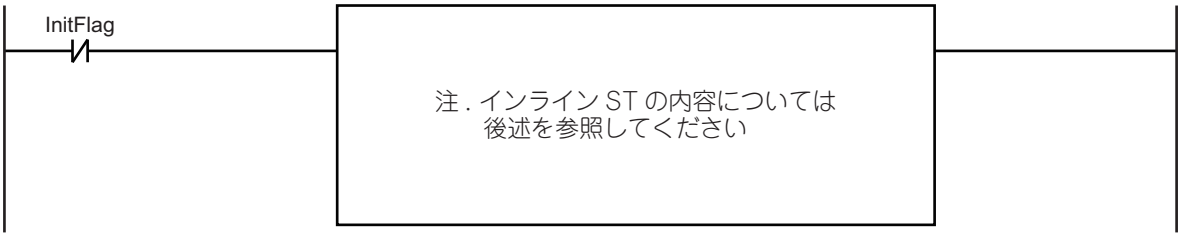
軸 1 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行する



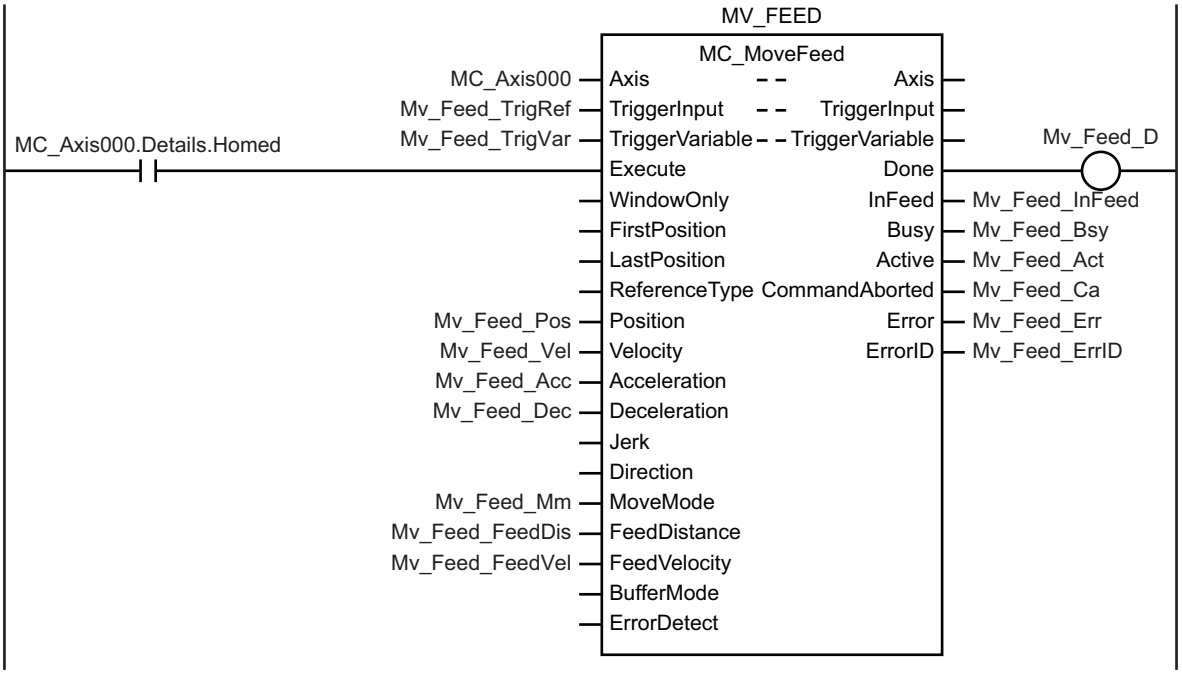
軸 2 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行する



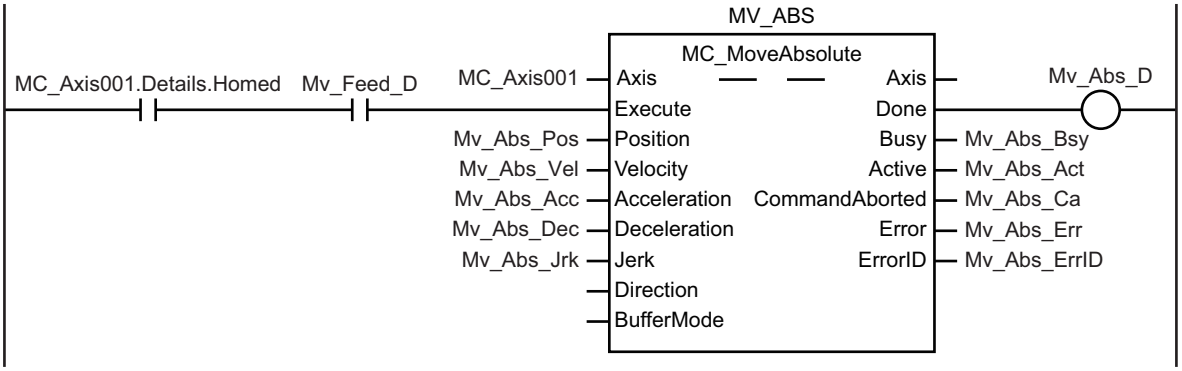
割り込み定寸位置決め、絶対値位置決め、高速原点復帰のパラメータを設定する



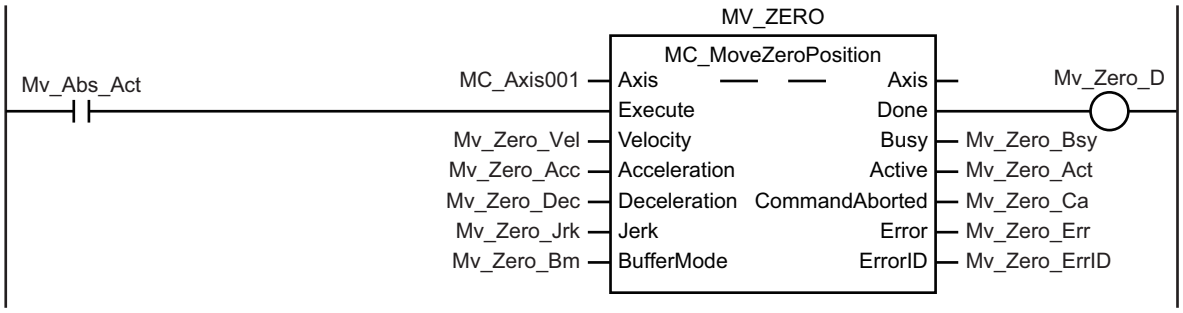
軸 1 が原点確定状態になったことを確認し、軸 1 で割り込み定寸を実行する



軸 2 が原点確定状態で、軸 1 の割り込み定寸位置決めが完了した場合に、軸 2 で絶対値位置決めを実行する



絶対値位置決め完了後、高速原点復帰を実行し、原点へ移動する



インライン ST の内容

```
//MV_FEED パラメータ
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 パラメータ
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 パラメータ
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;

//入力パラメータを設定後、InitFlag を TRUE にします
InitFlag := TRUE;
```

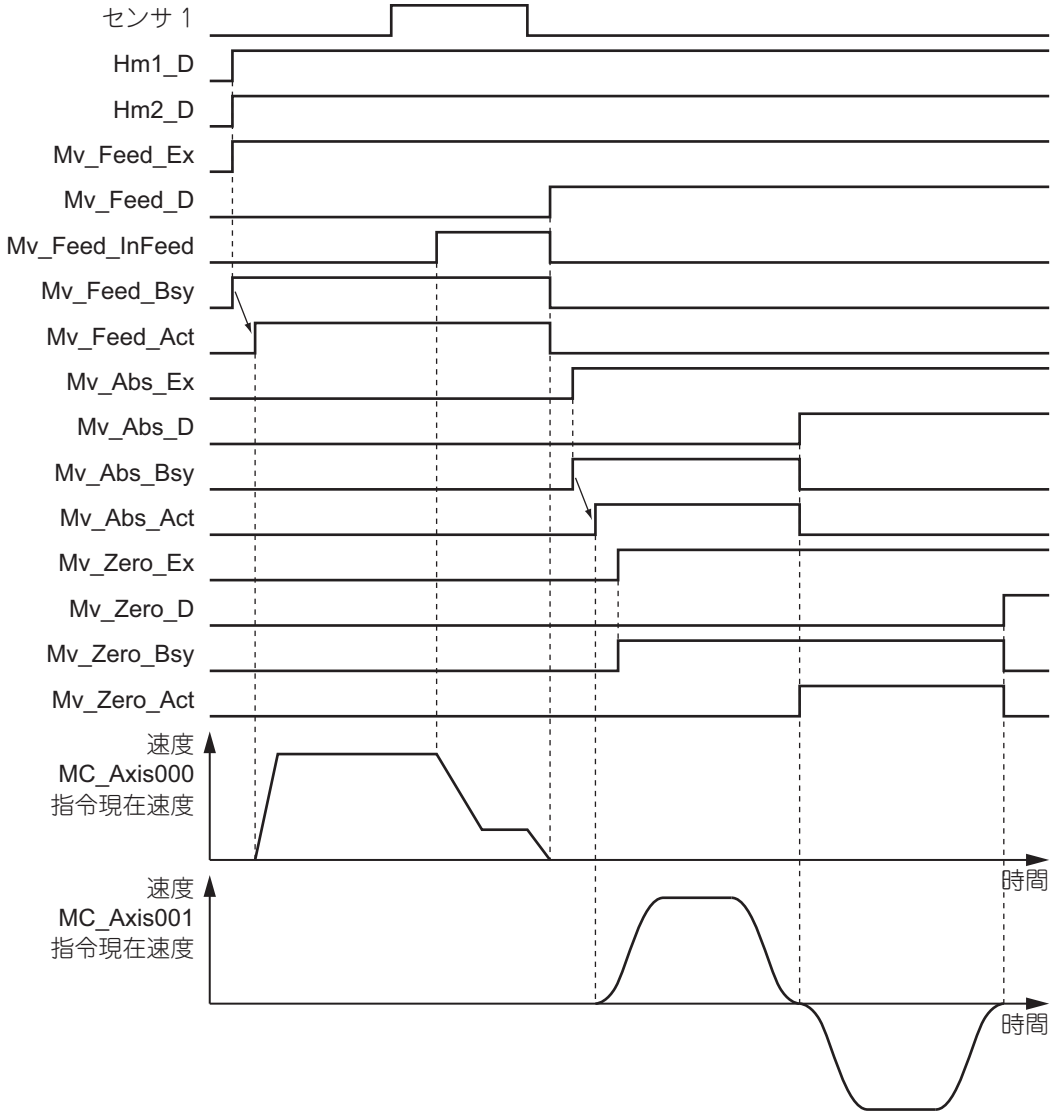
ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。

名称	データ型	初期値	コメント
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
MC_Axis001	_sAXIS_REF	−	軸 2 の軸変数です。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態のとき、TRUE となります。
TrigRef	_sTRIGGER_REF	−	割り込み入力の指定変数です。 このプログラムでは、サーボドライバのラッチ機能 1 を使用します。外部入力（センサ 1）の立ち上がりを検出し、定寸移動を実行します。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。
Hm1_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、 MC_Home のインスタンス HM1 が起動します。
Hm2_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、 MC_Home のインスタンス HM2 が起動します。
Mv_Feed_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、 MC_MoveFeed のインスタンス MV_FEED が起動します。
Mv_Abs_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、 MC_MoveAbsolute のインスタンス MV_ABS が起動します。
Mv_Zero_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、 MC_MoveZeroPosition のインスタンス MV_ZERO が起動します。

● タイミングチャート



● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag=FALSE THEN

    //MV_FEED パラメータ
    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 パラメータ
    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 パラメータ
    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;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag := TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE; //軸 1 をサーボ ON にします
ELSE
    Pwr1_En:=FALSE; //軸 1 をサーボ OFF にします
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 2 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE; //軸 2 をサーボ ON にします
ELSE
    Pwr2_En:=FALSE; //軸 2 をサーボ OFF にします
END_IF;

//軽度フォールトレベルの異常発生時の処理
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE)THEN
    FaultHandler();
END_IF;

//軸 1 がサーボ ON 状態で、かつ原点未確定のとき、軸 1 の原点復帰を行います
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//軸 2 がサーボ ON 状態で、かつ原点未確定のとき、軸 2 の原点復帰を行います
```

```
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//軸 1 が原点確定後に、MC_MoveFeed を実行します
IF MC_Axis000.Details.Homed=TRUE THEN
    Mv_Feed_Ex:=TRUE;
END_IF;

//軸 2 が原点確定状態で、かつ軸 1 の割込み定寸位置決めが完了した場合に、軸 2 の絶対位置決めを実行します
IF (MC_Axis001.Details.Homed=TRUE) AND (Mv_Feed_D=TRUE) THEN
    Mv_Abs_Ex := TRUE;
END_IF;

//MC_MoveAbsolute が開始されると、MC_MoveZeroPosition を多重起動します
IF Mv_Abs_Act=TRUE THEN
    Mv_Zero_Ex := TRUE;
END_IF;

//軸 1 の MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

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

//軸 1 の MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

//軸 2 の 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
);

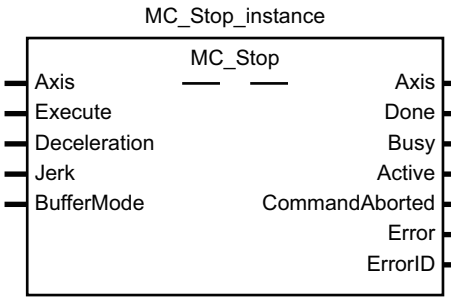
//MC_MoveAbsolute
MV_ABS(
    Axis := MC_Axis001,
    Execute := Mv_Abs_Ex,
    Position := Mv_Abs_Pos,
    Velocity := Mv_Abs_Vel,
    Acceleration := Mv_Abs_Acc,
    Deceleration := Mv_Abs_Dec,
    Jerk := Mv_Abs_Jrk,
    Done => Mv_Abs_D,
    Busy => Mv_Abs_Bsy,
    Active => Mv_Abs_Act,
    CommandAborted => 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

軸を減速停止させます。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_Stop	強制停止	FB		MC_Stop_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, BufferMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は、[指令単位／s^2]となります。*1
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は、[指令単位／s^3]となります。*1
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *2	モーション命令多重起動時の動作を指定します。 0: アボーティング

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。

出力変数	名称	データ型	有効範囲	内容
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	減速停止が完了し、速度が「0」に到達したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの MC_Stop（強制停止）命令による命令多重起動（[アボーティング]）で、本命令が中断されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 軸を現在速度から速度「0」へと減速制御します。
- 「Execute」（起動）の立ち上がりで、減速停止の動作を開始します。
- MC_Stop（強制停止）命令の起動により動作中の命令は「CommandAborted」（実行中断）します。

使用上の注意

軸の状態を表す「Status.ErrorStop」（エラー減速停止中）が TRUE のとき、本命令は起動できません。
エラー減速停止中に軸の動作を停止させるには、MC_ImmediateStop（即停止）命令 を使用してください。

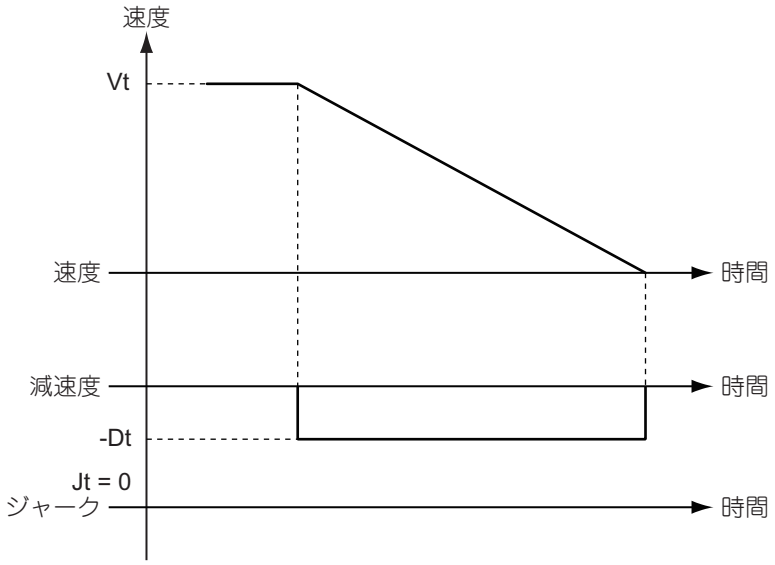
命令の詳細

以下に本命令の詳細について説明します。

●「Deceleration」（減速度）、「Jerk」（ジャーク）の指定

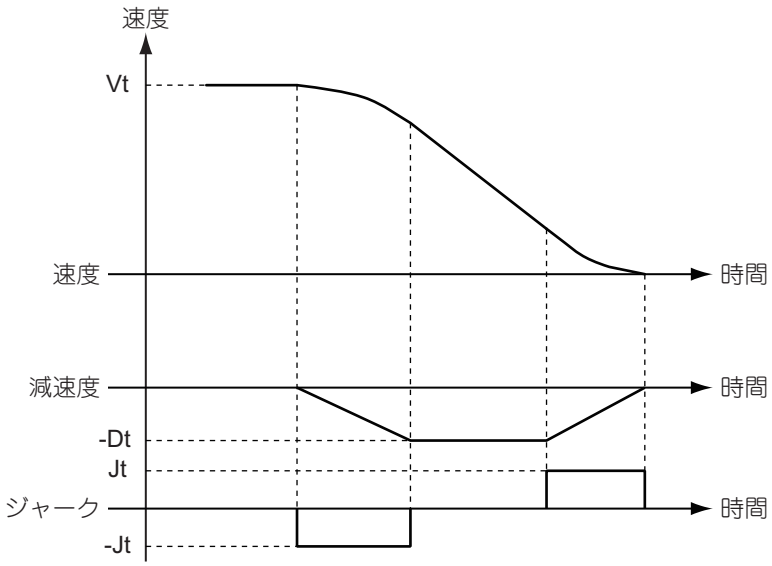
減速停止時の減速度、ジャークは、入力変数の「Deceleration」（減速度）、「Jerk」（ジャーク）で指定します。
「Jerk」（ジャーク）に「0」を指定した場合と「0」以外を指定した場合の減速度、速度の関係は次のようになります。

「Jerk」（ジャーク）が「0」のとき
減速度 Dt で、速度の指令値を作成します。



Vt: 減速起動時の速度、Dt: 減速度の指定値、Jt: ジャークの指定値

「Jerk」（ジャーク）が「0」以外のとき
減速度の上限を Dt として、現在速度を基点とし、速度の指令値を作成します。



Vt: 減速起動時の速度、Dt: 減速度の指定値、Jt: ジャークの指定値

参考

「Deceleration」（減速度）に「0」を指定すると、即停止の命令となります。減速せずに、すぐに停止します。
「Deceleration」（減速度）に「0」を指定した場合だけ、［加減速オーバー］の設定にかかわらず、即停止を実行します。

●「BufferMode」（バッファモード選択）の指定

1 つ前の軸動作と今回の動作のつなぎ方を指定します。
本命令の「BufferMode」（バッファモード選択）は将来拡張用としての予約パラメータです。
次の 1 種類があります。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令を実行します。

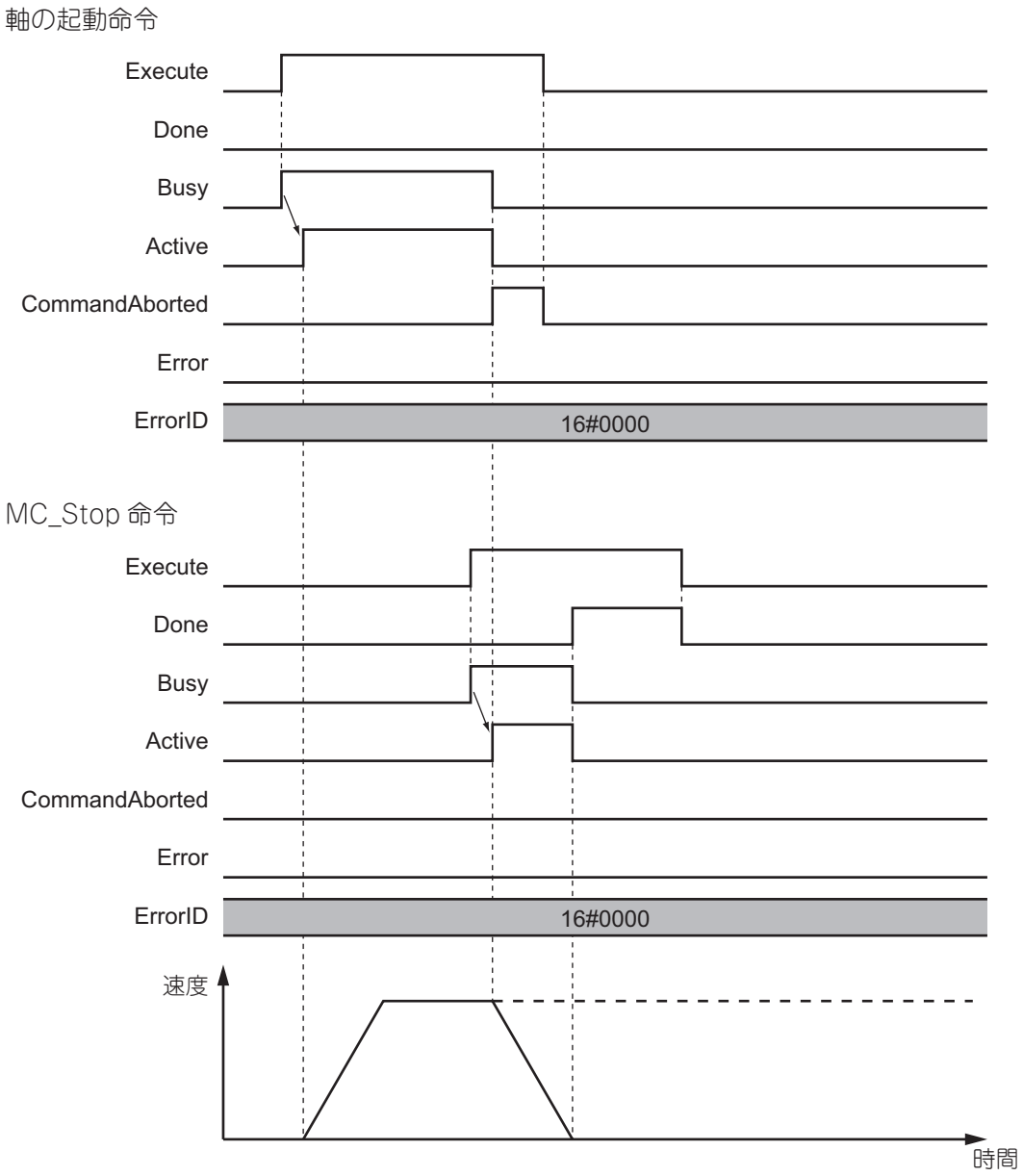
「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● インポジションチェック

本命令による停止時に、インポジションチェックは行いません。

タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- 速度「0」に到達すると、「Done」（完了）が TRUE となります。
- ほかの命令により、本命令が中断された場合は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）が FALSE となります。

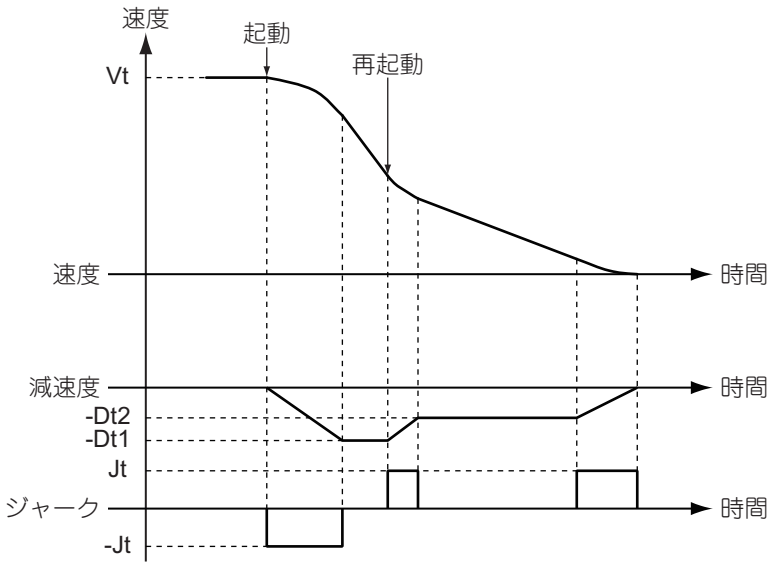


モーション命令再起動

本命令実行中に、再度、「Execute」（起動）を立ち上げると、「Deceleration」（減速度）の変更を行います。
モーション命令再起動では「Jerk」（ジャーク）の変更は行いません。

●「ジャーク」が「0」以外のとき

現在速度、現在減速度を基点とし、変更後の減速度の上限を Dt2 として速度の指令値を作成します。



Vt: 減速起動時の速度、Dt: 減速度の指定値、Jt: ジャークの指定値

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

軸グループ動作中

指定した軸が軸グループ動作中に、MC_Stop（強制停止）命令を実行すると軸は異常となります。軸グループも異常となり、軸グループ動作も止まります。

軸変数の「Status.ErrorStop」（エラー減速停止中）が TRUE のとき

軸で異常が発生している間、軸変数の「Status.ErrorStop」（エラー減速停止中）は TRUE となります。

エラー減速停止中に MC_Stop（強制停止）命令を起動した場合、正常起動できずに「CommandAborted」（実行中断）が TRUE になります。異常が発生している軸を停止するには、MC_ImmediateStop（即停止）命令を使用してください。

MC_ResetFollowingError（偏差カウンタリセット）命令実行中

MC_ResetFollowingError（偏差カウンタリセット）命令の実行中に MC_Stop（強制停止）命令を起動すると、MC_ResetFollowingError（偏差カウンタリセット）命令の「CommandAborted」（実行中断）が TRUE となり、MC_Stop（強制停止）命令が実行されます。ただし、本命令の「Deceleration」（減速度）は適用されずに即停止となります。

● 本命令実行中の他命令起動

- MC_Stop（強制停止）命令による軸の減速停止以外にも、以下の命令を起動した場合、軸変数の「Status.Stopping」（減速停止中）は TRUE となります。
  MC_ResetFollowingError（偏差カウンタリセット）命令
  MC_TouchProbe（外部ラッチ有効）命令の即停止中
  ほかの MC_Stop（強制停止）命令の「Execute」（起動）が 1 つでも TRUE になったとき。
- 「Status.Stopping」（減速停止中）が TRUE の軸に対しては、以下のようになります。
  a）単軸位置決め、連続動作、同期動作、手動動作を行うと、これらの命令の「CommandAborted」（実行中断）が TRUE となります。

b）MC_Stop（強制停止）命令実行中に MC_ResetFollowingError（偏差カウンタリセット）命令を起動すると、MC_Stop（強制停止）命令の「Done」（完了）が TRUE となり、MC_ResetFollowingError（偏差カウンタリセット）命令が実行されます。

c）MC_Stop（強制停止）命令同士の多重起動が可能です。実行中の MC_Stop（強制停止）命令の「Done」（完了）が TRUE となります。

・MC_Stop（強制停止）命令が起動した状態で、以下の状態になった場合、MC_Stop（強制停止）命令の「Done」（完了）が TRUE となります。

a）MC_Power（運転可）の入力変数「Enable」（有効）を FALSE にした場合（サーボ OFF にした場合）

b）MC_TouchProbe（外部ラッチ有効）命令の入力変数「StopMode」（停止方法選択）に［1: _mcImmediateStop］を選択した状態で、トリガ条件が成立し、オムロン製サーボドライバ 1S シリーズまたは G5 シリーズが即停止した場合

異常

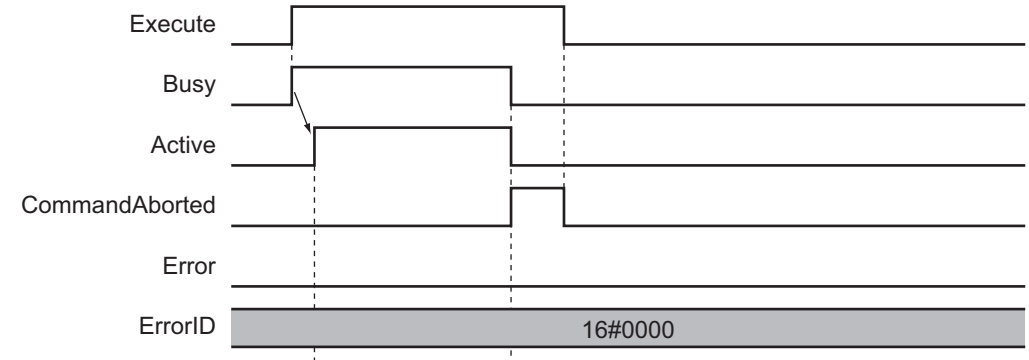
本命令実行中に軸等に異常が発生した場合、動作が停止します。
停止方法は、即停止、減速停止、サーボ OFF のいずれかを、軸パラメータで指定します。
停止方法に減速停止が指定されている場合、減速停止動作を継続します。

軸パラメータの停止方法の設定については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

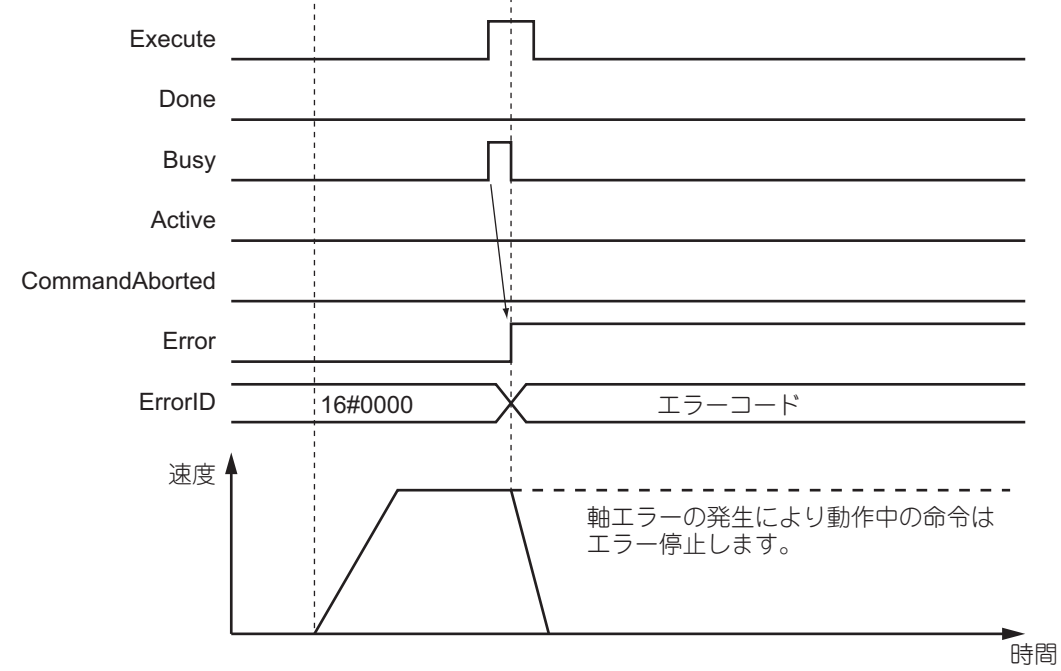
● 異常発生時のタイミングチャート

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

MC_MoveVelocity



MC_Stop

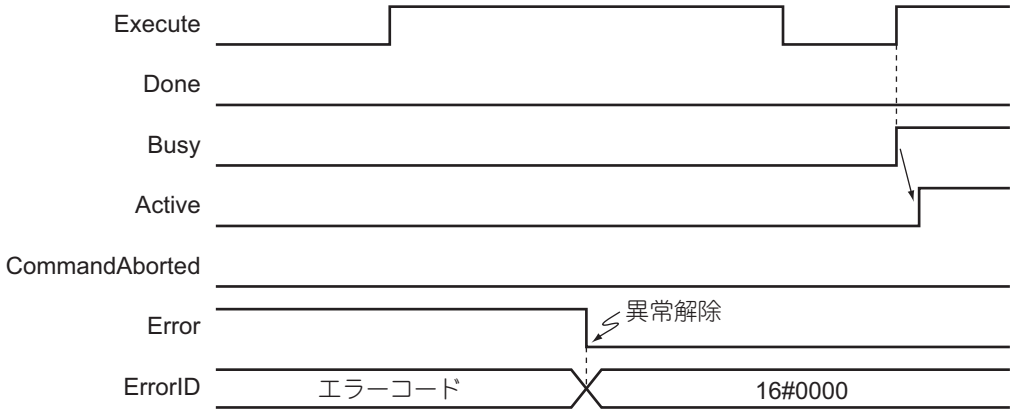


バージョン関連情報

異常解除時の動作は、CPU ユニットのバージョンにより、次のようになります。
なお、異常の解除は軸が停止してから行ってください。軸動作中は異常の解除ができません。

- **ユニットバージョン Ver.1.10 以降の CPU ユニットの場合**
  本命令の異常解除を行った場合、次の「Execute」（起動）の立ち上がりまで本命令は起動しません。

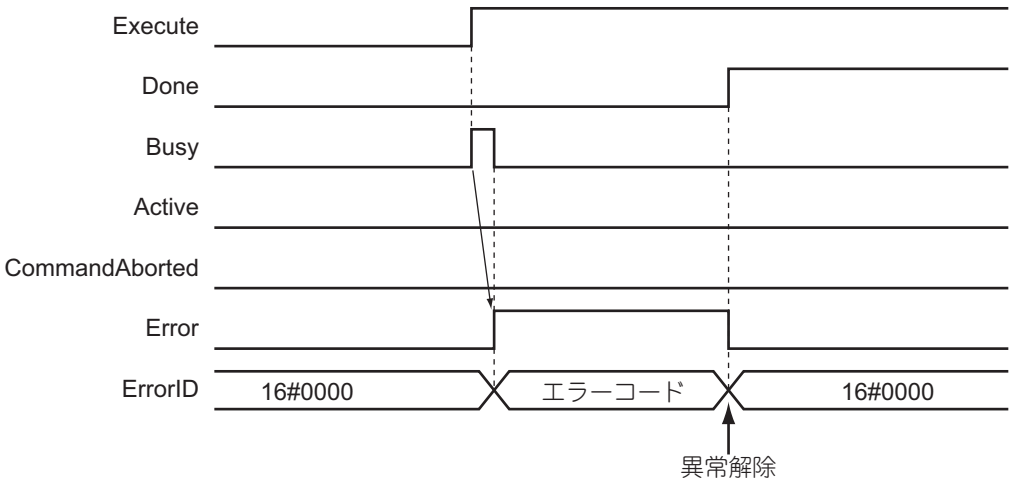
Ver.1.10 以降



- **ユニットバージョン Ver.1.09 以前の CPU ユニットの場合**
  本命令で異常が発生している状態で、「Execute」（起動）が TRUE のまま異常解除を行う場合、以下のような動作となります。
  a）異常要因が解除されていた場合、「Error」（エラー）が FALSE、「Done」（完了）が TRUE となります。軸変数の「Status.Stopping」（減速停止中）は、減速停止の正常実行時と同様に TRUE となります。
  b）異常要因が解除されていない場合、本命令は再び「Error」（エラー）になり、軸に異常が発生します。

以下のタイミングチャートは、異常要因が解除している場合です。

Ver.1.09 以前

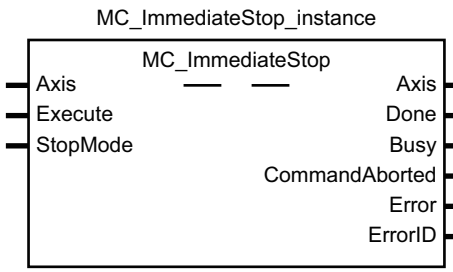


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_ImmediateStop

軸の状態によらず入力変数「StopMode」（停止方法選択）の停止方法に従って停止します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_ImmediateStop	即停止	FB		MC_ImmediateStop_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, StopMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
StopMode	停止方法選択	_eMC_STOP_MODE	1: _mcImmediateStop 2: _mcImmediateStopFEReset 3: _mcFreeRunStop	1 *1	停止方法を選択します。 1: 即停止します 2: 即停止し、偏差カウンタリセットを行います 3: 即停止し、サーボ OFF にします

*1. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	減速停止が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの MC_Stop（強制停止）命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

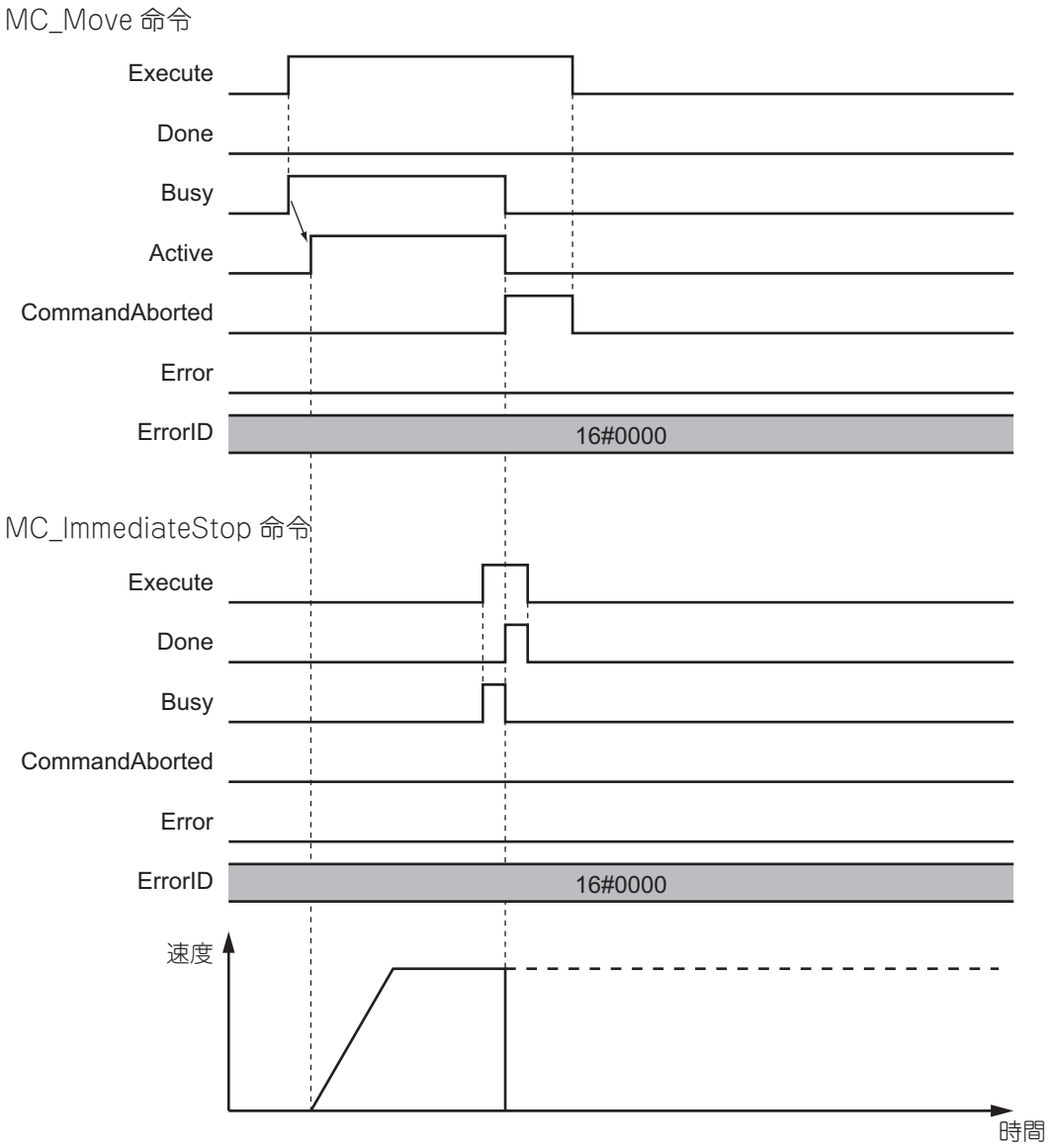
- 本命令は、どのような状態でも実行が可能な命令です。
  例えば、異常による減速停止中でも本命令で即停止にすることが可能です。
  MC_Stop（強制停止）命令の場合は、「Status.ErrorStop」（エラー減速停止中）が TRUE の状態では使用できませんが、MC_ImmediateStop（即停止）命令は使用可能です。
- 命令が実行されると、「StopMode」（停止方法選択）の指定に従い即停止し、動作中の命令は「CommandAborted」（実行中断）となります。
- 本命令を実行すると、軸の状態は「Status.ErrorStop」（エラー減速停止中）が TRUE となり、「即停止命令起動（エラーコード: 5485 Hex）」が発生します。

使用上の注意

主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。

タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となります。
- 即停止命令処理が完了すると、「Done」（完了）が TRUE となります。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

軸グループ動作中

指定した軸が、軸グループ動作中に、命令を実行すると軸は異常となり即停止します。
軸グループも異常となり、軸グループ動作も止まります。

軸変数の「Status.Stopping」（減速停止中）が TRUE のとき

下記の状態の場合、軸変数の「Status.Stopping」（減速停止中）が TRUE となります。
- MC_Stop（強制停止）命令による軸の減速中
- MC_ResetFollowingError（偏差カウンタリセット）命令実行中
- MC_TouchProbe（外部ラッチ有効）命令の即停止中
- MC_Stop（強制停止）命令の「Execute」が多重起動により 1 つでも TRUE となっている間

「Status.Stopping」（減速停止中）が TRUE の軸に対しても本命令は実行可能です。
本命令の起動により、下記実行中の命令は、「CommandAborted」（実行中断）が TRUE となります。
- MC_Stop（強制停止）命令
- MC_ResetFollowingError（偏差カウンタリセット）命令
- 即停止中の MC_TouchProbe（外部ラッチ有効）命令

軸変数の「Status.ErrorStop」（エラー減速停止中）が TRUE のとき
軸で異常が発生している間、軸の状態は「Status.ErrorStop」（エラー減速停止中）が TRUE となります。
異常による減速停止中でも本命令の指令は実行可能となります。

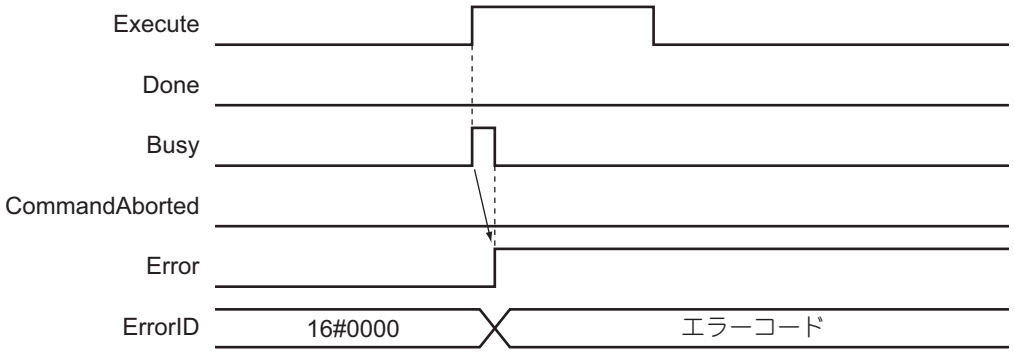
異常

本命令実行中に、軸等に異常が発生した場合も、そのまま即停止動作を継続します。

軸パラメータの停止方法の設定については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 異常発生時のタイミングチャート

本命令起動時に、異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_SetPosition

軸の指令現在位置とフィードバック現在位置を任意の値に変更します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_SetPosition	現在位置変更	FB	MC_SetPosition_instance MC_SetPosition Axis — — Axis Execute — Done Position — Busy ReferenceType — CommandAborted Relative — Error ExecutionMode — ErrorID	MC_SetPosition_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, Position :=「パラメータ」, ReferenceType :=「パラメータ」, Relative :=「パラメータ」, ExcutionMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Position	目標位置	LREAL	負数、正数、「0」	0	絶対座標の目標位置を指定します。 単位は、［指令単位］となります。 *1
ReferenceType	位置種別選択	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback	0 *2	位置の種別を指定します。 0: 指令位置（サーボ軸、仮想サーボ軸） 1: フィードバック位置（エンコーダ軸、仮想エンコーダ軸）
Relative (Reserved)	相対位置選択	BOOL	TRUE, FALSE	FALSE	(Reserved)
ExecutionMode (Reserved)	実行モード選択	_eMC_EXECUTION_MODE	0: _mcImmediately	0 *2	(Reserved)

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。

出力変数	名称	データ型	有効範囲	内容
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	指令現在位置とフィードバック現在位置の変更が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。
CommandAborted	• 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- サーボ軸の指令現在位置を、指定した目標位置に変更します。
  また、エンコーダ軸に対して実行した場合は、フィードバック現在位置を指定した目標位置に変更します。
- 目標位置は、絶対座標の位置で指定します。
- 指令現在位置を変更する時点で同時にフィードバック現在位置も変更されますが、その時点での位置偏差を維持したまま変更を行います。指令サーボ軸に対して実行した場合は、現在位置とフィードバック現在位置の差を維持したままの変更となります。
  このため、本命令実行後、軸のフィードバック現在位置は、次の計算式で示す値になります。

  変更後のフィードバック現在位置 ＝ 目標位置 − 変更前の位置偏差

- サーボ軸に対しフィードバック位置指定をした場合、または、エンコーダ軸に対し指令位置指定をした場合は、位置種別選択異常が発生します。

- カウントモードが［ロータリモード］の場合、目標位置は［リングカウンタ下限値］以上、［リングカウンタ上限値］未満の範囲で設定します。目標位置がこの範囲外であれば、リングカウンタ異常が発生します。
- カウントモードが［リニアモード］の場合、目標位置はソフトウェアリミットの範囲外であっても設定できます。
- 本命令は、軸が停止状態でも、軸が動作中でも使用できます。

使用上の注意

本命令完了後は、指定した軸が原点未確定になります。
このため、本命令完了後、次の機能や命令は使用できませんので、注意してください。
- ソフトウェアリミット機能
- MC_MoveZeroPosition（高速原点復帰）命令
- 多軸協調動作命令（直線補間と円弧補間）

● ReferenceType（位置種別選択）

- サーボ軸、または仮想サーボ軸を使用する場合は、[0: _mcCommand]（指令位置）を選択してください。
- エンコーダ軸、または仮想エンコーダ軸を使用する場合は、[1: _mcFeedback]（フィードバック位置）を選択してください。

● 軸種別と位置種別の関係

監視可能な軸種別と監視する位置種別の関係は以下のようになります。

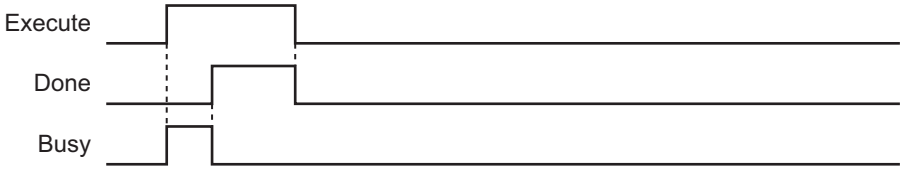
軸種別	ReferenceType	
	_mcCommand	_mcFeedback
サーボ軸	○	×
エンコーダ軸	× *1	○
仮想サーボ軸	○	×
仮想エンコーダ軸	× *1	○

*1. 命令起動時に、「位置種別選択範囲外（エラーコード: 5430 Hex）」が発生します。

タイミングチャート

● 軸が停止状態の時に本命令を使用する場合

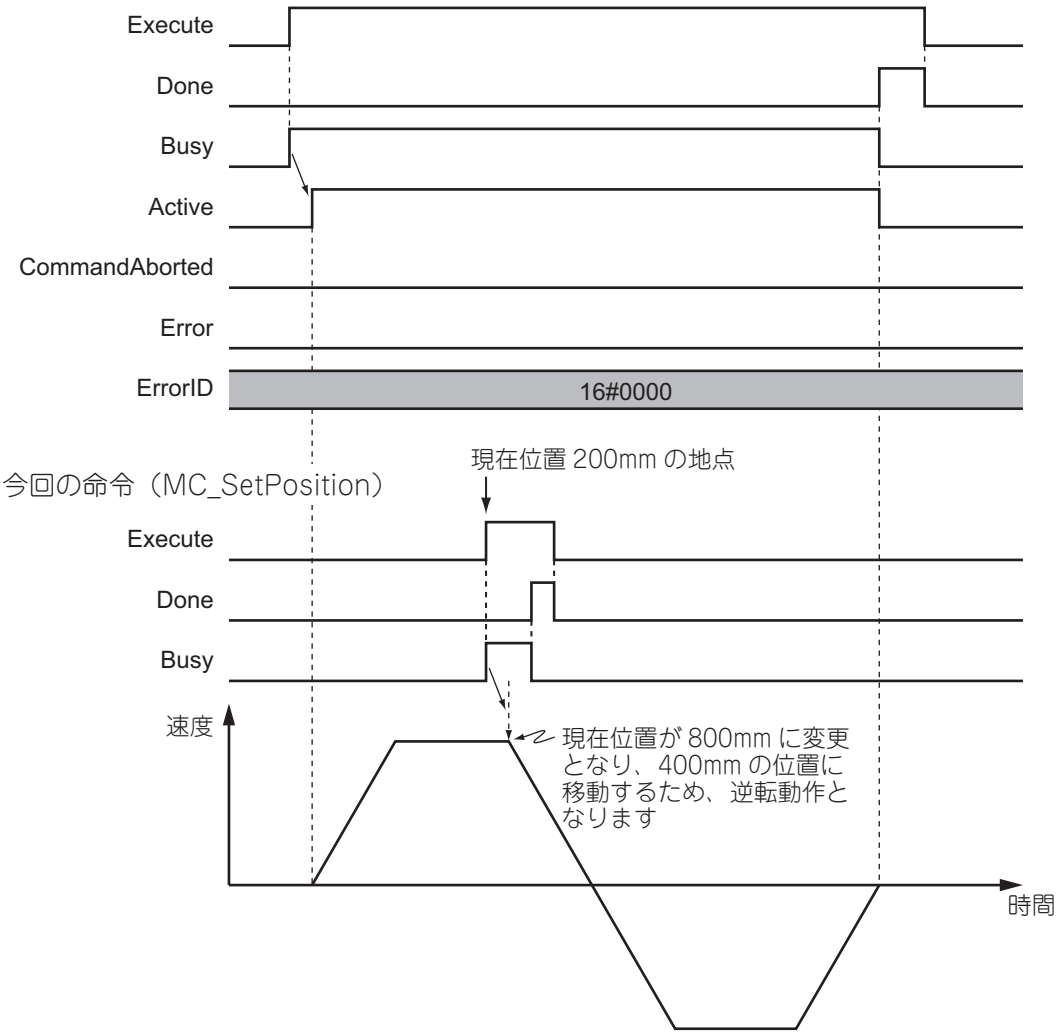
「Execute」（起動）の立ち上がりで、現在位置の変更を開始します。
「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、現在位置の変更が完了すると、「Done」（完了）が TRUE となります。



● 軸が動作中に本命令を使用する場合

絶対位置への位置決め中に本命令を実行した場合、位置の変更に伴い動作の目標値が変更となります。

例として、MC_MoveAbsolute（絶対値位置決め）命令で、400(mm)に移動中に、現在位置 200(mm)で、現在位置を 800(mm)に変更した場合の軸の動作とタイミングチャートを以下に示します。
この場合、現在値が 800(mm)となり、目標値が 400(mm)のため、軸は逆転します。
下図に示すように、MC_MoveAbsolute（絶対値位置決め）命令は、現在位置が変更されても、新しい現在位置から MC_MoveAbsolute（絶対値位置決め）命令で指定された目標位置へ位置決め完了後、「Done」（完了）が TRUE となります。



参考

- MC_MoveRelative（相対値位置決め）命令中や MC_MoveVelocity（速度制御）命令中に本命令を実行した場合は、現在位置は変更されます。しかし、MC_MoveRelative（相対値位置決め）命令や MC_MoveVelocity（速度制御）命令の動作には影響ありません。
- バッファリングされている命令がある場合は、バッファの切り替え時に変更後の位置に対して位置決めします。

モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex)」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

MC_SetPosition（現在位置変更）命令は、次の命令実行中の軸に対しては使用できません。
実行すると多重起動異常となります。

MC_MoveJog（ジョグ送り）命令	MC_MoveLink（台形パターンカム）命令
MC_Home（原点復帰）命令	MC_CombineAxes（加減算位置決め）命令
MC_HomeWithParameter(パラメータ指定原点復帰）命令	MC_MoveZeroPosition（高速原点復帰）命令
MC_CamIn（カム動作開始）命令	MC_TorqueControl（トルク制御）命令
MC_GearIn（ギア動作開始）命令	MC_SyncMoveVelocity(サイクリック同期速度制御）命令
MC_GearInPos（位置指定ギア動作）命令	

使用上の注意

ユニットバージョン Ver.1.09 以前の CPU ユニットの場合
MC_SetPosition（現在位置変更）命令は、MC_GearIn（ギア動作開始）命令等の同期用命令の「Master」（主軸）に対して使用しないでください。
「Master」（主軸）に対して使用した場合、「Master」（主軸）の指令現在位置とフィードバック現在位置が変化したとたん、「Slave」（従軸）は「Master」（主軸）が移動したと判断し、「Master」（主軸）の移動量に応じた動作を行います。このため、「Slave」（従軸）の動きが急激に変化したり、カム動作が終了したりすることがあります。
「Master」(主軸）に対して MC_SetPosition（現在位置変更）命令を使用する場合は、「Master」（主軸）と「Slave」（従軸）の関係をいったん解除してから実行してください。
「Master」（主軸）と「Slave」（従軸）の関係を解除するには、MC_GearOut（ギア動作解除）命令などを実行してください。
主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。

● 本命令実行中の他命令起動

MC_SetPosition（現在位置変更）命令実行中に、ほかの MC_SetPosition（現在位置変更）命令を実行すると、後の MC_SetPosition（現在位置変更）命令が優先して実行されます。
この場合、実行中だった先の MC_SetPosition（現在位置変更）命令の「Done」（完了）は TRUE となりますが、先に設定した位置に変更できていないので、注意してください。

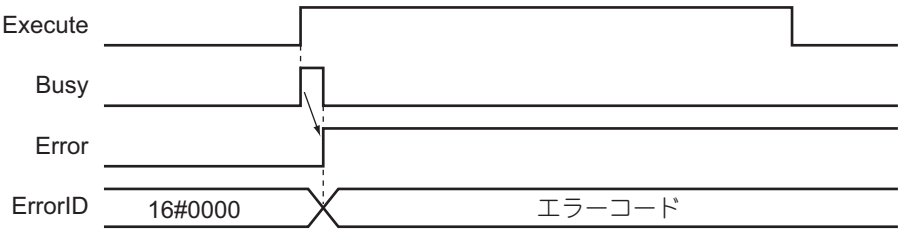
異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

参考

軸動作中に、本命令を実行した場合、現在動作中の軸の命令は、再起動の扱いとなり、位置決めのためのデータを再計算します。
このとき、異常が発生すれば、本命令の異常にはならず、現在動作中の軸の命令が異常となります。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_SetOverride

軸の目標速度を変更します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_SetOverride	オーバーライド値設定	FB	MC_SetOverride_instance MC_SetOverride Axis — — Axis Enable — Enabled VelFactor — Busy AccFactor — Error JerkFactor — ErrorID	MC_SetOverride_instance (Axis :=「パラメータ」, Enable :=「パラメータ」, VelFactor :=「パラメータ」, AccFactor :=「パラメータ」, JerkFactor :=「パラメータ」, Enabled =>「パラメータ」, Busy =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE にするとオーバーライドが有効になります。 FALSE にするとオーバーライドは「100%」に戻ります。
VelFactor	速度オーバーライド値	LREAL	0～500	100	速度のオーバーライド値を指定します。 オーバーライド値の有効範囲は「0.01～500.00」となります。 「500.00 以上」は「500」として扱われ、「0.01 未満（負数も含む）」は「0.01」として扱われます。 「0」を指定したときだけ「0」として動作します。 単位は[%]となります。
AccFactor (Reserved)	加減速度オーバーライド値	LREAL	0～500	100	(Reserved)
JerkFactor (Reserved)	ジャークオーバーライド値	LREAL	0～500	100	(Reserved)

出力変数

出力変数	名称	データ型	有効範囲	内容
Enabled	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。

出力変数	名称	データ型	有効範囲	内容
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Enabled	「Enable」が TRUE となったとき。	• 「Enable」が FALSE の場合、1 周期後 • 「Error」が TRUE となったとき。
Busy	「Enable」の立ち上がり	• 「Enable」が FALSE となったとき。 • 「Error」が TRUE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 軸の目標速度に対するオーバーライドを変更します。オーバーライドを変更することにより動作中の目標速度を変更します。
- オーバーライドが変更できる命令は以下の命令です。

MC_Move（位置決め）命令	MC_MoveJog（ジョグ送り）命令
MC_MoveAbsolute（絶対値位置決め）命令	MC_MoveFeed（割り込み定寸位置決め）命令
MC_MoveRelative（相対値位置決め）命令	MC_MoveZeroPosition（高速原点復帰）命令
MC_MoveVelocity（速度制御）命令	

- 新たな目標速度は、次の通りです。

  変更後の目標速度 = 現在実行中の命令の目標速度 × オーバーライド

- オーバーライドの単位は、[%]です。「100」で「100%」を表します。
- 変更後の目標速度が、軸パラメータの［最高速度］をオーバーする場合は、最高速度になります。
- 変更した目標速度に対して、軸を加速、または減速します。
- オーバーライドに「0」を指定した場合、目標速度「0」となり、軸の動作は減速し速度「0」で動作中となります。
  動作状態を継続しながら軸の動作を一時停止したい場合にオーバーライドを「0」とします。
  このとき、軸変数の「Status.Discrete」、「Status.Continuous」は変化しません。
- オーバーライド値は、「Enable」（有効）が TRUE の状態で、命令実行時に常時反映されます。
- オーバーライド値は、動作指令に対する停止中からの起動、モーション命令再起動、モーション命令多重起動で、設定された新たな目標速度に対して適応されます。
- 「Enable」（有効）が FALSE の場合、オーバーライドは、100(=100%)に戻ります。
- MC_SetOverride（オーバーライド値設定）命令実行中に軸異常が発生した場合、MC_SetOverride（オーバーライド値設定）命令は、「Enabled」（有効）が TRUE のまま継続します。

使用上の注意

本命令の「Enable」（有効）を FALSE にすると、本命令の「Enabled」（有効）および「Busy」（実行中）は FALSE になります。
その際、速度はオーバーライド「100%」に向かって加速、もしくは減速を行います。

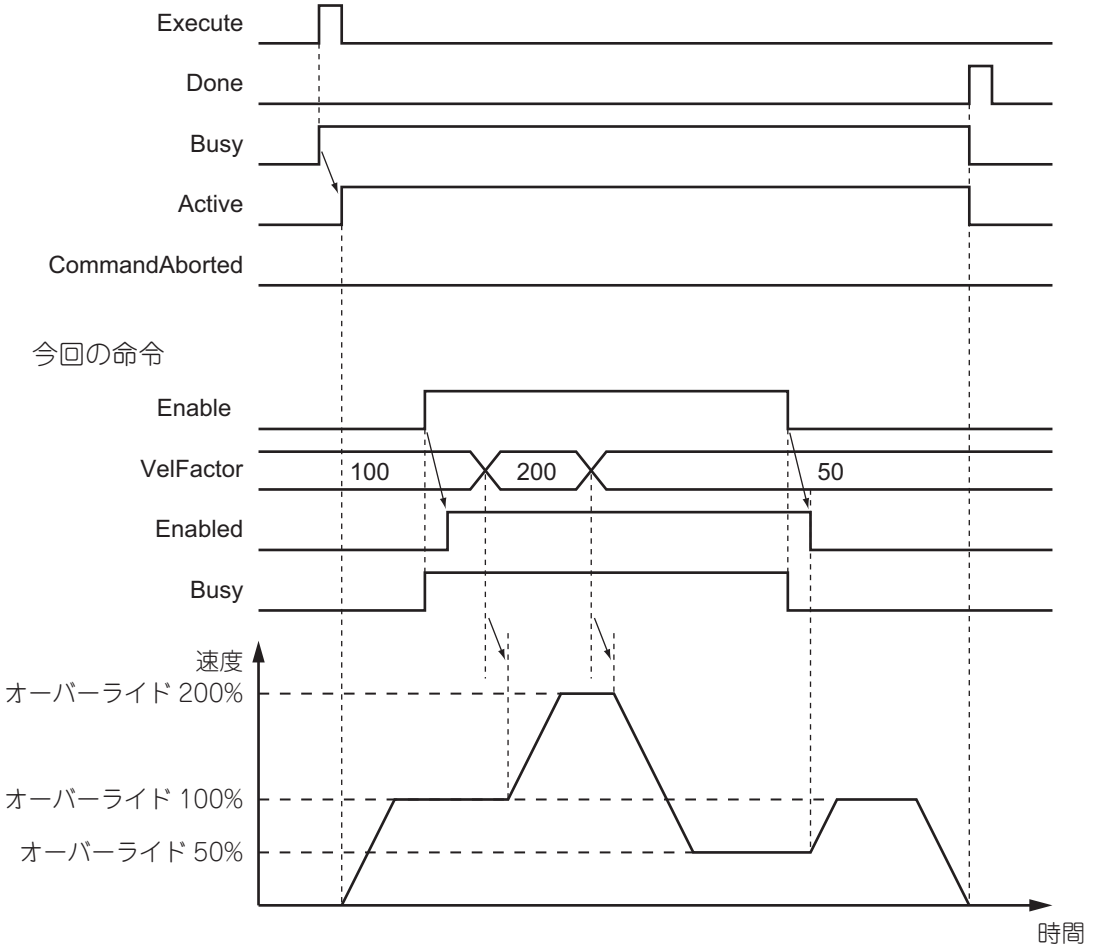
参考

他命令への影響
本命令は、各命令の目標速度を一時的に変更する場合に使用します。
そのため、目標速度を入力に持たない命令や、サイクリック同期速度制御のように目標速度を毎周期変更する命令に対しては、本命令は機能しません。
ただし、本命令が機能しない命令に対して MC_SetOverride（オーバーライド値設定）命令を有効にした場合でも、「Enabled」（有効）は TRUE のまま継続します。

タイミングチャート

● MC_MoveAbsolute（絶対値位置決め）命令へのオーバーライド

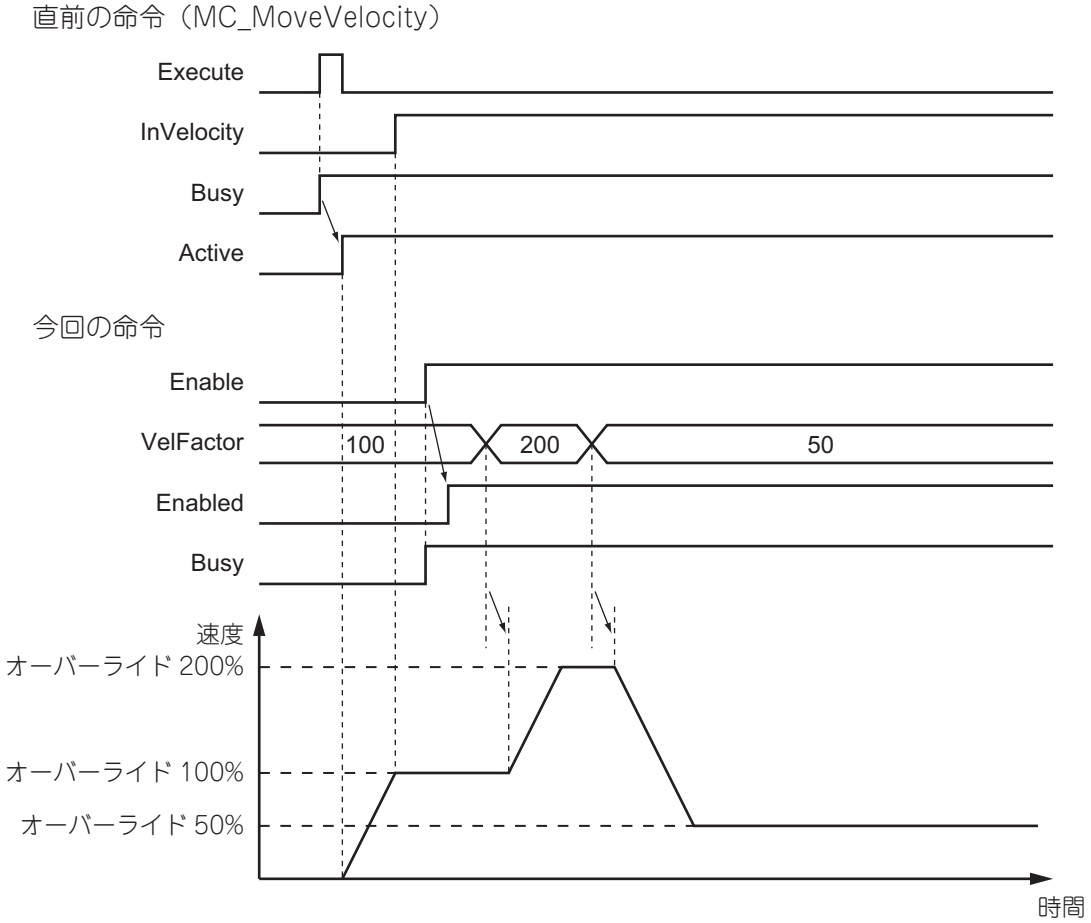
MC_MoveAbsolute（絶対値位置決め）命令に、MC_SetOverride（オーバーライド値設定）命令を使用した場合のタイミングチャートの例を示します。



MC_SetOverride（オーバーライド値設定）命令が無効になると、オーバーライド 100%の目標速度に戻ります。

● MC_MoveVelocity（速度制御）命令へのオーバーライド

MC_MoveVelocity（速度制御）命令に、MC_SetOverride（オーバーライド値設定）命令を使用した場合のタイミングチャートの例を示します。
「InVelocity」（目標速度到達）が TRUE となった後は、速度を変更しても、「InVelocity」（目標速度到達）は TRUE のままです。



モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

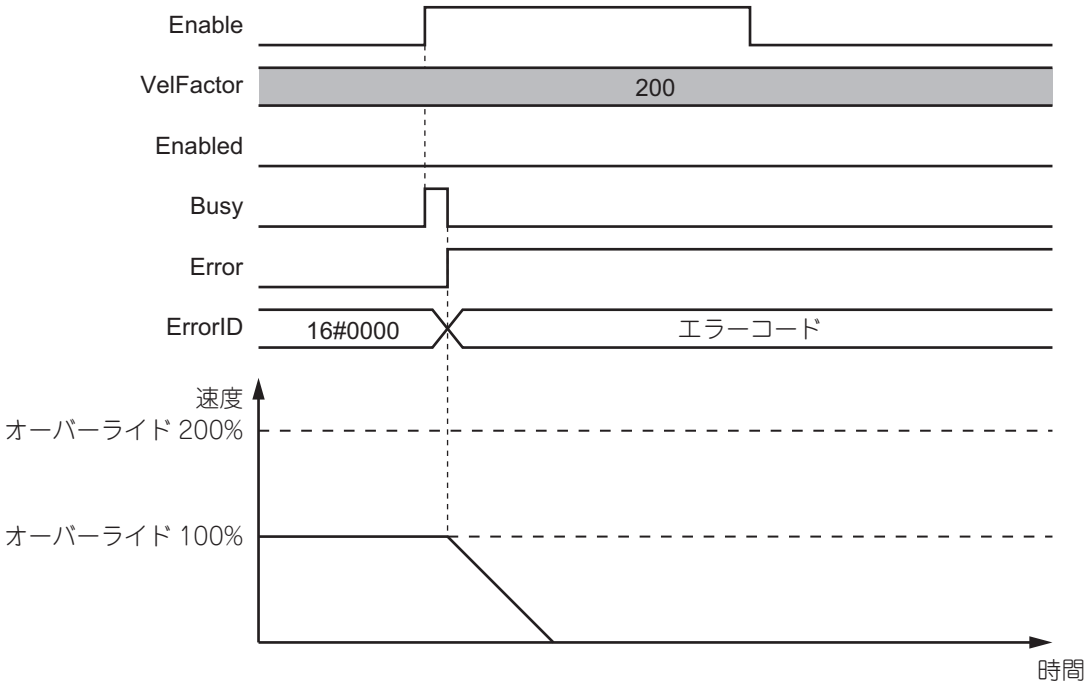
● 本命令同士の多重起動

MC_SetOverride（オーバーライド値設定）命令実行中の軸に、別インスタンスの MC_SetOverride（オーバーライド値設定）命令を起動した場合、後に実行されたインスタンスが優先して処理されます。
「Enabled」（有効）は両方の命令とも TRUE となります。
具体的な動作としては、後から実行されたインスタンスのオーバーライド値が有効となります。後から実行されたインスタンスの「Enable」（有効）を FALSE にすると、オーバーライドは無効となります。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート

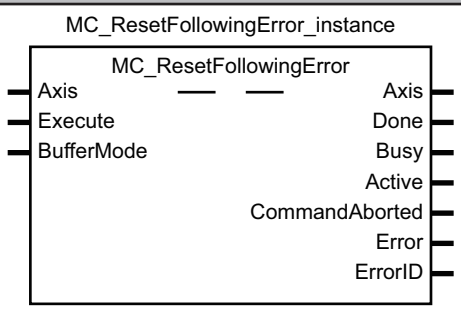


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_ResetFollowingError

指令現在位置とフィードバック現在位置の偏差をリセットします。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_ResetFollowingError	偏差カウンタリセット	FB		MC_ResetFollowingError_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, BufferMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *1	モーション命令多重起動時の動作を指定します。 0: アボーティング

*1. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	偏差カウンタリセットが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの MC_ResetFollowingError 命令によるモーション命令多重起動（[アボーティング]）で本命令が中断されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時。 • 「Execute」が FALSE の場合、1 周期後。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

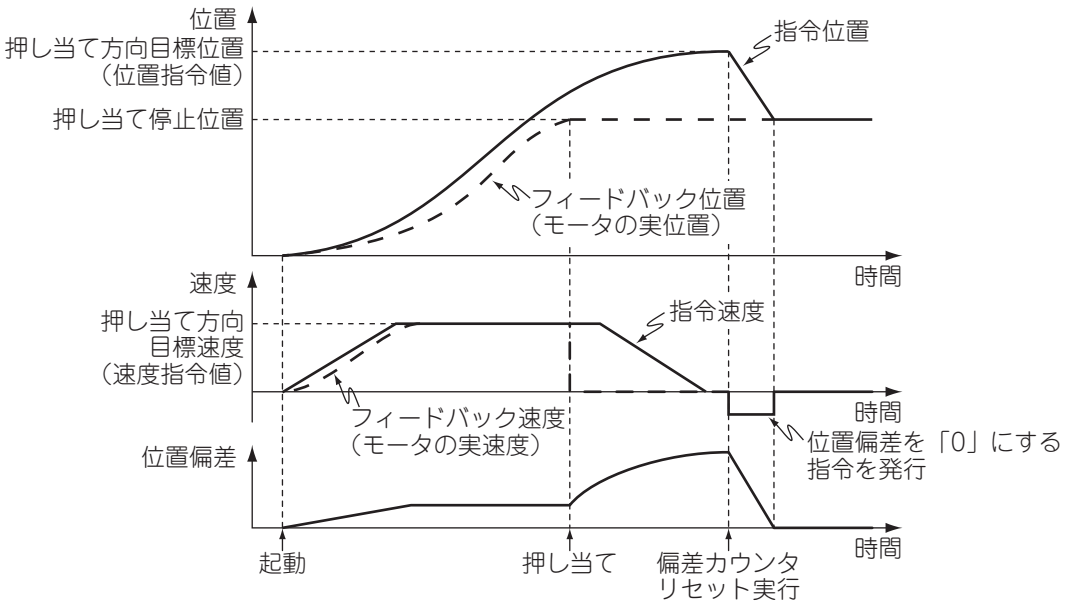
入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 本命令は、サイクリック同期位置モードにおいて、MC 機能モジュールの指令現在位置とフィードバック現在位置との偏差を「0」にする機能です。
- 「Execute」（起動）の立ち上がりを検出すると、その時点でのフィードバック現在位置を新しい目標位置として指令位置を与えます。
  たとえば下図のように位置偏差が発生する押し当て動作において、本命令を起動すると逆方向への位置指令を与えて位置偏差を「0」にします。
  位置偏差が発生した命令は、「CommandAborted」（命令中断）が TRUE となり、命令実行が中止されます。



- 位置偏差を「0」にする場合、軸パラメータに設定されている［最高速度］を用いて位置指令を与えます。［最大加速度］、［最大減速度］は適用されません。
- 新しい目標位置へ指令が完了したとき、出力変数「Done」（完了）が TRUE となります。
- 本命令は位置偏差が発生した動作とは逆方向へ指令位置を与えますが、軸パラメータの［反転時動作］は適用されません。

使用上の注意

- 軸の速度が小さい状態で本命令を起動してください。
  本命令は、前命令（押し当て方向）とは逆方向に指令値を与えます。そのため、軸速度が大きい状態で本命令を起動すると機械に衝撃を与える場合があります。
- 垂直軸など定常的にトルクをかけ続ける必要がある軸で本命令を使用する場合は、本命令起動によりトルク不足が発生しないことを確認してから使用してください。
- NX シリーズ パルス出力ユニットを使用する場合、パルス出力ユニットに接続したサーボドライバ内の偏差はリセットされません。詳細は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。
- **同期制御の主軸に対して、本命令を使用する場合**
  同期制御の主軸で、かつ指令位置を同期データとして使用する場合に、主軸に対して本命令を起動すると、従軸はギア比やカムデータ変数に従って反転しますので、注意してください。主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。

命令の詳細

以下に本命令の詳細について説明します。

● 使用可能軸と起動条件

- サーボ軸や仮想サーボ軸では、以下の場合に使用可能です。
  単軸の位置制御中
  MC_MoveVelocity（速度制御）命令
  同期制御中
- エンコーダ軸、仮想エンコーダ軸では起動時異常となります。

● 軸変数のステータス

軸変数のステータスは、「Status.Stopping」（減速停止中）が TRUE となります。

● 制御モード切り替え中の本命令の起動

MC_TorqueControl（トルク制御）命令または MC_SyncMoveVelocity（サイクリック同期速度制御）命令を実行している場合に、制御モードが位置制御モードとなる命令を実行し、実際に位置制御モードに切り替わるまでの間に本命令を起動すると、CPU ユニットのユニットバージョンにより、偏差が「0」となるタイミングが次のようになります。

ユニットバージョン Ver.1.10 以降の CPU ユニットの場合

- 位置制御モードに切り替わった直後に、指令現在位置とフィードバック現在位置との偏差を「0」にします。

ユニットバージョン Ver.1.09 以前の CPU ユニットの場合

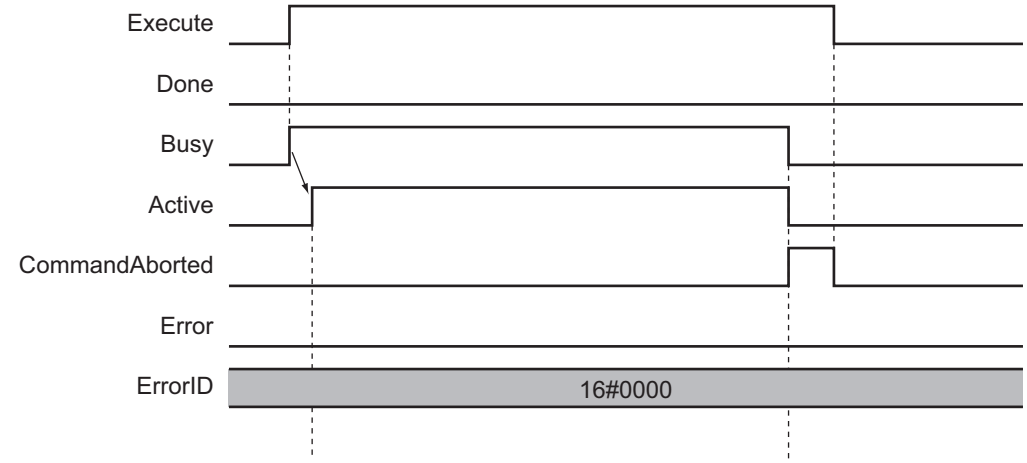
- 本命令の起動時に、指令現在位置とフィードバック現在位置との偏差を「0」にします。

制御モード切り替えの詳細については、「MC_TorqueControl (P.3-315)」および「MC_SyncMoveVelocity (P.3-370)」を参照してください。

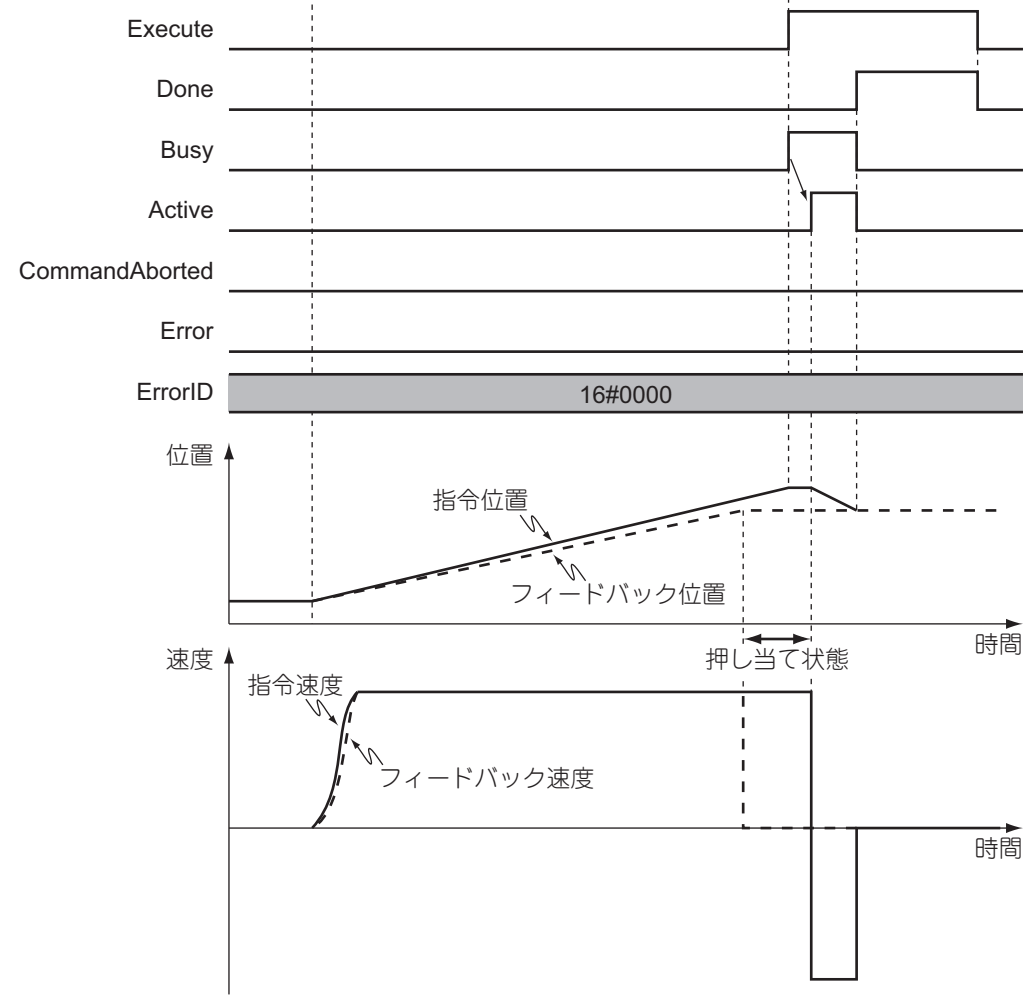
タイミングチャート

MC_MoveAbsolute（絶対値位置決め）命令を起動後、押し当て状態で本命令を起動した場合のタイミングチャートを以下に示します。

MC_MoveAbsolute 命令



MC_ResetFollowingError 命令



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して、本命令を起動することにより、本命令への切り替えや本命令のバッファリングを行うことができます。
各軸に 1 つまでバッファリングすることが可能です。
多重起動時の本命令の動作を、「BufferMode」（バッファモード選択）で指定します。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

MC_Stop（強制停止）命令との関係

MC_Stop（強制停止）命令実行中に MC_ResetFollowingError（偏差カウンタリセット）命令を起動した場合、MC_Stop（強制停止）命令の「Done」が TRUE とになり、MC_ResetFollowingError（偏差カウンタリセット）命令が実行されます。

● 本命令実行中の他命令起動

本命令実行中は、減速停止中となります。
そのため、減速停止中に多重起動可能な命令を起動してください。
起動できない命令では、本命令の「CommandAborted」（実行中断）が TRUE となります。

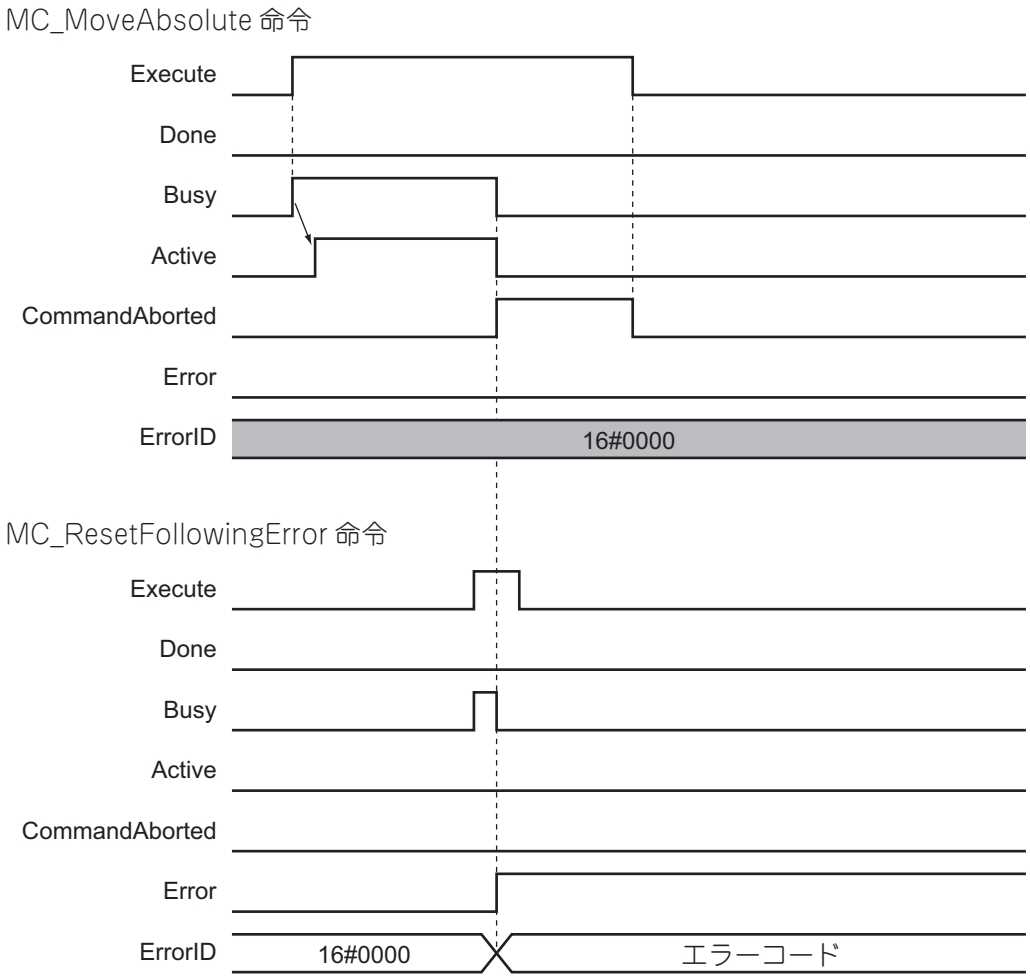
MC_Stop（強制停止）命令との関係

MC_ResetFollowingError（偏差カウンタリセット）命令実行中に MC_Stop（強制停止）命令を起動した場合、MC_ResetFollowingError（偏差カウンタリセット）命令の「CommandAborted」（実行中断）が TRUE になり、MC_Stop（強制停止）命令が実行されます。ただし、MC_Stop（強制停止）命令の「Deceleration」（減速度）は適用されず即停止となります。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

MC_MoveAbsolute（絶対値位置決め）命令を起動後、押し当て状態で本命令を起動した状態で異常が発生した場合のタイミングチャートを以下に示します。

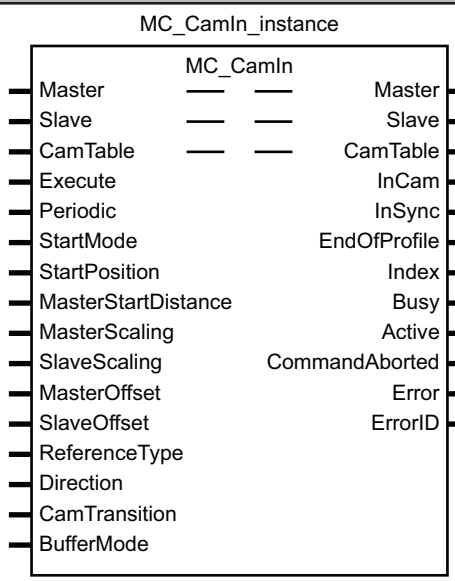


● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_CamIn

指定したカムテーブルを用いたカム動作を開始します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_CamIn	カム動作開始	FB		MC_CamIn_instance (Master :=「パラメータ」, Slave :=「パラメータ」, CamTable :=「パラメータ」, Execute :=「パラメータ」, Periodic :=「パラメータ」, StartMode :=「パラメータ」, StartPosition :=「パラメータ」, MasterStartDistance :=「パラメータ」, MasterScaling :=「パラメータ」, SlaveScaling :=「パラメータ」, MasterOffset :=「パラメータ」, SlaveOffset :=「パラメータ」, ReferenceType :=「パラメータ」, Direction :=「パラメータ」, CamTransition :=「パラメータ」, BufferMode :=「パラメータ」, InCam =>「パラメータ」, InSync =>「パラメータ」, EndOfProfile =>「パラメータ」, Index =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Periodic	繰り返しモード	BOOL	TRUE, FALSE	FALSE	指定されたカムテーブルを繰り返し実行するか、1 回しか実行しないかを指定します。 TRUE: 繰り返し FALSE: 繰り返しなし

入力変数	名称	データ型	有効範囲	初期値	内容
StartMode	開始位置方式指定	_eMC_START_MODE	0: _mcAbsolutePosition 1: _mcRelativePosition	0 *1	「MasterStartDistance」（主軸追従距離）が使用する座標系を指定します。 0: 絶対位置 1: 相対位置
StartPosition	カムテーブル開始位置	LREAL	負数、正数、「0」	0	カムテーブルの始点（位相=0）を、主軸の絶対位置で指定します。 単位は（指令単位）となります。*2
MasterStartDistance	主軸追従距離	LREAL	負数、正数、「0」	0	従軸がカム動作を開始する際の主軸の位置を指定します。 「StartMode」（開始位置方式指定）で［0: _mcAbsolutePosition］を指定した場合は主軸の絶対位置、［1: _mcRelativePosition］を指定した場合は「StartPosition」（カムテーブル開始位置）からの相対量で指定します。 単位は（指令単位）となります。*2
MasterScaling	主軸係数	LREAL	正値(>0.0)	1.0	指定されたスケールで、主軸の位相を拡大／縮小します。
SlaveScaling	従軸係数	LREAL	正値(>0.0)	1.0	指定されたスケールで、従軸の変位を拡大／縮小します。
MasterOffset	主軸オフセット	LREAL	負数、正数、「0」	0	指定されたオフセット値で、主軸の位相をずらします。
SlaveOffset	従軸オフセット	LREAL	負数、正数、「0」	0	指定されたオフセット値で、従軸の変位をずらします。
ReferenceType	位置種別選択	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *1	主軸の位置種別を指定します。 0: 指令位置（直前タスク周期 *3 での計算値） 1: フィードバック位置（同一タスク周期 *3 での取得値） 2: 指令位置（同一タスク周期*3 での計算値）
Direction	方向選択	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 4: _mcNoDirection	4 *1	指定された方向に主軸が動作している場合だけ、従軸がカム動作を行います。指定された方向と逆方向に主軸が動作している場合、従軸はカム動作を停止します。 0: 正方向指定 2: 負方向指定 4: 方向指定なし
CamTransition（Reserved）	カムトランジション選択	_eMC_CAM_TRANSITION	0: _mcCTNone	0 *1	（Reserved）

入力変数	名称	データ型	有効範囲	初期値	内容
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow *4 3: _mcBlendingPrevious *4 4: _mcBlendingNext *4 5: _mcBlendingHigh *4	0 *1	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2: ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ

*1. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*2. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*3. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。
*4. ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで使用可能です。4 種類のブレンディングのいずれを指定しても同じ動作となります。

出力変数

出力変数	名称	データ型	有効範囲	単位	内容
InCam	カム動作中	BOOL	TRUE, FALSE	−	カムテーブルの始点が実行されたときに TRUE となります。
InSync	同期中	BOOL	TRUE, FALSE	−	主軸に同期したときに TRUE となります。
EndOfProfile	カムサイクル完了	BOOL	TRUE, FALSE	−	カムテーブルの終点が実行されたときに TRUE となります。
Index	インデックス	UINT	0 または正値	−	カムデータのインデックス番号を出力します。*1
Busy	実行中	BOOL	TRUE, FALSE	−	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	−	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	−	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	−	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*2	−	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. 「InCam」（カム動作中）が FALSE の間は、出力は「0」となります。
*2. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
InSync	従軸がカム動作を開始したとき。	• 「Periodic」が FALSE の場合、「EndOfProfile」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
EndOfProfile	カムテーブルの終点の位相と変位が指令位置として出力された周期	「EndOfProfile」が TRUE となった次の 1 周期後
Busy	「Execute」の立ち上がり	• 「Periodic」が FALSE の場合、「EndOfProfile」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
InCam	主軸が「StartPosition」（カムテーブル開始位置）を通過したとき。	• 「Periodic」が FALSE の場合、「EndOfProfile」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Periodic」が FALSE の場合、「EndOfProfile」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。 • MC_CamOut 命令が起動されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Master	主軸	_sAXIS_REF	−	主軸を指定します。*1
Slave	従軸	_sAXIS_REF	−	従軸を指定します。*1
CamTable	カムテーブル	ARRAY[0..N] OF _sMC_CAM_REF	−	カムデータ構造体_sMC_CAM_REF 型の配列変数をカムテーブルとして指定します。*2

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. 配列要素“N”は Sysmac Studio により自動的に設定されます。Sysmac Studio のカムエディタで作成したカムデータ変数を指定してください。

使用上の注意

主軸と従軸に同じ軸を指定すると、軽度フォールトレベルの「主軸従軸同一（エラーコード: 5436 Hex)」異常となります。

機能説明

- 本命令は、位相（主軸）と変位（従軸）がカムテーブルに従って同期するカム動作を実行します。
- 本命令で指定するカムテーブルは、あらかじめカムエディタを使って作成し、コントローラ本体にダウンロードしておく必要があります。

・「Execute」（起動）の立ち上がりで、本命令が起動されます。

使用上の注意

カムテーブルは、Sysmac Studio のカムエディタで作成したカムデータ変数を使う必要があります。

参考

「ダウンロード」は、Sysmac Studio の「同期」機能を使用します。

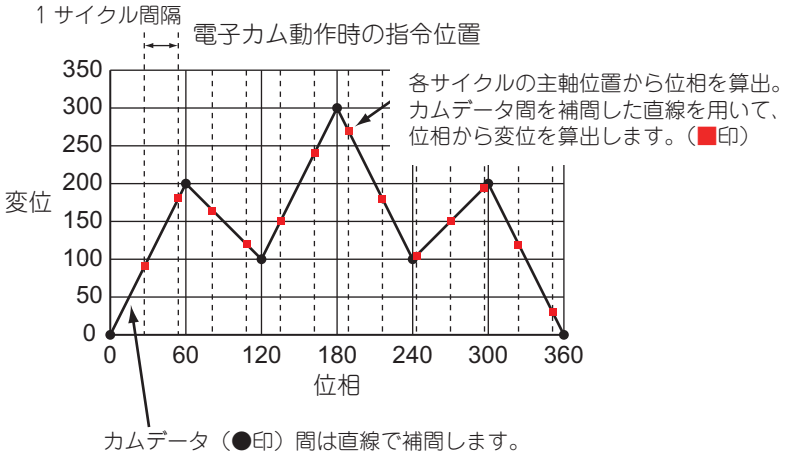
カムテーブルについては、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編 (SBCE-433)』を参照してください。

カムテーブルは位相、変位とも「0.0」の始点からの相対量で与えます。

カムテーブル

	位相	変位	
始点 →	0	0	← Index = 0
カムデータ →	60	200	1
	120	100	2
	180	300	3
	240	100	4
	300	200	5
終点 →	360	0	6

また、主軸、従軸への指令位置は、2 つのカムデータ間を直線で補間し、位相（主軸）に相当する変位（従軸）を求めます。
カムデータ数が少ないと位相の間隔が大きくなり、精度の荒いカム動作となります。逆にカムデータ数が多いと位相の間隔が小さくなり精度の高いカム動作となります。



使用上の注意

- ユニットバージョン Ver.1.09 以前の CPU ユニットで本命令を使用する場合は、「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行しないでください。「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行すると、「Slave」（従軸）が急激な追従をすることがあり危険です。
  「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を使用する場合は、「Master」（主軸）と「Slave」（従軸）の関係をいったん解除してから実行してください。
  主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。
- 本命令では、起動時に位相が昇順に並んでいるかをチェックします。位相が昇順に並んでいない場合、異常を検知します。
  本命令起動後、カムデータを変更した場合に、位相が昇順に並んでいることを確認するときは、MC_SetCamTableProperty（カムテーブルプロパティ更新）命令または MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令を実行してください。
  カム動作中に位相を変更する場合は、変更する前に昇順になっていることを確かめてから変更をお願いします。昇順になっていないとカム動作が途中で停止する場合があります。
- カムデータ変数はグローバル変数であり、複数のタスクからカムデータ変数の値を参照や変更をすることができます。複数のタスクからカムデータ変数の値を変更する場合、複数のタスクによる書き換え処理で競合が発生しないようにプログラムを作成してください。
- 「グローバル変数のタスク間排他制御」機能を用いてカムデータ変数を排他制御する場合、排他対象のカムデータ変数を用いてモーション命令を起動しないでください。軽度フォールトレベルの異常「カムテーブル指定不正（エラーコード: 5439 Hex）」が発生します。
- カム動作中に命令再起動によりカムテーブルを切り替える場合、切り替えの前後で従軸の速度や加速度が急激に変化することがあります。
  メカ機構に影響を与える場合があるので、速度や加速度が過度に不連続にならないか十分に検証したうえでカムテーブルを切り替えてください。
- カム動作中のカムテーブルに含まれるカムデータ変数の値をユーザプログラムから変更する場合、変更の前後で従軸の速度や加速度が急激に変化することがあります。
  メカ機構に影響を与える場合があるので、速度や加速度が過度に不連続にならないか十分に検証したうえでカムデータ変数の値を変更してください。

命令の詳細

以下に本命令の詳細について説明します。

● 命令起動条件

本命令は、主軸が停止中、または位置制御中、速度制御中、同期制御中のいずれの状態でも起動可能です。
従軸については「モーション命令再起動 (P.3-185)」、「モーション命令多重起動 (P.3-6)」を参照してください。

● ソフトウェアリミット

カム動作中に従軸がソフトウェアリミットを超えた場合、異常となります。

● カムデータ変数

カムデータ変数は、カムデータ構造体を配列宣言して使用します。カムデータ構造体は、あらかじめ次のように型宣言されています。

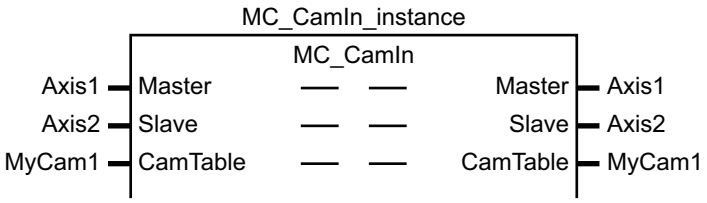
```
TYPE
    (* カムデータ構造体 *)
    _sMC_CAM_REF :
    STRUCT
    Phase : REAL; (* 位相 *)
    Distance : REAL; (* 変位 *)
    END_STRUCT;
END_TYPE
```

Sysmac Studio 上でカムデータ変数を作成します。
カムデータ変数の名称であるカムテーブル名は任意に決められます。

例えば、カムテーブル名「MyCam1」、カムデータ数を 1000 個にする場合、Sysmac Studio で、次のように自動的に変数宣言されます。

```
VAR
    (* カムテーブル *)
    MyCam1 : ARRAY [0..999] OF _sMC_CAM_REF;
END_VAR
```

本命令で「MyCam1」を指定する場合、次のように記述します。ここでは、主軸を「Axis1」、従軸を「Axis2」とします。



指定したカムテーブルがコントローラに存在しない場合、異常を検知します。また、複数軸で同一のカムテーブルを指定することが可能です。

カムデータ変数の値は、ユーザプログラムから書き換えることができます。ただし、書き換えた値はコントローラの電源 OFF、Sysmac Studio からのダウンロード実行時に消失します。
CPU ユニット電源投入後、およびダウンロード完了後は Sysmac Studio からダウンロードされた値に戻ります。書き換えた値を保存したい場合には、MC_SaveCamTable（カムテーブル保存）命令を使用してください。
CPU ユニットの動作モード変更では、書き換えたカムデータ変数の値は保持されます。

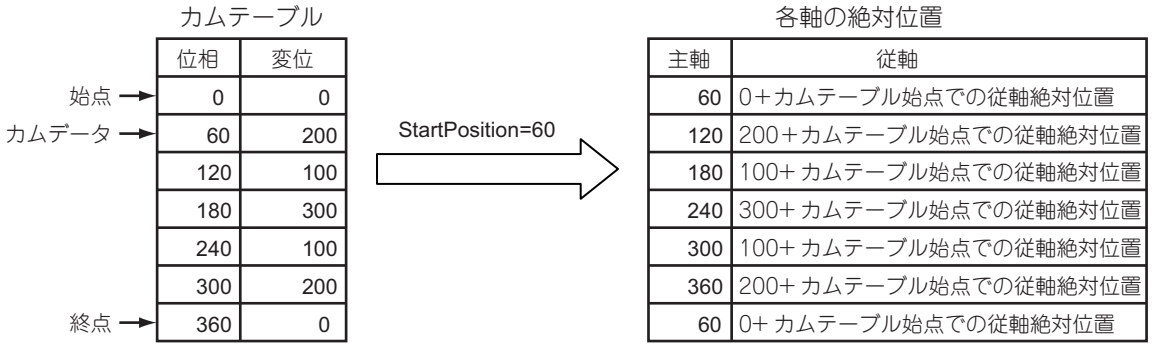
参考

- カムデータ変数はネットワーク非公開となります。
  例えば、「MyCam1[10].Phase」や「MyCam1[10].Distance」は、Sysmac Studio の Watch View で見ることはできますが、EtherNet/IP 経由で他のコントローラからアクセスすることはできません。
- 「ダウンロード」は、Sysmac Studio の「同期」機能を使用します。

● カム動作の開始

命令起動後、主軸が「StartPosition」（カムテーブル開始位置）に到達するのを待ちます。
主軸が「StartPosition」（カムテーブル開始位置）を通過すると、カムテーブルの始点が実行され、出力変数「InCam」（カム動作中）が TRUE となります。
カムテーブルの位相と変位は、始点をゼロとする相対量で与えているため、各位相での各軸の絶対位置は、カムテーブル始点での各軸の絶対位置を起点とする相対値となります。たとえば、主軸のカウントモードが 0～360° の［ロータリモード］で、「StartPosition」（カムテーブル開始位置）=60 と

します。主軸の絶対位置は、以下のとおりカムテーブルの位相に「StartPosition」を加えた値となり、従軸の絶対位置は、カムテーブルの変位にカムテーブル始点での従軸の絶対位置を加えた値となります。



さらに、「MasterStartDistance」（主軸追従距離）を通過すると、従軸のカム動作が開始し、出力変数「InSync」（同期中）が TRUE となります。
「MasterStartDistance」（主軸追従距離）は、絶対位置もしくは「StartPosition」（カムテーブル開始位置） からの相対距離で指定します。絶対位置、または相対位置は、「StartMode」（開始位置指定方式） で設定します。

例 1）「MasterStartDistance」 の相違による従軸動作の違い
例では、同じカムテーブル、同じ主軸を使用します。

カムテーブルの設定は以下です。

主軸	従軸	カム曲線	接続速度	接続加速度	位相間隔幅
0.000	0.000	−	−	−	−
80.000	80.000	直線	360.000	0.000	0.010
120.000	200.000	直線	1080.000	0.000	0.010
360.000	360.000	直線	240.000	0.000	0.010

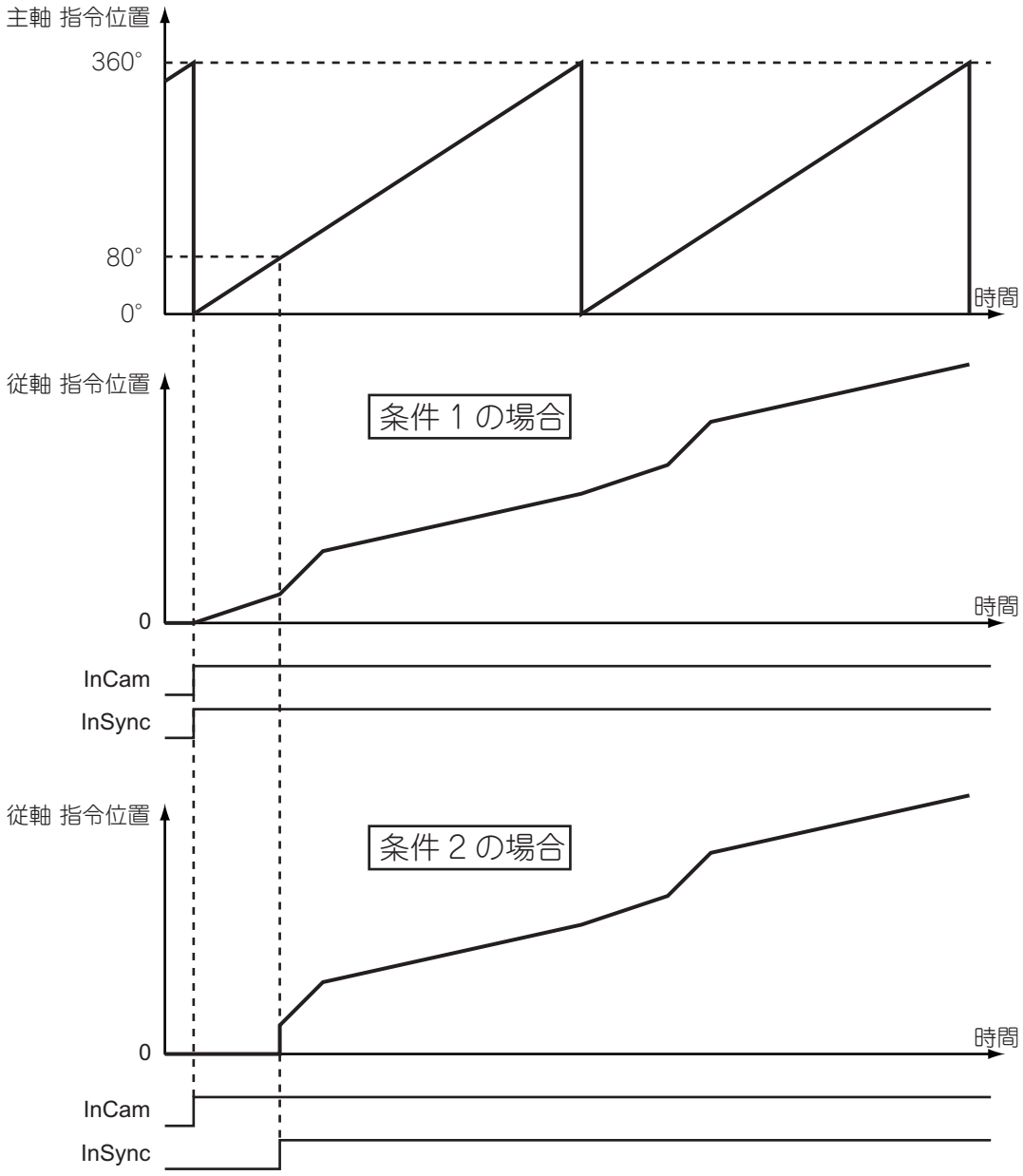
カム動作の開始条件は以下です。

入力変数	条件 1	条件 2
Periodic（繰り返しモード）	TRUE（繰り返し）	TRUE（繰り返し）
StartMode（開始位置方式指定）	_mcRelativePosition（相対位置）	_mcRelativePosition（相対位置）
StartPosition（カムテーブル開始位置）	0	0
MasterStartDistance（主軸追従距離）	0	80

条件 1 では、主軸が 0° を通過すると出力変数「InCam」（カム動作中）と出力変数「InSync」（同期中）の両方が TRUE となり、従軸がカム動作を開始します。

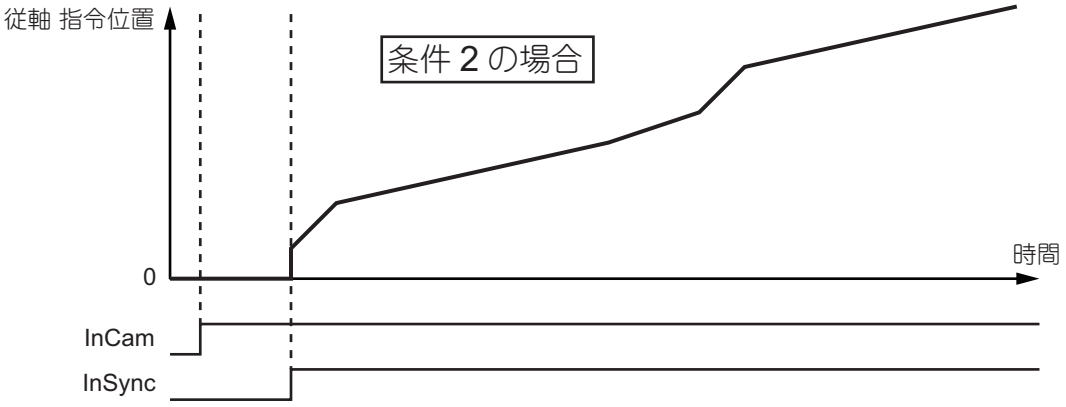
条件 2 では、主軸が 0° を通過すると出力変数「InCam」（カム動作中） が TRUE となり、さらに 80° *1 を通過すると出力変数「InSync」（同期中）が TRUE となり、従軸がカム動作を開始します。
条件 2 の場合、カムテーブルの途中からカム動作を開始するため、従軸は急激な加速となります。

*1.「StartMode」 が_mcRelativePosition のため、「StartPosition」 ＋ 「MasterStartDistance」=80° がカム動作開始位置となります。

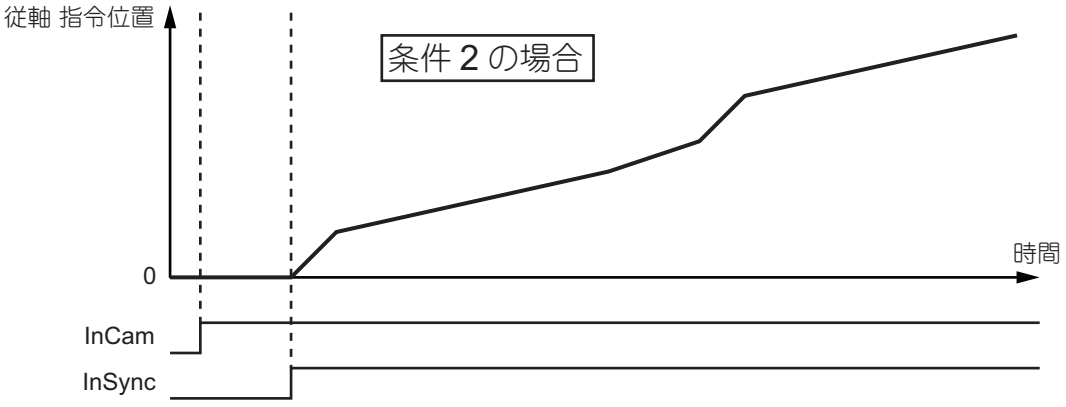

主軸 指令位置
360°
80°
0°
時間
従軸 指令位置
条件 1 の場合
0
時間
InCam
InSync
従軸 指令位置
条件 2 の場合
0
時間
InCam
InSync

参考

条件２の場合、「SlaveOffset」 = 0 とすると、従軸は下図のように急激な加速となります。



「SlaveOffset」 = -80 とすることにより、従軸は、「MasterStartDistance」（主軸追従距離）の変位から同期を開始し、急激な加速となりません。



例 2)「StartPosition」 と「MasterStartDistance」 の相違による従軸動作の違い

カムテーブルの設定は前述の例と同じです。カム動作の開始条件は以下です。

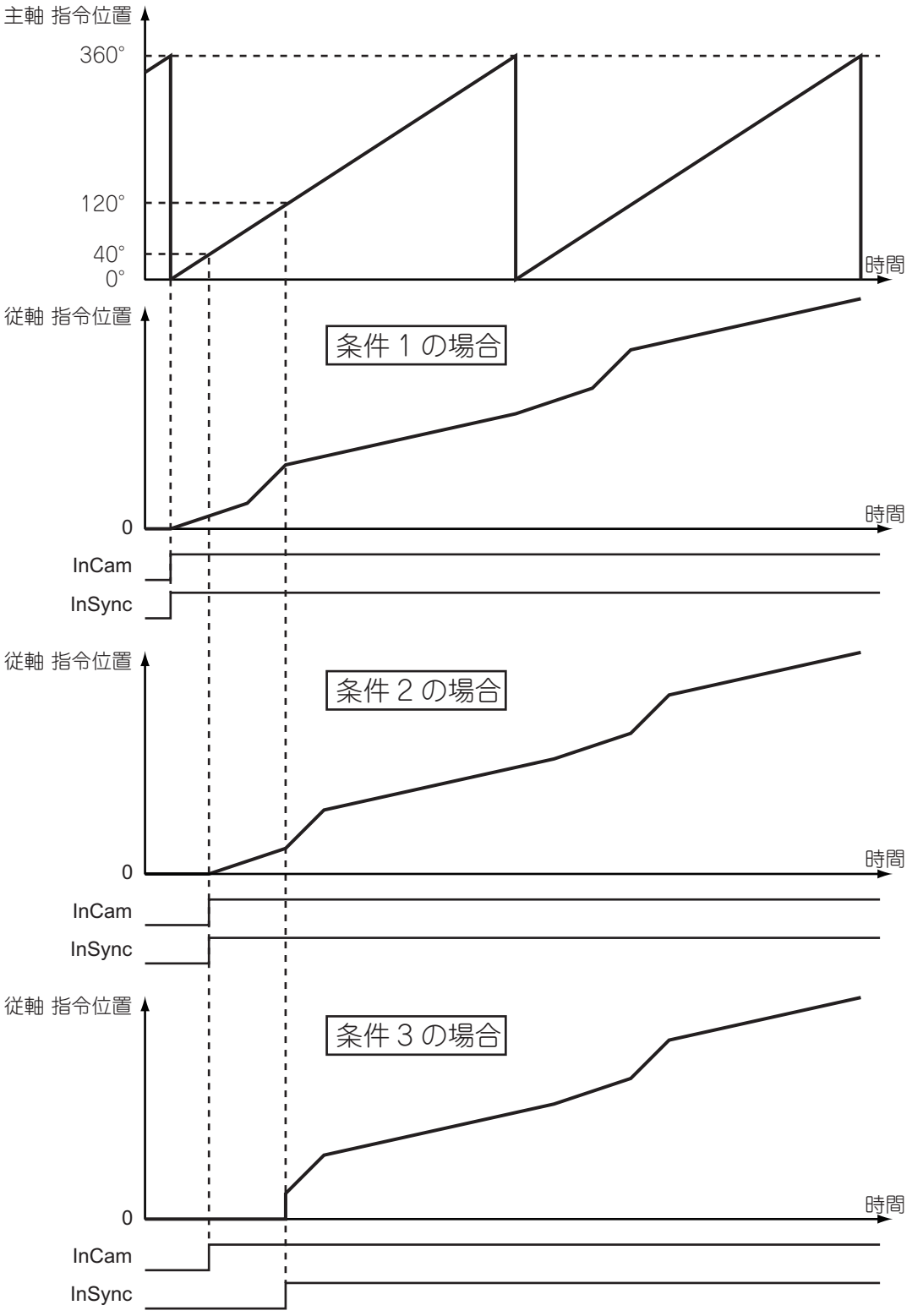
入力変数	条件 1	条件 2	条件 3
Periodic（繰り返しモード）	TRUE（繰り返し）	TRUE（繰り返し）	TRUE（繰り返し）
StartMode（開始位置方式指定）	_mcRelativePosition（相対位置）	_mcRelativePosition（相対位置）	_mcRelativePosition（相対位置）
StartPosition（カムテーブル開始位置）	0	40	40
MasterStartDistance（主軸追従距離）	0	0	80

条件 1 では、主軸が 0° を通過すると出力変数「InCam」（カム動作中）と出力変数「InSync」（同期中）の両方が TRUE となり、従軸がカム動作を開始します。

条件 2 では、「StartPosition」（カムテーブル開始位置）で指定した 40° を主軸が通過すると出力変数「InCam」（カム動作中） と出力変数「InSync」（同期中）の両方が TRUE となり、従軸がカム動作を開始します。

条件 3 では、主軸が 40° を通過すると出力変数「InCam」（カム動作中）が TRUE となり、さらに 120° *1 を通過すると出力変数「InSync」（同期中）が TRUE となり、従軸がカム動作を開始します。

*1.「StartMode 」が_mcRelativePosition のため、「StartPosition」 + 「MasterStartDistance」=120° がカム動作開始位置となります。



例 3)「StartMode」(開始位置方式指定) の相違による従軸のカム動作開始の違い

「StartMode」(開始位置方式指定) により、「MasterStartDistance」(主軸追従距離) に指定した値を絶対位置として扱うか、相対位置として扱うかを指定できます。

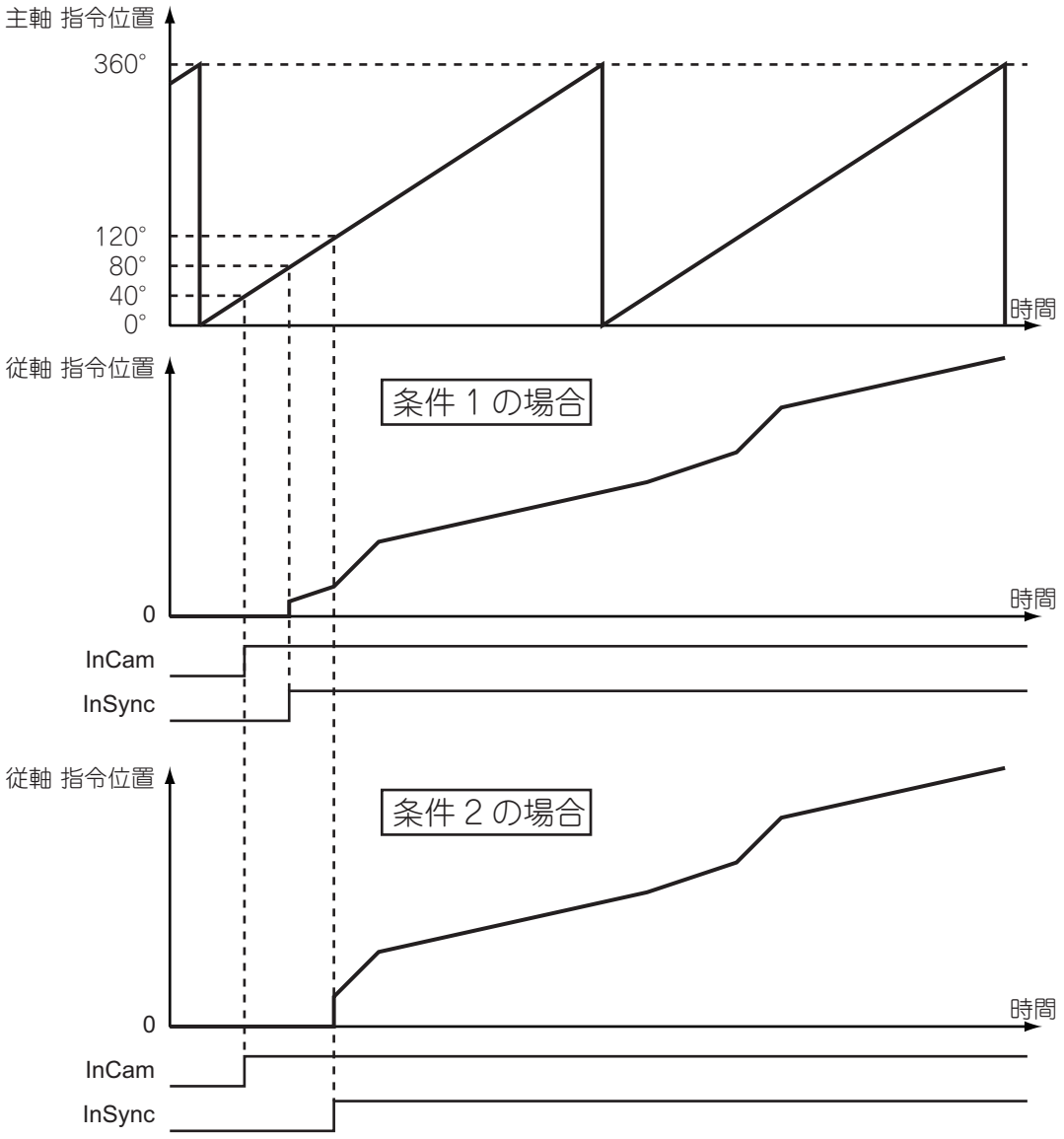
「StartMode」(開始位置方式指定) の相違による従軸のカム動作開始の違いを説明します。カムテーブルの設定は前述の例と同じです。

カム動作の開始条件は以下です。

入力変数	条件 1	条件 2
Periodic（繰り返しモード）	TRUE（繰り返し）	TRUE（繰り返し）
StartMode（開始位置方式指定）	_mcAbsolutePosition（絶対位置）	_mcRelativePosition（相対位置）
StartPosition（カムテーブル開始位置）	40	40
MasterStartDistance（主軸追従距離）	80	80

条件 1 と条件 2 の両方とも、主軸が 40° を通過すると出力変数「InCam」（カム動作中）が TRUE となります。条件 1 では、「StartMode」（開始位置方式指定）が_mcAbsolutePosition（絶対位置）のため、主軸が 80° を通過すると出力変数「InSync」（同期中）が TRUE となり、従軸がカム動作を開始します。

条件 2 では、「StartMode」（開始位置方式指定）が_mcRelativePosition（相対位置）のため、主軸が 120°（=40° + 80°）を通過すると出力変数「InSync」（同期中）が TRUE となり、従軸がカム動作を開始します。



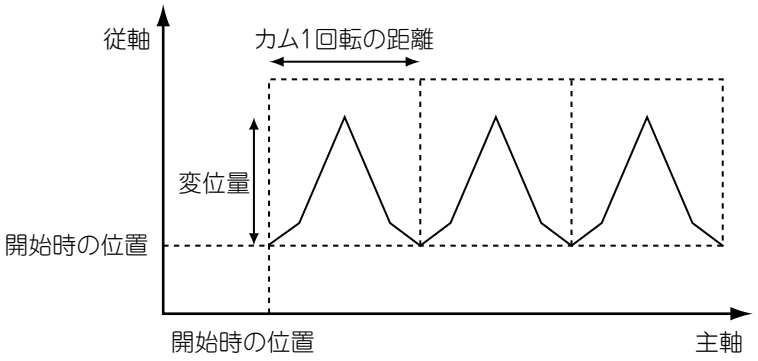
使用上の注意

カム動作を行うためには、Sysmac Studio のカムエディタを使って、カムプロファイルを作成の上、CPU ユニットにダウンロードしておく必要があります。
「ダウンロード」は、Sysmac Studio の「同期」機能を使用します。

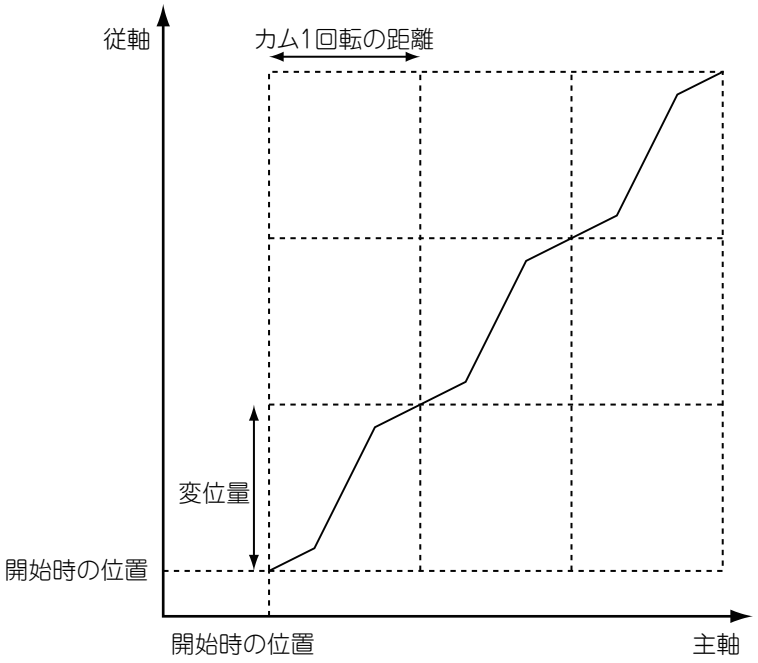
● Periodic（繰り返しモード）

「Periodic」（繰り返しモード）で TRUE［繰り返し］を指定した場合、カムテーブルの始点から終点までカム動作を繰り返し実行します。
FALSE［繰り返しなし］を指定した場合、カムテーブルの終点まで実行するとカム動作を終了します。
TRUE［繰り返し］のとき、カムテーブルの始点と終点で従軸のストローク位置が同一なら、往復カムとして動作します（下図、「往復カム動作」参照）。また、始点と終点で従軸のストローク位置が異なる場合、送りカムとして動作します（下図、「送りカム動作」参照）。
なお、下図では横軸が主軸の位置、縦軸が従軸の位置を表します。

往復カム動作



送りカム動作



● EndOfProfile（カムサイクル完了）

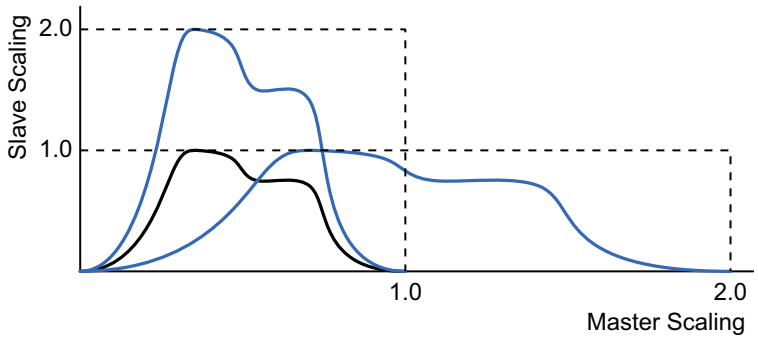
カムテーブルの終点で定義された位相、変位を用いたカム動作の指令値を出力した周期で、「EndOfProfile」（カムサイクル完了）を 1 周期の間、TRUE とします。
「StartPosition」（カムテーブル開始位置）には主軸の絶対位置を設定し、カムテーブルはその位置からの相対テーブルとなります。
「EndOfProfile」（カムサイクル完了）はカムテーブルの終端をあらわす出力となります。

● カム動作の終了

カム動作を途中で終了させる場合は、MC_CamOut（カム動作解除）命令、もしくは MC_Stop（強制停止）命令を使用します。

● 係数（スケーリング）

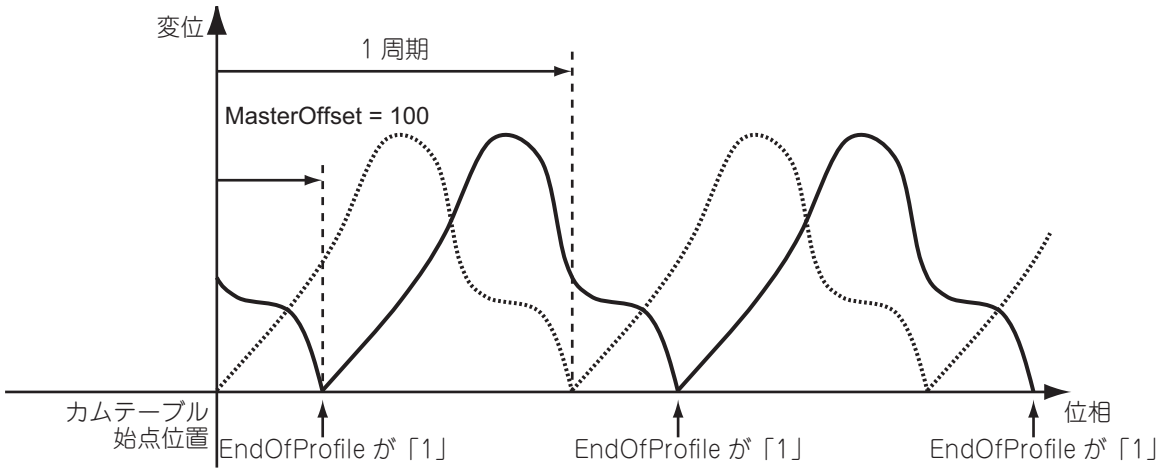
指定されたカムテーブルの主軸位相、従軸変位それぞれに対して、指定されたスケールで拡大縮小させることができます。
主軸、従軸とも別々の倍率を設定可能です。



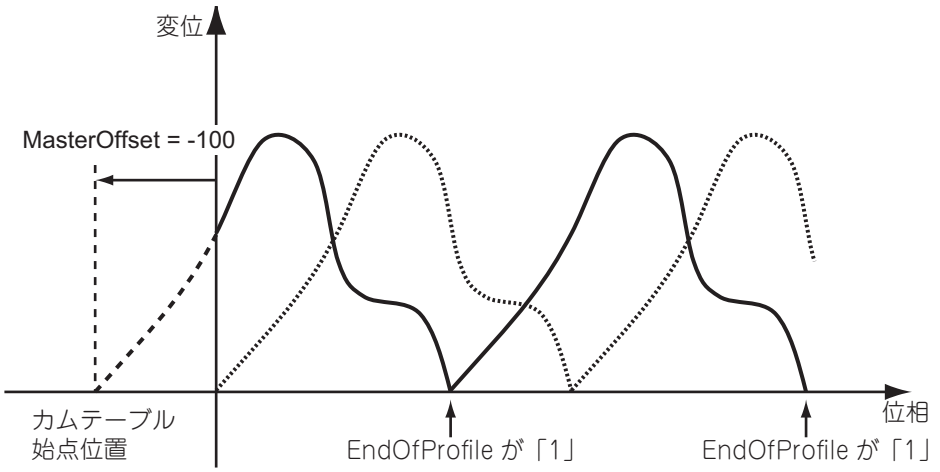
● オフセット

指定されたカムテーブルに対してオフセット量だけ位相と変位をシフトさせる機能です。
主軸位相、従軸変位に対して別々のオフセット量を指定できます。

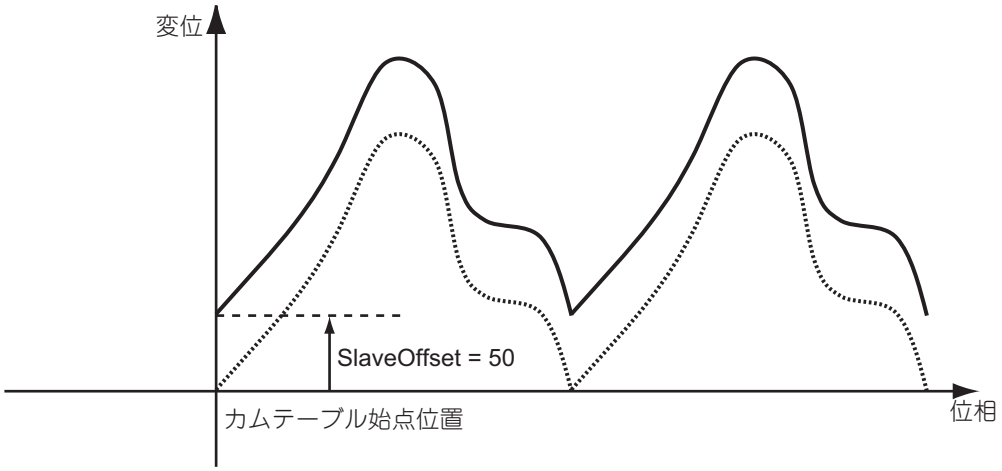
MasterOffset（主軸オフセット）＞0 の場合



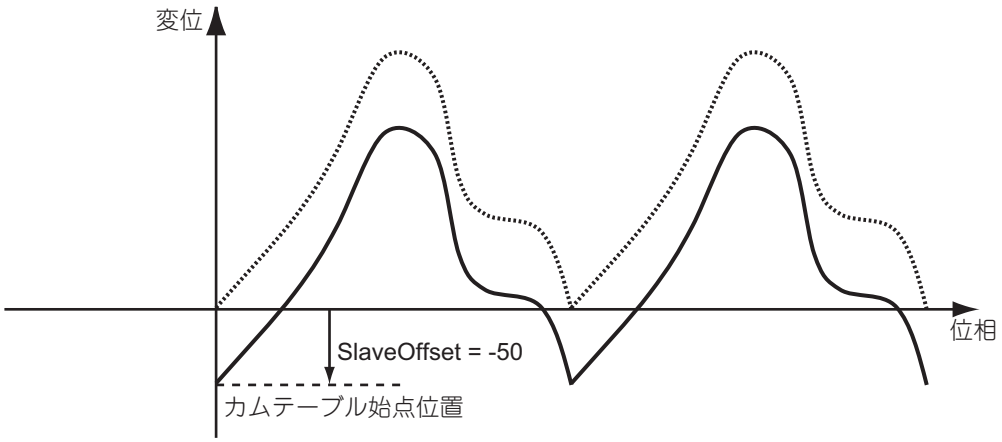
MasterOffset（主軸オフセット）＜0 の場合



SlaveOffset（従軸オフセット）＞0 の場合



SlaveOffset（従軸オフセット）＜0 の場合



● ReferenceType（位置種別選択）

従軸が同期する主軸の位置の種類を、以下から選択できます。

- _mcCommand: 指令位置（直前タスク周期での計算値）
  現周期に対して、直前のタスク周期で計算した主軸の指令位置を使用します。
  従軸の指令位置を計算する直前の定周期タスクで、計算した主軸の指令位置を使用します。
- _mcFeedback: 同一タスク周期での取得値

同一タスク周期で取得した主軸のフィードバック位置を使用します。

- _mcLatestCommand: 指令位置（同一タスク周期での計算値）
  同一タスク周期で計算した主軸の指令位置を使用します。
  _mcCommand よりも新しい情報を使用することができます。ただし、主軸番号は従軸番号より若い番号を設定しなければなりません。
  従軸が、主軸より若い軸番号だった場合、本命令の「Error」（エラー）が TRUE となります。「ErrorID」（エラーコード）には、「主軸／従軸 軸番号非昇順（エラーコード: 5438 Hex）」が出力されます。

使用上の注意

ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。同様に定周期タスクは、プライマリ定周期タスクまたは定周期タスク（実行優先度 5）を表します。

参考

同一タスク周期で計算した指令位置を選択した方が、直前タスク周期での計算値を選択した場合より、同期精度は向上します。
ただし、[「Master」（主軸）に設定されているモーション制御システム変数の軸番号 ＜「Slave」（従軸）に設定されているモーション制御システム変数の軸番号] となるように主軸、従軸を使用しなければなりません。

● 軸種別と位置種別の関係

監視可能な軸種別と監視する位置種別の関係は以下のようになります。

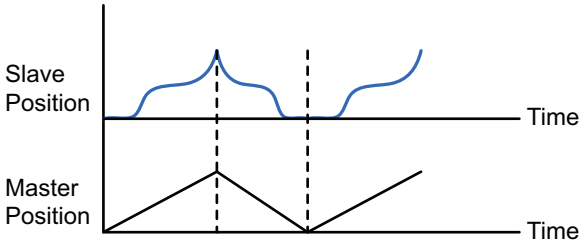
軸種別	ReferecneType	
	_mcCommand、または _mcLatestCommand	_mcFeedback
サーボ軸	○	○
エンコーダ軸	× *1	○
仮想サーボ軸	○	○
仮想エンコーダ軸	× *1	○

*1. 命令起動時に、「位置種別選択範囲外（エラーコード: 5430 Hex）」が発生します。

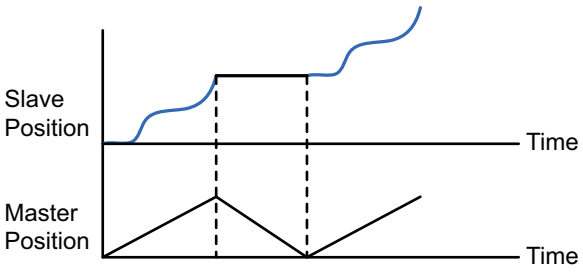
● Direction（方向選択）

主軸の移動方向が「Direction」（方向選択）と一致する場合だけ、従軸のカム動作を開始させることができます。
「Direction」（方向選択）が有効となるのは「InSync」（同期中）中になります。

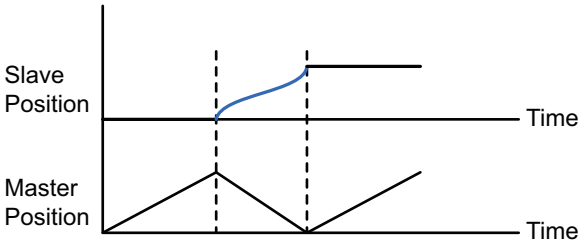
_mcNoDirection（方向指定なし）
主軸が正方向、負方向のどちらに移動中でもカム動作を開始します。



_mcPositiveDirection（正方向指定）
主軸が正方向に移動中にカム動作を開始します。



_mcNegativeDirection（負方向指定）
主軸が負方向に移動中にカム動作を開始します。



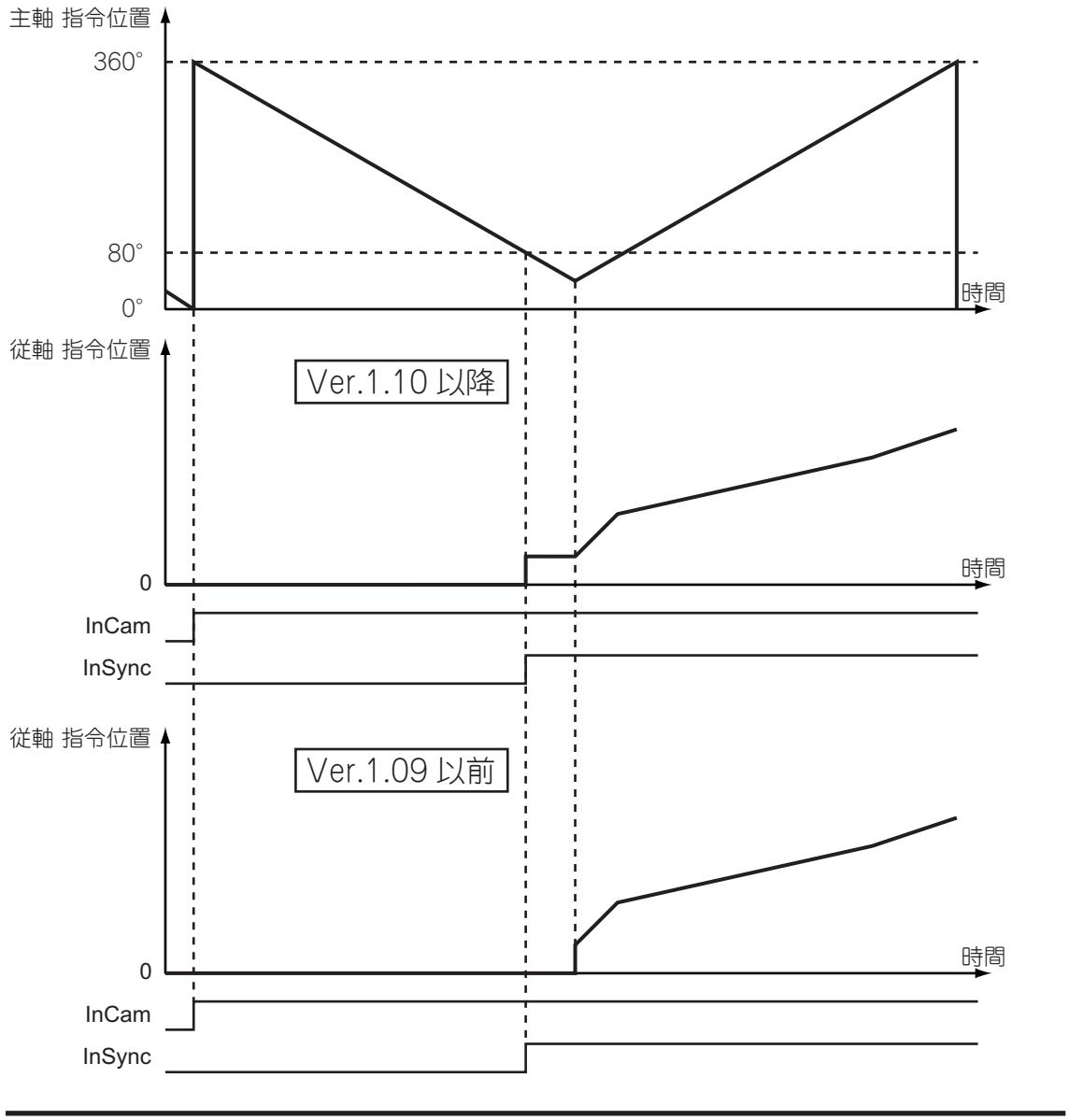
バージョン関連情報

主軸が「Direction」（方向選択）と逆方向に移動中、「MasterStartDistance」（主軸追従距離）を超え、「InSync」（同期中）が TRUE となる場合、CPU ユニットのユニットバージョンにより動作が次のようになります。

- **ユニットバージョン Ver.1.10 以降の CPU ユニットの場合**
  「InSync」（同期中）が TRUE となった時点で、従軸はカムテーブルの位相に従った変位に移動します。その後、主軸が「Direction」（方向選択）と一致する方向へ移動すると、従軸はカム動作を開始します。
- **ユニットバージョン Ver.1.09 以前の CPU ユニットの場合**
  「InSync」（同期中）が TRUE になった後、主軸が「Direction」（方向選択）と一致する方向へ移動すると、従軸はカム動作を開始します。

次に例を示します。カムテーブルの設定は前述の例と同じです。
カム動作の開始条件は以下です。

入力変数	条件
StartMode（開始位置方式指定）	_mcAbsolutePosition（絶対位置）
Direction（方向選択）	_mcPositiveDirection（正方向指定）
StartPosition（カムテーブル開始位置）	0
MasterStartDistance（主軸追従距離）	80



- **BufferMode（バッファモード選択）**

  1 つ前の軸動作と今回の動作のつなぎ方を指定します。
  次の 6 種類があります。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。 主軸が「StartPosition」（カムテーブル開始位置）を通過した後、「MasterStartDistance」（主軸追従距離）を通過すると、従軸のカム動作が開始し、出力変数「InSync」（同期中）が TRUE となります。 バッファリングされている本命令の出力変数「InSync」（同期中）が TRUE となるまで、従軸は停止しています。 カムテーブル始点は、「StartPosition」（カムテーブル開始位置）となります。

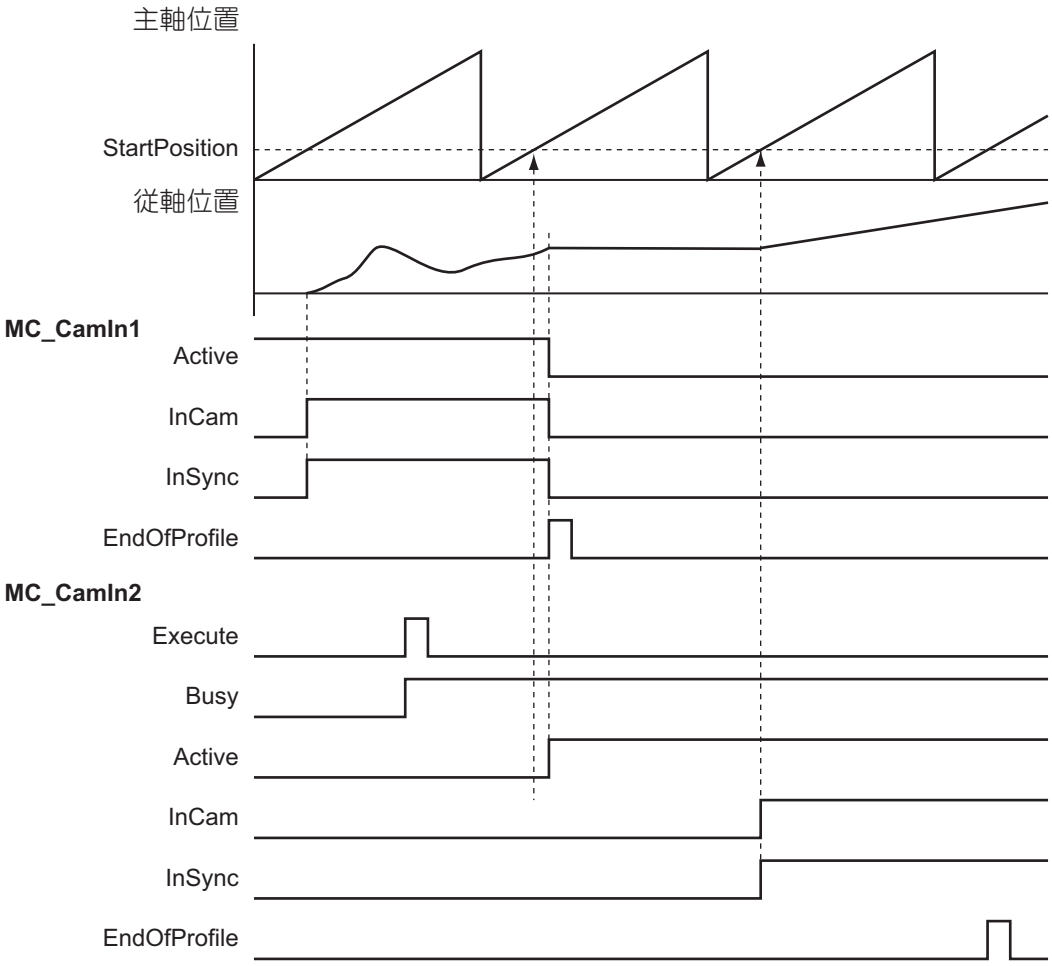
バッファモード選択		説明
バッファード		現在実行中の命令が正常終了した周期から、バッファリングされている本命令が自動的に起動します。主軸が「StartPosition」（カムテーブル開始位置）を通過した後、「MasterStartDistance」（主軸追従距離）を通過すると、従軸のカム動作が開始し、出力変数「InSync」（同期中）が TRUE となります。 バッファリングされている本命令の出力変数「InSync」（同期中）が TRUE となるまで、従軸は停止しています。 カムテーブル始点は、「StartPosition」（カムテーブル開始位置）となります。
ブレンディング *1		現在実行中の命令が正常終了した周期から、バッファリングされている本命令の出力変数「InSync」（同期中）が TRUE となり、従軸が停止することなくカム動作を開始します。 バッファリングされている本命令の「StartPosition」（カムテーブル開始位置）、「MasterStartDistance」（主軸追従距離）を指定している場合でも、指定値にかかわらず、命令の起動と同時に従軸がカム動作を開始します。 カムテーブル始点は、現在実行中の命令の終了位置となります。
	ブレンディングロウ	
	ブレンディングプレビアス	
	ブレンディングネクスト	
	ブレンディングハイ	

*1. ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで使用可能です。
4 種類のブレンディングのいずれを指定しても同じ動作となります。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

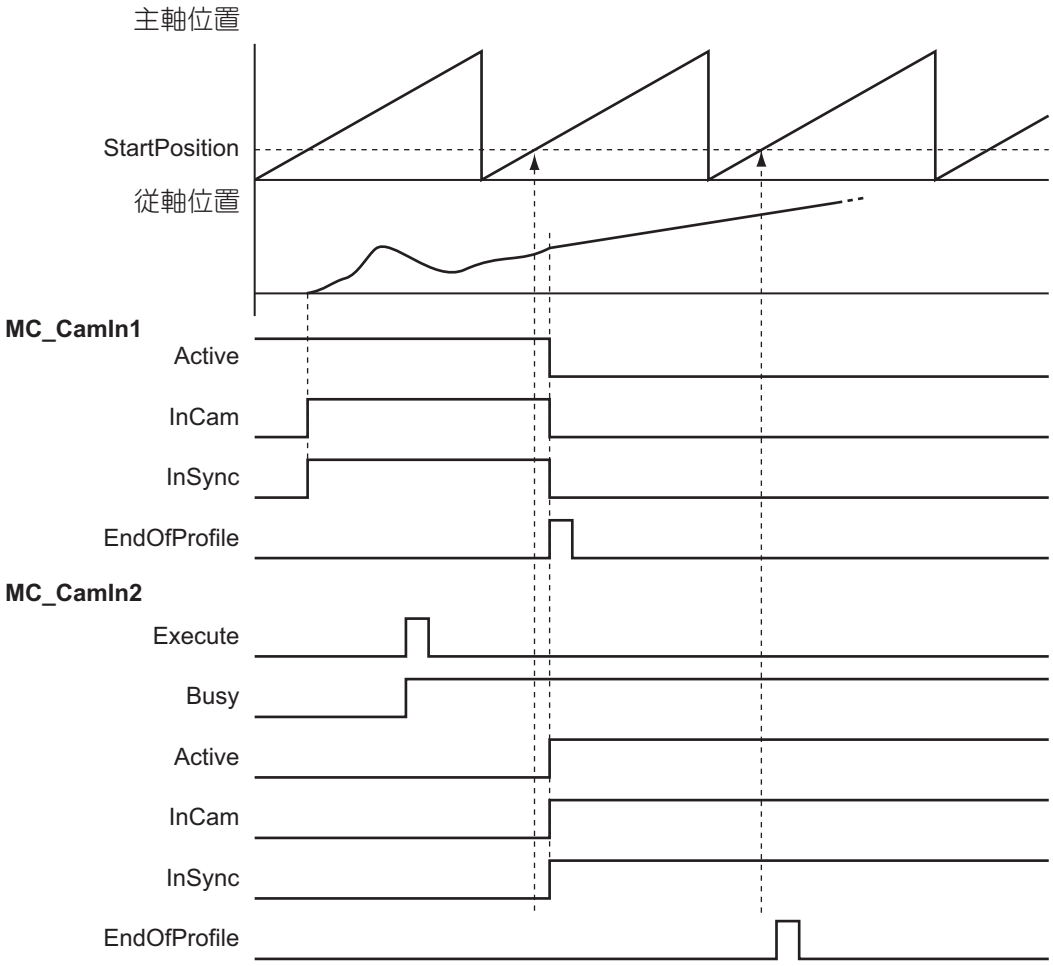
●［バッファード］を指定して多重起動する場合

MC_CamIn1 の実行中に、［バッファード］を指定して MC_CamIn2 を多重起動します。MC_CamIn1 が終了し、MC_CamIn2 の「Active」（制御中）が TRUE となった次の「StartPosition」（カムテーブル開始位置）で「InSync」（同期中）が TRUE となり、カム動作が開始します。



●［ブレンディング］を指定して多重起動する場合

MC_CamIn1 の実行中に、［ブレンディング］を指定して MC_CamIn2 を多重起動します。
MC_CamIn1 が終了した同じ周期で、MC_CamIn2 の「InSync」（同期中）が TRUE となり、カム動作が開始します。



● インデックス

主軸、従軸の指令位置を求めるために使用した 2 つのカムデータのうち、カムデータインデックス番号が小さい方を出力変数「Index」（インデックス）で出力します。
この値は、カムエディタもしくはユーザプログラムによりカムデータを微調整する場合などに使用します。

