OMRON

マシンオートメーションコントローラ

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

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□□

CPUユニット



SYSMAC
always in control

SBCE-433N

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。本マニュアルは、NJ/NX シリーズ CPU ユニットのモーション制御機能モジュールを使用する上で、必要な情報を記載しています。お使いになる前に本マニュアルをよく読んで、機能・性能などを十分ご理解いただき、システムの構築にご活用ください。
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。

対象となる商品

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)」を確認してください。

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 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 コマンドリファレンスマニュアル モーション制御編	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																			

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
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							○													

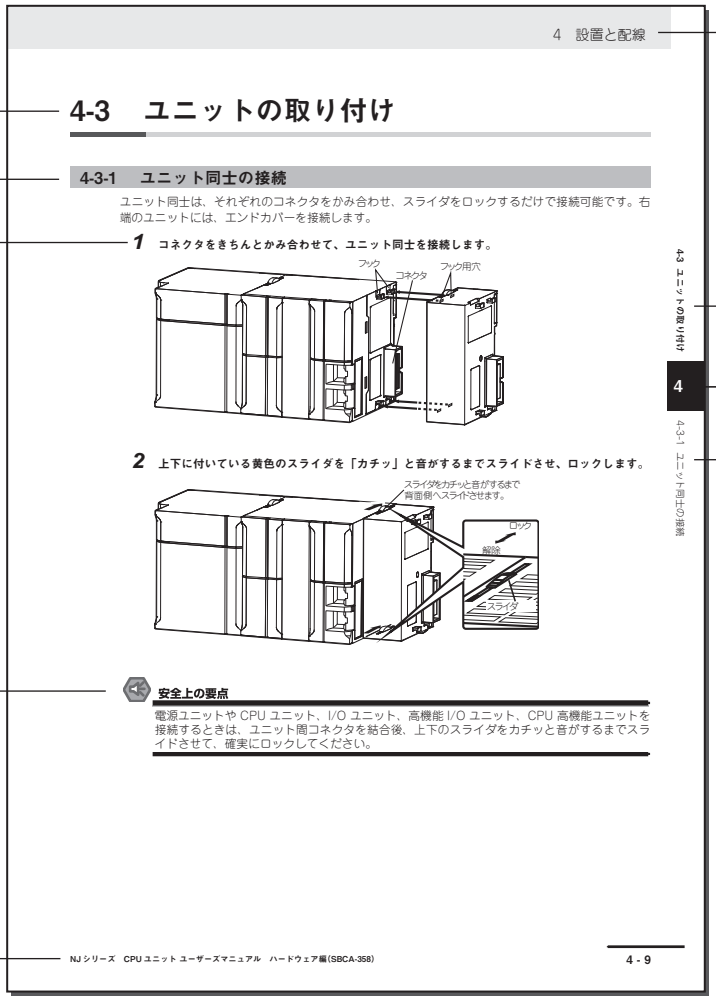
使用目的	マニュアル																				
	基本情報																				
	NX シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NJ シリーズ CPU ユニット ユーザーズマニュアル ハードウェア編	NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編	NX シリーズ 形 NX102 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX1P2 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 NX502 CPU ユニット ユーザーズマニュアル ハードウェア編	NX シリーズ 形 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 内蔵 EtherCAT ポート CPU ユニット ユーザーズマニュアル	NJ/NX シリーズ 内蔵 EtherNet/IP ポート CPU ユニット 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は次のとおりです。



節タイトル

項タイトル

操作手順

操作手順を示します。

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。

マニュアル名称

章タイトル

節タイトル

項タイトル

現在のページの章／節／項タイトルをそれぞれ示します。

章番号

現在のページの章番号を示します。

このページは、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 モーション制御機能モジュールの概要

2 モーション制御系の構成と原理

3 軸と軸グループの設定

4 Sysmac Studioによる配線確認

5 モーション制御パラメータ

6 モーション制御プログラム

7 手動運転

8 原点決め

9 モーション制御機能

10 サンプルプログラム

11 異常の確認と処置

A 付録

I 索引

目次

はじめに **1**
対象となる読者の方々 1
対象となる商品 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 6
表記についての留意事項 7

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　モーション制御機能モジュールの概要

1-1　特長 **1-2**
1-2　システム構成 **1-3**
1-3　基本的な使用フロー **1-5**
1-4　仕様 **1-7**
1-4-1　一般仕様 1-7
1-4-2　性能仕様 1-7
1-4-3　機能仕様 1-10

第 2 章　モーション制御系の構成と原理

2-1　CPU ユニット内部構成 **2-2**
2-2　モーション制御系の構成 **2-3**
2-3　モーション制御系の原理 **2-5**
2-3-1　CPU ユニットのタスク 2-5
2-3-2　モーション制御のタスク動作例 2-12
2-4　EtherCAT 通信との関係 **2-19**

2-4-1 CAN application protocol over EtherCAT (CoE) 2-19
2-4-2 EtherCAT マスタ機能とモーション制御機能モジュールの関係 2-19
2-4-3 プロセスデータ通信周期とモーション制御周期との関係 2-22

第 3 章　軸と軸グループの設定

3-1 軸 3-2
3-1-1 軸の概要 3-2
3-1-2 軸パラメータの概要 3-3
3-1-3 軸変数の概要 3-6
3-1-4 軸変数の同期 3-8
3-1-5 ユーザプログラムでの軸の指定方法 3-8
3-2 軸の設定手順 3-9
3-2-1 軸の設定フロー 3-9
3-2-2 設定手順 3-9
3-3 軸グループ 3-21
3-3-1 軸グループの概要 3-21
3-3-2 軸グループパラメータの概要 3-21
3-3-3 軸グループ変数の概要 3-22
3-3-4 ユーザプログラムでの軸グループの指定方法 3-24
3-4 軸グループの設定手順 3-25
3-4-1 軸グループの設定フロー 3-25
3-4-2 設定手順 3-25

第 4 章　Sysmac Studio による配線確認

4-1 MC 試運転機能 4-2
4-1-1 MC 試運転機能一覧 4-2
4-1-2 使用手順 4-3
4-1-3 軸パラメータの設定例 4-4
4-1-4 MC 試運転機能の立ち上げ手順 4-5
4-2 センサ信号のモニタ 4-7
4-3 モータ動作の確認 4-8
4-3-1 サーボ ON 4-8
4-3-2 ジョグ送り 4-9
4-3-3 原点復帰 4-9
4-3-4 絶対値位置決め 4-10
4-3-5 相対値位置決め 4-11

第 5 章　モーション制御パラメータ

5-1 概要 5-2
5-2 軸パラメータ 5-4
5-2-1 軸パラメータの一覧 5-4
5-2-2 軸基本設定 5-6
5-2-3 単位変換設定 5-11
5-2-4 動作設定 5-19
5-2-5 拡張動作設定 5-22
5-2-6 リミット設定 5-23
5-2-7 位置カウント設定 5-23
5-2-8 サーボドライバ設定 5-26
5-2-9 原点復帰設定 5-26
5-2-10 軸パラメータの設定例 5-27
5-3 軸グループパラメータ 5-30
5-3-1 軸グループパラメータの一覧 5-30

5-3-2 軸グループ基本設定 5-30
5-3-3 軸グループ動作設定 5-32
5-3-4 軸グループの有効 5-33

第 6 章 モーション制御プログラム

6-1 概要 6-2

6-2 モーション制御命令 6-5
6-2-1 PLCopen®モーション制御用ファンクションブロック 6-5
6-2-2 モーション制御機能モジュールのモーション制御命令 6-5

6-3 状態遷移 6-7
6-3-1 モーション制御機能モジュールの状態 6-7
6-3-2 軸の状態 6-7
6-3-3 軸グループの状態 6-9

6-4 モーション制御命令の起動とステータス 6-11
6-4-1 起動についての共通ルール 6-11
6-4-2 起動のタイミングチャート 6-13
6-4-3 モーション制御命令の再起動時のタイミングチャート 6-15
6-4-4 モーション制御命令の多重起動時のタイミングチャート 6-16

6-5 位置の扱い 6-17
6-5-1 位置の種類 6-17
6-5-2 各軸種別で使用可能な位置の種類 6-18

6-6 モーション制御システム変数 6-19
6-6-1 モーション制御システム変数の概要 6-19
6-6-2 モーション制御システム変数のしくみ 6-21
6-6-3 モーション制御システム変数の一覧 6-22

6-7 カムテーブルとカムデータ変数 6-32

6-8 モーション制御プログラムの作成方法 6-36

6-9 カムテーブルの作成方法 6-38

第 7 章 手動運転

7-1 概要 7-2

7-2 サーボ ON 7-3
7-2-1 サーボ ON の手順 7-3
7-2-2 軸パラメータの設定 7-4
7-2-3 プログラム例 7-4

7-3 ジョグ送り 7-5
7-3-1 ジョグ送りの手順 7-5
7-3-2 軸パラメータの設定 7-6
7-3-3 入力変数の設定例 7-6
7-3-4 プログラム例 7-7

第 8 章 原点決め

8-1 原点決めの概要 8-2

8-2 原点復帰の手順 8-5
8-2-1 原点復帰パラメータの設定 8-5
8-2-2 原点復帰のモニタ 8-11

8-3 原点復帰の動作 8-12

8-4 絶対値エンコーダ原点設定 8-13
8-4-1 機能概要 8-15

8-4-2 設定手順 8-16
8-5 高速原点復帰 8-18

第 9 章 モーション制御機能

9-1 単軸位置制御 9-3
9-1-1 動作概要 9-3
9-1-2 絶対値位置決め 9-4
9-1-3 相対値位置決め 9-4
9-1-4 割り込み定寸位置決め 9-4
9-1-5 サイクリック同期位置決め 9-5
9-1-6 停止 9-6
9-1-7 オーバーライド 9-10
9-2 単軸同期制御 9-12
9-2-1 同期制御の概要 9-12
9-2-2 ギア動作 9-12
9-2-3 位置指定ギア動作 9-13
9-2-4 カム動作 9-14
9-2-5 カムテーブル 9-15
9-2-6 台形パターンカム動作 9-22
9-2-7 加減算位置決め 9-23
9-2-8 主軸位相補正 9-24
9-2-9 従軸位置補正 9-25
9-2-10 マルチモーションでの同期制御の実現方法 9-26
9-3 単軸速度制御 9-28
9-3-1 速度制御 9-28
9-3-2 サイクリック同期速度制御 9-28
9-4 単軸トルク制御 9-30
9-5 単軸制御の共通機能 9-31
9-5-1 位置 9-31
9-5-2 速度 9-32
9-5-3 加速と減速 9-34
9-5-4 ジャーク 9-35
9-5-5 動作方向指定 9-36
9-5-6 モーション制御命令の再起動 9-39
9-5-7 モーション制御命令の多重起動（バッファモード） 9-44
9-6 多軸協調制御 9-50
9-6-1 動作概要 9-50
9-6-2 直線補間 9-52
9-6-3 円弧補間 9-53
9-6-4 軸グループサイクリック同期位置制御 9-54
9-6-5 多軸協調制御での停止 9-54
9-6-6 多軸協調制御でのオーバーライド 9-56
9-7 多軸協調制御の共通機能 9-57
9-7-1 多軸協調制御での速度 9-57
9-7-2 多軸協調制御での加速と減速 9-57
9-7-3 多軸協調制御でのジャーク 9-59
9-7-4 多軸協調制御のモーション制御命令の再起動 9-59
9-7-5 多軸協調制御のモーション制御命令の多重起動（バッファモード） 9-60
9-8 そのほかの機能 9-68
9-8-1 現在位置変更機能 9-68
9-8-2 トルク制限機能 9-68
9-8-3 ラッチ機能 9-69
9-8-4 ゾーン機能 9-70
9-8-5 ソフトウェアリミット機能 9-70
9-8-6 位置偏差監視機能 9-72
9-8-7 偏差カウンタリセット 9-72
9-8-8 軸間偏差監視機能 9-73
9-8-9 インポジションチェック 9-74

9-8-10　軸使用変更機能 9-76
9-8-11　デジタルカムスイッチ有効機能 9-76
9-8-12　任意座標系の 3D モーションモニタ表示 9-77

第 10 章　サンプルプログラム

10-1　サンプルプログラムの概要 10-2
10-1-1　使用機器 10-2
10-1-2　設置と配線 10-2
10-1-3　設定 10-2

10-2　基本プログラム例 10-3
10-2-1　EtherCAT 通信の監視とサーボ ON 10-3
10-2-2　マスタコントロール命令による軸動作のインタロック 10-4
10-2-3　異常の監視と異常解除（単軸動作、同期動作） 10-6
10-2-4　異常の監視と異常解除（多軸協調動作） 10-8
10-2-5　命令異常の監視方法 10-14
10-2-6　異常解除の監視方法 10-15
10-2-7　単軸での動作中における軸停止 10-17
10-2-8　多軸協調動作中での軸グループ停止 10-21
10-2-9　原点復帰と絶対値位置決め 10-28
10-2-10　命令の再起動により目標位置を変更する 10-33
10-2-11　割り込み定寸位置決め 10-38
10-2-12　命令の再起動によりカムテーブルを切り替える 10-43
10-2-13　カムプロファイル曲線を使って同期位置を補正する 10-53
10-2-14　カム動作時に主軸の位相を補正する 10-63
10-2-15　速度制御中に現在位置を変更する 10-72
10-2-16　カムデータ変数の変更とカムテーブルの保存 10-77
10-2-17　軸パラメータの一時書き換え 10-87
10-2-18　カムテーブルの終点インデックス更新 10-90

第 11 章　異常の確認と処置

11-1　異常確認と処置の概要 11-2

付録

A-1　サーボドライバ 1S シリーズとの接続 A-2
A-1-1　サーボドライバとの配線 A-2
A-1-2　サーボドライバの設定 A-2

A-2　サーボドライバ G5 シリーズとの接続 A-11
A-2-1　サーボドライバとの配線 A-11
A-2-2　サーボドライバの設定 A-11

A-3　エンコーダ入力ターミナルとの接続 A-20
A-3-1　エンコーダ入力ターミナルとの配線 A-20
A-3-2　エンコーダ入力ターミナルの設定 A-20

A-4　NX ユニットとの接続 A-26

A-5　PDS 状態遷移 A-27
A-5-1　PDS 状態制御方式 A-27
A-5-2　主回路電源 OFF 検出 A-28

A-6　用語説明 A-29
A-6-1　NJ/NX シリーズ共通 A-29
A-6-2　モーション制御機能 A-29
A-6-3　EtherCAT 通信 A-31

A-7　バージョン関連情報 A-32

A-8　マニュアル No.変更前の改訂履歴 A-36

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。
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、「A-7 バージョン関連情報 (P.A-32)」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
A	2018 年 7 月	• Man.No.の変更 • 誤記訂正
B	2019 年 1 月	誤記修正による改訂
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 • 機能追加に伴う改訂
D	2019 年 7 月	誤記修正による改訂
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	2023 年 4 月	形 NX502-1□□□の追加に伴う改訂
H	2023 年 7 月	誤記修正による改訂
J	2023 年 12 月	Sysmac Studio Ver.1.57 へのバージョンアップに伴う改訂
K	2024 年 4 月	形 NX502-1700、形 NX502-1600 の追加に伴う改訂
L	2024 年 10 月	誤記修正による改訂
M	2025 年 4 月	誤記修正による改訂
N	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

モーション制御機能モジュールの概要

モーション制御機能モジュールの特長とシステム構成、使用フローについて説明します。

1-1 特長 1-2
1-2 システム構成 1-3
1-3 基本的な使用フロー 1-5
1-4 仕様 1-7
1-4-1 一般仕様 1-7
1-4-2 性能仕様 1-7
1-4-3 機能仕様 1-10

1-1 特長

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」と省略する場合があります）とは、CPU ユニットに内蔵されたソフトウェアの機能モジュールを表します。
MC 機能モジュールは、CPU ユニットの内蔵 EtherCAT ポートを介して、最大 256 軸のモーション制御を行うことができます。
EtherCAT ポートに接続されたサーボドライバなどとサイクリック通信を行うことで、高速かつ高精度が要求されるマシン制御が実現可能です。

PLCopen®に準拠したモーション制御命令

MC 機能モジュールのモーション制御命令は、PLCopen®で標準化されたモーション制御用ファンクションブロックをベースにしています。
単軸での PTP 位置決めから補間制御、電子カムなどの同期制御に加え、速度制御やトルク制御まで、標準に準拠した命令でプログラミングができます。
また、モーション制御命令の起動ごとに、速度、加速度、減速度およびジャークを設定できるため、アプリケーションに応じた柔軟な動作制御が可能になります。



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。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で、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。

ジャークとは

ジャークは加速度や減速度の変化の比率で、「躍動」や「躍度」あるいは「加加速度」とも呼ばれています。ジャークを指定すると加減速時の速度波形が S 字型になります。

EtherCAT 通信によるデータ伝送

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 通信内蔵タイプと組み合わせることで、すべての制御情報が、高速なデータ通信により交換可能になります。
各種制御指令がデータ通信で伝達されることにより、エンコーダフィードバックパルスの応答周波数などインタフェースの仕様に制約されることなく、サーボモータの性能を最大限引き出すことができます。
また、サーボドライバの各種制御パラメータやモニタ情報を上位コントローラ側で扱えるようになるため、システムの情報管理が一元化できます。



参考

EtherCAT とは

EtherCAT は、Ethernet (IEEE802.3)準拠のオープンで超高速な産業用ネットワークシステムです。各ノードは Ethernet フレームを高速で転送するため、短い通信サイクルタイムを実現することができます。また、時計情報を共有するしくみにより、低い通信ジッタで高精度な同期制御が可能です。

1-2 システム構成

MC 機能モジュールは、装置や制御盤からのセンサ信号の状態と、ユーザプログラムのモーション制御命令からの指令を受けて、サーボドライバ、エンコーダ入力ターミナル、および NX シリーズ 位置インタフェースユニットのモーション制御を行います。

EtherCAT ネットワークでのモーション制御の構成

MC 機能モジュールでは、EtherCAT ネットワーク構成、EtherCAT カプラユニットのスレーブターミナル構成、および Sysmac Studio を使用します。

● EtherCAT ネットワーク構成

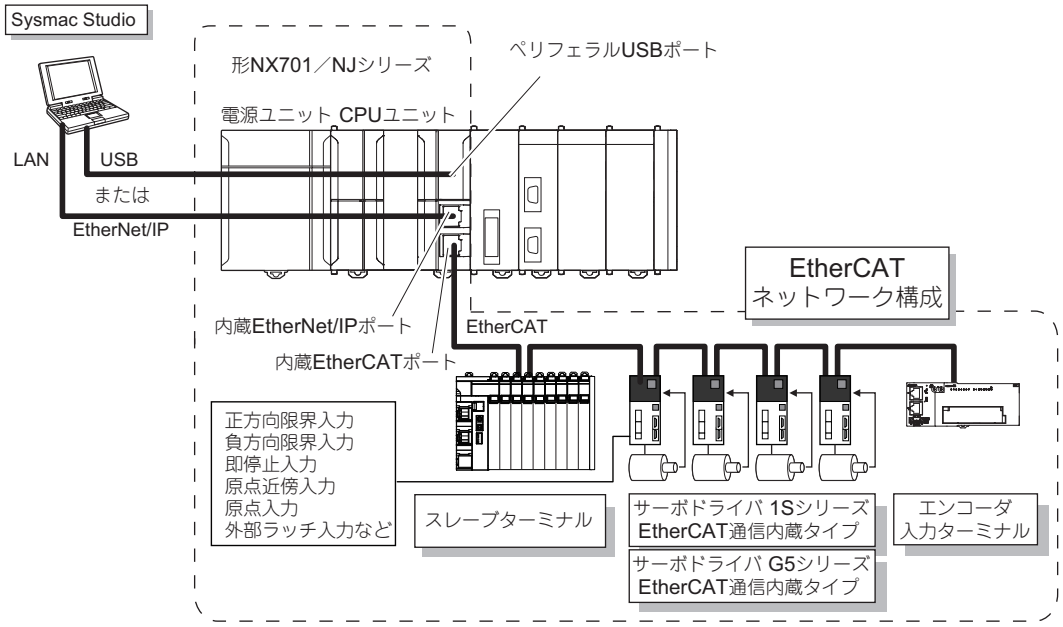
MC 機能モジュールは、CPU ユニットに内蔵された EtherCAT 通信マスタ機能ポートを使用して、サーボドライバ、エンコーダ入力ターミナルを制御します。
この構成を使用することで、定周期でゆらぎの少ない高精度なモーション制御を実現します。

● EtherCAT カプラユニットのスレーブターミナル構成

MC 機能モジュールは、EtherCAT カプラユニットに装着した位置インタフェースユニットを使用して、モータ制御用のパルス出力、およびエンコーダ入力の取り込みを行います。
この構成を使用することでも、定周期でゆらぎの少ない高精度なモーション制御を実現します。

● Sysmac Studio

CPU ユニットのペリフェラル USB ポートに市販 USB ケーブルで接続します。または CPU ユニット内蔵の EtherNet/IP ポートに Ethernet ケーブルで接続します。



使用上の注意

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、モーション制御機能の一部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。

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することができます。

CPU ユニット上でのモーション制御の構成

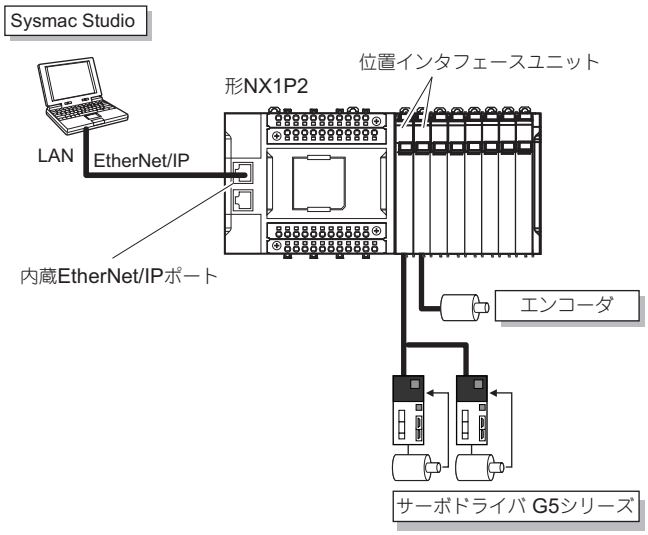
MC 機能モジュールでは、位置インタフェースユニットおよび Sysmac Studio を使用します。

- **位置インタフェースユニット**

  位置インタフェースユニットを使用して、モータ制御用のパルス出力、およびエンコーダ入力の取り込みを行います。

- **Sysmac Studio**

  CPU ユニット内蔵の EtherNet/IP ポートに Ethernet ケーブルで接続します。

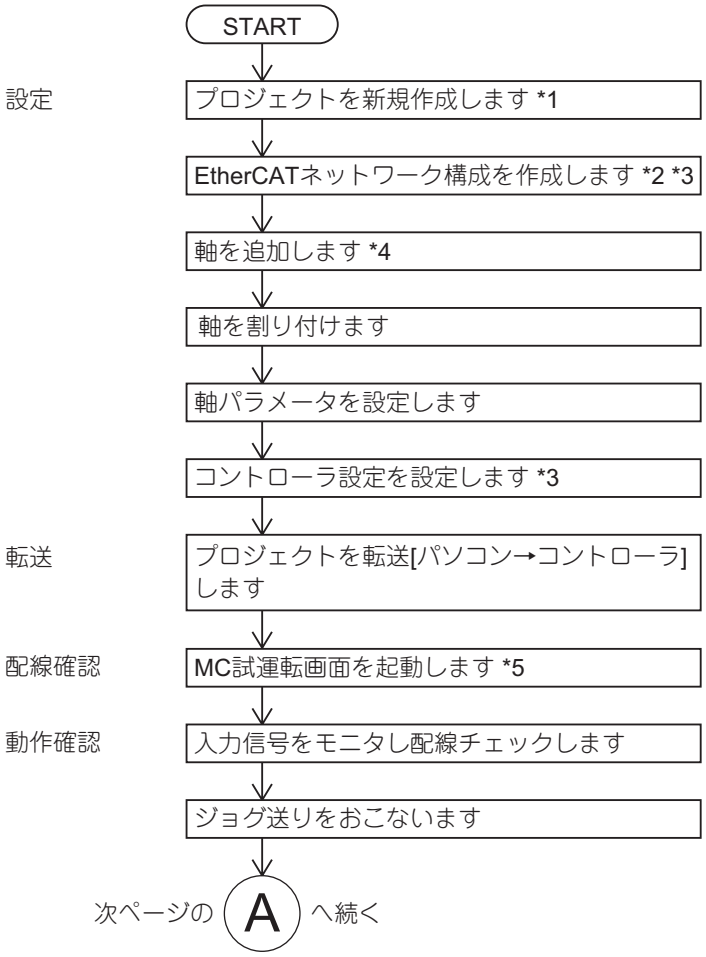


使用上の注意

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、モーション制御機能の一部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。

1-3 基本的な使用フロー

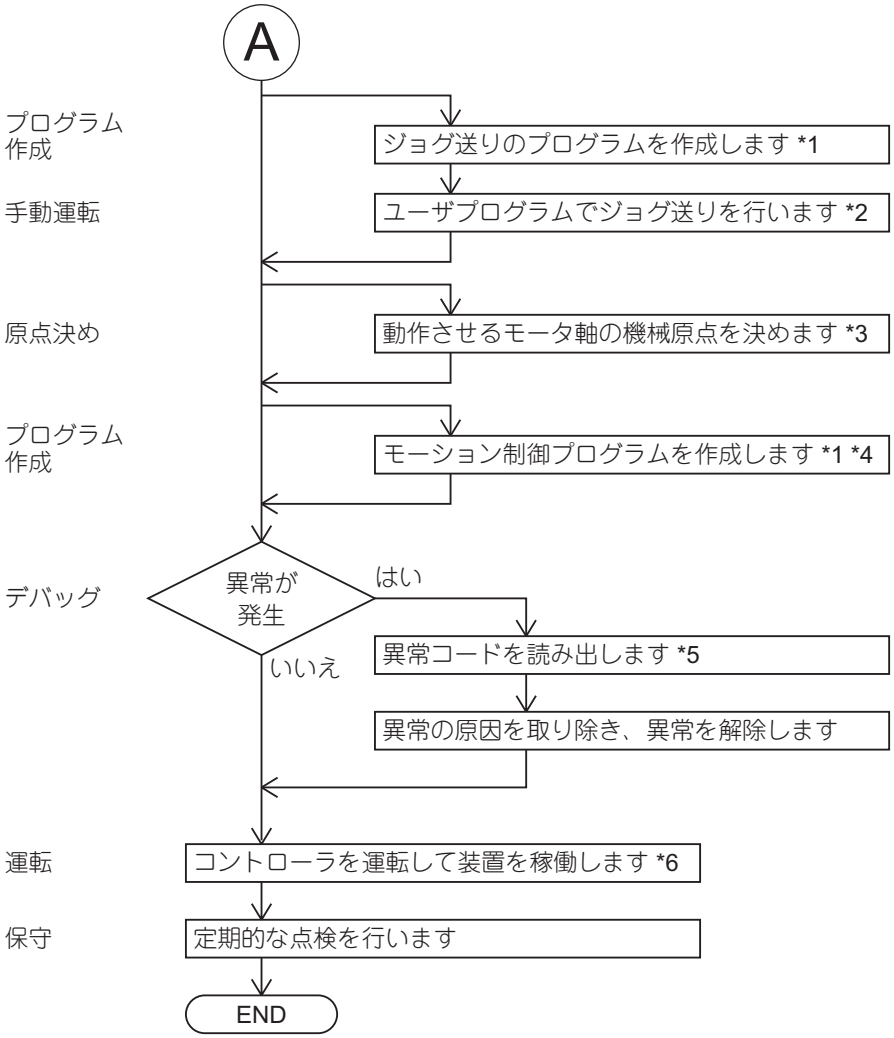
ここでは、MC 機能モジュールでモーション制御を行うための基本的な使用手順について説明します。



*1.『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。
*2. 実機を接続している場合は、オンラインで実機構成を設定することができます。
実機がない場合は、オフラインで使用する EtherCAT スレーブを選択して設定することができます。
*3.『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。
*4.「第 3 章 軸と軸グループの設定 (P.3-1)」を参照してください。
*5.「第 4 章 Sysmac Studio による配線確認 (P.4-1)」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の使用手順は、『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。



*1.「第 6 章 モーション制御プログラム (P.6-1)」を参照してください。
*2.「第 7 章 手動運転 (P.7-1)」を参照してください。
*3.「第 8 章 原点決め (P.8-1)」を参照してください。
*4.「第 10 章 サンプルプログラム (P.10-1)」を参照してください。
*5.「第 11 章 異常の確認と処置 (P.11-1)」を参照してください。
*6.「第 9 章 モーション制御機能 (P.9-1)」を参照してください。

1-4 仕様

ここでは、MC 機能モジュールの仕様について説明しています。

使用上の注意

形 NX102-90□□および形 NJ101-90□□は、モーション制御機能モジュールがありません。

1-4-1 一般仕様

一般仕様は CPU ユニットの一般仕様に準じます。
詳細については、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)』

1-4-2 性能仕様

CPU ユニットごとの性能仕様を説明します。

仕様項目			仕様（形 NX701-）	
			17□□	16□□
制御軸数	制御軸最大数 *1		256 軸	128 軸
		モーション制御軸*2	256 軸	128 軸
		単軸位置制御軸*3	−	
	使用実軸最大数*4		256 軸	128 軸
		使用モーション制御サーボ軸	256 軸	128 軸
		使用単軸位置制御サーボ軸	−	
	直線補間制御最大軸数		1 軸グループあたり 4 軸	
	円弧補間制御軸数		1 軸グループあたり 2 軸	
軸グループ最大数			64 グループ	
オーバーライド			0.00, 0.01～500.00%	
モーション制御周期			EtherCAT 通信のプロセスデータ通信周期と同じ	
マルチモーション			可	
カム	カムデータ点数	1 カムテーブルあたりの最大点数	65,535 点	
		全カムテーブルの最大点数	1,048,560 点	
	カムテーブル最大テーブル数		640 テーブル	

*1. すべての軸種別の合計です。 軸種別の詳細は、「 軸種別 (P.5-8)」を参照してください。
*2. モーション制御機能のすべての機能が使用可能な軸のことです。
*3. モーション制御機能のうち、「単軸位置制御」だけが使用可能な軸のことです。単軸位置制御軸の詳細は、「制御機能 (P.5-9)」を参照してください。
*4. 軸種別が［サーボ軸］または［エンコーダ軸］で、かつ軸使用が［使用軸］に設定された軸の合計です。

仕様項目		仕様（形 NX502-）				
		17□□*1	16□□*1	15□□	14□□	13□□
制御軸数	制御軸最大数 *2	256 軸	128 軸	128 軸	64 軸	32 軸

仕様項目			仕様（形 NX502-）				
			17□□*1	16□□*1	15□□	14□□	13□□
		モーション制御軸*3	256 軸	128 軸	128 軸	64 軸	32 軸
		単軸位置制御軸*4	−				
	使用実軸最大数*5		256 軸	128 軸	64 軸	32 軸	16 軸
		使用モーション制御サーボ軸	256 軸	128 軸	64 軸	32 軸	16 軸
		使用単軸位置制御サーボ軸	−				
	直線補間制御最大軸数		1 軸グループあたり 4 軸				
	円弧補間制御軸数		1 軸グループあたり 2 軸				
軸グループ最大数			64 グループ		32 グループ		
オーバーライド			0.00、0.01～500.00%				
モーション制御周期			EtherCAT 通信のプロセスデータ通信周期と同じ				
マルチモーション			不可				
カム	カムデータ点数	1 カムテーブルあたりの最大点数	65,535 点				
		全カムテーブルの最大点数	1,048,560 点				
	カムテーブル最大テーブル数		640 テーブル				

*1. ユニットバージョン Ver.1.66 から追加された形式です。
*2. すべての軸種別の合計です。 軸種別の詳細は、「 軸種別 (P.5-8)」を参照してください。
*3. モーション制御機能のすべての機能が使用可能な軸のことです。
*4. モーション制御機能のうち、「単軸位置制御」だけが使用可能な軸のことです。単軸位置制御軸の詳細は、「制御機能 (P.5-9)」を参照してください。
*5. 軸種別が［サーボ軸］または［エンコーダ軸］で、かつ軸使用が［使用軸］に設定された軸の合計です。

仕様項目			仕様（形 NX102-）			
			12□□	11□□	10□□	90□□
制御軸数	制御軸最大数 *1		15 軸			4 軸
		モーション制御軸 *2	11 軸			−
		単軸位置制御軸 *3	4 軸			
	使用実軸最大数 *4		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			
	直線補間制御最大軸数		1 軸グループあたり 4 軸			−
	円弧補間制御軸数		1 軸グループあたり 2 軸			−
軸グループ最大数			8 グループ			−
オーバーライド			0.00, 0.01～500.00%			
モーション制御周期			プライマリ定周期タスクの周期と同じ			
マルチモーション			不可			
カム	カムデータ点数	1 カムテーブルあたりの最大点数	65,535 点			
		全カムテーブルの最大点数	262,140 点			
	カムテーブル最大テーブル数		160 テーブル			

*1. すべての軸種別の合計です。軸種別の詳細は、「 軸種別 (P.5-8)」を参照してください。
*2. モーション制御機能のすべての機能が使用可能な軸のことです。
*3. モーション制御機能のうち、「単軸位置制御」だけが使用可能な軸のことです。単軸位置制御軸の詳細は、「制御機能 (P.5-9)」を参照してください。
*4. 軸種別が［サーボ軸］または［エンコーダ軸］で、かつ軸使用が［使用軸］に設定された軸の合計です。

仕様項目			仕様（形 NX1P2-）			
			11□□	10□□	90□□	9B□□
制御軸数	制御軸最大数 *1		12 軸	10 軸	4 軸	2 軸
		モーション制御軸 *2	8 軸	6 軸	−	
		単軸位置制御軸 *3	4 軸			2 軸
	使用実軸最大数 *4		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			2 軸
	直線補間制御最大軸数		1 軸グループあたり 4 軸		−	
	円弧補間制御軸数		1 軸グループあたり 2 軸		−	
軸グループ最大数			8 グループ		−	
オーバーライド			0.00, 0.01～500.00%			
モーション制御周期			プライマリ定周期タスクの周期と同じ			
マルチモーション			不可			
カム	カムデータ点数	1 カムテーブルあたりの最大点数	65,535 点		−	
		全カムテーブルの最大点数	262,140 点		−	
	カムテーブル最大テーブル数		80 テーブル		−	

*1. すべての軸種別の合計です。軸種別の詳細は、「 軸種別 (P.5-8)」を参照してください。
*2. モーション制御機能のすべての機能が使用可能な軸のことです。
*3. モーション制御機能のうち、「単軸位置制御」だけが使用可能な軸のことです。単軸位置制御軸の詳細は、「制御機能 (P.5-9)」を参照してください。
*4. 軸種別が［サーボ軸］または［エンコーダ軸］で、かつ軸使用が［使用軸］に設定された軸の合計です。

仕様項目			仕様（形 NJ501-）		
			□5□□	□4□□	□3□□
制御軸数	制御軸最大数 *1		64 軸	32 軸	16 軸
		モーション制御軸 *2	64 軸	32 軸	16 軸
		単軸位置制御軸 *3	−		
	使用実軸最大数 *4		64 軸	32 軸	16 軸
		使用モーション制御サーボ軸	64 軸	32 軸	16 軸
		使用単軸位置制御サーボ軸	−		
	直線補間制御最大軸数		1 軸グループあたり 4 軸		
	円弧補間制御軸数		1 軸グループあたり 2 軸		
軸グループ最大数			32 グループ		
オーバーライド			0.00, 0.01～500.00%		
モーション制御周期			EtherCAT 通信のプロセスデータ通信周期と同じ		
マルチモーション			不可		
カム	カムデータ点数	1 カムテーブルあたりの最大点数	65,535 点		
		全カムテーブルの最大点数	1,048,560 点		
	カムテーブル最大テーブル数		640 テーブル		

*1. すべての軸種別の合計です。軸種別の詳細は、「 軸種別 (P.5-8)」を参照してください。
*2. モーション制御機能のすべての機能が使用可能な軸のことです。
*3. モーション制御機能のうち、「単軸位置制御」だけが使用可能な軸のことです。単軸位置制御軸の詳細は、「制御機能 (P.5-9)」を参照してください。
*4. 軸種別が［サーボ軸］または［エンコーダ軸］で、かつ軸使用が［使用軸］に設定された軸の合計です。

仕様項目			仕様（形 NJ301-）		仕様（形 NJ101-）
			12□□	11□□	10□□
制御軸数	制御軸最大数 *1		15 軸 *2 *3	15 軸 *2 *4	6 軸
		モーション制御軸 *5	15 軸 *2 *3	15 軸 *2 *4	6 軸
		単軸位置制御軸 *6	−		
	使用実軸最大数 *7		8 軸	4 軸	2 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	−		
	直線補間制御最大軸数		1 軸グループあたり 4 軸		
	円弧補間制御軸数		1 軸グループあたり 2 軸		
軸グループ最大数			32 グループ		
オーバーライド			0.00, 0.01～500.00%		
モーション制御周期			EtherCAT 通信のプロセスデータ通信周期と同じ		
マルチモーション			不可		
カム	カムデータ点数	1 カムテーブルあたりの最大点数	65,535 点		
		全カムテーブルの最大点数	262,140 点		
	カムテーブル最大テーブル数		160 テーブル		

*1. すべての軸種別の合計です。軸種別の詳細は、「 軸種別 (P.5-8)」を参照してください。
*2. CPU ユニットのバージョンアップで追加された機能です。バージョンアップに関する情報は、「A-7 バージョン関連情報 (P.A-32)」を参照してください。
*3. 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の軸数です。それ以外の場合は 8 軸です。
*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の軸数です。それ以外の場合は 4 軸です。
*5. モーション制御機能のすべての機能が使用可能な軸のことです。
*6. モーション制御機能のうち、「単軸位置制御」だけが使用可能な軸のことです。単軸位置制御軸の詳細は、「制御機能 (P.5-9)」を参照してください。
*7. 軸種別が［サーボ軸］または［エンコーダ軸］で、かつ軸使用が［使用軸］に設定された軸の合計です。

1-4-3 機能仕様

オムロン製の制御対象機器と接続する場合の機能を以下に説明します。

機能項目		内容
制御対象サーボドライバ		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 *1 *2
制御対象エンコーダ入力ターミナル		オムロン製 EtherCAT リモート I/O ターミナル GX シリーズ 形 GX-EC0211/EC0241 *3
制御対象位置インタフェースユニット *2*4		オムロン製インクリメンタルエンコーダ入力ユニット 形 NX-EC0□□□ オムロン製 SSI 入力ユニット 形 NX-ECS□□□ オムロン製パルス出力ユニット 形 NX-PG0□□□
制御方式		EtherCAT 通信による制御コマンド発行
制御モード		位置制御 (cyclic synchronous position) 速度制御 (cyclic synchronous velocity) トルク制御 (cyclic synchronous torque)
単位変換	表示単位	パルス、mm、μm、nm、degree、inch
	電子ギア比	モータ 1 回転のパルス数／モータ 1 回転の移動量、または（モータ 1 回転のパルス数×モータ側ギア比）／（ワーク側 1 回転の移動量×ワーク側ギア比）
管理可能な位置		指令位置、フィードバック位置
軸種別		サーボ軸、仮想サーボ軸、エンコーダ軸、仮想エンコーダ軸

機能項目			内容
位置指令			倍精度実数型（LREAL 型）の負数、正数、0 （指令単位 *5）
速度指令			倍精度実数型（LREAL 型）の負数、正数、0 （指令単位／s）
加速度指令、減速度指令			倍精度実数型（LREAL 型）の正数、0 （指令単位／s^2）
ジャーク指令			倍精度実数型（LREAL 型）の正数、0 （指令単位／s^3）
単軸	単軸位置制御	絶対値位置決め	絶対座標の目標位置を指定して、位置決めを行う機能
		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、位置決めを行う機能
		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、位置決めを行う機能
		サイクリック同期絶対位置制御 *2	位置制御モードで制御周期ごとに指令位置を出力する機能
	単軸速度制御	速度制御	位置制御モードで速度制御を行う機能
		サイクリック同期速度制御	速度制御モードで制御周期ごとに速度指令を出力する機能
	単軸トルク制御	トルク制御	モータのトルク制御を行う機能
	単軸同期制御	カム動作開始	指定したカムテーブルを用いてカム動作を開始する機能
		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する機能
		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う機能
		位置指定ギア動作	主軸と従軸間のギア比と同期する位置を設定しギア動作を行う機能
		ギア動作解除	実行中のギア動作、位置指定ギア動作を中止する機能
		台形パターンカム	指定した主軸に同期して位置決めを行う機能
		主軸相対値位相補正	同期制御中の主軸の位相補正を行う機能
		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する機能
	単軸手動操作	運転可	サーボドライバの状態をサーボ ON 状態に切り替えて、軸動作を可能にする機能
		ジョグ送り	指定した目標速度にしたがって、ジョグ送りを行う機能
	単軸制御補助	軸エラーリセット	軸の異常を解除する機能
		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る機能
		パラメータ指定原点復帰 *2	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る機能
		高速原点復帰	絶対座標の「0」を目標位置とし、位置決めを行い、原点へ戻す機能
		強制停止	軸を減速停止する機能
		即停止	軸を即停止する機能
		オーバーライド値設定	軸の目標速度を変更する機能
		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する機能
		外部ラッチ有効	トリガの発生により軸の位置を記録する機能
		外部ラッチ無効	実行中のラッチを無効にする機能
		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する機能
		デジタルカムスイッチ有効 *2	軸の位置に応じてデジタル出力を ON または OFF する機能
		軸間偏差監視	指定した 2 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する機能

機能項目			内容
		偏差カウンタリセット	指令現在位置とフィードバック現在位置の間の偏差をゼロにする機能
		トルク制限	サーボドライバのトルク制限機能の有効／無効の切り替えとトルク制限値の設定を行うことで、出力トルクを制限する機能
		軸使用変更 *2	軸パラメータの［軸使用］を一時的に切り替える機能
		起動速度 *2	軸動作が開始するときの初速度を設定する機能
軸グループ	多軸協調制御	絶対値直線補間	絶対位置を指定して直線補間を行う機能
		相対値直線補間	相対位置を指定して直線補間を行う機能
		2 軸円弧補間	2 軸の円弧補間を行う機能
		軸グループサイクリック同期絶対位置制御 *2	位置制御モードで制御周期ごとに指令位置を出力する機能
	多軸協調制御補助	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する機能
		軸グループ有効	軸グループを有効にする機能
		軸グループ無効	軸グループを無効にする機能
		軸グループ構成軸書込 *2	軸グループパラメータの［構成軸］を一時的に書き換える機能
		軸グループ強制停止	補間動作中の全ての軸を減速停止する機能
		軸グループ即停止	補間動作中の全ての軸を即停止する機能
		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する機能
		軸グループ位置取得 *2	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する機能
共通	カム *6	カムテーブルプロパティ更新／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する機能
		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 CPU ユニット内蔵の不揮発性メモリに保存する機能
		カムテーブル生成 *2	入力パラメータで指定したカムプロパティとカムノードからカムテーブルを生成する機能
	パラメータ	MC 設定書込	軸パラメータおよび軸グループパラメータの一部を、一時的に書き換える機能
		軸パラメータの変更 *2	軸パラメータをユーザプログラムから参照・変更する機能
補助機能	カウントモード		リニアモード（有限長）またはロータリモード（無限長）を選択可能
	単位変換		各軸の表示単位を機械に合わせて設定可能
	加減速制御	自動加減速制御	軸や軸グループ動作時の加減速カーブをジャークで設定する機能（台形カーブ、S 字カーブ）
		加減速度変更	加減速動作中でも、加減速度の変更する機能
	インポジションチェック		位置決め完了をチェックするためのインポジション幅とインポジションチェック時間を設定する機能
	停止方法選択		即停止入力信号や限界入力信号が有効になったときの停止方法を設定する機能
	モーション制御命令の再起動		実行中のモーション制御命令の入力変数を変更し再起動することで、動作中に目標値を変更する機能
	モーション制御命令の多重起動（バッファモード）		動作中に別のモーション制御命令を起動した場合の実行開始タイミングと動作間の速度の接続方法を指定する機能

機能項目			内容
	監視機能	ソフトウェアリミット	軸の動作範囲を監視する機能
		位置偏差	軸ごとに指令現在値とフィードバック現在値との間の位置偏差を監視する機能
		速度／加減速度／トルク／補間速度／補間加減速度	軸ごとおよび軸グループごとに 警告値を設定し、監視する機能
	絶対値エンコーダ対応		オムロン製サーボモータ 1S シリーズまたは G5 シリーズの絶対値エンコーダ付きサーボモータを使用することで、電源投入時の原点復帰を不要とする機能 *7
	入力信号の論理反転 *2		即停止入力信号、正方向限界入力信号、負方向限界入力信号、原点近傍入力信号の論理を反転する機能
外部 I/F 信号			サーボドライバ側の下記入力信号を使用可能 • 原点信号、原点近傍信号、正方向限界信号、負方向限界信号、即停止信号、割込み入力信号

*1. 推奨するサーボドライバ G5 シリーズのバージョンは、モータがシリンダタイプの場合、Ver.2.1 以降で、リニアモータタイプの場合、Ver.1.1 以降です。
*2. CPU ユニットのバージョンアップで追加された機能です。バージョンアップに関する情報は、「A-7 バージョン関連情報 (P.A-32)」を参照してください。
*3. 推奨ユニットバージョンは Ver.1.1 以降です。
*4. 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、モーション制御機能の一部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。
*5. パルス単位に変換した値が符号付き整数型 40 ビットの範囲で設定可能です。
*6. カムテーブルは、Sysmac Studio のカムエディタまたはユーザプログラム中のカムテーブル生成命令で作成可能です。主軸の位相と従軸の変位を指定します。区間ごとに位相間隔幅を変更することが可能です。ユーザプログラム中でカムデータを書き換えることが可能です。
*7. 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では、絶対値タイプの外部スケールを使用することで可能です。

2

モーション制御系の構成と原理

この章では、CPU ユニットの内部構成の概要とモーション制御機能モジュールの構成と原理について説明します。

2-1　CPU ユニット内部構成 2-2
2-2　モーション制御系の構成 2-3
2-3　モーション制御系の原理 2-5
2-3-1　CPU ユニットのタスク 2-5
2-3-2　モーション制御のタスク動作例 2-12
2-4　EtherCAT 通信との関係 2-19
2-4-1　CAN application protocol over EtherCAT (CoE) 2-19
2-4-2　EtherCAT マスタ機能とモーション制御機能モジュールの関係 2-19
2-4-3　プロセスデータ通信周期とモーション制御周期との関係 2-22

2-1 CPU ユニット内部構成

ここでは、NJ/NX シリーズ CPU ユニットの内部のしくみの概要について説明します。
CPU ユニットは、以下のようなソフトウェア構造となっています。
モーション制御機能モジュールは、モーション制御を実行するソフトウェアです。

<table>
<tr><td></td><td>モーション制御機能モジュール</td><td>EtherCAT マスタ機能モジュール</td><td>そのほかの機能モジュール *1</td></tr>
<tr><td colspan="4">PLC 機能モジュール</td></tr>
<tr><td colspan="4">OS</td></tr>
</table>

*1. そのほかの機能モジュールについては、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編(SBCA-467)』を参照してください。

OS 上に、PLC 機能モジュールがあり、その上で各機能モジュールが動作します。
各機能モジュールの内容は以下のとおりです。

機能モジュール名	略称	内容
PLC 機能モジュール	PLC	全体のスケジュール管理、ユーザプログラムの実行、モーション制御モジュールへの指令、USB や SD メモリカードとのインタフェースなどを行います。
モーション制御機能モジュール	MC	ユーザプログラムのモーション制御命令の指令にしたがってモーション制御を実行し、EtherCAT マスタ機能モジュールにデータを送ります。
EtherCAT マスタ機能モジュール	ECAT	EtherCAT マスタとして、EtherCAT スレーブとの通信を行います。

(注) そのほかの機能モジュールについては、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編(SBCA-467)』を参照してください。

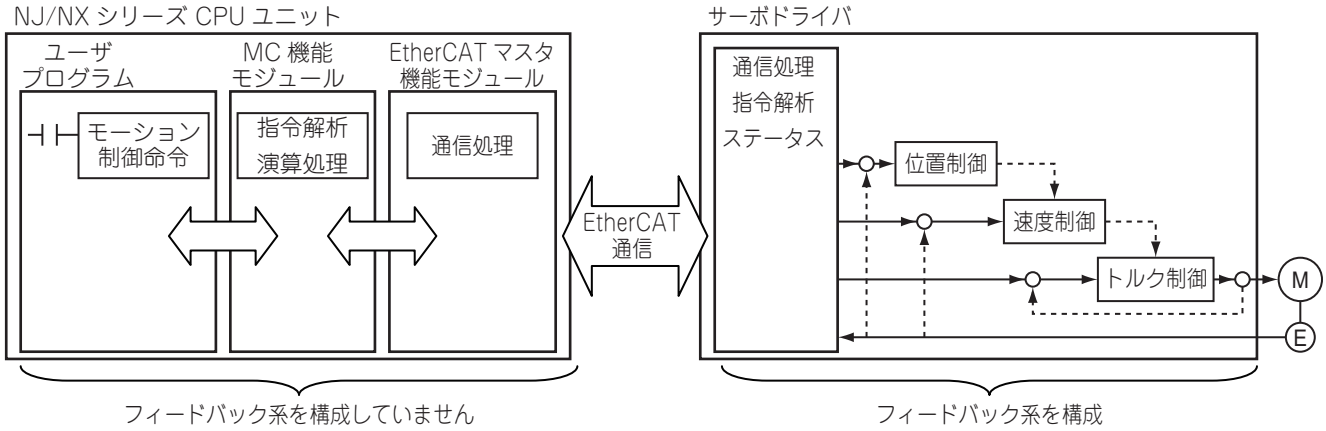
本マニュアルでは、モーション制御機能モジュール（以降、「MC 機能モジュール」と省略する場合があります）の仕様、取り扱いについて説明しています。
MC 機能モジュールを使用する場合は、NJ/NX シリーズの関連ユーザーズマニュアルも併せてご覧ください。

2-2 モーション制御系の構成

サーボにより構築される制御系は、一般的にセミクローズドループ方式によりモータ動作を制御しています。セミクローズドループ方式は、指令値に対するモータの回転量をモータに取り付けられたエンコーダにより検出します。これを機械の移動量としてフィードバックします。指令値と、モータの実回転量との偏差を算出し、これが「0」になるように制御する方法です。

MC 機能モジュールを使用した機器構成においては、CPU ユニットのユーザプログラムからの指令に関して、フィードバック情報は使用していません。サーボドライバ内でフィードバック系を構築しています。

EtherCAT ネットワーク構成の場合

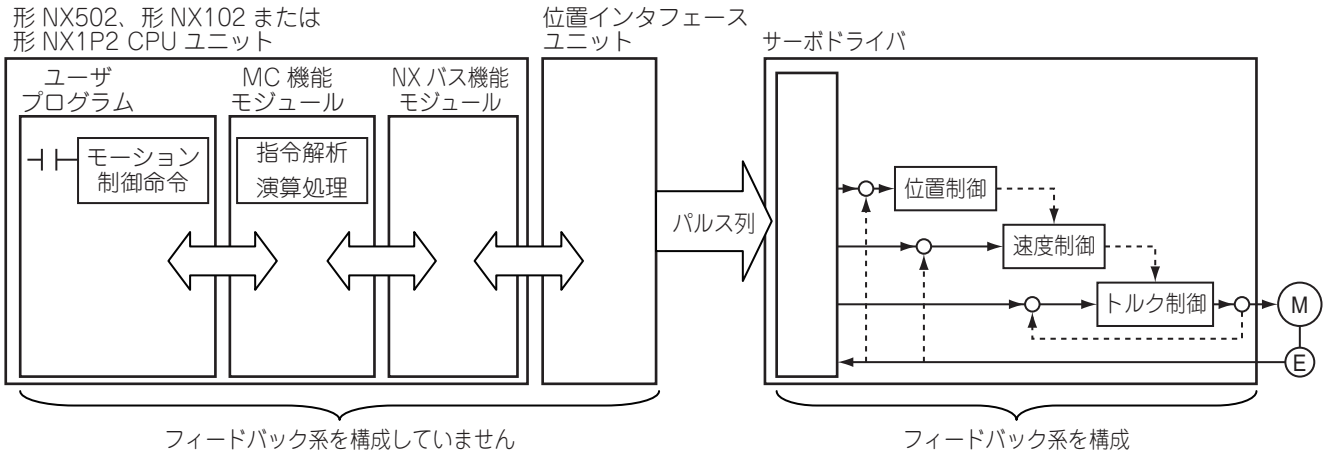


- ユーザプログラムでモーション制御命令を起動すると、MC 機能モジュールが指令を解析します。
- MC 機能モジュールは、指令の解析結果をもとに一定周期ごとにモーション演算を実行し、サーボドライバへの指令値を生成します。生成される指令値は、目標位置、目標速度、目標トルクです。
- 生成された指令値を EtherCAT 通信のプロセスデータ通信周期ごとに PDO 通信により送信します。
- サーボドライバは、EtherCAT 通信のプロセスデータ通信周期ごとの指令値に従って、位置ループ制御、速度ループ制御、トルクループ制御を実行します。
- エンコーダの現在値やサーボドライバのステータスは、EtherCAT 通信のプロセスデータ通信周期ごとに CPU ユニットへ送信します。

参考

- モーション演算と EtherCAT 通信のプロセスデータ通信は同じ周期です。
- MC 機能モジュールは、位置制御ループ、速度制御ループ、トルク制御ループが内蔵されているサーボドライバを対象としています。
- 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構成は、『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。

CPU ユニット上の構成の場合



- ユーザプログラムでモーション制御命令を起動すると、MC 機能モジュールが指令を解析します。
- MC 機能モジュールは、指令の解析結果をもとに一定周期ごとにモーション演算を実行し、位置インタフェースユニット（パルス出力ユニット）への指令値を生成します。生成される指令値は、目標位置、目標速度です。
- 生成された指令値をパルス列としてサーボドライバに出力します。
- サーボドライバは、パルス列の指令値に従って、位置ループ制御、速度ループ制御を実行します。

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の構成は、『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。

2-3 モーション制御系の原理

ここでは、CPU ユニットのタスクとモーション制御の関係について説明しています。

2-3-1 CPU ユニットのタスク

タスクとは、ユーザプログラムなどの処理の実行条件や順序の属性のことです。
NJ/NX シリーズ CPU ユニットには、次のタスクがあります。

タスクの種類	タスク名
一定周期ごとに実行されるタスク	プライマリ定周期タスク
	定周期タスク（実行優先度 5 *1, 16 *2, 17, 18）
あらかじめ指定した実行条件が成立したときに 1 回だけ実行されるタスク	イベントタスク（実行優先度 8, 48） *3

*1. 定周期タスク（実行優先度 5）は、形 NX701 CPU ユニットの場合だけ使用することができます。
*2. 定周期タスク（実行優先度 16）は、形 NX102 および形 NX1P2 CPU ユニットでは使用できません。
*3. イベントタスクは、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 以降との組み合わせで使用することができます。

ユーザプログラムおよびタスクの詳細や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 ソフトウェア編(SBCA-467)』を参照してください。

タスクの種類と優先度

形 NX701 CPU ユニットは、マルチコアプロセッサの搭載により、プライマリ定周期タスクと定周期タスク（実行優先度 5）を並列に実行することができます。
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 ユニットは、複数のタスクを同時に実行することはできません。
タスクには実行優先度があり、実行優先度が高いタスクを優先的に実行します。あるタスクを実行中に、別の実行優先度の高いタスクの実行条件が成立した場合は、実行優先度の高いタスクの方を優先的に実行します。
下表は NJ/NX シリーズ CPU ユニットで、モーション制御命令を使用できるタスクの種類と優先度です。イベントタスクは、モーション制御命令を使用できません。

タスクの種類	タスク数	優先度	動作
プライマリ定周期タスク	1 個	4	設定したタスク周期ごとに、I/O リフレッシュ、ユーザプログラム実行、モーション制御を実行します。 タスクの実行優先度が一番高く、高速・高精度に実行することができるため、同期制御や高速応答が必要な制御に適しています。 シングルタスクですべての制御を実行する場合は、プライマリ定周期タスクを使用します。

タスクの種類	タスク数	優先度	動作
定周期タスク	0～1 個	5 *1	設定したタスク周期ごとに、I/O リフレッシュ、ユーザプログラム実行、モーション制御を実行します。 タスクの実行優先度がプライマリ定周期タスクの次に高く、高速・高精度に実行することができます。 装置を構成する機能のうち、最高速で制御したい機能をプライマリ定周期タスクで制御し、その他の機能を定周期タスク（実行優先度 5）に分割して制御する場合に使用します。 マルチタスクでモーション制御を行う場合は、プライマリ定周期タスクと定周期タスク（実行優先度 5）を使用します。*2
		16 *3	設定したタスク周期ごとに、I/O リフレッシュとユーザプログラムを実行します。 プライマリ定周期タスクや定周期タスク（実行優先度 5）より遅い周期で、ユーザプログラムを実行する場合に定周期タスク（実行優先度 16）を使用します。 定周期タスク（実行優先度 16）は、プライマリ定周期タスクで I/O リフレッシュする一部のスレーブやユニットに対してユーザプログラムを記述することができます。 たとえば、同期制御や高速応答が必要な制御は、プライマリ定周期タスクや定周期タスク（実行優先度 5）で制御し、装置全体の制御は、定周期タスク（実行優先度 16）に分割して制御します。

*1. 定周期タスク（実行優先度 5）は、形 NX701 CPU ユニットの場合だけ使用することができます。
*2. プライマリ定周期タスクと定周期タスク（実行優先度 5）の両タスクを使用しモーション制御を並列実行することを「マルチモーション」と呼んでいます。
*3. 定周期タスク（実行優先度 16）は、形 NX102 および形 NX1P2 CPU ユニットでは使用できません。

（注）NJ/NX シリーズ CPU ユニットには、実行優先度が 17 および 18 の定周期タスクがあります。ただし、これらのタスクでは、モーション制御命令を使用できません。また、I/O リフレッシュは実行しません。

使用上の注意

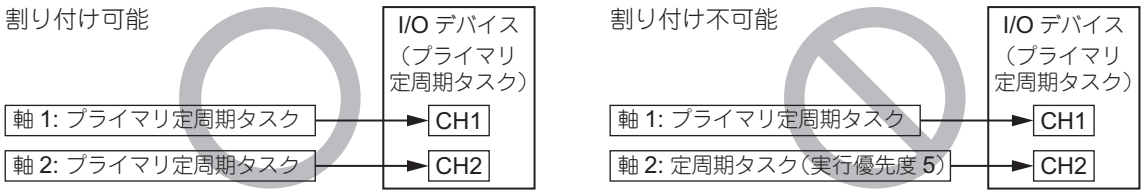
モーション制御命令は、プライマリ定周期タスクと定周期タスク（実行優先度 5 と 16）で使用することができます。
上記以外のタスクでモーション制御命令を使用すると、Sysmac Studio でのビルド時に異常となります。

参考

形 NX701 CPU ユニットでは、プライマリ定周期タスクと定周期タスク（実行優先度 5）でモーション制御を実行できます。2 つのモーション制御を区別する必要がある場合、プライマリ定周期タスクのモーション制御を「モーション制御 1」、定周期タスク（実行優先度 5）のモーション制御を「モーション制御 2」と呼びます。

タスクの割り付け

- 軸および軸グループは、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）のいずれかのタスクに割り付けることができます。軸に割り付ける I/O デバイスのタスクは、割り付ける軸と同じタスクでなければなりません。



- プライマリ定周期タスクに割り付けた軸および軸グループに対して、定周期タスク（実行優先度 16）で動作するユーザプログラムからモーション制御命令を実行することができます。

使用上の注意

- プライマリ定周期タスクに割り付けた軸および軸グループに対して、定周期タスク（実行優先度 5）で動作するユーザプログラムからモーション制御命令を実行することはできません。実行した場合、定周期タスク（実行優先度 5）に割り付けた軸の有無によって、下記のように動作が異なります。
  a）定周期タスク（実行優先度 5）に割り付けた軸が存在する場合、「軸指定不正（イベントコード: 54600000 Hex）」が発生します。
  b）定周期タスク（実行優先度 5）に割り付けた軸が存在しない場合、モーション制御命令は起動されますが、起動されるタイミングにばらつきがあるため、使用しないでください。
  同様に、定周期タスク（実行優先度 5）に割り付けた軸および軸グループに対して、プライマリ定周期タスクで動作するユーザプログラムからモーション制御命令を実行することはできません。
- 定周期タスク（実行優先度 5）に割り付けた軸および軸グループに対して、定周期タスク（実行優先度 16）で動作するユーザプログラムからモーション制御命令を実行することはできません。実行した場合は、「軸指定不正（イベントコード: 54600000 Hex）」が発生します。

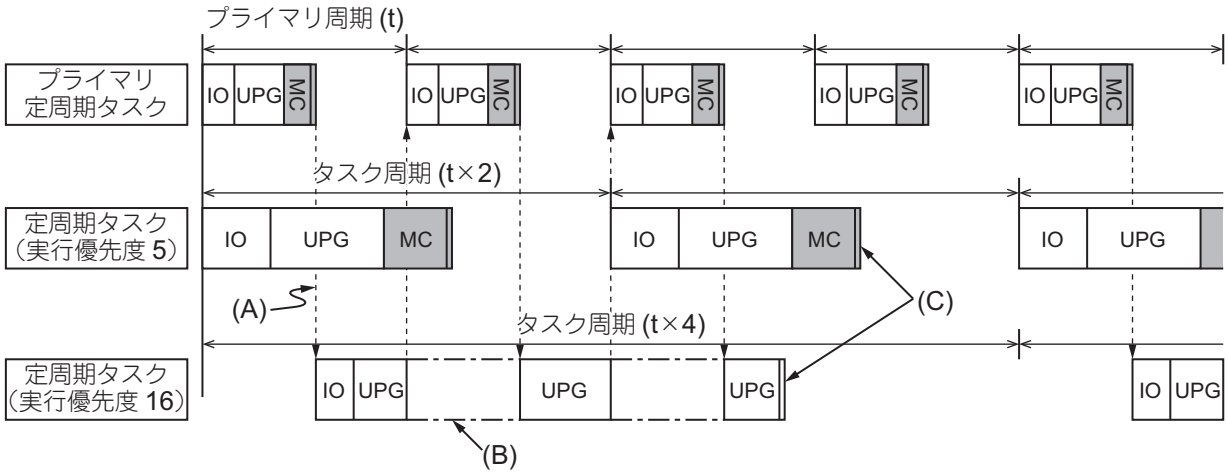
参考

軸および軸グループについては、「第 3 章 軸と軸グループの設定 (P.3-1)」を参照してください。

タスクの基本動作

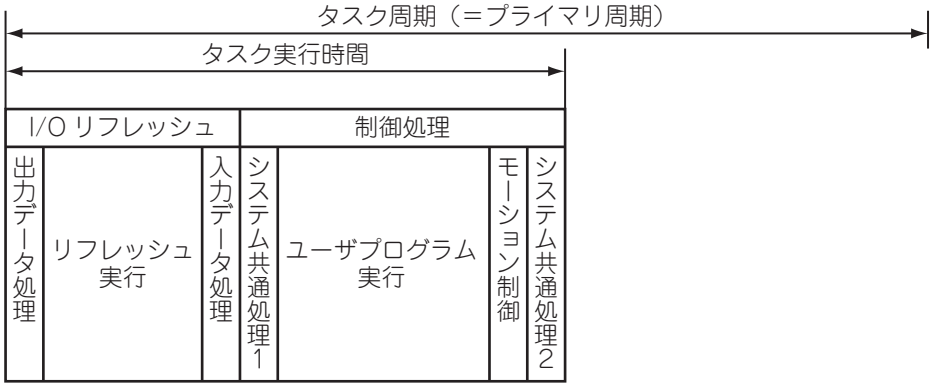
● タスク全体の動作

プライマリ定周期タスクと定周期タスクは、プライマリ定周期タスクのタスク周期（以降は「プライマリ周期」と呼びます）を基準として動作します。
プライマリ定周期タスクおよび定周期タスク（実行優先度 5）には、I/O リフレッシュ、ユーザプログラム実行のほかに、システム共通処理、モーション制御を含みます。
ユーザプログラムに記述されたモーション制御命令は、そのタスクの END 命令実行後、次のモーション制御(MC)のタイミングで演算を実行します。



記号	内容
IO	I/O リフレッシュ
UPG	ユーザプログラム実行
MC	モーション制御
(A)	点線は、タスク遷移を表します。
(B)	一点鎖線は、タスク処理中断状態を表します。
(C)	二重線は、タスク内処理の実行完了を表します。