● インポジションチェック

本命令では、インポジションチェックは行いません。

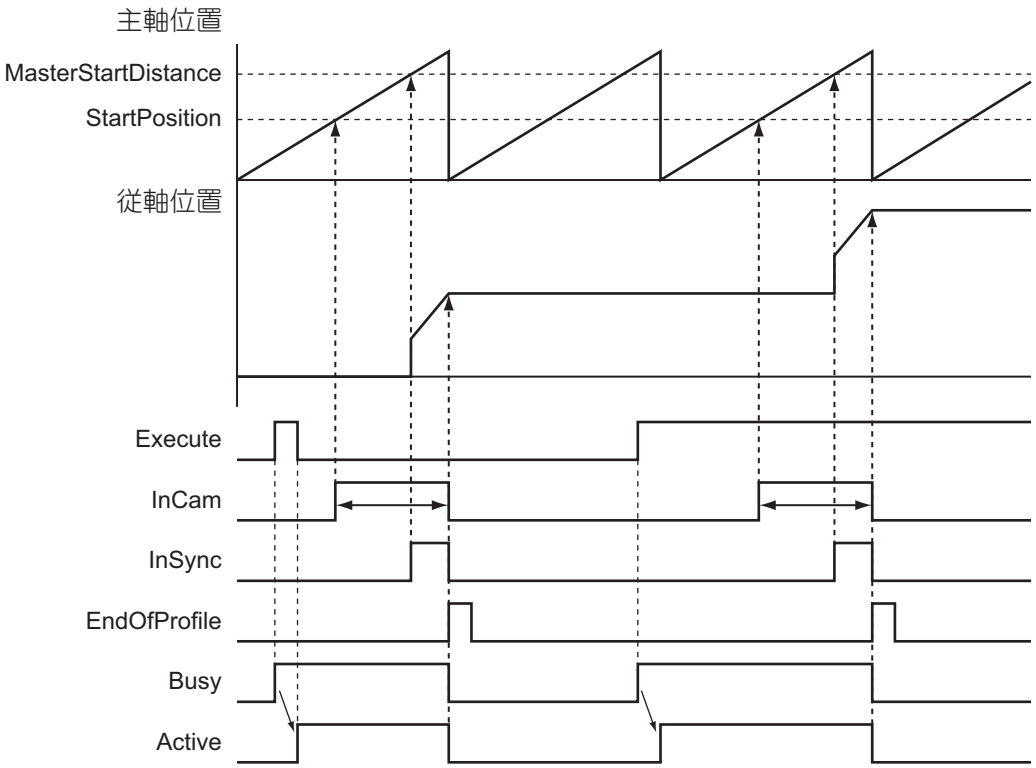
● オーバーライド

本命令では、MC_SetOverride（オーバーライド値設定）命令によるオーバーライドはできません。

タイミングチャート

● 繰り返しなしの場合

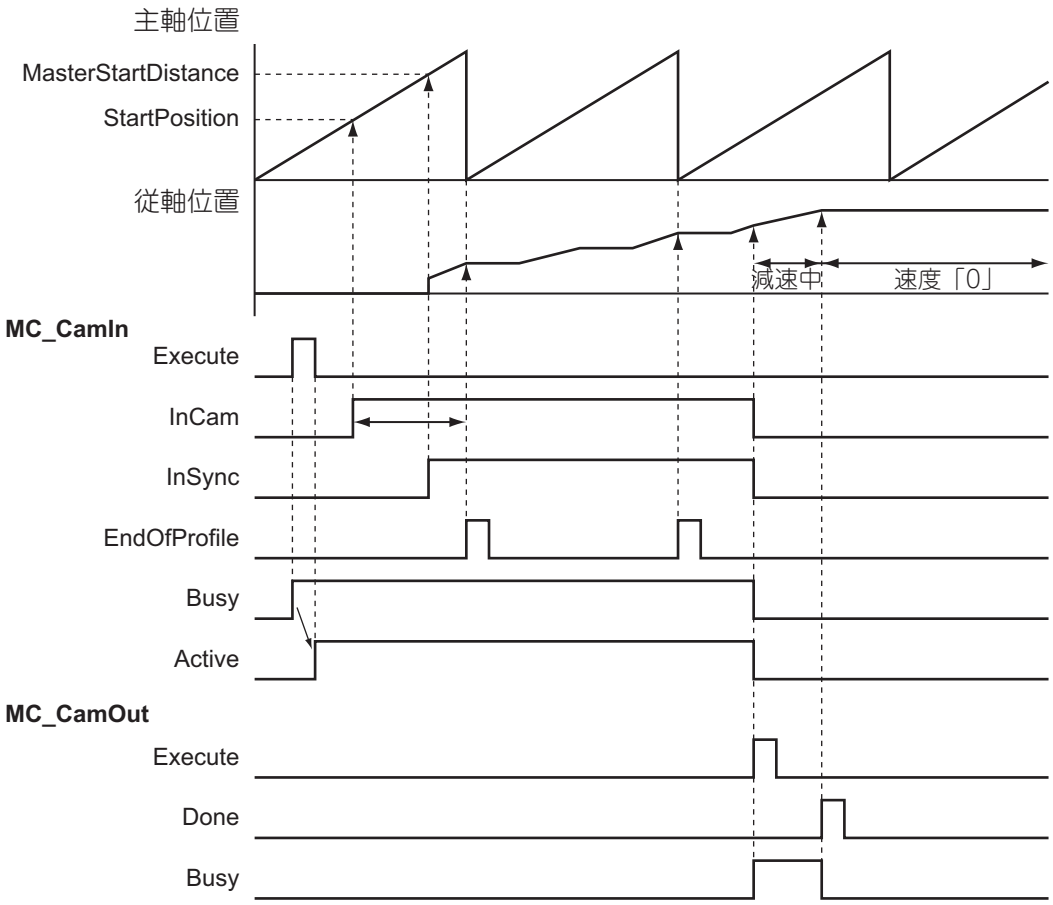
MC_CamIn（カム動作開始）命令の「Periodic」（繰り返しモード）が FALSE［繰り返しなし］の場合、下図のようになります。



● 繰り返しありの場合

カム動作が繰り返し実行されます。
MC_CamOut（カム動作解除）命令を用いてカムを解除する場合、従軸は速度「0」まで減速します。
主軸の動作には影響を与えません。

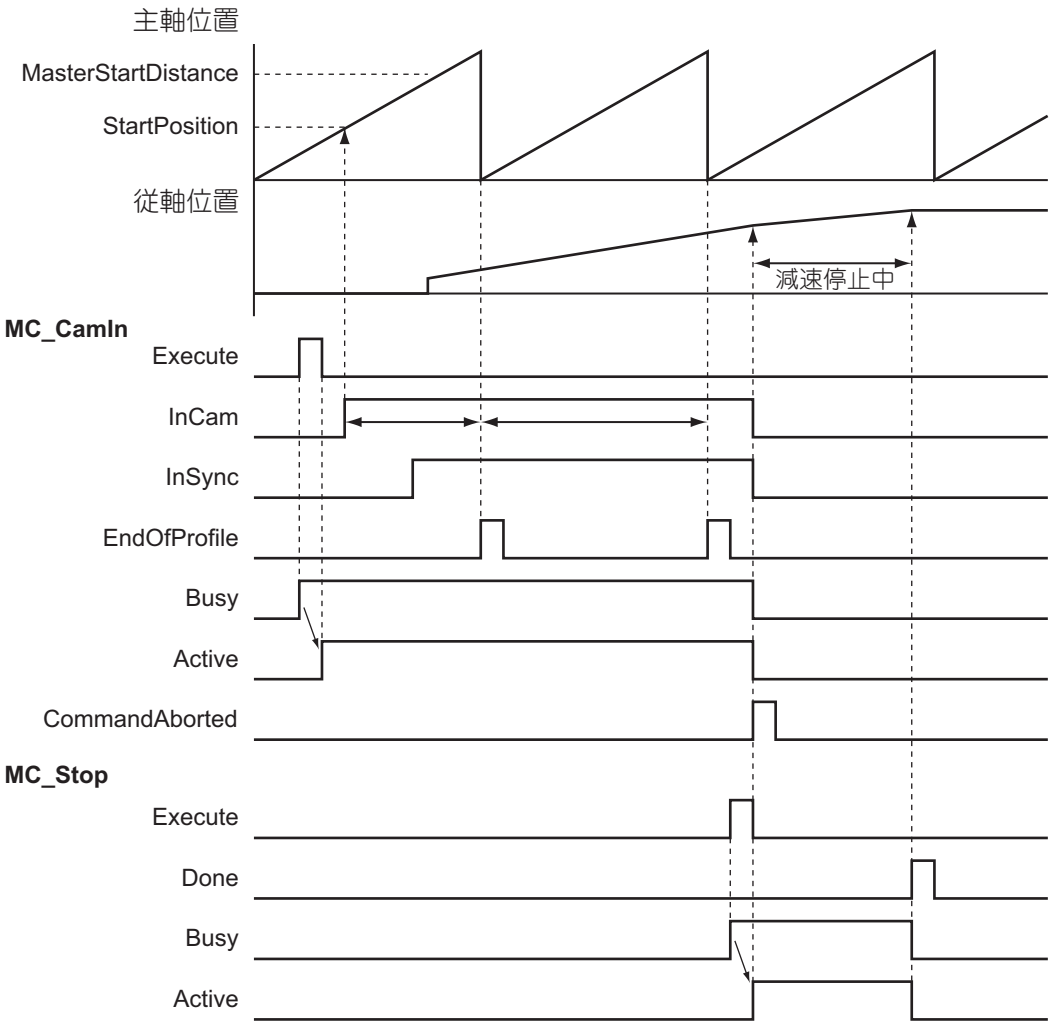
MC_CamIn（カム動作開始）命令の「Periodic」（繰り返しモード）が TRUE［繰り返し］と MC_CamOut（カム動作解除）命令を実行した場合は、下図のようになります。



● MC_Stop（強制停止）命令の場合

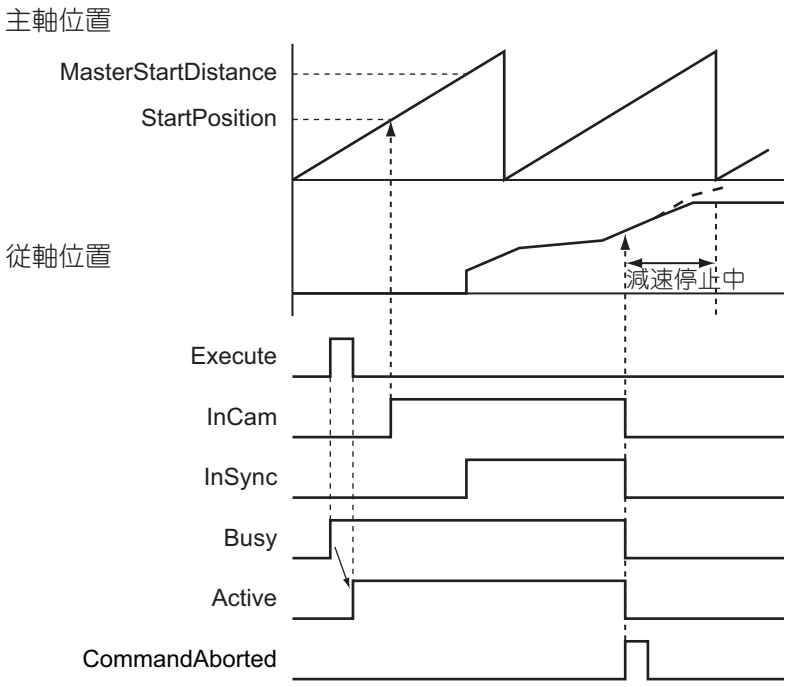
カム動作中に主軸に対して MC_Stop（強制停止）命令を起動した場合、主軸と従軸の同期関係は維持されます。
カム動作中の従軸に対して MC_Stop（強制停止）命令を起動した場合、主軸と従軸の同期関係は終了します。

MC_CamIn（カム動作開始）命令の「Periodic」（繰り返しモード）が TRUE［繰り返し］と従軸に対して MC_Stop（強制停止）命令を実行した場合は、下記のようになります。



命令の中止

本命令実行中に従軸で軸異常が発生した場合、従軸は軸の最大減速度で停止します。従軸に対して MC_Stop（強制停止）命令を起動した場合、「CommandAborted」（実行中断）が TRUE となり、従軸は MC_Stop（強制停止）命令で指定した減速度で停止します。
軸異常の発生要因の特定方法は、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。
なお、主軸で軸異常が発生した場合、カム動作は継続します。



モーション命令再起動

動作中のカムテーブルを切り替える場合、本命令の再起動を行うことが可能です。カムテーブルの切り替え以外の条件を変更する場合には、本命令に対して命令多重起動を実行してください。
命令再起動時、「InCam」（カム動作中）、「InSync」（同期中）は再起動前の状態を維持します。「InSync」（同期中）中に再起動した動作は、現在位置から求められる位相より大きい位相からカム動作を行います。
現在位置から求められる位相とは、カムデータ間を直線で補間した際に求められる位相のことです。

なお、再起動対象は、カムテーブルだけです。

使用上の注意

カム動作中に命令再起動によりカムテーブルを切り替える場合、切り替えの前後で従軸の速度や加速度が急激に変化することがあります。
メカ機構に影響を与える場合があるので、速度や加速度が過度に不連続にならないか十分に検証したうえでカムテーブルを切り替えてください。

モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 再起動方法

再起動方法については、「サンプルプログラム 1 (P.3-187)」、および「サンプルプログラム 2 (P.3-198)」を参照してください。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

本命令実行中に、同一の「CamTable」（カムテーブル）を使用する MC_GenerateCamTable（カムテーブル生成）命令を多重起動することはできません。
また、本命令に対するモーション命令多重起動を行う場合、従軸を指定してください。
なお、本命令動作中にほかの命令を多重起動する場合、「バッファモード」により以下の制約があります。

- ほかの命令を［アボーティング］で多重起動した場合、その時点でカム動作を中断し、次の動作を開始します。
- ほかの命令を［バッファード］で多重起動した場合、「EndOfProfile」（カムサイクル完了）が TRUE になったタイミングで、次の動作の指令値を出力します。

● 本命令同士のブレンディングでの多重起動

- 本命令同士で多重起動する場合だけ、［ブレンディング］を指定できます。ほかの命令を実行中に本命令を［ブレンディング］で多重起動できません。
- 「Master」（主軸）と「ReferenceType」（位置種別選択）は、現在実行中の命令とそれぞれ同じ値を指定してください。異なった値を指定すると、「モーション命令多重起動不可」異常が発生します。
- 現在実行中の命令が正常終了した周期から、バッファリングされている本命令の出力変数「InSync」（同期中）が TRUE となり、従軸が停止することなくカム動作を開始します。
- 「StartPosition」（カムテーブル開始位置）、「MasterStartDistance」（主軸追従距離）を指定している場合でも、指定値にかかわらず、命令の起動と同時に従軸がカム動作を開始します。カムテーブル始点は、現在実行中の命令の終了位置となります。4 種類のブレンディングのいずれを指定しても同じ動作となります。

● 他命令実行中の本命令起動

MC_GenerateCamTable（カムテーブル生成）命令を実行中に、同一の「CamTable」（カムテーブル）を使用する本命令を多重起動することはできません。

● 主軸または従軸に対する補正

同期制御中の主軸または従軸に対して補正を行う命令は、次の 3 つがあります。

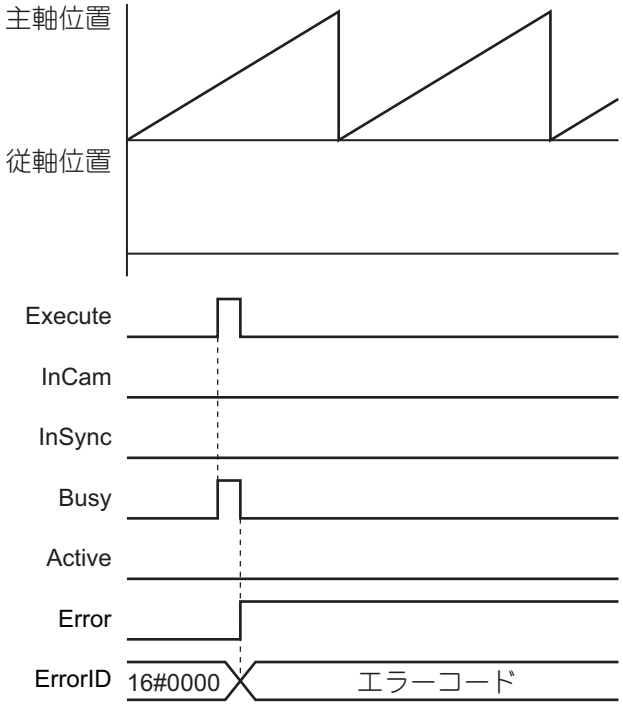
補正	命令
主軸に対する補正	MC_Phasing（主軸相対値位相補正）
従軸に対する補正	MC_SyncOffsetPosition（サイクリック同期位置オフセット補正） MC_OffsetPosition（位置オフセット補正）

MC_CamIn（カム動作開始）命令以外の同期制御命令では、多重起動により動作中の同期制御命令が「CommandAborted」（実行中断）になると、補正を行う命令も同時に「CommandAborted」（実行中断）となります。
MC_CamIn（カム動作開始）命令同士をブレンディングで多重起動する場合、補正を行う命令は「CommandAborted」（実行中断）とならずに処理を継続します。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● **異常発生時のタイミングチャート**



● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム 1

ここでは、繰り返しありのカム動作についてサンプルプログラムを掲載します。
本例では主軸を軸 1、従軸を軸 2 とします。

参考

予約(Reserved)扱いの入力変数には初期値以外の値を指定できません。
この例では各命令で定義された初期値を持つ変数を割り付けていますが、プログラミングされる際には変数やパラメータを割り付ける必要はありません。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● **軸パラメータの設定**

軸種別

軸	軸種別
軸 1	サーボ軸（主軸）
軸 2	サーボ軸（従軸）

カウンタモード

軸	カウンタモード
軸 1	ロータリモード
軸 2	リニアモード

リングカウンタ

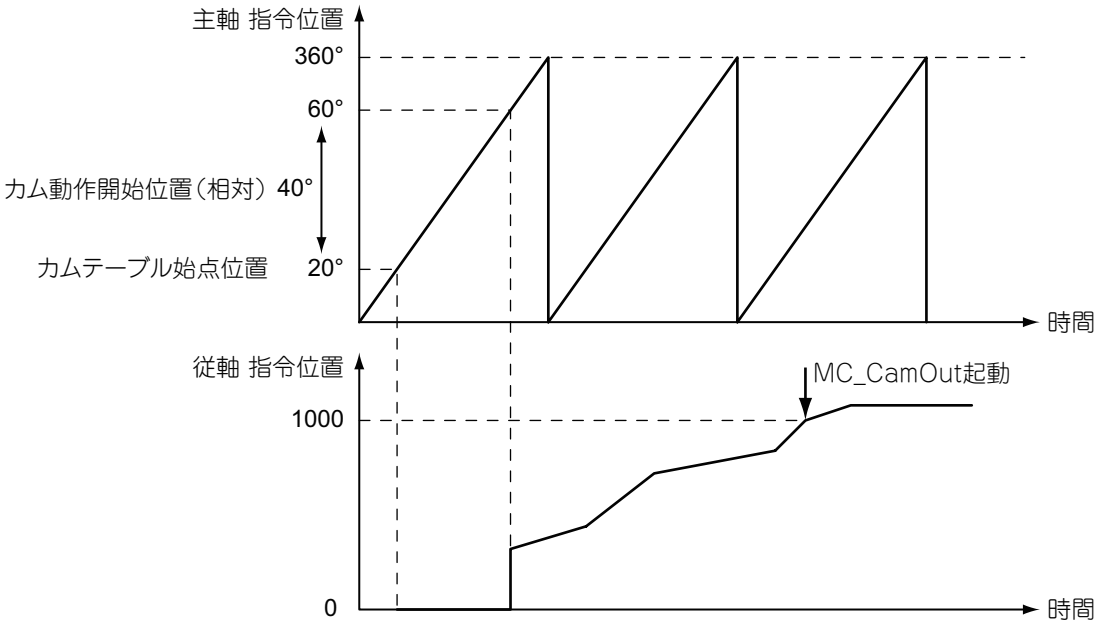
軸	上限値	下限値
軸 1	360	0

座標単位選択

軸	座標単位
軸 1	degree
軸 2	mm

動作例

● 動作パターン



1 **カム動作の開始**

カムテーブル始点位置（位相ゼロの位置）を 20° とし、そこから相対角度 40° 離れた角度に主軸が来たら、従軸はカム動作を開始します。カム動作は繰り返し動作で動きます。
「Periodic」（繰り返しモード） を TRUE とすることにより、繰り返し動作となります。

2 **カム動作の解除**

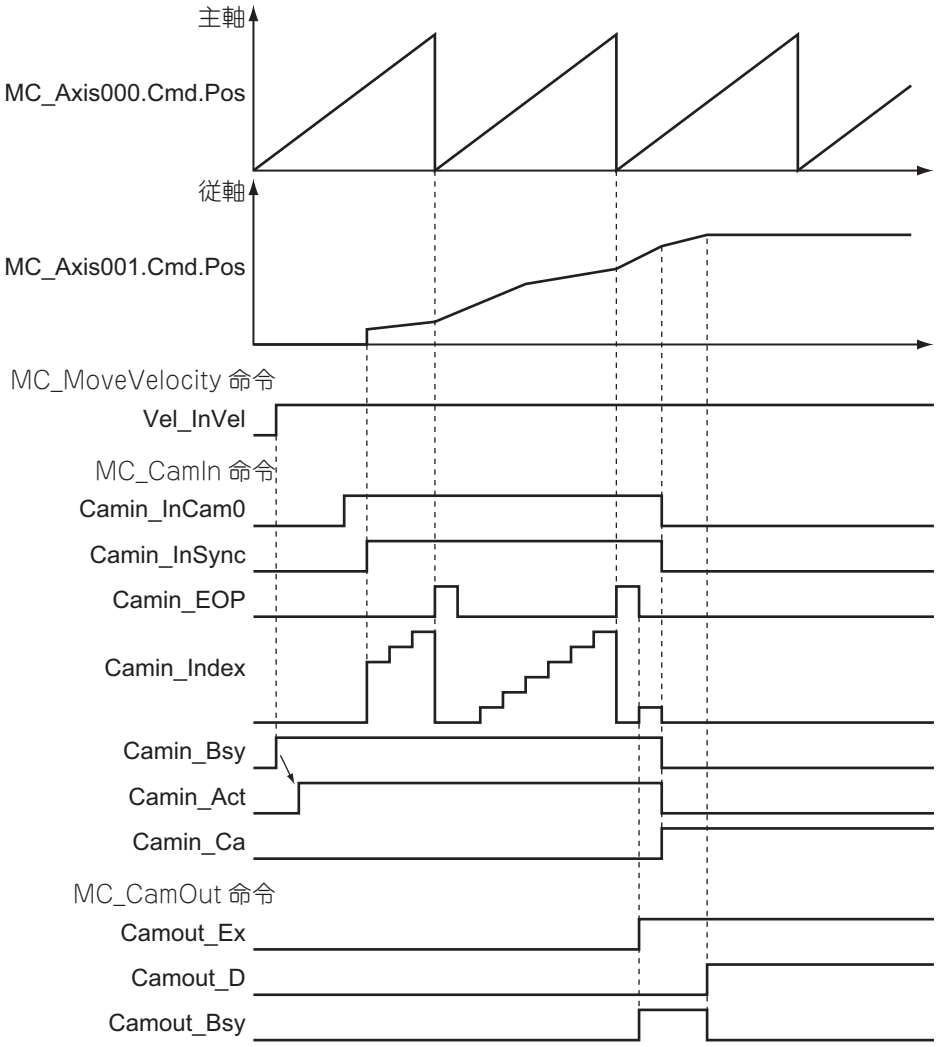
従軸のフィードバック位置(「MC_Axis001.Act.Pos」)が 1000.0 以上になったらカム動作を解除し、減速度(DecRate2)にて従軸を停止させます。

ラダー図

● 主な変数

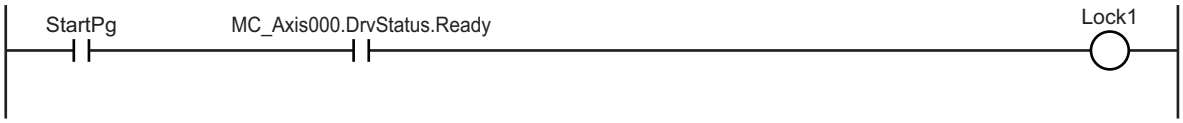
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態のとき、TRUE となります。
Pwr1_S	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr2_S	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	−	カムデータ変数です。 配列の要素数 ARRAY[0..N]は、カムエディタで設定します。この例では 0～360 としますが、カムエディタの設定により配列の要素数は異なります。
DecRate2	LREAL	10000.0	MC_CamOut 実行時の減速度を設定します。
Vel_InVel	BOOL	FALSE	MC_MoveVelocity のインスタンスである VEL の出力「InVelocity」を割り付けた変数です。目標速度に到達すると、この変数が TRUE となります。
Camin_InCam0	BOOL	FALSE	MC_CamIn のインスタンス CAMIN の出力「InCam」に割り付けた変数です。カム動作中のとき、この変数が TRUE となります。
Camout_Ex	BOOL	FALSE	この変数が TRUE のとき、MC_CamOut のインスタンス CAMOUT が起動されます。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。

● タイミングチャート



● サンプルプログラム

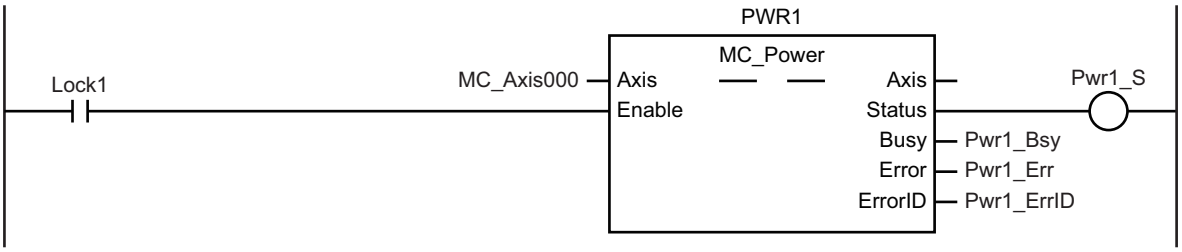
接点 StartPg が TRUE になったら、軸 1 のサーボドライバがサーボレディ状態になっていることを確認する



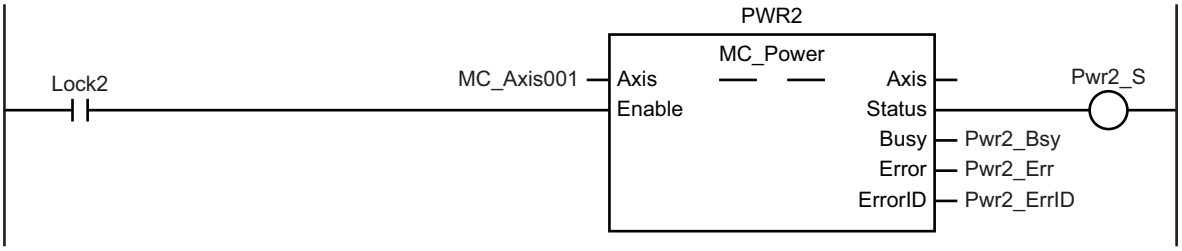
接点 StartPg が TRUE になったら、軸 2 のサーボドライバがサーボレディ状態になっていることを確認する



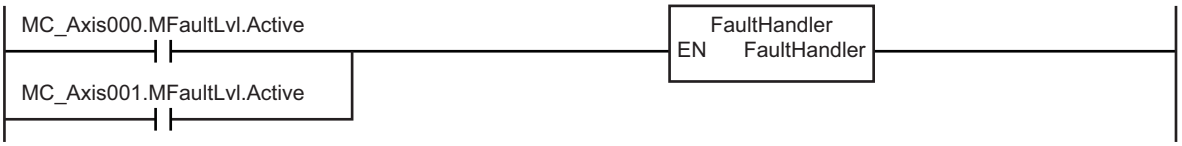
軸 1 がサーボレディ状態になっていれば、サーボ ON 状態にする



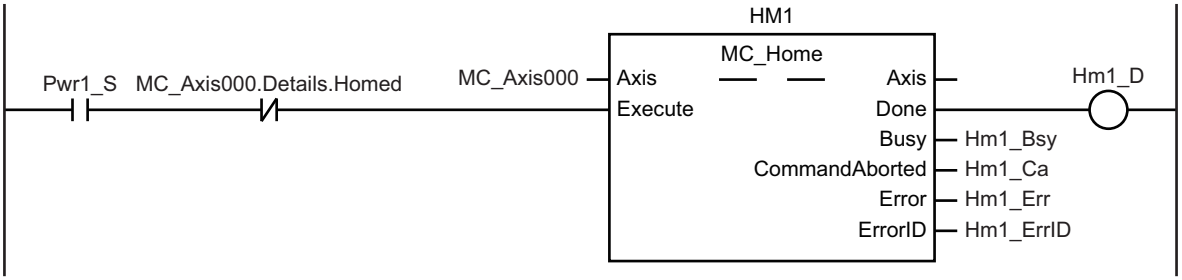
軸 2 がサーボレディ状態になっていれば、サーボ ON 状態にする



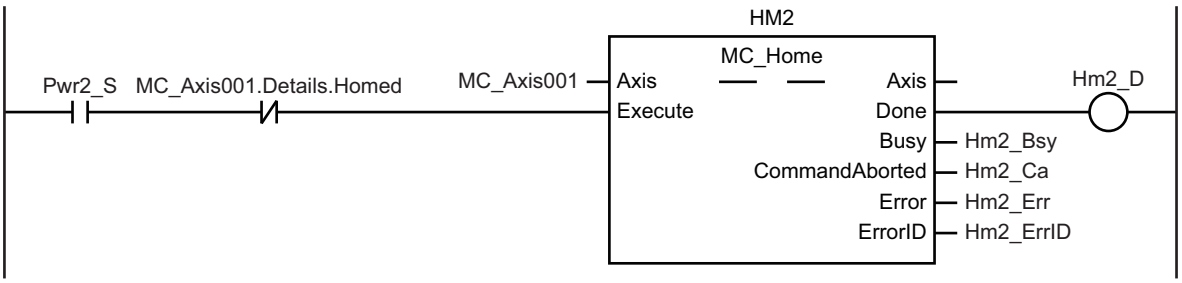
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



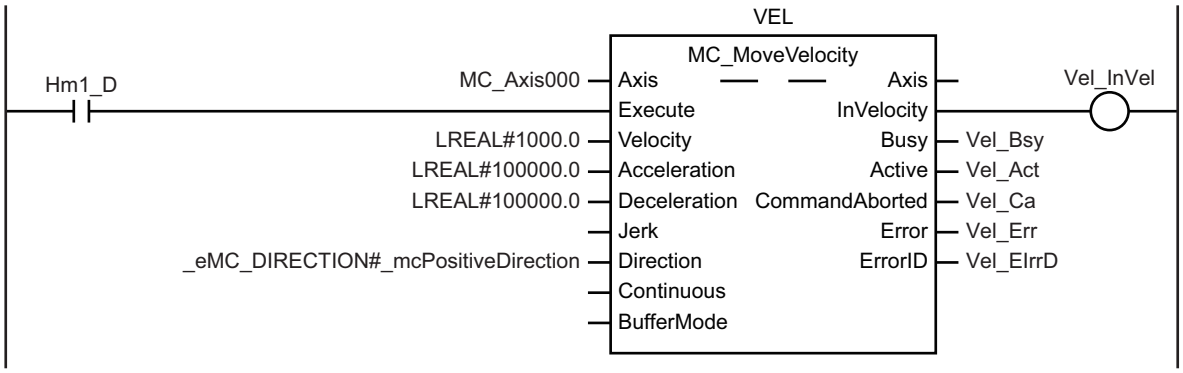
軸 1 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



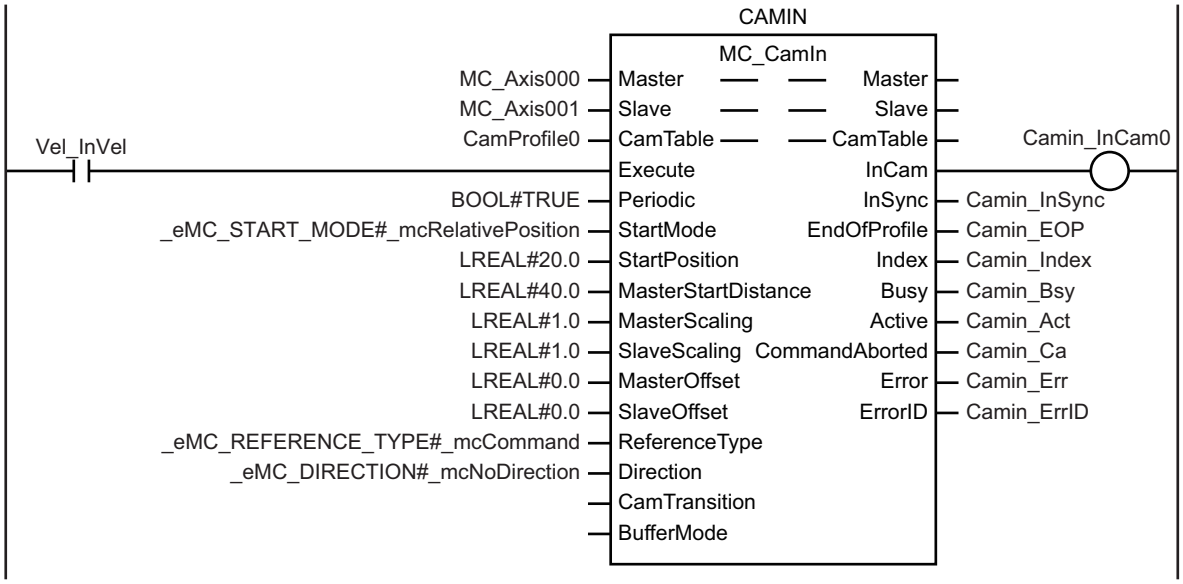
軸 2 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



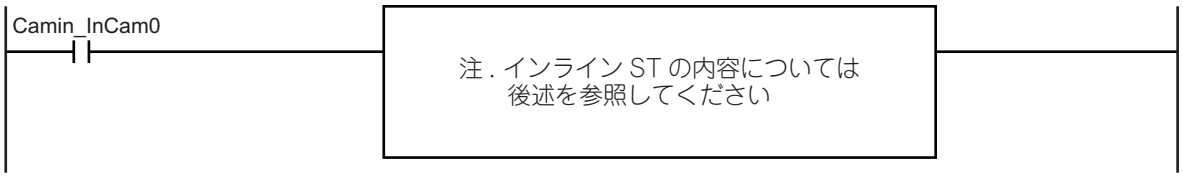
軸 1 が原点復帰完了状態になったら、MC_MoveVelocity(速度制御）命令を実行する



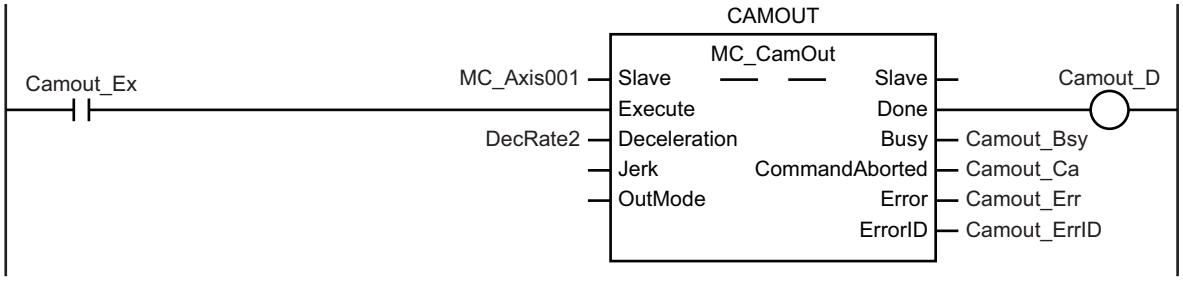
MC_MoveVelocity（速度制御）命令の Vel_InVel が TRUE になったら、MC_CamIn（カム動作開始）命令を起動する



Camin_InCam0 が TRUE で、MC_Axis001.Act.Pos が 1000 を超えたら、CamOut_Ex を TRUE にする



Camout_Ex を TRUE にすることで、MC_CamOut(カム動作解除）命令を起動する。
減速度として、DecRate2 を指定する



インライン ST の内容

```
IF MC_Axis001.Act.Pos>LREAL#1000.0 THEN
    Camout_Ex := TRUE;
END_IF;
```

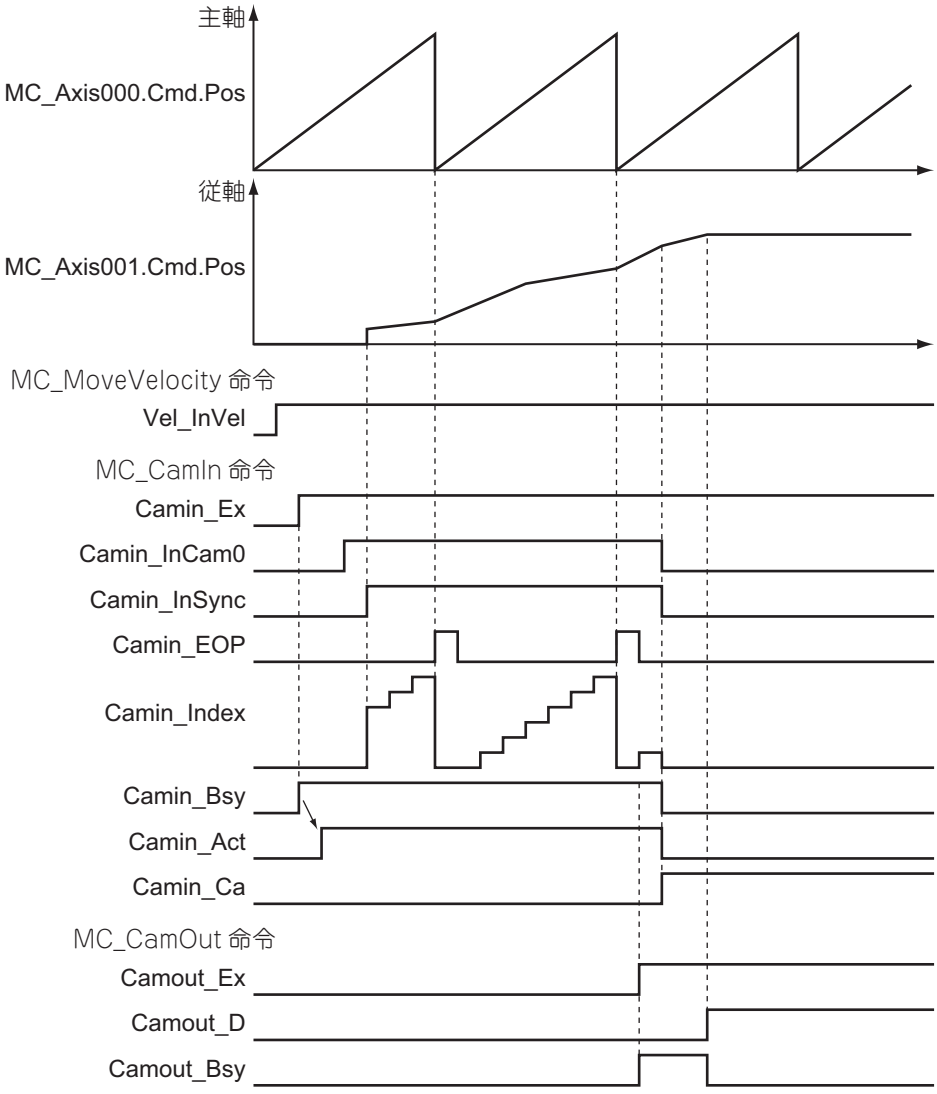
ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態のとき、TRUE となります。
Pwr1_S	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr2_S	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	−	カムデータ変数です。 配列の要素数 ARRAY[0..N]は、カムエディタで設定します。この例では 0～360 としますが、カムエディタの設定により配列の要素数は異なります。
DecRate2	LREAL	10000.0	MC_CamOut 実行時の減速度を設定します。
Vel_InVel	BOOL	FALSE	MC_MoveVelocity のインスタンスである VEL の出力「InVelocity」を割り付けた変数です。目標速度に到達すると、この変数が TRUE となります。
Camin_InCam0	BOOL	FALSE	MC_CamIn のインスタンス CAMIN の出力「InCam」に割り付けた変数です。カム動作中のとき、この変数が TRUE となります。
Camout_Ex	BOOL	FALSE	この変数が TRUE のとき、MC_CamOut のインスタンス CAMOUT が起動されます。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
C amin_Ex	BOOL	FALSE	Vel_InVel の立ち上がり時に TRUE にすることにより、MC_CamIn のインスタンス CAMIN の「Execute」を TRUE にします。

名称	データ型	初期値	コメント
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。FALSE のとき、入力パラメータを設定します。入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート



● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag=FALSE THEN

    //MC_CamIn パラメータ
    Camin_EM := TRUE; // 繰り返しモード
    Camin_StMode :=_eMC_START_MODE#_mcRelativePosition;
    Camin_StPos := LREAL#20.0; // 始点の主軸絶対位置
    Camin_MStDis := LREAL#40.0; // カム動作開始する主軸位置
    Camin_MSc := LREAL#1.0; // 主軸拡大
    Camin_SSc := LREAL#1.0; // 従軸拡大
    Camin_MO := LREAL#0.0; // 主軸オフセット
```

```
    Camin_SO := LREAL#0.0; // 従軸オフセット
    Camin_RT := _eMC_REFERENCE_TYPE#_mcCommand;//位置種別選択
    Camin_Dir := _eMC_DIRECTION#_mcNoDirection;//方向選択

    //MC_MoveVelocity パラメータ
    Vel_Vel := LREAL#1000.0;
    Vel_Acc := LREAL#100000.0;
    Vel_Dec := LREAL#100000.0;
    Vel_Dir :=_eMC_DIRECTION#_mcPositiveDirection;

    //MC_CamOut パラメータ
    Camout_Dec := DecRate2;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag := TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 2 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
END_IF;

//軸 1～軸 2 で軽度フォールトが発生したら、異常時処理を実行します
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF (MC_Axis000.MFaultLvl.Active=TRUE)
 OR (MC_Axis001.MFaultLvl.Active=TRUE)THEN
    FaultHandler();
END_IF;

// 軸 1 がサーボ ON 状態で、かつ原点未確定状態のときに、軸 1 の原点復帰を実行します
IF (Pwr1_S=TRUE)
 AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;
```

```
// 軸 2 がサーボ ON 状態で、かつ原点未確定状態のときに、軸 2 の原点復帰を実行します
IF (Pwr2_S=TRUE)
 AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

// 軸 1 が原点復帰完了状態で、MC_MoveVelocity を実行します
IF Hm1_D=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

// MC_MoveVelocity の InVel=TRUE で CamIn を起動します
IF Vel_InVel=TRUE THEN
    Camin_Ex := TRUE;
END_IF;

//Camin_InCam0=TRUE かつ MC_Axis001.Act.Pos>1000 で、CamOut を起動します
IF (Camin_InCam0=TRUE)
 AND (MC_Axis001.Act.Pos>LREAL#1000.0) THEN
    Camout_Ex := TRUE;
END_IF;

// 軸 1 の MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_S,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

// 軸 2 の MC_Power
PWR2(
    Axis := MC_Axis001,
    Enable := Pwr2_En,
    Status => Pwr2_S,
    Busy => Pwr2_Bsy,
    Error => Pwr2_Err,
    ErrorID => Pwr2_ErrID
);

// 軸 1 の MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
```

```
);

// 軸 2 の 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
);
```

サンプルプログラム 2

ここでは、液体充填のカム動作についてサンプルプログラムを掲載します。

参考

予約(Reserved)扱いの入力変数には初期値以外の値を指定できません。
この例ではパラメータを指定しません。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

軸種別

軸	軸種別
軸 1	サーボ軸（主軸）
軸 2	サーボ軸（従軸）
軸 3	サーボ軸（従軸）
軸 4	サーボ軸（従軸）
軸 5	サーボ軸（従軸）

カウンタモード

軸	カウンタモード
軸 1	ロータリモード
軸 2	ロータリモード
軸 3	ロータリモード
軸 4	ロータリモード
軸 5	ロータリモード

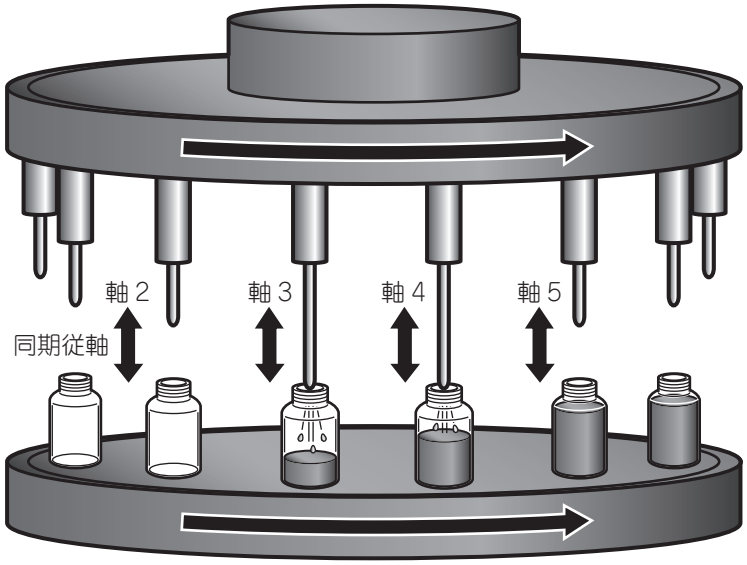
リングカウンタ

軸	上限値	下限値
軸 1	360	0
軸 2	360	0
軸 3	360	0
軸 4	360	0
軸 5	360	0

座標単位選択

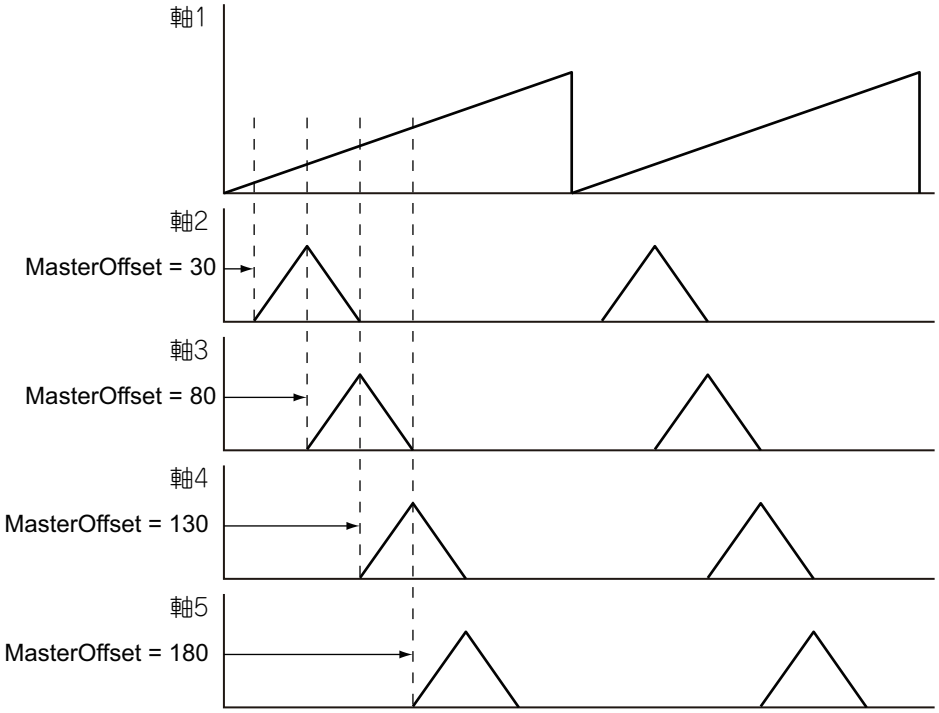
軸	座標単位
軸 1	degree
軸 2	degree
軸 3	degree
軸 4	degree
軸 5	degree

動作例



同期主軸　　軸 1

● 動作パターン



1 **カム動作開始**

主軸（軸 1）に同期して、従軸（軸 2～軸 5）がカム動作を行います。各軸は、50degree ずつ位相がシフトしてカム動作を開始します。

2 **繰り返し**

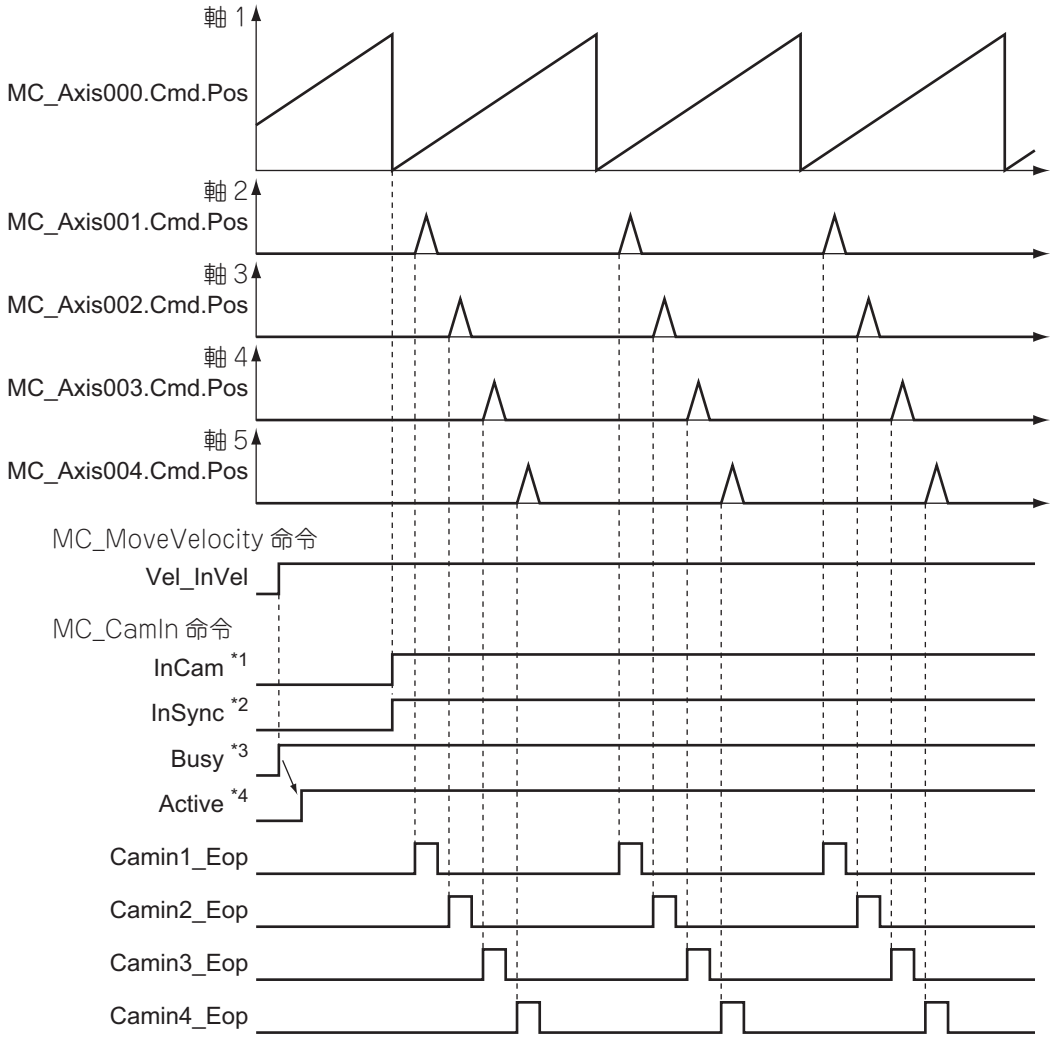
各軸は、指定されたカム動作を繰り返し実行します。

ラダー図

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	－	主軸である軸 1 の軸変数です。
MC_Axis001	_sAXIS_REF	－	従軸である軸 2 の軸変数です。
MC_Axis002	_sAXIS_REF	－	従軸である軸 3 の軸変数です。
MC_Axis003	_sAXIS_REF	－	従軸である軸 4 の軸変数です。
MC_Axis004	_sAXIS_REF	－	従軸である軸 5 の軸変数です。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Vel_InVel	BOOL	FALSE	MC_MoveVelocity のインスタンスである VEL の出力「InVelocity」を割り付けた変数です。目標速度に到達すると、この変数が TRUE となります。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	－	カムデータ変数です。 MC_CamIn のインスタンス CAMIN1～CAMIN4 の入力「CamTable」に、この変数を指定します。 配列の要素数 ARRAY[0..N]は、カムエディタで設定します。

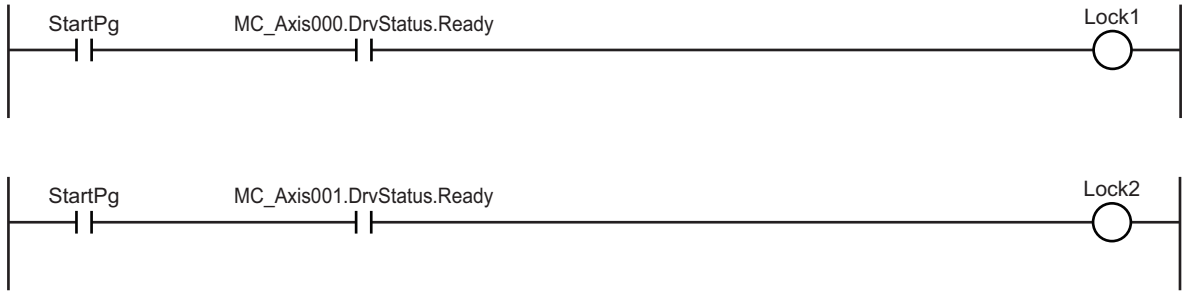
● タイミングチャート

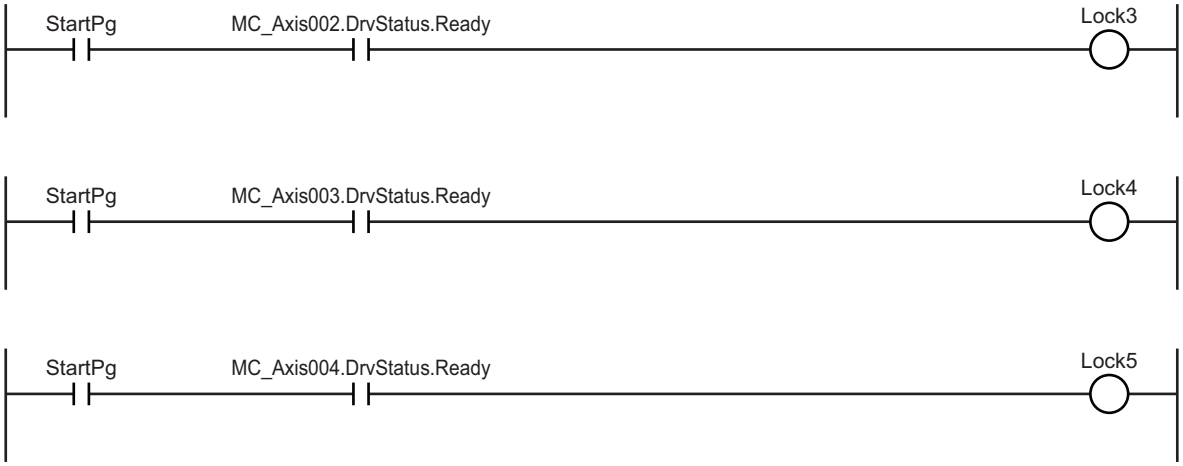


*1. Camin1_InCam, Camin2_InCam, Camin3_InCam, Camin4_InCam が同じタイミングです。
*2. Camin1_InSync, Camin2_InSync, Camin3_InSync, Camin4_InSync が同じタイミングです。
この例では、「MasterStartDistance」が「0」なので、主軸が「0° 」から「InSync」が TRUE となります。
*3. Camin1_Bsy, Camin2_Bsy, Camin3_Bsy, Camin4_Bsy が同じタイミングです。
*4. Camin1_Act, Camin2_Act, Camin3_Act, Camin4_Act が同じタイミングです。

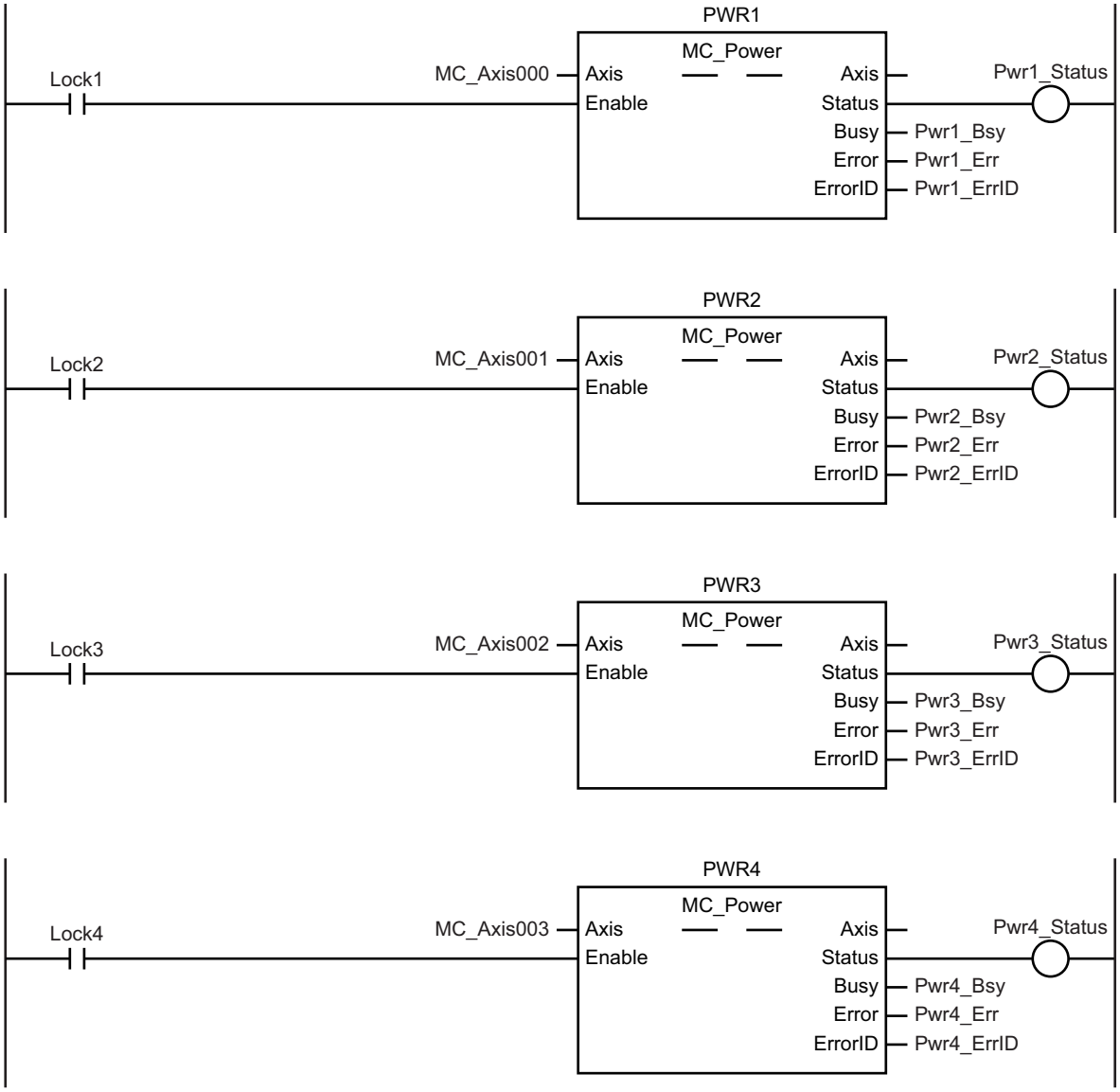
● サンプルプログラム

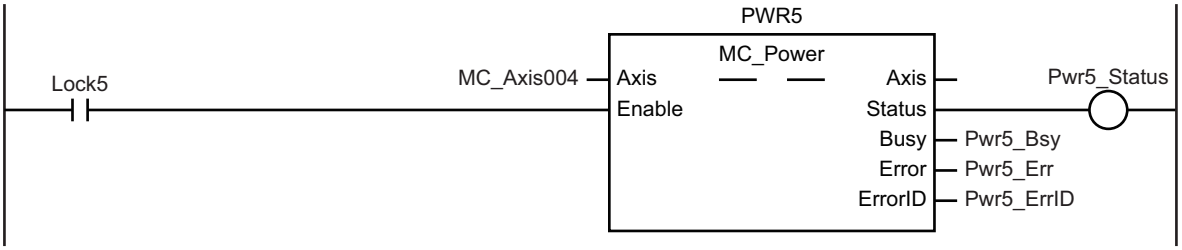
接点 StartPg が TRUE になったら、各軸のサーボドライバがサーボレディ状態になっていることを確認する



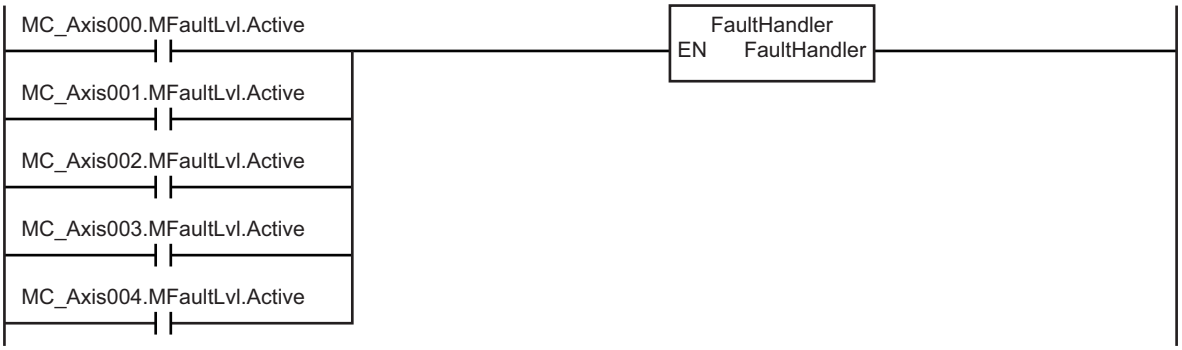


サーボレディ状態になっていれば、各軸をサーボ ON 状態にする

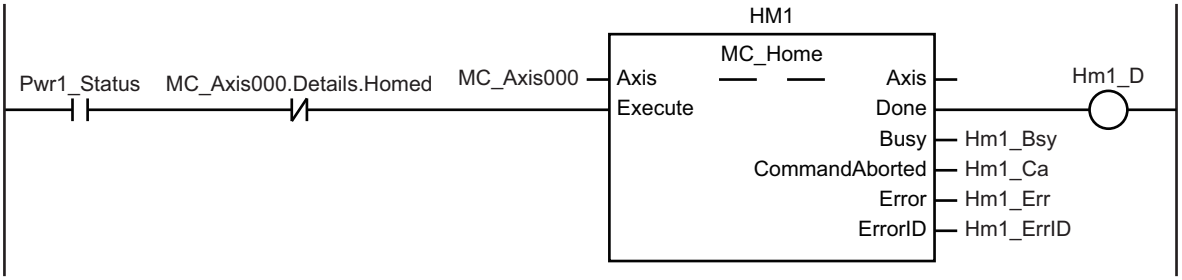


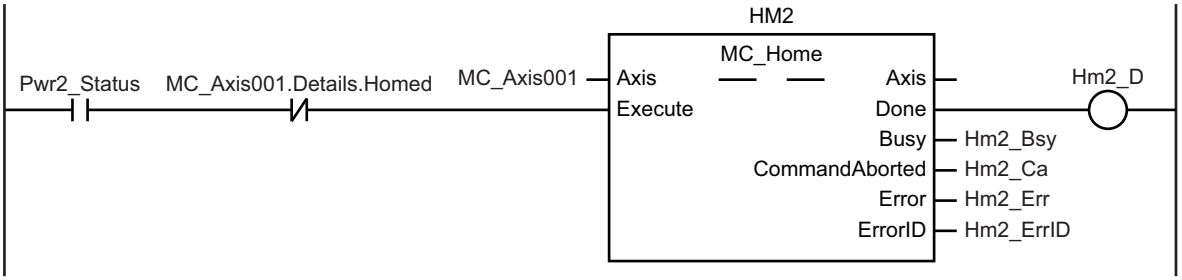


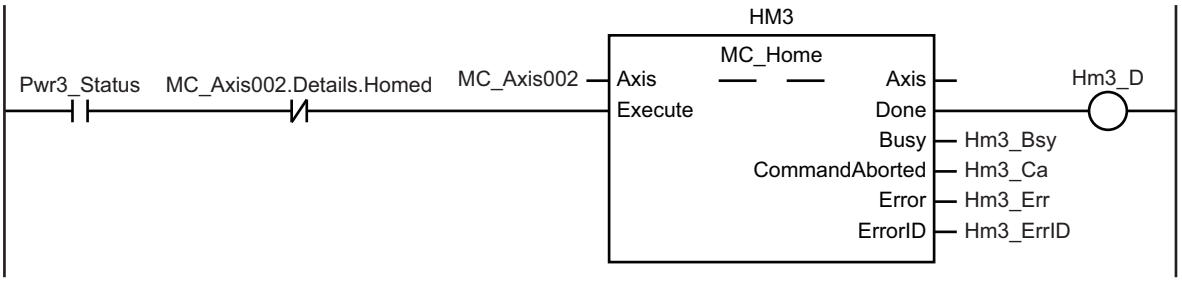
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください

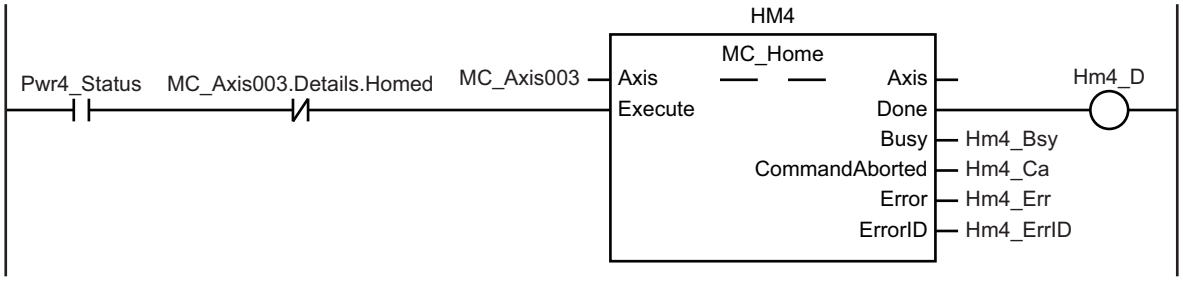


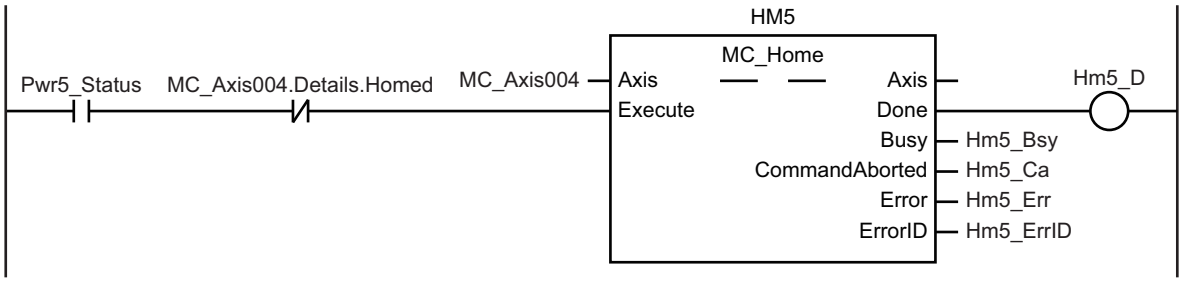
軸 1 ～軸 5 がサーボ ON 状態になり原点未確定状態なら、各軸ごとに原点復帰を実行する



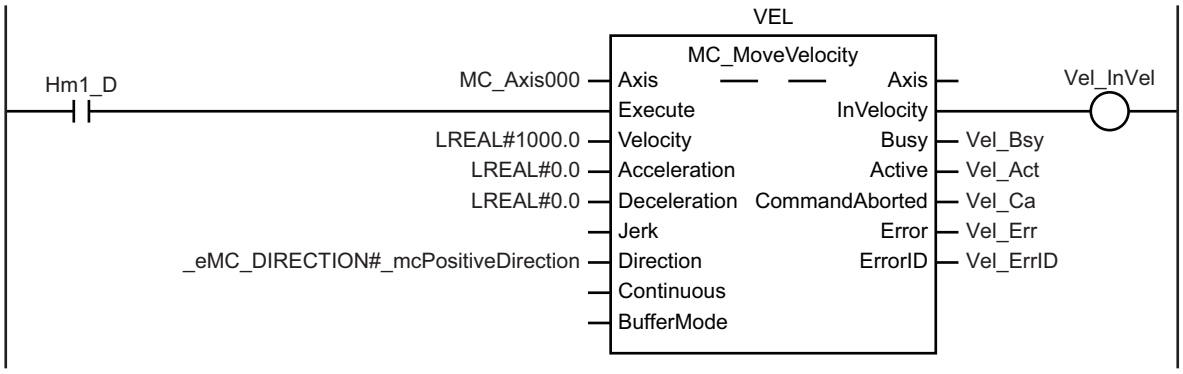




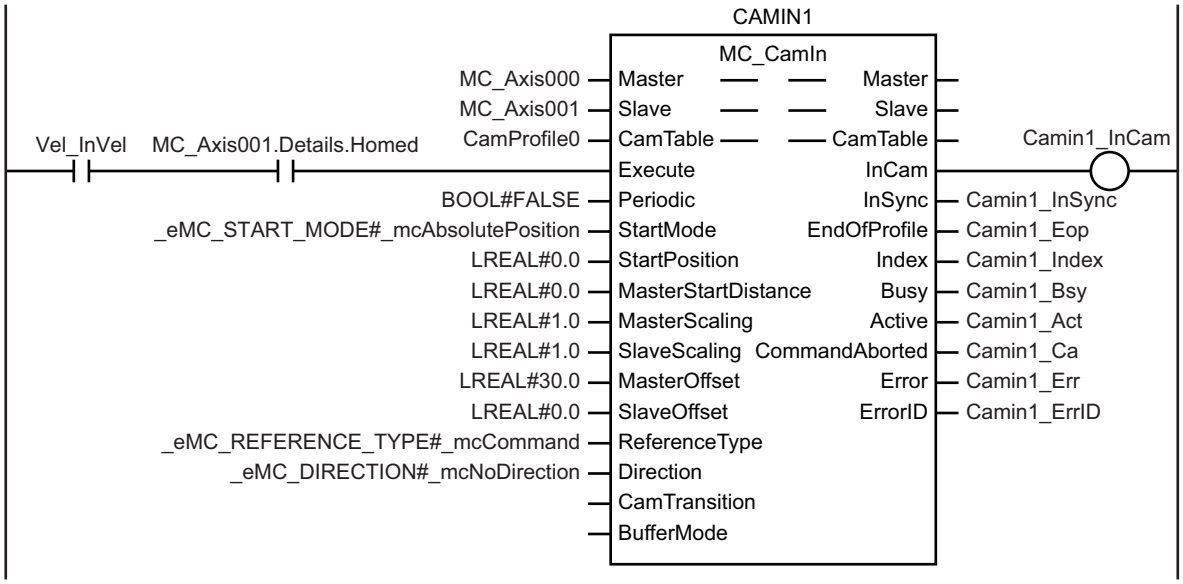




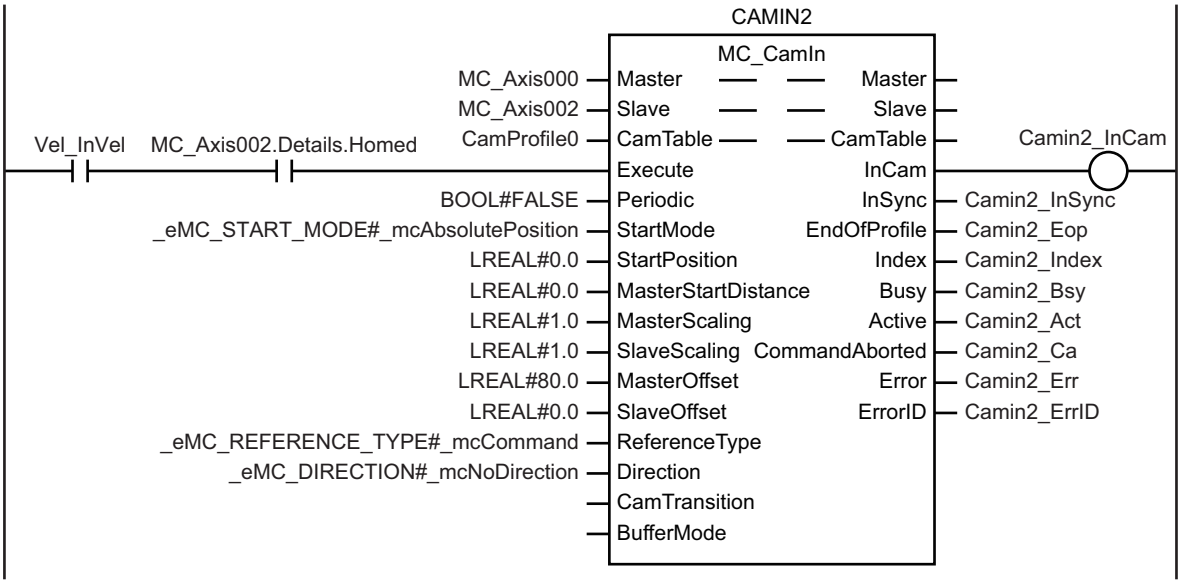
軸 1 の原点復帰完了後に、MC_MoveVelocity（速度制御）命令を起動する



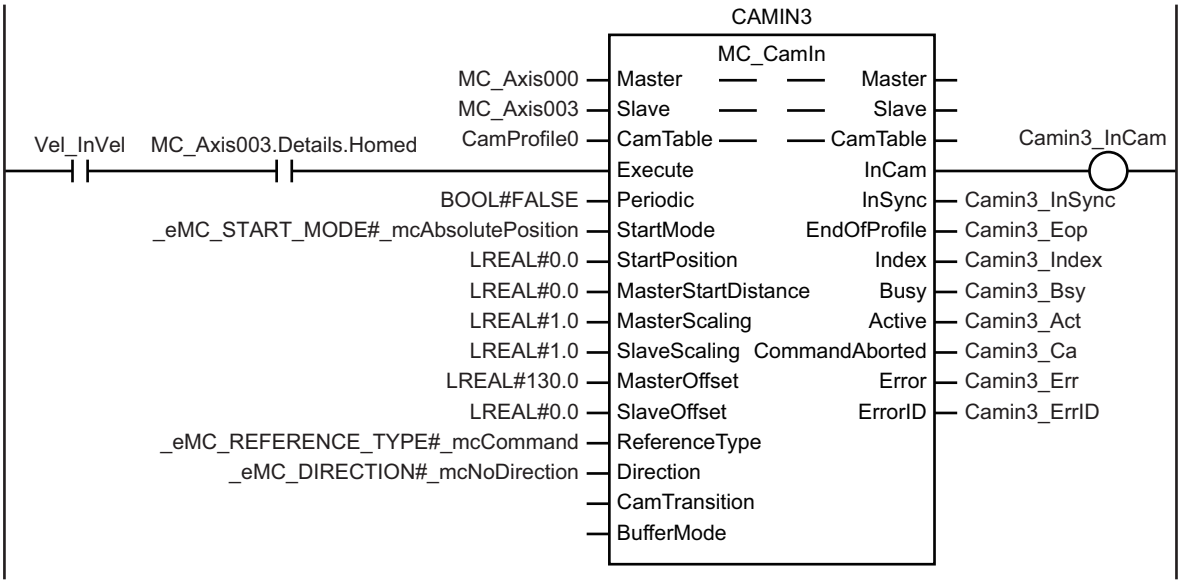
MC_MoveVelocity（速度制御）命令の Vel_InVel が TRUE で、軸 2 の原点復帰完了後に、従軸（軸 2）に設定された MC_CamIn（カム動作開始）命令を実行する



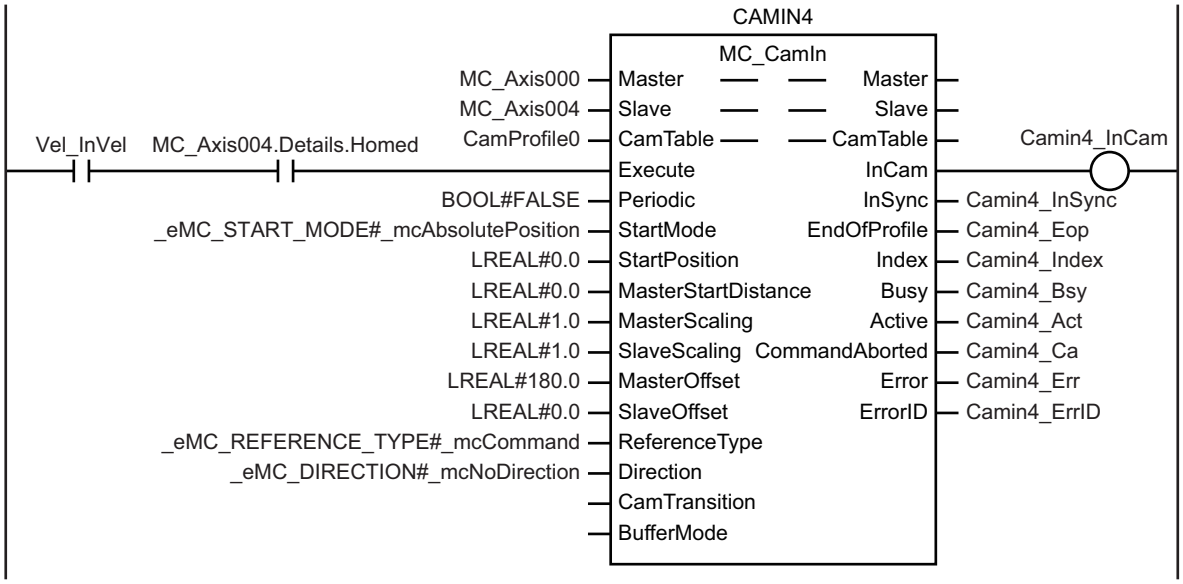
MC_MoveVelocity（速度制御）命令の Vel_InVel が TRUE で、軸 3 の原点復帰完了後に、従軸（軸 3）に設定された MC_CamIn（カム動作開始）命令を実行する



MC_MoveVelocity（速度制御）命令の Vel_InVel が TRUE で、軸 4 の原点復帰完了後に、従軸（軸 4）に設定された MC_CamIn（カム動作開始）命令を実行する



MC_MoveVelocity（速度制御）命令の Vel_InVel が TRUE で、軸 5 の原点復帰完了後に、従軸（軸 5）に設定された MC_CamIn（カム動作開始）命令を実行する

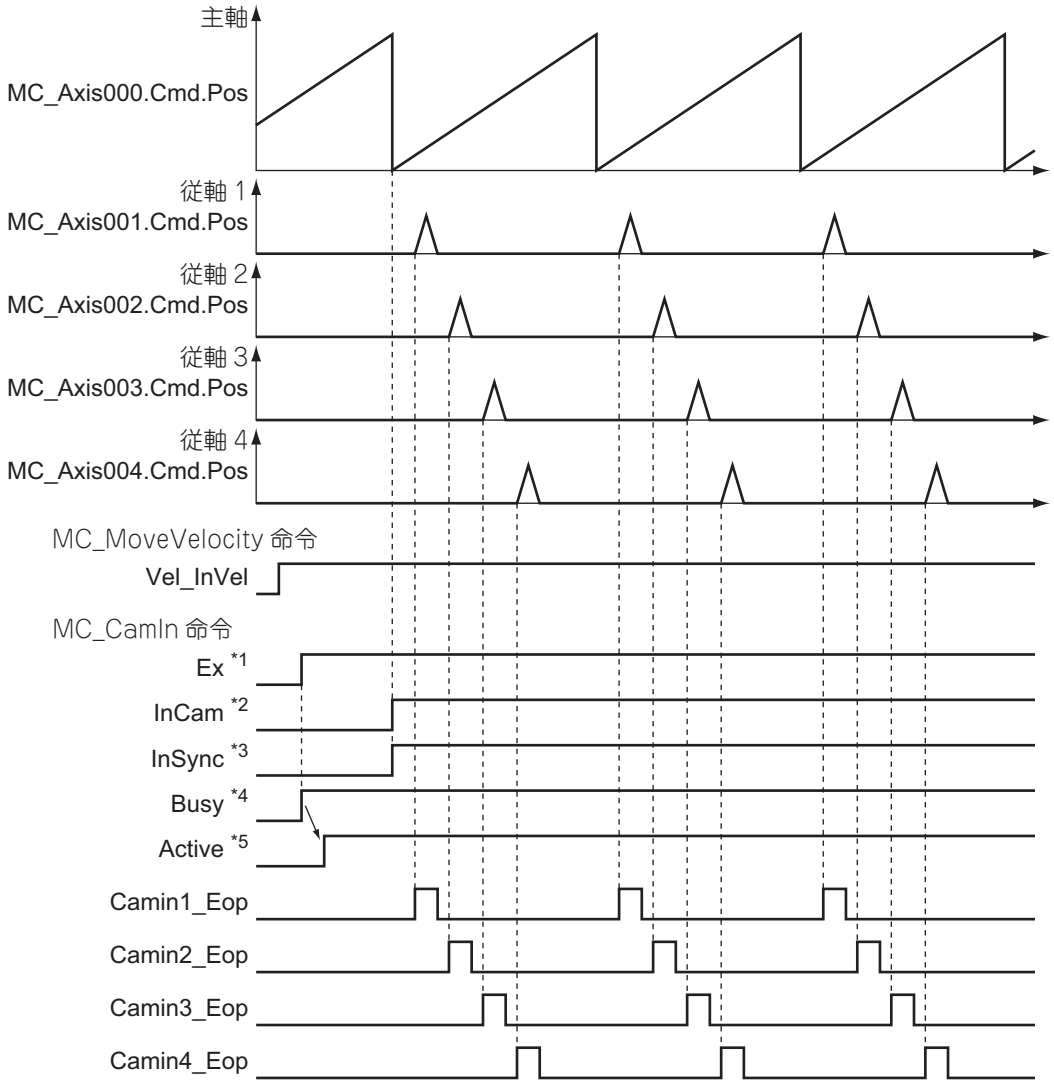


ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
MC_Axis002	_sAXIS_REF	−	従軸である軸 3 の軸変数です。
MC_Axis003	_sAXIS_REF	−	従軸である軸 4 の軸変数です。
MC_Axis004	_sAXIS_REF	−	従軸である軸 5 の軸変数です。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Vel_InVel	BOOL	FALSE	MC_MoveVelocity のインスタンスである VEL の出力「InVelocity」を割り付けた変数です。目標速度に到達すると、この変数が TRUE となります。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	−	カムデータ変数です。MC_CamIn のインスタンス CAMIN1～CAMIN4 の入力「CamTable」にこの変数を指定します。 配列の要素数 ARRAY[0..N]は、カムエディタで設定します。
Camin1_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、MC_CamIn のインスタンス CAMIN1～CAMIN4 が起動します。
Camin2_Ex	BOOL	FALSE	
Camin3_Ex	BOOL	FALSE	
Camin4_Ex	BOOL	FALSE	
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート



*1. Camin1_InCam, Camin2_InCam, Camin3_InCam, Camin4_InCam が同じタイミングです。
*2. Camin1_InSync, Camin2_InSync, Camin3_InSync, Camin4_InSync が同じタイミングです。
この例では、「MasterStartDistance」が「0」なので、主軸が「0°」から「InSync」が TRUE となります。
*3. Camin1_Bsy, Camin2_Bsy, Camin3_Bsy, Camin4_Bsy が同じタイミングです。
*4. Camin1_Act, Camin2_Act, Camin3_Act, Camin4_Act が同じタイミングです。

● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
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;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag := TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
```

```
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 2 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 3 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
AND (MC_Axis002.DrvStatus.Ready=TRUE) THEN
    Pwr3_En:=TRUE;
ELSE
    Pwr3_En:=FALSE;
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 4 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
AND (MC_Axis003.DrvStatus.Ready=TRUE) THEN
    Pwr4_En:=TRUE;
ELSE
    Pwr4_En:=FALSE;
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 5 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
AND (MC_Axis004.DrvStatus.Ready=TRUE) THEN
    Pwr5_En:=TRUE;
ELSE
    Pwr5_En:=FALSE;
END_IF;

//軸 1～軸 5 で軽度フォールトが発生したら、異常時処理 FaultHandler を実行します
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
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;

//軸 1 がサーボ ON 状態で、かつ原点未確定のとき、軸 1 の原点復帰を実行します
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//軸 2 がサーボ ON 状態で、かつ原点未確定のとき、軸 2 の原点復帰を実行します
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//軸 3 がサーボ ON 状態で、かつ原点未確定のとき、軸 3 の原点復帰を実行します
IF (Pwr3_Status=TRUE) AND (MC_Axis002.Details.Homed=FALSE) THEN
    Hm3_Ex:=TRUE;
END_IF;

//軸 4 がサーボ ON 状態で、かつ原点未確定のとき、軸 4 の原点復帰を実行します
IF (Pwr4_Status=TRUE) AND (MC_Axis003.Details.Homed=FALSE) THEN
    Hm4_Ex:=TRUE;
END_IF;

//軸 5 がサーボ ON 状態で、かつ原点未確定のとき、軸 5 の原点復帰を実行します
IF (Pwr5_Status=TRUE) AND (MC_Axis004.Details.Homed=FALSE) THEN
    Hm5_Ex:=TRUE;
END_IF;

//軸 1 の原点復帰完了後に、MC_MoveVelocity を実行します
IF Hm1_D=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

//軸 2 の原点確定で、かつ MC_MoveVelocity の Vel_InVel が TRUE のとき、従軸が軸 2 に設定された MC_CamIn を
起動します
IF (Vel_InVel=TRUE) AND (MC_Axis001.Details.Homed=TRUE) THEN
    Camin1_Ex := TRUE;
END_IF;

//軸 3 の原点確定で、かつ MC_MoveVelocity の Vel_InVel が TRUE のとき、従軸が軸 3 に設定された MC_CamIn を
起動します
IF (Vel_InVel=TRUE) AND (MC_Axis002.Details.Homed=TRUE) THEN
    Camin2_Ex := TRUE;
END_IF;

//軸 4 の原点確定で、かつ MC_MoveVelocity の Vel_InVel が TRUE のとき、従軸が軸 4 に設定された MC_CamIn を
起動します
IF (Vel_InVel=TRUE) AND (MC_Axis003.Details.Homed=TRUE) THEN
    Camin3_Ex := TRUE;
END_IF;
```

```
//軸5の原点確定で、かつMC_MoveVelocityのVel_InVelがTRUEのとき、従軸が軸5に設定されたMC_CamInを
起動します
IF (Vel_InVel=TRUE) AND (MC_Axis004.Details.Homed=TRUE) THEN
    Camin4_Ex := TRUE;
END_IF;

//軸1のMC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

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

//軸3のMC_Power
PWR3(
    Axis := MC_Axis002,
    Enable := Pwr3_En,
    Status => Pwr3_Status,
    Busy => Pwr3_Bsy,
    Error => Pwr3_Err,
    ErrorID => Pwr3_ErrID
);

//軸4のMC_Power
PWR4(
    Axis := MC_Axis003,
    Enable := Pwr4_En,
    Status => Pwr4_Status,
    Busy => Pwr4_Bsy,
    Error => Pwr4_Err,
    ErrorID => Pwr4_ErrID
);

//軸5のMC_Power
PWR5(
    Axis := MC_Axis004,
    Enable := Pwr5_En,
```

```
    Status => Pwr5_Status,
    Busy => Pwr5_Bsy,
    Error => Pwr5_Err,
    ErrorID => Pwr5_ErrID
);

//軸 1 の MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

//軸 2 の MC_Home
HM2(
    Axis := MC_Axis001,
    Execute := Hm2_Ex,
    Done => Hm2_D,
    Busy => Hm2_Bsy,
    CommandAborted => Hm2_Ca,
    Error => Hm2_Err,
    ErrorID => Hm2_ErrID
);

//軸 3 の MC_Home
HM3(
    Axis := MC_Axis002,
    Execute := Hm3_Ex,
    Done => Hm3_D,
    Busy => Hm3_Bsy,
    CommandAborted => Hm3_Ca,
    Error => Hm3_Err,
    ErrorID => Hm3_ErrID
);

//軸 4 の MC_Home
HM4(
    Axis := MC_Axis003,
    Execute := Hm4_Ex,
    Done => Hm4_D,
    Busy => Hm4_Bsy,
    CommandAborted => Hm4_Ca,
    Error => Hm4_Err,
    ErrorID => Hm4_ErrID
);

//軸 5 の 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
);

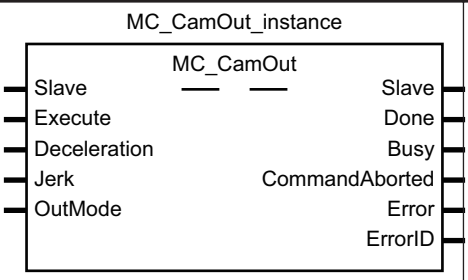
CAMIN3(
    Master := MC_Axis000,
    Slave := MC_Axis003,
    CamTable := CamProfile0,
    Execute := Camin3_Ex,
    Periodic := Camin3_Em,
    StartMode := Camin3_Sm,
    StartPosition := Camin3_Sp,
    MasterStartDistance := Camin3_Msd,
    MasterScaling := Camin3_Ms,
    SlaveScaling := Camin3_Ss,
    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

入力パラメータで指定した軸のカム動作を終了します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_CamOut	カム動作解除	FB		MC_CamOut_instance (Slave :=「パラメータ」, Execute :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, OutMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Deceleration	減速度	LREAL	正値、または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*1
Jerk (Reserved)	ジャーク	LREAL	0	0	(Reserved)
OutMode (Reserved)	同期解除モード選択	_eMC_OUT_MODE	0: _mcStop	0 *2	(Reserved)

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。

出力変数	名称	データ型	有効範囲	内容
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	速度「0」に到達したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Slave	従軸	_sAXIS_REF	−	従軸を指定します。 *1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

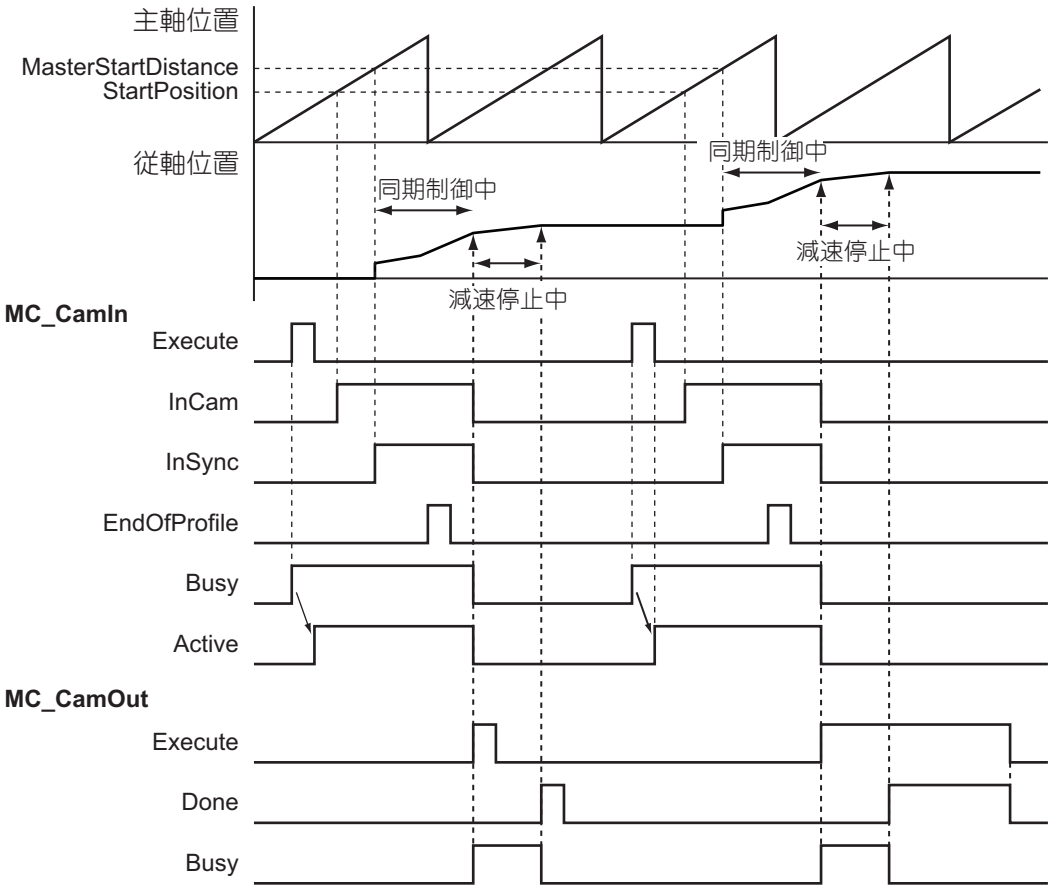
機能説明

- 本命令は、従軸のカム動作を解除します。
- 「Execute」（起動）を TRUE にすると「Deceleration」（減速度）に従って速度「0」に向かって減速します。
- 指令速度が「0」になったら、完了となります。
- カム動作をしていない軸に対して本命令を起動した場合、異常となります。

使用上の注意

- カムデータ変数はグローバル変数であり、複数のタスクからカムデータ変数の値を参照や変更をすることができます。複数のタスクからカムデータ変数の値を変更する場合、複数のタスクによる書き換え処理で競合が発生しないようにプログラムを作成してください。
- 「グローバル変数のタスク間排他制御」機能を用いてカムデータ変数を排他制御する場合、排他対象のカムデータ変数を用いてモーション命令を起動しないでください。軽度フォールトレベルの異常「カムテーブル指定不正（エラーコード: 5439 Hex）」が発生します。

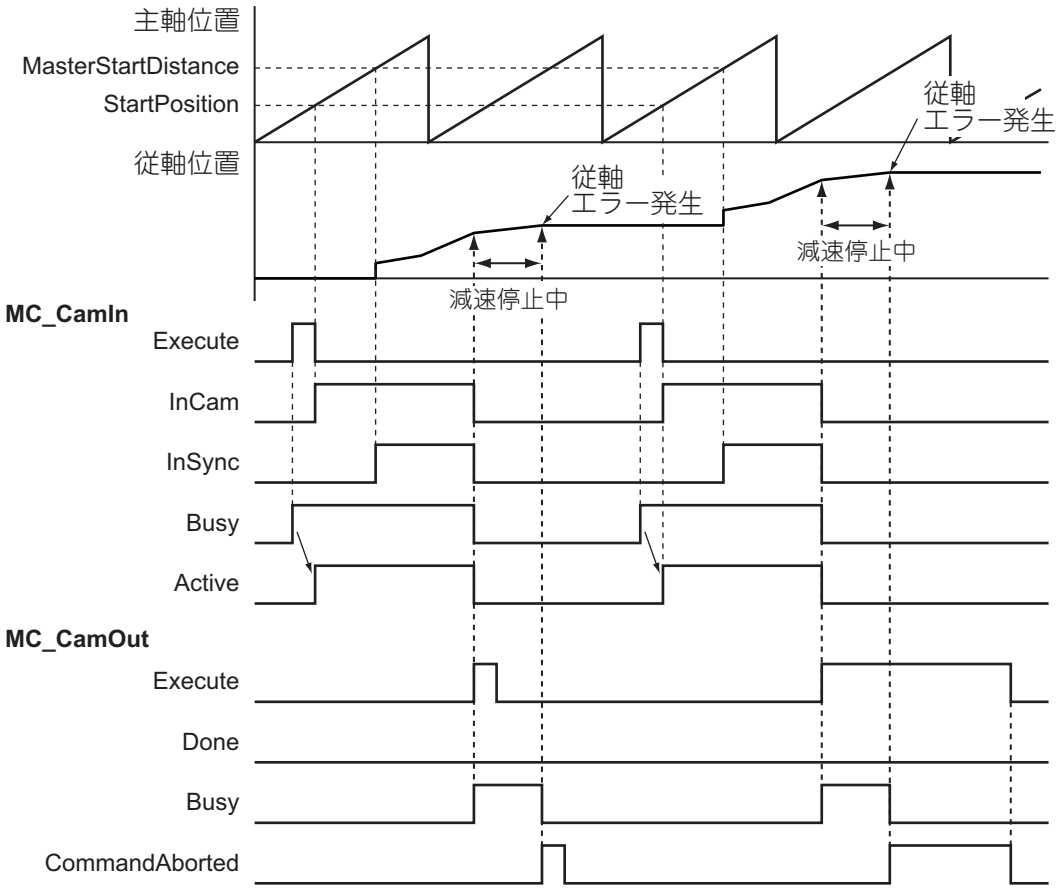
タイミングチャート



命令の中止

本命令実行中に従軸で軸異常が発生した場合、「CommandAborted」（実行中断）が TRUE となり、「Busy」（動作中）は FALSE となります。
また、本命令の「Deceleration」（減速度）に従って軸は減速します。

軸異常については、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

● 他命令実行中の本命令起動

MC_CamIn（カム動作開始）命令実行中に、本命令を起動した場合、MC_CamIn（カム動作開始）命令を「CommandAborted」（実行中断）し、本命令は「Busy」（実行中）となります。
MC_CamIn（カム動作開始）命令実行中以外で、本命令を起動した場合は、「Error」（エラー）となります。

● 本命令実行中の他命令起動

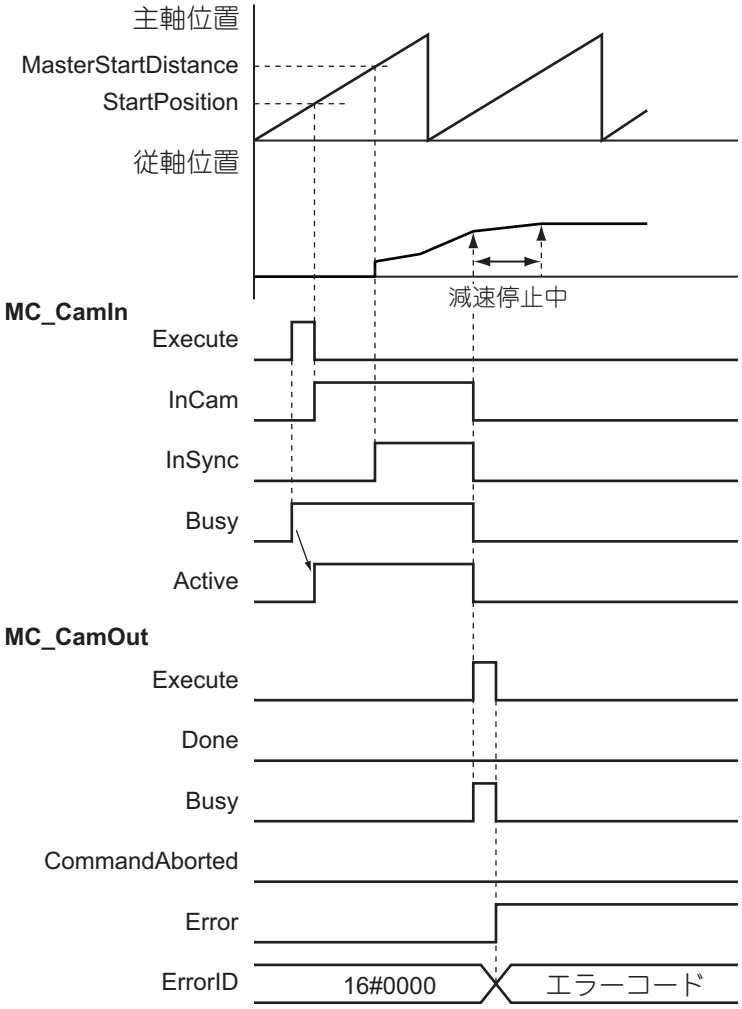
本命令に対するモーション命令多重起動を行う場合、従軸を指定してください。
本命令に対するモーション命令多重起動を行う場合、［アボーティング］、［バッファード］が可能です。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。

「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート

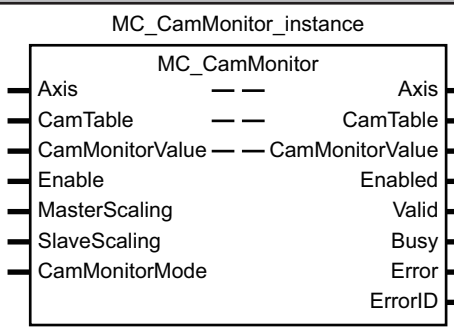


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_CamMonitor

カム動作に関する情報のモニタを行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_CamMonitor	カムモニタ	FB		MC_CamMonitor_instance(Axis :=「パラメータ」, CamTable :=「パラメータ」, CamMonitorValue :=「パラメータ」, Enable :=「パラメータ」, MasterScaling :=「パラメータ」, SlaveScaling :=「パラメータ」, CamMonitorMode :=「パラメータ」, Enabled =>「パラメータ」, Valid =>「パラメータ」, Busy =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

バージョン関連情報

本命令は、以下に示すユニットバージョンの CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。

CPU ユニット	使用可能なユニットバージョン
形 NX102 CPU ユニット	Ver.1.32 以降
形 NX1P2 CPU ユニット	Ver.1.21 以降
NJ シリーズ CPU ユニット	Ver.1.21 以降

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE の間、命令を実行します。
MasterScaling	主軸係数	LREAL	正値(>0.0)	1.0	指定されたスケールで、主軸の位相を拡大／縮小します。
SlaveScaling	従軸係数	LREAL	正値(>0.0)	1.0	指定されたスケールで、従軸の変位を拡大／縮小します。
CamMonitorMode	カムモニタモード選択	_eMC_CAM_MONITOR_MODE	0: _mcCalcCamDistanceDiff	0*1	モニタするカム動作の情報を指定します。 0: 変位偏差計算

*1. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Enabled	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。
Valid	カムモニタ値有効	BOOL	TRUE, FALSE	「CamMonitorValue」（カムモニタ値）が有効な値であるときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Enabled	「Enable」が TRUE となったとき	• 「Enable」が FALSE の場合、1 周期後 • 「Error」が TRUE となったとき。
Valid	「CamMonitorValue」（カムモニタ値）に出力された値が有効なとき	• 「Enable」が FALSE の場合、1 周期後 • 「Error」が TRUE となったとき。 • 「Valid」が FALSE になる条件を満たしたとき。*1
Busy	「Enable」の立ち上がり	• 「Enable」が FALSE となったとき。 • 「Error」が TRUE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

*1. 「Valid」が FALSE になる条件は、「使用上の注意 (P.3-229)」を参照してください。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1
CamTable	カムテーブル	ARRAY[0..N] OF_sMC_CAM_REF	−	カムデータ構造体_sMC_CAM_REF 型の配列変数をカムテーブルとして指定します。*2
CamMonitorValue	カムモニタ値	_sMC_CAM_MONITOR_DISTANCEDIFF	−	カム動作の情報を出力します。*3

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. 配列要素“N”は Sysmac Studio により自動的に設定されます。Sysmac Studio のカムエディタで作成したカムデータ変数を指定してください。
*3. モニタするカム動作の情報は、「CamMonitorMode」（カムモニタモード選択）で指定します。

● _sMC_CAM_MONITOR_DISTANCEDIFF（カムモニタ変位偏差）

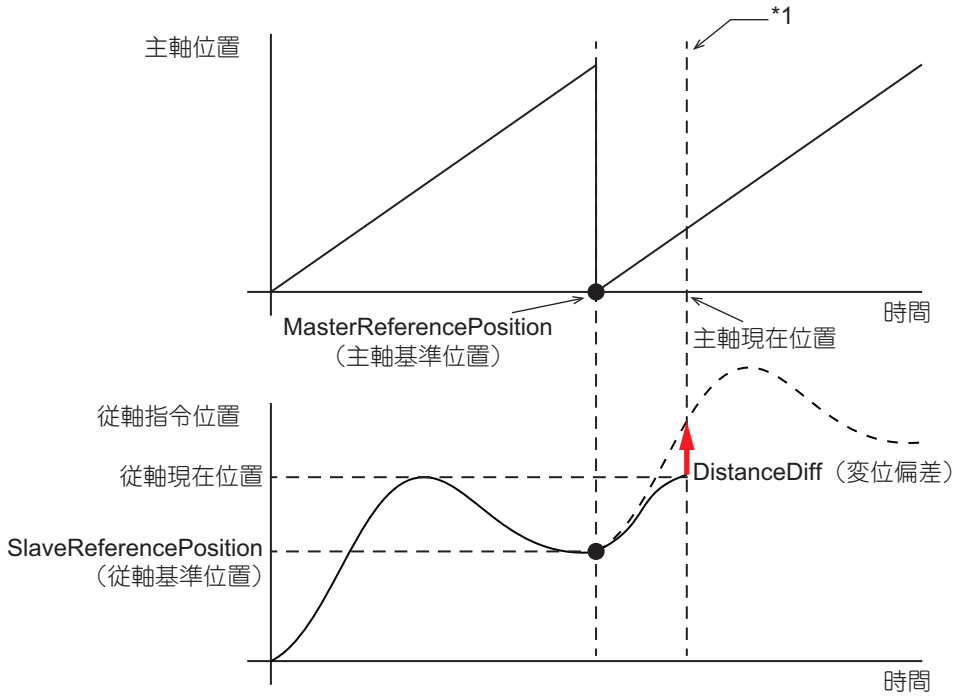
変数	名称	データ型	内容
CamProfileData	カムプロファイルデータ	_sMC_CAM_PROFILE_DATA	位相や変位などカム動作の基本的な情報を出力します。

変数	名称	データ型	内容
DistanceDiff	変位偏差	LREAL	軸の指令位置と「Distance」（従軸変位）との差を出力します。

● _sMC_CAM_PROFILE_DATA（カムプロファイルデータ）

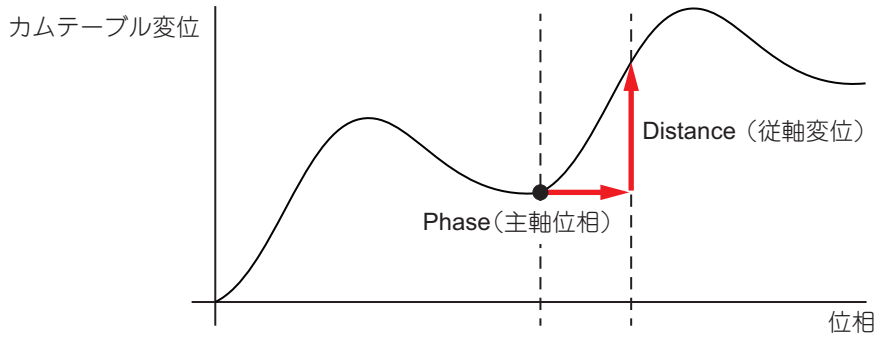
変数	名称	データ型	内容
Phase	主軸位相	LREAL	現在の位相を出力します。
Distance	従軸変位	LREAL	現在の変位を出力します。
MasterReferencePosition	主軸基準位置	LREAL	現在の位相の基準（位相=0.0）となる主軸の位置
SlaveReferencePosition	従軸基準位置	LREAL	現在の変位の基準（変位=0.0）となる従軸の位置
PhaseShift	位相補正量	LREAL	カム動作中に MC_Phasing（主軸相対値位相補正）命令によって補正された位相補正量
OffsetPosition	位置オフセット	LREAL	カム動作中に MC_SyncOffsetPosition（サイクリック同期位置オフセット補正）命令または MC_OffsetPosition（位置オフセット補正）命令によって補正された位置オフセット

変位偏差、主軸基準位置、従軸基準位置の関係は、以下のようになります。



*1. この時点での値がモニタされます。

また、主軸位相と従軸変位の関係は、以下のようになります。



機能説明

本命令は、「CamTable」（カムテーブル）と「CamMonitorMode」（カムモニタモード選択）で指定したカム動作の情報をモニタします。

直前に実行した MC_CamIn（カム動作開始）命令の主軸の位相に基づき、本命令で指定した「CamTable」（カムテーブル）と「CamMonitorMode」（カムモニタモード選択）により、カム動作の情報を「CamMonitorValue」（カムモニタ値）に出力します。
MC_CamIn（カム動作開始）命令の起動後、MC_CamIn（カム動作開始）命令の「InSync」（同期中）が TRUE になると、「Valid」（カムモニタ値有効）が TRUE となり、「CamMonitorValue」（カムモニタ値）を更新します。

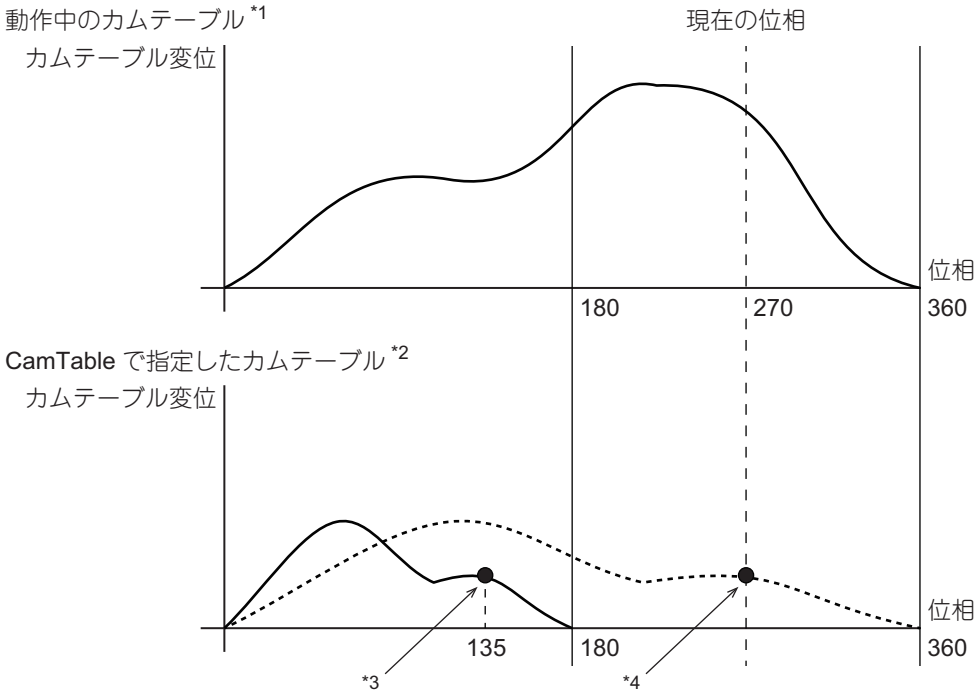
MC_CamIn（カム動作開始）命令の中断後、MC_CamIn（カム動作開始）命令の「CommandAborted」（実行中断）が TRUE となっても、本命令は実行可能です。

本命令は、以下のような場合に使用します。
- 現在動作中のカム動作が、従軸の異常などによって中断した場合に、現在の主軸の位相にもとづいて MC_CamIn（カム動作開始）命令の再開条件をモニタする。
- 段取り替えなどでカムテーブルを変更する場合に、現在の主軸の位相にもとづいて、カムテーブル変更後の MC_CamIn（カム動作開始）命令の開始条件をモニタする。

MC_CamIn（カム動作開始）命令は、「MasterOffset」（主軸オフセット）と「SlaveOffset」（従軸オフセット）を設定することで、カムテーブルの途中から起動することができます。
本命令でモニタされた「Phase」（主軸位相）、「Distance」（従軸変位）をもとに、後に起動する MC_CamIn（カム動作開始）命令の各オフセットを設定してください。
各オフセットの設定方法については、「MC_CamIn (P.3-160)」を参照してください。

CamTable（カムテーブル）

「CamTable」（カムテーブル）を指定することができます。
動作中のカムテーブルの位相の終点と、指定する「CamTable」（カムテーブル）の位相の終点が異なる場合、下記のように動作中のカムテーブルと同じ位相の終点となるようにスケーリングし、位相を計算します。
得られた位相から、スケーリングの影響を除いた値を位相として、「CamMonitorValue」（カムモニタ値）に出力します。



*1. 位相の終点は、360
*2. 位相の終点は、180
*3. スケーリングの影響を除いて「Phase」（主軸位相）を出力
*4. スケーリングしたカムテーブルと現在の位相を元に計算

「CamMonitorValue」（カムモニタ値）に出力される位相は、以下のようになります。

$$\text{Phase} = \frac{\text{現在の位相}}{\text{動作中のカムテーブルの位相の終点}} \times \text{CamTable に指定したカムテーブルの位相の終点}$$

$$= \frac{270}{360} \times 180$$

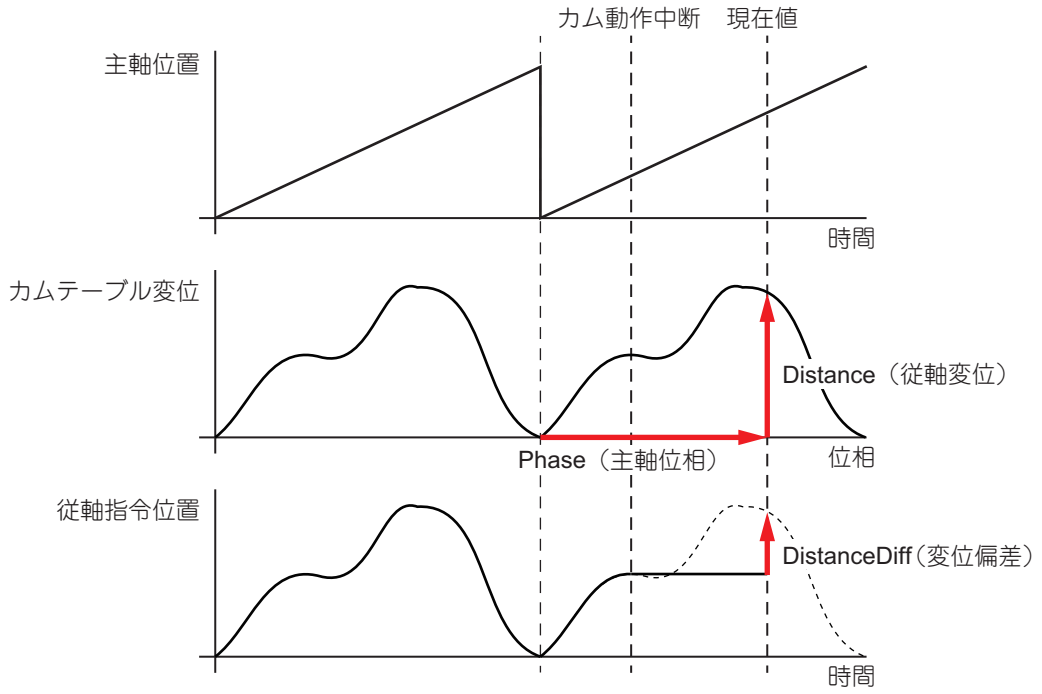
$$= 135$$

CamMonitorMode（カムモニタモード選択）

● 変位偏差計算

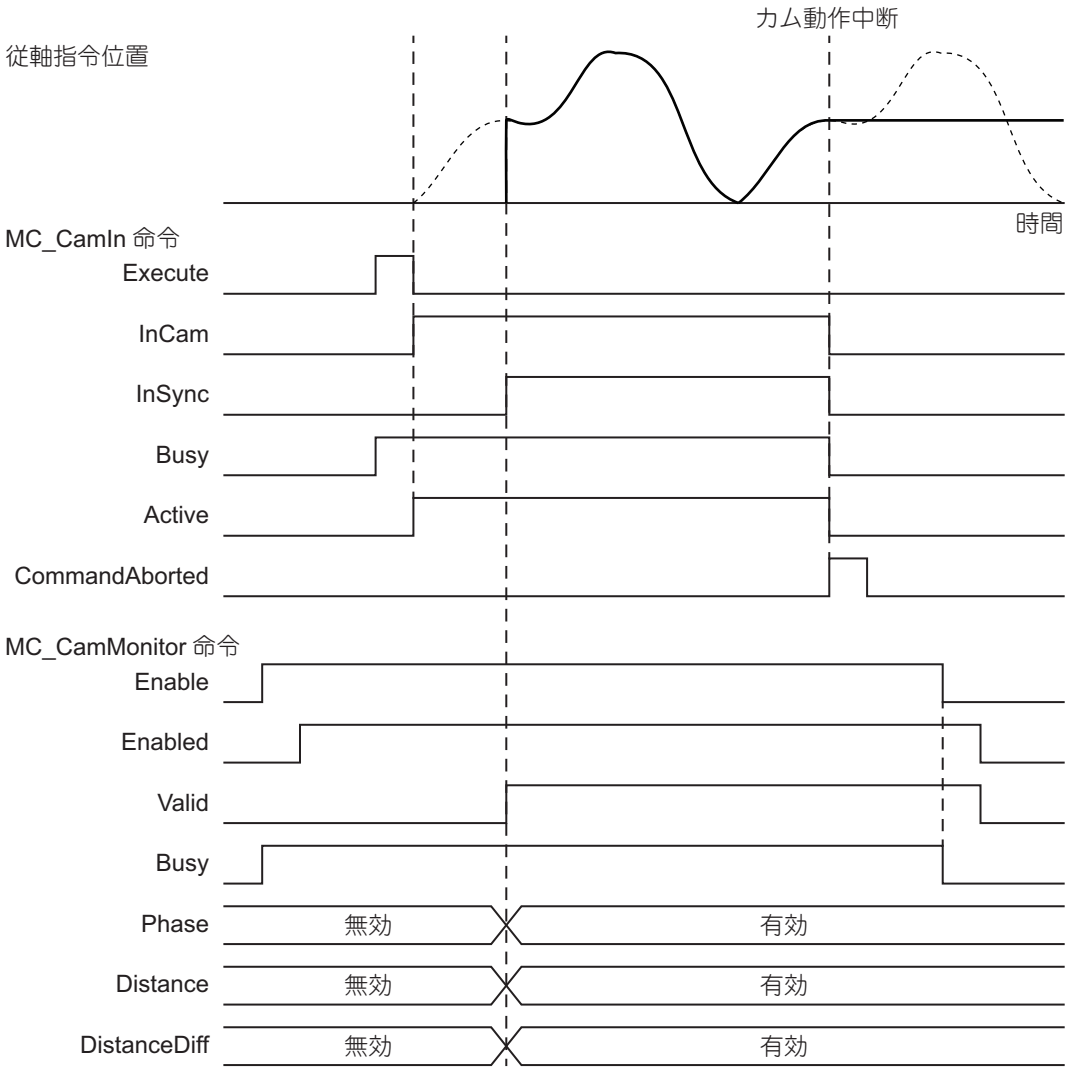
「CamMonitorMode」（カムモニタモード選択）で［0: _mcCalcCamDistanceDiff］を選択した場合は、以下のようになります。

主軸の位置から計算されるカムプロファイル曲線の「Phase」（主軸位相）と、「Phase」から得られる「Distance」（従軸変位）、および従軸指令位置の「DistanceDiff」（変位偏差）をモニタします。



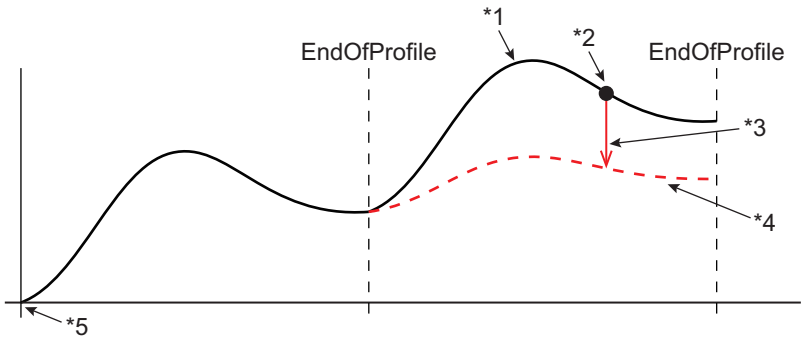
モニタした「DistanceDiff」（変位偏差）を利用して、従軸指令現在位置をカムプロファイル曲線上に移動させることができます。
さらに、モニタした「Phase」（主軸位相）と「Distance」（従軸変位）の正負逆転させた値を、次に起動する MC_CamIn（カム動作開始）命令の「MasterOffset」（主軸オフセット）と「SlaveOffset」（従軸オフセット）に指定することで、カムプロファイル曲線の途中から起動することができます。

以下に使用例として、MC_CamIn（カム動作開始）命令を実行中に MC_CamMonitor（カムモニタ）命令を起動し、その後 MC_CamIn（カム動作開始）命令を中断した場合のタイミングチャートを示します。


カム動作中断
従軸指令位置
時間
MC_CamIn 命令
Execute
InCam
InSync
Busy
Active
CommandAborted
MC_CamMonitor 命令
Enable
Enabled
Valid
Busy
Phase
Distance
DistanceDiff
無効
有効
無効
有効
無効
有効

使用上の注意

変位偏差計算は、カム動作を開始した位置ではなく、「MasterReferencePosition」（主軸基準位置）と「SlaveReferencePosition」（従軸基準位置）を基準にして、変位偏差を計算します。終点の変位が 0 にならないような送りカムの場合、以下のようになります。



*1. 動作中のカムテーブル
*2. 従軸の指令現在位置
*3. 計算される変位偏差
*4.「CamTable」（カムテーブル）で指定したカムテーブル
*5. カム動作を開始した位置

また、変位偏差計算でモニタされる位相は、直前の周期で計算した位相補正量に基づいて計算します。
そのため、MC_Phasing（主軸相対値位相補正）命令の実行中は、位相補正量が 1 制御周期ずれるので、MC_CamIn（カム動作開始）命令で計算される値とは異なります。

モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

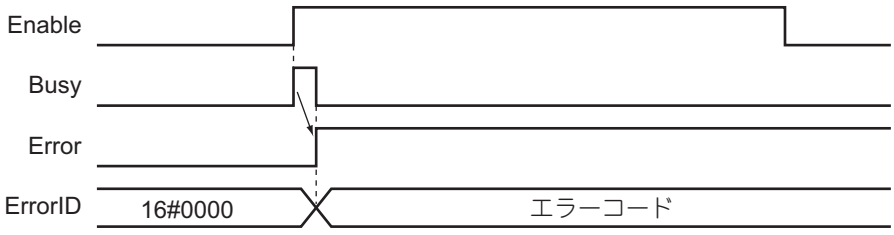
モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

使用上の注意

つぎの場合、「Valid」（カムモニタ値有効）が FALSE となり、「CamMonitorValue」（カムモニタ値）は更新されません。

- 「Axis」（軸）で指定した軸の状態が、同期制御命令の起動により、Synchronized（同期動作中）に遷移した場合。
- 「CamTable」（カムテーブル）で指定したカムプロファイル曲線に対して、MC_GenerateCamTable（カムテーブル生成）命令が実行中の場合。
- 「Axis」（軸）で指定した軸に対して、MC_Home（原点復帰）命令、MC_HomeWithParameter（パラメータ指定原点復帰）命令、MC_SetPosition（現在位置変更）命令を実行した場合。
- 「Axis」（軸）で指定した軸の［位置カウント設定］の［エンコーダ種別］を、［絶対値エンコーダ (ABS)］に設定し、MC_Power（運転可）命令により、「Status」（運転可）が TRUE となり、原点が確定した場合。
- 「Axis」（軸）で指定した軸、またはカム動作で同期していた主軸のいずれかに対して、MC_ChangeAxisUse（軸使用変更）命令を実行した場合。
- 「Axis」（軸）で指定した軸、またはカム動作で同期していた主軸のいずれかが、以下の状態になった場合。
  a）EtherCAT のプロセスデータ通信が確立していない場合。
  b）軽度フォールト「スレーブ通信異常(84400000Hex)」発生時。
  c）スレーブ離脱した時。
- 位相が「EndOfProfile」（カムサイクル完了）に到達し、カム動作が完了した後に、再度位相がカムプロファイル曲線内に入っても、「Valid」は TRUE にはなりません。
- CPU ユニットを運転モードからプログラムモードに切り替えた場合。
- MC 試運転を開始した場合。

MC_GearIn

主軸と従軸間のギア比を設定し、ギア動作を行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GearIn	ギア動作開始	FB	MC_GearIn_instance MC_GearIn Master — — Master Slave — — Slave Execute — InGear RatioNumerator — Busy RatioDenominator — Active ReferenceType — CommandAborted Acceleration — Error Deceleration — ErrorID Jerk BufferMode	MC_GearIn_instance (Master :=「パラメータ」, Slave :=「パラメータ」, Execute :=「パラメータ」, RatioNumerator :=「パラメータ」, RatioDenominator :=「パラメータ」, ReferenceType :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, BufferMode :=「パラメータ」, InGear =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
RatioNumerator	ギア比分子	DINT *1	正数または負数 *1	10000	主軸と従軸間の電子ギアの分子を指定します。
RatioDenominator	ギア比分母	UDINT *2	正数	10000	主軸と従軸間の電子ギアの分母を指定します。
ReferenceType *3	位置種別選択	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *4	位置の種別を指定します。 0: 指令位置（直前タスク周期 *5 での計算値） 1: フィードバック位置（同一タスク周期 *5 での取得値） 2: 指令位置（同一タスク周期 *5 での計算値）
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は［指令単位／s^2］となります。*6
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*6
Jerk (Reserved)	ジャーク	LREAL	0	0	(Reserved)

入力変数	名称	データ型	有効範囲	初期値	内容
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *4	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード

*1. ユニットバージョン Ver.1.02 以降の CPU ユニットと Sysmac Studio Ver.1.03 以降の組み合わせで使用可能です。それ以前の組み合わせでは、データ型は「UINT」、有効範囲「正数」となります。
*2. ユニットバージョン Ver.1.02 以降の CPU ユニットと Sysmac Studio Ver.1.03 以降の組み合わせで使用可能です。それ以前の組み合わせでは、データ型は「UINT」となります。
*3. _mcLatestCommand を使用する場合は、次の関係となるように主軸、従軸を使用しなければなりません。
「Master」（主軸）に設定されているモーション制御システム変数の軸番号 ＜ 「Slave」（従軸）に設定されているモーション制御システム変数の軸番号
*4. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*5. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。
*6. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

出力変数

出力変数	名称	データ型	有効範囲	内容
InGear	ギア比到達	BOOL	TRUE, FALSE	従軸が、目標速度に到達したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
InGear	以下の関係が成り立つとき。 • 加速中: 従軸の速度≧主軸の速度×ギア比 • 減速中: 従軸の速度≦主軸の速度×ギア比	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Busy	「Execute」の立ち上がり	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。 • MC_GearOut 命令が起動されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後

変数	TRUE となるタイミング	FALSE となるタイミング
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Master	主軸	_sAXIS_REF	−	主軸を指定します。*1
Slave	従軸	_sAXIS_REF	−	従軸を指定します。*1

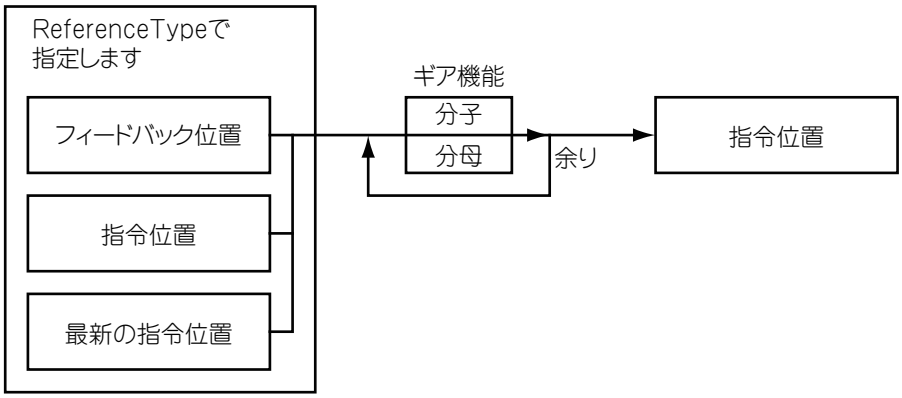
1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

使用上の注意

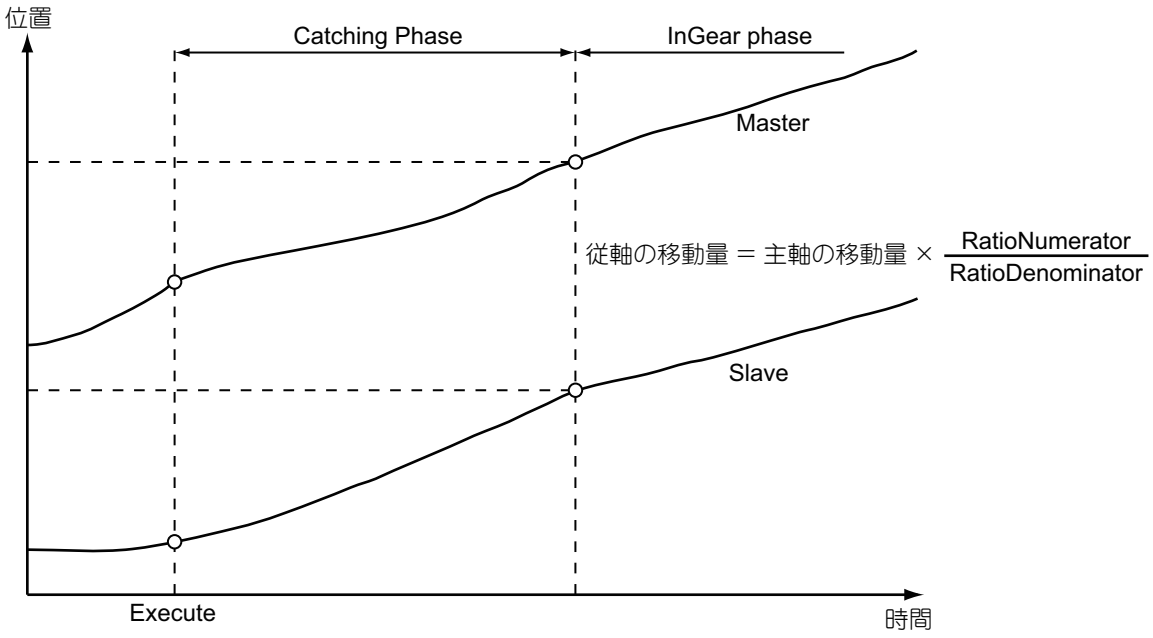
主軸と従軸に同じ軸を指定すると、軽度フォールトレベルの「主軸従軸同一（エラーコード: 5436 Hex)」異常となります。

機能説明

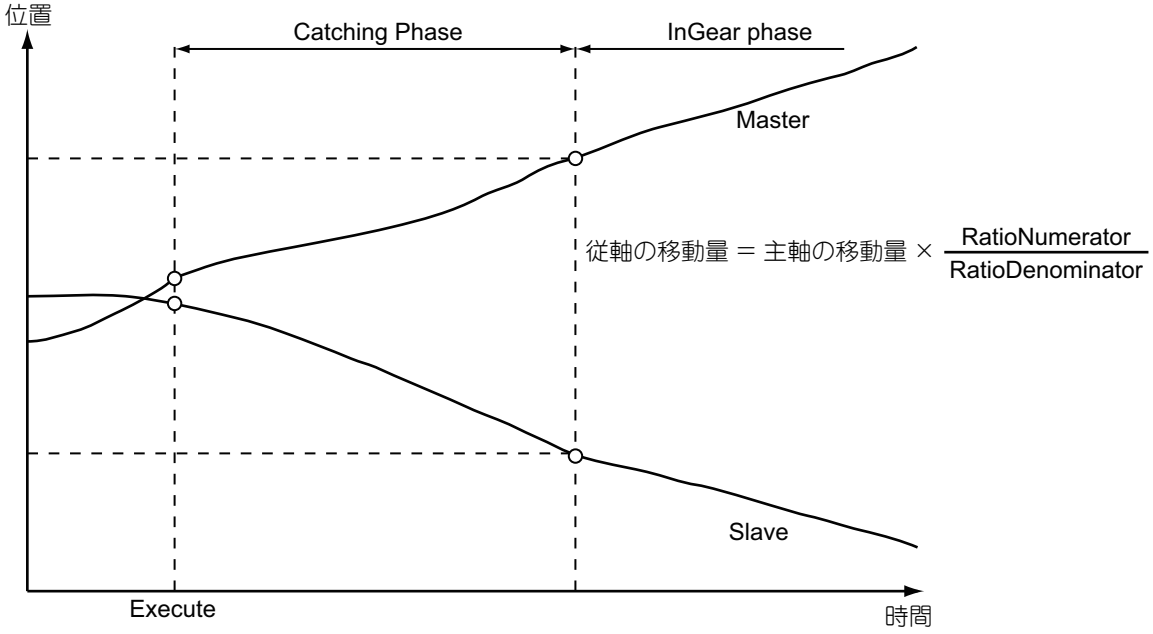
- 動作対象の軸を「Slave」(従軸) で指定し、「RatioNumerator」(ギア比分子)、「RatioDenominator」(ギア比分母)、「ReferenceType」(位置種別選択)、「Acceleration」(加速度)、「Deceleration」(減速度) を指定してギア動作を行います。
- 「Master」(主軸) には指令位置とフィードバック位置、最新の指令位置を指定することができます。



- 動作開始後、「Slave」(従軸) は「Master」(主軸) の速度にギア比を乗じた速度を目標速度として、加減速動作を行います。
- 目標速度に到達するまでを Catching phase（追いかけ中）とし、到達後を InGear phase（ギア同期中）と呼びます。
- ギア比が正数の場合、「Slave」(従軸) は「Master」(主軸) の同方向に移動します。



・ギア比が負数の場合、「Slave」（従軸）は「Master」（主軸）の逆方向に移動します。



・「Execute」（起動）の立ち上がりで、電子ギアの動作を開始します。

使用上の注意

ユニットバージョン Ver.1.09 以前の CPU ユニットで本命令を使用する場合は、「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行しないでください。「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行すると、「Slave」（従軸）が急激な追従をすることがあり、危険です。
「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を使用する場合は、「Master」（主軸）と「Slave」（従軸）の関係をいったん解除してから実行してください。
主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。

● ReferenceType（位置種別選択）

位置の種類を、以下から選択できます。

- _mcCommand: 指令位置（直前タスク周期での計算値）
  現周期に対して、直前のタスク周期で計算した主軸の指令位置を使用します。
  従軸の指令位置を計算する直前の定周期タスクで、計算した主軸の指令位置を使用します。
- _mcFeedback: 同一タスク周期での取得値
  同一タスク周期で取得した主軸のフィードバック位置を使用します。
- _mcLatestCommand: 指令位置（同一タスク周期での計算値）
  同一タスク周期で計算した主軸の指令位置を使用します。
  _mcCommand よりも新しい情報を使用することができます。ただし、主軸番号は従軸番号より若い番号を設定しなければなりません。
  従軸が、主軸より若い軸番号の場合、本命令の「Error」（エラー）が TRUE となります。「ErrorID」（エラーコード）には、「主軸／従軸 軸番号非昇順（エラーコード: 5438 Hex）」が出力されます。



使用上の注意

ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。同様に定周期タスクは、プライマリ定周期タスクまたは定周期タスク（実行優先度 5）を表します。



参考

同一タスク周期で計算した指令位置を選択した方が、直前タスク周期での計算値を選択した場合より、同期精度は向上します。
ただし、[「Master」（主軸）に設定されているモーション制御システム変数の軸番号 ＜「Slave」（従軸）に設定されているモーション制御システム変数の軸番号] となるように主軸、従軸を使用しなければなりません。

● 軸種別と位置種別の関係

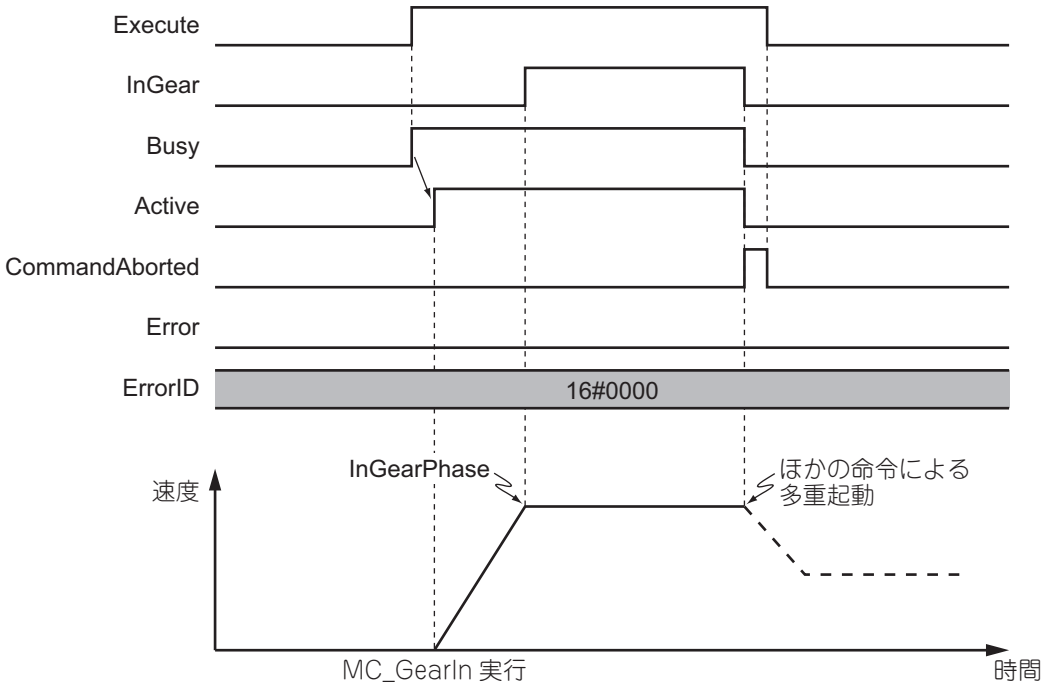
監視可能な軸種別と監視する位置種別の関係は以下のようになります。

軸種別	ReferecneType	
	_mcCommand、または _mcLatestCommand	_mcFeedback
サーボ軸	○	○
エンコーダ軸	× *1	○
仮想サーボ軸	○	○
仮想エンコーダ軸	× *1	○

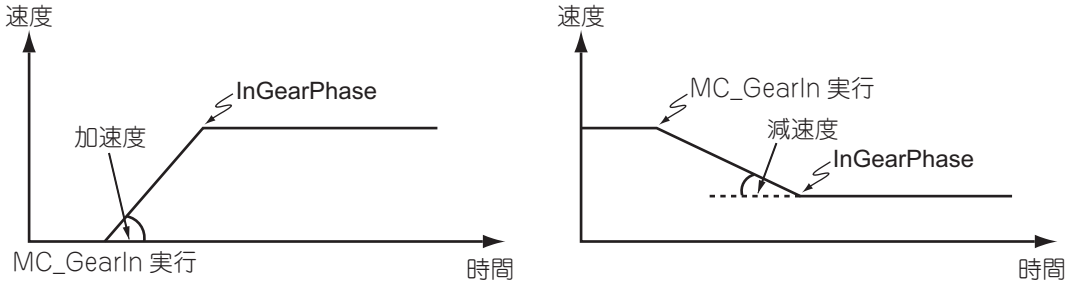
*1. 命令起動時に、「位置種別選択範囲外（エラーコード: 5430 Hex）」が発生します。

タイミングチャート

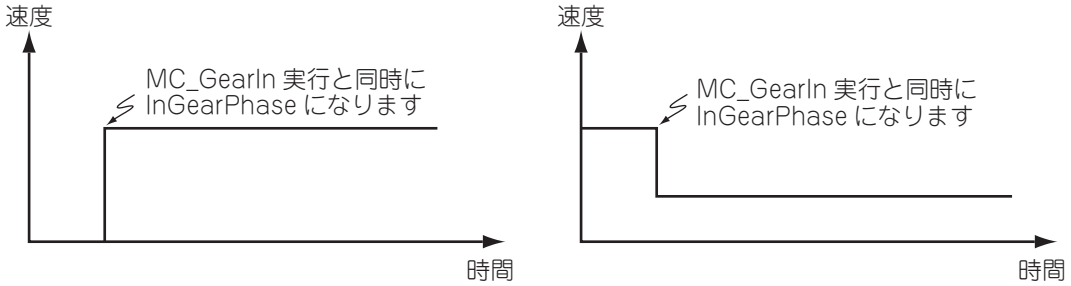
- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- 目標速度に到達すると、「InGear」（ギア比到達）が TRUE となります。
- ほかの命令により、本命令が中断された場合は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）、「InGear」（ギア比到達）が FALSE となります。
- 電子ギア動作を途中で終了させる場合には、MC_GearOut（ギア動作解除）命令、または MC_Stop（強制停止）命令を使用します。



入力変数に、「Acceleration」（加速度）、「Deceleration」（減速度）を指定することができます。



「Acceleration」（加速度）、または「Deceleration」（減速度）を「0」に指定して起動した場合は、加減速せずに目標速度に到達します。

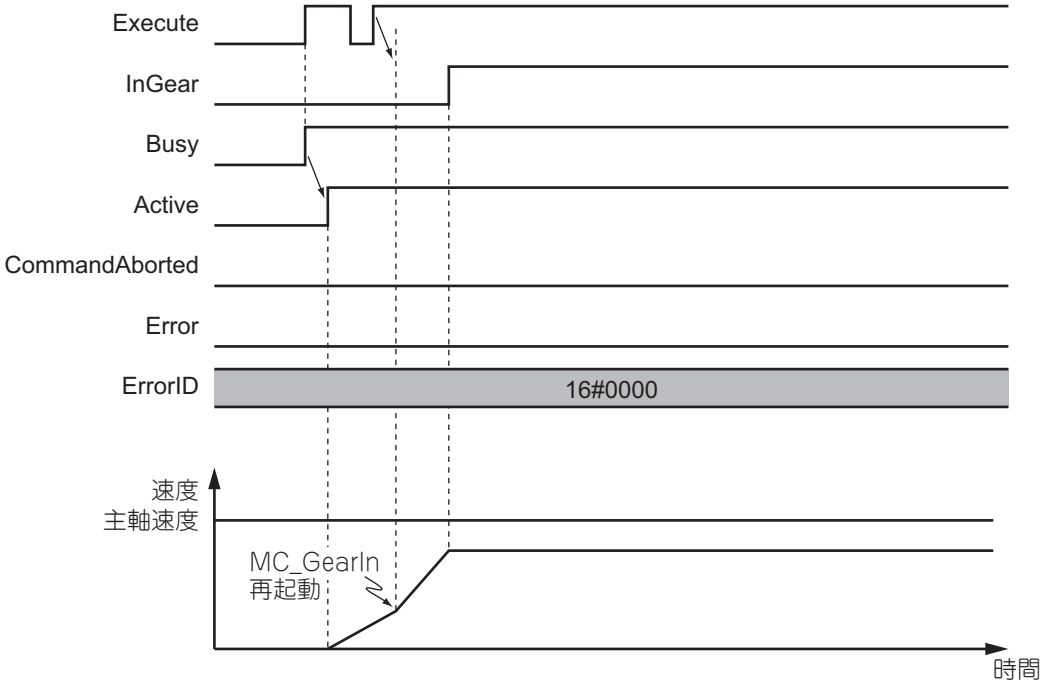


モーション命令再起動

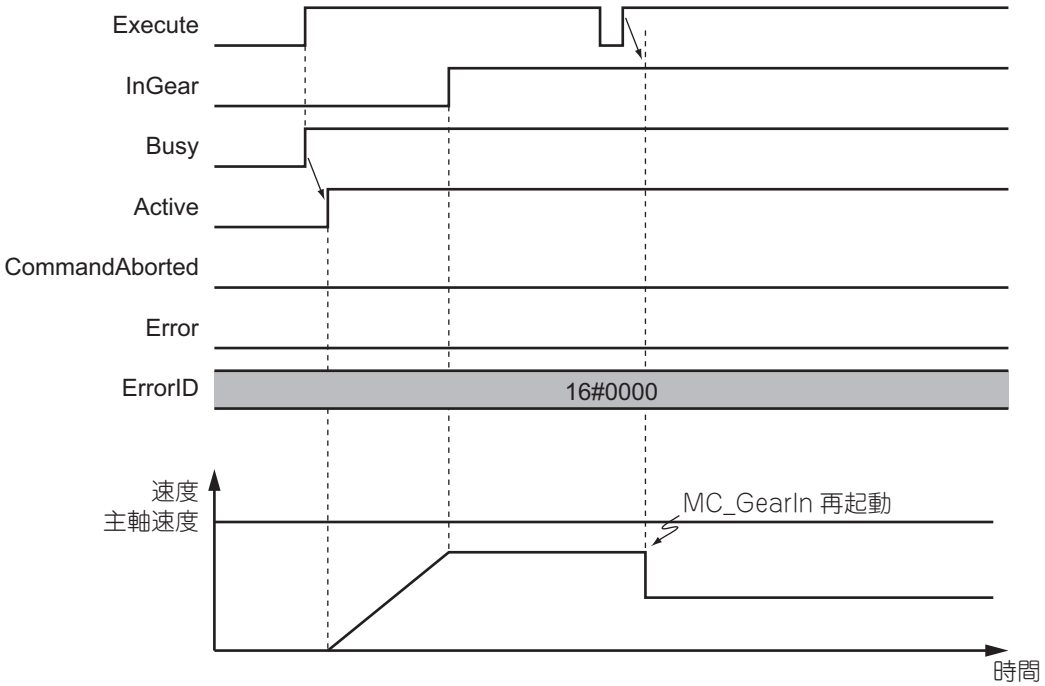
位置決め動作中に入力変数を変更し「Execute」（起動）を再度 TRUE にすることにより、本命令の動作を変更することができます。
モーション命令再起動で変更可能な入力変数は、「RatioNumerator」（ギア比分子）、「RatioDenominator」（ギア比分母）、「Acceleration」（加速度）、「Deceleration」（減速度）です。

モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

参考として、Catching phase（追いかけ中）で再起動し、「Acceleration」（加速度）を変更した場合のタイミングチャートを下記に示します。



InGear phase（ギア同期中）で再起動し、「RatioNumerator」（ギア比分子）、「RatioDenominator」（ギア比分母）を変更した場合のタイミングチャートを下記に示します。
動作としては、「Acceleration」（加速度）と「Deceleration」（減速度）がともに「0」に設定されている場合と同じ動作となります。



モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して、本命令を起動することにより、本命令への切り替えや本命令のバッファリングを行うことができます。
各軸に 1 つまでバッファリングすることが可能です。
多重起動時の本命令の動作を、「BufferMode」（バッファモード選択）で指定します。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。
バッファード	現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。

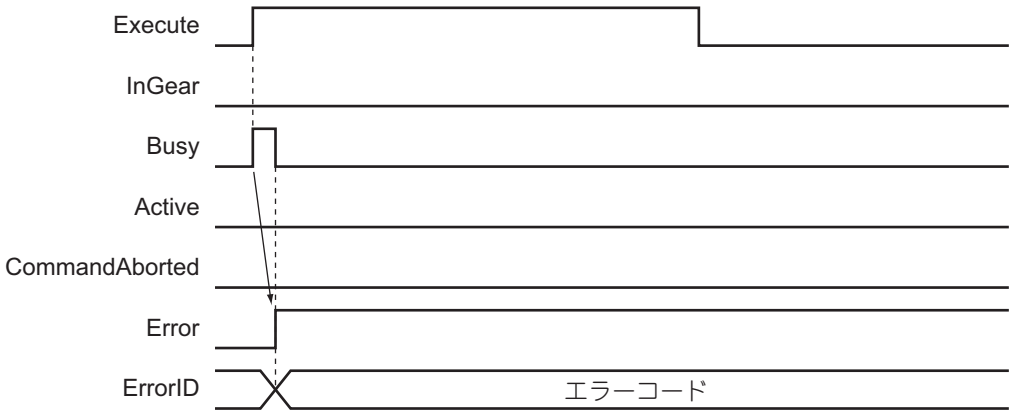
「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

従軸に対し、［アボーティング］による命令多重起動が可能です。
この場合、ギア動作を中止し、多重起動された動作を開始します。
［アボーティング］以外による命令多重起動はできません。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。



参考

- 同期制御中は、主軸の異常状態には影響を受けません。主軸の異常状態が解除され、主軸が正常動作すると、従軸は電子ギア動作を継続します。
- 本命令起動時および本命令実行中に、従軸に異常が発生した場合は、主軸に影響を与えることはありません。

● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、ギア比に比例して動作するサンプルプログラムを掲載します。



参考

予約(Reserved)扱いの入力変数には初期値以外の値を指定できません。
この例ではパラメータを指定しません。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

軸種別

軸	軸種別
軸 1	サーボ軸（主軸）
軸 2	サーボ軸（従軸）
軸 3	サーボ軸（従軸）

カウンタモード

軸	カウンタモード
軸 1	ロータリモード
軸 2	ロータリモード
軸 3	ロータリモード

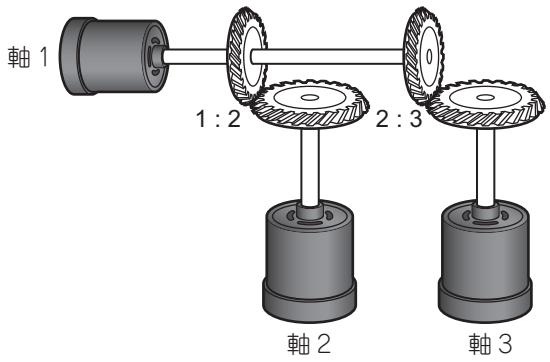
リングカウンタ

軸	上限値	下限値
軸 1	360	0
軸 2	360	0
軸 3	360	0

座標単位選択

軸	座標単位
軸 1	mm
軸 2	mm
軸 3	mm

動作例



1 **主軸の開始**
主軸（軸 1）は実サーボ軸であり速度制御で動作します。

2 **従軸の起動**
主軸のフィードバック速度が目標速度に到達したら、主軸のフィードバック位置に対して従軸（軸 2）をギア比 1:2 で、従軸（軸 3）をギア比 2:3 でギア動作させます。

3 **従軸の停止**
主軸のフィードバック位置(「MC_Axis000.Act.Pos」)が 1000.0 以上になれば、軸 2 のギア動作を解除し減速度 DecRate で軸 2 を減速停止させます。そのとき、軸 3 はギア動作を継続させます。

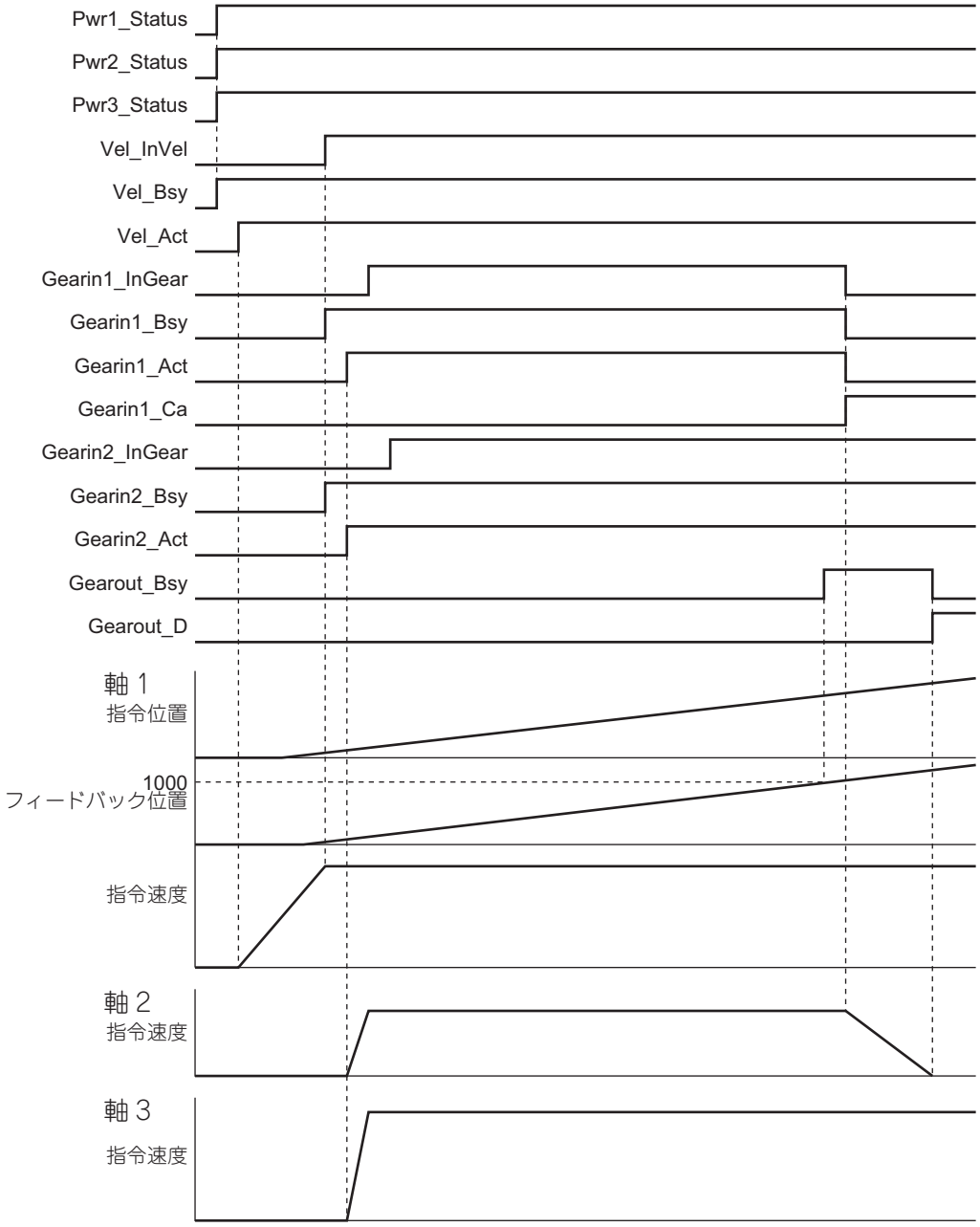
ラダー図

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis000.Act.Pos	LREAL	−	軸 1 のフィードバック現在位置を表します。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
MC_Axis002	_sAXIS_REF	−	従軸である軸 3 の軸変数です。
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr3_Status	BOOL	FALSE	MC_Power のインスタンス PWR3 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Vel_InVel	BOOL	FALSE	MC_MoveVelocity のインスタンス VEL の出力「InVelocity」に割り付けた変数です。目標速度に到達すると、この変数が TRUE となります。
Gearin1_Act	BOOL	FALSE	MC_GearIn のインスタンス GEARIN1 の出力「Active」に割り付けた変数です。GEARIN1 が制御中になると、この変数が TRUE となります。

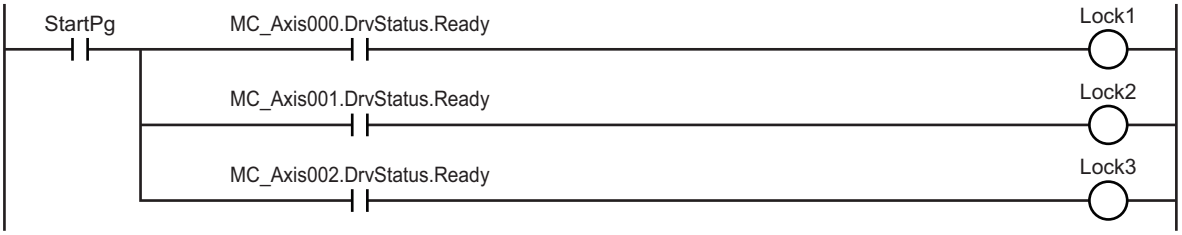
名称	データ型	初期値	コメント
Gearout_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、MC_GearOut のインスタンス GEAROUT を起動します。

● タイミングチャート

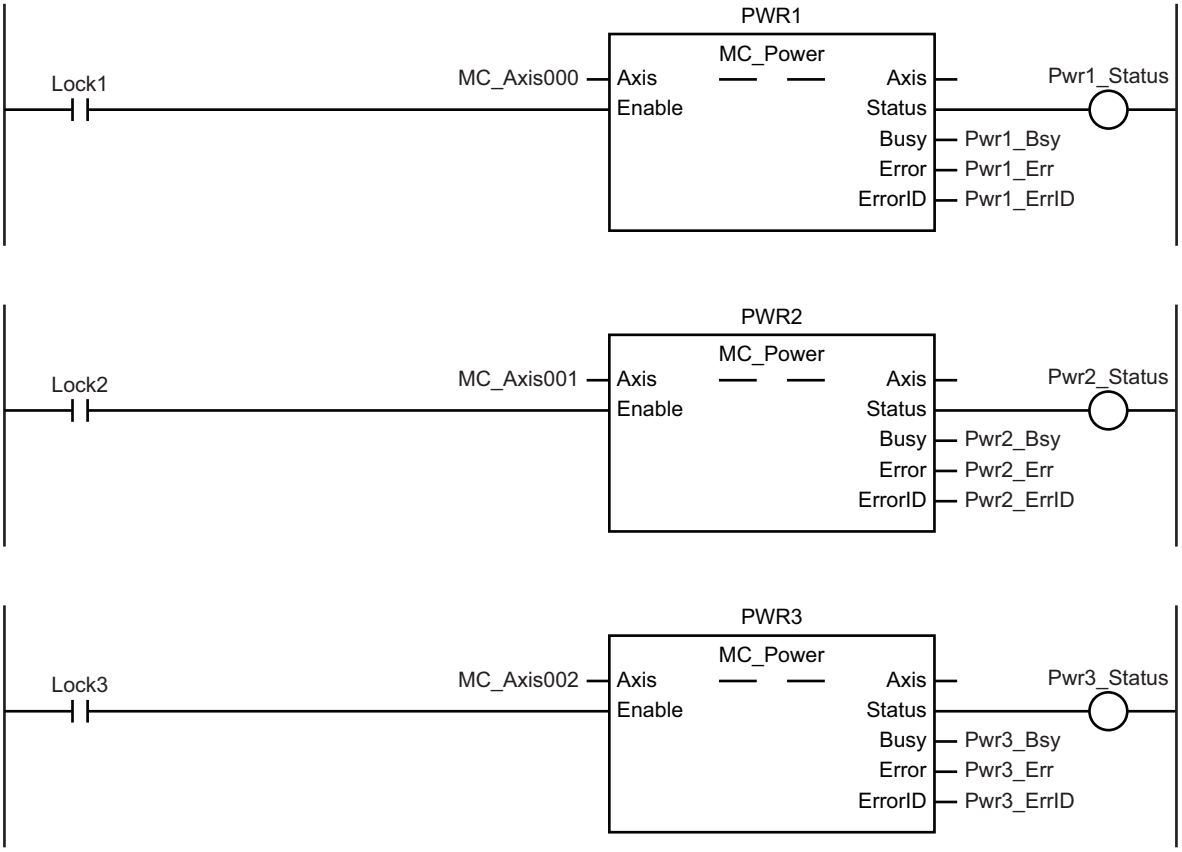


● サンプルプログラム

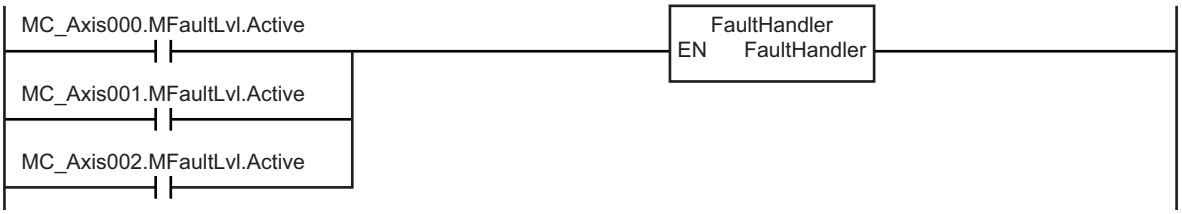
接点 StartPg が TRUE になったら、各軸のサーボドライバがサーボレディ状態になっていることを確認する



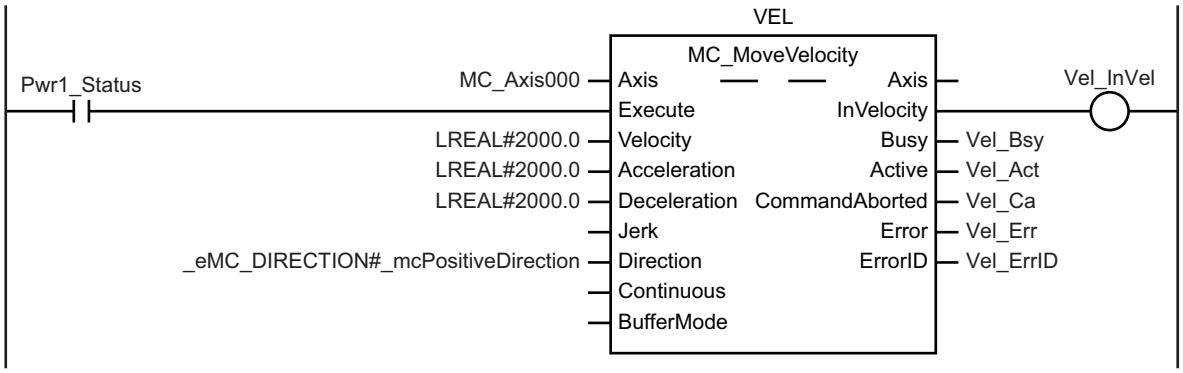
サーボレディ状態になっていれば、各軸をサーボ ON 状態にする



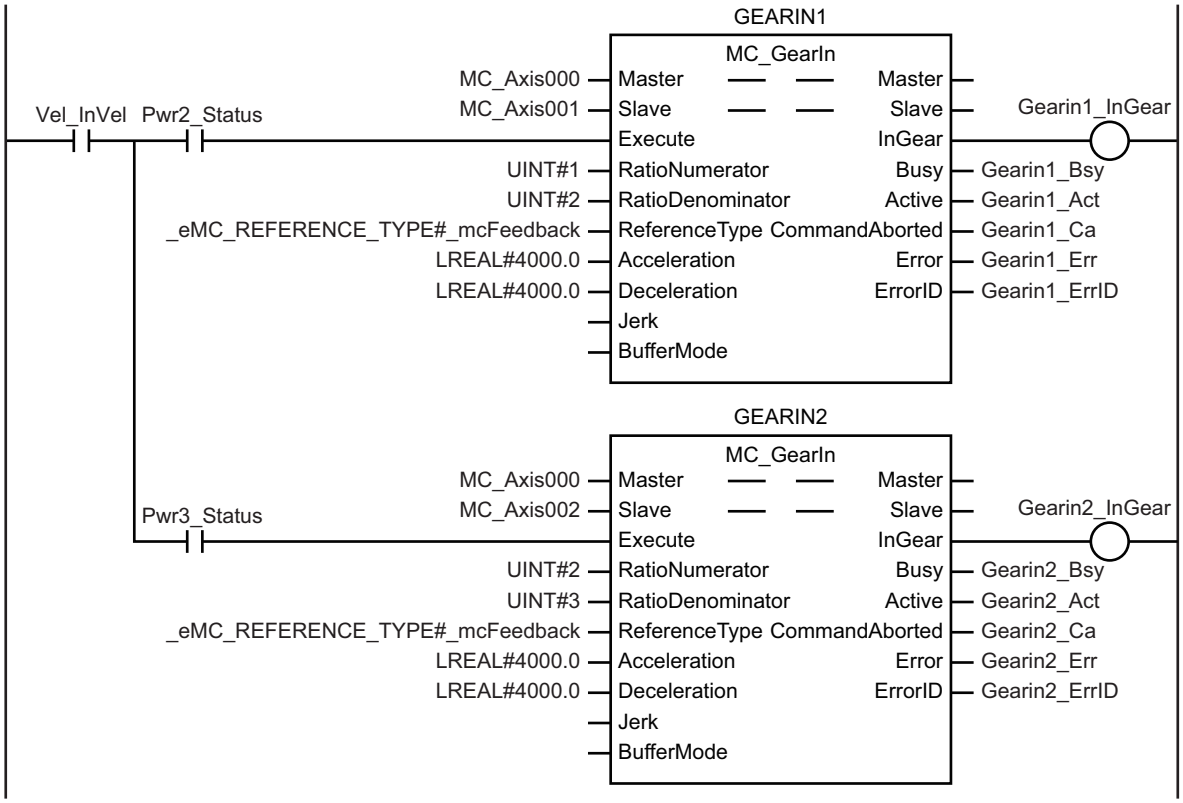
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



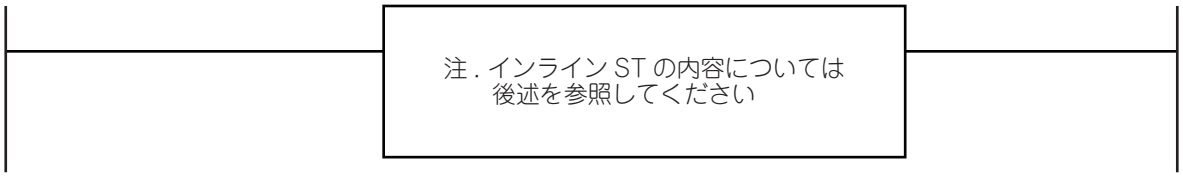
主軸（軸 1）が運転可能状態になったら、MC_MoveVelocity（速度制御）命令を起動する



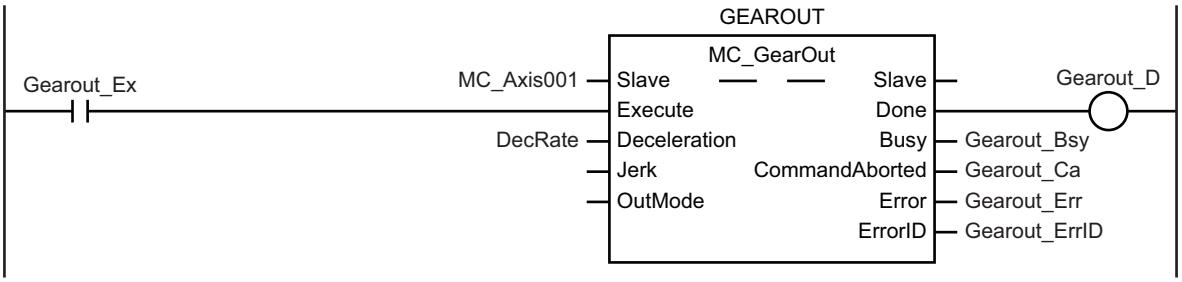
主軸（軸 1）のフィードバック速度が目標速度に到達したときに、従軸（軸 2）をギア比 1:2 で、従軸（軸 3）をギア比 2:3 でギア動作を開始する



従軸（軸 2）がギア動作中で、主軸のフィードバック位置が 1000.0 以上のとき、Gearout_Ex を TRUE にする



Gearout_Ex が TRUE になったら、従軸（軸 2）のギア動作を解除し、減速停止する



インライン ST の内容

```
IF (Gearin1_Act=TRUE) AND (MC_Axis000.Act.Pos>=LREAL#1000.0) THEN
    Gearout_Ex := TRUE;
END_IF;
```

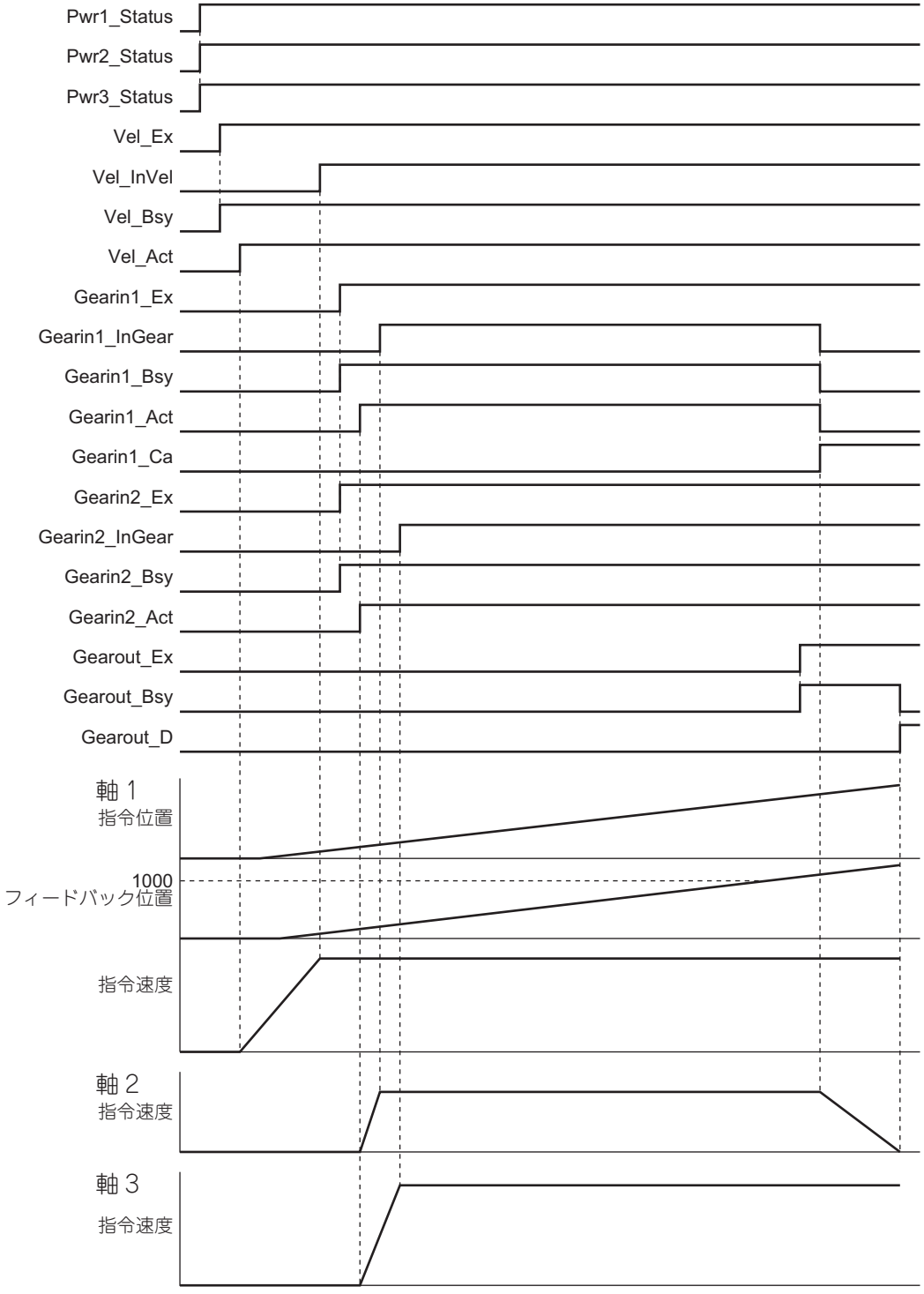
ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis000.Act.Pos	LREAL	−	軸 1 のフィードバック現在位置を表します。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
MC_Axis002	_sAXIS_REF	−	従軸である軸 3 の軸変数です。
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr3_Status	BOOL	FALSE	MC_Power のインスタンス PWR3 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Vel_InVel	BOOL	FALSE	MC_MoveVelocity のインスタンス VEL の出力「InVelocity」に割り付けた変数です。目標速度に到達すると、この変数が TRUE となります。
Gearin1_Act	BOOL	FALSE	MC_GearIn のインスタンス GEARIN1 の出力「Active」に割り付けた変数です。GEARIN1 が制御中になると、この変数が TRUE となります。
Gearout_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、MC_GearOut のインスタンス GEAROUT を起動します。
Vel_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、MC_MoveVelocity のインスタンス VEL を起動します。
Gearin1_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、MC_GearIn のインスタンス GEARIN1 を起動します。
Gearin2_Ex	BOOL	FALSE	この変数が FALSE → TRUE のとき、MC_GearIn のインスタンス GEARIN2 を起動します。

名称	データ型	初期値	コメント
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート



● サンプルプログラム

```
// 入力パラメータが設定されていない場合の処理
IF InitFlag = FALSE THEN

    //MC_MoveVelocity パラメータ
    Vel_Vel := LREAL#2000.0;
    Vel_Acc := LREAL#2000.0;
    Vel_Dec := LREAL#2000.0;
    Vel_Dir :=_eMC_DIRECTION#_mcPositiveDirection;

    //MC_GearIn1 パラメータ
    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 パラメータ
    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 パラメータ
    DecRate := LREAL#200.0;
    Gearout_Dec := DecRate;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag:=TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 2 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
```

```
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 3 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis002.DrvStatus.Ready=TRUE) THEN
    Pwr3_En:=TRUE;
ELSE
    Pwr3_En:=FALSE;
END_IF;

//軸 1～軸 3 で軽度フォールトが発生したら、異常時処理 FaultHandler を実行します
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (MC_Axis002.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

// 軸 1 がサーボ ON 状態のとき、MC_MoveVelocity を起動します
IF Pwr1_Status=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

//MC_MoveVelocity の InVelocity が TRUE で、かつ軸 2 がサーボ ON 状態のとき、主軸を軸 1,従軸を軸 2 に設定した MC_GearIn を起動します
IF (Vel_InVel=TRUE) AND (Pwr2_Status=TRUE) THEN
    Gearin1_Ex := TRUE;
END_IF;

//MC_MoveVelocity の InVelocity が TRUE で、かつ軸 3 がサーボ ON 状態のとき、主軸を軸 1,従軸を軸 3 に設定した MC_GearIn を起動します
IF (Vel_InVel=TRUE) AND (Pwr3_Status=TRUE) THEN
    Gearin2_Ex := TRUE;
END_IF;

//軸 2 がギア動作中で、かつ軸 1 のフィードバック位置が 1000.0 以上のとき、従軸を軸 2 に設定して、GearOut を起動します
IF (Gearin1_Act=TRUE) AND (MC_Axis000.Act.Pos>=LREAL#1000.0) THEN
    Gearout_Ex := TRUE;
END_IF;

//軸 1 の MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);
```

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

//軸 3 の 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
);

//軸 1 を主軸、軸 2 を従軸とした 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 => Gearin1_Bsy,
    Active => Gearin1_Act,
    CommandAborted => Gearin1_CA,
```

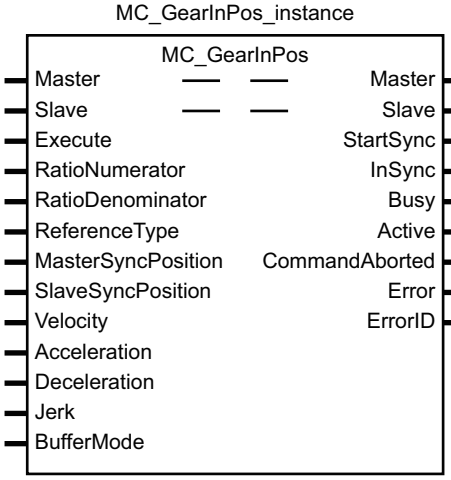
```
    Error => Gearin1_Err,
    ErrorID => Gearin1_ErrID
);

//軸 1 を主軸、軸 3 を従軸とした 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

主軸と従軸間のギア比を設定し、電子ギア動作を行います。
同期を開始する主軸、および従軸の位置を指定します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GearInPos	位置指定ギア動作	FB		MC_GearInPos_instance (Master :=「パラメータ」, Slave :=「パラメータ」, Execute :=「パラメータ」, RatioNumerator :=「パラメータ」, RatioDenominator :=「パラメータ」, ReferenceType :=「パラメータ」, MasterSyncPosition :=「パラメータ」, SlaveSyncPosition :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, BufferMode :=「パラメータ」, StartSync =>「パラメータ」, InSync =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
RatioNumerator	ギア比分子	DINT *1	正数または負数 *1	10000	主軸と従軸間の電子ギアの分子を指定します。
RatioDenominator	ギア比分母	UDINT *2	正数	10000	主軸と従軸間の電子ギアの分母を指定します。
ReferenceType *3	位置種別選択	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *4	位置の種別を指定します。 0: 指令位置（直前タスク周期 *5 での計算値） 1: フィードバック位置（同一タスク周期 *5 での取得値） 2: 指令位置（同一タスク周期 *5 での計算値）

入力変数	名称	データ型	有効範囲	初期値	内容
MasterSyncPosition	主軸同期位置	LREAL	負数、正数、「0」	0	絶対座標の主軸同期位置を指定します。 単位は［指令単位］となります。*6
SlaveSyncPosition	従軸同期位置	LREAL	負数、正数、「0」	0	絶対座標の従軸同期位置を指定します。 単位は［指令単位］となります。*6
Velocity	目標速度	LREAL	正数	0	目標速度を指定します。目標速度は必ず設定してください。設定せずに動作させると異常が発生します。 単位は［指令単位／s］となります。*6
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は［指令単位／s^2］となります。*6
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*6
Jerk（Reserved）	ジャーク	LREAL	0	0	（Reserved）
BufferMode（Reserved）	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *4	（Reserved）

*1. ユニットバージョン Ver.1.02 以降の CPU ユニットと Sysmac Studio Ver.1.03 以降の組み合わせで使用可能です。それ以前の組み合わせでは、データ型は「UINT」、有効範囲「正数」となります。
*2. ユニットバージョン Ver.1.02 以降の CPU ユニットと Sysmac Studio Ver.1.03 以降の組み合わせで使用可能です。それ以前の組み合わせでは、データ型は「UINT」となります。
*3. _mcLatestCommand を使用する場合は、次の関係となるように主軸、従軸を使用しなければなりません。
「Master」（主軸）に設定されているモーション制御システム変数の軸番号 ＜ 「Slave」（従軸）に設定されているモーション制御システム変数の軸番号
*4. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*5. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。
*6. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

出力変数

出力変数	名称	データ型	有効範囲	内容
StartSync	追従中	BOOL	TRUE, FALSE	同期のための加減速動作を開始したときに TRUE となります。
InSync	同期中	BOOL	TRUE, FALSE	従軸が従軸同期位置に到達したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
StartSync	軸が移動を開始したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
InSync	従軸の位置が、「SlaveSyncPosition」に到達したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Busy	「Execute」の立ち上がり	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を受け付けたとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に本命令を起動したとき。 • MC_GearOut 命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Master	主軸	_sAXIS_REF	−	主軸を指定します。*1
Slave	従軸	_sAXIS_REF	−	従軸を指定します。*1

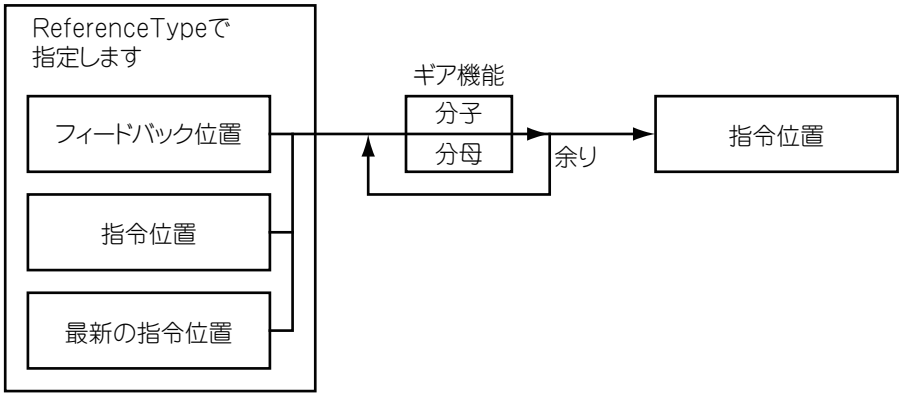
1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

使用上の注意

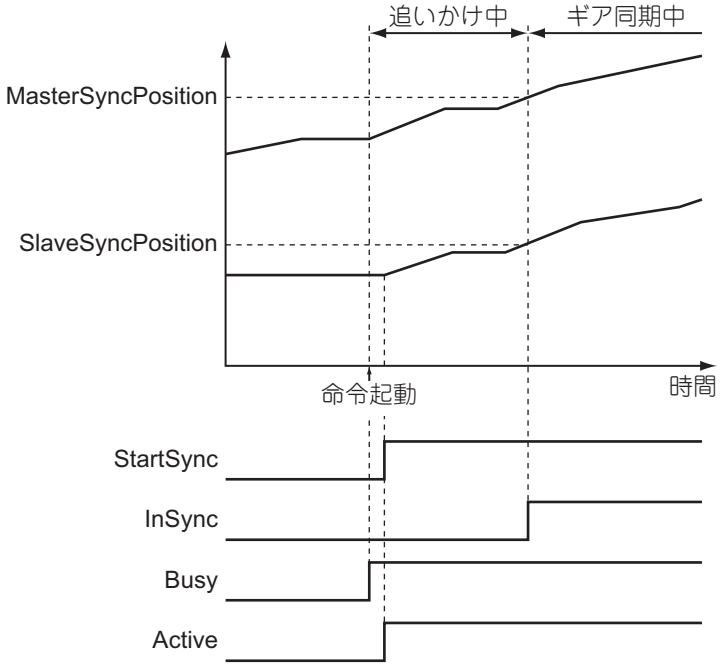
主軸と従軸に同じ軸を指定すると、軽度フォールトレベルの「主軸従軸同一（エラーコード: 5436 Hex)」異常となります。

機能説明

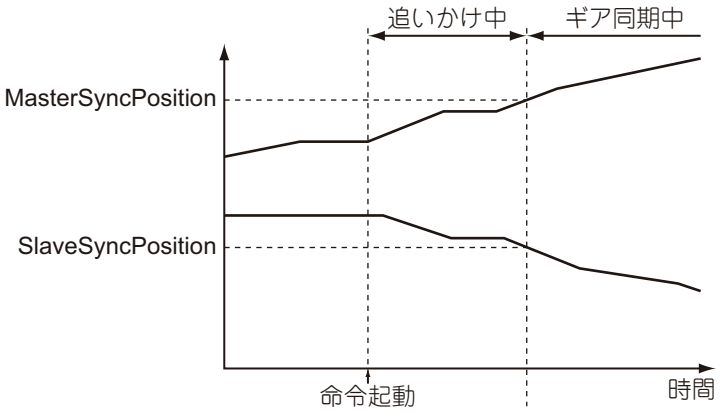
- 動作対象の軸を「Slave」（従軸）で指定し、「RatioNumerator」（ギア比分子）、「RatioDenominator」（ギア比分母）、「ReferenceType」（位置種別選択）、「Acceleration」（加速度）、「Deceleration」（減速度）を指定してギア動作を行います。
- 「Master」（主軸）には指令位置とフィードバック位置、最新の指令位置を指定することができます。



- 動作開始後、「Slave」（従軸）は、追いかけ動作により「Master」（主軸）に同期しながら加減速動作を行います。
- 従軸が従軸同期位置に到達するまでを Catching phase（追いかけ中）とし、到達後を InSync phase（ギア同期中）と呼びます。いずれの区間でも主軸の位置に同期します。
- 入力変数の「Velocity」（目標速度）は、Catching phase（追いかけ中）時の目標速度となります。
- 従軸は動作開始時の主軸と同方向に動作を行います。
  動作開始時に、主軸速度が「0」であった場合は異常となります。
  また、主軸が正方向動作、かつ「SlaveSyncPosition」（従軸同期位置）が起動時の従軸位置より小さい場合には、従軸は反転動作となります。
  同様に主軸が負方向動作で、かつ「SlaveSyncPosition」（従軸同期位置）が起動時の従軸位置より大きな場合も、従軸は反転動作となります。
- 主軸の速度が周期ごとに大きく変動する場合、従軸の速度も一定になりません。
- ギア比が正数の場合、「Slave」（従軸）は「Master」（主軸）の同方向に移動します。



- ギア比が負数の場合、「Slave」（従軸）は「Master」（主軸）の逆方向に移動します。

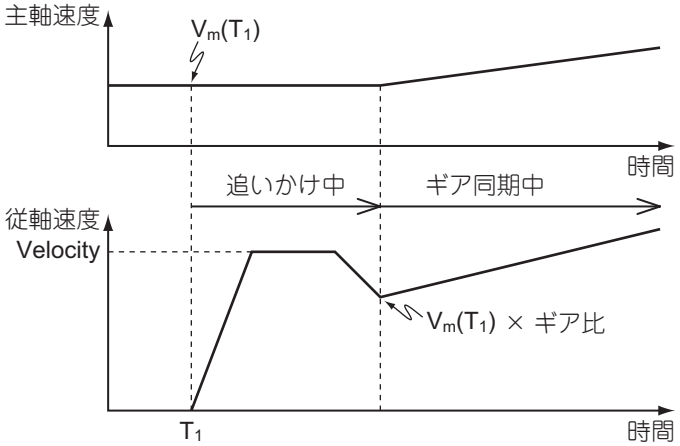


使用上の注意

ユニットバージョン Ver.1.09 以前の CPU ユニットで本命令を使用する場合は、「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行しないでください。「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行すると、「Slave」（従軸）が急激な追従をすることがあり、危険です。
「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を使用する場合は、「Master」（主軸）と「Slave」（従軸）の関係をいったん解除してから実行してください。

- MC 機能モジュールは、以下の 3 つの速度で、「Acceleration」（加速度）、および「Deceleration」（減速度）を用いて直線で加減速する速度プロファイルを計算します。
  a）命令起動時の「Slave」（従軸）の速度を初速
  b）命令起動時の「Master」（主軸）の速度にギア比を乗じた速度を終速
  c）「Velocity」（目標速度）を目標速度

  追いかけ中の移動量が短い場合には、目標速度に達しないこともあります。



「MasterSyncPosition」（主軸同期位置）、「SlaveSyncPosition」（従軸同期位置）において「Slave」（従軸）が「Master」（主軸）に追いつくためには、「Velocity」（目標速度）は以下の条件を満足する必要があります。

$$\text{Velocity} > \frac{\text{MC\_GearInPos 起動時の主軸速度} \times \text{ギア比分子}}{\text{ギア比分母}}$$

主軸速度として使用する情報は、「ReferenceType」（位置種別選択）の設定で異なります。

「ReferenceType」の設定	主軸速度として使用する情報
_mcCommand、_mcLatestCommand 設定時	指令現在速度を使用します。
_mcFeedback 設定時	フィードバック現在速度を使用します。

「MasterSyncPosition」（主軸同期位置）、「SlaveSyncPosition」（従軸同期位置）で、「Slave」（従軸）が「Master」（主軸）に追いつけない場合、「位置指定ギア動作目標速度不足（エラーコード: 5447 Hex）」となります。

- カウントモードが［ロータリモード］の場合、CPU ユニットのユニットバージョンで、以下のような違いがあります。

  a）ユニットバージョン Ver.1.10 以降の CPU ユニットの場合
  主軸カウントモードが［ロータリモード］の場合、リングカウンタ上下限設定値の範囲を超えた「MasterSyncPosition」（主軸同期位置）の指定が可能です。この場合、主軸現在位置と主軸同期位置の関係は、MC_MoveAbsolute（絶対値位置決め）命令で「Direction」（方向選択）に［方向指定なし］を指定した場合と同じになります。MC_MoveAbsolute（絶対値位置決め）命令は、「MC_MoveAbsolute (P.3-47)」を参照してください。
  同様に、従軸カウントモードが［ロータリモード］の場合、リングカウンタ上下限設定値の範囲を超えた「SlaveSyncPosition」（従軸同期位置）の指定が可能です。

  b）ユニットバージョン Ver.1.09 以前の CPU ユニットの場合
  カウントモードが［ロータリモード］の場合、リングカウンタが 1 周以内に同期動作が開始するようにしてください。

使用上の注意

主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。

● ReferenceType（位置種別選択）

位置の種類を、以下から選択できます。

- _mcCommand: 指令位置（直前タスク周期での計算値）
  現周期に対して、直前のタスク周期で計算した主軸の指令位置を使用します。
  従軸の指令位置を計算する直前の定周期タスクで、計算した主軸の指令位置を使用します。
- _mcFeedback: 同一タスク周期での取得値
  同一タスク周期で取得した主軸のフィードバック位置を使用します。
- _mcLatestCommand: 指令位置（同一タスク周期での計算値）
  同一タスク周期で計算した主軸の指令位置を使用します。
  _mcCommand よりも新しい情報を使用することができます。ただし、主軸番号は従軸番号より若い番号を設定しなければなりません。
  従軸が、主軸より若い軸番号の場合、本命令の「Error」（エラー）が TRUE となります。「ErrorID」（エラーコード）には、「主軸／従軸 軸番号非昇順（エラーコード: 5438 Hex）」が出力されます。

使用上の注意

ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。同様に定周期タスクは、プライマリ定周期タスクまたは定周期タスク（実行優先度 5）を表します。



参考

同一タスク周期で計算した指令位置を選択した方が、直前タスク周期での計算値を選択した場合より、同期精度は向上します。
ただし、[「Master」(主軸)に設定されているモーション制御システム変数の軸番号 <「Slave」(従軸)に設定されているモーション制御システム変数の軸番号]となるように主軸、従軸を使用しなければなりません。

● 軸種別と位置種別の関係

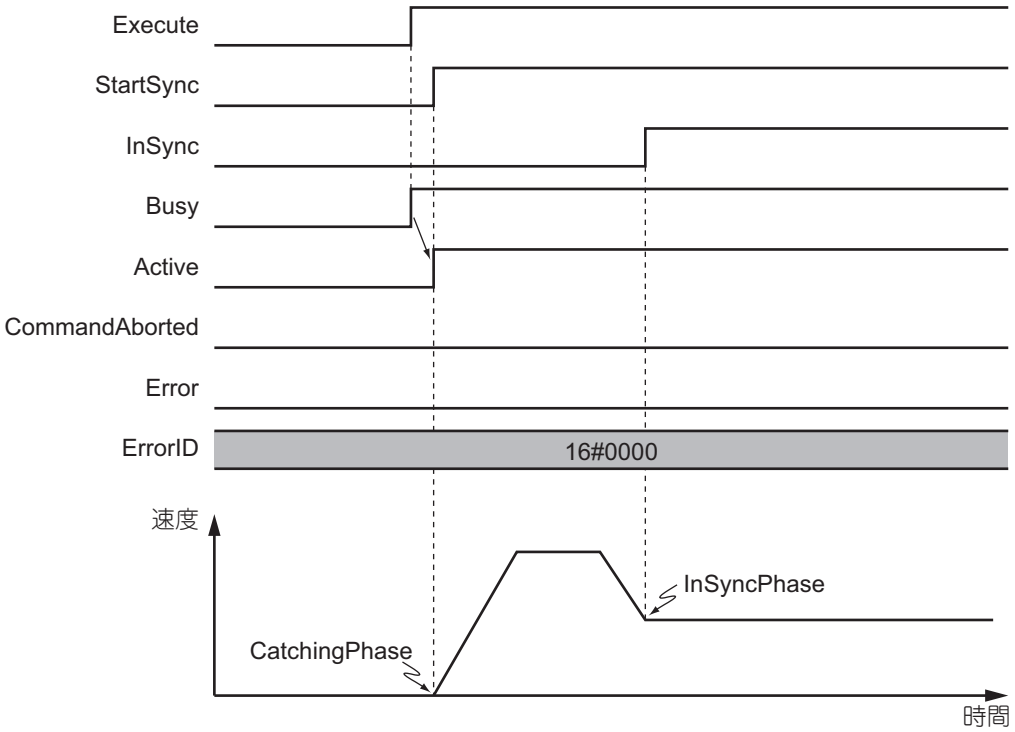
監視可能な軸種別と監視する位置種別の関係は以下のようになります。

軸種別	ReferecneType	
	_mcCommand、または _mcLatestCommand	_mcFeedback
サーボ軸	○	○
エンコーダ軸	× *1	○
仮想サーボ軸	○	○
仮想エンコーダ軸	× *1	○

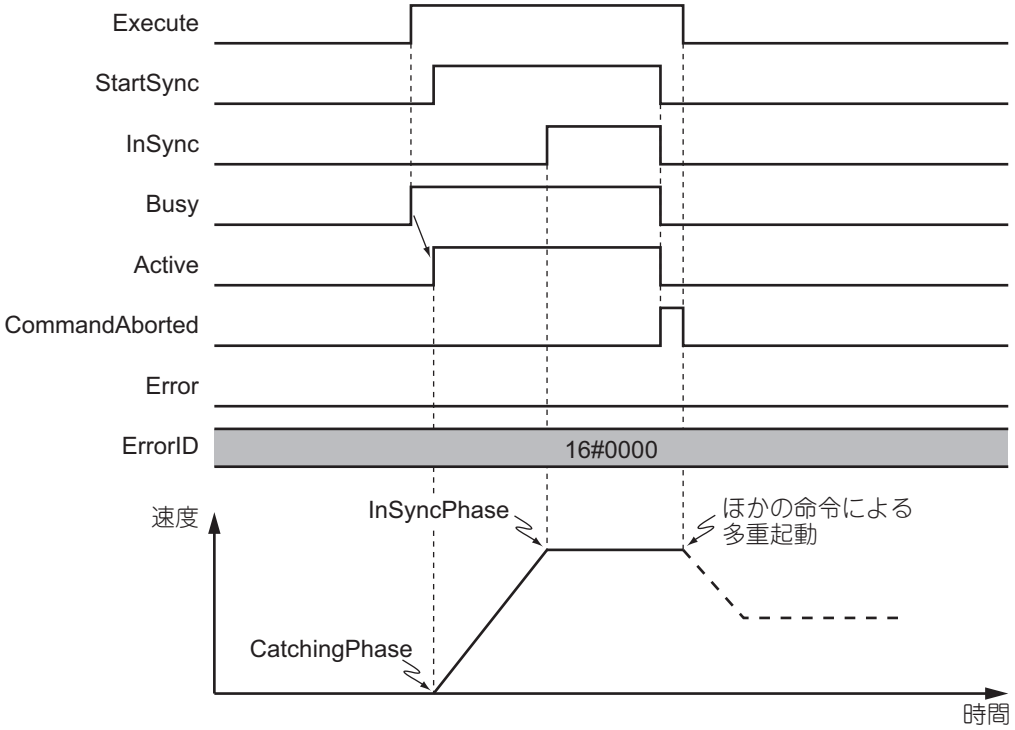
*1. 命令起動時に、「位置種別選択範囲外(エラーコード: 5430 Hex)」が発生します。

タイミングチャート

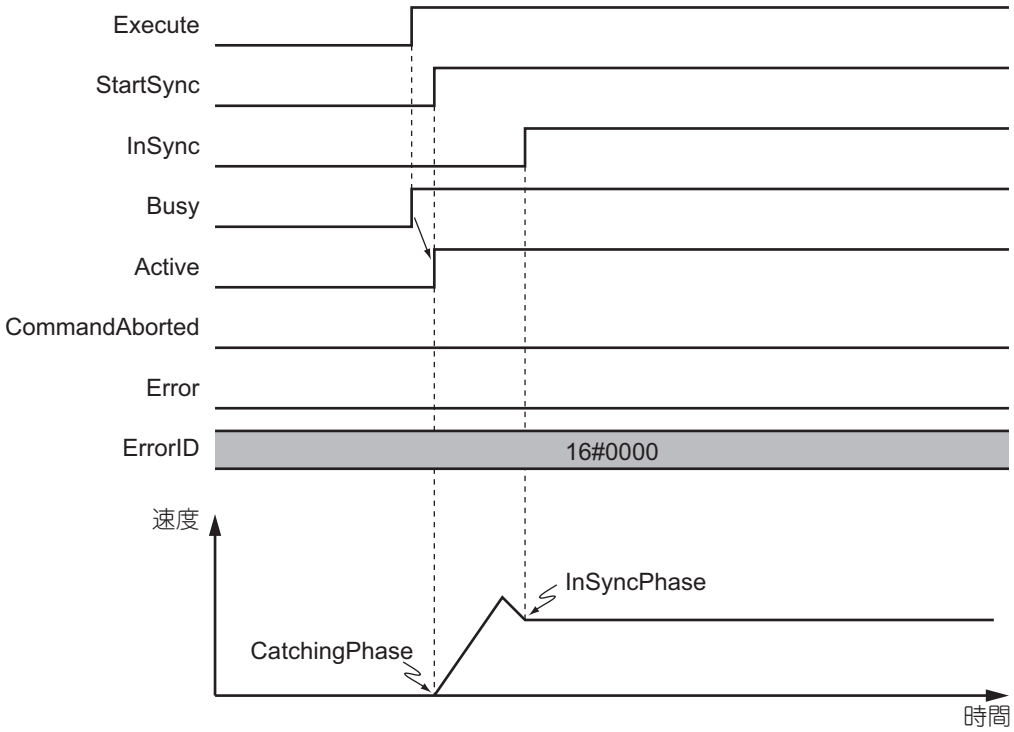
- 「Execute」(起動)の立ち上がりで、電子ギアの動作を開始します。
- 「Execute」(起動)の立ち上がりと同時に「Busy」(実行中)が TRUE となります。動作開始後に「Active」(制御中)と「StartSync」(追従中)が TRUE となり、「Slave」(従軸)がギア動作を開始します。
- 「MasterSyncPosition」(主軸同期位置)、「SlaveSyncPosition」(従軸同期位置)に到達しすると「InSync」(同期中)が TRUE となります。
- ほかの命令により、本命令が中断された場合は「CommandAborted」(実行中断)が TRUE となり、「Busy」(実行中)、「Active」(制御中)、「StartSync」(追従中)、「InSync」(同期中)が FALSE となります。



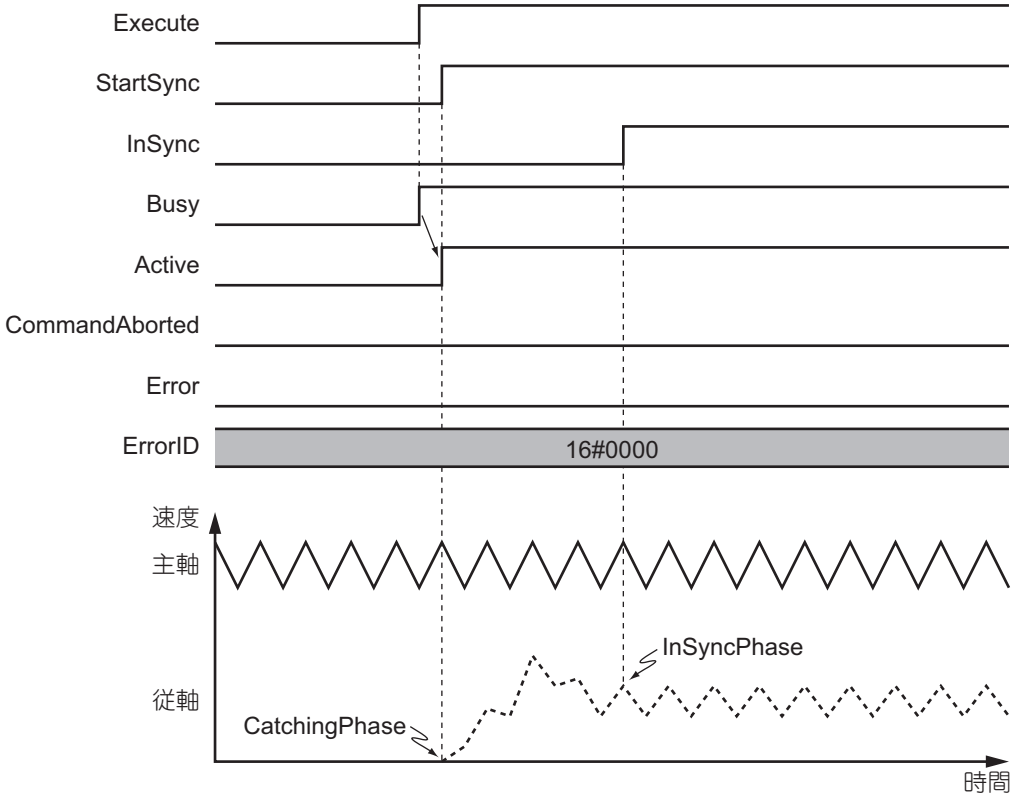
ほかの命令により、本命令が中断された場合を以下に示します。



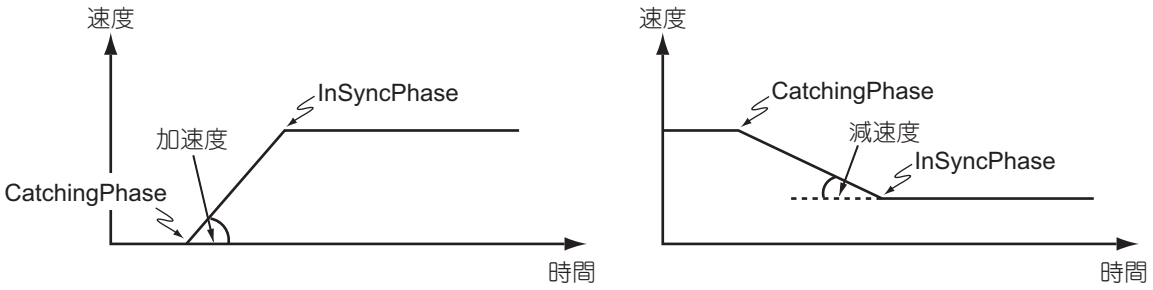
「SlaveSyncPosition」の設定によっては、目標速度に到達しない場合あります。その場合の例を下記に示します。



また、InSync phase 到達前も、主軸位置に追従します。その場合の例を下記に示します。



入力変数に、「Acceleration」（加速度）、「Deceleration」（減速度）を指定することができます。電子ギアの動作例を下記に示します。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

異常停止中、減速停止中を含む主軸の軸指令状態は、本命令実行には影響しません。
モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

多重起動時の本命令の動作を、「BufferMode」（バッファモード選択）で指定します。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

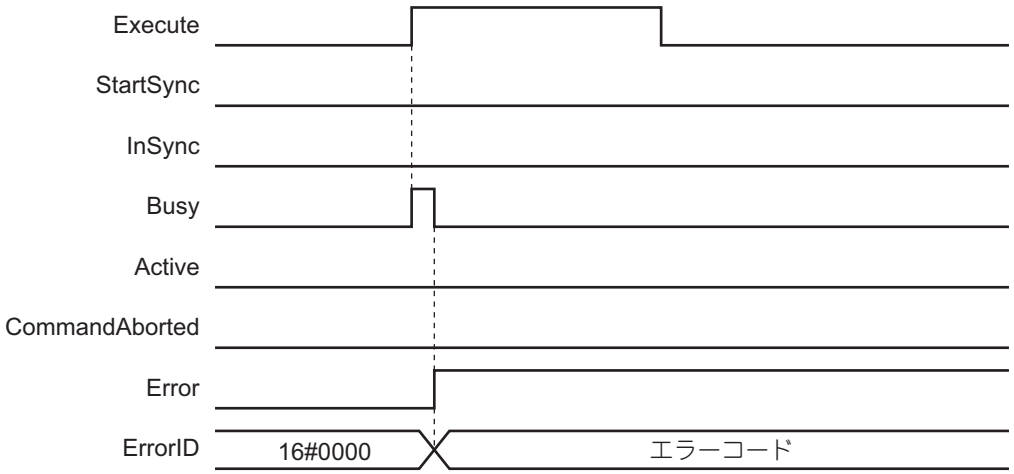
使用上の注意

他命令を「BufferMode」（バッファモード選択）の［0: アボーティング］以外で起動すると、他命令は異常になります。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



同期制御中は、主軸の異常状態には影響を受けません。主軸の異常状態が解除され、主軸の動作時には、従軸は電子ギア動作を継続します。
本命令起動時および本命令実行中に、従軸に異常が発生した場合、主軸に影響を与えることはありません。

● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、同期する位置を主軸は Pos1、従軸は Pos2 として動作するサンプルプログラムを掲載します。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

軸種別

軸	軸種別
軸 1	サーボ軸（主軸）
軸 2	サーボ軸（従軸）

カウンタモード

軸	カウンタモード
軸 1	ロータリモード
軸 2	ロータリモード

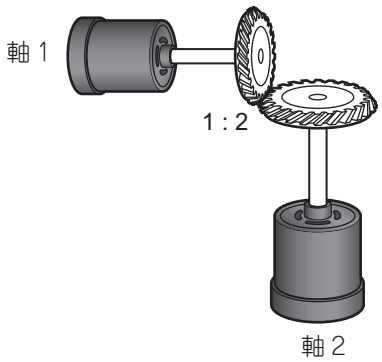
リングカウンタ

軸	上限値	下限値
軸 1	360	0
軸 2	360	0

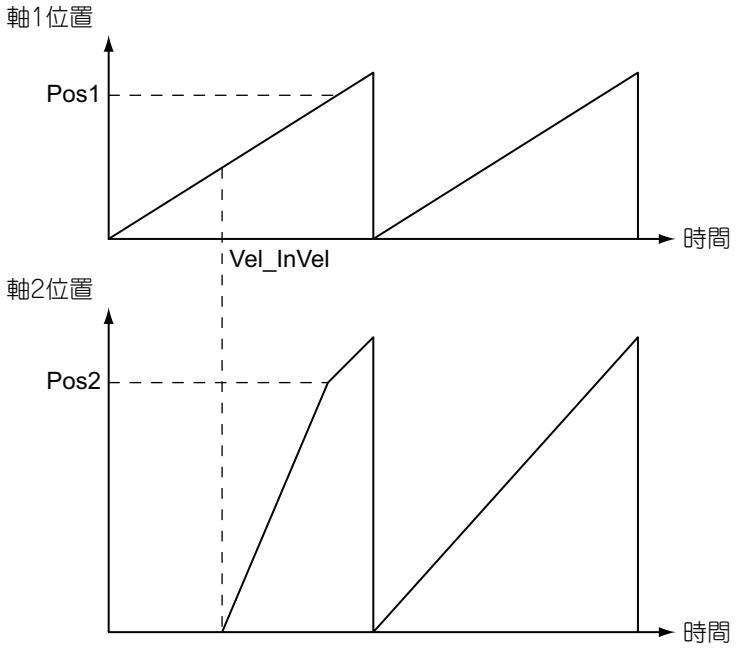
座標単位選択

軸	座標単位
軸 1	degree
軸 2	degree

動作例



● **動作パターン**



1 **主軸の開始**

主軸（軸 1）は実サーボ軸であり速度制御で動作します。

2 **主軸の目標速度到達**

主軸の指令速度が目標速度に到達したら、主軸の「InVelocity」（目標速度到達）が TRUE となります。

3 **従軸の起動**

主軸の「InVelocity」（目標速度到達）が TRUE となったら、主軸のフィードバック位置に対して従軸（軸 2）をギア比 1:2 でギア動作させます。

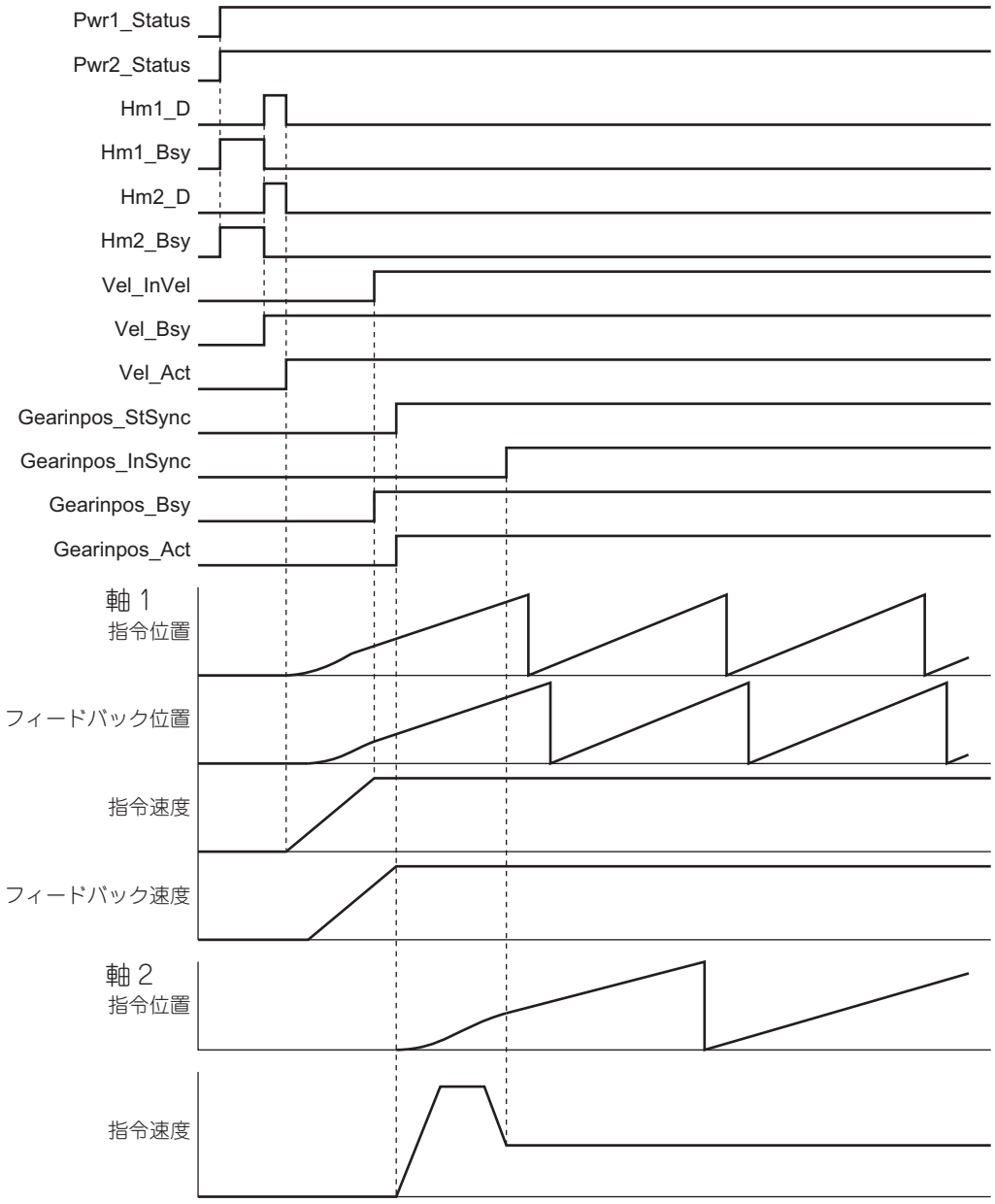
同期する位置は主軸が Pos1、従軸が Pos2 とします。

ラダー図

● 主な変数

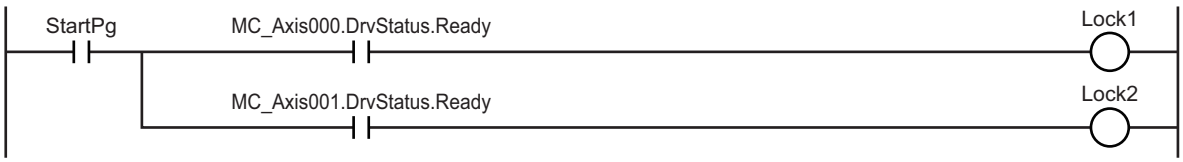
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態の場合に TRUE となります。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態の場合に TRUE となります。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Hm1_D	BOOL	FALSE	MC_Home のインスタンス HM1 の出力「Done」に割り付けた変数です。
Hm2_D	BOOL	FALSE	MC_Home のインスタンス HM2 の出力「Done」に割り付けた変数です。
Vel_InVel	BOOL	FALSE	MC_MoveVelocity のインスタンス VEL の出力「InVelocity」に割り付けた変数です。目標速度に到達すると、この変数が TRUE となります。
Pos1	LREAL	−	主軸の同期位置を表す変数です。
Pos2	LREAL	−	従軸の同期位置を表す変数です。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。

● タイミングチャート

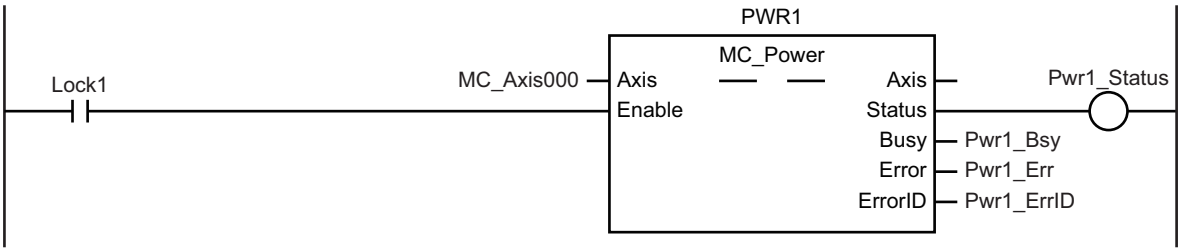


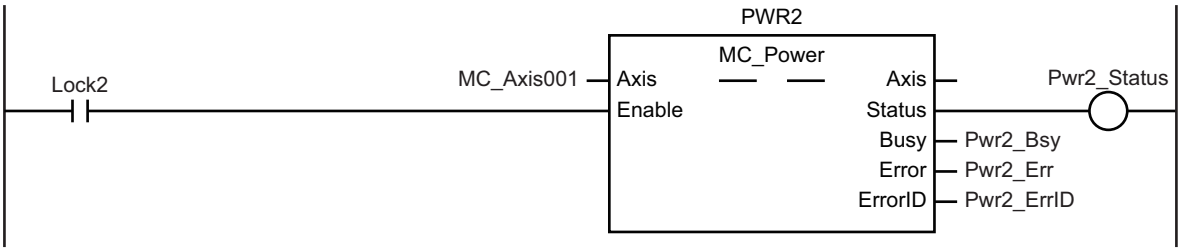
● サンプルプログラム

接点 StartPg が TRUE になったら、各軸のサーボドライバがサーボレディ状態になっていることを確認する

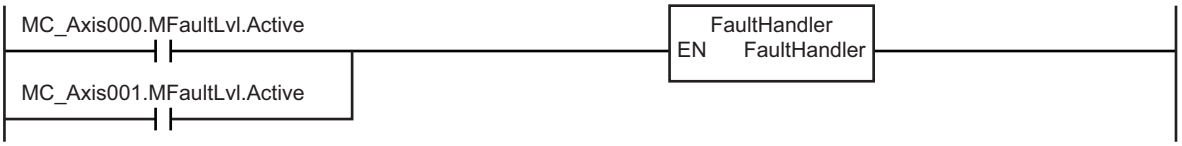


サーボレディ状態になっていれば、各軸をサーボ ON 状態にする

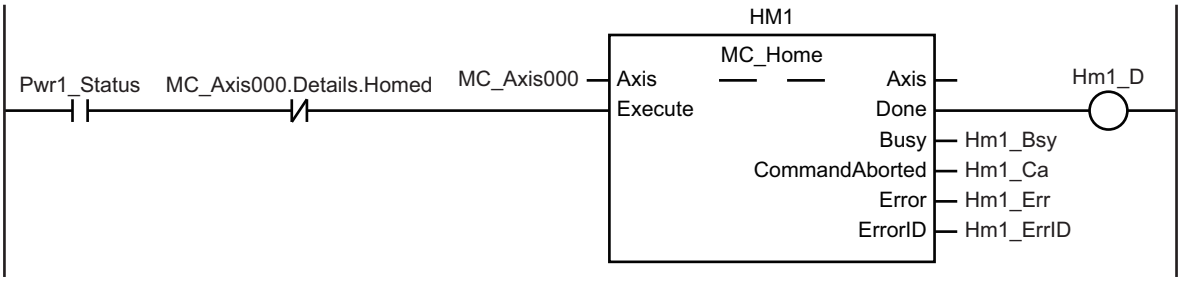




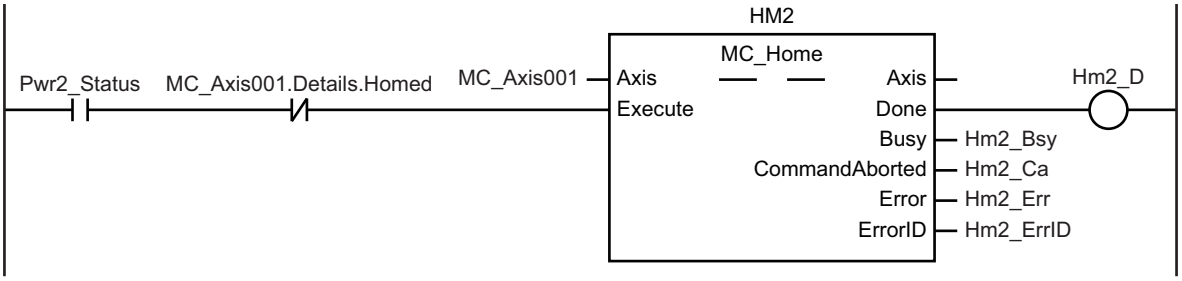
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



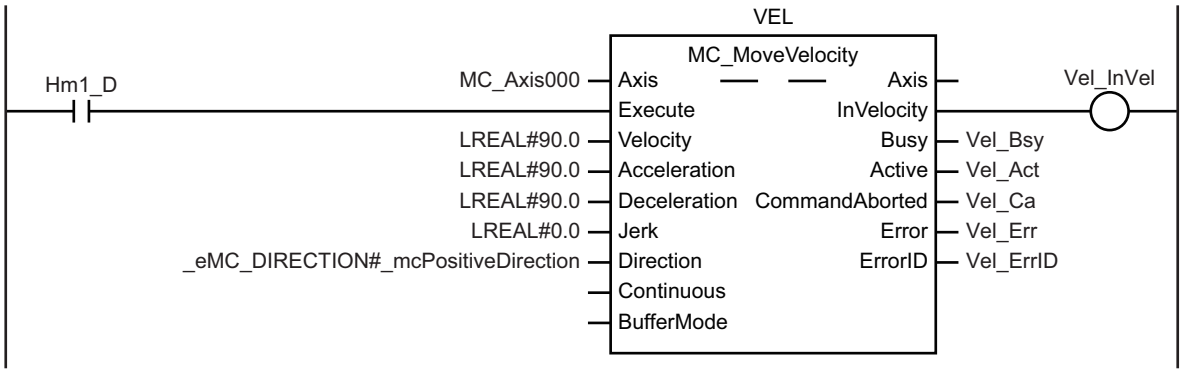
主軸（軸 1）がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



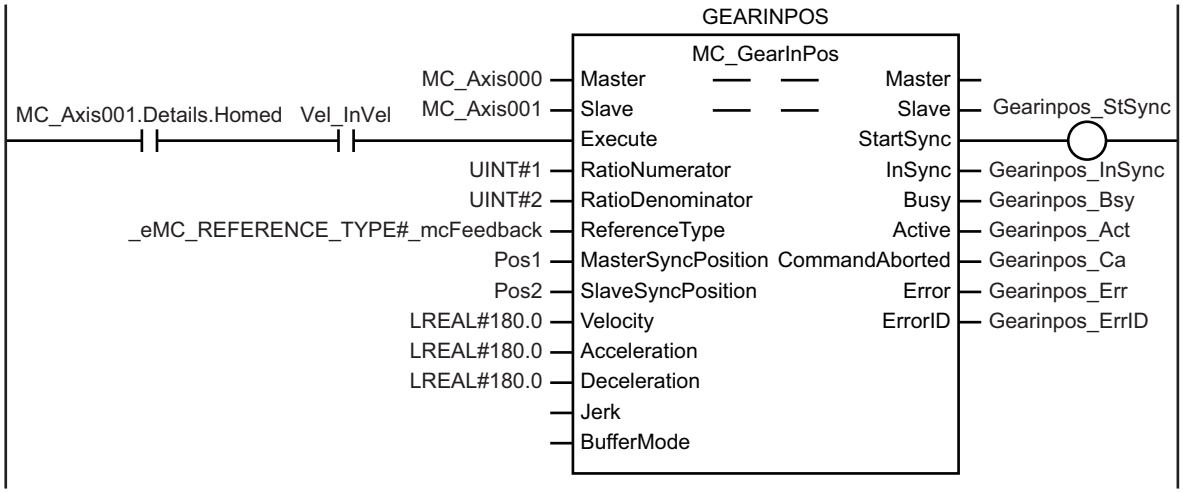
従軸（軸 2）がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



主軸（軸 1）が原点復帰完了後、MC_MoveVelocity（速度制御）命令を起動する



従軸（軸 2）が原点復帰完了後、MC_MoveVelocity の Vel_InVel が TRUE で MC_GearInPos（位置指定ギア動作）を起動し、ギア動作を開始する



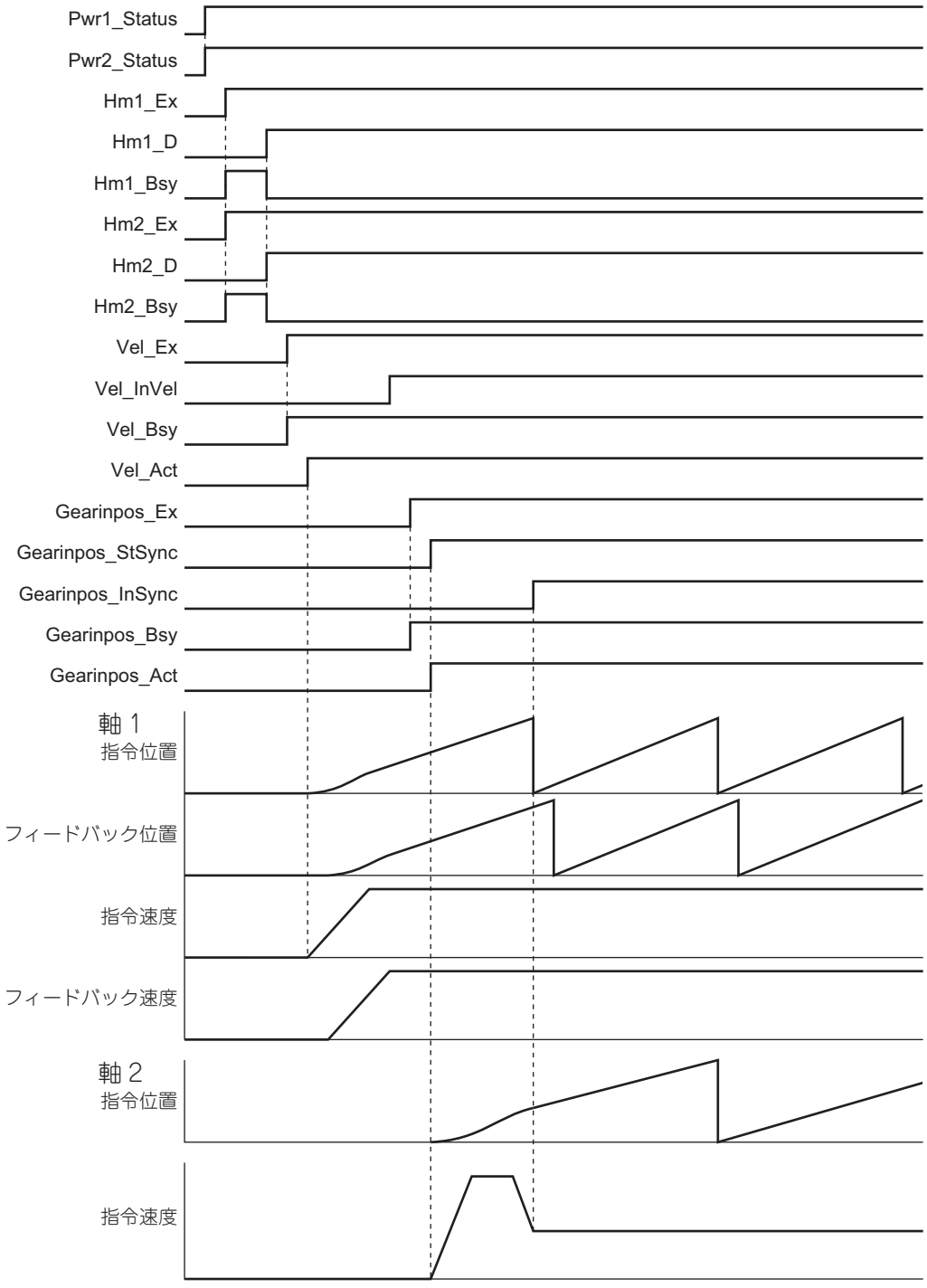
ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態の場合に TRUE となります。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態の場合に TRUE となります。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。

名称	データ型	初期値	コメント
Hm1_D	BOOL	FALSE	MC_Home のインスタンス HM1 の出力「Done」に割り付けた変数です。
Hm2_D	BOOL	FALSE	MC_Home のインスタンス HM2 の出力「Done」に割り付けた変数です。
Vel_InVel	BOOL	FALSE	MC_MoveVelocity のインスタンス VEL の出力「InVelocity」に割り付けた変数です。目標速度に到達すると、この変数が TRUE となります。
Pos1	LREAL	−	主軸の同期位置を表す変数です。
Pos2	LREAL	−	従軸の同期位置を表す変数です。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Gearinpos_Ex	BOOL	FALSE	この変数が FALSE → TRUE になったとき、MC_GearInPos のインスタンス GEARIN1 を起動します。
Vel_Ex	BOOL	FALSE	この変数が FALSE → TRUE になったとき、MC_MoveVelocity のインスタンス VEL を起動します。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート



● サンプルプログラム

```
// 入力パラメータが設定されていない場合の処理
IF InitFlag = FALSE THEN

    // MC_MoveVelocity パラメータ
    Vel_Vel := LREAL#90.0;
    Vel_Acc := LREAL#90.0;
    Vel_Dec := LREAL#90.0;
    Vel_Dir :=_eMC_DIRECTION#_mcPositiveDirection;
```

```
    // MC_GearInPos パラメータ
    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;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag:=TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 2 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE;
ELSE
    Pwr2_En:=FALSE;
END_IF;

//軸 1～軸 2 で軽度フォールトが発生したら、異常時処理 FaultHandler を実行します
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE)THEN
    FaultHandler();
END_IF;

// 軸 1 が運転可状態で、かつ原点未確定状態のとき、原点復帰を実行します
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

// 軸 2 が運転可状態で、かつ原点未確定状態のとき、原点復帰を実行します
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
```

```
    Hm2_Ex:=TRUE;
END_IF;

// 軸 1 が原点復帰完了後、MC_MoveVelocity を起動します
IF Hm1_D=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

//軸 2 が原点復帰完了後、MC_MoveVelocity の Vel_InVel が TRUE で、MC_GearInPos を起動します
IF (MC_Axis001.Details.Homed=TRUE) AND (Vel_InVel=TRUE) THEN
    Gearinpos_Ex := TRUE;
END_IF;

//軸 1 の MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

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

//軸 1 の MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

//軸 2 の 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

実行中の MC_GearIn（ギア動作開始）命令、MC_GearInPos（位置指定ギア動作）命令を中止します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GearOut	ギア動作解除	FB	MC_GearOut_instance MC_GearOut Slave — — Slave Execute — Done Deceleration — Busy Jerk — CommandAborted OutMode — Error ErrorID	MC_GearOut_instance (Slave :=「パラメータ」, Execute :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, OutMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*1
Jerk (Reserved)	ジャーク	LREAL	0	0	(Reserved)
OutMode (Reserved)	同期解除モード選択	_eMC_OUT_MODE	0: _mcStop	0 *2	(Reserved)

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。

出力変数	名称	データ型	有効範囲	内容
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Slave	従軸	_sAXIS_REF	−	従軸を指定します。 *1

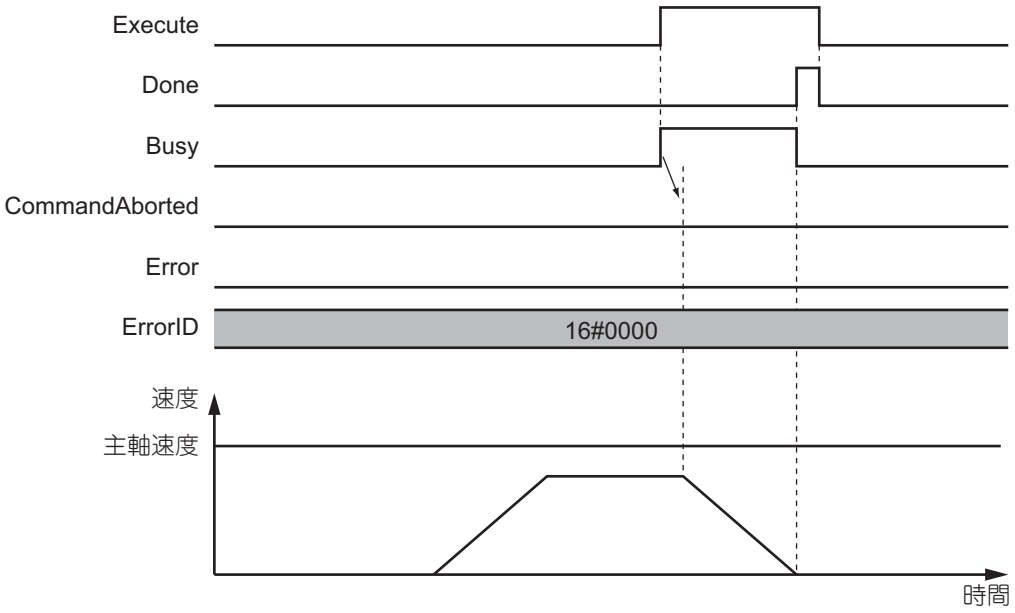
1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

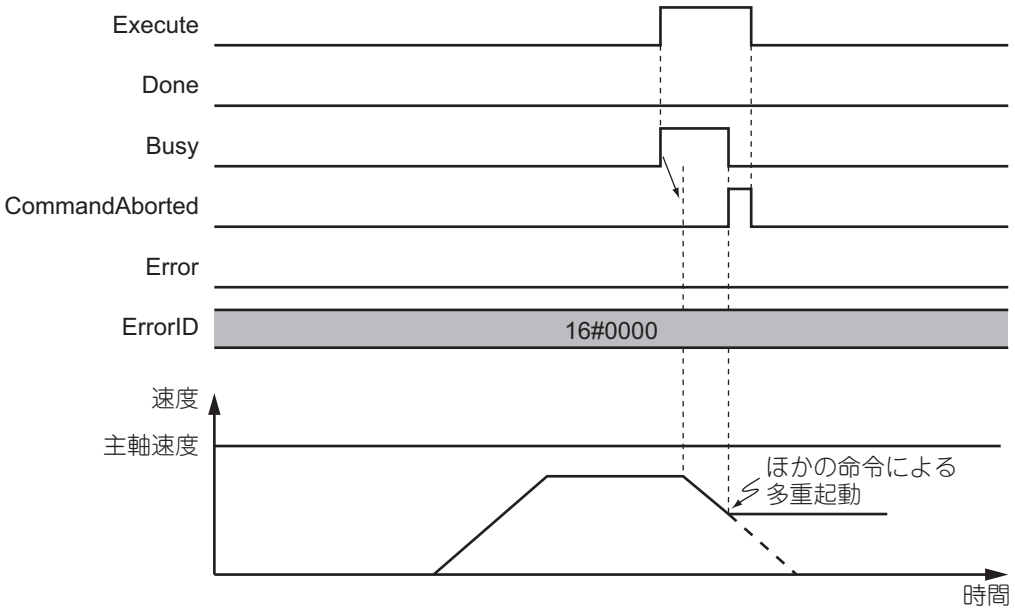
- 動作対象の軸を「Slave」（従軸）で指定し、「Deceleration」（減速度）を指定して、実行中の MC_GearIn（ギア動作開始）命令、MC_GearInPos（位置指定ギア動作）命令を中止します。
- 本命令は、MC_GearIn（ギア動作開始）命令、MC_GearInPos（位置指定ギア動作）命令の主軸の動作に対して、影響を与えません。

タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となります。
- 目標速度に到達すると、「Done」（完了）が TRUE となります。
- ほかの命令により、本命令が中断された場合は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）が FALSE となります。

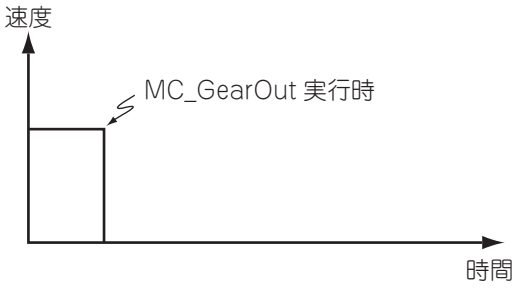


● **ほかの命令により、本命令が中断された場合**



● **「Deceleration」（減速度）を「0」に指定して起動した場合**

「Deceleration」（減速度）を「0」に指定して起動した場合は、減速せずに停止します。
「Deceleration」（減速度）が「0」の場合の動作例を下記に示します。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

MC_GearIn（ギア動作開始）命令または、MC_GearInPos（位置指定ギア動作）命令を実行中に、本命令を起動した場合、MC_GearIn（ギア動作開始）命令または MC_GearInPos（位置指定ギア動作）命令を「CommandAborted」（実行中断）し、本命令は「Busy」（実行中）となります。
MC_GearIn（ギア動作開始）命令または、MC_GearInPos（位置指定ギア動作）命令以外の命令を実行中に、本命令を起動した場合は、本命令は「Error」（エラー）となります。

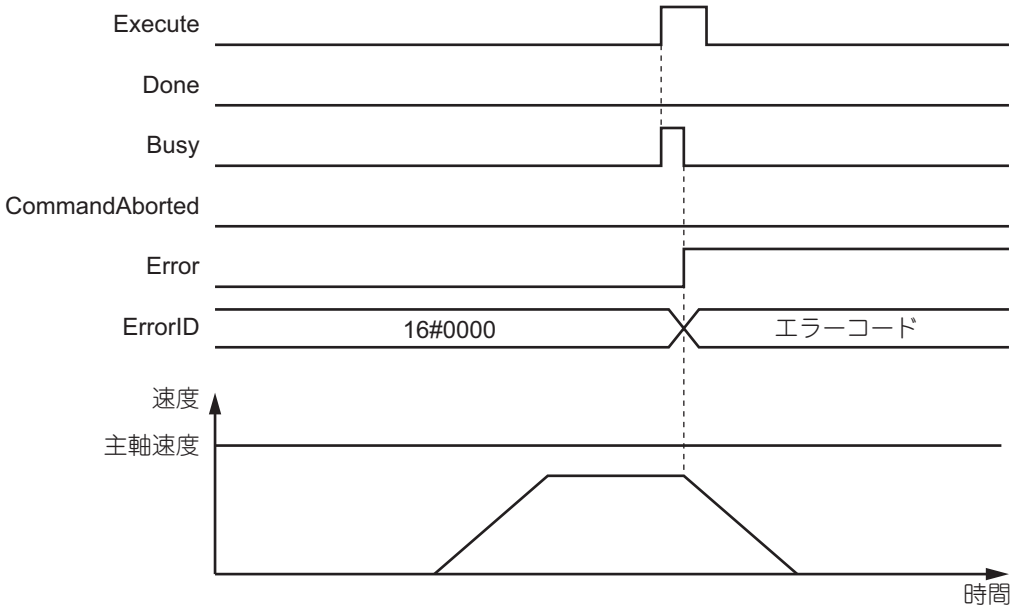
● 本命令実行中の他命令起動

本命令に対するモーション命令多重起動を行う場合、従軸を指定してください。
本命令に対するモーション命令多重起動を行う場合、［アボーティング］、［バッファード］が可能です。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_MoveLink

指定した主軸に同期して位置決めします。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveLink	台形パターンカム	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_MoveLink_instance (Master :=「パラメータ」, Slave :=「パラメータ」, TriggerInput :=「パラメータ」, TriggerVariable :=「パラメータ」, Execute :=「パラメータ」, ReferenceType :=「パラメータ」, SlaveDistance :=「パラメータ」, MasterDistance :=「パラメータ」, MasterDistanceInACC :=「パラメータ」, MasterDistanceInDEC :=「パラメータ」, LinkOption :=「パラメータ」, MasterStartDistance :=「パラメータ」, BufferMode :=「パラメータ」, Done =>「パラメータ」, InSync =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
ReferenceType *1	位置種別選択	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback 2: _mcLatestCommand	0 *2	位置の種別を指定します。 0: 指令位置（直前タスク周期 *3 での計算値） 1: フィードバック位置（同一タスク周期 *3 での取得値） 2: 指令位置（同一タスク周期 *3 での計算値）
SlaveDistance	従軸移動距離	LREAL	負数、正数、「0」	0	従軸の移動距離を指定します。単位は［指令単位］となります。 *4

入力変数	名称	データ型	有効範囲	初期値	内容
MasterDistance	主軸移動距離	LREAL	正数、または「0」	0	主軸の移動距離を符号なしの絶対値表現で指定します。主軸の移動方向は正負とも有効です。 単位は［指令単位］となります。*4
MasterDistanceInACC	主軸加速移動距離	LREAL	正数、または「0」	0	従軸加速時の主軸の移動距離を符号なしの絶対値表現で指定します。主軸の移動方向は正負とも有効です。 単位は［指令単位］となります。*4
MasterDistanceInDEC	主軸減速移動距離	LREAL	正数、または「0」	0	従軸減速時の主軸の移動距離を符号なしの絶対値表現で指定します。主軸の移動方向は正負とも有効です。 単位は［指令単位］となります。*4
LinkOption	同期開始条件選択	_eMC_LINKOPTION	0: _mcCommandExecution 1: _mcTriggerDetection 2: _mcMasterReach	0 *2	従軸が主軸に同期する条件を指定します。 0: 起動時 1: トリガを検出した時 2: 主軸が主軸追従距離に到達した時
MasterStartDistance	主軸追従距離	LREAL	負数、正数、「0」	0	従軸が主軸の追従を開始する絶対座標の主軸位置を指定します。 単位は［指令単位］となります。*4
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *2	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード

*1. _mcLatestCommand を使用する場合は、次の関係となるように主軸、従軸を使用しなければなりません。
「Master」（主軸）に設定されているモーション制御システム変数の軸番号 < 「Slave」（従軸）に設定されているモーション制御システム変数の軸番号
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*3. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。
*4. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
InSync	同期中	BOOL	TRUE, FALSE	同期動作を開始したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。

出力変数	名称	データ型	有効範囲	内容
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
InSync	同期開始条件が成立したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を受け付けたとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Master	主軸	_sAXIS_REF	−	主軸を指定します。*1
Slave	従軸	_sAXIS_REF	−	従軸を指定します。*1
TriggerInput	トリガ入力条件	_sTRIGGER_REF	−	トリガ条件を設定します。*2
TriggerVariable	トリガ変数	BOOL	TRUE, FALSE	トリガ条件でコントローラモードを指定したとき、トリガとなる入力変数を指定します。

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. _sTRIGGER_REF 型のユーザ定義変数を定義してください。

● _sTRIGGER_REF

メンバ変数	名称	データ型	有効範囲	機能
Mode	モード	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	トリガのモードを指定します。 0: ドライバモード 1: コントローラモード
LatchID	ラッチ ID 選択	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	[ドライバモード] のときに 2 つのラッチ機能のうちのどちらを使用するかを指定します。 0: ラッチ機能 1 1: ラッチ機能 2
InputDrive	ドライブトリガ入力信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	[ドライバモード] のときにサーボドライバのトリガ信号を指定します。 0: Z 相 1: 外部入力

機能説明

- 指定した主軸に同期して従軸が動作します。
- 電子カムの一種ですが、主軸に対して従軸は台形カーブで動作します。
- 本命令で動作中の軸を停止したい場合、MC_Stop（強制停止）命令を使用してください。

使用上の注意

ユニットバージョン Ver.1.09 以前の CPU ユニットで本命令を使用する場合は、「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行しないでください。
「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行すると、「Slave」（従軸）が急激な追従をすることがあり、危険です。
「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を使用する場合は、「Master」（主軸）と「Slave」（従軸）の関係をいったん解除してから実行してください。
主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。

● データオブジェクトのマッピング

「LinkOption」（同期開始条件選択）に［_mcTriggerDetection］を選択し、「Mode」に［ドライバモード］を選択して、MC_MoveLink（台形パターンカム）命令を使用する場合は、以下のオブジェクトデータをマッピングしてください。
マッピングは、Sysmac Studio の軸基本設定の［詳細設定］で行います。

- ラッチ機能(60B8Hex)
- ラッチステータス(60B9Hex)
- ラッチ 1 位置(60BAHex)
- ラッチ 2 位置(60BCHex)

これらを設定していない場合、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。

データオブジェクトのマッピングについては、「2-3 PDO マッピング (P.2-32)」、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。
NX シリーズ 位置インタフェースユニットを使用する場合は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』の「I/O エントリマッピング」を参照してください。

命令の詳細

以下に本命令の詳細について説明します。

● Master（主軸）指定

「Master」（主軸）で、主軸を指定します。

● Slave（従軸）指定

「Slave」（従軸）で、従軸を指定します。

● TriggerInput（トリガ入力条件）、TriggerVariable（トリガ変数）

同期開始条件指定が_mcTriggerDetectoin の時に、同期を開始する入力信号を指定します。
入力信号の選択、トリガが発生するタイミングの詳細については、「MC_TouchProbe (P.3-339)」を参照してください。
「TriggerInput」（トリガ入力条件）の「Mode」（モード）で［ドライバモード］を指定した場合は、従軸のドライバ入力を使用します。
コントローラモードを指定した場合は「TriggerVariable」（トリガ変数）をトリガ信号として使用します。

● ReferenceType（位置種別選択）

位置の種類を、以下から選択できます。

- _mcCommand: 指令位置（直前タスク周期での計算値）
  現周期に対して、直前のタスク周期で計算した主軸の指令位置を使用します。
  従軸の指令位置を計算する直前の定周期タスクで、計算した主軸の指令位置を使用します。
- _mcFeedback: 同一タスク周期での取得値
  同一タスク周期で取得した主軸のフィードバック位置を使用します。
- _mcLatestCommand: 指令位置（同一タスク周期での計算値）
  同一タスク周期で計算した主軸の指令位置を使用します。
  _mcCommand よりも新しい情報を使用することができます。ただし、主軸番号は従軸番号より若い番号を設定しなければなりません。
  従軸が、主軸より若い軸番号の場合、本命令の「Error」（エラー）が TRUE となります。「ErrorID」（エラーコード）には、「主軸／従軸 軸番号非昇順（エラーコード: 5438 Hex）」が出力されます。

使用上の注意

ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。同様に定周期タスクは、プライマリ定周期タスクまたは定周期タスク（実行優先度 5）を表します。

参考

同一タスク周期で計算した指令位置を選択した方が、直前タスク周期での計算値を選択した場合より、同期精度は向上します。
ただし、[「Master」（主軸）に設定されているモーション制御システム変数の軸番号 <「Slave」（従軸）に設定されているモーション制御システム変数の軸番号］となるように主軸、従軸を使用しなければなりません。

● 軸種別と位置種別の関係

監視可能な軸種別と監視する位置種別の関係は以下のようになります。

軸種別	ReferecneType	
	_mcCommand、または _mcLatestCommand	_mcFeedback
サーボ軸	○	○

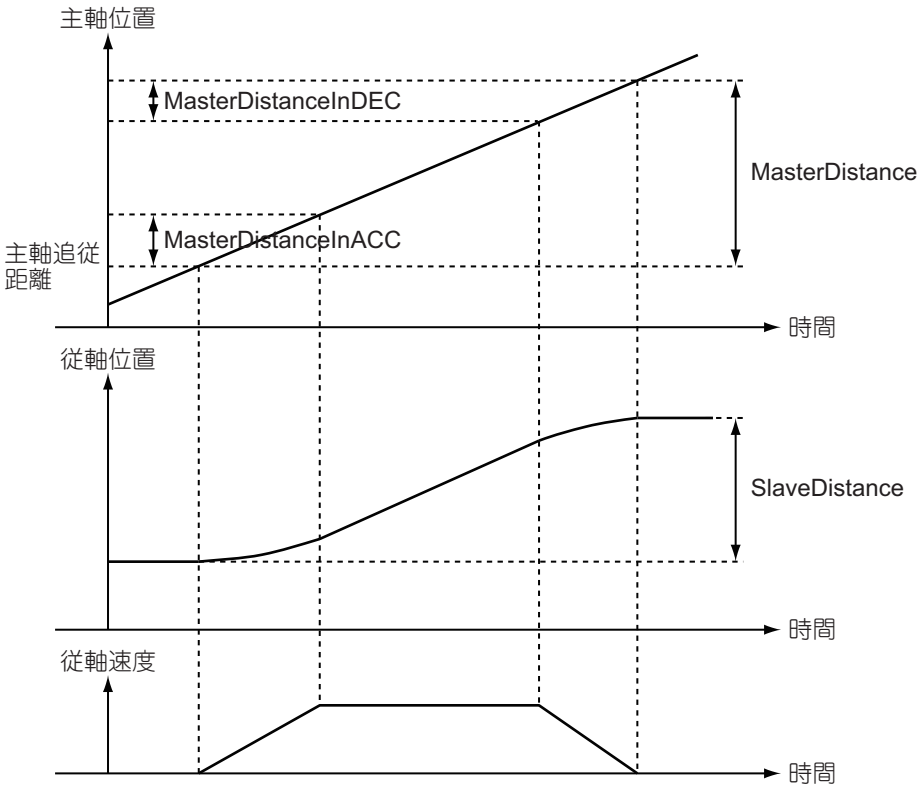
軸種別	ReferecneType	
	_mcCommand、または _mcLatestCommand	_mcFeedback
エンコーダ軸	× *1	○
仮想サーボ軸	○	○
仮想エンコーダ軸	× *1	○

*1. 命令起動時に、「位置種別選択範囲外（エラーコード: 5430 Hex）」が発生します。

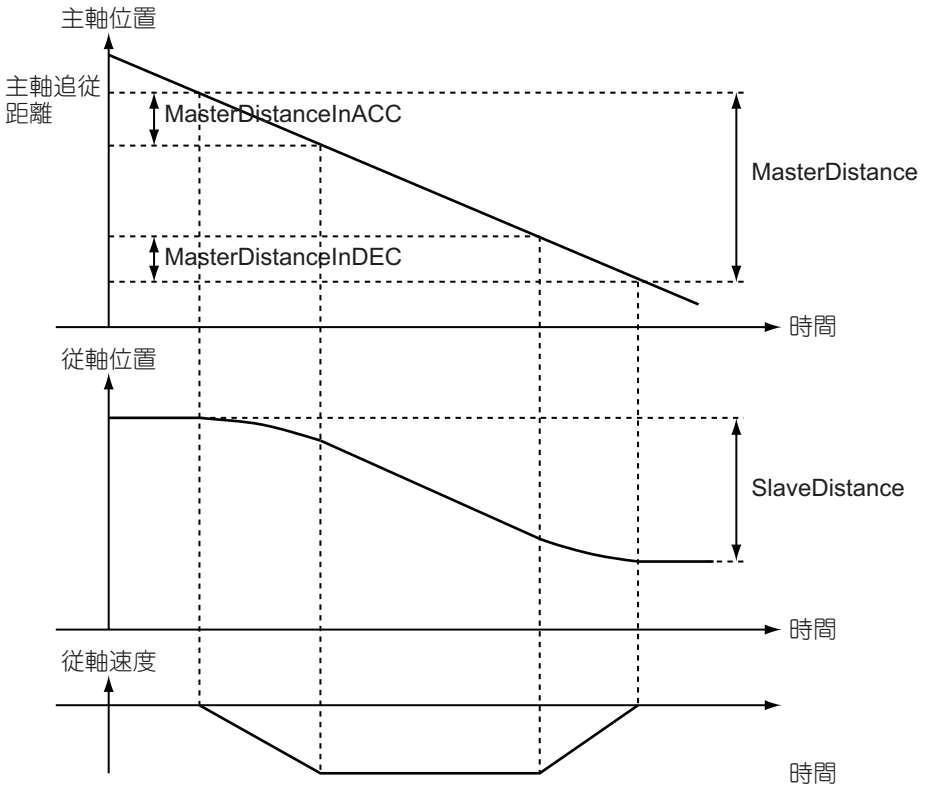
● SlaveDistance（従軸移動距離）、MasterDistance（主軸移動距離）、MasterDistanceInACC（主軸加速移動距離）、MasterDistanceInDEC（主軸減速移動距離）

従軸の速度と位置は、下図のように主軸と従軸の移動量の比で決まります。
下図の主軸追従距離は、同期開始条件が有効になった時の位置を示します。

主軸が正方向に移動する場合



主軸が負方向に移動する場合



主軸の移動量と従軸の移動量の関係は、次の表の通りとなります。

区間	主軸移動量と従軸移動量の関係	
加速	主軸	主軸加速移動距離
	従軸	従軸移動距離 × $\frac{\frac{\text{主軸加速移動距離}}{2}}{\frac{\text{主軸加速移動距離}}{2} + (\text{主軸移動距離} - \text{主軸加速移動距離} - \text{主軸減速移動距離}) + \frac{\text{主軸減速移動距離}}{2}}$
等速	主軸	主軸移動距離 − 主軸加速移動距離 − 主軸減速移動距離
	従軸	従軸移動距離 − 上記加速時の従軸移動距離 − 下記減速時の従軸移動距離
減速	主軸	主軸減速移動距離
	従軸	従軸移動距離 × $\frac{\frac{\text{主軸減速移動距離}}{2}}{\frac{\text{主軸加速移動距離}}{2} + (\text{主軸移動距離} - \text{主軸加速移動距離} - \text{主軸減速移動距離}) + \frac{\text{主軸減速移動距離}}{2}}$

主軸の等速区間が負数の場合、等速移動量異常が発生し停止します。
また、従軸を主軸と同一速度で送りたい場合、従軸の移動量には、次の値を指定してください。

$$\text{従軸移動距離} = \frac{\text{主軸加速移動距離}}{2} + (\text{主軸移動距離} - \text{主軸加速移動距離} - \text{主軸減速移動距離}) + \frac{\text{主軸減速移動距離}}{2}$$

使用上の注意

主軸のカウンタモードが［ロータリモード］の場合、「MasterDistance」（主軸移動距離）には主軸のリングカウンタ 1 周以内の値を指定してください。

● LinkOption（同期開始条件選択）

従軸が主軸に同期する条件を指定します。

- 命令開始時
  本命令起動時、次の周期より従軸が主軸に同期して位置決めします。
- トリガを検出した時
  入力トリガ指定で指定した入力信号が有効になった次の周期より、従軸が主軸に同期して位置決めします。
- 主軸が主軸追従距離に到達した時
  本命令が起動中に、主軸が主軸追従距離に到達した時点で、次の周期より従軸が同期して位置決めします。
  また、主軸が主軸追従距離に停止している状態で本命令が起動された時点でも、次の周期より従軸が同期して位置決めします。

使用上の注意

「LinkOption」（同期開始条件選択）に［_mcTriggerDetection］を選択し、「Mode」に［ドライバモード］を選択した場合、オブジェクトデータのマッピングが必要となります。
下記に示すオブジェクトの設定をしてください。
- ラッチ機能(60B8Hex)
- ラッチステータス(60B9Hex)
- ラッチ 1 位置(60BAHex)
- ラッチ 2 位置(60BCHex)

これらを設定していない場合、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。
オブジェクトデータのマッピングについては、「2-3 PDO マッピング (P.2-32)」、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● MasterStartDistance（主軸追従距離）

従軸が同期を開始する主軸の位置を、絶対位置で指定します。

ユニットバージョン Ver.1.10 以降の CPU ユニットの場合

主軸カウントモードが［ロータリモード］の場合、リングカウンタ上下限設定値の範囲を超えた「MasterStartDistance」（主軸追従距離）の指定が可能です。この場合、主軸現在位置と主軸追従距離の関係は、MC_MoveAbsolute（絶対値位置決め）命令で「Direction」（方向選択）に［方向指定なし］を指定した場合と同じになります。
MC_MoveAbsolute（絶対値位置決め）命令は、「MC_MoveAbsolute (P.3-47)」を参照してください。

ユニットバージョン Ver.1.09 以前の CPU ユニットの場合

主軸カウントモードが［ロータリモード］の場合、「MasterStartDistance」（主軸追従距離）は、リングカウンタ上下限設定値の範囲内にしてください。範囲外を指定した場合は、命令実行時に「カム動作（主軸追従）開始位置設定範囲外（エラーコード: 547B Hex)」を出力します。

● BufferMode（バッファモード選択）

1 つ前の軸動作と今回の動作のつなぎ方を指定します。
次の 2 種類があります。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。
バッファード	現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

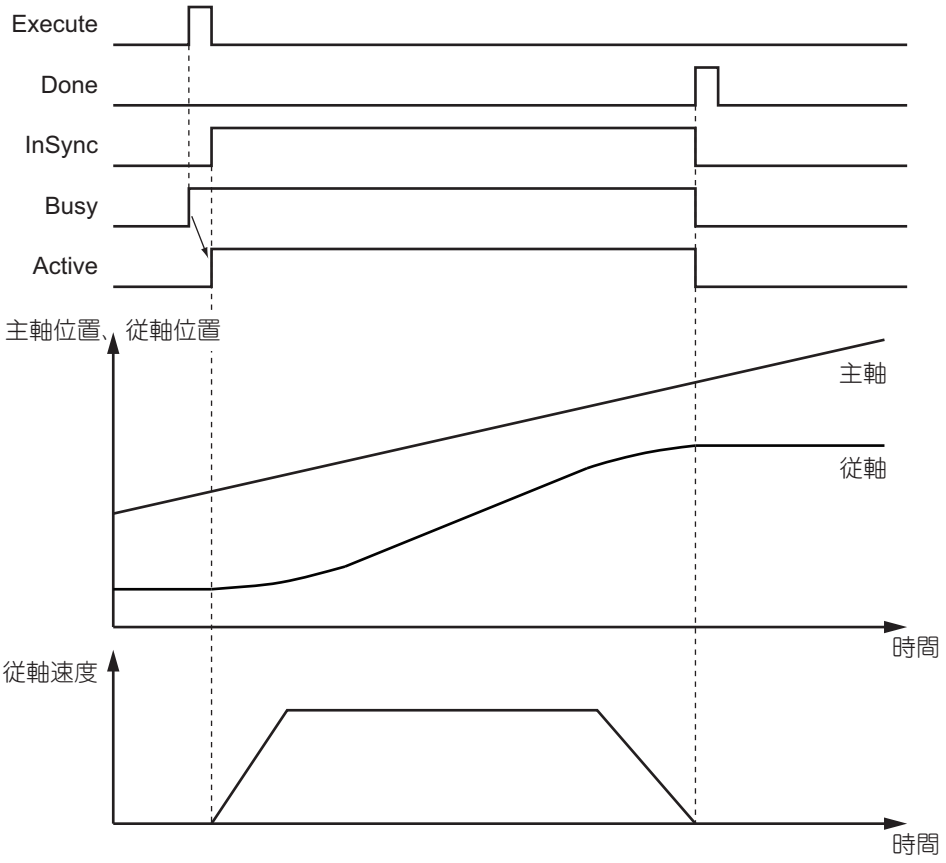
● インポジションチェック

従軸の動作が完了した時点でインポジションチェックを実行します。
本命令のインポジションチェックは軸パラメータの［インポジション幅］、［インポジションチェック時間］の設定に従って行われます。

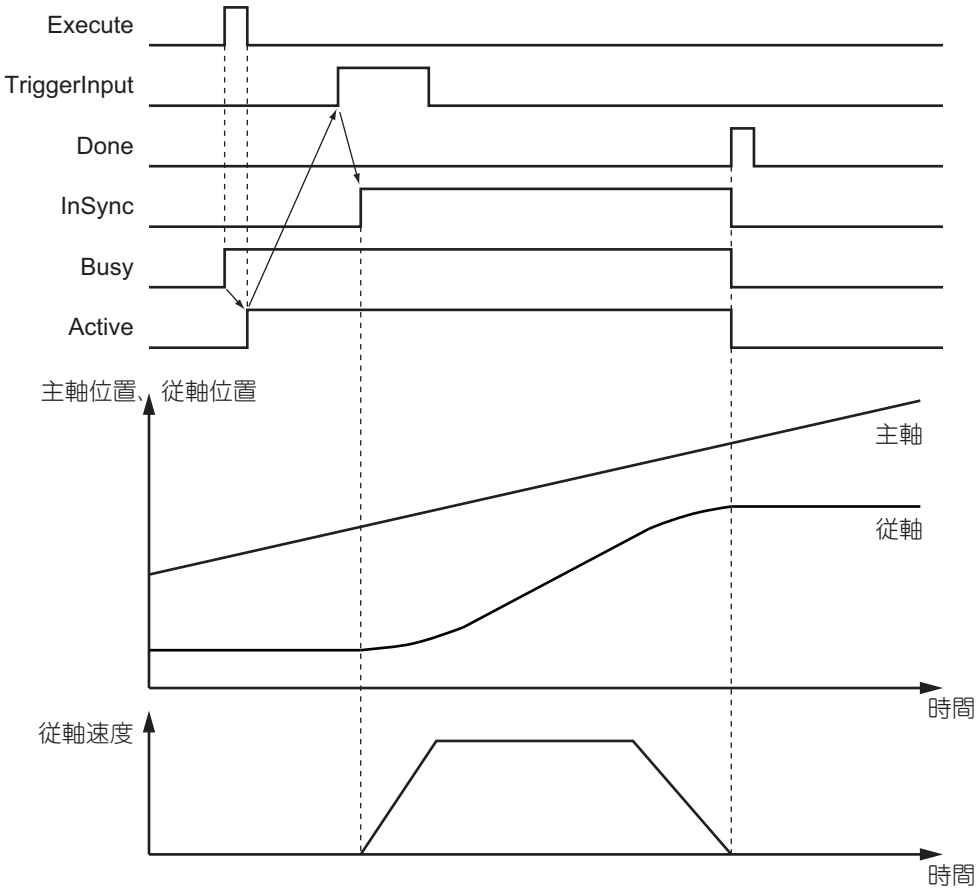
タイミングチャート

● 本命令を起動する場合

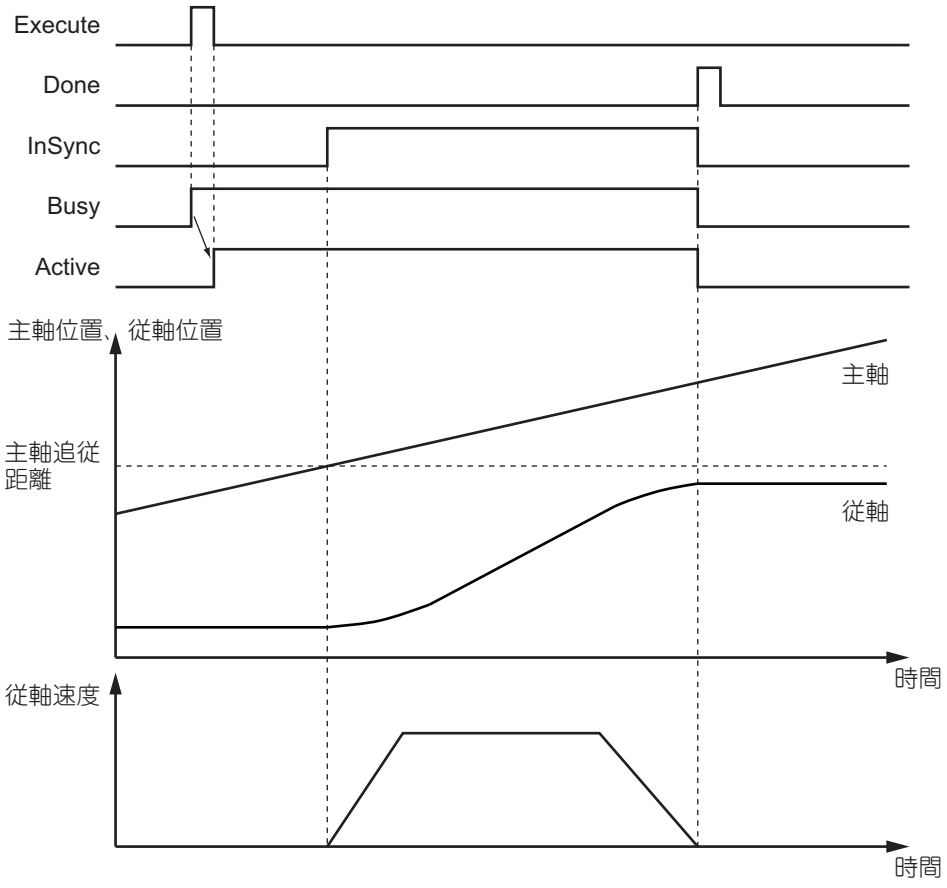
「LinkOption」（同期開始条件選択）が［0: 起動時］の場合



「LinkOption」（同期開始条件選択）が［1: トリガを検出した時］の場合



「LinkOption」（同期開始条件選択）が［2: 主軸が主軸追従距離に到達した時］の場合



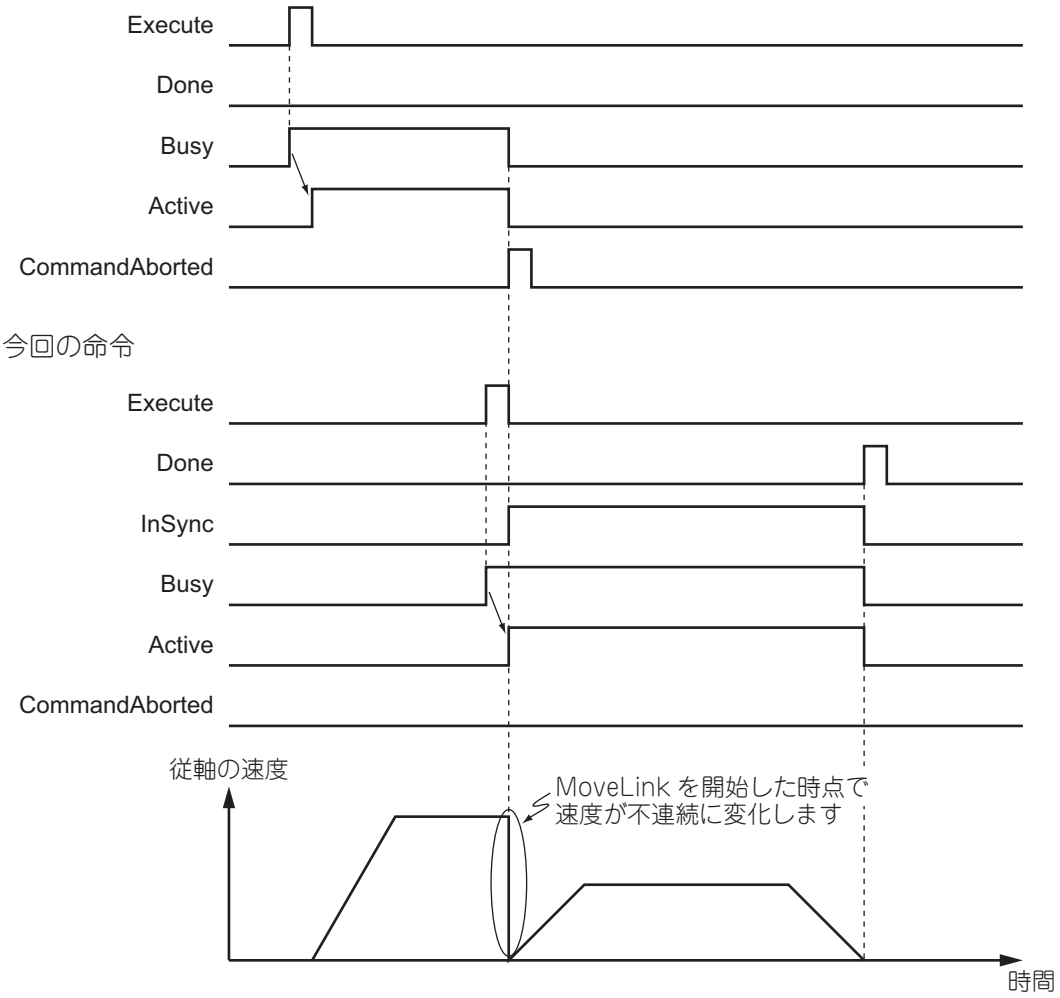
● 1 つ前の動作中に「BufferMode」（バッファモード選択）を［アボーティング］で起動する場合

1 つ前の動作が MC_MoveAbsolute（絶対値位置決め）命令を実行中に、「LinkOption」（同期開始条件選択）に［0: 起動時］を指定して本命令を実行した場合を説明します。

使用上の注意

下図に示すように、本命令開始時点で従軸の速度が不連続になりますので、注意してください。

直前の命令が MC_MoveAbsolute（絶対値位置決め）の場合

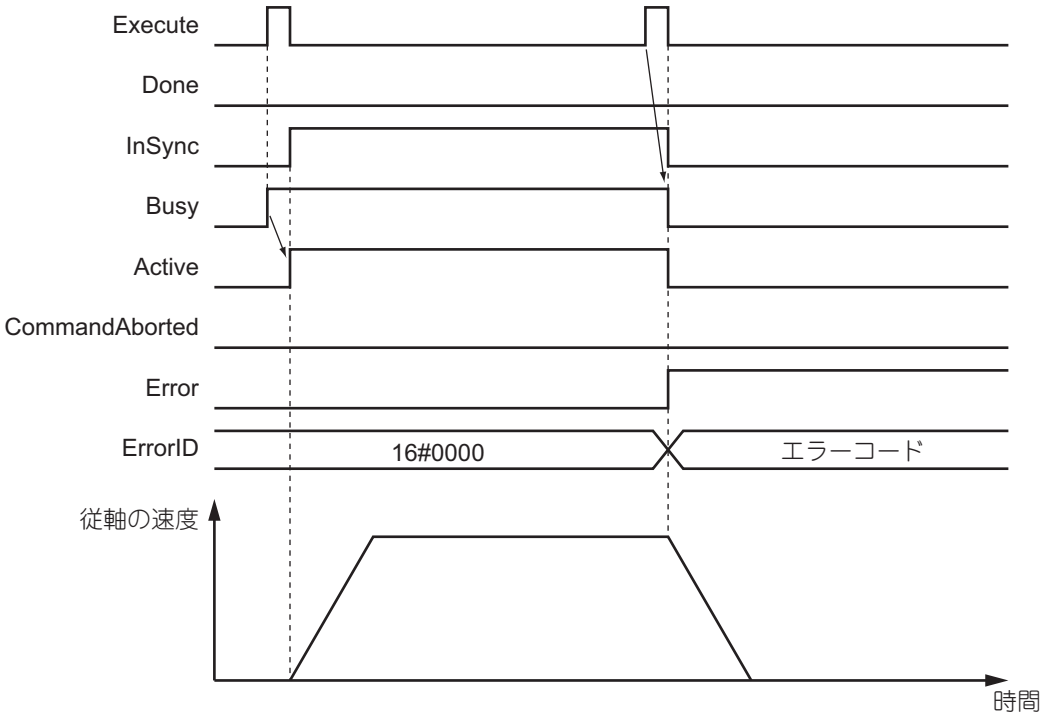


● 1 つ前の動作中に「BufferMode」（バッファモード選択）を［バッファード］で起動する場合

1 つ前の動作が完了してから、本命令を実行します。

モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となり、この軸の動作が停止します。



モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

本命令に対するモーション命令多重起動を行う場合、［アボーティング］、［バッファード］が可能です。
ブレンディングによる命令多重起動はできません。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

異常発生後のタイミングチャートは、「モーション命令再起動 (P.3-285)」を参照してください。

● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、裁断の制御についてサンプルプログラムを掲載します。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

軸種別

軸	軸種別
軸 1	サーボ軸（主軸）
軸 2	サーボ軸（従軸）

カウンタモード

軸	カウンタモード
軸 1	ロータリモード
軸 2	リニアモード

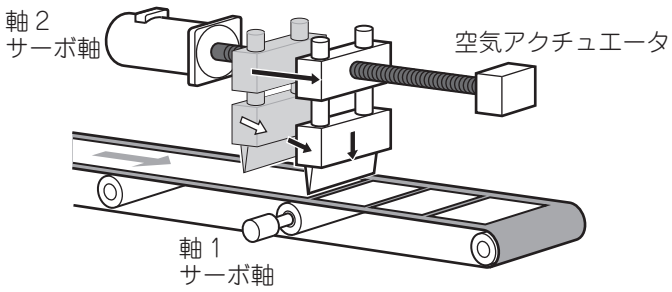
リングカウンタ

軸	上限値	下限値
軸 1	360	0

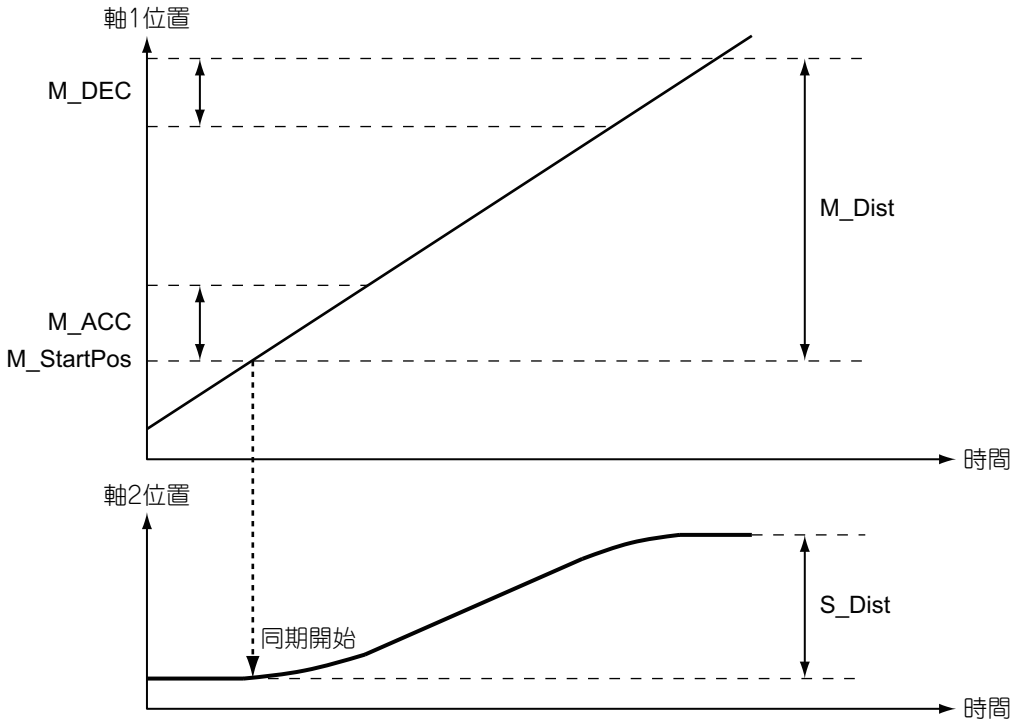
座標単位選択

軸	座標単位
軸 1	degree
軸 2	mm

動作例



● **動作パターン**



1 **主軸の開始**
ベルトコンベアに取り付けられた軸 1 を主軸として位置をフィードバックします。

2 **従軸の起動**
水平方向に動作するボールねじに取り付けられた軸 2 が軸 1 に同期して動作します。

3 **カッターによる裁断**
軸 2 が同期中に空気アクチュエータが「ON」となり、空気アクチュエータに接続されたカッターが垂直方向に降下してワークの裁断を行います。

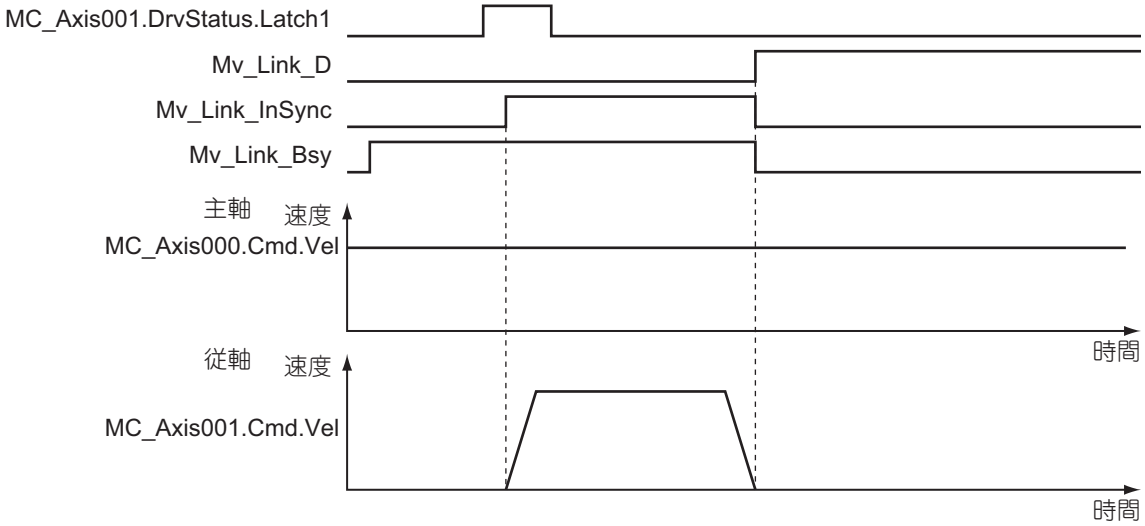
ラダー図

● **主な変数**

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態のとき、TRUE となります。
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。

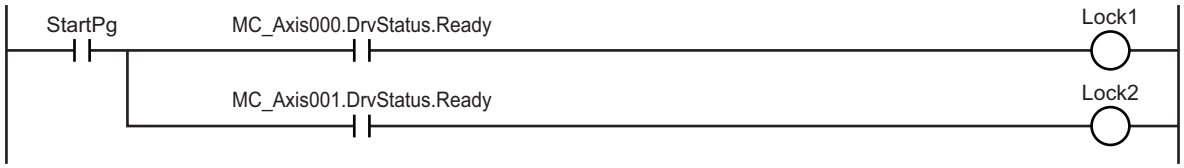
名称	データ型	初期値	コメント
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Actuator	BOOL	FALSE	軸 1 と軸 2 が同期すると、TRUE となります。「Actuator」が TRUE のとき、カッターが垂直方向に降下します。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート

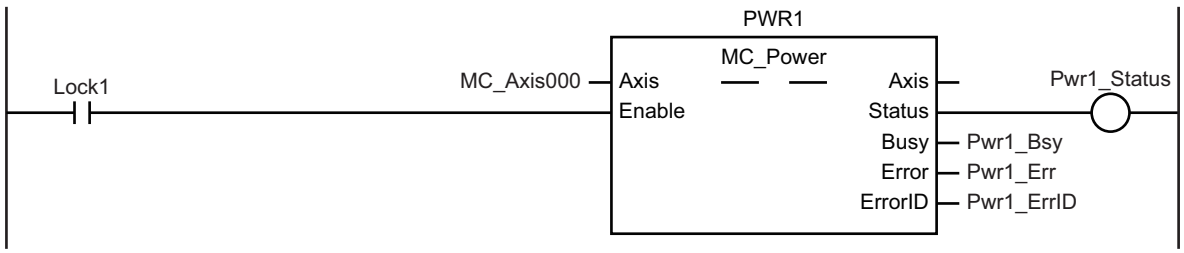


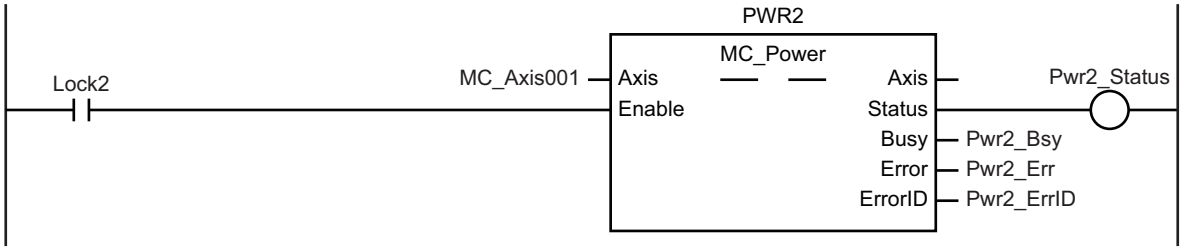
● サンプルプログラム

接点 StartPg が TRUE になったら、各軸のサーボドライバがサーボレディ状態になっていることを確認する

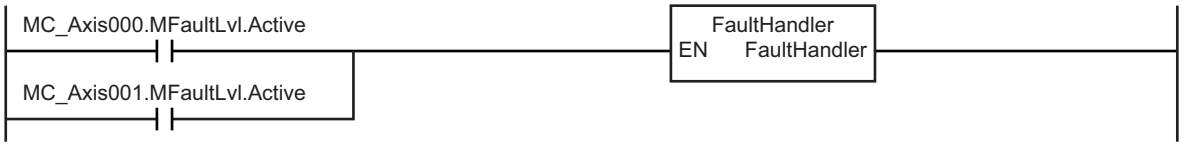


サーボレディ状態になっていれば、各軸をサーボ ON 状態にする

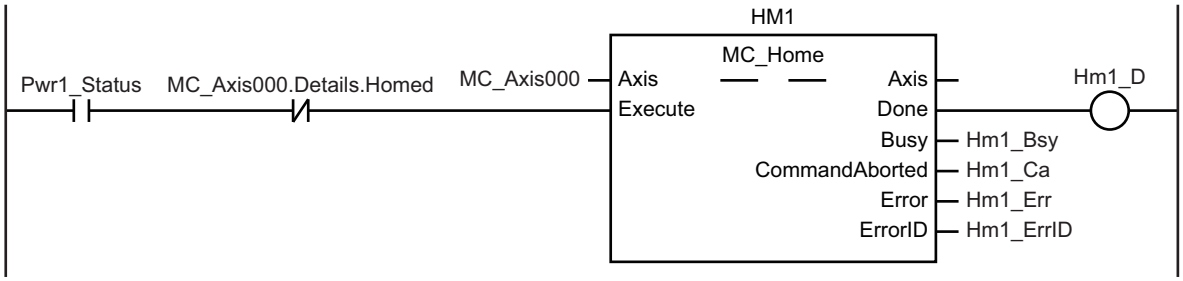




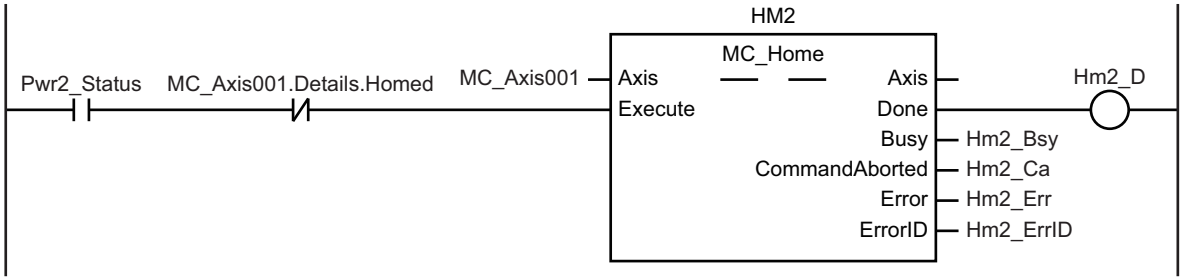
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



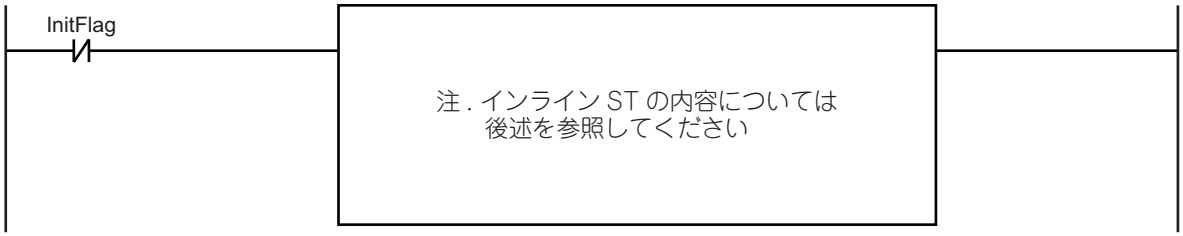
軸 1 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



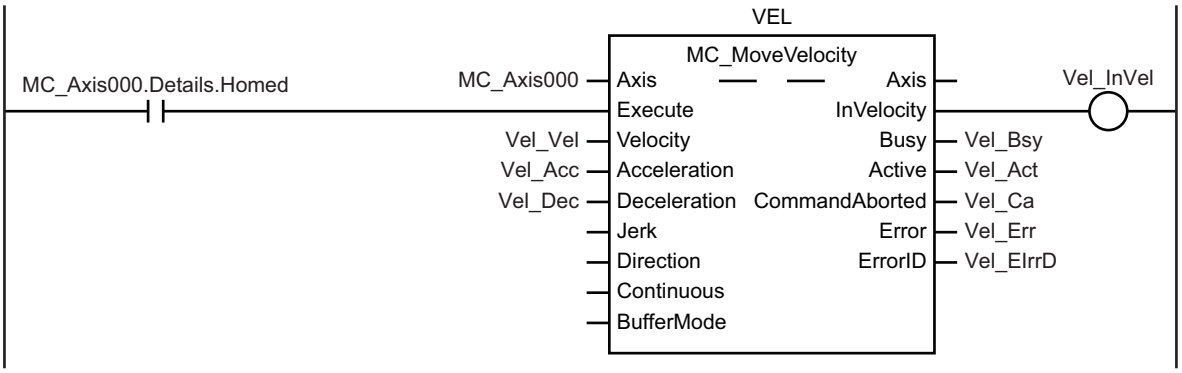
軸 2 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



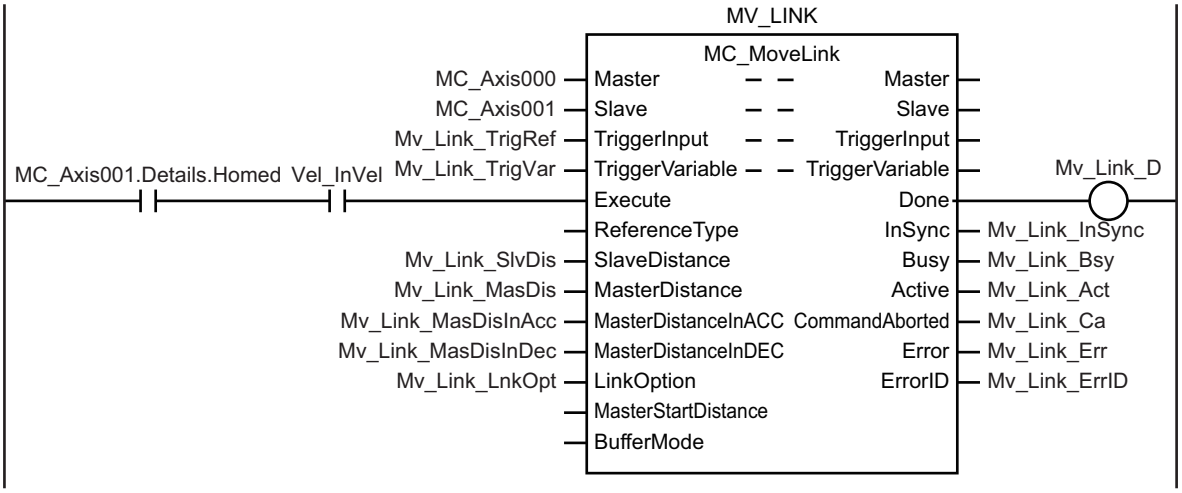
MC_MoveVelocity（速度制御）命令と MC_MoveLink（台形パターンカム）命令のパラメータを設定する



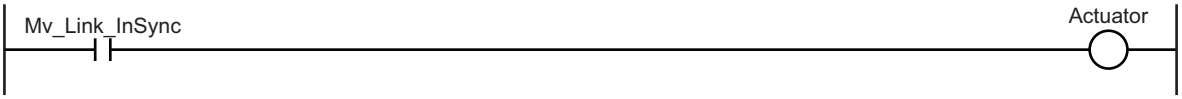
軸 1 が原点確定状態になったら、MC_MoveVelocity(速度制御）命令を実行する



従軸（軸 2）が原点確定状態になったら、MC_MoveLink（台形パターンカム）命令を実行する



同期中は Actuator を ON する



インライン ST の内容

```
//MC_MoveVelocity パラメータ
Vel_Vel := LREAL#1000.0;
Vel_Acc := LREAL#0.0;
Vel_Dec := LREAL#0.0;

//MC_MoveLink パラメータ
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;
```

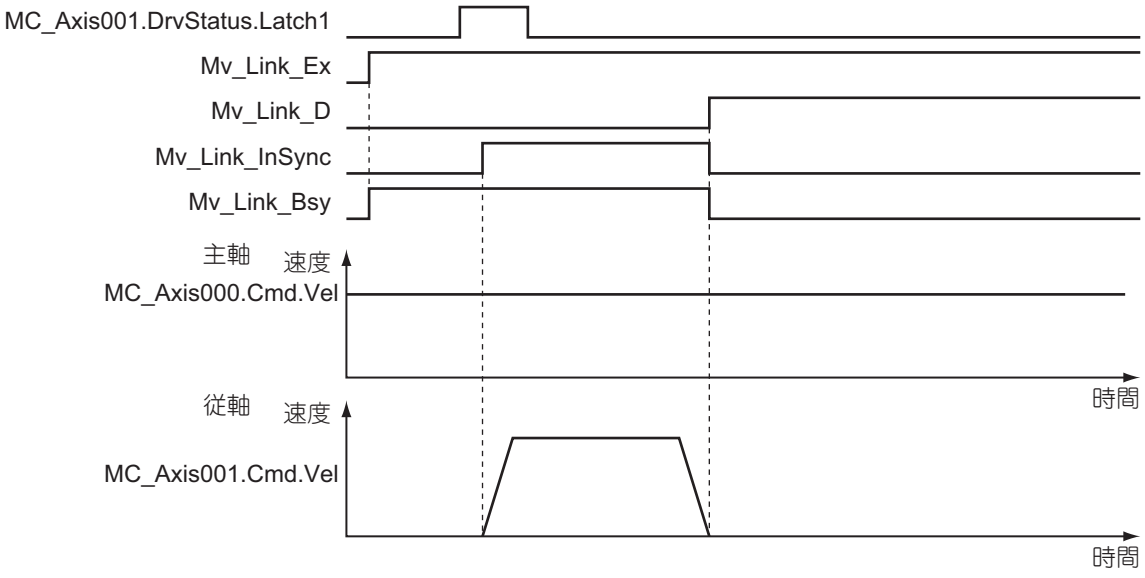
```
//入力パラメータを設定後、InitFlag を TRUE にします
InitFlag := TRUE;
```

ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態のとき、TRUE となります。
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
Actuator	BOOL	FALSE	軸 1 と軸 2 が同期すると、TRUE となります。「Actuator」が TRUE のとき、カッターが垂直方向に降下します。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。FALSE のとき、入力パラメータを設定します。入力パラメータの設定が完了すると、TRUE となります。

● タイミングチャート



● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag=FALSE THEN

    //MC_MoveVelocity パラメータ
    Vel_Vel := LREAL#1000.0;
    Vel_Acc := LREAL#0.0;
    Vel_Dec := LREAL#0.0;

    //MC_MoveLink パラメータ
    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;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag := TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE; //軸 1 をサーボ ON にします
ELSE
    Pwr1_En:=FALSE; //軸 1 をサーボ OFF にします
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 2 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE; //軸 2 をサーボ ON にします
ELSE
    Pwr2_En:=FALSE; //軸 2 をサーボ OFF にします
END_IF;

//軽度フォールトレベルの異常発生時の処理
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE)THEN
    FaultHandler();
END_IF;
```

```
//軸 1 がサーボ ON 状態で、かつ原点未確定状態のとき、原点復帰を実行します
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//軸 2 がサーボ ON 状態で、かつ原点未確定状態のとき、原点復帰を実行します
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//軸 1 が原点確定後に、軸 1 で MC_MoveVelocity を実行します
IF MC_Axis000.Details.Homed=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

//軸 2 が原点確定状態で、かつ軸 1 が目標速度に到達していることを確認し、軸 2 で MC_MoveLink を実行します
IF (MC_Axis001.Details.Homed=TRUE) AND (Vel_InVel=TRUE) THEN
    Mv_Link_Ex:=TRUE;
END_IF;

//軸 1 と軸 2 が同期すると、アクチュエータが ON します
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 => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

// MC_Home2
HM2(
    Axis := MC_Axis001,
    Execute := Hm2_Ex,
    Done => Hm2_D,
    Busy => Hm2_Bsy,
    CommandAborted => Hm2_Ca,
    Error => Hm2_Err,
    ErrorID => Hm2_ErrID
);

// MC_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

2 軸の位置を加算した値、または減算した値を指令位置として出力します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_CombineAxes	加減算位置決め	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 :=「パラメータ」, Auxiliary :=「パラメータ」, Slave :=「パラメータ」, Execute :=「パラメータ」, CombineMode :=「パラメータ」, RatioNumeratorMaster :=「パラメータ」, RatioDenominatorMaster :=「パラメータ」, RatioNumeratorAuxiliary :=「パラメータ」, RatioDenominatorAuxiliary :=「パラメータ」, ReferenceTypeMaster :=「パラメータ」, ReferenceTypeAuxiliary :=「パラメータ」, BufferMode :=「パラメータ」, InCombination =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
CombineMode	加減算方法選択	_eMC_COMBINE_MODE	0: _mcAddAxes 1: _mcSubAxes	0 *1	合成方法を指定します。 0: 加算 1: 減算
RatioNumeratorMaster (Reserved)	主軸ギア比分子	DINT *2	正数または負数 *2	10000	(Reserved)
RatioDenominatorMaster (Reserved)	主軸ギア比分母	UDINT *3	正数	10000	(Reserved)
RatioNumeratorAuxiliary (Reserved)	補助軸ギア比分子	DINT *2	正数または負数 *2	10000	(Reserved)

入力変数	名称	データ型	有効範囲	初期値	内容
RatioDenominatorAuxiliary (Reserved)	補助軸ギア比分母	UDINT *3	正数	10000	(Reserved)
ReferenceTypeMaster	主軸位置種別選択	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 *1	主軸の位置種別を指定します。 1: フィードバック位置（同一タスク周期 *4 での取得値） 2: 指令位置（同一タスク周期 *4 での計算値）
ReferenceTypeAuxiliary	補助軸位置種別選択	_eMC_REFERENCE_TYPE	1: _mcFeedback 2: _mcLatestCommand	2 *1	補助軸の位置種別を指定します。 1: フィードバック位置（同一タスク周期 *4 での取得値） 2: 指令位置（同一タスク周期 *4 での計算値）
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *1	モーション命令多重起動時の動作を指定します。 0: アボーティング

*1. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*2. ユニットバージョン Ver.1.02 以降の CPU ユニットと Sysmac Studio Ver.1.03 以降の組み合わせで使用可能です。それ以前の組み合わせでは、データ型は「UINT」、有効範囲「正数」となります。
*3. ユニットバージョン Ver.1.02 以降の CPU ユニットと Sysmac Studio Ver.1.03 以降の組み合わせで使用可能です。それ以前の組み合わせでは、データ型は「UINT」となります。
*4. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
InCombination	加減算中	BOOL	TRUE, FALSE	加減算中に TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
InCombination	加減算位置決めを開始したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Busy	「Execute」の立ち上がり	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Master	主軸	_sAXIS_REF	−	主軸を指定します。*1
Auxiliary	補助軸	_sAXIS_REF	−	補助軸を指定します。*1
Slave	従軸	_sAXIS_REF	−	従軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

使用上の注意

主軸および従軸、補助軸のいずれかに同じ軸を指定すると、以下の軽度フォールトレベルの異常となります。
•「主軸従軸同一（エラーコード: 5436 Hex）」異常
•「主軸補助軸同一（エラーコード: 5437 Hex）」異常
•「補助軸従軸同一（エラーコード: 548E Hex）」異常

機能説明

• 本命令は、「Execute」（起動）の立ち上がりで加減算位置決め動作を開始します。

使用上の注意

ユニットバージョン Ver.1.09 以前の CPU ユニットで本命令を使用する場合は、「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行しないでください。「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を実行すると、「Slave」（従軸）が急激な追従をすることがあり、危険です。
「Master」（主軸）に対して MC_SetPosition（現在位置変更）命令を使用する場合は、「Master」（主軸）と「Slave」（従軸）の関係をいったん解除してから実行してください。
主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。
なお、「Auxiliary」（補助軸）についても「Master」（主軸）と同様です。

命令の詳細

開始時点からの「Master」（主軸）の位置の相対値と「Auxiliary」（補助軸）の位置の相対値を加算または、減算した相対値を「Slave」（従軸）の指令位置として出力します。

• 位置の加算または減算については、軸パラメータの軸の座標単位選択に関係なく、数値として加算または減算します。

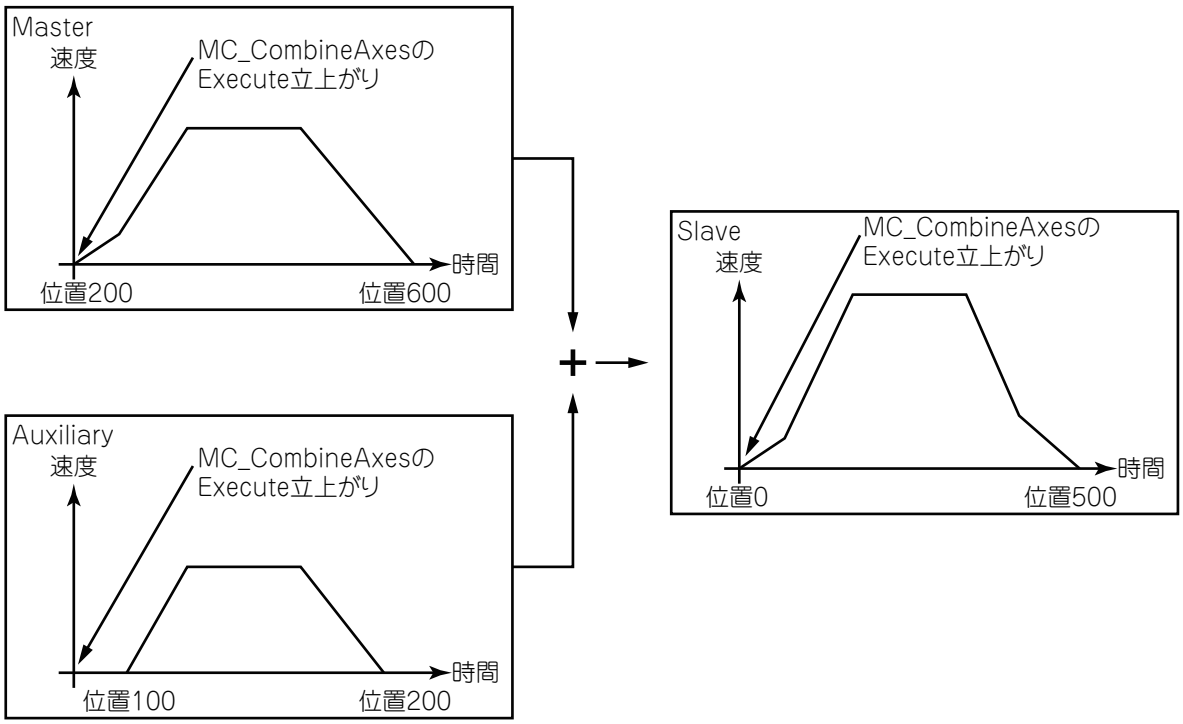
・本命令を終了するためには、MC_Stop（強制停止）命令を起動してください。

使用上の注意

「Master」（主軸）と「Auxiliary」（補助軸）の値によっては、「Slave」（従軸）の移動量、速度、加速度が急激に変化する場合があります。
十分に注意して使用してください。

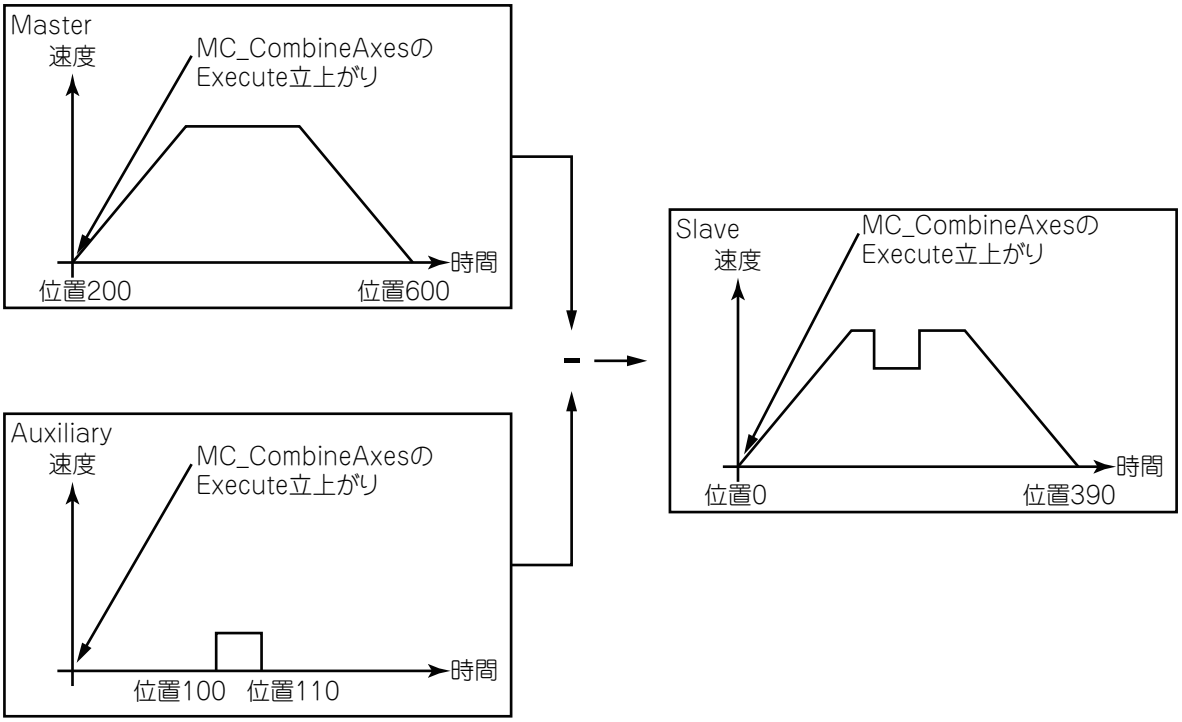
●「CombineMode」（加減算方法選択）が［0: 加算］の場合

「Slave」（従軸）位置 ＝ 「Master」（主軸）位置（相対位置） ＋ 「Auxiliary」（補助軸）位置（相対位置）



●「CombineMode」（加減算方法選択）が［1: 減算］の場合

「Slave」（従軸）位置 ＝ 「Master」（主軸）位置（相対位置） － 「Auxiliary」（補助軸）位置（相対位置）



● インポジションチェック

本命令では、インポジションチェックは行いません。

● オーバーライド

本命令では、MC_SetOverride（オーバーライド値設定）命令によるオーバーライドはできません。

● ReferenceType（位置種別選択）

位置の種類を、以下から選択できます。

- _mcFeedback: 同一タスク周期での取得値
  同一タスク周期で取得した主軸のフィードバック位置を使用します。
- _mcLatestCommand: 指令位置（同一タスク周期での計算値）
  同一タスク周期で計算した主軸の指令位置を使用します。
  _mcLatestCommand を選択する場合、主軸番号と補助軸番号は従軸番号より若い番号を設定しなければなりません。
  従軸が、主軸や補助軸より若い軸番号の場合、本命令の「Error」（エラー）が TRUE となります。「ErrorID」（エラーコード）には、「主軸／従軸 軸番号非昇順（エラーコード: 5438 Hex）」が出力されます。
  なお、主軸と補助軸の間には、軸番号に関する制約はありません。