● プライマリ定周期タスクの動作

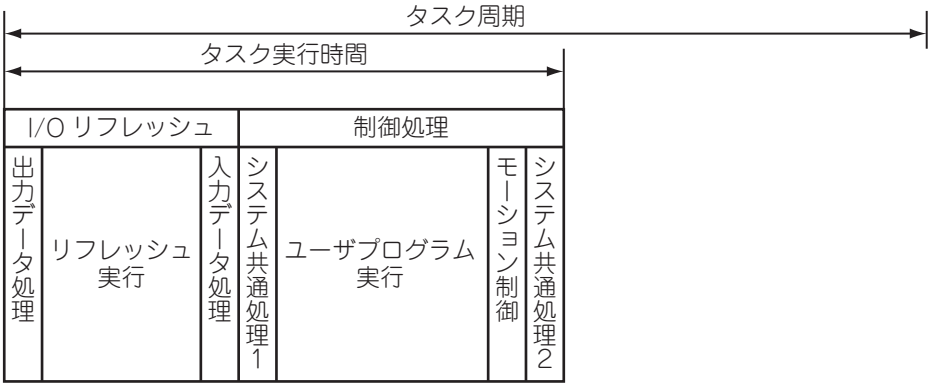


処理	処理内容
出力データ処理	• I/O リフレッシュを実行する I/O（出力）への OUT リフレッシュデータを生成します。 • 強制値リフレッシュが設定されている場合は、OUT リフレッシュデータに強制値（出力）を反映します。
リフレッシュ実行	I/O とのデータ交換を実行します。
入力データ処理	• イベントタスクの条件式の一致判定を実行します。 • I/O リフレッシュを実行した I/O（入力）ユニットの IN リフレッシュデータを取り込みます。 • 強制値リフレッシュが設定されている場合は、取り込んだ IN リフレッシュデータに強制値（入力）を反映します。
システム共通処理 1	• 変数のタスク間排他制御処理（参照タスクの場合） • モーション入力処理 *1 • データトレース処理（サンプリング、トリガ判定）
ユーザプログラム実行	タスクに割り付けられたユーザプログラムを割り付け順に実行します。

処理	処理内容
モーション制御 1 *2	プライマリ定周期タスク、および定周期タスク（実行優先度 16）のユーザプログラムのモーション制御命令からモーション制御指令を実行します。 モーション出力処理 *3
システム共通処理 2	• 変数のタスク間排他制御処理（更新タスクの場合） • タスク実行との同時性を確保させるためのコントローラ外部からの変数アクセス処理（[タスク設定] にて設定した「変数アクセス時間」分） • EtherNet/IP タグデータリンク処理の場合は、タグ（ネットワーク公開属性の変数）に対して更新タスクを設定すると、上記変数アクセス処理として実行します。

*1. サーボドライバなどから受信したデータをもとに、モーション制御システム変数のサーボドライバ状態や軸現在値などが更新されます。
*2. モーション制御 1 に割り付けた軸のシステム定義変数は、「_MC_AX[0-255]」、または「_MC1_AX[0-255]」を使用します。同様に軸グループのシステム定義変数は、「_MC_GRP[0-63]」、または「_MC1_GRP[0-63]」を使用します。
軸のシステム定義変数は、「3-1-3 軸変数の概要 (P.3-6)」を、軸グループのシステム定義変数は、「3-3-3 軸グループ変数の概要 (P.3-22)」を参照してください。
*3. 次のプライマリ定周期タスクの「I/O リフレッシュ」で、サーボドライバなどへ送信されます。

● 実行優先度 5 の定周期タスク動作



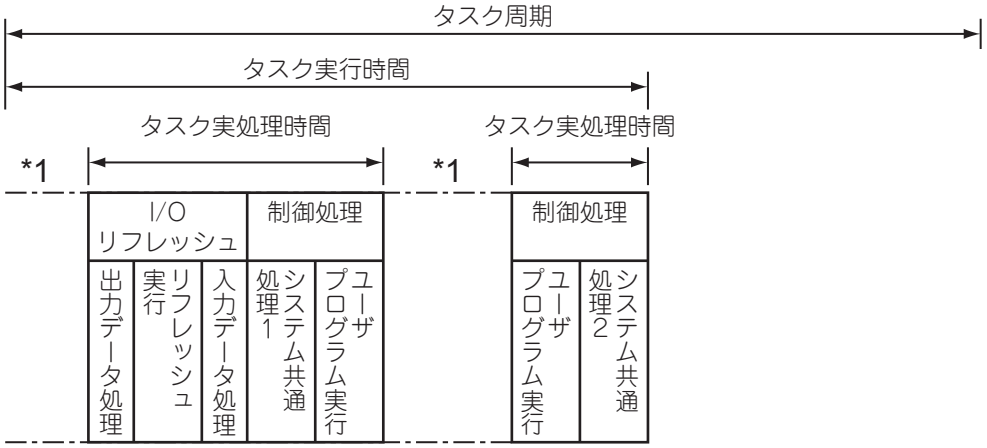
処理	処理内容
出力データ処理	• I/O リフレッシュを実行する I/O（出力）への OUT リフレッシュデータを生成します。 • 強制値リフレッシュが設定されている場合は、OUT リフレッシュデータに強制値（出力）を反映します。
リフレッシュ実行	I/O とのデータ交換を実行します。
入力データ処理	• I/O リフレッシュを実行した I/O（入力）ユニットの IN リフレッシュデータを取り込みます。 • 強制値リフレッシュが設定されている場合は、取り込んだ IN リフレッシュデータに強制値（入力）を反映します。
システム共通処理 1	• 変数のタスク間排他制御処理（参照タスクの場合） • モーション入力処理 *1 • データトレース処理（サンプリング、トリガ判定）
ユーザプログラム実行	タスクに割り付けられたユーザプログラムを割り付け順に実行します。
モーション制御 2 *2	• 定周期タスク（実行優先度 5）のユーザプログラムのモーション制御命令からモーション制御指令を実行します。 • モーション出力処理 *3
システム共通処理 2	• 変数のタスク間排他制御処理（更新タスクの場合） • タスク実行との同時性を確保させるためのコントローラ外部からの変数アクセス処理（[タスク設定] にて設定した「変数アクセス時間」分）

*1. サーボドライバなどから受信したデータをもとに、モーション制御システム変数のサーボドライバ状態や軸現在値などが更新されます。

*2. モーション制御 2 に割り付けた軸のシステム定義変数は、「_MC2_AX[0-255]」を使用します。同様に軸グループのシステム定義変数は、「_MC2_GRP[0-63]」を使用します。
軸のシステム定義変数は、「3-1-3 軸変数の概要 (P.3-6)」を、軸グループのシステム定義変数は、「3-3-3 軸グループ変数の概要 (P.3-22)」を参照してください。
*3. 次の定周期タスク（実行優先度 5）の「I/O リフレッシュ」で、サーボドライバなどへ送信されます。

● 実行優先度 16 の定周期タスク動作

定周期タスク（実行優先度 16）は I/O リフレッシュを実行することができます。



*1. 実行優先度の高いタスクの実行によって、タスク実行を一時的に中断します。

タスクの周期

シングルタスクの場合は、プライマリ定周期タスクのタスク周期であるプライマリ周期で動作します。
シングルタスクの場合、プライマリ周期が自動的にモーション制御周期（= EtherCAT 通信のプロセスデータ通信周期）となります。
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、シングルタスクだけの制御です。

マルチモーションの場合は、プライマリ周期と定周期タスク（実行優先度 5）のタスク周期で動作します。
マルチモーションの場合、モーション制御周期も 2 つとなり、EtherCAT 通信のプロセスデータ通信周期も自動的にそれぞれのタスク周期となります。

定周期タスクはプライマリ周期に同期して実行します。定周期タスクのタスク周期は、プライマリ周期の整数倍を設定してください。
たとえば、プライマリ周期を 1ms、定周期タスク（実行優先度 5）のタスク周期を 2ms、定周期タスク（実行優先度 16）のタスク周期を 4ms に設定した場合は、プライマリ周期の 2 回に 1 回の間隔で、プライマリ定周期タスクと定周期タスク（実行優先度 5）の周期の先頭がそろいます。同様に、定周期タスク（実行優先度 16）とプライマリ定周期タスクとは、4 回に 1 回の間隔で周期の先頭がそろいます。

タスクの周期の詳細は、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)』を参照してください。

参考

2 つのプロセスデータ通信周期を区別する必要がある場合、プライマリ定周期タスクの通信周期を「プロセスデータ通信周期 1」、定周期タスク（実行優先度 5）の通信周期を「プロセスデータ通信周期 2」と呼びます。

● 形 NX701 CPU ユニットの設定可能なタスク周期

形 NX701 CPU ユニットの場合、タスクの種類により、設定可能なタスク周期は次のようになります。

タスク	設定可能なタスク周期
プライマリ定周期タスク	125μs, 250μs～8ms（250μs 単位で指定可能）
定周期タスク（実行優先度 5）	125μs, 250μs～100ms（250μs 単位で指定可能）
定周期タスク（実行優先度 16）	1ms～100ms（250μs 単位で指定可能）

（注）プライマリ定周期タスクと定周期タスクのタスク周期の設定条件は次のようになります。
定周期タスクのタスク周期は、プライマリ定周期タスクの整数倍としてください。
各タスクのタスク周期の最小公倍数が、600ms 以下となるようにしてください。

● 形 NX502 CPU ユニットの設定可能なタスク周期

形 NX502 CPU ユニットの場合、タスクの種類により、設定可能なタスク周期は次のようになります。

タスク	設定可能なタスク周期
プライマリ定周期タスク	250μs～8ms（250μs 単位で指定可能）
定周期タスク（実行優先度 16）	1ms～100ms（250μs 単位で指定可能）

（注）プライマリ定周期タスクと定周期タスクのタスク周期の設定条件は次のようになります。
定周期タスクのタスク周期は、プライマリ定周期タスクの整数倍としてください。
各タスクのタスク周期の最小公倍数が、600ms 以下となるようにしてください。

● 形 NX102 CPU ユニットの設定可能なタスク周期

形 NX102 CPU ユニットの場合、タスクの種類はプライマリ定周期タスクだけです。設定可能なタスク周期は次のようになります。

タスク	設定可能なタスク周期
プライマリ定周期タスク	1ms～32ms（250μs 単位で指定可能）

● 形 NX1P2 CPU ユニットの設定可能なタスク周期

形 NX1P2 CPU ユニットの場合、タスクの種類はプライマリ定周期タスクだけです。設定可能なタスク周期は次のようになります。

タスク	設定可能なタスク周期
プライマリ定周期タスク	2ms～8ms（250μs 単位で指定可能）

● NJ シリーズ CPU ユニットの設定可能なタスク周期

NJ シリーズ CPU ユニットの場合、プライマリ定周期タスクと定周期タスク（実行優先度 16）のタスク周期の可能な組み合わせは下表となります。

プライマリ周期	定周期タスクの設定可能なタスク周期
500μs *1	1ms, 2ms, 3ms, 4ms, 5ms, 8ms, 10ms, 15ms, 20ms, 25ms, 30ms, 40ms, 50ms, 60ms, 75ms, 100ms
1ms	1ms, 2ms, 3ms, 4ms, 5ms, 8ms, 10ms, 15ms, 20ms, 25ms, 30ms, 40ms, 50ms, 60ms, 75ms, 100ms
2ms	2ms, 4ms, 8ms, 10ms, 20ms, 30ms, 40ms, 50ms, 60ms, 100ms
4ms	4ms, 8ms, 20ms, 40ms, 60ms, 100ms

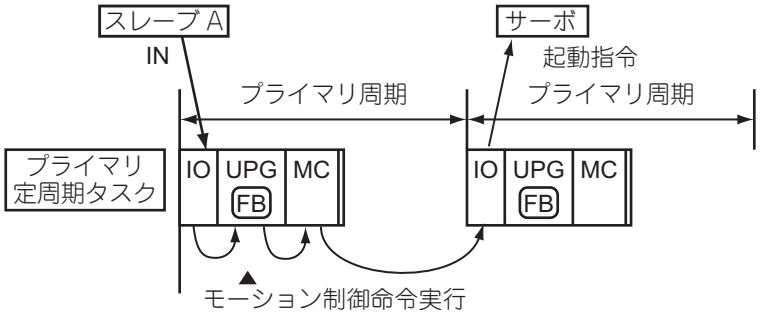
*1. 形 NJ301 CPU ユニットでは、ユニットバージョン Ver.1.03 以降で使用することができます。
形 NJ101-10□□では、使用できません。

2-3-2　モーション制御のタスク動作例

モーション制御命令は、プライマリ定周期タスク、および定周期タスク（実行優先度 5 と 16）に記述することができます。
ここでは、それぞれの場合のタスク動作例を説明します。

プライマリ定周期タスクでモーション制御命令を使用する場合

高速なモーション制御が要求される場合は、プライマリ定周期タスクにモーション制御命令(FB)を記述します。



1　データの取り込み
I/O リフレッシュ(IO)で、EtherCAT スレーブ（スレーブ A）の入力データが取り込まれます。

2　命令の起動
ユーザプログラム実行(UPG)で、取り込まれた入力データをもとにモーション制御命令(FB)が起動されます。
モーション制御命令の出力変数は、この時点で更新されます。

3　指令の生成
起動されたモーション制御命令(FB)に従って、そのプライマリ定周期タスクにある直後のモーション制御(MC)でモーション演算が実行され、サーボドライバなどへの起動指令が生成されます。

4　指令の送信
生成された起動指令は、次の周期の I/O リフレッシュ(IO)で、サーボドライバなどへ送信されます。

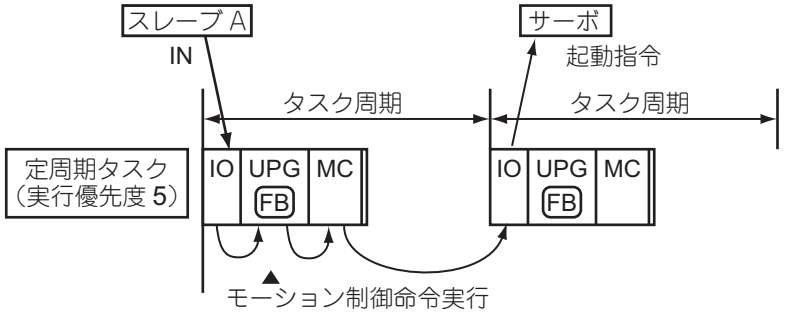
参考

入力からサーボドライバなどへの起動指令出力までが高速に処理されるので、モーション制御命令はプライマリ定周期タスクに記述することを推奨します。

実行優先度 5 の定周期タスクでモーション制御命令を使用する場合

プライマリ定周期タスクの次に高速なモーション制御が要求される場合は、定周期タスク（実行優先度 5）にモーション制御命令(FB)を記述します。
基本的な動作は、プライマリ定周期タスクと同じです。

● 処理タイミング



1 **データの取り込み**

I/O リフレッシュ(IO)で、EtherCAT スレーブ（スレーブ A）の入力データが取り込まれます。

2 **命令の起動**

ユーザプログラム実行(UPG)で、取り込まれた入力データをもとにモーション制御命令(FB)が起動されます。

モーション制御命令の出力変数は、この時点で更新されます。

3 **指令の生成**

起動されたモーション制御命令(FB)に従って、そのプライマリ定周期タスクにある直後のモーション制御(MC)でモーション演算が実行され、サーボドライバなどへの起動指令が生成されます。

4 **指令の送信**

生成された起動指令は、次の周期の I/O リフレッシュ(IO)で、サーボドライバなどへ送信されます。

参考

- 定周期タスク（実行優先度 5）は、形 NX701 CPU ユニットの場合だけ使用することができます。
- 装置を構成する機能のうち、最高速で制御したい機能をプライマリ定周期タスクに、その他の機能を定周期タスク（実行優先度 5）に分割して並列制御する場合に定周期タスク（実行優先度 5）を使用します。

● マルチモーションでの軸変数の反映タイミング

マルチモーションとは、プライマリ定周期タスクと定周期タスク（実行優先度 5）の両タスクを使用して並列制御することをいいます。

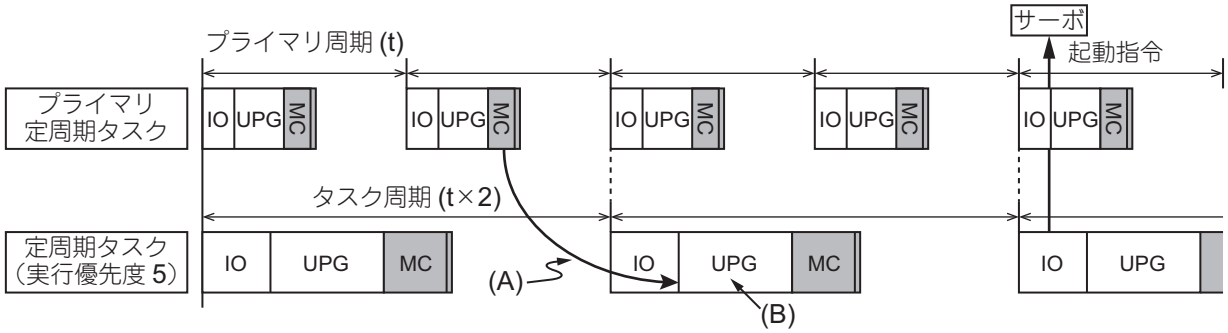
マルチモーションで、プライマリ定周期タスクで制御する軸の、軸変数の値を定周期タスク（実行優先度 5）のユーザプログラムで参照することができます。または、その逆に定周期タスク（実行優先度 5）で制御する軸の、軸変数の値をプライマリ定周期タスクのユーザプログラムで参照することもできます。

参考

- 軸変数の同期については、「3-1-4 軸変数の同期 (P.3-8)」を参照してください。
- 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 16）のユーザプログラムで参照することができます。詳細については、「実行優先度 16 の定周期タスクでモーション制御命令を使用する場合 (P.2-15)」、および「3-1-4 軸変数の同期 (P.3-8)」を参照してください。

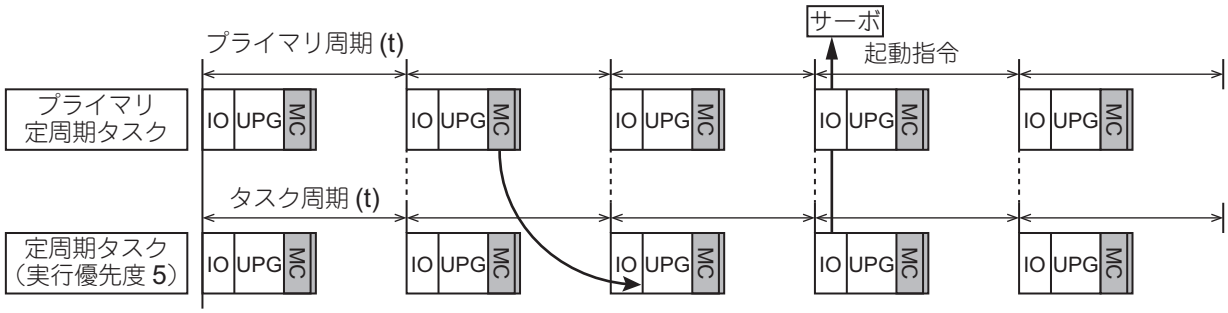
軸変数の値は、参照する側のタスク周期に同期して実行しています。
軸変数の値を参照する側のタスクのユーザプログラムが実行している途中で、参照している軸変数の値が書きかわることはありません。

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。



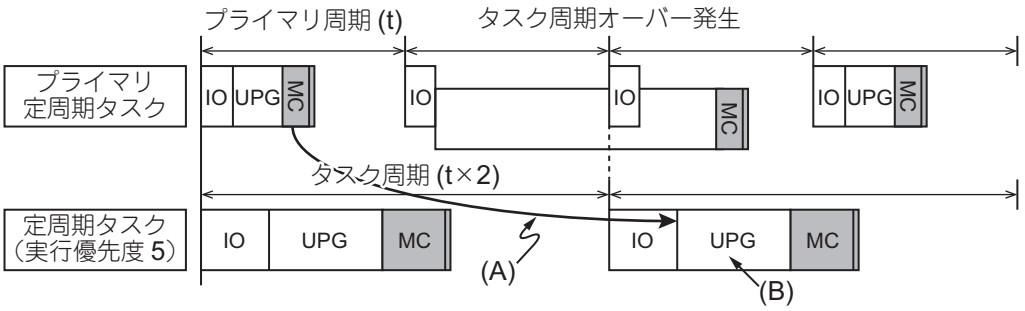
記号	説明
（A）	軸変数の更新。 定周期タスク（実行優先度 5）のユーザプログラム実行開始位置にかかわらず、タスク周期の先頭がそろう周期の直前のプライマリ定周期タスクの実行結果を反映します。
（B）	ユーザプログラム実行の途中で、軸変数の値が書きかわることはありません。

タスク周期が同じ場合も、同様になります。



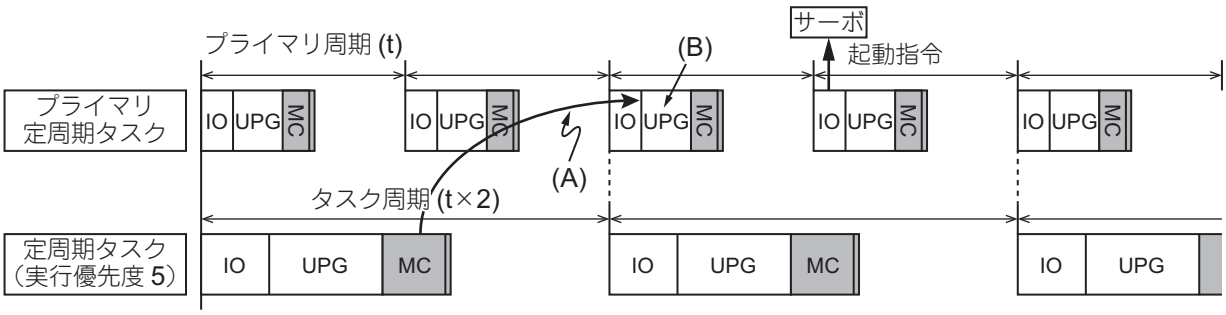
参考

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。

定周期タスク（実行優先度 5）の軸変数の値をプライマリ定周期タスクに反映するタイミングは、次のようになります。



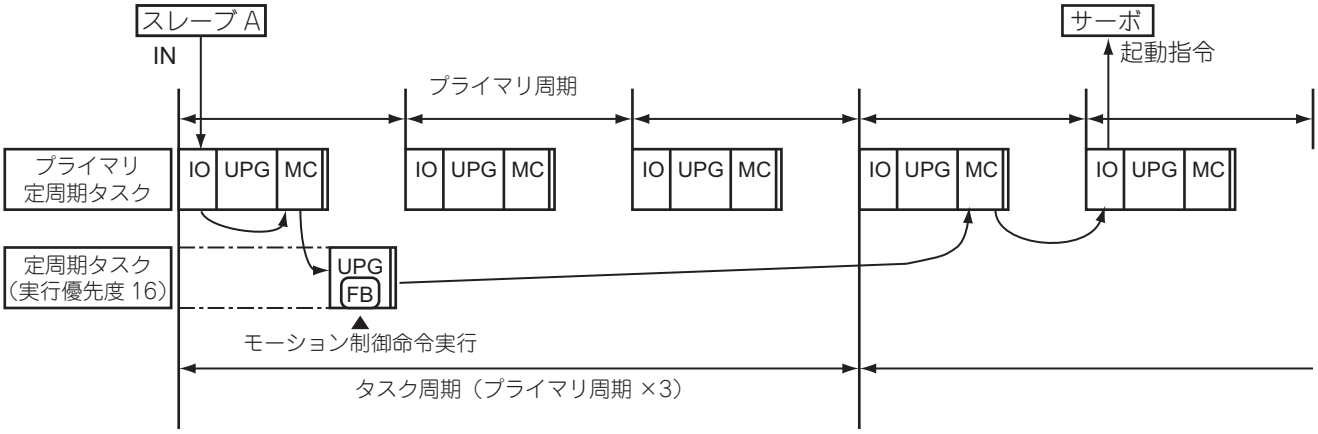
記号	説明
(A)	軸変数の更新。 タスク周期の先頭がそろう周期の直前の定周期タスク（実行優先度 5）の実行結果を反映します。
(B)	ユーザプログラム実行の途中で、軸変数の値が書きかわることはありません。

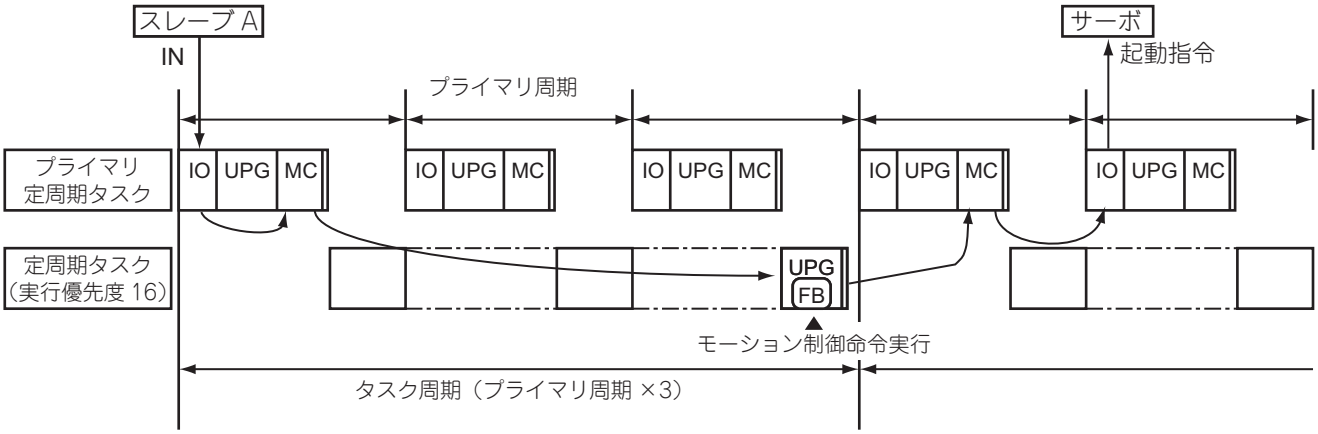
実行優先度 16 の定周期タスクでモーション制御命令を使用する場合

高速なモーション制御が要求されず、ユーザプログラムが大きくなる場合などは、定周期タスク（実行優先度 16）にモーション制御命令を記述します。

● 処理タイミング

定周期タスク（実行優先度 16）の同じタスク周期で実行したモーション制御命令(FB)は、同じタイミングでモーション制御(MC)を行います。
このため、複数の軸でも同時に起動あるいは中断できます。





1 **データの取り込み**

I/O リフレッシュ(IO)で、EtherCAT スレーブ（スレーブ A）の入力データが取り込まれます。

2 **命令の起動**

定周期タスク（実行優先度 16）のユーザプログラム実行(UPG)で、取り込まれた入力データをもとにモーション制御命令(FB)が起動されます。
モーション制御命令の出力変数は、この時点で更新されます。

3 **指令の生成**

起動されたモーション制御命令(FB)に従って、定周期タスク（実行優先度 16）のタスク周期で、モーション制御命令(FB)が実行されます。
その定周期タスクの次のプライマリ定周期タスクにあるモーション制御(MC)でモーション演算が実行され、サーボドライバなどへの起動指令が生成されます。

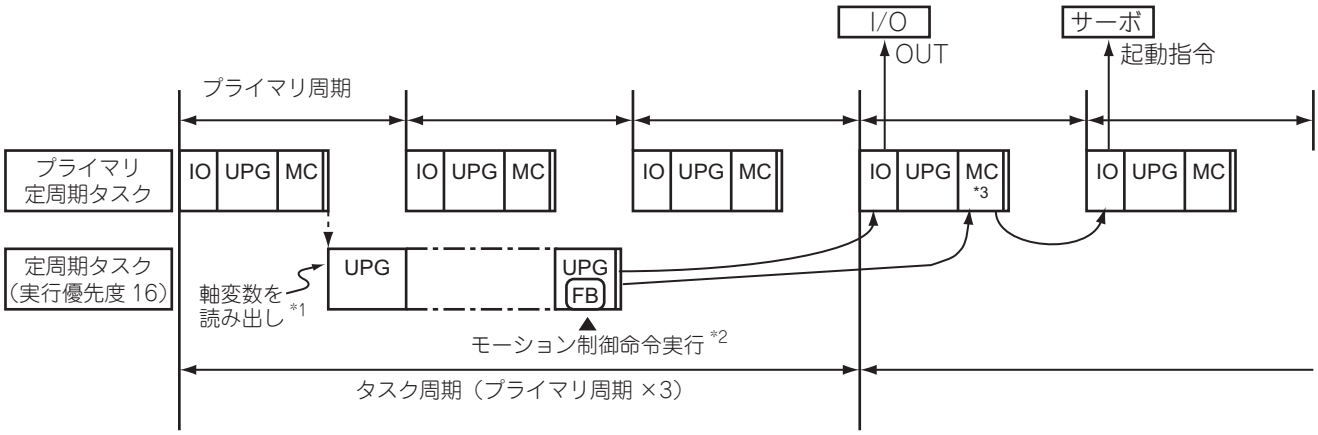
4 **指令の送信**

生成された起動指令は、さらに次の周期の I/O リフレッシュ(IO)で、サーボドライバなどへ送信されます。

● 軸変数の反映タイミング

軸変数は、MC 機能モジュールの軸パラメータの一部と現在位置や異常情報などのモニタ情報からなるシステム定義変数です。
定周期タスク（実行優先度 16）でプライマリ定周期タスクの軸変数を参照する場合、定周期タスク（実行優先度 16）の実行開始時の値が読み出されます。

また、モーション制御命令(FB)の実行時点では、軸変数の値は書き換わりません。次の定期タスク（実行優先度 16）の最初に処理するモーション制御(MC)で書き換わります。



*1. 定周期タスク（実行優先度 16）のユーザプログラム実行開始時のプライマリ定周期タスクの軸変数値が読み出されます。
*2. 定周期タスク（実行優先度 16）のモーション制御命令(FB)実行時点では、軸変数の値は書き換わりません。
*3. このモーション制御(MC)で書き換わります。

使用上の注意

- モーション制御命令を定周期タスク（実行優先度 16）に記述する場合、定周期タスク（実行優先度 16）のタスク周期が長くなると、サーボドライバなどの応答時間も長くなります。
- 非常停止など外部からの緊急停止については、安全に停止できることをあらかじめ考慮してください。
- 定周期タスク（実行優先度 16）に記述したモーション制御命令の実行タイミングと、I/O 制御の実行タイミングが異なることに注意してプログラムを作成してください。
- 定周期タスク（実行優先度 16）は、形 NX102 および形 NX1P2 CPU ユニットでは使用できません。

参考

軸変数については、「3-1-3 軸変数の概要 (P.3-6)」を参照してください。

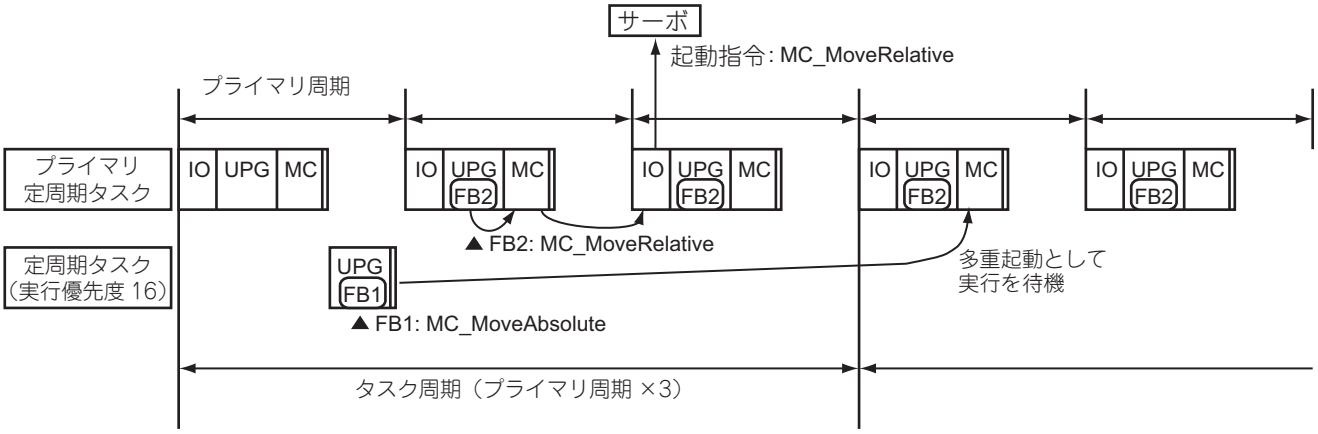
2 種類のタスクでモーション制御命令を使用する場合

同じ軸で高速なモーション制御が要求される処理と要求されない処理がある場合は、プライマリ定周期タスクと定周期タスク（実行優先度 16）の両方に、モーション制御命令(FB)を記述します。

定周期タスク（実行優先度 16）のタスク周期内で、両方のタスクのモーション制御命令(FB)を起動すると、MC 機能モジュールはプライマリ定周期タスクの命令のモーション演算を先に実行します。

たとえば、定周期タスク（実行優先度 16）で MC_MoveAbsolute を起動します。その後、プライマリ定周期タスクで、同じ軸に対して MC_MoveRelative を起動する場合は、以下のような動作となります。

- MC 機能モジュールは MC_MoveRelative を先に実行し、MC_MoveAbsolute を命令の多重起動として実行します。



モーション制御命令の出力変数の値やモーション制御システム変数の値は、その命令を起動したタスクの I/O リフレッシュのタイミングで変化します。そのため、複数のタスクで同じ軸に対してモーション制御命令を使用すると、タスクによって状態が異なって見える場合があります。
それぞれのタスクの処理を、理解していただいた上でプログラムを作成してください。

使用上の注意

- プライマリ定周期タスクと定周期タスク（実行優先度 16）の両方に、同じ軸のモーション制御命令を記述する場合は、モーション制御命令の実行順序やモーション制御システム変数の更新タイミング、および指令値の出力タイミングに注意してプログラムを作成してください。
- 同じ軸のモーション制御システム変数を複数のタスクで使用する場合、モーション制御システム変数の更新タイミングの違いに注意してプログラムを作成してください。
- 定周期タスク（実行優先度 16）は、形 NX102 および形 NX1P2 CPU ユニットでは使用できません。

参考

多重起動については、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44）」を参照してください。

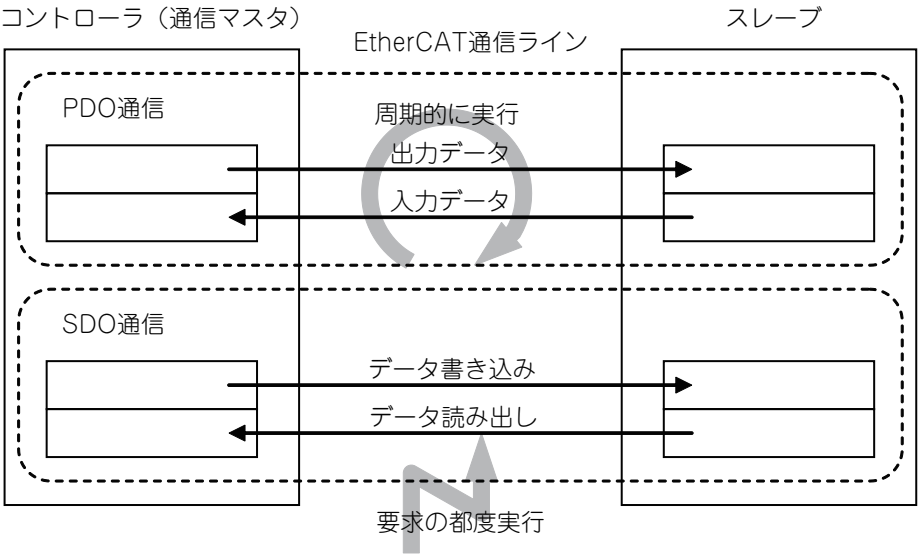
2-4 EtherCAT 通信との関係

MC 機能モジュールは、CPU ユニット内蔵の EtherCAT マスタ機能モジュールの PDO 通信を通じてサーボドライバ、カウンタ、および NX シリーズ 位置インタフェースユニットを制御します。
ここでは、EtherCAT 通信について MC 機能モジュールに関係する項目について説明しています。

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

MC 機能モジュールは、EtherCAT 上のスレーブとの情報交換に関して、CAN application protocol over EtherCAT (CoE)プロトコルを使用しています。
CoE では、各種スレーブが持つパラメータや制御情報は、オブジェクトディクショナリ(OD)と呼ばれるデータ仕様により規定されます。
これらのデータをコントローラ／通信マスタとスレーブ間で情報伝達するために、定周期でリアルタイムの情報交換を行うプロセスデータオブジェクト(Process Data Objects: PDO)と、任意のタイミングで情報伝達を行うサービスデータオブジェクト(Service Data Objects: SDO)の 2 つの手段があります。

MC 機能モジュールでは、サーボモータの位置制御など、一定の制御周期で入出力データ更新を行う指令に対しては PDO 通信を使用しています。
それとは異なり、パラメータ転送など指定されたタイミングでデータの読み出しや書き込みを行う指令には SDO 通信を使用しています。



2-4-2 EtherCAT マスタ機能とモーション制御機能モジュールの関係

NJ/NX シリーズ CPU ユニットでは、EtherCAT スレーブを接続してシーケンス制御やモーション制御を実行することができます。

シーケンス制御

- Sysmac Studio の EtherCAT 構成編集画面で EtherCAT 構成を作成すると、構成スレーブの「I/O ポート」が自動生成されます。
- Sysmac Studio の I/O マップでデバイス変数を割り付けます。

- モーション制御命令以外の命令でシーケンス制御を実行します。

モーション制御

- Sysmac Studio の EtherCAT 構成編集画面で EtherCAT 構成を作成すると、構成スレーブの「I/O ポート」が自動生成されます。
- モーション制御設定ビューで軸変数を作成し、モーション制御の対象とする EtherCAT スレーブを割り付けます。
- モーション制御命令でモーション制御を実行します。

軸変数に割り付けることができる機器は、EtherCAT スレーブのサーボドライバとエンコーダ入力ターミナル、および NX シリーズ 位置インタフェースユニットです。

参考

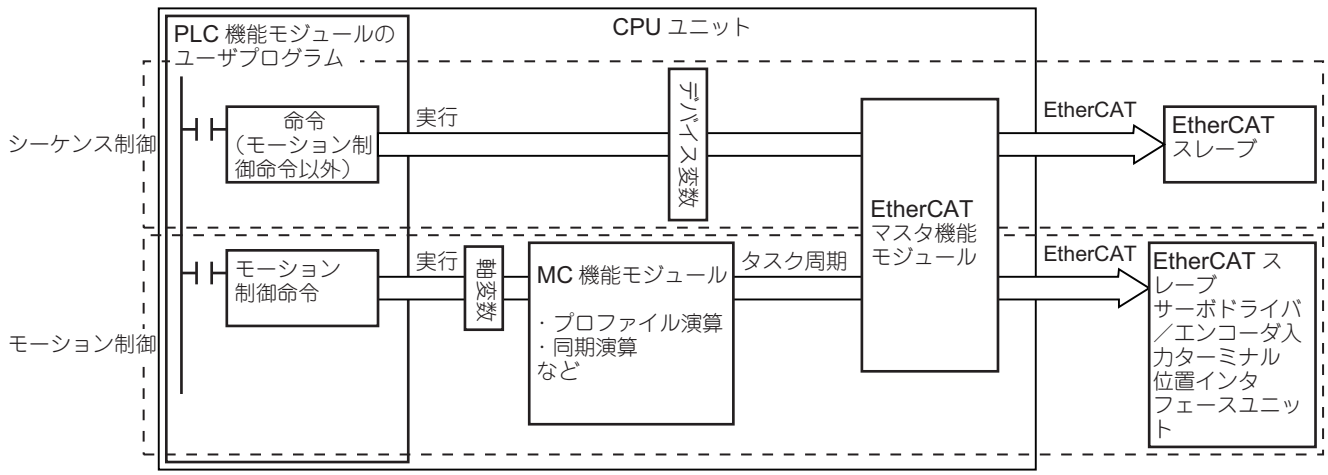
- モーション制御命令以外の命令では、軸変数に割り付けられた EtherCAT スレーブや NX シリーズ 位置インタフェースユニットに対して PDO 通信で直接的に指令することはできません。ただし、軸変数を通じて間接的にステータスを参照することはできます。
- SDO 通信を使用することで、軸変数に割り付けられた EtherCAT スレーブや NX シリーズ 位置インタフェースユニットのオブジェクトも読み書きすることができます。
  ただし、PDO 通信にマッピングされているオブジェクトについては、SDO 通信で書き込まないでください。書き込んだ場合の動作は、スレーブの仕様に依存します。オムロン製スレーブの場合、SDO 通信の異常となります。
- EtherCAT スレーブのサーボドライバやエンコーダ入力ターミナルおよび NX シリーズ 位置インタフェースユニットを軸変数に割り付けなかった場合は、汎用 EtherCAT スレーブと同様にシーケンス制御を実行してください。

バージョン関連情報

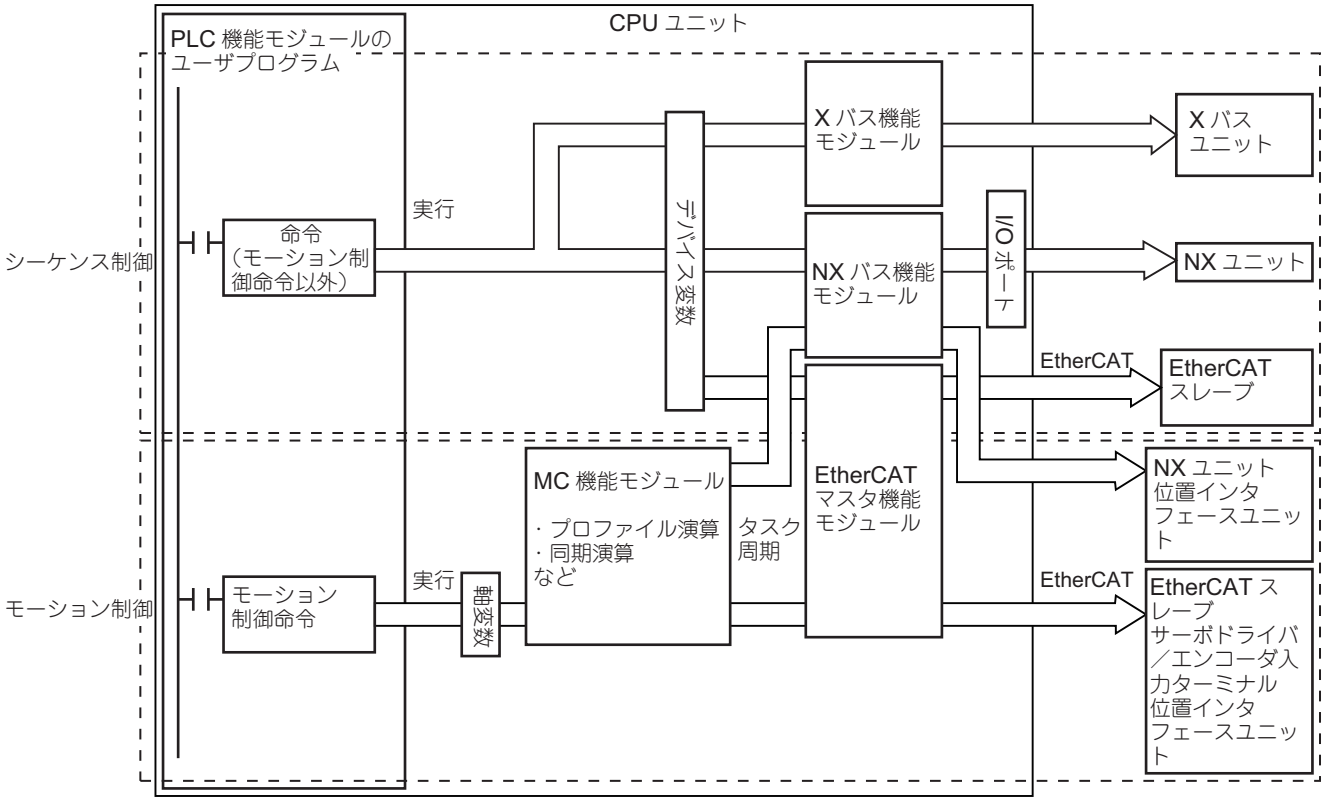
Sysmac Studio Ver.1.09 以降を使用することで、軸変数に割り付けたスレーブやユニットの I/O ポートにデバイス変数を割り付け可能となります。
割り付け可能な I/O ポートの条件は以下です。
- I/O ポートの Read/Write 属性が Read 属性（R: 読出専用）のポート
- I/O ポートの Read/Write 属性が Write 属性（W: 書込専用）であり、Sysmac Studio の［軸基本設定］画面の［詳細設定］でプロセスデータを［<割付なし>］としたポート

Sysmac Studio Ver.1.09 以降を使用し、軸変数にデバイス変数を割り付けたプロジェクトを、Sysmac Studio Ver.1.08 以前で開いた場合、デバイス変数の割り付けが解除されます。

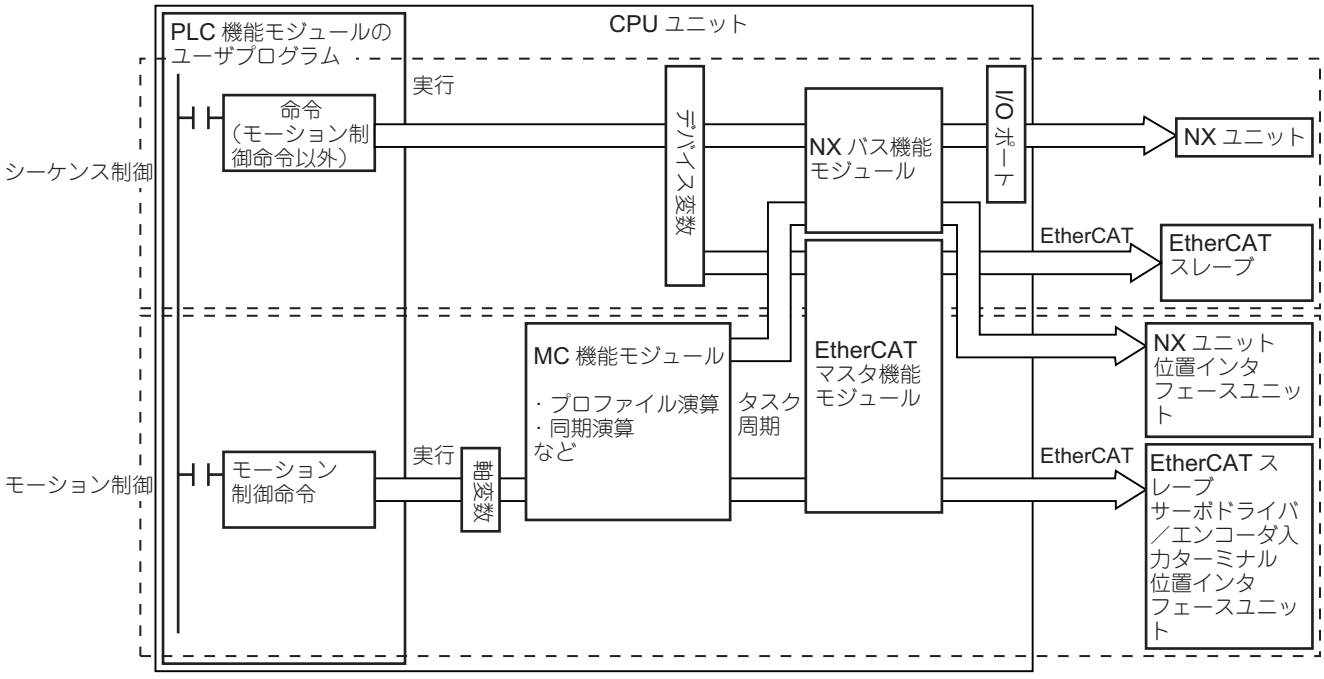
● 形 NX701 CPU ユニットの場合



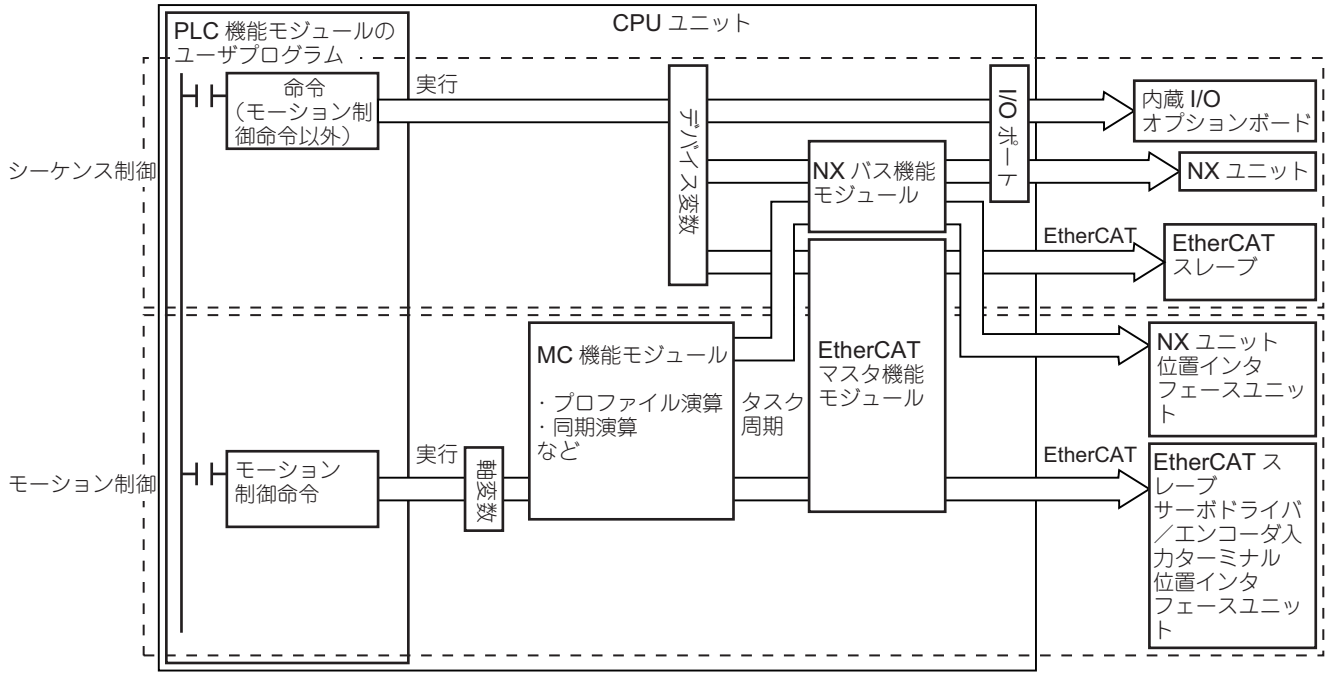
● 形 NX502 CPU ユニットの場合



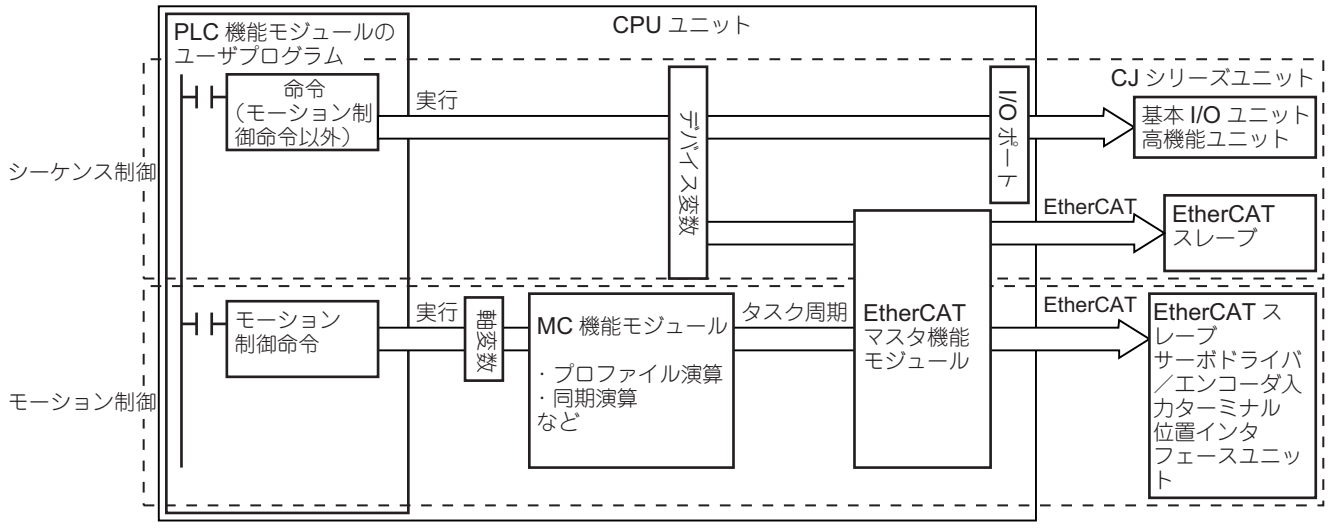
● 形 NX102 CPU ユニットの場合



● **形 NX1P2 CPU ユニットの場合**



● **NJ シリーズ CPU ユニットの場合**



2-4-3　プロセスデータ通信周期とモーション制御周期との関係

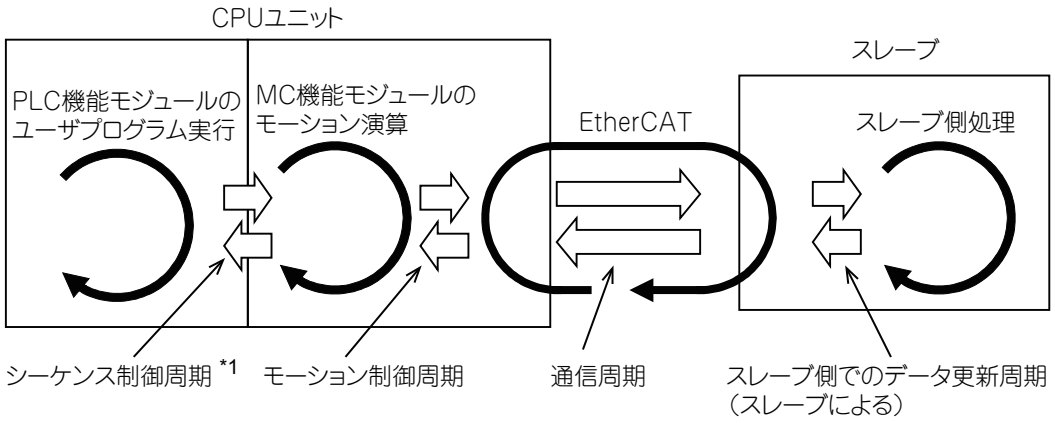
PLC 基本機能モジュールは、ユーザプログラム上のモーション制御命令を起動することによって、モーション制御の指令をモーション制御機能モジュールに対して発行します。それをもとにモーション制御機能モジュールはモーション演算を行い、演算結果を EtherCAT 上のサーボドライバなどに対して指令します。

これらのデータ交換は、以下のような周期で更新されます。

● **形 NX701 CPU ユニットの場合**

・「プライマリ周期」＝「モーション制御 1 の周期」 ＝「EtherCAT 通信のプロセスデータ通信周期 1」

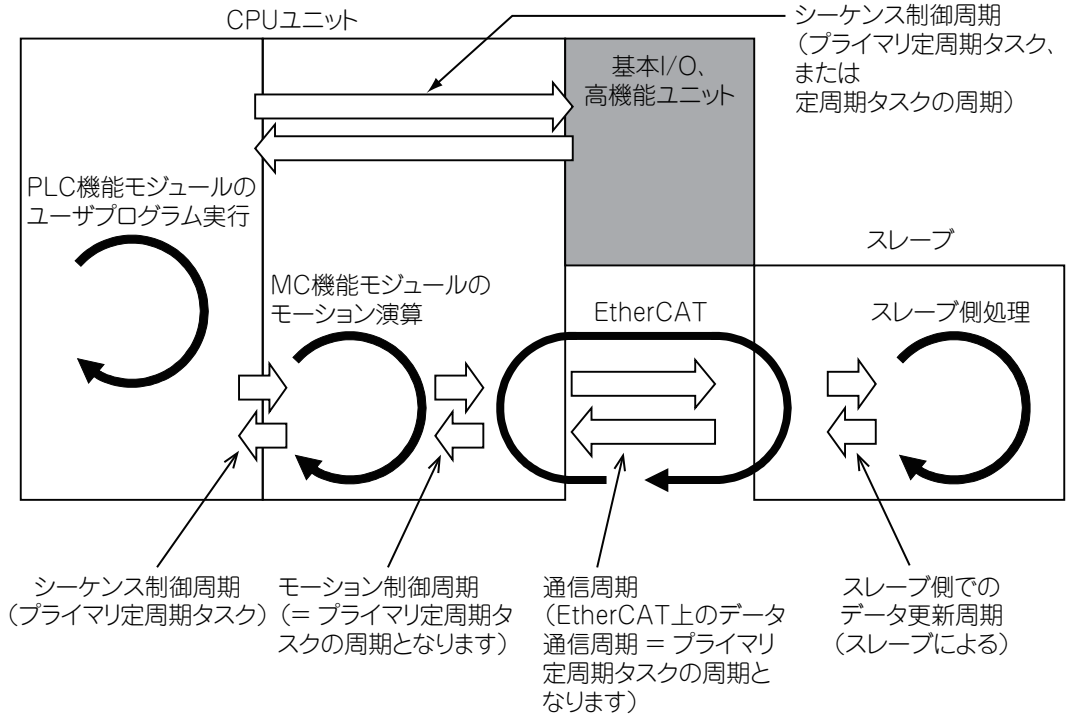
・「定周期タスク（実行優先度 5）のタスク周期」＝「モーション制御 2 の周期」＝「EtherCAT 通信のプロセスデータ通信周期 2」



*1. シーケンス制御周期がプライマリ周期の場合、モーション制御周期、および通信周期はプライマリ周期となります。
シーケンス制御周期が定周期タスク（実行優先度 5）のタスク周期の場合は、モーション制御周期、および通信周期は、定周期タスク（実行優先度 5）のタスク周期となります。

● 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 ユニットの場合

・「プライマリ周期」＝「モーション制御の周期」＝「EtherCAT 通信のプロセスデータ通信周期」



3

軸と軸グループの設定

この章では、軸と軸グループの概要説明、Sysmac Studio の MC 試運転機能で動作させるために必要な軸の設定項目、ならびに Sysmac Studio による作成と設定の手順について説明します。

3-1 軸 3-2
3-1-1 軸の概要 3-2
3-1-2 軸パラメータの概要 3-3
3-1-3 軸変数の概要 3-6
3-1-4 軸変数の同期 3-8
3-1-5 ユーザプログラムでの軸の指定方法 3-8

3-2 軸の設定手順 3-9
3-2-1 軸の設定フロー 3-9
3-2-2 設定手順 3-9

3-3 軸グループ 3-21
3-3-1 軸グループの概要 3-21
3-3-2 軸グループパラメータの概要 3-21
3-3-3 軸グループ変数の概要 3-22
3-3-4 ユーザプログラムでの軸グループの指定方法 3-24

3-4 軸グループの設定手順 3-25
3-4-1 軸グループの設定フロー 3-25
3-4-2 設定手順 3-25

3-1 軸

ここでは、MC 機能モジュールの軸について説明します。

3-1-1 軸の概要

モーション制御システムでは、モーション制御の対象を「軸」と呼びます。
軸には EtherCAT で接続する実際のサーボドライバやエンコーダなど、および MC 機能モジュール内部の仮想的なサーボドライバやエンコーダがあります。

また、形 NX701 CPU ユニット、形 NX502 CPU ユニット、および NJ シリーズ CPU ユニットでは、すべての「軸」がモーション制御軸ですが、形 NX102 および形 NX1P2 CPU ユニットでは、モーション制御軸に加え、「単軸位置制御軸」を追加しました。「単軸位置制御軸」への切り替えは、軸基本設定の［制御機能］で行います。
［制御機能］の詳細は、「制御機能 (P.5-9)」を参照してください。

MC 機能モジュールの軸には、次の表の軸種別があります。

軸種別	内容
サーボ軸	EtherCAT スレーブサーボドライバや NX シリーズ 位置インタフェースユニット *1 を使用する軸です。実際のサーボドライバなどと割り付けをおこなって使用します。 1 台のサーボモータを 1 軸として使用します。 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、パルス出力ユニットとデジタル入力ユニットなど、複数のデバイスを 1 軸に割り付けできます。
仮想サーボ軸	MC 機能モジュール内の仮想的な軸です。実際のサーボドライバは使用しません。同期制御の主軸などとして使用します。
エンコーダ軸	EtherCAT スレーブエンコーダ入力ターミナルや NX シリーズ 位置インタフェースユニット *1 を使用する軸です。実際のエンコーダ入力ターミナルなどと割り付けをおこなって使用します。 1 台のエンコーダ入力ターミナルに 2 個のカウンタがある場合は、各々 1 個のカウンタを 1 軸として使用します。
仮想エンコーダ軸	エンコーダの動作を仮想的に行う軸です。エンコーダの実物がない場合など、エンコーダ軸の代わりとして一時的に使用します。 *2

*1. 制御対象機器は、「1-4-3 機能仕様 (P.1-10)」を参照してください。
*2. 仮想エンコーダ軸の現在位置を更新するモーション制御命令と組み合わせて使用します。
この命令に対応していないバージョンの MC 機能モジュールでは、カウント機能を使用できません。

MC 機能モジュールの軸には、以下の関連要素があります。
これらは各形式の制御軸最大数分あります。制御軸最大数は、形式により異なります。
詳細は、「1-4-2 性能仕様 (P.1-7)」を参照してください。

構成要素	内容	参照ページ
軸パラメータ	MC 機能モジュールで制御する各軸に関して、最高速度やジョグ送り、原点復帰といった軸動作の設定を行います。 軸パラメータを設定するには、Sysmac Studio を使用します。	P.3-3
軸変数	軸変数は MC 機能モジュールで制御する各軸について、現在位置や異常情報などのモニタ情報からなるシステム定義変数です。 Sysmac Studio のマルチビューエクスプローラで軸を追加すると軸変数が作成されます。軸変数の名前（以降は、軸変数名と呼びます）は、このビューで設定することができます。	P.3-6

構成要素	内容	参照ページ
ユーザプログラムでの軸の指定方法	ユーザプログラムでは、モーション制御命令を使用してモーション制御を行います。単軸制御を行うモーション制御命令を「軸指令」と呼びます。 軸指令でどの軸を制御するかは、命令の入出力変数「Axis（軸）」に、システム定義変数の軸変数名、あるいは Sysmac Studio で設定した軸変数名を指定します。	P.3-8

3-1-2 軸パラメータの概要

● 軸パラメータの一覧

区分	パラメータ名称
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
	ワーク側ギア比 *3
	モータ側ギア比 *3
動作設定	最高速度
	起動速度 *4
	ジョグ最高速度
	最大加速度
	最大減速度
	加減速オーバー
	反転時動作
	速度警告値
	加速度警告値
	減速度警告値
	正方向トルク警告値 *5
	負方向トルク警告値 *5
	フィードバック速度フィルタ時定数
	インポジション幅
	インポジションチェック時間
	原点位置範囲
拡張動作設定	即停止入力停止方法
	限界入力停止方法
	ドライバエラーリセット監視時間
	正方向トルク制限上限値
	負方向トルク制限上限値
	即停止入力論理反転 *4
	正方向限界入力論理反転 *4
	負方向限界入力論理反転 *4

区分	パラメータ名称
	原点近傍入力論理反転 *4
リミット設定	ソフトウェアリミット機能
	正方向ソフトウェアリミット
	負方向ソフトウェアリミット
	位置偏差オーバー値
	位置偏差警告値
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
原点復帰設定	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

*1. 形 NX701 CPU ユニットの場合に設定します。
*2. 形 NX102 および形 NX1P2 CPU ユニットの場合に設定します。
*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することができます。
*4. 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することができます。
*5. トルク制御時にだけ有効です。

軸パラメータの詳細については、「5-2 軸パラメータ (P.5-4)」を参照してください。

● 軸を使用するための必須設定項目

Sysmac Studio で作成した軸について次の項目を設定することで、軸を使用することができます。

区分	パラメータ名称	設定内容	参照ページ
軸基本設定	軸番号	軸を作成した順に自動で設定されます。	P.5-6
	モーション制御 *1	［プライマリ定周期タスク］を選択します。	
	軸使用	［使用軸］を選択します。	
	軸種別	制御する軸の種別を選択します。	
	制御機能 *2	［すべて］を選択します。 *3	

区分	パラメータ名称	設定内容	参照ページ
	入力デバイス／出力デバイス	軸に割り付ける EtherCAT スレーブデバイスのノードアドレスを選択します。 軸種別が［仮想軸］の場合は選択できません。	

*1. 形 NX701 CPU ユニットの場合に設定します。
*2. 形 NX102 および形 NX1P2 CPU ユニットの場合に設定します。
*3. モーション制御軸として使用する場合は、［すべて］を選択します。単軸位置制御軸として使用する場合は、［単軸位置制御のみ］を選択してください。