使用上の注意

ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。同様に定周期タスクは、プライマリ定周期タスクまたは定周期タスク（実行優先度 5）を表します。

● 軸種別と位置種別の関係

監視可能な軸種別と監視する位置種別の関係は以下のようになります。

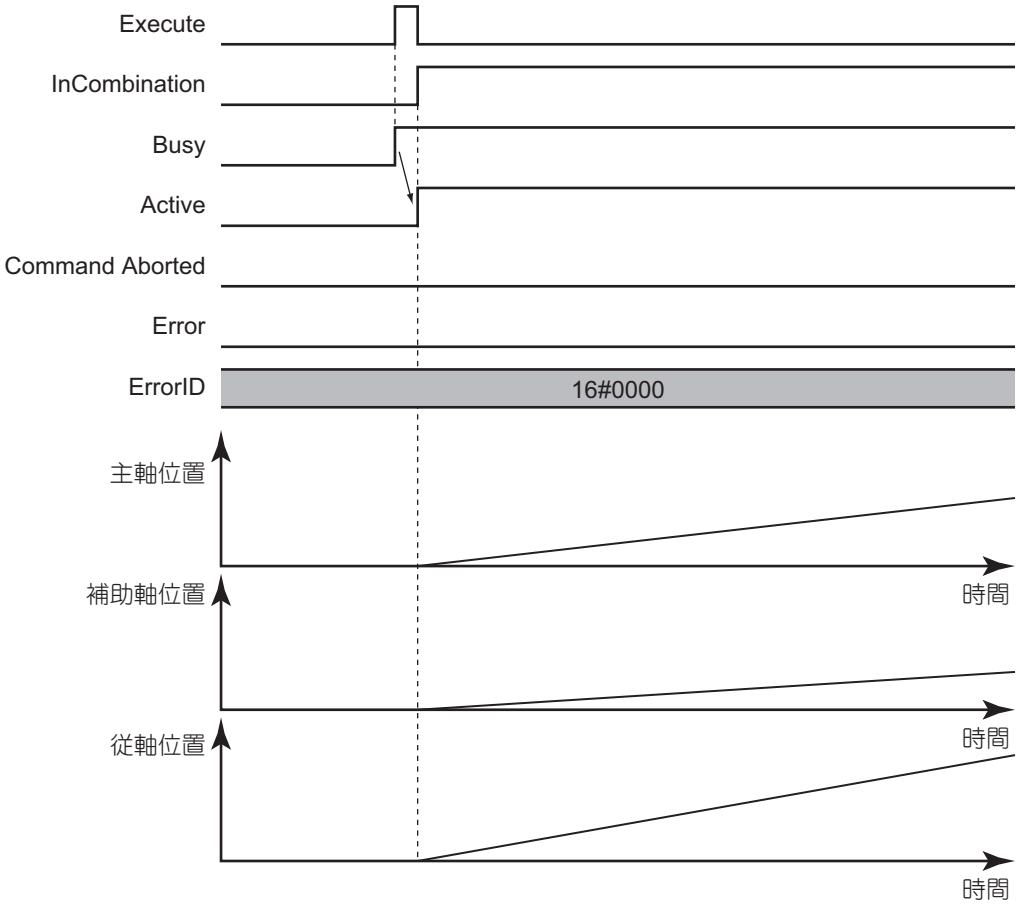
軸種別	ReferenceType	
	_mcFeedback	_mcLatestCommand
サーボ軸	○	○
エンコーダ軸	○	× *1
仮想サーボ軸	○	○
仮想エンコーダ軸	○	× *1

*1. 命令起動時に、「位置種別選択範囲外（エラーコード: 5430 Hex）」が発生します。

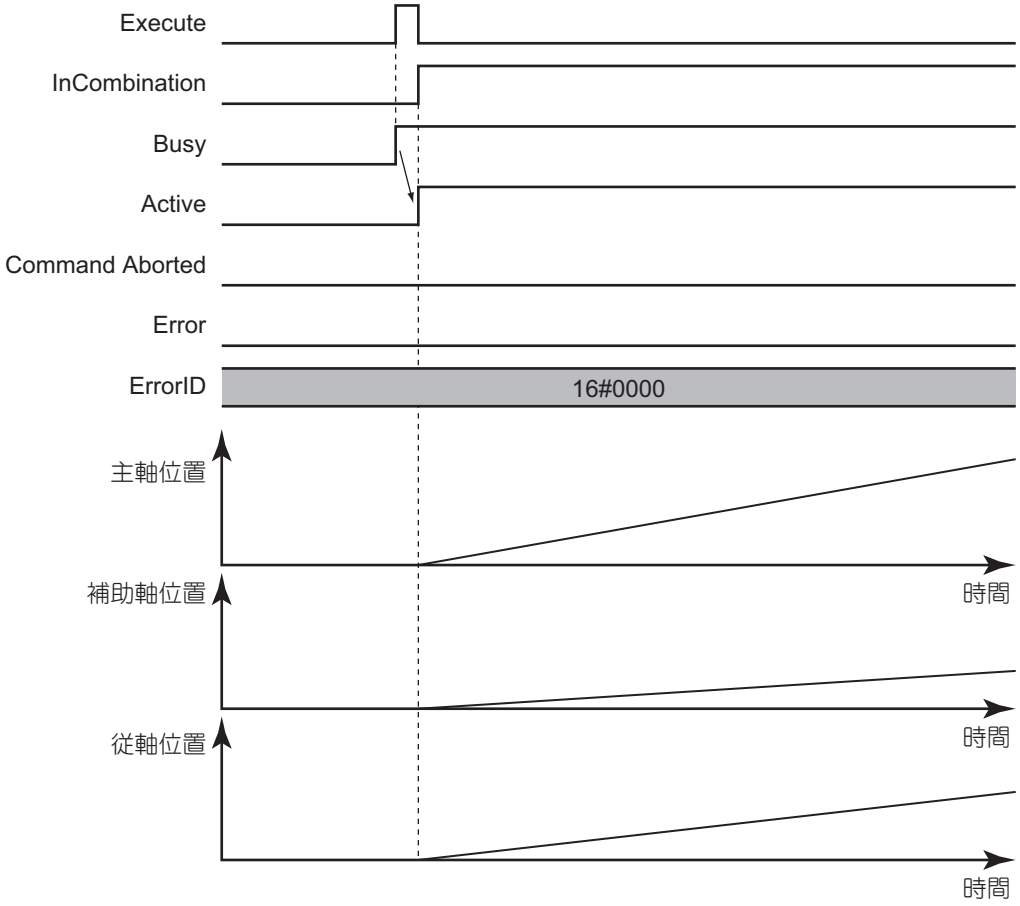
タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- 加減算出力を開始した周期で「InCombination」（加減算中）が TRUE となります。
- ほかの命令により、本命令が中断された場合は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）、「InCombination」（加減算中）が FALSE となります。

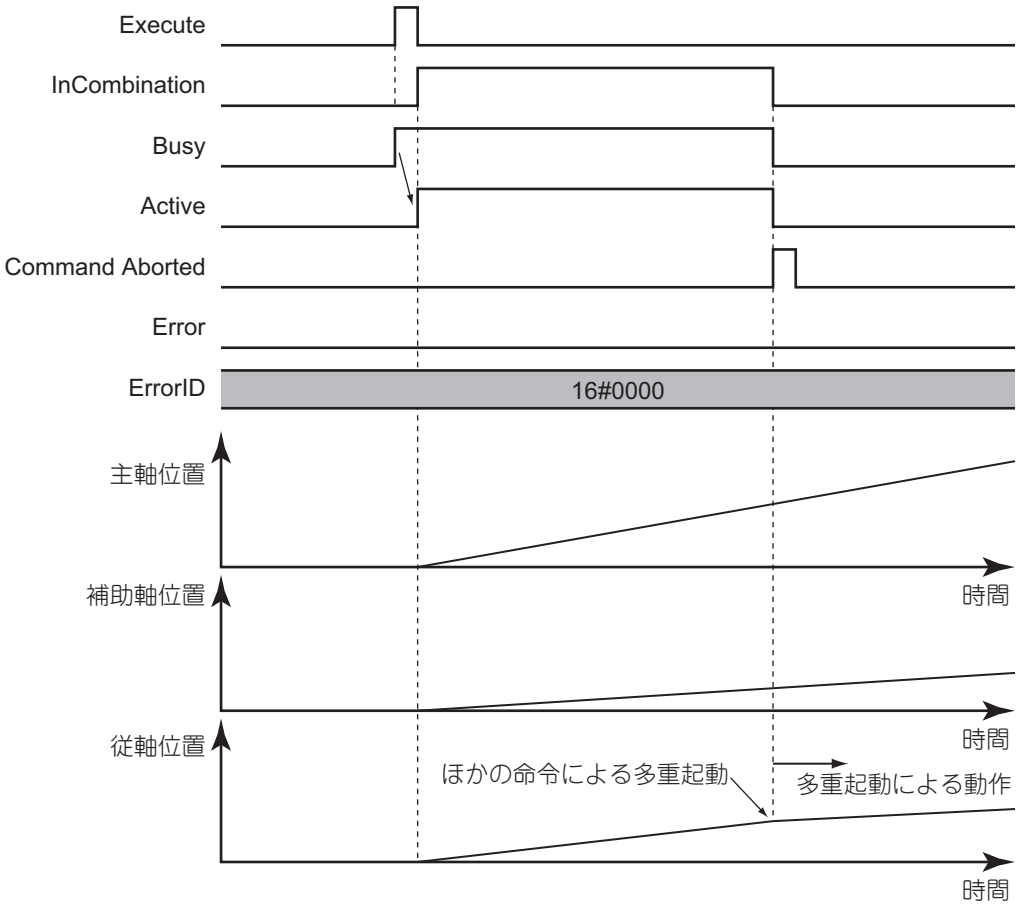
●「CombineMode」（加減算方法選択）が［0: 加算］の場合



● **「CombineMode」（加減算方法選択）が［1: 減算］の場合**



● 中断された場合



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex)」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

本命令に対する命令多重起動を行う場合、従軸を指定してください。
なお、本命令動作中にほかの命令を多重起動する場合、「BufferMode」（バッファモード選択）により以下の制約があります。

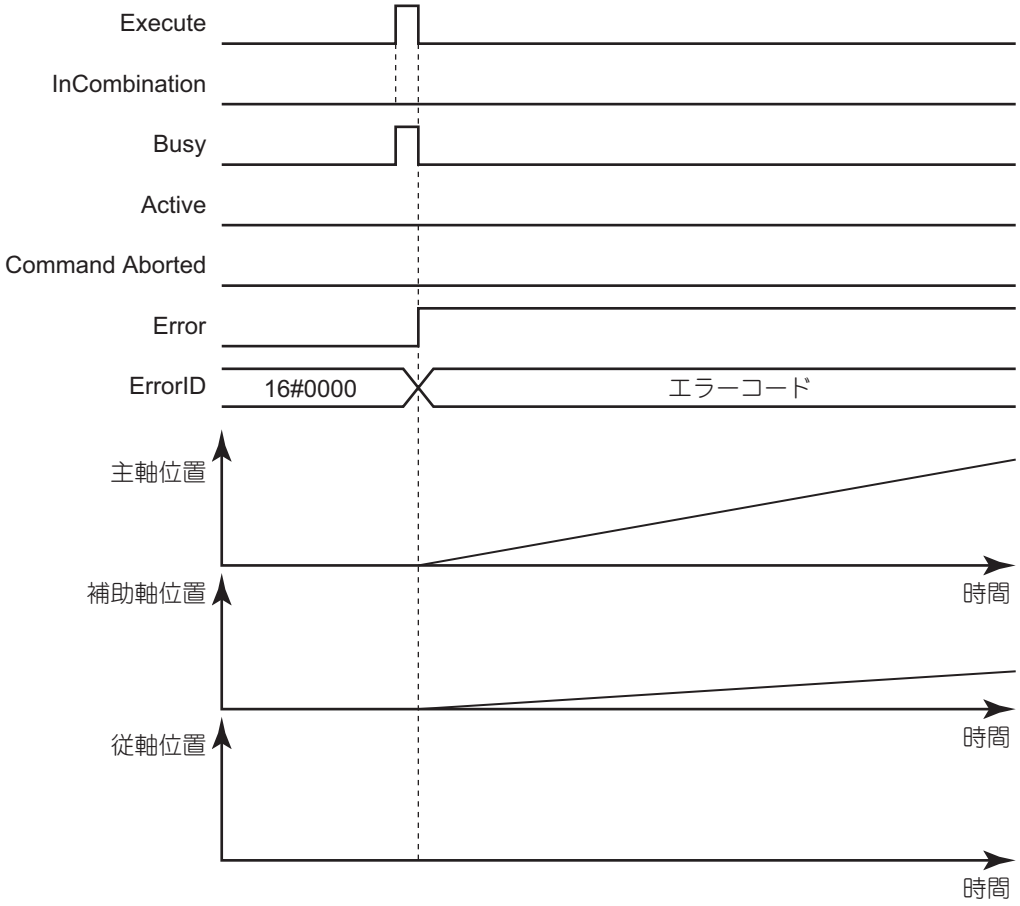
- 本命令実行中の他命令による多重起動は、［アボーティング］が可能です。
  ［バッファード］、［ブレンディング］による命令多重起動はできません。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。

「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

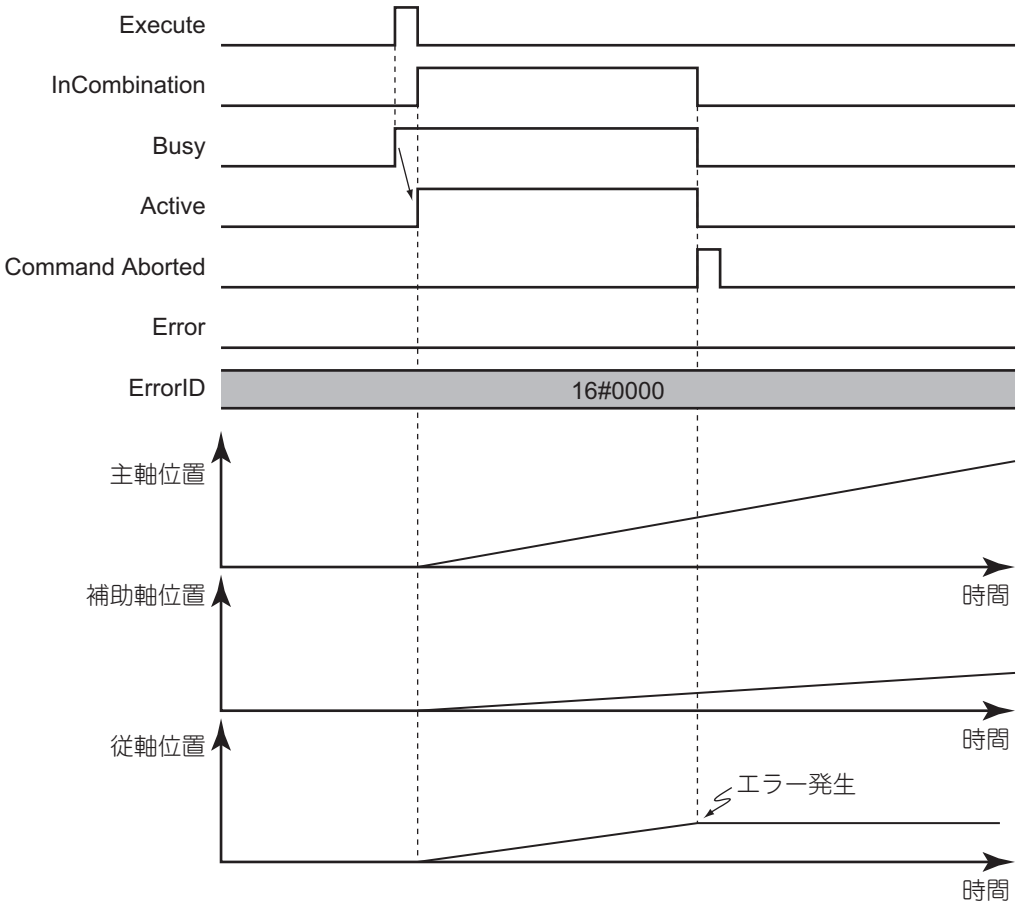
● 異常発生時のタイミングチャート



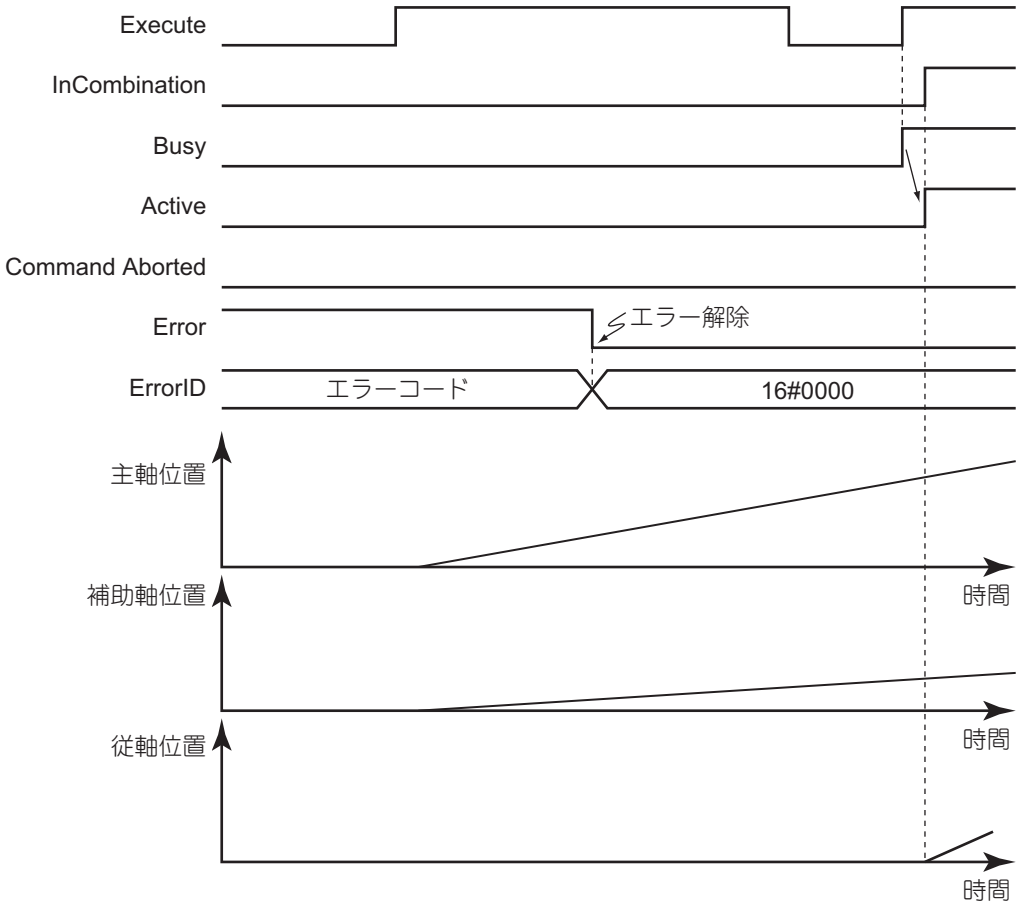
参考

- 本命令は、「Master」（主軸）、「Auxiliary」（補助軸）の異常状態には影響を受けません。
- 異常状態が解除され、「Master」（主軸）、「Auxiliary」（補助軸）の動作時には、「Slave」（従軸）は加減算位置決め動作を継続します。また、本命令起動時および本命令実行中に従軸に異常が発生した場合、本命令は中断しますが、「Master」（主軸）、「Auxiliary」（補助軸）に影響を与えることはありません。

本命令実行中に、軽度フォールトレベルの異常が発生した場合、「CommandAborted」（実行中断）がTRUE となり、軸が停止します。
減速停止する際の減速度は、軸パラメータの［最大減速度］です。
「Slave」（従軸）の軸変数「MFalutLvl.Code」に出力された値を参照することにより、異常の原因を知ることができます。



本命令の異常解除を行った場合、次の「Execute」（起動）の立ち上がりまで本命令は起動しません。

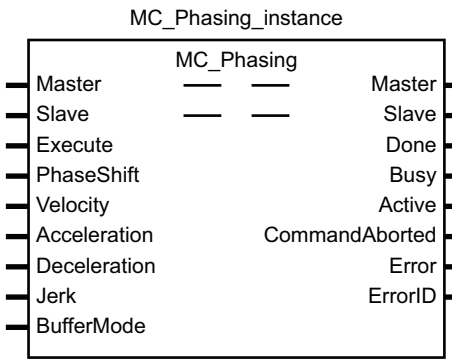


● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_Phasing

同期制御中の主軸の位相補正を行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_Phasing	主軸相対値位相補正	FB		MC_Phasing_instance (Master :=「パラメータ」, Slave :=「パラメータ」, Execute :=「パラメータ」, PhaseShift :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, BufferMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
PhaseShift	位相補正量	LREAL	負数、正数、「0」	0	主軸の位相補正量を指定します。 単位は［指令単位］となります。*1
Velocity	目標速度	LREAL	正数	0	位相補正量の目標速度（補正速度）を指定します。*2 単位は［指令単位／s］となります。*1
Acceleration	加速度	LREAL	正数または「0」	0	加速度を指定します。 単位は［指令単位／s^2］となります。*1
Deceleration	減速度	LREAL	正数または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*1
Jerk	ジャーク	LREAL	正数または「0」	0	ジャークを指定します。 単位は［指令単位／s^3］となります。*1

入力変数	名称	データ型	有効範囲	初期値	内容
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *3	モーション命令多重起動時の動作を指定します。 0: アボーティング

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 目標速度は必ず設定してください。設定せずに動作させると目標速度設定範囲外（エラーコード: 5422Hex）が発生します。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位相補正が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	位相補正を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。 • 実行中の同期制御命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Master	主軸	_sAXIS_REF	−	主軸を指定します。*1
Slave	従軸	_sAXIS_REF	−	従軸を指定します。*1

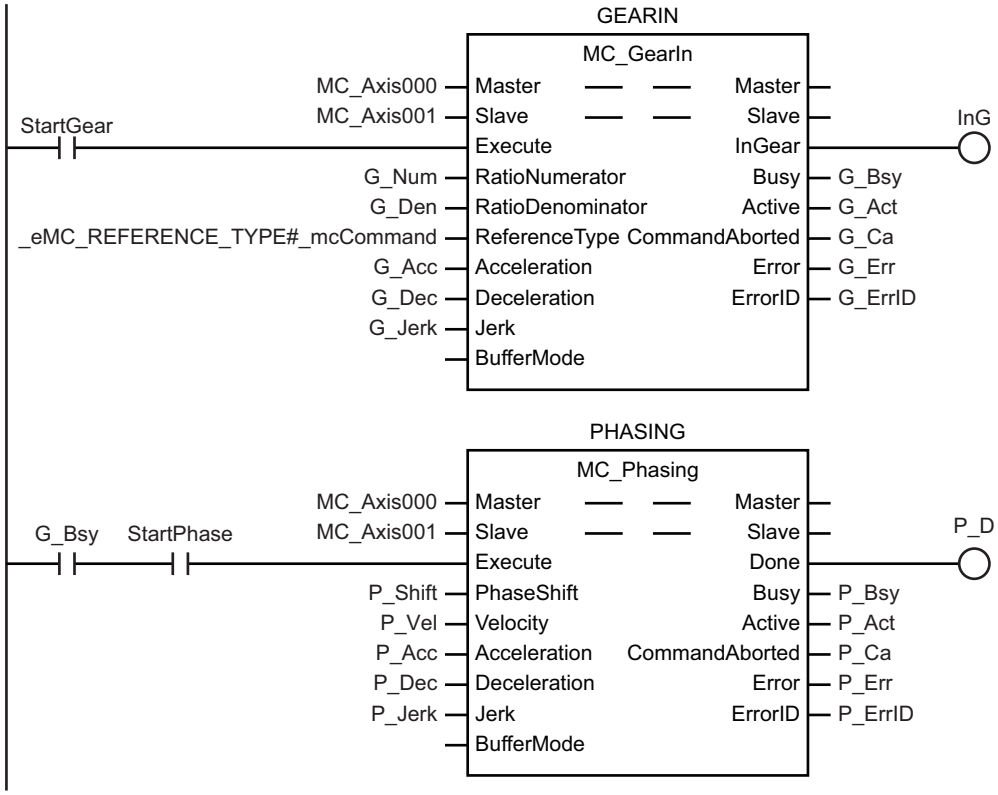
1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

使用上の注意

主軸と従軸に同じ軸を指定すると、軽度フォールトレベルの「主軸従軸同一（エラーコード: 5436 Hex)」異常となります。

機能説明

- MC_CombineAxes（加減算位置決め）命令実行中を除き、単軸同期制御中に本命令を起動すると、設定した「PhaseShift」（位相補正量)、「Velocity」（目標速度)、「Acceleration」（加速度)、「Deceleration」（減速度）によって主軸の位相を補正します。
- 主軸の指令現在位置、およびフィードバック現在位置は変化せず、主軸の指令現在位置、およびフィードバック現在位置を相対量で補正した値が「主軸の位相」となります。
  従軸は、補正された「主軸の位相」に対して同期します。
- 「PhaseShift」（位相補正量）に到達すると、「Done」（完了）が TRUE となります。
- 実行中の同期制御命令が完了すると、補正は終了します。再び、同期制御命令を実行した場合、以前の補正量は影響しません。
- 主軸位相補正が有効となる同期制御命令は MC_CamIn（カム動作開始）命令、MC_GearIn（ギア動作開始）命令、MC_GearInPos（位置指定ギア動作）命令、MC_MoveLink（台形パターンカム）命令の 4 つとなります。
- 本命令をユーザプログラム中に記載する場合は、下図のように同期制御命令の後に MC_Phasing（主軸相対値位相補正）命令を記述してください。



使用上の注意

主軸に対する注意事項については、「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。

命令の詳細

以下に本命令の詳細について説明します。

●「Master」（主軸）、「Slave」（従軸）の指定

「Master」（主軸）と「Slave」（従軸）は、軸変数名で軸を指定します。
同期制御命令が実行中でない「Master」（主軸）、「Slave」（従軸）の組み合わせを指定すると軸指定異常が発生します。

● PhaseShift（位相補正量）

「PhaseShift」（位相補正量）には、「Slave」（従軸）から見た「Master」（主軸）の位相補正量を指定します。
位相補正量は、相対量を指定します。

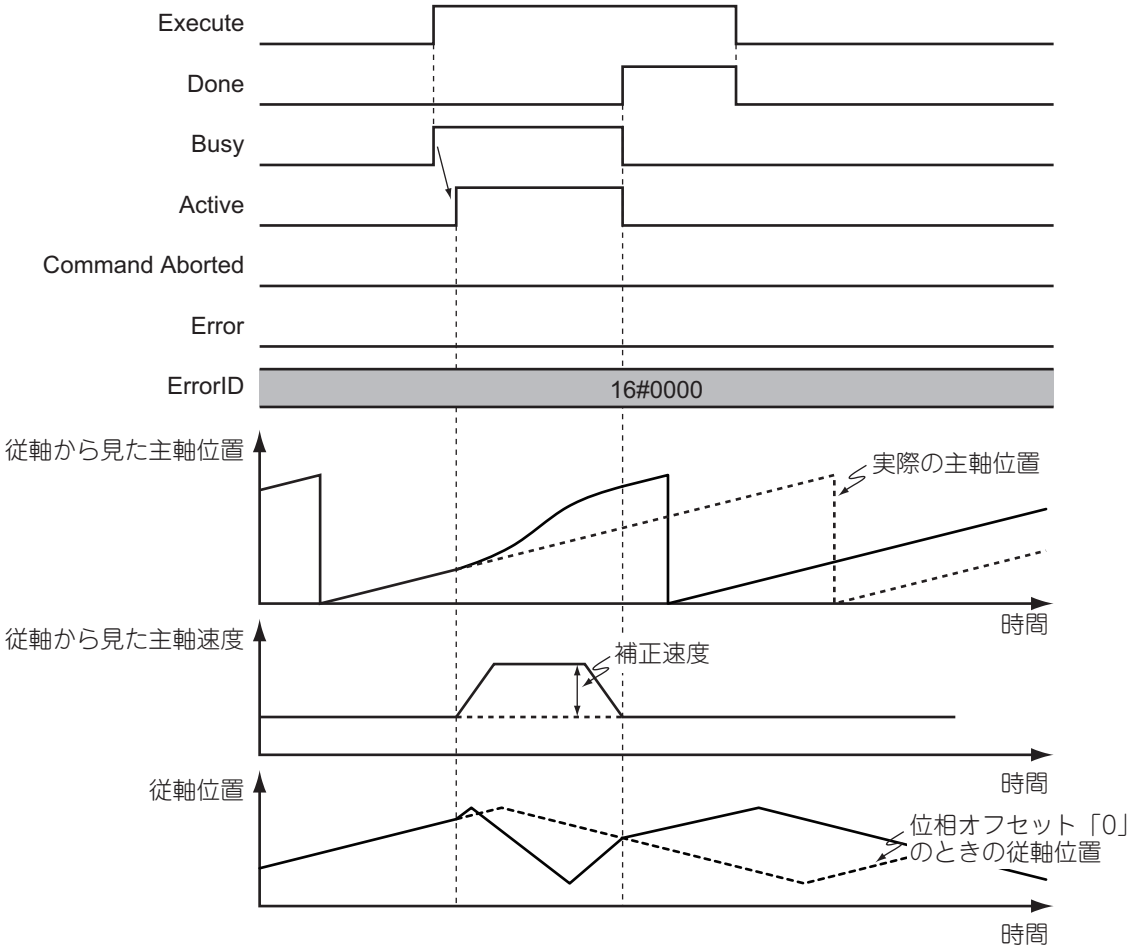
● Velocity（目標速度）、Acceleration（加速度）、Deceleration（減速度）、Jerk（ジャーク）

位相補正量の目標速度（補正速度）、加速度、減速度、ジャークは、「Velocity」、「Acceleration」、「Deceleration」、「Jerk」で指定します。
「Slave」（従軸）から見た「Master」（主軸）の目標速度（補正速度）は、主軸速度に対する相対速度となります。

「Slave」（従軸）から見た「Master」（主軸）の目標速度（補正速度）は、位相補正量、加速度、減速度、ジャークにより次のようになります。

例）電子カムの主軸相対値位相補正の場合

「Slave」（従軸）から見た「Master」（主軸）の補正速度は、主軸速度に対する相対速度となります。



- 位相補正量に「0」を指定すると、「Master」（主軸）の位相補正量「0」として正常終了します。
- 目標速度（補正速度）に「0」を指定すると、範囲外指定として「Slave」（従軸）の軸異常となります。
- 指定した目標速度（補正速度）と主軸速度の加算速度は、「Master」（主軸）の最高速度を超えて指定することが出来ます。

参考

MC_Phasing（主軸相対値位相補正）命令の結果は、同期した「Slave」（従軸）の動作として異常検知を行います。
そのため、命令の起動時には、本命令の「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）の設定に対する異常検知は行われません。

● BufferMode（バッファモード選択）

1 つ前の軸動作と今回の動作のつなぎ方を指定します。
次の 1 種類があります。

バッファモード選択	説明
アボーティング	本命令を再起動すると、即、本命令による補正を開始します。

多重起動による反転時動作は、主軸に設定された［反転時動作選択］に従います。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● インポジションチェック

位相補正動作の完了は補正値出力の終了となり、インポジションチェックを行いません。

モーション命令再起動

本命令実行中に、再度命令開始を立ち上げると、「PhaseShift」（位相補正量）、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）の変更を行います。
変更の動作は相対位置決めと同じになります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

「Slave」（従軸）において、本命令実行中の他命令による多重起動は、［アボーティング］が可能です。［バッファード］、［ブレンディング］による命令多重起動はできません。

● 本命令同士の多重起動

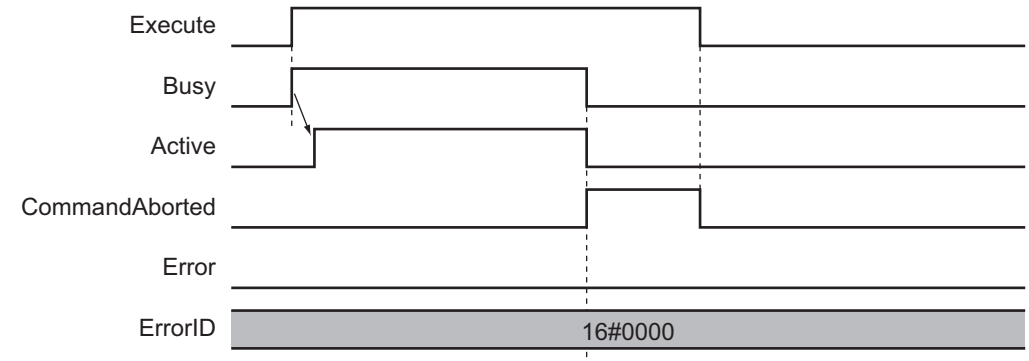
指定した従軸が MC_Phasing（主軸相対値位相補正）命令動作中に、MC_Phasing（主軸相対値位相補正）命令を起動することができます。

異常

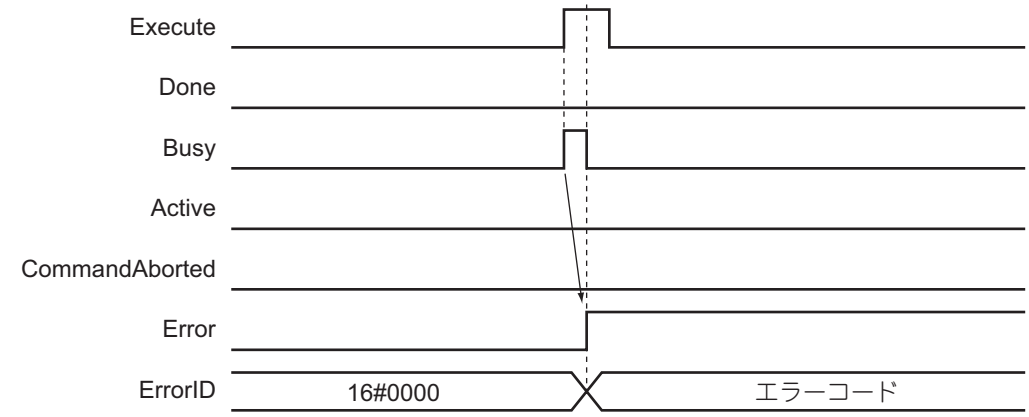
本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。
「Slave」（従軸）が異常の発生により同期状態解除となった時点で、MC_Phasing（主軸相対値位相補正）命令動作は「CommandAborted」（実行中断）となります。
なお、主軸の異常状態は本命令の動作に影響しません。

● 異常発生時のタイミングチャート

MC_CamIn（カム動作開始）命令



MC_Phasing（主軸相対値位相補正）命令



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_TorqueControl

サーボドライバのトルク制御モードを用いてトルクを制御します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_TorqueControl	トルク制御	FB	MC_TorqueControl_instance MC_TorqueControl Axis — — Axis Execute — InTorque Torque — Busy TorqueRamp — Active Velocity — CommandAborted Direction — Error BufferMode — ErrorID	MC_TorqueControl_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, Torque :=「パラメータ」, TorqueRamp :=「パラメータ」, Velocity :=「パラメータ」, Direction :=「パラメータ」, BufferMode :=「パラメータ」, InTorque =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

使用上の注意

NX シリーズ パルス出力ユニットに対しては、本命令を使用できません。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Torque	目標トルク	LREAL	0～1000.0	300.0	サーボドライバへ出力する目標トルクを[0.1%]単位で指定します。定格トルクを「100.0%」としたときの比率で指定します。*1 単位は[%]となります。
TorqueRamp	トルクランプ	LREAL	正数または「0」	0	現在のトルクから目標トルクを出力するまでのトルクの変化率を指定します。 単位は[%/s]となります。
Velocity	制限速度	LREAL	正数または「0」	0	目標速度を指定します。単位は、[指令単位／s] となります。*2
Direction	方向選択	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	0 *3	目標トルクの方向を指定します。 0: 正方向指定 2: 負方向指定

入力変数	名称	データ型	有効範囲	初期値	内容
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *3	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード

*1. 「1000.0%」を超える値が指定された場合は、「1000.0%」として扱われます。また、負値が指定された場合は、「0.0%」として扱われます。
*2. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
InTorque	目標トルク到達	BOOL	TRUE, FALSE	目標トルクに到達したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
InTorque	目標トルクを出力中	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。 • 命令再起動で、目標トルクを変更したとき。
Busy	「Execute」の立ち上がり	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	トルク指令値の出力を開始したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	－	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- トルク指令値を直接指定してサーボモータの出力トルクを制御します。
- 目標トルクは[0.1%]単位で指定します。小数点第 2 位以下を指定した場合、小数点第 2 位で四捨五入します。
- 本命令の実行を停止する場合、MC_Stop（強制停止）命令を使用してください。
- トルク制御は、サーボドライバのトルク制御モード機能を使用して実現しています。
- 制御モードが切り替わるまでは、直前の制御モードを維持します。

例
位置制御→トルク制御に切り替える時: サーボドライバがトルク制御に変わるまで、位置制御
トルク制御→位置制御に切り替える時: サーボドライバが位置制御に変わるまで、トルク制御

● データオブジェクトのマッピング

MC_TorqueControl（トルク制御）命令を使用する場合、Sysmac Studio の軸基本設定の［詳細設定］で、以下のオブジェクトデータをマッピングしてください。

- 目標トルク(6071Hex)
- オペレーションモード(6060Hex)
- 現在トルク(6077Hex)
- オペレーションモード表示(6061Hex)

これらを設定していない場合、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。

データオブジェクトのマッピングについては、「2-3 PDO マッピング (P.2-32)」、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

命令の詳細

以下に本命令の詳細について説明します。

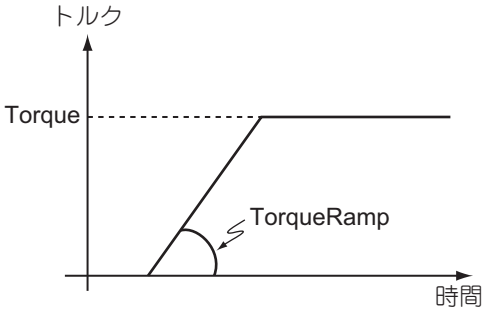
● Axis（軸）指定

「Axis」（軸）で、トルク制御する軸を指定します。

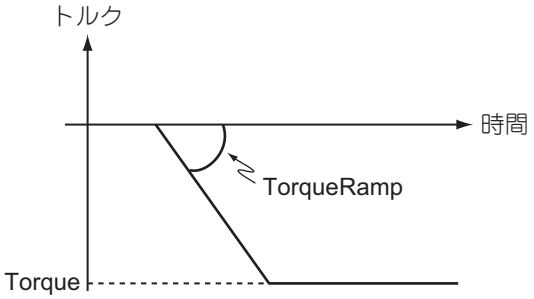
● TorqueRamp（トルクランプ）

現在指定している指令トルクから目標トルクを出力するまでの傾斜を指定します。

例 1）方向選択が［0: 正方向指定］の場合



例 2）方向選択が［2: 負方向指定］の場合



使用上の注意

目標トルクはモータ最大トルクより小さな値になるように設定してください。
モータ最大トルクを超えて設定した場合の動作は、サーボドライバに依存します。

● Velocity（制限速度）

トルク制御時の軸の最大速度を制限します。
軸の速度がこの制限速度値に達した場合、サーボドライバでトルクを落として軸の速度を下げます。
制限速度値の機能はサーボドライバの機能を使用して実現しています。

詳細については、各サーボドライバマニュアルのトルク制御機能を参照してください。

使用上の注意

- トルク制御時は、軸の速度が速くなります。安全のために必ず「Velocity」（制限速度）を設定してください。
- オムロン製サーボドライバ G5 シリーズを使用する場合、サーボドライバの「速度制限選択 (3317 Hex)」を「1: EtherCAT 通信からの速度制限」に必ず設定してください。
  設定していない場合、速度制限値が機能しません。また、限界入力信号が ON となっても軸が停止しません。
- 速度制限値は、プロセスデータ 607F Hex を使用します。
  オムロン製サーボドライバ 1S シリーズまたは G5 シリーズを使用する場合、Sysmac Studio の軸パラメータ設定画面の［詳細設定］において、速度制限値(607F Hex)を使用するように設定してください。
  なお、他社サーボドライバで速度制限機能を使用する場合には、各社サーボドライバのマニュアルを参照してください。

● Direction（方向選択）

目標トルクを出力する方向を指定します。
軸の正方向へ出力したい場合、正方向を、軸の負方向へ出力したい場合、負方向を指定します。

● BufferMode（バッファモード選択）

1 つ前の軸動作と今回の動作のつなぎ方を指定します。
次の 2 種類があります。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。
バッファード	現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● トルク制御中の軸停止

オムロン製サーボドライバ 1S シリーズまたは G5 シリーズを使用する場合は、本命令実行中に MC_Stop（強制停止）命令を起動すると、MC_Stop（強制停止）命令の減速度は適用されずに即停止となります。
また、減速停止扱いとなる異常が発生した場合でも、即停止の扱いとなります。

● トルク制御中の指令現在位置、フィードバック現在位置

本命令によるトルク制御中、モーション制御システム変数で取得できる現在位置は以下のようになります。

フィードバック現在位置　：サーボドライバから取得する値をギア比で変換した値となります。
指令現在位置　　：1 周期前のフィードバック現在位置と同じ値となります。

● 使用可能軸と起動条件

- サーボ軸において、MC_Power（運転可）命令の「Enable」（有効）が TRUE （サーボ ON 状態）のとき、本命令を起動することができます。
- 仮想サーボ軸は、常時受け付け可能です。ただし、サーボドライバの制御モード切り替え処理は行いません。
- エンコーダ軸、仮想エンコーダ軸では起動時異常となります。

● サーボ OFF 時の対応

MC_Power（運転可）の出力変数「Status」（運転可）が FALSE になった場合、MC 機能モジュールは CSP モードに切り替える処理を行います。
ただし、オムロン製サーボドライバ G5 シリーズは、サーボ OFF 状態では、MC 機能モジュールからの制御モード切替を受け付けません。

● 軸変数のステータス

軸変数のステータスは、「Status.Continuous」（連続動作中）が TRUE となります。
また、軸変数の「DrvStatus」（サーボドライバ状態）は、CST（サイクリック同期トルク(CST)モード中）が TRUE となります。

● 原点状態

原点確定状態が継続します。

● ソフトウェアリミット

本命令に対して、ソフトウェアリミットは有効となります。
軸パラメータで、以下の設定を選択した場合でも、有効となります。

- 指令位置に対して有効。減速停止させる。
- 指令位置に対して有効。たまりパルス停止させる。

● カウントモードが［リニアモード］の場合

オーバーフロー、またはアンダーフローした場合、目標位置を持たない動作の扱いとなります。

● 反転時動作

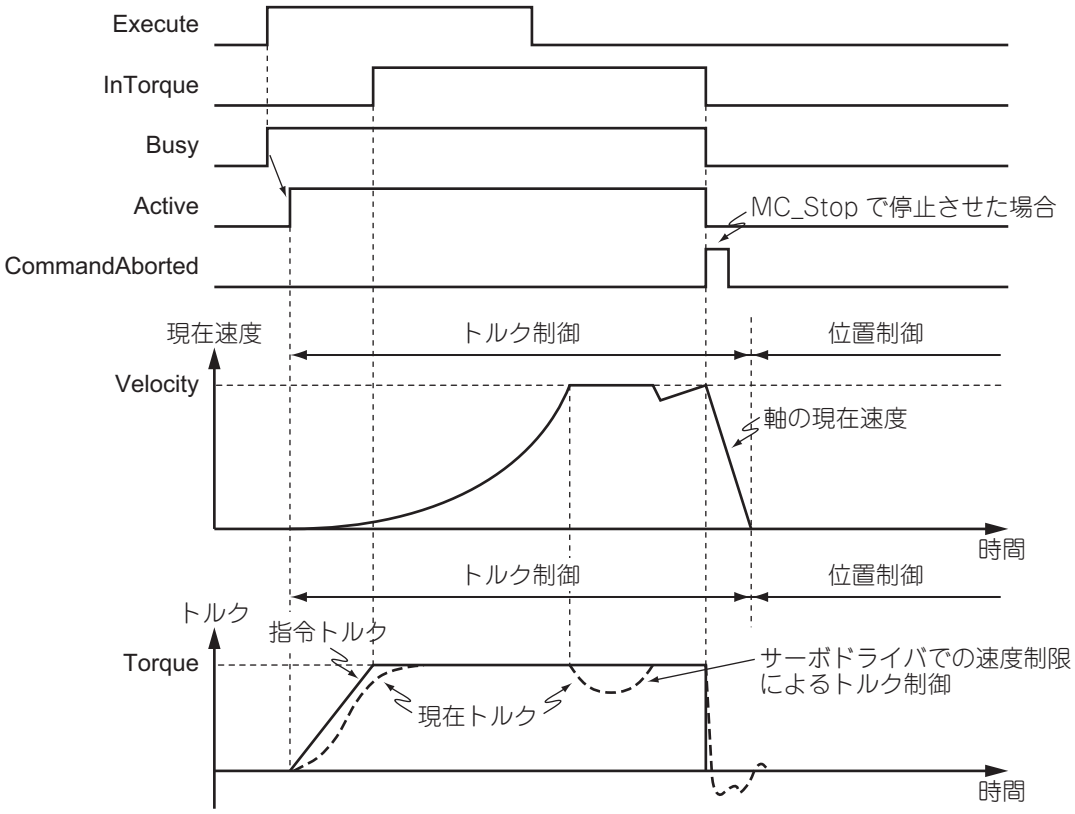
多重起動を行った際にトルク指令値が反転する場合、軸パラメータの［反転時動作選択］の設定内容に関係なく、本命令の「TorqueRamp」（トルクランプ）で動作します。

また多重起動による反転時の動作は以下の通りです。

- 本命令動作中に、CSP を使用する命令を多重起動したことで指令位置が反転する場合、軸パラメータの［反転時動作選択］にしたがって反転動作を行います。
- CSP や CSV を使用する命令の動作中に、本命令を多重起動したことでトルク指令値が反転する場合、「TorqueRamp」（トルクランプ）にしたがってトルク指令を反転させます。
- 本命令動作中に、本命令を多重起動したことでトルク指令値が反転する場合、「TorqueRamp」（トルクランプ）にしたがってトルク指令を反転させます。

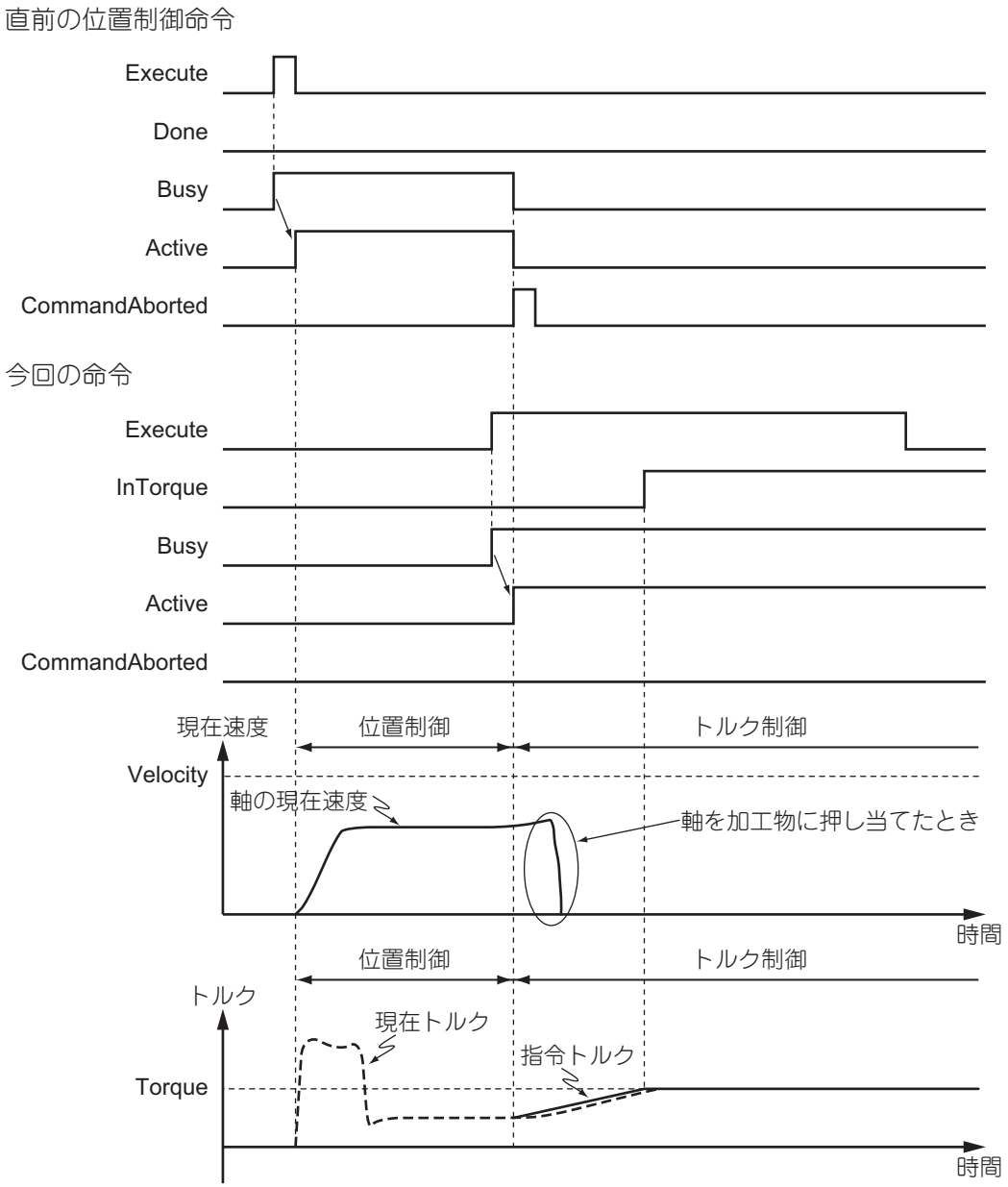
タイミングチャート

本命令を起動し、停止させる場合



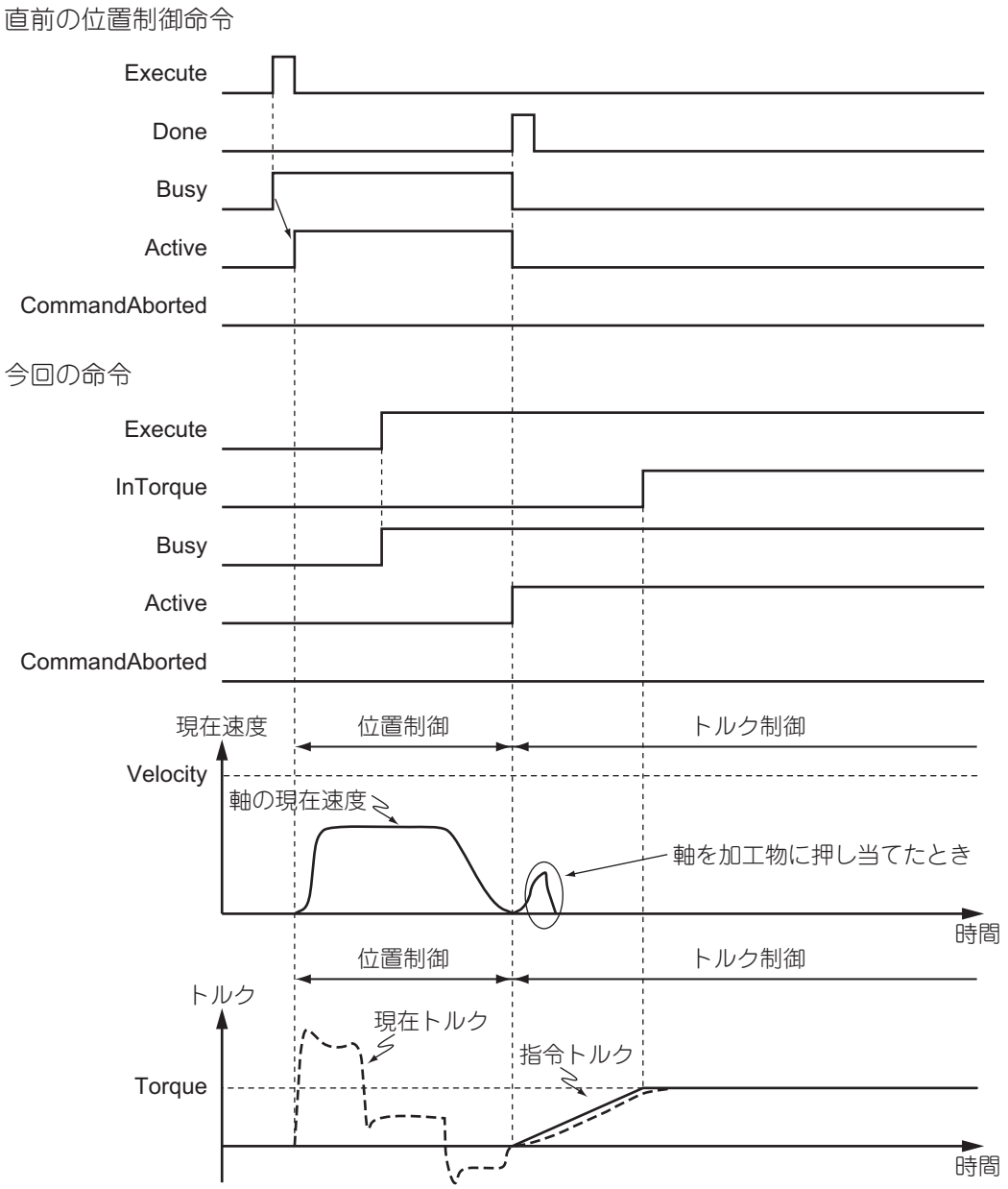
1 つ前の動作中に中止で起動する場合

本命令実行中に、軸が停止する押し当てのアプリケーションで説明します。



1 つ前の動作中に［バッファード］で起動する場合

本命令実行中に、軸が停止する押し当てのアプリケーションで説明します。



● 制御モードの切り替え

- MC_MoveAbsolute（絶対値位置決め）命令、MC_MoveRelative（相対値位置決め）命令などの位置制御の命令実行中に、MC_TorqueControl（トルク制御）命令を実行した場合、「BufferMode」（バッファモード選択）の設定によって動作が変わります。
  「BufferMode」（バッファモード選択）が［アボーティング］のときは、実行した時点でトルク制御に切り替わります。［バッファード］のときは、前の動作が完了した時点で、トルク制御に切り替わります。
- MC_TorqueControl（トルク制御）命令実行中に、MC_MoveAbsolute（絶対値位置決め）命令などのほかの命令で中止された場合や、軸に異常が発生した場合は、その時点で位置制御に切り替わります。
- 命令実行のタイミングで「Active」（制御中）は切り替わりますが、サーボドライバ側の制御モードが切り替わるまでには数周期が必要となります。制御モードが切り替わるまでの時間は、サーボドライバに依存します。

制御モード切り替え判定

オムロン製サーボドライバ 1S シリーズまたは G5 シリーズで軸を停止する場合、MC 機能モジュールは速度制限値(607F Hex)を「0」とします。その後、定周期タスクで 3 回連続してフィードバック現在速度が以下の判定式を満足した場合に、CSP への切り替えを行います。

フィードバック現在速度≦（最高速度）×0.1

他社サーボドライバの場合、サーボドライバの制御モードを CST から CSP に切り替えるとともに、切り替え処理のときのフィードバック現在位置でサーボロックをさせます。

使用上の注意

ここでの定周期タスクとは、プライマリ定周期タスクまたは定周期タスク（実行優先度 5）を表します。

制御モード切り替え失敗

MC 機能モジュールがサーボドライバに対して制御モード切り替え指令を行ってから 1 秒経過してもサーボドライバの切り替えが完了しない場合、「サーボドライバの制御モード切替異常（エラーコード: 7439 Hex)」とし、サーボ OFF します（フリーラン停止)。
「サーボドライバの制御モード切替異常（エラーコード: 7439 Hex)」については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

また、速度制限値を「0」にした後の経過時間が 10 秒経過しても切り替え判定式が成立しない場合も、上記と同様にサーボ OFF します。

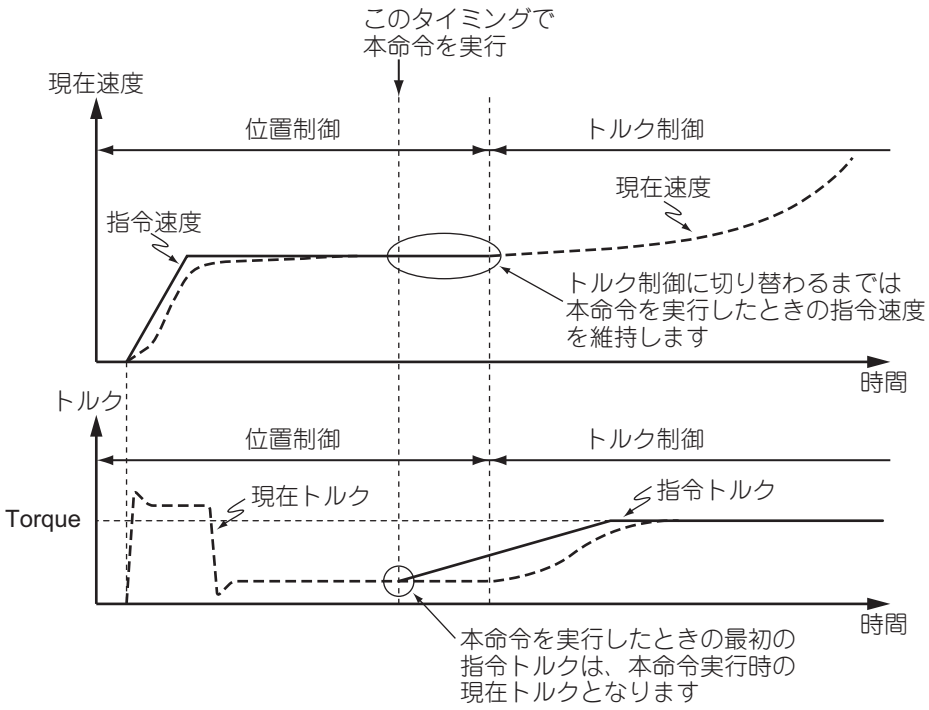
制御モード切り替え動作例

制御モードが切り替わるまでの指令トルクと指令速度の関係を、軸動作中に切り替えた場合の例を用いて説明します。

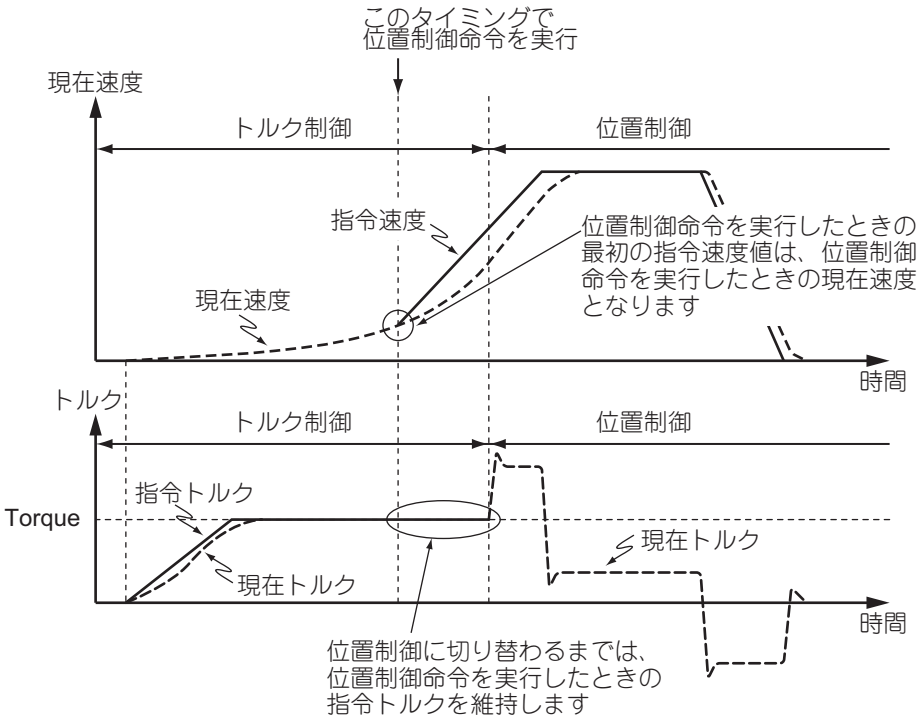
使用上の注意

軸が動作しているときに、サーボドライバ側で制御モードを切り替えると、サーボドライバによっては異常が発生することがあります。

位置制御からトルク制御に切り替えるとき



トルク制御から位置制御に切り替えるとき



モーション命令再起動

トルク制御動作中に入力パラメータを変更し、「Execute」（起動）を再度 TRUE にすることにより、本命令の動作を変更することができます。
モーション命令再起動で変更可能な入力変数は、「Torque」（目標トルク）、「TorqueRamp」（トルクランプ）、「Velocity」（制限速度）です。
モーション命令再起動によって「Torque」（目標トルク）を変更する場合、「InTorque」（目標トルク到達）は、再起動によって設定された新しい目標トルクに対して動作します。

モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して、本命令を起動することにより、本命令への切り替えや本命令のバッファリングを行うことができます。
各軸に 1 つまでバッファリングすることが可能です。
多重起動時の本命令の動作を、「BufferMode」（バッファモード選択）で指定します。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。
バッファード	現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。

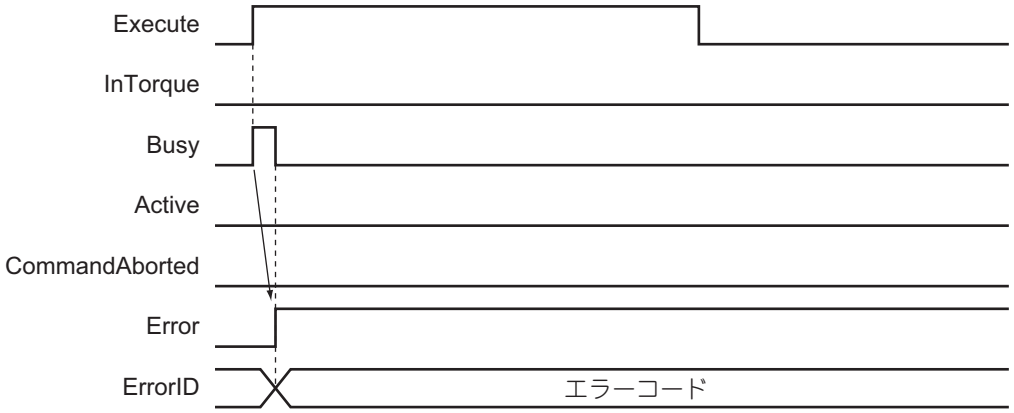
「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

本命令実行中のほかの命令による多重起動の実行は、ほかの命令の「BufferMode」（バッファモード選択）に［アボーティング］または［バッファード］を指定した時だけ可能です。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_SetTorqueLimit

サーボドライバのトルク制限機能を使用し、サーボドライバの出力トルクを制限します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_SetTorqueLimit	トルク制限	FB	MC_SetTorqueLimit_instance MC_SetTorqueLimit Axis — — Axis Enable — Enabled PositiveEnable — Busy PositiveValue — Error NegativeEnable — ErrorID NegativeValue	MC_SetTorqueLimit_instance (Axis :=「パラメータ」, Enable :=「パラメータ」, PositiveEnable :=「パラメータ」, PositiveValue :=「パラメータ」, NegativeEnable :=「パラメータ」, NegativeValue :=「パラメータ」, Enabled =>「パラメータ」, Busy =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

使用上の注意

NX シリーズ パルス出力ユニットに対しては、本命令を使用できません。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE の間、命令を実行します。
PositiveEnable	正方向有効	BOOL	TRUE, FALSE	FALSE	TRUE: 正方向のトルク制限を有効にします。 FALSE: 正方向のトルク制限を無効にします。
PositiveValue	正方向トルク制限値	LREAL	0.1～1000.0、または「0.0」	300.0	正方向へのトルク制限値を 0.1%単位で指定します。 軸パラメータの［正方向トルク制限上限値］を超える値が入力された場合、［正方向トルク制限上限値］となります。 「0」および負数を指定した場合は、「0」として動作します。
NegativeEnable	負方向有効	BOOL	TRUE, FALSE	FALSE	TRUE: 負方向のトルク制限を有効にします。 FALSE: 負方向のトルク制限を無効にします。
NegativeValue	負方向トルク制限値	LREAL	0.1～1000.0、または「0.0」	300.0	負方向へのトルク制限値を 0.1%単位で指定します。 軸パラメータの［負方向トルク制限上限値］を超える値が入力された場合、［負方向トルク制限上限値］となります。 「0」および負数を指定した場合は、「0」として動作します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Enabled	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。*1
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*2	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. 「Enabled」（有効）は本命令の状態を示すものであり、サーボドライバによるトルク制限状態を表すものではありません。
*2. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Enabled	• 「Enable」が TRUE となったとき。 • MC_Power 命令が実行中のとき。	• 「Enable」が FALSE となった次の 1 周期後 • 「Error」が TRUE となったとき。 • MC_Power 命令の「Enable」が FALSE となったとき。
Busy	「Enable」が TRUE となったとき。	• 「Error」が TRUE となったとき。 • 「Enable」が FALSE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 本命令は、サーボドライバに対してトルク制限値を設定します。
- 「Enable」（有効）が TRUE の状態で、「PositiveEnable」（正方向有効）を TRUE にすると、「PositiveValue」（正方向トルク制限値）で制限します。
  同様に、「Enable」（有効）が TRUE の状態で、「NegativeEnable」（負方向有効）を TRUE にすると、「NegativeValue」（負方向トルク制限値）で制限します。
- 「PositiveEnable」（正方向有効）を FALSE にした場合、軸パラメータの［正方向トルク制限上限値］をサーボドライバに設定します。
  同様に「NegativeEnable」（負方向有効）を FALSE にした場合も、軸パラメータの［負方向トルク制限上限値］をサーボドライバに設定します。
- 本命令の「Enable」（有効）を FALSE にした場合、［正方向トルク制限上限値］、［負方向トルク制限上限値］をサーボドライバに設定します。同時に「Busy」（実行中）、「Enabled」（有効）を FALSE にします。

・トルク制限値は、モータトルクに対して 0.1%単位で設定できます。小数点第 2 位以下を指定した場合、小数点第 2 位を四捨五入します。

使用上の注意

軸パラメータの［正方向トルク制限上限値］、［負方向トルク制限上限値］は、使用するサーボドライバに定められたトルク制限の上限値を設定してください。

● データオブジェクトのマッピング

MC_SetTorqueLimit（トルク制限）命令を使用する場合、Sysmac Studio の軸基本設定の［詳細設定］で、以下のオブジェクトデータをマッピングしてください。

・正方向トルク制限値(60E0 Hex)
・負方向トルク制限値(60E1 Hex)

これらを設定していない場合、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。

使用上の注意

・他社製サーボドライバを使用する場合、正方向トルク制限値(60E0 Hex)、負方向トルク制限値(60E1 Hex)がトルク制限値となるようにサーボドライバを設定してください。
設定方法については、各サーボドライバのマニュアルを参照してください。
・他社製サーボドライバで、正方向トルク制限値、負方向トルク制限値を PDO にマッピングできない場合は本命令は使用できません。
この場合、各社のサポートツールや SDO 通信を使用して、トルク制限値を設定してください。

データオブジェクトのマッピングについては、「2-3 PDO マッピング (P.2-32)」、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 入力パラメータの書き替え

「Enable」（有効） が TRUE のときには、以下の入力変数の値を常時変更できます。

・PositiveEnable（正方向有効）
・NegativeEnable（負方向有効）
・PositiveValue（正方向トルク制限値）
・NegativeValue（負方向トルク制限値）

● MC_Home（原点復帰）命令、MC_HomeWithParameter（パラメータ指定原点復帰）命令の押し当て動作との関係（オムロン製サーボドライバ 1S シリーズまたは G5 シリーズを使用する場合）

原点復帰の［12: 近傍退避・押し当て時間指定］、［13: 原点近傍入力なし・押し当て原点入力指定］は、サーボドライバにあらかじめ設定されたトルク制限値を用いて、押し当て方向に対してトルク制限を自動開始します。

使用上の注意

他社製サーボドライバを使用する場合、MC_Home（原点復帰）命令または MC_HomeWithParameter（パラメータ指定原点復帰）命令での自動トルク制限機能は機能しません。
MC_SetTorqueLimit（トルク制限）命令、SDO 通信、サーボドライバのサポートツール等を使用して、適切な値を設定してください。

参考

- 原点復帰の正常完了後もトルク制限は継続します。
- 逆方向へ移動する命令が起動された場合には、トルク制限が自動的に解除されます。
  原点復帰の詳細については、「MC_Home (P.3-16)」または「MC_HomeWithParameter (P.3-37)」を参照してください。

● オムロン製サーボドライバ 1S シリーズの設定

本命令を使用するには、Sysmac Studio を用いて、1S シリーズ サーボドライバの［トルク制限］の［切り替え選択］(3330Hex-01Hex)を設定する必要があります。

- 原点復帰の押し当て動作時に原点検出方向に対してトルク制限を行い、かつそのほかの動作は MC_SetTorqueLimit（トルク制限）命令のトルク制限方向やトルク制限値を使用する場合には「2」を使用してください。
  この場合、原点復帰の押し当て動作中は MC_SetTorqueLimit（トルク制限）命令の入力変数の内容は無視されます。
- 「0」を設定した場合、常に MC_SetTorqueLimit（トルク制限）命令の入力変数の内容が適用されます。原点復帰の押し当て動作時とそれ以外で適切なトルク制限値を設定する必要があります。

		［トルク制限］の［切り替え選択］(3330Hex-01Hex)	
		2	0
正方向トルク制限値	原点復帰中	サーボドライバの［トルク制限］の［正方向トルク制限値 2］(3330Hex-05Hex)が適用されます。	MC_SetTorqueLimit（トルク制限）命令の「PositiveValue」(正方向トルク制限値）が適用されます。
	原点復帰以外	MC_SetTorqueLimit（トルク制限）命令の「PositiveValue」(正方向トルク制限値）が適用されます。	
負方向トルク制限値	原点復帰中	サーボドライバの［トルク制限］の［負方向トルク制限値 2］(3330Hex-06Hex)が適用されます。	MC_SetTorqueLimit（トルク制限）命令の「NegativeValue」(負方向トルク制限値）が適用されます。
	原点復帰以外	MC_SetTorqueLimit（トルク制限）命令の「NegativeValue」(負方向トルク制限値）が適用されます。	

詳細については、『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-377)』、または『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ 安全機能対応 ユーザーズマニュアル(SBCE-438)』を参照してください。

● オムロン製サーボドライバ G5 シリーズの設定

本命令を使用するには、サーボドライバのサポートソフトを用いて、サーボドライバの［トルクリミット選択］(3521 Hex)を設定する必要があります。

- 原点復帰の押し当て動作時に原点検出方向に対してトルク制限を行い、かつそのほかの動作は本命令のトルク制限方向やトルク制限値を使用する場合には「6」を使用してください。
  この場合、原点復帰の押し当て動作中は本命令の入力変数の内容は無視されます。
- 「4」を設定した場合、常に本命令の入力変数の内容が適用されます。原点復帰の押し当て動作時とそれ以外で適切なトルク制限値を設定する必要があります。

		トルクリミット選択(3521 Hex)	
		6（推奨）	4
正方向トルク制限値	原点復帰中 *1	［第 3 トルクリミット］(3525 Hex) が適用されます。	本命令の「PositiveValue」(正方向トルク制限値）と、［第 1 トルクリミット］(3013 Hex)の小さい方が適用されます。
	原点復帰以外	本命令の「PositiveValue」(正方向トルク制限値）と、［第 1 トルクリミット］(3013 Hex)の小さい方が適用されます。	

		トルクリミット選択(3521 Hex)	
		6（推奨）	4
負方向トルク制限値	原点復帰中 *1	［第 4 トルクリミット］(3526 Hex) が適用されます。	本命令の「NegativeValue」(負方向トルク制限値）と、［第 2 トルクリミット］(3522 Hex)の小さい方が適用されます。
	原点復帰以外	本命令の「NegativeValue」(負方向トルク制限値）と、［第 2 トルクリミット］(3522 Hex)の小さい方が適用されます。	

*1. トルク制限自動解除まで

詳細については、『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-365)』、『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプ ユーザーズマニュアル(SBCE-366)』を参照してください。

● MC_TorqueControl（トルク制御）命令との関係

MC_SetTorqueLimit（トルク制限）命令と MC_TorqueControl（トルク制御）命令は、組み合わせて使用することができます。

● 軸グループに属する軸に対する使用

有効な軸グループに属している軸に対して、本命令を使用することができます。

● CPU ユニットの動作モードとの関係

運転モード中に本命令で設定した内容は、プログラムモードに遷移しても継続します。

● 使用可能な軸と起動条件

- サーボ軸で本命令を使用できます。
  ただし、サーボの状態によって本命令の出力変数「Enabled」(有効）の状態が異なります。

	サーボ ON 状態	サーボ OFF 状態
Enabled（有効）の状態	TRUE	FALSE *1

*1. 本命令の「Enabled」(有効）が FALSE の場合は、サーボドライバのトルク制限は機能しません。

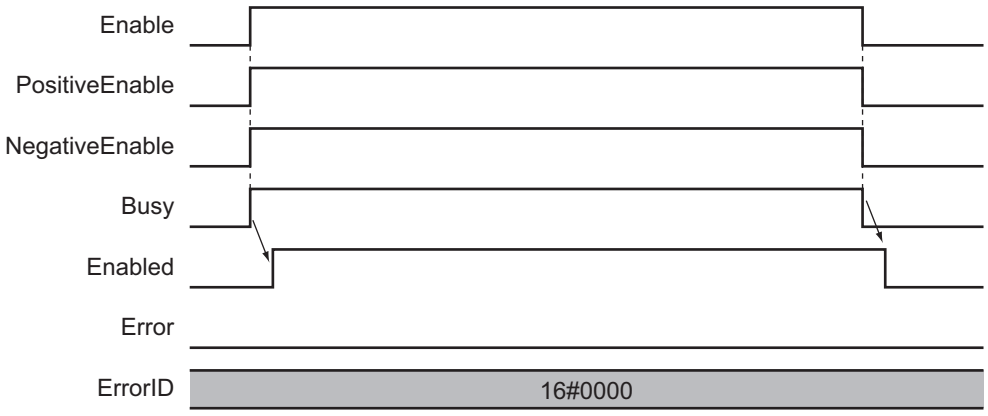
- 仮想サーボ軸でも命令は受け付け可能ですが、トルク制限は行いません。
- エンコーダ軸、仮想エンコーダ軸では起動時異常となります。

● 軸変数のステータス（サーボドライバ状態）

サーボドライバの内部制限機能が働いている場合に、軸変数の「DrvStatus」(サーボドライバ状態）の「ILA」(ドライバ内部機能制限中）が TRUE となります。
この変数は、サーボドライバのトルク制限、速度制限、駆動禁止入力、ソフトウェアリミットの 4 つの OR 出力です。

タイミングチャート

トルク制限を実行した場合のタイミングチャートは、次のようになります。



モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令同士の多重起動

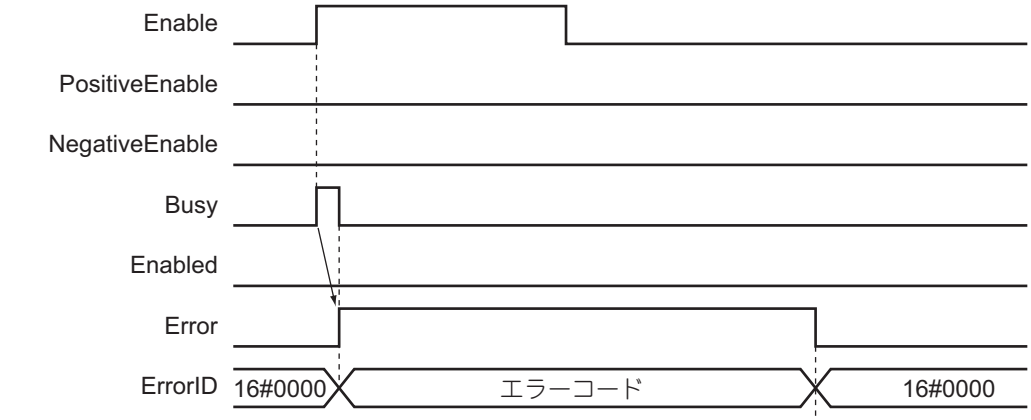
本命令実行中の軸に、本命令の別インスタンスを起動した場合、後から実行されたインスタンスが優先して処理されます。
「Enabled」（有効）は、両方とも TRUE となります。
具体的な動作としては、後から実行されたインスタンスのトルク制限値が有効となります。後から実行されたインスタンスの「Enable」（有効）を FALSE にすると、トルク制限は無効となります。

異常

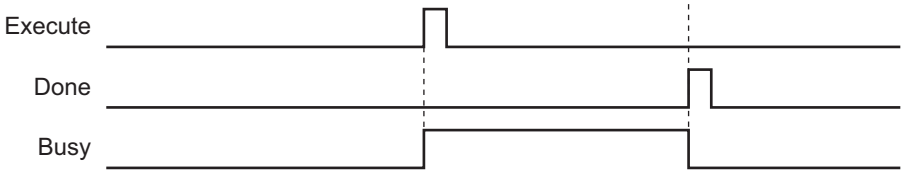
本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート

MC_SetTorqueLimit 命令



MC_Reset 命令

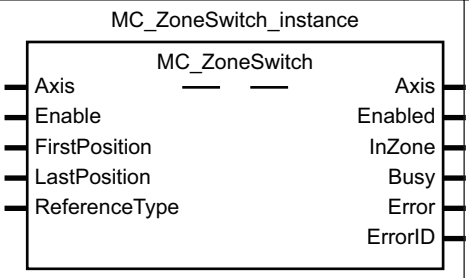


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_ZoneSwitch

軸の指令現在位置、もしくはフィードバック現在位置が指定された範囲内に存在するかを判定します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_ZoneSwitch	ゾーン監視	FB		MC_ZoneSwitch_instance (Axis :=「パラメータ」, Enable :=「パラメータ」, FirstPosition :=「パラメータ」, LastPosition :=「パラメータ」, ReferenceType :=「パラメータ」, Enabled =>「パラメータ」, InZone =>「パラメータ」, Busy =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE の間、命令を実行します。
FirstPosition	開始位置	LREAL	負数、正数、「0」	0	ゾーン範囲の開始位置を指定します。*1 単位は［指令単位］となります。*2
LastPosition	終了位置	LREAL	負数、正数、「0」	0	ゾーン範囲の終了位置を指定します。*3 単位は［指令単位］となります。*2
ReferenceType	位置種別選択	_eMC_REFERENCE_TYPE	0: _mcCommand 1: _mcFeedback	0 *4	監視する軸情報を指定します。 0: 指令位置（直前タスク周期 *5 での計算値） 1: フィードバック位置（同一タスク周期 *5 での取得値）

*1. 終了位置より小さい値を指定してください。
*2. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*3. 開始位置より大きい値を指定してください。
*4. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*5. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Enabled	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。

出力変数	名称	データ型	有効範囲	内容
InZone	ゾーン内	BOOL	TRUE, FALSE	軸位置がゾーン範囲内のときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Enabled	「Enable」が TRUE となったとき。	「Enable」が FALSE となったとき。
InZone	ゾーン内に入ったとき。	• ゾーンから出たとき。 • 「Enable」が FALSE となったとき。
Busy	「Enable」の立ち上がり	• 「Error」が TRUE となったとき。 • 「Enable」が FALSE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 本命令が「Enable」（有効）の状態で、軸の指令位置、またはフィードバック現在位置が「FirstPosition」（開始位置）以上、「LastPosition」（終了位置）以下のゾーン内に存在する場合、出力変数の「InZone」（ゾーン内）が TRUE となります。
「RererenceType」（位置種別選択）にて、監視対象の軸情報を指令位置もしくはフィードバック位置のいずれかに設定できます。
- すべての軸種別で、ゾーン監視が可能です。
- 「Enable」（有効）が TRUE の状態で、「FirstPosition」（開始位置）や「LastPosition」（終了位置）を書き替えた場合、書き替えを行った周期から新しい値が適用されます。
- 1 軸に対して複数のゾーンを設定でき、ゾーンに重複があっても設定可能です。また、ソフトウェアリミット範囲外にもゾーンを設定できます。

使用上の注意

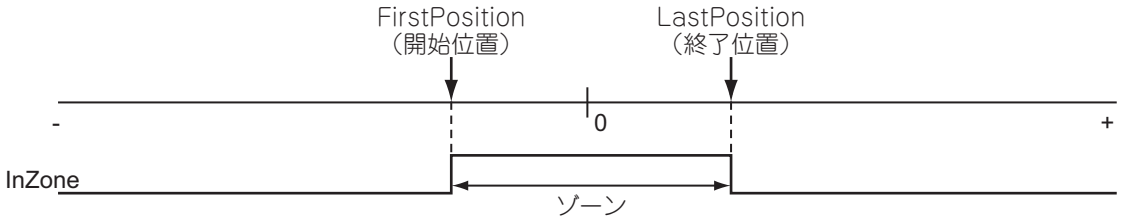
「FirstPosition」（開始位置）、「LastPosition」（終了位置）が割り切れない除算結果などを含んでいると、誤差のため、処理結果が期待どおりにならないことがあります。

命令の詳細

「FisrtPosition」（開始位置）と「LastPosition」（終了位置）は、カウントモードに応じて次の関係が成立するように設定してください。関係が成立しない場合は、異常となります。

● リニアモード

「FirstPosition」（開始位置）≦ 「LastPosition」（終了位置）となるように指定してください。

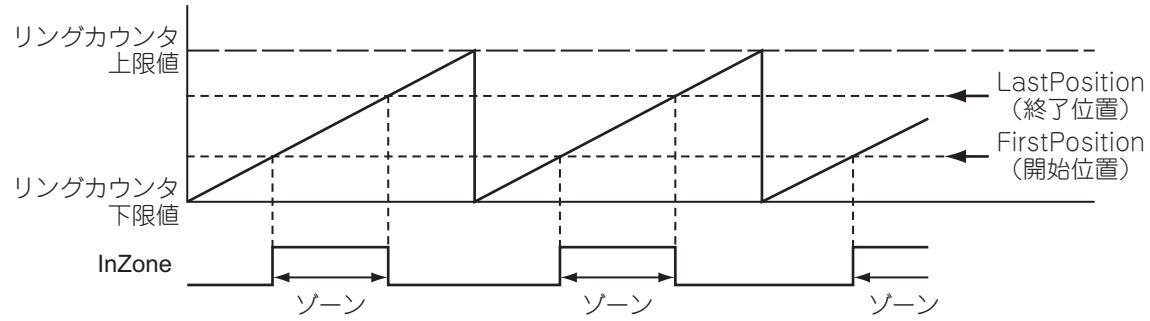


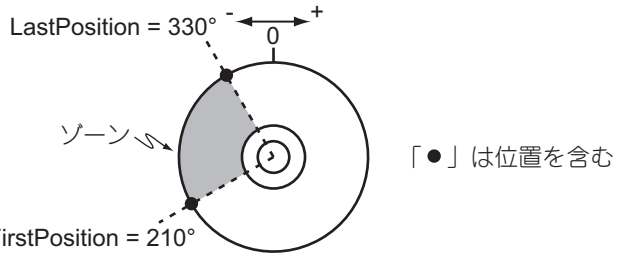
● ロータリモード

［ロータリモード］の場合は、リングカウンタの上下限位置をまたぐ場合と、またがない場合で下記のように異なります。

上下限位置をまたがない場合

「FirstPosition」（開始位置）≦ 「LastPosition」（終了位置）となるように指定してください。

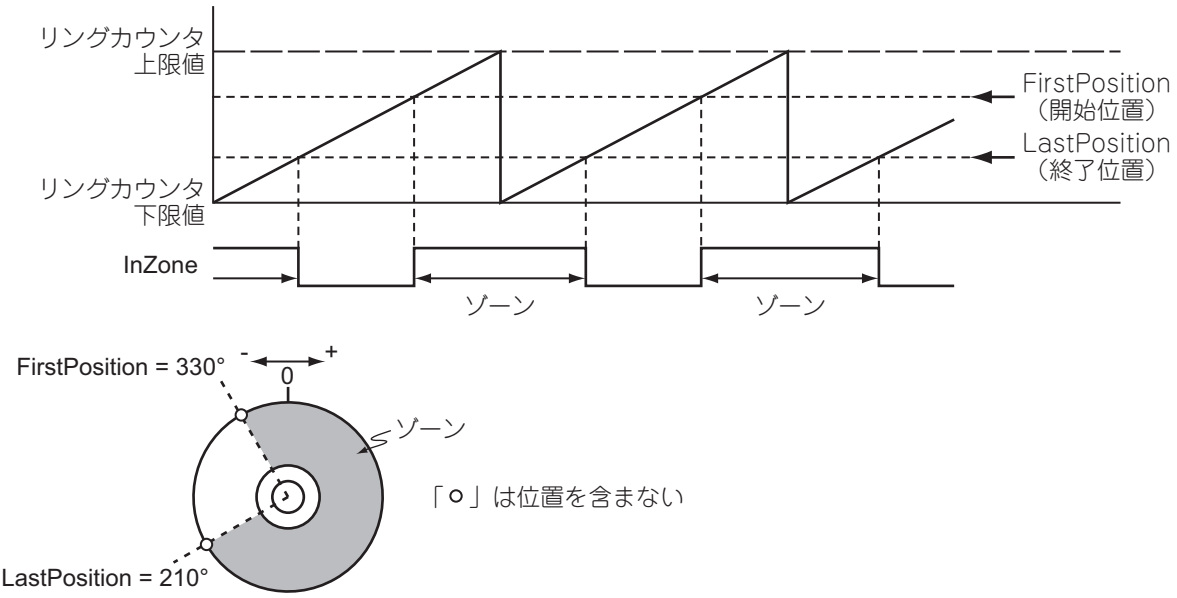




「●」は位置を含む

上下限位置をまたぐ場合

「FirstPosition」（開始位置）＞ 「LastPosition」（終了位置）となるように指定してください。



● ReferenceType（位置種別選択）

位置の種類を、以下から選択できます。

- _mcCommand: 指令位置（直前タスク周期での計算値）
  現周期に対して、直前のタスク周期で計算した主軸の指令位置を使用します。
  従軸の指令位置を計算する直前の定周期タスクで、計算した主軸の指令位置を使用します。
- _mcFeedback: 同一タスク周期での取得値
  同一タスク周期で取得した主軸のフィードバック位置を使用します。

使用上の注意

ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。同様に定周期タスクは、プライマリ定周期タスクまたは定周期タスク（実行優先度 5）を表します。

● 軸種別と位置種別の関係

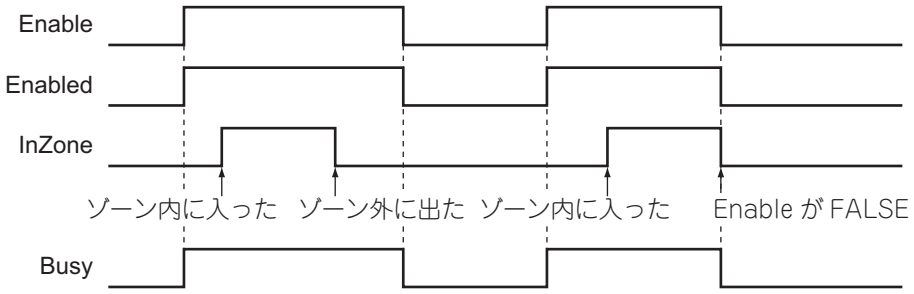
監視可能な軸種別と監視する位置種別の関係は以下のようになります。

軸種別	ReferenceType	
	_mcCommand	_mcFeedback
サーボ軸	○	○
エンコーダ軸	× *1	○
仮想サーボ軸	○	○
仮想エンコーダ軸	× *1	○

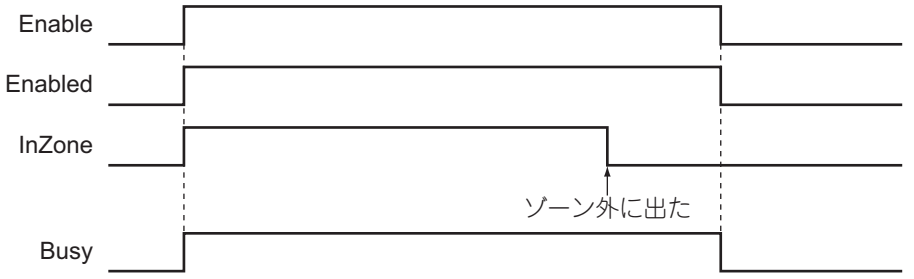
*1. 命令起動時に、「位置種別選択範囲外（エラーコード: 5430 Hex）」が発生します。

タイミングチャート

- **動作中にゾーン内に入った場合、およびゾーン内で「Enable」（有効）が FALSE になった場合**



- **命令起動前にゾーン内に位置し、途中でゾーンから出た場合**



モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

モーション命令多重起動

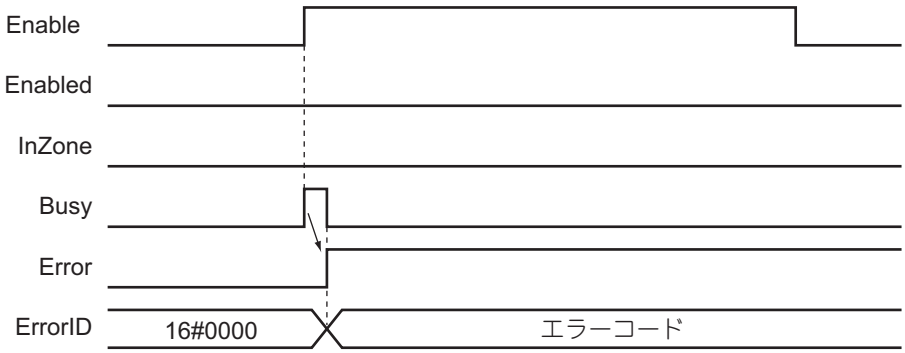
本命令は個々の命令インスタンスごとに独立して動作するため、モーション命令多重起動の制約はありません。

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート

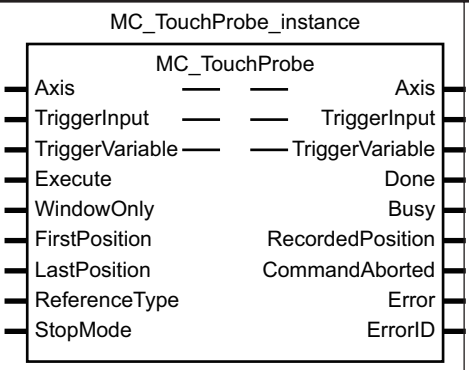


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_TouchProbe

トリガ信号の発生により、軸の位置を記録します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_TouchProbe	外部ラッチ有効	FB		MC_TouchProbe_instance (Axis :=「パラメータ」, TriggerInput :=「パラメータ」, TriggerVariable :=「パラメータ」 , Execute :=「パラメータ」, WindowOnly :=「パラメータ」, FirstPosition :=「パラメータ」, LastPosition :=「パラメータ」, ReferenceType :=「パラメータ」, StopMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, RecordedPosition =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
WindowOnly	ウィンドウ有効	BOOL	TRUE, FALSE	FALSE	ウィンドウマスクの有効／無効を指定します。
FirstPosition	開始位置	LREAL	負数、正数、「0」	0	トリガの受付を開始する位置を指定します。「ReferenceType」（位置種別選択）で指定した種別を使用します。*1 単位は［指令単位］となります。*2
LastPosition	終了位置	LREAL	負数、正数、「0」	0	トリガの受付を終了する位置を指定します。「ReferenceType」（位置種別選択）で指定した種別を使用します。*1 単位は［指令単位］となります。*2
ReferenceType（Reserved）	位置種別選択	_eMC_REFERENCE_TYPE	1: _mcFeedback	1 *3	（Reserved）

入力変数	名称	データ型	有効範囲	初期値	内容
StopMode	停止方法選択	_eMC_STOP_MODE	1: _mcImmediateStop 4: _mcNonStop	4 *3	停止方法を指定します。 1: 即停止します 4: 停止しません

*1. 詳細については、「WindowOnly（ウィンドウ有効）（P.3-344)」を参照してください。
*2. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
RecordedPosition	ラッチ位置	LREAL	負数、正数、「0」	ラッチした位置を出力します。 単位は［指令単位］となります。*1
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*2	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	• トリガ信号発生後、ラッチ位置を記録して命令が完了したとき。 • 停止指定有効時は、上記かつ、軸がラッチ位置に停止したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（［アボーティング］）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 「StopMode」を［_mcImmediateStop］として実行中に CSP モード以外に変更された場合 *1 • スレーブ離脱時 • スレーブ通信異常発生時（プロセスデータ通信中は除く） • MC_AbortTrigger 命令が起動されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

*1. NX シリーズ CPU ユニットを使用する場合は、制御モードによる制限はありませんので、CSP モード以外でも「CommandAborted」は発生しません。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1
TriggerInput	トリガ入力条件	_sTRIGGER_REF	−	トリガ条件を設定します。*2
TriggerVariable	トリガ変数	BOOL	TRUE, FALSE	トリガ条件でコントローラモードを指定したとき、トリガとなる入力変数を指定します。

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. _sTRIGGER_REF 型のユーザ定義変数を定義してください。

● _sTRIGGER_REF

メンバ変数	名称	データ型	有効範囲	機能
Mode	モード	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	トリガのモードを指定します。 0: ドライバモード 1: コントローラモード
LatchID	ラッチ ID 選択	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	[ドライバモード] のときに 2 つのラッチ機能のうちのどちらを使用するかを指定します。 0: ラッチ機能 1 1: ラッチ機能 2
InputDrive	ドライブトリガ入力信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	[ドライバモード] のときにサーボドライバのトリガ信号を指定します。 0: Z 相 1: 外部入力

機能説明

- センサ入力などのトリガの発生を基点とした位置の制御を行うことを目的とし、軸の位置をトリガの発生により記録（ラッチ）します。
- トリガ設定に従い、「Axis」（軸）で指定された軸の位置を「RecordedPosition」（ラッチ位置）に出力します。
  トリガ設定として、「TriggerInput」（トリガ入力条件）、「WindowOnly」（ウィンドウ有効）、「FirstPosition」（開始位置）、「LastPosition」（終了位置）、「StopMode」（停止方法選択）を指定することができます。
- 「RecordedPosition」（ラッチ位置）の出力値は同じ MC_TouchProbe（外部ラッチ有効）命令のインスタンスで再度、軸位置が記録されるまで保持されます。

● データオブジェクトのマッピング

「Mode」に［ドライバモード］を選択して、MC_TouchProbe（外部ラッチ有効）命令を使用する場合は、以下のオブジェクトデータをマッピングしてください。
マッピングは、Sysmac Studio の軸基本設定の［詳細設定］で行います。

軸種別がサーボ軸の場合

- ラッチ機能(60B8Hex)
- ラッチステータス(60B9Hex)
- ラッチ 1 位置(60BAHex)
- ラッチ 2 位置(60BCHex)

軸種別がエンコーダ軸の場合

- ラッチ機能(4020Hex)
- エンコーダ入力スレーブソフトスイッチ(4020Hex)
- ラッチステータス(4030Hex)
- ラッチ位置 1(4012Hex)
- ラッチ位置 2(4013Hex)
- エンコーダ入力スレーブステータス(4030Hex)

これらを設定していない場合、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。

データオブジェクトのマッピングについては、「2-3 PDO マッピング (P.2-32)」、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。
NX シリーズ 位置インタフェースユニットを使用する場合は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』の「I/O エントリマッピング」を参照してください。
エンコーダ入力スレーブを使用する場合は、『GX シリーズ EtherCAT スレーブ ユーザーズマニュアル(SBCD-350)』の「固定 PDO マッピング」を参照してください。

命令の詳細

以下に本命令の詳細について説明します。

● Axis（軸）指定

- 「Axis」（軸）にてラッチの対象とする軸を指定します。
- 指定した「Axis」（軸）が MC_GroupEnable（多軸協調制御）命令中は、MC_TouchProbe（外部ラッチ有効）命令は異常となり実行できません。
- 軸ごとに「LatchID」（ラッチ ID 選択）を指定し、同時に 2 つまでの MC_TouchProbe（外部ラッチ有効）命令を実行することができます。
- 「LatchID」（ラッチ ID 選択）は MC_AbortTrigger（外部ラッチ無効）命令で中止するラッチを指定する場合にも使用します。



参考

- オムロン製 GX シリーズ EtherCAT エンコーダ入力スレーブ 形 GX-EC02□□に接続したエンコーダ軸を使用して、本命令によるラッチを行うこともできます。
- NX シリーズ パルス出力ユニットを使用して、本命令によるラッチを行うこともできます。詳細は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。

● トリガ入力条件

「TriggerInput」（トリガ入力条件）の「Mode」、「LatchID」、「InputDrive」の指定によりトリガの条件を選択します。

Mode

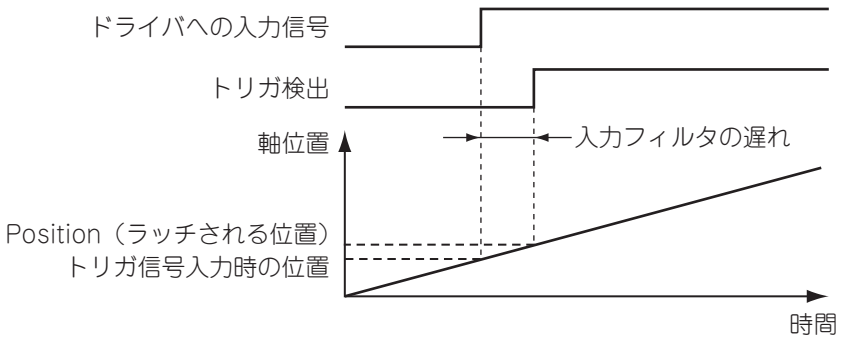
- 「Mode」の設定として、トリガにサーボドライバなどからの信号を指定する［ドライバモード］と、トリガに「TriggerVariable」（トリガ変数）を指定する［コントローラモード］があります。
- トリガ信号の立ち上がりをトリガ発生とみなし、ラッチ命令実行後の最初のトリガ発生（FALSE→TRUE）時の軸の位置を記録します。
- 本命令が「Busy」（動作中）中は「Execute」（起動）が FALSE となっても、「TriggerVariable」（トリガ変数）の変化をトリガとして動作します。

参考

オムロン製 GX シリーズ EtherCAT エンコーダ入力スレーブ 形 GX-EC02□□を使用する場合は、「Mode」は［_mcDrive］（ドライブモード）に設定してください。

ドライバモード

［ドライバモード］はサーボドライバなどの機能でトリガ検出と現在位置の取得を行うため［コントローラモード］に比べてラッチされる現在位置が正確です。



使用上の注意

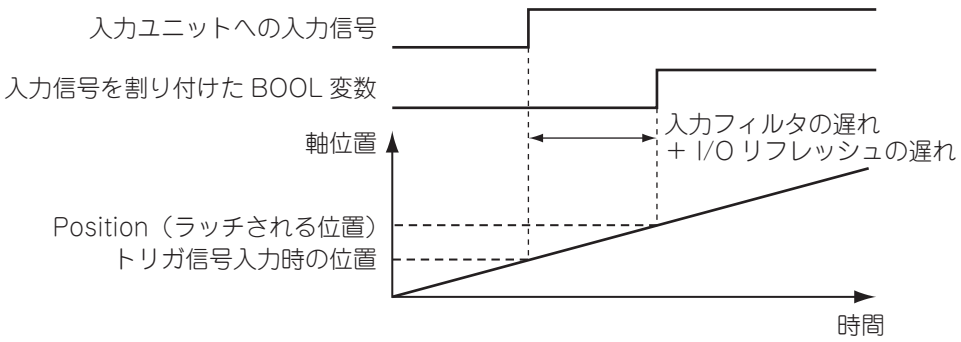
- ［ドライバモード］を使用する場合は、必ず使用する「LatchID」（ラッチ ID 選択）に、ラッチ信号を接続してください。
- ラッチ信号の幅は、サーボドライバなどの性能等に依存します。

参考

オムロン製 GX シリーズ EtherCAT エンコーダ入力スレーブ 形 GX-EC02□□を使用する場合、「InputDrive」には［_mcEXT］（外部入力）を指定してください。
オムロン製 GX シリーズ EtherCAT エンコーダ入力スレーブ 形 GX-EC02□□は Z 相によるラッチ機能を持っていません。そのため［_mcEncoderMark］（Z 相）を指定した場合、本命令起動時に異常となります。起動時に「Error」（エラー）が TRUE となり、「ErrorID」（エラーコード）に「プロセスデータオブジェクト設定不足（エラーコード: 3461 Hex）」が出力されます。

コントローラモード

- ［コントローラモード］では、トリガに BOOL 変数を指定することができます。
- 「TriggerVariable」（トリガ変数）にトリガとしたい BOOL 変数を指定します。
- ［コントローラモード］では、トリガ入力信号の BOOL 変数への反映が I/O リフレッシュの遅れにより［ドライバモード］よりも大きな遅れが生じます。



使用上の注意

［コントローラモード］を使用する場合、タスク周期の間隔でラッチ処理を行います。そのため、トリガ変数はタスク周期の 1 周期の間、TRUE でなければなりません。
また、トリガ変数が TRUE になったあと、MC 機能モジュールが処理するまでタスク周期の 1 周期分の時間が必要です。
ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。

LatchID

- 1 つの軸に対し同時に 2 つまで MC_TouchProbe（外部ラッチ有効）命令を実行することができます。
  「LatchID」（ラッチ ID 選択）で、そのどちらを使用するかを指定します。
- 同じ軸に対して指定したラッチ ID がすでに実行中の場合は、後から実行した命令が有効となります。それまでの命令は「CommandAborted」（実行中断）となります。
- サーボドライバなどでは、「LatchID」（ラッチ ID 選択）がサーボドライバのラッチ回路 1（第 1ch）、ラッチ回路 2（第 2ch）を意味します。

「LatchID」については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編 (SBCE-433)』を参照してください。

参考

オムロン製 GX シリーズ EtherCAT エンコーダ入力スレーブ 形 GX-EC02□□を使用する場合、以下の対応になります。

列挙子	エンコーダ入力ターミナルの信号名
_mcLatch1	外部ラッチ入力 A
_mcLatch2	外部ラッチ入力 B

InputDrive

- トリガとして_mcEncoderMark（Z 相）と_mcEXT（外部入力）の選択が可能です。
- _mcEncoderMark（Z 相）では、サーボドライバなどの Z 相をトリガに使用します。
  _mcEXT（外部入力）では、サーボドライバに入力された外部トリガ信号をトリガに使用します。
- _mcEXT（外部入力）は、オムロン製サーボドライバ 1S シリーズでは、設定されている Ext1, Ext2 のどちらかになります。G5 シリーズでは、設定されている Ext1, Ext2, Ext3 のうちの 1 つになります。設定は Sysmac Studio により行います。
  ドライバに設定される 2 つのトリガは同じ設定でも可能です。

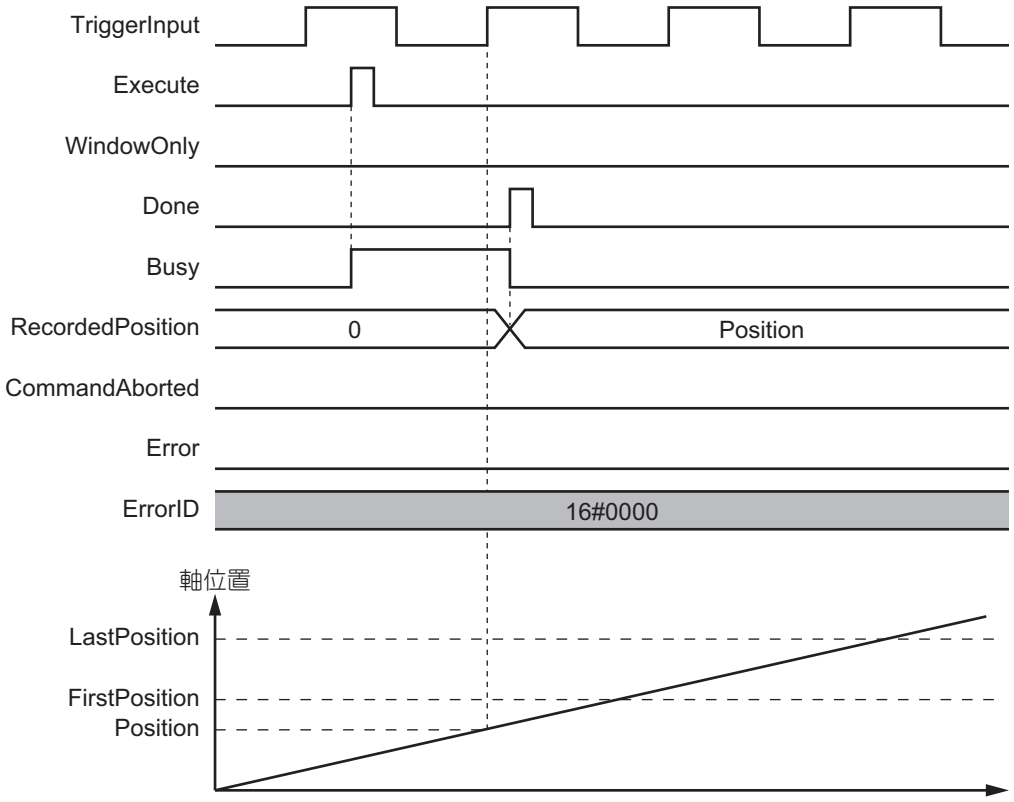
● WindowOnly（ウィンドウ有効）

- 「WindowOnly」（ウィンドウ有効）ではウィンドウの有効／無効を指定します。
- 「WindowOnly」（ウィンドウ有効）が FALSE の場合、すべての軸位置においてトリガを検出します。
- 「WindowOnly」（ウィンドウ有効）が TRUE の場合、軸位置が「FirstPosition」（開始位置）と「LastPosition」（終了位置）の範囲内にあるときだけトリガを検出します。

「WindowOnly」（ウィンドウ有効）の指定による動作の違いをタイミングチャートで示します。

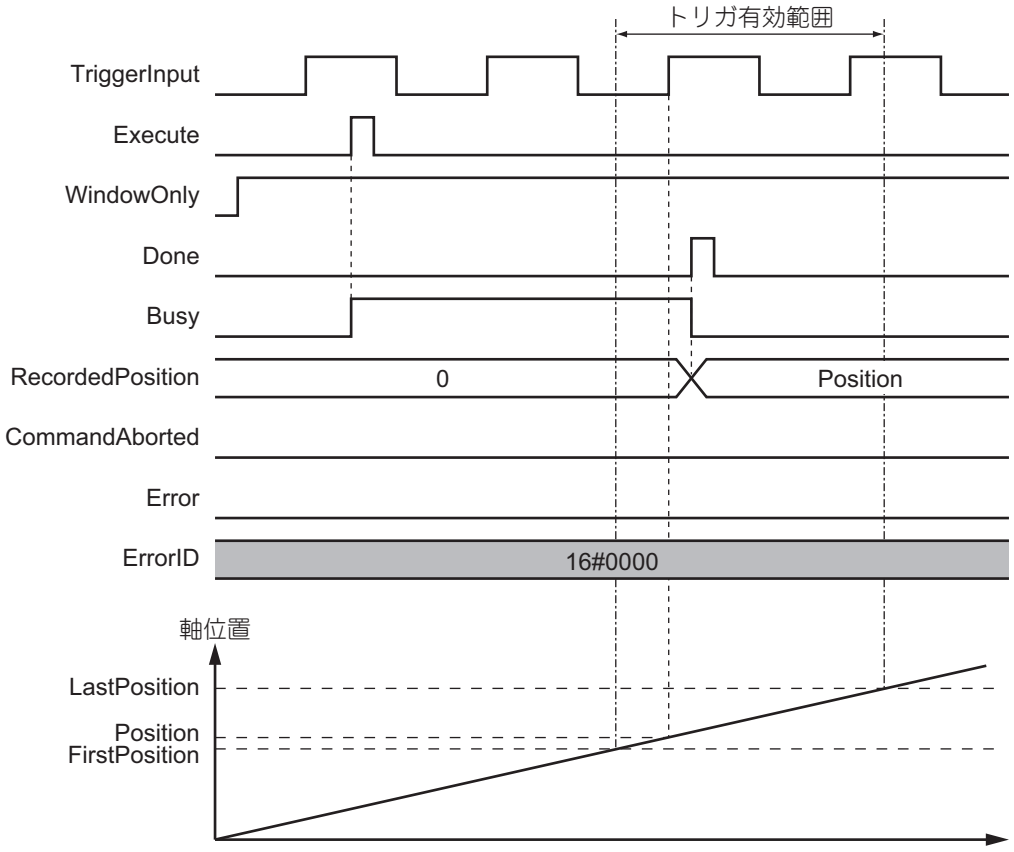
「WindowOnly」が FALSE の場合

「Execute」（起動）が TRUE となった後の最初のトリガ発生時の軸位置を「RecordedPosition」（ラッチ位置）に出力します。



「WindowOnly」が TRUE の場合

ウィンドウの範囲内だけ、トリガ入力を検出して軸位置を取得します。



使用上の注意

- 「WindowOnly」（ウィンドウ有効）が FALSE→TRUE に変化する瞬間とラッチ機能が動作するまでの時間はラッチできません。
- ラッチ機能が動作するまでに時間が必要なため、「WindowOnly」（ウィンドウ有効）の有効範囲がきわめて短い場合はラッチすることができません。ラッチ可能な有効範囲の限界値は、サーボドライバ、エンコーダ入力ターミナル、位置インタフェースユニットなどの性能および EtherCAT の通信に依存します。

また、カウントモードの違いにより、「FirstPosition」（開始位置）と「LastPosition」（終了位置）の範囲は以下のようになります。

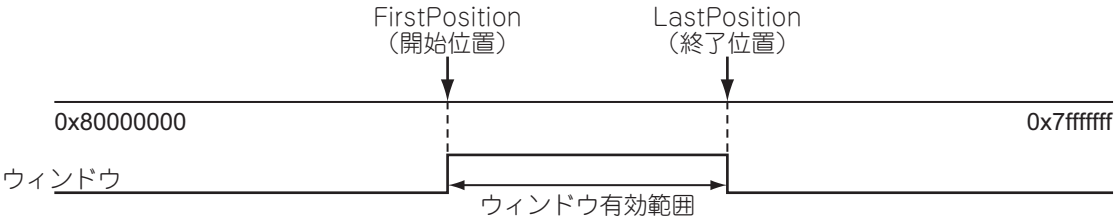
リニアモード

- ウィンドウの有効範囲は、以下の式で表されます。

  「FirstPosition」（開始位置） ≦ ウィンドウ範囲 ≦ 「LastPosition」（終了位置）

- 「FirstPosition」（開始位置） ＞ 「LastPosition」（終了位置）と指定した場合は異常となります。
- ［リニアモード］の位置範囲を超えて指定した場合も異常となります。
- また「FirstPosition」（開始位置）と「LastPosition」（終了位置）は LREAL 型の変数のため、「FirstPosition」（開始位置） ＝ 「LastPosition」（終了位置）を指定しないでください。
  LREAL 型については、「実数（REAL 型、LREAL 型）の扱い (P.1-12)」を参照してください。

［リニアモード］のときの、ウィンドウ有効範囲は下図のようになります。

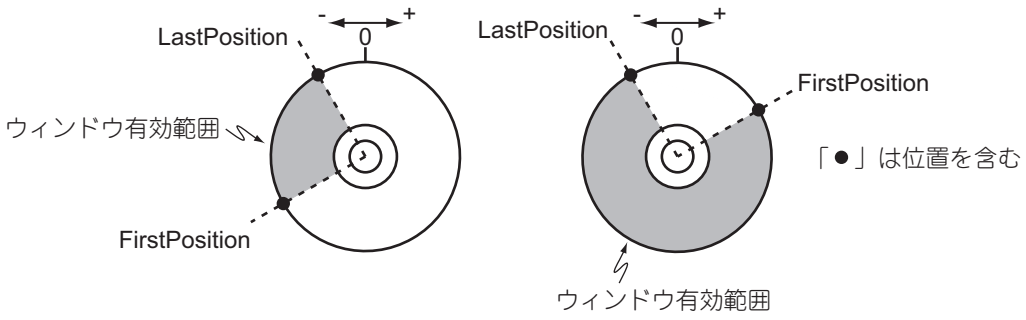


（注）「FirstPosition」（開始位置）と「LastPosition」（終了位置）は、ウィンドウ有効範囲に含まれます。

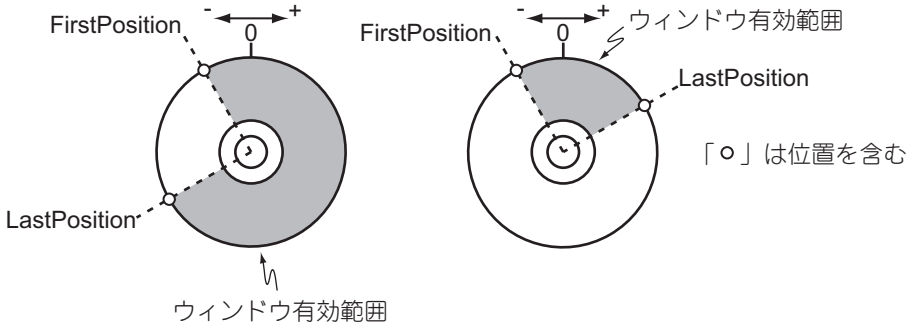
ロータリモード

- 「FirstPosition」（開始位置） ≦ 「LastPosition」（終了位置）、および「FirstPosition」（開始位置） ＞ 「LastPosition」（終了位置）の両方が指定可能です。
- 「FirstPosition」（開始位置） ＞ 「LastPosition」（終了位置）の場合はリングカウンタの上下限位置をまたぐ設定となります。
- リングカウンタ上下限の範囲を超えて指定した場合は異常となります。

FirstPosition ≦ LastPosition



FirstPosition > LastPosition



● StopMode（停止方法選択）

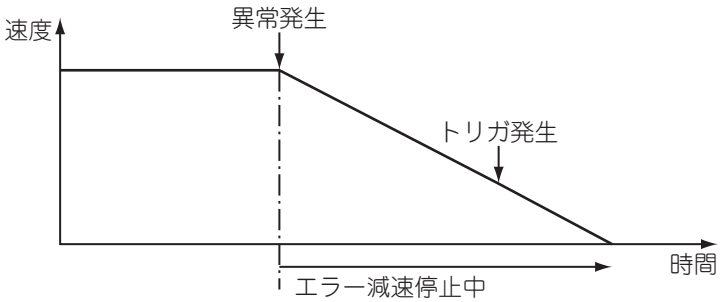
- トリガ発生時に、指定した「Axis」（軸）に対して「StopMode」（停止方法選択）を指定することができます。
- _mcNonStop（非停止）は、トリガが発生しても軸を停止させません。
- _mcImmediateStop（即停止）は、トリガ発生で軸をラッチした位置に即停止します。本停止により、その軸を動作させていた命令は「CommandAborted」（実行中断）となります。
- _mcImmediateStop（即停止）は、軸の指令が停止した時点で、「Done」（完了）となります。ラッチした位置へ停止するまで「Busy」（実行中）が TRUE となっています。
- _mcImmediateStop（即停止）による停止時のインポジションチェックは行いません。

参考

- オムロン製サーボドライバ 1S シリーズを使用する場合は、CPU ユニットのユニットバージョンにかかわらず、次に説明する Ver.1.10 以降の CPU ユニットと同じ動作となります。
- オムロン製サーボドライバ G5 シリーズを使用する場合は、CPU ユニットのユニットバージョンにより、次に説明するように動作が異なります。

ユニットバージョン Ver.1.10 以降の CPU ユニットの場合

- _mcImmediateStop（即停止）を指定した軸で軸の異常が発生し、軸の停止が完了する前にトリガが発生すると、軸は減速停止を継続します。また、トリガ発生により、「CommandAborted」（実行中断）となります。

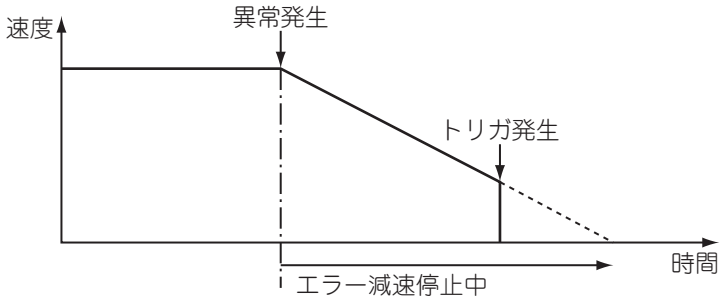


使用上の注意

- _mcImmediateStop（即停止）は、MC 機能モジュールが停止処理を行います。実際にはラッチ位置を超えて停止し、コントローラの指示によりラッチした位置まで戻りますので、注意してください。
- _mcImmediateStop（即停止）を使用する場合、指令速度が速いと、ラッチした位置に戻る量が大きくなります。指令速度が速くならないように注意して設定してください。
- エンコーダ軸には_mcNonStop（非停止）を指定してください。_mcImmediateStop（即停止）を指定した場合、本命令起動時に「Error」（エラー）が TRUE となります。同時に「ErrorID」（エラーコード）には「外部ラッチ有効命令実行不可（エラーコード: 5492 Hex）」を出力します。

ユニットバージョン Ver.1.09 以前の CPU ユニットの場合

- _mcImmediateStop（即停止）を指定した軸で軸の異常が発生し、軸の停止が完了する前にトリガが発生すると、オムロン製サーボドライバ G5 シリーズの機能により軸は即停止します。また、トリガ発生により、「CommandAborted」（実行中断）となります。
- モーション制御機能モジュールは、オムロン製サーボドライバ G5 シリーズの停止を検出後、指令停止を即停止させます。下図の点線は、トリガ未発生の目標軌道です。



使用上の注意

- _mcImmediateStop（即停止）は、CSP モードのときに機能します。
- CSV/CST モードの状態で、_mcImmediateStop（即停止）を指定して起動した場合、「外部ラッチ有効命令実行不可（エラーコード: 5492Hex）」となります。また、_mcImmediateStop（即停止）で動作中に CSV/CST モードに変更した場合には「CommandAborted」（実行中断）となります。
- _mcImmediateStop（即停止）は、オムロン製サーボドライバ G5 シリーズまたは NX シリーズ パルス出力ユニットを使用する場合には、サーボドライバまたは NX シリーズ パルス出力ユニットの機能を使用し停止させます。
  オムロン製サーボドライバ 1S シリーズ、および他社製サーボドライバの場合には、MC 機能モジュールが停止処理を行います。
  いずれの場合も、実際にはラッチ位置を超えて停止し、コントローラの指示によりラッチした位置まで戻りますので、注意してください。
- _mcImmediateStop（即停止）を使用する場合、指令速度が速いと、ラッチした位置に戻る量が大きくなります。
  指令速度が速くならないように注意して設定してください。
- エンコーダ軸には_mcNonStop（非停止）を指定してください。_mcImmediateStop（即停止）を指定した場合、本命令起動時に「Error」（エラー）が TRUE となります。同時に「ErrorID」（エラーコード）には「外部ラッチ有効命令実行不可（エラーコード: 5492Hex）」を出力します。
- _mcImmediateStop（即停止）を使用する場合に、以下の 2 つの処理が同じ制御周期で実行されると、オムロン製サーボドライバ G5 シリーズは Target Ignore 状態となります。
  a）ラッチ信号 ON 時のサーボドライバのカウンタラッチ処理
  b）MC_TouchProbe（外部ラッチ有効）命令で、「CommandAborted」（実行中断）が TRUE になったときのラッチ解除処理

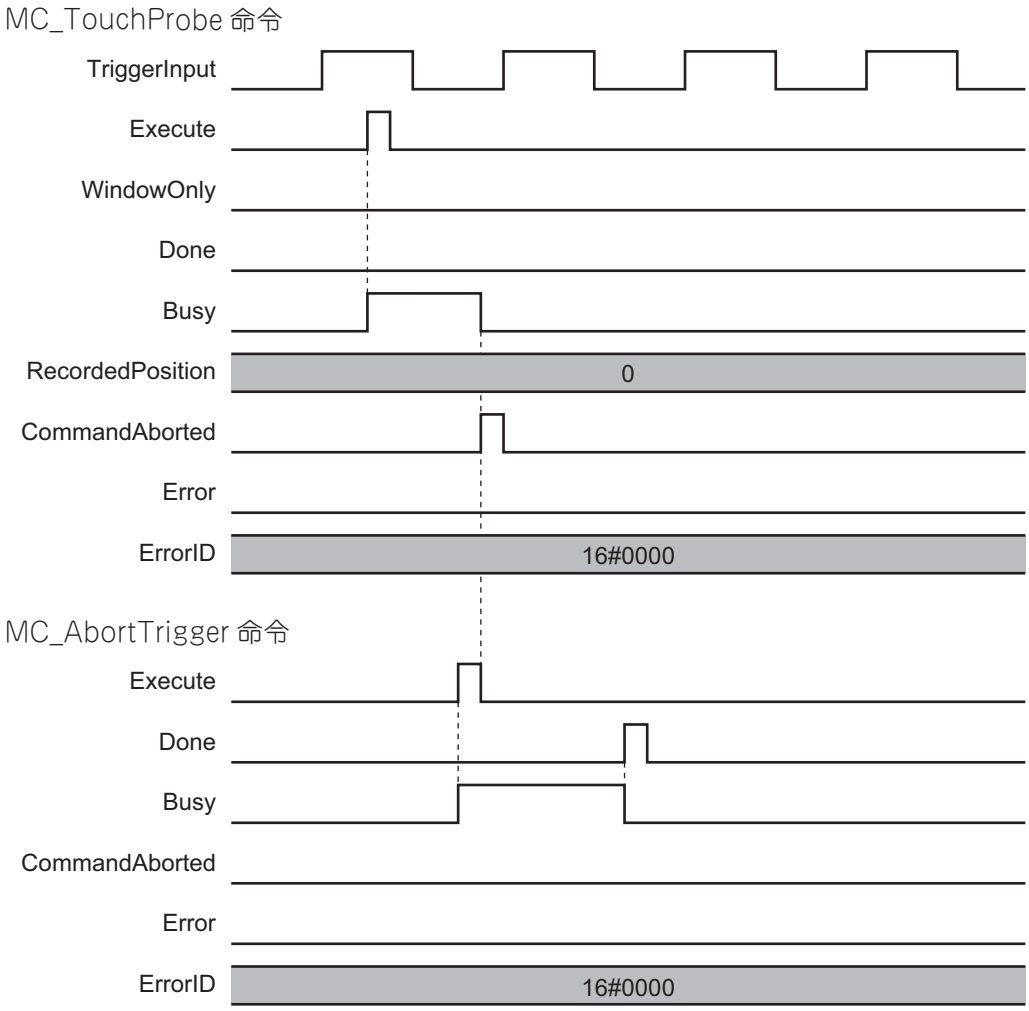
ラッチ信号が ON になるタイミングで、本命令の「CommandAborted」（実行中断）が TRUE にならないように注意してください。

● 軸変数のステータス

「StopMode」（停止方法選択）に_mcImmediateStop（即停止）を指定した場合、トリガ発生による停止処理中は、軸変数の「Status.Stopping」（減速停止中）が TRUE となります。

命令の中止

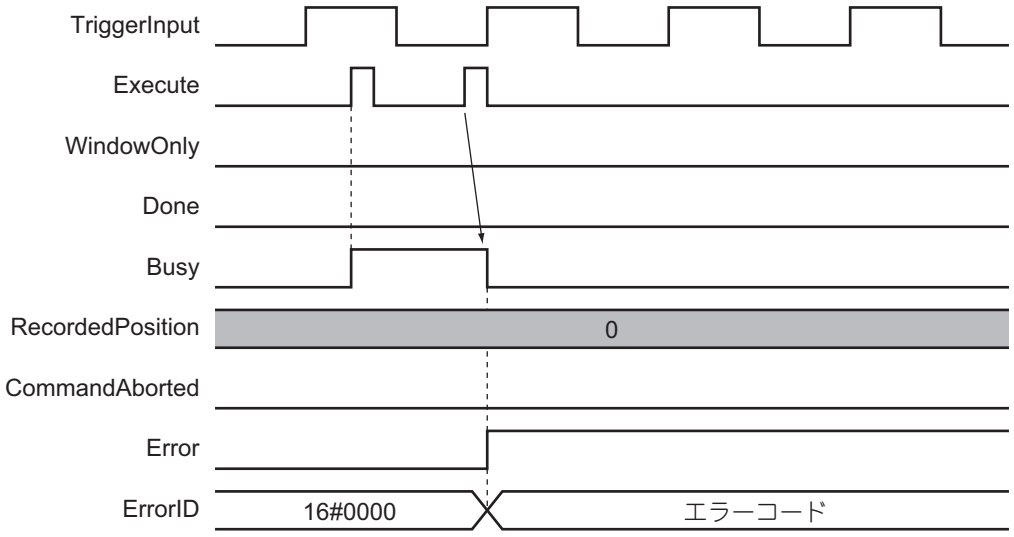
MC_TouchProbe（外部ラッチ有効）命令は、MC_AbortTrigger（外部ラッチ無効）命令により終了することができます。
中止させたい「Axis」（軸）と「LatchID」（ラッチ ID 選択）を MC_AbortTrigger（外部ラッチ無効）命令に指定して命令を起動することにより、終了します。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

同じ MC_TouchProbe（外部ラッチ有効）命令のインスタンスに対して「RecordedPosition」（ラッチ位置）の読み出し完了前に、再度「Execute」（起動）を FALSE → TRUE にした場合は、異常となります。

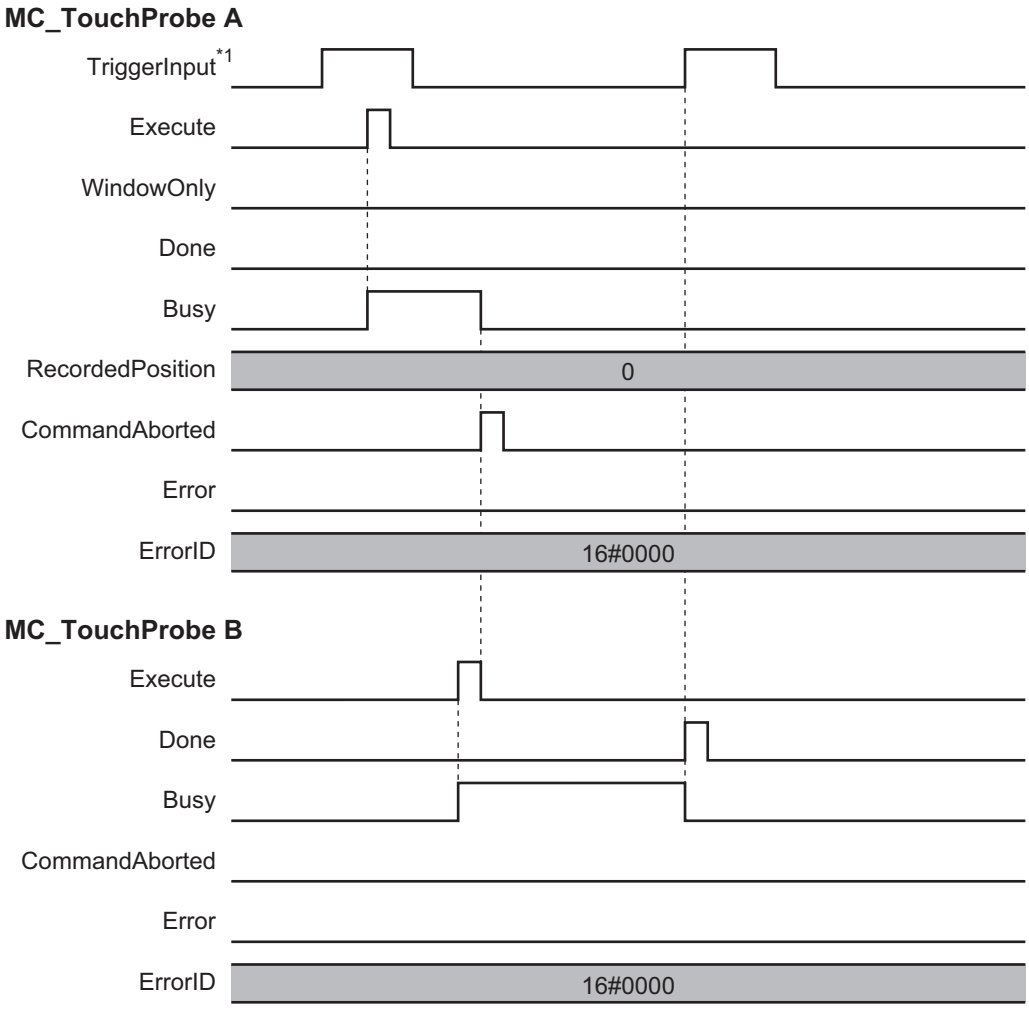


モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

同じ「Axis」（軸）の 1 つの「LatchID」（ラッチ ID 選択）に対しては同時に 1 つのトリガ設定しか実行することができません。
MC_TouchProbe（外部ラッチ有効）命令実行中に、同じ「LatchID」（ラッチ ID 選択）に対して別の MC_TouchProbe（外部ラッチ有効）命令のインスタンスが実行された場合、1 つ目が「CommandAborted」（実行中断）され、2 つ目が新規に実行されます。



*1. サーボドライバなどのトリガ入力信号の場合です。

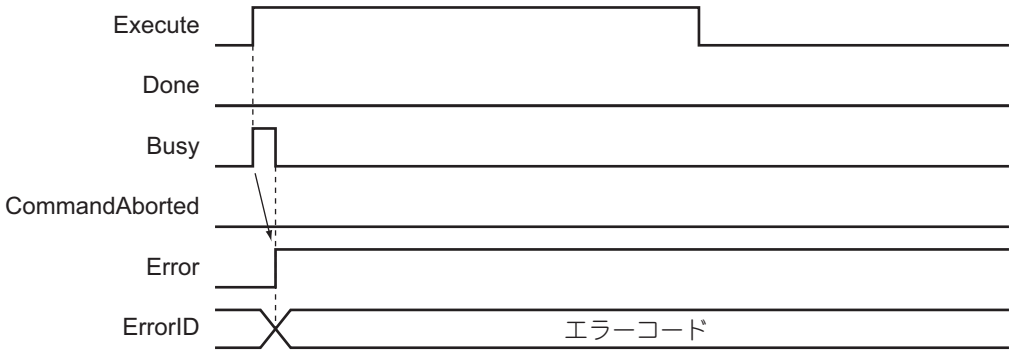
参考

「StopMode」（停止方法選択）に_mcImmediateStop（即停止）を指定した場合、トリガ発生による停止時に、多重起動された命令の「CommandAborted」（実行中断）が TRUE となります。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● **異常発生時のタイミングチャート**



● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、外部センサによる位置の取り出し制御についてサンプルプログラムを掲載します。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● **軸パラメータ**

軸種別

軸	軸種別
軸 1	サーボ軸

カウンタモード

軸	カウンタモード
軸 1	ロータリモード

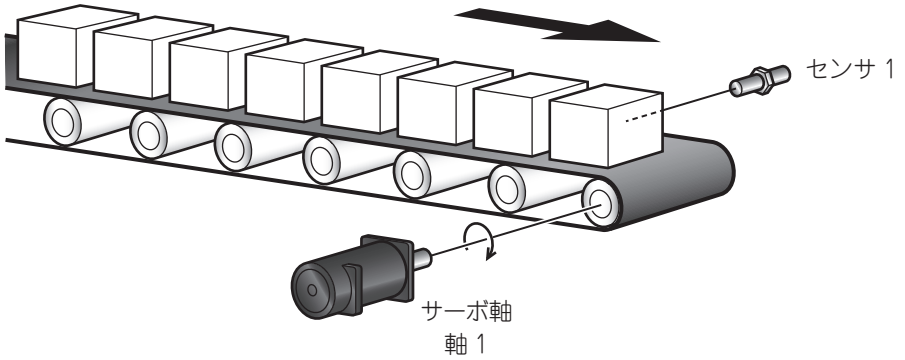
リングカウンタ

軸	上限値	下限値
軸 1	360	0

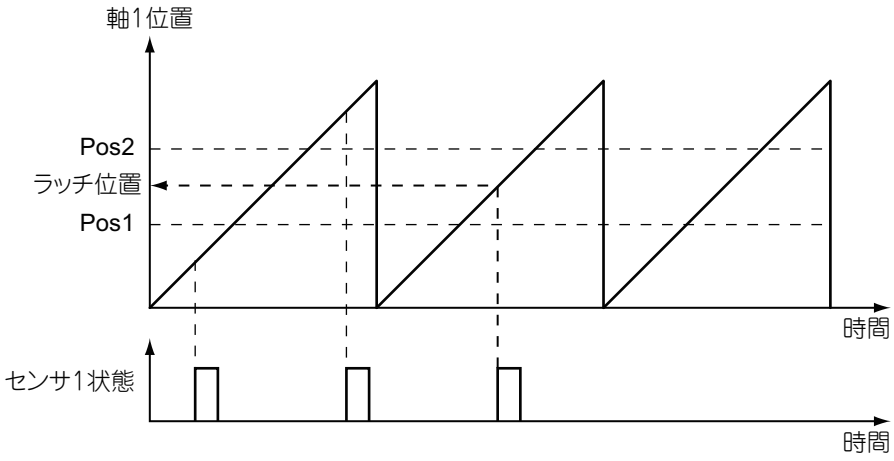
座標単位選択

軸	座標単位
軸 1	degree

動作例



● 動作パターン



1 **主軸の開始**
軸 1 は速度制御を行います。

2 **ワークの検出**
センサ 1 でワークの有無を検出します。

3 **位置の取得**
ワークがウィンドウの TRUE の範囲内（Pos1～Pos2）で検出されたら、その時の軸 1 のラッチ位置を取得します。

ラダー図

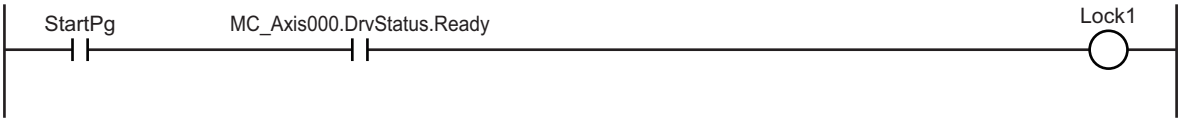
● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。

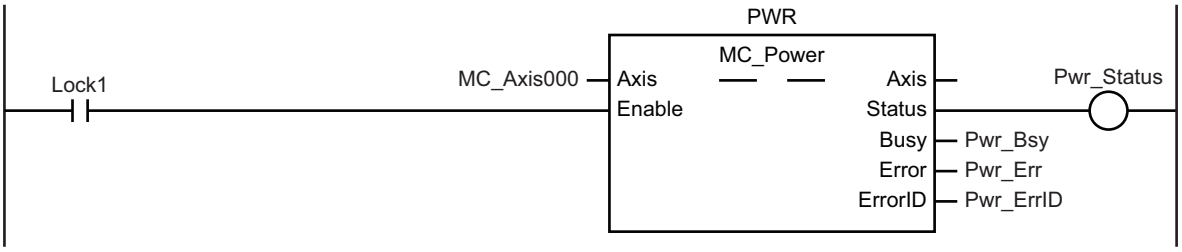
名称	データ型	初期値	コメント
Pwr_Status	BOOL	FALSE	MC_Power のインスタンス PWR の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pos1	LREAL	−	ウィンドウ有効の開始位置を示します。
Pos2	LREAL	−	ウィンドウ有効の終了位置を示します。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● サンプルプログラム

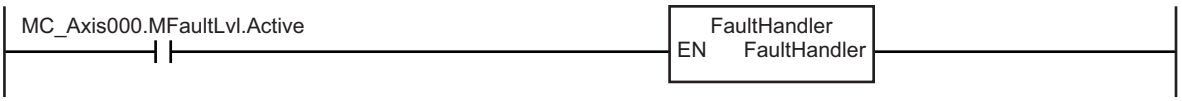
接点 StartPg が TRUE になったら、サーボドライバがサーボレディ状態になっていることを確認する



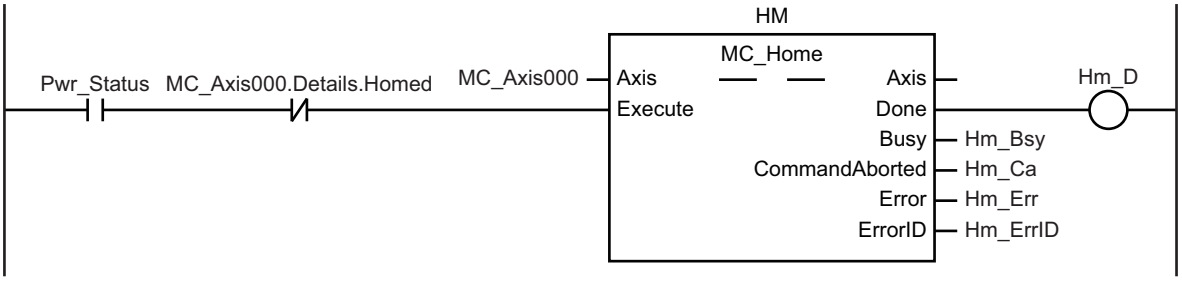
軸 1 がサーボレディ状態になっていれば、サーボ ON 状態にする



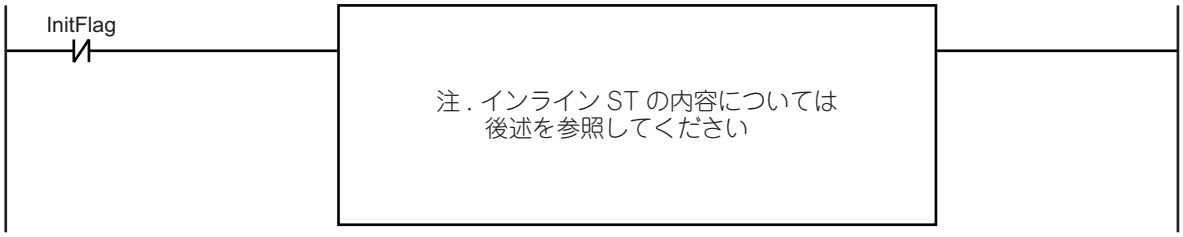
軸 1 で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



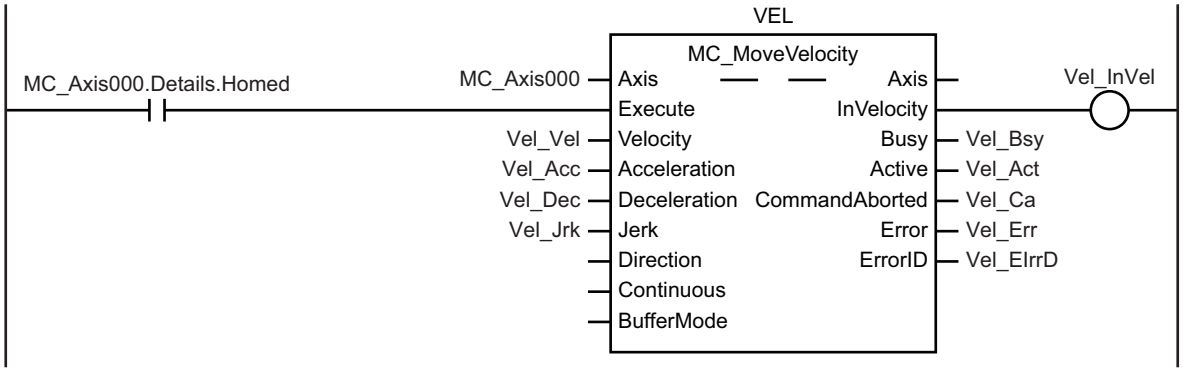
サーボ ON 状態になり原点未確定状態なら、原点復帰を実行する



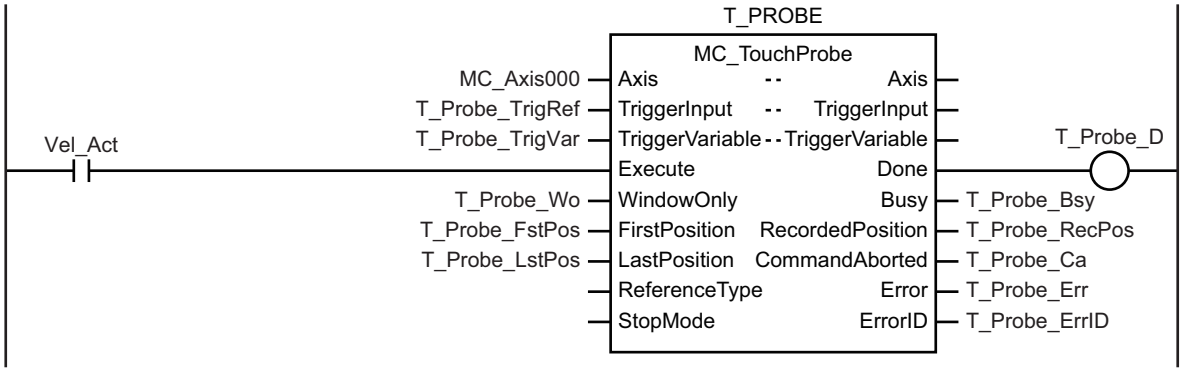
MC_MoveVelocity（速度制御）命令と MC_TouchProbe（外部ラッチ有効）命令のパラメータを設定する



軸 1 が原点確定状態になったら、MC_MoveVelocity(速度制御）命令を実行する



MC_MoveVelocity(速度制御）命令が開始されたら、ラッチ処理を実行する



インライン ST の内容

```
//MC_MoveVelocity パラメータ
Vel_Vel := LREAL#1000.0;
Vel_Acc := LREAL#1000.0;
Vel_Dec := LREAL#1000.0;
Vel_Jrk := LREAL#1000.0;

//MC_TouchProbe パラメータ
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;
```

```
//入力パラメータを設定後、InitFlag を TRUE にします
InitFlag:=TRUE;
```

ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき、TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power のインスタンス PWR の出力変数「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pos1	LREAL	−	ウィンドウ有効の開始位置を示します。
Pos2	LREAL	−	ウィンドウ有効の終了位置を示します。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag=FALSE THEN

    //MC_MoveVelocity パラメータ
    Vel_Vel := LREAL#1000.0;
    Vel_Acc := LREAL#1000.0;
    Vel_Dec := LREAL#1000.0;
    Vel_Jrk := LREAL#1000.0;

    //MC_TouchProbe パラメータ
    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;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag:=TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、サーボ ON 状態にします
```

```
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//軽度フォールトレベルの異常発生時の処理
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

//サーボ ON 状態で、かつ原点未確定なら原点復帰を実行します
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex:=TRUE;
END_IF;

//原点確定後に、MC_MoveVelocity を実行します
IF MC_Axis000.Details.Homed=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

//MC_MoveVelocity が起動後、MC_TouchProbe を実行します
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

実行中のラッチを無効にします。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_AbortTrigger	外部ラッチ無効	FB	MC_AbortTrigger_instance MC_AbortTrigger Axis — — Axis TriggerInput — — TriggerInput Execute — Done Busy CommandAborted Error ErrorID	MC_AbortTrigger_instance (Axis :=「パラメータ」, TriggerInput :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	• ラッチの停止を実行したとき。 • ラッチ実行中でないラッチ機能に対して本命令を実行して処理を終了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後

変数	TRUE となるタイミング	FALSE となるタイミング
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	ラッチ対象となる軸を指定します。*1
TriggerInput	トリガ入力条件	_sTRIGGER_REF	−	トリガ条件を設定します。*2

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. _sTRIGGER_REF 型のユーザ定義変数を定義してください。

● _sTRIGGER_REF

メンバ変数	名称	データ型	有効範囲	機能
Mode	モード	_eMC_TRIGGER_MODE	0: _mcDrive 1: _mcController	トリガのモードを指定します。 0: ドライバモード 1: コントローラモード
LatchID	ラッチ ID 選択	_eMC_TRIGGER_LATCH_ID	0: _mcLatch1 1: _mcLatch2	[ドライバモード] のときに 2 つのラッチ機能のうちのどちらを使用するかを指定します。 0: ラッチ機能 1 1: ラッチ機能 2
InputDrive	ドライブトリガ入力信号	_eMC_TRIGGER_INPUT_DRIVE	0: _mcEncoderMark 1: _mcEXT	[ドライバモード] のときにサーボドライバのトリガ信号を指定します。 0: Z 相 1: 外部入力

機能説明

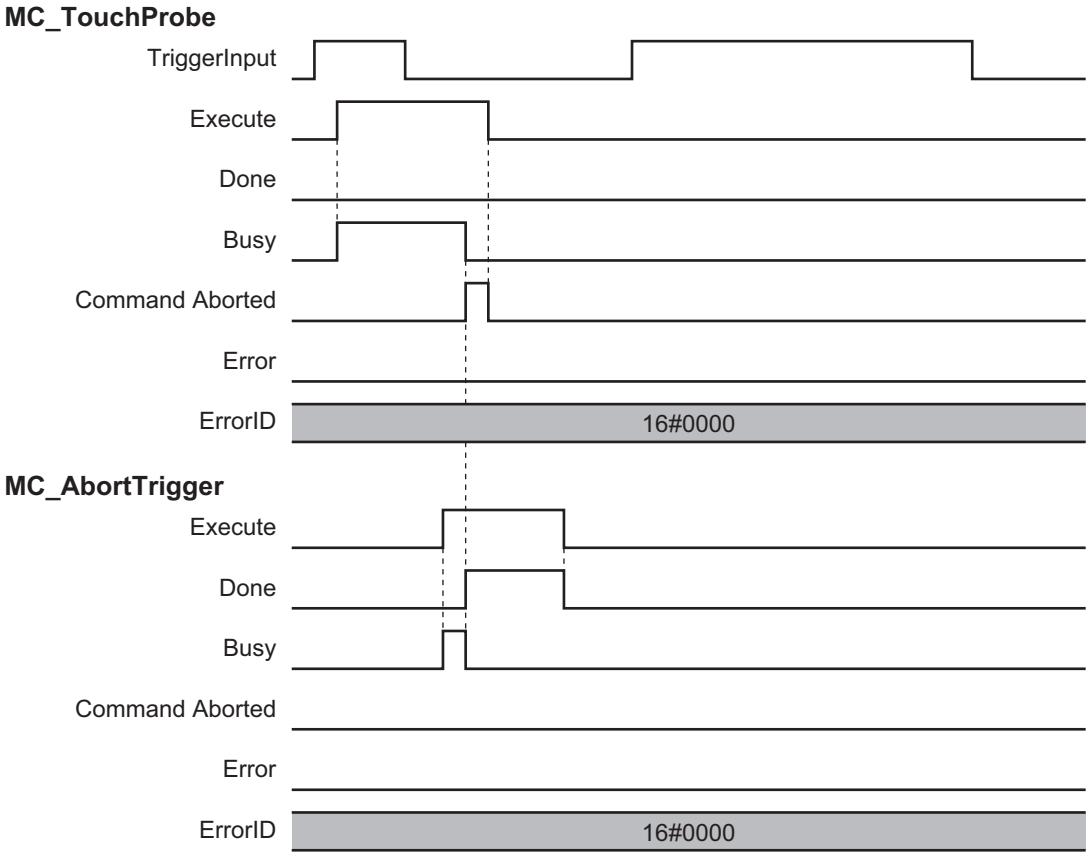
- 実行中のラッチを中止します。
- MC_AbortTrigger（外部ラッチ無効）命令では、「Axis」（軸）と「LatchID」（ラッチ ID 選択）を指定することにより中止するラッチを指定します。
- ラッチ要求が実行されていないトリガに対して MC_AbortTrigger（外部ラッチ無効）命令を実行した場合は、MC_AbortTrigger（外部ラッチ無効）命令は何もせずに正常終了します。
MC_TouchProbe（外部ラッチ有効）命令の実行が「Done」（完了）になったものに対して MC_AbortTrigger（外部ラッチ無効）命令を実行した場合も同様です。

使用上の注意

- MC_AbortTrigger（外部ラッチ無効）命令を指定した「Axis」（軸）が、MC_GroupEnable（軸グループ有効）命令有効の場合、MC_AbortTrigger（外部ラッチ無効）命令が異常となり実行できません。
- MC_Home（原点復帰）命令、MC_HomeWithParameter（パラメータ指定原点復帰）命令、MC_MoveFeed（割り込み定寸位置決め）命令、MC_MoveLink（台形パターンカム）命令実行中に、MC_AbortTrigger（外部ラッチ無効）命令を実行した場合、本命令は異常となります。

タイミングチャート

- MC_AbortTrigger（外部ラッチ無効）命令の「Done」（完了）は「Execute」（起動）の 1 周期後に TRUE となります。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

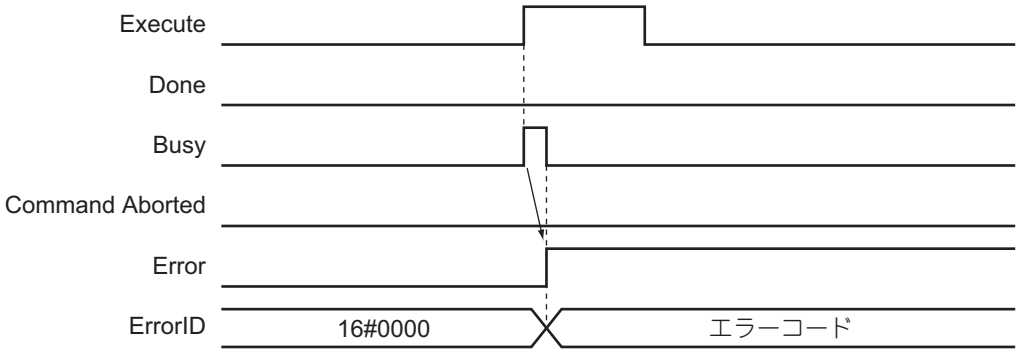
モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

MC_AbortTrigger



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_AxesObserve

指定された軸の指令位置、またはフィードバック位置の差分が許容値を超えることがないかを監視します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_AxesObserve	軸間偏差監視	FB	MC_AxesObserve_instance MC_AxesObserve Master — — Master Slave — — Slave Enable — Enabled ReferenceType — Invalid PermittedDeviation — Busy DeviatedValue Error ErrorID	MC_AxesObserve_instance (Master :=「パラメータ」, Slave :=「パラメータ」, Enable :=「パラメータ」, ReferenceType :=「パラメータ」, PermittedDeviation :=「パラメータ」, Enabled =>「パラメータ」, Invalid =>「パラメータ」, Busy =>「パラメータ」, DeviatedValue =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE の間、命令を実行します。
ReferenceType	位置種別選択	_eMC_REREFENCE_TYPE	0: _mcCommand 1: _mcFeedback	0 *1	位置の種別を指定します。 0: 指令位置（直前タスク周期 *2 での計算値） 1: フィードバック位置（同一タスク周期 *2 での取得値）
PermittedDeviation	軸間偏差許容値	LREAL	正数、または「0」	0	許容可能な主軸と従軸の位置偏差の最大値を指定します。 単位は［指令単位］となります。*3

*1. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*2. タスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。
*3. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

出力変数

出力変数	名称	データ型	有効範囲	内容
Enabled	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。

出力変数	名称	データ型	有効範囲	内容
Invalid	軸間偏差オーバー	BOOL	TRUE, FALSE	軸間偏差許容値を超えたときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
DeviatedValue	軸間偏差値	LREAL	負数、正数、「0」	指定された主軸と従軸の差分を出力します。 単位は［指令単位］となります。*1
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*2	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Enabled	「Enable」が TRUE のとき。	• 「Error」が TRUE となったとき。 • 「Enable」が FALSE となったとき。
Invalid	軸間偏差許容量を超えたとき。	• 「Error」が TRUE となったとき。 • 「Enable」が FALSE となったとき。 • 軸間偏差許容量を超えていないとき。
Busy	「Enable」の立ち上がり	• 「Error」が TRUE となったとき。 • 「Enable」が FALSE となったとき。
DeviatedValue *1	「Enable」が TRUE のとき。	
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

*1. 「DeviatedValue」は「Enable」の FALSE となるタイミングで FALSE には戻りません。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Master	主軸	_sAXIS_REF	−	主軸を指定します。*1
Slave	従軸	_sAXIS_REF	−	従軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

使用上の注意

主軸と従軸は同じタスクに割り付けてください。異なるタスクの軸を主軸に指定した場合、「主軸指定不正（エラーコード: 5462 Hex)」異常となります。

参考

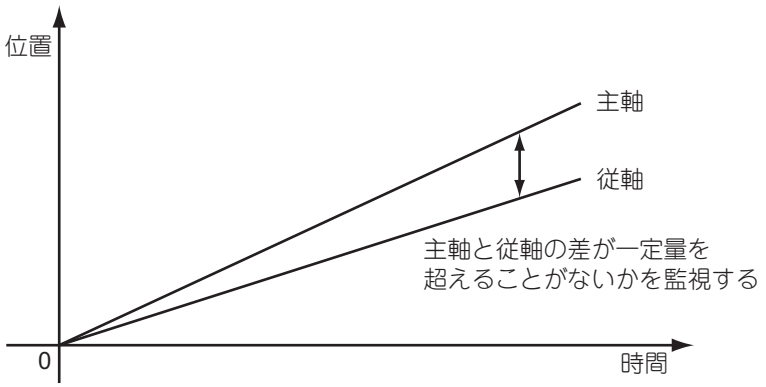
グループに属している軸も設定可能です。

機能説明

- 指定された「Master」（主軸）と「Slave」（従軸）の指令位置、またはフィードバック位置の差が軸間偏差許容値を超えた場合に「Invalid」（軸間偏差オーバー）が TRUE となります。
  つまり、次の条件が成立したときに、「Invalid」（軸間偏差オーバー）が TRUE となります。

  | DeviatedValue（軸間偏差値）| > PermittedDeviation（軸間偏差許容値）が成立するとき。

- 本命令は、軸の動作には、影響を与えません。
- 出力変数の「Invalid」（軸間偏差オーバー）の状態をもとに軸停止処理等の処理をプログラミングしてください。



使用上の注意

- 「Master」（主軸）と「Slave」（従軸）のカウントモードは同じモードにしてください。
  異なるモードの軸を指定した場合は、［リニアモード］として比較されます。
  また、［ロータリモード］同士であってもリングカウンタの設定範囲が異なる場合も、［リニアモード］として比較されます。
- 「PermittedDeviation」（軸間偏差許容値）が割り切れない除算結果などを含んでいると、誤差のため、処理結果が期待どおりにならないことがあります。
- NX シリーズ CPU ユニットでは、「Master」（主軸）と「Slave」（従軸）は、同じタスクに割り付けてください。異なるタスクに割り付けた場合、「Slave」（従軸）側で「主軸指定不正（エラーコード：5462 Hex）」が発生します。
- NX シリーズ パルス出力ユニットでは、指令位置=フィードバック現在位置となっているため、本命令で監視することの実効性はありません。

命令の詳細

以下に本命令の詳細について説明します。

● ReferenceType（位置種別選択）

従軸が同期する主軸の位置の種類を、以下から選択できます。

- _mcCommand: 指令位置（直前タスク周期での計算値）
  現周期に対して、直前のタスク周期で計算した主軸の指令位置を使用します。
  従軸の指令位置を計算する直前の定周期タスクで、計算した主軸の指令位置を使用します。
- _mcFeedback: 同一タスク周期での取得値
  同一タスク周期で取得した主軸のフィードバック位置を使用します。

使用上の注意

ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。同様に定周期タスクは、プライマリ定周期タスクまたは定周期タスク（実行優先度 5）を表します。

● 軸種別と位置種別の関係

監視可能な軸種別と監視する位置種別の関係は以下のようになります。

軸種別	ReferenceType	
	_mcCommand	_mcFeedback
サーボ軸	○	○
エンコーダ軸	× *1	○
仮想サーボ軸	○	○
仮想エンコーダ軸	× *1	○

*1. 命令起動時に、「位置種別選択範囲外（エラーコード: 5430 Hex）」が発生します。

●「DeviatedValue」（軸間偏差値）の計算方法

「DeviatedValue」（軸間偏差値）は以下のように計算します。

リニアモードの場合

「ReferenceType」（位置種別選択）が_mcCommand のとき

「DeviatedValue」（軸間偏差値）＝「Master」（主軸）の指令現在位置 − 「Slave」（従軸）の指令現在位置

「ReferenceType」（位置種別選択）が_mcFeedback のとき

「DeviatedValue」（軸間偏差値）＝「Master」（主軸）のフィードバック現在位置 − 「Slave」（従軸）のフィードバック現在位置

ロータリモードの場合

「ReferenceType」（位置種別選択）が_mcCommand のとき

「DeviatedValue」（軸間偏差値）はリングカウンタの範囲上で、「Master」（主軸）の指令現在位置と「Slave」（従軸）の指令現在位置の短い方の距離となります。
このとき、「DeviatedValue」（軸間偏差値）の符号は、短い方の距離から見て、以下のようになります。

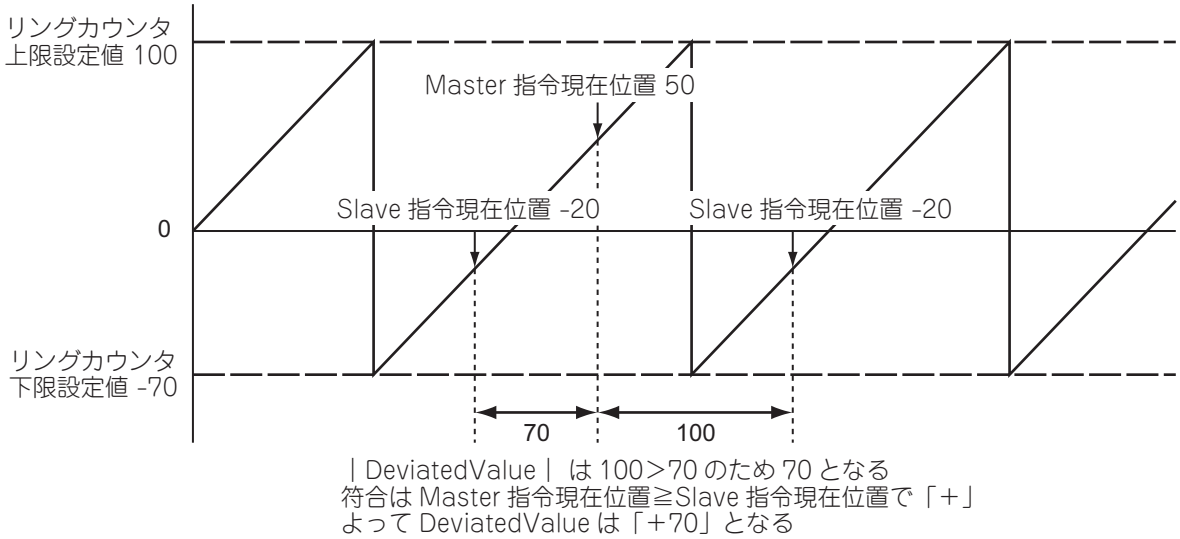
- 「Master」（主軸）の指令現在位置 ≧ 「Slave」（従軸）の指令現在位置なら「＋」
- 「Master」（主軸）の指令現在位置 ＜ 「Slave」（従軸）の指令現在位置なら「−」

「ReferenceType」（位置種別選択）が_mcFeedback のとき

「DeviatedValue」（軸間偏差値）はリングカウンタの範囲上で、「Master」（主軸）のフィードバック現在位置と「Slave」（従軸）のフィードバック現在位置の短い方の距離となります。
このとき、「DeviatedValue」（軸間偏差値）の符号は、短い方の距離から見て、以下のようになります。

- 「Master」（主軸）のフィードバック現在位置 ≧ 「Slave」（従軸）のフィードバック現在位置なら「＋」
- 「Master」（主軸）のフィードバック現在位置 ＜ 「Slave」（従軸）のフィードバック現在位置なら「−」

［ロータリモード］で、「ReferenceType」（位置種別選択）が_mcCommand の場合の「DeviatedValue」（軸間偏差値）の算出例

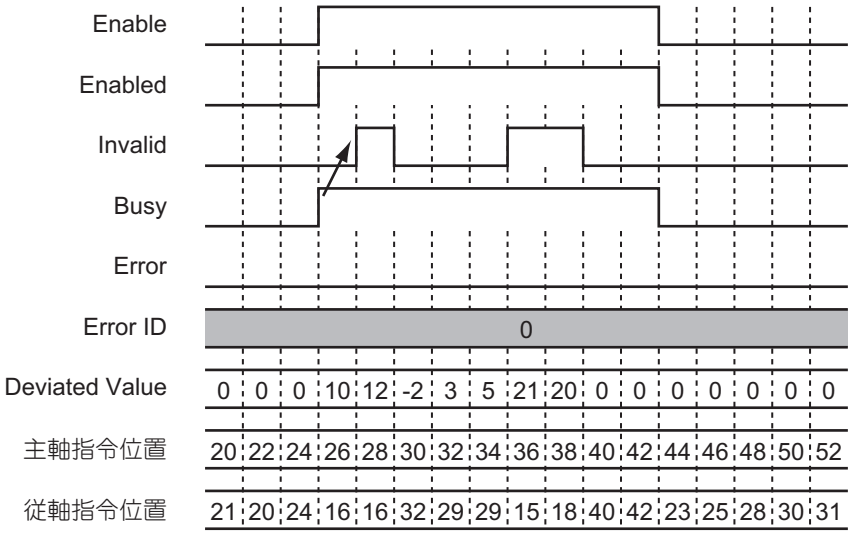


｜DeviatedValue｜ は 100>70 のため 70 となる
符合は Master 指令現在位置≧Slave 指令現在位置で「＋」
よって DeviatedValue は「＋70」となる

［ロータリモード］で、「ReferenceType」（位置種別選択）が_mcFeedback の場合は、上記図の“指令現在位置”が“フィードバック現在位置”になります。

タイミングチャート

本命令の異常は、軸及びグループの動作に影響を与えません。
「PermittedDeviation」（軸間偏差許容値）を「10.0」とした場合の、タイミングチャートを以下に示します。



モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

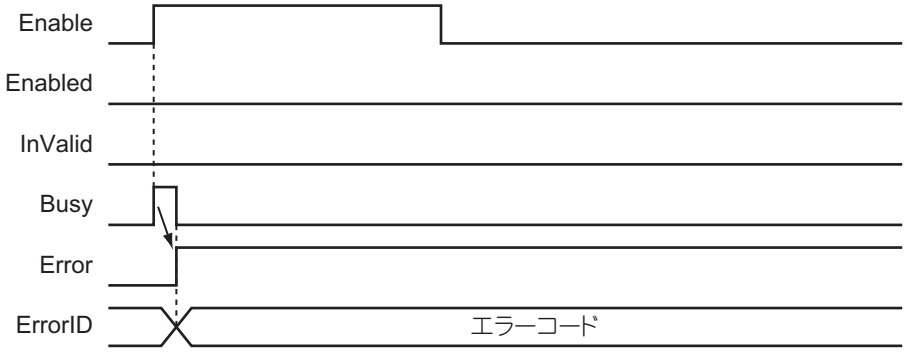
モーション命令多重起動

- 多重起動の制約はありません。

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

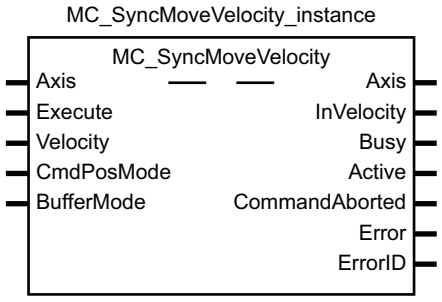


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_SyncMoveVelocity

目標速度に設定した値を、タスク周期ごとにサイクリック同期速度モード(CSV)でサーボドライバへ出力します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_SyncMoveVelocity	サイクリック同期速度制御	FB		MC_SyncMoveVelocity_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, Velocity :=「パラメータ」, CmdPosMode :=「パラメータ」, BufferMode :=「パラメータ」, InVelocity =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

使用上の注意

- NX シリーズ パルス出力ユニットに対しては、本命令を使用できません。
- ここでのタスク周期とは、タスクがプライマリ定周期タスクの場合は、プライマリ周期となり、定周期タスク（実行優先度 5）の場合は、定周期タスク（実行優先度 5）のタスク周期となります。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Velocity	目標速度	LREAL	負数、正数、「0」	0	目標速度を指定します。 0: 速度指令値を「0」にします 正数: 正方向に移動します 負数: 負方向に移動します 単位は、［指令単位／s］となります。*1
CmdPosMode	指令現在位置カウント選択	_eMC_CMDPOS_MODE	0: _mcCount	0 *2	0: フィードバック現在位置を使用し、指令現在位置を更新します。 原点確定状態が継続します。
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered	0 *2	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
InVelocity	目標速度到達	BOOL	TRUE, FALSE	目標速度に到達したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
InVelocity	目標速度に到達したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Busy	「Execute」の立ち上がり	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令により、本命令が中止されたとき。 • ほかの命令で異常が発生して、本命令が中止されたとき。 • 軸異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

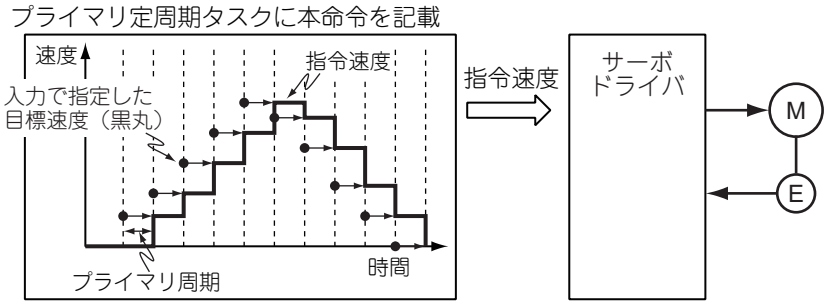
入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

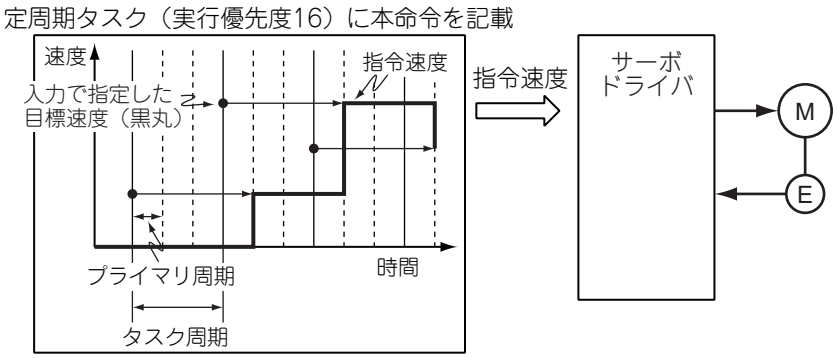
1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 本命令は、ユーザプログラムから与える目標速度を、タスク周期ごとにサイクリック同期速度モード(CSV)でサーボドライバに出力します。
- 「Execute」（起動）の立ち上りで、サーボドライバの制御モードを切り替えるとともに、指令速度を出力します。
- プライマリ定周期タスク、または定周期タスク（実行優先度 5）に本命令を記述した場合、次のタスク周期で目標速度に到達します。
  プライマリ定周期タスクに本命令を記載した例を以下に示します。定周期タスク（実行優先度 5）の場合も同様となります。



- 定周期タスク（実行優先度 16）に本命令を記述した場合、次の定周期タスクで目標速度に到達します。



使用上の注意

本命令を同期制御の主軸として使用する場合の注意事項については、 「1-1-3 同期制御の主軸および補助軸の注意事項 (P.1-6)」を参照してください。

参考

MC_SyncMoveVelocity（サイクリック同期速度制御）命令を使用した場合、MC_SetOverride（オーバーライド値設定）命令は機能しません。

● データオブジェクトのマッピング

MC_SyncMoveVelocity（サイクリック同期速度制御）命令を使用する場合、Sysmac Studio の軸基本設定の［詳細設定］で、以下のオブジェクトデータをマッピングしてください。

- 目標速度(60FFHex)
- オペレーションモード(6060Hex)
- オペレーションモード表示(6061Hex)

これらを設定していない場合、プロセスデータオブジェクト設定不足（エラーコード: 3461Hex）が発生します。

データオブジェクトのマッピングについては、「2-3 PDO マッピング (P.2-32)」、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

命令の詳細

以下に本命令の詳細について説明します。

● Velocity（目標速度）

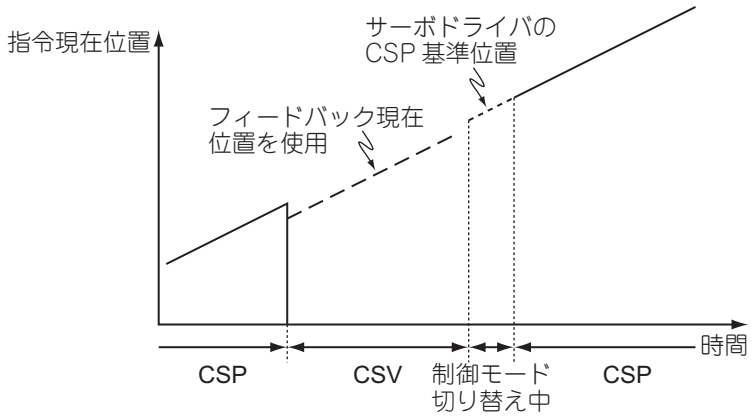
入力変数の「Velocity」（目標速度）は「0」を基準にした LREAL 型で設定することができます。
正数を設定すると正方向に移動し、負数を設定すると負方向に移動します。
「0」を設定すると、指令速度「0」となりますが、「Status.Continuous」（連続動作中）は TRUE のままです。
毎周期、ユーザプログラムから「Velocity」（目標速度）を設定することができます。
前周期と異なる目標速度が指定された場合、新しい目標速度が適用されます。前周期と同じ値の場合、前周期の目標速度で動作します。

使用上の注意

- 目標速度を設定する場合には、速度変化によって装置のメカ機構に過度な負担がかからない値を指定してください。
- 本命令を使用する軸を同期制御の主軸として使用する場合、主軸への目標速度の設定によっては、従軸が急激な動作をする場合がありますので、注意してください。
- 制御モード切り替え時に、指令現在位置が急に変化する場合がありますので、注意してください。

● 指令現在位置

入力変数の「CmdPosMode」（指令現在位置カウント選択）で_mcCount を選択すると、本命令起動時に指令現在位置は 1 周期前のフィードバック現在位置となり、本命令が終了するまでフィードバック現在位置を使用します。
オムロン製サーボドライバ 1S シリーズおよび G5 シリーズでは CSP への切り替え処理中、あらかじめマッピングしておいた「CSP 基準位置」が PDO 送信されます。CSV から CSP への切り替え処理が完了するまで、この基準位置を指令現在位置として用います。CSP への切り替えが完了したら、指令現在位置＝指令位置となります。



● オムロン製サーボドライバ 1S シリーズを使用する場合

制御モード切り替え途中のサーボドライバの「CSP 基準位置」を使用する場合には、プロセスデータに CSP 用参照位置(3010Hex-87Hex)をマッピングしてください。
Sysmac Studio の PDO 編集画面で CSP 用参照位置(3010Hex-87Hex)をプロセスデータにマッピングします。次に軸基本設定の［詳細設定］で MC 機能モジュールの「CSP 切り替え用基準位置」と CSP 用参照位置(3010Hex-87Hex)をマッピングしてください。

● オムロン製サーボドライバ G5 シリーズを使用する場合

制御モード切り替え途中のサーボドライバの「CSP 基準位置」を使用する場合には、プロセスデータに CSP 用参照位置(4020 Hex)をマッピングしてください。
Sysmac Studio の PDO 編集画面で CSP 用参照位置(4020 Hex)をプロセスデータにマッピングします。次に軸基本設定の［詳細設定］で MC 機能モジュールの「CSP 切り替え用基準位置」と CSP 用参照位置(4020 Hex)をマッピングしてください。

使用上の注意

PDO に「CSP 用参照位置(4020 Hex)」をマッピングした場合、プライマリ周期、または定周期タスク（実行優先度 5）のタスク周期を 1ms 以上に設定してください。
プライマリ周期、またはタスク周期を 1ms より小さい値に設定した場合、サーボドライバ G5 シリーズ側で異常が発生します。
詳細については、『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-365)』を参照してください。

参考

オムロン製サーボドライバ G5 シリーズのうち、リニアモータタイプは CSP 用参照位置(4020 Hex)に対応していません。

● オムロン製サーボドライバ 1S シリーズまたは G5 シリーズ以外のドライバを使用する場合

本命令を起動する場合や、本命令実行中に CSP を使用する命令を多重起動した場合、サーボドライバによっては制御モード切り替え処理時に異常が発生することがあります。
その場合は、軸を停止させた状態（速度「0」）で、本命令を起動したり、CSP を使用する命令を多重起動させたりしてください。

● NX シリーズ 位置インタフェースユニットを使用する場合

NX シリーズ 位置インタフェースユニットに対しては、本命令を使用できません。

● 停止処理

軸動作を停止させる場合の制御モードや指令速度について説明します。
減速停止を行う場合、本命令の目標速度を初速度として各停止方法が持つ減速度を適用して減速停止します。

MC_ImmediateStop（即停止）命令による停止
指令速度を「0」にします。下記の切り替え判定式が成立すれば、CSP に切り替えます。

MC_Stop（強制停止）命令による停止
命令の減速度で指令速度を「0」にします。下記の切り替え判定式が成立すれば、CSP に切り替えます。

軽度フォールトレベルの異常発生による停止

各異常が定める減速度で指令速度を「0」にします。下記の切り替え判定式が成立すれば、CSP に切り替えます。

全停止フォールトレベル、および部分停止フォールトレベルの異常発生による停止
指令速度を「0」にします。下記の切り替え判定式が成立すれば、CSP に切り替えます。
ただし、異常のレベルによっては制御モードの切り替えを正常に実行できず CSV で停止する場合があります。

サーボ OFF による停止
所定の方法により、指令速度を「0」にします。制御モードの切り替え処理は行いません。

CPU ユニットの動作モードをプログラムモードに切り替えた場合の停止
所定の方法により、指令速度を「0」にします。下記の切り替え判定式が成立すれば、CSP に切り替えます。

● 切り替え判定式

制御モードの切り替えの可否はサーボドライバの仕様に依存します。
停止命令の起動や、異常発生による停止処理の際に制御モード切り替えを確実に行うため、サーボモータが十分に減速したことを確認してから CSP への切り替えを行います。
指令速度が「0」になった後、定周期タスクで 3 回連続してフィードバック現在速度が以下の判定式を満足した場合に、CSP への切り替えを行います。

フィードバック現在速度 ≦ （最高速度）× 0.1

使用上の注意

ここでの定周期タスクは、プライマリ定周期タスクまたは定周期タスク（実行優先度 5）を表します。

● サイクリック同期位置モードへの復旧

停止によっては CSP への切り替えが正常に実行できない場合があります。
たとえば、モーション制御機能モジュールで部分停止フォールトレベルの異常が発生した場合などがそれにあたります。
そのため、MC_Power（運転可）命令の出力変数「Status」（運転可）が FALSE になった場合、CSP に切り替える処理を行います。

● 制御モード切り替え失敗時の動作

サーボドライバに対して制御モード切り替え指令を行ってから 1 秒経過してもサーボドライバの切り替えが完了しない場合は、「サーボドライバの制御モード切替異常（エラーコード: 7439 Hex）」となります。
また、サーボドライバに対して制御モード切り替え指令を行って、指令速度が「0」になってから 10 秒経過しても切り替え判定式が成立しない場合も上記と同様に「サーボドライバの制御モード切替異常（エラーコード: 7439 Hex）」となります。
「サーボドライバの制御モード切替異常（エラーコード: 7439 Hex）」が起こると、指令速度を「0」にし、サーボ OFF します（フリーラン停止）。

「サーボドライバの制御モード切替異常（エラーコード: 7439 Hex）」については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 使用可能軸と起動条件

- サーボ軸で本命令を使用できます。

使用する場合は、MC_Power（運転可）命令の「Enable」（有効）を TRUE（サーボ ON 状態）にしてください。
- 仮想サーボ軸は、常時受け付け可能です。
  ただし、サーボドライバの制御モード切り替え処理は行いません。
- エンコーダ軸、仮想エンコーダ軸では、起動時に異常となります。

● 軸変数のステータス

軸変数のステータスは、「Status.Continuous」（連続動作中）が TRUE となります。
また、サーボドライバに設定されている制御モードを確認したい場合、軸変数の「DrvStatus」（サーボドライバ状態）を確認してください。下記にサーボドライバ状態を示します。

変数名	データ型	名称	説明
CSP	BOOL	サイクリック同期位置(CSP)モード中	サーボ ON で、CSP モードのときに TRUE となります。
CSV	BOOL	サイクリック同期速度(CSV)モード中	サーボ ON で、CSV モードのときに TRUE となります。
CST	BOOL	サイクリック同期トルク(CST)モード中	サーボ ON で、CST モードのときに TRUE となります。

● 原点状態

「CmdPosMode」（指令現在位置カウント選択）が_mcCount のときは、原点確定状態は継続しますし、変化しません。

● オーバーライド

本命令に対して、オーバーライドは無効となります。

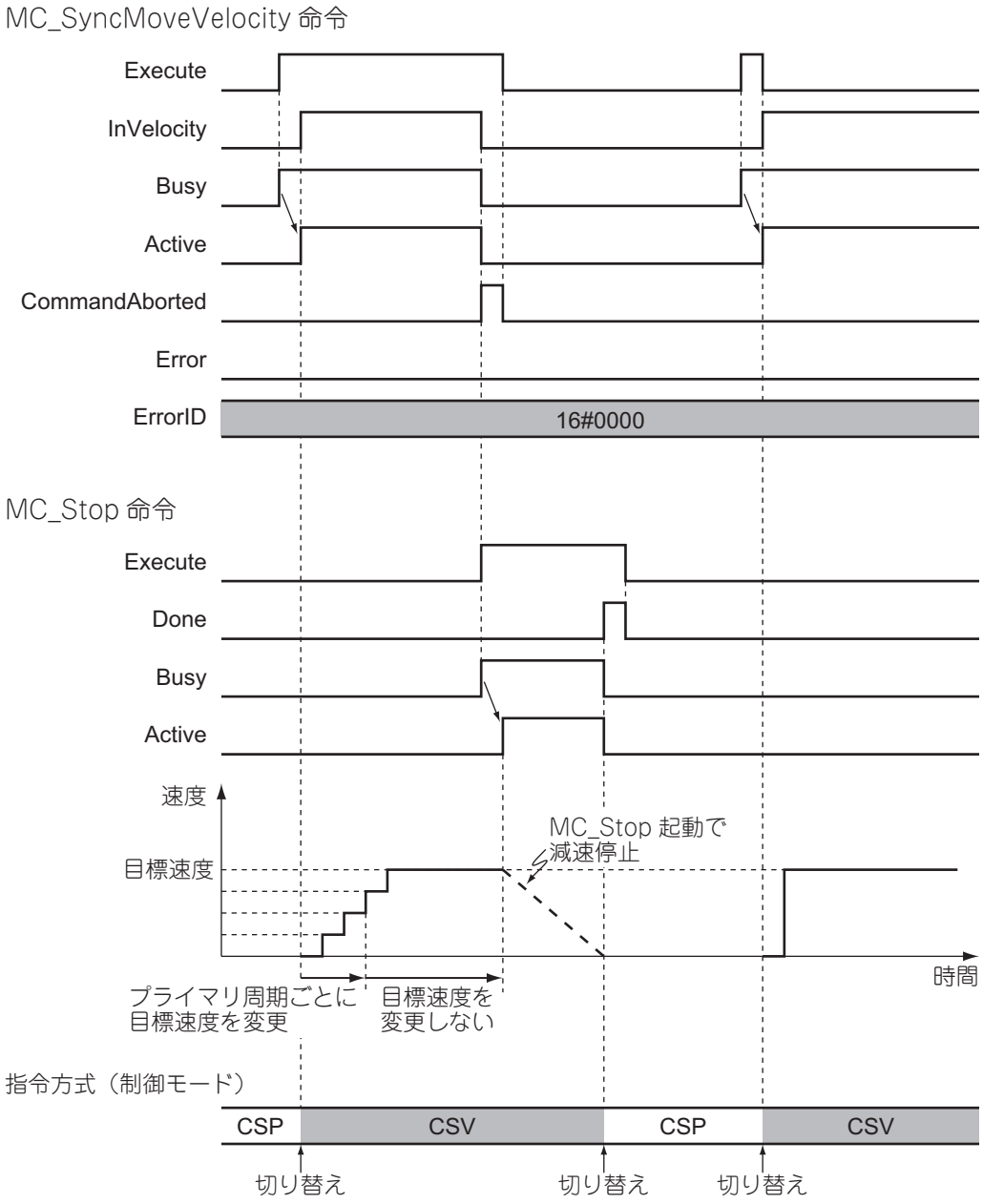
● ソフトウェアリミット

本命令に対して、ソフトウェアリミットは有効となります。
軸パラメータで、以下の設定を選択した場合でも、有効となります。
- 指令位置に対して有効。減速停止させる。
- 指令位置に対して有効。たまりパルス停止させる。

タイミングチャート

- 「Execute」（起動）の立ち上りと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- 指令速度が「Velocity」（目標速度）に到達すると、「InVelocity」（目標速度到達）が TRUE となります。
- ほかの命令により、本命令が中止された場合は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）、「InVelocity」（目標速度到達）が FALSE となります。
- 本命令を終了させるには、MC_Stop（強制停止）命令を起動します。

プライマリ定周期タスクに本命令を記載した場合を下記に示します。



参考

MC 機能モジュールはサードドライバに対して上記タイミングチャートで制御モードを指令します。しかし、サーボドライバに反映されるタイミングは、各サーボドライバの仕様により異なります。

モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex)」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

MC_MoveVelocity（速度制御）命令と同じく、「BufferMode」（バッファモード選択）の［アボーティング］と［バッファード］に対応します。
また、制御モードの切り替えは、多重起動された命令の処理開始時に行います。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

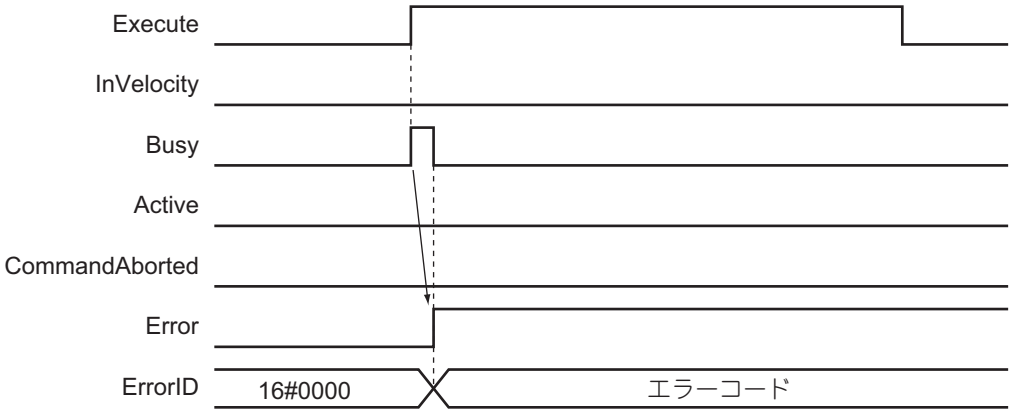
● 本命令実行中の他命令起動

MC_MoveVelocity（速度制御）命令と同じく、「BufferMode」（バッファモード選択）の［アボーティング］と［バッファード］に対応します。
［バッファード］は「InVelocity」（目標速度到達）が TRUE になったとき、他命令を起動します。
また、制御モードの切り替えは、多重起動された命令の処理開始時に行います。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_SyncMoveAbsolute

軸に指定した目標位置を周期的に出力します。

命令	名称	FB / FUN	グラフィック表現	ST 表現
MC_SyncMoveAbsolute	サイクリック同期絶対位置決め	FB	MC_SyncMoveAbsolute_instance MC_SyncMoveAbsolute Axis — — Axis Execute — InPosition Position — Busy Direction — Active BufferMode — CommandAborted — Error — ErrorID	MC_SyncMoveAbsolute_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, Position :=「パラメータ」, Direction :=「パラメータ」, BufferMode :=「パラメータ」, InPosition =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

バージョン関連情報

本命令は、ユニットバージョン Ver.1.03 以降の CPU ユニットと Sysmac Studio Ver.1.04 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Position	目標位置	LREAL	負数、正数、「0」	0	絶対座標の目標位置を指定します。単位は、［指令単位］となります。*1
Direction	方向選択	_eMC_DIRECTION	0: _mcPositiveDirection 1: _mcShortestWay 2: _mcNegativeDirection 3: _mcCurrentDirection 4: _mcNoDirection	1 *2	カウントモードが［ロータリモード］の場合に、回転方向を指定します。*3 0: 正方向指定 1: 近回り指定 2: 負方向指定 3: 現在方向指定 4: 方向指定なし

入力変数	名称	データ型	有効範囲	初期値	内容
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *2	モーション命令多重起動時の動作を指定します。 0: アボーティング

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*3. 回転方向の指定内容については、「Direction（方向選択）（P.3-50)」を参照してください。

出力変数

出力変数	名称	データ型	有効範囲	内容
InPosition	インポジション	BOOL	TRUE, FALSE	フィードバック現在位置が目標位置のインポジション幅内に入ったとき TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
InPosition	フィードバック現在位置が目標位置のインポジション幅内に入ったとき。	• フィードバック現在位置がインポジション幅外へ出たとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Busy	「Execute」の立ち上がり	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動で、本命令が中止されたとき。 • ほかの命令で異常が発生して、本命令が中止されたとき。 • 軸異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

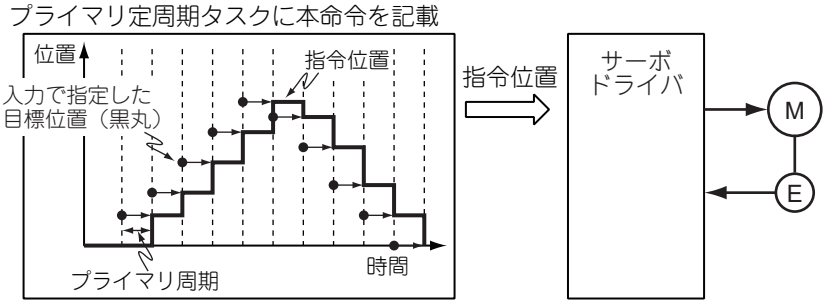
入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	―	軸を指定します。*1

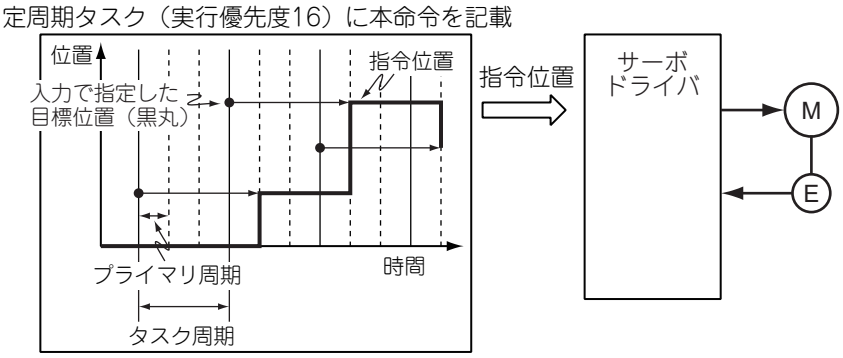
1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 本命令は、ユーザプログラムから与える目標位置を、タスク周期ごとにサイクリック同期位置モード(CSP)でサーボドライバなどに出力します。目標位置は絶対位置で与えます。
- 速度は軸パラメータの［最高速度］を上限とします。［最大加速度］と［最大減速度］は適用されません。
- プライマリ定周期タスク、または定周期タスク（実行優先度 5）に本命令を記述した場合、入力で指定した目標位置は次のタスク周期でサーボドライバに出力されます。
プライマリ定周期タスクに本命令を記載した例を以下に示します。定周期タスク（実行優先度 5）の場合も同様となります。



- 定周期タスク（実行優先度 16）に本命令を記述した場合、入力で指定した目標位置は次の定周期タスクでサーボドライバに出力されます。



使用上の注意

指示した目標位置による移動量が軸パラメータの［最高速度］を超えないように目標位置を指令してください。［最高速度］を超える目標位置を指示したときは、指令速度飽和となり、［最高速度］で制限された移動量を出力します。指示した目標位置から不足する分の移動量指示は次の周期以降に出力します。
なお、このとき軸制御ステータスの「Details.VelLimit」（指令速度飽和）が TRUE となります。

命令の詳細

以下に本命令の詳細について説明します。

● インポジションチェック

「Position」（目標位置）を更新しないときに目標位置とフィードバック位置の幅が軸パラメータの［インポジション幅］の範囲内になったとき、「InPosition」（インポジション）が TRUE となります。
「InPosition」（インポジション）が TRUE のときに目標位置を変更しても、その周期の中では TRUE のままで、次の周期で FALSE となります。
軸パラメータの［インポジションチェック時間］設定は無効です。

● 停止処理

軸動作を停止させるときの動作について説明します。
動作を停止させるには、MC_Stop（強制停止）命令、または MC_ImmediateStop（即停止）命令を使用します。これらの命令を起動すると、本命令の「CommandAborted」（実行中断）が TRUE となります。

MC_Stop（強制停止）命令による停止
即停止します。

MC_ImmediateStop（即停止）命令による停止
軸パラメータ設定［即停止入力停止方法］の指定に従い即停止します。

● 異常発生による停止

軸を停止させる異常が発生したとき、設定によらず「即停止」します。

● 使用可能軸

- サーボ軸で本命令を使用できます。
  使用する場合は、MC_Power（運転可）命令の「Enable」（有効）を TRUE（サーボ ON 状態）にしてください。
- 仮想サーボ軸は、常時受け付け可能です。
- エンコーダ軸、仮想エンコーダ軸では、起動時に異常となります。

● 軸変数のステータス

軸変数の軸ステータスは、「Status.Discrete」（位置決め動作中）が TRUE となります。
軸制御ステータスへは影響しません。

● オーバーライド

本命令に対して、オーバーライドは無効となります。

● 現在位置変更

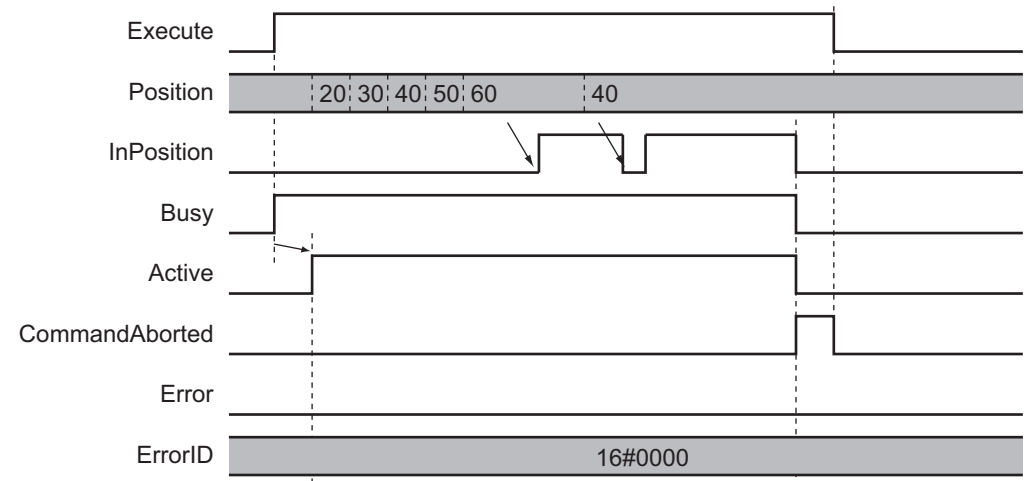
MC_SetPosition（現在位置変更）命令により現在位置を変更したときは、変更された現在位置から「Position」（目標位置）で指定した目標位置への移動量を出力します。
現在位置変更前に「InPosition」（インポジション）が TRUE のときは、現在位置変更後の次の周期で「InPosition」（インポジション）が FALSE となります。

タイミングチャート

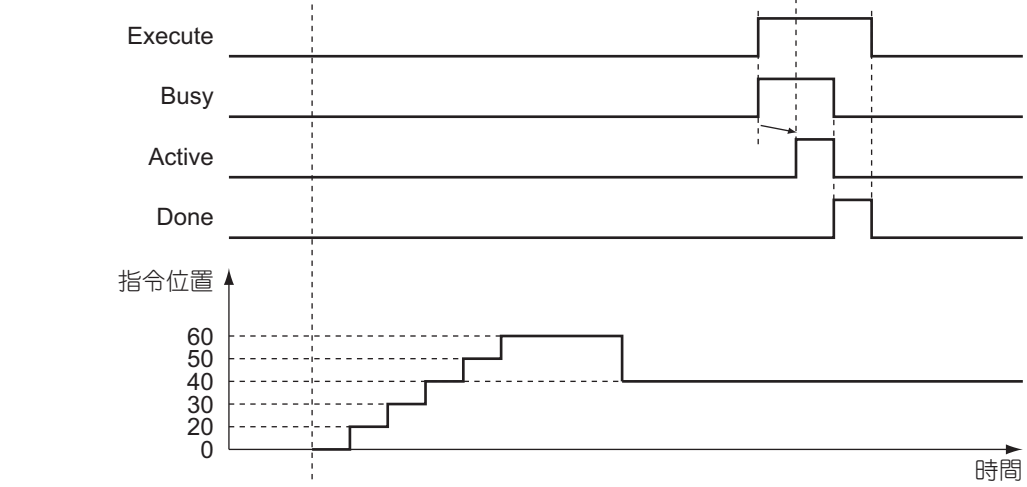
- 「Execute」（起動）の立ち上りと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- フィードバック現在位置が「Position」（目標位置）のインポジション幅内に入ったとき、「InPosition」（インポジション）が TRUE となります。
- ほかの命令により、本命令が中止された場合は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）、「InPosition」（インポジション）が FALSE となります。
- 本命令を終了させるには、MC_Stop（強制停止）命令を起動します。

プライマリ定周期タスクに本命令を記載した場合を下記に示します。

MC_SyncMoveAbsolute 命令



MC_Stop 命令



参考

MC 機能モジュールはサードドライバに対して上記タイミングチャートで制御モードを指令します。しかし、サーボドライバに反映されるタイミングは、各サーボドライバの仕様により異なります。

モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

現在実行中の命令に対して、本命令を起動することにより、本命令への切り替えや本命令のバッファリングを行うことができます。
各軸に 1 つまでバッファリングすることが可能です。
多重起動時の本命令の動作を「BufferMode」（バッファモード選択）で指定します。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

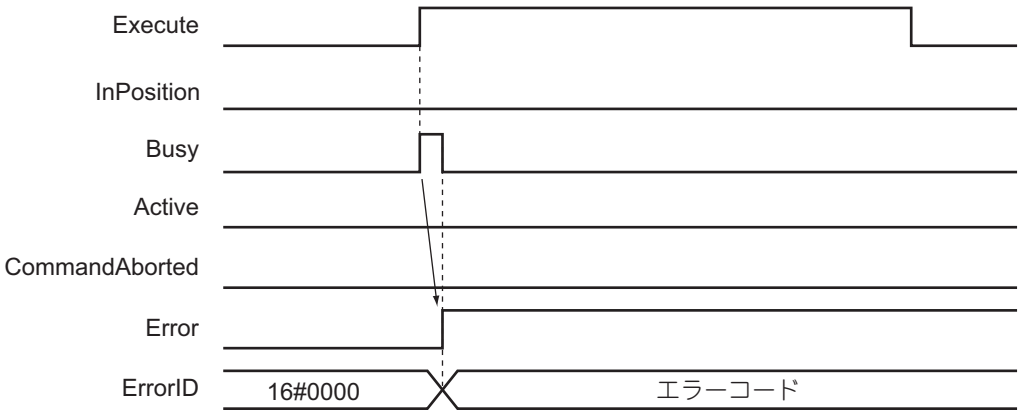
● 本命令実行中の他命令起動

本命令に対するモーション命令多重起動を行う場合、［アボーティング］が可能です。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_Reset

軸の異常を解除します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_Reset	軸エラーリセット	FB	MC_Reset_instance MC_Reset Axis — Axis Execute — Done Busy Failure Error ErrorID	MC_Reset_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Failure =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Failure	不正終了	BOOL	TRUE, FALSE	命令が正常に実行されなかったときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	異常解除処理が正常終了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「Failure」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
Failure	• 異常による軸の減速停止中に異常解除を実行したとき。 • 軸共通異常発生による軸異常発生中に異常解除を実行したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 「Execute」（起動）の立ち上がりで、「Axis」（軸）で指定した軸の異常解除処理を開始します。
異常解除処理とは、軸の異常解除、およびドライバ側で異常が発生している場合はドライバエラーリセットを行います。
また、軸の運転状態にかかわらず、異常解除処理を実行します。
- 本命令はすべての軸種別に対して使用可能です。
- 異常が発生している軸にだけ異常解除処理を実行します。
- ドライバエラーが発生している軸には、ドライバエラーリセット処理を実行したのち、異常解除処理を実行します。
ドライバエラーリセット処理はドライバエラーがクリアされるか、軸パラメータの［ドライバエラーリセット監視時間］の間、継続します。
- エラー減速停止中による軸減速停止中に本命令を実行した場合は、実行されず「Failure」（不正終了）が TRUE となります。これは、軸が停止する前に、異常解除ができないようにするためです。
また、MC 共通異常発生による軸異常も本命令では解除できないため「Failure」（不正終了）が TRUE となります。
MC 共通異常には「MC 共通部分停止フォールト」および「MC 共通軽度フォールト」があります。
- 解除される異常は「Execute」（起動）の立ち上がりのタイミングで発生している異常を対象とします。解除中に発生した異常については、異常解除は行われません。

使用上の注意

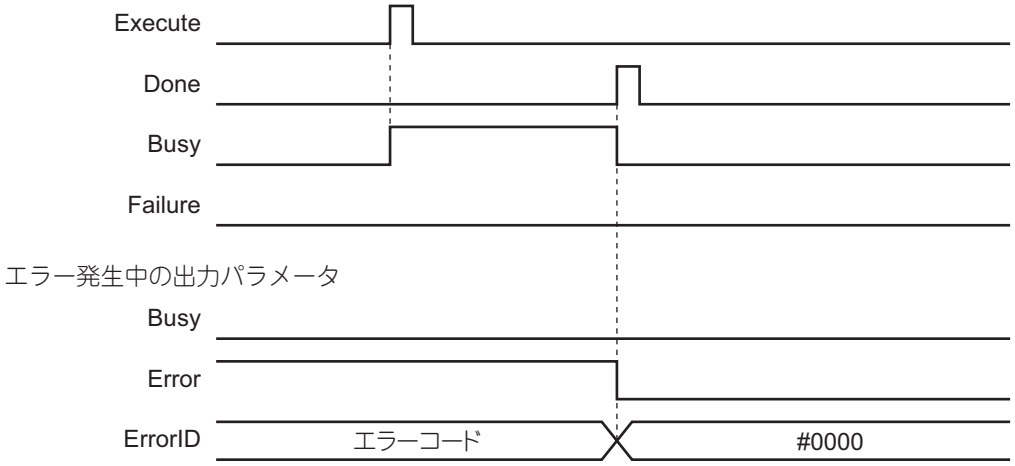
- 本命令による異常解除の処理は、複数の制御周期にまたがる場合があります。
- 「MC 共通 部分停止フォールト」、および「MC 共通 軽度フォールト」の共通異常が発生している場合や、軸が動作中の場合、本命令の「Failure」（不正終了）が TRUE となります。異常要因を取り除いた上で、「Done」（完了）になるまでリトライ処理を行ってください。
- 異常の発生要因を取り除く場合は、軸が完全に停止したことを確認してから行ってください。
軸が完全に停止したことを確認するには、軸変数の「Act.Vel」（フィードバック現在速度）が「0」になったことを確認する必要があります。
- オムロン製サーボドライバ G5 シリーズに対し本命令を使用する場合、ResetECError（EtherCAT 異常解除）命令が同時に実行されないよう排他処理を行ってください。
- NX シリーズ パルス出力ユニットに対し本命令を使用する場合、パルス出力ユニットに接続したドライバの異常状態は解除されません。詳細は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。

参考

本命令では以下の異常は解除できません。
- 軸共通異常 全般: ResetMcError（全エラーリセット）命令を実行してください。
- 軸グループ異常 全般: MC_GroupReset（軸グループエラーリセット）命令を実行してください。

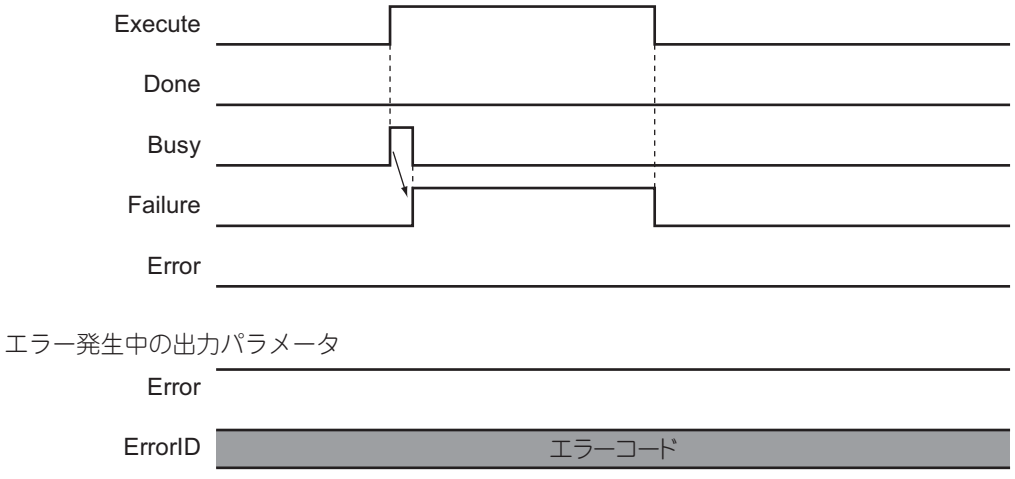
また、本命令を実行してもスレーブ通信異常などのネットワークに関する異常の要因解除は実施できません。ResetECError（EtherCAT 異常解除）命令を実行してください。

タイミングチャート



命令の中止

異常による軸の減速停止中、もしくは軸共通異常発生による軸異常発生中の異常が解除できない場合は、命令を中断します。



エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_ChangeAxisUse

軸パラメータの［軸使用］を一時的に切り替えます。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_ChangeAxisUse	軸使用変更	FB	MC_ChangeAxisUse_instance MC_ChangeAxisUse Axis — — Axis Execute — Done AxisUse — Busy CommandAborted Error ErrorID	MC_ChangeAxisUse_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, AxisUse :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

使用上の注意

本命令で切り替えた値は CPU ユニットの不揮発性メモリに保存されません。そのため、コントローラの電源を OFF、およびダウンロード、MC 機能モジュールの再起動処理が開始されると、切り替えた値は消失し、Sysmac Studio で設定した値に戻ります。
不揮発性メモリに保存する場合は、Sysmac Studio を使って、パラメータの転送を行ってください。

参考

- 「ダウンロード」は、Sysmac Studio の「同期」機能を使用します。
- 本命令を使用するアプリケーション例は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編(SBCA-467)』を参照してください。

バージョン関連情報

本命令は、ユニットバージョン Ver.1.04 以降の CPU ユニットと Sysmac Studio Ver.1.05 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
AxisUse	軸使用	_eMC_AXIS_USE	1: _mcUnusedAxis 2: _mcUsedAxis	1 *1	使用軸または未使用軸を指定します。 1: 未使用軸 2: 使用軸

*1. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	•「Execute」が TRUE の場合、「Execute」の FALSE と同時 •「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	•「Done」が TRUE となったとき。 •「Error」が TRUE となったとき。 •「CommandAborted」が TRUE となったとき。
CommandAborted	ほかの命令で異常が発生し、本命令が中止されたとき。	•「Execute」が TRUE の場合、「Execute」の FALSE と同時 •「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 「Execute」（起動）の立ち上がりで、「AxisUse」（軸使用）で指定した軸使用に切り替えます。
- 命令の実行完了と同時に、軸変数の_MC_AX[*].Cfg.AxEnable（軸使用）が指定した軸使用に切り替わります。
- 切り替えできる軸は、軸パラメータの［軸使用］が［未使用軸（使用軸に切替可）］または［使用軸］に設定されている軸です。［未使用軸（使用軸に切替不可）］に設定されている軸を［使用軸］に切り替えないでください。
- 原点確定と原点未確定のどちらの状態でも、命令を実行できます。原点確定のときに［未使用軸］に切り替えると、原点未確定となります。

- 絶対値エンコーダを使用している軸を、［使用軸］から［未使用軸］に切り替えた場合、CPU ユニットの電源 OFF 時に CPU ユニット内部に保存される［絶対値エンコーダ原点位置オフセット］は、［未使用軸］に切り替える直前の値です。

使用上の注意

- CPU ユニットの形式ごとの制御軸最大数を超える軸番号の軸を、使用軸に切り替えることはできません。また、使用軸に切り替え可能な実軸の数は使用実軸最大数までです。

仕様項目	形 NX701-17□□	形 NX701-16□□
設定可能な軸番号	0～255	0～127
使用実軸最大数	256 軸	128 軸

仕様項目	形 NX502-17□□*1	形 NX502-16□□*1	形 NX502-15□□	形 NX502-14□□	形 NX502-13□□
設定可能な軸番号	0～255	0～127	0～63	0～31	0～15
使用実軸最大数	256 軸	128 軸	64 軸	32 軸	16 軸

*1. ユニットバージョン Ver.1.66 から追加された形式です。

仕様項目	形 NX102-12□□	形 NX102-11□□	形 NX102-10□□	形 NX102-90□□
設定可能な軸番号	0～14	0～14	0～14	0～3
使用実軸最大数	12 軸	8 軸	6 軸	4 軸
使用モーション制御サーボ軸	8 軸	4 軸	2 軸	－
使用単軸位置制御サーボ軸	4 軸	4 軸	4 軸	4 軸

仕様項目	形 NX1P2-11□□	形 NX1P2-10□□	形 NX1P2-90□□	形 NX1P2-9B□□
設定可能な軸番号	0～11	0～9	0～3	0～1
使用実軸最大数	8 軸	6 軸	4 軸	2 軸
使用モーション制御サーボ軸	4 軸	2 軸	－	－
使用単軸位置制御サーボ軸	4 軸	4 軸	4 軸	2 軸

仕様項目	形 NJ501-		
	□5□□	□4□□	□3□□
設定可能な軸番号	0～63	0～31	0～15
使用実軸最大数	64 軸	32 軸	16 軸

仕様項目	形 NJ301-12□□	形 NJ301-11□□	形 NJ101-10□□
設定可能な軸番号	0～14 *1	0～14 *2	0～5
使用実軸最大数	8 軸	4 軸	2 軸

*1. ユニットバージョン Ver.1.05 以前の CPU ユニットの軸番号は、「0 ～ 7」です。
*2. ユニットバージョン Ver.1.05 以前の CPU ユニットは、「0 ～ 3」です。

- 軸変数の_MC_AX[*].Status.Disabled（軸無効）または_MC_AX[*].Status.Standstill（停止中）が TRUE のときだけ命令を実行できます。FALSE のときに実行すると異常となります。
- 軸変数の_MC_AX[*].Details.VelLimit（指令速度飽和）が TRUE のときに実行すると異常となります。
- 軸使用が_mcUnusedAxis（未使用軸）となっている軸に、MC_Write（MC 設定書込）命令で、軸パラメータを書き換えても無効です。軸使用を_mcUsedAxis（使用軸）に切り替えてから MC_Write（MC 設定書込）命令を実行してください。

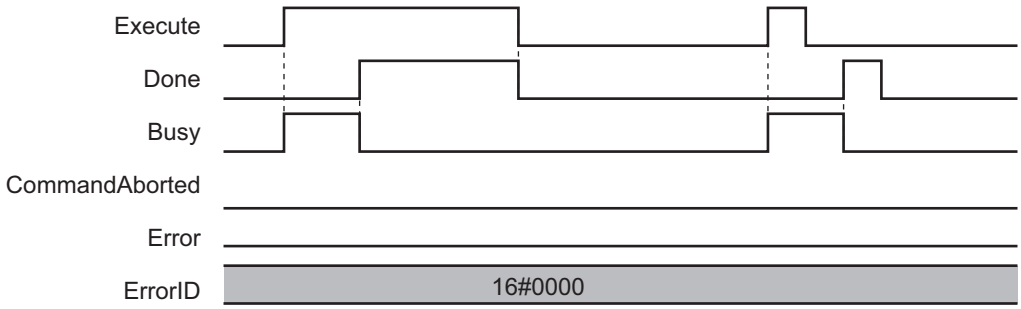
- 本命令で［未使用軸］に切り替えた軸を含む軸グループに、MC_GroupEnable（軸グループ有効）命令を実行すると異常となります。
- 使用軸に対する命令の実行で異常が発生した場合は軸の異常となり、未使用軸に対する命令の実行で異常が発生した場合は MC 共通の異常となります。
- 本命令による軸使用の切り替え後に他のモーション制御命令を実行する場合は、本命令の出力変数「Done」（完了）が TRUE となっていることを確認してから実行してください。

参考

NX シリーズ CPU ユニットの場合、_MC_AX[*]で始まる変数名は、_MC1_AX[*]、_MC2_AX[*]の場合があります。

タイミングチャート

MC_ChangeAxisUse 命令



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex)」となります。

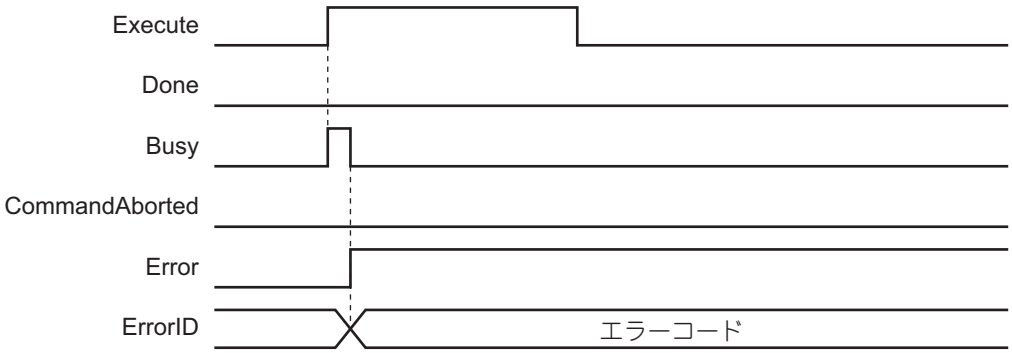
モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となり、軸使用は切り替わりません。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

使用上の注意

使用軸に対する命令の実行で異常が発生した場合は軸の異常となり、未使用軸に対する命令の実行で異常が発生した場合は MC 共通の異常となります。

MC_DigitalCamSwitch

軸の位置に応じてデジタル出力を ON または OFF します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_DigitalCamSwitch	デジタルカムスイッチ有効	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 :=「パラメータ」, Switches :=「パラメータ」, Outputs :=「パラメータ」, TrackOptions :=「パラメータ」, Enable :=「パラメータ」, EnableMask :=「パラメータ」, ValueSource :=「パラメータ」, InOperation =>「パラメータ」, Busy =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

安全上の要点

- MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令は、必ず軸を一定速度で使用してください。
- NX_AryDOutTimeStamp（タイムスタンプデジタル出力配列書込）命令は、MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令の「InOperation」（作動中）がTRUE になっていることを確認してから使用してください。

使用上の注意

- 本命令は、下記のユニットに対して有効です。
  a）NX シリーズ位置インタフェースユニットを割り付けている軸。
  使用可能な NX ユニットは、形 NX-EC0□□□、形 NX-ECS□□□です。
  b）オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプ。
- 本命令は、必ず NX_AryDOutTimeStamp 命令およびタイムスタンプ方式対応のデジタル出力ユニットと組み合わせて使用してください。

バージョン関連情報

本命令は、ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE の間、命令を実行します。FALSE のときは「Outputs」の値を更新しません。*1

入力変数	名称	データ型	有効範囲	初期値	内容
EnableMask	トラック有効	WORD	16#0000～FFFF	16#0000	トラックごとに有効または無効を指定します。最大 16 トラックで、0 ビットめの値がトラック番号 0 の、15 ビットめの値がトラック番号 15 の有効または無効を指定します。 0: 無効 *2 1: 有効
ValueSource (Reserved)	入力情報	_sMC_SOURCE	−	−	(Reserved)

*1. 「Enable」が FALSE のとき、「Outputs」(出力信号) の値は保持されます。また、NX_AryDOutTimeStamp 命令の「Enable」が FALSE のとき、デジタル出力ユニットのデジタル出力は OFF となります。
*2. 「EnableMask」のビットの値が［0］のとき、該当トラック番号の「Outputs」(出力信号) は OFF となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
InOperation	作動中	BOOL	TRUE, FALSE	有効な出力信号の出力中に TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID (エラーコード) となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
InOperation	「Enable」が TRUE となったとき。	• 「Enable」が FALSE となったとき。 • 「Error」が TRUE となったとき。
Busy	「Enable」の立ち上がり	• 「Enable」が FALSE となったとき。 • 「Error」が TRUE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	位置を参照する軸を指定します。*1
Switches	スイッチ	ARRAY[0..255] OF _sCAMSWITCH_REF *2	−	スイッチ構造体_sCAMSWITCH_REF 型の配列変数を、スイッチの ON/OFF パターンデータとして指定します。 配列要素番号は、スイッチ番号を示します。
Outputs	出力信号	ARRAY[0..15] OF _sOUTPUT_REF *2 *3	−	スイッチの ON/OFF パターンデータをもとに算出された、デジタル出力の ON/OFF 時刻の出力先として、出力信号_sOUTPUT_REF 型の配列変数を指定します。配列要素番号は、トラック番号を示します。 この配列変数を NX_AryDOutTimeStamp 命令の入出力変数に指定することにより、実際のデジタル出力を ON/OFF します。

入出力変数	名称	データ型	有効範囲	内容
TrackOptions	トラックオプション	ARRAY[0..15] OF _sTRACK_REF *2 *3	−	トラックオプション構造体_sTRACK_REF 型の配列変数を、スイッチの動作条件として指定します。配列要素番号は、トラック番号を示します。

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. 配列要素の開始番号が「0」以外の場合は異常となります。また、2 次元以上の配列の場合も異常となります。
*3. 「Outputs」の要素数と「TrackOptions」の要素数が一致しない場合は異常となります。

機能説明

- 本命令は、軸のフィードバック位置が「Switches」に設定した位置に到達する時刻を、デジタル出力が ON/OFF するための時刻として出力します。
- 本命令は、必ず NX_AryDOutTimeStamp 命令およびタイムスタンプ方式対応のデジタル出力ユニットと組み合わせて使用してください。
- NX_AryDOutTimeStamp 命令は、指定したタイムスタンプで、指定したデジタル出力を ON/OFF します。タイムスタンプ方式を使用することにより、制御処理のタイミングの影響を受けずに、任意のタイミングでデジタル出力信号を ON/OFF することができます。

使用上の注意

- 本命令だけでは、デジタル出力を ON/OFF できません。
- NX シリーズエンコーダ入力ユニットを使用する場合、本命令は変化時刻取得機能が動作している必要があります。
  次のような場合、変化時刻取得機能は動作しません。
  a）変化時刻取得機能に対応していないエンコーダ入力、またはサーボドライバを使用している場合。
  b）NX シリーズエンコーダ入力ユニットのオブジェクト 6010Hex（タイムスタンプ）を PDO に割り付けていない場合。
  c）EtherCAT カプラユニットの［DC 有効］を［無効］に設定した場合。
- オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプを使用する場合、本命令は次のようなときには動作しません。
  a）オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプのオブジェクト 3211-83Hex（現在位置タイムスタンプ）を PDO に割り付けていない場合。
  b）オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプの［DC 有効］を［無効］に設定した場合。
- タイムスタンプが更新されない場合でも、本命令は異常とはなりません。このときも、ON/OFF 時刻が算出されますが、意図した値とはなりません。
  Sysmac Studio の MC モニタテーブルまたはウォッチウィンドウで、軸変数のメンバ「TimeStamp」（タイムスタンプ）が更新されていることを確認したうえで本命令を使用してください。
- NX_AryDOutTimeStamp 命令およびタイムスタンプ方式対応のデジタル出力ユニットと組み合わせて使用する場合、最小 ON/OFF 幅は、タスク周期の値および回転速度の値に比例します。
  たとえば、ロータリテーブルの 1 回転が 360° で回転速度が 800r/min、タスク周期が 500 μs の場合、最小 ON/OFF 幅は 5° となります。タスク周期だけを 1000 μs に変更した場合、最小 ON/OFF 幅は 10° となります。
- スイッチ構造体型変数の「FirstOnPosition」、「LastOnPosition」、および「Duration」の値は、デジタル出力の ON/OFF 幅が最小 ON/OFF 幅より大きくなるように指定してください。最小 ON/OFF 幅より小さくなる場合は、実際のデジタル出力が ON/OFF しないときがあります。
- 本命令は、軸の現在位置と現在速度をもとに、指定した位置に到達するタイムスタンプを算出します。算出されたタイムスタンプの精度は、エンコーダ分解能および軸の回転速度の影響を受けます。エンコーダ分解能が低い場合や軸の回転速度が低い場合に誤差が大きくなります。最大誤差の目安は次の式で計算できます。

  タイムスタンプの最大誤差(s) = 180/［エンコーダ分解能（パルス／回転）×回転速度(r/min)］

  エンコーダ分解能と回転速度に対するタイムスタンプの最大誤差の例を次に示します。

エンコーダ分解能（パルス／回転）	回転速度 (r/min)	算出されたタイムスタンプの最大誤差 (μs)
3600	400	±125.0
	800	±62.5
131072	400	±3.4
	800	±1.7

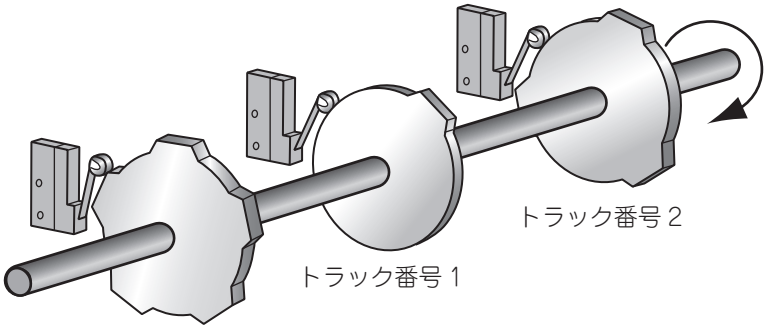
  また、軸が急な加減速をすると、算出誤差がさらに大きくなる可能性がありますので、軸が一定の速度の状態で使用してください。
  動作検証を十分に行い、安全を確認してください。
- 未使用軸を指定した場合、または MC 試運転中の場合、「Enable」が TRUE のときに「Busy」が TRUE となり、「InOperation」と「Error」は FALSE となります。
- 同じインスタンス名のインスタンスを 2 つ以上配置しないでください。配置すると、意図しない出力となる場合があります。

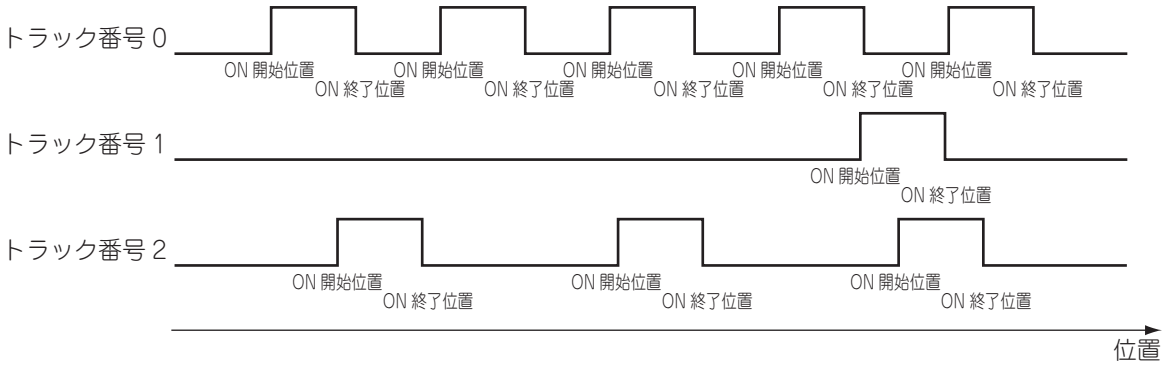
参考

NX_AryDOutTimeStamp 命令については、『NJ/NX シリーズ コマンドリファレンスマニュアル 基本編(SBCA-468)』を参照してください。

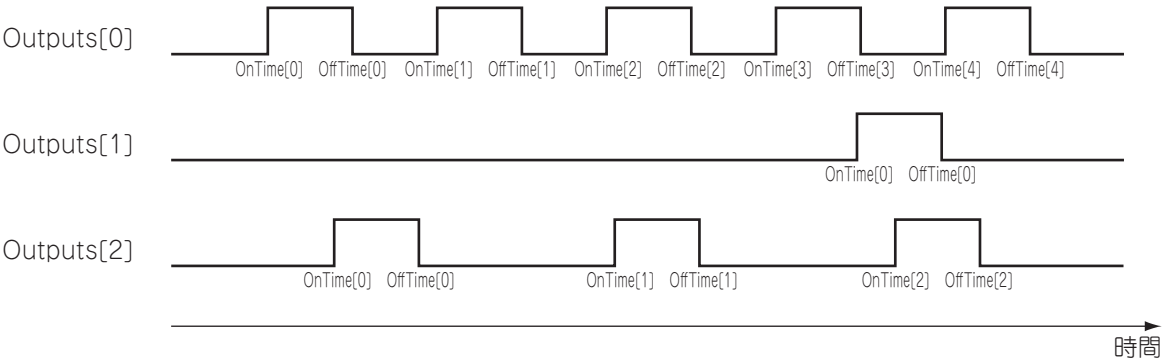
命令の詳細

本命令は、カムの回転角度によりセンサの出力信号を制御するメカ式カムスイッチの機能を、コントローラのプログラムで実現します。1 つのトラックは 1 枚のカムに相当します。



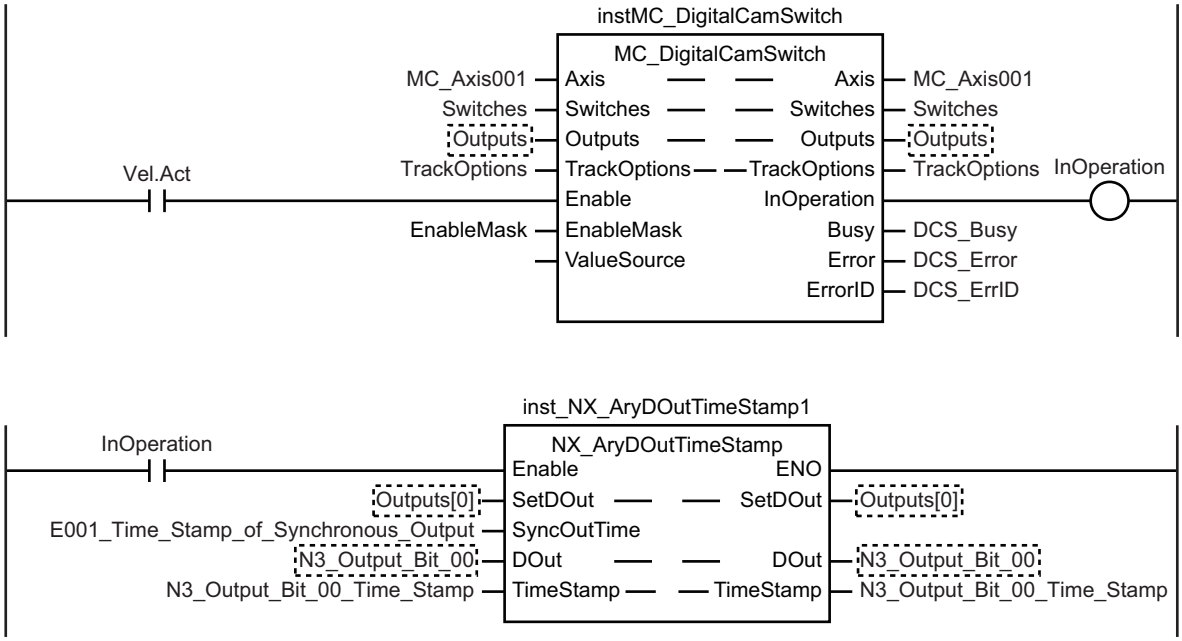


「TrackNumber」（トラック番号）は、カムの番号に相当します。「FirstOnPosition」（ON 開始位置）、「LastOnPosition」（ON 終了位置）は、カムの形状に相当します。
MC_DigitalCamSwitch 命令は、設定した ON 開始位置／ON 終了位置に到達する「OnTime」（ON 時刻）／「OffTime」（OFF 時刻）を算出し、「Outputs」（出力信号）で指定されたパラメータに格納します。



NX_AryDOutTimeStamp 命令は、「SetDOut」（出力パルス）で指定されたパラメータの「OnTime」（ON 時刻）／「OffTime」（OFF 時刻）に従って、実際のデジタル出力を ON/OFF します。

「SetDOut」（出力パルス）には、MC_DigitalCamSwitch 命令の「Outputs」（出力信号）のパラメータで指定した配列変数の要素を指定します。
「DOut」（DOut ユニット出力接点）には、実際のデジタル出力の指定として、タイムスタンプ方式対応のデジタル出力ユニットの出力接点に割り付けたデバイス変数を指定します。



次に命令の変数について説明します。

● Enable（有効）

- 「Enable」（有効）TRUE の間、命令を実行します。FALSE のときは「Outputs」の値を更新しません。

● EnableMask（トラック有効）

- 「Enable」（有効）が TRUE のとき、「EnableMask」（トラック有効）により、トラックごとの有効または無効を指定できます。デジタル出力ユニットの出力を OFF する場合は、対象のトラックを無効にしてください。
- 0 ビットめがトラック番号 0 を示し、15 ビットめがトラック番号 15 を示します。各ビットの値を「1」にすると対象トラックが有効となり、「0」にすると無効となります。ビットの値を「1」から「0」にすると、対象トラックのデジタル出力は OFF となります。
- 「EnableMask」に指定した値は、該当するトラック番号の「EnableOut」に反映されます。

● スイッチ構造体（_sCAMSWITCH_REF 型）

スイッチ構造体（_sCAMSWITCH_REF 型）は、出力信号の ON/OFF パターンを指定するための構造体型データ型です。本命令では、配列変数として最大 256 とおりの ON/OFF パターンを指定できます。1 つのトラックでは、最大 16 とおりの ON/OFF パターンを指定できます。
次に、スイッチ構造体のメンバについて説明します。

変数	名称	データ型	有効範囲	初期値	内容
TrackNumber	トラック番号	UINT	0～15 *1	0	対象となるトラック番号を指定します。
FirstOnPosition	ON 開始位置	LREAL	負数、正数、「0」	0	出力が ON する位置を指定します。*2

変数	名称	データ型	有効範囲	初期値	内容
LastOnPosition	ON 終了位置	LREAL	負数、正数、「0」	0	出力が OFF する位置を指定します。*2 スイッチモード選択を位置基準に指定した場合に有効となります。
AxisDirection	軸方向選択	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection 4: _mcNoDirection	0 *3	回転方向を指定します。 0: 正方向指定 2: 負方向指定 4: 方向指定なし（両方向）
CamSwitchMode	スイッチモード選択	_eMC_SWITCH_MODE	0: _mcSwitchDisable 1: _mcPositionBased 2: _mcTimeBased	0 *3	スイッチモードを指定します。 0: スイッチ無効 1: 位置基準 *4 2: 時間基準 *5
Duration	ON 時間	TIME	正数、T#0s	T#0s	出力が ON する時間を指定します。 スイッチモード選択を時間基準に指定した場合に有効となります。

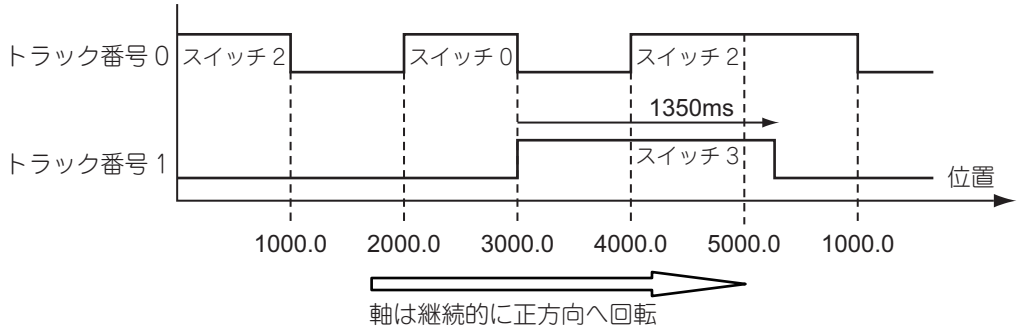
*1. 「Outputs」（出力信号）に指定した変数の要素番号までが指定可能です。
*2. 指令単位となります。指令単位は[mm]、[μm]、[nm]、[degree]、[inch] または[pulse]で表されます。
カウントモードが[リニアモード]の場合、設定範囲はパルス単位変換後で符号付き整数型 40 ビット（0x8000000000 ～ 7FFFFFFFFF）です。カウントモードが［ロータリモード］の場合、設定範囲はリングカウンタ下限値以上、リングカウンタ上限値未満です。
*3. 有効範囲が列挙型の変数は、実際の初期値は数値ではなく、列挙子となります。
*4. [1: _mcPositionBased]（位置基準）を指定した場合、「FirstOnPosition」（ON 開始位置）と「LastOnPosition」（ON 終了位置）の値に従って動作します。「Duration」（ON 時間）の値は関係ありません。
*5. [2: _mcTimeBased]（時間基準）を指定した場合は、「FirstOnPosition」（ON 開始位置）と「Duration」（ON 時間）の値に従って動作します。「LastOnPosition」（ON 終了位置）の値は関係ありません。

設定例

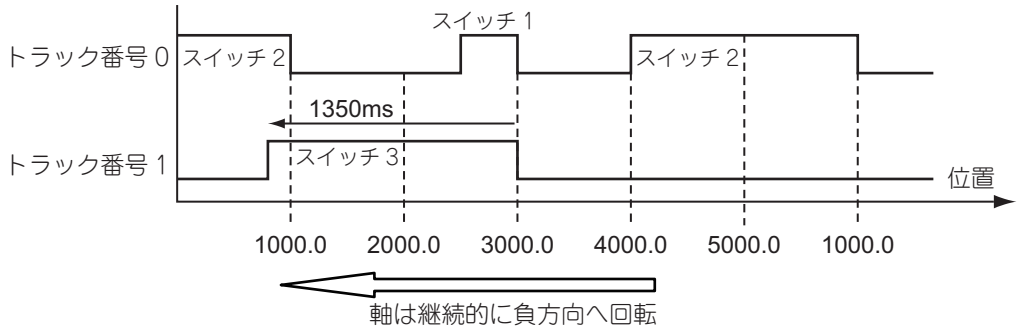
変数	名称	スイッチ 0	スイッチ 1	スイッチ 2	スイッチ 3	・・・	スイッチ 255
TrackNumber	トラック番号	0	0	0	1		
FirstOnPosition	ON 開始位置	2000.0	2500.0	4000.0	3000.0		
LastOnPosition	ON 終了位置	3000.0	3000.0	1000.0	— *1		
AxisDirection	軸方向選択	正方向指定	負方向指定	方向指定なし	方向指定なし		
CamSwitchMode	スイッチモード選択	位置基準	位置基準	位置基準	時間基準		
Duration	ON 時間	— *2	— *2	— *2	T#1350ms		

*1. スイッチモード選択が時間基準の場合、ON 開始位置と ON 時間で動作します。ON 終了位置は関係ありません。
*2. スイッチモード選択が位置基準の場合、ON 開始位置と ON 終了位置で動作します。ON 時間は関係ありません。

軸が継続的に正方向へ回転している場合の動作を次に示します。軸パラメータのカウントモードは［ロータリモード］でリングカウンタの範囲は 0～5000 です。



軸が継続的に負方向へ回転している場合の動作を次に示します。軸パラメータの設定は上記と同じです。



● 出力信号構造体（_sOUTPUT_REF 型）

出力信号構造体（_sOUTPUT_REF 型）は、スイッチの ON/OFF パターンデータをもとに算出されたデジタル信号の ON/OFF 時刻を表示するための構造体型データ型です。本命令では、「Outputs」（出力信号）の配列変数として最大 16 個の要素を扱います。「Outputs」（出力信号）の配列要素番号は、トラック番号を示します。

次に、出力信号構造体のメンバについて説明します。

変数	名称	データ型	有効範囲	内容
EnableOut	出力有効	BOOL	TRUE, FALSE	該当トラック番号の出力の有効／無効を示します。 「EnableMask」の同じトラック番号に相当するビットの値が反映されます。*1 TRUE: 該当トラック番号の出力が有効 FALSE: 該当トラック番号の出力が無効
OnTime	ON 時刻	ARRAY[0..15] OF ULINT	正数、「0」	デジタル出力を ON するタイムスタンプを示します。NX シリーズエンコーダ入力ユニット内の時刻を基準とするタイムスタンプが示されます。タスク周期ごとに値が更新されます。単位は[ns]です。
OffTime	OFF 時刻	ARRAY[0..15] OF ULINT	正数、「0」	デジタル出力を OFF するタイムスタンプを示します。NX シリーズエンコーダ入力ユニット内の時刻を基準とするタイムスタンプが示されます。タスク周期ごとに値が更新されます。単位は[ns]です。

*1. 「EnableMask」の i ビットの値が、「Outputs[i].EnableOut」に反映されます。

● トラックオプション構造体（_sTRACK_REF 型）

トラックオプション構造体（_sTRACK_REF 型）は、スイッチの動作条件を指定するための構造体型データ型です。本命令では、配列変数として最大 16 出力の条件を指定できます。

「TrackOptions」に指定する変数の要素数と「Outputs」に指定する変数の要素数は、同じ要素数としてください。
次に、トラックオプション構造体のメンバについて説明します。

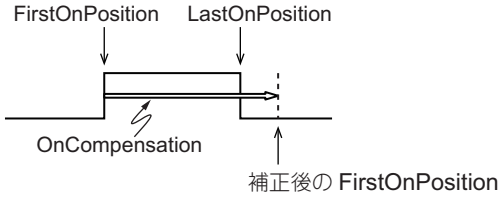
変数	名称	データ型	有効範囲	初期値	内容
OnCompensation	ON 時間補正	TIME	T#-1s～T#1s *1	T#0s	出力が ON するタイミングを補正します。正数の場合はタイミングを遅らせ、負数の場合は早めます。
OffCompensation	OFF 時間補正	TIME	T#-1s～T#1s *1	T#0s	出力が OFF するタイミングを補正します。正数の場合はタイミングを遅らせ、負数の場合は早めます。

*1. カウントモードを［ロータリモード］で使用する場合、CPU ユニットのバージョンにより以下の制約があります。
- ユニットバージョン Ver.1.09 以降の CPU ユニットでは、軸の±半回転の範囲を超える値を設定すると「InOperation」（作動中）が FALSE になります。
- ユニットバージョン Ver.1.08 以前の CPU ユニットでは、有効範囲が「T#-1s～T#0s」となります。

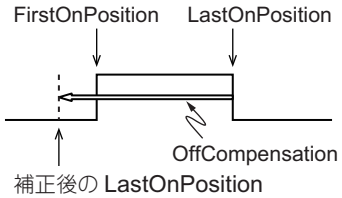
「OnCompensation」（ON 時間補正）および「OffCompensation」（OFF 時間補正）は、機械の微小な動作遅れやずれなどを補正するために使用します。

安全上の要点

MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令は、必ず軸を一定速度で使用してください。
カウントモードを［ロータリモード］にして、「OnCompensation」（ON 時間補正）または「OffCompensation」（OFF 時間補正）を使用し、軸速度が急変すると、以下のような動作となります。
- 「OnCompensation」（ON 時間補正）または「OffCompensation」（OFF 時間補正）の値が、軸の半回転以上に相当する時間になる場合は、「InOperation」（作動中）が FALSE となります。
- 「OnCompensation」（ON 時間補正）の値が、「LastOnPosition」（ON 終了位置）を超える場合は、出力タイミングが不定となります。



- 「OffCompensation」（OFF 時間補正）の値が、「FirstOnPosition」（ON 開始位置）を超える場合は、出力タイミングが不定となります。



使用上の注意

カウントモードを［ロータリモード］にして、「OnCompensation」（ON 時間補正）または「OffCompensation」（OFF 時間補正）を使用する場合は、「FirstOnPosition」（ON 開始位置）と「LastOnPosition」（ON 終了位置）の位置関係が逆転しないように設定してください。出力タイミングが不定となります。
動作については、前述の“安全上の要点”を参照してください。

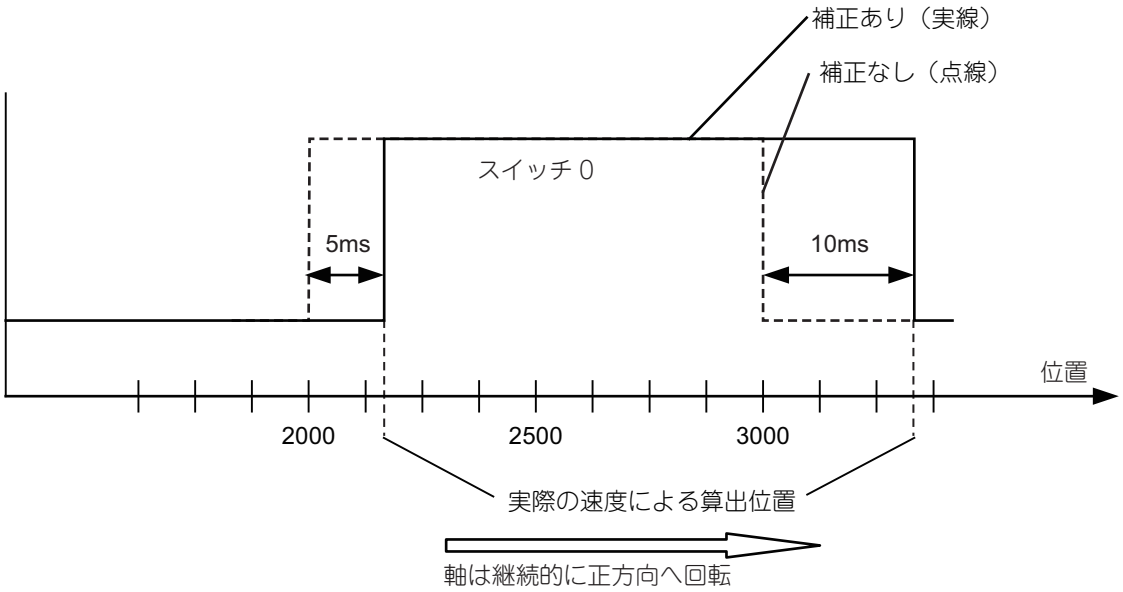
バージョン関連情報

カウントモードを［ロータリモード］で使用する場合、CPU ユニットのバージョンにより以下の制約があります。
ユニットバージョン Ver.1.09 以降の CPU ユニットと Ver.1.08 以前では有効範囲が異なります。CPU ユニットを Ver.1.09 以降にバージョンアップする場合は、有効範囲に注意してください。

ユニットバージョン Ver.1.09 以降の CPU ユニット
有効範囲は「T#-1s～T#1s」ですが、設定にあたっては以下の制約があります。
- 軸の半回転分の時間を超えない範囲で設定してください。
  例えば、500r/min で回転させる場合、1 回転にかかる時間は 120ms となります。半回転分の時間ですので、「OnCompensation」（ON 時間補正）、「OffCompensation」（OFF 時間補正）は、-60ms から 60ms の間の値を設定してください。
- 軸の半回転の時間を超える時間を設定した場合、「InOperation」（作動中）が FALSE になると同時に、「EnableOut」（出力有効）が FALSE になります。必ず「InOperation」（作動中）の状態を確認しながら使用してください。
- 上記により「InOperation」（作動中）が FALSE になったあと、値を正常範囲内に戻すと、「InOperation」（作動中）が TRUE となります。

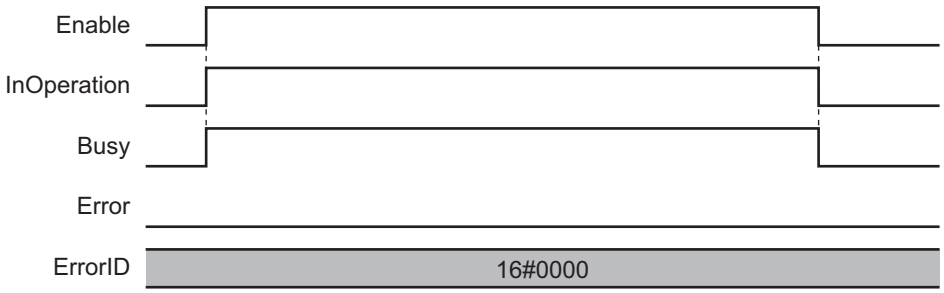
ユニットバージョン Ver.1.08 以前の CPU ユニット
- 有効範囲が「T#-1s～T#0s」となります。

前述の「設定例(P.3-401)」で、さらに「OnCompensation」（ON 時間補正）に T#5ms を、「OffCompensation」（OFF 時間補正）に T#10ms を設定したときの動作を次に示します。

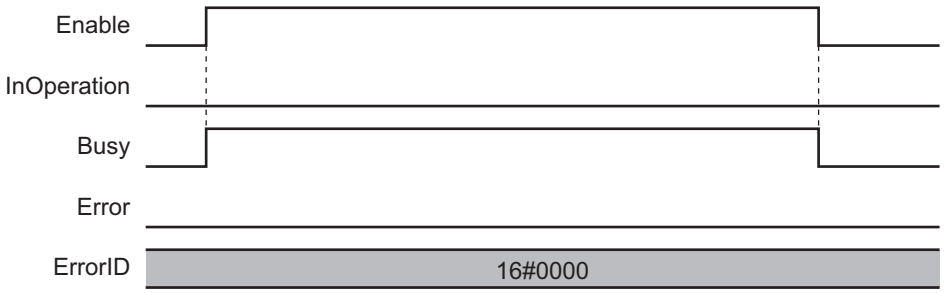


タイミングチャート

本命令を実行した場合のタイミングチャートを下記に示します。



未使用軸を指定した場合、または MC 試運転中の場合のタイミングチャートを示します。
ユニットバージョン Ver.1.09 以降の CPU ユニットでカウントモードが［ロータリモード］の場合は、「OnCompensation」（ON 時間補正）、または「OffCompensation」（OFF 時間補正）に軸の半回転分を超える時間を設定したときも同様のタイミングチャートになります。



このとき、「Outputs」の「EnableOut」は FALSE となります。

モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

モーション命令多重起動

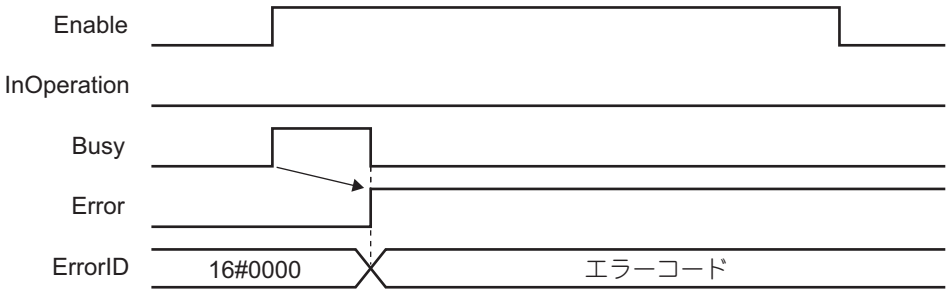
本命令は個々の命令インスタンスごとに独立して動作するため、モーション命令多重起動の制約はありません。

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● **異常発生時のタイミングチャート**



このとき、「Outputs」の「EnableOut」は FALSE となります。

● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、前述の「設定例(P.3-401)」で示した例のサンプルプログラムを掲載します。

使用機器

このサンプルプログラムで使用するおもな機器を次に示します。

機器	形式
EtherCAT カプラユニット	形 NX-ECC201(Ver.1.1) *1
パルス出力ユニット	形 NX-PG0122 *2
インクリメンタルエンコーダ入力ユニット	形 NX-EC0122 *3
デジタル出力ユニット	形 NX-OD2154 *4

*1. ノードアドレスを「1」、デバイス名を「E001」とします。
*2. NX ユニット番号を「1」、デバイス名を「N1」とします。軸 1 に割り付けます。
*3. NX ユニット番号を「2」、デバイス名を「N2」とします。軸 2 に割り付けます。
*4. NX ユニット番号を「3」、デバイス名を「N3」とします。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● **軸パラメータ**

軸種別

軸	軸種別
軸 1	サーボ軸
軸 2	エンコーダ軸

カウンタモード

軸	カウンタモード
軸 1	ロータリモード

軸	カウンタモード
軸 2	ロータリモード

リングカウンタ

軸	上限値	下限値
軸 1	5000	0
軸 2	5000	0

座標単位選択

軸	座標単位
軸 1	pulse
軸 2	pulse

ラダー図

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis001	_sAXIS_REF	−	軸 2 の軸変数です。
E001_Time_Stamp_of_Synchronous_Output	ULINT	−	デバイス変数です。*1
N3_Output_Bit_00	BOOL	−	デバイス変数です。
N3_Output_Bit_00_Time_Stamp	ULINT	−	デバイス変数です。
N3_Output_Bit_01	BOOL	−	デバイス変数です。
N3_Output_Bit_01_Time_Stamp	ULINT	−	デバイス変数です。
Switches	ARRAY[0..3] OF _sCAMSWITCH_REF	−	入出力変数「Switches」の入力パラメータです。 要素番号は、スイッチ番号に対応します。
Outputs	ARRAY[0..1] OF _sOUTPUT_REF	−	入出力変数「Outputs」の入力パラメータです。 要素番号は、トラック番号に対応します。
TrackOptions	ARRAY[0..1] OF _sTRACK_REF	−	入出力変数「TrackOptions」の入力パラメータです。 要素番号は、トラック番号に対応します。
EnableMask	WORD	16#0003	入力変数「EnableMask」の入力パラメータです。 トラック 0 とトラック 1 が有効です。

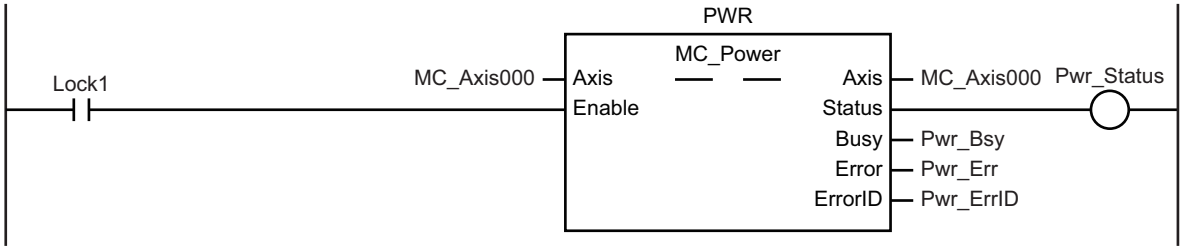
*1. EtherCAT カプラユニットの I/O エントリに、0x200A:02(Time Stamp of Synchronous Output)を追加しておく必要があります。

● サンプルプログラム

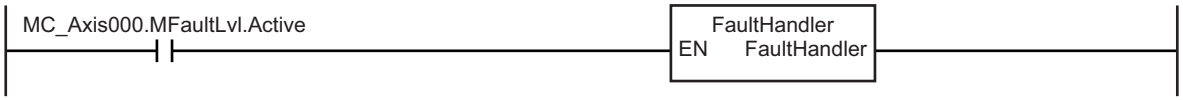
接点 StartPg が TRUE になったら、軸 1 のサーボドライバがサーボレディ状態になっていることを確認する



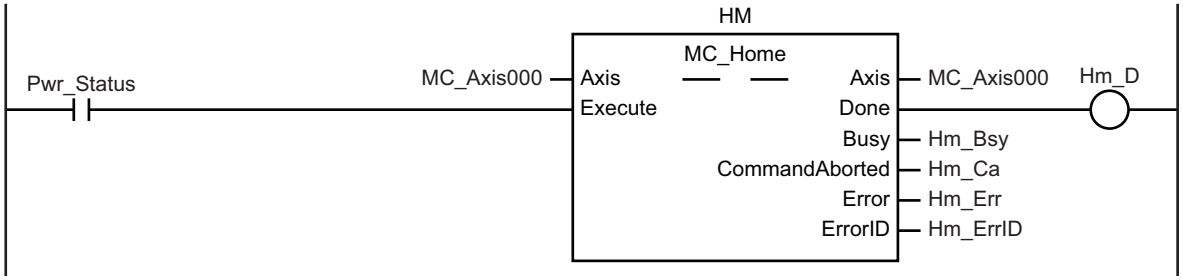
軸 1 がサーボレディ状態になっていれば、サーボ ON 状態にする



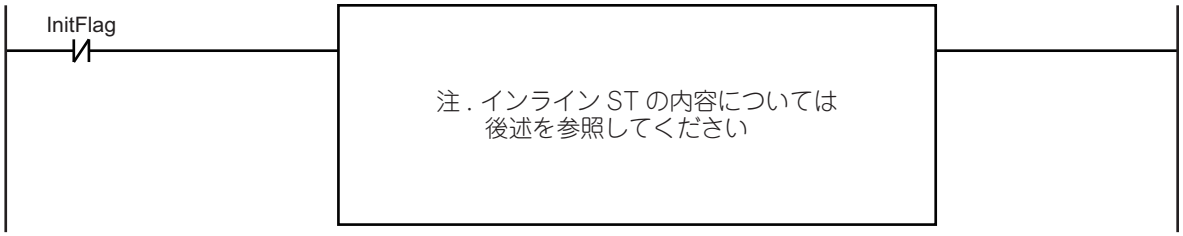
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



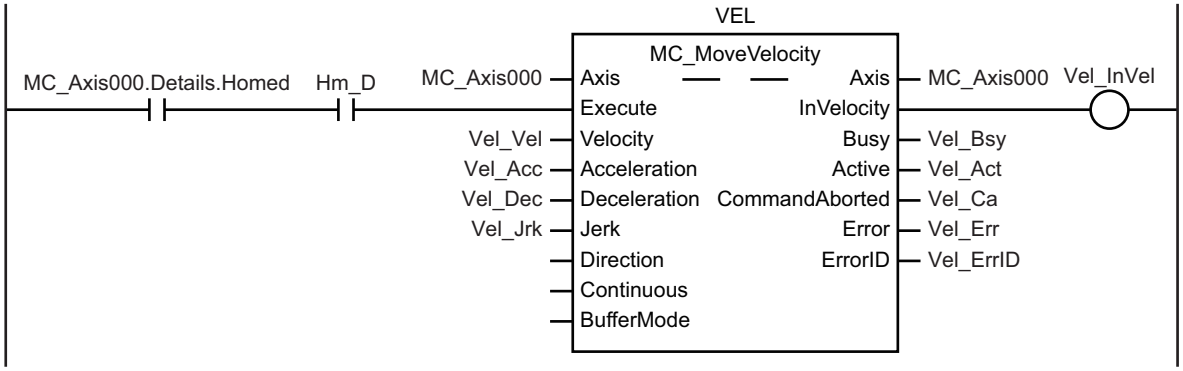
軸 1 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



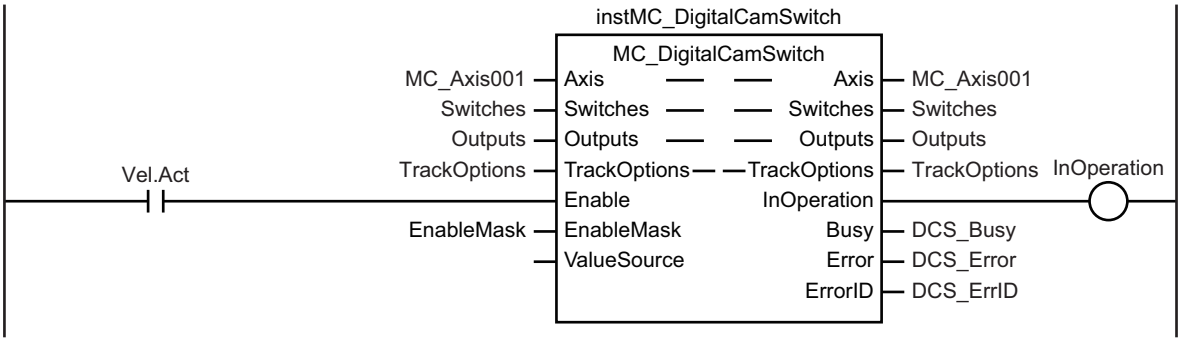
MC_MoveVelocity（速度制御）命令のパラメータと MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令の Switches のパラメータを設定する



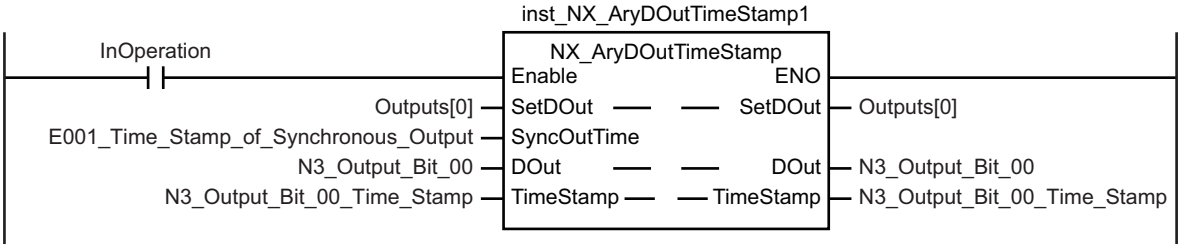
軸 1 が原点確定状態になったら、MC_MoveVelocity(速度制御）命令を実行する

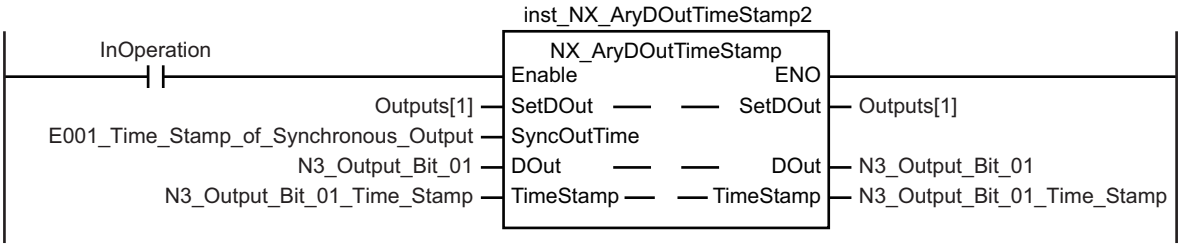


軸 1 の MC_MoveVelocity(速度制御）命令が実行されたら、軸 2 の MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令を実行する



軸 2 の MC_DigitalCamSwitch(デジタルカムスイッチ有効)命令が作動したら、NX_AryDOutTimeStamp(タイムスタンプデジタル出力配列書込）命令を実行する





インライン ST の内容

```
//MC_MoveVelocity パラメータ
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 パラメータ
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;
```

ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis001	_sAXIS_REF	−	軸 2 の軸変数です。
E001_Time_Stamp_of_Synchronous_Output	ULINT	−	デバイス変数です。*1
N3_Output_Bit_00	BOOL	−	デバイス変数です。
N3_Output_Bit_00_Time_Stamp	ULINT	−	デバイス変数です。
N3_Output_Bit_01	BOOL	−	デバイス変数です。
N3_Output_Bit_01_Time_Stamp	ULINT	−	デバイス変数です。
Pwr_En	BOOL	−	MC_Power のインスタンス PWR の入力変数「Enable」に割り付けた変数です。
Switches	ARRAY[0..3] OF _sCAMSWITCH_REF	−	入出力変数「Switches」の入力パラメータです。 要素番号は、スイッチ番号に対応します。
Outputs	ARRAY[0..1] OF _sOUTPUT_REF	−	入出力変数「Outputs」の入力パラメータです。 要素番号は、トラック番号に対応します。
TrackOptions	ARRAY[0..1] OF _sTRACK_REF	−	入出力変数「TrackOptions」の入力パラメータです。 要素番号は、トラック番号に対応します。
EnableMask	WORD	16#0003	入力変数「EnableMask」の入力パラメータです。 トラック 0 とトラック 1 が有効です。
Hm_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_Home のインスタンス HM を起動します。
Vel_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveVelocity のインスタンス VEL を起動します。

*1. EtherCAT カプラユニットの I/O エントリに、0x200A:02(Time Stamp of Synchronous Output)を追加しておく必要があります。

● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag=FALSE THEN

    //MC_MoveVelocity パラメータ
    Vel_Vel := LREAL#1000.0;
    Vel_Acc := LREAL#0.0;
    Vel_Dec := LREAL#0.0;
    Vel_Jrk := LREAL#1000.0;

    //MC_DigitalCamSwitch パラメータ
    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;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag:=TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//軸 1 で軽度フォールトが発生したら、異常時処理 FaultHandler を実行します
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
```

```
END_IF;

//サーボ ON 状態で、かつ原点未確定なら原点復帰を実行します
IF (Pwr_Status=TRUE) THEN
    Hm_Ex:=TRUE;
END_IF;

//原点確定後に、MC_MoveVelocity を実行します
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

指定したタイムスタンプの軸の位置を算出します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_TimeStampToPos	タイムスタンプ→軸位置算出	FB	MC_TimeStampToPos_instance MC_TimeStampToPos Axis — — Axis Enable — Enabled TimeStamp — Busy ValueSource — CalcPosition — Error — ErrorID	MC_TimeStampToPos_instance (Axis :=「パラメータ」, Enable :=「パラメータ」, TimeStamp :=「パラメータ」, ValueSource :=「パラメータ」, Enabled =>「パラメータ」, Busy =>「パラメータ」, CalcPosition =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

使用上の注意

- 本命令は、下記のユニットに対して有効です。
  a）NX シリーズ位置インタフェースユニットを割り付けている軸。
  使用可能な NX ユニットは、形 NX-EC0□□□、形 NX-ECS□□□です。
  b）オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプ。
- 本命令をオムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプで使用する場合は、オブジェクト 3211-83Hex（現在位置タイムスタンプ）を PDO に割り付ける必要があります。

バージョン関連情報

本命令は、ユニットバージョン Ver.1.06 以降の CPU ユニットと Sysmac Studio Ver.1.07 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE の間、命令を実行します。
TimeStamp	タイムスタンプ	ULINT	正数、または「0」	0	位置を算出するタイムスタンプを指定します。 タイムスタンプ方式対応のデジタル入力ユニットやエンコーダ入力ユニット、オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプ内の時刻を基準とするタイムスタンプを指定します。単位は[ns]です。
ValueSource（Reserved）	入力情報	_sMC_SOURCE	−	−	（Reserved）

出力変数

出力変数	名称	データ型	有効範囲	内容
Enabled	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CalcPosition	算出位置	LREAL	負数、正数、「0」	指定したタイムスタンプの位置を出力します。 単位は［指令単位］です。*1
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*2	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Enabled	「Enable」が TRUE となったとき。	• 「Enable」が FALSE の場合、1 周期後 • 「Error」が TRUE となったとき。
Busy	「Enable」の立ち上がり	• 「Enable」が FALSE となったとき。 • 「Error」が TRUE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

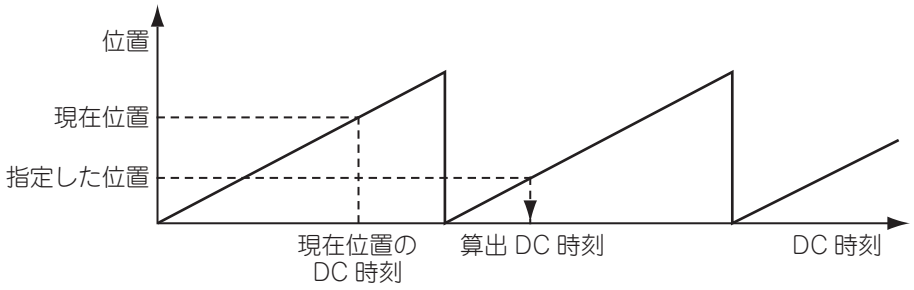
入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1 *2

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. NX シリーズエンコーダ入力ユニットを割り付けたエンコーダ軸、またはオムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプの軸を指定してください。

機能説明

- 本命令は、軸のフィードバック現在位置とタイムスタンプをもとに、入力変数で指定したタイムスタンプのフィードバック現在位置を算出します。
- 軸には、NX シリーズエンコーダ入力ユニットを指定してください。
- 軸パラメータのカウントモードが［リニアモード］の場合、算出した位置がオーバーフローまたはアンダーフローとなったとき、算出値はオーバーフローまたはアンダーフローの値となります。この場合、異常とはなりません。

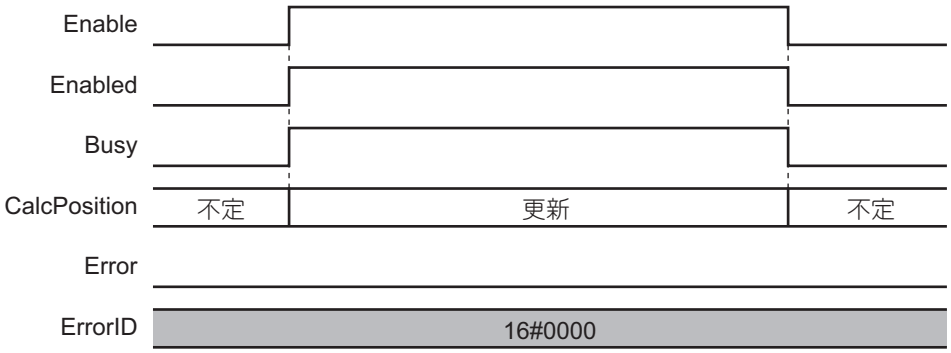


使用上の注意

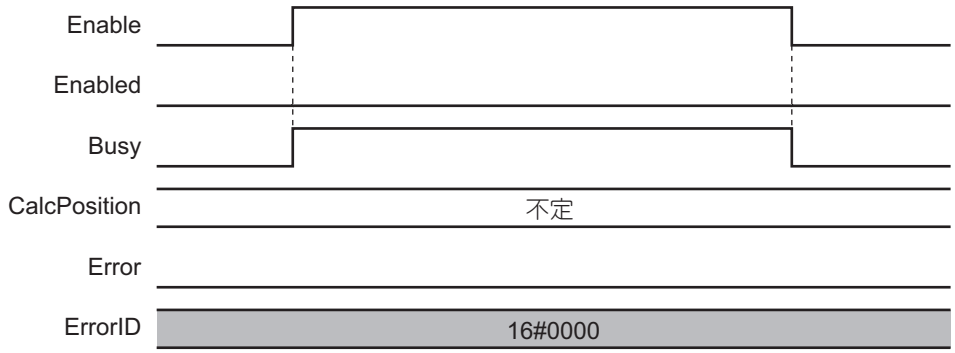
- NX シリーズエンコーダ入力ユニットを使用する場合、本命令は変化時刻取得機能が動作している必要があります。
  次のような場合、変化時刻取得機能は動作しません。
  a）変化時刻取得機能に対応していないエンコーダ入力またはサーボドライバを使用している場合。
  b）NX シリーズエンコーダ入力ユニットのオブジェクト 6010Hex（タイムスタンプ）を PDO に割り付けていない場合。
  c）EtherCAT カプラユニットの［DC 有効］を［無効］に設定した場合。
- オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプを使用する場合、本命令は次のようなときには動作しません。
  a）オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプのオブジェクト 3211-83Hex（現在位置タイムスタンプ）を PDO に割り付けていない場合。
  b）オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプの［DC 有効］を［無効］に設定した場合。
- タイムスタンプが更新されない場合でも、本命令は異常とはなりません。このときも、位置が算出されますが、指定したタイムスタンプでの位置とはなりません。
  Sysmac Studio の MC モニタテーブルまたはウォッチウィンドウで、軸変数のメンバ「TimeStamp」（タイムスタンプ）が更新されていることを確認したうえで本命令を使用してください。
- 本命令は、軸の現在位置と現在速度をもとに、指定したタイムスタンプでの位置を算出します。軸が急な加減速をするときは、算出誤差が大きくなる可能性がありますので、軸が一定の速度の状態で使用してください。
  動作検証を十分に行い、安全を確認してください。
- 未使用軸を指定した場合、または MC 試運転中の場合、「Enable」が TRUE のときに「Busy」が TRUE となり、「Enabled」と「Error」は FALSE となります。
- 同じインスタンス名のインスタンスを 2 つ以上配置しないでください。配置すると、意図しない出力となる場合があります。

タイミングチャート

本命令を実行した場合のタイミングチャートを下記に示します。



未使用軸を指定した場合、または MC 試運転中の場合のタイミングチャートを示します。



モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

モーション命令多重起動

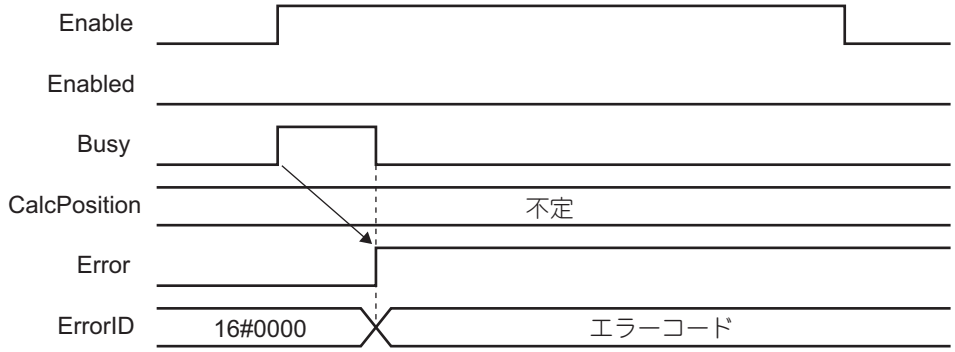
本命令は個々の命令インスタンスごとに独立して動作するため、モーション命令多重起動の制約はありません。

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、ベルトコンベア上を移動するワーク間の距離を計測するサンプルプログラムを掲載します。

使用機器

このサンプルプログラムで使用するおもな機器を次に示します。

機器	形式
EtherCAT カプラユニット	形 NX-ECC201(Ver.1.1) *1
パルス出力ユニット	形 NX-PG0122 *2
インクリメンタルエンコーダ入力ユニット	形 NX-EC0122 *3
デジタル入力ユニット	形 NX-ID3344 *4

*1. ノードアドレスを「1」、デバイス名を「E001」とします。
*2. NX ユニット番号を「1」、デバイス名を「N1」とします。軸 1 に割り付けます。
*3. NX ユニット番号を「2」、デバイス名を「N2」とします。軸 2 に割り付けます。
*4. NX ユニット番号を「3」、デバイス名を「N3」とします。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータ

軸種別

軸	軸種別
軸 1	サーボ軸
軸 2	エンコーダ軸

カウンタモード

軸	カウンタモード
軸 1	ロータリモード
軸 2	ロータリモード

リングカウンタ

軸	上限値	下限値
軸 1	360	0
軸 2	360	0

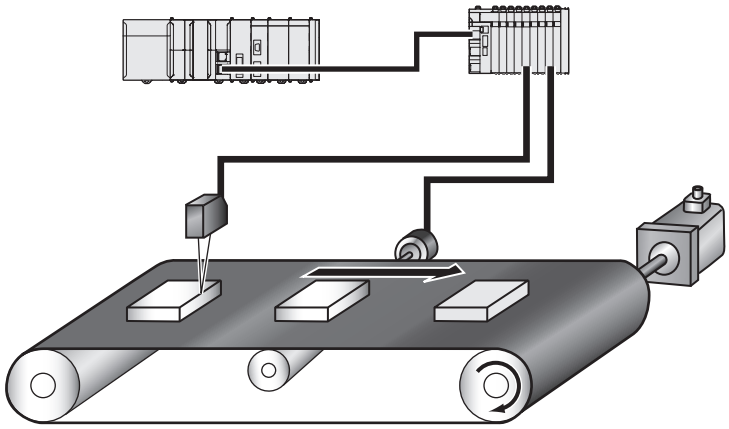
座標単位選択

軸	座標単位
軸 1	degree
軸 2	degree

動作例

センサがワークを検出するとセンサ出力は ON となり、ワークが通過するとセンサ出力は OFF となります。センサが次のワークを検出すると、センサ出力は再度 ON となります。

センサ出力が ON したときのタイムスタンプをもとに、エンコーダ入力の位置を計算します。2 つの位置の差がワーク間の距離となります。



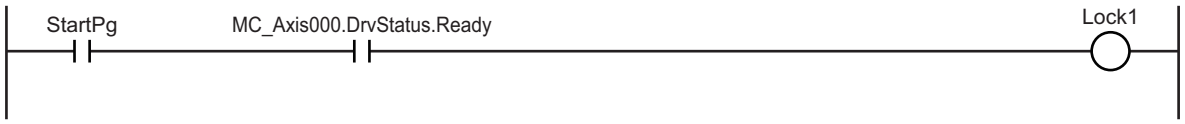
ラダー図

● 主な変数

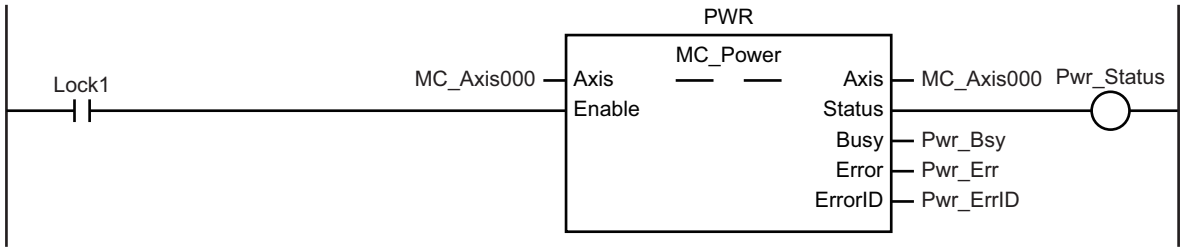
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis001	_sAXIS_REF	−	軸 2 の軸変数です。
N3_Input_Bit_00	BOOL	−	デバイス変数です。
N3_Input_Bit_00_Time_Stamp	ULINT	−	デバイス変数です。
Position	ARRAY[0..1] OF LREAL	−	算出位置を格納します。
Count	ARRAY[0..1] OF ULINT	−	多回転数を格納します。
FirstPoint	UINT	−	処理用の変数です。
LastPoint	UINT	−	処理用の変数です。
Distance	LREAL	−	ワーク間の距離です。

● サンプルプログラム

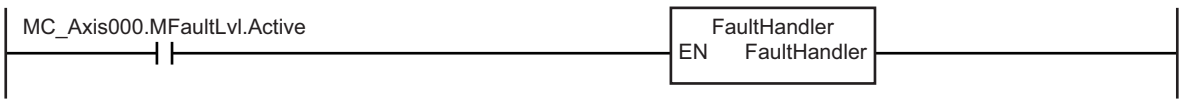
接点 StartPg が TRUE になったら、軸 1 のサーボドライバがサーボレディ状態になっていることを確認する



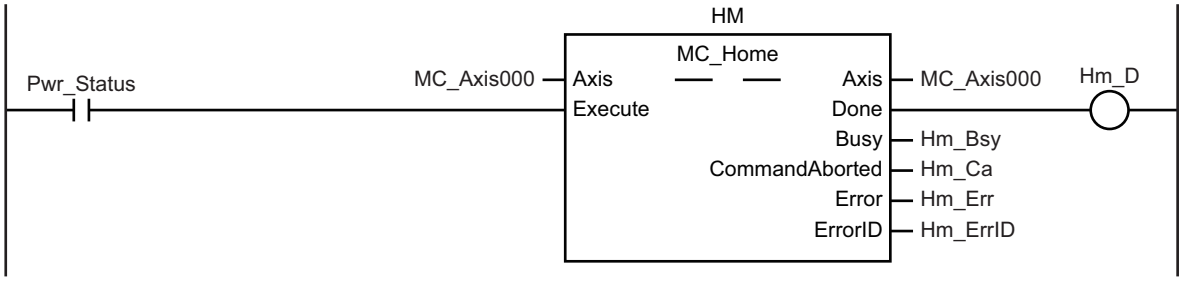
軸 1 がサーボレディ状態になっていれば、サーボ ON 状態にする



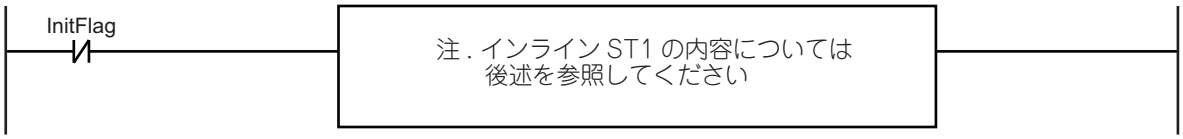
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



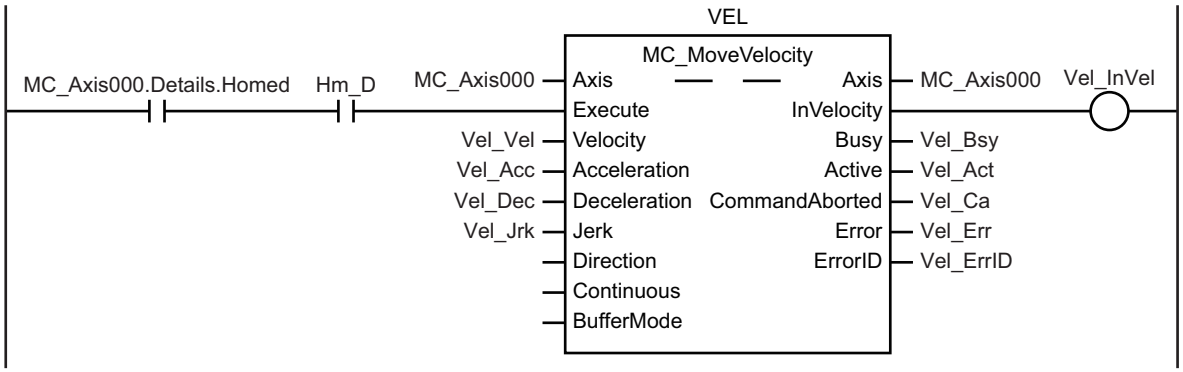
軸 1 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



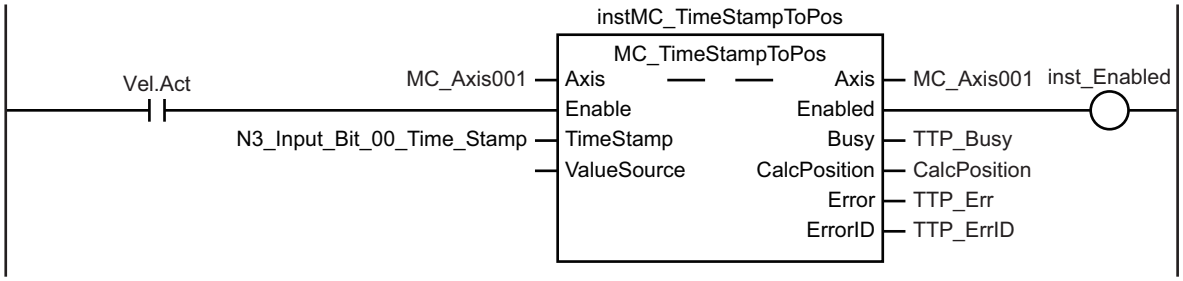
MC_MoveVelocity（速度制御）命令のパラメータを設定する



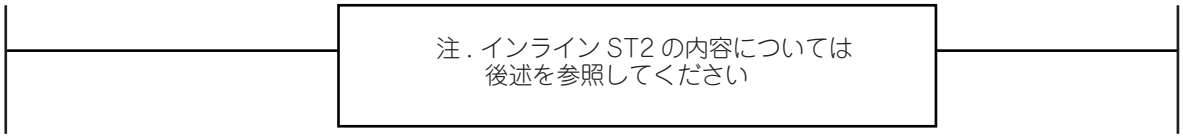
軸 1 が原点確定状態になったら、MC_MoveVelocity(速度制御）命令を実行する



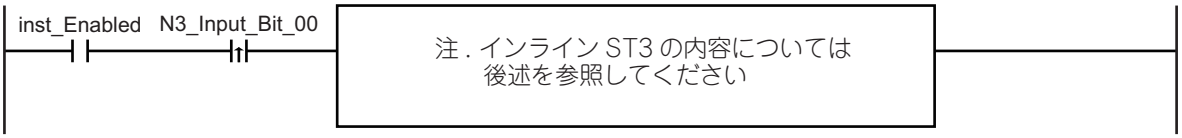
軸 1 の MC_MoveVelocity(速度制御）命令が実行されたら、軸 2 の MC_TimeStampToPos（タイムスタンプ→軸位置算出）命令を実行する



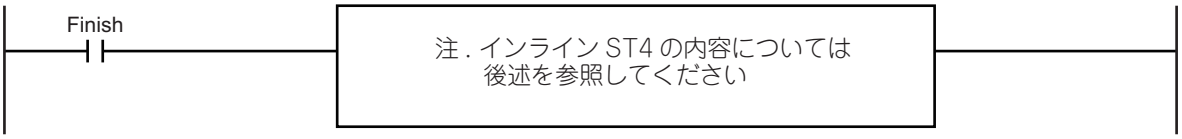
エンコーダ軸の多回転数をカウントする（今回値 < 前回値のとき、リングカウンタの上限値を超えたと判断）



センサ出力の ON 立ち上がり時に CalcPosition（算出位置）を取得する（Position[0] と Position[1] に交互に保存）



２つめのワークの検出以降、ワーク間の距離を算出する



インライン ST1 の内容

```
//MC_MoveVelocity パラメータ
Vel_Vel := LREAL#1000.0;
Vel_Acc := LREAL#0.0;
Vel_Dec := LREAL#0.0;
Vel_Jrk := LREAL#1000.0;
InitFlag := BOOL#TRUE;
```

インライン ST2 の内容

```
IF MC_Axis001.Act.Pos < PreAxis001ActPos THEN
    Inc(RotaryCount);
END_IF;
PreAxis001ActPos := MC_Axis001.Act.Pos;
```

インライン ST3 の内容

```
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;
```

インライン ST4 の内容

```
//1 回めの比較：ワーク 1=Position[0]、ワーク 2=Position[1]
//2 回めの比較：ワーク 2=Position[1]、ワーク 3=Position[0]
//3 回めの比較：ワーク 3=Position[0]、ワーク 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]);
```

ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis001	_sAXIS_REF	−	軸 2 の軸変数です。
N3_Input_Bit_00	BOOL	−	デバイス変数です。
N3_Input_Bit_00_Time_Stamp	ULINT	−	デバイス変数です。
Hm_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_Home のインスタンス HM を起動します。
Vel_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveVelocity のインスタンス VEL を起動します。
PreN3InputBit00	BOOL	−	デバイス変数 N3_Input_Bit_00 の前回値です。
Position	ARRAY[0..1] OF LREAL	−	算出位置を格納します。
Count	ARRAY[0..1] OF ULINT	−	多回転数を格納します。
FirstPoint	UINT	−	処理用の変数です。
LastPoint	UINT	−	処理用の変数です。
Distance	LREAL	−	ワーク間の距離です。

● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag=FALSE THEN

    //MC_MoveVelocity パラメータ
    Vel_Vel := LREAL#1000.0;
    Vel_Acc := LREAL#0.0;
    Vel_Dec := LREAL#0.0;
    Vel_Jrk := LREAL#1000.0;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag:=TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
ELSE
    Pwr_En:=FALSE;
END_IF;

//軸 1 で軽度フォールトが発生したら、異常時処理 FaultHandler を実行します
```

```
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
    FaultHandler();
END_IF;

//サーボ ON 状態で、かつ原点未確定なら原点復帰を実行します
IF (Pwr_Status=TRUE) THEN
    Hm_Ex:=TRUE;
END_IF;

//原点確定後に、MC_MoveVelocity を実行します
IF MC_Axis000.Details.Homed=TRUE AND Hm_D=TRUE THEN
    Vel_Ex:=TRUE;
END_IF;

//エンコーダ軸の多回転数をカウントします
//（今回値<前回値のとき、リングカウンタの上限値を超えたと判断）
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 );

//センサ出力の ON 立ち上がり時に CalcPosition（算出位置）を取得します
//（Position[0]と Position[1]に交互に保存）
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;