● Sysmac Studio でサーボドライバを試運転するための必須設定項目

Sysmac Studio の MC 試運転機能で、EtherCAT 接続したサーボドライバなどを動作するには、次の項目を設定します。

区分	パラメータ名称	設定内容	参照ページ
軸基本設定	軸番号	軸を作成した順に自動で設定されます。	P.5-6
	モーション制御 *1	［プライマリ定周期タスク］を選択します。	
	軸使用	［使用軸］を選択します。	
	軸種別	［サーボ軸］を選択します。	
	制御機能 *2	［すべて］を選択します。*3	
	入力デバイス／出力デバイス	軸に割り付ける EtherCAT スレーブデバイスのノードアドレスを選択します。 軸種別が［仮想軸］の場合は選択できません。	
単位変換設定	表示単位	[mm]、[degree]など、表示する単位を選択します。	P.5-11
	モータ 1 回転のパルス数	エンコーダ分解能にあわせて、モータ 1 回転あたりのパルス数を設定します。*4	
	モータ 1 回転の移動量	機械の仕様にあわせて、モータ 1 回転あたりのワークの移動量を設定します。	
	減速機使用 *5	減速機設定を使用するかを指定します。	
	ワーク側 1 回転の移動量 *5	ワーク側が 1 回転した場合の移動量を設定します。	
	ワーク側ギア比 *5	ワーク側のギア比を設定します。	
	モータ側ギア比 *5	モータ側のギア比を設定します。	
位置カウント設定	カウントモード	機械の仕様にあわせて選択します。	P.5-23
リミット設定	ソフトウェアリミット機能	装置の仕様にあわせて選択します。	P.5-23

*1. 形 NX701 CPU ユニットの場合に設定します。
*2. 形 NX102 および形 NX1P2 CPU ユニットの場合に設定します。
*3. モーション制御軸として使用する場合は、［すべて］を選択します。単軸位置制御軸として使用する場合は、［単軸位置制御のみ］を選択してください。
*4. たとえば、エンコーダ分解能が 10,000 パルス／回転の場合は、「10,000」を設定してください。
*5. 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することができます。

使用上の注意

- 実際にモータを動作させる場合、最高速度や最大加減速度および停止設定などは、機械の動作条件に従って適切な値を設定してください。
- オムロン製サーボドライバ 1S シリーズおよび G5 シリーズは、ノードアドレスをドライバ前面のロータリスイッチにより固定値を設定することができます。
  このロータリスイッチを「00」と設定した場合、ノードアドレスは Sysmac Studio の EtherCAT エディタで設定した値に従います。
  接続されるすべてのドライバでロータリスイッチを「00」と設定した場合、ドライバの接続位置を変更しても異常とはなりません。装置の各制御対象に対して特定のサーボドライバを割り付ける場合は、ドライバのノードアドレスをロータリスイッチにより設定してください。

3-1-3 軸変数の概要

軸変数は、MC 機能モジュールの軸パラメータの一部と現在位置や異常情報などのモニタ情報からなるシステム定義変数です。
Sysmac Studio で軸を作成すると、作成した順番に軸変数が変数テーブルに登録されます。
軸変数のデータ型は構造体型で、データ型名は「_sAXIS_REF」です。

軸変数名

MC 機能モジュールの軸変数名には、システム定義変数の軸変数名と、Sysmac Studio で軸を追加したときの軸変数名の 2 種類があります。
システム定義変数の軸変数名は、「_MC_AX[0]～_MC_AX[255]」、「_MC1_AX[0]～_MC1_AX[255]」、「_MC2_AX[0]～_MC2_AX[255]」です。
Sysmac Studio で軸を追加すると、「MC_Axis000～MC_Axis255」というデフォルト名称が、「_MC_AX[0]～_MC_AX[255]」に対して設定されます。
番号は追加した順番に前詰めです。これらの軸変数名は、Sysmac Studio で軸ごとに任意の変数名に変更することができます。
ユーザプログラムでは、システム定義変数の軸変数名でも、Sysmac Studio で追加した軸変数名のどちらも使用することができます。

● _MC_AX[0-255]を使用した場合の例

システム定義変数の軸変数名（グローバル変数ビューの割付先 *1）	Sysmac Studio で追加した軸変数名のデフォルト	軸番号例
_MC_AX[0]	MC_Axis000	軸 0
_MC_AX[1]	MC_Axis001	軸 1
:	:	:
_MC_AX[255]	MC_Axis255	軸 255

*1. Sysmac Studio のグローバル変数ビューにある割付先欄の名称を変更すると異常となります。

参考

- 形 NX701 CPU ユニットの場合は、「_MC_AX[0-255]」または「_MC1_AX[0-255]」と、「_MC2_AX[0-255]」が使用できます。
  形 NX502 CPU ユニットの場合は、「_MC_AX[0-127]」だけが使用できます。
  形 NX102 CPU ユニットの場合は、「_MC_AX[0-14]」だけが使用できます。
  形 NX1P2 CPU ユニットの場合は、「_MC_AX[0-11]」だけが使用できます。
  NJ シリーズ CPU ユニットの場合は、「_MC_AX[0-63]」だけが使用できます。
- 形 NX701 CPU ユニットの場合「_MC_AX[0-255]」、と「_MC1_AX[0-255]」は、軸番号が同じであれば同じ値を参照します。どちらの軸変数でも使用可能ですし、混在させても使用可能です。
- プライマリ定周期タスクに割り付けた軸変数は、「_MC_AX[0-255]」、または「_MC1_AX[0-255]」となります。
  定周期タスク（実行優先度 5）に割り付けた軸変数は、「_MC2_AX[0-255]」となります。

軸変数の階層と軸変数名の変更例

_MC_AX[0]を例に説明します。ほかの軸変数も同様となります。

```
_MC_AX[0]                          : 軸変数
    _MC_AX[0].Status               : 「軸ステータス」を表す階層
        _MC_AX[0].Status.Ready     : 「軸起動準備完了」を表す変数
        _MC_AX[0].Status.Disabled  : 「軸無効」を表す変数
            :
    _MC_AX[0].Details              : 「軸制御ステータス」を表す階層
        _MC_AX[0].Details.Idle     : 「停止中」を表す変数
        _MC_AX[0].Details.InPosWaiting : 「インポジション待ち」を表す変数
            :
    _MC_AX[0].Cmd                  : 「軸指令値」を表す階層
        _MC_AX[0].Cmd.Pos          : 「指令現在位置」を表す変数
        _MC_AX[0].Cmd.Vel          : 「指令現在速度」を表す変数
        _MC_AX[0].Cmd.AccDec       : 軸モニタの「指令現在加減速度」を表す変数
            :
    _MC_AX[0].Act                  : 「軸現在値」を表す階層
        _MC_AX[0].Act.Pos          : 「フィードバック現在位置」を表す変数
        _MC_AX[0].Act.Vel          : 「フィードバック現在速度」を表す変数
            :
    _MC_AX[0].Cfg                  : 「軸基本設定」を表す階層
        _MC_AX[0].Cfg.AxNo         : 「軸番号」を表す変数
        _MC_AX[0].Cfg.AxEnable     : 「軸使用」を表す変数
        _MC_AX[0].Cfg.AxType       : 「軸種別」を表す変数
            :
        _MC_AX[0].Scale.Units      : 「表示単位」を表す変数
_MC_AX[1]                          : 軸変数
            :
```

● **例**

「MC_Axis000」を「MyAxis1」に変更した場合、「フィードバック現在位置」を表す変数は、「MyAxis1.Act.Pos」と、「_MC_AX[0].Act.Pos」のどちらでも扱うことができます。

軸変数の詳細については、「軸変数 (P.6-22)」を参照してください。

3-1-4 軸変数の同期

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。または、その逆に定周期タスク（実行優先度 5）で制御する軸の、軸変数の値をプライマリ定周期タスクのユーザプログラムで参照することもできます。
また、プライマリ定周期タスクで制御する軸変数の値を定周期タスク（実行優先度 16）のユーザプログラムで参照することもできます。
参照するタスクの種類、ならびに軸変数の関係は、次のようになります。

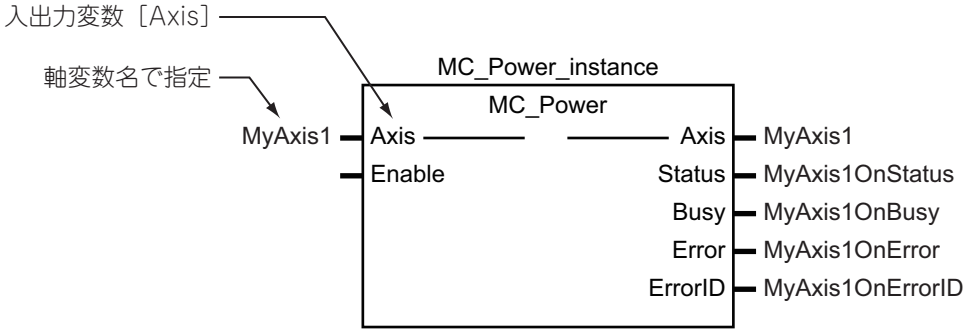
○: 参照可 ×: 参照不可

タスクの種類	モーション制御 1 の軸変数		モーション制御 2 の軸変数	
	ユーザ定義 *1	システム定義 *2	ユーザ定義 *1	システム定義 *3
プライマリ定周期タスク	○	○	○ *4	×
定周期タスク（実行優先度 5）	○ *5	×	○	○
定周期タスク（実行優先度 16）	○ *5	○ *6	○ *4	×
定周期タスク（実行優先度 17, 18）	○ *5	○ *7	○ *4	○ *8
イベントタスク（8, 48）	○ *5	○ *7	○ *4	○ *8

*1. Sysmac Studio で「軸」を作成したときに自動生成されるユーザ定義の軸変数「MC_Axis000」や「Loader1」などのことです。ユーザ定義の軸変数を使用することを推奨します。
*2. 「_MC_AX[*]」や「_MC1_AX[*]」のことです。
*3. 「_MC2_AX[*]」のことです。
*4. 定周期タスク（実行優先度 5）のタスク周期ごとに、1 軸分の軸変数を処理します。
*5. プライマリ周期ごとに、1 軸分の軸変数を処理します。
*6. プライマリ周期ごとに、最大軸数分の軸変数を処理します。プライマリ定周期タスクのタスク実行時間が増加するため、推奨しません。
*7. プライマリ周期ごとに、最大軸数分の軸変数を処理します。
*8. 定周期タスク（実行優先度 5）のタスク周期ごとに、最大軸数分の軸変数を処理します。

3-1-5 ユーザプログラムでの軸の指定方法

ユーザプログラムでは、モーション制御命令の入出力変数「Axis」に、軸変数名を指定します。
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。



「MyAxis1」のかわりに、システム定義変数の「_MC_AX[0]」も使用することができます。

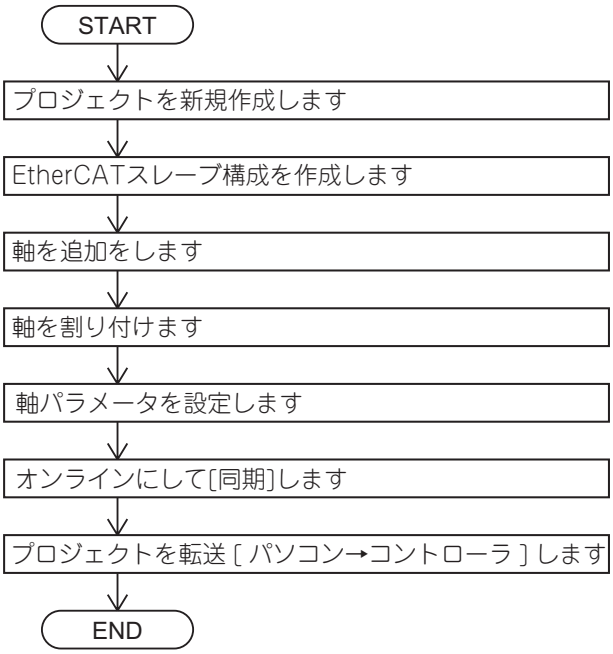
モーション制御命令の詳細については、「6-2 モーション制御命令 (P.6-5)」を参照してください。

各モーション制御命令の詳細機能については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の各命令の説明を参照してください。

3-2 軸の設定手順

ここでは、Sysmac Studio を使用して新規にサーボ軸を設定する手順について説明します。

3-2-1 軸の設定フロー



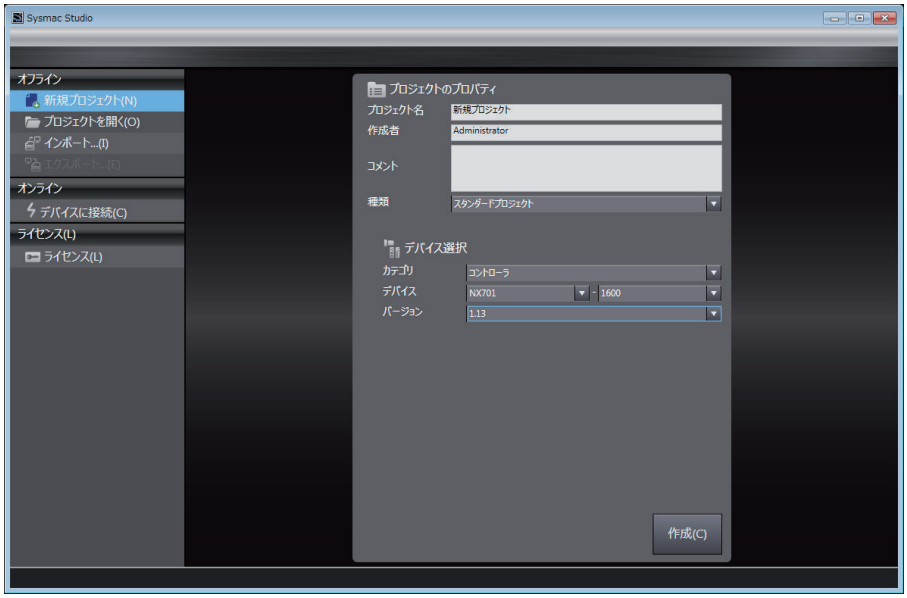
3-2-2 設定手順

ここでは、軸の設定方法について説明します。

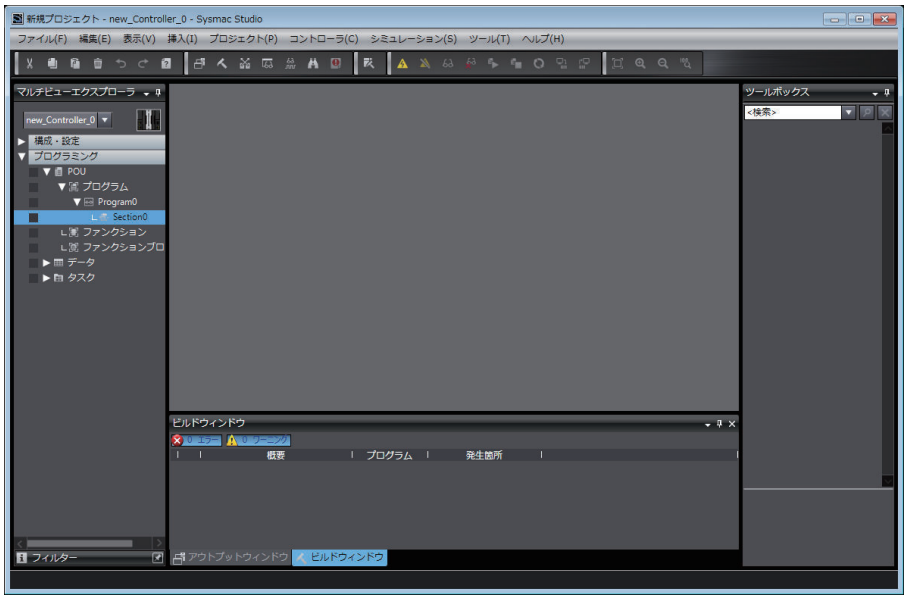
Sysmac Studio の起動

1 **Sysmac Studio を起動し、[新規プロジェクト] ボタンをクリックします。**

2 **プロジェクトのプロパティとデバイス選択を設定し、[作成] ボタンをクリックします。**



新規プロジェクト画面が表示されます。



EtherCAT ネットワーク構成の作成

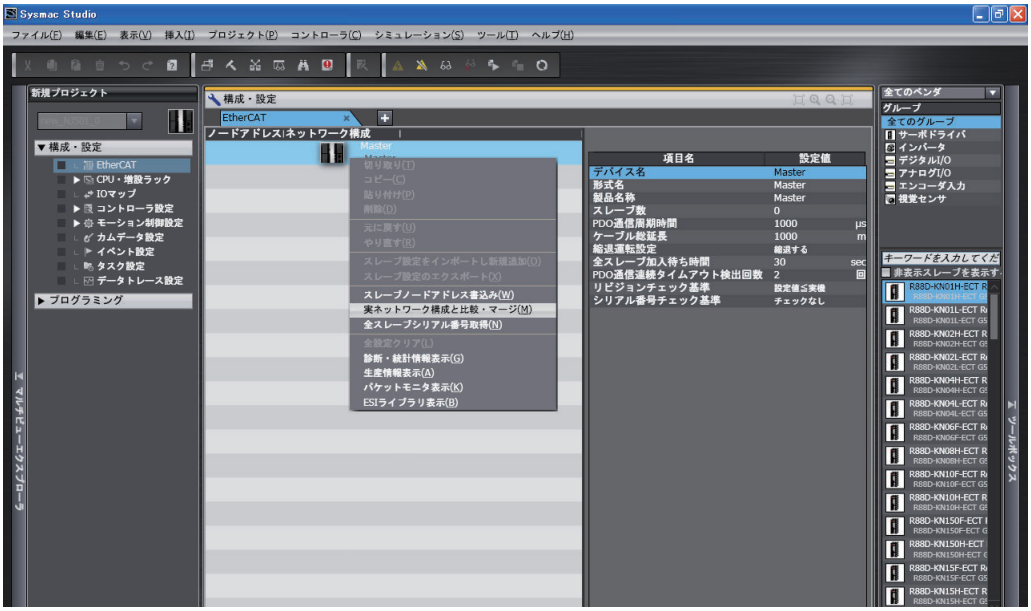
EtherCAT ネットワーク構成を作成する方法は、オンラインで行う方法とオフラインで行う方法の 2 つがあります。

● オンラインによる方法

1 **マルチビューエクスプローラで、[EtherCAT] をダブルクリックします。**
EtherCAT 構成編集画面が表示されます。

2 **[コントローラ] メニューの [オンライン] をクリックしオンラインにします。**

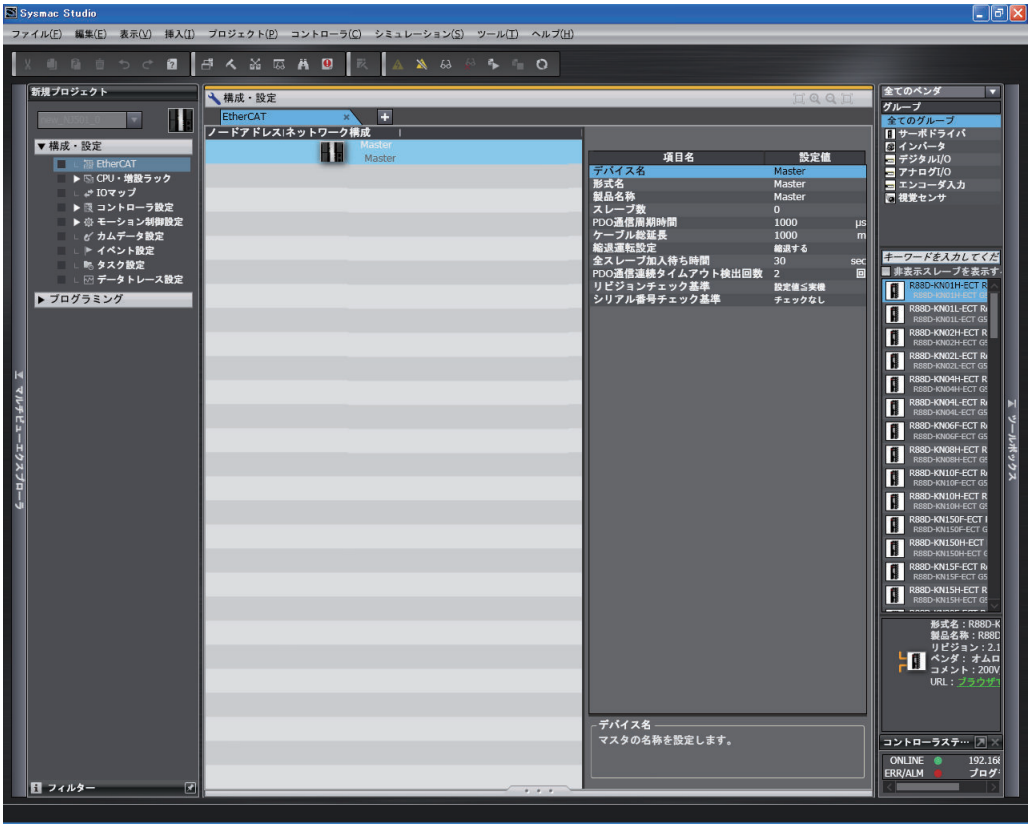
3 **画面内の [Master] 上で右クリックし [実ネットワーク構成と比較・マージ] を選択します。**



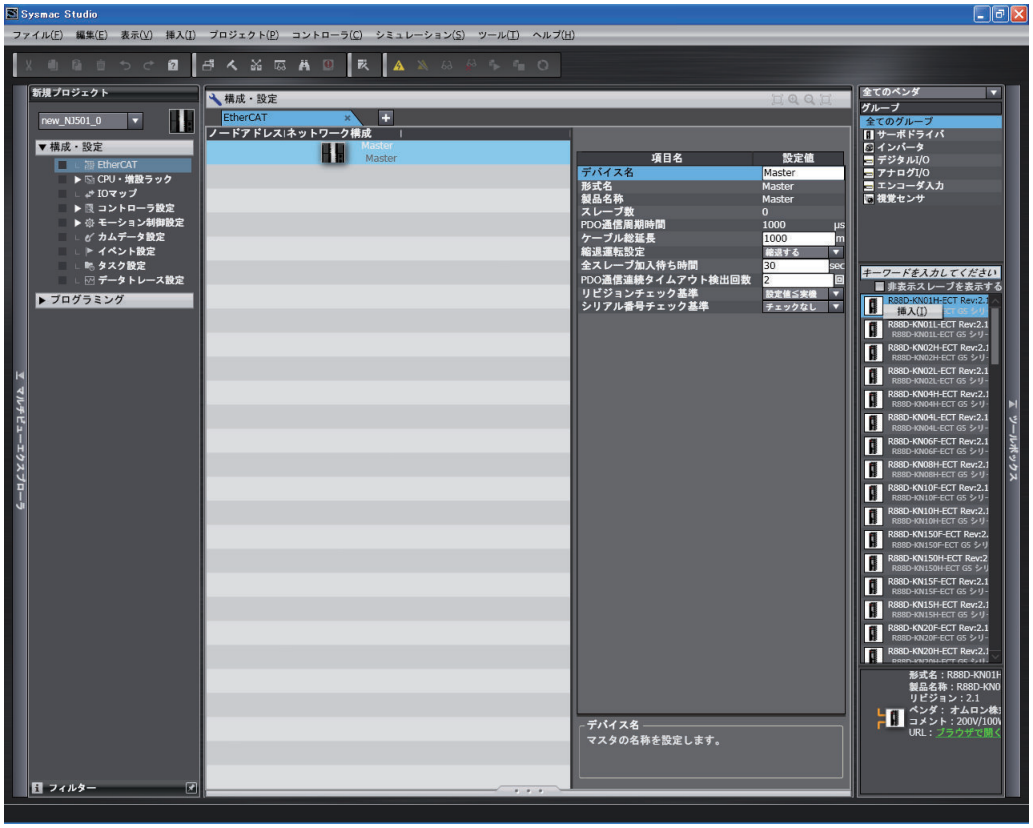
情報取得が終了すると、EtherCAT スレーブの実機構成が表示されます。
表示された実機構成上で右クリックし［実ネットワーク構成を使用］をクリックします。

● オフラインで作成する場合

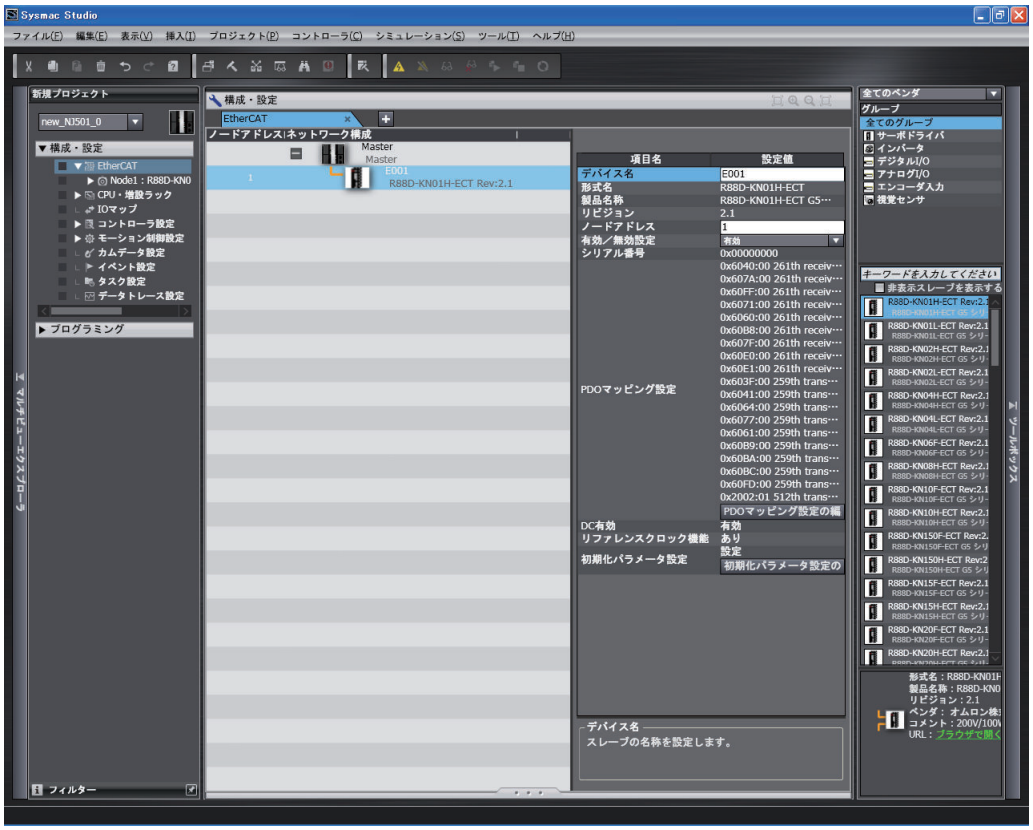
1 **マルチビューエクスプローラで、[EtherCAT] をダブルクリックします。**
EtherCAT 構成編集画面が表示されます。



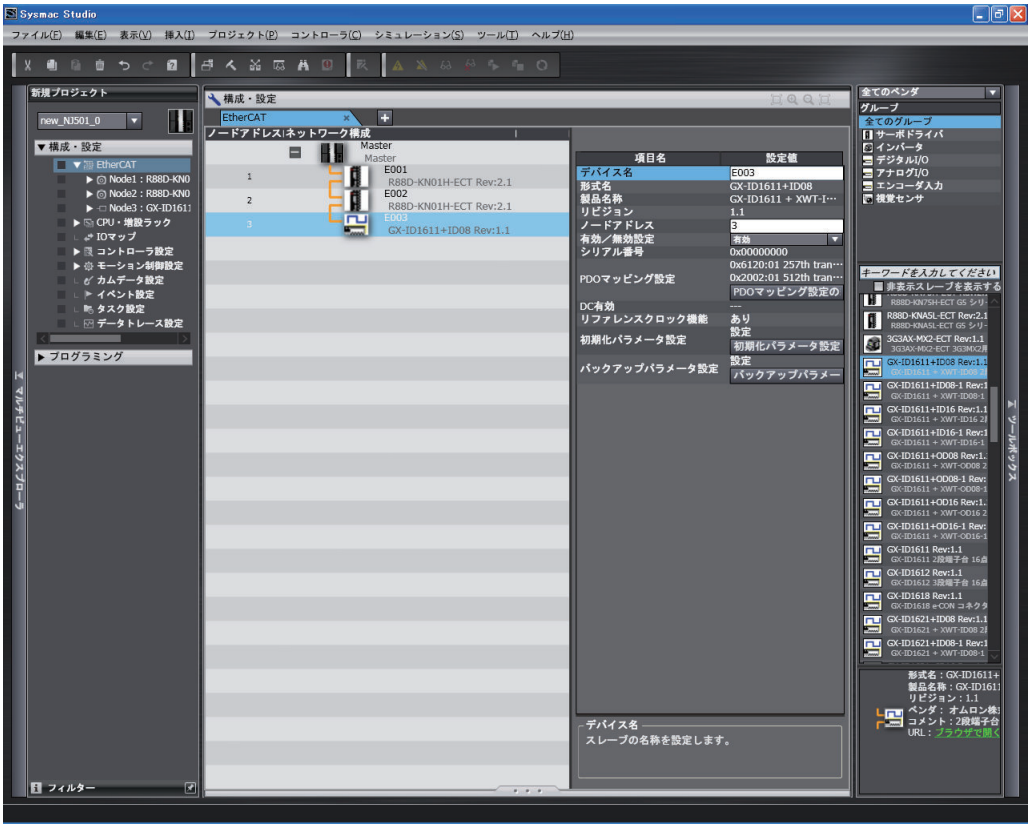
2 **接続するスレーブを右クリックし、[挿入] をクリックします。**



挿入したスレーブが表示されます。



3 **使用するすべてのスレーブを挿入します。**

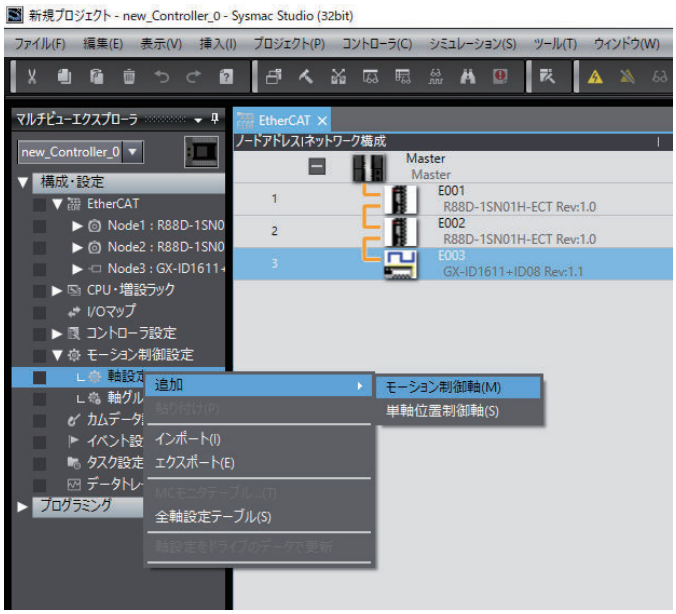


軸の追加

1 **マルチビューエクスプローラで［軸設定］を右クリックし、［追加］から［モーション制御軸］または［単軸位置制御軸］をクリックします。**

参考

- ［単軸位置制御軸］は、形 NX102 および形 NX1P2 CPU ユニットの場合に表示されます。
- ［モーション制御軸］と［単軸位置制御軸］の違いについては、「制御機能 (P.5-9)」を参照してください。



マルチビューエクスプローラに軸が追加されます。 追加された軸の変数名のデフォルトは「MC_Axis000」です。

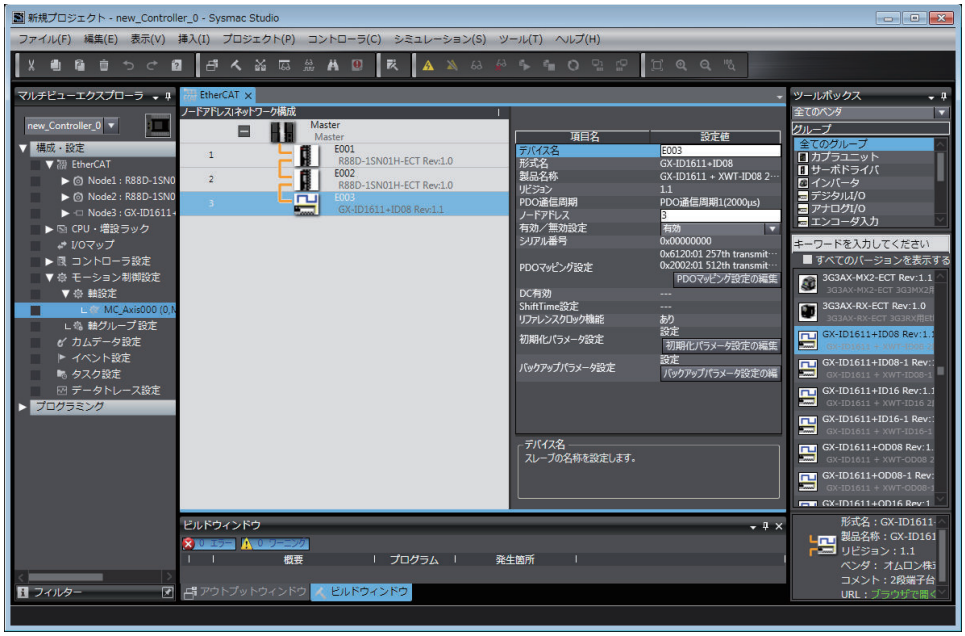
参考

ユーザプログラムで使用している軸変数名の変更

すでに使用している軸変数名を変更する場合、次の 2 つの操作を行ってください。

- 変数宣言内の変数テーブルにある軸変数名の変更。
- ユーザプログラムの軸変数名の変更。

変数テーブルの軸変数名を変更しても、ユーザプログラムの軸変数名には反映されません。変数テーブルで宣言されていない変数名をユーザプログラムで使用すると、異常が発生します。必ず両方とも変更してください。



● インポートによる軸の追加

事前に作成しておいた軸設定ファイルをインポートすることで［軸設定］を追加することもできます。

軸設定ファイルは［軸設定］をエクスポートすることで作成できます。
詳細は、『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。

バージョン関連情報

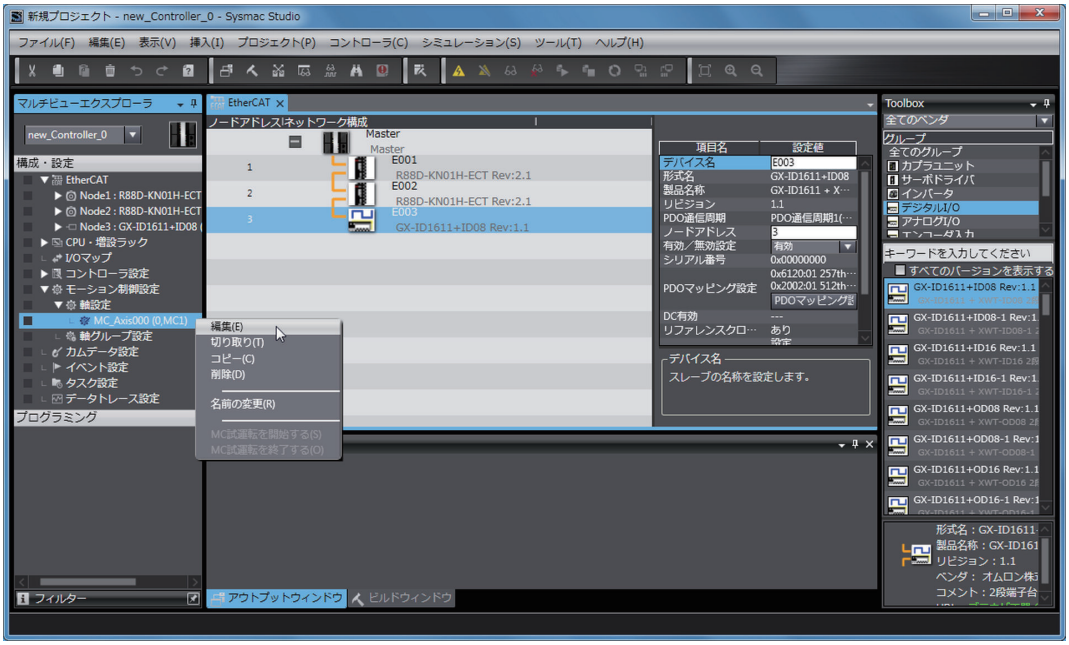
［インポート］および［エクスポート］は、Sysmac Studio Ver.1.57 以降で使用可能です。

- **コピーによる軸の追加**

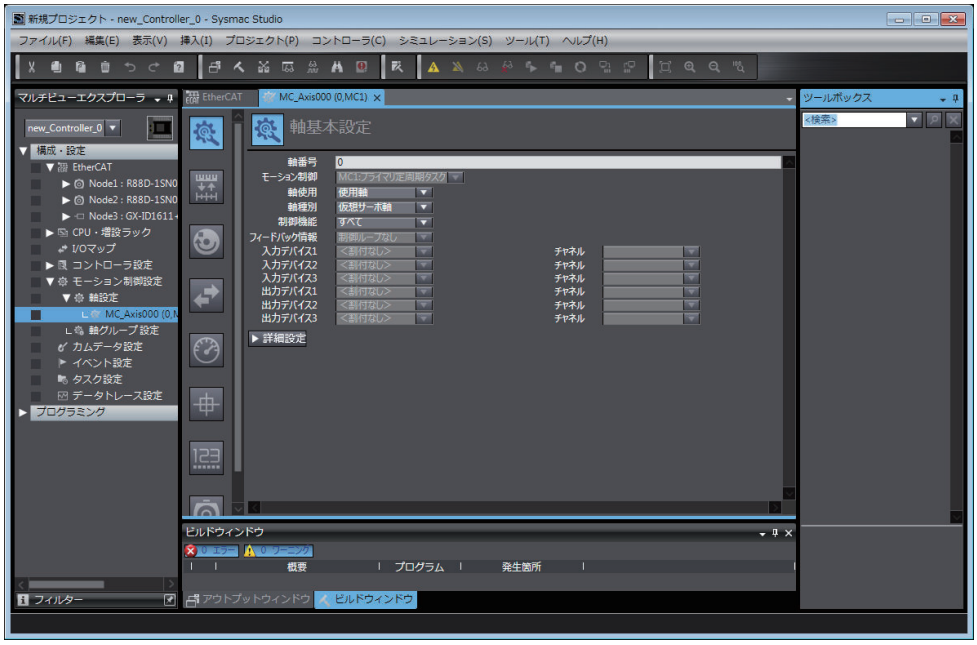
既存の軸設定をコピーして貼り付けることで、軸を追加することもできます。

軸の割付

1 **マルチビューエクスプローラに追加された軸を右クリックし、［編集］をクリックします。**



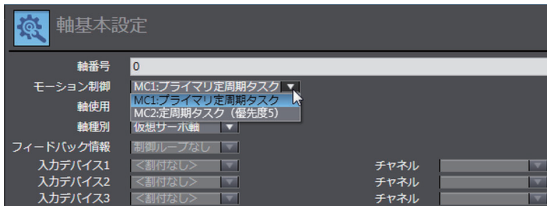
軸パラメータ設定画面の［軸基本設定］画面が表示されます。



参考

［制御機能］は、形 NX102 および形 NX1P2 CPU ユニットの場合に表示されます。

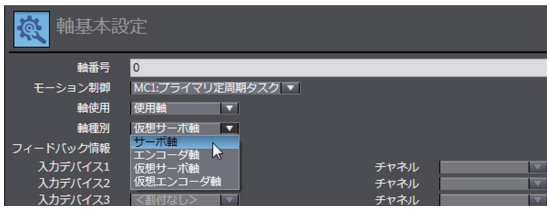
2 **［モーション制御］から［プライマリ定周期タスク］、または［定周期タスク（優先度 5）］を選択します。**



参考

形 NX701 CPU ユニットの場合に設定します。

3 **［軸種別］から［サーボ軸］を選択します。**



4 **［制御機能］から［すべて］を選択します。**

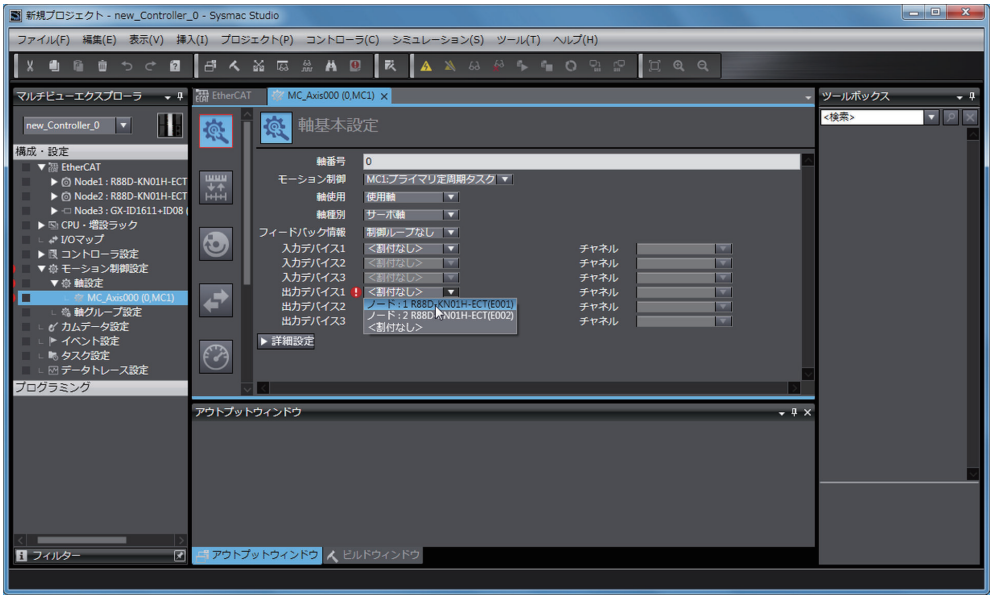


参考

- 形 NX102 および形 NX1P2 CPU ユニットの場合に設定します。
- モーション制御軸として使用する場合は、［すべて］を選択します。単軸位置制御軸として使用する場合は、［単軸位置制御のみ］を選択してください。

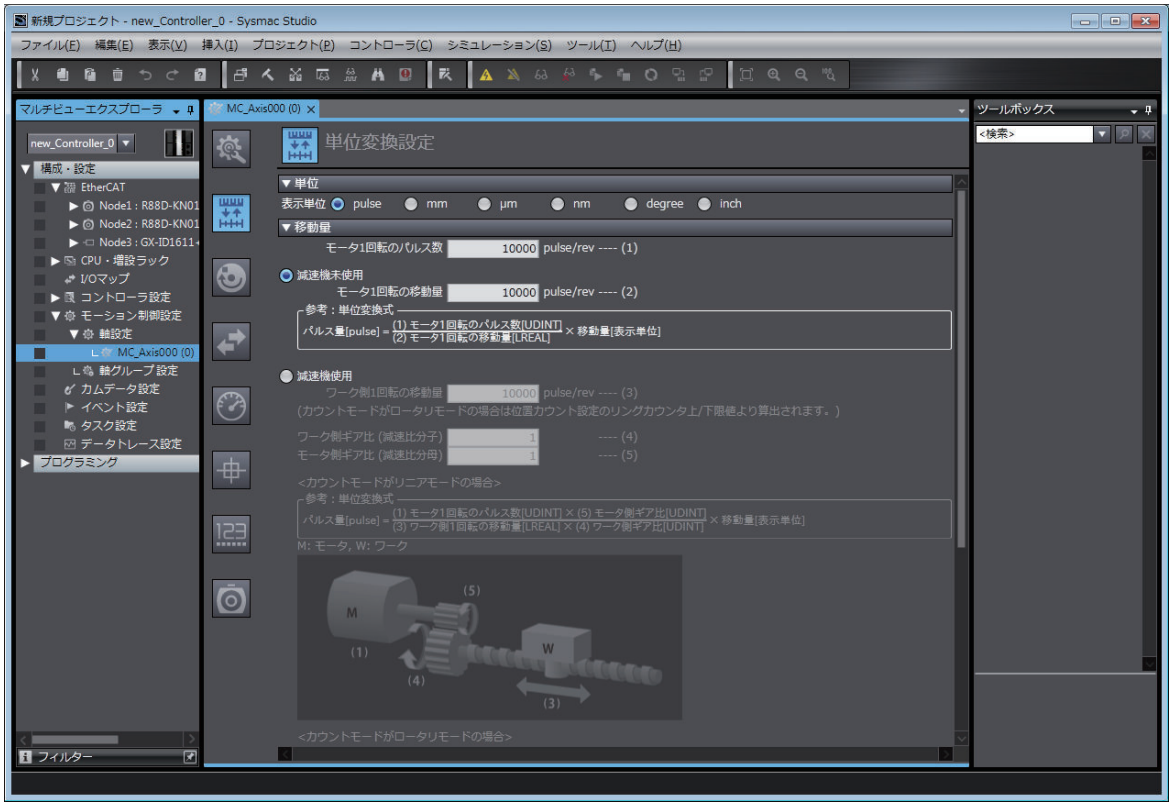
5 **使用するサーボドライバを選択します。**

［出力デバイス 1］に使用するサーボドライバを選択します。
この操作により EtherCAT スレーブのサーボドライバを軸として使用できます。

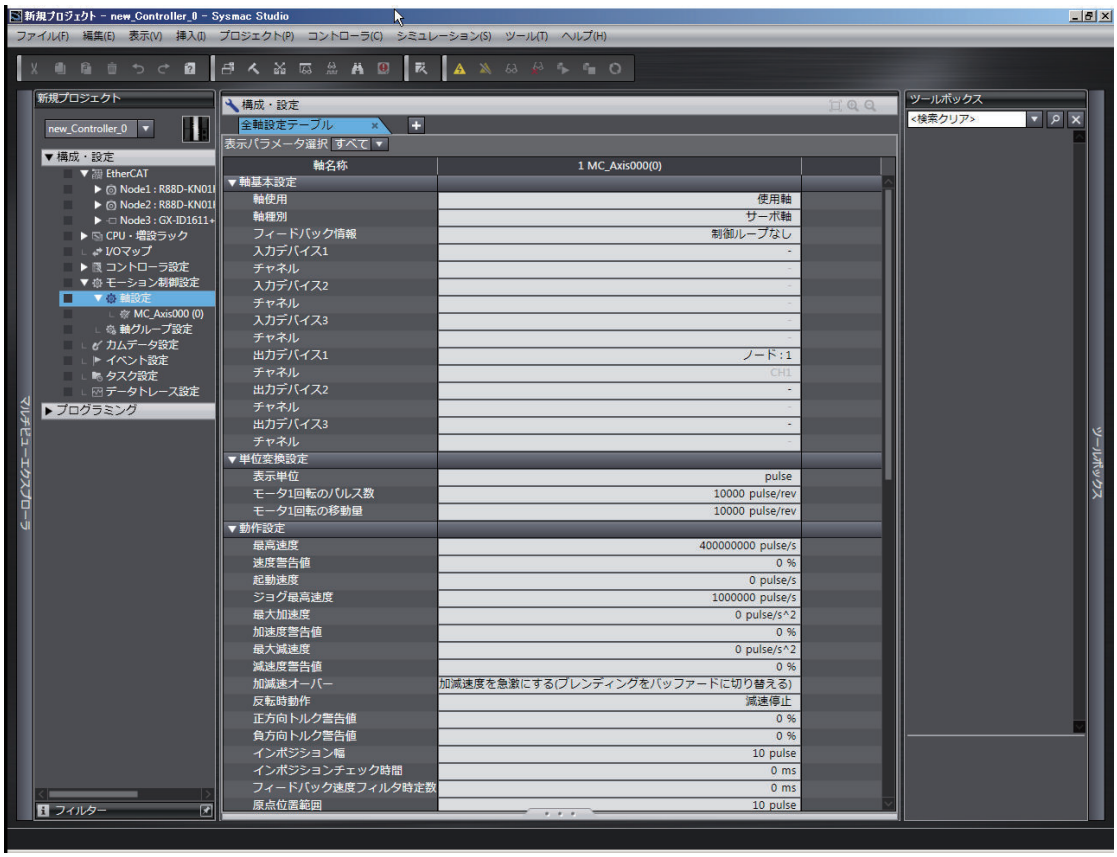


軸パラメータの設定

軸パラメータ設定画面にあるアイコンをクリックします。
アイコンごとに分類された軸パラメータ設定画面が表示されます。



マルチビューエクスプローラの［軸設定］を右クリックし［全軸設定テーブル］をクリックすることにより、全軸の軸パラメータを一覧画面で設定することもできます。



使用上の注意

表示単位や電子ギア（単位変換の式）、および最高速度や最大加減速度などの動作設定は、装置の動作条件に従って適切な値を設定してください。

参考

ユーザプログラムで使用している軸変数名の変更

すでに使用している軸変数名を変更する場合、次の 2 つの操作を行ってください。

- 変数宣言内の変数テーブルにある軸変数名の変更。
- ユーザプログラムの軸変数名の変更。

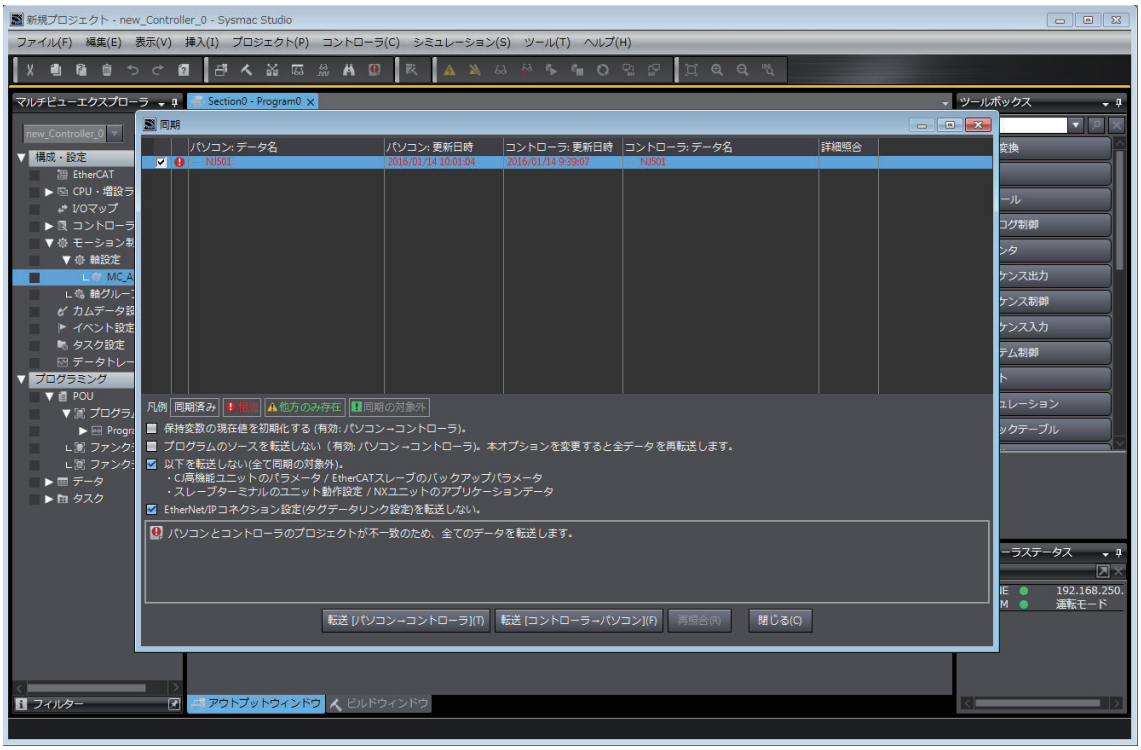
変数テーブルの軸変数名を変更しても、ユーザプログラムの軸変数名には反映されません。
変数テーブルで宣言されていない変数名をユーザプログラムで使用すると、異常が発生します。
必ず両方とも変更してください。

CPU ユニットへのダウンロード

NJ/NX シリーズ CPU ユニットへのダウンロードは、Sysmac Studio の「同期」機能を使用します。

1 **［コントローラ］メニューの［オンライン］をクリックしオンラインにします。**

2 **［コントローラ］メニューから［同期］を選択し、［転送［パソコン→コントローラ］］をクリックします。**



参考

MC 機能モジュールは、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、および G5 シリーズ EtherCAT 通信内蔵タイプ、NX シリーズパルス出力ユニットを接続対象としています。

サーボドライバの接続対象機種

- サーボドライバの対象機種の形式は、形 R88D-1SN□□□-ECT、形 R88D-1SAN□□□-ECT、および形 R88D-KN□□□-ECT、形 R88D-KN□□□-ECT-L です。
- 形 R88D-KN□□□-ECT-R は、位置制御 (cyclic synchronous position)モードにだけ対応しているサーボドライバです。そのため、速度制御 (cyclic synchronous velocity)モード、あるいはトルク制御 (cyclic synchronous torque)モードを使用する機能は使用できません。

サーボドライバ側の設定

- MC 機能モジュールは、サーボドライバ側の入力信号や一部機能を使用しています。MC 機能モジュールの機能を正しく使用するためには、サーボドライバ側の信号配線やオブジェクトの設定を行う必要があります。
  設定内容については、「A-1 サーボドライバ 1S シリーズとの接続 (P.A-2)」、または「A-2 サーボドライバ G5 シリーズとの接続 (P.A-11)」を参照してください。

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。

3-3 軸グループ

ここでは、MC 機能モジュールの軸グループについて説明しています。

3-3-1 軸グループの概要

直線補間や円弧補間など、複数の軸を同時に動作させる場合に軸グループを使用します。
そのため、軸グループは複数の軸で構成されています。
Sysmac Studio を使用して軸グループ変数を設定することで、軸グループのモーション制御命令を起動したり、軸グループの状態を取得したりすることができます。
MC 機能モジュールでは、最大で 64 グループまで扱うことができます。
軸グループの仕様概要を下記に示します。

項目	仕様					
	形 NX701	形 NX502-		形 NX102	形 NX1P2	NJ シリーズ
		17□□*1 16□□*1	15□□ 14□□ 13□□			
軸グループ数	最大 64 グループ	最大 64 グループ	最大 32 グループ	最大 8 グループ	最大 8 グループ	最大 32 グループ
構成軸数	最大 4 軸／軸グループ					

*1. ユニットバージョン Ver.1.66 から追加された形式です。

MC 機能モジュールの軸グループには、以下の関連要素があります。

構成要素	内容	参照ページ
軸グループパラメータ	MC 機能モジュールで制御する各軸グループに関して、最高補間速度や最大補間加減速度といった軸グループ動作の設定を行います。 軸グループパラメータを設定するには、Sysmac Studio を使用します。	P.3-21
軸グループ変数	軸グループ変数は MC 機能モジュールで制御する各軸グループについて、軸グループパラメータの一部と指令補間速度や異常情報などのモニタ情報からなるシステム定義変数です。 Sysmac Studio のマルチビューエクスプローラで軸グループを追加すると軸グループ変数が作成されます。軸グループ変数の名前（以降は、軸グループ変数名と呼びます）は、このビューで設定することができます。	P.3-22
ユーザプログラムでの軸グループの指定方法	ユーザプログラムでは、モーション制御命令を使用してモーション制御を行います。多軸協調制御を行うモーション制御命令を「軸グループ指令」と呼びます。 軸グループ指令でどの軸グループを制御するかは、命令の入出力変数「AxesGroup（軸グループ）」に、システム定義変数の軸グループ変数名、あるいは Sysmac Studio 設定した軸グループ変数名を指定します。	P.3-24

使用上の注意

EtherCAT ネットワーク上の軸と、CPU ユニット上の軸で、軸グループを構成しないでください。
ネットワーク間の同期が行われないため、指令のタイミングにズレが発生し、意図しない動作になる可能性があります。

3-3-2 軸グループパラメータの概要

● 軸グループパラメータの一覧

区分	パラメータ名称
軸グループ基本設定	軸グループ番号
	モーション制御 *1
	軸グループ使用
	軸構成
	構成軸
軸グループ動作設定	最高補間速度
	最大補間加速度
	最大補間減速度
	補間加減速オーバー
	補間速度警告値
	補間加速度警告値
	補間減速度警告値
	軸グループ停止方法
	中心点補正許容率

*1. 形 NX701 CPU ユニットの場合に設定します。

軸グループパラメータの詳細については、「5-3 軸グループパラメータ (P.5-30)」を参照してください。

● 軸グループを使用するための必須設定項目

Sysmac Studio で作成した軸グループについて次の項目を設定することで、軸グループを使用することができます。

区分	パラメータ名称	設定内容	参照ページ
軸グループ基本設定	軸グループ番号	軸グループを作成した順に自動で設定されます。	P.5-30
	モーション制御 *1	［プライマリ定周期タスク］を選択します。	
	軸グループ使用	［使用］を選択します。	
	軸構成	制御する軸構成を選択します。	
	構成軸	軸グループに割り付ける軸を選択します。	

*1. 形 NX701 CPU ユニットの場合に設定します。

使用上の注意

- 最高補間速度や停止方法などは、動作条件に従って適切な値を設定してください。
- ［制御機能］の設定が［単軸位置制御のみ］の軸は、軸グループの構成軸に割り付けられません。

3-3-3 軸グループ変数の概要

軸グループ変数は、MC 機能モジュールの各軸グループの設定情報と現在位置や異常情報などのモニタ情報からなるシステム定義変数です。
Sysmac Studio で軸グループを作成すると、作成した順番に軸グループ変数が変数テーブルに登録されます。軸グループ変数のデータ型は構造体型で、データ型名は「_sGROUP_REF」です。

軸グループ変数名

MC 機能モジュールの軸グループ変数名には、システム定義変数の軸グループ変数名と、Sysmac Studio で軸グループを追加したときの軸グループ変数名の 2 種類があります。
システム定義変数の軸グループ変数名は、「_MC_GRP[0]～_MC_GRP[63]」、「_MC1_GRP[0]～_MC1_GRP[63]」、「_MC2_GRP[0]～_MC2_GRP[63]」です。
Sysmac Studio で軸グループを追加すると、「MC_Group000～MC_Group063」というデフォルト名称が、「_MC_GRP[0]～_MC_GRP[63]」に対して設定されます。
番号は追加した順番に前詰めです。これらの軸グループ変数名は、Sysmac Studio で軸グループごとに任意の変数名に変更することができます。
ユーザプログラムでは、システム定義変数の軸グループ変数名でも、Sysmac Studio で追加した軸グループ変数名のどちらも使用することができます。

● _MC_GRP[0-63]を使用した場合の例

システム定義変数の軸グループ変数名（グローバル変数ビューの割付先 *1）	Sysmac Studio で追加した軸グループ変数名のデフォルト	軸グループ番号例
_MC_GRP[0]	MC_Group000	軸グループ 0
_MC_GRP[1]	MC_Group001	軸グループ 1
:	:	:
_MC_GRP[63]	MC_Group063	軸グループ 63

*1. Sysmac Studio のグローバル変数ビューにある割付先欄の名称を変更すると異常となります。

参考

- 形 NX701 CPU ユニットの場合は、「_MC_GRP[0-63]」または「_MC1_GRP[0-63]」と、「_MC2_GRP[0-63]」が使用できます。
  形 NX502 CPU ユニットの場合は、「_MC_GRP[0-31]」だけが使用できます。
  形 NX102 および形 NX1P2 CPU ユニットの場合は、「_MC_GRP[0-7]」だけが使用できます。
  NJ シリーズ CPU ユニットの場合は、「_MC_GRP[0-31]」だけが使用できます。
- 形 NX701 CPU ユニットの場合、「_MC_GRP[0-63]」と「_MC1_GRP[0-63]」は、軸グループ番号が同じであれば同じ値を参照します。どちらの軸グループ変数でも使用可能ですし、混在させても使用可能です。
- プライマリ定周期タスクに割り付けた軸グループ変数は、「_MC_GRP[0-63]」、または「_MC1_GRP[0-63]」となります。
  定周期タスク（実行優先度 5）に割り付けた軸グループ変数は、「_MC2_GRP[0-63]」となります。

軸グループ変数の階層と軸グループ変数名の変更例

_MC_GRP[0]を例に説明します。ほかの軸グループ変数も同様となります。

```
_MC_GRP[0]                          : 軸グループ変数
    _MC_GRP[0].Status               : 「軸グループステータス」を表す階層
        :
    _MC_GRP[0].Cmd                  : 「軸グループ指令値」を表す階層
        _MC_GRP[0].Cmd.Vel          : 「指令補間速度」を表す変数
        _MC_GRP[0].Cmd.AccDec       : 「指令補間加減速度」を表す変数
        :
    _MC_GRP[0].Cfg                  : 「軸グループ基本設定」を表す階層
        _MC_GRP[0].Cfg.GrNo         : 「軸グループ番号」を表す変数
        _MC_GRP[0].Cfg.GrEnable     : 「軸グループ使用」を表す変数
```

```
_MC_GRP[0].Kinematics                 :「運動学変換設定」を表す階層
    _MC_GRP[0].Kinematics.GrType      :「軸構成」を表す変数
    _MC_GRP[0].Kinematics.Axis[0]     :「構成軸（軸 A0）」の軸番号を表す変数
        :
    _MC_GRP[0].Kinematics.Axis[3]     :「構成軸（軸 A3）」の軸番号を表す変数
_MC_GRP[1]                            :軸グループ変数
        :
```

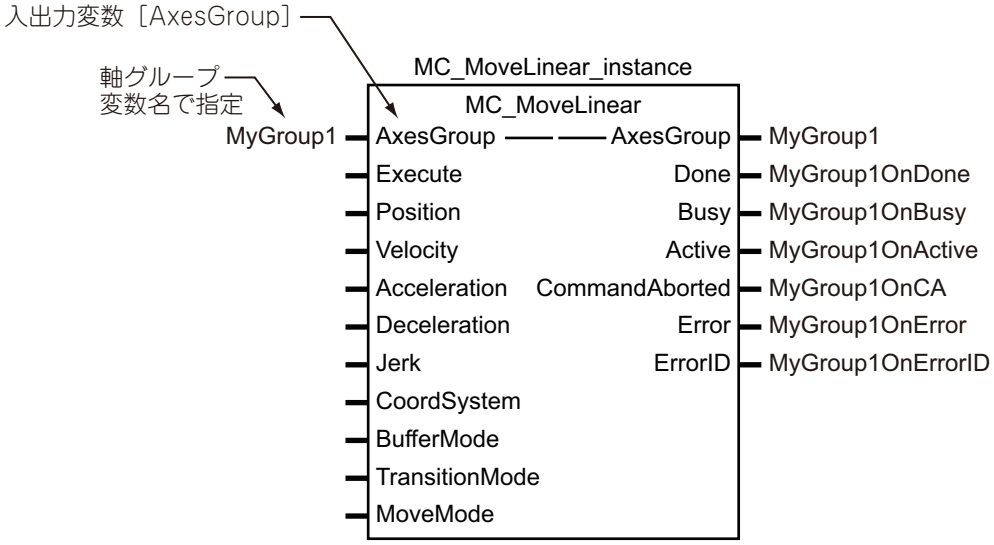
● **例**

「MC_Group000」を「MyGroup1」に変更した場合、「指令補間速度」を表す変数は、「MyGroup1.Cmd.Vel」と、「_MC_GRP[0].Cmd.Vel」のどちらでも扱うことができます。

軸グループ変数の詳細については、「軸グループ変数 (P.6-30)」を参照してください。

3-3-4　ユーザプログラムでの軸グループの指定方法

ユーザプログラムでは、モーション制御命令の入出力変数「AxesGroup」に、軸グループ変数名を指定します。
下図はシステム定義変数の軸変数名が「_MC_GRP[0]」の軸グループについて、Sysmac Studio で追加したときの軸グループ変数名を「MyGroup1」に変更した場合の例です。



「MyGroup1」のかわりに、システム定義変数の「_MC_GRP[0]」も使用することができます。

モーション制御命令の詳細については、「6-2 モーション制御命令 (P.6-5)」を参照してください。

各モーション制御命令の詳細機能については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の各命令の説明を参照してください。

3-4 軸グループの設定手順

ここでは、Sysmac Studio を使用して軸グループを設定する手順について説明します。
直線補間や円弧補間などの軸グループ指令の命令を使用しない場合は、設定する必要はありません。

3-4-1 軸グループの設定フロー

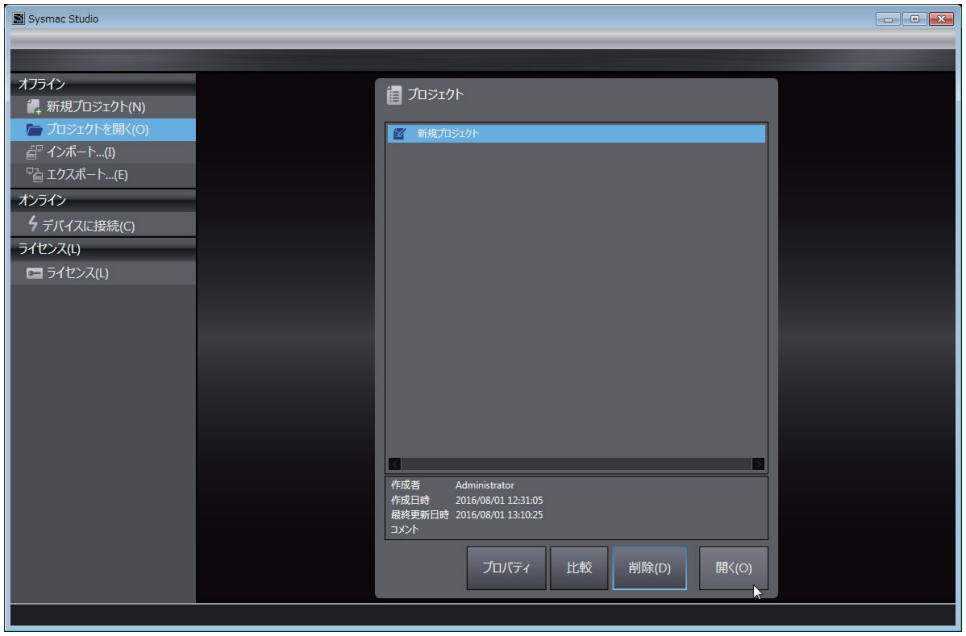


3-4-2 設定手順

ここでは、すでに軸が追加されている既存のプロジェクトで、軸グループを設定する方法について説明します。

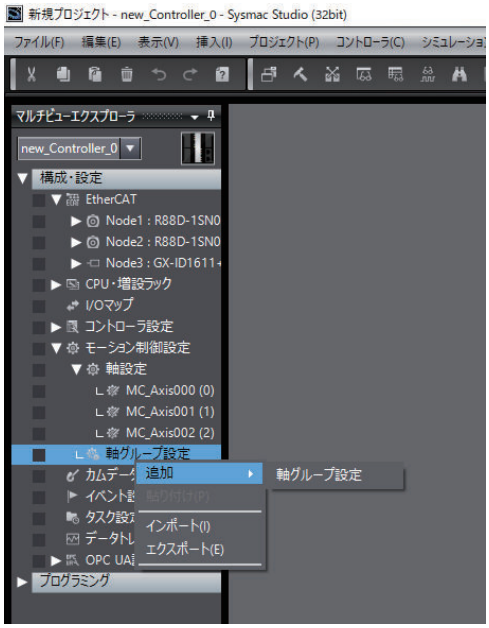
Sysmac Studio の起動

Sysmac Studio を起動して既存プロジェクトを開きます。

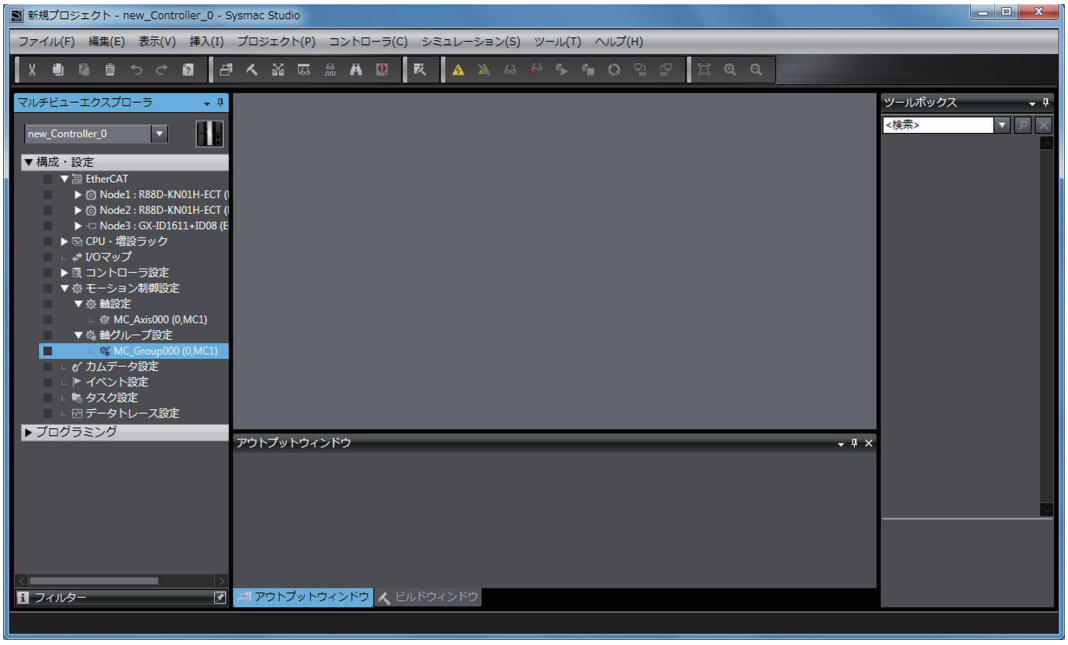


軸グループの追加

マルチビューエクスプローラで［軸グループ設定］を右クリックし、［追加］から［軸グループ設定］をクリックします。



マルチビューエクスプローラに軸グループが追加されます。
追加された軸グループの変数名のデフォルトは「MC_Group000」です。



● インポートによる軸グループの追加

事前に作成しておいた軸グループ設定ファイルをインポートすることで［軸グループ設定］を追加することもできます。
軸グループ設定ファイルは［軸グループ設定］をエクスポートすることで作成できます。
詳細は、『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。

バージョン関連情報

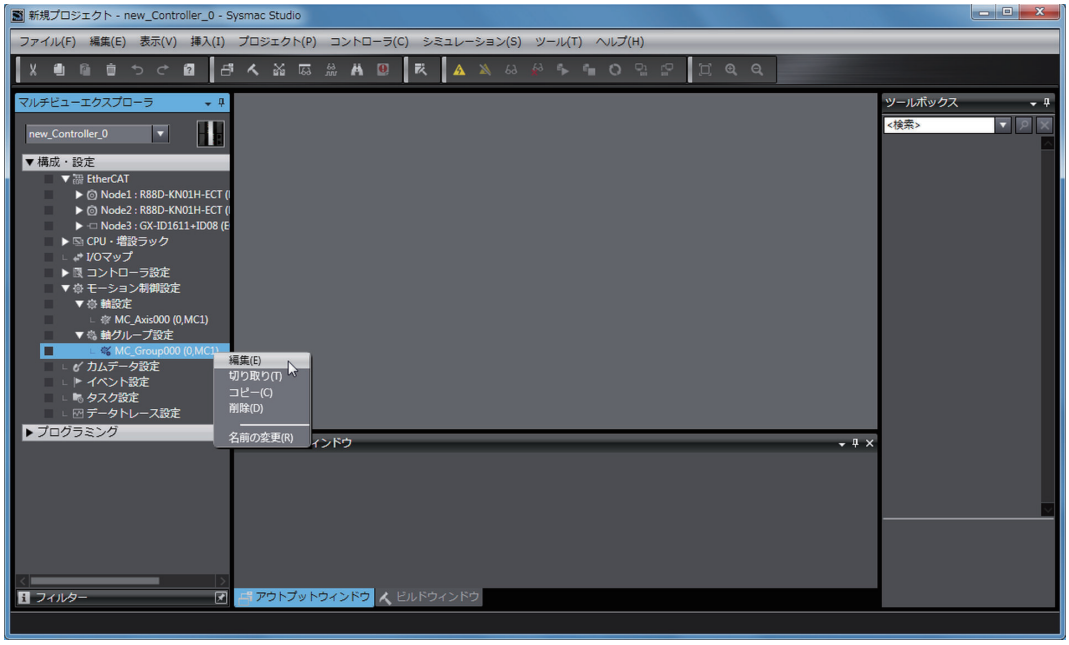
［インポート］および［エクスポート］は、Sysmac Studio Ver.1.57 以降で使用可能です。

● コピーによる軸グループの追加

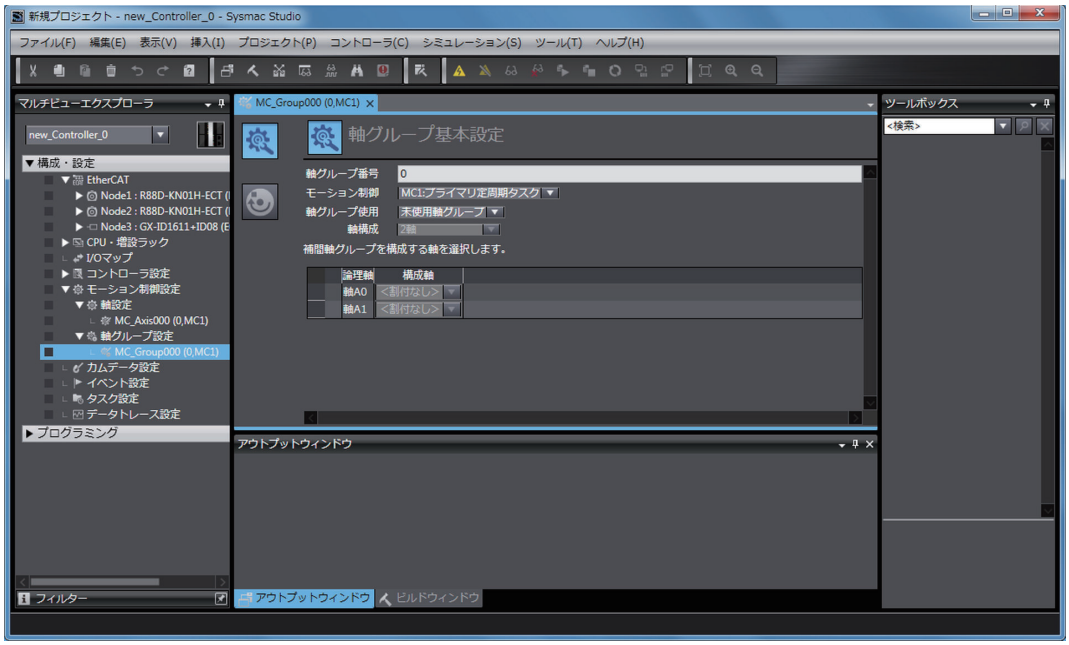
既存プロジェクトの軸グループをコピーして貼り付けることで、新規に軸グループを作成することも可能です。

軸グループパラメータの設定

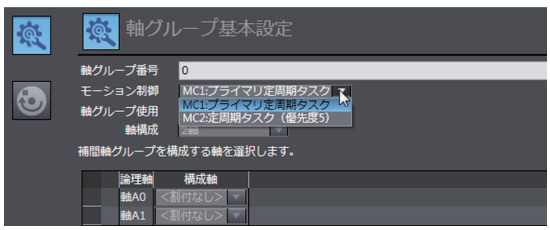
1 **マルチビューエクスプローラに追加された軸グループを右クリックし、[編集] をクリックします。**



軸グループパラメータ設定画面の［軸グループ基本設定］画面が表示されます。



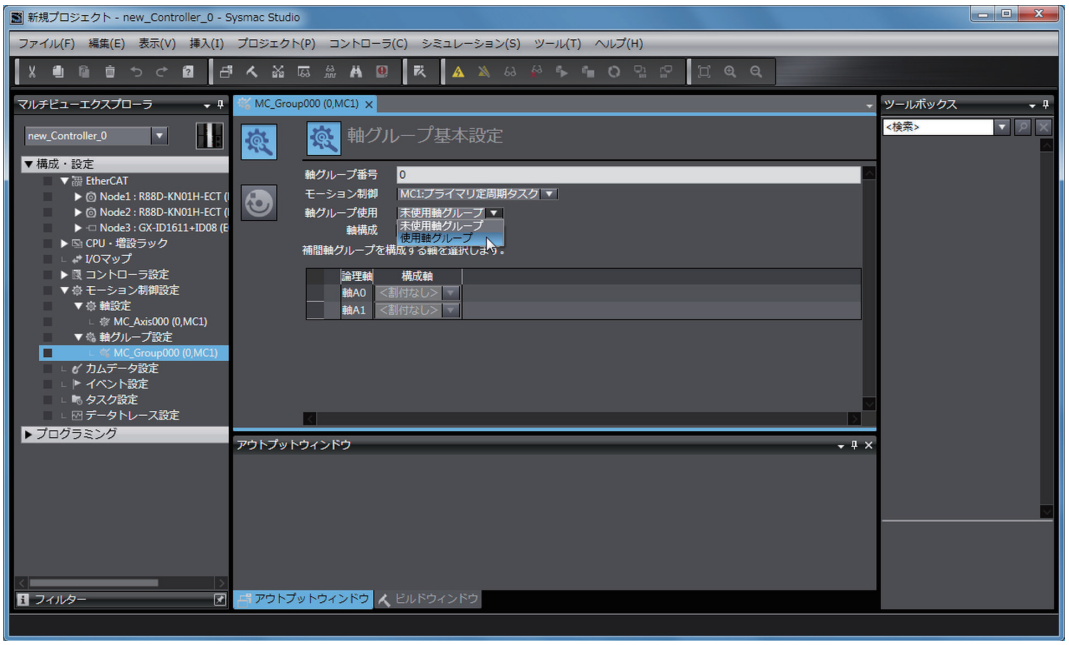
2 **［モーション制御］から［プライマリ定周期タスク］、または［定周期タスク（優先度 5）］を選択します。**



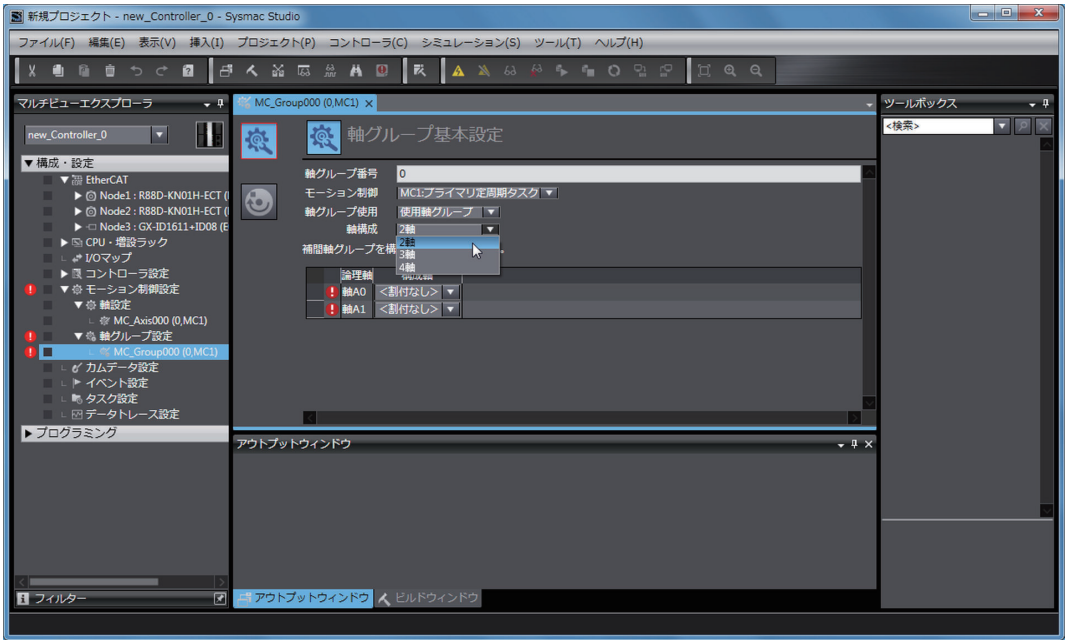
参考

形 NX701 CPU ユニットの場合に設定します。

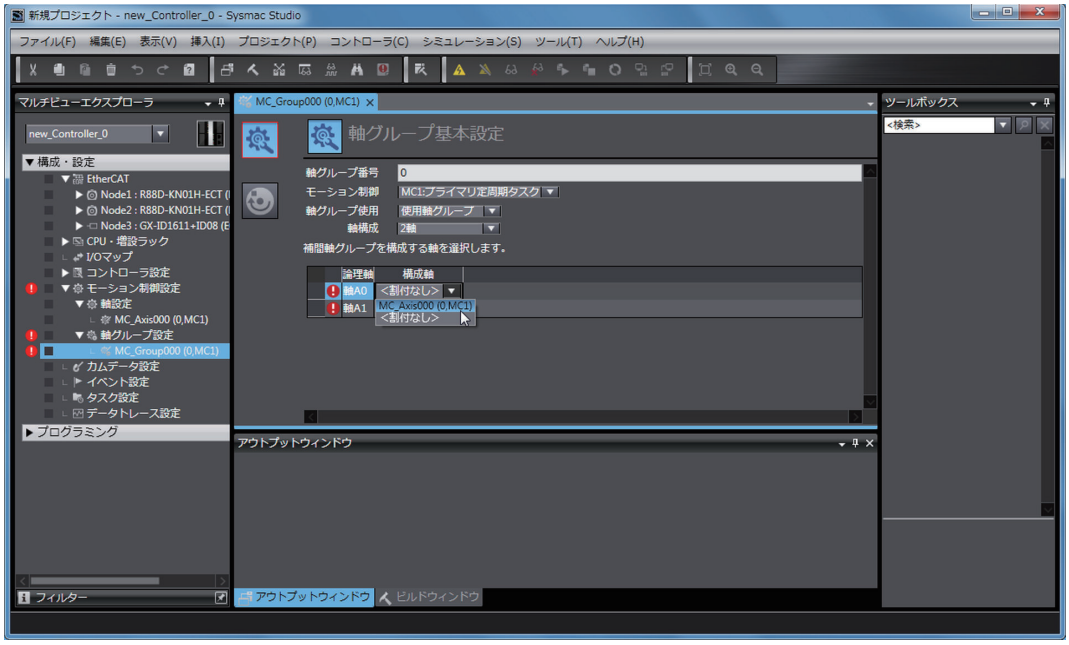
3 ［軸グループ使用］で、［使用軸グループ］を選択します。



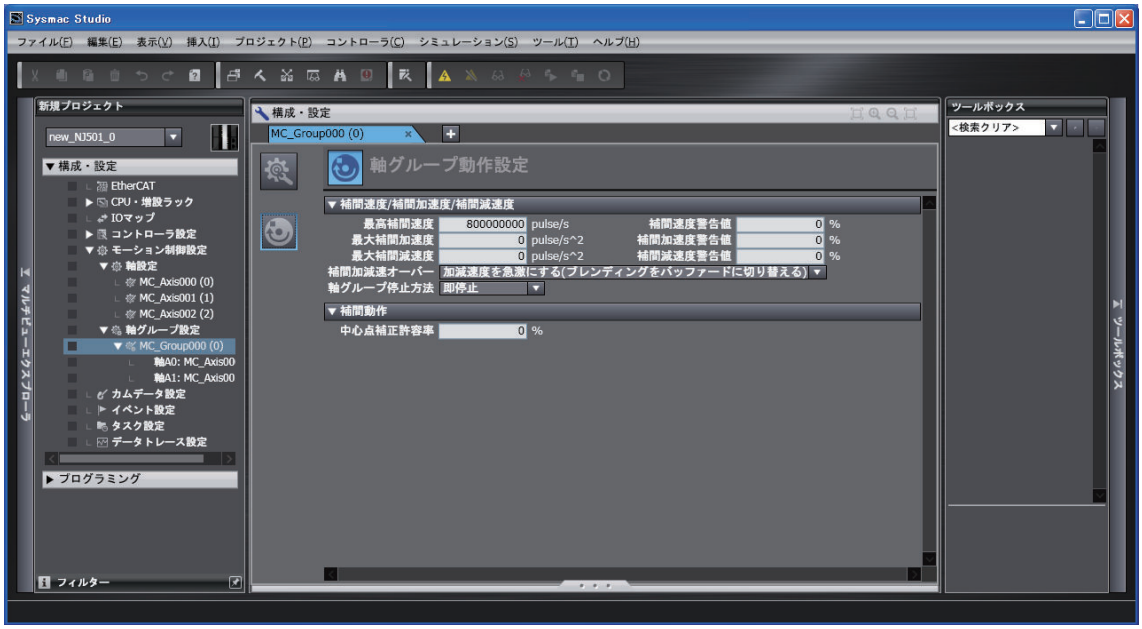
4 ［軸構成］で軸グループの構成を選択します。下記は 2 軸構成を選択した場合です。



5 ［論理軸］に使用する軸を割り付けます。



6 **下側のアイコンをクリックすると［軸グループ動作設定］画面が表示されます。**
各設定は、装置の動作条件に従って適切な値を設定してください。



参考

ユーザプログラムで使用している軸グループ変数名の変更
すでに使用している軸グループ変数名を変更する場合、次の 2 つの操作を行ってください。
- 変数宣言内の変数テーブルにある軸グループ変数名の変更。
- ユーザプログラムの軸グループ変数名の変更。

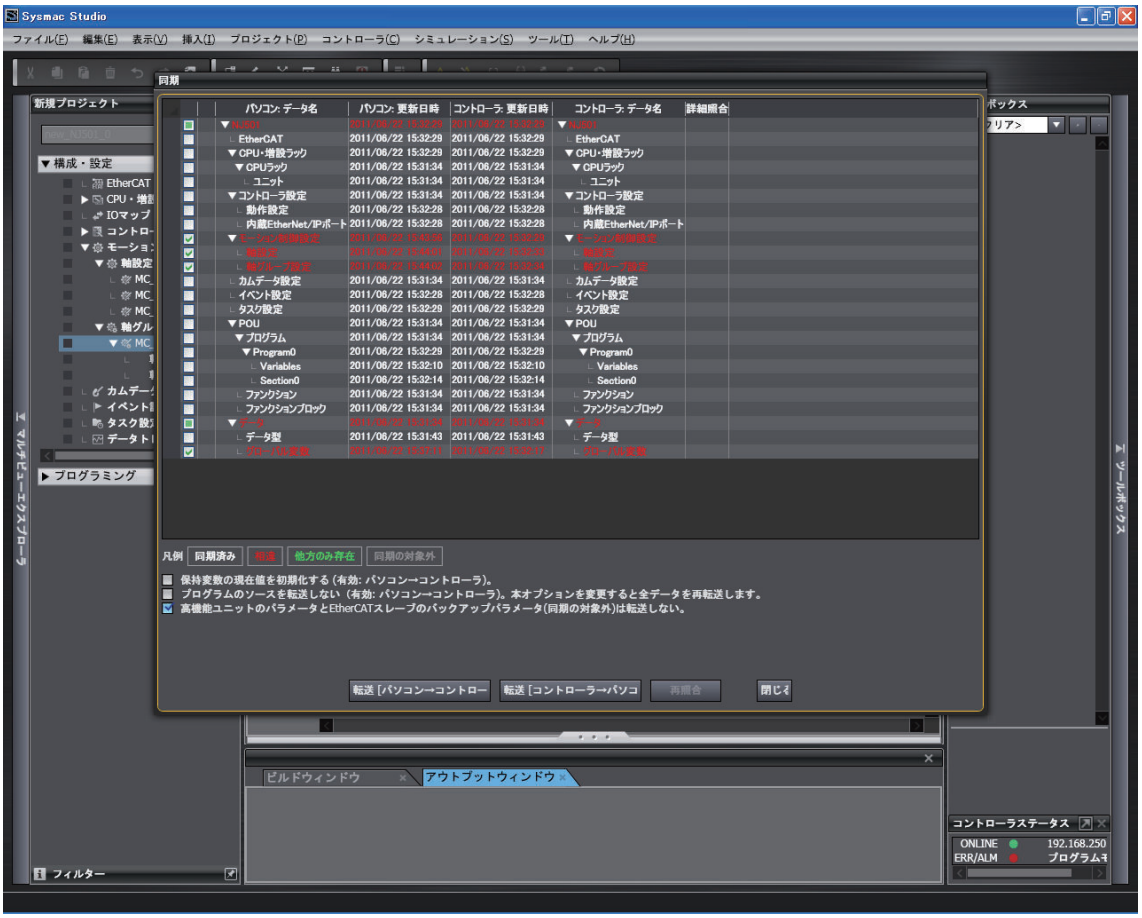
変数テーブルの軸グループ変数名を変更しても、ユーザプログラムの軸グループ変数名には反映されません。
変数テーブルで宣言されていない変数名をユーザプログラムで使用すると、異常が発生します。必ず両方とも変更してください。

CPU ユニットへのダウンロード

NJ/NX シリーズ CPU ユニットへのダウンロードは、Sysmac Studio の「同期」機能を使用します。

1 **［コントローラ］メニューの［オンライン］をクリックしオンラインにします。**

2 **［コントローラ］メニューから［同期］を選択し、［転送［パソコン→コントローラ］］をクリックします。**



4

Sysmac Studio による配線確認

Sysmac Studio の MC 試運転機能について説明します。MC 試運転機能により、プログラムレスでセンサ信号のモニタやモータ配線の確認を行うことができます。

4-1 MC 試運転機能 4-2
4-1-1 MC 試運転機能一覧 4-2
4-1-2 使用手順 4-3
4-1-3 軸パラメータの設定例 4-4
4-1-4 MC 試運転機能の立ち上げ手順 4-5
4-2 センサ信号のモニタ 4-7
4-3 モータ動作の確認 4-8
4-3-1 サーボ ON 4-8
4-3-2 ジョグ送り 4-9
4-3-3 原点復帰 4-9
4-3-4 絶対値位置決め 4-10
4-3-5 相対値位置決め 4-11

4-1 MC 試運転機能

ここでは、MC 試運転機能を使用して、配線確認と基本的な設定の確認について説明します。
Sysmac Studio の MC 試運転機能を利用することにより、プログラムレスで配線確認ができます。

使用上の注意

MC 試運転機能は、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、NX シリーズパルス出力ユニットを対象としています。他社製のドライバでは使用しないでください。

参考

- オムロン製サーボドライバ 1S シリーズとの接続方法については、「A-1 サーボドライバ 1S シリーズとの接続 (P.A-2)」を参照してください。
- オムロン製サーボドライバ G5 シリーズとの接続方法については、「A-2 サーボドライバ G5 シリーズとの接続 (P.A-11)」を参照してください。

4-1-1 MC 試運転機能一覧

MC 試運転機能には下記の機能があります。
特に、モニタによるセンサ類の配線確認、軸動作によるサーボ ON、ジョグ送り、原点復帰の動作を確認することにより、CPU ユニットとサーボドライバとの配線確認ができます。

分類	機能	説明	設定項目／モニタ項目
軸動作	サーボ ON/OFF	サーボ ON/OFF します。	−
	ジョグ送り	正方向または負方向へジョグ送りを行います。	目標速度 加速度／減速度
	原点復帰	原点復帰パラメータの設定で原点復帰を行います。	原点復帰パラメータ
	絶対値位置決め	絶対値位置決めを行います。 *1	目標位置 目標速度 加速度／減速度 ジャーク
	相対値位置決め	相対値位置決めを行います。	移動距離 目標速度 加速度／減速度 ジャーク
	減速停止	MC 試運転中に減速停止します。	−
	異常解除	MC 機能モジュールで発生している異常を解除します。	−
モニタ	サーボドライバ状態	サーボドライバの状態をモニタします。	サーボ ON/OFF サーボレディ 主回路電源
	入力信号	入力信号の状態をモニタします。	正方向限界入力／負方向限界入力 即停止入力 原点近傍入力 原点入力 外部ラッチ入力 1～2
	現在位置モニタ	現在位置をモニタします。	現在位置（指令／フィードバック）
	現在速度モニタ	現在速度をモニタします。	現在速度（指令／フィードバック）

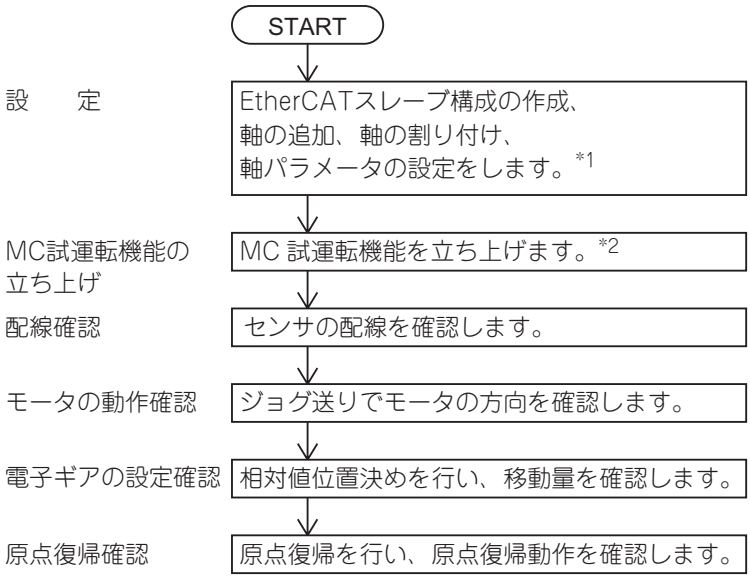
分類	機能	説明	設定項目／モニタ項目
	軸ステータス	軸の状態をモニタします。	軸起動準備完了 停止中 位置決め動作中 連続動作中 原点復帰中 減速停止中 原点確定 原点停止
	異常一覧	MC 機能モジュールで発生している異常をモニタします。	MC 共通の異常 軸の異常 軸グループの異常

*1. 軸の［カウントモード］が［ロータリモード］の場合は、正方向の目標位置に向かって位置決めを行います。
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 モーション編(SBCE-434)』の MC_MoveAbsolute（絶対値位置決め）命令を参照してください。

4-1-2 使用手順

MC 試運転を行うためには、まず下記の 2 点について確認をしてください。
- Sysmac Studio とコントローラ本体がオンラインでつながっていますか。
- 他の Sysmac Studio で MC 試運転を起動していませんか。

確認ができましたら、以下のフローにしたがって操作をしてください。



*1.「第 3 章 軸と軸グループの設定 (P.3-1)」を参照してください。
*2.「4-1-4 MC 試運転機能の立ち上げ手順 (P.4-5)」を参照してください。

使用上の注意

- Sysmac Studio からの指令により、サーボ ON、ジョグ送り、相対値位置決め、絶対値位置決め、原点復帰を実行すると、設定された速度でモータが動作します。
  モータの動作による危険がないことを十分に確認の上、実行してください。
- Sysmac Studio から操作をする場合は、安全にモータが停止できるように外部に非常停止回路を設置してください。パソコンの異常が発生した場合など、Sysmac Studio から指令が与えられなくなる場合があります。
- Sysmac Studio から操作する場合は、EtherCAT 通信の設定を行い、通信を確立してから実行してください。
- 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の手順は、『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。

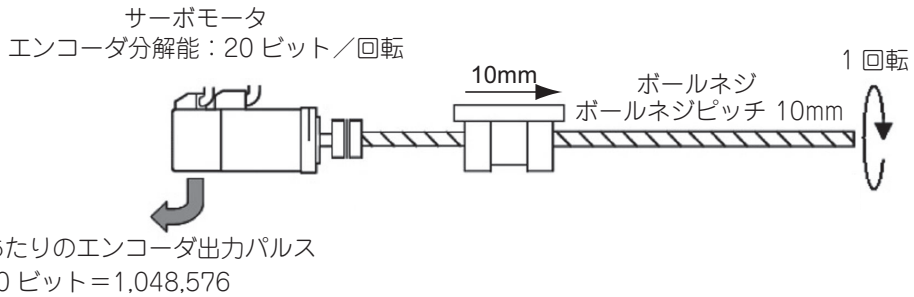
参考

次の操作をすることで、いつでも MC 試運転機能を終了できます。
- Sysmac Studio の［コントローラ］メニューにある［MC 試運転］で［終了］を選択する。
- Sysmac Studio のマルチビューエクスプローラの対称軸で右クリックし、［MC 試運転を終了する］を選択する。
- Sysmac Studio の MC 試運転画面を閉じる。
- Sysmac Studio を終了する。

操作方法の詳細については、『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。

4-1-3 軸パラメータの設定例

Sysmac Studio の MC 試運転機能を行う場合、次の軸パラメータを設定します。
1 軸の装置を例に設定例を説明します。



設定項目	設定内容
軸変数名	Axis1 *1
軸番号	1 *2
軸使用	使用軸
軸種別	サーボ軸
入力デバイス／出力デバイス	1 *3
表示単位	μm
モータ 1 回転のパルス数	1,048,576 *4
モータ 1 回転の移動量	10,000 *4
最高速度	500,000 *5
JOG 最高速度	50,000 *6
最大加速度	5,000,000 *7
最大減速度	5,000,000 *7

設定項目	設定内容
ソフトウェアリミット機能	指令位置に対して有効で即停止させる。
正方向ソフトウェアリミット	500,000 *8
負方向ソフトウェアリミット	0 *8
カウントモード	リニアモード

*1. 複数軸の場合は、軸ごとに異なる変数名を設定します。
*2. 複数軸の場合は、軸ごとに異なる値を設定します。
*3. サーボドライバのノードアドレスと同じ値を設定します。複数軸の場合は、軸ごとに異なる値を設定します。
*4. 位置の指令単位は 1(μm)となります。
*5. 最高速度は、3,000r/min = 30m/min = 0.5m/s = 500,000 μm/s としています。
*6. ジョグ最高速度は、最高速度の 10%の 0.05m/s = 50,000 μm/s としています。
*7. 最大加速度および最大減速度は、5m/s^2 です。最高速度の 3,000r/min までの加速時間を 0.1s としています。
*8. 装置の可動範囲の内側の位置を設定します。正方向ソフトウェアリミットの設定値は、50cm = 500,000 μm としています。

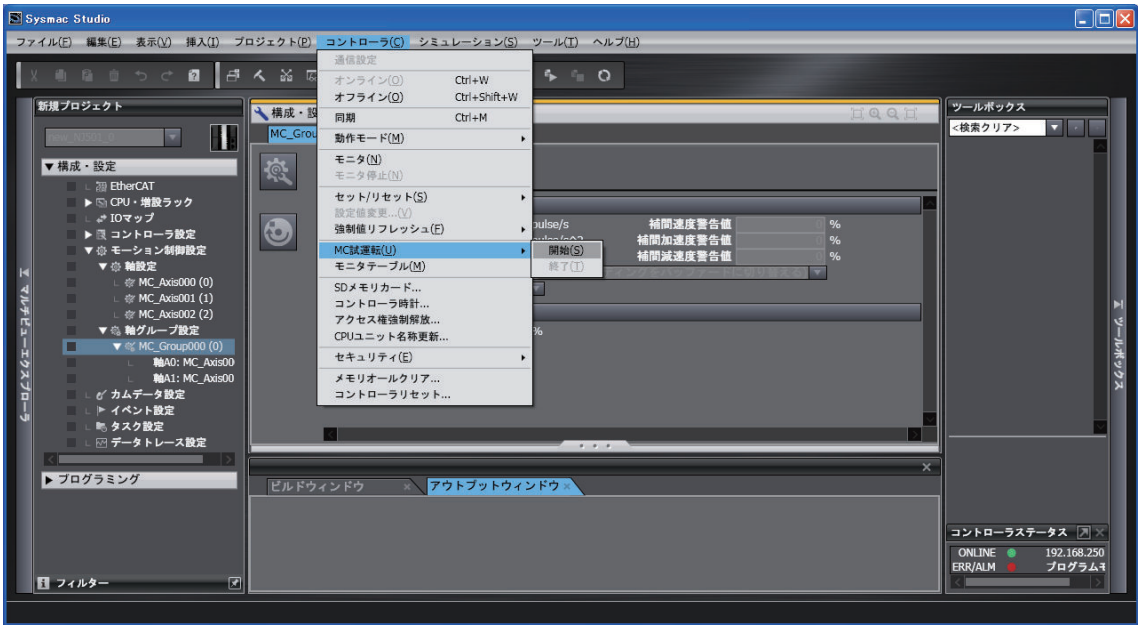
4-1-4 MC 試運転機能の立ち上げ手順

MC 試運転機能は、Sysmac Studio から起動します。

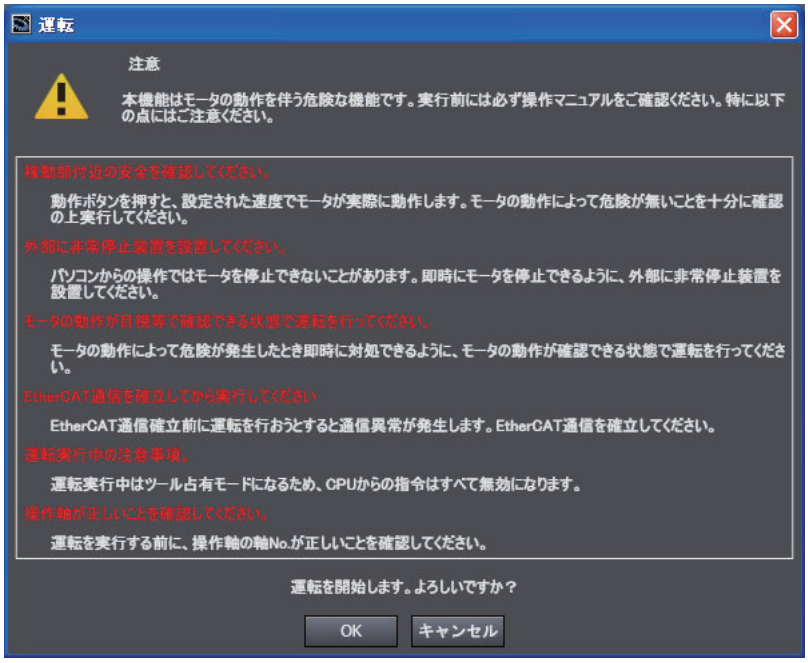
1 **Sysmac Studio を起動して、サーボ軸の設定が完了している既存プロジェクトを開きます。**

2 **［コントローラ］メニューの［オンライン］をクリックしオンラインにします。**

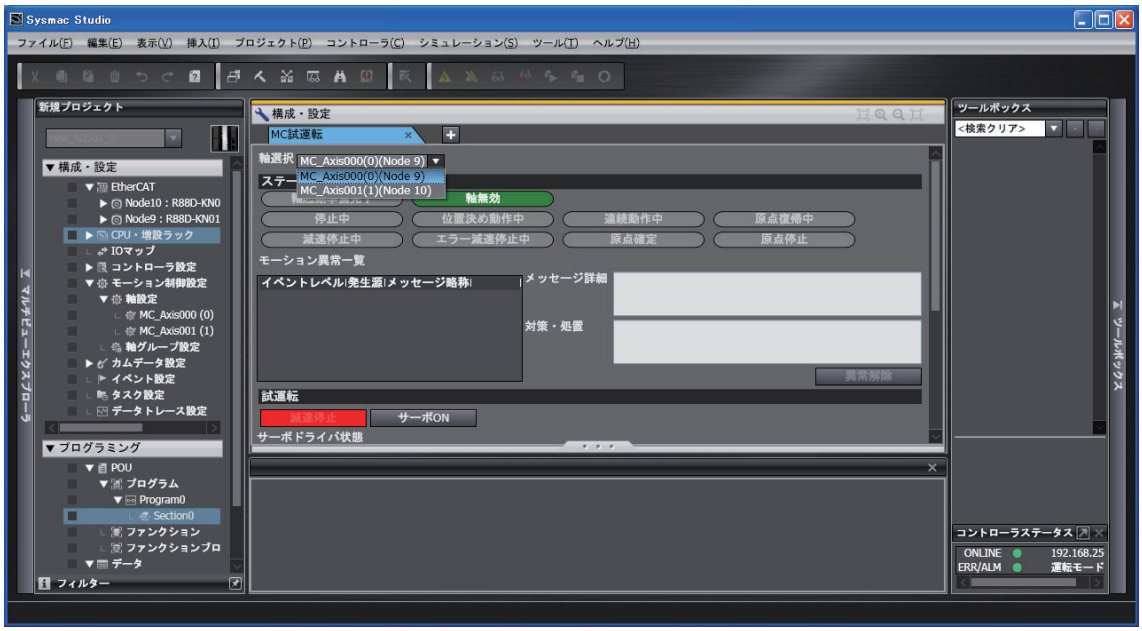
3 **［コントローラ］メニューの［MC 試運転］から［開始］をクリックします。**



以下の注意ダイアログが表示されるので、内容をよく読み、安全が確認できた場合は［OK］をクリックしてください。



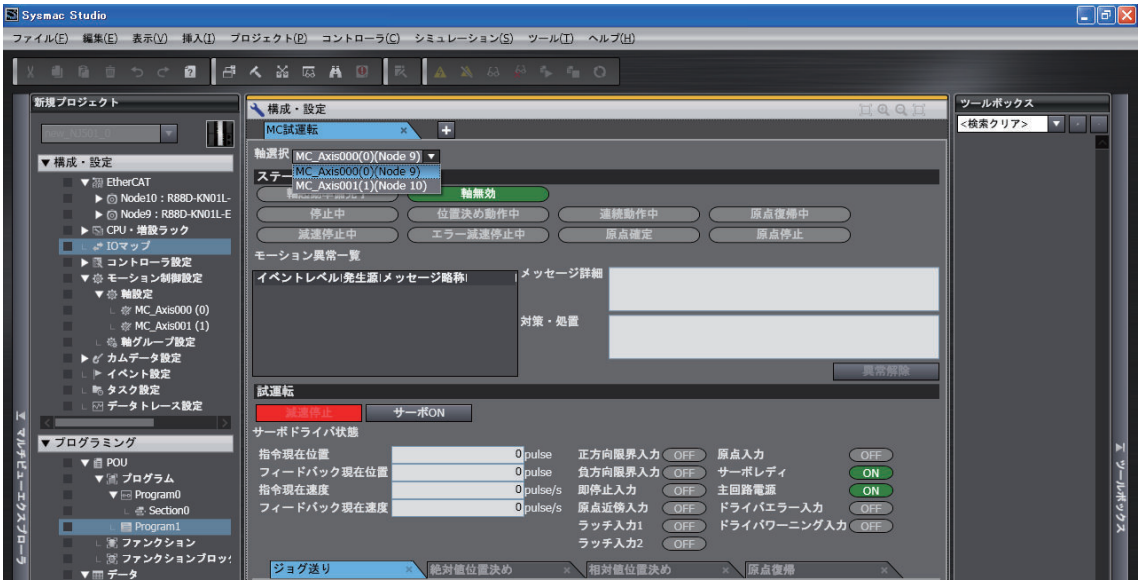
MC 試運転画面が表示されます。



4-2　センサ信号のモニタ

センサ信号の配線確認は、各入力信号の表示を使用します。

1　**MC 試運転画面で確認する軸を選びます。**



2　**各入力信号に接続したセンサを ON/OFF して、モニタ画面の該当する信号が正しく ON/OFF するか確認します。**

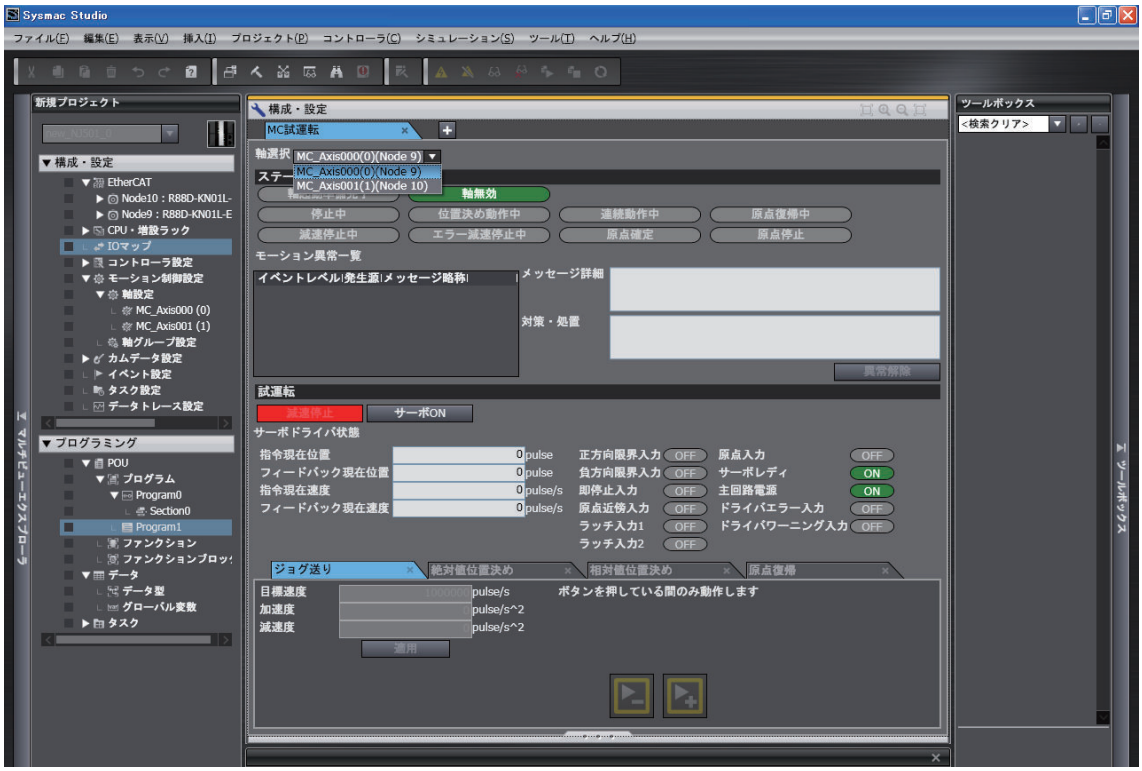
4-3　モータ動作の確認

モータ動作の確認は、MC 試運転の各機能を使用します。

4-3-1　サーボ ON

［サーボ ON］ボタンを押すことにより、サーボ ON とサーボ OFF を切り替えることができます。

1　サーボ ON する軸を選びます。



2　［サーボ ON］ボタンを押して、サーボ ON 状態にします。

3　［サーボ OFF］ボタンを押すと、サーボ OFF 状態に戻ります。

使用上の注意

- Sysmac Studio からの指令により、サーボ ON、ジョグ送り、相対値位置決め、絶対値位置決め、原点復帰を実行すると、設定された速度でモータが動作します。
  モータの動作による危険がないことを十分に確認の上、実行してください。
- Sysmac Studio から操作をする場合は、安全にモータが停止できるように外部に非常停止回路を設置してください。パソコンの異常が発生した場合など、Sysmac Studio から指令が与えられなくなる場合があります。
- Sysmac Studio から操作する場合は、EtherCAT 通信の設定を行い、通信を確立してから実行してください。
- NX シリーズパルス出力ユニットを使用する場合は、モータドライバの通電 ON/OFF を別途行う必要があります。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。

4-3-2 ジョグ送り

1 **MC 試運転画面の［ジョグ送り］タブでジョグ送りする軸を選びます。**

2 **［サーボ ON］ボタンを押して、サーボ ON 状態にします。**

3 **目標速度、加速度、減速度の値を入力し［適用］ボタンを押します。**

4 **ボタン、あるいは ボタンを押します。**

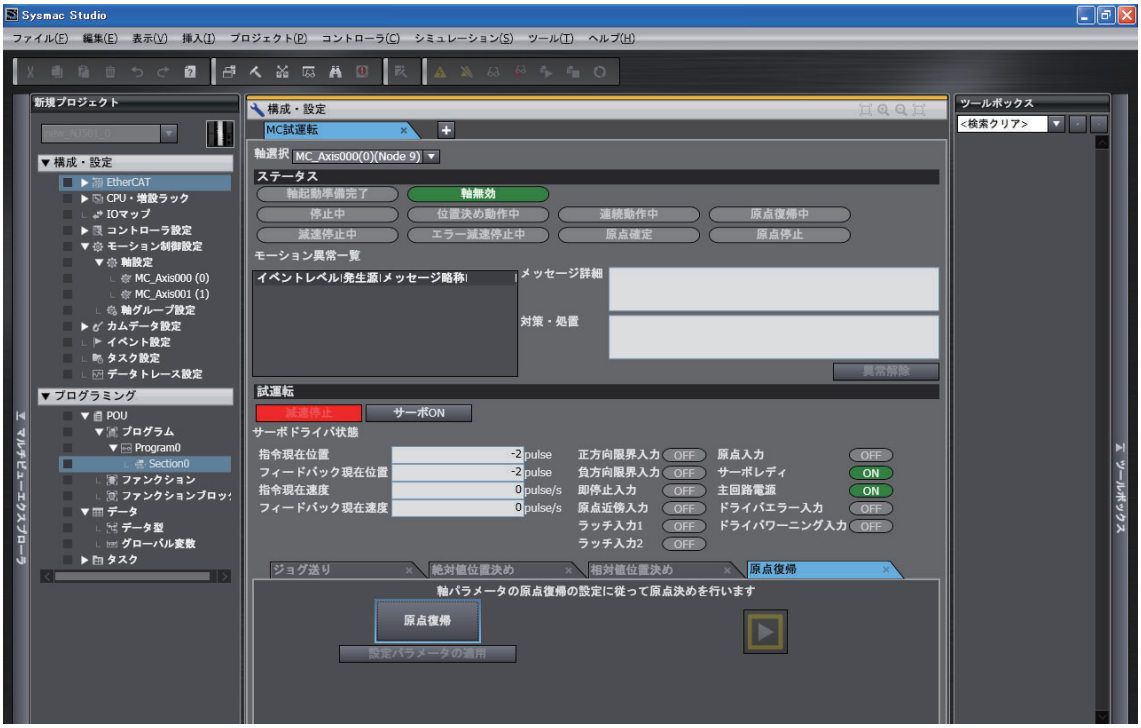
ボタンを押している間、正方向あるいは負方向に動作します。
設定した方向に動作するか確認してください。

4-3-3 原点復帰

MC 試運転機能を使って原点復帰を行うには、軸パラメータの［原点復帰設定］が必要になります。原点復帰の詳細については、「第 8 章 原点決め (P.8-1)」ならびに「5-2-9 原点復帰設定 (P.5-26)」を参照してください。

1 **あらかじめ、軸パラメータ設定画面の［原点復帰設定］で原点復帰パラメータを設定しておきます。**

2 **試運転画面の［原点復帰］タブを選びます。**
以下の画面が表示されます。



3 **原点復帰を行う軸を選びます。**

4 **［サーボ ON］ボタンを押して、サーボ ON 状態にします。**

5 **［設定パラメータの適用］ボタンを押します。**

6 **ボタンを押します。**

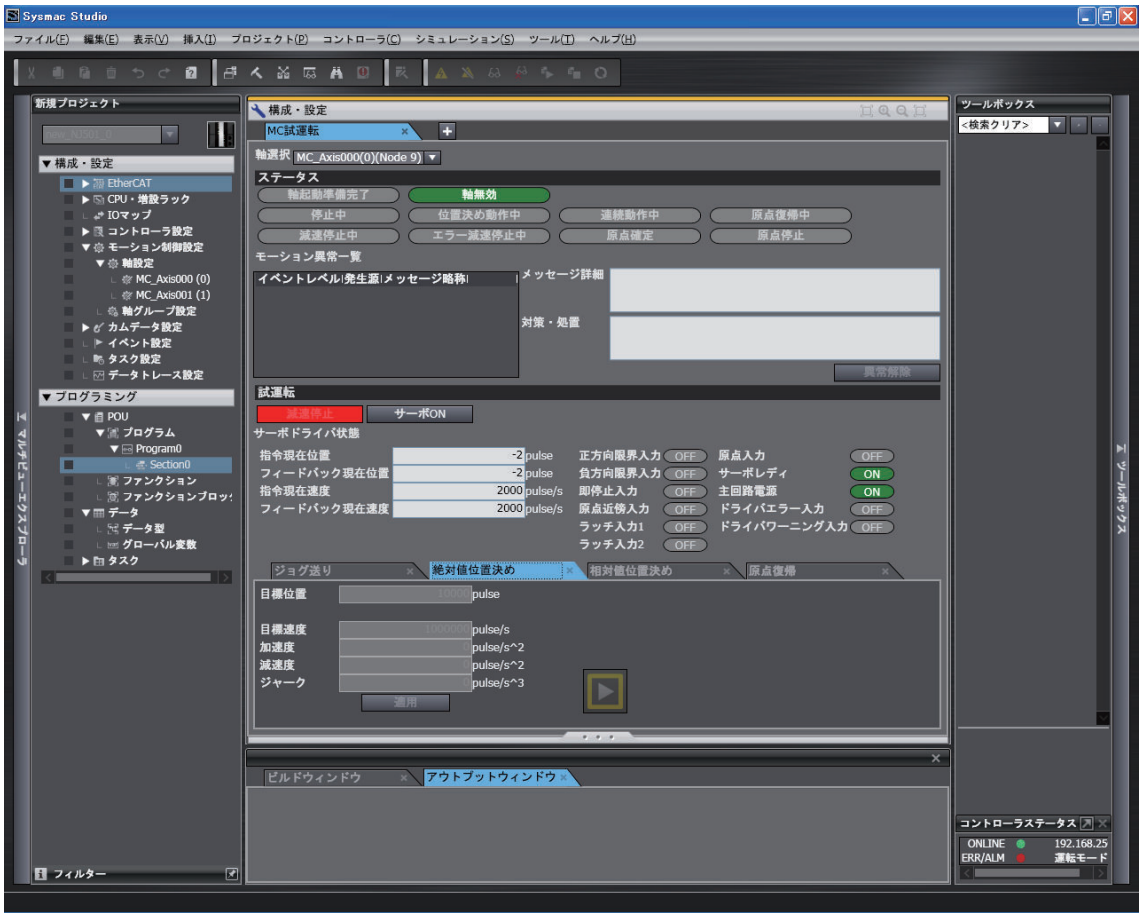
原点復帰動作が設定とおりになっているか、確認してください。

参考

- ［原点復帰設定］ボタンを押すと、［軸パラメータ設定］の［原点復帰設定］画面が表示されるので、［原点復帰パラメータ］を設定してください。
- あらかじめ［原点復帰パラメータ］を設定しておけば、［設定パラメータの適用］ボタンを押すと、その設定が適用されます。

4-3-4 絶対値位置決め

1 **MC 試運転画面で、［絶対値位置決め］タブを選択します。**
以下の画面が表示されます。



2 **絶対値位置決めを行う軸を選びます。**

3 **［サーボ ON］ボタンを押して、サーボ ON 状態にします。**

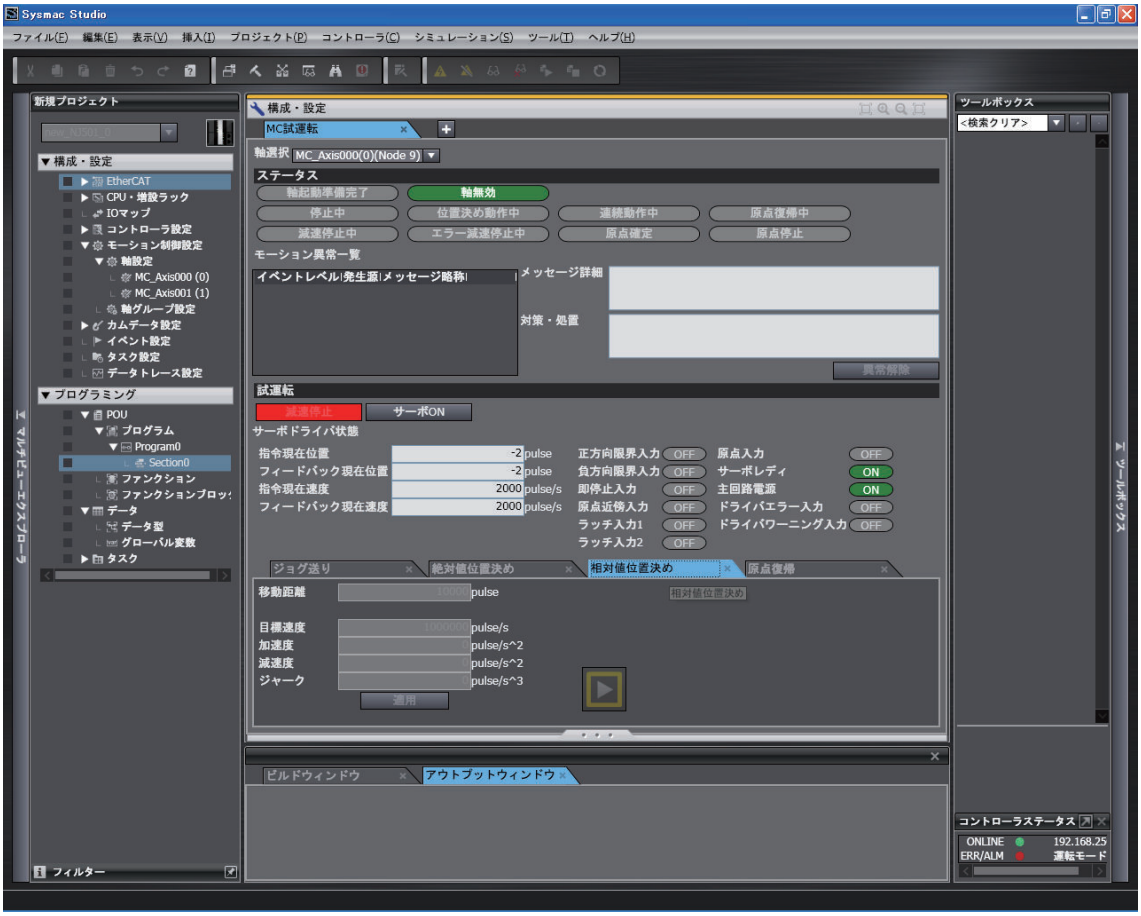
4 **目標位置、目標速度、加速度、減速度、ジャークの値を入力し［適用］ボタンを押します。**

5 ボタンを押すと、絶対値位置決めを開始します。

位置決めが完了した位置が設定とおりになっているか、確認してください。

4-3-5 相対値位置決め

1 MC 試運転画面で、[相対値位置決め] タブを選択します。
以下の画面が表示されます。



2 相対値位置決めを行う軸を選びます。

3 [サーボ ON] ボタンを押して、サーボ ON 状態にします。

4 目標移動距離、目標速度、加速度、減速度、ジャークの値を入力し [適用] ボタンを押します。

5 ボタンを押すと、相対値位置決めを開始します。

移動距離が設定とおりになっているか、確認してください。

5

モーション制御パラメータ

この章では、モーション制御で使用する軸パラメータと軸グループパラメータについて説明します。

5-1 概要 5-2

5-2 軸パラメータ 5-4

5-2-1 軸パラメータの一覧 5-4
5-2-2 軸基本設定 5-6
5-2-3 単位変換設定 5-11
5-2-4 動作設定 5-19
5-2-5 拡張動作設定 5-22
5-2-6 リミット設定 5-23
5-2-7 位置カウント設定 5-23
5-2-8 サーボドライバ設定 5-26
5-2-9 原点復帰設定 5-26
5-2-10 軸パラメータの設定例 5-27

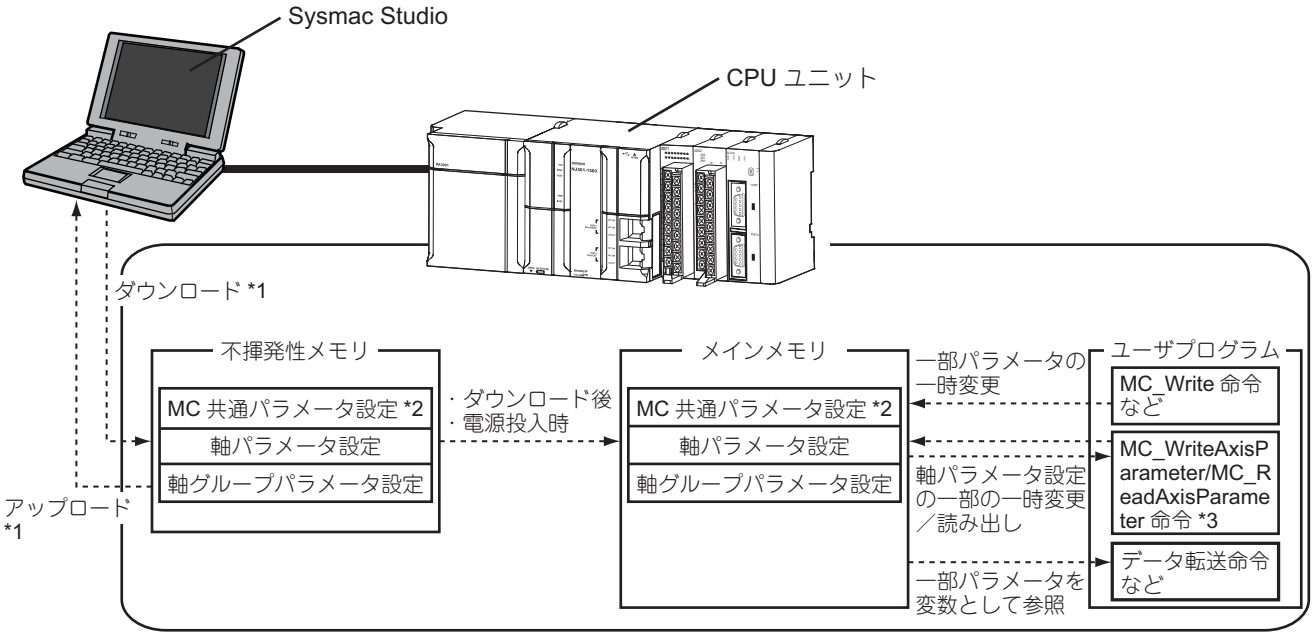
5-3 軸グループパラメータ 5-30

5-3-1 軸グループパラメータの一覧 5-30
5-3-2 軸グループ基本設定 5-30
5-3-3 軸グループ動作設定 5-32
5-3-4 軸グループの有効 5-33

5-1 概要

NJ/NX シリーズ CPU ユニットの MC 機能モジュールでは、モーション制御命令により単軸での動作と軸グループでの多軸動作を行うことができます。
軸パラメータと軸グループパラメータは、これらの動作を設定するためのパラメータです。
軸パラメータは常に設定する必要がありますが、軸グループでの多軸動作を使用しない場合、軸グループパラメータは設定する必要はありません。

これらのパラメータをモーション制御パラメータ設定（略称: MC パラメータ設定）とよびます。



*1.「ダウンロード」および「アップロード」は、Sysmac Studio の「同期」機能を使用します。
*2. 現バージョンの MC 機能モジュールでは、「MC 共通パラメータ設定」の設定項目はありません。
*3. MC_WriteAxisParameter（軸パラメータ書込）命令と MC_ReadAxisParameter（軸パラメータ読出）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することができます。

MC パラメータ設定のデータフロー

- Sysmac Studio で設定した MC パラメータ設定を CPU ユニットへダウンロードすると、CPU ユニット内蔵の不揮発性メモリに保存されます。
  また、Sysmac Studio に MC パラメータ設定をアップロードすると、不揮発性メモリに保存された設定がアップロードされます。
- 不揮発性メモリに保存された設定はダウンロード後や電源投入時にメインメモリに反映されます。
- 設定内容が正しければ、MC 機能モジュールはメインメモリの設定内容で制御を実行します。
- 一部のパラメータは、モーション制御システム変数として設定内容を参照できます。
- MC パラメータ設定のアップロードおよびダウンロードは、CPU ユニットのモードや MC 機能モジュールの状態にかかわらず実行可能です。
- ダウンロードを開始すると、動作中の軸は即停止します。

バージョン関連情報

ユニットバージョン Ver.1.13 以降の CPU ユニットと Sysmac Studio Ver.1.17 以降の組み合わせで、ダウンロード中も I/O デバイスへの指令を継続することが可能です。
ユニットバージョン Ver.1.12 以前の CPU ユニットの場合は、ダウンロード中の I/O デバイスへの指令は、停止します。
ダウンロード開始時に、I/O デバイスへ指令を停止するか、継続するかの設定は、『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。

I/O デバイスへの指令を停止する場合

ダウンロード中は、サーボ OFF となり、軸状態は、「Disable」（軸無効）となります。

I/O デバイスへの指令を継続する場合

ダウンロード中は、ダウンロード直前の MC_Power 命令によるサーボ ON 状態、および MC_SetTorqueLimit 命令による出力トルクの制限を継続します。
また、ダウンロードによりユーザプログラムが更新され、MC_Power 命令および MC_SetTorqueLimit 命令が削除された場合も、サーボ ON 状態およびトルク制限を継続します。

MC パラメータ設定の命令による書き換え

- モーション制御命令の MC_Write（MC 設定書込）命令、MC_ChangeAxesInGroup（軸グループ構成軸書込）命令、および MC_WriteAxisParameter（軸パラメータ書込）命令などは、メインメモリにある一部の MC パラメータ設定の内容を、ユーザプログラムの実行中に変更することができます。
- 指定された設定値が範囲外であれば、命令の出力変数「Error（エラー）」が TRUE になり、MC パラメータ設定は変更されません。
- 変更した MC パラメータ設定の内容が有効になるタイミングは以下のいずれかのときです。
  a）軸や軸グループが停止している状態で、軸指令や軸グループ指令の命令を起動したとき。
  b）命令のバッファモード選択をアボーティング（中断）にして、多重起動したとき。
- MC_Write（MC 設定書込）命令、MC_ChangeAxesInGroup（軸グループ構成軸書込）命令、および MC_WriteAxisParameter（軸パラメータ書込）命令などの詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』を参照してください。