//2 つめのワークの検出以降、ワーク間の距離を算出します
```

```
//1 回めの比較：ワーク 1=Position[0]、ワーク 2=Position[1]
//2 回めの比較：ワーク 2=Position[1]、ワーク 3=Position[0]
//3 回めの比較：ワーク 3=Position[0]、ワーク 4=Position[1]
//...
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
);
```

MC_PeriodicSyncVariables

軸変数のタスク間同期を定周期に実行します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_PeriodicSyncVariables	軸変数タスク間定周期同期	FB	MC_PeriodicSyncVariables_instance MC_PeriodicSyncVariables Axis — — Axis Enable — Enabled ExecID — Busy — Error — ErrorID	MC_PeriodicSyncVariables_instance (Axis :=「パラメータ」, Enable :=「パラメータ」, ExecID :=「パラメータ」, Enabled =>「パラメータ」, Busy =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

使用上の注意

本命令は、形 NX502 CPU ユニット、形 NX102 CPU ユニット、形 NX1P2 CPU ユニットおよび NJ シリーズ CPU ユニットでは使用できません。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
ExecID	実行 ID	UINT	2	2	変数同期指定先の ID を指定します。 2: 定周期タスク（実行優先度 5）

出力変数

出力変数	名称	データ型	有効範囲	内容
Enabled	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Enabled	「Enable」が TRUE となったとき。	• 「Enable」が FALSE の場合、1 周期後 • 「Error」が TRUE となったとき。
Busy	「Enable」の立ち上がり	• 「Enable」が FALSE となったとき。 • 「Error」が TRUE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 「Enable」（有効）を TRUE にすると、プライマリ定周期タスクと定周期タスク（実行優先度 5）のタスク間で、軸変数を周期的に同期します。
- 同期した軸変数は、指定先のタスクの軸変数領域（システム変数）に出力されます。



使用上の注意

- 本命令は、プライマリ定周期タスクから定周期タスク（実行優先度 5）への変数同期だけに対応しています。
- 「ExecID」（実行 ID）に指定した値と、その軸が割り付けられているタスクが同じ場合、異常は出さずに本命令は「Busy」（実行中）のままとなります。



参考

異なるタスクで制御する軸の軸変数をユーザプログラムで外部変数宣言して参照すると、ユーザプログラムの実行開始位置によって、どの時点で更新された軸変数を参照するかわかりません。
電子カムなど制御する軸が主従関係を持つアプリケーションを、異なるタスクで制御する場合は、MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令を使用してください。

命令の詳細

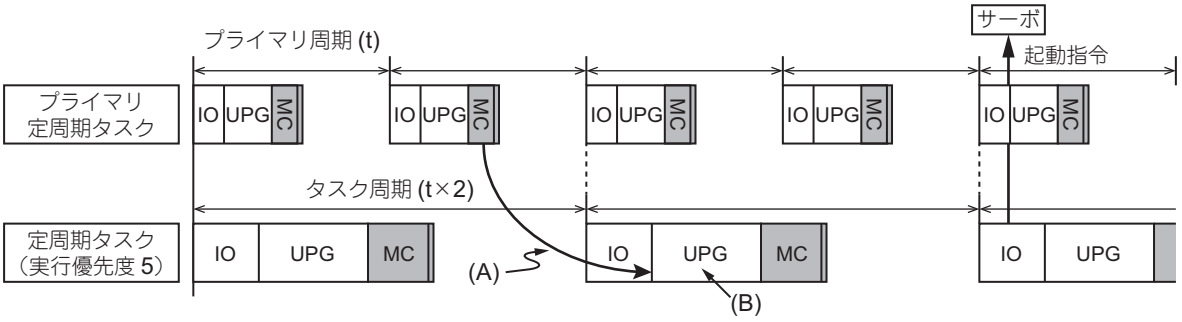
以下に本命令の詳細について説明します。

● 同期タイミング

プライマリ定周期タスクで制御する軸の軸変数の値を、定周期タスク（実行優先度 5）のユーザプログラムで参照することができます。
定周期タスク（実行優先度 5）のユーザプログラムが実行している途中で、プライマリ定周期タスクの軸の軸変数の値が書きかわることはありません。

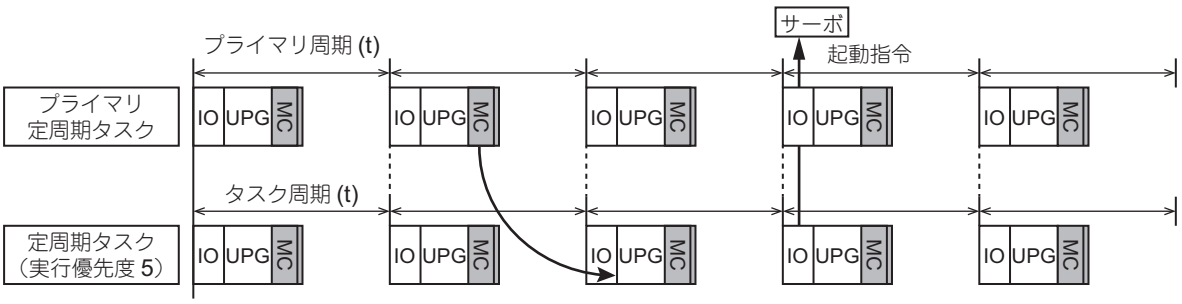
定周期タスク（実行優先度 5）のユーザプログラムは、プライマリ定周期タスクと定周期タスク（実行優先度 5）のタスク周期の先頭がそろう周期の直前のプライマリ定周期タスクの実行結果を反映します。

プライマリ定周期タスクの軸変数の値を定周期タスク（実行優先度 5）に反映するタイミングは、次のようになります。



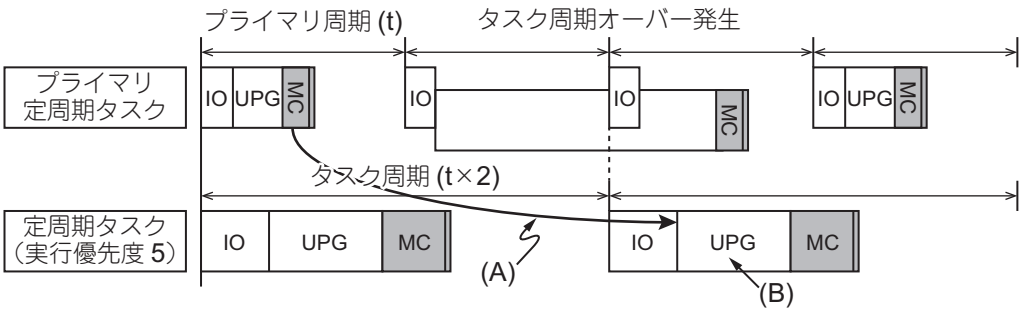
記号	説明
（A）	軸変数の更新。 定周期タスク（実行優先度 5）のユーザプログラム実行開始位置にかかわらず、タスク周期の先頭がそろう周期の直前のプライマリ定周期タスクの実行結果を反映します。
（B）	ユーザプログラム実行の途中で、軸変数の値が書きかわることはありません。

タスク周期が同じ場合も、同様になります。



参考

タスク周期オーバーが発生した場合、タスク周期の先頭がそろう周期の直前のタスク周期の実行結果ではなく、さらにひとつ前の周期の実行結果が反映されます。



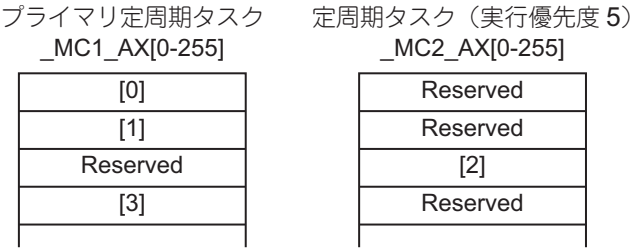
記号	説明
（A）	軸変数の更新。 タスク周期オーバーが発生した場合、ひとつ前の周期の値を反映します。
（B）	ユーザプログラム実行の途中で、軸変数の値が書きかわることはありません。

● 変数の出力領域

同期した軸変数は、指定先のタスクの軸変数領域（システム変数）に出力されます。

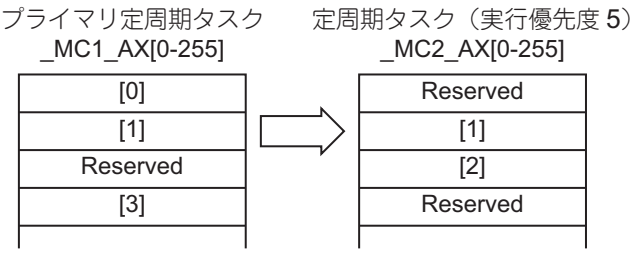
例えば、軸パラメータの［モーション制御］を次のように設定した場合、軸変数（システム変数）は、下図の領域が更新対象となります。

軸番号	モーション制御（割り付けタスク）
軸 0	プライマリ定周期タスク
軸 1	プライマリ定周期タスク
軸 2	定周期タスク（実行優先度 5）
軸 3	プライマリ定周期タスク



（注）Reserved は、基本設定を除き、初期値になります。

MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令の「Axis」（軸）に軸 1 を、「ExecID」（実行 ID）に「10#2」を指定し実行すると、_MC2_Ax[1]の領域が更新されるようになります。



モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令同士の多重起動

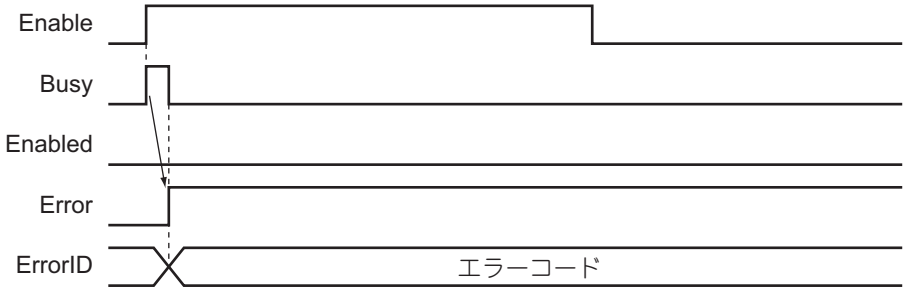
MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令を実行中の軸に、別インスタンスの MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令を起動した場合、両命令とも実行します。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。

「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



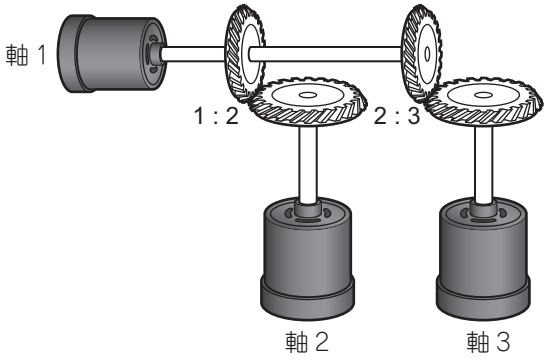
● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

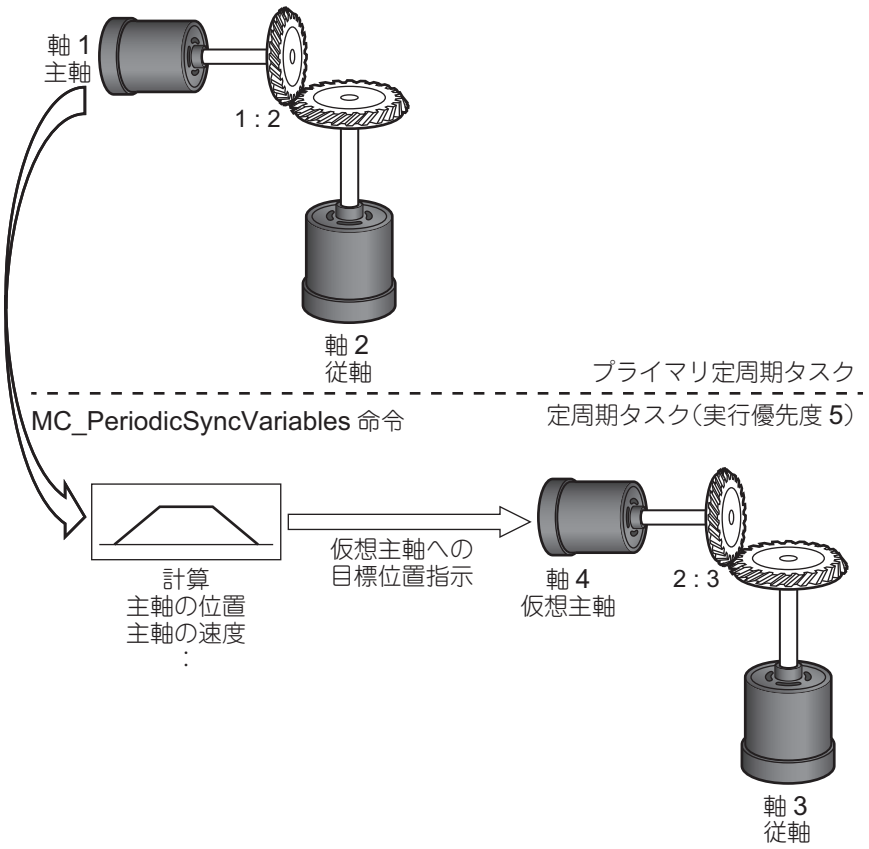
サンプルプログラム

構成として、主軸（軸 1）に対して、高速で高精度が必要な従軸（軸 2）をプライマリ定周期タスクに、精度を必要としない従軸（軸 3）を定周期タスク（実行優先度 5）のタスクに割り付けて制御します。なお、主軸（軸 1）は、プライマリ定周期タスクに割り付けます。

● 物理的な軸構成



● **論理的な軸構成**



プライマリ定周期タスクと定周期タスク（実行優先度 5）にそれぞれプログラムを組むことで、上記のアプリケーションの動作を実現します。

プライマリ定周期タスクのプログラム

- プライマリ定周期タスクで、プライマリ定周期タスクに割り付けた主軸(Primary_MasterAxis)に対し、MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令を実行します。
- MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令の実行により、定周期タスク（実行優先度 5）に対して主軸の軸変数を周期ごとに更新します。
- プログラムで使用する主な変数は、下記を使用します。

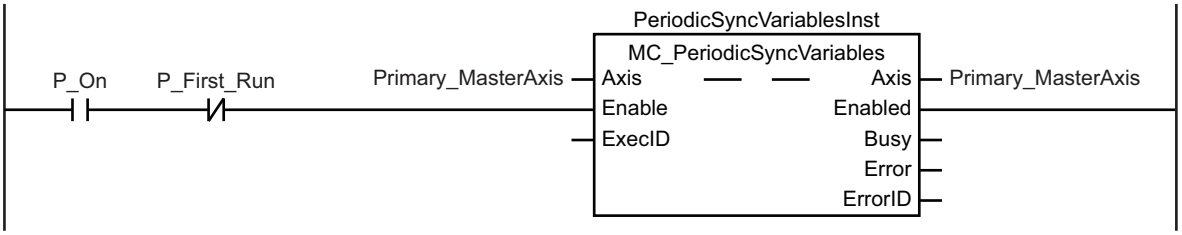
名称	データ型	コメント
Primary_MasterAxis	_sAXIS_REF	主軸（軸 1）。軸変数は、_MC1_AX[0]とする。 プライマリ定周期タスクに割り付け *1

*1. 主軸は、必ずプライマリ定周期タスクに割り付けてください。

● **ラダー図**

プライマリ定周期タスクのラダー図プログラムは、次のようになります。

プライマリ定周期タスクで、MC_PeriodicSyncVariables（軸変数定周期同期）命令を実行します。



定周期タスク（実行優先度 5）のプログラム

- MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令により、定周期タスク（実行優先度 5）のプログラムで、周期ごとに更新される主軸(Primary_MasterAxis)の軸変数が参照可能となります。
- 主軸の情報をもとに、定周期タスク（実行優先度 5）に割り付けた仮想主軸(Secondary_VirtualMasterAxis)に対し、MC_SyncMoveAbsolute（サイクリック同期絶対位置決め）命令を実行し主軸に追従させます。
- 従軸(Secondary_SlaveAxis)に対して、仮想主軸(Secondary_VirtualMasterAxis)を主軸指定します。
- MC_GearIn（ギア動作開始）命令などの同期制御命令を実行することにより、異なるタスク周期に割り付けている軸間の同期制御機能が実現できます。
- プログラムで使用する主な変数は、下記を使用します。

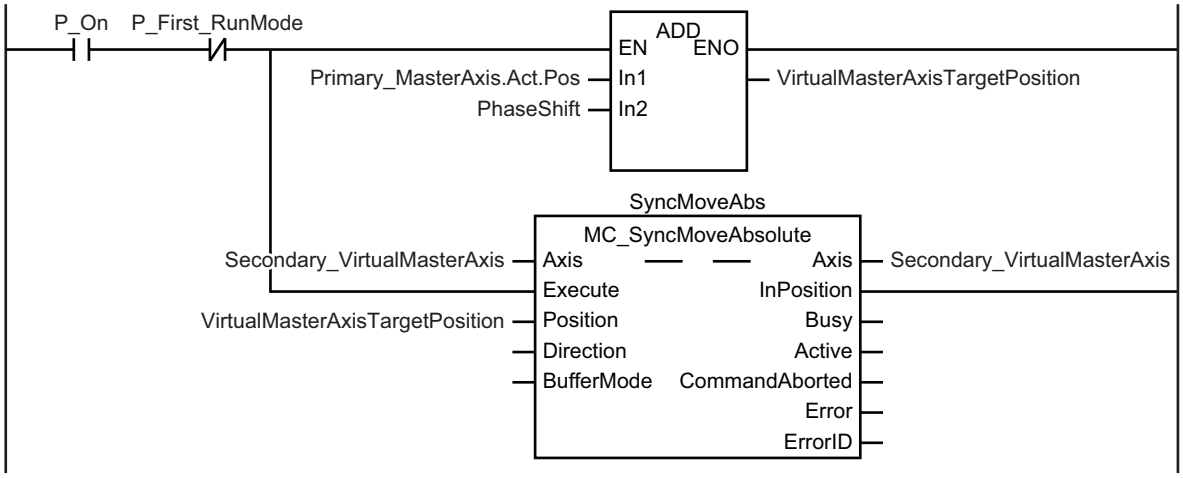
名称	データ型	コメント
Secondary_VirtualMasterAxis	_sAXIS_REF	仮想主軸。定周期タスク（実行優先度 5）に割り付け
Secondary_SlaveAxis	_sAXIS_REF	従軸（軸 3)。定周期タスク（実行優先度 5）に割り付け
Primary_MasterAxis *1	_sAXIS_REF	主軸。軸変数は、_MC2_AX[4]とする。MC_PeriodicSyncVariables(軸変数タスク間定周期同期）命令により、周期ごとに同期される。
PhaseShift	LREAL	位相補正量
VirtualMasterAxisTargetPosition	LREAL	仮想主軸のタスク周期ごとの目標位置
SyncExecute	BOOL	従軸の同期制御開始信号

*1. このサンプルでは、プライマリ定周期タスクの主軸と同一名の内部変数「Primary_MasterAxis」を定義し、使用しています。

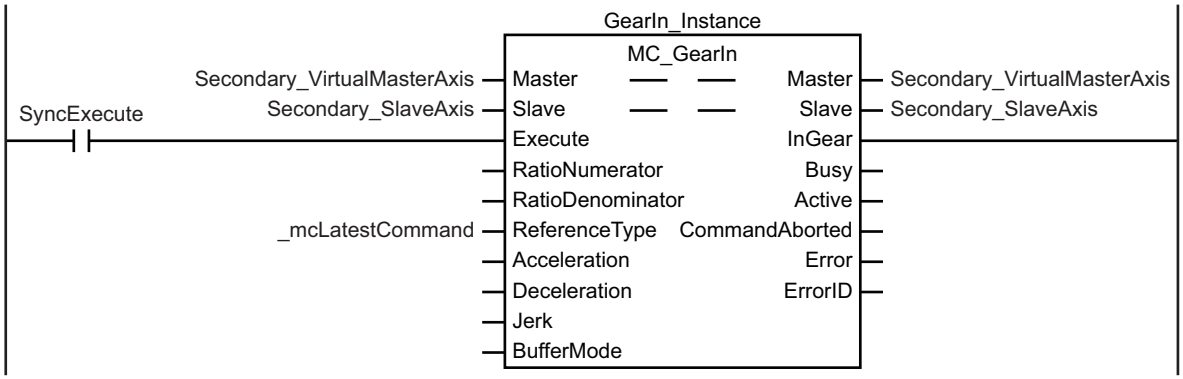
● ラダー図

定周期タスク（実行優先度 5）のラダー図プログラムは、次のようになります。

Primary_MasterAxis の現在位置と位相補正量 (PhaseShift) を加算し、Secondary_VirtualMasterAxis の指令位置 (VirtualMasterAxisTargetPosition) とします。
MC_SyncMoveAbsolute（サイクリック同期絶対位置決め）命令を使って、常時 Secondary_VirtualMasterAxis を Primary_MasterAxis に追従させておきます。

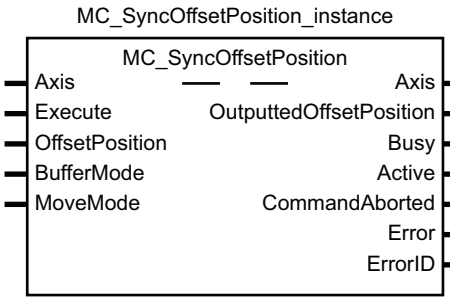


Secondary_VirtualMasterAxis を仮想主軸にして、Secondary_SlaveAxis に同期制御命令を実行します。



MC_SyncOffsetPosition

同期制御中の従軸に対して、指定した位置オフセットを周期的に指令現在位置に加算して出力します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_SyncOffsetPosition	サイクリック同期位置オフセット補正	FB		MC_SyncOffsetPosition_instance (Axis :=「パラメータ」, Execute :=「パラメータ」, OffsetPosition :=「パラメータ」, BufferMode :=「パラメータ」, MoveMode :=「パラメータ」, OutputtedOffsetPosition => 「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

バージョン関連情報

本命令は、ユニットバージョン Ver.1.10 以降の CPU ユニットと Sysmac Studio Ver.1.12 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
OffsetPosition	位置オフセット	LREAL	負数、正数、「0」	0	指令現在位置に加算する位置オフセットを指定します。 単位は［指令単位］です。*1
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *2	モーション命令多重起動時の動作を指定します。 0: アボーティング
MoveMode	移動方法選択	_eMC_MOVE_MODE	1: _mcRelative	0 *2	移動方法を選択します。 1: 相対値位置決め

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
OutputtedOffsetPosition	位置オフセット出力値	LREAL	負数、正数、「0」	指令現在位置に加算された位置オフセットを出力します。 「Active」が TRUE のときに値を更新します。 「CommandAborted」または「Error」が TRUE のときは、更新を停止し値を保持します。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Busy	「Execute」の立ち上がり	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• 実行中の同期制御命令が停止したとき。 • ほかの命令によるモーション命令多重起動で、本命令が中止されたとき。 • ほかの命令で異常が発生して、本命令が中止されたとき。 • 軸異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

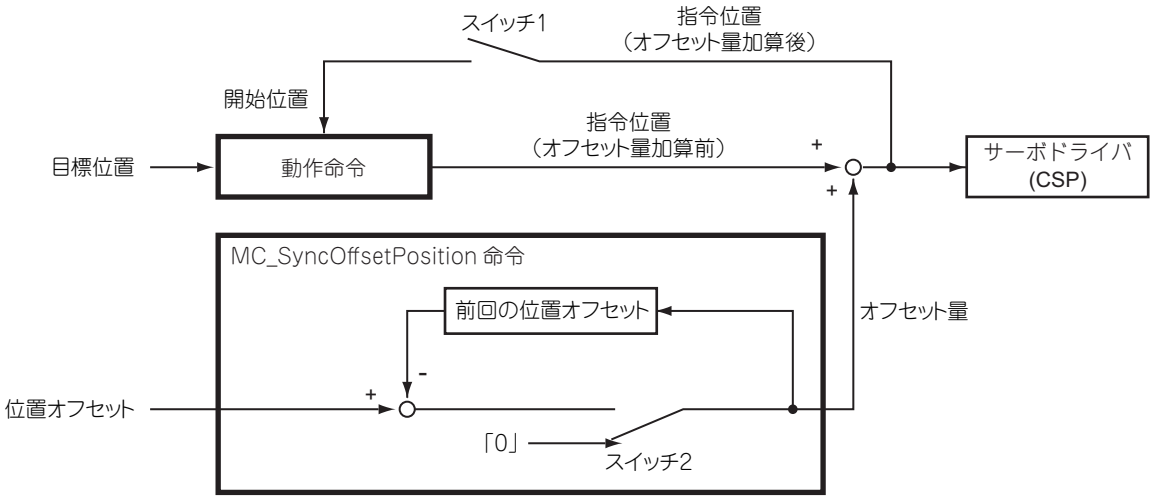
入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1 *2

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

*2. NX シリーズエンコーダ入力ユニットを割り付けたエンコーダ軸、またはオムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプの軸を指定してください。

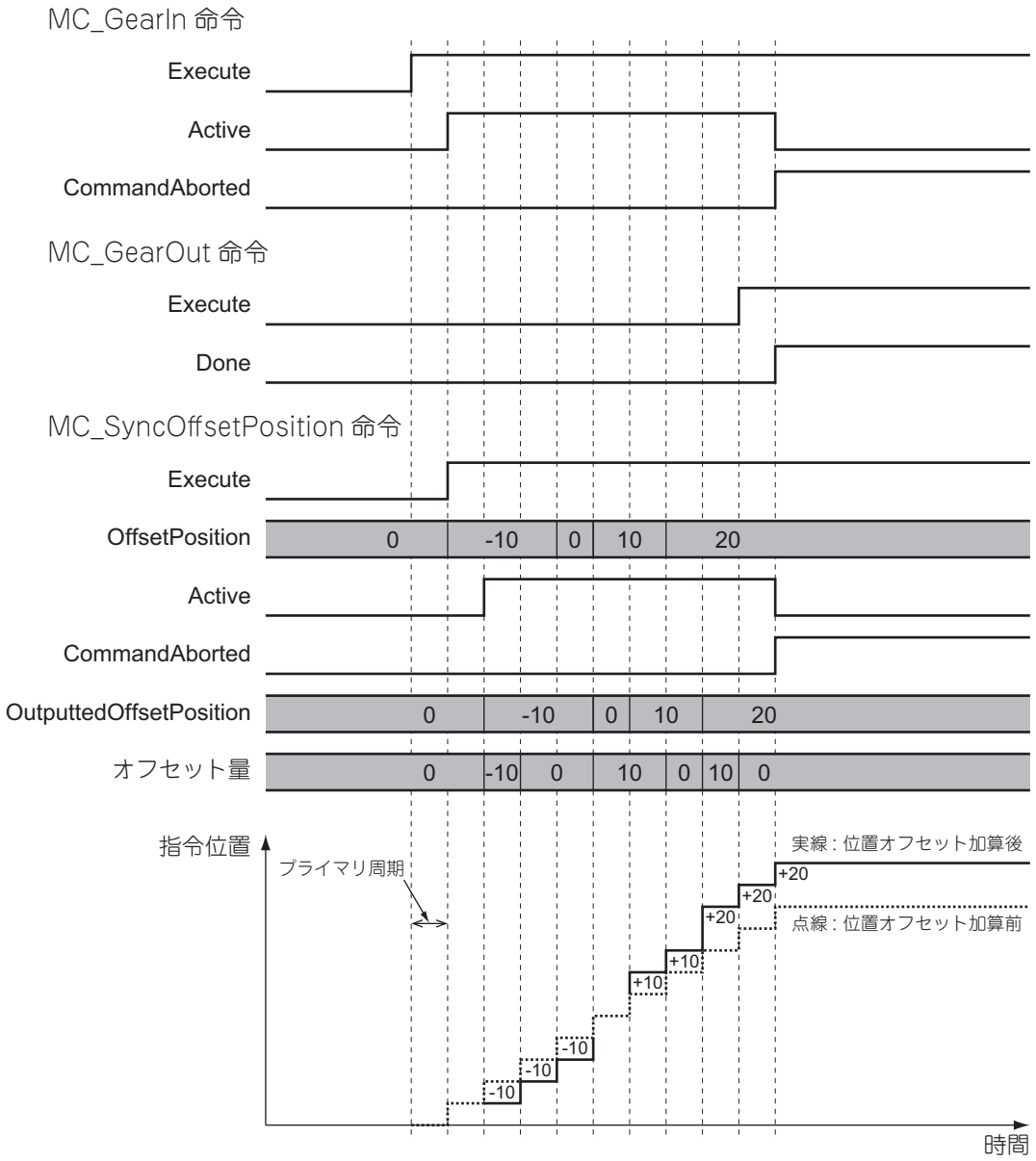
機能説明

- 本命令は、同期制御中の従軸に対して、入力変数の「OffsetPosition」（位置オフセット）の値をもとに演算したオフセット量を指令現在位置に加算してサーボドライバに出力します。
- 出力変数の「Active」（制御中）が TRUE の間、入力変数の「OffsetPosition」（位置オフセット）の値を変更できます。
- 本命令の「Active」（制御中）が TRUE の間、「OffsetPosition」（位置オフセット）は、本命令の起動時を起点として扱います。詳細は、「本命令の起動と停止を繰り返し実行する場合 (P.3-436)」を参照してください。



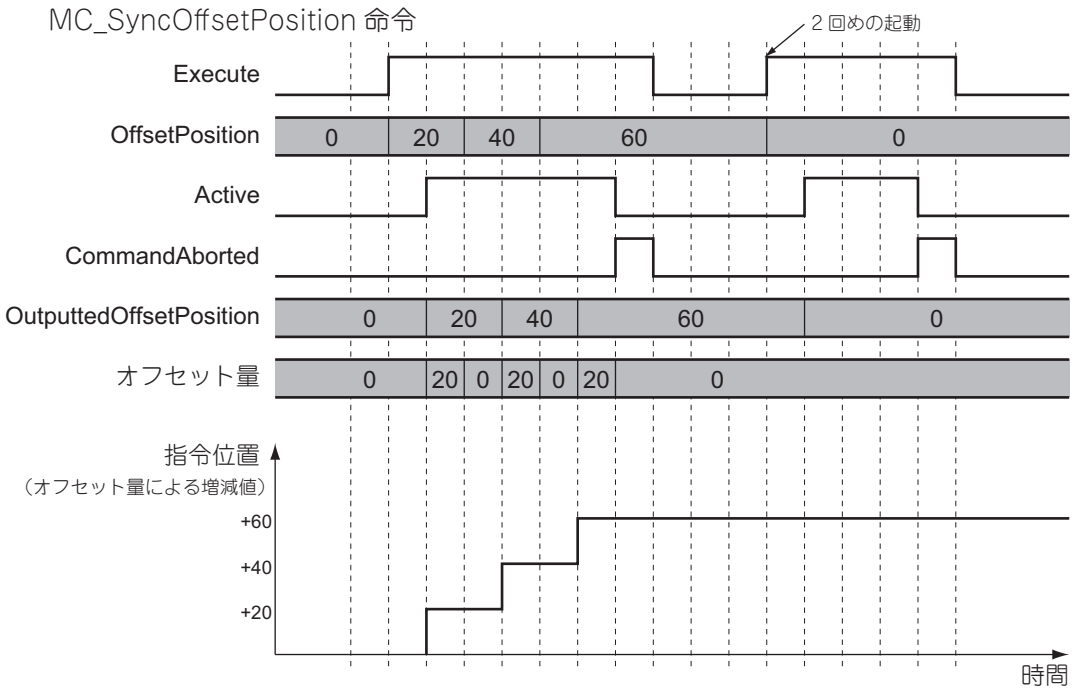
a）スイッチ 1 は、動作命令の「Execute」（起動）の立ち上がり時に一度だけ ON します。
b）スイッチ 2 は、MC_SyncOffsetPosition 命令の「Active」（制御中）が TRUE で ON します。
c）スイッチ 2 が OFF のとき、指令位置に加算されるオフセット量は「0」となります。

- 本命令は、同期制御中の従軸に対してだけ実行可能です。
- 本命令の起動後に、対象の同期制御命令の「Done」（完了）、「CommandAborted」（実行中断）、あるいは「Error」（エラー）が TRUE になると、本命令の「CommandAborted」（実行中断）が TRUE となります。
- 本命令の「CommandAborted」（実行中断）、または「Error」（エラー）が TRUE になった場合、指令位置は直前に加算したオフセット量を保持します。
- プライマリ定周期タスクでの MC_GearIn（ギア動作開始）命令の動作中に本命令を実行し、その後 MC_GearIn（ギア動作開始）命令の「CommandAborted」（実行中断）が TRUE となった場合のタイミングチャートを次に示します。

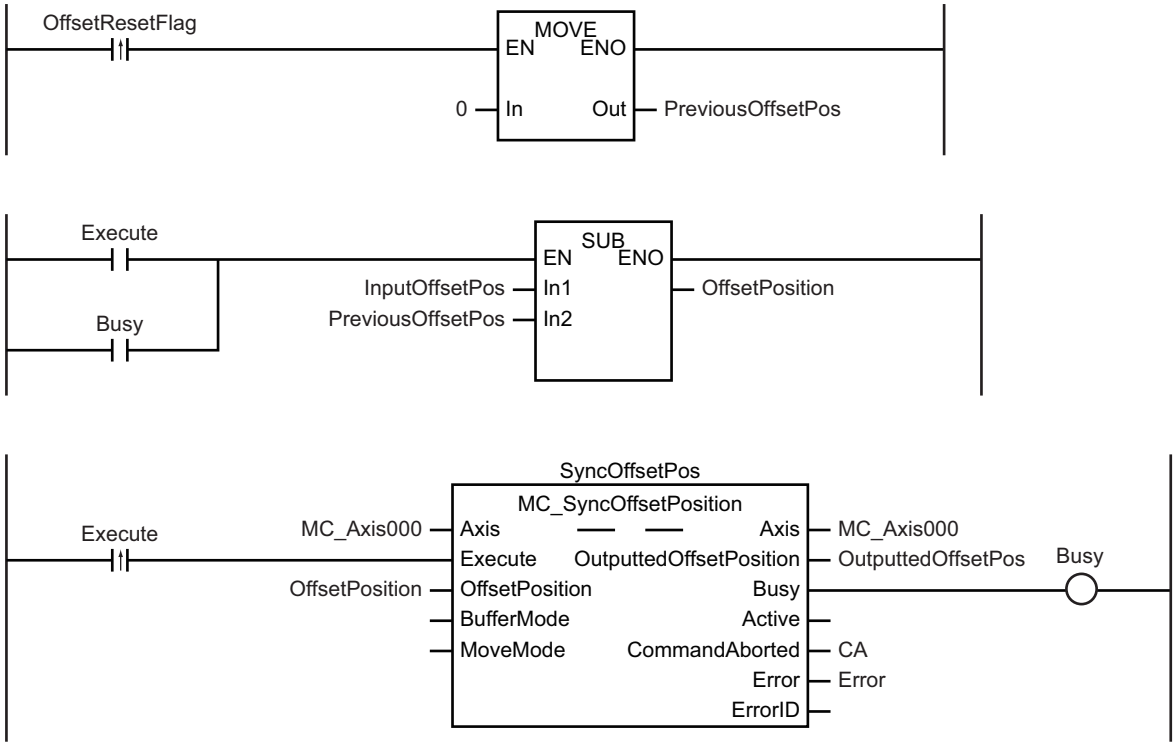


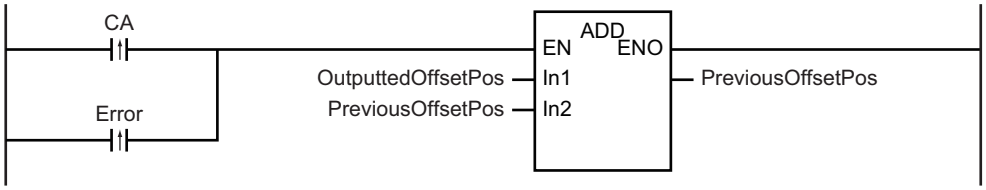
● 本命令の起動と停止を繰り返し実行する場合

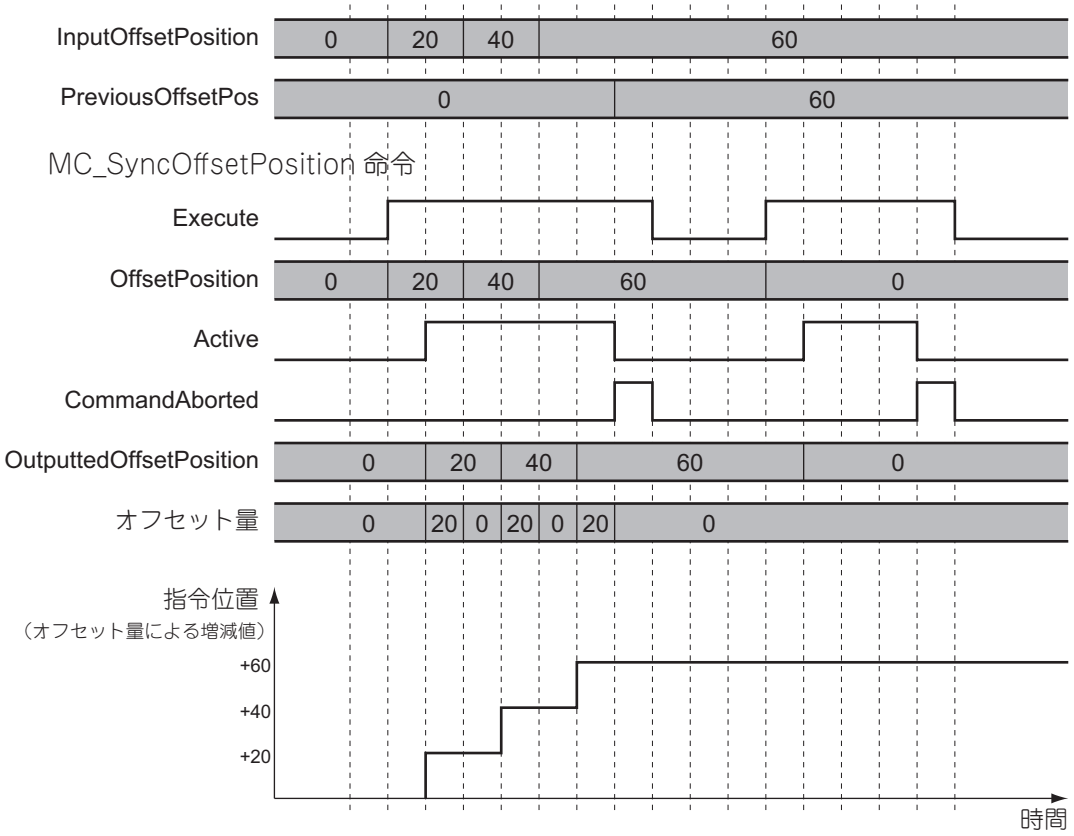
本命令の実行中断後に再度実行すると、入力変数の「OffsetPosition」（位置オフセット）は「0」が基点となります。2 回めの起動時に「OffsetPosition」（位置オフセット）に「0」を指定すると、指令位置に加算されるオフセット量は「0」となります。



本命令の実行中断後に再度実行する場合に、前回の命令と同じ基点で指令したいときは、出力変数の「OutputtedOffsetPosition」（位置オフセット出力値）を利用します。
次のサンプルプログラムのように、「OutputtedOffsetPosition」（位置オフセット出力値）で実行中断時の位置オフセットを参照し、本命令に指定する入力パラメータの変数から減算します。







● 実行可能な軸の状態

本命令は、「Axis」（軸）に指定した軸の状態が、「Status.Synchronized」（同期動作中）が TRUE のときにだけ実行できます。ほかの状態で実行すると異常となります。

● 指令位置の扱い

軸の指令位置は、位置オフセットが加算された後の値で管理されています。そのため、位置オフセットに設定する値によっては、次の項目に関連する異常が発生する場合があります。

- 軸パラメータの動作設定
  - a）最高速度
  - b）最大加速度
  - c）最大減速度
  - d）速度警告値
  - e）加速度警告値
  - f）減速度警告値
  - g）インポジションチェック時間
- 軸パラメータのリミット設定
  - a）ソフトウェアリミット機能
  - b）位置偏差オーバー値
  - c）位置偏差警告値

- 指令位置オーバーフロー
- 指令位置アンダーフロー

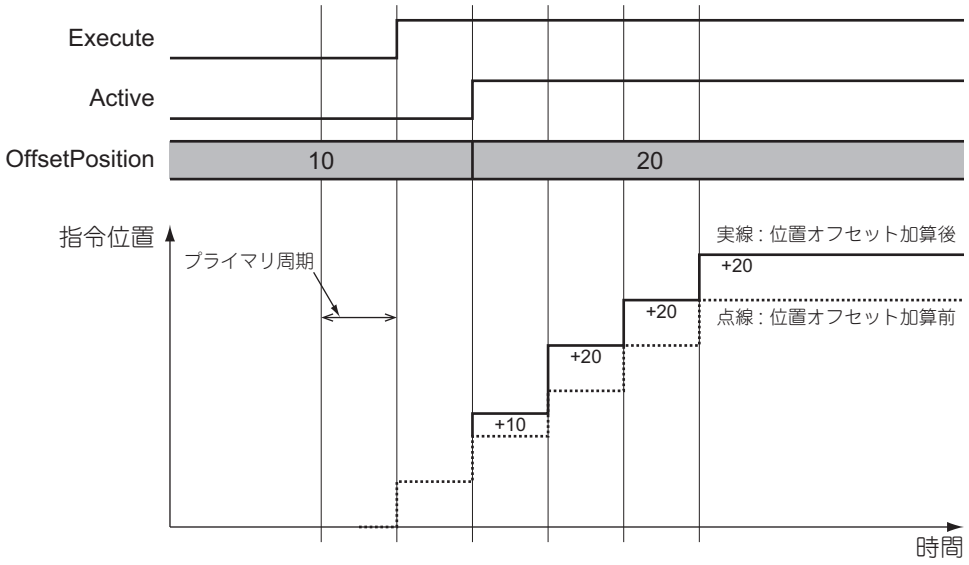
タイミングチャート

本命令を実行した場合の位置オフセットの反映タイミングを次に示します。

● プライマリ定周期タスク、または定周期タスク（実行優先度 5）に本命令を記述した場合

入力で指定した位置オフセットは次のタスク周期でサーボドライバに出力されます。
プライマリ定周期タスクに本命令を記述した例を次に示します。
定周期タスク（実行優先度 5）の場合も同様となります。

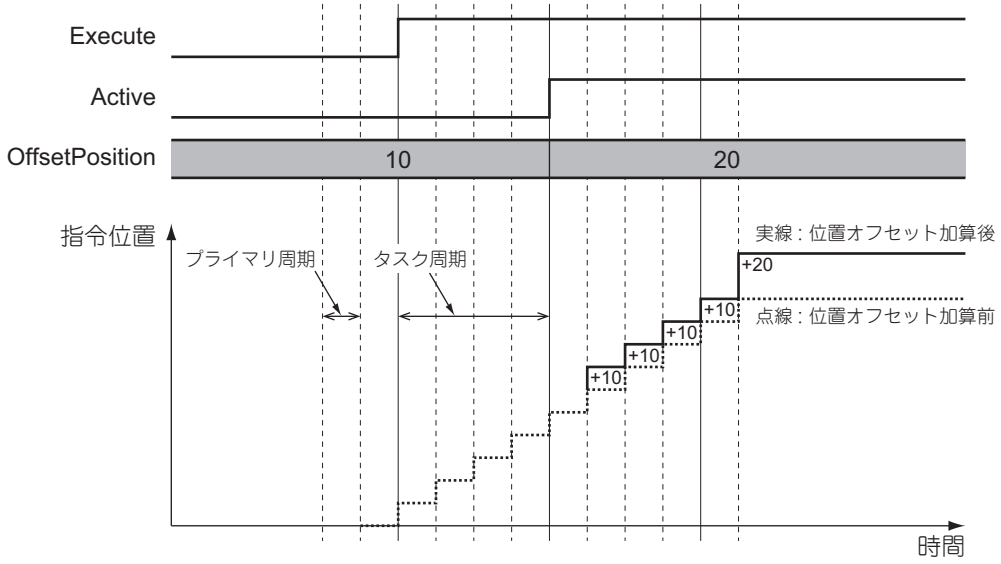
MC_SyncOffsetPosition 命令



● 定周期タスク（実行優先度 16）に本命令を記述した場合

入力で指定した位置オフセットは次の定周期タスク（実行優先度 16）+プライマリ周期の 1 周期後のタイミングでサーボドライバに出力されます。

MC_SyncOffsetPosition 命令



モーション命令再起動

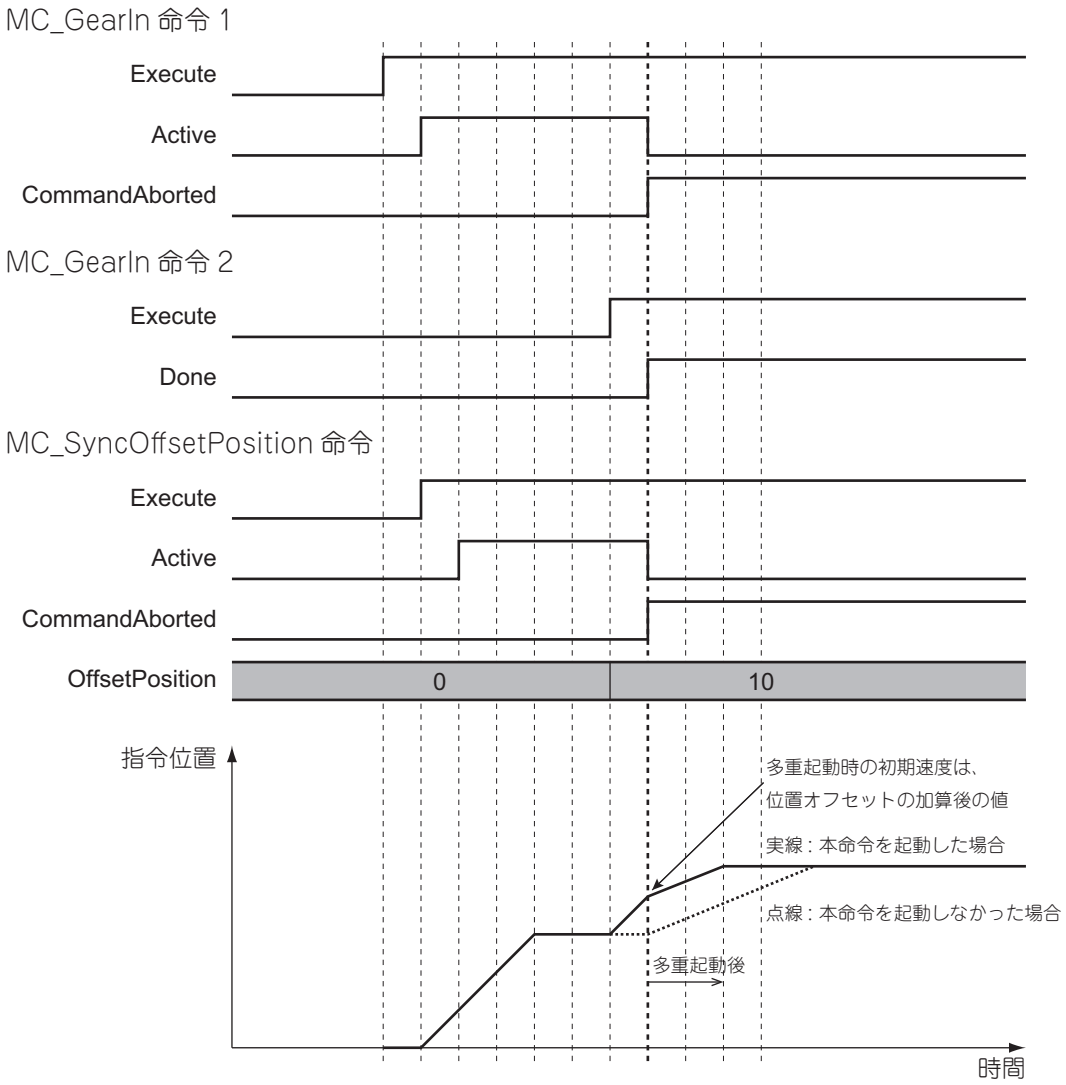
本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

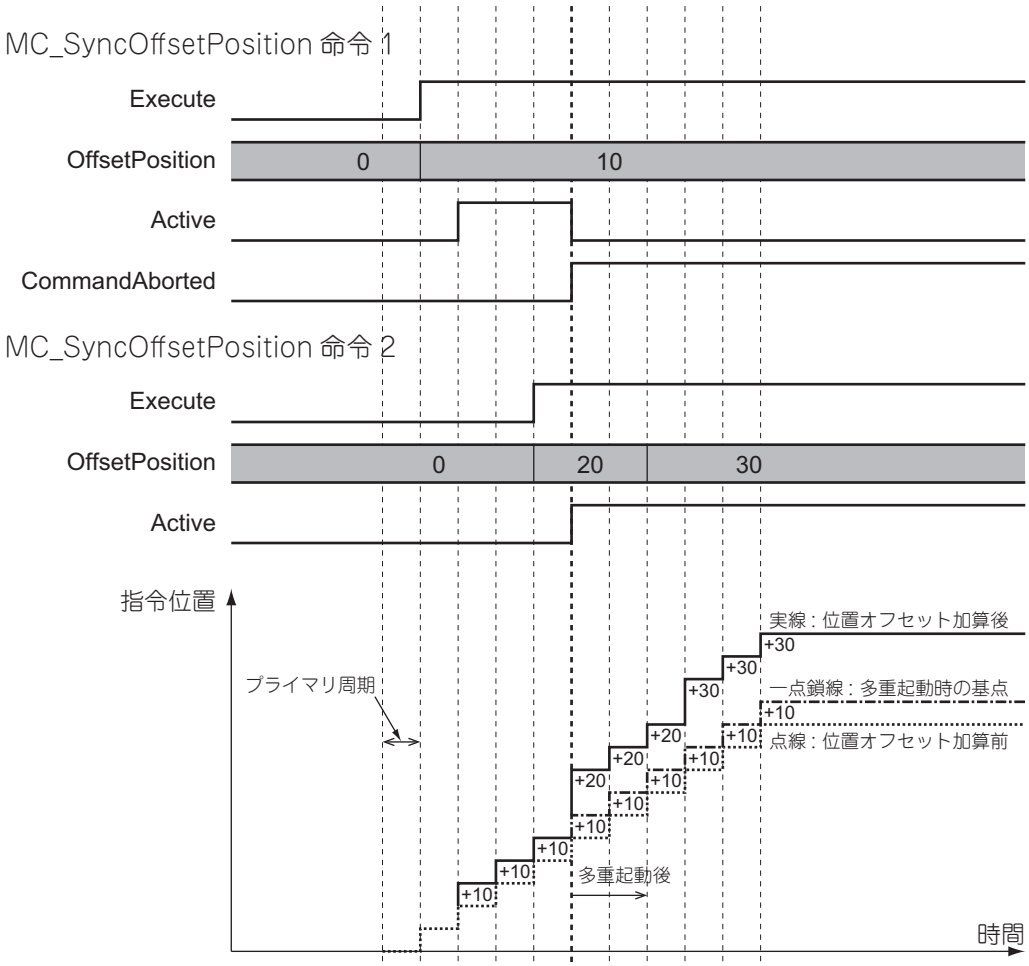
● 対象の同期制御命令に対する多重起動

本命令の対象の同期制御命令に多重起動を行うと、本命令の「CommandAborted」（実行中断）が TRUE となります。ただし、MC_CamIn（カム動作開始）命令同士のブレンディングでの多重起動の場合、本命令の「CommandAborted」（実行中断）は FALSE のままで処理を継続します。このときの初期速度は、「OffsetPosition」（位置オフセット）が加算された値となります。
対象の同期制御命令が MC_GearIn（ギア動作開始）命令で本命令を実行中に、別インスタンスの MC_GearIn（ギア動作開始）命令を多重起動した場合のタイミングチャートを次に示します。



● 本命令に対する多重起動

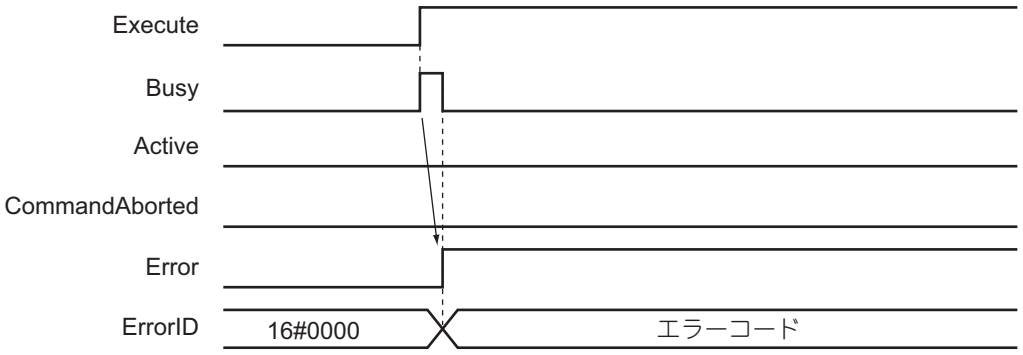
本命令同士で多重起動する場合、実行中のインスタンスの「CommandAborted」（実行中断）は TRUE となり、後から実行したインスタンスが実行されます。後から実行したインスタンスの基点は、先に実行していたインスタンスが中断される 1 周期前に出力していた位置オフセット加算後の位置となります。本命令の多重起動により、実行中断となる他の命令はありません。
本命令同士の多重起動時のタイミングチャートを次に示します。



異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_OffsetPosition

同期制御中の従軸に対して、指定した位置オフセットを指令現在位置に加減速カーブをつけて加算して出力します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_OffsetPosition	位置オフセット補正	FB	MC_OffsetPosition_instance MC_OffsetPosition Axis — — Axis Execute — Done OffsetPosition — OutputtedOffsetPosition Velocity — Busy Acceleration — Active Deceleration — CommandAborted Jerk — Error BufferMode — ErrorID MoveMode	MC_OffsetPosition_instance(Axis :=「パラメータ」, Execute :=「パラメータ」, OffsetPosition :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, BufferMode :=「パラメータ」, MoveMode :=「パラメータ」, Done =>「パラメータ」, OutputtedOffsetPosition => 「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

バージョン関連情報

本命令は、以下に示すユニットバージョンの CPU ユニットと Sysmac Studio Ver.1.28 以降の組み合わせで使用可能です。

CPU ユニット	使用可能なユニットバージョン
形 NX102 CPU ユニット	Ver.1.32 以降
形 NX1P2 CPU ユニット	Ver.1.21 以降
NJ シリーズ CPU ユニット	Ver.1.21 以降

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
OffsetPosition	位置オフセット	LREAL	負数、正数、「0」	0	指令現在位置に対して加える位置オフセットを指定します。 単位は、［指令単位］となります。*1
Velocity	目標速度	LREAL	正数	0	目標速度を指定します。*2 単位は、［指令単位／s］となります。*1

入力変数	名称	データ型	有効範囲	初期値	内容
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は、[指令単位／s^2]となります。*1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は、[指令単位／s^2]となります。*1
Jerk (Reserved)	ジャーク	LREAL	0	0	ジャークを指定します。 単位は、[指令単位／s^3]となります。*1
BufferMode (Reserved)	バッファーモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *3	モーション命令多重起動時の動作を指定します。 0: アボーティング
MoveMode	移動方法選択	_eMC_MOVE_MODE	1: _mcRelative	1 *3	移動方法を選択します。 1: 相対位置決め

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 目標速度は必ず設定してください。設定せずに動作させると異常が発生します。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
OutputtedOffsetPosition	位置オフセット出力値	LREAL	負数、正数、「0」	指令現在位置に加算された位置オフセットを出力します。 「Active」が TRUE のときに値を更新します。 「CommandAborted」または「Error」が TRUE のときは、更新を停止し値を保持します。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位置決めが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
Active	命令を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• 実行中の同期制御命令が停止したとき。 • ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_Stop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

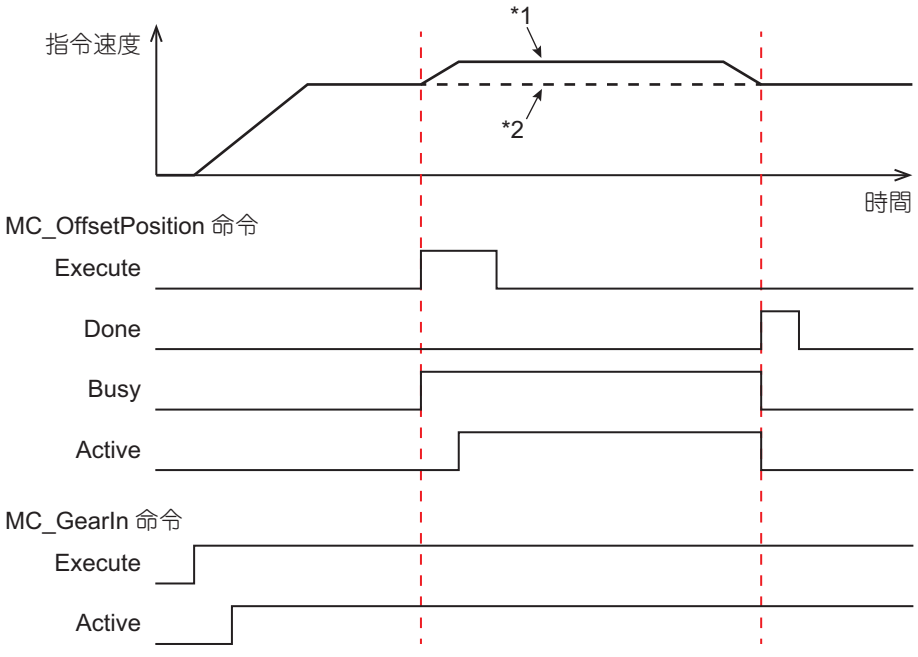
入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	軸を指定します。*1

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

同期制御中の従軸に対して、指定した「OffsetPosition」（位置オフセット）、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）を基に、オフセット量を指令現在位置に加算してサーボドライバに出力します。

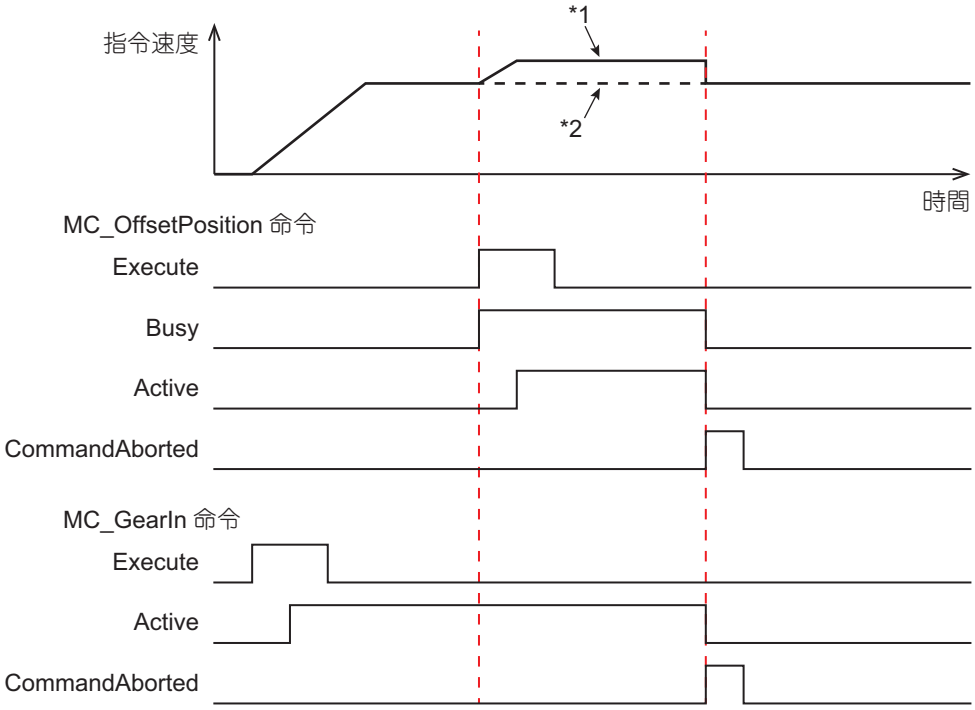
下記に、例として MC_GearIn（ギア動作開始）命令動作中に本命令を実行し、本命令が完了するまでのタイミングチャートを示します。



*1. MC_OffsetPosition 命令を起動した場合。
*2. MC_OffsetPosition 命令を起動しない場合。

「OffsetPosition」（位置オフセット）に到達すると、「Done」（完了）が TRUE になります。
本命令のインポジションチェックは行いません。
現在実行中の同期制御命令が完了した時点で、本命令は「CommandAborted」（実行中断）となります。

下記に、例として MC_GearIn（ギア動作開始）命令動作中に本命令を実行し、その後 MC_GearIn（ギア動作開始）命令が「CommandAborted」（実行中断）され、本命令が停止するまでのタイミングチャートを示します。



実行可能な軸の状態

本命令は、「Axis」（軸）に指定した軸の状態が、「Status.Synchronized」（同期動作中）が TRUE のときにだけ実行できます。ほかの状態で実行すると異常となります。

軸の制限機能

軸の制限機能は、位置オフセットが加算された指令位置に対して機能します。
各制限機能の仕様は、動作中の同期制御命令の仕様によって異なります。
以下に関連する軸の制限値を示します。

- 動作設定値
  a）最高速度
  b）最大加速度／最大減速度
  c）速度警告値
  d）加速度警告値／減速度警告値
- リミット設定値
  a）位置偏差オーバー値
  b）位置偏差警告値

c）ソフトウェアリミット値
- 指令位置オーバーフロー／指令位置アンダーフロー
- インポジションチェック

モーション命令再起動

位置決め動作中に入力変数を変更し、「Execute」（起動）を再度 TRUE にすることによって、本命令の動作を変更することができます。
モーション命令再起動で変更可能な入力変数は、「OffsetPosition」（位置オフセット）、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）です。

モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

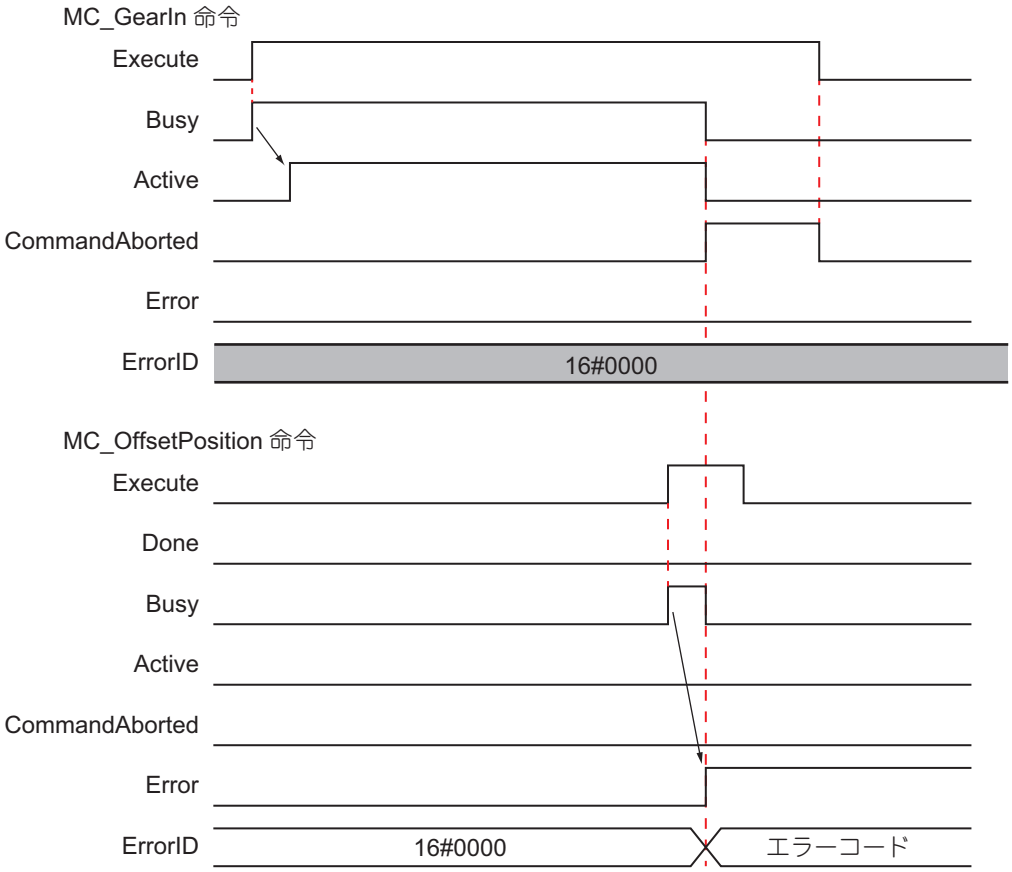
モーション命令多重起動

本命令の多重起動はできません。
多重起動を行った場合、「モーション命令多重起動不可（エラーコード: 543CHex）」となります。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

4

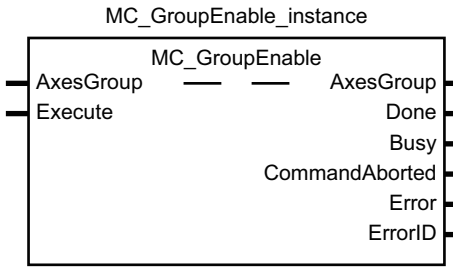
軸グループ指令の命令

MC 機能モジュールが多軸協調制御を行うための命令について説明します。

MC_GroupEnable....4-2
MC_GroupDisable....4-6
MC_MoveLinear....4-11
MC_MoveLinearAbsolute....4-38
MC_MoveLinearRelative....4-41
MC_MoveCircular2D....4-44
MC_GroupStop....4-71
MC_GroupImmediateStop....4-79
MC_GroupSetOverride....4-83
MC_GroupReadPosition....4-87
MC_ChangeAxesInGroup....4-91
MC_GroupSyncMoveAbsolute....4-95
MC_GroupReset....4-101

MC_GroupEnable

軸グループを有効にします。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GroupEnable	軸グループ有効	FB		MC_GroupEnable_instance (AxesGroup :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後

変数	TRUE となるタイミング	FALSE となるタイミング
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	本命令が中止されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	−	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

本命令の機能を説明する前に、軸グループの状態について説明します。

- 軸グループは、「GroupEnable」（グループ有効）と「GroupDisable」（グループ無効）との 2 つの状態を持ちます。

状態	説明
GroupEnable（グループ有効）	多軸協調制御が有効な状態です。 MC_ChangeAxesInGroup（軸グループ構成軸書込）命令を除くすべての多軸協調命令を実行可能です。
GroupDisable(グループ無効)	多軸協調制御が無効な状態です。 多軸協調制御命令は、以下の命令だけ実行可能です。 • MC_GroupEnable（軸グループ有効）命令 • MC_GroupDisable（軸グループ無効）命令 • MC_GroupReset（軸グループエラーリセット）命令 • MC_GroupSetOverride（軸グループオーバーライド値設定）命令 • MC_GroupReadPosition（軸グループ位置取得）命令 • MC_ChangeAxesInGroup（軸グループ構成軸書込）命令

- 多軸協調制御を行うには、軸グループの状態が「GroupEnable」（グループ有効）になっている必要があります。

モーション制御システム変数の軸グループ変数をモニタすることにより、軸グループの有効、無効を確認することができます。
ここでは、_MC_GRP[*] を例に軸グループ変数を説明しています。
_MC1_GRP[*] 、_MC2_GRP[*] についても同様です。

変数名	名称	説明
_MC_GRP*.Status.Ready	軸グループ有効	軸グループが停止中で、かつ軸グループ命令による起動準備が整っているときに TRUE となります。
_MC_GRP*.Status.Disabled	軸グループ無効	軸グループ無効で停止しているときに TRUE となります。

（注）「_MC_GRP*.」の「*」には、形 NX701 CPU ユニットの場合は「0～63」、形 NX102 および形 NX1P2 CPU ユニットの場合は「0～7」、形 NX502 および NJ シリーズ CPU ユニットの場合は「0～31」の数字が入ります。

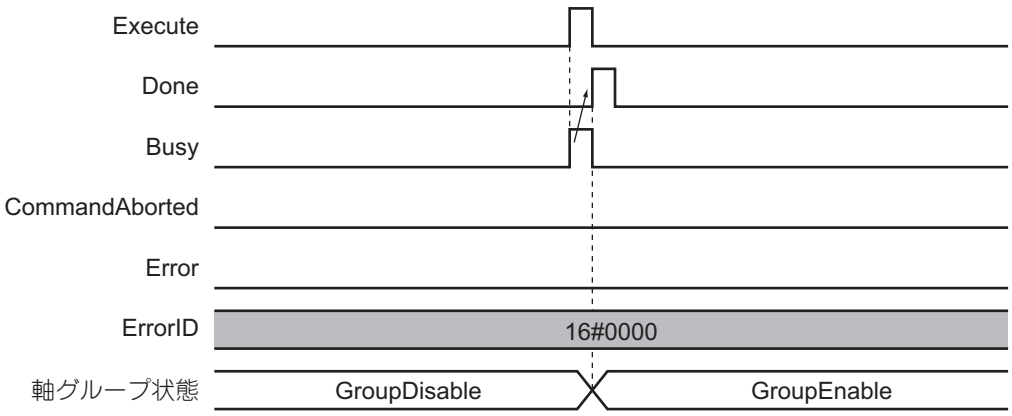
基本機能

- MC_GroupEnable（軸グループ有効）命令は、「AxesGroup」（軸グループ）で指定した軸グループの状態を「GroupEnable」（グループ有効）にします。
- 「GroupEnable」（グループ有効）の状態にすることにより、軸グループはすべての多軸協調命令を実行できるようになります。
- 軸グループに指定できる軸種別は、［サーボ軸］、［仮想サーボ軸］だけで、ほかの軸種別を含む場合は異常となります。
- 軸グループに属するすべての軸が停止中であることが必要です。
  軸が停止中かどうかは、軸変数の「Status.Disabled」（軸無効）、または「Status.Standstill」（停止中）が TRUE となっていれば停止中であることがわかります。
- 既にほかの軸グループに属し、軸グループ有効とされている軸がある場合は、MC_GroupEnable（軸グループ有効）命令は実行できず、異常となります。
- 軸グループが有効になると、軸グループに属している軸の状態は、「多軸協調動作中」となります。
  軸変数の「Status.Coordinated」（多軸協調動作中）が TRUE となります。
- 軸グループが無効になる条件は、MC_GroupDisable（軸グループ無効）命令の実行と、プログラムモード移行による運転停止、および MC 試運転開始時です。

使用上の注意

- 軸グループを使用するには、Sysmac Studio で軸グループを作成したうえで、CPU ユニットに設定をダウンロードしておく必要があります。プログラム中で軸グループの軸を変更することはできません。
  ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせでは、MC_ChangeAxesInGroup（軸グループ構成軸書込）命令を使用して、一時的に変更することができます。
- 「ダウンロード」は、Sysmac Studio の「同期」機能を使用します。

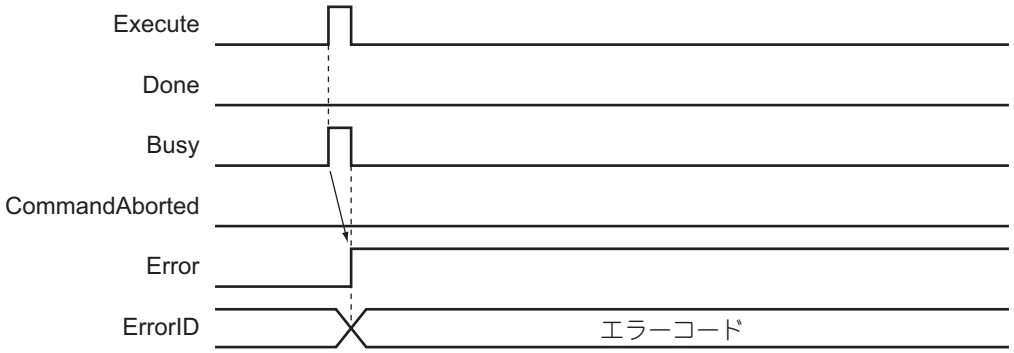
タイミングチャート



異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。

「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。



参考

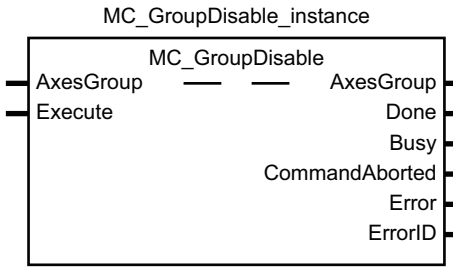
軸グループで異常が発生しても、グループに含まれる各軸は異常にはなりません。

● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_GroupDisable

軸グループを無効にします。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GroupDisable	軸グループ無効	FB		MC_GroupDisable_instance (AxesGroup :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	軸グループの状態が「GroupDisable」に変更されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後

変数	TRUE となるタイミング	FALSE となるタイミング
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	TRUE となる条件はありません。(Reserved)	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

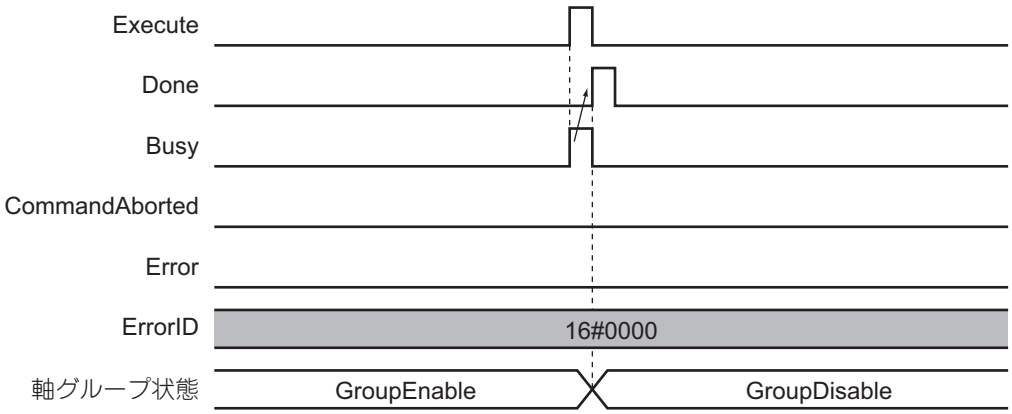
入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	―	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

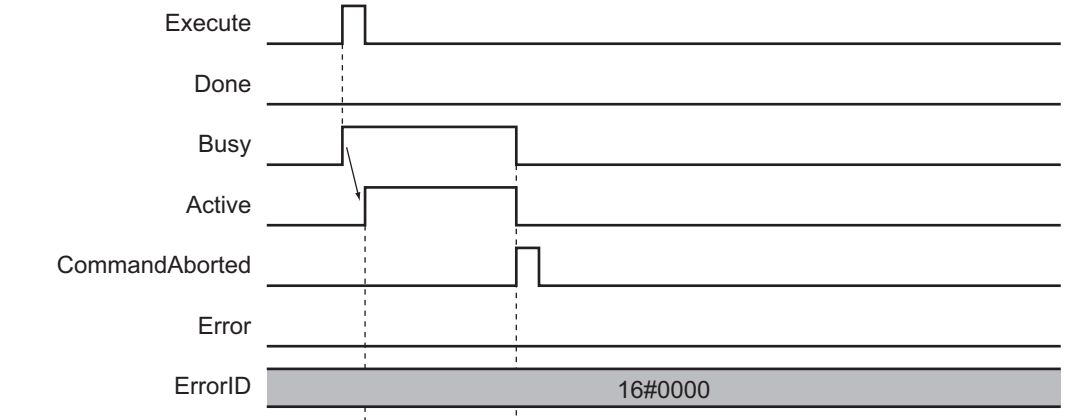
- 本命令は、軸グループ状態を無効にします。
  軸グループの状態説明については、「機能説明 (P.4-3)」を参照してください。
- MC_GroupDisable（軸グループ無効）命令は、「AxesGroup」（軸グループ）で指定した軸グループの状態を「GroupDisable」（グループ無効）にします。
  「GroupDisable」（グループ無効）の状態にすることにより、軸グループは多軸協調命令を受け付けなくなります。
- 軸グループの状態を「GroupDisable」（グループ無効）にした場合、指定した「AxesGroup」（軸グループ）のバッファリング命令はクリアされます。
- 軸グループは、運転停止時にも無効となります。
  軸グループが無効になると、軸グループに属している軸の状態は、「Status.Coordinated」（多軸協調動作中）が TRUE になっている状態から、各軸の状態に変化します。
  各軸の状態を知るには、軸変数の「Status」（軸ステータス）を確認してください。

タイミングチャート

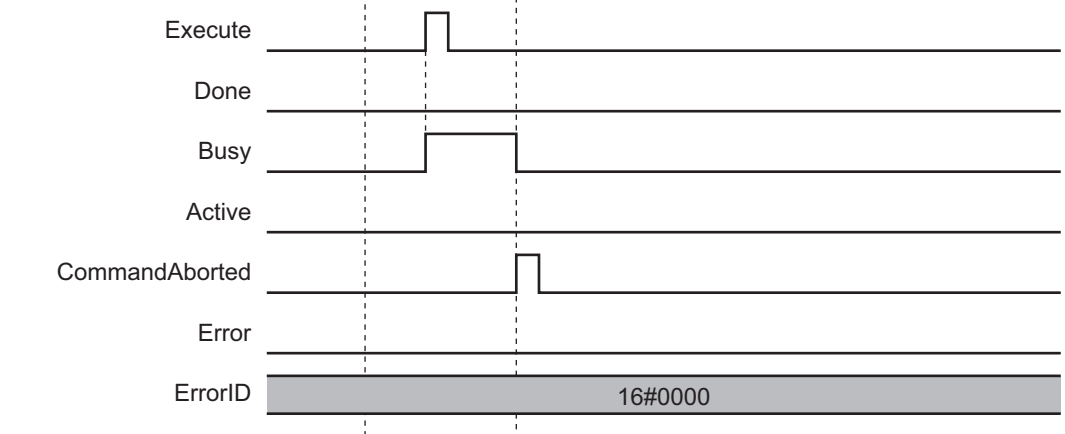


多軸協調制御中の「AxesGroup」（軸グループ）に MC_GroupDisable（軸グループ無効）命令を実行した場合は、多軸協調制御は、「CommandAborted」（実行中断）されます。
この場合、バッファリングされている多軸協調制御命令も同様に「CommandAborted」（実行中断）されます。
動作中の軸は各軸の最大減速度で減速停止します。

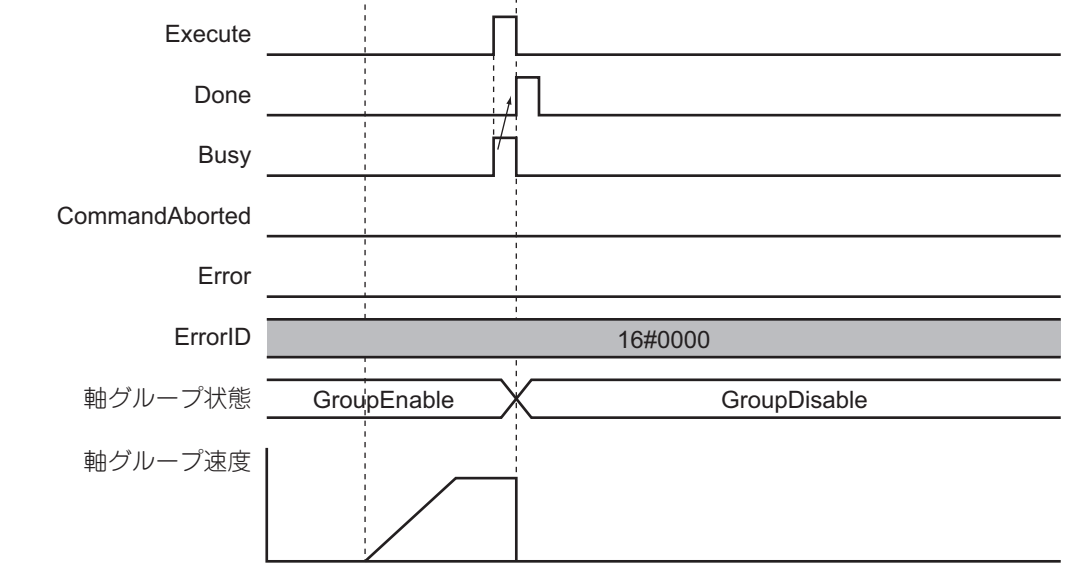
多軸協調制御 1（Ex. MC_MoveLinearAbsolute）



多軸協調制御 2（Ex. MC_MoveLinearAbsolute.Bufferd）

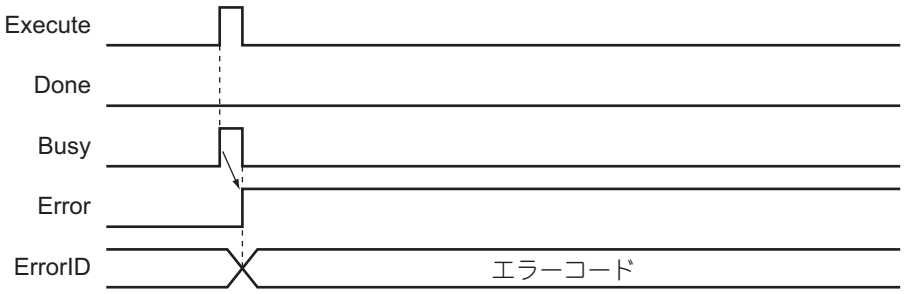


MC_GroupDisable



異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

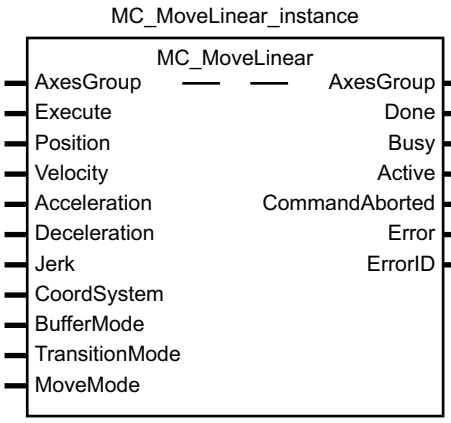


● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_MoveLinear

直線補間を行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveLinear	直線補間	FB		MC_MoveLinear_instance (AxesGroup :=「パラメータ」, Execute :=「パラメータ」, Position :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, CoordSystem :=「パラメータ」, BufferMode :=「パラメータ」, TransitionMode :=「パラメータ」, MoveMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Position	目標位置	ARRAY [0..3] OF LREAL	負数、正数、「0」	0	直線補間の目標位置を指定します。 単位は［指令単位］となります。*1
Velocity *2	目標速度	LREAL	正数	0	目標速度を指定します。 単位は［指令単位／s］となります。*1
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は［指令単位／s^2］となります。*1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*1
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は［指令単位／s^3］となります。*1

入力変数	名称	データ型	有効範囲	初期値	内容
CoordSystem	座標系	_eMC_COORD_SYSTEM	0: _mcACS	0 *3	座標系を指定します。 0: 軸座標系(ACS)
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2: ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ
TransitionMode	トランジションモード（遷移モード）	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	動作の経路を指定します。 0: トランジション無効 10: コーナースーパーインポーズ
MoveMode	移動方法選択	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative	0 *3	移動方法を選択します。 0: 絶対値位置決め 1: 相対値位置決め

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 目標速度は必ず設定してください。設定せずに動作させると異常が発生します。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位置決めが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	軸が移動を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_GroupStop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	―	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

• 2 軸～4 軸の直線補間を行います。

使用上の注意

• 軸グループの構成軸のいずれかが、原点未確定の場合は、「原点未確定状態での命令起動異常（エラーコード: 5466 Hex）」となります。
• 直線補間を行う場合、その軸グループに属している論理軸のうち、いずれかの限界入力が「ON」していると、命令を起動できません。

命令の詳細

以下に本命令の詳細について説明します。

● 直線補間の手順

直線補間は次の手順で実行します。

1 補間を行う軸グループの登録

• 補間を行う軸グループを決めます。
軸グループとは_MC_GRP[*]、または_MC1_GRP[*]、_MC2_GRP[*]です。
• この軸グループ変数の［軸構成］を指定します。指定できる軸数は 2～4 軸です。
• この軸グループ変数の［構成軸選択］で、補間を行う軸の組み合わせを指定します。
• 軸は、軸番号ではなく、論理軸（軸 A0～軸 A3）を使用します。
• ［構成軸選択］で、論理軸 A0～A3 にそれぞれ軸番号を前詰めで指定します。
• 軸番号は、形式やシリーズによって以下のようになります。

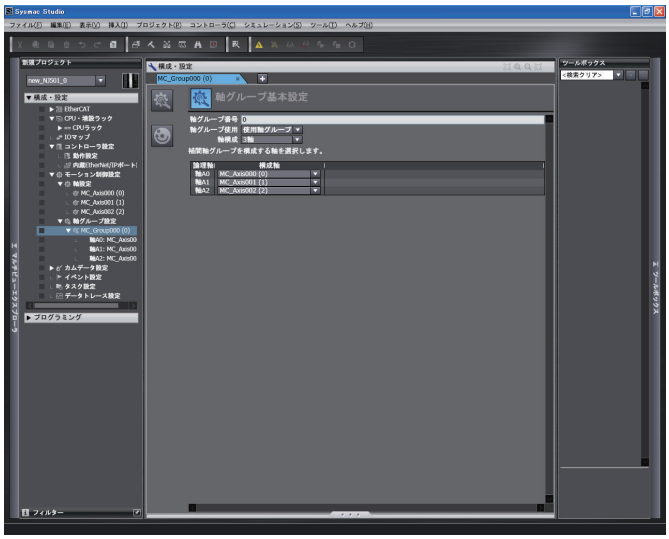
形式またはシリーズ名	軸番号
形 NX701 CPU ユニット	軸 0～軸 255

形式またはシリーズ名	軸番号
形 NX502 CPU ユニット	軸 0～軸 255
形 NX102 CPU ユニット	軸 0～軸 14
形 NX1P2 CPU ユニット	軸 0～軸 11
NJ シリーズ CPU ユニット	軸 0～軸 63

参考

軸番号の詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編 (SBCE-433)』を参照してください。

- Sysmac Studio の軸グループ設定で、使用する［軸構成］を選択し、論理軸に軸番号を割り付けます。たとえば、3 軸を有する軸グループ［MC_Group000］に対して、「MC_Axis000」，「MC_Axis001」，「MC_Axis002」を登録する例を以下に示します。



使用上の注意

軸グループにエンコーダ軸、または仮想エンコーダ軸が含まれている場合は「軸種別不適合（エラーコード: 543D Hex)」となり、動作が終了します。
必ず［サーボ軸］、または［仮想サーボ軸］を選択してください。

2 軸グループの有効化

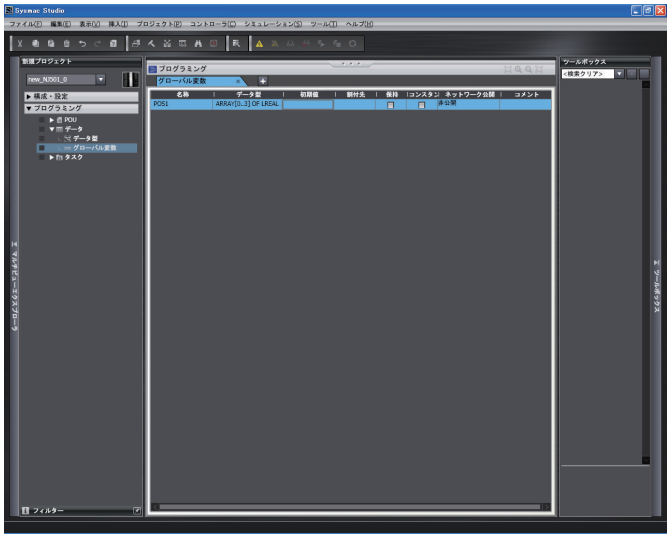
- 軸グループの各構成軸をサーボオン状態にし、さらにそれらを原点確定状態とします。
- MC_GroupEnable（軸グループ有効）命令を実行し、登録した軸グループを有効にします。

これで、直線補間命令が使用できるようになります。

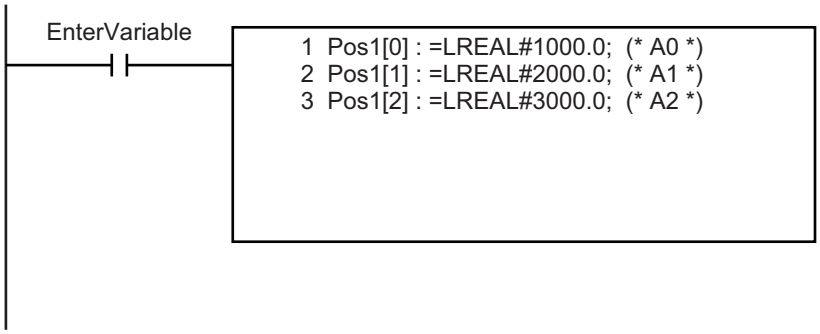
● Position（目標位置）

- 「Position」（目標位置）には、軸グループ変数の［構成軸選択］で指定した軸数分の目標位置を指定します。
- 「Position」(A0, A1, A2, A3)に代入する変数は、Sysmac Studio 上で 1×4 の配列変数を作成する必要があります。変数名は任意でつけることができます。
  その配列に各軸の目標位置を代入します。
  使用する軸グループに属する軸数が 4 軸より少ない場合でも、必ず 1×4 の配列を宣言してください。

Sysmac Studio で配列変数 「Pos1」を宣言した場合の例は下記のようになります。



また、インライン ST を用いて「Pos1」に目標位置を代入する例は以下のようになります。
図では、A0～A2 の各軸に目標位置(1000.0, 2000.0, 3000.0)を代入しています。

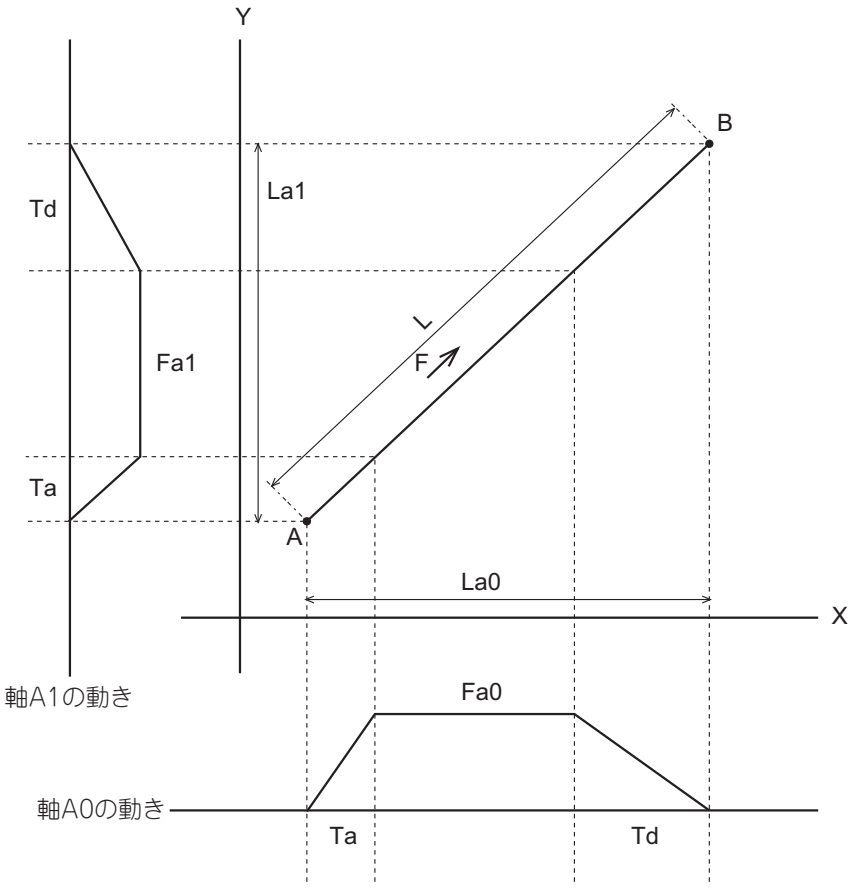


- カウントモード選択で［ロータリモード］を設定している軸を補間軸に指定した場合、絶対位置を指定すると目標値は「Direction」（方向選択）が［方向指定なし］と同様になります。
詳細については、「Direction（方向選択）（P.3-50）」を参照してください。

● Velocity（目標速度）、Acceleration（加速度）、Deceleration（減速度）、Jerk（ジャーク）

- 直線補間時の補間速度、加速度、減速度、ジャークは、「Velocity」、「Acceleration」、「Deceleration」、「Jerk」で指定します。
- 直線補間の場合、直線を各軸に分解して動かします。

例として 2 軸を使用して、A 点から B 点までの直線補間を表すと、下図のようになります。



4 軸の直線補間の場合、各軸の目標速度は補間速度と各軸の移動距離により、次のようになります。
F: 指定された補間送り速度
Fa0: F を軸 A0 へ展開した補間送り速度
Fa1: F を軸 A1 へ展開した補間送り速度
Fa2: F を軸 A2 へ展開した補間送り速度
Fa3: F を軸 A3 へ展開した補間送り速度
Ta: 補間加速時間
Td: 補間減速時間
L: 指定された軌跡の移動量
La0、La1、La2、La3: 軸 A0、軸 A1、軸 A2、軸 A3 の移動量

前述とすると、L, Fa0, Fa1, Fa2, Fa3 は次式で表すことができます。

$$Fa0 = F \times \frac{La0}{L}$$

$$Fa1 = F \times \frac{La1}{L}$$

$$Fa2 = F \times \frac{La2}{L}$$

$$Fa3 = F \times \frac{La3}{L}$$

$$L = \sqrt{La0^2 + La1^2 + La2^2 + La3^2}$$

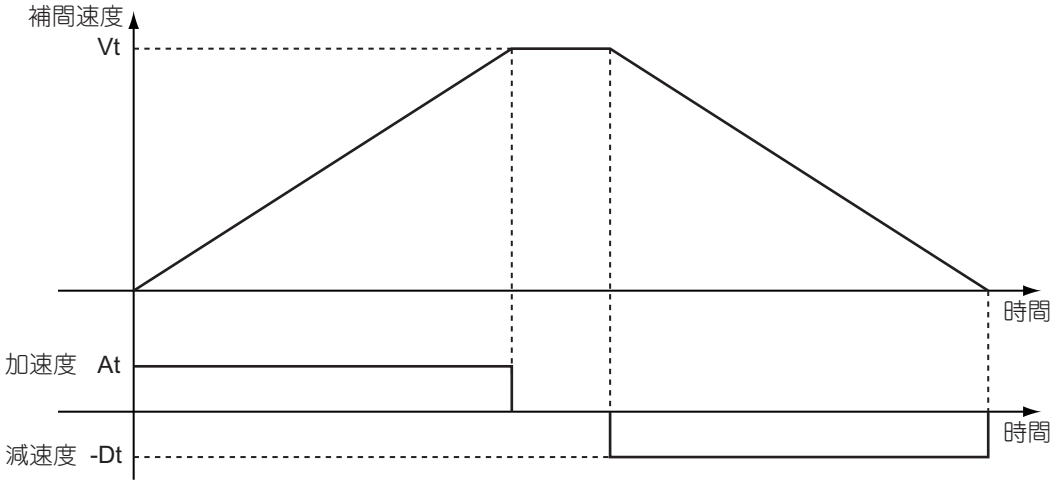
Velocity（目標速度）

- 「Velocity」（目標速度）に「0」を指定すると、補間速度指定異常が発生します。指定された軸グループの軸が動作中であれば、全軸停止します。
- 算出した「Velocity」（目標速度）Fa0～Fa3 のいずれかが、各軸の最高速度をオーバーする場合、いずれかの軸が最高速度で動くように「Velocity」（目標速度）を自動的に調整します。

Jerk（ジャーク）

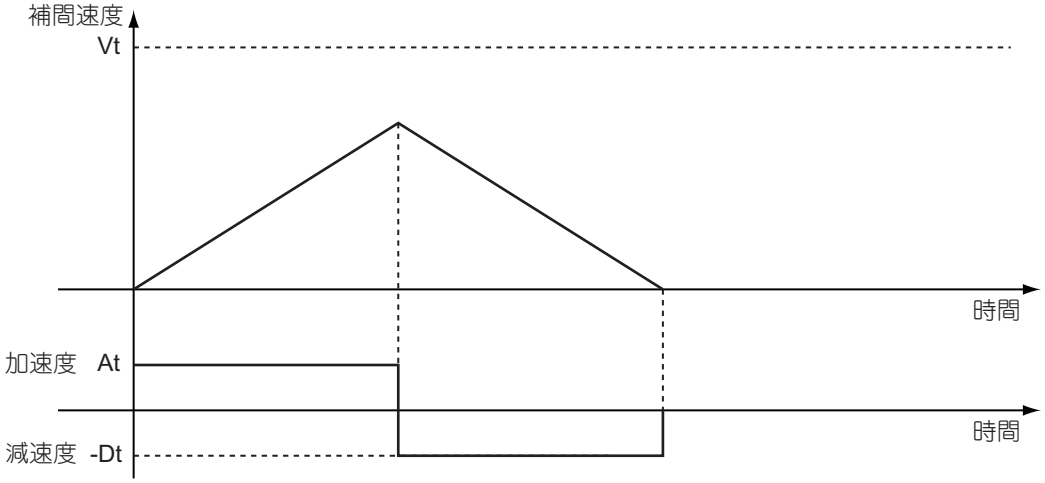
「Jerk」（ジャーク）に「0」を指定した場合と「0」以外を指定した場合の「Acceleration」（加速度）、「Deceleration」（減速度）、「Velocity」（目標速度）の関係は次のようになります。

- 「Jerk」（ジャーク）が「0」の場合
  加速度 At、減速度 Dt で速度の指令値を作成します。



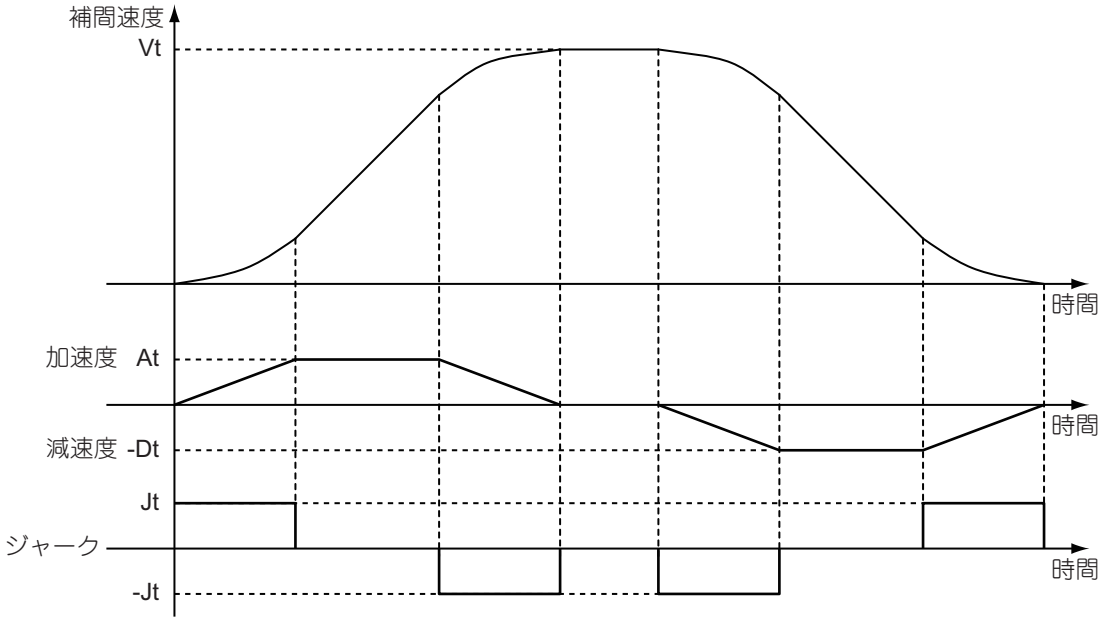
Vt: 補間速度の指定値、At: 加速度の指定値、Dt: 減速度の指定値

- 「Jerk」（ジャーク）が「0」で移動距離が短い場合
  補間速度は、指定値 Vt（目標速度）まで達しません。



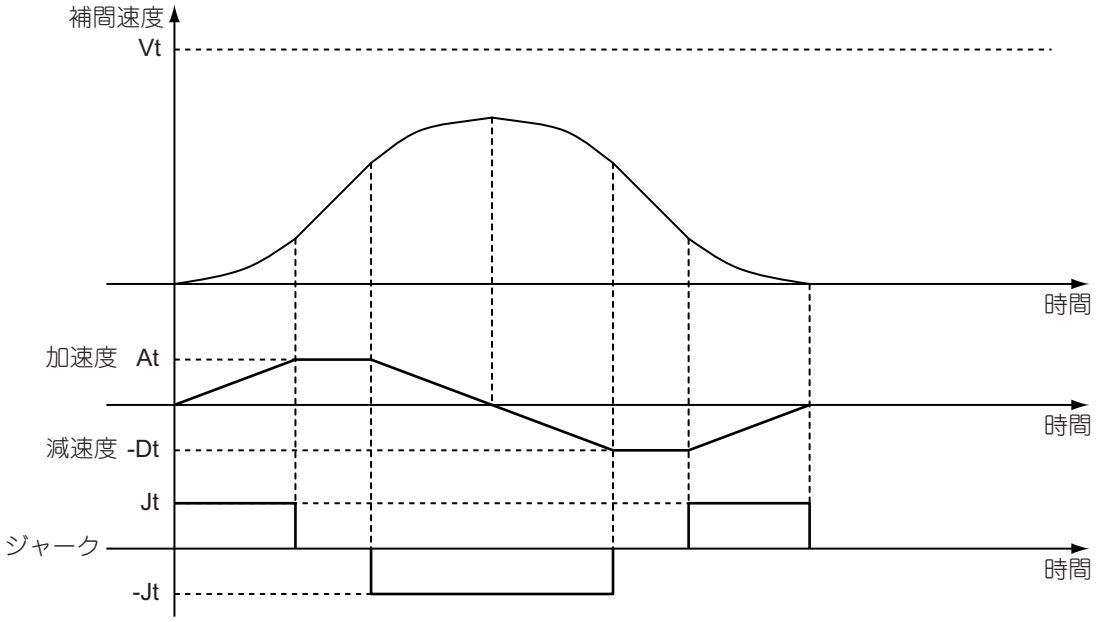
Vt: 補間速度の指定値、At: 加速度の指定値、Dt: 減速度の指定値

- 「Jerk」（ジャーク）が「0」以外の場合
  加速度の上限は At、減速度の上限は Dt として速度の指令値を作成します。



Vt: 補間速度の指定値、At: 加速度の指定値、Dt: 減速度の指定値、Jt: ジャークの指定値

- 「Jerk」（ジャーク）が「0」以外で移動距離が短い場合
  補間速度は、指定値 Vt（目標速度）まで達しません。



Vt: 補間速度の指定値、At: 加速度の指定値、Dt: 減速度の指定値、Jt: ジャークの指定値

参考

- 「Acceleration」（加速度）に「0」を指定すると、すぐに指定した「Velocity」（目標速度）で動かします。
- 「Deceleration」（減速度）に「0」を指定すると、すぐに停止します。ただし、バッファモードでブレンディングを指定している場合は、停止せずに、次の動作で指定した目標速度に移行します。
  詳細については、「BufferMode（バッファモード選択）（P.4-19）」を参照してください。
- 「Acceleration」（加速度）、または「Deceleration」（減速度）に「0」を指定した場合、「Jerk」（ジャーク）の設定は無効となります。

● CoordSystem（座標系）

- 直線補間を行う座標系を指定します。
- 2 軸以上で構成される軸座標系(ACS)だけ対応しています。

● BufferMode（バッファモード選択）

- 1 つ前の補間動作と今回の補間動作のつなぎ方を指定します。
- 次の 6 種類があります。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。
バッファード	現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。
ブレンディング	現在実行中の命令が目標位置に到達したときの速度（中継速度）を起動速度として、バッファリングされている本命令を連続して動作させます。中継速度で目標位置に到達するように、現在実行中の命令の動作を変更します。中継速度の指定方法は、以下の 4 種類があります。 また、ブレンディングのオプションとして、トランジション指定（後述）があります。
ブレンディングロウ	現在実行中の目標速度とバッファリングされている目標速度の、どちらか低い方を中継速度とします。
ブレンディングプレビアス	現在実行中の目標速度を中継速度とします。
ブレンディングネクスト	バッファリングされている本命令の目標速度を中継速度とします。
ブレンディングハイ	現在実行中の目標速度とバッファリングされている目標速度の、どちらか高い方を中継速度とします。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● TransitionMode（トランジションモード）

- 1 つ前の補間動作と今回の補間動作で作る軌跡のつなぎ方を指定します。
- 「TransitionMode」（トランジションモード）では、_mcTMNone（トランジション無効）、または _mcTMCornerSuperimposed（コーナースーパーインポーズ）のいずれか指定します。
- 「BufferMode」（バッファモード選択）で［ブレンディング］を指定すると、「TransitionMode」（トランジションモード）が有効になります。
- ［ブレンディング］を指定せず「TransitionMode」（トランジションモード）を_mcTMNone（トランジション無効）以外に指定した場合は異常となります。

トランジション無効

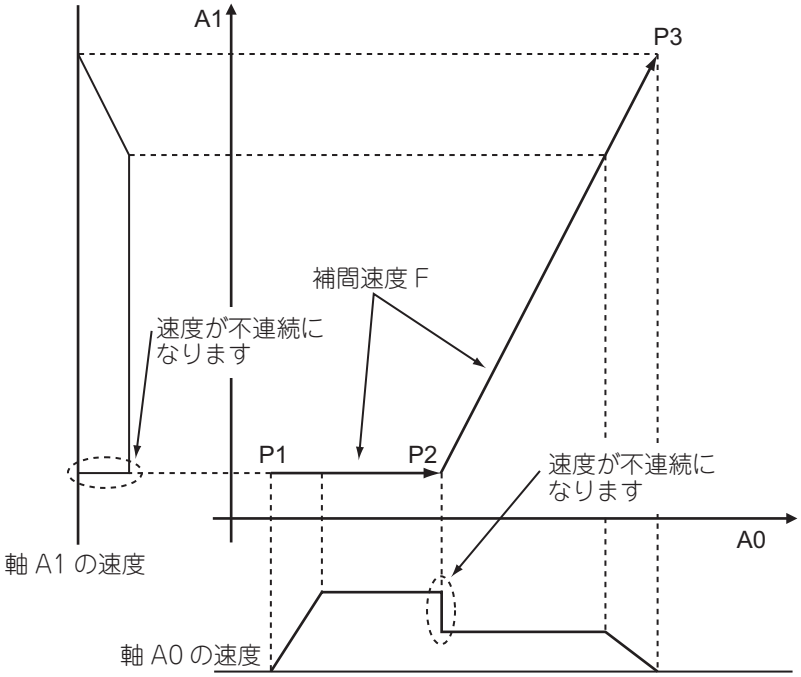
軌跡を優先して速度指令値を作成するため、各軸の速度指令値は、1 つ 1 つの動作の切れ目で急激に速度が変化することがあります。

動作例

P1 から P2、P2 から P3 への「Velocity」（目標速度）、「BufferMode」（バッファモード選択）、「TransitionMode」（トランジションモード）は次の通りです。

- P1 から P2 への動作：Velocity = F、BufferMode = アボーティング、TransitionMode = _mcTMNone（トランジション無効）
- P2 から P3 への動作：Velocity = F、BufferMode = ブレンディングネクスト、TransitionMode = _mcTMNone（トランジション無効）
- 位置 P1 から動き始め、位置 P2 を通過して、位置 P3 へ直線補間で動作します。

- 位置 P2 を通過するとき、直線補間の速度 F を維持します。このため、位置 P2 では、図に示すように、速度が不連続になります。



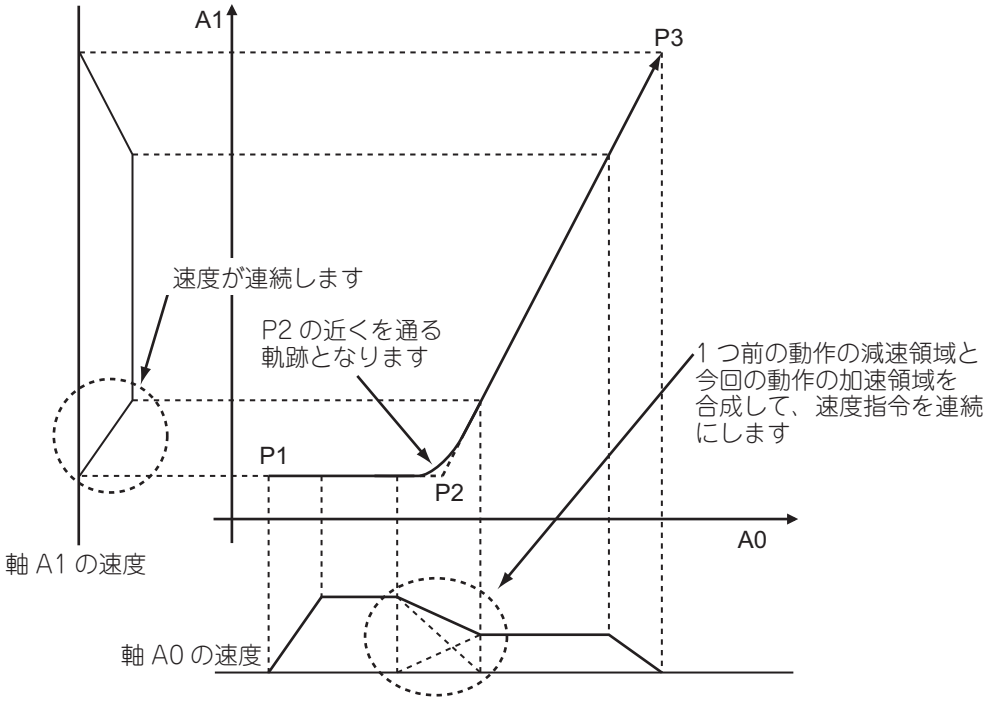
コーナースーパーインポーズ

各軸の指令速度を連続で動かしたい場合に使用します。

動作例

P1 から P2、P2 から P3 への「Velocity」（目標速度）、「BufferMode」（バッファモード選択）、「TransitionMode」（トランジションモード）は次の通りです。

- P1 から P2 への動作：Velocity = F、BufferMode = アボーティング、TransitionMode = _mcTMNone（トランジション無効）
- P2 から P3 への動作：Velocity = F、BufferMode = ブレンディングネクスト、TransitionMode = _mcTMCornerSuperimposed（コーナースーパーインポーズ）
- 位置 P1 から動き始め、位置 P2 の近くを通過して、位置 P3 へ直線補間で動作します。
- 各軸の指令速度を連続にするため、1 つ前の動作の減速領域と今回の動作の加速領域を合成して、指令速度を作成します。このため、今回の動作の加速時間は、1 つ前の動作の減速時間と同じになります。



合成した軌跡は、P2 の近くを通る軌跡になります。
P2 からこの軌跡への距離は、以下のようになります。

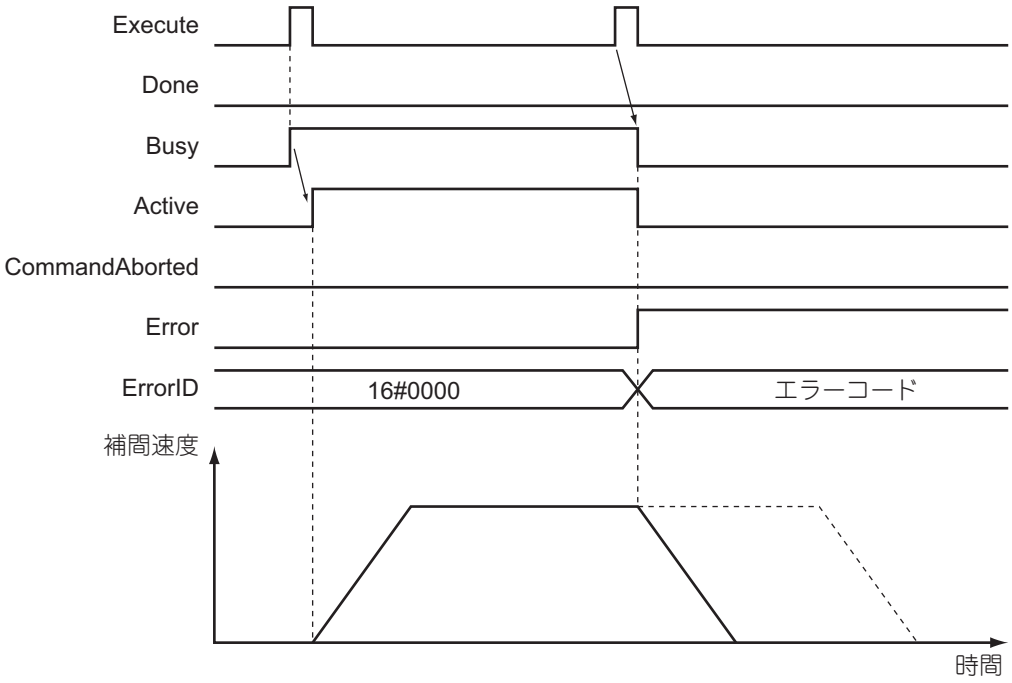
- 補間速度が速くなる、または 1 つ前の命令の減速度が小さくなると、長くなります。
- 補間速度が遅くなる、または 1 つ前の命令の減速度が大きくなると、短くなります。

参考

コーナースーパーインポーズする区間では、「Jerk」（ジャーク）の指定は無効になります。

モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となり、直線補間動作中の全軸が停止します。



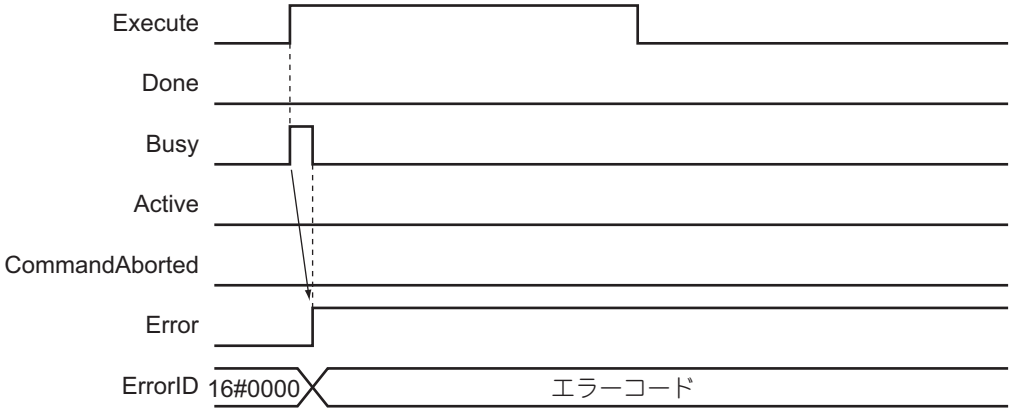
モーション命令多重起動

本命令実行中に、使用できる命令には制限があります。

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、多重起動の繰り返しによる直線補間についてサンプルプログラムを掲載します。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

軸種別

軸	軸種別
軸 1	サーボ軸
軸 2	サーボ軸

カウンタモード

軸	カウンタモード
軸 1	リニアモード
軸 2	リニアモード

座標単位選択

軸	座標単位
軸 1	mm
軸 2	mm

● 軸グループパラメータの設定

軸構成

［2 軸］とします。

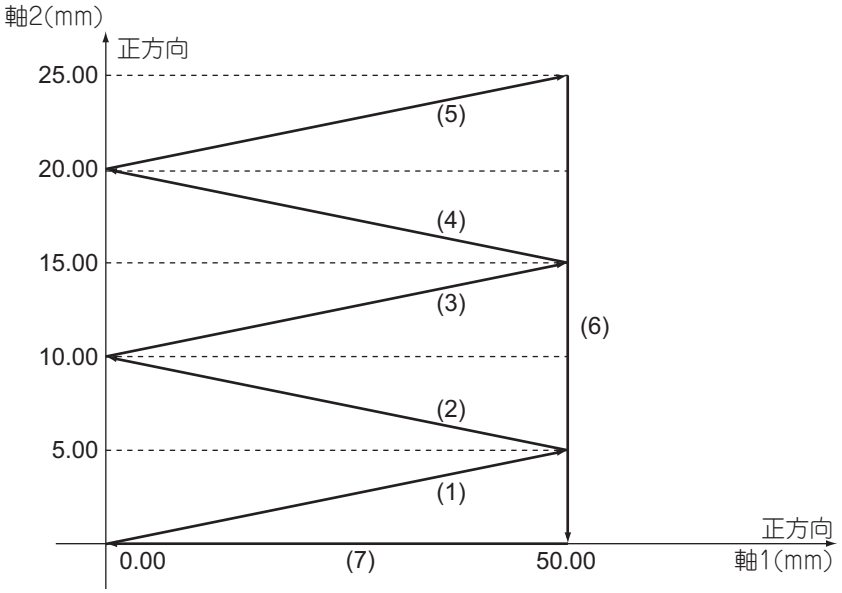
軸選択

［軸 1］、［軸 2］とします。

動作例

直線補間動作を自動で行い、原点に戻ったところで停止する動作例を以下に示します。
直線補間(1)が動作中に直線補間(2)～(7)を多重起動します。そのときの「バッファモード選択」は［バッファード］にします。
この例では直線補間(1)の出力変数「Active」（制御中）が TRUE になったときに、(2)～(7)の命令多重起動を行っています。多軸協調動作では最大 7 命令まで多重起動が可能です。

● 動作パターン



直線補間を用いて、(軸 1、軸 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)の順に位置決めを行い、停止します。

ラダー図

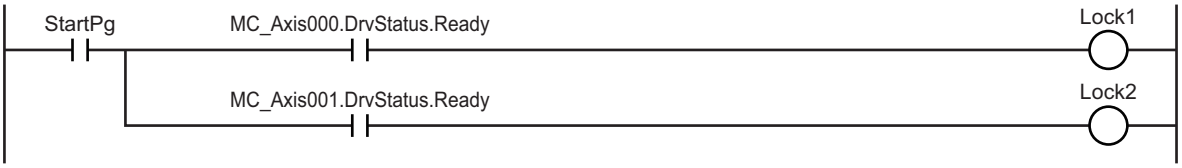
● 主な変数

名称	データ型	初期値	コメント
MC_Group000	_sGROUP_REF	－	軸グループ 0 の軸グループ変数です。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	軸グループ 0 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Group000.Status.Disabled	BOOL	FALSE	軸グループ 0 が無効な場合に TRUE となります。
MC_Axis000	_sAXIS_REF	－	軸 1 の軸変数です。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態の場合に TRUE となります。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001	_sAXIS_REF	－	軸 2 の軸変数です。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態の場合に TRUE となります。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、TRUE となります。

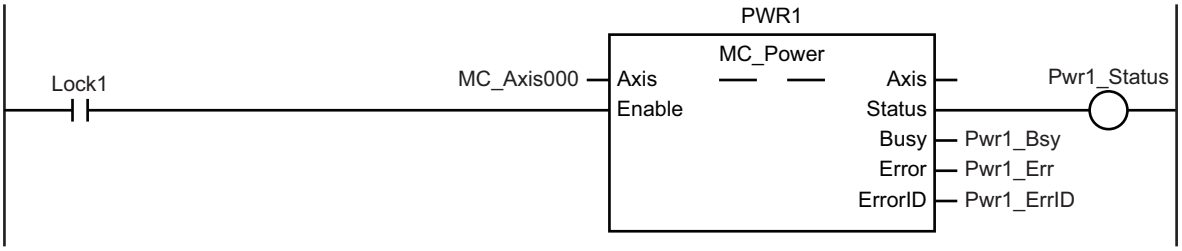
名称	データ型	初期値	コメント
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、軸グループに属する各軸をサーボ ON 状態にします。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

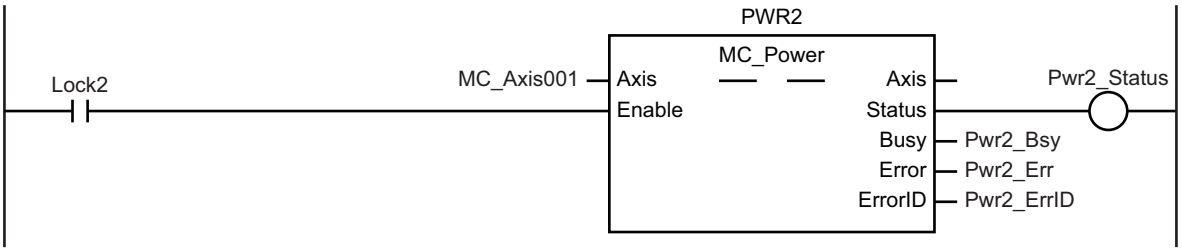
● サンプルプログラム

接点 StartPg が TRUE になったら、各軸のサーボドライバがサーボレディ状態になっていることを確認する

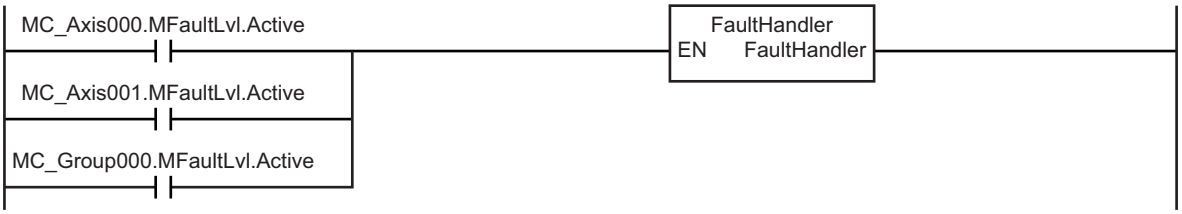


サーボレディ状態になっていれば、各軸をサーボ ON 状態にする

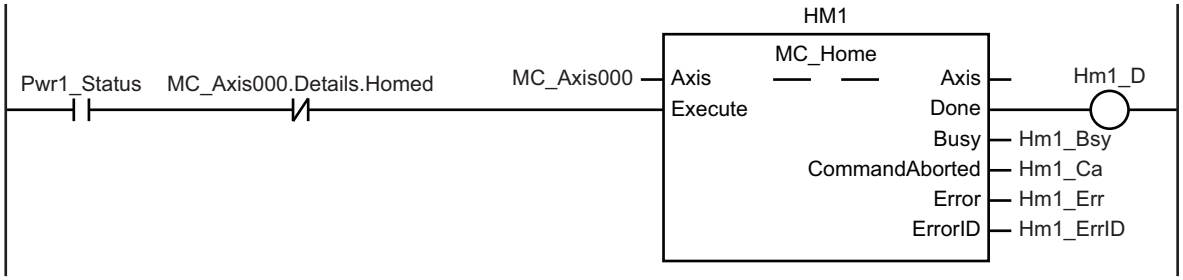




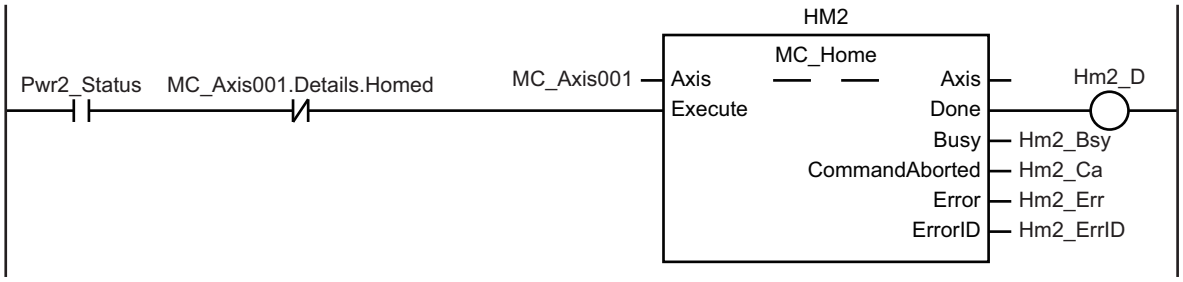
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



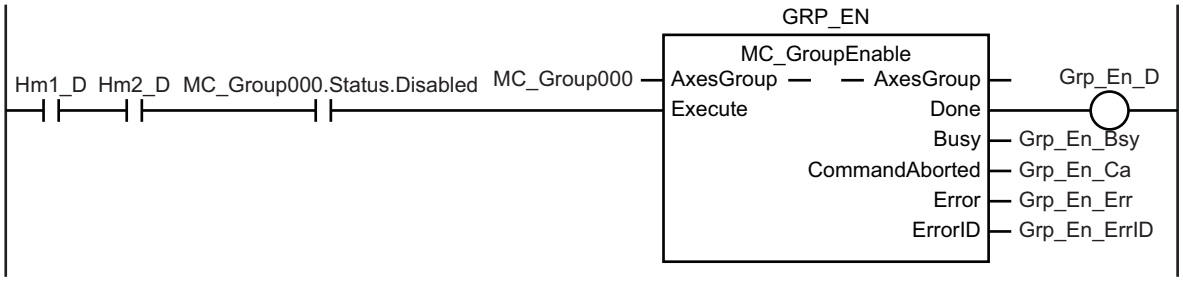
軸 1 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



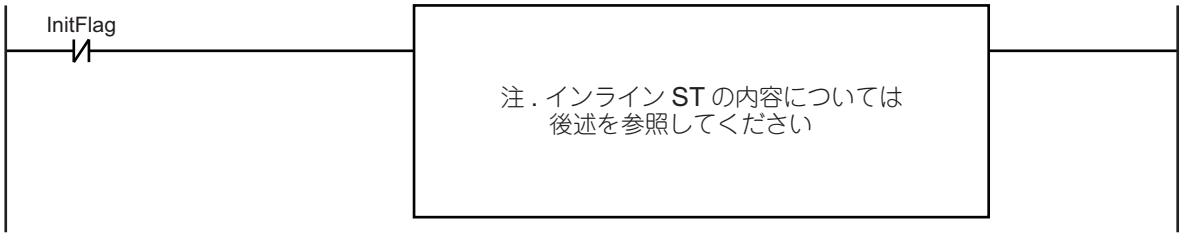
軸 2 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



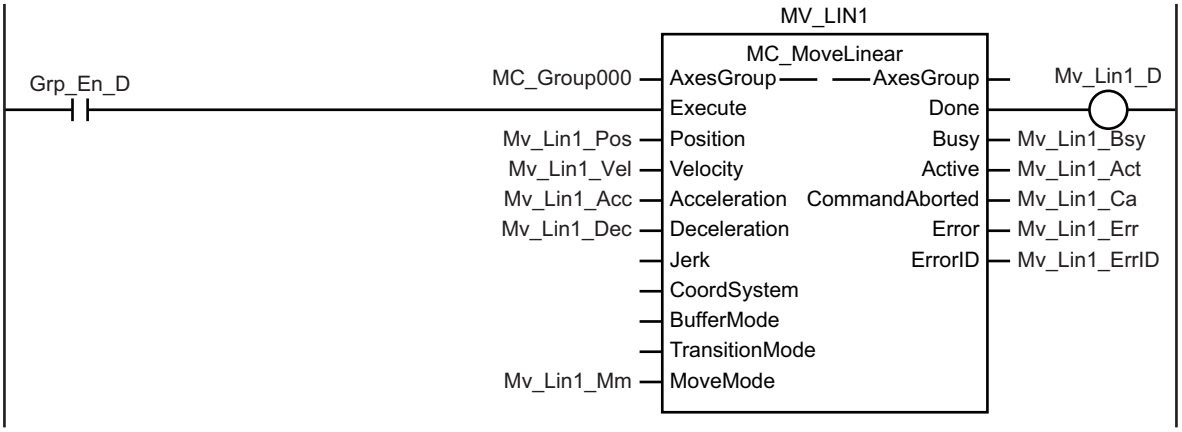
軸 1 と軸 2 の原点確定後、軸グループを有効にする



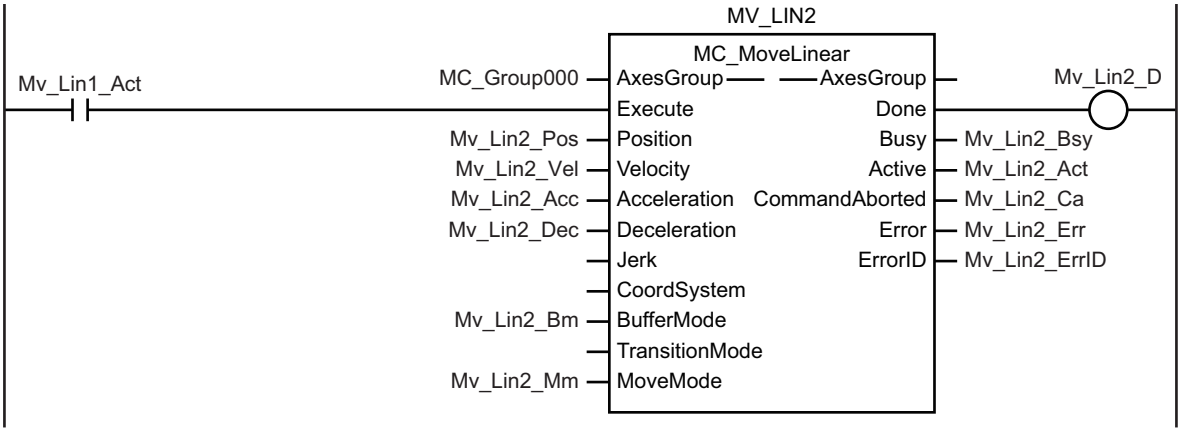
直線補間のパラメータを設定する



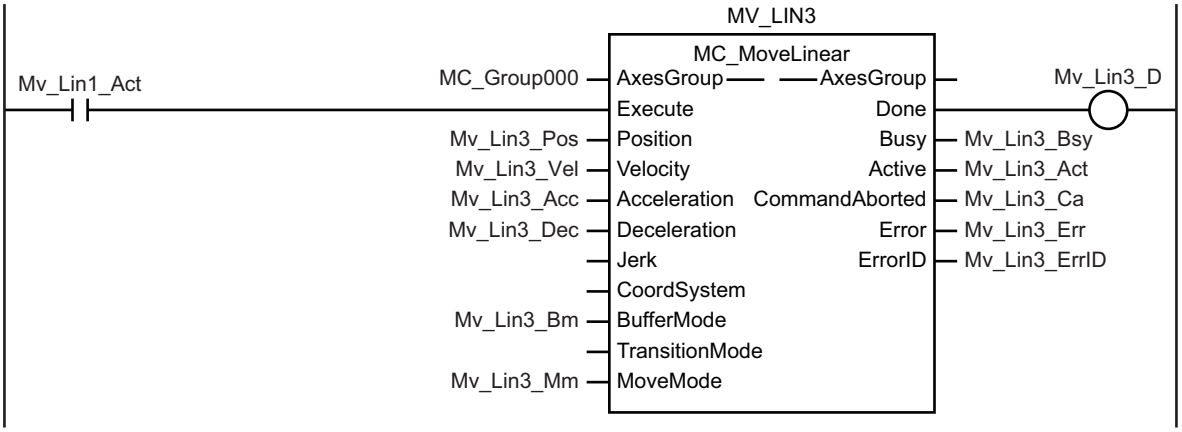
軸グループが有効になったことを確認し、直線補間 (1) を実行する



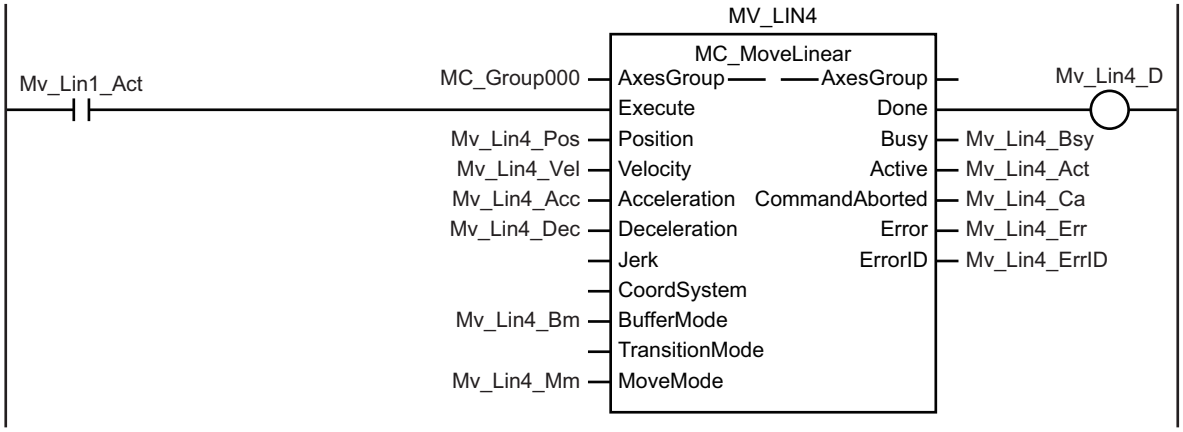
直線補間 (1) の開始後、直線補間 (2) を多重起動する



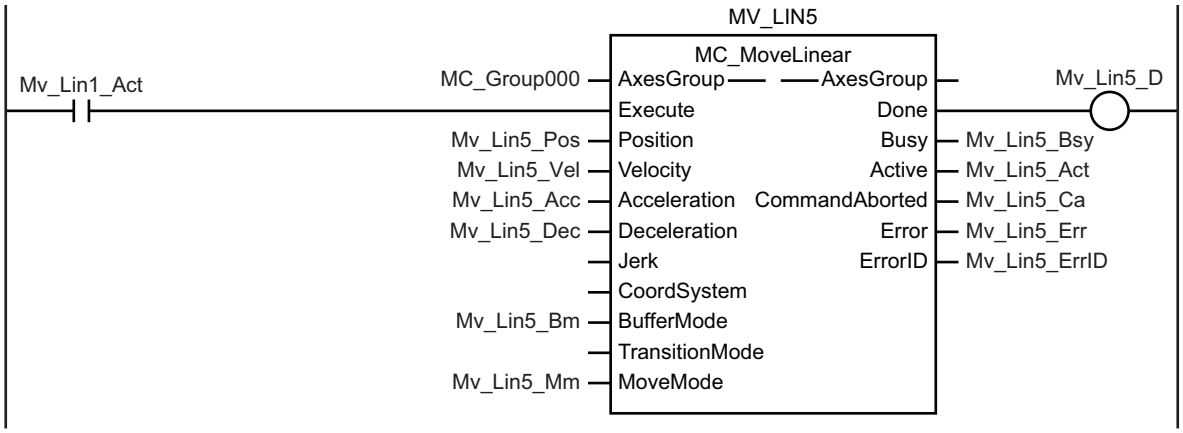
直線補間 (1) の開始後、直線補間 (3) を多重起動する



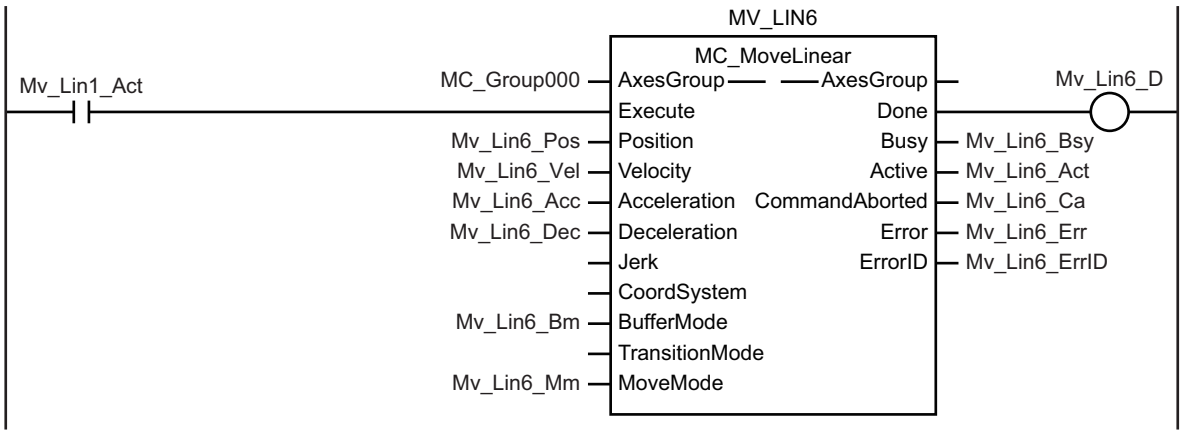
直線補間 (1) の開始後、直線補間 (4) を多重起動する



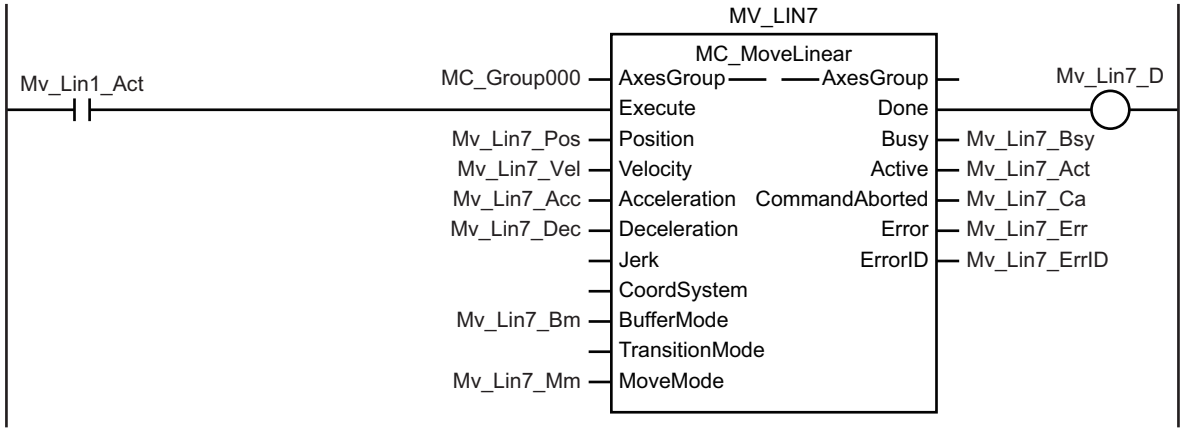
直線補間 (1) の開始後、直線補間 (5) を多重起動する



直線補間 (1) の開始後、直線補間 (6) を多重起動する



直線補間 (1) の開始後、直線補間 (7) を多重起動する



インライン ST の内容

```
//MV_LIN1 パラメータ
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 パラメータ
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 パラメータ
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 パラメータ
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 パラメータ
```

```
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 パラメータ
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 パラメータ
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;