使用上の注意

- MC パラメータ設定の MC_Write（MC 設定書込）命令などの命令による変更は、CPU ユニット内のメインメモリに保存されます。CPU ユニット内蔵の不揮発性メモリには保存されません。
  そのため、電源の再投入または Sysmac Studio からのダウンロードにより不揮発性メモリのパラメータ設定内容に戻ります。また、Sysmac Studio でこれらのデータをアップロードすることはできません。
  不揮発性メモリに保存する必要がある場合は、Sysmac Studio でパラメータ設定を変更し、CPU ユニットへダウンロードしてください。
- ダウンロード後も MC_Power（運転可）命令、および MC_SetTorqueLimit（トルク制限）命令を維持する場合、命令の入力接点を保持属性を保持にしてください。
- MC パラメータ設定を変更できる命令は以下のとおりです。
  MC_Write（MC 設定書込）命令
  MC_ChangeAxesInGroup（軸グループ構成軸書込）命令
  MC_ChangeAxisUse（軸使用変更）命令
  MC_WriteAxisParameter（軸パラメータ書込）命令
- 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。

5-2 軸パラメータ

軸パラメータは、MC 機能モジュールで制御する軸に関して、最高速度やジャーク、原点復帰といった軸動作の設定を行います。
軸パラメータは各形式の制御軸最大数分あります。制御軸最大数は、形式により異なります。
詳細は、「1-4-2 性能仕様 (P.1-7)」を参照してください。
なお、各軸のパラメータ設定は同じなので、ここでは、1 軸分だけについて説明します。

5-2-1 軸パラメータの一覧

軸パラメータは Sysmac Studio を使用して軸ごとに設定します。

区分	パラメータ名称	一時変更 *1		変数読出 *2	参照ページ
		可否	使用命令		
軸基本設定	軸番号			可	P.5-6
	モーション制御 *3			可 *4	
	軸使用	可 *5	MC_ChangeAxisUse	可	
	軸種別			可	
	制御機能 *6				
	入力デバイス／出力デバイス			可	
単位変換設定	表示単位	可 *7	MC_WriteAxisParameter *8	可	P.5-11
	モータ 1 回転のパルス数			可	
	モータ 1 回転の移動量			可	
	減速機使用 *9				
	ワーク側 1 回転の移動量 *9				
	ワーク側ギア比 *9				
	モータ側ギア比 *9				
動作設定	最高速度	可 *7	MC_WriteAxisParameter *8		P.5-19
	起動速度 *10				
	ジョグ最高速度				
	最大加速度				
	最大減速度				
	加減速オーバー				
	反転時動作				
	速度警告値	可	MC_Write MC_WriteAxisParameter *8		
	加速度警告値				
	減速度警告値				
	正方向トルク警告値 *11				
	負方向トルク警告値 *11				
	インポジション幅	可 *12			
	インポジションチェック時間	可			
	フィードバック速度フィルタ時定数	可 *7	MC_WriteAxisParameter *8		
	原点位置範囲				

区分	パラメータ名称	一時変更 *1		変数読出 *2	参照ページ
		可否	使用命令		
拡張動作設定	即停止入力停止方法	可 *7	MC_WriteAxisParameter *8		P.5-22
	限界入力停止方法				
	ドライバエラーリセット監視時間				
	正方向トルク制限上限値				
	負方向トルク制限上限値				
	即停止入力論理反転 *10				
	正方向限界入力論理反転 *10				
	負方向限界入力論理反転 *10				
	原点近傍入力論理反転 *10				
リミット設定	ソフトウェアリミット機能	可	MC_Write MC_WriteAxisParameter *8		P.5-23
	正方向ソフトウェアリミット				
	負方向ソフトウェアリミット				
	位置偏差オーバー値				
	位置偏差警告値				
位置カウント設定	カウントモード	可 *7	MC_WriteAxisParameter *8		P.5-23
	リングカウンタ上限設定値			可 *4	
	リングカウンタ下限設定値			可 *4	
	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				P.5-26
	リングカウンタ下限設定値				
	PDS 状態制御方式 *4				
原点復帰設定	原点復帰動作	可 *7	MC_WriteAxisParameter *8		P.5-26
	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				

*1. 命令による一時変更の可否を示します。
*2. モーション制御システム変数としてユーザプログラムからの読み出し可否を示します。
*3. 形 NX701 CPU ユニットの場合に設定します。
*4. 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することができます。
*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一時変更が可能となります。
*6. 形 NX102 および形 NX1P2 CPU ユニットの場合に設定します。
*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一時変更が可能となります。
*8. MC_WriteAxisParameter 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することができます。

MC_WriteAxisParameter 命令で一時変更可能なパラメータは、MC_ReadAxisParameter 命令で読み出すことができます。

*9. 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することができます。

*10. 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することができます。

*11. トルク制御時にだけ有効です。

*12. MC_Write 命令での一時変更は、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可能です。
MC_WriteAxisParameter 命令での一時変更は、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可能です。

軸パラメータの設定方法については、「3-2 軸の設定手順 (P.3-9)」を参照してください。
モーション制御システム変数については、「6-6 モーション制御システム変数 (P.6-19)」を参照してください。
MC_Write（MC 設定書込）命令など各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 モーション編(SBCE-434)』を参照してください。

5-2-2 軸基本設定

軸を使用／不使用と、使用する場合の軸種別、および EtherCAT スレーブデバイスのノードアドレスを設定します。

パラメータ名称	機能	設定範囲	初期値
軸番号	軸の論理番号を設定します。システム定義変数の変数名「_MC_AX[0-255]」、「_MC1_AX[0-255]」、「_MC2_AX[0-255]」の番号は、ここで設定した番号となります。	−	−
モーション制御 *1	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）のどちらかのタスクに割り付けます。 1: プライマリ定周期タスク 2: 定周期タスク（実行優先度 5）	1〜2	1
軸使用	軸を使用するかしないかを設定します。 *2 0: 未作成軸 1: 未使用軸 *3 2: 使用軸	0〜2	0
軸種別	軸種別を設定します。 仮想軸では入出力配線が不要となります。 0: サーボ軸 1: エンコーダ軸 2: 仮想サーボ軸 3: 仮想エンコーダ軸	0〜3	2
制御機能 *4	制御する軸の機能を選択します。 *5 0: すべて 1: 単軸位置制御のみ	0, 1	0
入力デバイス／出力デバイス	軸に割り付ける EtherCAT スレーブデバイスのノードアドレスを選択します。 *6 軸種別が［仮想軸］の場合は選択できません。	0〜65535	−

*1. 形 NX701 CPU ユニットの場合に設定します。

*2. 未使用軸および未作成軸でモーション制御命令を実行すると、「Busy」（実行中）が TRUE となります。「Execute」（起動）あるいは「Enable」（有効）を FALSE にすると、「Busy」（実行中）も FALSE となります。
［未使用軸］に設定すると、1 つのユーザプログラムを軸構成が異なる装置で共用する場合などで、使用しない軸のプログラムを削除する必要がありません。

*3. 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 以降の組み合わせの場合、Sysmac Studio の設定画面では、［未使用軸（使用軸に切替可）］、［未使用軸（使用軸に切替不可）］と表示されます。

［未使用軸（使用軸に切替可）］に設定すると、軸パラメータを設定することができ、MC_ChangeAxisUse（軸使用変更）命令で、一時的に［軸使用］を変更できます。
軸使用変更機能については、「9-8-10 軸使用変更機能 (P.9-76)」を参照してください。

*4. 形 NX102 および形 NX1P2 CPU ユニットの場合に設定します。
*5. モーション制御軸として使用する場合は、［すべて］を選択します。単軸位置制御軸として使用する場合は、［単軸位置制御のみ］を選択してください。
*6. NX シリーズ 位置インタフェースユニットの場合は、EtherCAT カプラユニットのノードアドレスおよび位置インタフェースユニットの NX ユニット番号を選択します。

使用上の注意

絶対値エンコーダを使用する場合
［絶対値エンコーダ原点位置オフセット］が、軸番号と関連付けて軸ごとに CPU ユニット内部に保存されています。
軸番号を変更すると、保存されたオフセット量が失われます。軸番号を変更した場合は、再度原点復帰を行ってください。

軸番号

軸番号に設定できる番号は、制御軸最大数までです。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□□	形 NJ501-□4□□	形 NJ501-□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。

モーション制御

形 NX701 CPU ユニットの場合、使用する軸を、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）のどちらかのタスクに割り付けます。

参考

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 ユニットには、プライマリ定周期タスクしかありませんので、［モーション制御］設定はありません。

軸種別

軸種別で選択できる軸種別の内容を、次の表に説明します。

軸種別	内容
サーボ軸	EtherCAT スレーブサーボドライバや NX シリーズ 位置インタフェースユニット *1 を使用する軸です。実際のサーボドライバなどと割り付けをおこなって使用します。 1 台のサーボモータを 1 軸として使用します。 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、パルス出力ユニットとデジタル入力ユニットなど、複数のデバイスを 1 軸に割り付けできます。
仮想サーボ軸	MC 機能モジュール内の仮想的な軸です。実際のサーボドライバは使用しません。 同期制御の主軸などとして使用します。
エンコーダ軸	EtherCAT スレーブエンコーダ入力ターミナルや NX シリーズ 位置インタフェースユニット *1 を使用する軸です。実際のエンコーダ入力ターミナルなどと割り付けをおこなって使用します。 1 台のエンコーダ入力ターミナルに 2 個のカウンタがある場合は、各々 1 個のカウンタを 1 軸として使用します。
仮想エンコーダ軸	エンコーダの動作を仮想的に行う軸です。エンコーダの実物がない場合など、エンコーダ軸の代わりとして一時的に使用します。*2

*1. 制御対象機器は、「1-4-3 機能仕様 (P.1-10)」を参照してください。
*2. 仮想エンコーダ軸の現在位置を更新するモーション制御命令と組み合わせて使用します。
この命令に未対応のバージョンの MC 機能モジュールでは、エンコーダ軸の代わりとして使用することはできません。

● 仮想サーボ軸

仮想サーボ軸は物理的なエンコーダや外部入出力信号がありません。
そのため仮想サーボ軸はサーボ軸に対して以下の違いがあります。

- 常時サーボ ON 状態として扱われます。
- フィードバック現在位置 ＝ 指令現在位置となります。
  ただし、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。
- フィードバック現在速度 ＝ 指令現在速度となります。
  ただし、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。
- 外部入力信号は使用できません。

- MC_Home（原点復帰）命令または MC_HomeWithParameter（パラメータ指定原点復帰）命令で原点復帰する場合、軸パラメータの原点復帰動作の設定に関わらず［原点プリセット］として扱われます。
- ラッチ機能を使用するモーション制御命令では、トリガ入力条件のモードをコントローラモードに設定する必要があります。ドライバモードに設定しても異常となりませんが、ラッチのトリガが発生しないため命令の実行は完了しません。
  ラッチ機能を使用する命令には、MC_TouchProbe（外部ラッチ有効）命令、MC_MoveFeed（割り込み定寸位置決め）命令、および MC_MoveLink（台形パターンカム）命令などがあります。
- 即停止入力や限界入力信号がないため、即停止入力や正方向／負方向限界入力検出の異常は発生しません。

● エンコーダ軸／仮想エンコーダ軸

エンコーダ軸／仮想エンコーダ軸は、サーボ軸／仮想サーボ軸に対して以下の違いがあります。

- 指令位置を持ちません。フィードバック位置だけを持ちます。
- 動作系のモーション制御命令は使用できません。

使用上の注意

単軸位置制御軸の軸種別にエンコーダ軸／仮想エンコーダ軸は、指定できません。

制御機能

制御する軸の機能を選択します。
形 NX102 または形 NX1P2 CPU ユニットを使用する場合だけ、選択することが可能です。

制御機能	内容
すべて	全機能が使用可能な軸である「モーション制御軸」に設定されます。
単軸位置制御のみ	単軸位置制御だけが使用可能な「単軸位置制御軸」に設定されます。

制御機能の選択により、使用可能な軸種別は、以下のようになります。

軸種別	制御機能	
	［すべて］	［単軸位置制御のみ］
サーボ軸	○	○
仮想サーボ軸	○	○
エンコーダ軸	○	×
仮想エンコーダ軸	○	×

また、動作可能な機能は、以下のようになります。

モーション制御機能	制御機能	
	［すべて］	［単軸位置制御のみ］
単軸位置制御	○	○
単軸同期制御	○	×
単軸速度制御	○	○ *1
単軸トルク制御	○	×
多軸協調制御	○	×

*1. MC_MoveVelocity（速度制御）命令だけが使用できます。

モーション制御機能の詳細は、「第 9 章 モーション制御機能 (P.9-1)」を参照してください。

参考

形 NX102-90□□または形 NX1P2-9□□□□□□を使用する場合は、［単軸位置制御のみ］を選択してください。

入力デバイス／出力デバイス

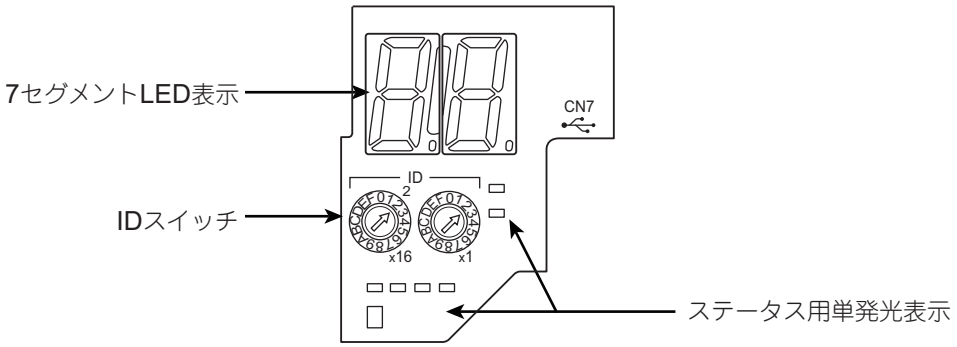
軸種別がサーボ軸またはエンコーダ軸の場合、軸に割り付ける EtherCAT デバイスのノードアドレスを選択します。
NX シリーズ 位置インタフェースユニットの場合は、EtherCAT カプラユニットのノードアドレスおよび位置インタフェースユニットの NX ユニット番号を選択します。
軸種別が［仮想軸］の場合は選択できません。

使用上の注意

- オムロン製サーボドライバ 1S シリーズおよび G5 シリーズは、ノードアドレスをドライバ前面のノードアドレススイッチにより固定値を設定することができます。
  このノードアドレススイッチを「00」と設定した場合、ノードアドレスは Sysmac Studio の EtherCAT エディタで設定した値に従います。
  接続されるすべてのドライバでノードアドレススイッチを「00」と設定した場合、ドライバの接続位置を変更しても異常とはなりません。装置の各制御対象に対して特定のサーボドライバを割り付ける場合は、ドライバのノードアドレスはノードアドレススイッチにより設定するようにしてください。
- サーボドライバのノードアドレススイッチの設定値はサーボドライバの制御電源の投入時に一度だけ読み込まれます。
  電源投入後に変更しても制御に反映されず、次回の電源投入時に有効となります。
  電源投入後はノードアドレススイッチの値を変更しないでください。
- ノードアドレスが重複している場合は異常となります。

参考

- EtherCAT デバイスのノードアドレスの設定例として、オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプの場合を下記に示します。



- サーボドライバの表示部のロータリスイッチを使用して、EtherCAT のノードアドレスを設定します。

ロータリスイッチの設定値	ノードアドレスの設定値	
	オムロン製スレーブ	他社製スレーブ *1
00	Sysmac Studio による設定値 (1～512) *2	Sysmac Studio による設定値 (1～512) *2
01～FF	ノードアドレススイッチの設定値	

*1. 他社製スレーブではスレーブ側の設定有無に関係なく、Sysmac Studio による設定値が有効となります。
*2. NJ シリーズ CPU ユニットの場合、設定値は「1～192」です。ただし、形 NJ101 CPU ユニットだけ接続できるスレーブの最大数は、64 スレーブです。
形 NX502 CPU ユニットの場合、設定値は「1～256」で、接続できるスレーブの最大数は、256 スレーブです。
形 NX102 CPU ユニットの場合、設定値は「1～192」で、接続できるスレーブの最大数は、64 スレーブです。
形 NX1P2 CPU ユニットの場合、設定値は「1～192」で、接続できるスレーブの最大数は、16 スレーブです。

5-2-3 単位変換設定

位置の単位系を設定します。

パラメータ名称	機能	設定範囲	初期値
表示単位	指令位置の単位を設定します。 • pulse • mm • μm • nm • degree • inch	0～5	0
モータ 1 回転のパルス数 *1	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。 *2 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。	1～4294967295	10000
モータ 1 回転の移動量 *3	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。	0.000000001～4294967295	10000
減速機使用 *4 *5	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	TRUE, FALSE	FALSE

パラメータ名称	機能	設定範囲	初期値
ワーク側 1 回転の移動量 *4 *6 *7	ワーク側が 1 回転した場合の移動量を設定します。	倍精度実数型の正数	10000
ワーク側ギア比 *4 *6	ワーク側のギア比を設定します。	1～4294967295	1
モータ側ギア比 *4 *6	モータ側のギア比を設定します。	1～4294967295	1

*1. 電子ギア比（単位変換の式）の分子です。
*2. たとえば、エンコーダ分解能が 10,000 パルス／回転の場合は、「10,000」を設定してください。
*3. 減速機使用を［使用しない］と設定した場合の電子ギア比（単位変換の式）の分母です。
減速機使用を［使用する］と設定した場合は、無効となります。
*4. 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することができます。
*5. 減速機使用を［使用する］に設定する場合、単位変換設定の設定には条件があります。
条件については、「減速機使用の条件 (P.5-13)」を参照してください。
*6. 減速機使用を［使用する］と設定した場合に有効となります。
*7. 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。

使用上の注意

- ［カウントモード］を［ロータリモード］で使用する場合は、減速機使用を［使用する］に設定してください。
減速機使用を［使用しない］と設定した場合、リングカウンタの 1 周をパルス単位に変換するときの演算誤差により、リングカウンタ 1 周あたりのパルス数が期待した整数とはならないことがあります。期待した整数にならない場合、位置ずれの要因となります。
- 表示単位の変換を除き、単位変換設定を変更した場合は、物理的な機械位置と MC 機能モジュールの指令現在位置との関係にずれが生じます。そのため、単位変換設定を変更した場合は、原点復帰命令を実行し、再度、原点を確定してください。

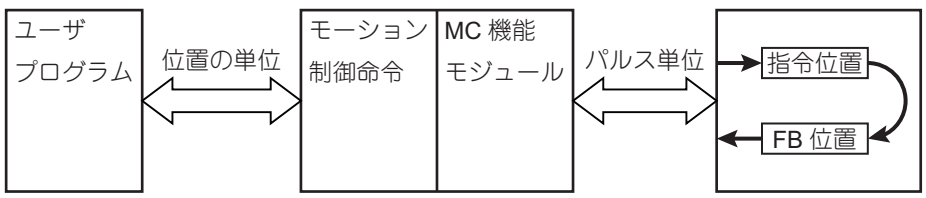
サーボドライバやエンコーダ入力ターミナルと MC 機能モジュール間は、基本的にパルス単位の位置が使用されます。
モーション制御命令で使用する位置は、実際の動作を分かりやすくするために、mm や degree などの表示単位を使用します。



表示単位と電子ギア（単位変換の式）の設定により、パルス単位の表記を mm や degree の表記に変換できます。

参考

仮想サーボ軸の場合、指令現在値をパルス単位に変換し、その値を再度、表示単位に変換した値をフィードバック現在値とします。



表示単位

表示単位は、Sysmac Studio に表示する表示単位の設定です。表示は、位置の表示単位です。
設定できる単位について説明します。

設定可能な単位	説明
pulse	パルス単位で表現する場合に使用します。
mm	比較的長い距離を直動する場合に使用します。
μm	精度の高い直動を行う場合に使用します。
nm	μm よりも精度の高い直動を行う場合に使用します。
degree	ロータリテーブルなどの回転する軸に使用します。
inch	直動などで使用します。

減速機使用の条件

減速機使用を［使用する］に設定する場合、単位変換設定の設定には、次の条件を満たす必要があります。

- 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が、0.000000001 以上、4294967295 以下であること。

［カウントモード］が［ロータリモード］の場合は、さらに次の条件も満たす必要があります。

- モータ 1 回転のパルス数×モータ側ギア比が、1099511627775 以下であること。

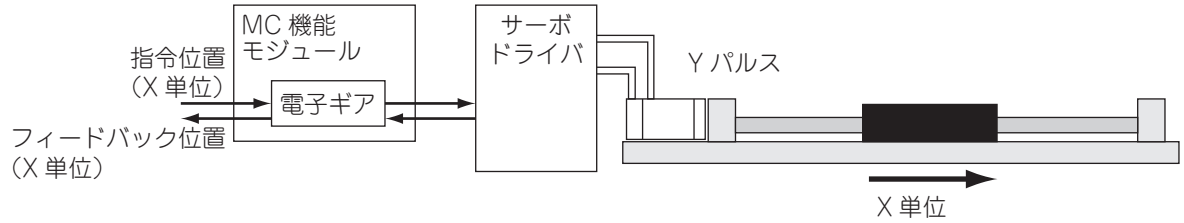
ただし、これらの条件のワーク側ギア比とモータ側ギア比の値は、両者の最大公約数で割った値として計算しています。
したがって、設定した値のまま計算すると条件を満たさない場合でも、実際には条件を満たす場合があります。

● 例

ワーク側ギア比が「4」、モータ側ギア比が「6」の場合、それぞれワーク側ギア比が「2」、モータ側ギア比が「3」とみなして計算します。

電子ギア比（単位変換の式）

MC 機能モジュールでは、表示単位とパルス単位の関係を設定するために電子ギアを使用します。
Sysmac Studio を使用して、電子ギア比を設定します。



指令位置（パルス）値 = 指令位置（X 単位）× 電子ギア比

● 減速機使用を［使用しない］場合

減速機使用を［使用しない］と設定した場合、電子ギア比は、次の式によって表されます。

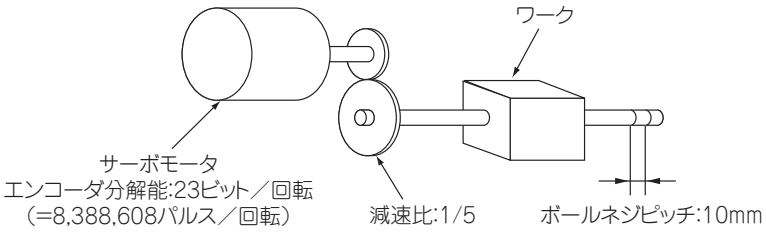
$$電子ギア比 = \frac{モータ1回転のパルス数^{*1}（Yパルス）}{モータ1回転の移動量^{*2}（X単位）}$$

*1. エンコーダ軸の場合は、エンコーダ 1 回転あたりのパルス数となります。
*2. エンコーダ軸の場合は、エンコーダ 1 回転あたりの移動量となります。

使用上の注意

MC 機能モジュールやモーション制御命令で使用する位置は、電子ギアで単位変換した値を使用します。
モーション制御命令では目標位置を LREAL 型で指定しますが、電子ギアでパルス単位に変換した後の指令位置が 40 ビットを超える場合には、命令で異常が発生します。

例として、オムロン製サーボモータ 1S シリーズの 23 ビット絶対値エンコーダ付を使用する場合で説明します。
機械系は、減速機の減速比が 1/5 で、ボールネジが 1 回転するとワークが 10mm 移動するとします。



表示単位は、mm とします。モータ 1 回転のパルス数には、サーボモータのエンコーダ分解能を設定します。
減速比が 1/5 の減速機を使用しているため、サーボモータが 5 回転するとボールネジは 1 回転します。このときワークは 2mm （=10mm×1/5）移動するため、モータ 1 回転の移動量には、「2」を設定します。

設定項目	設定内容
表示単位	mm
モータ 1 回転のパルス数	8,388,608
モータ 1 回転の移動量	2

この設定により、ユーザプログラムでの位置の指令単位は 1（mm）となります。
たとえば、100.5mm の位置へ絶対値位置決めする場合は、MC_MoveAbsolute（絶対値位置決め）命令の入力変数「Position（目標位置）」に「100.5」を設定します。

参考

例と同じ機械で減速比が 1/9 の減速機を使用する場合
サーボモータが 1 回転した場合のワークの移動量は、10mm×1/9 = 1.111･･･mm となり割り切れません。
このように割り切れない数字となる場合は、［モータ 1 回転のパルス数］と［モータ 1 回転の移動量］に同じ係数をかけた値を設定します。
減速比が 1/9 なので、係数を「9」とします。
- ［モータ 1 回転のパルス数］: 75,497,472 (=8,388,608×9)
- ［モータ 1 回転の移動量］: 10 (=10×1/9×9)

● 減速機使用を［使用する］で、［カウントモード］が［リニアモード］の場合

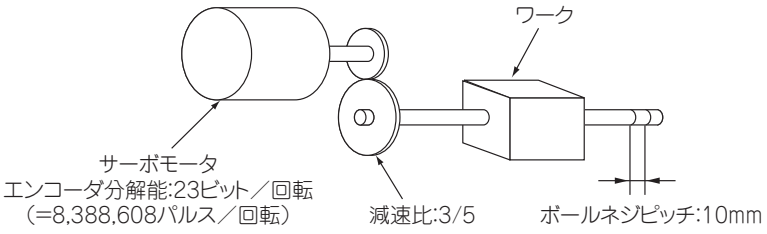
減速機使用を［使用する］と設定し、かつ［カウントモード］が［リニアモード］の場合、電子ギア比は、次の式によって表されます。

$$\text{電子ギア比} = \frac{\text{モータ 1 回転のパルス数} \times \text{モータ側ギア比}}{\text{ワーク 1 回転の移動量} \times \text{ワーク側ギア比}}$$

使用上の注意

MC 機能モジュールやモーション制御命令で使用する位置は、電子ギアで単位変換した値を使用します。
モーション制御命令では目標位置を LREAL 型で指定しますが、電子ギアでパルス単位に変換した後の指令位置が 40 ビットを超える場合には、命令で異常が発生します。

例として、オムロン製サーボモータ 1S シリーズの 23 ビット絶対値エンコーダ付を使用する場合で説明します。
機械系は、減速機の減速比が 3/5 で、ボールネジが 1 回転するとワークが 10mm 移動するとします。



表示単位は、mm とします。モータ 1 回転のパルス数には、サーボモータのエンコーダ分解能を設定します。
ワーク 1 回転の移動量には、ボールネジピッチの 10mm を設定します。
減速比が 3/5 の減速機を使用しているため、サーボモータが 5 回転するとボールネジは 3 回転します。この場合の減速比の設定は、ワーク側ギア比に「3」を、モータ側ギア比に「5」を設定します。

設定項目	設定内容
表示単位	mm
モータ 1 回転のパルス数	8,388,608
ワーク 1 回転の移動量	10
ワーク側ギア比	3
モータ側ギア比	5

この設定により、ユーザプログラムでの位置の指令単位は 1(mm)となります。
たとえば、100.5mm の位置へ絶対値位置決めする場合は、MC_MoveAbsolute（絶対値位置決め）命令の入力変数「Position（目標位置）」に「100.5」を設定します。

● 減速機使用を［使用する］で、［カウントモード］が［ロータリモード］の場合

減速機使用を［使用する］と設定し、かつ［カウントモード］が［ロータリモード］の場合、電子ギア比は、次の式によって表されます。
［ロータリモード］の場合、ワーク 1 回転の移動量は、「｜リングカウンタ上限値 - リングカウンタ下限値｜」で、自動的に設定されます。

$$\text{電子ギア比} = \frac{\text{モータ 1 回転のパルス数} \times \text{モータ側ギア比}}{\text{ワーク 1 回転の移動量} \times \text{ワーク側ギア比}}$$

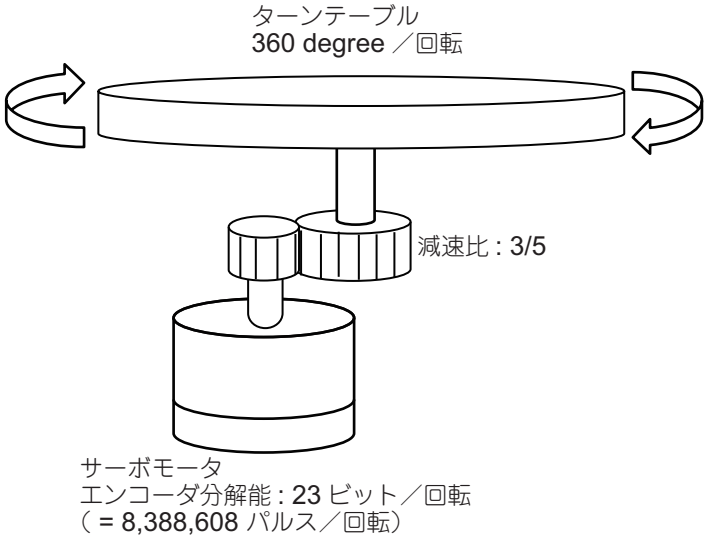
$$= \frac{\text{モータ 1 回転のパルス数} \times \text{モータ側ギア比}}{|\text{リングカウンタ上限値} - \text{リングカウンタ下限値}| \times \text{ワーク側ギア比}}$$



使用上の注意

MC 機能モジュールやモーション制御命令で使用する位置は、電子ギアで単位変換した値を使用します。
モーション制御命令では目標位置を LREAL 型で指定しますが、電子ギアでパルス単位に変換した後の指令位置が 40 ビットを超える場合には、命令で異常が発生します。

例 1 として、オムロン製サーボモータ 1S シリーズの 23 ビット絶対値エンコーダ付を使用する場合で説明します。
機械系は、減速機の減速比が 3/5 で、ターンテーブルが 1 回転するとワークが 360degree 移動します。



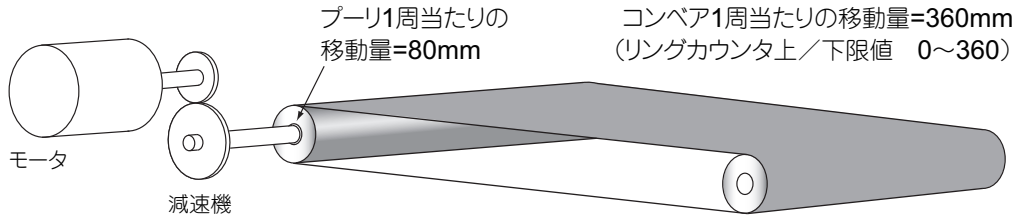
表示単位は、degree とします。モータ 1 回転のパルス数には、サーボモータのエンコーダ分解能を設定します。
ワーク 1 回転の移動量は、「｜リングカウンタ上限値 - リングカウンタ下限値｜」で、自動的に設定されます。
減速比が 3/5 の減速機を使用しているため、サーボモータが 5 回転するとターンテーブル（ワーク）は 3 回転します。この場合の減速比の設定は、ワーク側ギア比に「3」を、モータ側ギア比に「5」を設定します。

設定項目	設定内容
表示単位	degree
モータ 1 回転のパルス数	8,388,608
ワーク 1 回転の移動量	｜リングカウンタ上限値 - リングカウンタ下限値｜
ワーク側ギア比	3
モータ側ギア比	5

この設定により、ユーザプログラムでの位置の指令単位は 1(degree)となります。
たとえば、100.5degree の位置へ絶対値位置決めする場合は、MC_MoveAbsolute（絶対値位置決め）命令の入力変数「Position（目標位置）」に「100.5」を設定します。

例 2 として、オムロン製サーボモータ 1S シリーズの 23 ビット絶対値エンコーダ付を使用する場合で説明します。
機械系は、減速機の減速比が 3/5 で、プーリが 1 回転するとコンベアが 80mm 移動します。コンベアは 360mm で 1 周します。

サーボモータ
エンコーダ分解能: 23ビット／回転
（=8,388,608パルス／回転）



減速機
比率: 3/5

メカ的な機構
比率: 80/360

最終的な比率:

$$= \frac{3 \times 80}{5 \times 360} = \frac{\text{ワーク側ギア比}}{\text{モータ側ギア比}}$$

表示単位は、mm とします。モータ 1 回転のパルス数には、サーボモータのエンコーダ分解能を設定します。
コンベア 1 回転の移動量は、「｜リングカウンタ上限値 - リングカウンタ下限値｜」で、自動的に設定されます。

設定項目	設定内容
リングカウンタ上限値	360
リングカウンタ下限値	0

減速比が 3/5 の減速機を使用しているため、サーボモータが 5 回転するとプーリは 3 回転します。また、プーリ 1 周あたりのコンベアの移動量が 80mm、コンベアの 1 周が 360mm のため、減速比の設定は、ワーク側ギア比に「3×80」の「240」を、モータ側ギア比に「5×360」の「1800」を設定します。

設定項目	設定内容
表示単位	mm
モータ 1 回転のパルス数	8,388,608
ワーク 1 回転の移動量	｜リングカウンタ上限値 - リングカウンタ下限値｜
ワーク側ギア比	240
モータ側ギア比	1800

参考

- プーリの 1 回転あたりの移動量が 80.1mm など整数でない場合は、整数となる係数「10」を掛けた比率を使用して、ワーク側ギア比、モータ側ギア比を算出してください。上記の例を元にすると、ワーク側ギアの比率が「3×80.1×10」の「2403」となり、モータ側ギアの比率は「5×360×10」の「18000」となります。
- 減速機が多段な場合も上記の例 2 と同様に、複数の比率を掛け合わせた最終的な比率をワーク側ギア比、モータ側ギア比として設定してください。

5-2-4 動作設定

制御する装置の仕様にあわせて、最高速度や最大加減速度などの軸の動作について設定します。

パラメータ名称	機能	設定範囲	初期値
最高速度	各軸の最高速度を設定します。*1 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。(単位: 指令単位／s)	倍精度実数型の正数 *2	400,000,000
起動速度 *3	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。(単位: 指令単位／s)	倍精度実数型の正数	0
ジョグ最高速度	各軸のジョグ最高速度を設定します。*4 最高速度を超えないように設定してください。(単位: 指令単位／s)	倍精度実数型の正数	1,000,000
最大加速度	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。(単位: 指令単位／s^2)	倍精度実数型の正数、0	0
最大減速度	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。(単位: 指令単位／s^2)	倍精度実数型の正数、0	0
加減速オーバー	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)。*5 1: 加減速度を急激にする。 2: 異常として停止する。*6	0～2	0
反転時動作	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。*7 0: 減速停止 1: 即停止	0～1	0
速度警告値	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。(単位: %)	0～100	0
加速度警告値	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。(単位: %)	0～100	0
減速度警告値	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。(単位: %)	0～100	0
正方向トルク警告値 *8	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。(単位: %)	0～1000	0
負方向トルク警告値 *8	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。(単位: %)	0～1000	0

パラメータ名称	機能	設定範囲	初期値
フィードバック速度フィルタ時定数	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。（単位: ms） 軸の速度が遅い場合のフィードバック現在速度のばらつきを抑えたい場合に使用します。	0～100	0
インポジション幅 *9	位置決め完了幅を設定します。 （単位: 指令単位）	倍精度実数型の正数、0	10
インポジションチェック時間 *9	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だけ位置決め完了のチェックを継続し、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。（単位: ms）	0～10,000	0
原点位置範囲	原点位置検出幅を設定します。 （単位: 指令単位）	倍精度実数型の正数、0	10

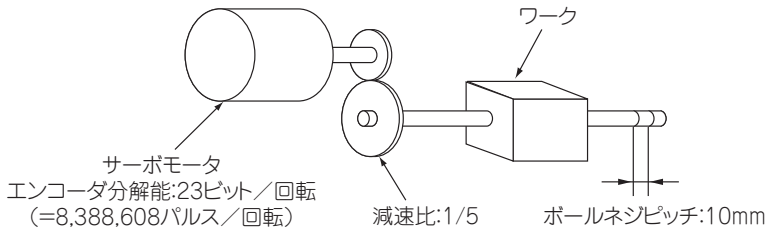
*1. 最高速度を超える速度指令値を指定した場合は、最高速度で指令します。補間制御動作時も有効です。
軸に接続するサーボモータ／ドライバの出力可能なモータ最高速度（最高回転数）を超える値は、設定しないでください。接続するサーボモータ／ドライバの仕様を確認の上、設定してください。
*2. 設定できる最大値は、パルス単位に変換した値で次のとおりです。
ユニットバージョン Ver.1.11 以降の CPU ユニットは、2,147,483,647 [pulse/s]です。ユニットバージョン Ver.1.10 以前で Ver.1.03 以降の CPU ユニットは、500,000,000 [pulse/s]、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]です。
*3. 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することができます。
*4. ジョグ最高速度を超える速度指令値を指定した場合は、ジョグ最高速度で指令します。
*5. 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、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44)」を参照してください。
*6. 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、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44)」を参照してください。
*7. 反転時動作の詳細は、「9-5-6 モーション制御命令の再起動 (P.9-39)」、および「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44)」を参照してください。
*8. トルク制御時にだけ有効です。
*9. インポジションチェックは、MC 機能モジュール側で処理しています。サーボドライバ側の機能は使用していません。

最高速度

最高速度の設定例を説明します。

● 最高速度の設定例

前述の「減速機使用を［使用しない］場合 (P.5-13)」と同じ機械で、サーボモータの最大回転数が 6,000r/min の場合について説明します。



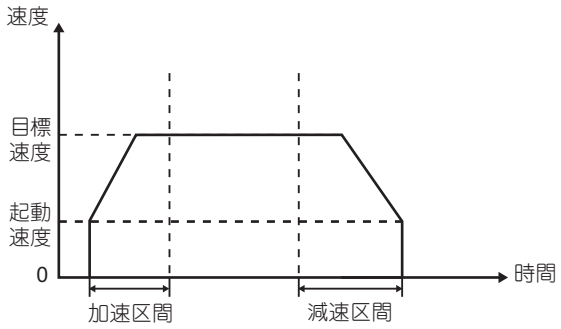
最大回転数:6,000r/min、減速比:1/5、ボールネジピッチ:10mm の各条件から計算すると、6,000r/min × 1/5 × 10mm = 12,000mm/min = 200mm/s となるため、最高速度には「200」を設定します。
初期値の 400,000,000 はモータの最大回転数を超えているため、必ず設定値を変更してください。

設定項目	設定内容
表示単位	mm
モータ 1 回転のパルス数	8,388,608
モータ 1 回転の移動量	2
最高速度	200

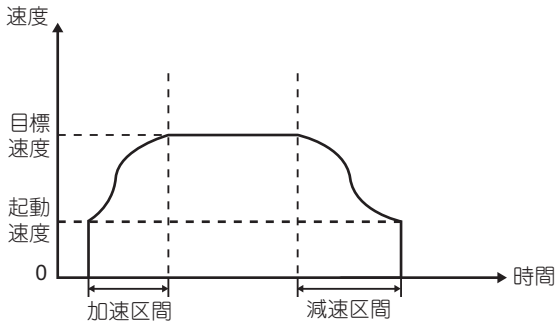
起動速度

サーボモータを使用する場合は、「0」としてください。
ステッピングモータを使用する場合は、起動時の脱調を防ぐために、通常、最大自起動周波数の 1/10～1/2 の値を設定します。負荷状況によって変わりますので、詳しくはステッピングモータのマニュアルを確認してください。

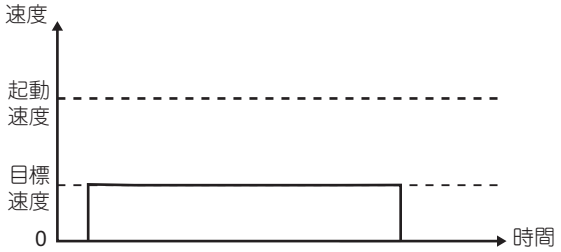
ジャークを指定していないとき



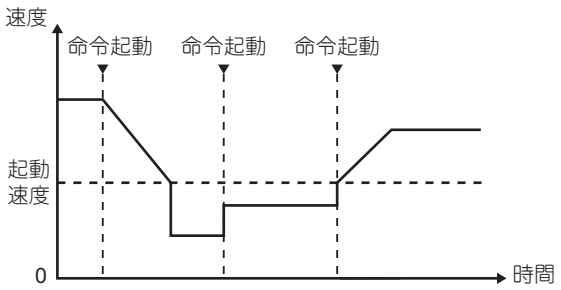
ジャークを指定したとき



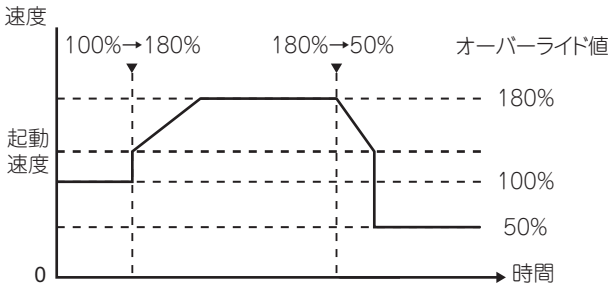
目標速度が起動速度以下の場合は、加減速せずに目標速度で動作します。



動作中にモーション制御命令の再起動や多重起動により、目標速度を変更した場合にも起動速度が適用されます。目標速度が起動速度を超えている場合は、指定した加速度と減速度で加減速動作を行います。目標速度が起動速度以下の場合は、加減速せずに動作します。



MC_SetOverride（オーバーライド値設定）命令により、速度を変更した場合にも起動速度が適用されます。目標速度が起動速度を超えている場合は、指定した加速度と減速度で加減速動作を行います。目標速度が起動速度以下の場合は、加減速せずに動作します。



次の機能では、起動速度は適用されません。

- トルク制御モード
- サイクリック同期速度制御
- サイクリック同期位置決め
- 同期制御
  ただし、MC_GearOut（ギア動作解除）、MC_CamOut（カム動作解除）では、起動速度が適用されます。
- 多軸協調制御
  ただし、「各軸を減速停止」で減速する場合は、各軸の起動速度が適用されます。

5-2-5 拡張動作設定

各種入力信号が有効になったときの停止方法やトルク制限値などを設定します。

パラメータ名称	機能	設定範囲	初期値
即停止入力停止方法	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。 0: 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	0, 2, 3	0
限界入力停止方法	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: 指令値を即停止 1: 指令値を減速停止 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	0〜3	0
ドライバエラーリセット監視時間	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。	1〜1000	200
正方向トルク制限上限値	正方向トルク制限値の上限を設定します。*1 （単位: %）	0.0〜1000.0	300.0
負方向トルク制限上限値	負方向トルク制限値の上限を設定します。*1 （単位: %）	0.0〜1000.0	300.0
即停止入力論理反転 *2	即停止入力信号の論理の反転を設定します。 FALSE: 反転しない TRUE: 反転する	FALSE, TRUE	FALSE *3
正方向限界入力論理反転 *2	正方向限界入力信号の論理の反転を設定します。 FALSE: 反転しない TRUE: 反転する	FALSE, TRUE	FALSE *3

パラメータ名称	機能	設定範囲	初期値
負方向限界入力論理反転 *2	負方向限界入力信号の論理の反転を設定します。 FALSE: 反転しない TRUE: 反転する	FALSE, TRUE	FALSE *3
原点近傍入力論理反転 *2	原点近傍入力信号の論理の反転を設定します。 FALSE: 反転しない TRUE: 反転する	FALSE, TRUE	FALSE

*1. PDO マッピングに［正方向トルク制限値(60E0Hex)］および［負方向トルク制限値(60E1Hex)］をマッピングした場合に、これらの設定値が EtherCAT のプロセスデータ通信で送信されます。
MC_SetTorqueLimit（トルク制限）命令でトルク制限を有効にしている場合は、命令の入力変数に指定した値が送信されます。
*2. 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することができます。ユニットバージョン Ver.1.04 以前の CPU ユニットでは、論理を反転しません。
NX シリーズデジタル入力ユニットなど、入力信号の論理を設定できない機器を割り付けた場合に設定します。オムロン製サーボドライバ 1S シリーズなど、入力信号の論理を設定できる機器では、通常、［反転しない］に設定します。
*3. NX シリーズ パルス出力ユニットを割り付けた軸では、初期値は「TRUE」となります。

5-2-6 リミット設定

位置偏差に関するリミット機能とソフトウェアによるリミット機能を選択します。

パラメータ名称	機能	設定範囲	初期値
ソフトウェアリミット機能 *1	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	0～4	0
正方向ソフトウェアリミット	正方向側のソフトウェアリミットを設定します。 （単位: 指令単位）	倍精度実数型の負数、正数、0	2147483647
負方向ソフトウェアリミット	負方向側のソフトウェアリミットを設定します。 （単位: 指令単位）	倍精度実数型の負数、正数、0	-2147483648
位置偏差オーバー値	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。（単位: 指令単位）	倍精度実数型の正数、0	0
位置偏差警告値	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。（単位: 指令単位）	倍精度実数型の正数、0（位置偏差オーバー値以下）	0

*1. この機能は、カウントモードがリニアモードで原点確定している場合にだけ機能します。
ソフトウェアリミット機能の詳細については、「9-8-5 ソフトウェアリミット機能 (P.9-70)」を参照してください。

5-2-7 位置カウント設定

位置のカウントモードを設定します。
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。

パラメータ名称	機能	設定範囲	初期値
カウントモード	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）	0～1	0
リングカウンタ上限設定値	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。 （単位: 指令単位）	倍精度実数型の負数、正数、0	2147483647

パラメータ名称	機能	設定範囲	初期値
リングカウンタ下限設定値	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。 （単位: 指令単位）	倍精度実数型の負数、正数、0	-2147483648
エンコーダ種別	エンコーダ種別を選択します。*1 *2 0: インクリメンタルエンコーダ(INC) 1: 絶対値エンコーダ(ABS)	0～1	0

*1. 以下を使用する場合は、エンコーダ種別に［1: 絶対値エンコーダ(ABS)］を設定してください。
- オムロン製サーボモータ／ドライバ 1S シリーズを使用する場合
- オムロン製サーボモータ／ドライバ G5 シリーズで絶対値エンコーダ付きサーボモータを使用する場合
- オムロン製サーボモータ／ドライバ G5 シリーズでフルクローズ制御用に絶対値タイプの外部スケールを使用する場合
- オムロン製サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプで絶対値タイプの外部スケールを使用する場合

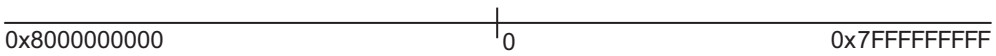
*2. オムロン製サーボモータ／ドライバ G5 シリーズ でフルクローズ制御用に外部スケールを使用する場合、またはオムロン製サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプを使用する場合は、それぞれ以下のとおり読み替えてください。
0: インクリメンタルタイプの外部スケール
1: 絶対値タイプの外部スケール

カウントモード

軸送りモードのことです。
軸ごとに指令位置のカウントモードを選択します。
軸送り範囲が有限長である［リニアモード］と、無限長である［ロータリモード］があります。

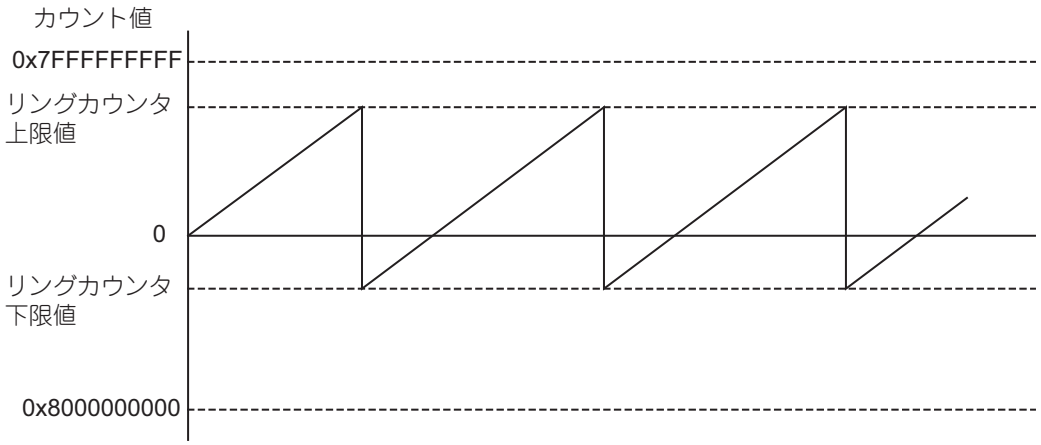
● リニアモード（有限長軸）

- 「0」を中心とする直線的な設定となります。
  X-Y ステージなど機械的な動作範囲が限定される装置などで使用します。
- 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。
- 相対値移動や絶対値移動などの目標位置を指定する位置決めでは、範囲を超える目標位置を指令できません。
- 速度制御や原点復帰およびトルク制御などの目標位置を指定しない動作で、動作範囲を超えると監視情報（指令位置オーバーフロー／アンダーフロー）となります。指令位置の出力は継続されますが、現在位置は更新されず、上限値または下限値で固定されます。
- 現在位置が固定の状態では、目標位置を指定しない動作で範囲内に入る方向への指令や停止については、起動することができます。範囲内から遠ざかる方向への指令は命令起動時に異常となります。
- オーバーフロー／アンダフロー状態を解除するまで、現在位置の更新はおこなわれません。



● ロータリモード（無限長軸）

- 設定した範囲で繰り返し無限にカウントするリングカウンタ形式のモードです。
  ロータリテーブルや巻き取り軸などで使用します。
- リングカウンタの範囲は、Sysmac Studio でリングカウンタ上限値とリングカウンタ下限値を設定します。
- 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。



- リングカウンタの 1 周をパルス単位に変換する場合の演算誤差により、リングカウンタ 1 周あたりのパルス数が期待した整数とはならないことがあります。期待した整数にならない場合、位置ずれの要因となります。

バージョン関連情報

ユニットバージョン V1.11 以降の CPU ユニットを使用する場合、上記の位置ずれの要因を解消するには、減速機使用を［使用する］に設定してください。
減速機使用については、「5-2-3 単位変換設定 (P.5-11)」を参照ください。

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

カウントモードをロータリモードとした場合に、有効となる設定値です。
リングカウンタの上限値と下限値を設定します。

使用上の注意

- リングカウンタの上限値と下限値の間に「0」を含まない場合、MC_MoveZeroPosition（高速原点復帰）命令を実行すると異常となります。
- MC_MoveAbsolute（絶対値位置決め）命令や MC_Move（位置決め）命令で絶対値位置決めを実行する場合は、目標位置をリングカウンタの範囲内に設定してください。範囲外の場合は異常となります。
  ただし、命令の入力変数「Direction」（方向選択）で［4: 方向指定なし］を指定した場合は、範囲外も目標位置として設定できます。このときの動作は、目標位置と指令現在位置の差分を目標距離とする相対値位置決めとなります。
- 位置カウント設定を変更した場合は、物理的な機械位置と MC 機能モジュールの指令現在位置との関係にずれが生じます。そのため、位置カウント設定を変更した場合は、原点復帰命令を実行し、再度、原点を確定してください。

エンコーダ種別

フィードバック入力するエンコーダの種別を選択します。
以下を使用する場合は、エンコーダ種別に［1: 絶対値エンコーダ(ABS)］を設定してください。

- オムロン製サーボモータ／ドライバ 1S シリーズを使用する場合
- オムロン製サーボモータ／ドライバ G5 シリーズで絶対値エンコーダ付きサーボモータを使用する場合
- オムロン製サーボモータ／ドライバ G5 シリーズでフルクローズ制御用に絶対値タイプの外部スケールを使用する場合

- オムロン製サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプで絶対値タイプの外部スケールを使用する場合

5-2-8 サーボドライバ設定

接続されているサーボドライバ、またはエンコーダ入力ターミナルで設定されている値を設定します。

パラメータ名称	機能	設定範囲	初期値 *1
リングカウンタ上限設定値	サーボドライバ、またはエンコーダ入力ターミナルで設定されているリングカウンタ上限値を設定します。*2	-2^{63}～$2^{63}-1$（単位: パルス）	2147483647
リングカウンタ下限設定値	サーボドライバ、またはエンコーダ入力ターミナルで設定されているリングカウンタ下限値を設定します。*2	-2^{63}～$2^{63}-1$（単位: パルス）	-2147483648
PDS 状態制御方式 *3	MC_Power 命令でサーボ OFF する時の PDS 状態の遷移先を設定します。*4 0: サーボ OFF 時に Switeched on 1: サーボ OFF 時に Ready to switch on	0～1	0

*1. 初期値は整数型(DINT)の全範囲となっています。オムロン製サーボドライバ 1S シリーズおよび G5 シリーズは、整数型(DINT)の全範囲のため初期値のままで使用できます。
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。

*2. オムロン製 GX シリーズ EtherCAT スレーブエンコーダ入力ターミナルを使用する場合、エンコーダ入力ターミナルのリングカウンタ最大値（対象インデックス 0x4003）とリングカウンタ上限設定値を、一致させる必要があります。また、リングカウンタ下限設定値を「0」にする必要があります。

*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することができます。

*4. オムロン製サーボドライバ G5 シリーズを使用する場合、「1」に設定すると、サーボドライバのサーボレディ(Switeched on)が使用できません。サーボレディ(Switeched on)を使用する場合は、「0」に設定してください。PDS 状態の遷移については、「A-5 PDS 状態遷移 (P.A-27)」を参照してください。

5-2-9 原点復帰設定

機械原点を決めるためのモータ動作について設定します。

パラメータ名称	機能	設定範囲	初期値
原点復帰動作 *1 *2	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	0, 1, 4, 5, 8, 9, 11～14	14
原点入力信号	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。*3	0～1	0
原点復帰開始方向	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	0, 2	0
原点検出方向	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	0, 2	0

パラメータ名称	機能	設定範囲	初期値
正方向限界入力時動作	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	0～2	1
負方向限界入力時動作	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	0～2	1
原点復帰速度	原点復帰中の速度を設定します。 （単位: 指令単位／s）	倍精度実数型の正数	10,000
原点復帰アプローチ速度	原点近傍入力が「ON」した後の速度を設定します。 （単位: 指令単位／s）	倍精度実数型の正数	1,000
原点復帰加速度	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。（単位: 指令単位／s^2）	倍精度実数型の正数、0	0
原点復帰減速度	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。（単位: 指令単位／s^2）	倍精度実数型の正数、0	0
原点復帰ジャーク	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。 （単位: 指令単位／s^3）	倍精度実数型の正数、0	0
原点入力マスク距離	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: 指令単位）	倍精度実数型の正数、0	10,000
原点位置オフセット	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。（単位: 指令単位）	倍精度実数型の負数、正数、0	0
原点復帰押し当て時間	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: ms）	0～10,000	100
原点復帰補正値	原点復帰の原点確定後の原点復帰補正量を設定します。 （単位: 指令単位）	倍精度実数型の負数、正数、0	0
原点復帰補正速度	原点復帰補正時の速度を設定します。 （単位: 指令単位／s）	倍精度実数型の正数	1,000

*1. 原点復帰動作の設定です。
詳細については、「第 8 章 原点決め (P.8-1)」を参照してください。

*2. 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 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［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。

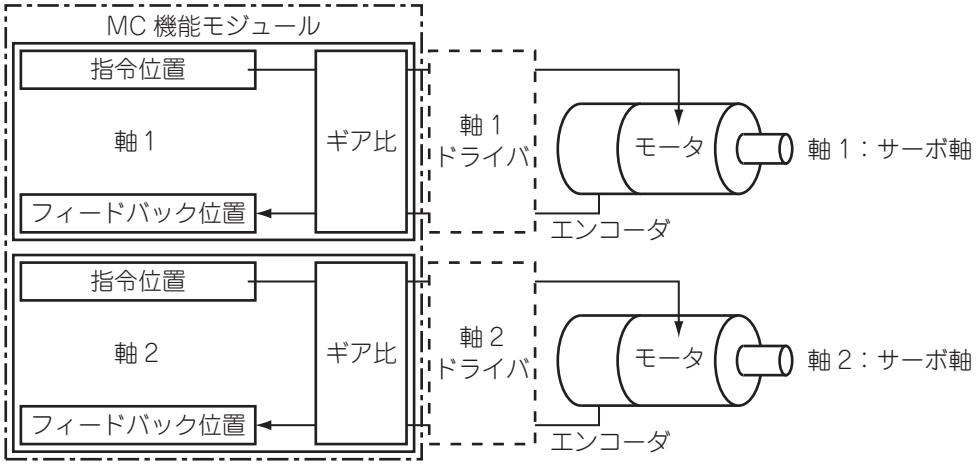
*3. オムロン製サーボドライバ 1S シリーズおよび G5 シリーズを使用する場合に指定できます。サーボドライバのラッチ機能 1 に割り付けられた入力を外部原点入力として使用します。1S シリーズおよび G5 シリーズの初期状態では、ラッチ機能 1 に外部ラッチ入力 1 が割り付けられています。
詳細については、『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル (SBCE-377)』、『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ 安全機能対応 ユーザーズマニュアル(SBCE-438)』、および『AC サ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

5-2-10 軸パラメータの設定例

軸パラメータのうち位置に関する項目の設定例を説明します。

単軸での位置決め

2 軸をそれぞれ単軸で位置決めする装置を例に説明します。

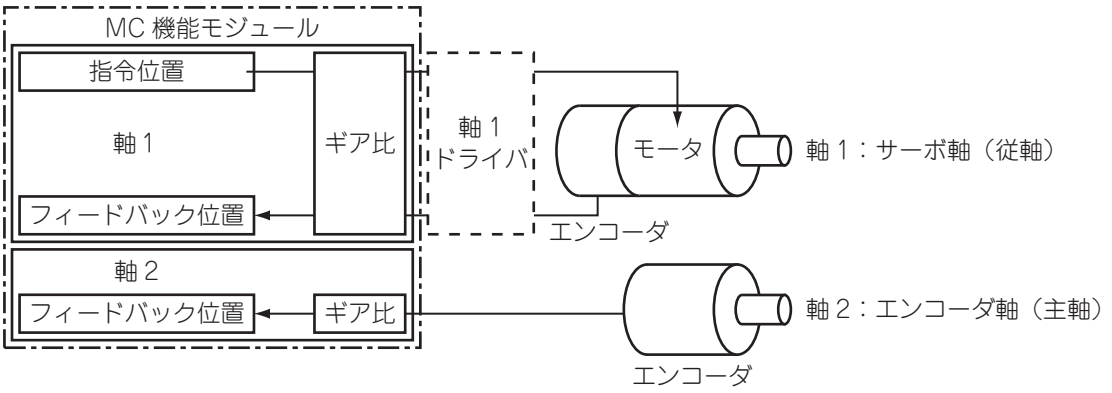


設定項目	設定内容	
	軸 1	軸 2
軸変数名	Axis1	Axis2
軸番号	1	2
軸使用	使用軸	使用軸
軸種別	サーボ軸	サーボ軸
入力デバイス／出力デバイス	1	2
表示単位	μm *1	μm
モータ 1 回転のパルス数	1,048,576	1,048,576
モータ 1 回転の移動量	10,000	10,000
最高速度	500,000 *2	500,000 *2
ジョグ最高速度	50,000 *3	50,000 *3
最大加速度	5,000,000 *4	5,000,000 *4
最大減速度	5,000,000 *4	5,000,000 *4
ソフトウェアリミット機能	指令位置に対して有効で即停止	指令位置に対して有効で即停止
正方向ソフトウェアリミット	500,000 *5	500,000 *5
負方向ソフトウェアリミット	0 *5	0 *5
カウントモード	リニアモード	リニアモード

*1. 位置の指令単位は 1（μm)となります。
*2. 最高速度は、3,000r/min = 30m/min = 0.5m/s = 500,000 μm/s としています。
*3. ジョグ最高速度は、最高速度の 10%の 0.05m/s = 50,000 μm/s としています。
*4. 最大加速度および最大減速度は、5m/s^2 です。最高速度の 3,000r/min までの加速時間を 0.1s としています。
*5. 装置の可動範囲の内側の位置を設定します。
正方向ソフトウェアリミットの設定値は、50cm = 500,000 μm としています。

エンコーダが主軸の同期制御

コンベアに取り付けられた軸 2（エンコーダ）のフィードバック位置を主軸として、軸 1 のサーボドライバが有限長範囲内で同期する装置を例に説明します。



設定項目	設定内容	
	軸 1	軸 2
軸変数名	Axis1	Axis2
軸番号	1	2
軸使用	使用軸	使用軸
軸種別	サーボ軸	エンコーダ軸
入力デバイス／出力デバイス	1	2
表示単位	μm *1	μm *1
モータ 1 回転のパルス数	1,048,576	1,048,576
モータ 1 回転の移動量	10,000	10,000
最高速度	500,000 *2	−
ジョグ最高速度	50,000 *3	−
最大加速度	5,000,000 *4	−
最大減速度	5,000,000 *4	−
ソフトウェアリミット機能	指令位置に対して有効で即停止	無効
正方向ソフトウェアリミット	500,000 *5	−
負方向ソフトウェアリミット	0 *5	−
カウントモード	リニアモード	ロータリモード
リングカウンタ上限値	−	1,000,000 *6
リングカウンタ下限値	−	0 *6

*1. 位置の指令単位は 1（μm)となります。
*2. 最高速度は、3,000r/min = 30m/min = 0.5m/s = 500,000 μm/s としています。
*3. ジョグ最高速度は、最高速度の 10%の 0.05m/s = 50,000 μm/s としています。
*4. 最大加速度および最大減速度は、5m/s^2 です。最高速度の 3,000r/min までの加速時間を 0.1s としています。
*5. 装置の可動範囲の内側の位置を設定します。
正方向ソフトウェアリミットの設定値は、50cm = 500,000 μm としています。
*6. 位置の繰り返し範囲を、0～1m (1,000,000 μm)としています。

参考

主軸は装置の構成によって、軸種別を選択することができます。軸種別にはサーボ軸、仮想サーボ軸、エンコーダ軸、仮想エンコーダ軸の 4 種類あります。
この例では主軸の軸種別がエンコーダ軸であるため、モーション制御命令の入力変数「ReferenceType」（位置種別選択）は、フィードバック位置を指定してください。

5-3 軸グループパラメータ

軸グループパラメータは、MC 機能モジュールで制御する軸グループに関して、軸構成や最高補間速度、軸グループ停止方法といった軸グループ動作の設定を行います。
軸グループパラメータは、軸グループ数分存在します。軸グループ数は、形式によって異なります。詳細は、「1-4-2 性能仕様 (P.1-7)」を参照してください。
なお、各軸グループのパラメータ設定は同じなので、ここでは、1 グループ分だけについて説明します。

5-3-1 軸グループパラメータの一覧

軸グループパラメータは Sysmac Studio を使用して軸グループごとに設定します。

区分	パラメータ名称	一時変更 *1		変数読出 *2	参照ページ
		可否	使用命令		
軸グループ基本設定	軸グループ番号			可	P.5-30
	モーション制御 *3			可 *4	
	軸グループ使用			可	
	軸構成			可	
	構成軸	可 *5	MC_ChangeAxesInGroup	可	
軸グループ動作設定	最高補間速度				P.5-32
	最大補間加速度				
	最大補間減速度				
	補間加減速オーバー				
	補間速度警告値	可	MC_Write		
	補間加速度警告値				
	補間減速度警告値				
	軸グループ停止方法				
	中心点補正許容率				

*1. 命令による一時変更の可否を示します。
*2. 変数としてユーザプログラムからの読み出し可否を示します。
*3. 形 NX701 CPU ユニットの場合に設定します。
*4. 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することができます。
*5. 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することができます。

軸グループパラメータの設定方法については、「3-4 軸グループの設定手順 (P.3-25)」を参照してください。
MC_Write（MC 設定書込）命令と MC_ChangeAxesInGroup（軸グループ構成軸書込）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 モーション編(SBCE-434)』を参照してください。
モーション制御システム変数については、「6-6 モーション制御システム変数 (P.6-19)」を参照してください。

5-3-2 軸グループ基本設定

軸グループの使用／不使用と、使用する場合の軸構成と使用する軸を設定します。

パラメータ名称	機能	設定範囲	初期値
軸グループ番号	軸グループの論理番号を設定します。 システム定義変数の変数名「_MC_GRP[0-63]」、「_MC1_GRP[0-63]」、「_MC2_GRP[0-63]」の番号は、ここで設定した番号となります。	−	−
モーション制御 *1	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）のどちらかのタスクに割り付けます。 1: プライマリ定周期タスク 2: 定周期タスク（実行優先度 5）	1〜2	1
軸グループ使用	軸グループを使用するかしないかを設定します。未作成軸グループおよび未使用軸グループに対してモーション制御命令を実行すると異常となります。 *2 0: 未作成軸グループ 1: 未使用軸グループ 2: 使用軸グループ	0〜2	0
軸構成	軸グループの軸構成を設定します。 0: 2 軸 1: 3 軸 2: 4 軸	0〜2	0
構成軸	軸グループに割り付ける軸番号を設定します。 Sysmac Studio では、軸 A0〜A3 に対して、軸変数名を選択します。	2〜4 軸	0

*1. 形 NX701 CPU ユニットの場合に設定します。
*2. 未使用軸を含む軸グループに対して、MC_GroupEnable（軸グループ有効）命令を実行すると異常となります。

軸構成

次の表は、MC 機能モジュールで使用可能な軸構成です。
Sysmac Studio を用いて実際の装置にあわせて軸構成を設定します。

軸構成	説明
2 軸	2 軸の軸構成です。たとえば、2 軸の直交座標系の機械です。
3 軸	3 軸の軸構成です。たとえば、3 軸の直交座標系の機械です。
4 軸	4 軸の軸構成です。たとえば、3 軸の直交座標系と先端ツールの回転軸を持つ機械です。

構成軸

軸グループに属する軸を構成軸と呼びます。
また、軸グループ指令の補間命令の場合、プログラムの再利用性を高めるために、軸番号（軸 0〜軸 255）の代わりに論理軸（軸 A0〜軸 A3）を使用します。
構成軸は、軸グループに属する軸の軸番号と論理軸の番号の割り付けを設定します。
論理軸には、サーボ軸、または仮想サーボ軸から選択できます。
Sysmac Studio を用いて軸 A0 から順に、軸構成で選択した軸数分の軸を設定します。
複数の軸グループを作成する場合でも、軸グループごとに軸 A0 から順に軸番号を設定します。
また、おなじ軸番号を複数の軸グループに設定することもできます。

軸構成の設定	構成軸の設定
2 軸	軸 A0、軸 A1 の軸変数名（軸番号）を設定します。
3 軸	軸 A0、軸 A1、軸 A2 の軸変数名（軸番号）を設定します。
4 軸	軸 A0、軸 A1、軸 A2、軸 A3 の軸変数名（軸番号）を設定します。

使用上の注意

- 軸グループの構成軸は、すべて同じタスクに割り付けてください。
- おなじ軸番号を複数の軸グループに設定した場合、おなじ軸番号が設定された軸グループを同時に有効にはできません。
- ［制御機能］の設定が［単軸位置制御のみ］の軸は、軸グループの構成軸に割り付けられません。

バージョン関連情報

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 以降の組み合わせの場合、サーボ軸、または仮想サーボ軸であれば、使用軸だけでなく未使用軸（使用軸に切替可）に設定されている軸も、軸グループに設定できます。

● 構成軸の設定例

- 軸グループに軸番号 1, 2, 5, 8 の 4 軸を割り付ける場合の設定は、以下のようになります。

論理軸	軸番号
軸 A0	軸 1
軸 A1	軸 2
軸 A2	軸 5
軸 A3	軸 8

- 軸グループに軸番号 1, 8, 2 の 3 軸を割り付ける場合の設定は、以下のようになります。

論理軸	軸番号
軸 A0	軸 1
軸 A1	軸 8
軸 A2	軸 2
軸 A3	（なし）

5-3-3 軸グループ動作設定

制御する装置の仕様にあわせて、最高補間速度や軸グループ停止方法など、軸グループの動作について設定します。
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。

パラメータ名称	機能	設定範囲	初期値
最高補間速度	軌跡の最高補間速度を設定します。 「0」を設定すると補間速度の制限はなくなります。 軸グループ動作命令で最高補間速度を超える補間の目標速度が指定された場合は、最高補間速度で動作します。（単位: 指令単位／s）	倍精度実数型の正数、0	800000000
最大補間加速度	軌跡の最大補間加速度を設定します。 「0」を設定すると補間加速度の制限はなくなります。 （単位: 指令単位／s^2）	倍精度実数型の正数、0	0
最大補間減速度	軌跡の最大補間減速度を設定します。 「0」を設定すると補間減速度の制限はなくなります。 （単位: 指令単位／s^2）	倍精度実数型の正数、0	0

パラメータ名称	機能	設定範囲	初期値
補間加減速オーバー	軸グループの加減速制御で、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補間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	0～2	0
補間速度警告値	補間速度警告を出力するための最高補間速度に対する割合を設定します。 「0」を設定すると補間速度警告は出力しません。 (単位: %)	0～100	0
補間加速度警告値	補間加速度警告を出力するための最高補間加速度に対する割合を設定します。 「0」を設定すると補間加速度警告は出力しません。 (単位: %)	0～100	0
補間減速度警告値	補間減速度警告を出力するための最高補間減速度に対する割合を設定します。 「0」を設定すると補間減速度警告は出力しません。 (単位: %)	0～100	0
軸グループ停止方法	多軸協調動作中にある軸で即停止する異常が発生した場合に、異常が発生していない構成軸の停止方法を設定します。 0: 即停止 1: 各軸を減速停止（各軸の最大減速度） 3: 即停止し、同時にサーボ OFF を実行	0, 1, 3	0
中心点補正許容率	円弧補間命令の中心点指定方式で、始点と中心点、終点と中心点の距離が異なる場合は、距離が等しくなるように補正します。その補正の許容範囲を半径に対する比率で設定します。比率は 0.1%以上で設定します。 「0」を設定すると誤差のチェックは行いません。	単精度浮動小数点型で 0～100	0

*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、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44)」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、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44)」を参照してください。

5-3-4 軸グループの有効

ユーザプログラムで軸グループへの動作命令を有効にするためには、MC_GroupEnable（軸グループ有効）命令で有効にする軸グループ番号を指定します。
有効にされていない軸グループに対してモーション制御命令を起動すると、命令で異常が発生します。また、同時に複数の軸グループを有効にできますが、同じ軸を含む軸グループを有効にすると、命令で異常が発生します。

ある 1 つの軸を作業工程ごとに、異なる軸グループで動作させたい場合、その軸を含む複数の軸グループを作成しておきます。その上で、MC_GroupEnable(軸グループ有効)命令と MC_GroupDisable（軸グループ無効）命令を順次切り替えていくことで、実現可能です。

多軸での動作中に、MC_GroupDisable（軸グループ無効）命令を起動すると減速停止します。

6

モーション制御プログラム

この章では、モーション制御プログラムの仕様と、プログラム作成までの操作手順について説明します。

6-1　概要 6-2

6-2　モーション制御命令 6-5

6-2-1　PLCopen®モーション制御用ファンクションブロック 6-5
6-2-2　モーション制御機能モジュールのモーション制御命令 6-5

6-3　状態遷移 6-7

6-3-1　モーション制御機能モジュールの状態 6-7
6-3-2　軸の状態 6-7
6-3-3　軸グループの状態 6-9

6-4　モーション制御命令の起動とステータス 6-11

6-4-1　起動についての共通ルール 6-11
6-4-2　起動のタイミングチャート 6-13
6-4-3　モーション制御命令の再起動時のタイミングチャート 6-15
6-4-4　モーション制御命令の多重起動時のタイミングチャート 6-16

6-5　位置の扱い 6-17

6-5-1　位置の種類 6-17
6-5-2　各軸種別で使用可能な位置の種類 6-18

6-6　モーション制御システム変数 6-19

6-6-1　モーション制御システム変数の概要 6-19
6-6-2　モーション制御システム変数のしくみ 6-21
6-6-3　モーション制御システム変数の一覧 6-22

6-7　カムテーブルとカムデータ変数 6-32

6-8　モーション制御プログラムの作成方法 6-36

6-9　カムテーブルの作成方法 6-38

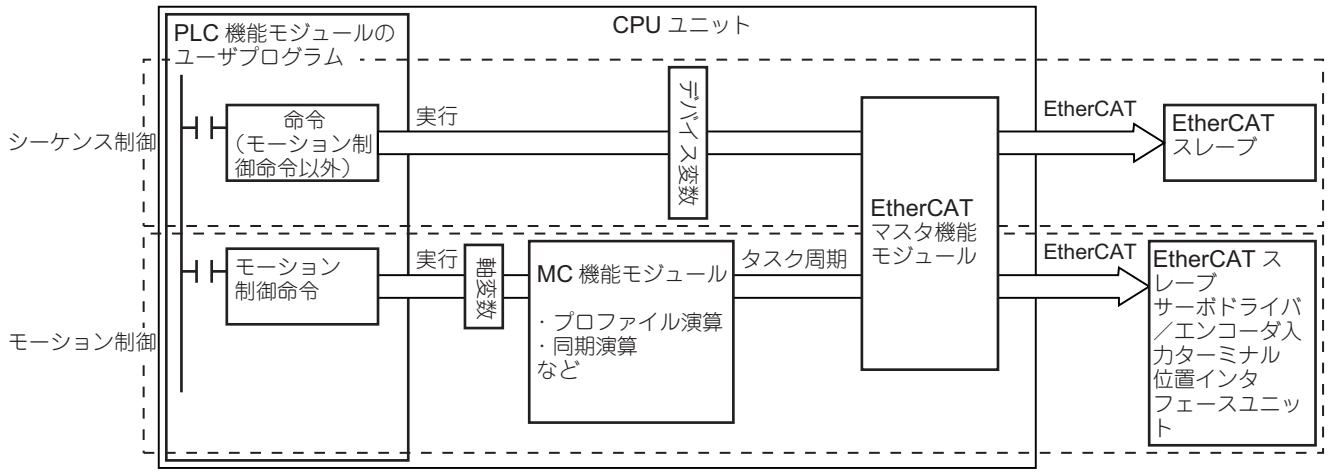
6-1 概要

NJ/NX シリーズ CPU ユニットでは、シーケンス制御とモーション制御を行うことができます。
ユーザプログラムにモーション制御命令を記述することで、EtherCAT スレーブサーボドライバや NX シリーズ 位置インタフェースユニットなどを対象にモーション制御を実行します。
本マニュアルでは、モーション制御命令を記述したプログラムを、モーション制御プログラムと呼びます。
EtherCAT スレーブサーボドライバや NX シリーズ 位置インタフェースユニットには、軸変数を割り付ける必要があります。軸変数に割り付けない場合は、汎用スレーブと同じく I/O デバイス変数に割り付けます。
モーション制御命令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 と 16）で使用することができます。

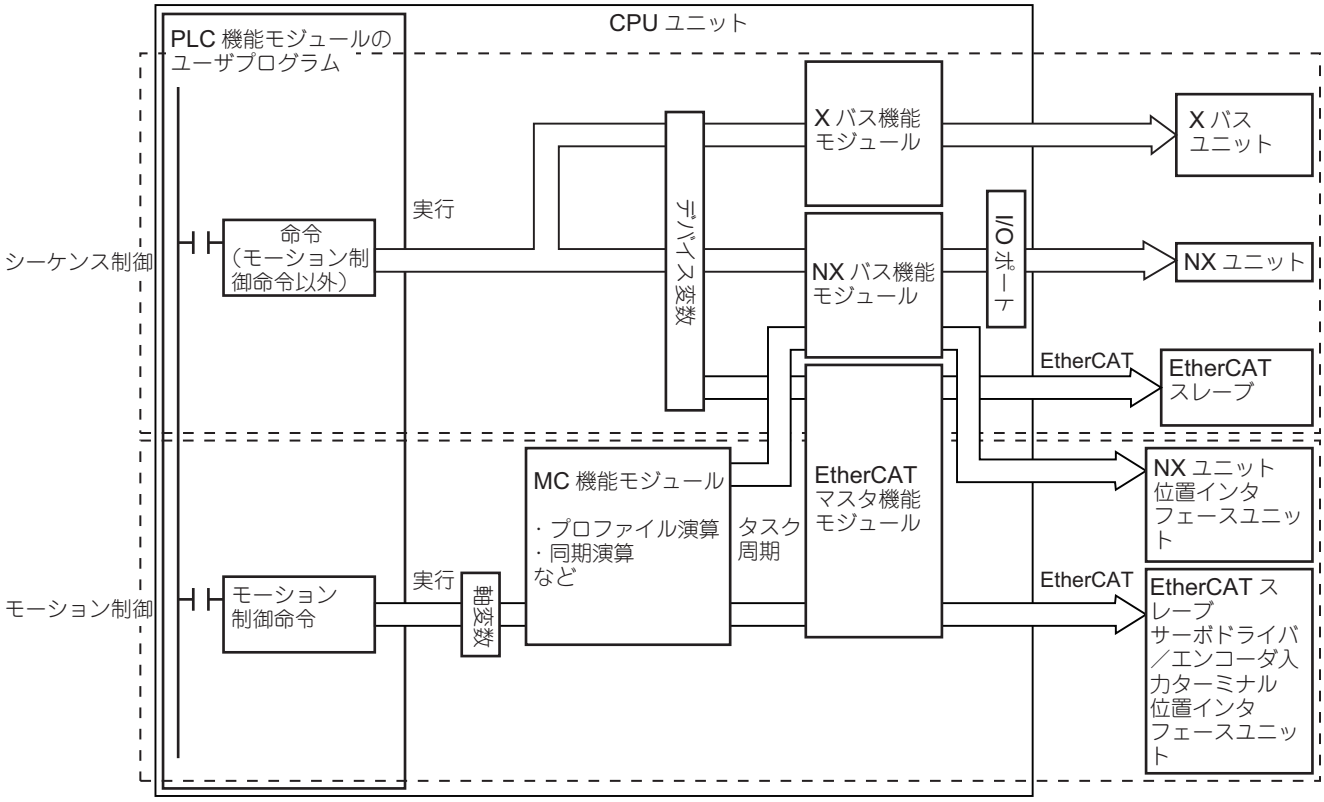
バージョン関連情報

Sysmac Studio Ver.1.09 以降を使用することで、軸変数に割り付けたスレーブやユニットの I/O ポートにデバイス変数を割り付け可能となります。
詳細は、「2-4-2 EtherCAT マスタ機能とモーション制御機能モジュールの関係 (P.2-19)」を参照してください。

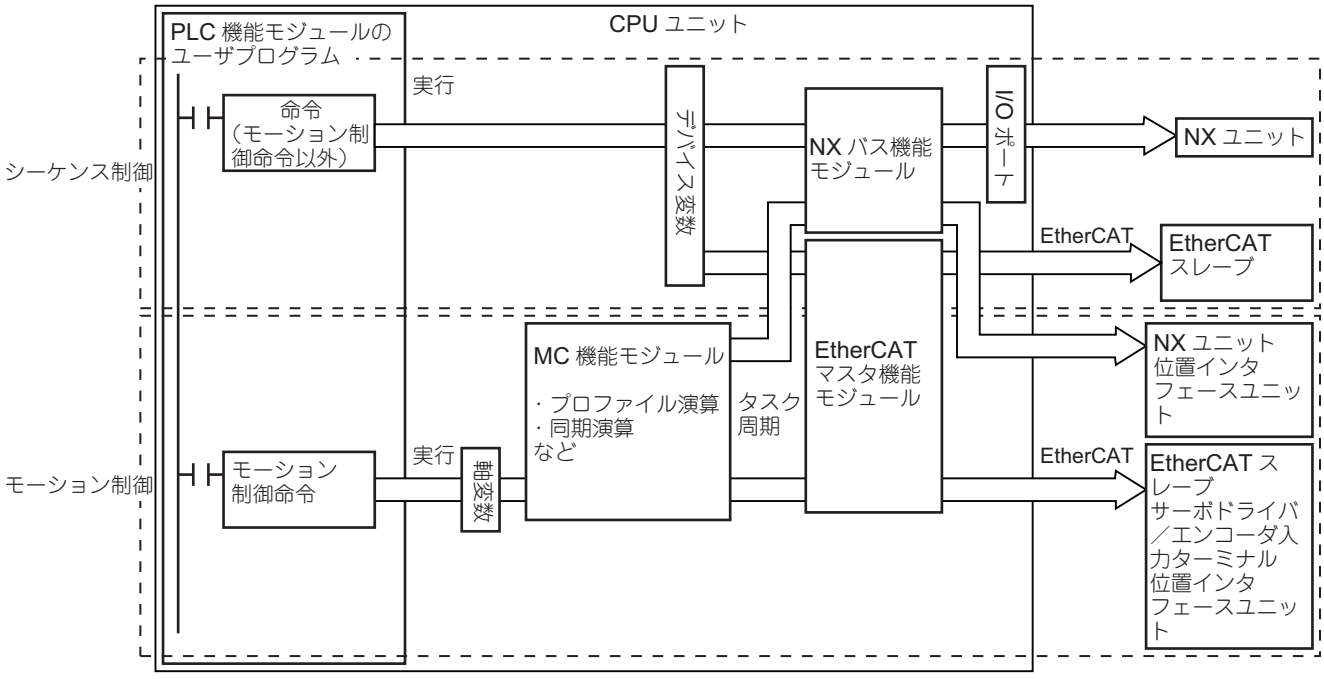
● **形 NX701 CPU ユニットの場合**



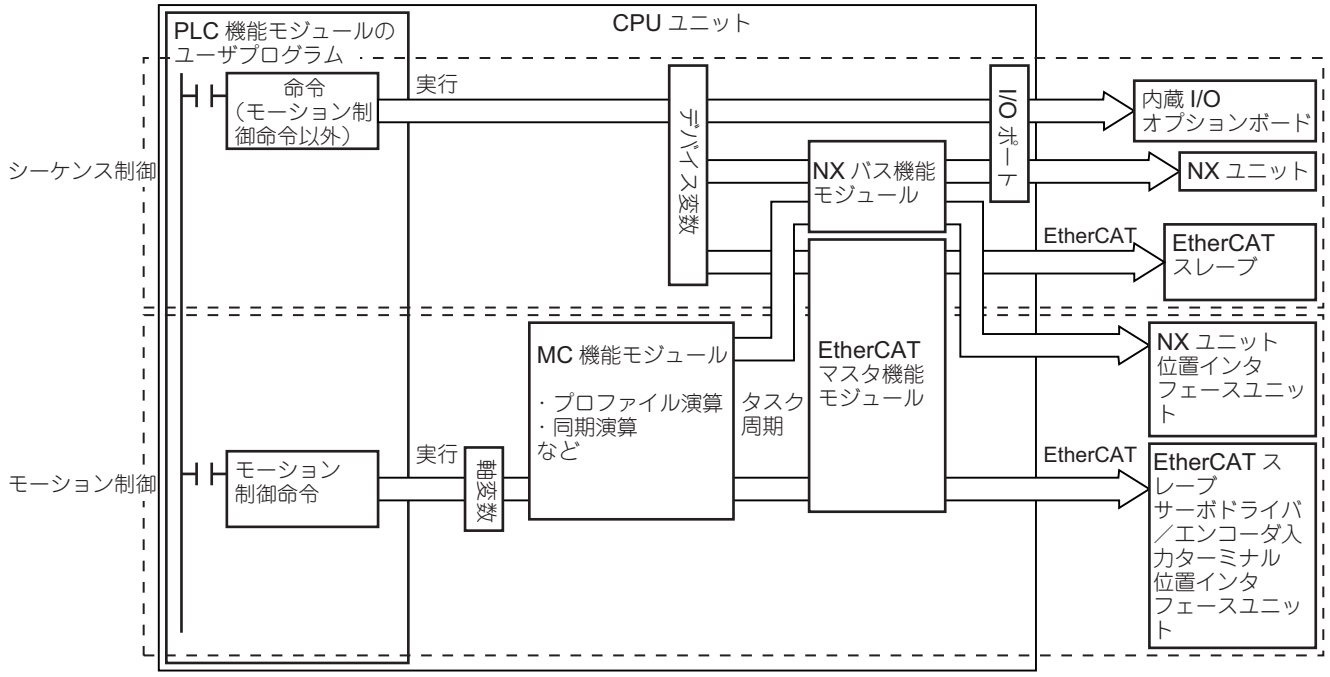
● **形 NX502 CPU ユニットの場合**



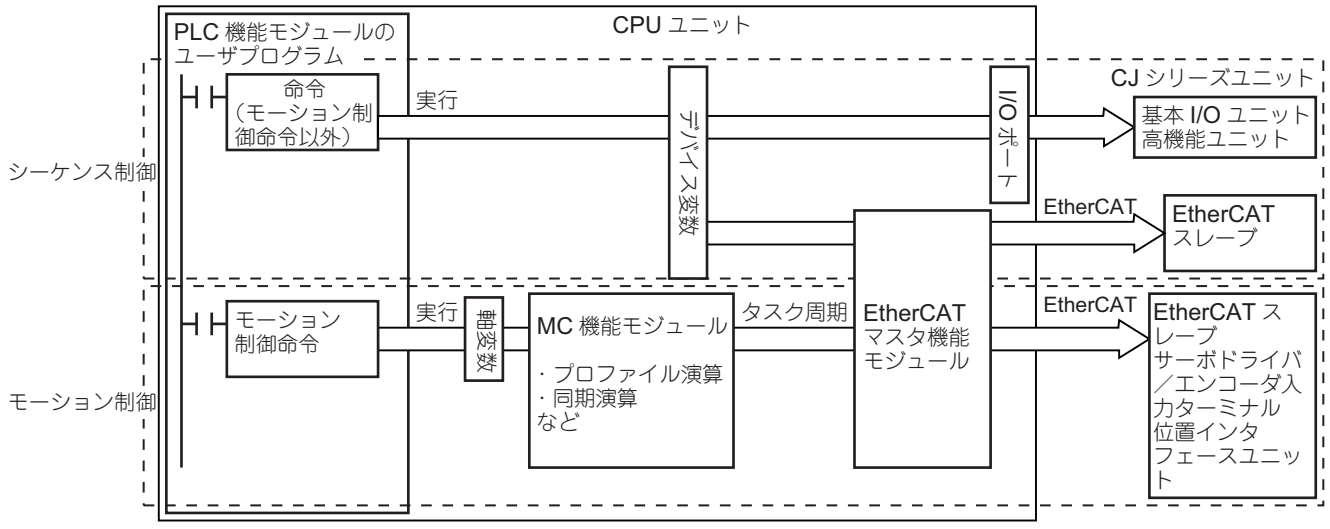
● **形 NX102 CPU ユニットの場合**



● 形 NX1P2 CPU ユニットの場合



● NJ シリーズ CPU ユニットの場合



6-2 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（以降、FB と省略する場合があります）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 FB の概要と MC 機能モジュールの仕様について説明します。

NJ/NX シリーズ CPU ユニットのファンクションブロック(FB)の基本事項については、『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。
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

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、NX シリーズ 位置インタフェースユニットを使用することができます。

6-2-1 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で、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 の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/

6-2-2 モーション制御機能モジュールのモーション制御命令

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

MC 機能モジュールで使用可能な命令の一覧と詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』を参照してください。

6-3 状態遷移

MC 機能モジュールの軸と軸グループの状態、および命令の起動による状態遷移は、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。
ここでは、MC 機能モジュール全体も含め、それぞれの状態と状態遷移について説明します。

6-3-1 モーション制御機能モジュールの状態

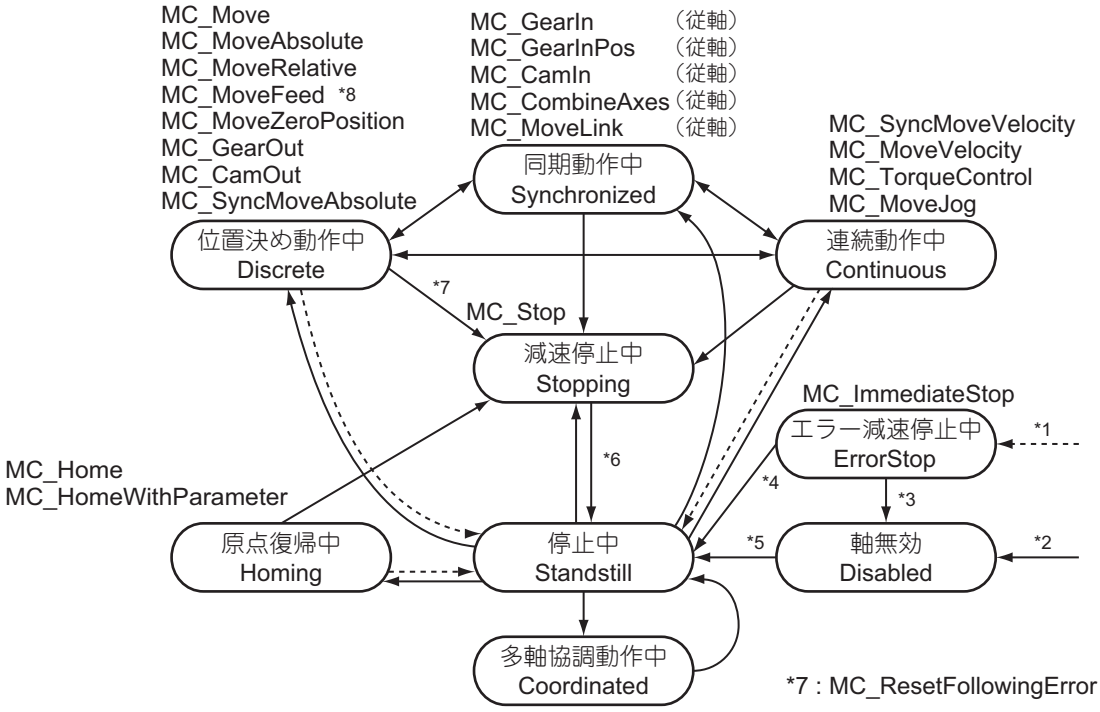
MC 機能モジュール全体の状態を下表に示します。

状態名	定義
MC 機能モジュール運転中 *1	モーション制御命令が有効な状態です。 ユーザプログラムのモーション制御命令を解析し、モーション制御を行います。 CPU ユニットの動作モードに関係なく「MC 機能モジュール運転中」にできます。
MC 機能モジュール試運転中 *2	Sysmac Studio から試運転をおこなっている状態です。
カムテーブルファイル保存中 *3	カムテーブルファイルの保存中または待機中の状態です。
カムテーブル生成実行中 *4	カムテーブルの生成実行中の状態です。*5

*1. MC 共通変数の「_MC_COM.Status.RunMode」でモニタできます。
*2. MC 共通変数の「_MC_COM.Status.TestMode」でモニタできます。
*3. MC 共通変数の「_MC_COM.Status.CamTabeleBusy」でモニタできます。
*4. MC 共通変数の「_MC_COM.Status.GenerateCamBusy」でモニタできます。
*5. CPU ユニットの電源を OFF する場合は、カムテーブル生成実行中でないことを確認してください。カムテーブルの生成実行中に CPU ユニットの電源を OFF すると、カムテーブルが正しく生成されません。

6-3-2 軸の状態

軸に対するモーション制御命令が起動されたときの軸の動作を下図に表します。
原則として、モーション制御命令は逐次実行され、軸は下表のいずれかひとつの状態になります。



*1.「多軸協調動作中」以外のいずれの状態からでも、軸の異常発生で遷移します。
*2. いずれの状態からでも、MC_Power 命令の「Status」（運転可）出力が「FALSE」で、軸の異常が発生していないときに遷移します。（サーボ OFF）
*3. サーボ OFF 状態で、MC_Reset 命令または ResetMCError 命令による異常解除で遷移します。
*4. サーボ ON 状態で、MC_Reset 命令または ResetMCError 命令による異常解除で遷移します。
*5. MC_Power 命令の「Enable」（有効）入力を「TRUE」にして、MC_Power 命令の「Status」（運転可）出力が「TRUE」で遷移します。（サーボ ON）
*6. MC_Stop 命令の「Done」（完了）出力が「TRUE」で、MC_Stop 命令の「Execute」（起動）入力が「FALSE」のときに遷移します。
*7. MC_ResetFollowingError 命令により、「減速停止中」へ遷移します。
*8. MC_MoveFeed 命令の入力変数「MoveMode」（移動方法選択）に、[2: _mcVelocity]（速度制御）を設定した場合、トリガ入力が検出されるまでは「連続動作中」となります。

状態名		定義
サーボ OFF		軸がサーボ OFF している状態です。 この状態に遷移すると命令多重起動の待機状態を解除します。
	軸無効	軸がサーボ OFF で停止中で起動準備が整っている状態です。
	エラー減速停止中 *1	軸がサーボ OFF で停止中で軸の異常が発生している状態です。
サーボ ON		軸がサーボ ON している状態です。
	停止中	軸がサーボ ON で停止している状態です。
	位置決め動作中	目標位置が指定された位置決めを実行している状態です。 位置決め完了待ちの状態や、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。
	連続動作中	目標位置が指定されない連続的な動作を実行している状態です。 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。
	同期動作中	同期制御指令により、軸を同期制御している状態です。 同期制御命令切り替え後の同期待ち状態も含みます。
	減速停止中	MC_Stop（強制停止）命令、または MC_TouchProbe（外部ラッチ有効）命令により軸が停止するまでの状態です。 MC_Stop（強制停止）命令で停止後、Execute（起動）が TRUE の状態も含みます。 この状態では軸指令による動作を実行できません。 実行すると命令は CommandAborted（実行中断）が TRUE となります。
	エラー減速停止中 *1	軸がサーボ ON で軸の異常が発生している状態です。 MC_ImmediateStop（即停止）命令が実行されている状態や、軸の異常の発生により軸が減速している状態も含みます。 この状態では軸動作指令を実行できません。実行すると命令は中断（CommandAborted = TRUE）状態になります。
	原点復帰中	MC_Home（原点復帰）命令、または MC_HomeWithParameter（パラメータ指定原点復帰）命令により、原点を探している状態です。
	多軸協調動作中	軸が、軸グループ指令の命令により軸グループ有効となっている状態です。 軸グループの状態の、「グループ動作中」、「減速停止中」、または「エラー停止中」で動作している状態です。

*1. 「エラー減速停止中」には、軸がサーボ OFF のときと、サーボ ON のときの両方があります。

（注）軸の状態は、軸変数の「_MC_AX[0-255].Status」の各メンバ変数でモニタできます。

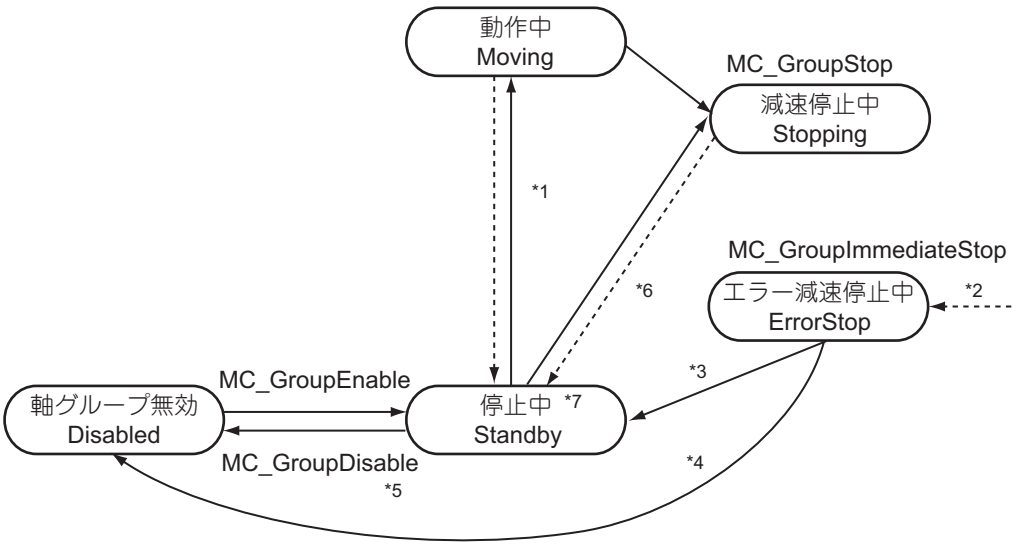
バージョン関連情報

形 NX701 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。

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。

6-3-3 軸グループの状態

軸グループに対するモーション制御命令が起動されたときの軸グループの動作を下図に表します。



*1. すべての軸グループ動作命令で遷移します。
*2. すべての状態から遷移します。軸グループが無効の状態でも、異常が発生すると遷移します。
*3. 軸グループが有効の場合に、MC_GroupReset 命令、または ResetMCError 命令で遷移します。
*4. 軸グループが無効の場合に、MC_GroupReset 命令、または ResetMCError 命令で遷移します。
*5. エラー減速停止中状態のときに、MC_GroupDisable 命令を起動しても、同じ状態にとどまります。
*6. MC_GroupStop 命令の出力変数「Done」＝ TRUE で、入力変数「Execute」＝ FALSE のときに遷移します。
*7. 停止中（Standby）状態で、下記のすべての条件を満たすとき、起動準備完了(Ready)となります。
すべての構成軸がサーボ ON している。
すべての構成軸で MC_Stop 命令が実行されていない。
すべての構成軸が原点確定している。

状態名		定義
軸グループ無効		軸グループが無効な状態です。 本状態へ遷移すると命令多重起動の待機状態を解除します。
	エラー減速停止中 *1	軸グループが無効な状態で、軸グループの異常が発生している状態です。
軸グループ有効		軸グループが有効な状態です。
	停止中	軸グループ指令の動作命令を実行していない状態です。 軸グループの構成軸のサーボ ON/OFF 状態には関係ありません。
	動作中	軸グループ指令の動作命令により、目標位置が指定された位置決めを実行している状態です。 インポジション待ちの状態や、グループ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。
	減速停止中	MC_GroupStop（軸グループ強制停止）命令が実行されている状態です。 MC_GroupStop（軸グループ強制停止）命令で停止後、「Execute」（起動）が TRUE の状態も含みます。 この状態では軸グループ指令による動作を実行できません。 実行すると命令は「CommandAborted」（実行中断）が TRUE となります。
	エラー減速停止中 *1	軸グループで異常が発生している状態です。 MC_GroupImmediateStop（軸グループ即停止）命令が実行されている状態や、軸グループの異常の発生により軸グループが減速している状態も本状態に含みます。 この状態では多軸協調指令の実行を禁止します。 実行時は「CommandAborted」（実行中断）が TRUE となります。

*1. 「エラー減速停止中」には、軸グループ無効のときと軸グループ有効のときの両方があります。

（注）軸グループの状態は、軸グループ変数の「_MC_GRP[0-63].Status」、「_MC1_GRP[0-63].Status」、「_MC2_GRP[0-63].Status」の各メンバ変数でモニタできます。

6-4 モーション制御命令の起動とステータス

MC 機能モジュールの共通ルールとして、モーション制御命令を起動する変数や命令の実行状態を示す変数を定義しています。
モーション制御命令の機能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。

6-4-1 起動についての共通ルール

MC 機能モジュールの共通ルールを下表に記載しています。
起動例を「6-4-2 起動のタイミングチャート (P.6-13)」に記載しています。
あわせて参照してください。

項目	原則
出力の排他性	出力変数の「Busy」（実行中）、「Done」（完了）、「Error」（エラー）、「CommandAborted」（実行中断）は、相互に排他的であり、複数が同時に TRUE になることはありません。 同様に、出力変数の「Active」（制御中）、「Done」（完了）、「Error」（エラー）、「CommandAborted」（実行中断）についても、複数が同時に TRUE になることはありません。 「Busy」（実行中）と「Active」（制御中）は、同時に TRUE になる場合があります。
出力ステータス	出力変数の「Done」（完了）、「InGear」（ギア比到達）、「InSync」（同期中）、「InVelocity」（目標速度到達）、および「CommandAborted」（実行中断）は、入力変数の「Execute」（起動）の立ち下がりエッジで FALSE になります。 「Execute」（起動）の立ち下がりエッジは、実際のモーション制御命令の実行を停止させません。 命令インスタンスが実行完了する以前に、「Execute」（起動）が FALSE になっていても、命令インスタンスの状態が変化すれば、少なくとも 1 周期の期間、対応する出力変数が TRUE になります。 出力変数の「Error」（エラー）と「ErrorID」（エラーコード）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、あるいは 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。*1 モーション制御命令が実行中に、同じ命令インスタンスの「Execute」（起動）を再度 FALSE→TRUE に立ち上げても（命令再起動）、「CommandAborted」（実行中断）は TRUE になりません。
入力パラメータ	入力変数の「Execute」（起動）で起動するモーション制御命令の入力パラメータは、「Execute」（起動）の立ち上がりエッジ発生時の値が使用されます。 入力変数の「Enable」（有効）で起動するモーション制御命令の入力パラメータは、「Enable」（有効）が TRUE の間、周期ごとの値が使用されます。
入力パラメータの省略	命令インスタンスの入力パラメータを省略した場合、初期値が適用されます。*2
「Position（目標位置）」と「Distance（移動距離）」	入力変数の「Position」（目標位置）は、座標系で定義された値です。 入力変数の「Distance」（移動距離）は、相対的な長さであり 2 つの位置間の差異です。
符号規則	入力変数の「Acceleration」（加速度）、「Deceleration」（減速度）、「Jerk」（ジャーク）は、正の値、または「0」です。 「Position」（目標位置）、「Distance」（移動距離）、「Velocity」（目標速度）は、正の値、負の値、または「0」のいずれの値にもなり得ます。

項目	原則
異常処理動作	命令インスタンスの実行中に発生する異常を示す出力変数は 2 つあります。 これらの出力は以下のように定義されています。 • Error（エラー）: 出力変数「Error」の立ち上りエッジは、命令インスタンスの実行中に異常が発生したことを示します。 • ErrorID（エラーコード）: 異常の原因を示すコードです。 出力変数の「Done」(完了)、「InVelocity」(目標速度到達)、「InGear」(ギア比到達)、「InSync」(同期中）は、正常完了または正常に動作していることを意味するため、出力変数の「Error」(エラー）に対して排他的になります。 異常の種類は、以下に示す 2 つあります。 • 命令インスタンスの異常（例: パラメータ範囲外、状態遷移の条件違反） • 軸の異常（例: 位置偏差オーバー、サーボドライバ側の異常） 命令インスタンスの異常では、軸の異常を発生させないで軸を停止させる場合があります。
出力変数「Done」（完了）の動作	出力変数の「Done」（完了）、「InGear」（ギア比到達）、「InSync」（同期中）は、命令の動作が正常完了または指令した状態に到達した場合に TRUE になります。目標位置を指定する動作命令の場合、軸パラメータの［インポジションチェック時間］の設定によって TRUE となるタイミングが異なります。 •「0 以外」の場合、位置決めが完了した次の周期で TRUE となります。 •「0」の場合、指令終了（払い出し完了）の次の周期で TRUE となります。 同じ軸で動作する複数の命令では、軸の動作が最終目標位置に到達する前に、他の動作命令で割り込まれた場合、最初の命令の出力変数「Done」（完了）は TRUE になりません。
出力変数「CommandAborted」（実行中断）の動作	出力変数の「CommandAborted」（実行中断）は、指令された動作が、ほかの動作命令で割り込まれたときに TRUE になります。 本 MC 機能モジュールでは、モーション制御命令の起動時に対象の軸や軸グループが異常発生中または減速停止中のときも、TRUE になります。 「CommandAborted」（実行中断）が TRUE になると、ほかの出力変数は FALSE になります。
有効範囲外の入力	入力変数の値が有効範囲外で起動された場合、命令インスタンスは異常を発生します。
出力変数「Busy」（実行中）の動作	出力変数の「Busy」（実行中）は、命令インスタンスが実行中であることを示します。 「Busy」（実行中）は、入力変数の「Execute」（起動）の立ち上がりエッジで TRUE になり、出力変数の「Done」（完了）、「CommandAborted」（実行中断）、「Error」（エラー）が TRUE になったときに FALSE になります。 上記の出力変数は、いつ変化するかわからないので、「Busy」（実行中）が TRUE の間は、出力変数の変化を監視するために、命令インスタンスが毎周期実行 *3 されるようにプログラムしてください。 ひとつの軸またはひとつの軸グループで、複数の命令インスタンスの「Busy」（実行中）が TRUE になり得ますが、出力変数「Active」（制御中）が TRUE になるのは、一度にひとつの命令インスタンスだけです。 ただし、MC_Phasing（主軸相対値位相補正）命令は例外です。
出力変数「Active」（制御中）	出力変数の「Active」（制御中）は、命令インスタンスが該当する軸の制御権を獲得したときに TRUE になります。 出力変数の「Active」（制御中）は、「Busy」（実行中）より遅れて変化することがあります。*4

*1. 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、「Error」（エラー）と「ErrorID」（エラーコード）は「0」となります。
また、「Error」（エラー）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」（起動）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、異常を解除した時に「Enable」（有効）が TRUE であれば起動します。

*2. ラダー図言語でプログラムを作成する場合は、入力変数の「Execute」（起動）または「Enable」（有効）と左母線との間に接点を配置してください。接点を配置せずに左母線と直接接続すると、ビルド実施時に異常となります。
予約(Reserved)扱いの入力変数に対しては、入力パラメータを省略するか初期値を設定してください。

*3. ST 構造命令で条件式や選択値が不一致の場合、その文に記述した命令は実行されません。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 モーション編(SBCE-434)』を参照してください。

*4. 出力変数のタイミングチャートについては、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』を参照してください。

使用上の注意

- モーション制御命令を起動する場合は、EtherCAT 通信のプロセスデータ通信が正常に実行中であることを確認してください。
  詳細は、「10-2-1 EtherCAT 通信の監視とサーボ ON (P.10-3)」を参照してください。
- 「Execute」（起動）は、ユーザプログラム実行の初回の周期では FALSE となるようにユーザプログムを作成してください。

6-4-2 起動のタイミングチャート

MC 機能モジュールのモーション制御命令は、常時実行型の FB です。
ここでは、モーション制御命令の機能を起動するための入力変数が「Execute」（起動）の命令を Execute 型、「Enable」（有効）の命令を Enable 型として説明します。

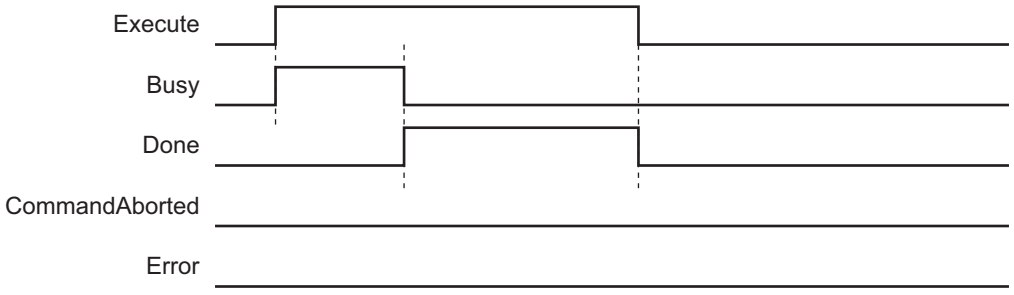
起動条件	内容
Execute 型	モーション制御命令の入力変数「Execute」（起動）の立ち上がり時に起動する命令です。 次の状態になるまで命令の実行を継続します。 • 命令で指定した動作の完了 • ほかのモーション制御命令の起動による動作の中断 • Execute（起動）の再立ち上がりエッジによる再起動 そのほかの入力変数は、「Execute」（起動）の立ち上がり時に値が取り込まれます。
Enable 型	モーション制御命令の入力変数「Enable」（有効）が TRUE の間、毎周期実行する命令です。 「Enable」（有効）が TRUE の間、そのほかの入力変数も毎周期、取り込まれます。 ただし例外として、MC_MoveJog（ジョグ送り）の入力変数「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）の立ち上がり時の値が取り込まれます。

使用上の注意

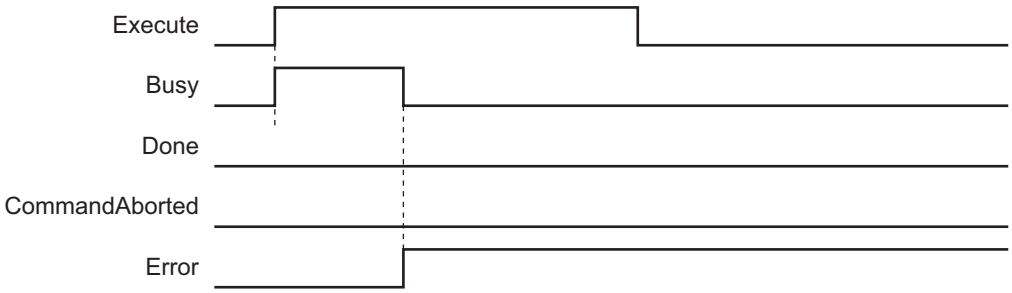
マニュアルに記載しているタイミングチャートは、Sysmac Studio のデータトレースで表示したタイミングと異なる場合があります。
データトレースの詳細については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編(SBCA-467)』を参照してください。

Execute 型の命令実行タイミングチャート

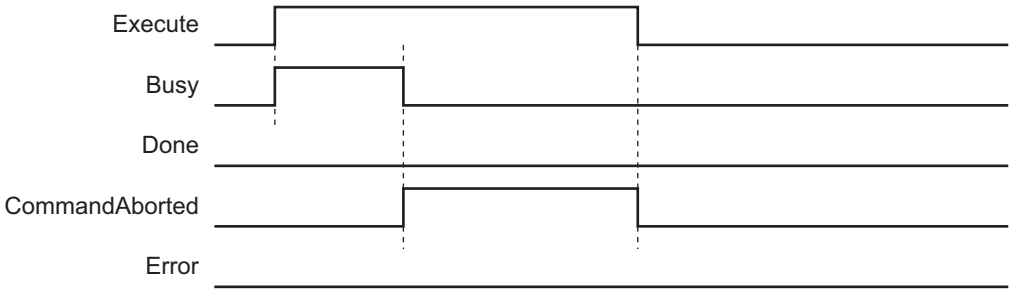
- 入力変数の「Execute」（起動）が TRUE の状態で命令の動作が完了します。
  その後、「Execute」（起動）を FALSE にするまで異常が発生しなかった場合を下図に示します。



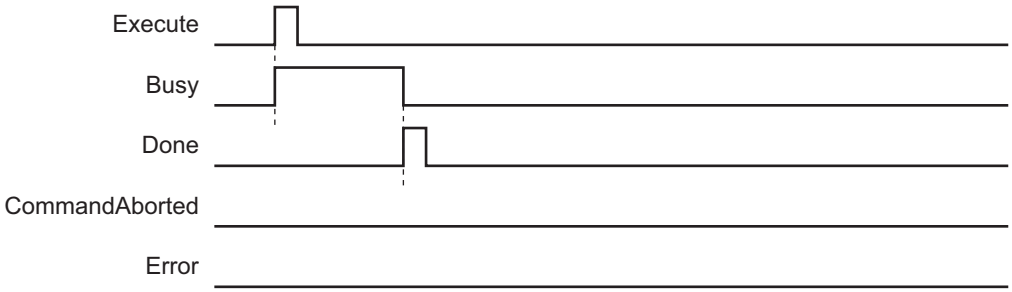
- 入力変数の「Execute」（起動）が TRUE の状態で、異常が発生した場合を下図に示します。
  その後、「Execute」（起動）を FALSE にしても、出力変数の「Error」（エラー）は TRUE のままです。



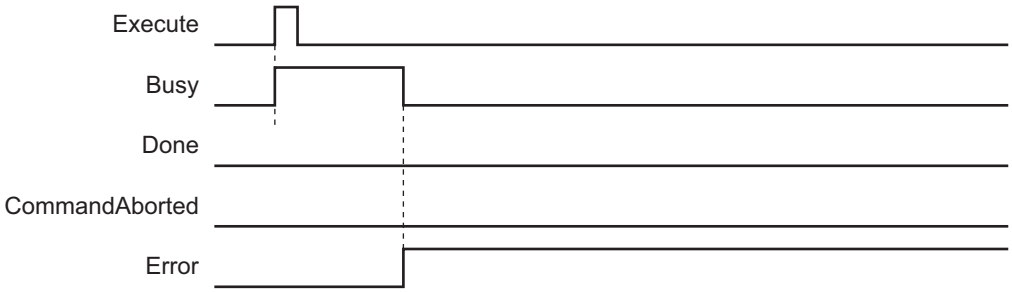
- 入力変数の「Execute」（起動）が TRUE の状態で、命令の実行が途中で中断された場合を下図に示します。



- 入力変数の「Execute」（起動）を 1 周期だけ TRUE にし、命令で異常が発生しなかった場合を下図に示します。
  命令の動作が完了すると、出力変数の「Done」（完了）は 1 周期だけ TRUE となります。

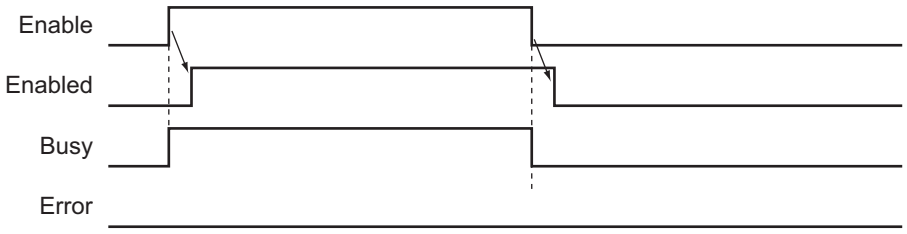


- 入力変数の「Execute」（起動）を 1 周期だけ TRUE にし、命令で異常が発生した場合を下図に示します。
  出力変数の「Error」（エラー）は TRUE のままです。

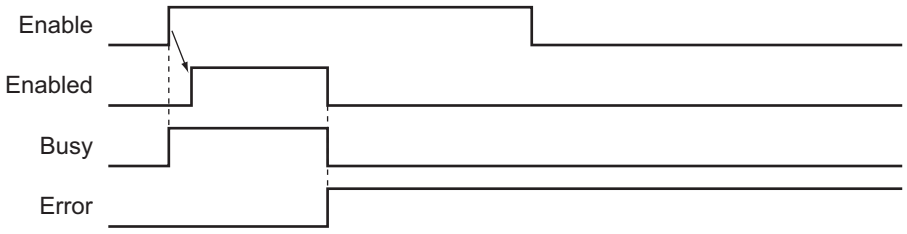


Enable 型の命令実行タイミングチャート

- 入力変数「Enable」（有効）を TRUE にし、命令で異常が発生しなかった場合を下図に示します。



• 入力変数「Enable」（有効）を TRUE にし、命令で異常が発生した場合を下図に示します。



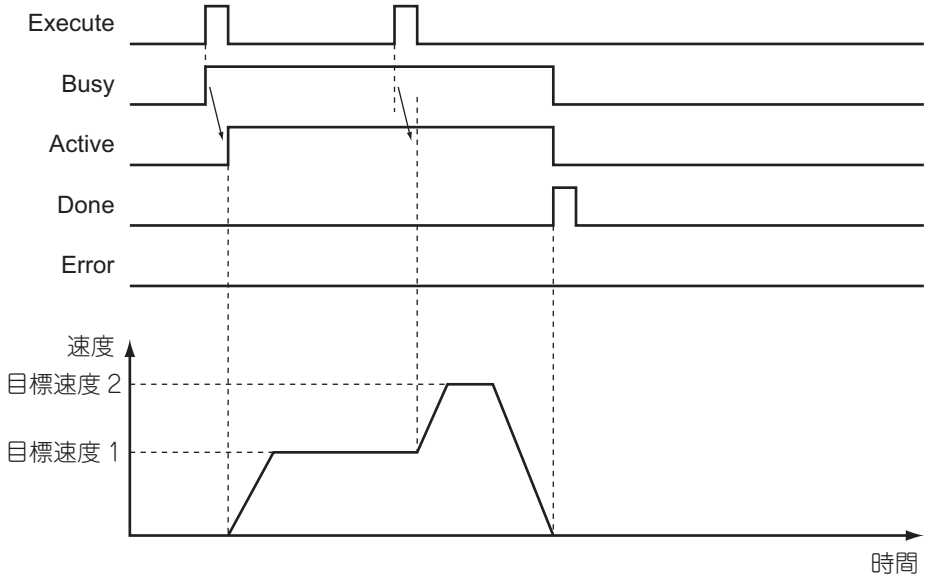
参考

MC_ZoneSwitch（ゾーン監視）命令や MC_AxesObserve（軸間偏差監視）命令など、「Enable」と「Enabled」が同じタイミングとなる命令もあります。命令ごとのタイミングの詳細は、『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)』を参照してください。

6-4-3 モーション制御命令の再起動時のタイミングチャート

モーション制御命令の実行中に同じインスタンスの入力変数の値を変更して、入力変数「Execute」（起動）を TRUE→FALSE→TRUE とすると、変更した入力変数の値で動作します。

MC_MoveAbsolute（絶対値位置決め）で速度を変更した場合を下図に示します。

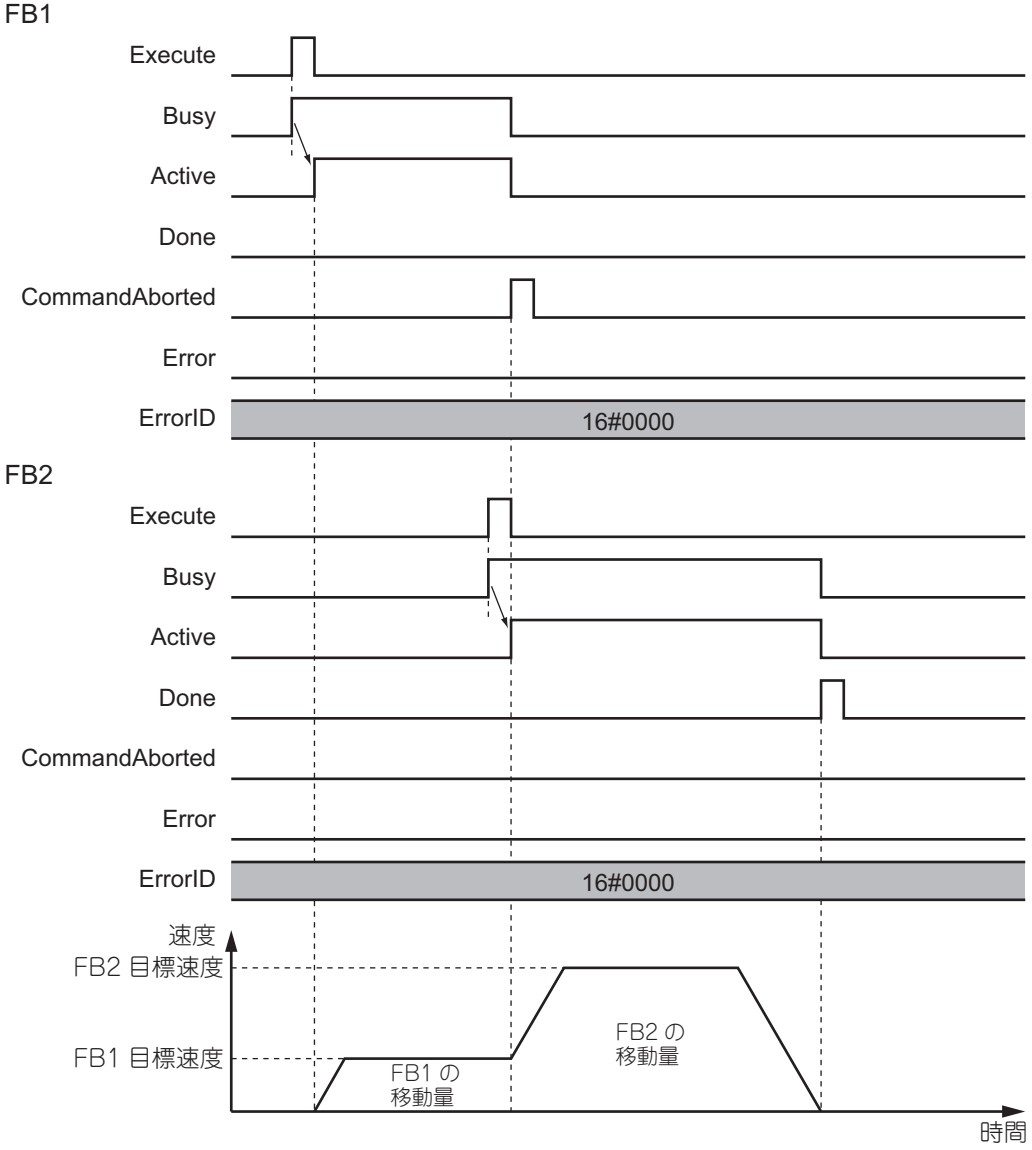


MC 機能モジュールでの命令再起動の詳細については、「9-5-6 モーション制御命令の再起動 (P.9-39)」、「9-7-4 多軸協調制御のモーション制御命令の再起動 (P.9-59)」を参照してください。

6-4-4 モーション制御命令の多重起動時のタイミングチャート

動作中の軸に対して別のインスタンスを起動することができます。
入力変数の「BufferMode」（バッファモード選択）を設定することにより、動作開始のタイミングを指定することができます。

下図は、「BufferMode」（バッファモード選択）をアボーティングに設定し、2 つの MC_MoveAbsolute（絶対値位置決め）命令を多重起動した場合です。
図中の「FB1」および「FB2」は、命令のインスタンス名です。



MC 機能モジュールでの命令多重起動については、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44）」、「9-7-5 多軸協調制御のモーション制御命令の多重起動（バッファモード）（P.9-60）」を参照してください。

6-5 位置の扱い

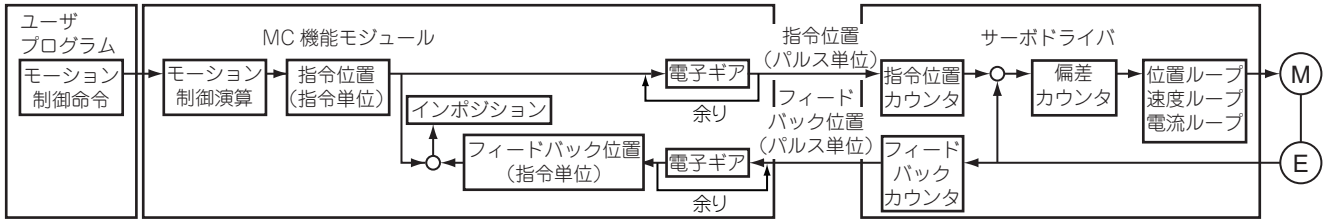
ここでは、モーション制御プログラムで扱う位置について説明します。

6-5-1 位置の種類

MC 機能モジュールには、つぎの 2 種類の位置があります。

位置種別	定義
指令位置	MC 機能モジュールが、軸を制御するために出力する位置です。
フィードバック位置	サーボドライバやカウンタから入力した実際の位置です。

EtherCAT スレーブサーボドライバの場合の指令位置とフィードバック位置の関係を下図に示します。



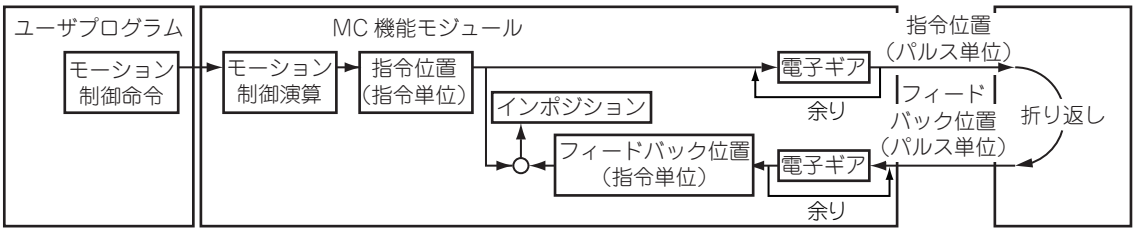
指令位置とフィードバック位置は下記の項目を共有しています。

項目	指令位置	フィードバック位置
カウントモード	リニアモードまたはロータリモードを設定します。	指令位置と同じカウントモードになります。
位置の単位	mm, μm, nm, inch, degree, pulse からいずれかを設定します。	指令位置と同じ単位になります。
ソフトウェアリミット	ソフトウェア的な動作可能範囲を設定します。	指令位置と同じ範囲になります。
現在位置変更	現在位置を任意の位置に変更します。	指令位置と同じ値が同時に設定されます。*1
原点確定	原点確定または原点未確定の状態を持ちます。	指令位置と同じ原点状態になります。

*1. 変更前に位置偏差があれば、位置偏差の値を保持した値がフィードバック位置に設定されます。

参考

- 仮想サーボ軸の場合、指令現在値をパルス単位に変換し、その値を再度、表示単位に変換した値をフィードバック現在値とします。



- NX シリーズ 位置インタフェースユニットの場合は、『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。

6-5-2 各軸種別で使用可能な位置の種類

軸種別で使用可能な位置の種類を下記に示します。

軸種別	位置の種類	
	指令位置	フィードバック位置
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 *2

*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。
*2. 実際のエンコーダが存在しない場合などに使用します。

6-6 モーション制御システム変数

ここでは、MC 機能モジュールに属する変数について説明しています。

6-6-1 モーション制御システム変数の概要

NJ/NX シリーズ CPU ユニットは IEC 61131-3 に準拠したコントローラです。
NJ/NX シリーズ CPU ユニットのユーザプログラムではパラメータ設定やステータス情報などのデータを変数として扱います。
これらの変数のうち、MC 機能モジュールに属する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。

モーション制御システム変数の種類

下表はモーション制御システム変数の種類の一覧です。

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。

● MC 共通変数

MC 機能モジュール共通の状態をモニタする変数です。変数名は「_MC_COM」です。

● 軸変数

EtherCAT スレーブのサーボドライバやエンコーダ入力ターミナル、NX シリーズ 位置インタフェースユニット、および仮想的なサーボドライバやエンコーダ入力を扱うための変数です。
ユーザプログラムでは、システム定義変数の変数名と Sysmac Studio で作成した変数名のどちらも使用することができます。
Sysmac Studio で作成した変数名は、軸ごとに任意の変数名に変更することができます。

a. システム定義変数の変数名は、次のようになります。
- _MC_AX[0]～_MC_AX[255]
- _MC1_AX[0]～_MC1_AX[255]
- _MC2_AX[0]～_MC2_AX[255]

b. Sysmac Studio で作成した変数名は、次のようになります。
- MC_Axis000～MC_Axis255（デフォルト）

● 軸グループ変数

複数の軸をまとめたグループを扱うための変数です。
ユーザプログラムでは、システム定義変数の変数名と Sysmac Studio で作成した変数名のどちらも使用することができます。
Sysmac Studio で作成した変数名は、軸グループごとに任意の変数名に変更することができます。

a. システム定義変数の変数名は、次のようになります。
- _MC_GRP[0]～_MC_GRP[63]
- _MC1_GRP[0]～_MC1_GRP[63]
- _MC2_GRP[0]～_MC2_GRP[63]

b. Sysmac Studio で作成した変数名は、次のようになります。

・ MC_Group000～MC_Group063（デフォルト）

NJ/NX シリーズ CPU ユニットで扱う変数については、『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。

モーション制御システム変数で扱うデータ型

モーション制御システム変数には、基本データ型と派生データ型があります。

● 基本データ型

分類	データ型	サイズ	値の範囲	記述方法
ブール	BOOL	2 *1	FALSE, TRUE	FALSE, TRUE
整数	UINT	2	0～+65535	2 進表現は、先頭に「2#」を付けます。*2 8 進表現は、先頭に「8#」を付けます。*3 10 進表現は、先頭に「10#」を付けます。*4 16 進表現は、先頭に「16#」を付けます。*5 先頭に何も付けなければ、10 進数として解釈されます。
	UDINT	4	0～+4294967295	
浮動小数点	LREAL	8	-1.79769313486231e+308 ～-2.22507385850721e-308, 0, 2.22507385850721e-308～ 1.79769313486231e+308, +∞/-∞	（符号）+整数部+（小数点）+（小数部）+（指数部）で記述します。*6 （ ）付きは、省略可

*1. BOOL 型のデータサイズは 1 ビットですが、メモリは 2 バイト分消費します。
*2. 2 進表現例: 2#1111_1111, 2#1110_0000
*3. 8 進表現例: 8#377, 8#340
*4. 10 進表現例: -12, 0, 123_456, +986, 10#1234
*5. 16 進表現例: 16#FF, 16#ff, 16#E0, 16#e0
*6. 例: 2, -12.0, 0.0, 0.4560, 3.14159_26, -1.34E-12, -1.34e-12, 1.0E+6, 1.0e+6, 1.234E6, 1.234e6

● 派生データ型

種類	内容
列挙型	あらかじめ用意された名前リスト内のひとつを値とするデータ型です。 データ型の先頭が「_e」のものです。
構造体型	複数のデータ型を階層的にひとつにまとめたデータ型です。 データ型の先頭が「_s」のものです。

NJ/NX シリーズ CPU ユニットで扱うその他の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 ソフトウェア編(SBCA-467)』を参照してください。

モーション制御システム変数の属性

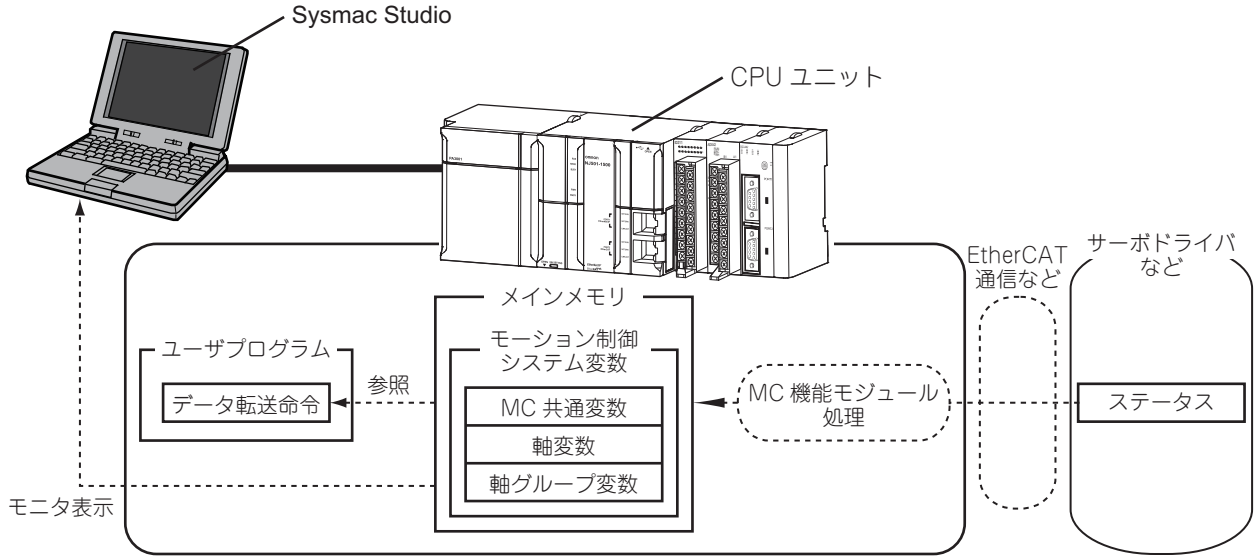
つぎの属性について、モーション制御システム変数はすべて同じ属性を持ちます。

属性	モーション制御システム変数の属性
グローバル／ローカル	グローバル変数
R/W アクセス	読出専用
保持	非保持
ネットワーク公開	公開 *1

*1. システム定義変数の変数名でネットワーク公開されます。Sysmac Studio で軸または軸グループを作成したときの変数名はネットワーク非公開です。

6-6-2　モーション制御システム変数のしくみ

モーション制御システム変数は、MC 機能モジュールの状態を示す情報、EtherCAT 通信で接続されたスレーブ機器や CPU ユニット上の NX ユニットのステータス情報、およびモーション制御を行うための MC パラメータ設定の一部から構成されています。
モーション制御システム変数は、ユーザプログラムから変数として参照することができ、Sysmac Studio でモニタ表示することもできます。