//入力パラメータを設定後、InitFlag を TRUE にします
InitFlag := TRUE;
```

ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Group000	_sGROUP_REF	−	軸グループ 0 の軸グループ変数です。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	軸グループ 0 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Group000.Status.Disabled	BOOL	FALSE	軸グループ 0 が無効な場合に TRUE となります。
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態の場合に TRUE となります。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001	_sAXIS_REF	−	軸 2 の軸変数です。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態の場合に TRUE となります。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。

名称	データ型	初期値	コメント
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、軸グループに属する各軸をサーボ ON 状態にします。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag=FALSE THEN

    //MV_LIN1 パラメータ
    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 パラメータ
    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 パラメータ
    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 パラメータ
    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 パラメータ
    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 パラメータ
    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 パラメータ
    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;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag := TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状
態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE; //軸 1 をサーボ ON にします
ELSE
    Pwr1_En:=FALSE; //軸 1 をサーボ OFF にします
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 2 をサーボ ON 状
態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE; //軸 2 をサーボ ON にします
ELSE
```

```
    Pwr2_En:=FALSE; //軸 2 をサーボ OFF にします
END_IF;

//軽度フォールトレベルの異常発生時の処理
//異常発生時の処理(FaultHandler)の内容は、装置にあわせてお客様でプログラミングしてください
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (MC_Group000.MFaultLvl.Active=TRUE)THEN
    FaultHandler();
END_IF;

//軸 1 がサーボ ON 状態で、かつ原点未確定状態なら、原点確定を行います
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//軸 2 がサーボ ON 状態で、かつ原点未確定状態なら、原点確定を行います
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//軸 1 と軸 2 が原点確定状態で、かつ軸グループ 0 が無効のとき、軸グループ 0 を有効にします
IF (Hm1_D=TRUE) AND (Hm2_D=TRUE) AND (MC_Group000.Status.Disabled=TRUE)THEN
    Grp_En_Ex:= TRUE;
END_IF;

//軸グループ有効 MC_GroupEnable が完了したら、直線補間(1)を実行します
IF Grp_En_D=TRUE THEN
    Mv_Lin1_Ex:=TRUE;
END_IF;

//直線補間(1)の出力 Active が TRUE になったら、直線補間(2)～(7)を多重起動します
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;

//軸 1 の MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

//軸 2 の MC_Power
```

```
PWR2(
    Axis := MC_Axis001,
    Enable := Pwr2_En,
    Status => Pwr2_Status,
    Busy => Pwr2_Bsy,
    Error => Pwr2_Err,
    ErrorID => Pwr2_ErrID
);

//軸 1 の MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

//軸 2 の MC_Home
HM2(
    Axis := MC_Axis001,
    Execute := Hm2_Ex,
    Done => Hm2_D,
    Busy => Hm2_Bsy,
    CommandAborted => Hm2_Ca,
    Error => Hm2_Err,
    ErrorID => Hm2_ErrID
);

//軸グループ 0 を有効にする
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
);

//直線補間(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
);

//直線補間(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
);

//直線補間(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
);

//直線補間(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
);

//直線補間(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
);

//直線補間(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
);

//直線補間(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

絶対位置を指定して直線補間を行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveLinearAbsolute	絶対値直線補間	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 :=「パラメータ」, Execute :=「パラメータ」, Position :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, CoordSystem :=「パラメータ」, BufferMode :=「パラメータ」, TransitionMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Position	目標位置	ARRAY [0..3] OF LREAL	負数、正数、「0」	0	直線補間の目標位置を指定します。 単位は［指令単位］となります。*1
Velocity *2	目標速度	LREAL	正数	0	目標速度を指定します。 単位は［指令単位／s］となります。*1
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は［指令単位／s^2］となります。*1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*1
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は［指令単位／s^3］となります。*1

入力変数	名称	データ型	有効範囲	初期値	内容
CoordSystem	座標系	_eMC_COORD_SYSTEM	0: _mcACS	0 *3	座標系を指定します。 0: 軸座標系(ACS)
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2: ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ
TransitionMode	トランジションモード(遷移モード)	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	動作の経路を指定します。 0: トランジション無効 10: コーナースーパーインポーズ

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 目標速度は必ず設定してください。設定せずに動作させると異常が発生します。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位置決めが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	軸が移動を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_GroupStop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	−	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 2 軸～4 軸の直線補間を行います。
- 目標位置は、絶対位置で指定します。

そのほかは、MC_MoveLinear（直線補間）命令と同一仕様です。
詳細については、MC_MoveLinear（直線補間）命令の「機能説明 (P.4-13)」を参照してください。

使用上の注意

- 軸グループの構成軸のいずれかが、原点未確定の場合は「原点未確定状態での命令起動異常（エラーコード: 5466 Hex)」となります。
- 直線補間を行う場合、その軸グループに属している論理軸のうち、いずれかの限界入力が「ON」していると、命令を起動できません。

MC_MoveLinearRelative

相対位置を指定して直線補間を行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveLinearRelative	相対値直線補間	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 :=「パラメータ」, Execute :=「パラメータ」, Distance :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, CoordSystem :=「パラメータ」, BufferMode :=「パラメータ」, TransitionMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Distance	移動距離	ARRAY [0..3] OF LREAL	負数、正数、「0」	0	直線補間の目標位置を指定します。 単位は［指令単位］となります。*1
Velocity *2	目標速度	LREAL	正数	0	目標速度を指定します。 単位は［指令単位／s］となります。*1
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は［指令単位／s^2］となります。*1
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*1
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は［指令単位／s^3］となります。*1

入力変数	名称	データ型	有効範囲	初期値	内容
CoordSystem	座標系	_eMC_COORD_SYSTEM	0: _mcACS	0 *3	座標系を指定します。 0: 軸座標系(ACS)
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *3	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2: ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ
TransitionMode	トランジションモード(遷移モード)	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *3	動作の経路を指定します。 0: トランジション無効 10: コーナースーパーインポーズ

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 目標速度は必ず設定してください。設定せずに動作させると異常が発生します。
*3. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位置決めが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	軸が移動を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_GroupStop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	−	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 2 軸～4 軸の直線補間を行います。
- 目標位置は、相対位置で指定します。

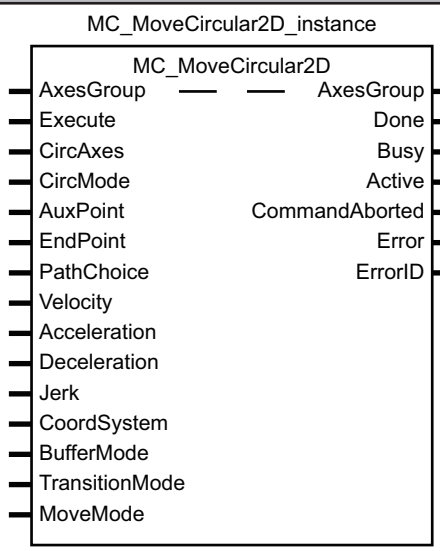
そのほかは、MC_MoveLinear（直線補間）命令と同一仕様です。
詳細については、MC_MoveLinear（直線補間）命令の「機能説明 (P.4-13)」を参照してください。

使用上の注意

- 軸グループの構成軸のいずれかが、原点未確定の場合は「原点未確定状態での命令起動異常（エラーコード: 5466 Hex)」となります。
- 直線補間を行う場合、その軸グループに属している論理軸のうち、いずれかの限界入力が「ON」していると、命令を起動できません。

MC_MoveCircular2D

2 軸の円弧補間を行います。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_MoveCircular2D	2 軸円弧補間	FB		MC_MoveCircular2D_instance (AxesGroup :=「パラメータ」, Execute :=「パラメータ」, CircAxes :=「パラメータ」, CircMode :=「パラメータ」, AuxPoint :=「パラメータ」, EndPoint :=「パラメータ」, PathChoice :=「パラメータ」, Velocity :=「パラメータ」, Acceleration :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, CoordSystem :=「パラメータ」, BufferMode :=「パラメータ」, TransitionMode :=「パラメータ」, MoveMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
CircAxes	円弧軸指定	ARRAY [0,1] OF UINT	0～3	0	円弧補間を行う軸を指定します。 0: 軸 A0 1: 軸 A1 2: 軸 A2 3: 軸 A3
CircMode	円弧補間モード	_eMC_CIRC_MODE	0: _mcBorder 1: _mcCenter 2: _mcRadius	0 *1	円弧補間の方法を指定します。 0: 通過点指定 1: 中心点指定 2: 半径指定

入力変数	名称	データ型	有効範囲	初期値	内容
AuxPoint	補助点	ARRAY [0,1] OF LREAL	負数、正数、「0」	0	通過点位置／中心位置／半径を指定します。 単位は［指令単位］となります。*2
EndPoint	終点	ARRAY [0,1] OF LREAL	負数、正数、「0」	0	目標位置を指定します。 単位は［指令単位］となります。*2
PathChoice	経路選択	_eMC_CIRC_PATHCHOICE	0: _mcCW 1: _mcCCW	0 *1	経路方向を指定します。 0: CW 1: CCW
Velocity *3	目標速度	LREAL	正数	0	目標速度を指定します。 単位は［指令単位／s］となります。*2
Acceleration	加速度	LREAL	正数、または「0」	0	加速度を指定します。 単位は［指令単位／s^2］となります。*2
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*2
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は［指令単位／s^3］となります。*2
CoordSystem	座標系	_eMC_COORD_SYSTEM	0: _mcACS	0 *1	座標系を指定します。 0: 軸座標系(ACS)
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting 1: _mcBuffered 2: _mcBlendingLow 3: _mcBlendingPrevious 4: _mcBlendingNext 5: _mcBlendingHigh	0 *1	モーション命令多重起動時の動作を指定します。 0: アボーティング 1: バッファード 2: ブレンディングロウ 3: ブレンディングプレビアス 4: ブレンディングネクスト 5: ブレンディングハイ
TransitionMode	トランジションモード（遷移モード）	_eMC_TRANSITION_MODE	0: _mcTMNone 10: _mcTMCornerSuperimposed	0 *1	動作の経路を指定します。 0: トランジション無効 10: コーナースーパーインポーズ
MoveMode	移動方法選択	_eMC_MOVE_MODE	0: _mcAbsolute 1: _mcRelative	0 *1	移動方法を選択します。 0: 絶対値位置決め 1: 相対値位置決め

*1. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。
*2. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*3. 目標速度は必ず設定してください。設定せずに動作させると異常が発生します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。

出力変数	名称	データ型	有効範囲	内容
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	位置決めが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	軸が移動を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。 • MC_GroupStop 命令実行中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	−	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

• 2 次元平面で 2 軸の円弧補間を行います。

使用上の注意

- 軸グループの構成軸のいずれかが、原点未確定の場合は「原点未確定状態での命令起動異常（エラーコード: 5466 Hex）」となります。
- 2 軸円弧補間を行う場合、その軸グループに属している論理軸のうち、いずれかの限界入力が「ON」していると、命令を起動できません。

命令の詳細

以下に本命令の詳細について説明します。

● 円弧補間の手順

円弧補間は次の手順で実行します。

1 補間を行う軸グループの登録

- 補間を行う軸グループを決めます。
  軸グループとは_MC_GRP[*]、または_MC1_GRP[*]、_MC2_GRP[*]です。
- この軸グループ変数の［軸構成］を指定します。
- この軸グループ変数の［構成軸選択］で、補間を行う軸の組み合わせを指定します。
- 軸は、軸番号ではなく、論理軸（軸 A0～軸 A3）を使用します。
- ［構成軸選択］で、論理軸 A0～A3 にそれぞれ軸番号を前詰めで指定します。
- 軸番号は、形式やシリーズによって以下のようになります。

形式またはシリーズ名	軸番号
形 NX701 CPU ユニット	軸 0～軸 255
形 NX502 CPU ユニット	軸 0～軸 255
形 NX102 CPU ユニット	軸 0～軸 14
形 NX1P2 CPU ユニット	軸 0～軸 11
NJ シリーズ CPU ユニット	軸 0～軸 63

例: 軸構成が 2 軸で、軸番号 0～1 を、軸 A0～A1 に指定する場合、次のように指定します。

論理軸	軸番号	説明
軸 A0	軸 0	軸 A0 から軸番号の小さい順に指定します。
軸 A1	軸 1	

使用上の注意

軸グループにエンコーダ軸、または仮想エンコーダ軸が含まれている場合は「軸種別不適合（エラーコード: 543D Hex）」となり、動作が終了します。
必ず［サーボ軸］、または［仮想サーボ軸］を選択してください。

参考

軸番号の詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編 (SBCE-433)』を参照してください。

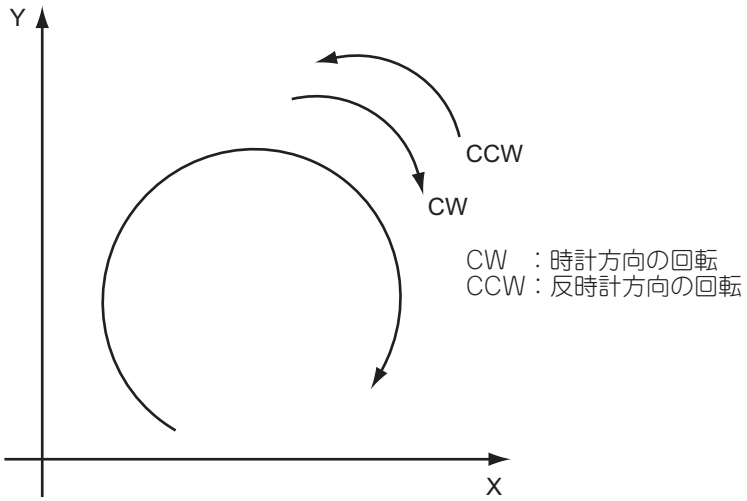
2 軸グループの有効化

- 軸グループの各構成軸をサーボオン状態にし、さらにそれらを原点確定状態とします。
- MC_GroupEnable（軸グループ有効）命令を実行し、登録した軸グループを有効にします。

これで、円弧補間命令が使用できるようになります。

● CircAxes（円弧軸指定）

円弧補間で使用する軸は、X 軸、Y 軸です。



「CircAxes」（円弧軸指定）で X 軸、Y 軸に割り付ける軸を指定します。
指定する軸は、軸番号ではなく、論理軸（軸 A0～軸 A3）で指定します。

使用上の注意

X 軸、または Y 軸に指定する軸のカウントモードは、［リニアモード］としてください。
［ロータリモード］として起動した場合、「カウントモード設定による命令起動異常（エラーコード: 544A Hex）」が発生します。

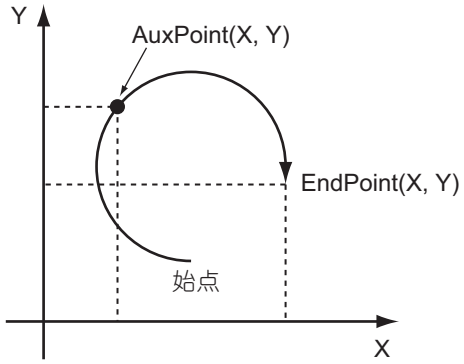
● CircMode（円弧補間モード）

円弧補間の方式には、「通過点指定／中心点指定／半径指定」の 3 種類あり、「CircMode」（円弧補間モード）で指定します。
これらの方式で指定する位置は、［絶対値位置決め］と［相対値位置決め］があり、「MoveMode」（移動方法選択）で指定します。

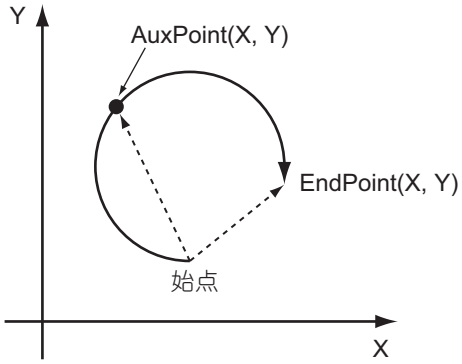
MoveMode	説明
絶対値位置決め	通過点指定で指定する通過点位置、中心点指定で指定する中心点位置と終点は、軸座標系原点からの絶対位置で指定します。
相対値位置決め	通過点指定で指定する通過点位置、中心点指定で指定する中心点位置と終点は、始点からの相対位置で指定します。

通過点指定を例に、［絶対値位置決め］と［相対値位置決め］の違いについて下記に示します。

円弧方式：通過点指定＜絶対値位置決め＞



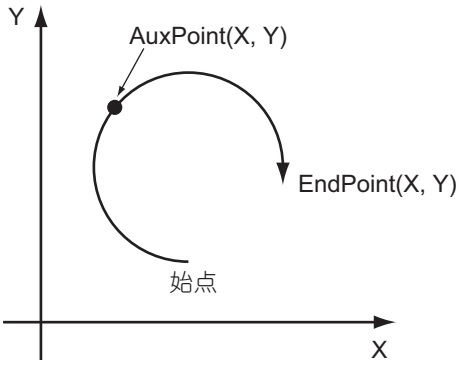
円弧方式：通過点指定＜相対値位置決め＞



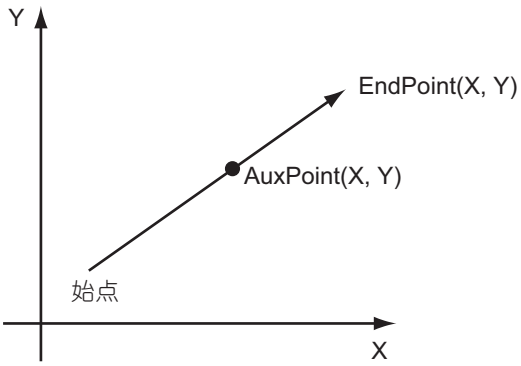
これ以降の説明は、「MoveMode」（移動方式選択）で［絶対値位置決め］を指定していることを前提に説明していきます。

通過点指定

現在位置を始点として、通過点位置の AuxPoint(X, Y)と終点の EndPoint(X, Y)を結ぶ円弧補間を行います。



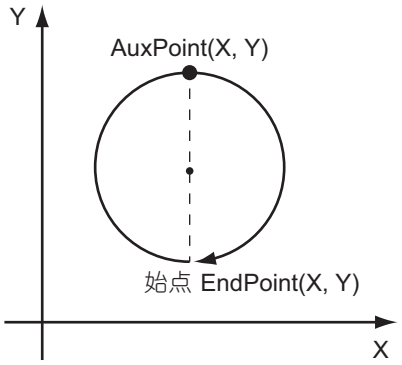
始点、通過点位置と終点が同一直線上の場合、または通過点位置と終点が同一点の場合、始点と通過点位置が同一点の場合は、始点から終点まで直線補間を行います。



使用上の注意

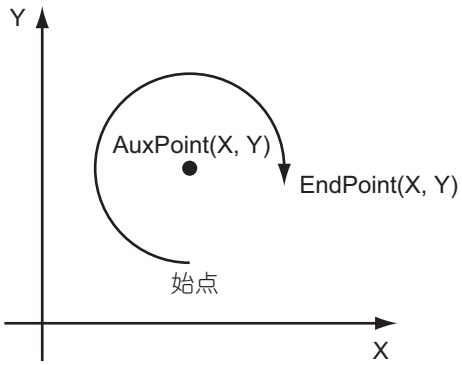
- 同一直線上にあるとは、始点と終点とを結ぶ直線と通過点位置との距離が、X 座標、Y 座標ともに 1 パルス未満の場合、同一直線上にあると判断します。
- 始点、通過点位置、終点が同一点の場合は、異常となります。
  始点、通過点位置、終点が同一点にあるとは、指令単位の指定位置が同一の場合のことです。指令単位での指令位置が異なり、パルス換算位置が同じ場合は同一点と見なされず、異常とはなりません。

始点と終点が同一点の場合、始点と通過点を直径とする真円を描きます。この場合、円弧の回転方向を、「PathChoice」（経路選択）で指定します。



中心点指定

現在位置を始点として、中心位置の AuxPoint(X, Y)で指定された円弧で、終点の EndPoint(X, Y)を結ぶ円弧補間を行います。
円弧の回転方向は、「PathChoice」（経路選択）で指定します。
始点と終点が同一点の場合、真円を描きます。
指定した中心位置から始点までの半径と、終点までの半径が異なっている場合、2 つの半径の平均値を使用して円弧補間を行います。この場合、半径指定と同様に中心位置を算出し、その半径と中心位置を使用します。



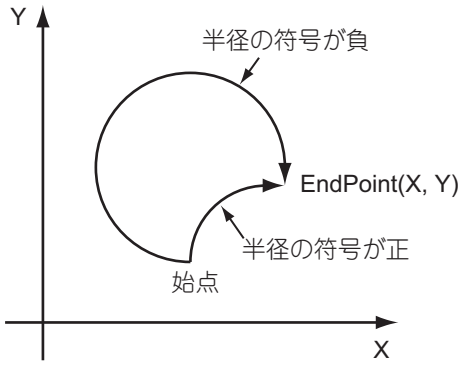
使用上の注意

軸グループパラメータの［中心点補正許容率］に「0」以外が設定されている場合で、指定した中心点の位置が、下式の円をオーバーする場合、「円弧補間中心点指定位置範囲外（エラーコード: 5449 Hex）」が発生します。

- 補正した中心点から算出した半径を「100%」とし、「半径×中心点指定方式チェック方法[%]／100」の円をオーバーする場合

半径指定

現在位置を始点として、半径の AuxPoint(X, Y)で指定した円弧で、終点の EndPoint(X, Y)を結ぶ円弧補間を行います。
半径は、AuxPoint(X, Y)の最初の要素で指定します。2 つめの要素は使用しません。
たとえば、半径を 100 にする場合、AuxPoint(X, Y)に、AuxPoint(100, 0)と指定します。
半径の符号が負であれば円弧長が長い円弧を、半径の符号が正であれば円弧長が短い円弧を描きます。
円弧の回転方向は、「PathChoice」（経路選択）で指定します。



使用上の注意

- 始点と終点が同一点の場合、円弧補間始点終点同一異常が発生し、このグループの全軸の動作が停止します。
- 指定した半径が始点と終点の距離の 1/2 以下の場合は、円弧を作成できないため異常となります。

● Velocity（目標速度）、Acceleration（加速度）、Deceleration（減速度）、Jerk（ジャーク）

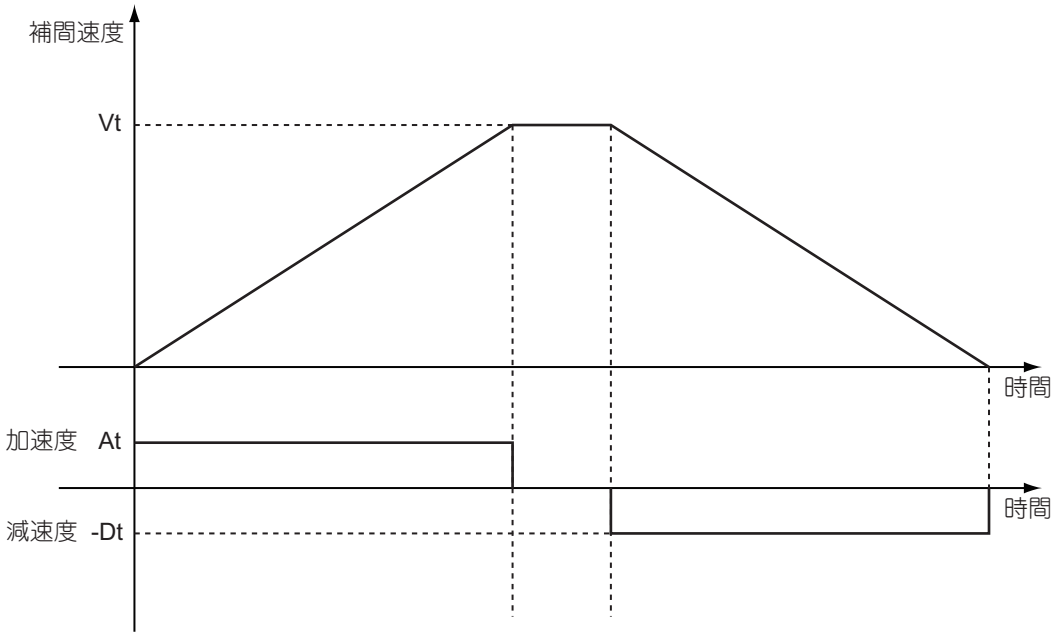
- 円弧補間時の補間速度、加速度、減速度、ジャークは、「Velocity」、「Acceleration」、「Deceleration」、「Jerk」で指定します。
- 円弧補間の補間速度に「0」を指定すると、Velocity 指定異常が発生し、このグループの全軸の動作が停止します。
- 指定した補間速度が、いずれかの軸の最高速度をオーバーする場合、次の動作となります。

1 軸だけオーバーする場合	：	この軸の最高速度で動作するように、補間速度を自動的に調整します。
2 軸ともにオーバーする場合	：	2 軸の最高速度のうち、遅い方の最高速度で動作するように、補間速度を自動的に調整します。

Jerk（ジャーク）

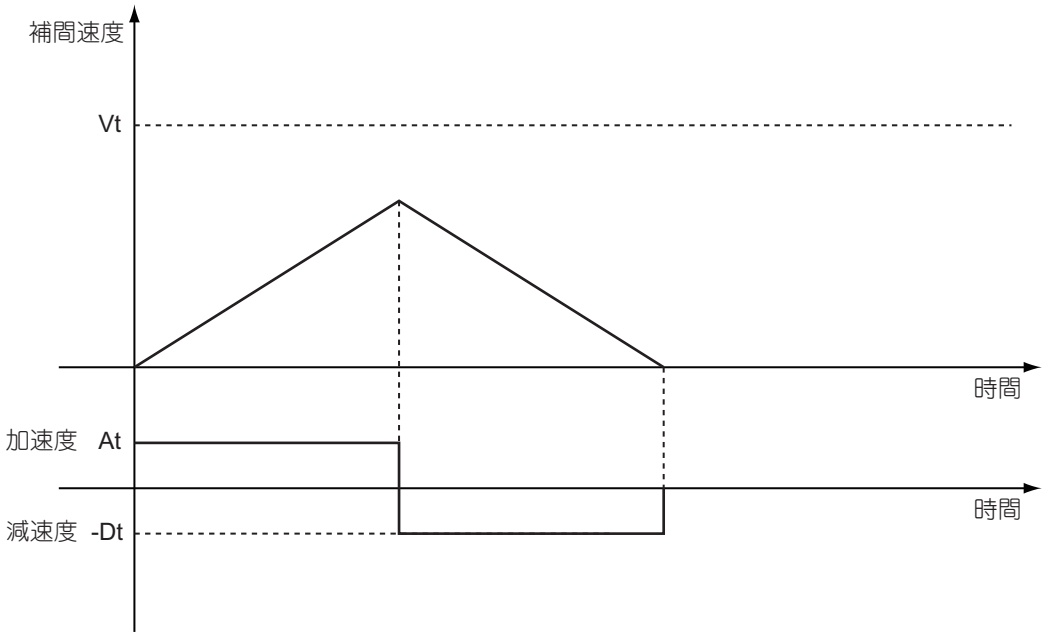
「Jerk」（ジャーク）に「0」を指定した場合と「0」以外を指定した場合の「Acceleration」（加速度）、「Deceleration」（減速度）、「Velocity」（目標速度）の関係は次のようになります。

- 「Jerk」（ジャーク）が「0」の場合
  加速度 At、減速度 Dt で速度の指令値を作成します。



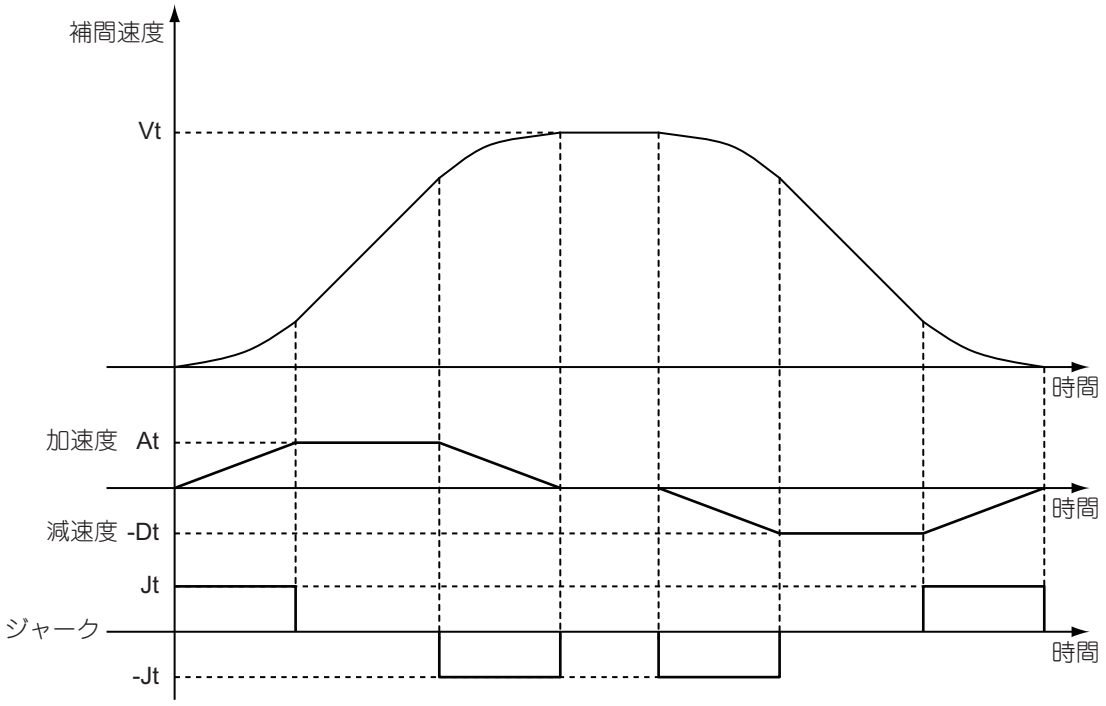
Vt: 補間速度の指定値、At: 加速度の指定値、Dt: 減速度の指定値

- 「Jerk」（ジャーク）が「0」で移動距離が短い場合
  補間速度は、指定値 Vt（目標速度）まで達しません。



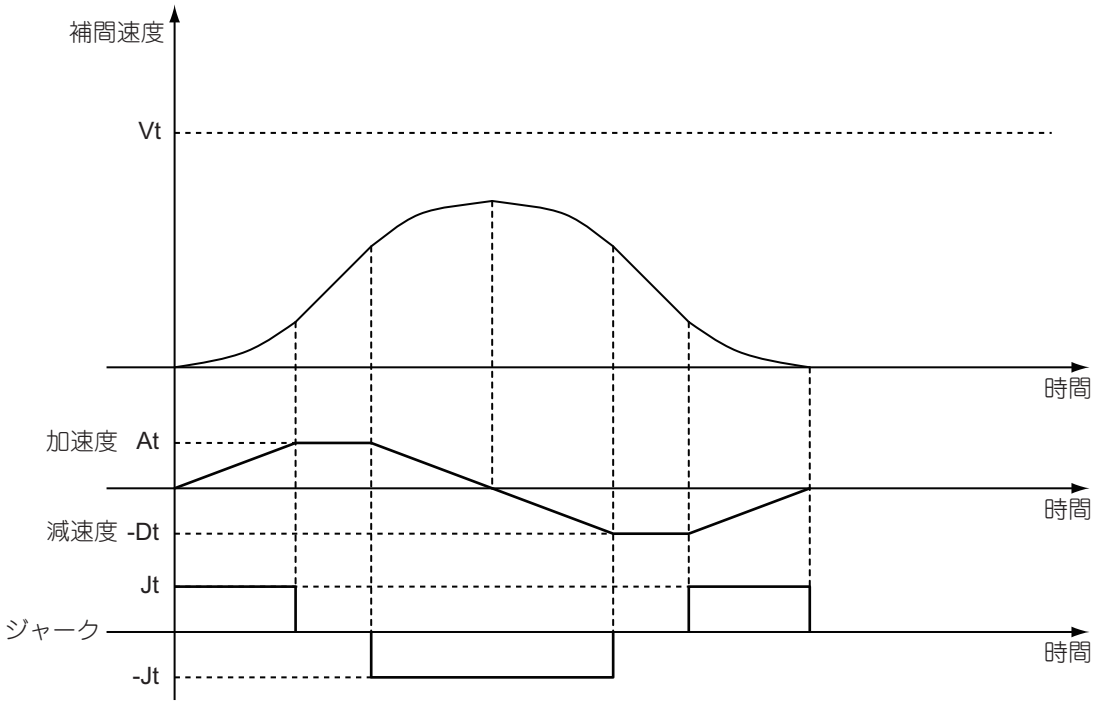
Vt: 補間速度の指定値、At: 加速度の指定値、Dt: 減速度の指定値

- 「Jerk」（ジャーク）が「0」以外の場合
  加速度の上限は At、減速度の上限は Dt として速度の指令値を作成します。



Vt: 補間速度の指定値、At: 加速度の指定値、Dt: 減速度の指定値、Jt: ジャークの指定値

- 「Jerk」（ジャーク）が「0」以外で移動距離が短い場合
  補間速度は、指定値 Vt（目標速度）まで達しません。



Vt: 補間速度の指定値、At: 加速度の指定値、Dt: 減速度の指定値、Jt: ジャークの指定値

参考

- 「Acceleration」（加速度）に「0」を指定すると、すぐに指定した補間速度で動かします。
- 「Deceleration」（減速度）に「0」を指定すると、すぐに停止します。ただし、「バッファモード」で［ブレンディング］を指定している場合は、停止せずに、次の動作で指定した補間速度にすぐに移行します。詳細については、「BufferMode（バッファモード選択）（P.4-19）」を参照してください。
- 「Acceleration」（加速度）、または「Deceleration」（減速度）に「0」を指定した場合、「Jerk」（ジャーク）の設定は無効となります。

● CoordSystem（座標系）

- 円弧補間を行う座標系を指定します。
- 2 軸以上から構成される軸座標系(ACS)だけ対応しています。

● BufferMode（バッファモード選択）

- 1 つ前の補間動作と今回の補間動作のつなぎ方を指定します。
- 次の 6 種類があります。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令に切り替えます。 命令の切り替えにより軸の動作方向が反転する場合は、軸パラメータの［反転時動作］に従い反転動作します。
バッファード	現在実行中の命令が正常完了したのち、バッファリングされている本命令が自動的に起動します。
ブレンディング	現在実行中の命令が目標位置に到達したときの速度（中継速度）を起動速度として、バッファリングされている本命令を連続して動作させます。中継速度で目標位置に到達するように、現在実行中の命令の動作を変更します。中継速度の指定方法は、以下の 4 種類があります。 また、ブレンディングのオプションとして、トランジション指定（後述）があります。
ブレンディングロウ	現在実行中の目標速度とバッファリングされている目標速度の、どちらか低い方を中継速度とします。
ブレンディングプレビアス	現在実行中の目標速度を中継速度とします。
ブレンディングネクスト	バッファリングされている本命令の目標速度を中継速度とします。
ブレンディングハイ	現在実行中の目標速度とバッファリングされている目標速度の、どちらか高い方を中継速度とします。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

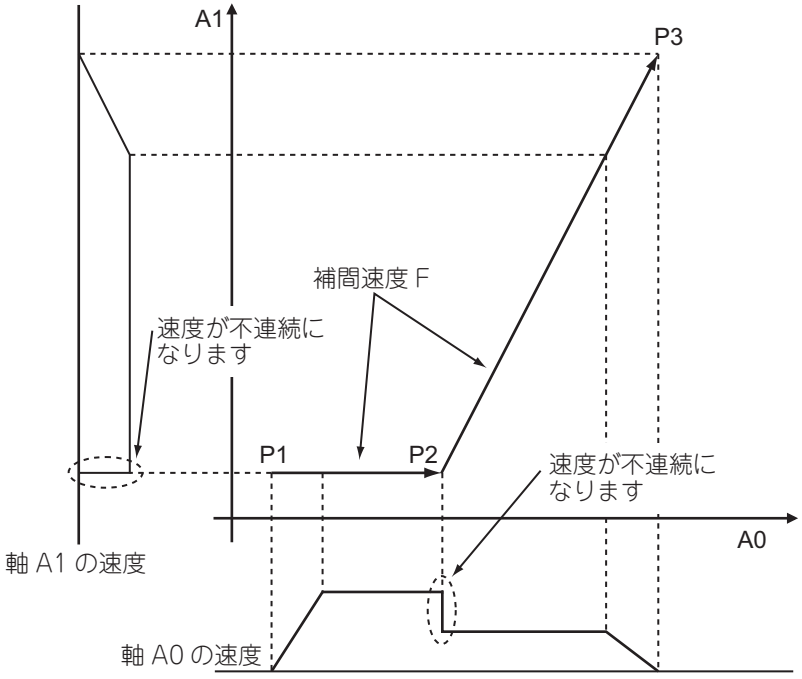
● TransitionMode（トランジションモード）

- 1 つ前の補間動作と今回の補間動作で作る軌跡のつなぎ方を指定します。
- 「TransitionMode」（トランジションモード）では、_mcTMNone（トランジション無効）、または _mcTMCornerSuperimposed（コーナースーパーインポーズ）のいずれか指定します。
- 「BufferMode」（バッファモード選択）で［ブレンディング］を指定すると、「TransitionMode」（トランジションモード）が有効になります。
- ［ブレンディング］を指定せず「TransitionMode」（トランジションモード）を_mcTMNone（トランジション無効）以外に指定した場合は異常となります。

トランジション無効

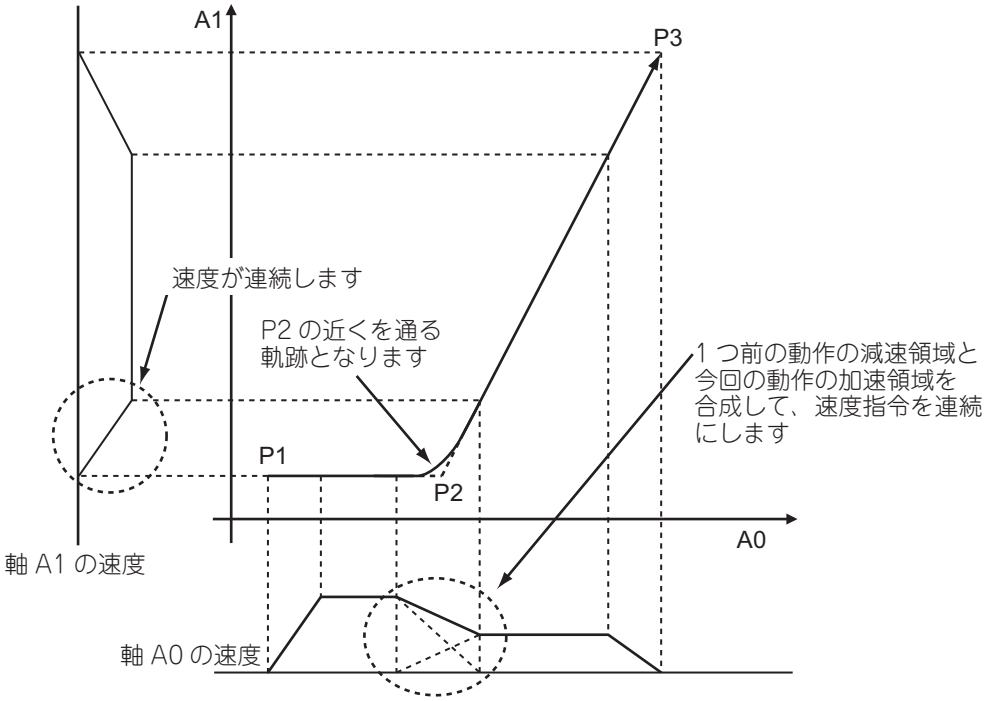
軌跡を優先して速度指令値を作成するため、各軸の速度指令値は、1 つ 1 つの動作の切れ目で急激に速度が変化することがあります。

動作例
P1 から P2、P2 から P3 への「Velocity」（目標速度）、「BufferMode」（バッファモード選択）、「TransitionMode」（トランジションモード）は次の通りです。
- P1 から P2 への動作：Velocity = F、BufferMode = アボーティング、TransitionMode = _mcTMNone（トランジション無効）
- P2 から P3 への動作：Velocity = F、BufferMode = ブレンディングネクスト、TransitionMode = _mcTMNone（トランジション無効）
- 位置 P1 から動き始め、位置 P2 を通過して、位置 P3 へ直線補間で動作します。
- 位置 P2 を通過するとき、直線補間の速度 F を維持します。このため、位置 P2 では、図に示すように、速度が不連続になります。



コーナースーパーインポーズ
各軸の指令速度を連続で動かしたい場合に使用します。

動作例
P1 から P2、P2 から P3 への「Velocity」（目標速度）、「BufferMode」（バッファモード選択）、「TransitionMode」（トランジションモード）は次の通りです。
- P1 から P2 への動作：Velocity = F、BufferMode = アボーティング、TransitionMode = _mcTMNone（トランジション無効）
- P2 から P3 への動作：Velocity = F、BufferMode = ブレンディングネクスト、TransitionMode = _mcTMCornerSuperimposed（コーナースーパーインポーズ）
- 位置 P1 から動き始め、位置 P2 の近くを通過して、位置 P3 へ直線補間で動作します。
- 各軸の指令速度を連続にするため、1 つ前の動作の減速領域と今回の動作の加速領域を合成して、指令速度を作成します。このため、今回の動作の加速時間は、1 つ前の動作の減速時間と同じになります。



合成した軌跡は、P2 の近くを通る軌跡になります。
P2 からこの軌跡への距離は、以下のようになります。

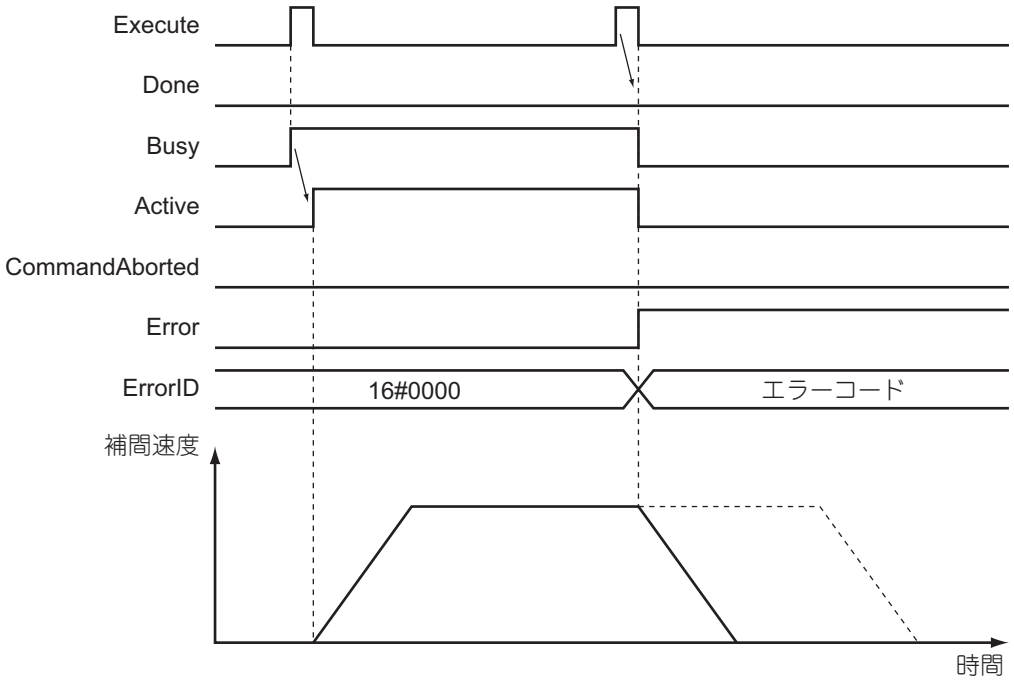
- 補間速度が速くなる、または 1 つ前の命令の減速度が小さくなると、長くなります。
- 補間速度が遅くなる、または 1 つ前の命令の減速度が大きくなると、短くなります。

参考

コーナースーパーインポーズする区間では、「Jerk」（ジャーク）の指定は無効になります。

モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となり、円弧補間動作中の全軸が停止します。



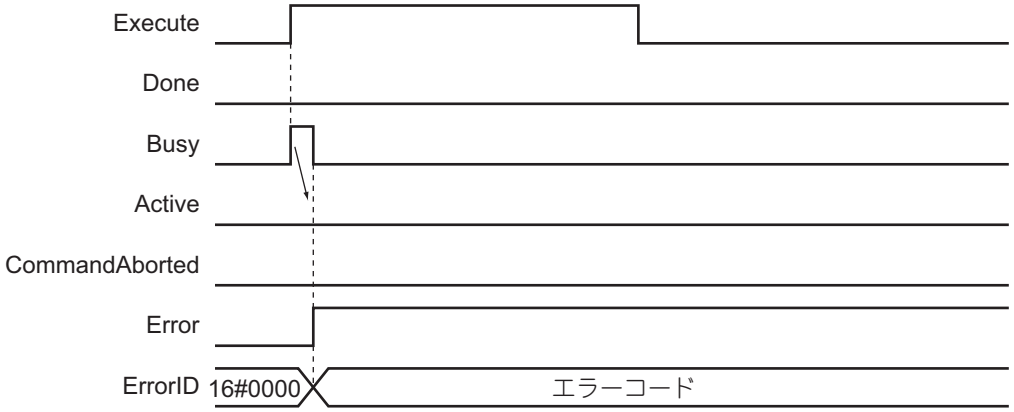
モーション命令多重起動

本命令実行中に、使用できる命令には制限があります。

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、多重起動による円弧補間についてサンプルプログラムを掲載します。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● 軸パラメータの設定

軸種別

軸	軸種別
軸 1	サーボ軸
軸 2	サーボ軸

カウンタモード

軸	カウンタモード
軸 1	リニアモード
軸 2	リニアモード

座標単位選択

軸	座標単位
軸 1	mm
軸 2	mm

● 軸グループパラメータの設定

軸構成

［2 軸］とします。

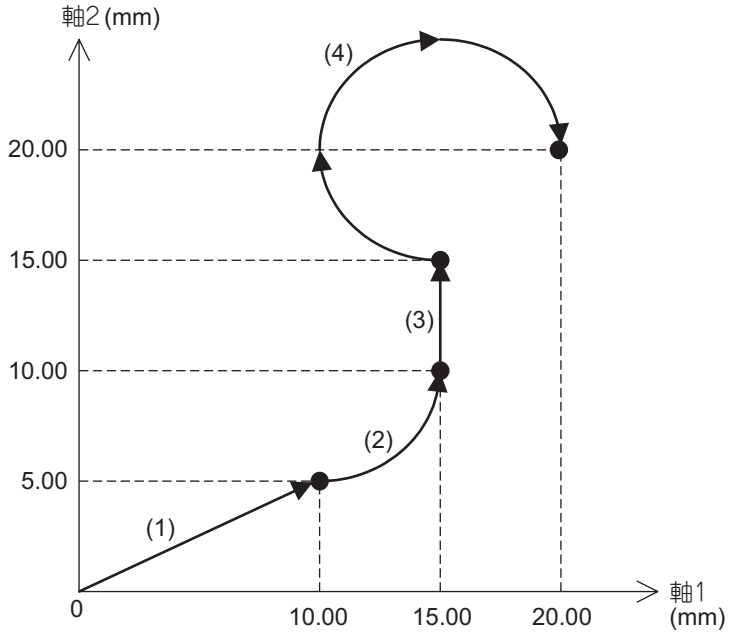
軸選択

［軸 1］、［軸 2］とします。

動作例

直線補間動作から円弧補間を自動で位置決めする動作例を以下に示します。
直線補間と円弧補間を用いて、最終目標位置(20.00mm、20.00mm)に移動します。
「バッファモード 選択」を［バッファード］に指定し、多重起動により指令します。
この例では直線補間(1)の出力「Active」が TRUE になったときに、(2)～(4)の命令多重起動を行っています。多軸協調動作では最大 7 命令まで多重起動することが可能です。

● **動作パターン**



1 **起動**

原点で動作開始スイッチを「ON」すると、直線補間動作で、(10.00mm, 5.00mm)に移動します。

2 **連続移動**

そのまま、円弧補間動作で(15.00mm, 10.00mm)へ移動し、直線補間動作で(15.00mm, 15.00mm)、さらに円弧補間動作で(20.00mm, 20.00mm)へ位置決めし停止します。
このときの速度は 10.00mm/s とします。

ラダー図

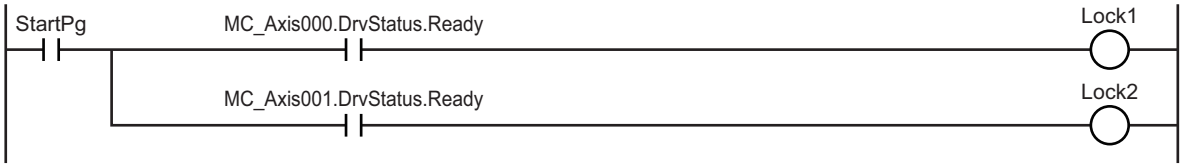
● **主な変数**

名称	データ型	初期値	コメント
MC_Group000	_sGROUP_REF	−	軸グループ 0 の軸グループ変数です。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	軸グループ 0 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Group000.Status.Disabled	BOOL	FALSE	軸グループ 0 が無効な場合に TRUE となります。
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態の場合に TRUE となります。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001	_sAXIS_REF	−	軸 2 の軸変数です。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態の場合に TRUE となります。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。

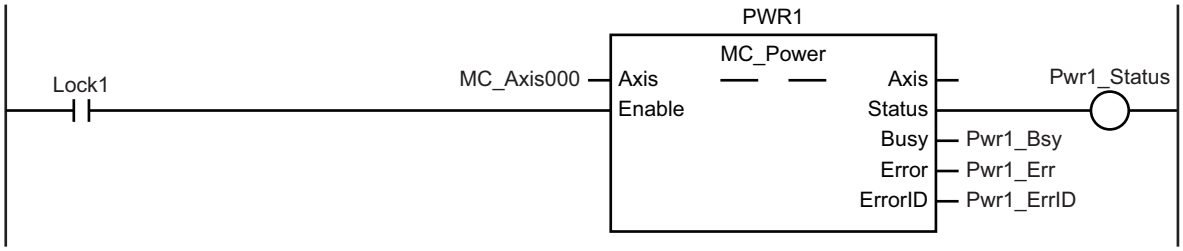
名称	データ型	初期値	コメント
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、軸グループに属する各軸をサーボ ON 状態にします。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。

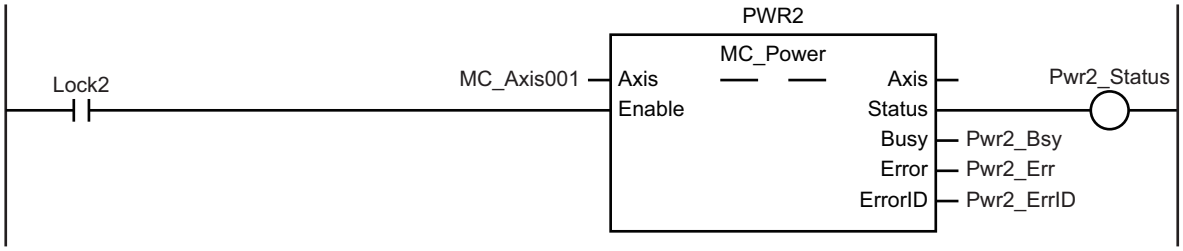
● サンプルプログラム

接点 StartPg が TRUE になったら、各軸のサーボドライバがサーボレディ状態になっていることを確認する

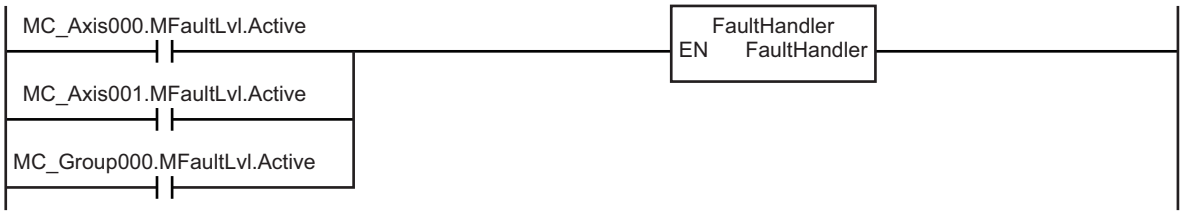


サーボレディ状態になっていれば、サーボ ON 状態にする

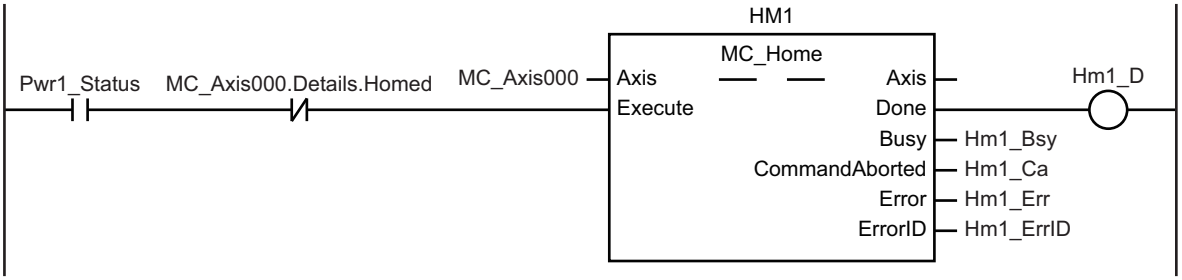




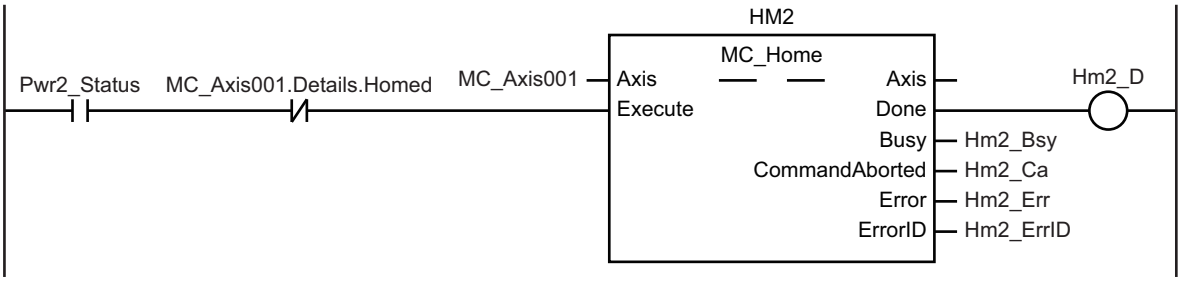
軸構成で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。
異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



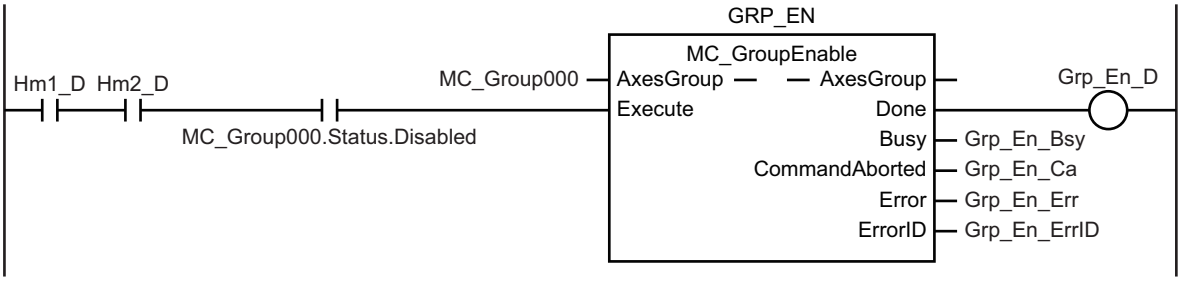
軸 1 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行する



軸 2 がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行する



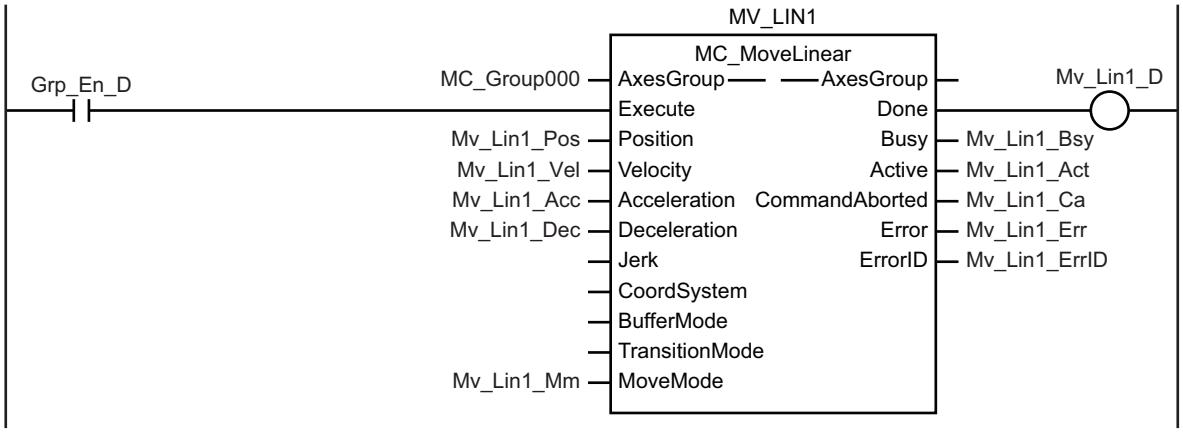
軸 1 と軸 2 の原点確定後、軸グループを有効にする



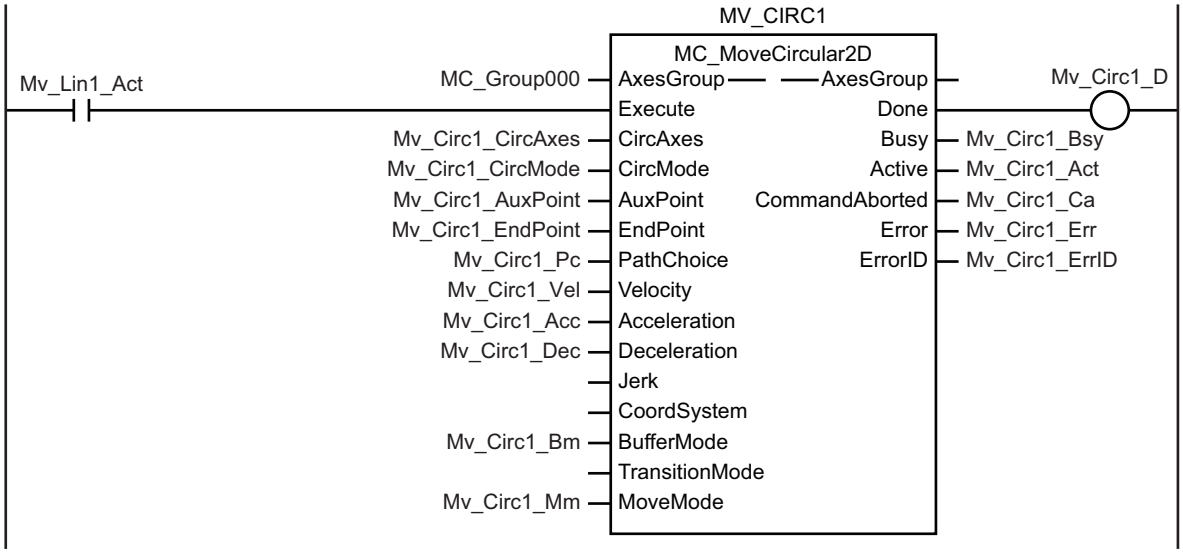
直線補間と円弧補間のパラメータを設定する



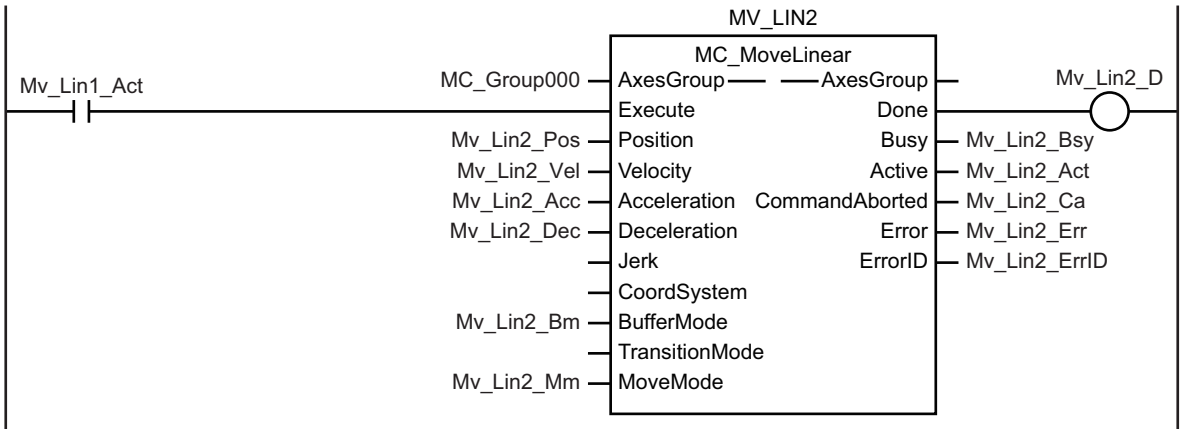
軸グループが有効になったことを確認し、直線補間 (1) を実行する



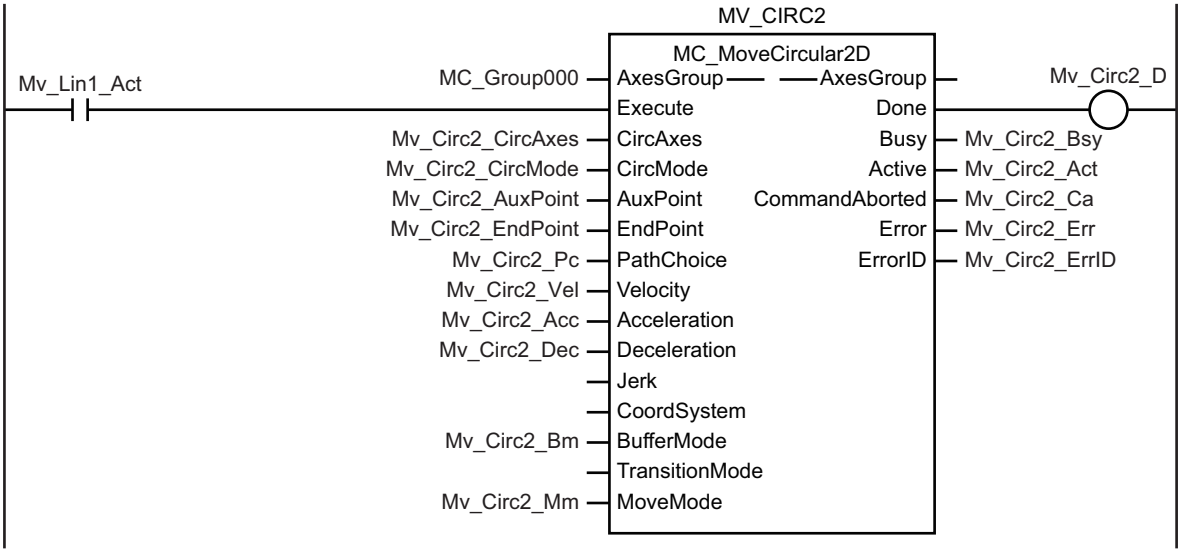
直線補間 (1) が起動後、円弧補間 (2) を多重起動する



直線補間 (1) が起動後、直線補間 (3) を多重起動する



直線補間 (1) が起動後、円弧補間 (4) を多重起動する



インライン ST の内容

```
//MV_CIRC1 パラメータ
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 パラメータ
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 パラメータ
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 パラメータ
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;

//入力パラメータを設定後、InitFlag を TRUE にします
InitFlag := TRUE;
```

ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Group000	_sGROUP_REF	−	軸グループ 0 の軸グループ変数です。
MC_Group000.MFaultLvl.Active	BOOL	FALSE	軸グループ 0 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Group000.Status.Disabled	BOOL	FALSE	軸グループ 0 が無効な場合に TRUE となります。
MC_Axis000	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態の場合に TRUE となります。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
MC_Axis001	_sAXIS_REF	−	軸 2 の軸変数です。
MC_Axis001.Details.Homed	BOOL	FALSE	軸 2 が原点確定状態の場合に TRUE となります。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 2 で軽度フォールトレベルの異常が発生した場合に TRUE となります。
Pwr1_Status	BOOL	FALSE	MC_Power のインスタンス PWR1 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power のインスタンス PWR2 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、軸グループに属する各軸をサーボ ON 状態にします。
InitFlag	BOOL	FALSE	入力パラメータの設定完了を表します。 FALSE のとき、入力パラメータを設定します。 入力パラメータの設定が完了すると、TRUE となります。
Hm1_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_Home のインスタンス HM1 を起動します。

名称	データ型	初期値	コメント
Hm2_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_Home のインスタンス HM2 を起動します。
Grp_En_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_GroupEnable のインスタンス GRP_EN を起動します。
Mv_Lin1_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveLinear のインスタンス MV_LIN1 を起動します。
Mv_Lin2_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveLinear のインスタンス MV_LIN2 を起動します。
Mv_Circ1_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveCircular のインスタンス MV_CIRC1 を起動します。
Mv_Circ2_Ex	BOOL	FALSE	この変数が FALSE→TRUE のとき、MC_MoveCircular のインスタンス MV_CIRC2 を起動します。

● サンプルプログラム

```
//入力パラメータが設定されていない場合の処理
IF InitFlag=FALSE THEN

    //MV_CIRC1 パラメータ
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 パラメータ
    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 パラメータ
    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 パラメータ
    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;

    //入力パラメータを設定後、InitFlag を TRUE にします
    InitFlag := TRUE;

END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE; //軸 1 をサーボ ON にします
ELSE
    Pwr1_En:=FALSE; //軸 1 をサーボ OFF にします
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 2 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE; //軸 2 をサーボ ON にします
ELSE
    Pwr2_En:=FALSE; //軸 2 をサーボ OFF にします
END_IF;

//軽度フォールトレベルの異常発生時の処理
//異常発生時の処理(FaultHandler)の内容は、装置にあわせてお客様でプログラミングしてください
IF (MC_Axis000.MFaultLvl.Active=TRUE) OR (MC_Axis001.MFaultLvl.Active=TRUE)
 OR (MC_Group000.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

//軸 1 がサーボロック状態で、かつ原点未確定のとき、原点復帰を行います
```

```
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//軸 2 がサーボロック状態で、かつ原点未確定のとき、原点復帰を行います
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//軸 1 と軸 2 の原点復帰が完了し、かつ軸グループ 0 が無効のとき、軸グループ 0 を有効にします
IF (Hm1_D=TRUE) AND (Hm2_D=TRUE) AND (MC_Group000.Status.Disabled=TRUE)THEN
    Grp_En_Ex:= TRUE;
END_IF;

//軸グループ 0 が有効になったら、直線補間(1)を起動します
IF Grp_En_D=TRUE THEN
    Mv_Lin1_Ex:=TRUE;
END_IF;

//直線補間(1)の出力 Active が TRUE になったら、以降の命令を多重起動します
IF Mv_Lin1_Act=TRUE THEN
    Mv_Circ1_Ex:=TRUE;
    Mv_Lin2_Ex:=TRUE;
    Mv_Circ2_Ex:=TRUE;
END_IF;

//軸 1 の MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);

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

//軸 1 の MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
```

```
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

//軸 2 の MC_Home
HM2(
    Axis := MC_Axis001,
    Execute := Hm2_Ex,
    Done => Hm2_D,
    Busy => Hm2_Bsy,
    CommandAborted => Hm2_Ca,
    Error => Hm2_Err,
    ErrorID => Hm2_ErrID
);

//軸グループ 0 有効
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
);

//直線補間(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
);

//円弧補間(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
);

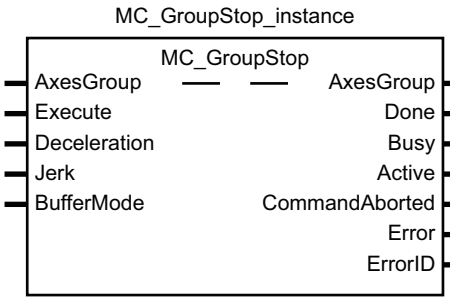
//直線補間(3)
MV_LIN2(
    AxesGroup := MC_Group000,
    Execute := Mv_Lin2_Ex,
    Position := Mv_Lin2_Pos,
    Velocity := Mv_Lin2_Vel,
    Acceleration := Mv_Lin2_Acc,
    Deceleration := Mv_Lin2_Dec,
    BufferMode := Mv_Lin2_Bm,
    MoveMode := Mv_Lin2_Mm,
    Done => Mv_Lin2_D,
    Busy => Mv_Lin2_Bsy,
    Active => Mv_Lin2_Act,
    CommandAborted => Mv_Lin2_Ca,
    Error => Mv_Lin2_Err,
    ErrorID => Mv_Lin2_ErrID
);

//円弧補間(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

補間動作中のすべての軸を減速停止します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GroupStop	軸グループ強制停止	FB		MC_GroupStop_instance (AxesGroup :=「パラメータ」, Execute :=「パラメータ」, Deceleration :=「パラメータ」, Jerk :=「パラメータ」, BufferMode :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Deceleration	減速度	LREAL	正数、または「0」	0	減速度を指定します。 単位は［指令単位／s^2］となります。*1
Jerk	ジャーク	LREAL	正数、または「0」	0	ジャークを指定します。 単位は［指令単位／s^3］となります。*1
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *2	モーション命令多重起動時の動作を指定します。 0: アボーティング

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。

出力変数	名称	データ型	有効範囲	内容
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。 • 異常発生中に、本命令を起動したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	−	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 本命令は、軸グループ指令で動作中の「AxesGroup」（軸グループ）で指定した軸グループに属するすべての軸を停止させ、軸グループ指令を無効状態とします。
  軸を停止するときに「Deceleration」（減速度）で減速停止する命令は、MC_MoveLinear（直線補間）命令、MC_MoveLinearAbsolute（絶対値直線補間）命令、MC_MoveLinearRelative（相対値直線補間）命令、MC_MoveCircular2D（2 軸円弧補間）命令です。
  軸を停止するときに「Deceleration」（減速度）の設定値に関係なく即停止する命令は MC_GroupSyncMoveAbsolute（軸グループサイクリック同期絶対位置制御）命令です。
- 本命令の起動により補間動作中の命令は「CommandAborted」（実行中断）します。

- 補間命令を実行している場合に本命令を実行すると、直線補間、または円弧補間の軌跡上に減速停止します。
- 「Execute」（起動）の立ち上がりで、減速停止の動作を開始します。

使用上の注意

軸グループ変数の「Status.ErrorStop」（エラー減速停止中）が TRUE の場合、本命令は起動できません。
エラー減速停止中に軸グループの動作を停止させるには、MC_GroupImmediateStop（即停止）命令を使用してください。

命令の詳細

以下に本命令の詳細について説明します。

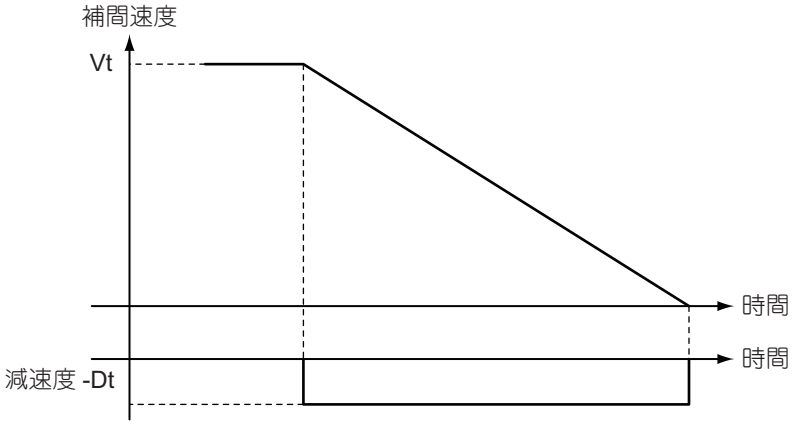
● Deceleration（減速度）、Jerk（ジャーク）

減速停止時の減速度、ジャークは、入力変数の「Deceleration」（減速度）、「Jerk」（ジャーク）で指定します。
本命令を実行した場合、補間速度に対して、ここで指定した「Deceleration」（減速度）、「Jerk」（ジャーク）を使用します。

Jerk（ジャーク）

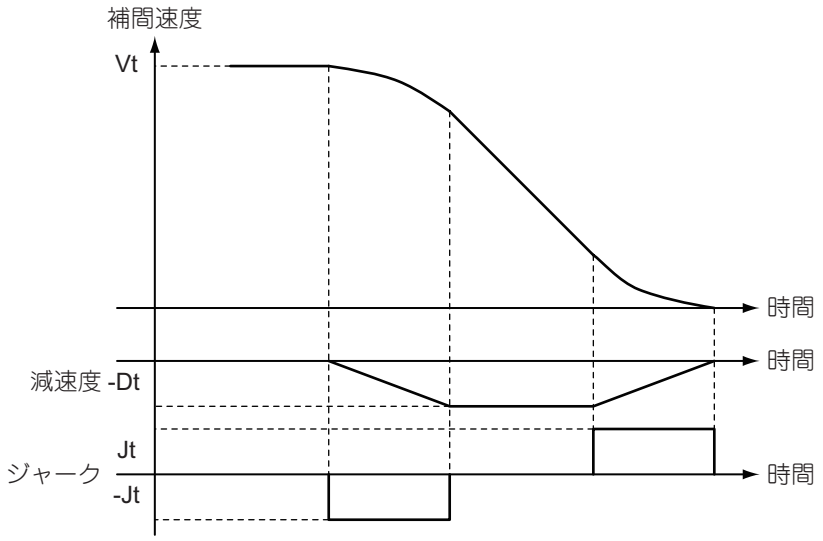
「Jerk」（ジャーク）に「0」を指定した場合と「0」以外を指定した場合の減速度、補間速度の関係は次のようになります。

- 「Jerk」（ジャーク）が「0」の場合
  減速度 Dt で、速度の指令値を作成します。



Vt: 減速起動時の速度、Dt: 減速度の指定値

- 「Jerk」（ジャーク）が「0」以外の場合
  減速度の上限は Dt として現在速度を基点とし、速度の指令値を作成します。



Vt: 減速起動時の速度、Dt: 減速度の指定値、Jt: ジャークの指定値

参考

- 「Deceleration」（減速度）に「0」を指定すると、即停止の命令となり、すぐに停止します。
- 「Deceleration」（減速度）に「0」を指定した場合だけ、軸パラメータの［加減速オーバー］の設定にかかわらず、即停止を実行します。

● BufferMode（バッファモード選択）

1 つ前の補間動作と今回の補間動作のつなぎ方を指定します。
次の 1 種類があります。

バッファモード選択	説明
アボーティング	現在実行中の命令を直ちに中止して、本命令を実行します。

「BufferMode」（バッファモード選択）については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

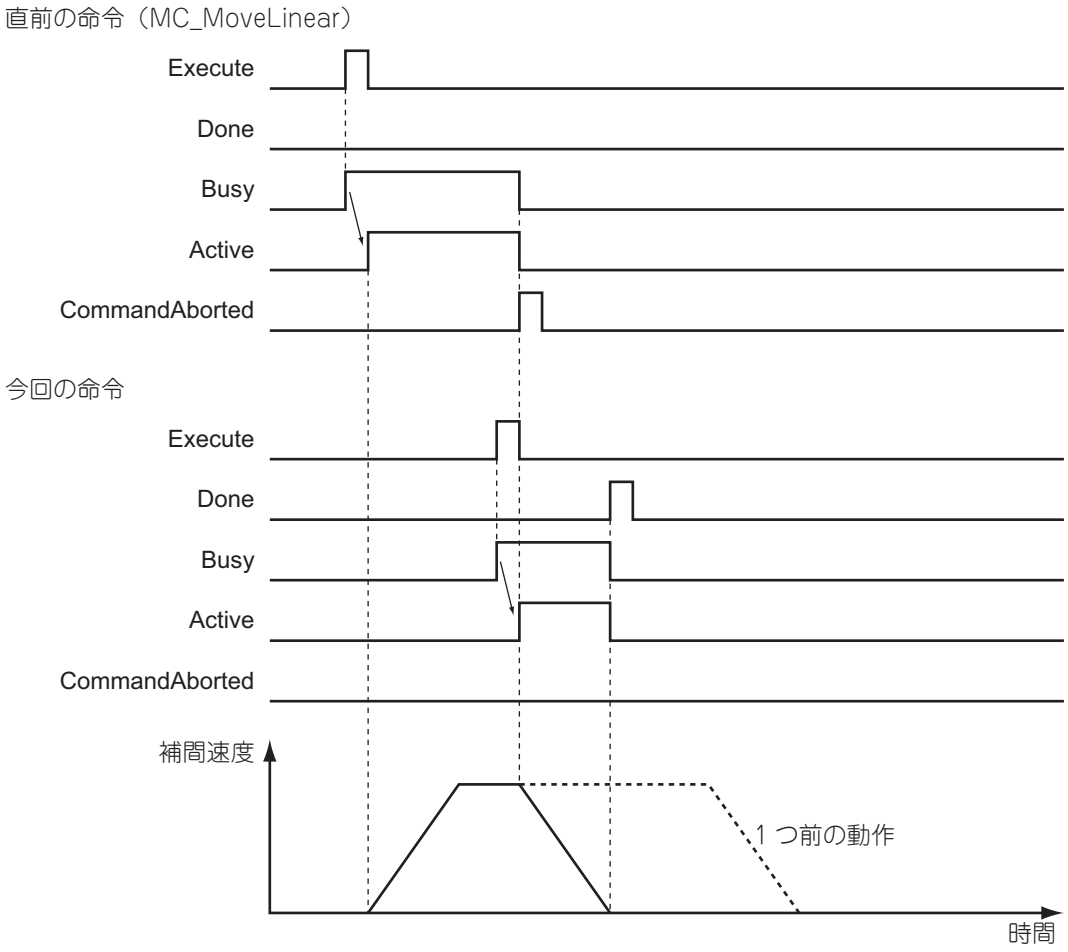
● インポジションチェック

本命令による停止時に、インポジションチェックは行いません。

タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- 速度「0」に到達すると、「Done」（完了）が TRUE となります。
- ほかの命令により、本命令が中断された場合は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）が FALSE となります。

直線補間中の減速停止のタイミングチャートを以下に記します。
本命令の起動により補間動作中の命令は「CommandAborted」（実行中断）します。

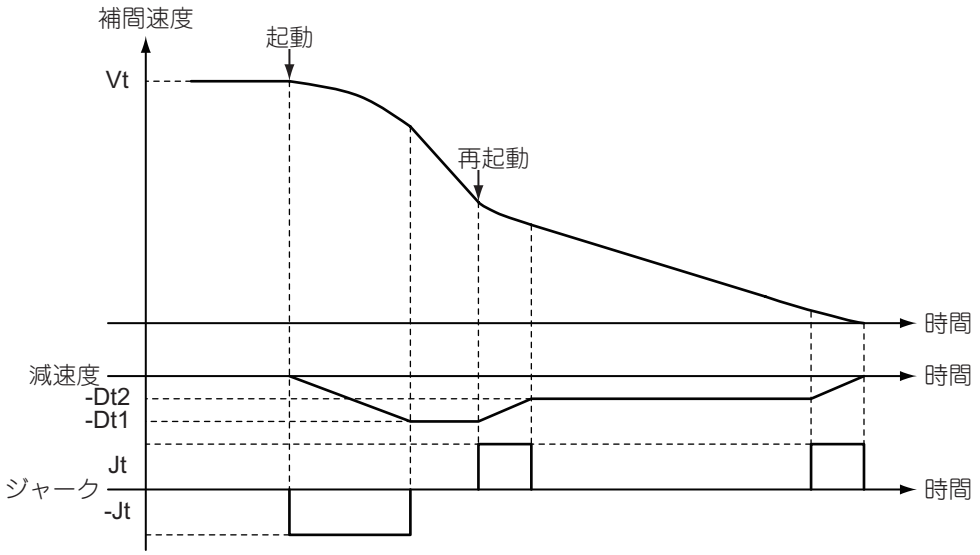


モーション命令再起動

本命令実行中に、再度「Execute」（起動）を立ち上げると、減速度の変更を行います。
モーション命令再起動では「Jerk」（ジャーク）の変更は行いません。

●「Jerk」（ジャーク）が「0」以外の場合

現在速度、現在減速度を基点とし、変更後の減速度の上限を Dt2 として速度の指令値を作成します。



Vt: 減速起動時の速度、Dt: 減速度の指定値、Jt: ジャークの指定値

モーション命令多重起動

本命令の実行、および本命令実行中に使用できる状態、命令には制限があります。

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

軸グループ無効中

無効の軸グループに、MC_GroupStop（軸グループ強制停止）命令を実行すると軸グループは異常となります。軸への影響はありません。

軸グループ変数の「Status.Stopping」（減速停止中）が TRUE のとき

下記の状態の場合、軸変数の「Status.Stopping」（減速停止中）が TRUE となります。

- MC_GroupStop（軸グループ強制停止）命令による軸グループの減速中
- MC_GroupStop（軸グループ強制停止）命令の「Execute」が多重起動により、1 つでも TRUE となっている間

減速停止中状態の軸グループに対し、以下の命令を起動した場合、起動した命令の「CommandAborted」（実行中断）が TRUE となります。

- MC_MoveLinear（直線補間）命令
- MC_MoveLinearAbsolute（絶対値直線補間）命令
- MC_MoveLinearRelative（相対値直線補間）命令
- MC_MoveCircular2D（2 軸円弧補間）命令

MC_GroupStop（軸グループ強制停止）命令を実行しているとき、MC_GroupStop（軸グループ強制停止）命令の多重起動が可能です。実行中の MC_GroupStop（軸グループ強制停止）命令は「Done」（完了）が TRUE となります。

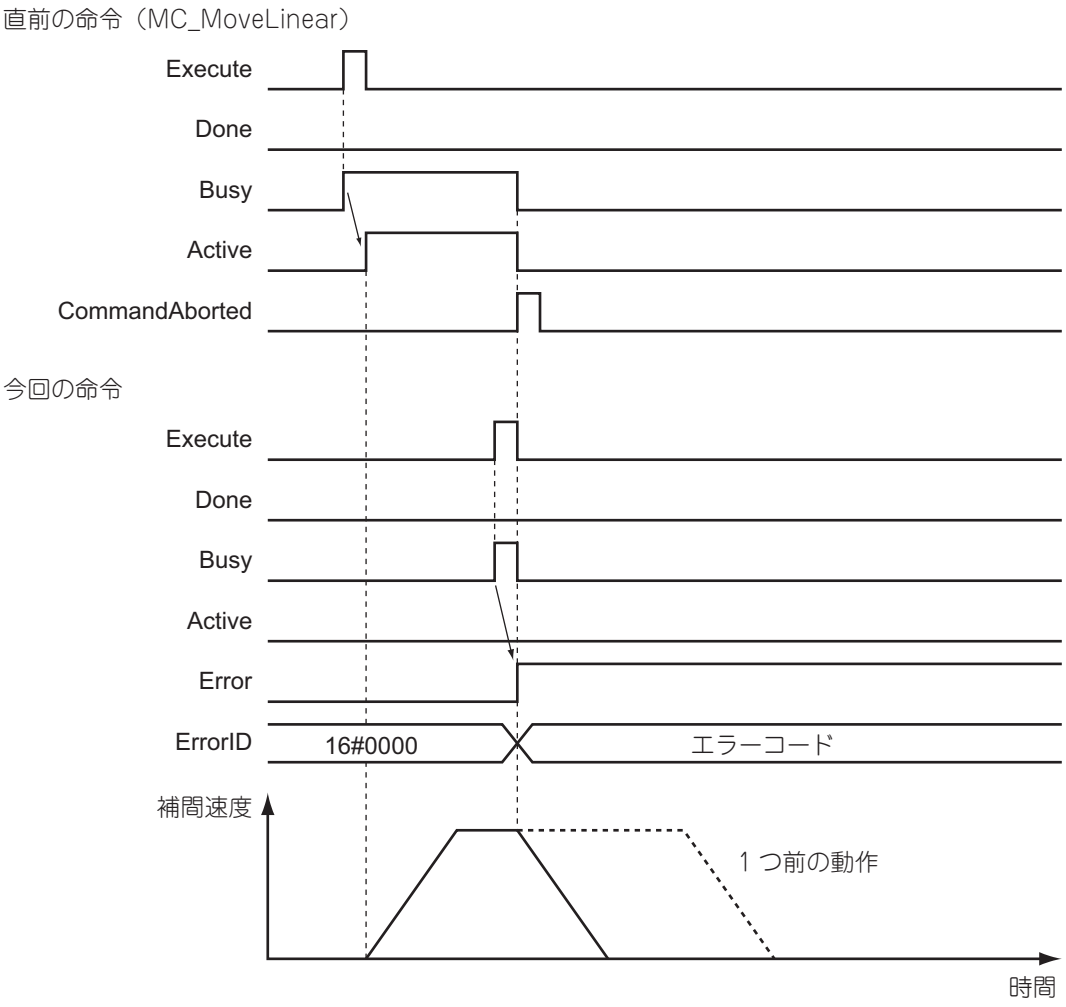
軸グループ変数の「Status.ErrorStop」（エラー減速停止中）が TRUE のとき

軸グループで異常が発生している間、軸グループは「Status.ErrorStop」（エラー減速停止中）が TRUE となります。

「Status.ErrorStop」（エラー減速停止中）が TRUE の場合に MC_GroupStop（軸グループ強制停止）命令を起動すると、「CommandAborted」（実行中断）が TRUE となります。
この場合、MC_GroupImmediateStop（軸グループ即停止）命令を使用してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

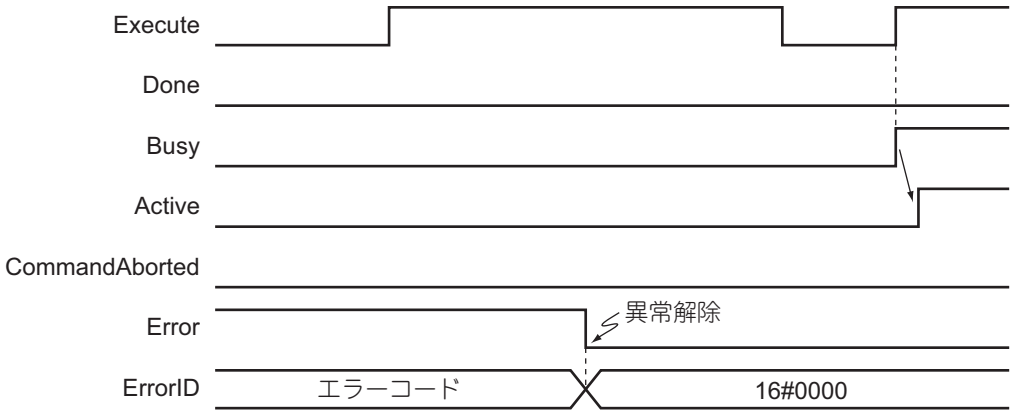


バージョン関連情報

異常解除時の動作は、CPU ユニットのバージョンにより、次のようになります。
なお、異常の解除は軸グループが停止してから行ってください。軸グループ動作中は異常の解除ができません。

- **ユニットバージョン Ver.1.10 以降の CPU ユニットの場合**
  本命令の異常解除を行った場合、次の「Execute」（起動）の立ち上がりまで本命令は起動しません。

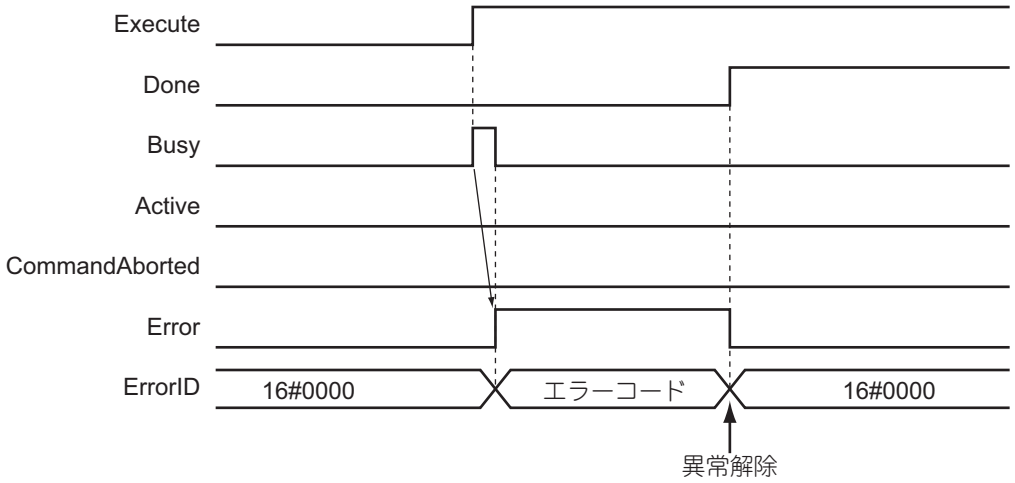
Ver.1.10 以降



- **ユニットバージョン Ver.1.09 以前の CPU ユニットの場合**
  本命令で異常が発生している状態で、「Execute」（起動）が TRUE のまま異常解除を行う場合、以下のような動作となります。
  a）異常要因が解除されていた場合、「Error」（エラー）が FALSE、「Done」（完了）が TRUE となります。軸変数の「Status.Stopping」（減速停止中）は、減速停止の正常実行時と同様に TRUE となります。
  b）異常要因が解除されていない場合、本命令は再び「Error」（エラー）になり、軸グループに異常が発生します。

  以下のタイミングチャートは、異常要因を解除している場合です。

Ver.1.09 以前



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_GroupImmediateStop

補間動作中のすべての軸が即停止します。
対象軸グループが有効となっていれば軸の状態によらず軸パラメータの［即停止入力停止方法］の停止方法に従って構成軸のすべての軸が停止します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GroupImmediateStop	軸グループ即停止	FB	MC_GroupImmediateStop_instance MC_GroupImmediateStop AxesGroup ── ── AxesGroup Execute — Done Busy CommandAborted Error ErrorID	MC_GroupImmediateStop_instance (AxesGroup :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	即停止が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後

変数	TRUE となるタイミング	FALSE となるタイミング
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの MC_GroupStop（軸グループ強制停止）命令によるモーション命令多重起動（[アボーティング]）で、本命令が中止されたとき。 • 異常発生により、本命令が中止されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	−	軸グループを指定します。*1

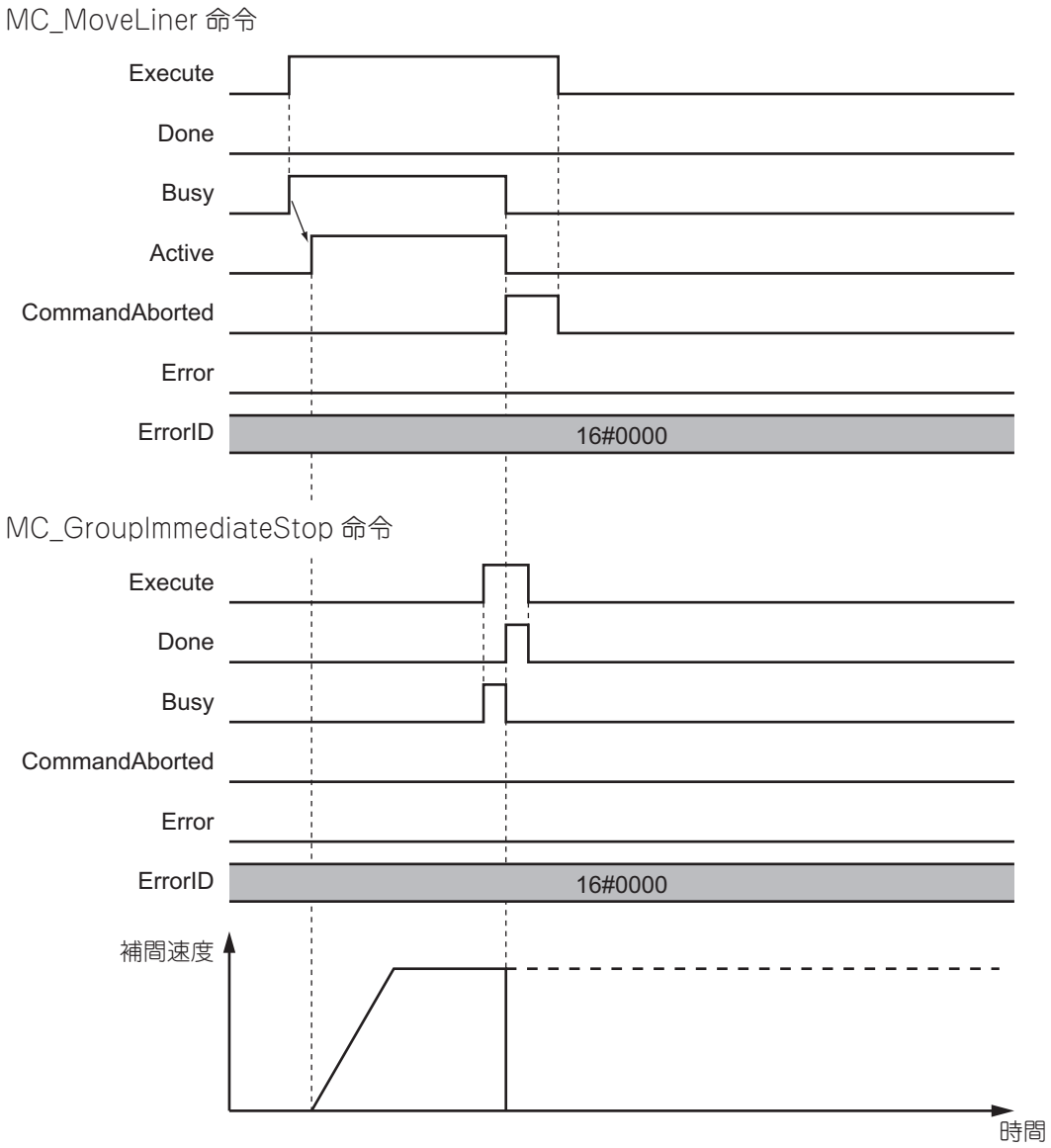
1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 本命令は、軸グループ指令で動作中の「AxesGroup」（軸グループ）で指定した軸グループに属するすべての軸、および MC_GroupStop（軸グループ強制停止）命令、もしくは異常発生による停止状態の軸グループに対して実行可能です。
- 命令が実行されると、各軸のパラメータ設定［即停止入力停止方法］の指定に従い即停止し、動作中の命令は「CommandAborted」（実行中断）となります。
- 本命令を実行すると、軸グループの状態は「Status.ErrorStop」（エラー減速停止中）が TRUE となり、「軸グループ即停止命令起動（エラーコード: 5486 Hex）」が発生します。
「軸グループ即停止命令起動（エラーコード: 5486 Hex）」については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

タイミングチャート

- 「Execute」（起動）の立ち上がりと同時に「Busy」（実行中）が TRUE となります。
- 本命令の処理が完了すると、「Done」（完了）が TRUE となります。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex)」となり、この軸の動作が停止します。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

軸グループ無効中

無効の軸グループに、本命令を実行すると軸グループは異常となります。軸への影響はありません。

軸グループ変数の「Status.Stopping」（減速停止中）が TRUE のとき

下記の状態の場合、軸グループ変数の「Status.Stopping」（減速停止中）が TRUE となります。
- MC_GroupStop（軸グループ強制停止）命令による軸グループの減速中
- MC_GroupStop（軸グループ強制停止）命令の「Execute」が多重起動により 1 つでも TRUE となっている間

減速停止中状態の軸グループに対しても本命令による多重起動が可能です。
本命令の起動により、実行中の MC_GroupStop（軸グループ強制停止）命令の「CommandAborted」（実行中断）が TRUE となります。

軸グループ変数の「Status.ErrorStop」（エラー減速停止中）が TRUE のとき
軸グループで異常が発生している間、軸グループの状態は「Status.ErrorStop」（エラー減速停止中）が TRUE となります。
軸グループエラー減速停止中でも本命令は実行可能となります。

異常

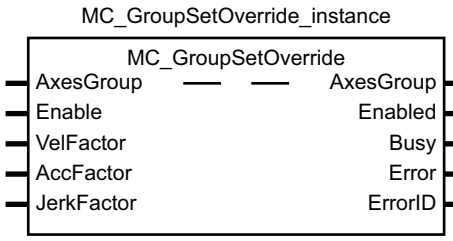
本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_GroupSetOverride

補間動作中の合成目標速度を変更します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GroupSetOverride	軸グループオーバーライド値設定	FB		MC_GroupSetOverride_instance (AxesGroup :=「パラメータ」, Enable :=「パラメータ」, VelFactor :=「パラメータ」, AccFactor :=「パラメータ」, JerkFactor :=「パラメータ」, Enabled =>「パラメータ」, Busy =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	TRUE にするとオーバーライドが有効になります。 FALSE にするとオーバーライドは「100%」に戻ります。
VelFactor	速度オーバーライド値	LREAL	0～500	100	速度のオーバーライド値を指定します。 オーバーライド値の有効範囲は「0.01～500.00」となります。 「500.00 以上」は「500」として扱われ、「0.01 未満（負数も含む）」は「0.01」として扱われます。 「0」を指定したときだけ「0」として動作します。 単位は[%]となります。
AccFactor (Reserved)	加減速度オーバーライド値	LREAL	0～500	100	(Reserved)
JerkFactor (Reserved)	ジャークオーバーライド値	LREAL	0～500	100	(Reserved)

出力変数

出力変数	名称	データ型	有効範囲	内容
Enabled	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。

出力変数	名称	データ型	有効範囲	内容
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Enabled	「Enable」が TRUE となったとき。	• 「Enable」が FALSE の場合、1 周期後 • 「Error」が TRUE となったとき。
Busy	「Enable」の立ち上がり	• 「Error」が TRUE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	−	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 次の命令で動かしているグループの補間目標速度に対するオーバーライドを変更します。
オーバーライドを変更することにより動作中の目標速度を変更します。
対象となる命令は、以下の命令となります。

MC_MoveLinear（直線補間）命令	MC_MoveLinearRelative（相対値直線補間）命令
MC_MoveLinearAbsolute（絶対値直線補間）命令	MC_MoveCircular2D（2 軸円弧補間）命令

- 新たな目標速度は、次の通りです。
変更後の目標速度 = 現在実行中の命令の補間速度 × オーバーライド[%]
- オーバーライド単位は、[%]です。「100」で「100%」を示します。
- 変更後の補間速度が、軸グループ変数の最高補間速度をオーバーする場合は、最高補間速度になります。
- 変更した補間速度に対して、軸を加速、または減速します。
- オーバーライドに「0」を指定した場合、目標速度「0」となり、グループの動作は減速停止し、速度「0」で動作中となります。動作中は軸グループ変数の「Status.Moving」（動作中）が TRUE となります。
動作状態を継続しながら軸の動作を一時停止したい場合にオーバーライドを「0」とします。
- 「Enable」（有効）が FALSE の場合、オーバーライドは 100(=100%)に戻ります。
- MC_GroupSetOverride（軸グループオーバーライド値設定）命令実行中に、グループ異常が発生した場合、MC_GroupSetOverride（軸グループオーバーライド値設定）命令は「Enabled」（有効）を継続します。

使用上の注意

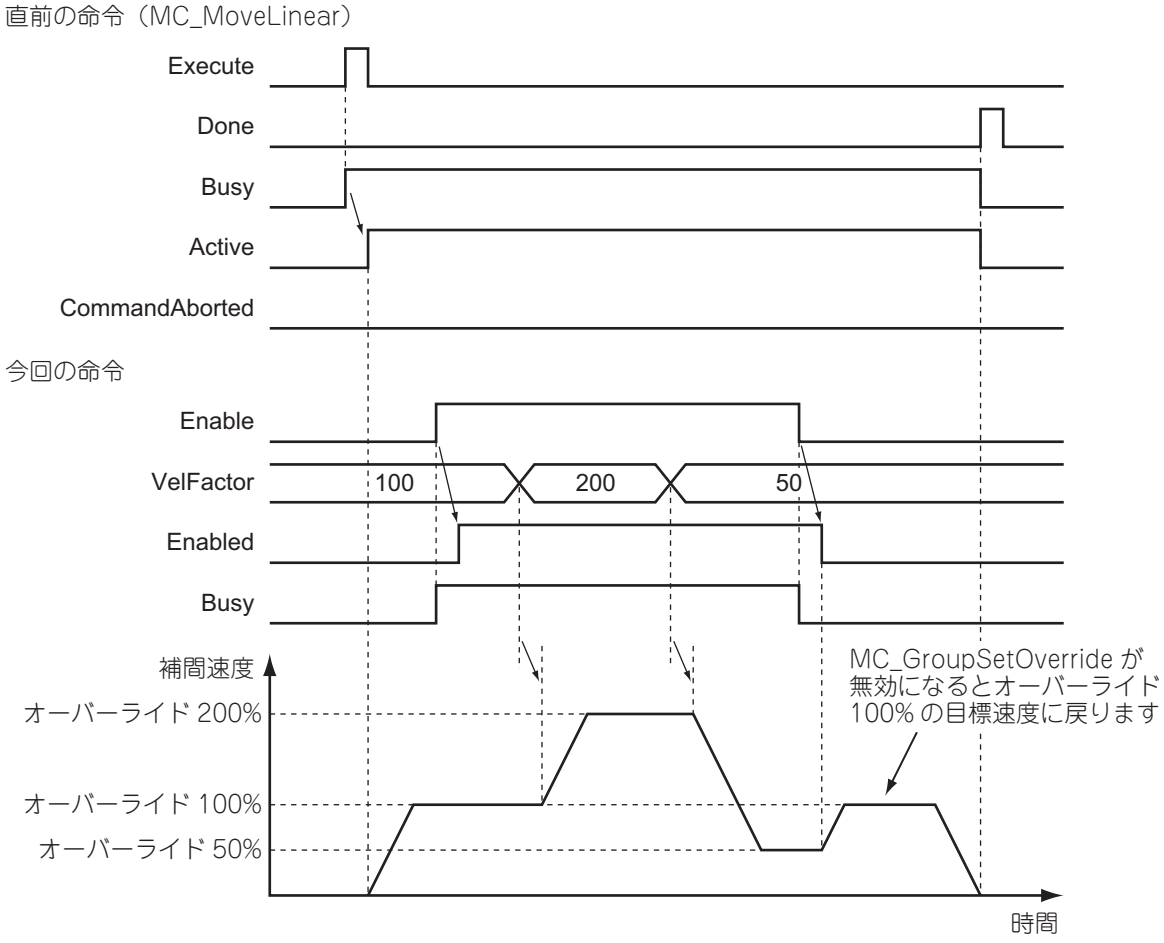
本命令の「Enable」（有効）を FALSE にすると、本命令の「Enabled」（有効）および「Busy」（実行中）は FALSE になります。
その際、速度はオーバーライド「100%」に向かって加速、もしくは減速を行います。

参考

他命令への影響
本命令は、各命令の目標速度を一時的に変更する場合に使用します。
そのため、目標速度を入力に持たない命令に対しては、本命令は機能しません。
ただし、本命令が機能しない命令に対して MC_GroupSetOverride（軸グループオーバーライド値設定）命令を有効にした場合でも、「Enabled」（有効）は TRUE のまま継続します。

タイミングチャート

● MC_MoveLinear（直線補間）命令に本命令を使用した場合



モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

モーション命令多重起動

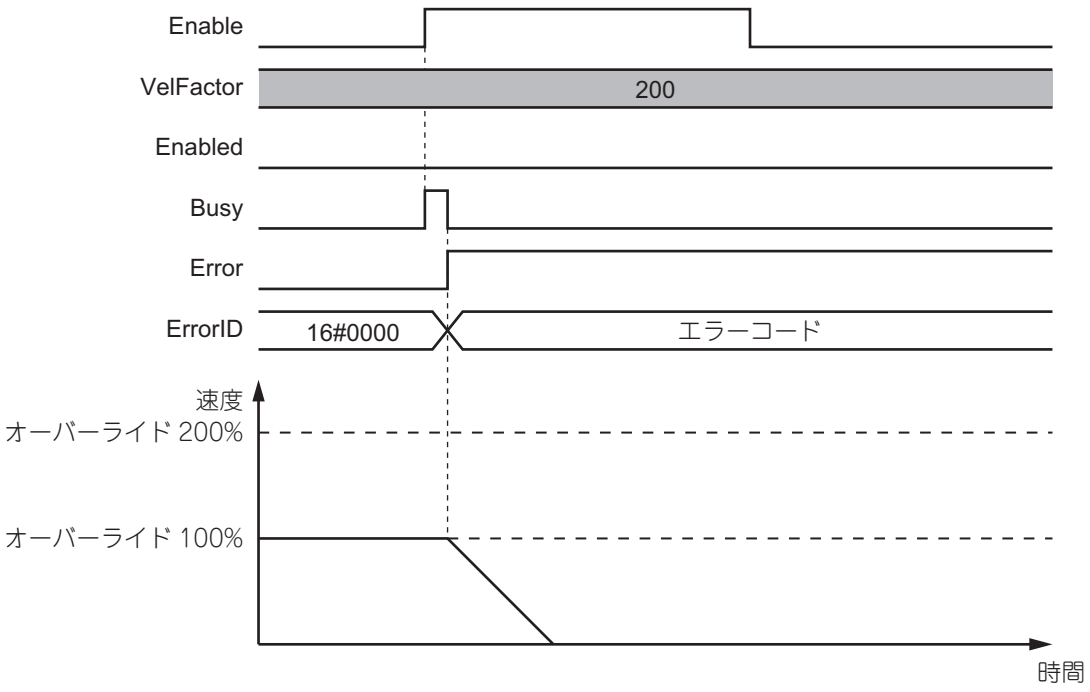
モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

MC_GroupSetOverride（軸グループオーバーライド値設定）命令実行中に、同じ軸グループに対して、ほかのインスタンスの MC_GroupSetOverride（軸グループオーバーライド値設定）命令を起動した場合、後から実行されたインスタンスが優先して処理されます。
「Enabled」（有効）は両方とも TRUE となります。
具体的な動作としては、後から実行されたインスタンスのオーバーライド値が有効となります。後から実行されたインスタンスの「Enable」（有効）を FALSE にすると、オーバーライドは無効となります。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_GroupReadPosition

軸グループの指令現在位置とフィードバック現在位置を取得します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GroupReadPosition	軸グループ位置取得	FB	MC_GroupReadPosition_instance MC_GroupReadPosition AxesGroup — AxesGroup Enable — Valid CoordSystem — Busy Error ErrorID CommandPosition ActualPosition	MC_GroupReadPosition_instance (AxesGroup :=「パラメータ」, Enable :=「パラメータ」, CoordSystem :=「パラメータ」, Valid =>「パラメータ」, Busy =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」, CommandPosition =>「パラメータ」, ActualPosition =>「パラメータ」);

バージョン関連情報

本命令は、ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Enable	有効	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
CoordSystem	座標系	_eMC_COORD_SYSTEM	0: _mcACS	0 *1	座標系を指定します。 0: 軸座標系(ACS)

*1. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
Valid	有効	BOOL	TRUE, FALSE	制御中に TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。
CommandPosition	指令現在位置	ARRAY [0..3] OF LREAL	負数、正数、「0」	指令位置の現在値を出力します。 単位は、［指令単位］となります。*2

出力変数	名称	データ型	有効範囲	内容
ActualPosition	フィードバック現在位置	ARRAY [0..3] OF LREAL	負数、正数、「0」	フィードバック位置の現在値を出力します。 単位は、［指令単位］となります。*2

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。
*2. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

（注）1. 軸グループの軸構成が 2 軸または 3 軸構成のとき、空きの構成軸の「CommandPosition」（指令現在位置）や「ActualPosition」（フィードバック現在位置）の値は不定となります。
（注）2. 「Enable」（有効）が FALSE になると「CommandPosition」（指令現在位置）や「ActualPosition」（フィードバック現在位置）の値は不定となります。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Valid	「Enable」の立ち上がり	• 「Enable」が FALSE となったとき。 • 「Error」が TRUE となったとき。
Busy	「Enable」の立ち上がり	「Error」が TRUE となったとき。
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	－	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 「Valid」（有効）が TRUE のとき、軸グループの構成軸の指令現在位置とフィードバック現在位置を制御周期ごとに取得します。
- 軸種別ごとの使用可能な現在位置の種類は以下のとおりです。

軸種別	現在位置の種類	
	指令現在位置	フィードバック現在位置
サーボ軸	使用可能	使用可能
仮想サーボ軸	使用可能	使用可能 *1
エンコーダ軸	使用不可	使用可能
仮想エンコーダ軸	使用不可	使用可能

*1. 仮想サーボ軸では、「フィードバック位置 ＝ 指令位置」となります。
MC 機能モジュール内部では倍精度実数型で処理しているため、演算誤差を含む場合があります。

- 原点未確定でも本命令を実行することができます。
- 軸グループ有効、無効に関係なく、本命令を使用できます。

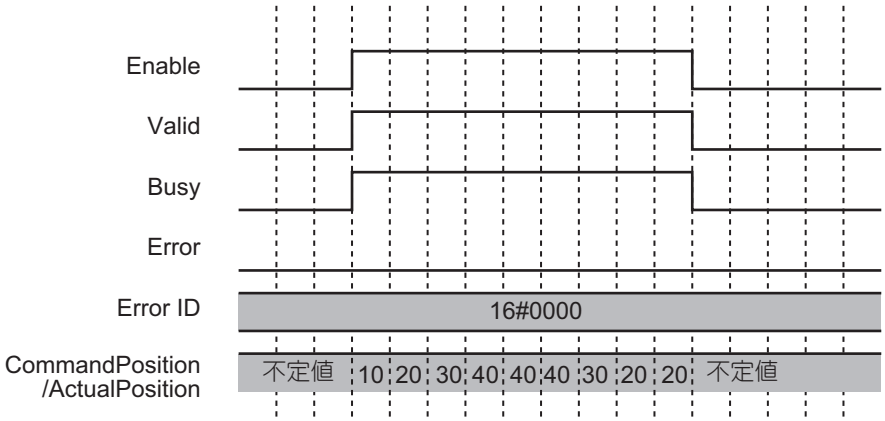
使用上の注意

軸グループを割り付けたタスク以外で本命令を使用する場合は、使用するタスクで「Axes」（軸グループ構成軸）で指定した構成軸に該当する軸変数を、外部変数宣言してから使用してください。

タイミングチャート

- 「Enable」（有効）を TRUE にすると、「Busy」（実行中）と「Valid」（有効）が TRUE となります。
- 「Enable」（有効）を FALSE にすると、「Busy」（実行中）と「Valid」（有効）は FALSE となります。

MC_GroupReadPosition



モーション命令再起動

入力が「Enable 型」の命令にはモーション命令再起動はありません。

モーション命令多重起動

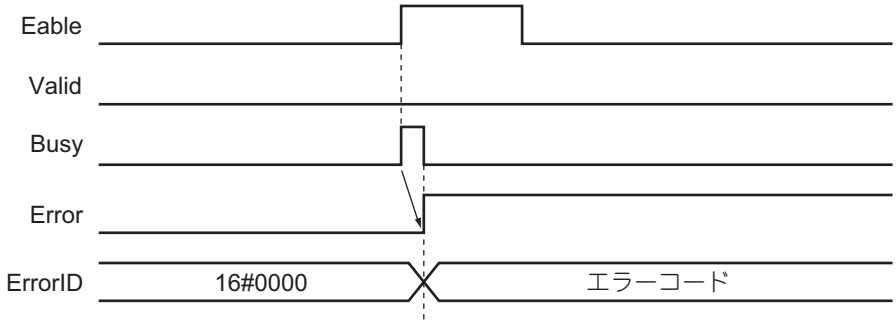
本命令は個々の命令インスタンスごとに独立して動作するため、モーション命令多重起動の制約はありません。

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

MC_GroupReadPosition



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_ChangeAxesInGroup

軸グループパラメータの［構成軸］を一時的に書き換えます。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_ChangeAxesInGroup	軸グループ構成軸書込	FB	MC_ChangeAxesInGroup_instance MC_ChangeAxesInGroup AxesGroup — — AxesGroup Axes — — Axes Execute — Done Busy CommandAborted Error ErrorID	MC_ChangeAxesInGroup_instance (AxesGroup :=「パラメータ」, Axes :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

使用上の注意

本命令で書き換えた値は CPU ユニットの不揮発性メモリに保存されません。そのため、コントローラの電源を OFF、およびダウンロード、MC 機能モジュールの再起動処理が開始されると、書き換えた値は消失し、Sysmac Studio で設定した値に戻ります。
不揮発性メモリに保存する場合は、Sysmac Studio を使って、パラメータの転送を行ってください。

参考

「ダウンロード」は、Sysmac Studio の「同期」機能を使用します。

バージョン関連情報

本命令は、ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。

出力変数	名称	データ型	有効範囲	内容
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	ほかの命令で異常が発生し、本命令が中止されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

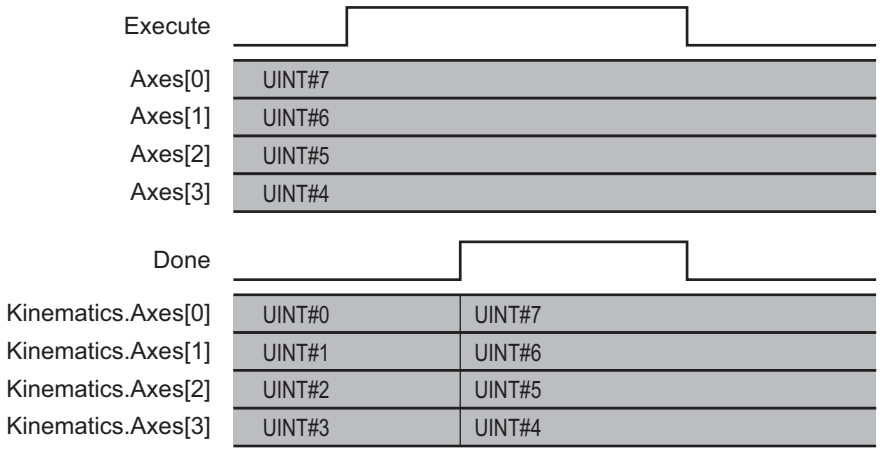
入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	−	書き換え対象の軸グループを指定します。*1
Axes	軸グループ構成軸	ARRAY [0..3] OF UINT	−	書き換え後の構成軸の軸番号を指定します。*2

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. 構成軸にはサーボ軸または仮想サーボ軸を設定してください。また、書き換え後の構成軸数は書き込み前の構成軸数と同一にしてください。

機能説明

- 「Execute」（起動）の立ち上がりで、「AxesGroup」（軸グループ）で指定した軸グループの軸グループパラメータに、「Axes」（軸グループ構成軸）で指定した構成軸を書き込みます。
- 命令の実行が完了すると、軸グループ変数の「Kinematics」（運動学変換設定）の「Axis[ ]」が指定した構成軸に書き換わっています。
- この命令は軸構成（構成軸の数）を書き換えません。たとえば軸構成が 3 軸の軸グループに本命令を実行しても軸構成は 3 軸のままです。
- 構成軸の軸番号を「0, 1, 2, 3」から「7, 6, 5, 4」に変更するときの動作は、次のようになります。



- 原点の確定、未確定に関係なく本命令を実行できます。
- 軸グループが無効のときだけ本命令を実行できます。

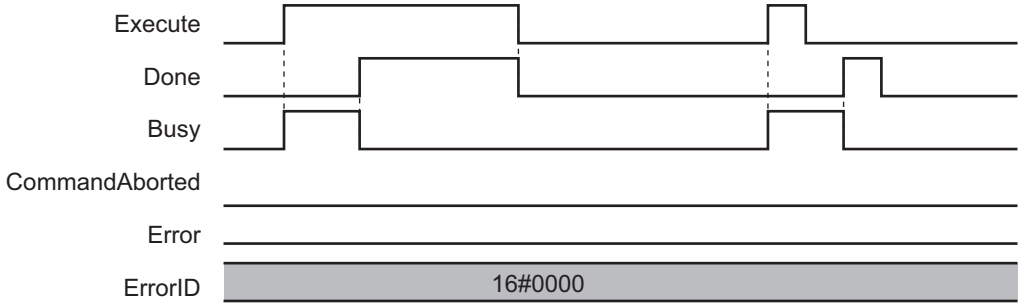
使用上の注意

- 本命令で書き換えた値は CPU ユニットの不揮発性メモリに保存されません。そのため、コントローラの電源を OFF、およびダウンロード、MC 機能モジュールの再起動処理が開始されると、書き換えた値は消失し、Sysmac Studio で設定した値に戻ります。
  不揮発性メモリに保存する場合は、Sysmac Studio を使って、パラメータの転送を行ってください。
- NX シリーズ CPU ユニットを使用する場合、「Axes」（軸グループ構成軸）で指定したすべての構成軸が、命令を実行する対象の軸グループと同じタスクに割り付けられている必要があります。異なるタスクに割り付けられている軸を指定した場合、「軸グループ構成軸設定範囲外（エラーコード: 549D Hex）」となります。

タイミングチャート

- 「Execute」（起動）を TRUE にすると、「Busy」（実行中）が TRUE となります。
- 構成軸の書き換えが完了すると「Done」（完了）が TRUE となります。

MC_ChangeAxesInGroup 命令



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

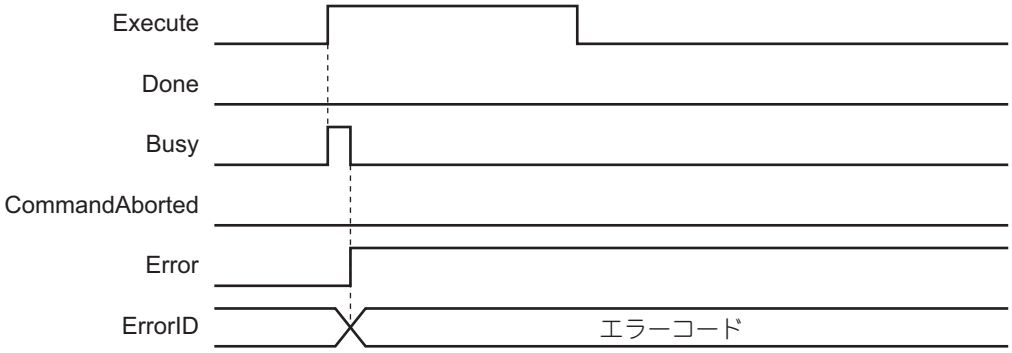
モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となり、パラメータの書き込みは行いません。命令実行前の値が保持されます。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート



● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_GroupSyncMoveAbsolute

各軸に指定した目標位置を周期的に出力します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GroupSyncMoveAbsolute	軸グループサイクリック同期絶対位置制御	FB	MC_GroupSyncMoveAbsolute_instance MC_GroupSyncMoveAbsolute AxesGroup — AxesGroup Execute — InPosition Position — Busy CoordSystem — Active BufferMode — CommandAborted Error ErrorID	MC_GroupSyncMoveAbsolute_instance (AxesGroup :=「パラメータ」, Execute :=「パラメータ」, Position :=「パラメータ」, CoordSystem :=「パラメータ」, BufferMode :=「パラメータ」, InPosition =>「パラメータ」, Busy =>「パラメータ」, Active =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

バージョン関連情報

本命令は、ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
Position	目標位置	ARRAY [0..3] OF LREAL	負数、正数、「0」	0	絶対座標の目標位置を指定します。 単位は、[指令単位] となります。*1
CoordSystem	座標系	_eMC_COORD_SYSTEM	0: _mcACS	0 *2	座標系を指定します。 0: 軸座標系(ACS)
BufferMode	バッファモード選択	_eMC_BUFFER_MODE	0: _mcAborting	0 *2	モーション命令多重起動時の動作を指定します。 0: アボーティング

*1. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。
*2. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

出力変数

出力変数	名称	データ型	有効範囲	内容
InPosition	インポジション	BOOL	TRUE, FALSE	すべての構成軸のフィードバック現在位置が目標位置のインポジション幅内に入ったとき TRUE となります。

出力変数	名称	データ型	有効範囲	内容
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Active	制御中	BOOL	TRUE, FALSE	制御中に TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
InPosition	すべての構成軸のフィードバック現在位置が目標位置のインポジション幅内に入ったとき。	• フィードバック現在位置がインポジション幅外へ出たとき。 •「Error」が TRUE となったとき。 •「CommandAborted」が TRUE となったとき。
Busy	「Execute」の立ち上がり	•「Error」が TRUE となったとき。 •「CommandAborted」が TRUE となったとき。
Active	命令を開始したとき。	•「Error」が TRUE となったとき。 •「CommandAborted」が TRUE となったとき。
CommandAborted	• ほかの命令によるモーション命令多重起動で、本命令が中止されたとき。 • ほかの命令で異常が発生して、本命令が中止されたとき。 • 軸異常発生中に、本命令を起動したとき。 • MC_GroupStop 命令実行中に、本命令を起動したとき。	•「Execute」が TRUE の場合、「Execute」の FALSE と同時 •「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

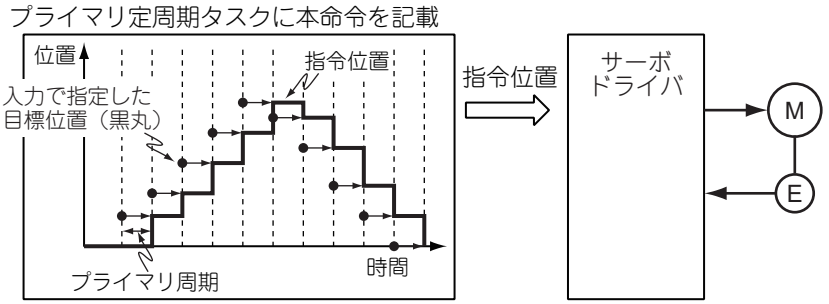
入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	―	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

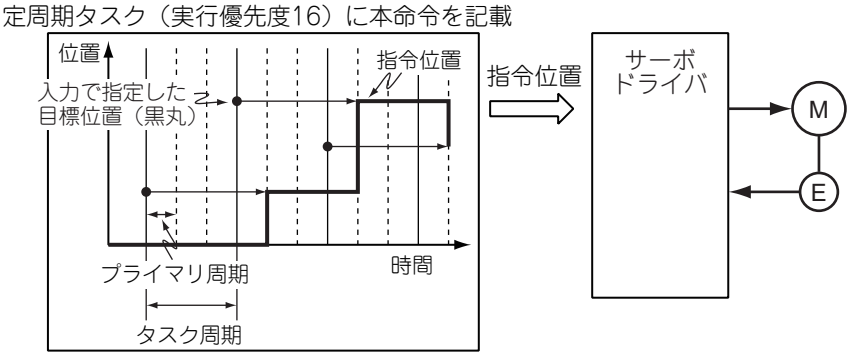
機能説明

• 本命令は、ユーザプログラムから与える目標位置を、タスク周期ごとにサイクリック同期位置モード(CSP)でサーボドライバなどに出力します。目標位置は絶対位置で与えます。

- 速度は軸パラメータの［最高速度］を上限とします。［最大加速度］と［最大減速度］は適用されません。
- プライマリ定周期タスク、または定周期タスク（実行優先度 5）に本命令を記述した場合、入力で指定した目標位置は次のタスク周期でサーボドライバに出力されます。
  プライマリ定周期タスクに本命令を記載した例を以下に示します。
  定周期タスク（実行優先度 5）の場合も同様となります。



- 定周期タスク（実行優先度 16）に本命令を記述した場合、入力で指定した目標位置は次の定周期タスクでサーボドライバに出力されます。



使用上の注意

- 軸グループの構成軸のいずれかが、原点未確定の場合は、「原点未確定状態での命令起動異常（エラーコード: 5466 Hex）となります。
- 指示した目標位置による移動量が軸パラメータの［最高速度］を超えないように目標位置を指令してください。［最高速度］を超える目標位置を指示したときは、指令速度飽和となり、［最高速度］で制限された移動量を出力します。指示した目標位置から不足する分の移動量指示は次の周期以降に出力します。
  なお、このとき軸制御ステータスの「Details.VelLimit」（指令速度飽和）が TRUE となります。

命令の詳細

以下に本命令の詳細について説明します。

● インポジションチェック

「Position」（目標位置）を更新しないときに目標位置とフィードバック位置の幅が軸パラメータの［インポジション幅］の範囲内になったとき、「InPosition」（インポジション）が TRUE となります。
「InPosition」（インポジション）が TRUE のときに目標位置を変更しても、その周期の中では TRUE のままで、次の周期で FALSE となります。
軸パラメータの［インポジションチェック時間］設定は無効です。

● 停止処理

軸グループ動作を停止させるときの動作について説明します。
動作を停止させるには、MC_GroupStop（軸グループ強制停止）命令、または MC_GroupImmediateStop（軸グループ即停止）命令を使用します。これらの命令を起動すると、本命令の「CommandAborted」（実行中断）が TRUE となります。

MC_GroupStop（軸グループ強制停止）命令による停止
即停止します。

MC_GroupImmediateStop（軸グループ即停止）命令による停止
各軸の軸パラメータ設定［即停止入力停止方法］の指定に従い即停止します。

● 異常発生による停止

軸を停止させる異常が発生したとき、設定によらず「即停止」します。

● 使用可能軸

- サーボ軸で本命令を使用できます。
  使用する場合は、MC_Power（運転可）命令の「Enable」（有効）を TRUE（サーボ ON 状態）にしてください。
- 仮想サーボ軸は、常時受け付け可能です。

● 起動条件

- 軸パラメータの［カウントモード］を［リニアモード］としてください。
- すべての構成軸を原点確定状態としてください。
- MC_GroupEnable（軸グループ有効）命令で軸グループを有効にしてください。

● 軸変数のステータス

軸グループ変数の軸グループステータスは、「Status.Moving」（動作中）が TRUE となります。
軸グループ制御ステータスへは影響しません。

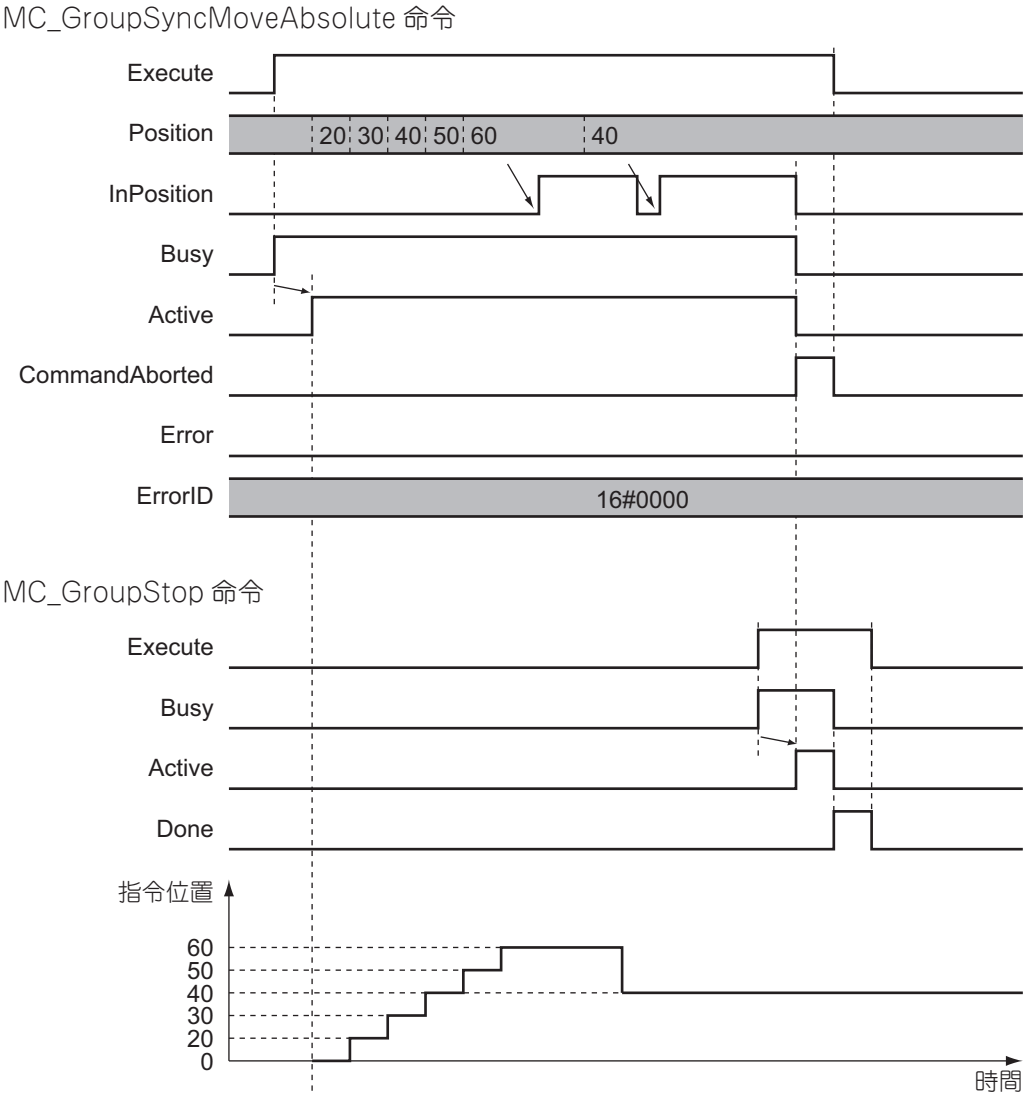
● オーバーライド

本命令に対して、オーバーライドは無効となります。

タイミングチャート

- 「Execute」（起動）の立ち上りと同時に「Busy」（実行中）が TRUE となり、次の周期で「Active」（制御中）が TRUE となります。
- すべての構成軸のフィードバック現在位置が「Position」（目標位置）のインポジション幅内に入ったとき、「InPosition」（インポジション）が TRUE となります。
- ほかの命令により、本命令が中止された場合は「CommandAborted」（実行中断）が TRUE となり、「Busy」（実行中）、「Active」（制御中）、「InPosition」（インポジション）が FALSE となります。
- 本命令を終了させるには、MC_GroupStop（軸グループ強制停止）命令を起動します。

プライマリ定周期タスクに本命令を記載した場合を下記に示します。



参考

MC 機能モジュールはサーボドライバに対して上記タイミングチャートで制御モードを指令します。しかし、サーボドライバに反映されるタイミングは、各サーボドライバの仕様により異なります。

モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex)」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

本命令の「BufferMode」（バッファモード選択）は［アボーティング］だけに対応します。

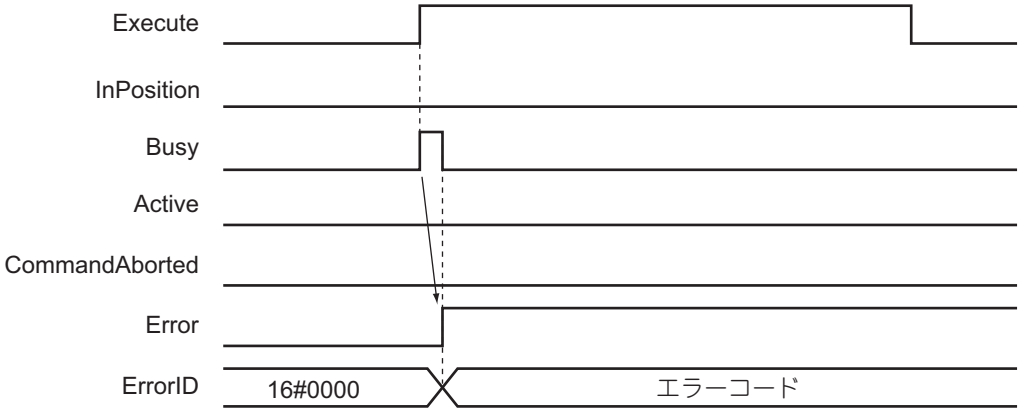
● 本命令実行中の他命令起動

本命令に対するモーション命令多重起動を行う場合、［アボーティング］だけが可能です。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート

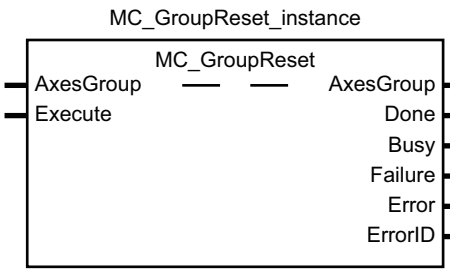


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_GroupReset

軸グループ、および軸の異常を解除します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GroupReset	軸グループエラーリセット	FB		MC_GroupReset_instance (AxesGroup :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, Failure =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
Failure	不正終了	BOOL	TRUE, FALSE	命令が正常に実行されなかったときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	異常解除処理が正常に終了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「Failure」が TRUE となったとき。

変数	TRUE となるタイミング	FALSE となるタイミング
Failure	• 異常による軸、軸グループの減速停止中に命令を実行したとき。 • 軸共通異常発生による軸グループ異常発生中に命令を実行したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
AxesGroup	軸グループ	_sGROUP_REF	−	軸グループを指定します。*1

1. Sysmac Studio の軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group）またはシステム定義変数の軸グループ変数名(_MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。

機能説明

- 「Execute」（起動）の立ち上がりで、「GroupEnable」（グループ有効）状態の「AxesGroup」（軸グループ）で指定した軸グループの異常および軸グループに属する軸異常の解除処理を行います。解除を行えるのは、軸や軸グループで発生する「軽度フォールト」レベル、「監視情報」レベルの異常の解除、およびドライバエラーリセットです。
- 軸がサーボ ON 状態／サーボ OFF 状態にかかわらず、異常解除処理を実行します。
- ドライバエラーが発生している軸にはドライバエラーリセット処理を実行したのち、異常解除処理を実行します。
- ドライバエラーリセット処理はドライバエラーがクリアされるか、軸パラメータの［ドライバエラーリセット監視時間］の間継続します。ドライバエラーリセット処理は軸グループに属する軸に対して同時に行います。
- 解除される異常は「Execute」（起動）の立ち上がりのタイミングで発生している異常を対象とします。解除中に発生した異常については、異常解除は行われません。
- 軸グループのエラー減速停止中に命令を実行した場合は、実行されず「Failure」（不正終了）が TRUE となります。これは、軸が停止する前に、異常解除ができないようにするためです。
また、軸共通異常発生による軸異常も、本命令では解除できないため、「Failure」（不正終了）が TRUE となります。

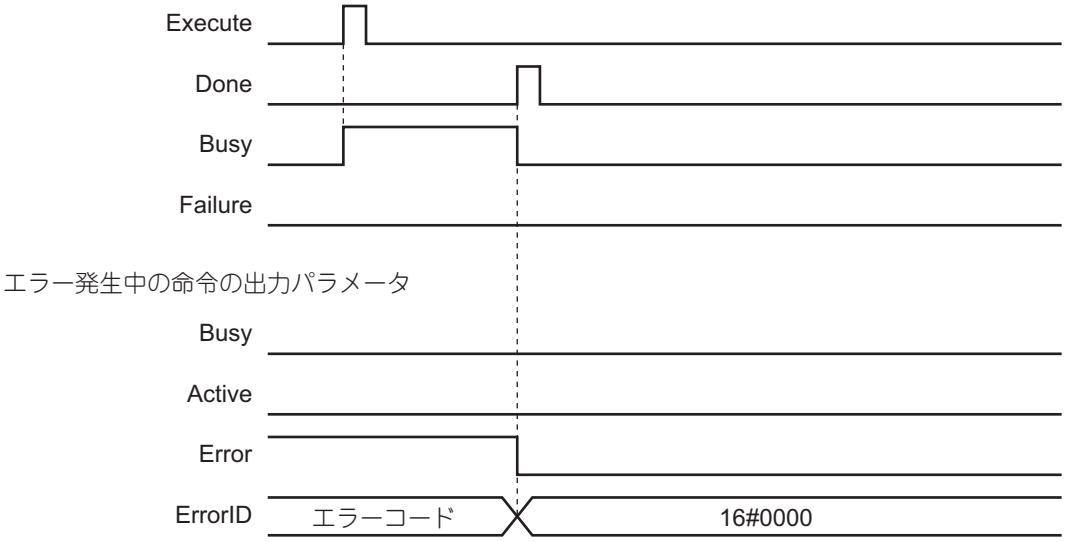
使用上の注意

- 本命令による異常解除の処理は、複数の制御周期にまたがる場合があります。
- 本命令を起動しても軸が動作中の場合、本命令の出力変数「Failure」（不正終了） が TRUE となります。
異常要因を除去した上で、「Done」（完了）が TRUE となるまでリトライ処理を行ってください。
- 異常の発生要因を除去する場合には、各軸が完全に停止したことを確認してから行ってください。
- オムロン製サーボドライバ G5 シリーズに対し本命令を使用する場合、ResetECError（EtherCAT 異常解除）命令が同時に実行されないよう排他処理を行ってください。
- NX シリーズ パルス出力ユニットに対し本命令を使用する場合、パルス出力ユニットに接続したドライバの異常状態は解除されません。詳細は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル(SBCE-374)』を参照してください。

参考

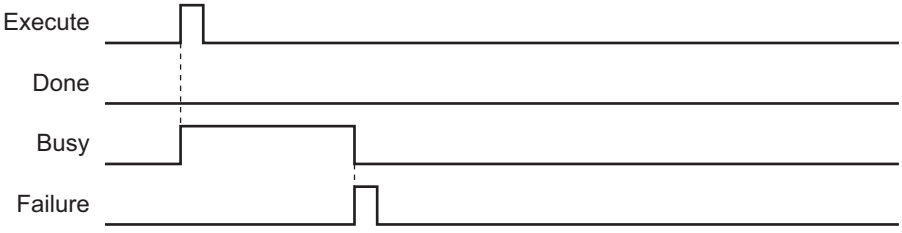
- 軸異常は軸グループが有効の場合だけ、解除可能です。
- 本命令では以下の異常は解除できません。
  軸共通異常 全般
  異常を解除するには、ResetMCError（全エラーリセット）命令を別途実行してください。
- 異常が発生していない軸グループに対して本命令を実行した場合、命令は正常終了し、状態を継続します。

タイミングチャート



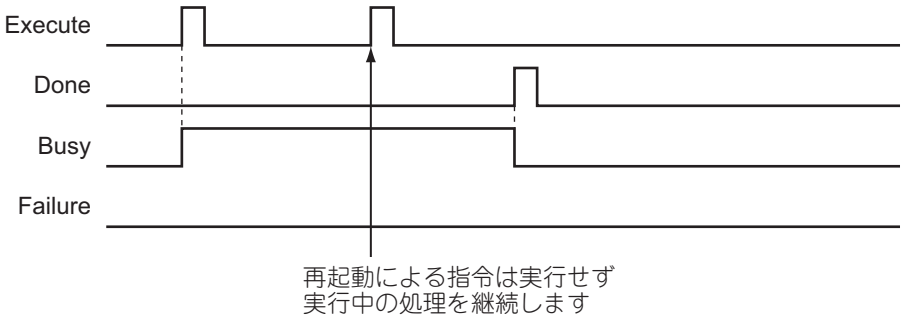
命令の中止

異常による軸、軸グループの減速停止中、もしくは軸共通異常発生による軸グループ異常発生中の異常が解除できない場合は、命令を中断します。



モーション命令再起動

本命令実行中に再度、「Execute」（起動）を立ち上げた場合、再起動の指令は無視して、異常解除処理を継続します。

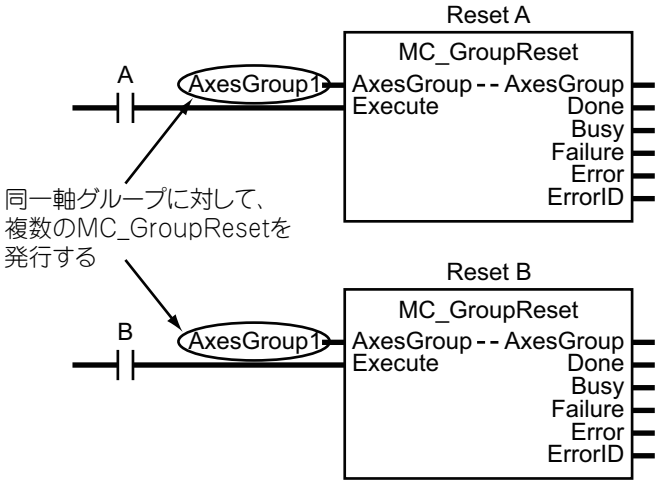


モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 本命令実行中の他命令起動

同一の軸グループに対して、別インスタンスの MC_GroupReset（軸グループエラーリセット）命令が同時に実行された場合、両方の命令をそれぞれ実行します。
スレーブ系の異常発生中は、軸設定の［ドライバエラーリセット監視時間］まで待つことがあります。経過時間も命令インスタンスごとにカウントします。
本命令実行中に指定した軸グループに属する軸に対して MC_Reset（軸エラーリセット）命令が実行された場合も、同様に両方の命令をそれぞれ実行します。



エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

5

共通指令の命令

軸および軸グループに共通する命令について説明します。

MC_SetCamTableProperty 5-2
MC_SetCamTableEndPointIndex.....5-8
MC_SaveCamTable.....5-13
MC_Write.....5-18
MC_GenerateCamTable 5-23
MC_WriteAxisParameter.....5-50
MC_ReadAxisParameter 5-63

MC_SetCamTableProperty

入出力パラメータで指定したカムテーブルの終点インデックスを更新します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_SetCamTableProperty	カムテーブルプロパティ更新	FB	MC_SetCamTableProperty_instance MC_SetCamTableProperty CamTable — CamTable Execute — Done EndPointIndex MaxDataNumber Busy CommandAborted Error ErrorID	MC_SetCamTableProperty_instance (CamTable :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, EndPointIndex =>「パラメータ」, MaxDataNumber =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
EndPointIndex	終点インデックス	UINT	正数、または「0」	カムテーブルの終点インデックスを出力します。
MaxDataNumber	最大カムデータ数	UINT	正数	最大カムデータ数を出力します。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	カムテーブルの書き替えが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	本命令を中止する要因が発生したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
CamTable	カムテーブル	ARRAY[0..N] OF _sMC_CAM_REF	−	カムデータ構造体_sMC_CAM_REF 型の配列変数をカムテーブルとして指定します。*1

*1. 配列要素「N」は Sysmac Studio により自動的に設定されます。Sysmac Studio のカムエディタで作成したカムデータ変数を指定してください。

機能説明

- 本命令は、入出力パラメータで指定したカムテーブルの終点インデックスを更新します。
- 終点とは、カムテーブルを始点から順に検索した際に、位相が最初に「0」となるカムデータより 1 つ前に配置されたデータのことです。
- 終点の配列番号を「EndPointIndex」（終点インデックス）に出力します。
- 位相「0」を検出したら、それ以降は無効カムデータとして扱うため、位相／変位の値は無視されます。
- 最大カムデータ数とは、カムデータ構造体配列の最大要素数を表します。
  最大カムデータ数は、Sysmac Studio で構造体配列を宣言する際に指定した値です。
- ユーザプログラムから、カムデータの終点インデックスを変更した場合、終点の更新が必要です。
  この場合に、本命令を使用して有効カムデータ数の更新を行ってください。
- 終点インデックスの更新処理において実行から完了までに必要な周期を調節される場合、本命令のかわりに「MC_SetCamTableEndPointIndex (P.5-8)」をご使用ください。

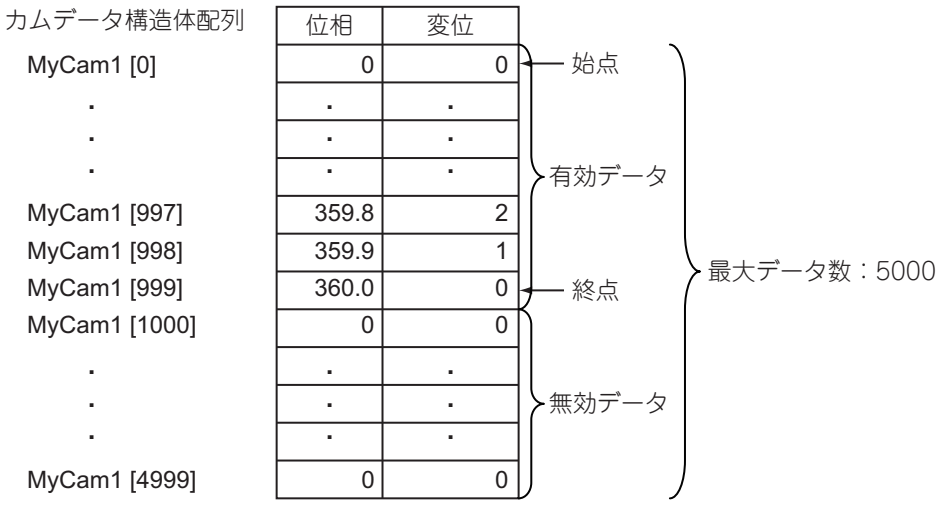
カムテーブルについては、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編 (SBCE-433)』を参照してください。

使用上の注意

- カムテーブルを検索中に、位相「0」を見つける前に位相が昇順になっていないことを検出すると異常となります。
- 最大カムデータ数はユーザプログラムから変更することはできません。
- カムデータの終点を変更したり、カムデータの値の書き換えを行ったりした後は、本命令を実行してください。
  終点インデックスを変更したり、位相が昇順でなかったりした場合、カム動作や MC_CamIn（カム動作開始）命令の「EndOfProfile」（カムサイクル完了）が意図した動作になりません。
- 本命令処理中にカムテーブルデータを変更した場合、更新処理が正しく行われません。プログラムでカムデータを変更する場合は、本命令の完了後に行ってください。
- カムデータ変数はグローバル変数であり、複数のタスクからカムデータ変数の値を参照や変更をすることができます。複数のタスクからカムデータ変数の値を変更する場合、複数のタスクによる書き換え処理で競合が発生しないようにプログラムを作成してください。
- 「グローバル変数のタスク間排他制御」機能を用いてカムデータ変数を排他制御する場合、排他対象のカムデータ変数を用いてモーション命令を起動しないでください。軽度フォールトレベルの異常「カムテーブル指定不正（エラーコード: 5439 Hex）」が発生します。
- 本命令処理中に処理対象のカムテーブルを用いた命令を実行すると、意図しない実行結果になる可能性があります。カムテーブルを用いた命令を実行する場合は本命令が完了してから起動してください。

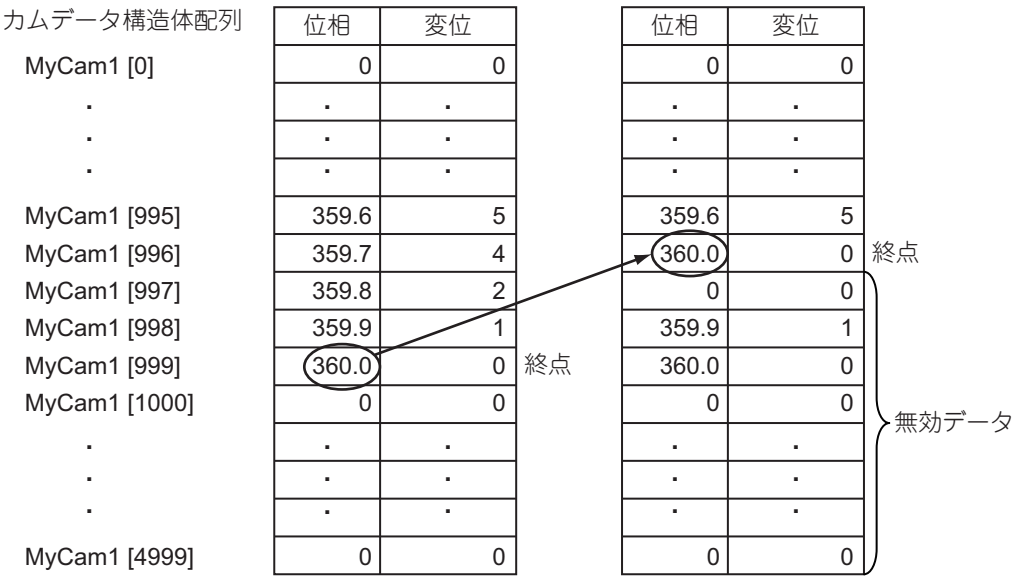
命令の詳細

例えば、次のようなカムテーブルの場合、「EndPointIndex」（終点インデックス）は「999」、「MaxDataNumber」（最大カムデータ数）は「5000」となります。

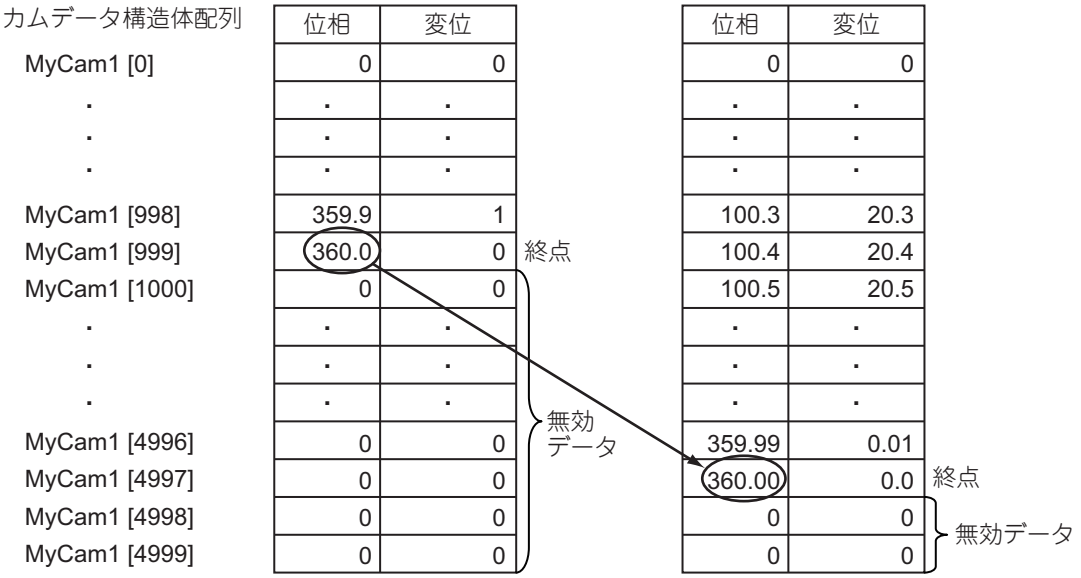


以下に、カムデータの書き換えと「EndPointIndex」（終点インデックス）の関係を示します。

MyCam1[1000]以降の位相が「0」のカムテーブルで本命令を起動すると「EndPointIndex」（終点インデックス）が「999」となります。
MyCam1[997]の位相を「0」に書き換えた後に、本命令を起動すると「EndPointIndex」（終点インデックス）が「996」となります。



また、MyCam1[1000]から MyCam1[4997]までの位相を「0」以外の数値に書き換えた後、本命令を起動すると「EndPointIndex」（終点インデックス）が「4997」となります。

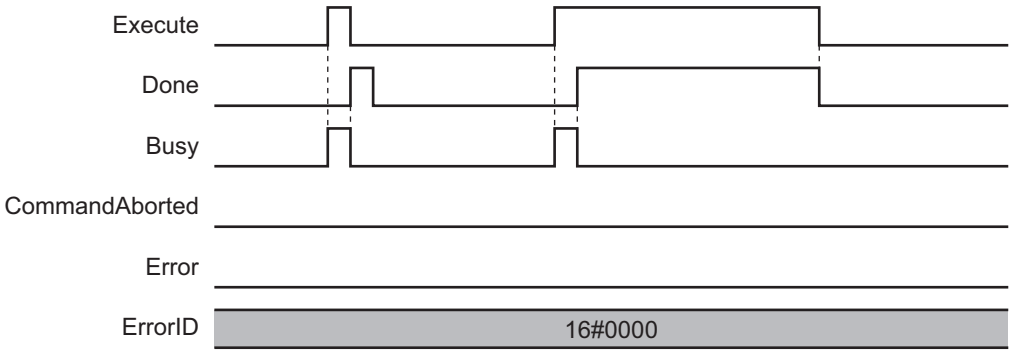


なお、MyCam1[1000]から MyCam1[4997]までの位相を「0」以外に書き換えたにも関わらず本命令を起動しなかった場合、カム動作は MyCam1[0]から MyCam1[999]の間で実行されます。
カムテーブルは書き換わっていますが「EndPointIndex」（終点インデックス）は変わりません。

カムテーブルのデータ構造の詳細については、「MC_CamIn (P.3-160)」を参照してください。

タイミングチャート

本命令の起動方法を 2 種類示します。動作としては、どちらの起動方法の場合も正常終了を表します。
1 回目の起動は、「Execute」（起動）を TRUE とし、完了前に「Execute」（起動）を FALSE にした場合です。
2 回目の起動は、「Execute」（起動）を保持した場合を表しています。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

● 他命令実行中の本命令起動

本命令と同一の「CamTable」（カムテーブル）を使用する以下の命令がすでに起動されている場合、本命令を多重起動することはできません。

- MC_SaveCamTable（カムテーブル保存）命令
- MC_SetCamTableProperty（カムテーブルプロパティ更新）命令
- MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令
- MC_GenerateCamTable（カムテーブル生成）命令

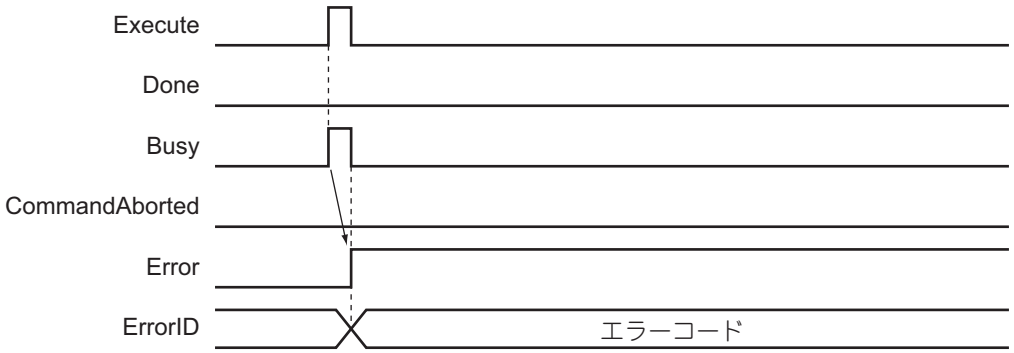
● 本命令実行中の他命令起動

本命令実行中に、同一の「CamTable」（カムテーブル）を使用する以下の命令を多重起動することはできません。

- MC_SaveCamTable（カムテーブル保存）命令
- MC_SetCamTableProperty（カムテーブルプロパティ更新）命令
- MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令
- MC_GenerateCamTable（カムテーブル生成）命令

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。



● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_SetCamTableEndPointIndex

入出力パラメータで指定したカムテーブルの終点インデックスを更新します。
カムテーブルに対する操作と得られる結果は、MC_SetCamTableProperty（カムテーブルプロパティ更新）命令と同じですが、命令起動から完了まで必要なタスク周期を調節できます。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_SetCamTableEndPointIndex	カムテーブル終点インデックス更新	FB	MC_SetCamTableEndPointIndex_instance MC_SetCamTableEndPointIndex CamTable ── ── CamTable Execute — Done SeachNum — EndPointIndex MaxDataNumber Busy CommandAborted Error ErrorID	MC_SetCamTableEndPointIndex_instance (CamTable :=「パラメータ」, Execute :=「パラメータ」, SearchNum :=「パラメータ」, Done =>「パラメータ」, EndPointIndex =>「パラメータ」, MaxDataNumber =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。
SearchNum	検索数	UINT	0～65535	10	• プライマリ定周期タスク 1 周期あたりの検索するカムデータ数を指定します。 •「10」を指定した場合、MC_SetCamTableProperty（カムテーブルプロパティ更新）命令を使用したときと同一のタイミングで命令が完了します。 •「0」を指定した場合、「10」として扱われます。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
EndPointIndex	終点インデックス	UINT	正数、または「0」	カムテーブルの終点インデックスを出力します。
MaxDataNumber	最大カムデータ数	UINT	正数	最大カムデータ数を出力します。

出力変数	名称	データ型	有効範囲	内容
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	カムテーブルの書き替えが完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	本命令を中止する要因が発生したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
CamTable	カムテーブル	ARRAY[0..N] OF _sMC_CAM_REF	－	カムデータ構造体_sMC_CAM_REF 型の配列変数をカムテーブルとして指定します。*1

*1. 配列要素「N」は Sysmac Studio により自動的に設定されます。Sysmac Studio のカムエディタで作成したカムデータ変数を指定してください。

機能説明

- 本命令は、最新のカムテーブルに更新するとともに、終点インデックスと最大カムデータ数を出力します。
- カムテーブルに対する操作と得られる結果は MC_SetCamTableProperty（カムテーブルプロパティ更新）命令と同じですが、カムデータの終点を始点から順に検索する際に、プライマリ定周期タスク 1 周期あたりの検索するカムデータ数が指定されることにより命令起動から完了まで必要なタスク周期を調節できます。
- 本命令の基本機能については「MC_SetCamTableProperty (P.5-2)」を参照してください。
- 「SearchNum」（検索数）を「10」に指定した場合、MC_SetCamTableProperty（カムテーブルプロパティ更新）命令を使用したときと同一のタイミングで命令が完了します。

カムテーブルについては、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

使用上の注意

- カムテーブルを検索中に、位相「0」を見つける前に位相が昇順になっていないことを検出すると異常となります。
- 最大カムデータ数はユーザプログラムから変更することはできません。
- カムデータの終点を変更したり、カムデータの値の書き換えを行ったりした後は、本命令を実行してください。
  終点インデックスを変更したり、位相が昇順でなかったりした場合、カム動作や MC_CamIn（カム動作開始）命令の「EndOfProfile」（カムサイクル完了）が意図した動作になりません。
- 本命令処理中にカムテーブルデータを変更した場合、更新処理が正しく行われません。プログラムでカムデータを変更する場合は、本命令の完了後に行ってください。
- カムデータ変数はグローバル変数であり、複数のタスクからカムデータ変数の値を参照や変更をすることができます。複数のタスクからカムデータ変数の値を変更する場合、複数のタスクによる書き換え処理で競合が発生しないようにプログラムを作成してください。
- 「グローバル変数のタスク間排他制御」機能を用いてカムデータ変数を排他制御する場合、排他対象のカムデータ変数を用いてモーション命令を起動しないでください。軽度フォールトレベルの異常「カムテーブル指定不正（エラーコード: 5439 Hex）」が発生します。
- 「SearchNum」（検索数）の値を大きくするとプライマリ定周期タスクのタスク実行時間が長くなるため、タスク周期超過が発生する可能性があります。「SearchNum」（検索数）の値を設定する場合は小さな値から段階的に引き上げていき、タスク周期超過が発生しないことを確認してください。
- 本命令処理中に処理対象のカムテーブルを用いた命令を実行すると、意図しない実行結果になる可能性があります。カムテーブルを用いた命令を実行する場合は本命令が完了してから起動してください。

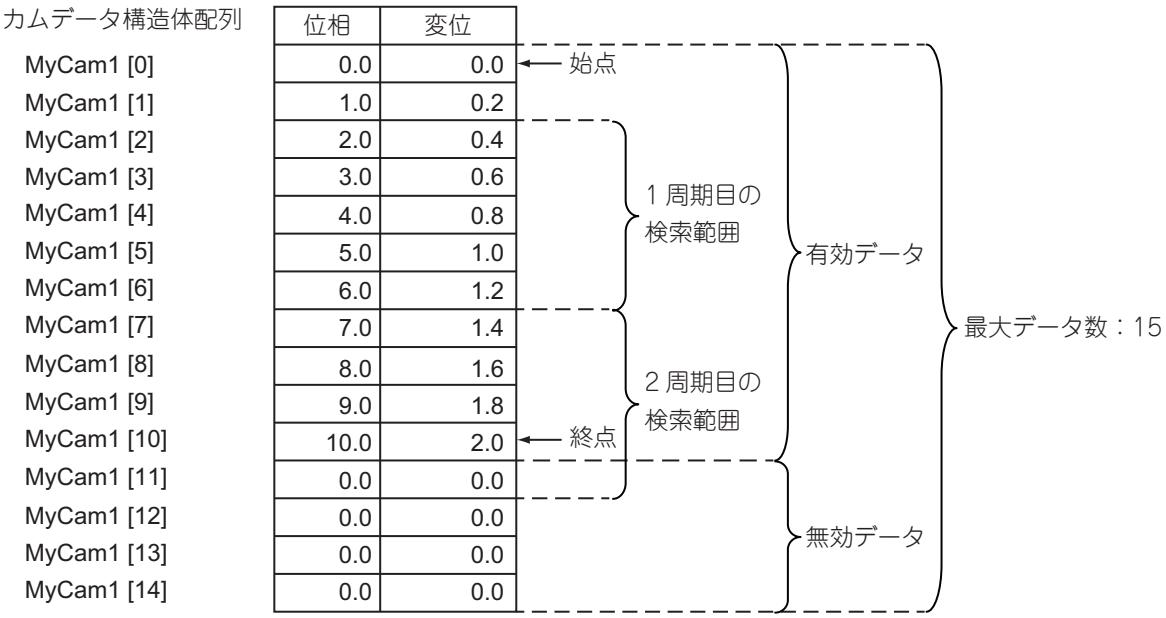
命令の詳細

ここでは本命令の実行時間について説明します。
本命令のカムテーブルの終点インデックス更新の詳細については「MC_SetCamTableProperty (P.5-2)」を参照してください。
本命令の実行時間は以下のとおり算出することができます。

- 「EndPointIndex」（終点インデックス）が、カムデータ構造体配列の末尾をささない場合
  本命令の実行時間［タスク周期］ = RoundUp(「EndPointIndex」（終点インデックス） / 「SearchNum」（検索数）)
- 「EndPointIndex」（終点インデックス）が、カムデータ構造体配列の末尾をさす場合
  本命令の実行時間［タスク周期］ = RoundUp((「EndPointIndex」（終点インデックス）-1) / 「SearchNum」（検索数）)

ただし、RoundUp は小数点以下切り上げです。

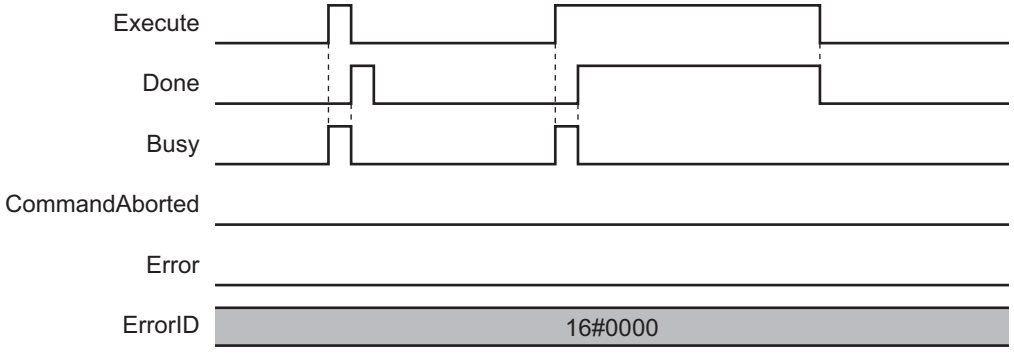
例えば、次のようなカムテーブルで「SearchNum」（検索数）を「5」と指定した場合、本命令の実行時間は「2」タスク周期となります。



終点の検索は MyCam1[2]から実行され、タスク周期毎に「SearchNum」（検索数）だけ検索します。終点は、カムテーブルを始点側から順に検索した際、位相が最初に「0」となるカムデータより 1 つ前に配置されたデータであるため、MyCam1[11]まで検索し、「EndPointIndex」（終点インデックス）を検出します。

タイミングチャート

本命令の起動方法を 2 種類示します。動作としては、どちらの起動方法の場合も正常終了を表します。
1 回目の起動は、「Execute」（起動）を TRUE とし、完了前に「Execute」（起動）を FALSE にした場合です。
2 回目の起動は、「Execute」（起動）を保持した場合を表しています。
「Busy」（実行中）が TRUE のまま維持されるタスク周期は「SearchNum」（検索数）とカムテーブルの終点インデックスに依存します。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex）」となります。

モーション命令多重起動

● 他命令実行中の本命令起動

本命令と同一の「CamTable」（カムテーブル）を使用する以下の命令がすでに起動されている場合、本命令を多重起動することはできません。

- MC_SaveCamTable（カムテーブル保存）命令
- MC_SetCamTableProperty（カムテーブルプロパティ更新）命令
- MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令
- MC_GenerateCamTable（カムテーブル生成）命令

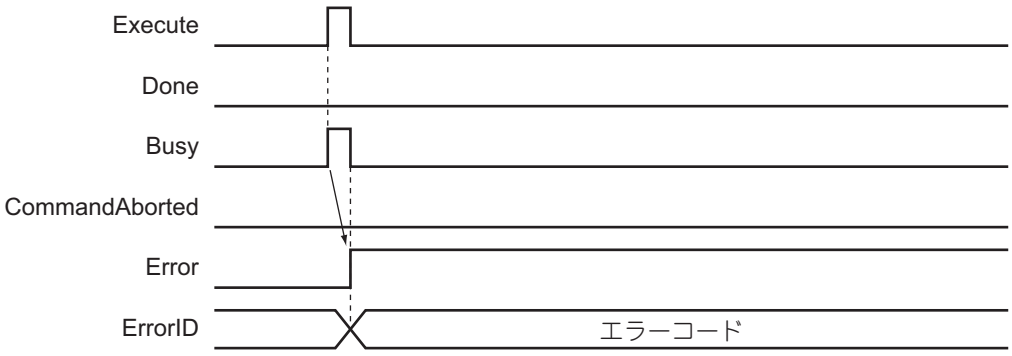
● 本命令実行中の他命令起動

本命令実行中に、同一の「CamTable」（カムテーブル）を使用する以下の命令を多重起動することはできません。

- MC_SaveCamTable（カムテーブル保存）命令
- MC_SetCamTableProperty（カムテーブルプロパティ更新）命令
- MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令
- MC_GenerateCamTable（カムテーブル生成）命令

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

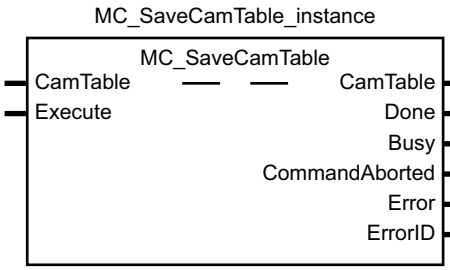


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_SaveCamTable

入力パラメータで指定したカムテーブルを不揮発性メモリに保存します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_SaveCamTable	カムテーブル保存	FB		MC_SaveCamTable_instance (CamTable :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	カムテーブルの保存が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後

変数	TRUE となるタイミング	FALSE となるタイミング
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	本命令を中止する要因が発生したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
CamTable	カムテーブル	ARRAY[0..N] OF _sMC_CAM_REF	−	カムデータ構造体_sMC_CAM_REF 型の配列変数をカムテーブルとして指定します。*1

*1. 配列要素「N」は Sysmac Studio により自動的に設定されます。Sysmac Studio のカムエディタで作成したカムデータ変数を指定してください。

機能説明

- 本命令は、入出力変数で指定したカムテーブルを不揮発性メモリに保存します。
- カムテーブルの保存処理中は、モーション制御システム変数の_MC_COM.Status.CamTableBusy（カムテーブルファイル保存実行中／待機中）が TRUE となります。

カムテーブルのデータ仕様については、「MC_CamIn (P.3-160)」を参照してください。

使用上の注意

- カムデータの書き換えを行った後は、コントローラの電源を OFF する前に本命令で保存を行ってください。保存を行わずにコントローラ電源を OFF した場合、書き換えたデータは消失します。
- 本命令処理中にカムテーブルデータを変更した場合、更新処理が正しく行われません。プログラムでカムテーブルを変更する場合は、本命令処理中にカムテーブルへの書き込みを実行しないでください。
- 本命令は、他の命令に比べて著しく処理時間が長くなります。また、CPU ユニットの演算負荷量により処理時間が大きく異なる場合があります。そのため、本命令の完了をトリガとして次命令を起動する場合、次命令の起動タイミングに注意してください。
- 本命令処理中にコントローラの電源を OFF しないでください。OFF した場合、正しくデータが保存されません。また、不揮発性メモリに保存されているカムテーブルが壊れる可能性があります。
- 本命令処理中にコントローラの電源を OFF すると、次回電源投入時に全停止フォールトが発生する場合があります。
- 本命令処理中は、カムデータのダウンロードやアップロード、オンライン開始の操作、オンラインエディット、データトレース開始の操作などが使用できません。
  また、カムデータのアップロードやダウンロード、オンラインエディットなどを行っている場合に本命令を実行すると、カムテーブル保存命令実行不可（エラーコード: 743CHex）となります。カムデータが正しく保存されるまでリトライ処理を行ってください。
  なお、「ダウンロード」や「アップロード」は、Sysmac Studio の「同期」機能を使用します。
- 不揮発性メモリの書き込み回数には上限があります。本命令を頻繁に実行すると不揮発性メモリの寿命が短くなりますので、本命令の使用は必要最小限にしてください。
- カムデータ変数はグローバル変数であり、複数のタスクからカムデータ変数の値を参照や変更をすることができます。複数のタスクからカムデータ変数の値を変更する場合、複数のタスクによる書き換え処理で競合が発生しないようにプログラムを作成してください。
- 「グローバル変数のタスク間排他制御」機能を用いてカムデータ変数を排他制御する場合、排他対象のカムデータ変数を用いてモーション命令を起動しないでください。軽度フォールトレベルの異常「カムテーブル指定不正（エラーコード: 5439 Hex）」が発生します。
- オンラインエディット反映中に本命令を実行しないでください。実行すると、オンラインエディットによる変更が正しく反映されない可能性があります。
  オンラインエディット反映中とは、以下の状態を指します。
  a）ユニットバージョン Ver.1.04 以降の CPU ユニットと Sysmac Studio Ver.1.05 以降の組み合わせでは、内蔵不揮発性メモリバックアップ中ウィンドウが閉じるまでの間
  b）ユニットバージョン Ver.1.03 以前の CPU ユニット、および Sysmac Studio Ver.1.04 以前の組み合わせでは、反映確認ウィンドウで［はい］を選択してから、反映確認ウィンドウが閉じるまでの間

● CPU ユニットの動作モードとの関係

本命令によりカムデータを保存中、CPU ユニットの動作モードをプログラムモードに切り替えた後も、カムデータの保存処理は継続します。
プログラムモード中にカムデータ保存処理が実行されているか確認する場合は、Sysmac Studio をオンラインにしてモーション制御システム変数の_MC_COM.Status.CamTableBusy をモニタしてください。

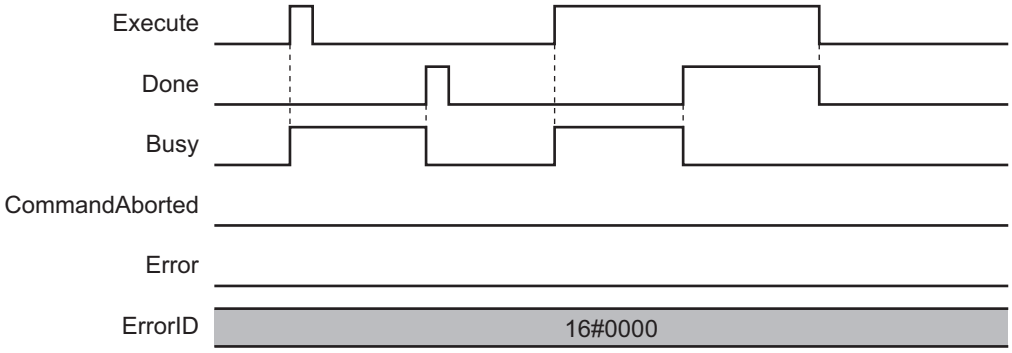
● オンラインエディットによる命令削除時

本命令によりカムデータを保存中、オンラインエディットにより本命令を削除してもカムデータの保存処理は継続します。

タイミングチャート

本命令の起動方法を 2 種類示します。動作としては、どちらの起動方法の場合も正常終了を表します。

1 回目の起動は、「Execute」（起動）を TRUE とし、完了前に「Execute」（起動）を FALSE にした場合です。
2 回目の起動は、「Execute」（起動）を保持した場合を表しています。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex)」となります。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

● 他命令実行中の本命令起動

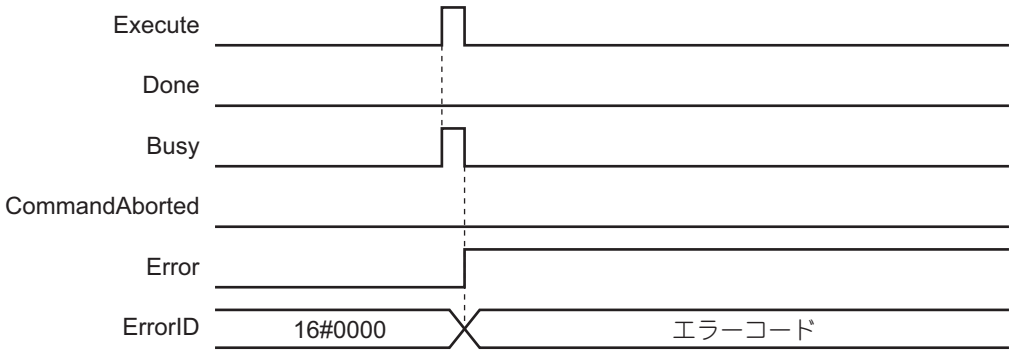
本命令と同一の「CamTable」（カムテーブル）を使用する MC_SetCamTableProperty（カムテーブルプロパティ更新）命令、MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令が既に起動されている場合、本命令は多重起動できません。
また、MC_SaveCamTable（カムテーブル保存）命令実行中は、本命令を多重起動できません。

● 本命令実行中の他命令起動

本命令実行中に、同一の「CamTable」（カムテーブル）を使用する別命令の MC_SetCamTableProperty(カムテーブルプロパティ更新)命令、MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令を多重起動することはできません。
また、本命令実行中に MC_SaveCamTable（カムテーブル保存）命令を多重起動することはできません。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

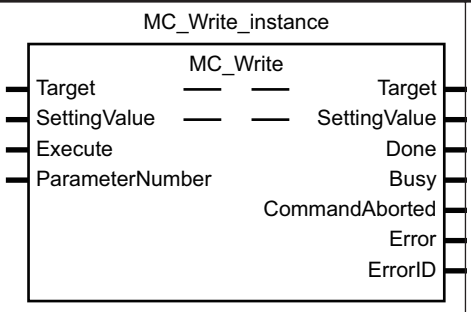


● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_Write

MC 機能モジュールのモーション制御パラメータ設定の一部を書き換えます。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_Write	MC 設定書込	FB		MC_Write_instance (Target :=「パラメータ」, SettingValue :=「パラメータ」, Execute :=「パラメータ」, ParameterNumber :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

使用上の注意

本命令で書き換えた値は CPU ユニットの不揮発性メモリに保存されません。そのため、コントローラの電源を OFF、およびダウンロード、MC 機能モジュールの再起動処理が開始されると、書き換えた値は消失し、Sysmac Studio で設定した値に戻ります。
不揮発性メモリに保存する場合は、Sysmac Studio を使って、パラメータの転送を行ってください。

参考

「ダウンロード」は、Sysmac Studio の「同期」機能を使用します。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

入力変数	名称	データ型	有効範囲	初期値	内容
Parameter Number	パラメータ番号	_eMC_PARAMETER_NUMBER	0: _mcChkVel 1: _mcChkAcc 2: _mcChkDec 3: _mcPosiChkTrq *1 4: _mcNegaChkTrq *1 5: _mcFELmt 6: _mcChkFELmt 7: _mcSwLmtMode 8: _mcPosiSwLmt 9: _mcNegaSwLmt 10: _mcInPosTime 11: _mcInPosRange *2 12: _mcStartVel *3	0 *4	書き込み先のパラメータを指定します。 0: 速度警告値／補間速度警告値 1: 加速度警告値／補間加速度警告値 2: 減速度警告値／補間減速度警告値 3: 正方向トルク警告値 4: 負方向トルク警告値 5: 位置偏差オーバー値 6: 位置偏差警告値 7: ソフトウェアリミット機能 8: 正方向ソフトウェアリミット 9: 負方向ソフトウェアリミット 10: インポジションチェック時間 11: インポジション幅 12: 起動速度

*1. トルク制御時にだけ有効です。
*2. ユニットバージョン Ver.1.01 以降の CPU ユニットと Sysmac Studio Ver.1.02 以降の組み合わせで指定可能です。
*3. ユニットバージョン Ver.1.10 以降の CPU ユニットと Sysmac Studio Ver.1.12 以降の組み合わせで指定可能です。
*4. 有効範囲が列挙型の変数の初期値は、実際には数値ではなく、列挙子となります。