モーション制御システム変数の更新タイミング

モーション制御システム変数は、形 NX701 CPU ユニット、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 ユニットでは次のように更新タイミングが異なります。

● 形 NX701 CPU ユニットの場合

プライマリ定周期タスクと定周期タスク（実行優先度 5）に軸を割り付けた場合は以下のようになります。

- プライマリ周期ごとにプライマリ定周期タスクに割り付けたタスク用領域と定周期タスク（実行優先度 5）の基本設定領域だけが更新されます。
- 定周期タスク（実行優先度 5）のそのほかの領域は、初期値のまま更新されず、定周期タスク（実行優先度 5）のタスク周期で更新されます。

更新対象の変数領域	プライマリ定周期タスク	定周期タスク（実行優先度 5）
_MC_AX[0-255]	○	
_MC1_AX[0-255]	○	
_MC2_AX[0-255]		○
_MC_GRP[0-63]	○	
_MC1_GRP[0-63]	○	
_MC2_GRP[0-63]		○

● 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 ユニットの場合

モーション制御システム変数は、プライマリ周期ごとに更新されます。

6-6-3 モーション制御システム変数の一覧

ここでは、モーション制御システム変数の種類ごとに一覧表で説明しています。

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 からの試運転中のときに 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 部分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 部分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することができます。

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	

変数名	データ型	名称	機能
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中しているときに 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」の状態も含みます。
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。

変数名	データ型	名称	機能
Details	_sAXIS_REF_DET	軸制御ステータス	指令の制御状態を表す。
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。*1 Idle と InPosWaiting は排他的で、同時には TRUE になりません。
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。*2 FALSE: 原点未確定 TRUE: 原点確定
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 *3	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	指令している移動方向を表します。
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	ドライバなどの状態を表します。
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 *4 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。*5 *6
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。

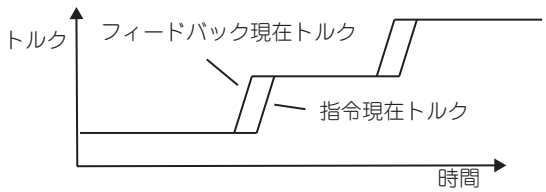
変数名	データ型	名称	機能
ILA	BOOL	ドライバ内部機能制限中	サーボドライバ側の制限機能により実際に制限されているときに TRUE となります。*7
CSP	BOOL	サイクリック同期位置(CSP)モード中	サーボドライバ側がサーボ ON で、CSP モードのときに TRUE となります。*8
CSV	BOOL	サイクリック同期速度(CSV)モード中	サーボドライバ側がサーボ ON で、CSV モードのときに TRUE となります。*8
CST	BOOL	サイクリック同期トルク(CST)モード中	サーボドライバ側がサーボ ON で、CST モードのときに TRUE となります。*8
Cmd	_sAXIS_REF_CMD_DATA	軸指令値	
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。*9
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。*10
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) *9
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。

変数名	データ型	名称	機能
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。
Cfg	_sAXIS_REF_CFG	軸基本設定	軸基本設定の設定内容を表します。
AxNo	UINT	軸番号	軸の論理番号を表します。
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。*12 16#FFFF は、なしを示します。
ExecID *13	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	電子ギア比の設定値を表します。
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。*14
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 *13	_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）

変数名			データ型	名称	機能
		MaxPos *13	LREAL	現在位置上限値	現在位置表示の上限値を表します。*15
		MinPos *13	LREAL	現在位置下限値	現在位置表示の下限値を表します。*16

*1. 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。
*2. 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。
*3. VelLimit は、同期制御中の従軸にだけ使用してください。
*4. サーボドライバの PDS 状態が「Ready to switch on」、「Switched on」、「Operation enabled」のいずれかの場合で、かつ主回路電源(voltage enabled)が ON している場合に TRUE となります。ただし、［主回路電源 OFF 検出］が［検出しない］に設定されている場合、主回路電源の ON/OFF は無視します。
PDS 状態、主回路電源 OFF 検出の詳細は、「A-5 PDS 状態遷移 (P.A-27)」を参照してください。
*5. 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。
*6. 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この信号を PDO にマッピングすることができません。
*7. 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。
*8. PDO にマッピングされた「ステータスワード(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 となります。
*9. 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。
*10. 指令現在トルクとフィードバック現在トルクをデータトレースで比較表示した場合、トルク制御モード以外では、指令現在トルクが 1 タスク周期後から変化するように表示されます。これは、データトレースの処理タイミングによる影響です。データトレースの処理タイミングについては、『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)』の「データトレース」を参照してください。

データトレース画面の例



指令現在トルクが、フィードバック現在トルクよりも遅れて変化するように表示されます。

*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することができます。
*12. 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。
*13. 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することができます。

*14. 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。
*15. 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。
*16. 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。

● 軸変数と軸種別の関係

軸変数については、軸種別ごとに有効／無効なメンバが異なります。無効なメンバを参照すると「FALSE」あるいは「0」が読み出されます。

ここでは、_MC_AX[0-255]を例にして説明しています。_MC1_AX[0-255]、_MC2_AX[0-255]の場合も同様です。
また、軸種別ごとに各メンバの有効／無効を一覧で記載しています。

○: 有効、－: 無効

変数名	データ型	名称	サーボ軸	仮想サーボ軸	エンコーダ軸	仮想エンコーダ軸
_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	○	○	－	－
Disabled	BOOL	軸無効	○	○	○	○
Standstill	BOOL	停止中	○	○	－	－
Discrete	BOOL	位置決め動作中	○	○	－	－
Continuous	BOOL	連続動作中	○	○	－	－
Synchronized	BOOL	同期動作中	○	○	－	－
Homing	BOOL	原点復帰中	○	○	－	－
Stopping	BOOL	減速停止中	○	○	－	－
ErrorStop	BOOL	エラー減速停止中	○	○	－	－
Coordinated	BOOL	多軸協調動作中	○	○	－	－
Details	_sAXIS_REF_DET	軸制御ステータス				
Idle	BOOL	停止中	○	○	○	○
InPosWaiting	BOOL	インポジション待ち	○	○	－	－
Homed	BOOL	原点確定	○	○	－	－
InHome	BOOL	原点停止	○	○	－	－
VelLimit	BOOL	指令速度飽和	○	○	－	－
Dir	_sAXIS_REF_DIR	指令方向ステータス				
Posi	BOOL	正方向指令中	○	○	－	－
Nega	BOOL	負方向指令中	○	○	－	－
DrvStatus	_sAXIS_REF_STA_DRV	サーボドライバ状態				
ServoOn	BOOL	サーボ ON	○	－	－	－
Ready	BOOL	サーボレディ	○	－	－	－
MainPower	BOOL	主回路電源	○	－	－	－
P_OT	BOOL	正方向限界入力	○	－	－	－
N_OT	BOOL	負方向限界入力	○	－	－	－
HomeSw	BOOL	原点近傍入力	○	－	－	－
Home	BOOL	原点入力	○	－	－	－
ImdStop	BOOL	即停止入力	○	－	－	－
Latch1	BOOL	外部ラッチ入力 1	○	－	－	－
Latch2	BOOL	外部ラッチ入力 2	○	－	－	－
DrvAlarm	BOOL	ドライバエラー入力	○	－	－	－

変数名	データ型	名称	サーボ軸	仮想サーボ軸	エンコーダ軸	仮想エンコーダ軸
DrvWarning	BOOL	ドライバワーニング入力	○	−	−	−
ILA	BOOL	ドライバ内部機能制限中	○	−	−	−
CSP	BOOL	サイクリック同期位置(CSP)モード中	○	−	−	−
CSV	BOOL	サイクリック同期速度(CSV)モード中	○	−	−	−
CST	BOOL	サイクリック同期トルク(CST)モード中	○	−	−	−
Cmd	_sAXIS_REF_CMD_DATA	軸指令値				
Pos	LREAL	指令現在位置	○	○	−	−
Vel	LREAL	指令現在速度	○	○	−	−
AccDec	LREAL	指令現在加減速度	○	○	−	−
Jerk	LREAL	指令現在ジャーク	○	○	−	−
Trq	LREAL	指令現在トルク	○	○	−	−
Act	_sAXIS_REF_ACT_DATA	軸現在値				
Pos	LREAL	フィードバック現在位置	○	○	○	○
Vel	LREAL	フィードバック現在速度	○	○	○	○
Trq	LREAL	フィードバック現在トルク	○	○	−	−
TimeStamp	ULINT	タイムスタンプ	○	−	○	−
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	○	○	○	○
Code	WORD	軸 軽度フォールトコード	○	○	○	○
Obsr	_sMC_REF_EVENT	軸 監視情報				
Active	BOOL	軸 監視情報発生中	○	○	○	○
Code	WORD	軸 監視情報コード	○	○	○	○
Cfg	_sAXIS_REF_CFG	軸基本設定				
AxNo	UINT	軸番号	○	○	○	○
AxEnable	_eMC_AXIS_USE	軸使用	○	○	○	○
AxType	_eMC_AXIS_TYPE	軸種別	○	○	○	○
NodeAddress	UINT	ノードアドレス	○	−	○	−
ExecID *1	UINT	実行 ID	○	○	○	○
Scale	_sAXIS_REF_SCALE	単位変換				
Num	UDINT	モータ 1 回転のパルス数	○	○	○	○
Den *2	LREAL	モータ 1 回転の移動量	○	○	○	○
Units	_eMC_UNITS	表示単位	○	○	○	○
CountMode *1	_eMC_COUNT_MODE	カウントモード	○	○	○	○
MaxPos *1	LREAL	現在位置上限値	○	○	○	○
MinPos *1	LREAL	現在位置下限値	○	○	○	○

*1. 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することができます。

*2. 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。

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]、_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 が TRUE（停止中） • 構成軸がサーボ ON 状態 • 構成軸の_MC_AX[*].Details.Homed が TRUE（原点確定）
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。 Disabled/ 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 命令で軸が停止した後に Execute が TRUE のときも含みます。 本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(_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	指令の制御状態を表します。
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。*1 Idle と InPosWaiting は排他的で、同時には TRUE になりません。

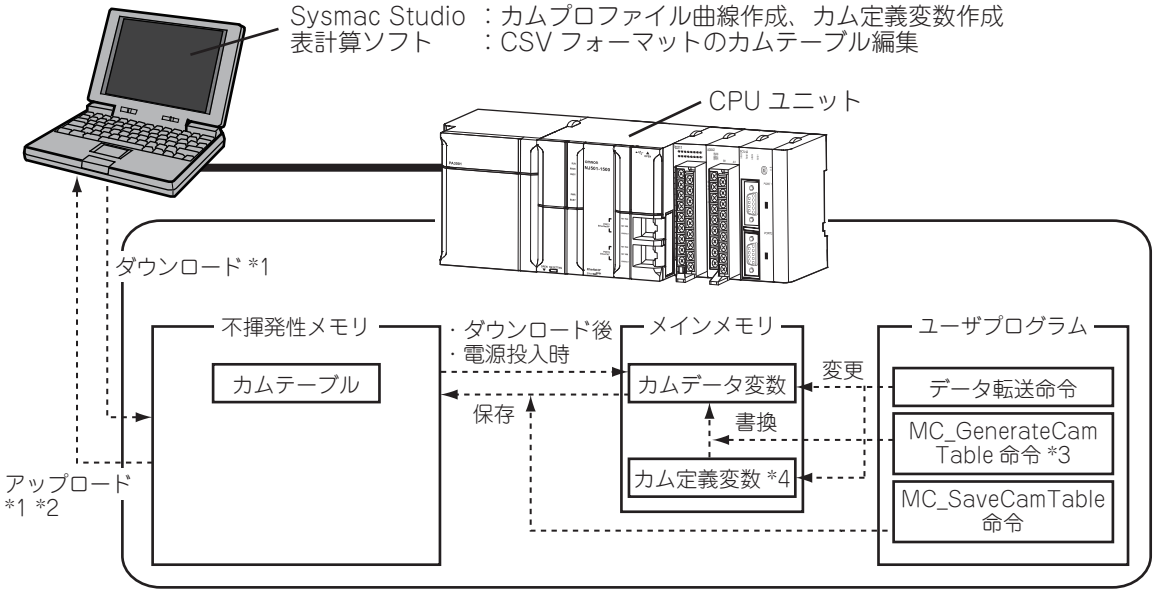
変数名	データ型	名称	機能
InPosWaiting	BOOL	インポジション待ち	いずれかの構成軸がインポジション待ちのときに TRUE となります。 インポジションチェックを行う位置決めで、インポジションチェックを行っている状態です。*2
Cmd	_sGROUP_REF_CMD_DATA	軸グループ指令値	
Vel	LREAL	指令補間速度	指令補間速度の現在値を出力します。 補間指令現在位置の差分から算出した補間速度です。正方向移動時はプラス符号、負方向移動時はマイナス符号がつきます。 軸グループが無効の場合は「0」となります。
AccDec	LREAL	指令補間加減速度	指令補間加減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。
Code	WORD	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	軸グループ基本設定の設定内容を表します。
GrpNo	UINT	軸グループ番号	軸グループの論理番号を表します。
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	軸グループの運動学変換のための定義を表します。
GrpType	_eMC_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（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（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（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（軸 A3）	軸 A3 に割り付けられた軸番号を表します。

*1. 速度「0」での動作も演算している状態に含まれます。
*2. 軸グループ内の全構成軸が、各軸パラメータのインポジション幅の範囲内に入ると FALSE となります。
*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することができます。

6-7　カムテーブルとカムデータ変数

MC 機能モジュールでは、Sysmac Studio のカムエディタ機能で作成したカムプロファイル曲線をカムテーブルとして使用することができます。
NJ/NX シリーズ CPU ユニットのユーザプログラムではカムテーブルのデータをカムデータ変数として扱います。

カムテーブルの作成と保存のしくみ



*1.「ダウンロード」および「アップロード」は、Sysmac Studio の「同期」機能を使用します。
*2. アップロードしたカムデータ変数は、カムエディタ機能で編集できません。ユーザプログラムで編集したカムデータ変数をパソコンで再編集する方法は、「ユーザプログラムで編集したカムデータ変数をパソコンで再編集する (P.6-33)」を参照してください。
また、Sysmac Studio でリビルドしたり、カムプロファイル曲線を編集したりすると、アップロードしたカムデータ変数は、カムプロファイル曲線のデータで上書きされます。
*3. MC_GenerateCamTable（カムテーブル生成）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することができます。
*4. カム定義変数は、カムプロパティ変数とカムノード変数の総称です。

● カムテーブルのデータフロー

- Sysmac Studio で作成したカムプロファイル曲線を CPU ユニットへダウンロードすると、カムテーブルとして CPU ユニット内蔵の不揮発性メモリに保存されます。
また Sysmac Studio にカムテーブルをアップロードすると、不揮発性メモリに保存されたカムテーブルがアップロードされます。
- 不揮発性メモリに保存されたカムテーブルはダウンロード後や電源投入時に、カムデータ変数としてメインメモリに反映されます。
- ユーザプログラムにより、メインメモリのカムデータ変数およびカム定義変数を編集することができます。カムデータ変数およびカム定義変数については、「9-2-5 カムテーブル (P.9-15)」を参照してください。
- ユーザプログラム中の MC_GenerateCamTable（カムテーブル生成）命令は、カム定義変数の値に従ってメインメモリのカムデータ変数を書き換えることができます。
- モーション制御命令の MC_SaveCamTable（カムテーブル保存）命令は、メインメモリのカムデータ変数を不揮発性メモリに保存します。

- Sysmac Studio のカムデータ設定で作成したカム定義変数は、ユーザプログラムで変更した場合でもアップロードできます。ユーザ定義変数として作成したカム定義変数は、ユーザプログラムで変更した場合にはアップロードできません。
- Sysmac Studio でのカム定義変数の作成と転送については、『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。
- MC_GenerateCamTable（カムテーブル生成）命令および MC_SaveCamTable（カムテーブル保存）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 モーション編(SBCE-434)』を参照してください。
- カムテーブルのダウンロードおよびアップロードは、NJ/NX シリーズ CPU ユニットのモードや MC 機能モジュールの状態にかかわらず実行可能です。ただし、カムテーブル保存中はカムデータのダウンロードおよびアップロードのほか、オンライン開始の操作や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、オンラインエディットなどを行っている場合は MC_SaveCamTable（カムテーブル保存）命令は使用できません。
- ダウンロードを開始すると、動作中の軸は最大減速度で減速します。

使用上の注意

ユーザプログラムで変更したカムデータは、電源の再投入または Sysmac Studio からのダウンロードにより、不揮発性メモリのカムテーブルの内容に戻ります。また、Sysmac Studio でこれらのデータをアップロードすることはできません。

バージョン関連情報

ユニットバージョン Ver.1.13 以降の CPU ユニットと Sysmac Studio Ver.1.17 以降の組み合わせで、ダウンロード中も I/O デバイスへの指令を継続することが可能です。
ユニットバージョン Ver.1.12 以前の CPU ユニットの場合は、ダウンロード中の I/O デバイスへの指令は、停止します。
ダウンロード開始時に、I/O デバイスへ指令を停止するか、継続するかの設定は、『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。

I/O デバイスへの指令を停止する場合

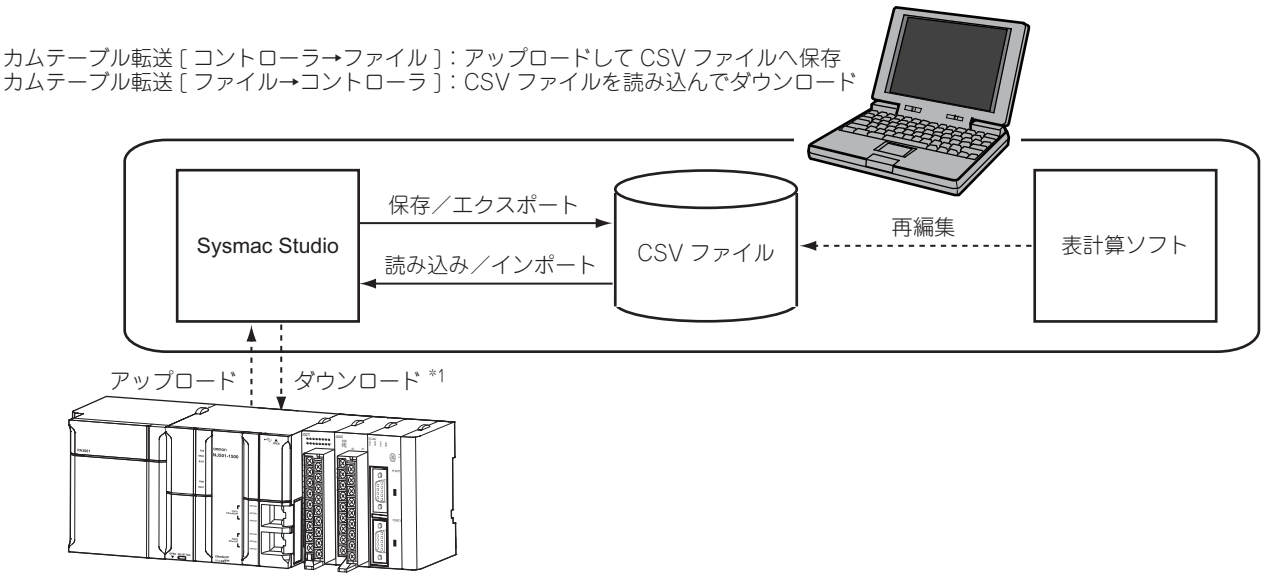
ダウンロード中は、サーボ OFF となり、軸状態は、「Disable」（軸無効）となります。

I/O デバイスへの指令を継続する場合

MC_Power（運転可）命令および MC_SetTorqueLimit（トルク制限）命令は、ダウンロード中も動作可能です。
ダウンロード中は、ダウンロード直前の MC_Power 命令によるサーボ ON 状態、および MC_SetTorqueLimit 命令による出力トルクの制限を継続します。
また、ダウンロードによりユーザプログラムが更新され、MC_Power 命令および MC_SetTorqueLimit 命令が削除された場合も、サーボ ON 状態およびトルク制限を継続します。

ユーザプログラムで編集したカムデータ変数をパソコンで再編集する

ユーザプログラムでカムデータ変数を編集または書き換え、MC_SaveCamTable（カムテーブル保存）命令で不揮発性メモリに保存したカムテーブルは、Sysmac Studio のカムエディタ機能でデータを再編集できません。ここでは表計算ソフトなどでデータを再編集しカムテーブルとして使用する方法を説明します。



*1.「ダウンロード」および「アップロード」は、Sysmac Studio の「同期」機能を使用します。

● 不揮発性メモリに保存したカムテーブルを CSV ファイルへ保存する

- Sysmac Studio の［カムデータ設定］で再編集するカムプロファイルを右クリックし、メニューから［カムテーブル転送［コントローラ→ファイル］］を選択します。
- ［保存］ダイアログボックスでファイルの保存先とファイル名を指定し、［保存］ボタンをクリックします。

● CSV ファイルを再編集する

- 表計算ソフトなどを使用してデータを編集し CSV ファイルで保存します。

● CSV ファイルを CPU ユニットへ転送する

- ダウンロードするカムプロファイルを右クリックし、メニューから［カムテーブル転送［ファイル→コントローラ］］を選択します。
- ［ファイルを開く］ダイアログボックスで、転送するファイル名を指定し、［OK］ボタンをクリックします。
- 転送したカムテーブルを有効にするために、転送完了後にコントローラリセットまたはコントローラの電源再投入を行ってください。

使用上の注意

- ［カムテーブル転送［ファイル→コントローラ］］を行う前にコントローラと同期してください。
- ［カムテーブル転送［ファイル→コントローラ］］でカムテーブルを転送した後に、同期機能でカムデータ設定をコントローラへ転送すると、カムデータ設定のデータに変更されます。この場合は、再度［カムテーブル転送［ファイル→コントローラ］］で転送するか、カムデータ設定を同期の対象外としてください。

カムエディタで作成したカムデータ設定を CSV ファイルへエクスポートすることもできます。
カムデータ設定と操作方法の詳細については、『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。

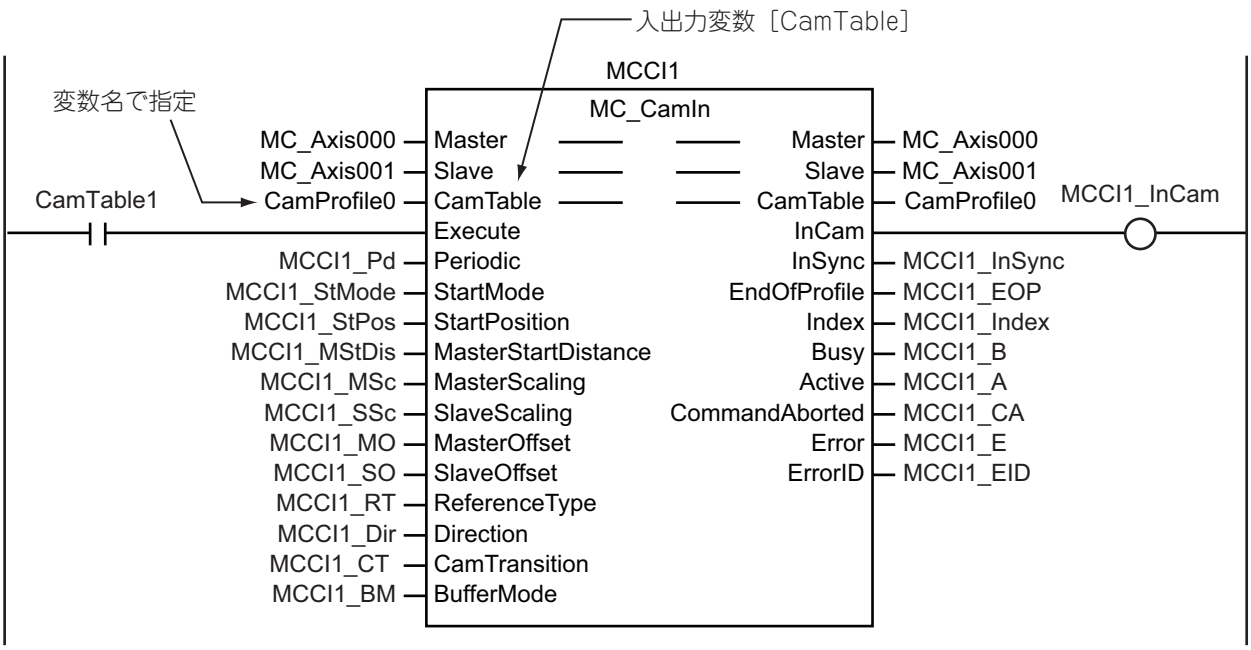
カムプロファイル曲線の名前

Sysmac Studio でカムプロファイルを追加すると、［CamProfile0］という名前がデフォルトで設定されます。末尾の数字には、カムプロファイルを追加するごとに追番が設定されます。
これらの名前は、Sysmac Studio でカムプロファイルごとに任意の名前に変更することができます。

ここで設定したカムプロファイルの名前が、カムテーブル名として扱われます。

ユーザプログラムでの指定方法

ユーザプログラムでは、モーション制御命令の入出力変数「CamTable」にカムテーブル名を指定します。



6-8 モーション制御プログラムの作成方法

NJ/NX シリーズ CPU ユニットのユーザプログラム中にモーション制御命令を記述することで、モーション制御を実行することができます。モーション制御命令を記述したプログラムをモーション制御プログラムとよびます。

使用上の注意

- Sysmac Studio では、いずれの形式の CPU ユニットでも最大 256 軸の設定とプログラミングができます。ただし、CPU 形式の最大制御軸数を超えるプロジェクトを CPU ユニットへダウンロードすることはできません。
- プロジェクトを流用する場合などは、CPU ユニットの形式を確認し最大制御軸数を超えないようにしてください。
- 軸使用が［未使用軸］の軸も制御軸数の対象となります。

プログラミングの詳細については、『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。

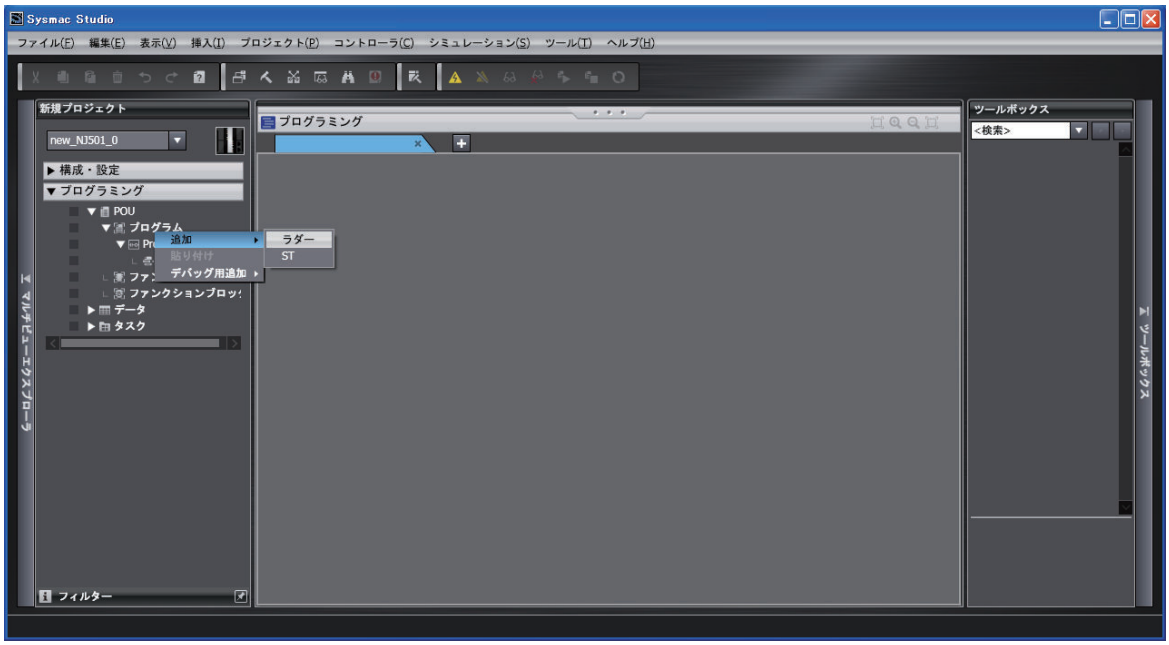
ここでは、Sysmac Studio を使用して既存のプロジェクトにプログラムを追加する手順を説明します。

1 **Sysmac Studio の起動**

Sysmac Studio を起動して既存プロジェクトを開きます。

2 **プログラムの追加**

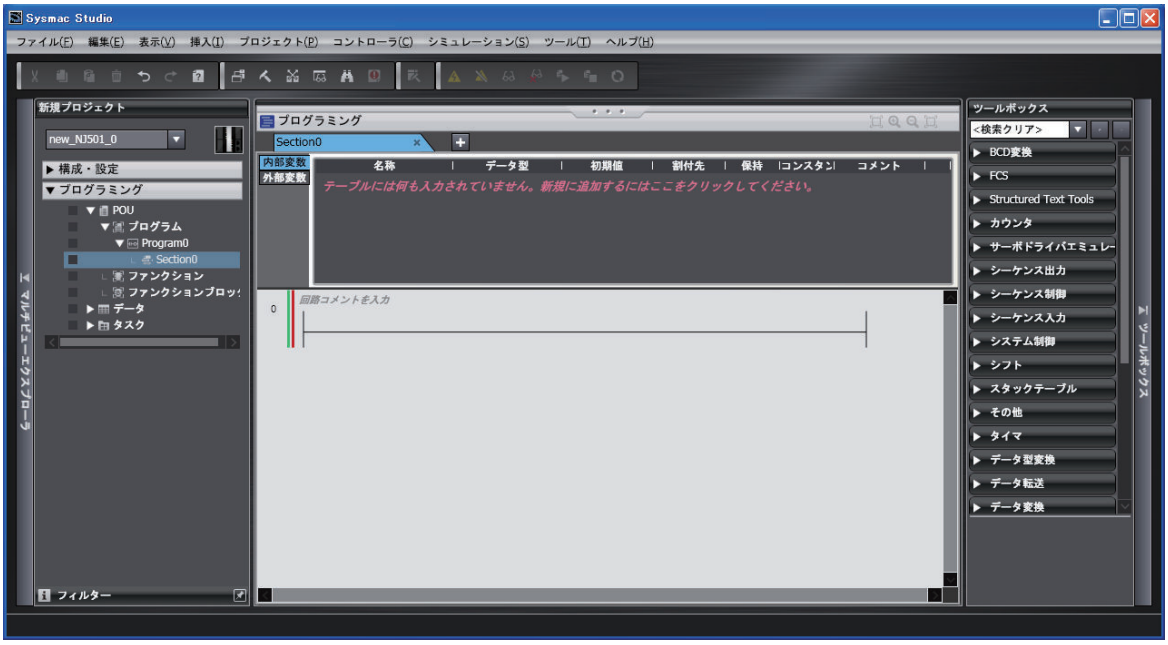
マルチビューエクスプローラの［プログラム］を右クリックし、［追加］から［ラダー］または［ST］をクリックします。



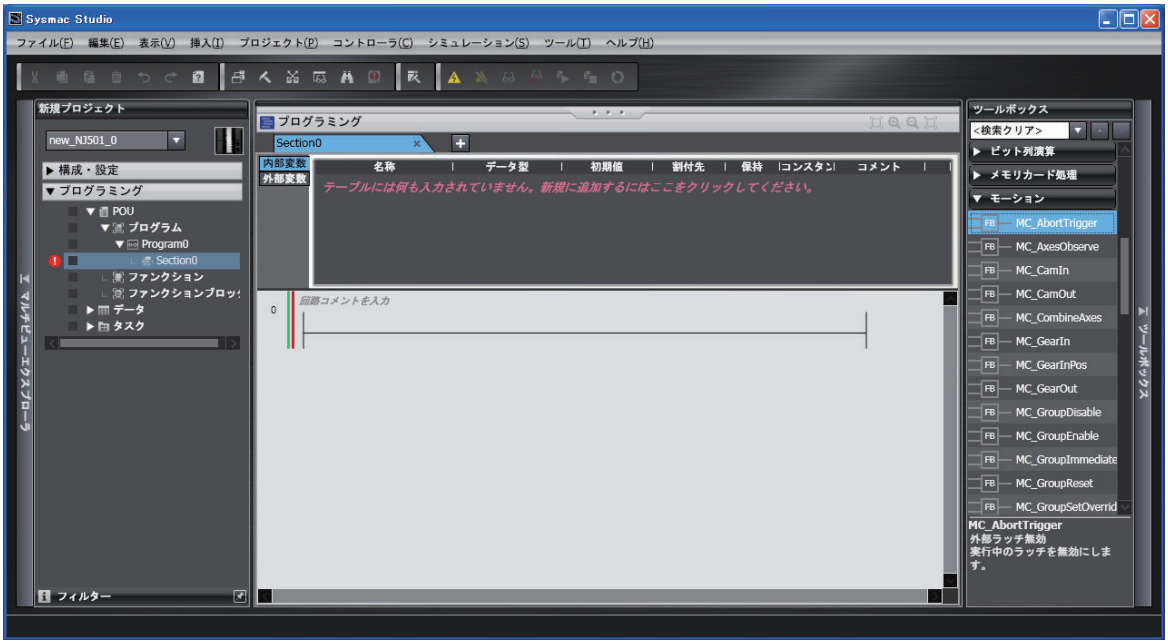
マルチビューエクスプローラにプログラムが追加されます。

3 **プログラムの編集**

追加されたプログラムの下にあるセクションを右クリックして［編集］をクリックすると、プログラムの編集画面が開きます。



ツールボックスから命令を選択して、必要なプログラムを作成してください。



プログラミングの詳細については、『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)』参照してください。

操作方法の詳細については、『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。

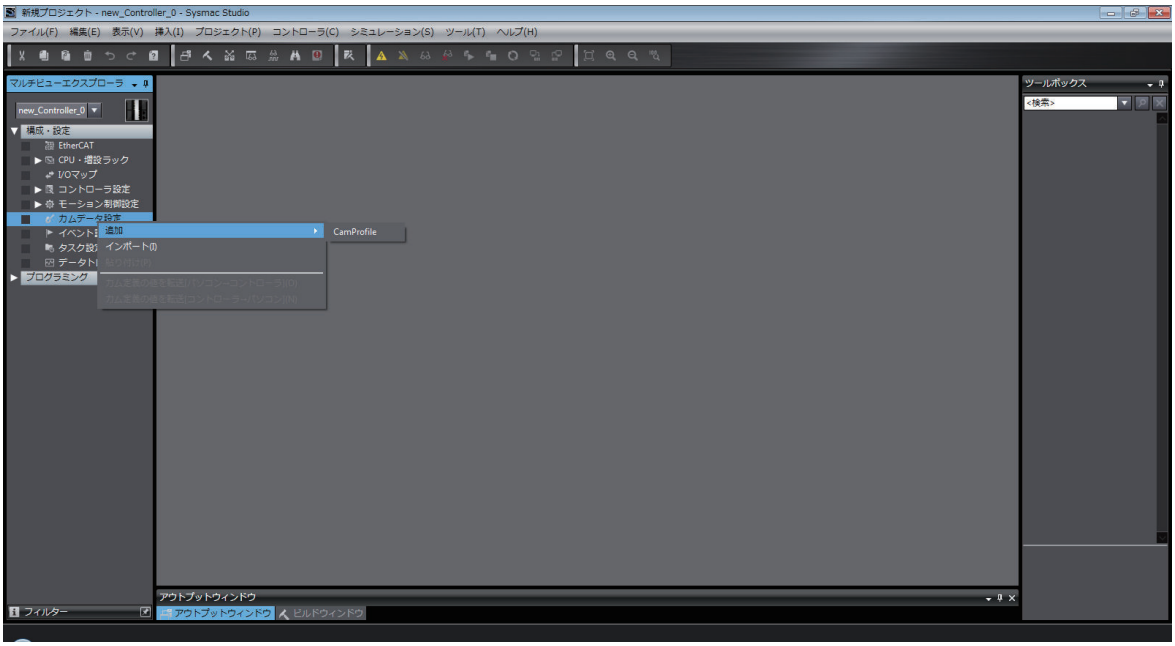
6-9 カムテーブルの作成方法

ここでは Sysmac Studio のカムエディタ機能を使用して、カムテーブルを作成する手順を説明します。

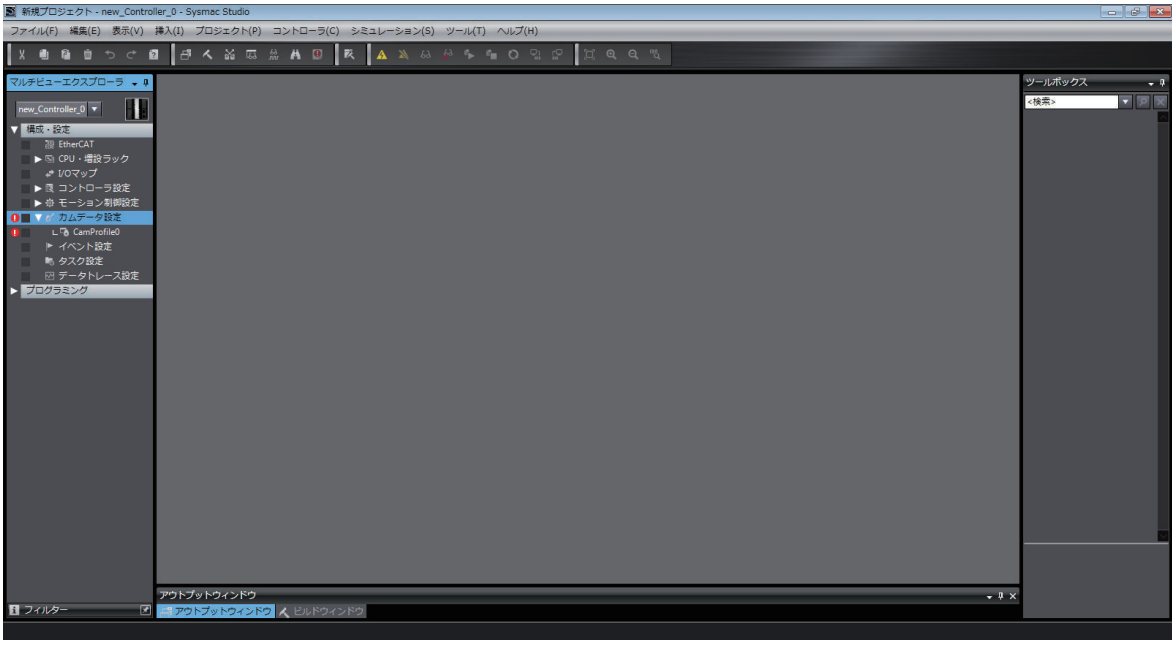
カムエディタ機能の詳細については、『Sysmac Studio Version 1 オペレーションマニュアル (SBCA-470)』を参照してください。

1 カムプロファイルの追加

マルチビューエクスプローラの［カムデータ設定］を右クリックし、［追加］から［CamProfile］をクリックします。

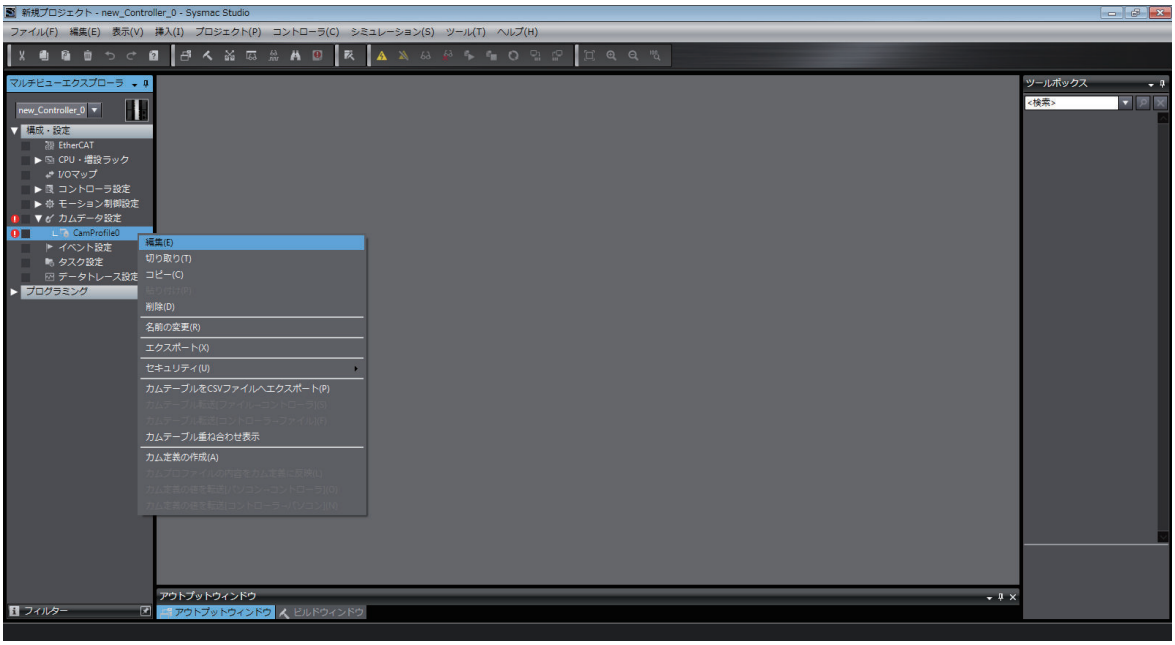


マルチビューエクスプローラにカムプロファイルが追加されます。追加されたカムプロファイルの名前は、デフォルトの「CamProfile0」から任意の名前に変更できます。

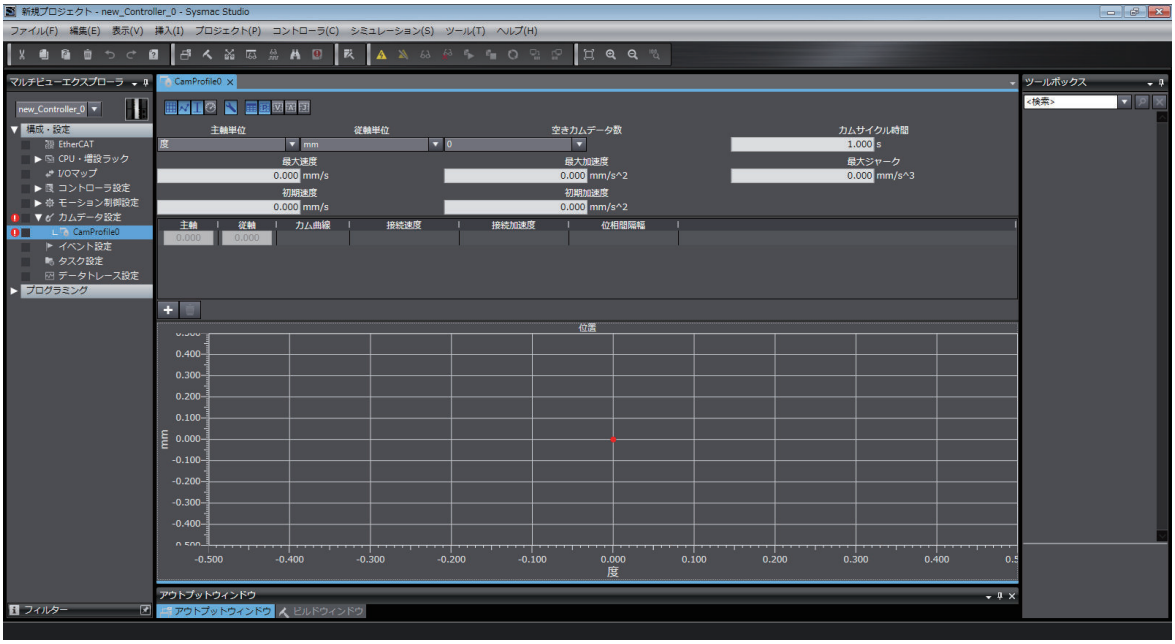


2 カムプロファイルの編集

マルチビューエクスプローラに追加されたカムプロファイルを右クリックし、[編集] をクリックします。



カムプロファイルの編集画面が表示されます。



各項目を設定してカムプロファイルを編集してください。

操作方法の詳細については、『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。

7

手動運転

オムロン製サーボドライバ 1S シリーズとの組み合わせで、手動操作のための機能について説明します。

7-1 概要 7-2

7-2 サーボ ON 7-3

7-2-1 サーボ ON の手順 7-3
7-2-2 軸パラメータの設定 7-4
7-2-3 プログラム例 7-4

7-3 ジョグ送り 7-5

7-3-1 ジョグ送りの手順 7-5
7-3-2 軸パラメータの設定 7-6
7-3-3 入力変数の設定例 7-6
7-3-4 プログラム例 7-7

7-1 概要

MC 機能モジュールとオムロン製サーボドライバ 1S シリーズとの組み合わせで、ユーザプログラムからモーション制御命令を使用して手動運転を行う場合について説明します。

手動運転のモーション制御命令は、サーボドライバをサーボ ON 状態にする MC_Power（運転可）命令とジョグ送りを行う MC_MoveJog（ジョグ送り）命令です。

使用上の注意

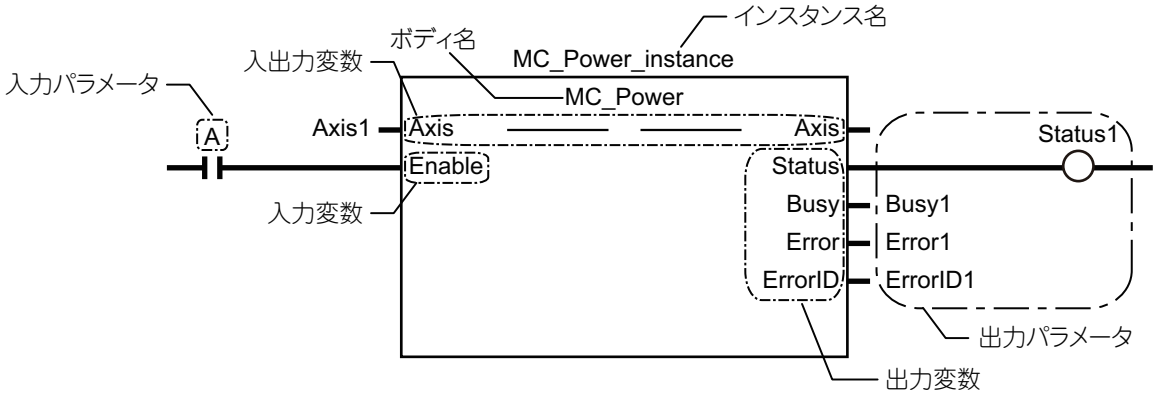
手動運転を行うためには、軸の設定が必要です。
軸の設定方法については、「第 3 章 軸と軸グループの設定 (P.3-1)」を参照してください。

参考

- プログラムを組まずに手動運転を行う場合は、Sysmac Studio の MC 試運転機能を使用してください。
  Sysmac Studio を使った手動運転については、「4-3 モータ動作の確認 (P.4-8)」を参照してください。
- ユーザプログラムの作成方法は、「第 6 章 モーション制御プログラム (P.6-1)」を参照してください。
- オムロン製サーボドライバ G5 シリーズを使用する場合も手順や操作は同じです。

7-2 サーボ ON

サーボドライバに対する運転指令を ON/OFF する機能です。
モーション制御命令の MC_Power（運転可）命令を使用します。



動作させたい軸は、入出力変数である「Axis」（軸）で指定します。
MC_Power（運転可）命令の入力変数である「Enable」（有効）を TRUE にすることにより、サーボ ON 状態となります。「Enable」（有効）を FALSE にすることにより、サーボ OFF 状態となります。

使用上の注意

- 「Enable」（有効）を軸の動作中に FALSE にした場合は、指令を即停止させ、その軸に対するモーション制御命令が無効となります。
- NX シリーズパルス出力ユニットを使用する場合は、モータドライバの通電 ON/OFF を別途行う必要があります。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。

参考

オムロン製サーボモータ 1S シリーズおよび G5 シリーズの絶対値エンコーダ付きサーボモータ、またはオムロン製モータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプで絶対値タイプの外部スケールを使用する場合、MC_Power（運転可）命令の入力変数「Enable」（有効）を TRUE にすると、原点確定します。

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、オムロン製サーボモータ 1S シリーズおよび G5 シリーズの絶対値エンコーダ付きサーボモータ、またはオムロン製モータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプで絶対値タイプの外部スケールを使用する場合、前述の「参考」に加えて、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、原点確定します。

7-2-1 サーボ ON の手順

1 **軸の追加と設定**

Sysmac Studio で軸の追加と設定を行います。
詳細については、「3-2-2 設定手順 (P.3-9)」を参照してください。

2 **軸パラメータの設定**

Sysmac Studio で軸パラメータの設定を行います。
詳細については、「3-2-2 設定手順 (P.3-9)」を参照してください。

3 **ユーザプログラムの作成**
Sysmac Studio でユーザプログラムを作成します。
詳細については、「6-8 モーション制御プログラムの作成方法 (P.6-36)」を参照してください。

4 **軸パラメータとユーザプログラムのダウンロード**
設定した軸パラメータとユーザプログラムを CPU ユニットへダウンロードします。
CPU ユニットへのダウンロードは、Sysmac Studio の「同期」機能を使用します。
詳細については、「3-2-2 設定手順 (P.3-9)」を参照してください。

5 **ユーザプログラムの起動**
ユーザプログラムを起動し、MC_Power 命令の入力変数「Enable」を TRUE にします。
サーボドライバがサーボ ON 状態になります。

7-2-2 軸パラメータの設定

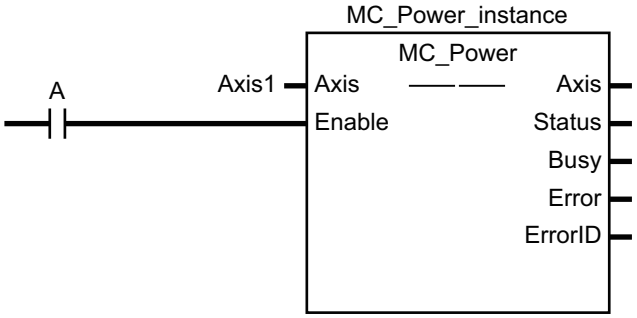
サーボ ON 状態にするだけの場合、設定が必要な軸パラメータは、次の設定項目だけです。
下表は設定例です。

設定項目	設定内容
軸変数名	Axis1 *1
軸番号	1 *2
軸使用	使用軸
軸種別	サーボ軸
入力デバイス／出力デバイス	1 *3

*1. 複数軸の場合は、軸ごとに異なる変数名を設定してください。
*2. 複数軸の場合は、軸ごとに異なる値を設定してください。
*3. サーボドライバのノードアドレスと同じ値を設定してください。複数軸の場合は、軸ごとに異なる値を設定してください。

7-2-3 プログラム例

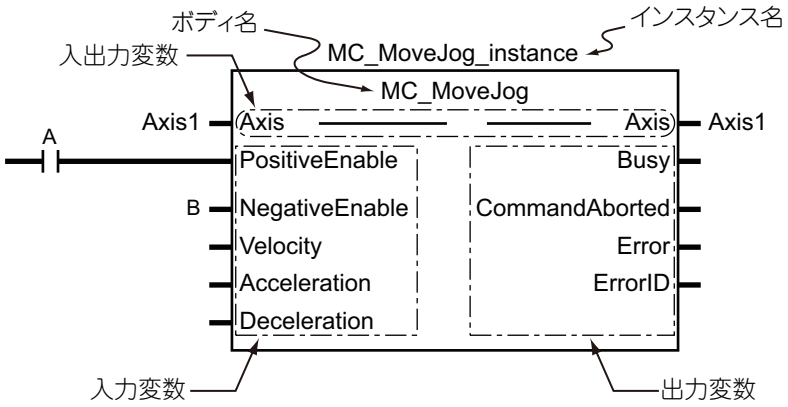
接点 A によって、Axis1 という軸のサーボ ON/OFF を切り替えるサンプルプログラムは以下のようになります。



MC_Power (運転可) 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 モーション編(SBCE-434)』を参照してください。

7-3 ジョグ送り

ジョグ送りは、モーション制御命令の MC_MoveJog（ジョグ送り）命令を使用します。



ジョグ送りしたい軸は、入出力変数の「Axis」（軸）で指定します。
入力変数の「PositiveEnable」を TRUE にすると、正方向に指定した「Velocity」（目標速度）、「Acceleration」（加速度）で軸起動します。「PositiveEnable」を FALSE にすると指定した「Deceleration」（減速度）で減速し、停止します。
同様に入力変数の「NegativeEnable」を TRUE にすると、負方向に起動し、FALSE で停止します。
ジョグ送りは、原点未確定の状態でも実行できます。

7-3-1 ジョグ送りの手順

1 **軸の追加と設定**
Sysmac Studio で軸の追加と設定を行います。
詳細については、「3-2-2 設定手順 (P.3-9)」を参照してください。

2 **軸パラメータの設定**
Sysmac Studio で軸パラメータの設定を行います。
詳細については、「3-2-2 設定手順 (P.3-9)」を参照してください。

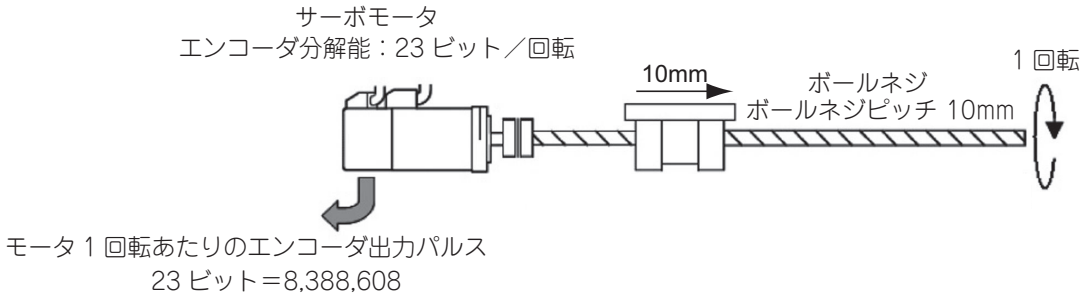
3 **ユーザプログラムの作成**
Sysmac Studio でユーザプログラムを作成します。
詳細については、「6-8 モーション制御プログラムの作成方法 (P.6-36)」を参照してください。

4 **軸パラメータとユーザプログラムのダウンロード**
設定した軸パラメータとユーザプログラムを CPU ユニットへダウンロードします。
CPU ユニットへのダウンロードは、Sysmac Studio の「同期」機能を使用します。
詳細については、「3-2-2 設定手順 (P.3-9)」を参照してください。

5 **ユーザプログラムの起動**
ユーザプログラムを起動し、MC_Power（運転可）命令の入力変数「Enable」を TRUE にし、サーボドライバをサーボ ON 状態にします。
MC_MoveJog（ジョグ送り）命令の入力変数「PositiveEnable」または「NegativeEnable」を TRUE にし、ジョグ送りを行います。

7-3-2 軸パラメータの設定

原点未確定状態でジョグ送りを実行する場合、次の軸パラメータを設定してください。
1 軸の装置を例に設定例を説明します。



設定項目	設定内容
軸変数名	Axis1 *1
軸番号	1 *2
軸使用	使用軸
軸種別	サーボ軸
入力デバイス／出力デバイス	1 *3
モータ 1 回転のパルス数	8,388,608 *4
モータ 1 回転の移動量	10,000 *4
ソフトウェアリミット機能	フィードバック位置に対して有効
表示単位	μm
カウントモード	リニアモード
最高速度	500,000 *5
JOG 最高速度	50,000 *6
最大加速度	5,000,000 *7
最大減速度	5,000,000 *7

*1. 複数軸の場合は、軸ごとに異なる変数名を設定してください。
*2. 複数軸の場合は、軸ごとに異なる値を設定してください。
*3. サーボドライバのノードアドレスと同じ値を設定してください。複数軸の場合は、軸ごとに異なる値を設定してください。
*4. 位置の指令単位は 1(μm)となります。
*5. 最高速度は、3,000r/min = 30m/min = 0.5m/s = 500,000 μm/s としています。
*6. ジョグ最高速度は、最高速度の 10%の 0.05m/s = 50,000 μm/s としています。
*7. 最大加速度および最大減速度は、5m/s^2 です。最高速度の 3,000r/min までの加速時間を 0.1s としています。

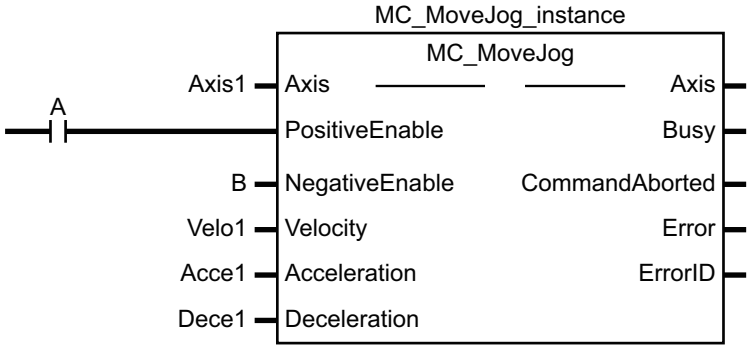
7-3-3 入力変数の設定例

ここでは、MC_MoveJog（ジョグ送り）命令の入力変数「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）の設定について説明します。

- たとえば 0.03m/s (30,000 μm/s)の速度でジョグ送りをするには「Velocity」に「30,000」を指定します。
- また、3m/s^2 (3,000,000 μm/s^2)で加減速するには「Acceleration」と「Deceleration」に「3,000,000」を指定します。

7-3-4 プログラム例

接点 A によって Axis1 という軸を正方向に、接点 B によって負方向にジョグ送りさせるサンプルプログラムは以下のようになります。



このときの「Velocity」（目標速度）は「Velo1」、「Acceleration」（加速度）は「Acce1」、「Deceleration」（減速度）は「Dece1」となります。

入力変数の設定例の条件で動作させるには、あらかじめユーザプログラムで各変数の値を設定しておきます。

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

MC_MoveJog（ジョグ送り）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 モーション編(SBCE-434)』を参照してください。

8

原点決め

この章では、原点決めについて説明します。

8-1　原点決めの概要 8-2

8-2　原点復帰の手順 8-5

8-2-1　原点復帰パラメータの設定 8-5

8-2-2　原点復帰のモニタ 8-11

8-3　原点復帰の動作 8-12

8-4　絶対値エンコーダ原点設定 8-13

8-4-1　機能概要 8-15

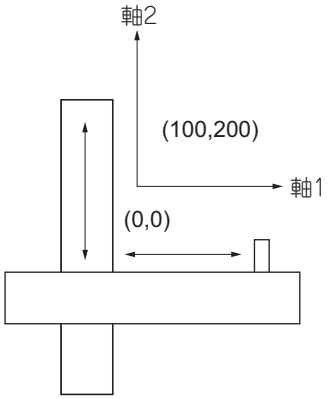
8-4-2　設定手順 8-16

8-5　高速原点復帰 8-18

8-1 原点決めの概要

位置決めを行うシステムで、システムの絶対的な位置に対して位置決めを行う場合、最初に原点決めを行う必要があります。

たとえば、下図のような XY ステージで、（軸 1、軸 2）＝（100mm, 200mm)の位置に位置決めしたい場合、機械の原点 (0, 0)を決定しなければなりません。
この機械原点を決定することを、原点決めと呼びます。



MC 機能モジュールでは、機械原点を決める方法としてモーション制御命令の MC_Home（原点復帰）命令、または MC_HomeWithParameter（パラメータ指定原点復帰）命令を使用します。

名称	内容
原点復帰	モータを実際に動かして、限界センサ、原点近傍センサ、原点入力信号を用いて機械原点を決めます。 原点入力信号として、近接センサや、エンコーダの Z 相等を使います。

使用上の注意

- 次の場合、原点未確定となります。
  - MC_SetPosition（現在位置変更）命令の実行
  - リニアモード時のオーバーフロー、およびアンダーフロー
  - 原点復帰開始時
  - EtherCAT 通信の制御状態が、オペレーショナル状態以外
- 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、一部の機能に制約があります。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。

参考

オムロン製サーボモータ 1S シリーズおよび G5 シリーズの絶対値エンコーダ付きサーボモータ、またはオムロン製モータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプで絶対値タイプの外部スケールを使用する場合、MC_Power（運転可）命令の入力変数「Enable」（有効）を TRUE にすると、原点確定します。
絶対値エンコーダによる原点設定については、「8-4 絶対値エンコーダ原点設定 (P.8-13)」を参照してください。

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、オムロン製サーボモータ 1S シリーズおよび G5 シリーズの絶対値エンコーダ付きサーボモータ、またはオムロン製モータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプで絶対値タイプの外部スケールを使用する場合、前述の「参考」に加えて、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、原点確定します。

また、前記の方法で決定した原点に位置決めする機能として、下記の MC_MoveZeroPosition（高速原点復帰）命令も搭載しています。

名称	内容
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。

なお、MC 機能モジュールは、原点未確定状態でもモータを動作することはできます（高速原点復帰を除く）。

機能	動作
ジョグ送り、速度制御	原点未確定状態の場合は、電源投入時を「0」として移動します。
高速原点復帰	使用できません。 使用した場合、命令の異常となります。
位置決め	原点未確定状態の場合は、電源投入時を「0」として移動します。
割り込み定寸位置決め	
カム動作開始	
ギア動作開始	
台形パターンカム	
加減算位置決め	
トルク制御	
ゾーン監視	
直線補間	使用できません。 使用した場合、命令の異常となります。
円弧補間	

参考

原点未確定時には、ソフトウェアリミット機能は無効となります。

各操作、またはイベントに対する原点確定状態の変化を以下に示します。

操作またはイベント	変化条件	原点確定状態の変化
サーボ ON 軸有効時	インクリメンタルエンコーダ	変化なし
	絶対値エンコーダで絶対値データ正常読出時	原点確定
	絶対値エンコーダで絶対値データ読出失敗時	原点未確定
現在位置変更	起動時	原点未確定
原点復帰	起動時	原点未確定
	完了時	原点確定
オーバーフロー／アンダーフロー	発生時	原点未確定

使用上の注意

- 仮想軸については、軸パラメータの［原点復帰動作］の設定に関係なく、［原点プリセット］として原点確定します。
- MC 機能モジュールは、サーボドライバの正転側駆動禁止入力(POT)、逆転側駆動禁止入力(NOT)、原点近傍入力(DEC)を、それぞれ正方向限界入力、負方向限界入力、原点近傍入力として使用します。
  これらの入力信号に関しては、サーボドライバが検出できる信号幅を確保し、MC 機能モジュールの制御周期以上となるようにしてください。
  入力信号が制御周期以下の幅になる場合、MC 機能モジュールが入力信号を検出できず、正常な動作が実行できなくなります。
- サーボドライバの各入力信号は、サーボドライバ側のパラメータを設定する必要があります。接続するサーボドライバのマニュアルと付録を参照し、正しく設定してください。

8-2 原点復帰の手順

原点復帰を行う場合の手順について説明します。

1 **軸の追加と設定**
Sysmac Studio で軸の追加と設定を行います。

2 **軸パラメータの設定**
原点復帰パラメータで原点復帰方法を設定します。

3 **ユーザプログラムの作成**
Sysmac Studio でユーザプログラムを作成します。
詳細については、「6-8 モーション制御プログラムの作成方法 (P.6-36)」を参照してください。

4 **軸パラメータとユーザプログラムのダウンロード**
設定した軸パラメータとユーザプログラムを CPU ユニットへダウンロードします。
CPU ユニットへのダウンロードは、Sysmac Studio の「同期」機能を使用します。
詳細については、「3-2-2 設定手順 (P.3-9)」を参照してください。