● パラメータ番号のデータ型と有効範囲

パラメータ	データ型	有効範囲	備考
0 : 速度警告値／補間速度警告値	UINT	0～100	単位は[%]です。
1 : 加速度警告値／補間加速度警告値	UINT	0～100	単位は[%]です。
2 : 減速度警告値／補間減速度警告値	UINT	0～100	単位は[%]です。
3 : 正方向トルク警告値 *1	UINT	0～1000	単位は[%]です。
4 : 負方向トルク警告値 *1	UINT	0～1000	単位は[%]です。
5 : 位置偏差オーバー値	LREAL	正数	単位は［指令単位］です。*2
6 : 位置偏差警告値	LREAL	正数	位置偏差オーバー値以下にします。 単位は［指令単位］です。*2
7 : ソフトウェアリミット機能	_eMC_SWLMT_MODE	0: _mcNonSwLmt 1: _mcCmdDecelerationStop 2: _mcCmdImmediateStop 3: _mcActDecelerationStop 4: _mcActImmediateStop	0: ソフトフェアリミット機能の無効 1: 指令位置に対して有効で減速停止 2: 指令位置に対して有効で即停止（たまりパルス停止） 3: フィードバック位置に対して有効で減速停止 4: フィードバック位置に対して有効で即停止（たまりパルス停止）
8 : 正方向ソフトウェアリミット	LREAL	負数、正数、「0」	単位は［指令単位］です。*2
9 : 負方向ソフトウェアリミット	LREAL	負数、正数、「0」	単位は［指令単位］です。*2
10 : インポジションチェック時間	UINT	0～10000	単位は[ms]です。
11 : インポジション幅	LREAL	正数、「0」	単位は［指令単位］です。*2
12 : 起動速度	LREAL	正数、「0」	単位は［指令単位／s］です。*2

*1. トルク制御時にだけ有効です。
*2. 指令単位については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』の「単位変換設定」を参照してください。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	•「Execute」が TRUE の場合、「Execute」の FALSE と同時 •「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	•「Done」が TRUE となったとき。 •「Error」が TRUE となったとき。 •「CommandAborted」が TRUE となったとき。
CommandAborted	ほかの命令で異常が発生し、本命令が中止されたとき。	•「Execute」が TRUE の場合、「Execute」の FALSE と同時 •「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Target	書込対象	_sAXIS_REF、または _sGROUP_REF	−	書き込み対象となる軸、または軸グループを指定します。*1
SettingValue	設定値	指定した変数の型によります。*2	−	書き込む値を指定します。 有効範囲は「ParameterNumber」（パラメータ番号）で指定したモーション制御パラメータ指定に従います。初期値は「0」。

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）、または軸グループ基本設定画面で作成したユーザ定義変数の軸グループ変数名（デフォルト: MC_Group*）を使用してください。いずれもシステム定義変数の変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*], _MC_GRP[*], _MC1_GRP[*], _MC2_GRP[*])も使用できます。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名、または軸グループ変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. 変数のデータ型については、「パラメータ番号のデータ型と有効範囲 (P.5-19)」を参照してください。

● **入出力変数の反映タイミング**

入出力変数	書き込みタイミング
SettingValue	「Done」の立ち上がり時

機能説明

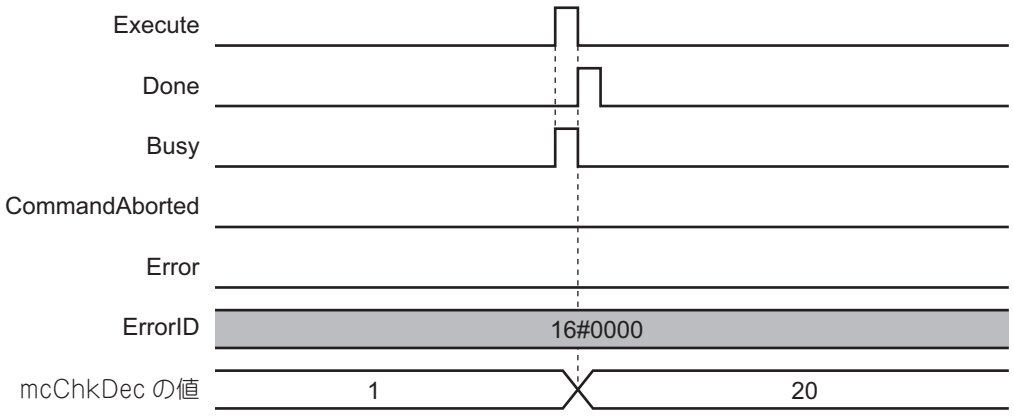
- 「Execute」（起動）の立ち上がりで、「Target」（書込対象）、および「ParameterNumber」（パラメータ番号）で指定したモーション制御システム変数に「SettingValue」（設定値）を書き込みます。
- 入力変数で指定されたパラメータは、軸変数の「Status.Standstill」（停止中）が TRUE のとき、または軸グループ変数の「Status.Standby」（停止中）が TRUE のときの動作開始時と、［アボーティング］指定での多重起動時に適用されます。
  このため、再起動および［アボーティング］指定以外での多重起動で動作を継続させた場合は、適用されません。

使用上の注意

本命令で書き換えた値は CPU ユニットの不揮発性メモリに保存されません。そのため、コントローラの電源を OFF、およびダウンロード、MC 機能モジュールの再起動処理が開始されると、書き換えた値は消失し、Sysmac Studio で設定した値に戻ります。
不揮発性メモリに保存する場合は、Sysmac Studio を使って、パラメータの転送を行ってください。

タイミングチャート

軸パラメータ設定の_mcChkDec（減速度警告値）にデータ「20」を書き込む場合のタイミングチャートは、次のようになります。



モーション命令再起動

本命令の「Busy」（実行中）が TRUE のときに、同じインスタンスの「Execute」（起動）の立ち上りがある場合、命令の再起動となります。このとき、あとの「Execute」（起動）の立ち上がり時点の「Target」（書込対象）と「ParameterNumber」（パラメータ番号）、「SettingValue」（設定値）に書き換わります。

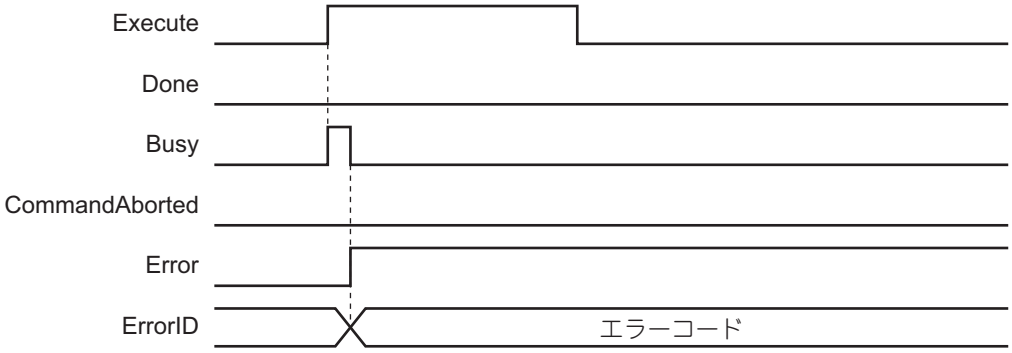
モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となり、パラメータの書き込みは行いません。命令実行前の値が保持されます。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。

● 異常発生時のタイミングチャート

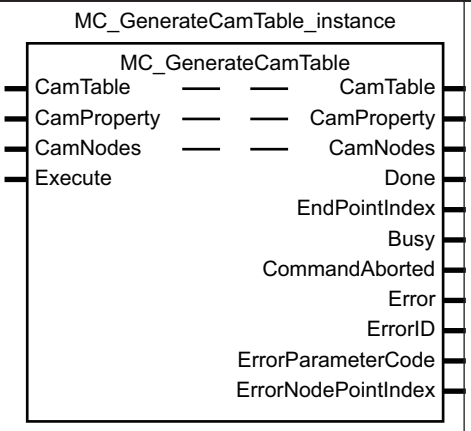


● エラーコード

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_GenerateCamTable

入出力変数で指定した「CamProperty」（カムプロパティ）と「CamNodes」（カムノード）をもとにカムテーブルを生成します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_GenerateCamTable	カムテーブル生成	FB		MC_GenerateCamTable_instance (CamTable :=「パラメータ」, CamProperty :=「パラメータ」, CamNodes :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, EndPointIndex =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」, ErrorParameterCode =>「パラメータ」, ErrorNodePointIndex =>「パラメータ」);

使用上の注意

本命令で指定するカムテーブルは、あらかじめ Sysmac Studio のカムエディタで作成し、CPU ユニットにダウンロードしておく必要があります。
「ダウンロード」は、Sysmac Studio の「同期」機能を使用します。

バージョン関連情報

本命令は、ユニットバージョン Ver.1.08 以降の CPU ユニットと Sysmac Studio Ver.1.09 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
EndPointIndex	終点インデックス	UINT	正数、または「0」	命令の実行が完了したときにカムテーブルの終点インデックスを出力します。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。
ErrorParameterCode	パラメータ詳細コード	WORD	*1	エラーコードによっては、付属情報が存在します。存在する場合は、異常となっているパラメータの詳細コードを出力します。
ErrorNodePointIndex	ノードポイント要素番号	UINT	*1	エラーコードによっては、付属情報が存在します。存在する場合は、異常となっているノードポイントの要素番号を出力します。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	カムテーブルの生成が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	本命令を中止する要因が発生したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

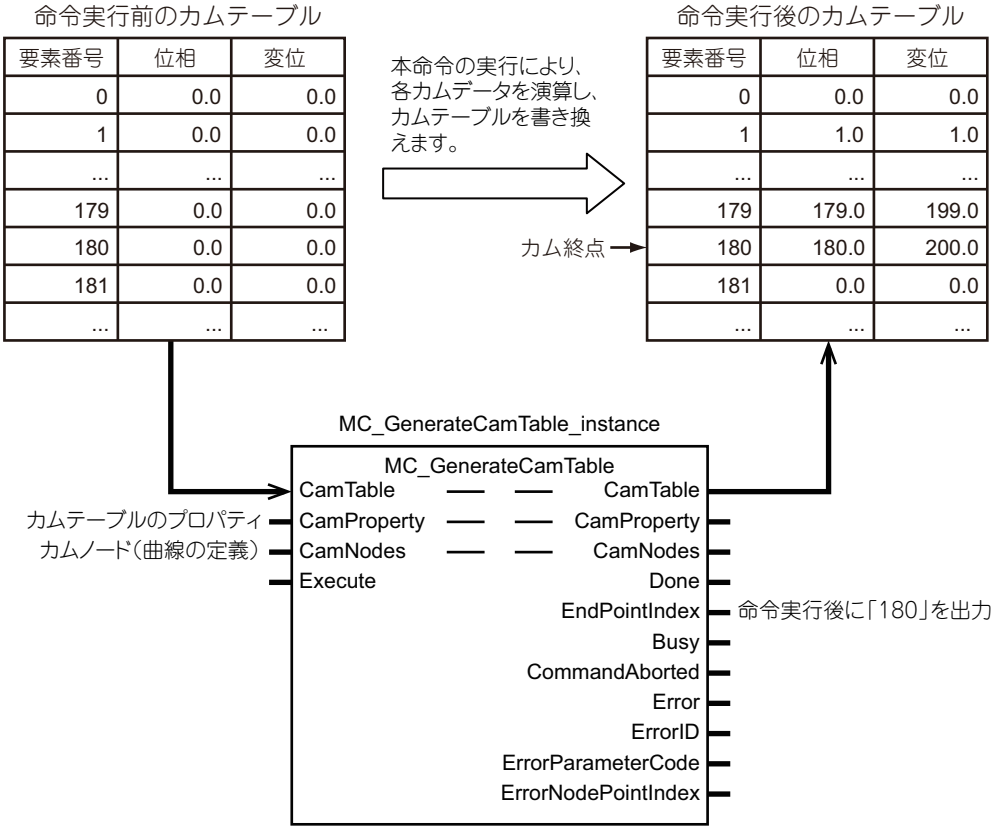
入出力変数	名称	データ型	有効範囲	内容
CamTable	カムテーブル	ARRAY[0..N] OF _sMC_CAM_REF	−	カムデータ構造体_sMC_CAM_REF 型の配列変数をカムテーブルとして指定します。*1 Sysmac studio のカムエディタで作成したカムデータ変数を指定してください。
CamProperty	カムプロパティ	_sMC_CAM_PROPERTY	−	カムプロパティ構造体_sMC_CAM_PROPERTY 型の変数を指定します。 _sMC_CAM_PROPERTY 型のユーザ定義変数、または Sysmac studio で作成したカムプロパティ変数を指定してください。

入出力変数	名称	データ型	有効範囲	内容
CamNodes	カムノード	ARRAY[0..N] OF _sMC_CAM_NODE	―	カムノード構造体_sMC_CAM_NODE 型の配列変数を指定します。 _sMC_CAM_NODE 型のユーザ定義変数 *2、または Sysmac studio で作成したカムノード変数を指定してください。

*1. 配列要素「N」は Sysmac Studio により自動的に設定されます。
*2. ユーザ定義変数の場合、配列変数の要素の開始番号が 0, 配列要素「N」が 358 以下となるよう作成してください。

機能説明

- 本命令は、「Execute」（起動）の立ち上がりで、入出力変数に指定した「CamProperty」（カムプロパティ）と「CamNodes」（カムノード）の値をもとにカムデータを演算します。
- 入出力変数に指定した「CamTable」（カムテーブル）を、演算したカムデータの値に書き換えます。
- 「CamProperty」（カムプロパティ）と「CamNodes」（カムノード）の各項目は、Sysmac Studio のカムエディタの各設定項目と一致しています。
- カムテーブルの書き換えが完了すると、カムテーブルの終点インデックスを更新し、カム終点の要素番号を「EndPointIndex」（終点インデックス）に出力します。
  本命令の完了後に MC_SetCamTableProperty（カムテーブルプロパティ更新）命令、MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令を実行する必要はありません。
- カムテーブルの生成処理中は、モーション制御システム変数の MC 共通変数にある _MC_COM.Status.GenerateCamBusy（カムテーブル生成実行中）が TRUE となります。



使用上の注意

- 指定したカムテーブルの最大カムデータ数は、本命令が生成するカムテーブルのカムデータ数以上としてください。
- Sysmac Studio のカムエディタでは、曲線の形状として「自由曲線」を指定できますが、本命令では「自由曲線」を指定できません。
- 本命令は、他の命令に比べて著しく処理時間が長くなります。また、CPU ユニットの演算負荷量により処理時間が大きく異なる場合があります。そのため、本命令の完了をトリガとして次命令を起動する場合、次命令の起動タイミングに注意してください。
- Sysmac Studio のカムエディタでの設定値と本命令での設定項が同じ場合でも、内部の演算処理の相違により、生成されるカムデータの値が異なるときがあります。
- カムデータ変数はグローバル変数であり、複数のタスクからカムデータ変数の値を参照や変更をすることができます。複数のタスクからカムデータ変数の値を変更する場合、複数のタスクによる書き換え処理で競合が発生しないようにプログラムを作成してください。
- 「グローバル変数のタスク間排他制御」機能を用いてカムデータ変数を排他制御する場合、排他対象のカムデータ変数を用いてモーション命令を起動しないでください。軽度フォールトレベルの異常「カムテーブル指定不正（エラーコード: 5439 Hex）」が発生します。
- 本命令の入出力変数「CamProperty」（カムプロパティ）および「CamNodes」（カムノード）に指定する変数をユーザ定義変数として作成した場合、変数の保持属性の初期値は「非保持」です。変数の値を変更し、プログラムモードへの切り替え後や電源再投入後に再度使用するときは、変数の保持属性を「保持」に設定してください。
  Sysmac studio で作成したカムプロパティ変数およびカムノード変数は、保持属性が「保持」で固定されています。
- 本命令で生成されたカムデータ変数は、CPU ユニットの不揮発性メモリに保存されません。不揮発性メモリに保存する場合は、MC_SaveCamTable（カムテーブル保存）命令を実行してください。
- 本命令の実行中に、「CamNodes」（カムノード）に指定した配列変数の値を変更しないでください。本命令が意図しない動作を起こす可能性があります。
- 「CamNodes」（カムノード）の配列サイズが大きいと、命令実行時間が長くなり、タスク周期を超えてしまう場合があります。その場合、タスク周期オーバー（エラーコード: 6001 Hex）となりますので、配列サイズを調整するか、タスク周期を変更してください。
- 本命令の実行中に、運転モードからプログラムモードに切り替えた場合でも、カムテーブルの生成処理は継続しています。この状態で再度運転モードに切り替え、カムテーブルを対象とする命令を実行すると、軽度フォールトレベルの異常「モーション命令多重起動不可（エラーコード: 543C Hex）」が発生します。

参考

- カムテーブルの最大カムデータ数は、Sysmac Studio を使用してグローバル変数のデータ型で確認するか、あるいはユーザプログラムで SizeOfAry（配列要素数取得）命令を実行することにより確認できます。
- 本命令によって生成するカムテーブルのカムデータ数は、次の式で計算できます。

$$T_{cd} = \sum_{k=0}^{n-1} \left(\frac{M_k - M_{k-1}}{P_k} \right) + 1$$

T_{cd} ： カムテーブルのカムデータ数
k ： カムノード変数の配列の要素番号
M_k ： カムノード変数の配列の要素番号 k の「Phase」（主軸位相）
ただし、M_{k-1} の「Phase」（主軸位相）は「0」とします。
P_k ： カムノード変数の配列の要素番号 k の「PhasePitch」（位相間隔幅）
n ： ノードポイント数

（注）「Phase」（主軸位相）／「PhasePitch」（位相間隔幅）が割り切れない場合、小数第 1 位を切り上げてください。

- 本命令によって生成するカムテーブルのカムデータ数は、65535 以下としてください。
- 生成されたカムテーブルのカムデータ数は、「EndPointIndex」（終点インデックス）の値で確認できます。表示器などで「EndPointIndex」（終点インデックス）の値を表示して、カムテーブルの分解能の改善やノードポイントの追加などの調整に利用することができます。
- カムテーブルのカムデータの詳細については、「MC_CamIn (P.3-160)」命令を参照してください。
- 終点インデックスの詳細については、「MC_SetCamTableProperty (P.5-2)」命令を参照してください。
- 「CamProperty」（カムプロパティ）と「CamNodes」（カムノード）の各設定項目については、『Sysmac Studio Version 1 オペレーションマニュアル(SBCA-470)』を参照してください。
- 曲線の形状と接続速度、接続加速度との関係については、『Sysmac Studio Version 1 オペレーションマニュアル(SBCA-470)』を参照してください。

● CPU ユニットの動作モードとの関係

本命令によるカムテーブルの生成処理中に、CPU ユニットの動作モードをプログラムモードに切り替えた後も、カムテーブルの生成処理は継続します。

● オンラインエディットによる命令削除時

本命令の実行中に、オンラインエディットにより本命令を削除してもカムテーブルの生成処理は継続します。

● MC 試運転の実行時

本命令によるカムテーブルの生成処理中に MC 試運転を実行した場合は、本命令の出力変数「CommandAborted」（実行中断）が TRUE となりますが、カムテーブルの生成処理は継続します。

命令の詳細

以下に本命令の詳細について説明します。

● カムプロパティ構造体（_sMC_CAM_PROPERTY 型）

カムプロパティ構造体（_sMC_CAM_PROPERTY 型）は、プロパティを指定するための構造体型データ型です。各メンバは、Sysmac Studio のカムエディタで設定する「カムテーブルのプロパティ」の項目の一部に相当します。
本命令では、入出力変数の「CamProperty」（カムプロパティ）に指定する変数として使用します。
次に、カムプロパティ構造体のメンバについて説明します。

変数	名称	データ型	有効範囲	内容
InitVel	初期速度	REAL	負数 *1、正数 *1、「0」	開始ノード（位相=「0」、変位=「0」）での、動作開始時の速度を指定します。 開始ノードの次のカムノードで指定した曲線形状が 3 次曲線、または 5 次曲線の場合に、初期速度が有効となります。 単位は［指令単位／s］となります。
InitAcc	初期加速度	REAL	負数 *1、正数 *1、「0」	開始ノード（位相=「0」、変位=「0」）での、動作開始時の加速度を指定します。 開始ノードの次のカムノードで指定した曲線形状が 5 次曲線の場合に、初期加速度が有効となります。 単位は［指令単位／s^2］となります。
CycleTime	サイクルタイム	REAL	正数 *1	カム動作の 1 サイクルの時間を指定します。 単位は[s]となります。

*1. 絶対値が 0.001 以上となるように指定してください。小数第 4 位は四捨五入されます。

● カムノード構造体（_sMC_CAM_NODE 型）

カムノード構造体（_sMC_CAM_NODE 型）は、曲線の定義を指定するための構造体型データ型です。各メンバは、Sysmac Studio のカムエディタで設定する［カムノード］の項目に相当します。



本命令では、入出力変数の「CamNodes」（カムノード）に指定する配列変数として使用します。
カムノードの配列変数の要素を「ノードポイント」と呼びます。配列変数の要素数は、設定するノードポイント数以上としてください。
カムプロファイル曲線の開始点（位相=「0」、変位=「0」）のノードポイントを「開始ノード」と呼びます。配列変数には、開始ノードを除くノードポイントを要素番号順に指定します。
「Phase」（主軸位相）に、「0.001」以上の正数を指定すると有効ノードとなります。「0」を指定すると、以降のノードポイントが無効となります。ただし、要素番号が「0」の「Phase」（主軸位相）に「0」を指定すると異常となります。

次に、有効ノード数が 5、配列変数の要素数が 10 の場合の例を示します。

	要素番号	Phase（主軸位相）
有効ノード	0	正数
	...	...
	4	正数
無効ノード	5	0
	6	任意
	...	...
	9	任意

配列番号の昇順で「Phase」（主軸位相）の値が昇順となるように指定してください。（前回値＜今回値）

次に、カムノード構造体のメンバについて説明します。

変数	名称	データ型	有効範囲	内容
Phase	主軸位相	REAL	正数 *1 *2、「0」	ノードポイントでの主軸の位相の値を指定します。 単位は［指令単位］となります。
Distance	従軸変位	REAL	負数 *1 *2、正数 *1 *2、「0」	ノードポイントでの従軸の変位の値を指定します。 単位は［指令単位］となります。
Curve	曲線形状	_eMC_CAM_CURVE	0: _mcConstantLine 1: _mcStraightLine 2: _mcParabolic 3: _mcModifiedConstantVel 4: _mcModifiedTrapezoid 5: _mcModifiedSine 6: _mcCycloidal 7: _mcTrapecloid 8: _mcReverseTrapecloid 9: _mcSimpleHarmonic 10: _mcDoubleHarmonic 11: _mcReverseDoubleHarmonic 12: _mcNC2Curve 13: _mcPolynomic3 14: _mcPolynomic5	ノードポイントまでのカム曲線の形状を指定します。 0: 直線（変位維持）*3 1: 直線 2: 等加速度 3: 変形等速度 4: 変形台形 5: 変形正弦 6: サイクロイド 7: トラペクロイド 8: 逆トラペクロイド 9: 単弦 10: 複弦 11: 逆複弦 12: NC2 曲線 13: 3 次曲線 14: 5 次曲線
ConnectingVelEnable	接続速度有効	BOOL	TRUE, FALSE	TRUE にすると、指定した曲線形状が 3 次曲線、または 5 次曲線の場合に、指定した接続速度を有効にします。 FALSE にすると接続速度を無効にします。
ConnectingVel	接続速度	REAL	負数 *1、正数 *1、「0」	指定した曲線形状が 3 次曲線、または 5 次曲線の場合に、後続する曲線との接続部での速度を指定できます。 曲線と曲線をなめらかに接続する場合に指定します。 単位は［指令単位／s］となります。
ConnectingAccEnable	接続加速度有効	BOOL	TRUE, FALSE	TRUE にすると、指定した曲線形状が 5 次曲線の場合に、指定した接続加速度を有効にします。 FALSE にすると接続加速度を無効にします。
ConnectingAcc	接続加速度	REAL	負数 *1、正数 *1、「0」	指定した曲線形状が 5 次曲線の場合に、後続する曲線との接続部での加速度を指定できます。 曲線と曲線をなめらかに接続する場合に指定します。 単位は［指令単位／s^2］となります。
PhasePitch	位相間隔幅	REAL	正数 *1	指定した間隔幅でノードポイント間の位相を分割します。*4 単位は［指令単位］となります。

*1. 絶対値が 0.001 以上となるように指定してください。小数第 4 位は四捨五入されます。

*2. 「Phase」（主軸位相）と「Distance」（従軸変位）の有効桁は 7 桁です。8 桁以上を入力すると、有効桁 8 桁め以降は切り捨てとなります。切り捨ての結果、他の「Phase」（主軸位相）と同じ値となる場合は「カムノード主軸位相非昇順（エラーコード: 5740Hex）」となります。有効桁 7 桁以内で昇順となる値を入力してください。

*3. 直線（変位維持）を指定すると、「Distance」（従軸変位）は無効となり、ひとつ前のノードポイントで指定した値と同じ値として処理されます。配列の要素番号が 0 の場合に直線（変位維持）を指定すると、「Distance」（従軸変位）は 0 として処理されます。
*4. ノードポイントごとに生成されるカムデータのすべての合計が、65535 以下となるように指定してください。

● カムテーブルの生成例

本命令でのカムテーブルの生成例を説明します。
「CamNodes」（カムノード）に指定する配列変数の要素数は 4、「Curve」（曲線形状）は _mcStraightLine（直線）とします。「ConnectingVel」（接続速度）と「ConnectingAcc」（接続加速度）、および「CamProperty」（カムプロパティ）の「InitVel」（初期速度）と「InitAcc」（初期加速度）は無効となるため、記載を省略しています。「Phase」（主軸位相）と「Distance」（従軸変位）の値は、次の表のとおりとします。

CamNodes（カムノード）の配列変数の要素番号	Phase（主軸位相）	Distance（従軸変位）	Curve（曲線形状）	PhasePitch（位相間隔幅）
0	180.000	180.000	_mcStraightLine	0.100
1	360.000	0.000	_mcStraightLine	0.100
2	0.00	-	-	-
3	-	-	-	-

あらかじめ Sysmac studio のカムエディタで作成したカムデータ変数の要素数は 4000 とし、位相と変位の値は、すべて「不定値」とします。
本命令の実行前のカムテーブルのカムデータ変数は次の表のとおりです。

要素番号	Phase（位相）	Distance（変位）
0	（不定値）	（不定値）
1	（不定値）	（不定値）
...		
1799	（不定値）	（不定値）
1800	（不定値）	（不定値）
1801	（不定値）	（不定値）
...		
3599	（不定値）	（不定値）
3600	（不定値）	（不定値）
3601	（不定値）	（不定値）
...		
3999	（不定値）	（不定値）

本命令の実行によるカムデータ変数の変更過程を順に説明します。変更箇所は、背景を塗りつぶして表示しています。
本命令を実行すると、カムデータ変数の要素番号「0」の位相と変位を「0」に書き換えます。

要素番号	Phase（位相）	Distance（変位）
0	**0.0**	**0.0**
1	（不定値）	（不定値）
...		
1799	（不定値）	（不定値）
1800	（不定値）	（不定値）
1801	（不定値）	（不定値）
...		
3599	（不定値）	（不定値）
3600	（不定値）	（不定値）
3601	（不定値）	（不定値）
...		

要素番号	Phase（位相）	Distance（変位）
3999	（不定値）	（不定値）

次に、「CamNodes」（カムノード）の要素番号「0」の指定値に従い、開始ノードからノードポイントまでのカムデータ数および各カムデータの位相と変位を計算します。カムデータ数が「1800」となるため、カムデータ変数の要素番号「1」から「1800」までの位相と変位を書き換えます。

要素番号	Phase（位相）	Distance（変位）
0	0.0	0.0
1	**0.1**	**0.1**
...		
1799	**179.9**	**179.9**
1800	**180.0**	**180.0**
1801	（不定値）	（不定値）
...		
3599	（不定値）	（不定値）
3600	（不定値）	（不定値）
3601	（不定値）	（不定値）
...		
3999	（不定値）	（不定値）

同様に、「CamNodes」（カムノード）の要素番号「1」の指定値に従い、ノードポイント間のカムデータ数および各カムデータの位相と変位を計算します。カムデータ数が「1800」となるため、カムデータ変数の要素番号「1801」から「3600」までの位相と変位が書き換わります。

要素番号	Phase（位相）	Distance（変位）
0	0.0	0.0
1	0.1	0.1
...		
1799	179.9	179.9
1800	180.0	180.0
1801	**180.1**	**179.9**
...		
3599	**359.9**	**0.1**
3600	**360.0**	**0.0**
3601	（不定値）	（不定値）
...		
3999	（不定値）	（不定値）

次に、「CamNodes」（カムノード）の要素番号「2」は無効ノードであるため、計算を終了します。カムデータ変数の要素番号「3601」以降のカムデータを無効とするために、位相の値を「0」に書き換えます。

要素番号	Phase（位相）	Distance（変位）
0	0.0	0.0
1	0.1	0.1
...		
1799	179.9	179.9
1800	180.0	180.0
1801	180.1	179.9
...		
3599	359.9	0.1
3600	360.0	0.0
3601	**0.0**	（不定値）

要素番号	Phase（位相）	Distance（変位）
...		
3999	（不定値）	（不定値）

ただし、Sysmac studio のカムエディタで作成したカムデータ変数の要素数が 3601 の場合、無効カムデータが存在しないため、位相の値を「0」に書き換えることはありません。

要素番号	Phase（位相）	Distance（変位）
0	0.0	0.0
1	0.1	0.1
...		
1799	179.9	179.9
1800	180.0	180.0
1801	180.1	179.9
...		
3599	359.9	0.1
3600	360.0	0.0

最後に書き換えたカムデータ変数の要素番号を、カムテーブルの終点インデックスの値として、本命令の出力変数「EndPointIndex」（終点インデックス）に出力します。この例では「3600」を出力します。

以上で本命令の実行は完了となります。

● カムテーブル生成変位オーバーフローが発生する場合

本命令で計算したカムデータの「Distance」（従軸変位）の値が、REAL 型の有効範囲を超えた場合、「カムテーブル生成変位オーバーフロー（エラーコード: 5742 Hex）」が発生します。このとき、該当する「Distance」（従軸変位）の値は更新せずに、カムテーブルの生成処理を中止します。
カムテーブル生成変位オーバーフローは、「Curve」（曲線形状）に 3 次曲線または 5 次曲線を指定した場合にだけ発生する可能性があります。この異常が発生した場合は、以降に記載した 3 次曲線の変位の計算方法または 5 次曲線の変位の計算方法を参照し、入出力変数「CamProperty」（カムプロパティ）および「CamNodes」（カムノード）の値を修正してください。

● 3 次曲線の変位の計算方法

「CamNodes」（カムノード）に指定する配列変数の要素番号を「m」とします。
要素番号が「m」の「Curve」（曲線形状）に 3 次曲線を指定したとき、要素番号「m-1」の「Phase」を主軸位相の初期値 P_{init} とします。同様に、「Distance」を従軸変位の初期値 d_{init} とします。m=0 のとき、P_{init} と d_{init} は「0」として計算します。また、要素番号「m」の「Phase」を主軸位相の終端値 P_{final} とします。同様に、要素番号「m」の「Distance」を従軸変位の終端値 d_{final} とします。
d_{init} から n 番目のカムデータの「Distance」（従軸変位）を d(n)とすると、d(n)は次の式となります。

0≦n<N の場合

$$d(n) = d_{init} + (d_{final} - d_{init}) \cdot \sum_{i=f}^{3} a_i \cdot \left(\frac{pitch}{(P_{final} - P_{init})} \cdot n \right)^i$$

n=N の場合

$d(n)=d_{final}$

ただし、「(P_{final} - P_{phase}) / pitch」が整数の場合

$$N = \frac{P_{final} - P_{phase}}{pitch}$$

「(P_{final} - P_{phase}) / pitch」が整数でない場合

$$N = floor\left(\frac{P_{final} - P_{phase}}{pitch}\right) + 1$$

$$a_1 = \frac{V_{init} \cdot T}{(d_{final} - d_{init})}$$

$$a_2 = 3 - (2 \cdot V_{init} + V_{fin}) \cdot \frac{T}{(d_{final} - d_{init})}$$

$$a_3 = (V_{init} + V_{fin}) \cdot \frac{T}{(d_{final} - d_{init})} - 2$$

$$T = CycleTime \cdot \frac{(P_{final} - P_{init})}{P_{max}}$$

pitch: 「CamNodes」（カムノード）の要素番号「m」の「PhasePitch」（位相間隔幅）
CycleTime: 「CamProperty」（カムプロパティ）の「CycleTime」（サイクルタイム）
P_{max}: 「CamNodes」（カムノード）の有効ノードでの「Phase」（主軸位相）の最大値
V_{init}: カムプロファイル曲線の初期速度［指令単位／s］

ただし、m=0 の場合

V_{init} = 「CamProperty」（カムプロパティ）の「InitVel」（初期速度）

m≧1 の場合で、「CamNodes」（カムノード）の要素番号「m-1」の「Curve」（曲線形状）が直線(_mcStraightLine)のとき

$$V_{init} = \frac{(\text{要素番号 m-1 の } d_{final} - \text{要素番号 m-1 の } d_{init})}{(\text{要素番号 m-1 の } P_{final} - \text{要素番号 m-1 の } P_{init})} \cdot \frac{P_{max}}{CycleTime}$$

m≧1 の場合で、「CamNodes」（カムノード）の要素番号「m-1」の「Curve」（曲線形状）が 3 次曲線または 5 次曲線のとき

- 「CamNodes」（カムノード）の要素番号「m-1」の「ConnectingVelEnable」（接続速度有効） が TRUE

  V_{init} = 「CamNodes」（カムノード）の要素番号「m-1」の「ConnectingVel」（接続速度）
- 「CamNodes」（カムノード）の要素番号「m-1」の「ConnectingVelEnable」（接続速度有効） が FALSE

  $V_{init} = 0$

m≧1 の場合で、上記以外のとき

$V_{init} = 0$

V_{fin}: カム曲線の終端速度［指令単位／s］

ただし、「CamNodes」（カムノード）の要素番号「m」の「ConnectingVelEnable」（接続速度有効）が TRUE の場合

V_{fin} = 「CamNodes」（カムノード）の要素番号「m」の「ConnectingVel」（接続速度）

「CamNodes」（カムノード）の要素番号「m」の「ConnectingVelEnable」（接続速度有効） が FALSE の場合

- 「CamNodes」（カムノード）の要素番号「m+1」の「Curve」（曲線形状）が直線(_mcStraightLine)

$$V_{fin} = \frac{(\text{要素番号 m+1 の } d_{final} - \text{要素番号 m+1 の } d_{init})}{(\text{要素番号 m+1 の } P_{final} - \text{要素番号 m+1 の } P_{init})} \cdot \frac{P_{max}}{CycleTime}$$

- 「CamNodes」（カムノード）の要素番号「m+1」の「Curve」（曲線形状）が直線(_mcStraightLine)以外

$V_{fin} = 0$

● 5 次曲線の変位の計算方法

「CamNodes」（カムノード）に指定する配列変数の要素番号を「m」とします。
要素番号が「m」の「Curve」（曲線形状）に 5 次曲線を指定したとき、要素番号「m-1」の「Phase」を主軸位相の初期値 P_{init} とします。同様に、「Distance」を従軸変位の初期値 d_{init} とします。m=0 のとき、P_{init} と d_{init} は「0」として計算します。また、要素番号「m」の「Phase」を主軸位相の終端値 P_{final} とします。同様に、要素番号「m」の「Distance」を従軸変位の終端値 d_{final} とします。d_{init} から n 番目のカムデータの「Distance」（従軸変位）を d(n)とすると、d(n)は次の式となります。

0≦n<N の場合

$$d(n) = d_{init} + (d_{final} - d_{init}) \cdot \sum_{i=f}^{5} a_i \cdot \left(\frac{pitch}{(P_{final} - P_{init})} \cdot n\right)^i$$

n=N の場合

$d(n) = d_{final}$

ただし、「(P_{final} - P_{phase}) / pitch」が整数の場合

$$N = \frac{P_{final} - P_{phase}}{pitch}$$

「(P_{final} - P_{phase}) / pitch」が整数でない場合

$$N = floor\left(\frac{P_{final} - P_{phase}}{pitch}\right) + 1$$

$$a_1 = \frac{V_{init} \cdot T}{(d_{final} - d_{init})}$$

$$a_2 = \frac{A_{init} \cdot T^2}{2 \cdot (d_{final} - d_{init})}$$

$$a_3 = \frac{-\left((3A_{init} - A_{fin}) \cdot \frac{T^2}{(d_{final} - d_{init})} + (8V_{fin} + 12V_{init}) \cdot \frac{T}{(d_{final} - d_{init})} - 20\right)}{2}$$

$$a_4 = \left((1.5A_{init} - A_{fin}) \cdot \frac{T^2}{(d_{final} - d_{init})} + (7V_{fin} + 8V_{init}) \cdot \frac{T}{(d_{final} - d_{init})} - 15\right)$$

$$a_5 = \frac{-\left((A_{init} - A_{fin}) \cdot \frac{T^2}{(d_{final} - d_{init})} + (6V_{fin} + 6V_{init}) \cdot \frac{T}{(d_{final} - d_{init})} - 12\right)}{2}$$

$$T = CycleTime \cdot \frac{(P_{final} - P_{init})}{P_{max}}$$

pitch: 「CamNodes」（カムノード）の要素番号「m」の「PhasePitch」（位相間隔幅）
CycleTime: 「CamProperty」（カムプロパティ）の「CycleTime」（サイクルタイム）
P_{max}: 「CamNodes」（カムノード）の有効ノードでの「Phase」（主軸位相）の最大値

A_{init}: カムプロファイル曲線の初期加速度［指令単位／s^2］

ただし、m=0 の場合

A_{init} = 「CamProperty」（カムプロパティ）の「InitAcc」（初期加速度）

m≧1 の場合で、「CamNodes」（カムノード）の要素番号「m-1」の「Curve」（曲線形状）が 5 次曲線のとき

- 「CamNodes」（カムノード）の要素番号「m-1」の「ConnectingVelEnable」（接続速度有効） が TRUE

  A_{init} = 「CamNodes」（カムノード）の要素番号「m-1」の「ConnectingAcc」（接続加速度）
- 「CamNodes」（カムノード）の要素番号「m-1」の「ConnectingVelEnable」（接続速度有効） が FALSE

  $A_{init} = 0$

m≧1 の場合で、「CamNodes」（カムノード）の要素番号「m-1」の「Curve」（曲線形状）が 5 次曲線以外のとき

$A_{init} = 0$

A_{fin}: カム曲線の終端加速度［指令単位／s^2］
ただし、「CamNodes」（カムノード）の要素番号「m」の「ConnectingAccEnable」（接続加速度有効） が TRUE の場合

A_{fin} = 「CamNodes」（カムノード）の要素番号「m」の「ConnectingAcc」（接続加速度）

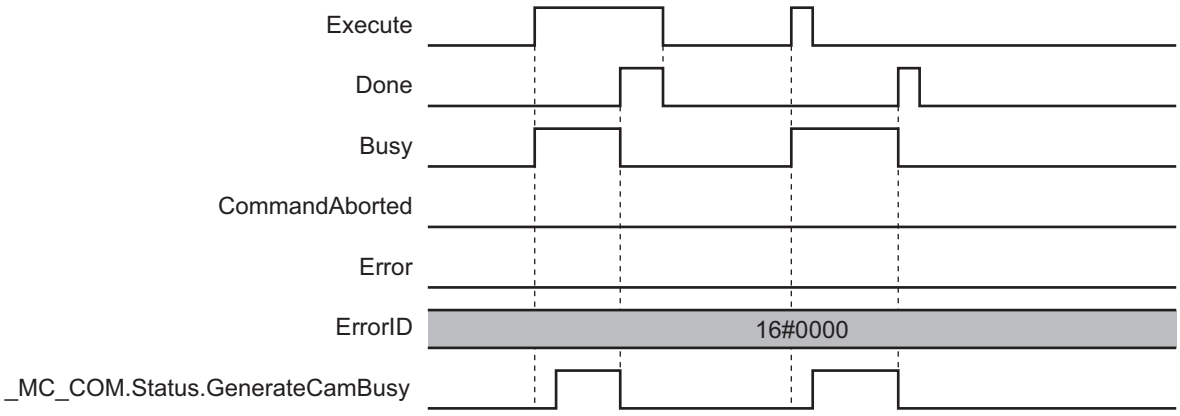
「CamNodes」（カムノード）の要素番号「m」の「ConnectingAccEnable」（接続加速度有効） が FALSE の場合

$A_{fin} = 0$

V_{init}: カムプロファイル曲線の初期速度［指令単位／s］、および V_{fin}: カム曲線の終端速度［指令単位／s］については、「3 次曲線の変位の計算方法 (P.5-32)」を参照してください。

タイミングチャート

「Execute」（実行）の立ち上がりと同時に「Busy」（実行中）が TRUE となり、次の周期でカムテーブル生成実行中(_MC_COM.Status.GenerateCamBusy)が TRUE となります。



モーション命令再起動

本命令の再起動はできません。
再起動を行った場合、「モーション命令再起動不可（エラーコード: 543B Hex)」となります。ただし、実行中のカムテーブル生成処理は継続します。

モーション命令多重起動

● 他命令実行中の本命令起動

本命令と同一の「CamTable」（カムテーブル）を使用する以下の命令がすでに起動されている場合、本命令を多重起動することはできません。

- MC_CamIn（カム動作開始）命令
- MC_SaveCamTable（カムテーブル保存）命令
- MC_SetCamTableProperty（カムテーブルプロパティ更新）命令
- MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令

● 本命令実行中の他命令起動

本命令実行中に、同一の「CamTable」（カムテーブル）を使用する以下の命令を多重起動することはできません。

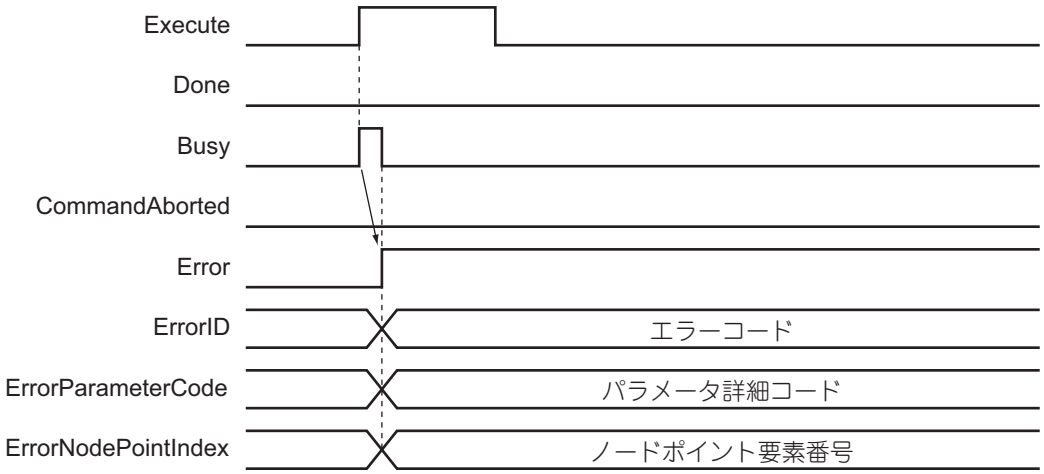
- MC_CamIn（カム動作開始）命令
- MC_SaveCamTable（カムテーブル保存）命令
- MC_SetCamTableProperty（カムテーブルプロパティ更新）命令
- MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令

● 本命令実行中の本命令起動

本命令実行中に、本命令の別のインスタンスを起動することはできません。また、カムテーブル生成実行中(_MC_COM.Status.GenerateCamBusy)が TRUE のとき、本命令を起動することはできません。カムテーブル生成実行中(_MC_COM.Status.GenerateCamBusy)または本命令の出力変数を使用し排他処理を行ってください。
本命令の複数のインスタンスを同じ周期に起動する場合、その周期ではカムテーブル生成実行中(_MC_COM.Status.GenerateCamBusy)は FALSE となっています。別のインスタンスの「Busy」(実行中）を使用し排他処理を行ってください。

異常

本命令起動時に異常が発生した場合、「Error」(エラー）が TRUE となります。
「ErrorID」(エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。
また、「ErrorID」(エラーコード）によっては、付属情報が「ErrorParameterCode」(パラメータ詳細コード）と「ErrorNodePointIndex」(ノードポイント要素番号）に出力されます。



本命令起動時に異常が発生した場合でも、「CamTable」(カムテーブル）に指定したカムデータ変数は、本命令の実行前の値を保持しています。
ただし、次の場合はカムデータ変数の値を保持していません。

- 「Execute」(実行）を立ち上げた後に、「CamNodes」(カムノード）に指定したパラメータを変更した場合
- 本命令で計算したカムデータの「Distance」(従軸変位）の値が、REAL 型の有効範囲を超えた場合

カムデータ変数の値が保持されていない場合、カムテーブルが不正状態となります。不正状態のカムテーブルを指定し、以下の命令を実行すると異常が発生します。

- MC_CamIn（カム動作開始）命令
- MC_SaveCamTable（カムテーブル保存）命令
- MC_SetCamTableProperty（カムテーブルプロパティ更新）命令
- MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令

カムテーブルの不正状態を解除するには、命令で発生中の異常を解除した後に、次のいずれかの操作を実行してください。

- 不正状態のカムテーブルに対して本命令を再度実行しカムテーブルの生成を完了
- Sysmac Studio でカムテーブルをダウンロード
- CPU ユニットの電源を再投入

プログラムモードまたは MC 試運転中に、実行中の本命令が異常となった場合、「ErrorID」（エラーコード）にエラーコードが出力されません。この場合は、イベントログ機能またはモーション制御システム変数の MC 共通 軽度フォールトコード(_MC_COM.MFaultLvl.Code)により、異常の原因を確認してください。

● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

サンプルプログラム

ここでは、Sysmac Studio のカムエディタで作成したカムテーブルのカムデータ変数を書き換えるサンプルプログラムを掲載します。

パラメータ設定

このサンプルプログラムで使用する必要最低限の設定を以下に示します。

● **軸パラメータ**

軸種別

軸	軸種別
軸 1	サーボ軸
軸 2	サーボ軸

カウンタモード

軸	カウンタモード
軸 1	ロータリモード
軸 2	ロータリモード

リングカウンタ

軸	上限値	下限値
軸 1	360	0
軸 2	360	0

座標単位選択

軸	座標単位
軸 1	degree
軸 2	degree

書き換え内容

Sysmac Studio のカムエディタで、あらかじめ作成したカムプロパティ変数とカムノード変数の初期値、およびこのサンプルプログラムで書き換える値を説明します。

● **カムプロパティ**

カムプロパティ変数の初期値を次に示します。

InitVel（初期速度）	InitAcc（初期加速度）	CycleTime（サイクルタイム）
0.000	0.000	1.000

このサンプルプログラムでは、「CycleTime」（サイクルタイム）を「0.8」に変更します。

InitVel（初期速度）	InitAcc（初期加速度）	CycleTime（サイクルタイム）
0.000	0.000	**0.800**

● カムノード

カムノード変数の初期値を次に示します。

要素番号	Phase（主軸位相）	Distance（従軸変位）	Curve（曲線形状）	ConnectingVelEnable（接続速度有効）	ConnectingVel（接続速度）	ConnectingAccEnable（接続加速度有効）	ConnectingAcc（接続加速度）	PhasePitch（位相間隔幅）
0	180.000	-100.000	_mcModifiedSine	FALSE	0.000	FALSE	0.000	0.010
1	360.000	0.000	_mcPolynomic5	FALSE	0.000	FALSE	0.000	0.010
2	0.0	0.0	_mcConstantLine	FALSE	0.000	FALSE	0.0	0.0
3	0.0	0.0	_mcConstantLine	FALSE	0.000	FALSE	0.0	0.0

このサンプルプログラムでは、要素番号「0」の曲線形状を変更し、要素番号「1」にノードポイントを追加します。

要素番号	Phase（主軸位相）	Distance（従軸変位）	Curve（曲線形状）	ConnectingVelEnable（接続速度有効）	ConnectingVel（接続速度）	ConnectingAccEnable（接続加速度有効）	ConnectingAcc（接続加速度）	PhasePitch（位相間隔幅）
0	180.000	-100.000	**_mcPolynomic5**	FALSE	0.000	FALSE	0.000	0.010
1	**200.000**	**-102.000**	**_mcStraightLine**	FALSE	0.000	FALSE	0.000	0.010
2	360.000	0.000	_mcPolynomic5	FALSE	0.000	FALSE	0.000	0.010
3	0.0	0.0	_mcConstantLine	FALSE	0.000	FALSE	0.0	0.0

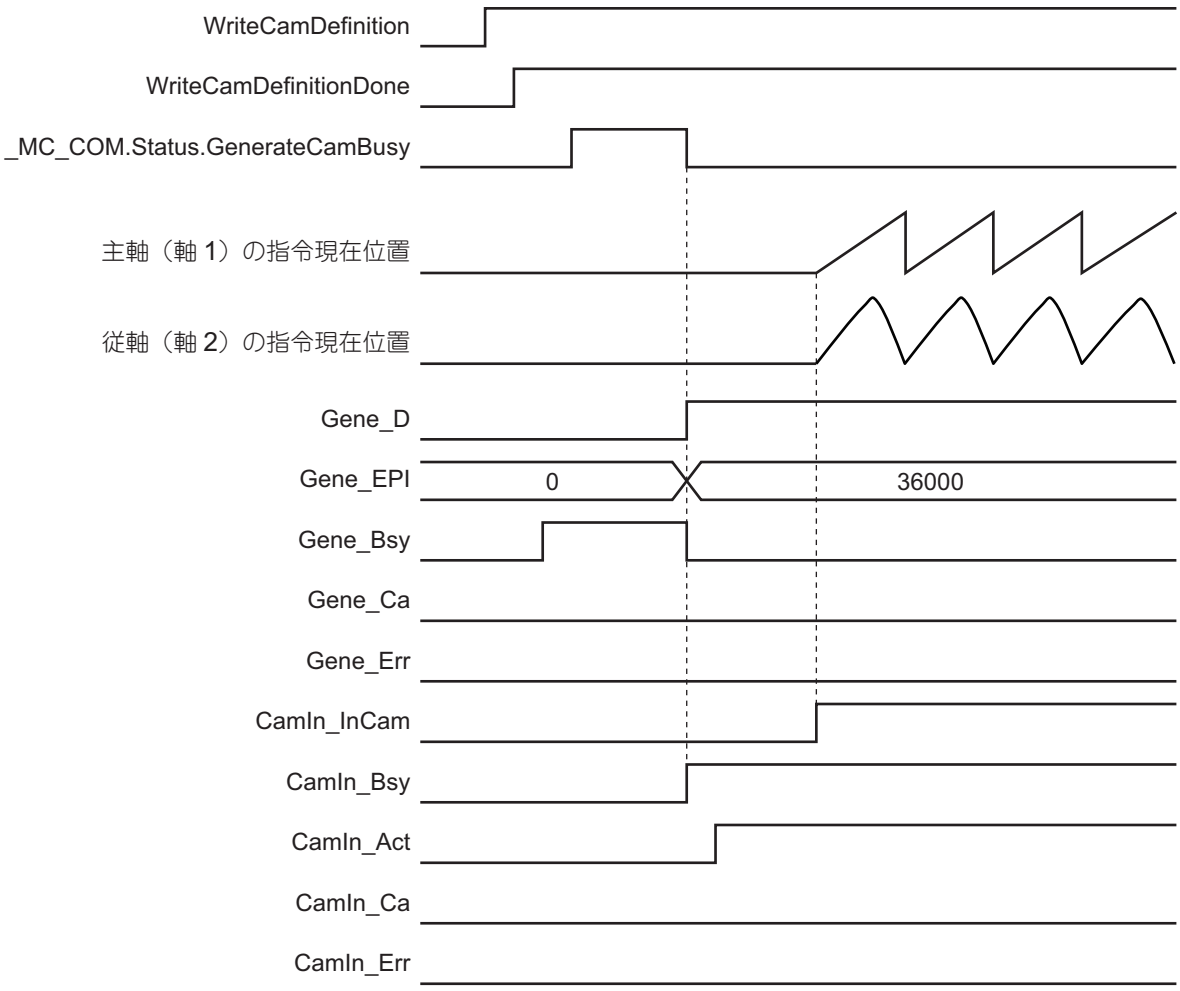
ラダー図

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	主軸である軸 1 の軸変数です。
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
CamProfile0	ARRAY[0..36000] OF _sMC_CAM_REF	−	カムデータ変数です。 あらかじめ、Sysmac Studio のカムエディタで作成しておきます。
Pwr1_Status	BOOL	−	MC_Power のインスタンス PWR1 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。

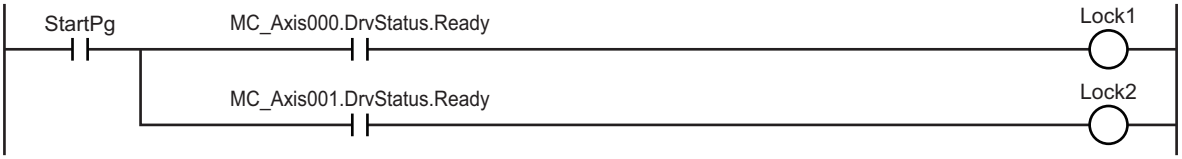
名称	データ型	初期値	コメント
Pwr2_Status	BOOL	−	MC_Power のインスタンス PWR2 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	−	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
WriteCamDefinition	BOOL	−	この変数が TRUE のときにカムプロパティ変数とカムノード変数の値を書き換えます。
CamProperty0	_sMC_CAM_PROPERTY	−	カムプロパティ変数です。
CamNode0	ARRAY[0..3] OF _sMC_CAM_NODE	−	カムノード変数です。
_MC_COM.Status.GenerateCamBusy	BOOL	−	モーション制御システム変数です。カムテーブルの生成処理中に TRUE となります。

● **タイミングチャート**

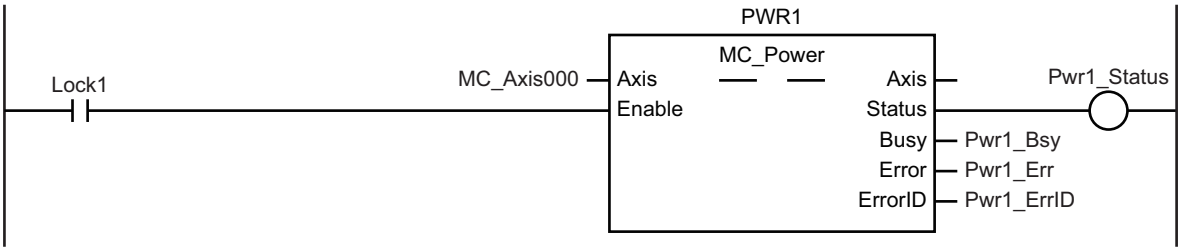


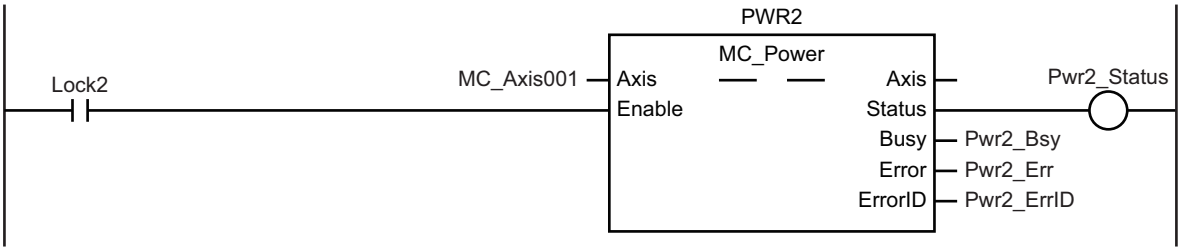
● サンプルプログラム

接点 StartPg が TRUE になったら、各軸のサーボドライバがサーボレディ状態になっていることを確認する

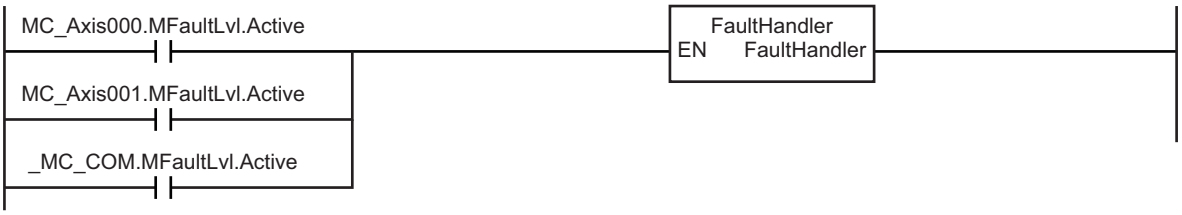


サーボレディ状態になっていれば、各軸をサーボ ON 状態にする

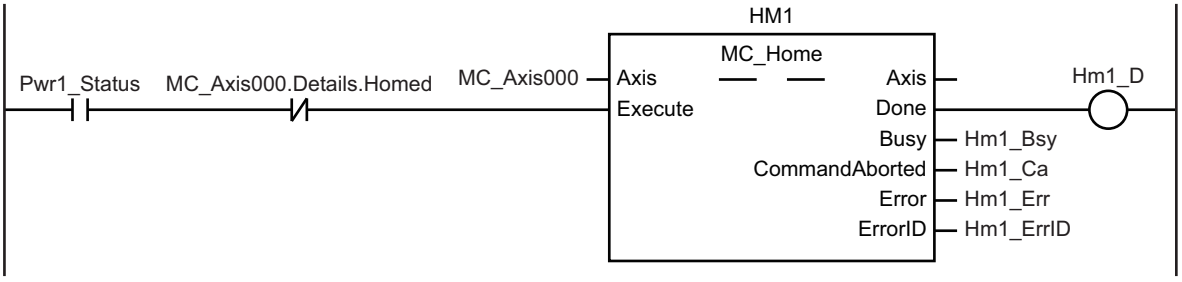




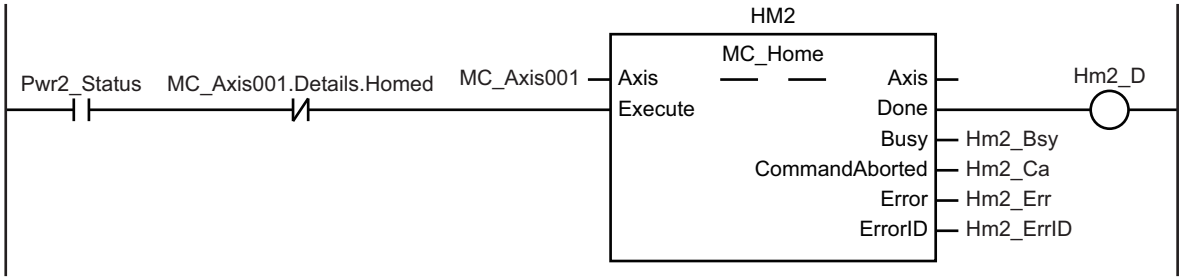
軸または MC 共通で軽度フォールトレベルの異常が発生したら、装置における異常発生時の処理（FaultHandler）を実行する。異常発生時の処理 (FaultHandler) は、装置にあわせてお客様でプログラミングしてください



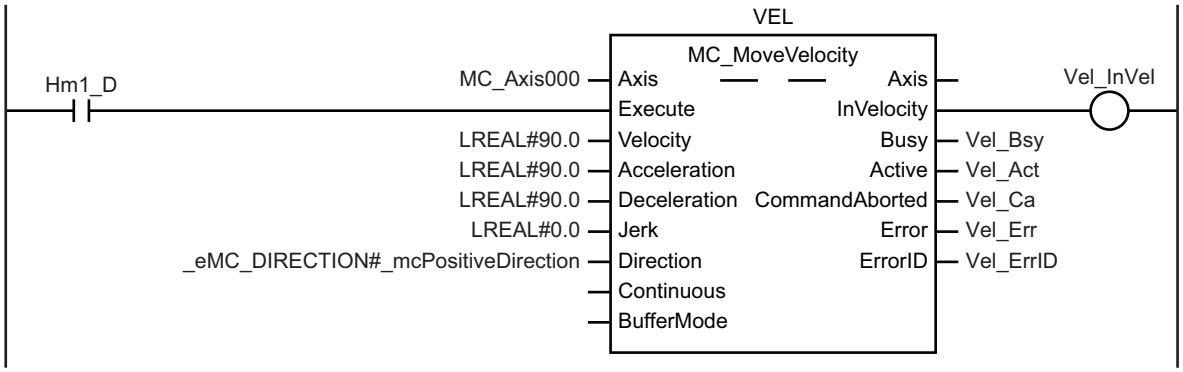
主軸（軸 1）がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



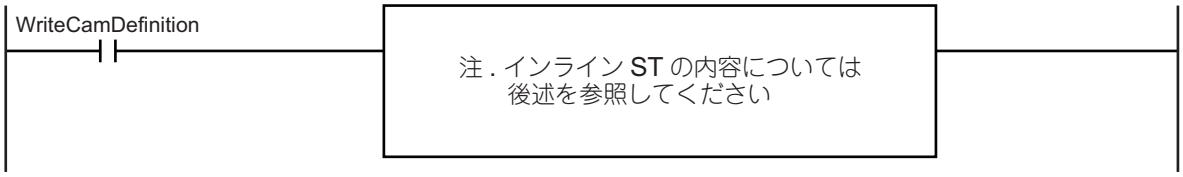
従軸（軸 2）がサーボ ON 状態になり原点未確定状態なら、原点復帰を実行し、原点を確定する



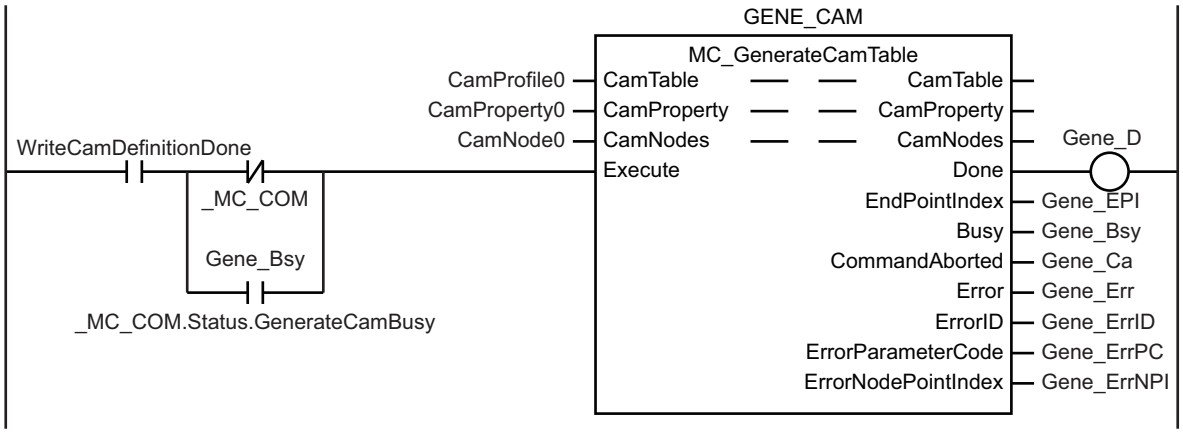
主軸（軸 1）が原点復帰完了後、MC_MoveVelocity（速度制御）命令を起動する



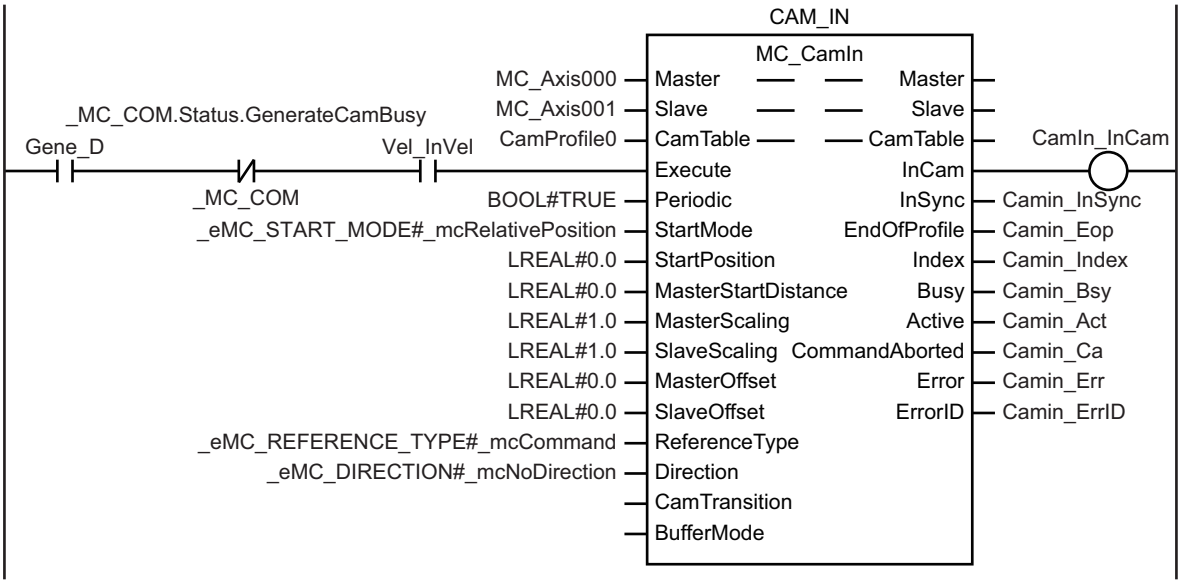
WriteCamDefinition が TRUE で、MC_GenerateCamTable（カムテーブル生成）命令が未実行状態のとき、カムプロパティとカムノードを書き換える。書き換え後に WriteCamDefinitionDone を TRUE とする



WriteCamDefinitionDone が TRUE で、カムテーブル生成処理中以外のとき、MC_GenerateCamTable（カムテーブル生成）命令を起動する



カムテーブル生成処理が完了し、主軸（軸 1）が目標速度に到達すれば、MC_CamIn（カム動作開始）命令を起動する



インライン ST の内容

```
CamProperty0.CycleTime := REAL#0.800;
CamNode0[0].Curve := _eMC_CAM_CURVE#_mcPolynomic5;
CamNode0[0].ConnectingVelEnable := FALSE;
CamNode0[0].ConnectingVel := REAL#0.000;
CamNode0[0].ConnectingAccEnable := FALSE;
CamNode0[0].ConnectingAcc := REAL#0.000;
CamNode0[1].Phase := REAL#200.000;
CamNode0[1].Distance := REAL#-102.000;
CamNode0[1].Curve := _eMC_CAM_CURVE#_mcStraightLine;
CamNode0[1].ConnectingVelEnable := FALSE;
CamNode0[1].ConnectingVel := REAL#0.000;
CamNode0[1].ConnectingAccEnable := FALSE;
CamNode0[1].ConnectingAcc := REAL#0.000;
CamNode0[2].Phase := REAL#360.000;
CamNode0[2].Distance := REAL#0.000;
CamNode0[2].Curve := _eMC_CAM_CURVE#_mcPolynomic5;
CamNode0[2].ConnectingVelEnable := FALSE;
CamNode0[2].ConnectingVel := REAL#0.000;
CamNode0[2].ConnectingAccEnable := FALSE;
CamNode0[2].ConnectingAcc := REAL#0.000;
CamNode0[2].PhasePitch := REAL#0.010;
CamNode0[3].Phase := REAL#0.000;
WriteCamDefinitionDone := TRUE;
```

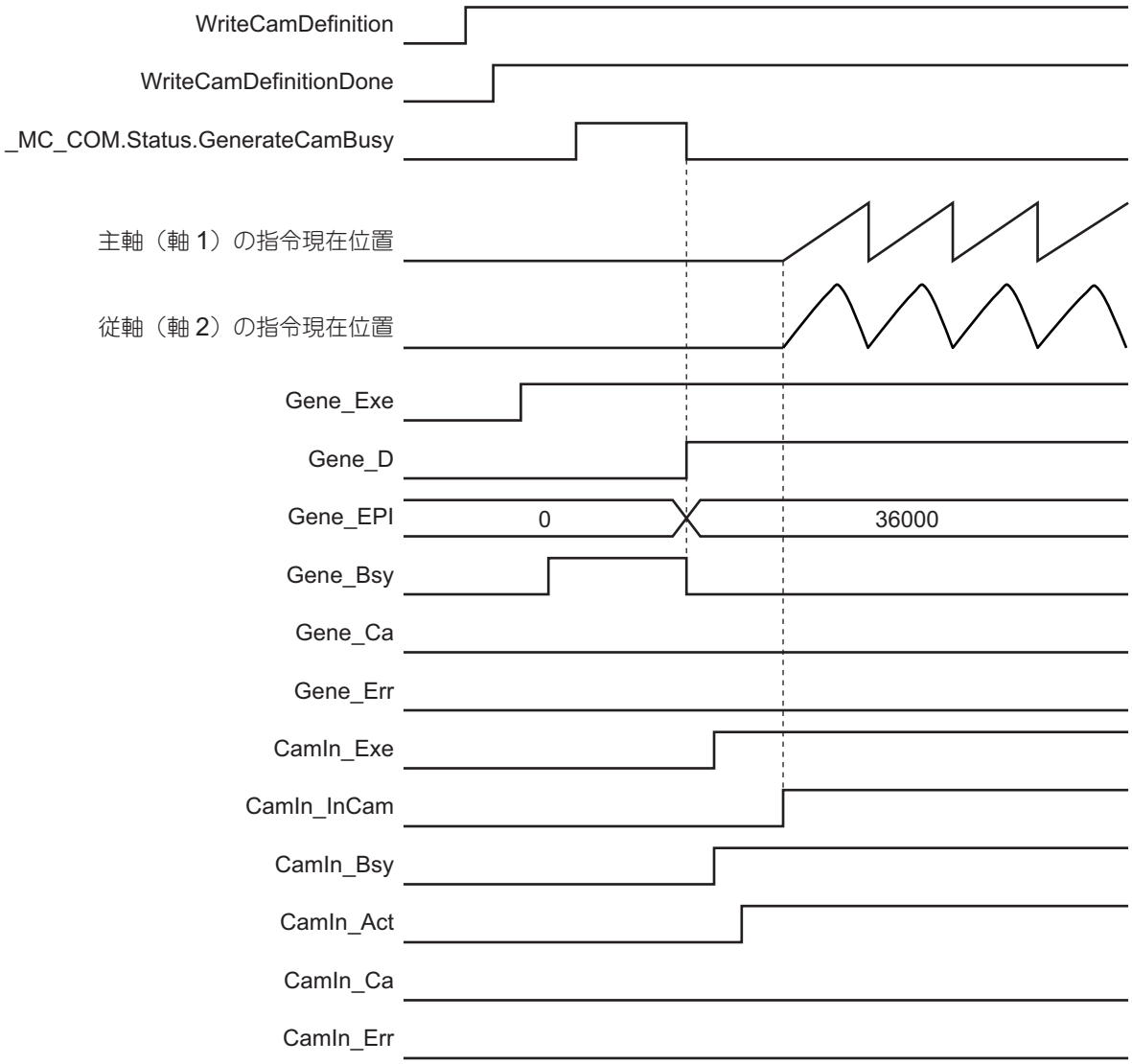
ストラクチャードテキスト(ST)

● 主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	－	主軸である軸 1 の軸変数です。

名称	データ型	初期値	コメント
MC_Axis001	_sAXIS_REF	−	従軸である軸 2 の軸変数です。
CamProfile0	ARRAY[0..36000] OF _sMC_CAM_REF	−	カムデータ変数です。 あらかじめ、Sysmac Studio のカムエディタで作成しておきます。
Pwr1_Status	BOOL	−	MC_Power のインスタンス PWR1 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
Pwr2_Status	BOOL	−	MC_Power のインスタンス PWR2 の出力「Status」に割り付けた変数です。サーボ ON 状態になると、この変数が TRUE となります。
StartPg	BOOL	−	この変数が TRUE で、EtherCAT のプロセスデータ通信が確立状態なら、サーボ ON 状態にします。
WriteCamDefinition	BOOL	−	この変数が TRUE のときにカムプロパティ変数とカムノード変数の値を書き換えます。
CamProperty0	_sMC_CAM_PROPERTY	−	カムプロパティ変数です。
CamNode0	ARRAY[0..3] OF _sMC_CAM_NODE	−	カムノード変数です。
_MC_COM.Status.GenerateCamBusy	BOOL	−	モーション制御システム変数です。カムテーブルの生成処理中に TRUE となります。

● タイミングチャート



● サンプルプログラム

```
//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 1 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
ELSE
    Pwr1_En:=FALSE;
END_IF;

//StartPg が TRUE の場合に、サーボドライバがサーボレディ状態になっていることを確認し、軸 2 をサーボ ON 状態にします
//サーボレディ状態になっていない場合は、サーボ OFF にします
IF (StartPg=TRUE)
 AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE;
```

```
ELSE
    Pwr2_En:=FALSE;
END_IF;

//軸またはMC共通で軽度フォールトが発生したら、異常時処理FaultHandlerを実行します
//異常発生時の処理(FaultHandler)は、装置にあわせてお客様でプログラミングしてください
IF (MC_Axis000.MFaultLvl.Active=TRUE)
OR (MC_Axis001.MFaultLvl.Active=TRUE)
OR (_MC_COM.MFaultLvl.Active=TRUE) THEN
    FaultHandler();
END_IF;

//軸1が運転可状態で、かつ原点未確定状態のとき、原点復帰を実行します
IF (Pwr1_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm1_Ex:=TRUE;
END_IF;

//軸2が運転可状態で、かつ原点未確定状態のとき、原点復帰を実行します
IF (Pwr2_Status=TRUE) AND (MC_Axis001.Details.Homed=FALSE) THEN
    Hm2_Ex:=TRUE;
END_IF;

//軸1が原点復帰完了後、MC_MoveVelocityを起動します
IF Hm1_D=TRUE THEN
    Vel_Ex := TRUE;
END_IF;

//WriteCamDefinitionがTRUEで、MC_GenerateCamTable（カムテーブル生成）命令が未実行状態のとき、カムプロパティとカムノードを書き換えます
//書き換え後にWriteCamDefinitionDoneをTRUEとします
IF(WriteCamDefinition = TRUE) THEN
    //カムプロパティの書き換え
    CamProperty0.CycleTime := REAL#0.800;

    //カムノードの書き換え
    CamNode0[0].Curve :=_eMC_CAM_CURVE#_mcPolynomic5;
    CamNode0[0].ConnectingVelEnable := FALSE;
    CamNode0[0].ConnectingVel := REAL#0.000;
    CamNode0[0].ConnectingAccEnable := FALSE;
    CamNode0[0].ConnectingAcc := REAL#0.000;
    CamNode0[1].Phase := REAL#200.000;
    CamNode0[1].Distance := REAL#-102.000;
    CamNode0[1].Curve :=_eMC_CAM_CURVE#_mcStraightLine;
    CamNode0[1].ConnectingVelEnable := FALSE;
    CamNode0[1].ConnectingVel := REAL#0.000;
    CamNode0[1].ConnectingAccEnable := FALSE;
    CamNode0[1].ConnectingAcc := REAL#0.000;
    CamNode0[2].Phase := REAL#360.000;
    CamNode0[2].Distance := REAL#0.000;
    CamNode0[2].Curve :=_eMC_CAM_CURVE#_mcPolynomic5;
    CamNode0[2].ConnectingVelEnable := FALSE;
```

```
    CamNode0[2].ConnectingVel := REAL#0.000;
    CamNode0[2].ConnectingAccEnable := FALSE;
    CamNode0[2].ConnectingAcc := REAL#0.000;
    CamNode0[2].PhasePitch := REAL#0.010;
    CamNode0[3].Phase := REAL#0.000;

    //WriteCamDefinitionDone を TRUE とします
    WriteCamDefinitionDone := TRUE;
END_IF;

//WriteCamDefinitionDone が TRUE で、カムテーブル生成処理中以外のとき、MC_GenerateCamTable（カムテーブル生成）命令を起動します
IF(WriteCamDefinitionDone = TRUE)
AND(_MC_COM.Status.GenerateCamBusy = FALSE) THEN
    Gene_Exe := TRUE;
END_IF;

//カムテーブル生成処理が完了し、主軸（軸 1）が目標速度に到達すれば、MC_CamIn（カム動作開始）命令を起動します
IF (Gene_D=TRUE)
AND(_MC_COM.Status.GenerateCamBusy = FALSE)
AND (Vel_InVel=TRUE) THEN
    Camin_Ex := TRUE;
END_IF;

// MC_GenerateCamTable
GENE_CAM(
    CamTable := CamProfile0,
    CamProperty := CamProperty0,
    CamNodes := CamNode0,
    execute := Gene_Exe,
    Done => Gene_D,
    EndPointIndex => Gene_EPI,
    Busy => Gene_Bsy,
    CommandAborted => Gene_CA,
    Error => Gene_Err,
    ErrorID => Gene_ErrID,
    ErrorParameterCode => Gene_ErrPC,
    ErrorNodePointIndex => Gene_ErrNPI
);

// 主軸（軸 1）の MC_Power
PWR1(
    Axis := MC_Axis000,
    Enable := Pwr1_En,
    Status => Pwr1_Status,
    Busy => Pwr1_Bsy,
    Error => Pwr1_Err,
    ErrorID => Pwr1_ErrID
);
```

```
// 従軸（軸 2）の MC_Power
PWR2(
    Axis := MC_Axis001,
    Enable := Pwr2_En,
    Status => Pwr2_Status,
    Busy => Pwr2_Bsy,
    Error => Pwr2_Err,
    ErrorID => Pwr2_ErrID
);

// 主軸（軸 1）の MC_Home
HM1(
    Axis := MC_Axis000,
    Execute := Hm1_Ex,
    Done => Hm1_D,
    Busy => Hm1_Bsy,
    CommandAborted => Hm1_Ca,
    Error => Hm1_Err,
    ErrorID => Hm1_ErrID
);

// 従軸（軸 2）の MC_Home
HM2(
    Axis := MC_Axis001,
    Execute := Hm2_Ex,
    Done => Hm2_D,
    Busy => Hm2_Bsy,
    CommandAborted => Hm2_Ca,
    Error => Hm2_Err,
    ErrorID => Hm2_ErrID
);

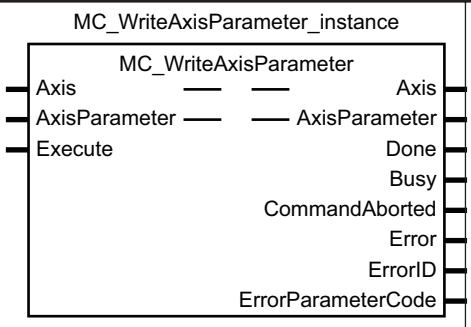
//MC_MoveVelocity
VEL(
    Axis := MC_Axis000,
    Execute := Vel_Ex,
    Velocity := Vel_Vel,
    Acceleration := Vel_Acc,
    Deceleration := Vel_Dec,
    Direction := Vel_Dir,
    InVelocity => Vel_InVel,
    Busy => Vel_Bsy,
    Active => Vel_Act,
    CommandAborted => Vel_Ca,
    Error => Vel_Err,
    ErrorID => Vel_ErrID
);

//MC_CamIn
CAM_IN(
    Master := MC_Axis000,
```

```
    Slave := MC_Axis001,
    CamTable := CamProfile0,
    Execute := Camin_Ex,
    Periodic := Camin_P,
    StartMode := Camin_Sm,
    StartPosition := Camin_Sp,
    MasterStartDistance := Camin_Msd,
    MasterScaling := Camin_Ms,
    SlaveScaling := Camin_Ss,
    MasterOffset := Camin_Mo,
    SlaveOffset := Camin_So,
    ReferenceType := Camin_Rt,
    Direction := Camin_Dir,
    InCam => Camin_InCam,
    InSync => Camin_InSync,
    EndOfProfile => Camin_Eop,
    Index => Camin_Index,
    Busy => Camin_Bsy,
    Active => Camin_Act,
    CommandAborted => Camin_Ca,
    Error => Camin_Err,
    ErrorID => Camin_ErrID
);
```

MC_WriteAxisParameter

MC 機能モジュールの軸パラメータ設定を書き換えます。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_WriteAxisParameter	軸パラメータ書込	FB		MC_WriteAxisParameter_instance (Axis :=「パラメータ」, AxisParameter :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」, ErrorParameterCode =>「パラメータ」);

バージョン関連情報

本命令は、ユニットバージョン Ver.1.08 以降の CPU ユニットと Sysmac Studio Ver.1.09 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。16#0000 は正常です。

出力変数	名称	データ型	有効範囲	内容
ErrorParameterCode	パラメータ詳細コード	WORD	*1	エラーコードによっては、付属情報が存在します。存在する場合は、異常となっているパラメータの詳細コードを出力します。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	異常発生により、本命令が中止されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	書き換え対象となる軸を指定します。*1
AxisParameter	軸パラメータ	_sAXIS_PARAM	−	書き換える値を指定します。*2

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. _sAXIS_PARAM 型のユーザ定義変数を定義してください。

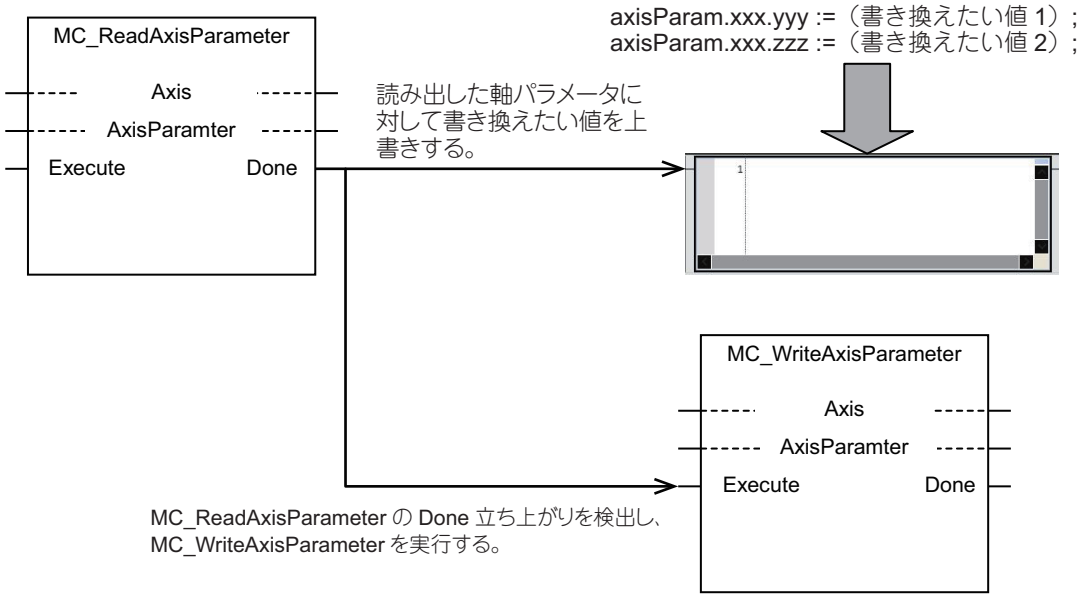
機能説明

- 本命令は、「Execute」（起動）の立ち上りで、指定された「Axis」（軸）の軸パラメータに、「AxisParameter」（軸パラメータ）で指定した値を書き込みます。
- 指定した軸の軸使用が未使用軸の場合だけ書き込みできます。これ以外で実行すると、実行時エラーとなり軸パラメータの書き込みを行いません。命令実行前の値が保持されます。
- MC_ChangeAxisUse（軸使用変更）命令により軸使用を切り替えた後に本命令を実行する場合は、MC_ChangeAxisUse（軸使用変更）命令の出力変数「Done」（完了）が TRUE となっていることを確認してから実行してください。
- 「AxisParameter」（軸パラメータ）が不正値の場合、または軸パラメータ間の整合がとれていない場合、実行時エラーとなり軸パラメータの書き込みを行いません。命令実行前の値が保持されます。

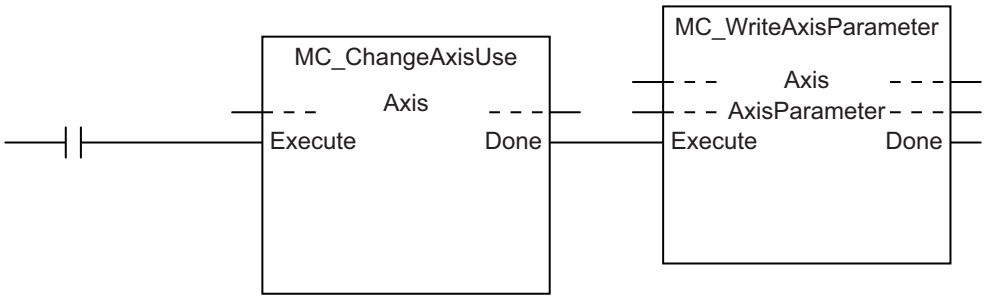
軸パラメータの設定可能範囲チェックの内容、および軸パラメータ間の整合チェックについては、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

使用上の注意

- 本命令で書き換えた値は CPU ユニットの不揮発性メモリに保存されません。そのため、CPU ユニットの電源 OFF、ダウンロード、MC 機能モジュールの再起動処理が開始されると、書き換えた値は消失し、Sysmac Studio で設定した値に戻ります。
- 本命令で書き換えた値は Sysmac Studio でアップロードできません。
- 本命令は書き込む軸パラメータだけでなく、書き込み対象の軸の軸パラメータすべてに対してパラメータの設定が必要です。対象パラメータについては、「書き込みや読み出し対象となる軸パラメータ (P.5-59)」を参照してください。変更が不要の軸パラメータは Sysmac Studio で設定した値、もしくは MC_ReadAxisParameter（軸パラメータ読出）命令により読み出した値を設定してください。
- 本命令を MC_ReadAxisParameter（軸パラメータ読出）命令と合わせて使用する場合の実行イメージは以下のとおりです。



- 本命令を MC_ChangeAxisUse（軸使用変更）命令と合わせて使用する場合の例を示します。MC_ChangeAxisUse（軸使用変更）命令の「Done」（完了）を確認してから本命令を実行してください。



- エンコーダ種別を［絶対値エンコーダ］(ABS)に設定した軸に対して、本命令を使用して［単位変換設定］や［位置カウント設定］を変更すると、機械の物理的な位置と MC 機能モジュール上の軸の位置の関係が変化します。この場合、MC_ChangeAxisUse（軸使用変更）命令で使用軸に変更したあとで MC_Home（原点復帰）命令を実行し、再度原点を確定してください。
- 使用軸に変更した時に「絶対値エンコーダ現在位置算出不可（エラーコード: 6458 Hex)」が発生した場合は、軸の異常を解除したのちに、MC_Home（原点復帰）命令を実行し、再度原点を確定してください。
- 軸パラメータの Reserved 変数は、「0」以外の値を指定しないでください。

命令の詳細

以下に本命令の詳細について説明します。

● _sAXIS_PARAM

_sAXIS_PARAM 型は、軸パラメータの値を指定するための構造体型データ型です。軸パラメータの区分ごとの構造体型データ型で構成されています。
本命令では、入出力変数の「AxisParameter」（軸パラメータ）に指定する変数として使用します。
次に、各メンバについて説明します。

メンバ変数	パラメータ名称	データ型	内容
UnitConversion	単位変換設定	_sAXIS_UNIT_CONVERSION_SETTINGS	単位変換設定の各メンバ変数に対して書き換える値を指定します。
Operation	動作設定	_sAXIS_OPERATION_SETTINGS	動作設定の各メンバ変数に対して書き換える値を指定します。
OtherOperation	拡張動作設定	_sAXIS_OTHER_OPERATION_SETTINGS	拡張動作設定の各メンバ変数に対して書き換える値を指定します。
Limit	リミット設定	_sAXIS_LIMIT_SETTINGS	リミット設定の各メンバ変数に対して書き換える値を指定します。
PosCount	位置カウント設定	_sAXIS_POSITION_COUNT_SETTINGS	位置カウント設定の各メンバ変数に対して書き換える値を指定します。
Homing	原点復帰設定	_sAXIS_HOMING_SETTINGS	原点復帰設定の各メンバ変数に対して書き換える値を指定します。
(Reserved)	（予備エリア）	ARRAY[0..255] OF BYTE	－

● _sAXIS_UNIT_CONVERSION_SETTINGS（単位変換設定）

_sAXIS_UNIT_CONVERSION_SETTINGS 型は、軸パラメータの単位変換設定の値を指定するための構造体型データ型です。
次に、各メンバについて説明します。

メンバ変数	パラメータ名称	データ型	有効範囲	内容
UnitDisplay	表示単位	_eMC_UNITS	0: _mcPls 1: _mcMm 2: _mcUm 3: _mcNm 4: _mcDeg 5: _mcInch	指令位置の単位を設定します。 0: pulse 1: mm 2: μm 3: nm 4: degree 5: inch
CmdPlsCountPerMotorRotation	モータ 1 回転のパルス数	UDINT	1～4294967295	エンコーダ分解能にあわせて、指令位置のモータ 1 回転あたりのパルス数を設定します。 電子ギア比に従い指令値からパルス量への変換を行います。
WorkTravelDistancePerMotorRotation	モータ 1 回転の移動量	LREAL	0.000000001～4294967295	指令位置のモータ 1 回転あたりのワークの移動量を設定します。
ReducerUse *1	減速機使用	BOOL	TRUE, FALSE	減速機設定を使用するかを指定します。 TRUE: 使用する FALSE: 使用しない

メンバ変数	パラメータ名称	データ型	有効範囲	内容
WorkTravelDistancePerWorkSideRotation *1 *2 *3	ワーク側 1 回転の移動量	LREAL	倍精度実数型の正数	ワーク側が 1 回転した場合の移動量を設定します。
WorkSideGearRatio *1 *2	ワーク側ギア比	UDINT	1～4294967295	ワーク側のギア比を設定します。
MotorSideGearRatio *1 *2	モータ側ギア比	UDINT	1～4294967295	モータ側のギア比を設定します。
(Reserved)	（予備エリア）	ARRAY [0..7] OF BYTE *4	－	－

*1. ユニットバージョン Ver.1.11 以降の CPU ユニットと Sysmac Studio Ver.1.15 以降の組み合わせで使用可能です。
*2. ［減速機使用］を TRUE と設定した場合に有効となります。
*3. カウンタモードが［リニアモード］の場合にだけ設定が可能です。［ロータリモード］の場合は、リングカウンタ上限設定値およびリングカウンタ下限設定値から算出されます。
*4. ユニットバージョン Ver.1.10 以前の CPU ユニットは、ARRAY [0..31] OF BYTE です。

● _sAXIS_OPERATION_SETTINGS（動作設定）

_sAXIS_OPERATION_SETTINGS 型は、軸パラメータの動作設定の値を指定するための構造体型データ型です。
次に、各メンバについて説明します。

メンバ変数	パラメータ名称	データ型	有効範囲	内容
MaxVel	最高速度	LREAL	倍精度実数型の正数	各軸の最高速度を設定します。 使用するモータの最大回転数を超えないように設定してください。
StartVel	起動速度	LREAL	0.0～倍精度実数型の正の上限値	各軸の起動速度を設定します。 最高速度を超えないように設定してください。
MaxJogVel	ジョグ最高速度	LREAL	倍精度実数型の正数	各軸のジョグ最高速度を設定します。 最高速度を超えないように設定してください。
MaxAcc	最大加速度	LREAL	倍精度実数型の正数、0	軸動作指令時の各軸の最大加速度を設定します。 「0」を設定すると加速度の制限なしとなります。
MaxDec	最大減速度	LREAL	倍精度実数型の正数、0	軸動作指令時の各軸の最大減速度を設定します。 「0」を設定すると減速度の制限なしとなります。
AccDecOver	加減速オーバー	_eMC_ACCDECOVER	0: _mcAccDecOverBuffer 1: _mcAccDecOverRapid 2: _mcAccDecOverErrorStop	軸の加減速制御で、目標位置への停止を優先した結果、減速オーバーとなり最大加減速度を超過してしまう場合の動作を指定します。 0: 加減速度を急激にする（ブレンディングをバッファードに切り替える）*1 1: 加減速度を急激にする 2: 異常として停止する*2

メンバ変数	パラメータ名称	データ型	有効範囲	内容
ReverseMode	反転時動作	_eMC_REVERSE_MODE	0: _mcReverseModeDecelerationStop 1: _mcReverseModeImmediateStop	命令多重起動時、命令再起動時、割り込み定寸位置決め時の反転時動作を指定します。 0: 減速停止 1: 即停止
VelWarningVal	速度警告値	UINT	0、1～100	各軸の速度警告を出力するための最高速度に対する割合を設定します。 「0」を設定すると速度警告は出力しません。
AccWarningVal	加速度警告値	UINT	0、1～100	各軸の加速度警告を出力するための最高加速度に対する割合を設定します。 「0」を設定すると加速度警告は出力しません。
DecWarningVal	減速度警告値	UINT	0、1～100	各軸の減速度警告を出力するための最高減速度に対する割合を設定します。 「0」を設定すると減速度警告は出力しません。
PosiTrqWarningVal *3	正方向トルク警告値	UINT	0、1～1000	各軸の正方向トルク警告を出力するためのトルク指令値を設定します。 「0」を設定すると正方向トルク警告は出力しません。
NegaTrqWarningVal *3	負方向トルク警告値	UINT	0、1～1000	各軸の負方向トルク警告を出力するためのトルク指令値を設定します。 「0」を設定すると負方向トルク警告は出力しません。
InPosRange	インポジション幅	LREAL	倍精度実数型の正数、0	位置決め完了幅を設定します。
InPosCheckTime	インポジションチェック時間	UINT	0、1～10000	位置決め完了のチェック時間をミリ秒単位で設定します。 「0」を設定すると、原点復帰時の原点確定時にだけ位置決め完了のチェックを継続します。原点復帰時以外では位置決め完了のチェックはしません。
ActVelFilterTimeConstant	フィードバック速度フィルタ時定数	UINT	0、1～100	フィードバック速度の移動平均を算出する期間をミリ秒単位で設定します。 「0」を設定すると移動平均は算出しません。
ZeroPosRange	原点位置範囲	LREAL	倍精度実数型の正数、0	原点位置検出幅を設定します。
(Reserved)	(予備エリア)	ARRAY[0..31] OF BYTE	－	－

*1. ユニットバージョン Ver.1.10 以降の CPU ユニットでは、［ブレンディング］を［バッファード］に切り替えません。詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

*2. ユニットバージョン Ver.1.10 以降の CPU ユニットでは、ブレンディング動作の場合、異常として停止しません。詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

*3. トルク制御時にだけ有効です。

● _sAXIS_OTHER_OPERATION_SETTINGS（拡張動作設定）

_sAXIS_OTHER_OPERATION_SETTINGS 型は、軸パラメータの拡張動作設定の値を指定するための構造体型データ型です。
次に、各メンバについて説明します。

メンバ変数	パラメータ名称	データ型	有効範囲	内容
ImmediateStopInputStopMode	即停止入力停止方法	_eMC_STOP_MODE	1: _mcImmediateStop 2: _mcImmediateStopFEReset 3: _mcFreeRunStop	即停止入力が有効になったときの、MC機能モジュール側での停止方法を設定します。 1: 指令値を即停止 2: 指令値を即停止し、同時に偏差カウンタリセットを実行 3: 指令値を即停止し、同時にサーボOFF を実行
LimitInputStopMode	限界入力停止方法	_eMC_STOP_MODE	0: _mcDeccelerationStop 1: _mcImmediateStop 2: _mcImmediateStopFEReset 3: _mcFreeRunStop	正方向限界入力、または、負方向限界入力が有効になったときの、MC 機能モジュール側での停止方法を設定します。 0: 指令値を減速停止 1: 指令値を即停止 2: 指令値を即停止し、同時に偏差カウンタリセットを実行 3: 指令値を即停止し、同時にサーボOFF を実行
DriveErrorResetMonitoringTime	ドライバエラーリセット監視時間	UINT	1～1000	ドライバエラーリセット実行時の監視時間を設定します。（単位: ms） 監視時間を過ぎると、ドライバの異常がリセットされていなくてもリセット処理を終了します。
MaxPosiTrqLimit	正方向トルク制限上限値	LREAL	0.0～1000.0	正方向トルク制限値の上限を設定します。
MaxNegaTrqLimit	負方向トルク制限上限値	LREAL	0.0～1000.0	負方向トルク制限値の上限を設定します。
ImmediateStopInputLogicInversion	即停止入力論理反転	BOOL	TRUE, FALSE	即停止入力信号の論理の反転を設定します。 TRUE: 反転する FALSE: 反転しない
PosiLimitInputLogicInversion	正方向限界入力論理反転	BOOL	TRUE, FALSE	正方向限界入力信号の論理の反転を設定します。 TRUE: 反転する FALSE: 反転しない
NegaLimitInputLogicInversion	負方向限界入力論理反転	BOOL	TRUE, FALSE	負方向限界入力信号の論理の反転を設定します。 TRUE: 反転する FALSE: 反転しない
HomeProximityInputLogicInversion	原点近傍入力論理反転	BOOL	TRUE, FALSE	原点近傍入力信号の論理の反転を設定します。 TRUE: 反転する FALSE: 反転しない
(Reserved)	（予備エリア）	ARRAY[0..31] OF BYTE	－	－

● _sAXIS_LIMIT_SETTINGS（リミット設定）

_sAXIS_LIMIT_SETTINGS 型は、軸パラメータのリミット設定の値を指定するための構造体型データ型です。
次に、各メンバについて説明します。

メンバ変数	パラメータ名称	データ型	有効範囲	内容
SwLimitMode	ソフトウェアリミット機能	_eMC_SWLMT_MODE	0: _mcNonSwLmt 1: _mcCmdDecelerationStop 2: _mcCmdImmediateStop 3: _mcActDecelerationStop 4: _mcActImmediateStop	ソフトウェアリミットの機能を選択します。 0: 無効 1: 指令位置に対して有効で減速停止 2: 指令位置に対して有効で即停止 3: フィードバック位置に対して有効で減速停止 4: フィードバック位置に対して有効で即停止
PosiSwLimit	正方向ソフトウェアリミット	LREAL	倍精度実数型の負数、正数、0	正方向側のソフトウェアリミットを設定します。
NegaSwLimit	負方向ソフトウェアリミット	LREAL	倍精度実数型の負数、正数、0	負方向側のソフトウェアリミットを設定します。
FollowingErrorOverVal	位置偏差オーバー値	LREAL	倍精度実数型の正数、0	位置偏差オーバーのチェック値を設定します。 「0」を設定すると位置偏差オーバーチェックが無効になります。
FollowingErrorWarningVal	位置偏差警告値	LREAL	倍精度実数型の正数、0	位置偏差警告のチェック値を設定します。 「0」を設定すると位置偏差警告チェックが無効になります。
(Reserved)	(予備エリア)	ARRAY[0..31] OF BYTE	−	−

● _sAXIS_POSITION_COUNT_SETTINGS（位置カウント設定）

_sAXIS_POSITION_COUNT_SETTINGS 型は、軸パラメータの位置カウント設定の値を指定するための構造体型データ型です。

次に、各メンバについて説明します。

メンバ変数	パラメータ名称	データ型	有効範囲	内容
CountMode	カウントモード	_eMC_COUNT_MODE	0: _mcCountModeLinear 1: _mcCountModeRotary	位置のカウントモードを設定します。 0: リニアモード（有限長） 1: ロータリモード（無限長）
ModuloMaxPosVal	リングカウンタ上限設定値	LREAL	倍精度実数型の負数、正数、0	カウントモードを［ロータリモード］とした場合の、リングカウンタ上限値を設定します。
ModuloMinPosVal	リングカウンタ下限設定値	LREAL	倍精度実数型の負数、正数、0	カウントモードを［ロータリモード］とした場合の、リングカウンタ下限値を設定します。
(Reserved)	(予備エリア)	ARRAY[0..31] OF BYTE	−	−

● _sAXIS_HOMING_SETTINGS （原点復帰設定）

_sAXIS_HOMING_SETTINGS 型は、軸パラメータの原点復帰設定の値を指定するための構造体型データ型です。

次に、各メンバについて説明します。

メンバ変数	パラメータ名称	データ型	有効範囲	内容
Mode	原点復帰動作	_eMC_HOMING_MODE	0: _mcHomeSwTurnHomeSwOff 1: _mcHomeSwTurnHomeSwOn 4: _mcHomeSwOff 5: _mcHomeSwOn 8: _mcLimitInputOff 9: _mcHomeSwTurnHomeMask 11: _mcLimitInputOnly 12: _mcHomeSwTurnHoldingTime 13: _mcNoHomeSwHoldingHomeInput 14: _mcHomePreset	原点復帰の動作方法を設定します。 0: 近傍退避･原点近傍入力 OFF 指定 1: 近傍退避･原点近傍入力 ON 指定 4: 原点近傍入力 OFF 指定 5: 原点近傍入力 ON 指定 8: 限界入力 OFF 指定 9: 近傍退避･原点入力マスク距離指定 11: 限界入力のみ 12: 近傍退避･押し当て時間指定 13: 原点近傍入力なし･押し当て原点入力指定 14: 原点プリセット
HomeInputSignal	原点入力信号	_eMC_HOME_INPUT	0: _mcZPhase 1: _mcExternalSignal	原点入力信号を選択します。 0: Z 相入力を使用する 1: 外部原点入力を使用する
StartDirection	原点復帰開始方向	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	原点復帰起動時の開始方向を設定します。 0: 正方向 2: 負方向
HomeInputDetectionDirection	原点検出方向	_eMC_DIRECTION	0: _mcPositiveDirection 2: _mcNegativeDirection	原点復帰の原点入力検出方向を設定します。 0: 正方向 2: 負方向
PosiLimitInputMode	正方向限界入力時動作	_eMC_LIMIT_REVERSE_MODE	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDecelerationStop	原点復帰中に正方向限界入力が「ON」した場合の停止方法を設定します。 0: 反転しない・軸の異常で停止（停止方法は限界入力停止方法に従う） 1: 反転する・即停止 2: 反転する・減速停止
NegaLimitInputMode	負方向限界入力時動作	_eMC_LIMIT_REVERSE_MODE	0: _mcErrorStop 1: _mcRevImmediateStop 2: _mcRevDecelerationStop	原点復帰中に負方向限界入力が「ON」した場合の停止方法を設定します。 0: 反転しない・軸の異常で停止（停止方法は限界入力停止方法に従う） 1: 反転する・即停止 2: 反転する・減速停止
Vel	原点復帰速度	LREAL	倍精度実数型の正数	原点復帰中の速度を設定します。
ApproachVel	原点復帰アプローチ速度	LREAL	倍精度実数型の正数	原点近傍入力が「ON」した後の速度を設定します。
Acc	原点復帰加速度	LREAL	倍精度実数型の正数、0	原点復帰時の加速度を指定します。 「0」を設定すると加速せずに原点復帰速度などの目標とする速度に到達します。
Dec	原点復帰減速度	LREAL	倍精度実数型の正数、0	原点復帰時の減速度を指定します。 「0」を設定すると減速せずに原点復帰アプローチ速度などの目標とする速度に到達します。
Jerk	原点復帰ジャーク	LREAL	倍精度実数型の正数、0	原点復帰時のジャークを指定します。 「0」を設定するとジャークなしとなります。

メンバ変数	パラメータ名称	データ型	有効範囲	内容
HomeInputMaskDistance	原点入力マスク距離	LREAL	倍精度実数型の正数、0	原点復帰動作モードが近傍退避・原点入力マスク距離指定の場合の原点入力のマスク距離を設定します。
HomeOffset	原点位置オフセット	LREAL	倍精度実数型の負数、正数、0	原点復帰完了時に設定した値で現在位置をプリセットします。
HoldingTime	原点復帰押し当て時間	UINT	0～10000	原点復帰動作モードが近傍退避・押し当て時間指定の場合の押し当て時間をミリ秒単位で設定します。
CompensationVal	原点復帰補正値	LREAL	倍精度実数型の負数、正数、0	原点復帰の原点確定後の原点復帰補正量を設定します。
CompensationVel	原点復帰補正速度	LREAL	倍精度実数型の正数	原点復帰補正時の速度を設定します。
(Reserved)	(予備エリア)	ARRAY[0..31] OF BYTE	－	－

書き込みや読み出し対象となる軸パラメータ

本命令での書き込み対象となる軸パラメータを以下に記載します。
MC_ReadAxisParameter（軸パラメータ読出）命令での読み出し対象となる軸パラメータも以下と同じです。

軸パラメータ区分	軸パラメータ名称	○: 書込／読出対象 ×: 書込／読出対象外
軸基本設定	軸番号	×
	モーション制御 *1	×
	軸使用	×
	軸種別	×
	制御機能 *2	×
	入力デバイス／出力デバイス	×
単位変換設定	表示単位	○
	モータ 1 回転のパルス数	○
	モータ 1 回転の移動量	○
	減速機使用 *3	○
	ワーク側 1 回転の移動量 *3	○ *4
	ワーク側ギア比 *3	○
	モータ側ギア比 *3	○
動作設定	最高速度	○ *5
	起動速度	○ *5
	ジョグ最高速度	○ *5
	最大加速度	○ *5
	最大減速度	○ *5
	加減速オーバー	○ *5
	反転時動作	○ *5
	速度警告値	○ *5
	加速度警告値	○ *5
	減速度警告値	○ *5
	正方向トルク警告値 *6	○ *5
	負方向トルク警告値 *6	○ *5

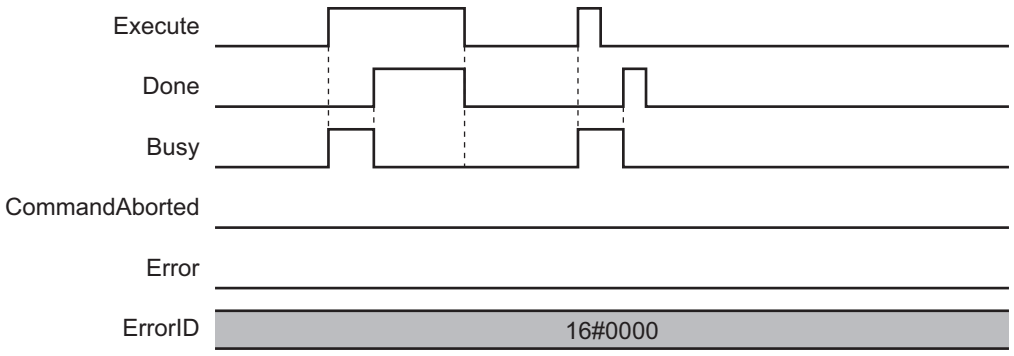
軸パラメータ区分	軸パラメータ名称	○: 書込／読出対象 ×: 書込／読出対象外
	インポジション幅	○ *5
	インポジションチェック時間	○ *5
	フィードバック速度フィルタ時定数	○
	原点位置範囲	○ *5
拡張動作設定	即停止入力停止方法	○ *5
	限界入力停止方法	○ *5
	ドライバエラーリセット監視時間	○ *5
	正方向トルク制限上限値	○ *5
	負方向トルク制限下限値	○ *5
	即停止入力論理反転	○ *5
	正方向限界入力論理反転	○ *5
	負方向限界入力論理反転	○ *5
	原点近傍入力論理反転	○ *5
リミット設定	ソフトウェアリミット機能	○ *5
	正方向ソフトウェアリミット	○ *5
	負方向ソフトウェアリミット	○ *5
	位置偏差オーバー値	○ *5
	位置偏差警告値	○ *5
位置カウント設定	カウントモード	○
	リングカウンタ上限設定値	○
	リングカウンタ下限設定値	○
	エンコーダ種別	×
サーボドライバ設定	リングカウンタ上限設定値	×
	リングカウンタ下限設定値	×
	PDS 状態制御方式 *7	×
原点復帰設定	原点復帰動作	○ *5
	原点入力信号	○ *5
	原点復帰開始方向	○ *5
	原点検出方向	○ *5
	正方向限界入力時動作	○ *5
	負方向限界入力時動作	○ *5
	原点復帰速度	○ *5
	原点復帰アプローチ速度	○ *5
	原点復帰加速度	○ *5
	原点復帰減速度	○ *5
	原点復帰ジャーク	○ *5
	原点入力マスク距離	○ *5
	原点位置オフセット	○ *5
	原点復帰押し当て時間	○ *5
	原点復帰補正値	○ *5
	原点復帰補正速度	○ *5

*1. 形 NX701 CPU ユニットだけのパラメータです。
*2. 形 NX102 および形 NX1P2 CPU ユニットだけのパラメータです。
*3. ユニットバージョン Ver.1.11 以降の CPU ユニットと Sysmac Studio Ver.1.15 以降の組み合わせで使用可能です。

*4. 書き込み対象となる軸の［減速機使用］が TRUE かつ［カウントモード］が［ロータリモード］の場合、［ワーク側 1 回転の移動量］の値は書き込みを行いません。｜リングカウンタ上限値-リングカウンタ下限値｜の値が［ワーク側 1 回転の移動量］として扱われます。また、読み出し対象となる軸の［減速機使用］が TRUE かつ［カウントモード］が［ロータリモード］の場合、［ワーク側 1 回転の移動量］の値は｜リングカウンタ上限値-リングカウンタ下限値｜の値が読み出されます。
*5. 書き込み対象となる軸の［軸種別］が［エンコーダ軸］、または［仮想エンコーダ軸］である場合、軸パラメータへの書き込みを行いません。また、読み出し対象となる軸の［軸種別］が［エンコーダ軸］、または［仮想エンコーダ軸］である場合、軸パラメータの初期値が読み出されます。
軸パラメータの初期値は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編 (SBCE-433)』を参照してください。
*6. トルク制御時にだけ有効です。
*7. ユニットバージョン Ver.1.10 以降の CPU ユニットと Sysmac Studio Ver.1.12 以降の組み合わせで指定可能です。

タイミングチャート

MC_WriteAxisParameter（軸パラメータ書込）命令を実行した場合のタイミングチャートを以下に示します。



モーション命令再起動

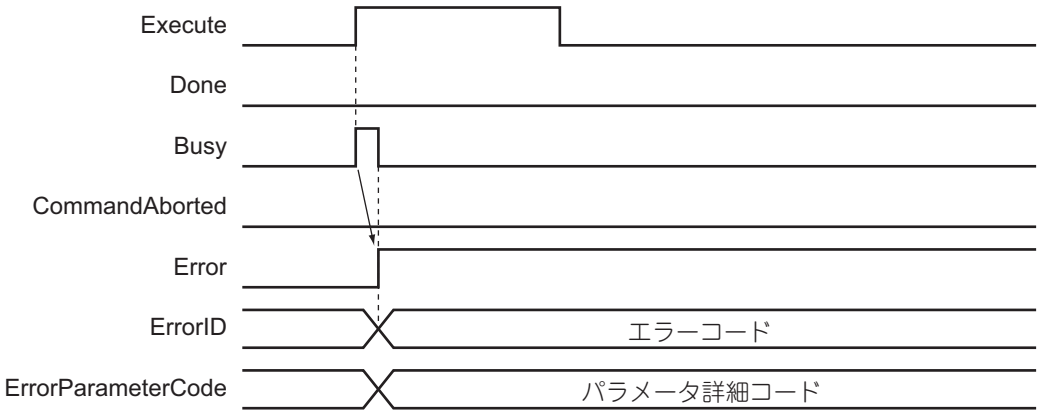
モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となり、パラメータの書き込みを行いません。命令実行前の値が保持されます。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。また、「ErrorID」（エラーコード）によっては、付属情報が「ErrorParameterCode」（パラメータ詳細コード）に出力されます。



● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

MC_ReadAxisParameter

MC 機能モジュールの軸パラメータ設定を読み出します。

命令	名称	FB/FUN	グラフィック表現	ST 表現
MC_ReadAxisParameter	軸パラメータ読出	FB	MC_ReadAxisParameter_instance MC_ReadAxisParameter Axis — Axis AxisParameter — AxisParameter Execute — Done Busy CommandAborted Error ErrorID	MC_ReadAxisParameter (Axis :=「パラメータ」, AxisParameter :=「パラメータ」, Execute :=「パラメータ」, Done =>「パラメータ」, Busy =>「パラメータ」, CommandAborted =>「パラメータ」, Error =>「パラメータ」, ErrorID =>「パラメータ」);

バージョン関連情報

本命令は、ユニットバージョン Ver.1.08 以降の CPU ユニットと Sysmac Studio Ver.1.09 以降の組み合わせで使用可能です。

変数

入力変数

入力変数	名称	データ型	有効範囲	初期値	内容
Execute	起動	BOOL	TRUE, FALSE	FALSE	立ち上がり時に命令を開始します。

出力変数

出力変数	名称	データ型	有効範囲	内容
Done	完了	BOOL	TRUE, FALSE	命令の実行が完了したときに TRUE となります。
Busy	実行中	BOOL	TRUE, FALSE	命令を受け付けたときに TRUE となります。
CommandAborted	実行中断	BOOL	TRUE, FALSE	命令が中止されたときに TRUE となります。
Error	エラー	BOOL	TRUE, FALSE	異常が発生したときに TRUE となります。
ErrorID	エラーコード	WORD	*1	異常が発生したときに、エラーコードを出力します。 16#0000 は正常です。

*1. イベントコードの上 4 けたが ErrorID（エラーコード）となります。詳細は、『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』を参照してください。

● 出力変数の反映タイミング

変数	TRUE となるタイミング	FALSE となるタイミング
Done	命令が完了したとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Busy	「Execute」の立ち上がり	• 「Done」が TRUE となったとき。 • 「Error」が TRUE となったとき。 • 「CommandAborted」が TRUE となったとき。
CommandAborted	異常発生により、本命令が中止されたとき。	• 「Execute」が TRUE の場合、「Execute」の FALSE と同時 • 「Execute」が FALSE の場合、1 周期後
Error	本命令の起動条件や入力パラメータに異常要因が含まれていたとき。	異常を解除したとき。

入出力変数

入出力変数	名称	データ型	有効範囲	内容
Axis	軸	_sAXIS_REF	−	読み出し対象となる軸を指定します。*1
AxisParameter	軸パラメータ	_sAXIS_PARAM	−	読み出した値を格納します。*2

1. Sysmac Studio の軸基本設定画面で作成したユーザ定義変数の軸変数名（デフォルト: MC_Axis）またはシステム定義変数の軸変数名(_MC_AX[*], _MC1_AX[*], _MC2_AX[*])を使用してください。
Sysmac Studio Ver.1.29 以降を使用する場合、ユーザ定義変数の AT（割付先）指定に、システム定義変数の軸変数名を設定することができます。設定することで、ユーザ定義変数を指定することが可能になります。
Sysmac Studio Ver.1.28 以前を使用する場合、変数テーブルで作成したユーザ定義変数は、指定しないでください。
*2. _sAXIS_PARAM 型のユーザ定義変数を定義してください。

● _sAXIS_PARAM

メンバ変数	パラメータ名称	データ型 *1	機能
UnitConversion	単位変換設定	_sAXIS_UNIT_CONVERSION_SETTINGS	読み出した単位変換設定の値を各メンバ変数に格納します。
Operation	動作設定	_sAXIS_OPERATION_SETTINGS	読み出した動作設定の値を各メンバ変数に格納します。
OtherOperation	拡張動作設定	_sAXIS_OTHER_OPERATION_SETTINGS	読み出した拡張動作設定の値を各メンバ変数に格納します。
Limit	リミット設定	_sAXIS_LIMIT_SETTINGS	読み出したリミット設定の値を各メンバ変数に格納します。
PosCount	位置カウント設定	_sAXIS_POSITION_COUNT_SETTINGS	読み出した位置カウント設定の値を各メンバ変数に格納します。
Homing	原点復帰設定	_sAXIS_HOMING_SETTINGS	読み出した原点復帰設定の値を各メンバ変数に格納します。
(Reserved)	（予備エリア）	ARRAY[0..255] OF BYTE	−

*1. データ型の詳細については、「_sAXIS_PARAM (P.5-53)」を参照してください。

機能説明

- 「Execute」（起動）の立ち上りで、指定した「Axis」（軸）の軸パラメータを「AxisParameter」（軸パラメータ）に読み出します。
- 本命令は、軸変数の「Cfg.AxEnable」（軸使用）の状態に関係なく軸パラメータの読み出しが可能です。

使用上の注意

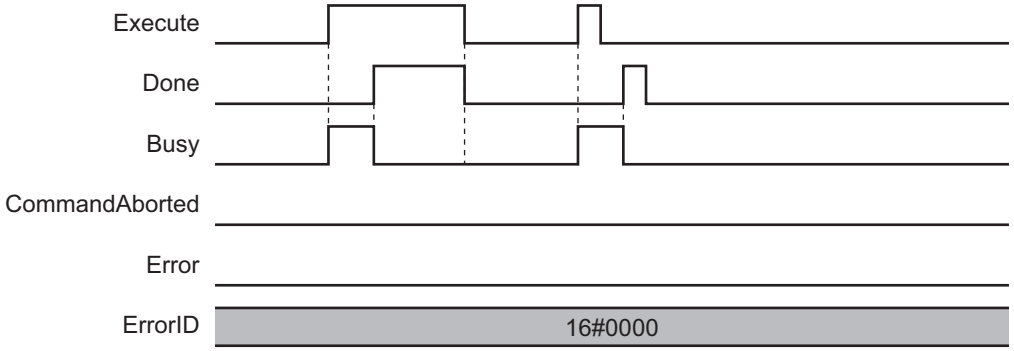
- 本命令で読み出せる値は、CPU ユニットの不揮発性メモリに保存されている値ではなく、本命令が実行されたタイミングで有効となっている軸パラメータが読み出されます。
  例えば、MC_Write（MC 設定書込）命令を使用して軸パラメータを書き込んだ場合、書き込んだ軸パラメータが読み出されます。
- 本命令を MC_WriteAxisParameter（軸パラメータ書込）命令と合わせて使用する場合、MC_WriteAxisParameter（軸パラメータ書込）命令の「Done」（完了）を確認してから起動してください。
- 本命令が完了したタイミングにだけ「AxisParameter」（軸パラメータ）に読み出します。その後は、「AxisParameter」（軸パラメータ）に読み出しません。そのため、「AxisParameter」（軸パラメータ）に書き込みをした場合は上書きされます。

読み出し対象となる軸パラメータ

本命令での読み出し対象となる軸パラメータは、「書き込みや読み出し対象となる軸パラメータ (P.5-59)」を参照してください。

タイミングチャート

MC_ReadAxisParameter（軸パラメータ読出）命令を実行した場合のタイミングチャートを以下に示します。



モーション命令再起動

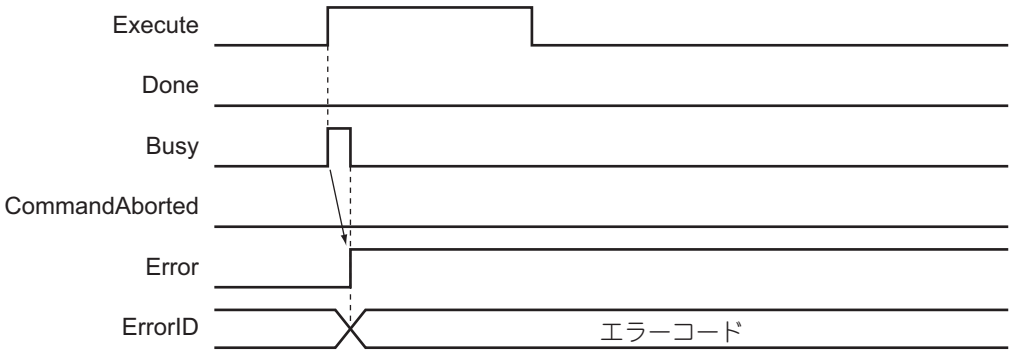
モーション命令再起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

モーション命令多重起動

モーション命令多重起動の詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

異常

本命令起動時に異常が発生した場合、「Error」（エラー）が TRUE となります。
「ErrorID」（エラーコード）に出力された値を参照することにより、異常の原因を知ることができます。



● **エラーコード**

本命令で発生する異常については、『NJ/NX シリーズ トラブルシューティングマニュアル (SBCA-469)』を参照してください。

A

付録

多重起動の可否、バージョン関連情報について説明します。

A-1　多重起動の可否 A-2
A-1-1　軸の状態、軸グループの状態 A-2
A-1-2　状態遷移と多重起動の可否 A-4
A-2　バージョン関連情報 A-10
A-3　マニュアル No.変更前の改訂履歴 A-12

A-1　多重起動の可否

モーション命令の多重起動の可否は、現在の軸の状態または現在の軸グループの状態、および実行する命令の種類によって決まります。ここでは、それらの関係について説明します。
なお、命令の多重起動の詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

使用上の注意

- 1 つの単軸でバッファできる多重起動は 1 つだけです。2 つ以上の命令を多重起動しようとすると、命令で異常が発生します。
- 1 つの軸グループでバッファできる多重起動は 7 つまでです。8 つ以上の命令を多重起動しようとすると、命令で異常が発生します。
- 軸指令の命令によって動作している軸に対して、軸グループ指令の命令を多重起動することはできません。また、軸グループ指令の命令によって動作している軸に対して、軸指令の命令を多重起動することはできません。これらの多重起動を行おうとした場合、命令で異常が発生します。

A-1-1　軸の状態、軸グループの状態

多重起動の可否は、現在の軸の状態または現在の軸グループの状態で変わります。軸の状態、軸グループの状態は、該当する軸の軸変数および軸グループ変数の値で知ることができます。
なお、軸の状態、軸グループの状態、軸変数、軸グループ変数の詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

軸の状態を示す軸変数

軸の状態を示す軸変数には、以下の 9 種類があります。これらは排他的であり、常にこのうちのどれか 1 つだけが TRUE になります。
ここでは、_MC_AX[0-255]を例に説明しています。_MC1_AX[0-255]、_MC2_AX[0-255]についても同様です。

変数名	データ型	名称	機能
_MC_AX[0-255]	_sAXIS_REF	軸変数	
Status	_sAXIS_REF_STA	軸ステータス	
Disabled	BOOL	軸無効	軸がサーボ OFF で停止しているときに TRUE となります。
Standstill	BOOL	停止中	軸がサーボ ON で停止しているときに TRUE となります。
Descrete	BOOL	位置決め動作中	目標位置に向かって、位置決め制御を実行しているときに TRUE となります。 位置決め動作中にオーバーライド値を「0」に設定したことによる速度「0」の状態も含みます。
Continuous	BOOL	連続動作中	目標位置をもたずに、連続動作しているときに TRUE となります。 速度制御またはトルク制御のときに、この状態となります。 目標速度を「0」にしたことによる速度「0」の状態や、連続動作中にオーバーライド値を「0」に設定したことによる速度「0」の状態も含みます。
Synchronized	BOOL	同期動作中	同期制御を実行しているときに TRUE となります。 同期制御命令切替後の同期待ち状態も含みます。

変数名	データ型	名称	機能
Homing	BOOL	原点復帰中	MC_Home 命令、MC_HomeWithParameter 命令により、原点復帰を実行しているときに TRUE となります。
Stopping	BOOL	減速停止中	MC_Stop 命令、MC_TouchProbe 命令により、軸が停止するまでの状態のとき TRUE となります。 MC_Stop 命令で軸が停止した後に、「Execute」が TRUE のときも含みます。 本状態では軸動作命令を起動できません (CommandAborted=TRUE)。
ErrorStop	BOOL	エラー減速停止中	MC_ImmediateStop 命令または軸 軽度フォールト発生中 (_MC_AX[*].MFaultLvl.Active=TRUE) により、軸が停止するまで、および停止しているとき TRUE となります。 この状態では軸動作命令を起動できません (CommandAborted=TRUE)。
Coordinated	BOOL	多軸協調動作中	多軸協調命令により、軸グループ有効となっているとき TRUE となります。

軸グループの状態を示す軸グループ変数

軸グループの状態を示す軸グループ変数には、以下の 5 種類があります。これらは排他的であり、常にこのうちのどれか 1 つだけが TRUE になります。
ここでは、_MC_GRP[0-63]を例に説明しています。
_MC1_GRP[0-63] 、_MC2_GRP[0-63] についても同様です。

変数名	データ型	名称	機能
_MC_GRP[0-63]	_sGROUP_REF	軸グループ変数	
Status	_sGROUP_REF_STA	軸グループステータス	
Disabled	BOOL	軸グループ無効	軸グループ無効で停止しているときに TRUE となります。
Standby	BOOL	停止中	軸グループ動作命令が停止しているときに TRUE となります。 軸グループの構成軸のサーボ ON/OFF 状態によりません。
Moving	BOOL	動作中	目標位置に向かって、軸グループ動作命令を実行しているときに TRUE となります。 インポジション待ちの状態や、オーバーライドによる、速度「0」の状態も含まれます。
Stopping	BOOL	減速停止中	MC_GroupStop 命令により、軸グループが停止するまでの状態のとき TRUE となります。 MC_GroupStop 命令で軸グループが停止した後に、「Execute」が TRUE のときも含みます。 本状態では軸動作命令を起動できません (CommandAborted=TRUE)。
ErrorStop	BOOL	エラー減速停止中	MC_GroupImmediateStop 命令、または軸グループ 軽度フォールト発生 (_MC_GRP[*].MFaultLvl.Active=TRUE)により、軸グループが停止するまで、および停止しているとき TRUE となります。 この状態では軸グループ動作命令を起動できません (CommandAborted=TRUE)。

A-1-2　状態遷移と多重起動の可否

各軸の状態および各軸グループの状態における、各命令の多重起動の可否を示します。それと併せて、各軸の状態および各軸グループの状態の遷移先も示します。
なお、モーション制御機能モジュールの状態遷移と、状態遷移に関係するモーション制御命令の詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル モーション制御編(SBCE-433)』を参照してください。

サーボ軸、仮想サーボ軸に対する命令の多重起動

命令の対象がサーボ軸およびサーボ軸グループの場合の、多重起動の可否と状態遷移先を、以下の表に示します。
多重起動の可否は、表のセルの色で表しています。

白色　　：多重起動可能
グレー　：多重起動不可。異常が発生します。
黄色　　：多重起動不可。「CommandAborted」（実行中断）が TRUE になり、命令実行が無効になります。

また、表内の記号は、以下のように遷移先の状態を示します。

軸の状態

A　：軸無効(Disabled)
B　：停止中(Standstill)
C　：位置決め動作中(Discrete)
D　：連続動作中(Continuous)
E　：同期動作中(Synchronized)
F　：原点復帰中(Homing)
G　：減速動作中(Stopping)
H　：エラー減速停止中(ErrorStop)
I　：多軸協調動作中(Coordinated)

軸グループの状態

a　：軸グループ無効(Disabled)
b　：停止中(Standby)
c　：動作中(Moving)
d　：減速動作中(Stopping)
e　：エラー減速停止中(ErrorStop)

● 軸指令の命令

命令	命令実行前の軸の状態									命令実行前の軸グループの状態				
	A	B	C	D	E	F	G	H	I	a	b	c	d	e
	軸無効	停止中	位置決め動作中	連続動作中	同期動作中	原点復帰中	減速停止中	エラー減速停止中	多軸協調動作中	軸グループ無効	停止中	動作中	減速停止中	エラー減速停止中
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

命令	命令実行前の軸の状態									命令実行前の軸グループの状態				
	A	B	C	D	E	F	G	H	I	a	b	c	d	e
	軸無効	停止中	位置決め動作中	連続動作中	同期動作中	原点復帰中	減速停止中	エラー減速停止中	多軸協調動作中	軸グループ無効	停止中	動作中	減速停止中	エラー減速停止中
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_CamIn *6	H	H	H	H	—	H	G	H	—	—	—	—	—	—
MC_CamOut	H	H	H	H	C*7	H	G	H	I	a	e	e	e	e
MC_CamMonitor	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_GearIn	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_GearIn *6	H	H	H	H	—	H	G	H	—	—	—	—	—	—
MC_GearInPos	H	E	H	H	H	H	G	H	I	a	e	e	e	e
MC_GearInPos *6	H	H	H	H	—	H	G	H	—	—	—	—	—	—
MC_GearOut	H	H	H	H	C*8	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_MoveLink *6	H	H	H	H	—	H	G	H	—	—	—	—	—	—
MC_CombineAxes	H	E	E	E	E	H	G	H	I	a	e	e	e	e
MC_CombineAxes *6	H	H	H	H	—	H	G	H	—	—	—	—	—	—
MC_Phasing	H	H	H	H	E*9	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_TorqueControl *6	H	H	H	H	—	H	G	H	—	—	—	—	—	—
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*10	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_SyncMoveVelocity *6	H	H	H	H	—	H	G	H	—	—	—	—	—	—
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	*11	I	a	b	c	d	e

命令	命令実行前の軸の状態									命令実行前の軸グループの状態				
	A 軸無効	B 停止中	C 位置決め動作中	D 連続動作中	E 同期動作中	F 原点復帰中	G 減速停止中	H エラー減速停止中	I 多軸協調動作中	a 軸グループ無効	b 停止中	c 動作中	d 減速停止中	e エラー減速停止中
MC_ChangeAxisUse（未使用軸→使用軸）	A	—	—	—	—	—	—	—	—	—	—	—	—	—
MC_ChangeAxisUse（使用軸→未使用軸）	—	*12	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
MC_PeriodicSyncVariables	A	B	C	D	E	F	G	H	I	a	b	c	d	e
MC_SyncOffsetPosition	H	H	H	H	E	H	G	H	I	a	e	e	e	e
MC_OffsetPosition	H	H	H	H	E	H	G	H	I	a	e	e	e	e

*1. MC_Stop 命令の「Execute」が FALSE なら B: 停止中に遷移します。TRUE なら G: 減速停止中に遷移します。
*2. 最大減速度に応じて、b: 停止中または d: 減速停止中に遷移します。
*3. 目標位置を設定している場合は C: 位置決め動作中に遷移します。そうでない場合は D: 連続動作中に遷移します。
*4. MC_MoveZeroPosition 命令が実行中の場合は異常となり、状態遷移先は H: エラー減速停止中になります。
*5. MC_TorqueControl 命令または MC_SyncMoveVelocity 命令が実行中の場合は異常となり、状態遷移先は H: エラー減速停止中になります。
*6. 軸機能制限設定が、[1: 単軸制御／動作プロファイルを指定する制御（指令位置出力のみ）実行可能] の場合。
*7. MC_CamIn 命令が実行中でない場合は異常となり、状態遷移先は H: エラー減速停止中になります。
*8. MC_GearIn 命令または MC_GearInPos 命令が実行中でない場合は異常となり、状態遷移先は H: エラー減速停止中になります。
*9. MC_CombineAxes 命令が実行中の場合は異常となり、状態遷移先は H: エラー減速停止中になります。
*10. MC_AbortTrigger 命令の「LatchID」が、実行中の MC_MoveFeed 命令の「LatchID」と重複している場合は異常となり、状態遷移先は H: エラー減速停止中になります。
*11. 条件によって、状態遷移先は以下のように変わります。
- サーボ OFF で異常解除成功の場合、またはサーボ OFF で MC_Stop 命令の「Execute」が TRUE の場合、A: 軸無効に遷移します。
- サーボ ON で異常解除成功の場合、B: 停止中に遷移します。
- サーボ ON で MC_Stop 命令の「Execute」が TRUE の場合、G: 減速停止中に遷移します。
- サーボ ON/OFF に関係なく異常解除失敗の場合、H: エラー減速停止中に遷移します。

*12. 軸変数の_MC_AX[*].Details.VelLimit（指令速度飽和）が TRUE の場合、H: エラー減速停止中に遷移します。FALSE の場合、未使用軸に切り替わるため、ー: 軸の状態なしとなります。

使用上の注意

NX シリーズ CPU ユニットの場合、_MC_AX[*]で始まる変数名は、_MC1_AX[*]、_MC2_AX[*]の場合があります。

● 軸グループ指令の命令

命令	命令実行前の軸の状態									命令実行前の軸グループの状態				
	A 軸無効	B 停止中	C 位置決め動作中	D 連続動作中	E 同期動作中	F 原点復帰中	G 減速停止中	H エラー減速停止中	I 多軸協調動作中	a 軸グループ無効	b 停止中	c 動作中	d 減速停止中	e エラー減速停止中
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. サーボ OFF の場合、「CommandAborted」（実行中断）が TRUE になります。
*2. 条件によって、状態遷移先は以下のように変わります。
- サーボ OFF で異常が発生していない場合、A: 軸無効に遷移します。
- サーボ OFF で MC_GroupStop 命令の「Execute」が FALSE の場合、B: 停止中に遷移します。
- サーボ ON で MC_GroupStop 命令の「Execute」が TRUE の場合、G: 減速停止中に遷移します。
- サーボ ON/OFF に関係なく異常発生の場合、H: エラー減速停止中に遷移します。

*3. MC_GroupEnable 命令が実行中の場合は異常となり、状態遷移先は e: エラー減速停止中になります。
単軸位置制御軸の場合は、軸グループ構成軸として設定できないので、異常となります。
*4. サーボ ON の場合、異常になります。
*5. 条件によって、状態遷移先は以下のように変わります。
- サーボ OFF で異常解除成功の場合、a: 軸グループ無効に遷移します。
- サーボ ON で異常解除成功の場合、b: 停止中に遷移します。
- サーボ ON で MC_GroupStop 命令の「Execute」が TRUE の場合、d: 減速停止中に遷移します。

● 共通指令の命令

共通指令の命令は、軸の状態、軸グループの状態にかかわらず、常に多重起動可能です。
また、共通指令の命令を実行することにより、軸の状態、軸グループの状態が他の状態に遷移することはありません。現在の状態が維持されます。

エンコーダ軸、仮想エンコーダ軸に対する命令の多重起動

命令の対象がエンコーダ軸およびエンコーダ軸グループの場合の、状態遷移と多重起動の可否を、以下の表に示します。

多重起動の可否は、表のセルの色で表しています。

白色　：多重起動可能
グレー　：多重起動不可。異常が発生します。
黄色　：多重起動不可。「CommandAborted」（実行中断）が TRUE になり、命令実行が無効になります。

また、表内の記号は、以下のように遷移先の状態を示します。

軸の状態

A ：軸無効(Disabled)
H ：エラー減速停止中(ErrorStop)

● 軸指令の命令

命令	命令実行前の軸の状態	
	A	H
	軸無効	エラー減速停止中
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_CamMonitor	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 (未使用軸→使用軸)	A	H
MC_ChangeAxisUse (使用軸→未使用軸)	−	H
MC_DigitalCamSwitch	A	H
MC_TimeStampToPos	A	H
MC_PeriodicSyncVariables	A	H
MC_SyncOffsetPosition	H	H
MC_OffsetPosition	H	H

● 軸グループ指令の命令

エンコーダ軸、仮想エンコーダ軸は、軸グループ構成軸として設定することができません。したがって、エンコーダ軸、仮想エンコーダ軸に対して軸グループ指令の命令を多重起動しようとすると、すべて異常になります。

● 共通指令の命令

共通指令の命令は、軸の状態にかかわらず、常に多重起動可能です。
また、共通指令の命令を実行することにより、軸の状態が他の状態に遷移することはありません。現在の状態が維持されます。

A-2 バージョン関連情報

ここでは、NJ/NX シリーズ CPU ユニットと Sysmac Studio の各バージョンの更新に関連する情報を記載します。

バージョンの更新により、仕様変更された命令と新規命令の一覧

命令には、CPU ユニット、Sysmac Studio の各バージョンによって、対応状況や仕様が異なるものがあります。以下に、その内容を示します。

種類	命令	名称	新規／仕様変更	バージョン		ページ
				CPU ユニット	Sysmac Studio	
共通指令	MC_SetCamTableEndPointIndex	カムテーブル終点インデックス更新	新規	Ver.1.36/ Ver.1.69	Ver.1.63	P.5-8
	MC_Write	MC 設定書込	仕様変更	Ver.1.01	Ver.1.02	P.5-18
			仕様変更	Ver.1.10	Ver.1.12	
	MC_GenerateCamTable	カムテーブル生成	新規	Ver.1.08	Ver.1.09	P.5-23
	MC_WriteAxisParameter	軸パラメータ書込	新規	Ver.1.08	Ver.1.09	P.5-50
			仕様変更	Ver.1.10	Ver.1.12	
	MC_ReadAxisParameter	軸パラメータ読出	新規	Ver.1.08	Ver.1.09	P.5-63
			仕様変更	Ver.1.10	Ver.1.12	
軸グループ指令	MC_GroupReadPosition	軸グループ位置取得	新規	Ver.1.01	Ver.1.02	P.4-87
	MC_ChangeAxesInGroup	軸グループ構成軸書込	新規	Ver.1.01	Ver.1.02	P.4-91
	MC_GroupSyncMoveAbsolute	軸グループサイクリック同期絶対位置制御	新規	Ver.1.01	Ver.1.02	P.4-95
軸指令	MC_Power	運転可	仕様変更	Ver.1.10	Ver.1.12	P.3-3
	MC_SetPosition	現在位置変更	仕様変更	Ver.1.10	Ver.1.12	P.3-142
	MC_ResetFollowingError	偏差カウンタリセット	仕様変更	Ver.1.10	Ver.1.12	P.3-153
	MC_CamIn	カム動作開始	仕様変更	Ver.1.10	Ver.1.12	P.3-160
	MC_CamMonitor	カムモニタ	新規	Ver.1.21/ Ver.1.32	Ver.1.28	P.3-221
	MC_CombineAxes	加減算位置決め	仕様変更	Ver.1.10	Ver.1.12	P.3-297
	MC_GearIn	ギア動作開始	仕様変更	Ver.1.02	Ver.1.03	P.3-230
			仕様変更	Ver.1.10	Ver.1.12	
	MC_GearInPos	位置指定ギア動作	仕様変更	Ver.1.02	Ver.1.03	P.3-249
			仕様変更	Ver.1.10	Ver.1.12	
	MC_MoveLink	台形パターンカム	仕様変更	Ver.1.10	Ver.1.12	P.3-275
	MC_TouchProbe	外部ラッチ有効	仕様変更	Ver.1.10	Ver.1.12	P.3-339
	MC_HomeWithParameter	パラメータ指定原点復帰	新規	Ver.1.03	Ver.1.04	P.3-37
	MC_SyncMoveAbsolute	サイクリック同期絶対位置決め	新規	Ver.1.03	Ver.1.04	P.3-379
	MC_ChangeAxisUse	軸使用変更	新規	Ver.1.04	Ver.1.05	P.3-389
	MC_DigitalCamSwitch	デジタルカムスイッチ有効	新規	Ver.1.06	Ver.1.07	P.3-395
			仕様変更	Ver.1.09	Ver.1.10	
	MC_TimeStampToPos	タイムスタンプ→軸位置算出	新規	Ver.1.06	Ver.1.07	P.3-414
	MC_PeriodicSyncVariables	軸変数タスク間定周期同期	新規	Ver.1.10	Ver.1.13	P.3-425

種類	命令	名称	新規／仕様変更	バージョン		ページ
				CPU ユニット	Sysmac Studio	
	MC_SyncOffsetPosition	サイクリック同期位置オフセット補正	新規	Ver.1.10	Ver.1.12	P.3-433
	MC_OffsetPosition	位置オフセット補正	新規	Ver.1.21/ Ver.1.32	Ver.1.28	P.3-443

NX シリーズ 位置インタフェースユニットを使用する場合

NX シリーズ 位置インタフェースユニットは、ユニットバージョン Ver.1.05 以降の CPU ユニットと Sysmac Studio Ver.1.06 以降の組み合わせで使用可能です。
モーション制御命令を NX シリーズ 位置インタフェースユニットに対して使用する場合、機能に制約のある命令や使用できない命令があります。詳細は、『NX シリーズ 位置インタフェースユニット ユーザーズマニュアル (SBCE-374)』を参照してください。

A-3　マニュアル No.変更前の改訂履歴

本マニュアルは、Man.No.（マニュアル No.）を次のように変更しています。

現在の Man.No.	変更前の Man.No.
SBCE-434	SBCE-364

変更前の Man.No.での改訂履歴を示します。

改訂記号	改訂年月	改訂理由・改訂ページ
A	2011 年 7 月	初版
B	2012 年 1 月	• 以下の軸グループ指令命令の追加 　• MC_GroupReadPosition（軸グループ位置取得） 　• MC_ChangeAxesInGroup（軸グループ構成軸書込） 　• MC_GroupSyncMoveAbsolute（軸グループサイクリック同期絶対位置制御） • 誤記訂正
C	2012 年 5 月	• CPU ユニット Ver.1.02 へのバージョンアップに伴う改訂 • 誤記訂正
D	2012 年 8 月	CPU ユニット Ver.1.03 へのバージョンアップに伴う改訂
E	2013 年 2 月	CPU ユニット Ver.1.04 へのバージョンアップに伴う改訂
F	2013 年 4 月	• CPU ユニット Ver.1.05 へのバージョンアップに伴う改訂 • 誤記訂正
G	2013 年 6 月	• CPU ユニット Ver.1.06 へのバージョンアップに伴う改訂 • 誤記訂正
H	2013 年 12 月	• CPU ユニット Ver.1.08 へのバージョンアップに伴う改訂 • 誤記訂正
J	2014 年 7 月	• CPU ユニット Ver.1.09 へのバージョンアップに伴う改訂 • 誤記訂正
K	2015 年 1 月	• CPU ユニット Ver.1.10 へのバージョンアップに伴う改訂 • 誤記訂正
L	2015 年 4 月	• NX シリーズ CPU ユニット 形 NX701-□□□□、NJ シリーズ CPU ユニット 形 NJ101-□□□□の追加に伴う改訂 • 誤記訂正
M	2016 年 4 月	• CPU ユニット Ver.1.11 へのバージョンアップに伴う改訂 • 誤記訂正
N	2016 年 7 月	• CPU ユニット Ver.1.12 へのバージョンアップに伴う改訂 • 誤記訂正
P	2016 年 10 月	• NX シリーズ 形 NX1P2 CPU ユニットの追加に伴う改訂 • CPU ユニット Ver.1.13 へのバージョンアップに伴う改訂 • 誤記訂正
R	2017 年 4 月	誤記訂正による改訂
S	2017 年 10 月	誤記訂正による改訂
T	2018 年 4 月	• NX シリーズ 形 NX102 CPU ユニットの追加に伴う改訂 • イベントコードの『NJ/NX シリーズ トラブルシューティングマニュアル(SBCA-469)』への移動に伴う改訂 • 誤記訂正

I

索引

索引

Numerics

2 軸円弧補間 4-44

M

MC_AbortTrigger（外部ラッチ無効） 3-360
MC_AxesObserve（軸間偏差監視） 3-364
MC_CamIn（カム動作開始） 3-160
MC_CamMonitor（カムモニタ） 3-221
MC_CamOut（カム動作解除） 3-216
MC_ChangeAxesInGroup（軸グループ構成軸書込） 4-91
MC_ChangeAxisUse（軸使用変更） 3-389
MC_CombineAxes（加減算位置決め） 3-297
MC_DigitalCamSwitch（デジタルカムスイッチ） ... 3-395
MC_GearInPos（位置指定ギア動作） 3-249
MC_GearIn（ギア動作開始） 3-230
MC_GearOut（ギア動作解除） 3-270
MC_GenerateCamTable（カムテーブル生成） 5-23
MC_GroupDisable（軸グループ無効） 4-6
MC_GroupEnable（軸グループ有効） 4-2
MC_GroupImmediateStop（軸グループ即停止） 4-79
MC_GroupReadPosition（軸グループ位置取得） 4-87
MC_GroupReset（軸グループエラーリセット） 4-101
MC_GroupSetOverride（軸グループオーバーライド値設定） 4-83
MC_GroupStop（軸グループ強制停止） 4-71
MC_GroupSyncMoveAbsolute（軸グループサイクリック同期絶対位置制御） 4-95
MC_HomeWithParameter（パラメータ指定原点復帰） 3-37
MC_Home（原点復帰） 3-16
MC_ImmediateStop（即停止） 3-138
MC_MoveAbsolute（絶対値位置決め） 3-47
MC_MoveCircular2D（2 軸円弧補間） 4-44
MC_MoveFeed（割り込み定寸位置決め） 3-102
MC_MoveJog（ジョグ送り） 3-8
MC_MoveLinearAbsolute（絶対値直線補間） 4-38
MC_MoveLinearRelative（相対値直線補間） 4-41
MC_MoveLinear（直線補間） 4-11
MC_MoveLink（台形パターンカム） 3-275
MC_MoveRelative（相対値位置決め） 3-73
MC_MoveVelocity（速度制御） 3-81
MC_MoveZero（高速原点復帰） 3-95
MC_Move（位置決め） 3-42
MC_OffsetPosition（位置オフセット補正） 3-443
MC_PeriodicSyncVariables（軸変数タスク周定周期同期） 3-425
MC_Phasing（主軸相対値位相補正） 3-308
MC_Power（運転可） 3-3
MC_ReadAxisParameter（軸パラメータ読出） 5-63
MC_ResetFollowingError（偏差カウンタリセット） 3-153
MC_Reset（軸エラーリセット） 3-386
MC_SaveCamTable（カムテーブル保存） 5-13
MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新） 5-8
MC_SetCamTableProperty（カムテーブルプロパティ更新） 5-2
MC_SetOverride（オーバーライド値設定） 3-148
MC_SetPosition（現在位置変更） 3-142
MC_SetTorqueLimit（トルク制限） 3-326
MC_Stop（強制停止） 3-129
MC_SyncMoveAbsolute（サイクリック同期絶対位置決め） 3-379
MC_SyncMoveVelocity（サイクリック同期速度制御） 3-370
MC_SyncOffsetPosition（サイクリック同期位置オフセット補正） 3-433
MC_TimeStampToPos（タイムスタンプ→軸位置算出） 3-414
MC_TorqueControl（トルク制御） 3-315
MC_TouchProbe（外部ラッチ有効） 3-339
MC_WriteAxisParameter（軸パラメータ書込） 5-50
MC_Write（MC 設定書込） 5-18
MC_ZoneSwitch（ゾーン監視） 3-333
MC 設定書込 5-18

い

位相補正 3-308
位置オフセット補正 3-443
位置決め 3-42, 3-47, 3-73, 3-102
位置指定ギア動作 3-249

う

運転可 3-3

え

エラーリセット 3-386, 4-101
円弧補間 4-44

お

オーバーライド値設定 3-148, 4-83

か

外部ラッチ無効 3-360
外部ラッチ有効 3-339
加減算位置決め 3-297
カムテーブル終点インデックス更新 5-8
カムテーブル生成 5-23
カムテーブルプロパティ更新 5-2
カムテーブル保存 5-13
カム動作開始 3-160
カム動作解除 3-216
カムモニタ 3-221

き

ギア動作.....3-249
ギア動作開始.....3-230
ギア動作解除.....3-270
強制停止.....3-129, 4-71

け

現在位置変更.....3-142
原点復帰.....3-16

こ

高速原点復帰.....3-95

さ

サイクリック同期位置オフセット補正.....3-433
サイクリック同期絶対位置決め.....3-379
サイクリック同期速度制御.....3-370

し

軸エラーリセット.....3-386
軸間偏差監視.....3-364
軸グループ位置取得.....4-87
軸グループエラーリセット.....4-101
軸グループオーバーライド値設定.....4-83
軸グループ強制停止.....4-71
軸グループ構成軸書込.....4-91
軸グループサイクリック同期絶対位置制御.....4-95
軸グループ即停止.....4-79
軸グループ無効.....4-6
軸グループ有効.....4-2
軸使用変更.....3-389
軸パラメータ書込.....5-50
軸パラメータ読出.....5-63
軸変数タスク間定周期同期.....3-425
実行 ID.....2-10
主軸相対値位相補正.....3-308
ジョグ送り.....3-8

せ

絶対値位置決め.....3-47
絶対値直線補間.....4-38
設定書込.....5-18

そ

相対値位置決め.....3-73
相対値直線補間.....4-41
即停止.....3-138
速度制御.....3-81
ゾーン監視.....3-333

た

台形パターンカム.....3-275
タイムスタンプ→軸位置算出.....3-414

ち

直線補間.....4-11, 4-38, 4-41

て

デジタルカムスイッチ有効.....3-395
電子ギア.....3-249

と

トルク制御.....3-315
トルク制限.....3-326

は

パラメータ指定原点復帰.....3-37
バージョン.....22

へ

偏差カウンタリセット.....3-153

わ

割り込み定寸位置決め.....3-102

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー

製品に関するお問い合わせ先

お客様相談室

フリー通話 **0120-919-066**（クイック　オムロン）

携帯電話の場合、

055-982-5015（有料）をご利用ください。

受付時間：9:00～17:00（土・日・12/31～1/3を除く）

オムロンFAクイックチャット

www.fa.omron.co.jp/contact/tech/chat/

技術相談員にチャットでお問い合わせいただけます。（I-Webメンバーズ限定）

受付時間：平日9:00～12:00 / 13:00～17:00（土日祝日・年末年始・当社休業日を除く）

※受付時間、営業日は変更の可能性がございます。最新情報はリンク先をご確認ください。

その他のお問い合わせ：納期・価格・サンプル・仕様書は貴社のお取引先、または貴社担当オムロン販売員にご相談ください。オムロン制御機器販売店やオムロン販売拠点は、Webページでご案内しています。

オムロン制御機器の最新情報をご覧いただけます。緊急時のご購入にもご利用ください。 **www.fa.omron.co.jp**

●本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娯楽機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途に使用される際には、当社の意図した特別な商品用途の場合や特別の合意がある場合を除き、当社は当社商品に対して一切保証をいたしません。

●本製品の内、外国為替及び外国貿易法に定める輸出許可、承認対象貨物（又は技術）に該当するものを輸出（又は非居住者に提供）する場合は同法に基づく輸出許可、承認（又は役務取引許可）が必要です。

オムロン商品のご用命は

Man. No. SBCE-434L　2025年7月現在

©OMRON Corporation 2011-2025 All Rights Reserved.
お断りなく仕様などを変更することがありますのでご了承ください