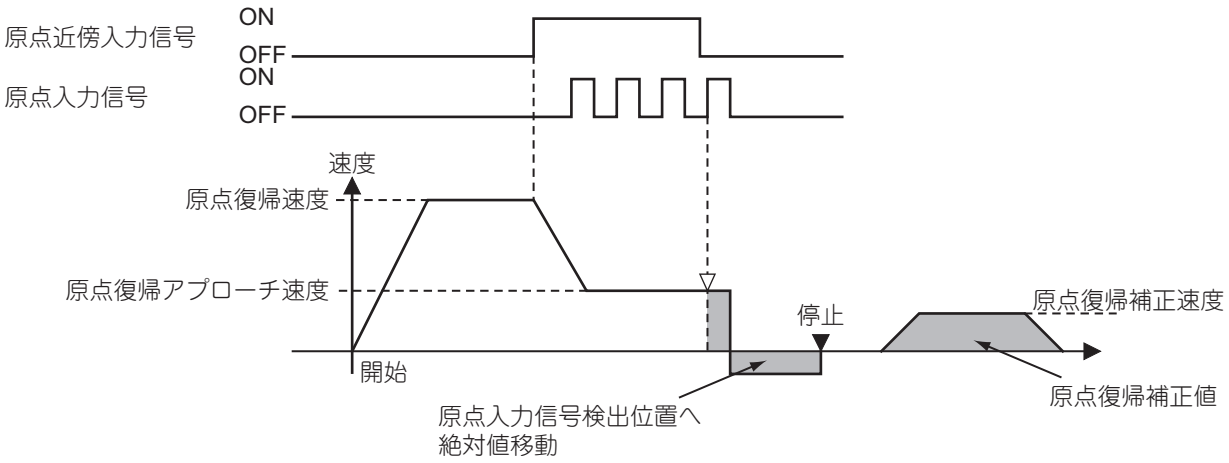
5 **ユーザプログラムの起動**
ユーザプログラムを起動し、MC_Power（運転可）命令の入力変数「Enable」を TRUE にし、サーボドライバをサーボ ON 状態にします。
MC_Home（原点復帰）命令、または MC_HomeWithParameter（パラメータ指定原点復帰）命令の入力変数「Execute」を TRUE にし、原点復帰を行います。

8-2-1 原点復帰パラメータの設定

原点復帰の方法は、原点復帰パラメータの設定値によって指定します。
原点復帰パラメータは Sysmac Studio で設定します。

設定	概要
原点復帰動作	原点復帰の動作方法を選択します。
原点入力信号	原点入力信号の入力先を選択します。
原点復帰開始方向	原点復帰起動時の開始方向を設定します。
原点検出方向	原点復帰の原点入力検出方向を設定します。
正方向限界入力時動作	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。
負方向限界入力時動作	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。
原点復帰速度	原点復帰中の速度を設定します。(単位: 指令単位／s)
原点復帰アプローチ速度	原点近傍入力が「ON」した後の速度を設定します。(単位: 指令単位／s)
原点復帰加速度	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に到達します。 (単位: 指令単位／s^2)
原点復帰減速度	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ゼロ速度に到達します。 (単位: 指令単位／s^2)
原点復帰ジャーク	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。(単位: 指令単位／s^3)
原点入力マスク距離	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: 指令単位)
原点位置オフセット	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。 (単位: 指令単位)

設定	概要
原点復帰押し当て時間	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: ms)
原点復帰補正値	原点復帰の原点確定後の原点復帰補正量を設定します。(単位: 指令単位)
原点復帰補正速度	原点復帰補正時の速度を設定します。(単位: 指令単位／s)



原点復帰動作

機械原点を決定する動作として、10 種類の動作モードから選択できます。

- 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

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。

○: パラメータを使用する、×: パラメータを使用しない

	原点復帰パラメータ														
原点復帰動作モード	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	○	○	○	○	○	○	○	○	○	○	×	○	×	○	○

	原点復帰パラメータ														
原点復帰動作モード	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	×	×	×	×	×	×	×	×	×	×	×	○	×	×	×

各原点復帰動作モードの詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』を参照してください。

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モードの［原点近傍入力なし・押し当て原点入力指定］を使用する場合は、原点入力信号に Z 相入力を選択しないでください。

原点入力信号

原点入力信号を使用する原点復帰動作モードで、原点決めするための信号をサーボドライバの Z 相信号、外部原点信号から選択します。

使用上の注意

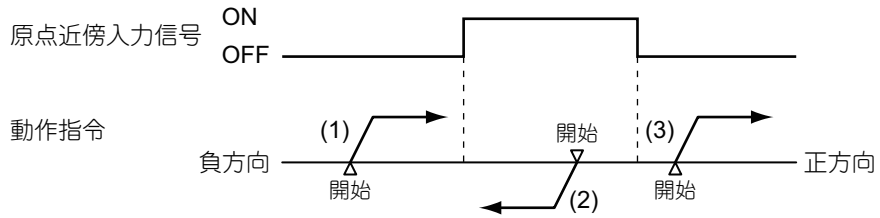
この設定で原点入力信号を選択できるのはオムロン製サーボドライバ 1S シリーズまたは G5 シリーズを接続した場合だけです。

原点復帰開始方向

原点復帰の起動時に動き出す方向を正方向と負方向から選択します。
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。

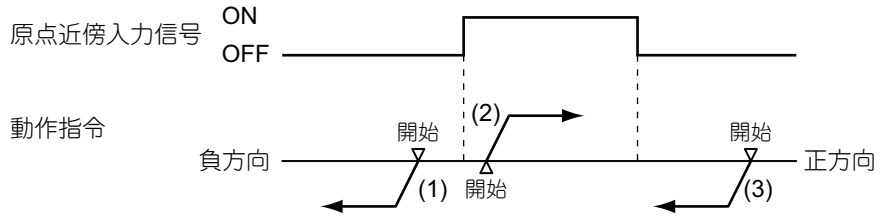
近傍退避による反転動作を含む原点復帰動作モードは以下の 4 つです。
- 0: 近傍退避・原点近傍入力 OFF 指定
- 1: 近傍退避・原点近傍入力 ON 指定
- 9: 近傍退避・原点入力マスク距離指定
- 12: 近傍退避・押し当て時間指定

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



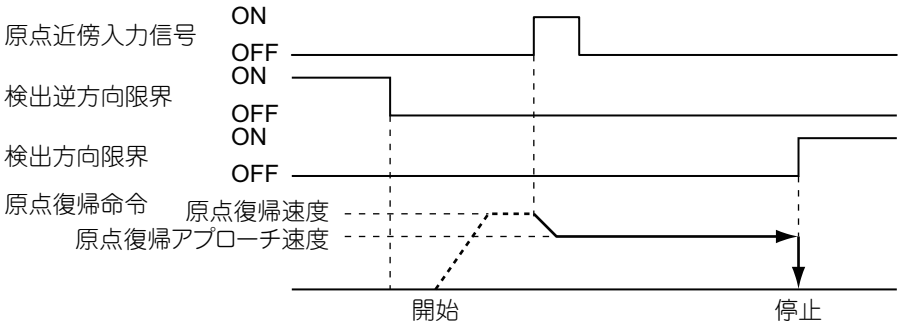
(1), (3)　: 原点近傍信号「OFF」状態から開始のため、原点復帰開始方向に動作開始
(2)　: 原点近傍信号「ON」状態から開始のため、原点検出方向と反対方向に動作開始

原点検出方向

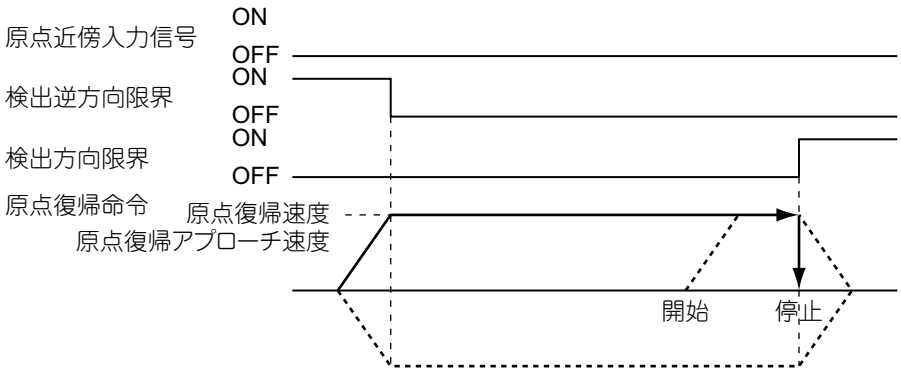
原点検出する方向を正方向と負方向から選択します。
原点検出方法と原点復帰の起動時に動き出す方向との関係については、「原点復帰開始方向 (P.8-7)」を参照してください。

正方向限界入力時動作／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。



原点復帰速度

原点復帰中の速度を、指令単位／s で設定します。

原点復帰アプローチ速度

原点近傍入力が「ON」した後の速度を、指令単位／s で設定します。

原点復帰加速度

原点復帰時の加速度を、指令単位／s^2 で設定します。
「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。

原点復帰減速度

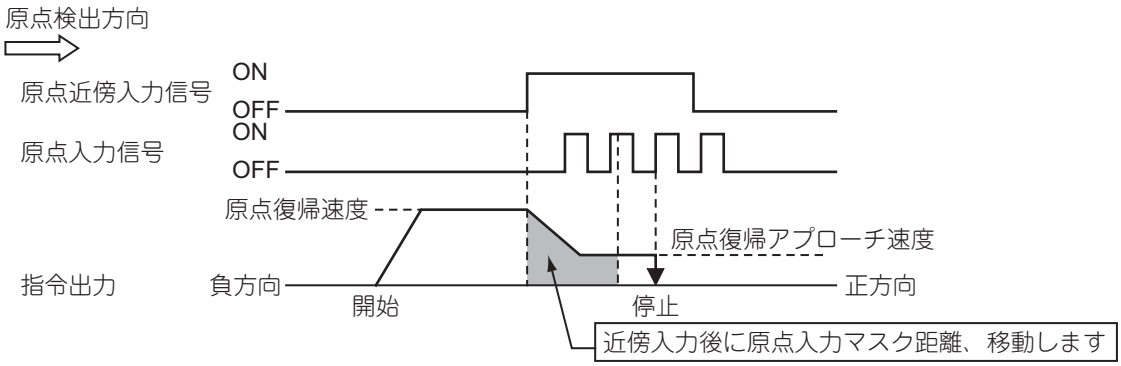
原点復帰時の減速度を、指令単位／s^2 で設定します。
「0」を設定すると減速せずに原点復帰アプローチ速度などの目標とする速度に到達します。

原点復帰ジャーク

原点復帰時のジャークを、指令単位／s^3 で設定します。
「0」を設定するとジャークなしでの加減速となります。

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］の場合の原点入力のマスク距離を指令単位で設定します。
原点近傍入力信号の「OFF」→「ON」による減速開始からの距離です。

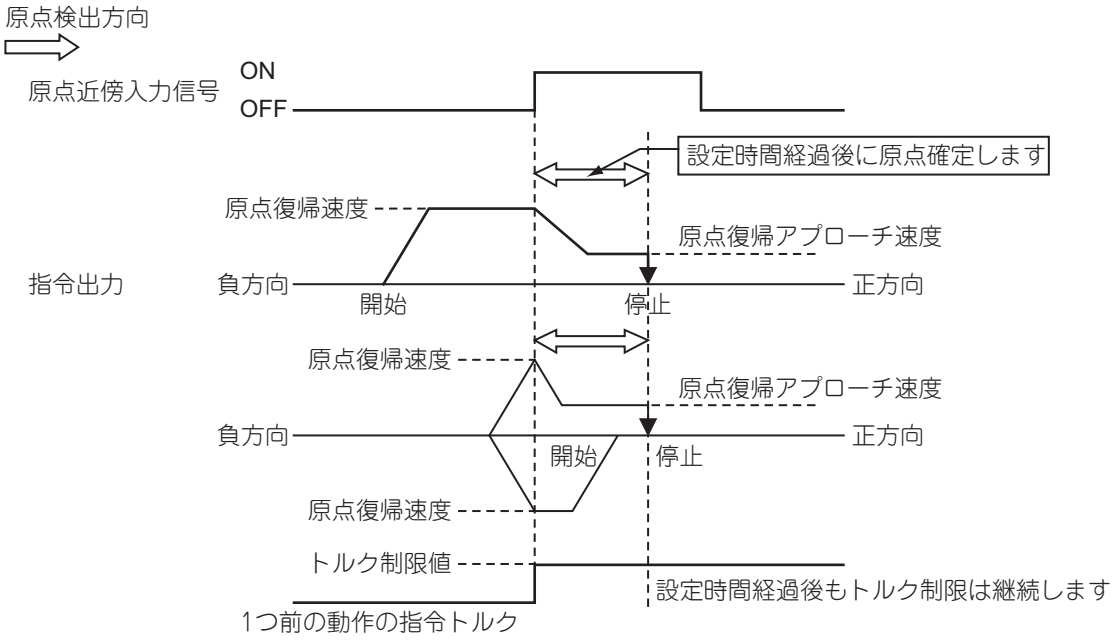


原点位置オフセット

機械原点が確定した後、原点復帰補正値が設定されている場合は原点復帰補正の動作完了後に、設定した値で現在値をプリセットします。
機械原点を「0」としてではなく、任意の値で使用するための機能です。
また、絶対値エンコーダシステムの場合は、同時に絶対値エンコーダ原点位置オフセットの値を自動で計算し、CPU ユニット内部に保存します。

原点復帰押し当て時間

原点復帰動作モードが［12: 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。
原点近傍入力信号の「OFF」→「ON」による減速開始からの時間です。



原点復帰補正値

機械原点が確定した後に、設定した値で相対位置決めを行い、機械原点を補正します。そのときの移動速度は、原点復帰補正速度となります。

機械原点が確定した後ではワークの調整が困難なため、微調整が必要な場合に最初に確定した機械原点からのずれ量を補正データとして設定します。
原点近傍センサの再設置が困難な場合や、モータの交換により機械原点がずれた場合に使用すると便利です。

原点復帰補正速度

原点復帰補正値を設定した場合の速度を、指令単位／s で設定します。

8-2-2 原点復帰のモニタ

ユーザプログラムからは、軸変数を読み取ることで原点復帰状態や入力信号状態をモニタできます。
なお、変数名は、_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_AX[0-255].Status.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。
_MC_AX[0-255].Dtails.Homed	BOOL	原点確定	原点確定状態時に TRUE となります。 FALSE: 原点未確定 TRUE: 原点確定
_MC_AX[0-255].Dtails.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。
_MC_AX[0-255].DrvStatus.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。
_MC_AX[0-255].DrvStatus.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。
_MC_AX[0-255].DrvStatus.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。
_MC_AX[0-255].DrvStatus.Home	BOOL	原点入力	原点入力が有効なときに TRUE となります。

8-3 原点復帰の動作

原点復帰は、位置決めシステムの構成や用途によって、機械原点の確定方法を選択する必要があります。
MC 機能モジュールでは、10 種類の原点復帰動作モードが準備されています。
また、原点確定した後に、原点復帰補正値により微調整することも可能です。

参考

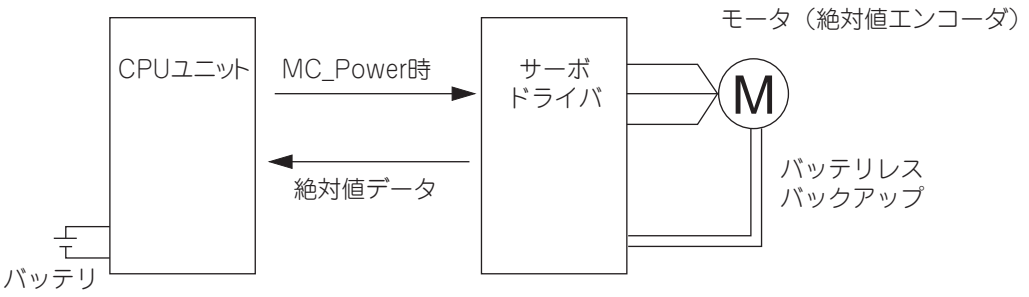
- どのモードが適しているかは、位置決めシステムの構成や用途によって異なります。
  リニアモード（有限長）で、原点近傍センサと正方向限界入力、負方向限界入力を備えた機械系では、［近傍退避・原点近傍入力 OFF 指定の動作］を選択するのが一般的です。
- 原点復帰ではオーバーライドは無効となります。
- 原点確定時および原点復帰補正移動完了時にだけ、インポジションチェックの設定に従いインポジションチェックを行います。
- 原点復帰中に他のモーション制御命令を用いた多重起動を行う場合、バッファードおよびブレンディングは行えません。

原点復帰の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』を参照してください。

8-4　絶対値エンコーダ原点設定

ここではオムロン製サーボモータ 1S シリーズおよび G5 シリーズの絶対値エンコーダ付きサーボモータ、またはオムロン製モータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプで絶対値タイプの外部スケールを使用する場合について説明します。

絶対値エンコーダは CPU ユニットの電源が切れても、絶対値データを保持しています。
このため、MC_Power（運転可）命令実行時に、絶対値エンコーダより現在位置を読み込むことにより、位置を確定させています。
一度、原点確定しておけば、インクリメンタルエンコーダのように原点復帰操作は不要です。



使用上の注意

- オムロン製サーボドライバ 1S シリーズを使用する場合、形 NX502、形 NX102 または形 NX1P2 以外の CPU ユニットを使用するときは、CPU ユニットのバッテリを接続してください。
- オムロン製サーボドライバ G5 シリーズで絶対値エンコーダを使用する場合、形 NX502、形 NX102 または形 NX1P2 の CPU ユニットを使用するときは、サーボドライバの絶対値エンコーダバックアップ用バッテリを接続してください。
- オムロン製サーボドライバ G5 シリーズで絶対値エンコーダを使用する場合、形 NX502、形 NX102 または形 NX1P2 以外の CPU ユニットを使用するときは、CPU ユニットのバッテリとサーボドライバの絶対値エンコーダバックアップ用バッテリの両方を接続してください。
- 絶対値エンコーダの初回使用時や、モータの交換時、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絶対値エンコーダのバッテリ切れなど、絶対値データが失われているときには、必ず MC_Home（原点復帰）命令、または MC_HomeWithParameter（パラメータ指定原点復帰）命令を実行し、原点確定をしてください。
  それ以降は、MC_Power（運転可）命令の入力変数「Enable」（有効）を TRUE にすると、原点確定します。
  原点復帰については、「8-1 原点決めの概要 (P.8-2)」を参照してください。
  また、ユニットバージョン Ver.1.10 以降の CPU ユニットでは、前述に加えて、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、原点確定します。
- 形 NX502、形 NX102 または形 NX1P2 の CPU ユニットの場合、電源を切るときに CPU ユニット内部のメモリに異常が発生して絶対値データの保存に失敗したとき、次に電源を投入すると、「絶対値エンコーダ原点位置オフセット読み出し異常（イベントコード: 14600000Hex)」が発生します。
  形 NX502、形 NX102 または形 NX1P2 以外の CPU ユニットの場合、CPU ユニット側のバッテリが異常のときに CPU ユニットの電源を投入すると、「絶対値エンコーダ原点位置オフセット読み出し異常（イベントコード: 14600000Hex)」が発生します。
  この場合でも、ResetMCError（MC 全エラーリセット）命令で異常を解除するとサーボ ON することができます。
  ユニットバージョン Ver.1.10 以降の CPU ユニットでは、異常解除後に MC_Home（原点復帰）命令、または MC_HomeWithParameter（パラメータ指定原点復帰）命令を実行して原点確定するまで、原点未確定状態を維持します。
  正しい位置で動作を行うためには、MC_Home（原点復帰）命令、または MC_HomeWithParameter（パラメータ指定原点復帰）命令を実行し、正しい位置で原点確定してください。
  ユニットバージョン Ver.1.09 以前の CPU ユニットでは、異常解除後のサーボ ON 時に、[絶対値エンコーダ原点位置オフセット] の値を「0」として原点確定します。
- CPU ユニットの電源を切ると原点未確定状態となります。

参考

オムロン製サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプを使用する場合も絶対値エンコーダ原点設定ができます。リニアモータタイプを使用するときは以下の点に注意して本節の説明を読んでください。

- リニアモータタイプは「エンコーダ」は使用しませんが、同様の機能の「外部スケール」を使用します。
- 本節の説明の「絶対値エンコーダ」は「絶対値タイプの外部スケール」と読み替え可能です。
- 絶対値タイプの外部スケールは絶対値エンコーダにはある回転量データを持ちません。そのため、絶対値エンコーダでは必要であったセットアップが不要です。また、回転量データを保持するバッテリは使用しません。
- リニアモータタイプの仕様の詳細については、『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。

8-4-1 機能概要

絶対値エンコーダシステムの原点設定は、MC_Power（運転可）命令実行時に絶対値エンコーダのオフセット補正を行い、原点確定します。
ユニットバージョン Ver.1.10 以降の CPU ユニットでは、前述に加えて、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、原点確定します。
また、インクリメンタルエンコーダと同様に原点復帰により原点確定させることもできます。
その場合、原点確定後の指令位置と絶対値エンコーダより読み出した絶対値データとの差を［絶対値エンコーダ原点位置オフセット］として CPU ユニット内部に取り込みます。
原点復帰については、「8-1 原点決めの概要 (P.8-2)」を参照してください。

MC_Home（原点復帰）命令、または MC_HomeWithParameter（パラメータ指定原点復帰）命令による現在位置プリセットを実行した場合も、原点確定後の指令位置と絶対値データとの差をオフセット量として［絶対値エンコーダ原点位置オフセット］に取り込みます。
MC 機能モジュールでは、［絶対値エンコーダ原点位置オフセット］を、CPU ユニットの電源 OFF 時に CPU ユニット内部に保存しています。

使用上の注意

- ［絶対値エンコーダ原点位置オフセット］は、軸番号と関連付けて軸ごとに CPU ユニット内部に保存されています。軸番号を変更すると、保存されたオフセット量が失われます。軸番号を変更した場合は、再度原点復帰を行ってください。
- CPU ユニットまたは CPU ユニット側のバッテリを交換する場合は、交換前に原点確定状態で［絶対値エンコーダ原点位置オフセット］のバックアップを行ってください。
- 交換後にリストアを行うことで、交換前に確定した原点を使用することができます。
- バックアップ／リストアは Sysmac Studio を使用します。
  操作方法の詳細は、『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。

対象となるサーボモータ

絶対値エンコーダ原点設定を使用できるサーボモータを以下に示します。

メーカ名	シリーズ名	サーボドライバ形式	サーボモータ形式
オムロン	1S シリーズ	形 R88D-1SN□□□-ECT	形 R88M-1□□□□□□□S 形 R88M-1□□□□□□□T 形 R88M-1□□□□□□□C
		形 R88D-1SAN□□□-ECT	
	G5 シリーズ	形 R88D-KN□□□-ECT	形 R88M-K□□□□□□S 形 R88M-K□□□□□□T 形 R88M-K□□□□□□C
		形 R88D-KN□□□-ECT-L	形 R88L-EC□□□□□□□□□

使用上の注意

NX シリーズパルス出力ユニットでは、絶対値エンコーダを使用できません。

サーボドライバとの接続

『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)』にしたがって、正しく接続してください。

参考

- オムロン製サーボドライバ 1S シリーズと接続する場合の設定例については、「A-1 サーボドライバ 1S シリーズとの接続 (P.A-2)」を参照してください。
- オムロン製サーボドライバ G5 シリーズと接続する場合の設定例については、「A-2 サーボドライバ G5 シリーズとの接続 (P.A-11)」を参照してください。

8-4-2 設定手順

絶対値エンコーダシステムの原点設定の手順について説明します。

1 **絶対値エンコーダのセットアップ**
セットアップ方法については、『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。

2 **軸パラメータの設定**
MC 機能モジュールの軸パラメータにある［位置カウント設定］の［エンコーダ種別］で［1: 絶対値エンコーダ (ABS)］を選択します。
詳細については、「5-2-7 位置カウント設定 (P.5-23)」を参照してください。

3 **原点復帰の実行**
原点復帰を実行します。
原点確定方法は、MC 機能モジュールの軸パラメータの［原点復帰設定］の［原点復帰動作］により選択できます。
原点確定後、指令位置と絶対値エンコーダより読み出した絶対値データとの差を［絶対値エンコーダ原点位置オフセット］として CPU ユニット内部に保存します。

絶対値エンコーダのセットアップ

絶対値エンコーダの初回使用時や、回転量データを「0」に初期化したい場合、また絶対値エンコーダにバッテリを接続せずに長期間放置したときなど、絶対値エンコーダのセットアップが必要となります。
セットアップ方法の詳細については、『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。

使用上の注意

オムロン製サーボドライバ 1S シリーズおよび G5 シリーズは、絶対値エンコーダのセットアップ後、電源の再投入が必要です。
絶対値エンコーダのセットアップ処理が完了すると、サーボドライバに絶対値クリアエラー (A27.1)が発生します。サーボドライバの制御電源を一度 OFF した後、再度電源を投入することで、この異常は解除され、絶対値エンコーダのセットアップ手順が終了します。

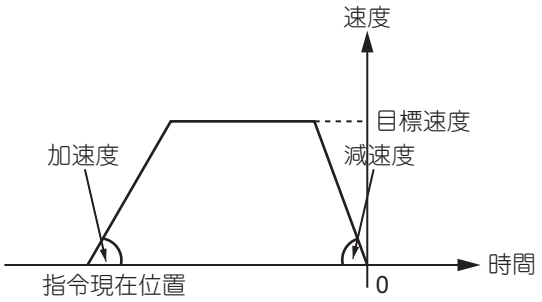
ロータリモードでの絶対値エンコーダの使用

軸パラメータの［カウントモード選択］に［ロータリモード］を設定している場合、現在位置は［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］の範囲でリング状のカウンタとなります。
絶対値エンコーダを使用して［ロータリモード］に設定している場合、MC 機能モジュールはモーション制御周期ごとに、自動的に［絶対値エンコーダ原点位置オフセット］を計算して更新します。

更新された［絶対値エンコーダ原点位置オフセット］は、CPU ユニットの電源 OFF 時に CPU ユニット内部に保存されます。
これにより、CPU ユニットやサーボドライバの電源が OFF されても、次の電源投入時に絶対値エンコーダから回転軸での現在位置を復元することが可能となります。

8-5 高速原点復帰

原点確定状態で原点位置へ高速に位置決めする機能です。
MC_MoveZeroPosition（高速原点復帰）命令を使用し、目標速度、加速度、減速度、ジャークを指定することができます。
原点未確定状態で実行すると命令で異常が発生します。



参考

MC_MoveZeroPosition（高速原点復帰）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 モーション編(SBCE-434)』を参照してください。

9

モーション制御機能

ここではオムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプと接続する場合のモーション制御機能について説明します。

9-1 単軸位置制御 9-3
9-1-1 動作概要 9-3
9-1-2 絶対値位置決め 9-4
9-1-3 相対値位置決め 9-4
9-1-4 割り込み定寸位置決め 9-4
9-1-5 サイクリック同期位置決め 9-5
9-1-6 停止 9-6
9-1-7 オーバーライド 9-10

9-2 単軸同期制御 9-12
9-2-1 同期制御の概要 9-12
9-2-2 ギア動作 9-12
9-2-3 位置指定ギア動作 9-13
9-2-4 カム動作 9-14
9-2-5 カムテーブル 9-15
9-2-6 台形パターンカム動作 9-22
9-2-7 加減算位置決め 9-23
9-2-8 主軸位相補正 9-24
9-2-9 従軸位置補正 9-25
9-2-10 マルチモーションでの同期制御の実現方法 9-26

9-3 単軸速度制御 9-28
9-3-1 速度制御 9-28
9-3-2 サイクリック同期速度制御 9-28

9-4 単軸トルク制御 9-30

9-5 単軸制御の共通機能 9-31
9-5-1 位置 9-31
9-5-2 速度 9-32
9-5-3 加速と減速 9-34
9-5-4 ジャーク 9-35
9-5-5 動作方向指定 9-36
9-5-6 モーション制御命令の再起動 9-39
9-5-7 モーション制御命令の多重起動（バッファモード） 9-44

9-6 多軸協調制御 9-50
9-6-1 動作概要 9-50
9-6-2 直線補間 9-52
9-6-3 円弧補間 9-53
9-6-4 軸グループサイクリック同期位置制御 9-54

9-6-5　多軸協調制御での停止 9-54
9-6-6　多軸協調制御でのオーバーライド 9-56

9-7　多軸協調制御の共通機能 9-57
9-7-1　多軸協調制御での速度 9-57
9-7-2　多軸協調制御での加速と減速 9-57
9-7-3　多軸協調制御でのジャーク 9-59
9-7-4　多軸協調制御のモーション制御命令の再起動 9-59
9-7-5　多軸協調制御のモーション制御命令の多重起動（バッファモード） 9-60

9-8　そのほかの機能 9-68
9-8-1　現在位置変更機能 9-68
9-8-2　トルク制限機能 9-68
9-8-3　ラッチ機能 9-69
9-8-4　ゾーン機能 9-70
9-8-5　ソフトウェアリミット機能 9-70
9-8-6　位置偏差監視機能 9-72
9-8-7　偏差カウンタリセット 9-72
9-8-8　軸間偏差監視機能 9-73
9-8-9　インポジションチェック 9-74
9-8-10　軸使用変更機能 9-76
9-8-11　デジタルカムスイッチ有効機能 9-76
9-8-12　任意座標系の 3D モーションモニタ表示 9-77

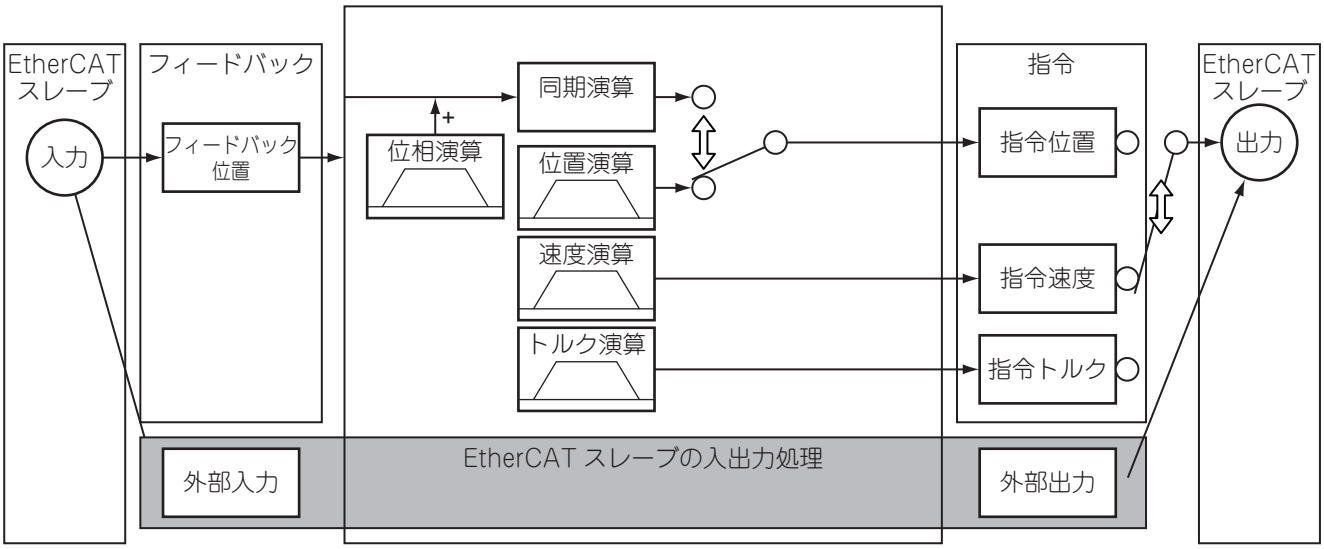
9-1 単軸位置制御

MC 機能モジュールは、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、および G5 シリーズ EtherCAT 通信内蔵タイプと接続することにより、位置制御、速度制御、トルク制御を行うことができます。
ここでは、単軸での位置決め動作について説明します。

NX シリーズ パルス出力ユニットを使用する場合は、モーション制御機能の一部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。

9-1-1 動作概要

MC 機能モジュールの単軸制御機能には、動作プロファイルを指令する制御と同期制御があります。
動作プロファイルを指令する制御では、位置制御、速度制御、トルク制御の 3 つの制御モードで指令することができます。
同期制御では、制御対象である従軸が、主軸に対するカムプロファイル曲線やギア比などで表現される同期関係を持ちながら動作します。
また、ジョグ送りなどの手動運転や原点復帰にも対応しています。



（注）同期演算には、指令位置またはフィードバック位置を入力として使用できます。

軸エラーリセット

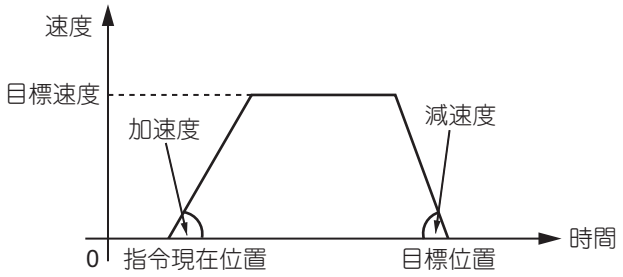
軸で異常が発生した場合、異常の原因を取り除いたのちに MC_Reset（軸エラーリセット）命令で、異常を解除します。

軸エラーリセットの詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_Reset（軸エラーリセット）命令を参照してください。

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

9-1-2　絶対値位置決め

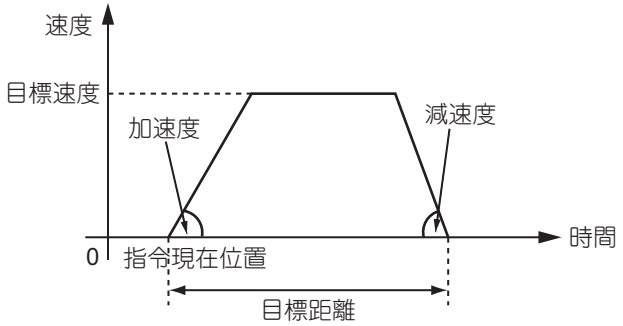
原点を基準とする絶対座標の目標位置を指定する位置決めです。
［カウントモード］を［ロータリモード］にして動作方向を指定することで、ロータリテーブルでの近回りなどの位置決めができます。



機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_MoveAbsolute（絶対値位置決め）命令および MC_Move（位置決め）命令を参照してください。

9-1-3　相対値位置決め

現在位置からの距離を指定する位置決めです。
［カウントモード］を［ロータリモード］にすることで、リングカウンタ範囲を超える移動距離の指定ができます。



機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_MoveRelative（相対値位置決め）命令および MC_Move（位置決め）命令を参照してください。

9-1-4　割り込み定寸位置決め

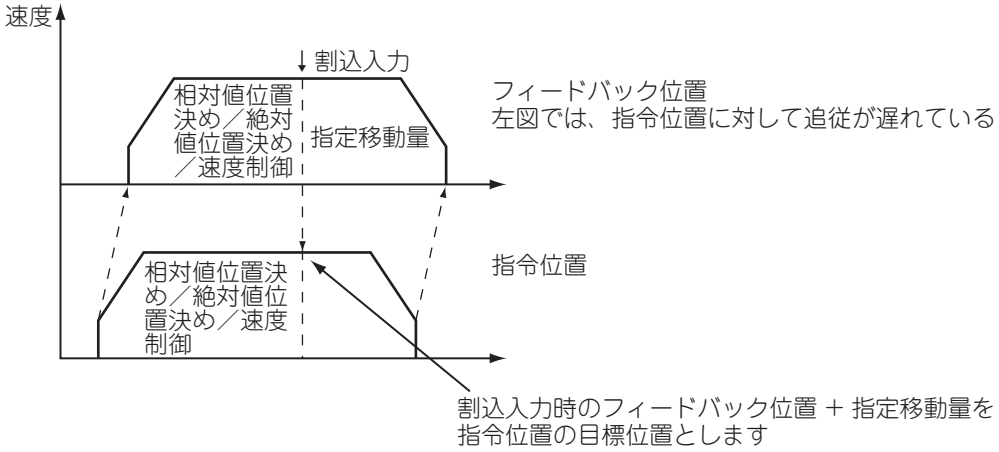
指定した速度で軸を送り、トリガ信号が発生したフィードバック位置から、指定した距離を移動する位置決めです。
絶対値位置決め、または相対値位置決めの場合、指定した移動量以内にトリガ信号が発生しなかった場合に異常を出力することも選択できます。
トリガ信号発生時のフィードバック位置は、サーボドライバ側のラッチ機能を使用しているので、位置偏差の影響を受けない定寸位置決めを実現できます。また、指定した位置範囲外のトリガ信号発生を無効にするウィンドウ機能により、包装機などでフィルムのマーク印刷以外の影響をなくすこともできます。

フィードバック位置を基準

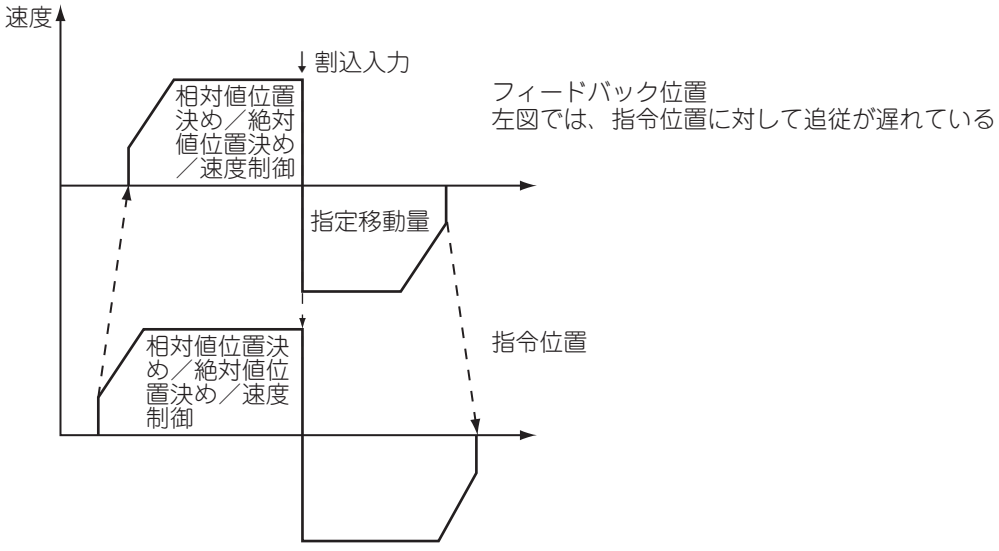
機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_MoveFeed（割り込み定寸位置決め）命令を参照してください。

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

● 進行方向への指定量移動



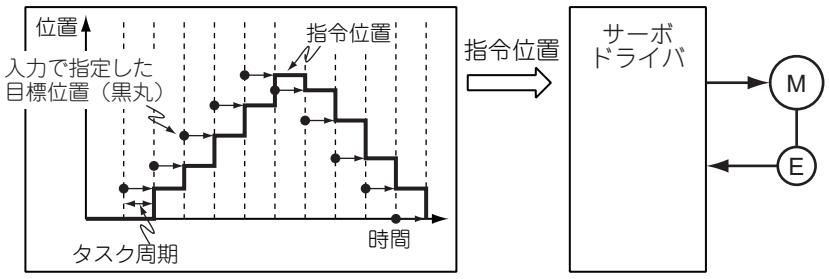
● 進行方向とは逆方向への指定量移動



軸パラメータの反転時動作で［反転時減速停止］を選択すると、加減速カーブで反転します。

9-1-5　サイクリック同期位置決め

サイクリック同期位置決めは、軸に指定した目標位置を、プライマリ周期または定周期タスクのタスク周期ごとに出力する機能です。目標位置は、絶対位置で指定します。
ユーザが作成した任意の軌跡で動作させたい場合などに使用します。



機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_SyncMoveAbsolute（サイクリック同期絶対位置決め）命令を参照してください。

バージョン関連情報

サイクリック同期位置決め機能は、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することができます。

9-1-6　停止

軸の動作を停止する機能には、サーボドライバに接続された即停止入力信号や限界入力信号と、ユーザプログラムのモーション制御命令による停止機能、および異常の発生などによる停止があります。

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

サーボドライバの入力信号による停止

サーボドライバの即停止入力信号または限界入力信号により軸の動作を停止します。
Sysmac Studio を使用して停止方法を選択することができます。

● 即停止入力

サーボドライバ側の入力信号の状態により、MC 機能モジュール側で停止処理を実行します。
MC 機能モジュールの停止方法は、次から選択することができます。

- 指令値を即停止
- 指令値を即停止し、同時に偏差カウンタリセットを実行
- 指令値を即停止し、同時にサーボ OFF を実行

使用上の注意

オムロン製サーボドライバ 1S シリーズおよび G5 シリーズの即停止入力は、サーボドライバ側でも異常となり停止処理を実行しています。

● 限界入力（正方向限界入力または負方向限界入力）

サーボドライバ側の入力信号の状態により、MC 機能モジュール側で停止処理を実行します。
MC 機能モジュールの停止方法は、次から選択することができます。

- 指令値を即停止
- 指令値を減速停止
- 指令値を即停止し、同時に偏差カウンタリセットを実行
- 指令値を即停止し、同時にサーボ OFF を実行

使用上の注意

- 限界入力信号が ON している場合、軸指令の命令は、ON している限界入力信号と同じ方向には起動しないでください。
- 軸グループのいずれかの軸の限界入力が ON している場合、軸グループ指令の命令は起動しないでください。
- 入力変数に「Deceleration」（減速度）を持つ同期動作命令を実行中に、減速停止となる信号が入力された場合は、「Deceleration」（減速度）で減速停止します。
- 入力変数に「Deceleration」（減速度）を持たない同期動作命令を実行中に、減速停止となる信号が入力された場合は、軸パラメータの最大減速度で減速停止します。

参考

- サーボドライバの入力信号を使用するためには、サーボドライバの設定を行う必要があります。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 通信内蔵タイプでは、サーボドライバ側の初期値で即停止入力と限界入力が割り付けられています。
- オムロン製サーボドライバ 1S シリーズと接続する場合の設定例については、「A-1 サーボドライバ 1S シリーズとの接続 (P.A-2)」を参照してください。
- オムロン製サーボドライバ G5 シリーズと接続する場合の設定例については、「A-2 サーボドライバ G5 シリーズとの接続 (P.A-11)」を参照してください。

モーション制御命令による停止

単軸動作の停止には、MC_Stop（強制停止）命令、または MC_ImmediateStop（即停止）命令を使用します。

機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_Stop（強制停止）命令、MC_ImmediateStop（即停止）命令を参照してください。

参考

MC_Power（運転可）命令の入力変数「Enable」（有効）を FALSE にすると、MC 機能モジュールは、指令値を即停止しサーボ OFF を実行します。
サーボ OFF 時のサーボドライバの動作は、サーボドライバ側の設定に従います。

● MC_Stop（強制停止）命令

単軸制御および同期制御機能に対して減速度とジャークを指定して減速停止させることができます。
減速度に「0」を指定すると即時に停止する指令をサーボドライバへ送信します。
本命令による減速停止中と入力変数の「Execute」（起動）が TRUE の間は、ほかの動作指令を受け付けません。

● MC_ImmediateStop（即停止）命令

単軸制御および同期制御機能に対して即停止することができます。
軸グループ有効となっている軸に対しても実行できます。

異常の発生などによる停止

● 単軸動作中の異常発生による停止

単軸動作中に異常が発生した場合は、異常の内容により即停止あるいは減速停止となります。
異常ごとの停止方法については、『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。

● ソフトウェアリミットによる停止

ソフトウェアリミットによる停止は、軸パラメータ設定の［ソフトウェアリミット機能］で設定します。
ソフトウェアリミットによる停止方法は、次から選択することができます。

- 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。即停止させる。
- フィードバック位置に対して有効。減速停止させる。
- フィードバック位置に対して有効。即停止させる。

ソフトウェアリミット機能については、「9-8-5 ソフトウェアリミット機能 (P.9-70)」を参照してください。

● モーション制御周期オーバーの発生による停止

モーション制御の処理が 2 周期以内で完了せずモーション制御周期オーバーが発生したときは、すべての軸が即停止します。

使用上の注意

形 NX701 CPU ユニットを使用してマルチモーションで動作させている場合に、いずれか一方のタスクでモーション制御周期オーバーが発生したときは、両タスクのすべての軸が即停止します。
マルチモーションは、「A-6-2 モーション制御機能 (P.A-29)」を参照してください。

● サーボ OFF となる異常の発生

サーボ OFF となる異常が発生した場合は即停止します。
サーボ OFF 時のサーボドライバの動作は、サーボドライバ側の設定に従います。

● MC 試運転機能の開始による停止

Sysmac Studio の MC 試運転機能を開始すると、各軸は「最大減速度」で減速停止します。

● MC 試運転機能の終了による停止

Sysmac Studio の MC 試運転機能を終了すると、各軸は「最大減速度」で減速停止します。

- Sysmac Studio の MC 試運転画面にある［MC 試運転終了］を選択する。
- Sysmac Studio の MC 試運転画面を閉じる。
- Sysmac Studio を終了する。

● CPU ユニットのモード切り替えによる停止

NJ/NX シリーズ CPU ユニットのモードを切り替えると、各軸は「最大減速度」で減速停止します。

使用上の注意

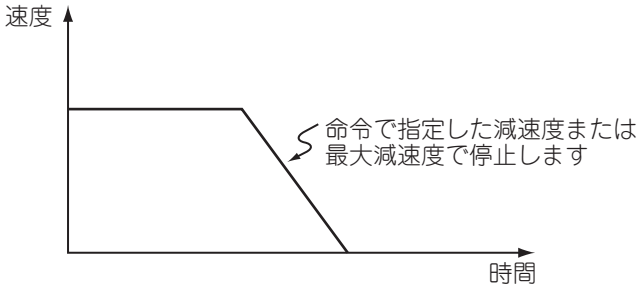
- 入力変数に「Deceleration」（減速度）を持つ同期動作命令を実行中に、減速停止となる異常が発生した場合は、「Deceleration」（減速度）で減速停止します。
- 入力変数に「Deceleration」（減速度）を持たない同期動作命令を実行中に、減速停止となる異常が発生した場合は、軸パラメータの最大減速度で減速停止します。

参考

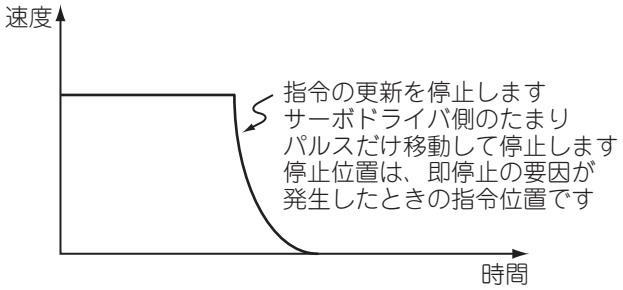
- 運転モードからプログラムモードへ切り替えると、動作中のモーション制御命令は中断されます。このとき、命令の出力変数「CommandAborted」（実行中断）は FALSE のままです。プログラムモードへ切り替わり後もサーボ ON/OFF 状態を維持しています。
- 運転モードからプログラムモードへ切り替えて減速停止しているときに、運転モードに戻すと、モーション制御命令の出力変数がクリアされるため、モーション制御命令の出力変数「CommandAborted」（実行中断）は FALSE のままです。
- MC_SaveCamTable（カムテーブル保存）命令で保存中の場合、保存処理は継続されます。
- MC_GenerateCamTable（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、生成処理は継続されます。

停止方法

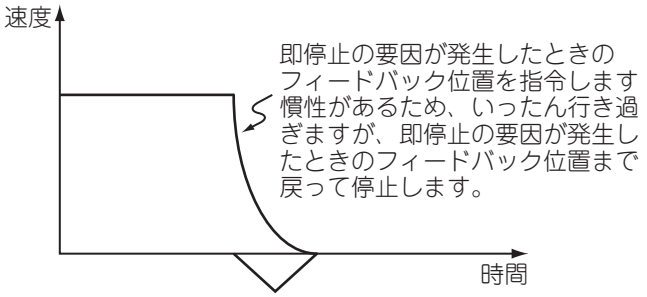
● 指令値を減速停止



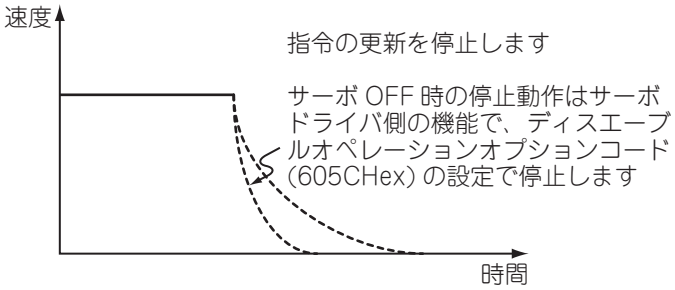
● 指令値を即停止



● **指令値を即停止し、同時に偏差カウンタリセットを実行**



● **指令値を即停止し、同時にサーボ OFF を実行**



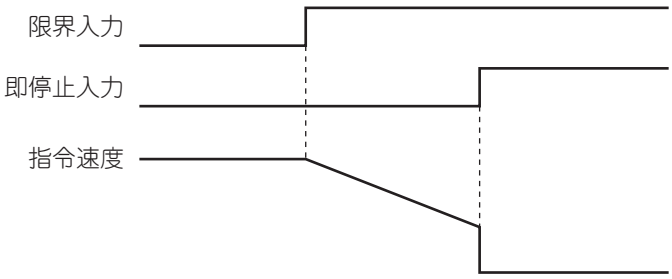
停止の優先順位

停止の優先順位はつぎの表のとおりです。停止中に優先順位の高い停止が発生した場合は、優先順位の高い停止方法に切り替わります。

停止方法	優先順位 (数字が大きい方が高優先)
指令値を即停止し、同時にサーボ OFF を実行	4
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	2
指令値を減速停止	1

● **例**

下図は、限界入力信号の ON で減速停止中に即停止入力が ON し、即停止する場合の例です。



9-1-7　オーバーライド

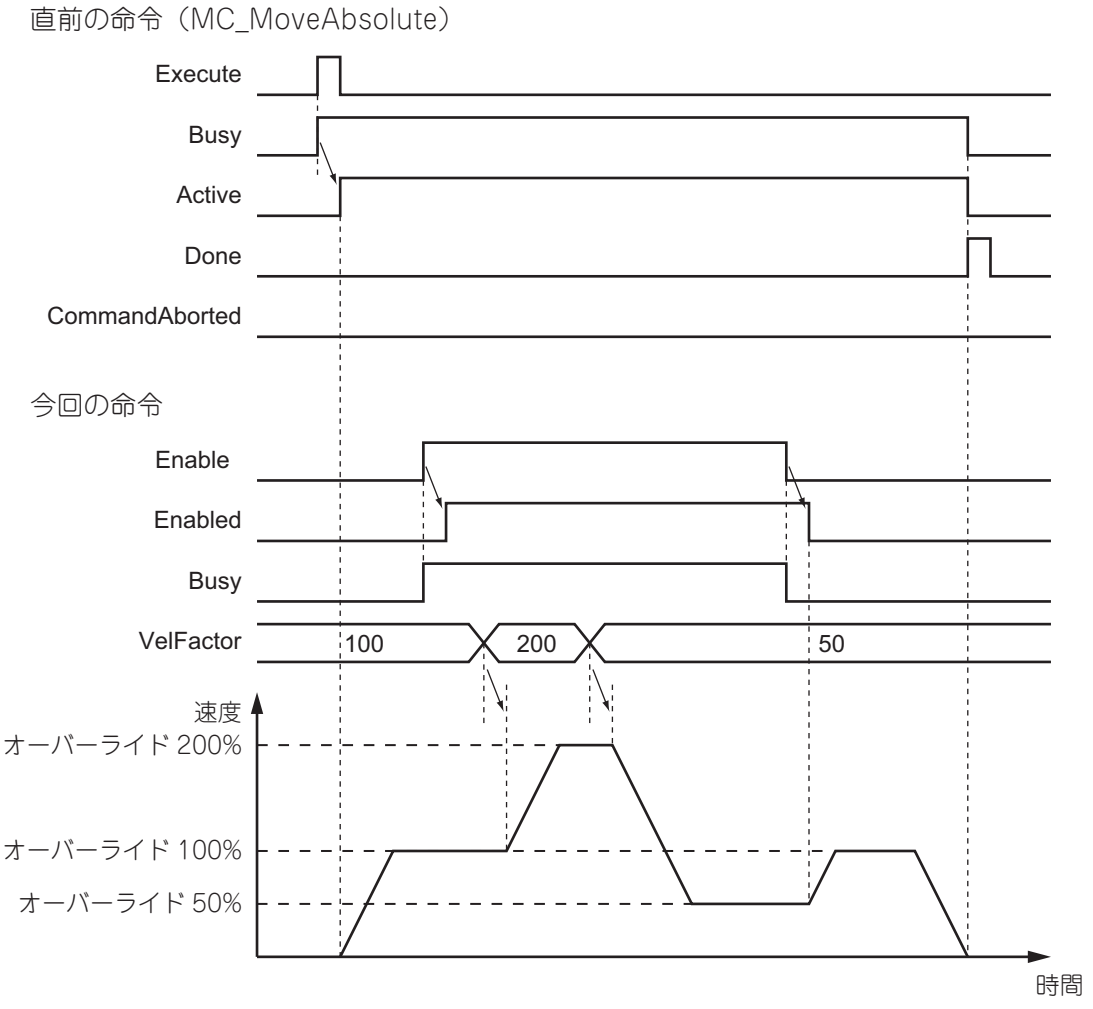
MC_SetOverride（オーバーライド値設定）命令により、起動中の軸の動作に対するオーバーライド値を設定する機能です。

オーバーライド値は目標速度に対する比率で、0～500%を設定できます。
目標速度に対して 0%を設定すると、速度「0」で停止したまま動作状態を継続します。

オーバーライドが有効の間、設定したオーバーライド値が取り込まれます。
オーバーライドを無効にすると、オーバーライド値は 100%に戻ります。
オーバーライド値で変更後の値が最高速度をオーバーする場合は、各軸の「最高速度」になります。

● 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、オーバーライド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。



機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編（SBCE-434）』の MC_SetOverride（オーバーライド値設定）命令を参照してください。

9-2 単軸同期制御

ここでは、単軸の同期制御の動作について説明します。

9-2-1 同期制御の概要

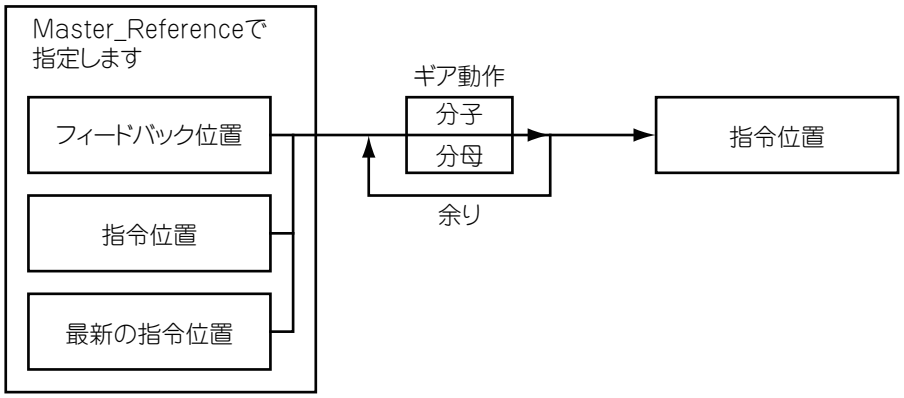
主軸の位置に同期して従軸の位置を制御する機能です。
主軸には各軸の指令位置またはフィードバック位置を指定できます。
従軸への指令速度が軸パラメータの［最高速度］を超過した場合は、最高速度で指令します。この場合、不足する移動量は次の周期以降に分配して出力します。

使用上の注意

- 従軸には、エンコーダ軸、仮想エンコーダ軸、単軸位置制御軸を指定できません。
- 形 NX701 CPU ユニットを使用してマルチモーションで動作させている場合に、同期制御するときは、構成する主軸と従軸を同じタスクにしてください。
  MC_GearIn（ギア動作開始）命令や MC_CamIn（カム動作開始）命令などの同期制御命令により、タスクの異なる軸を主軸、従軸に指定した場合、「主軸指定不正（イベントコード: 54620000 Hex）」が発生します。
  タスクの異なる軸を主軸、従軸に指定したい場合は、「9-2-10 マルチモーションでの同期制御の実現方法 (P.9-26)」を参照してください。

9-2-2 ギア動作

主軸と従軸間のギア比を設定しギア動作を行う機能です。
MC_GearIn（ギア動作開始）命令でギア動作を開始し、MC_GearOut（ギア動作解除）命令、または MC_Stop（強制停止）命令で同期を解除します。



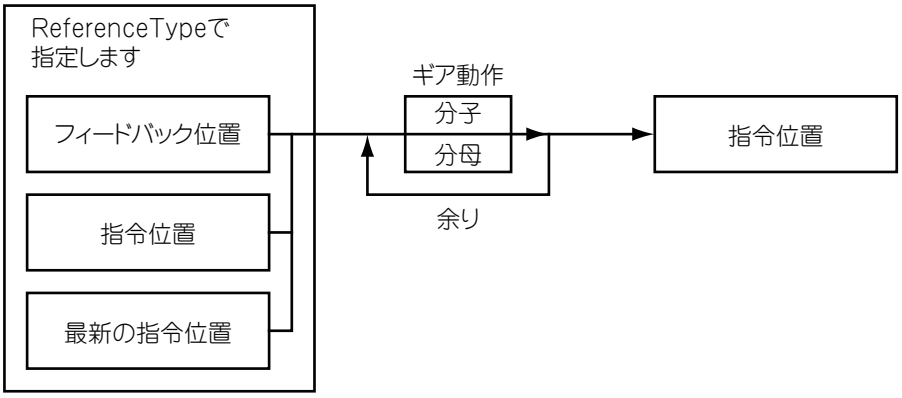
動作対象の従軸に対し、ギア比分子、ギア比分母、位置種別選択、加速度、減速度を指定することができます。また、主軸には指令位置、フィードバック位置、最新の指令位置のいずれかを指定することができます。

動作開始後、従軸は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。
ギア比が正数のときは Slave（従軸）は Master（主軸）の同方向に移動し、負数のときは Slave（従軸）は Master（主軸）の逆方向に移動します。

ギア動作の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_GearIn（ギア動作開始）命令、および MC_GearOut（ギア動作解除）命令、MC_Stop（強制停止）命令を参照してください。

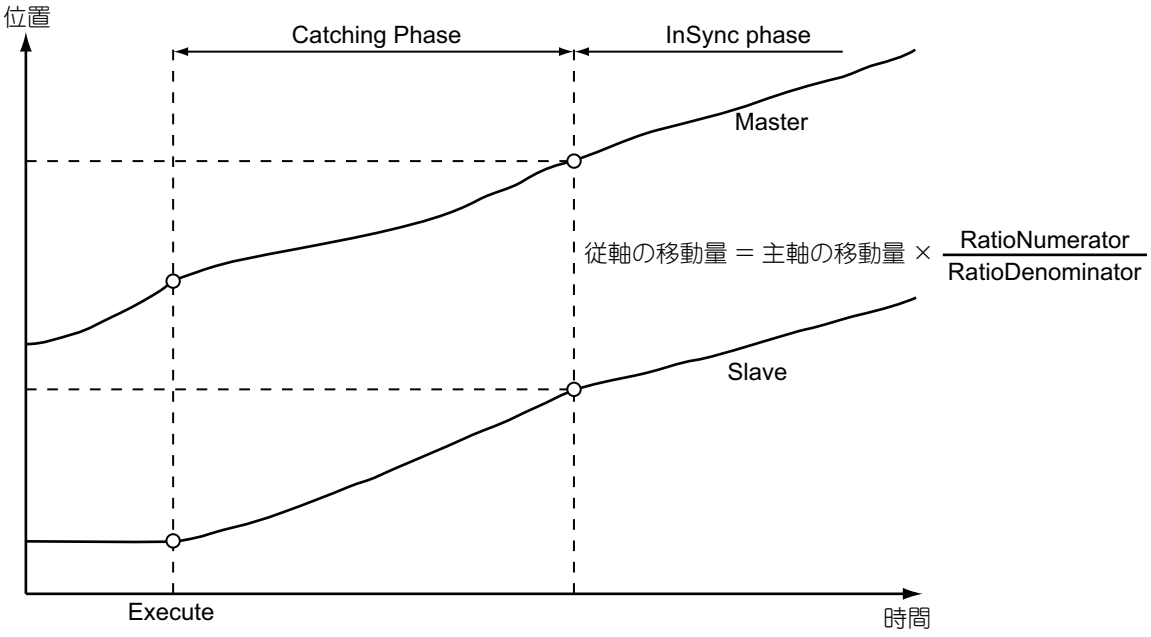
9-2-3 位置指定ギア動作

主軸と従軸間のギア比を設定し、ギア動作を行う機能です。
位置指定ギア動作では、同期を開始する主軸の位置と従軸の位置を指定することができます。
MC_GearInPos（位置指定ギア動作）命令で位置指定ギア動作を開始し、MC_GearOut（ギア動作解除）命令または、MC_Stop（強制停止）命令で同期を終了します。



動作対象の従軸に対し、ギア比分子、ギア比分母、位置種別選択、加速度、減速度を指定することができます。また、主軸には指令位置、フィードバック位置、最新の指令位置のいずれかを指定することができます。

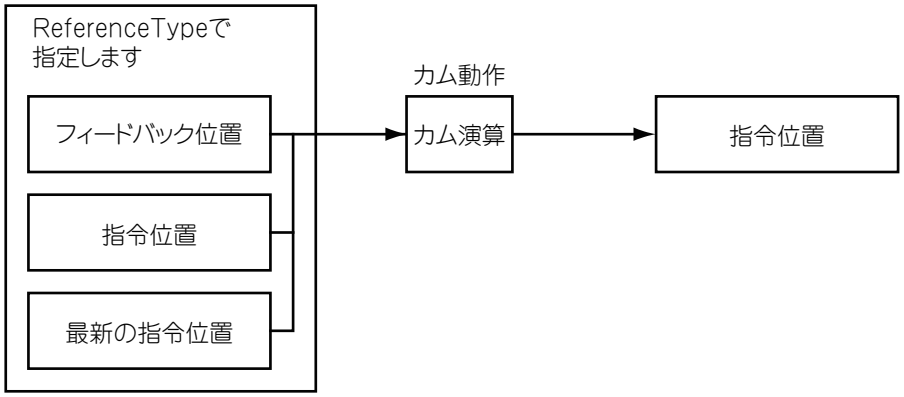
動作開始後、従軸は主軸の速度にギア比を乗じた速度を目標として、加減速動作を行います。
従軸が従軸同期位置に到達するまでが Catching phase（追いかけ中）で、従軸が従軸同期位置に到達すると InSync（ギア同期中）となります。いずれも、従軸は主軸の位置に同期します。
ギア比が正数のときは Slave（従軸）は Master（主軸）の同方向に移動し、負数のときは Slave（従軸）は Master（主軸）の逆方向に移動します。下の図はギア比が正数のときの動作を示します。

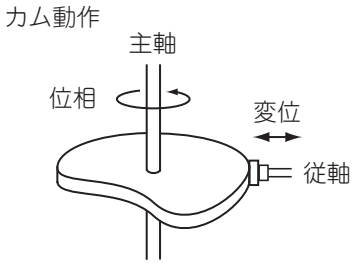


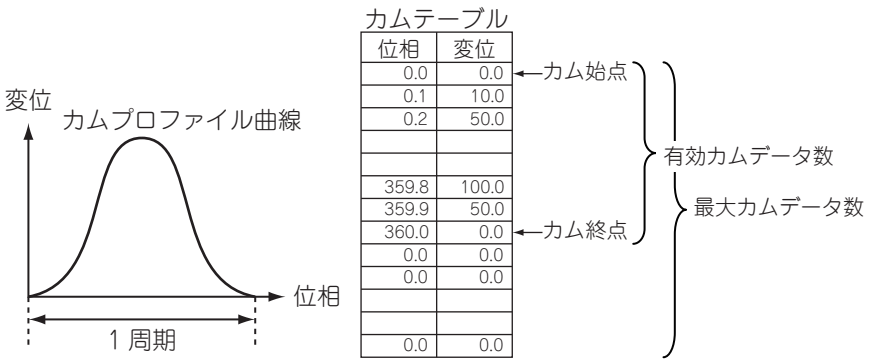
位置指定ギア動作の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_GearInPos（位置指定ギア動作）命令、MC_GearOut（ギア動作解除）命令、および MC_Stop（強制停止）命令を参照してください。

9-2-4 カム動作

カムテーブルに従って従軸が主軸に位置同期して動作する機能です。
MC_CamIn（カム動作開始）命令でカム動作を開始し、MC_CamOut（カム動作解除）命令、または MC_Stop（強制停止）命令でカム動作を解除します。
カム動作に関する情報のモニタは、MC_CamMonitor（カムモニタ）命令を使用します。
カムテーブルは、Sysmac Studio のカムエディタ機能を使用して作成し、CPU ユニット本体にダウンロードしておきます。
CPU ユニットへのダウンロードは、Sysmac Studio の「同期」機能を使用します。







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は、MC_CamIn（カム動作開始）命令の実行中に、ブレンディングを指定して別の MC_CamIn（カム動作開始）命令を多重起動することで、従軸が停止することなくカムテーブルを切り替えながら動作させることができます。

カム動作の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_CamIn（カム動作開始）命令、および MC_CamOut（カム動作解除）命令、MC_Stop（強制停止）命令を参照してください。
また、カムエディタ機能の詳細については、『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。

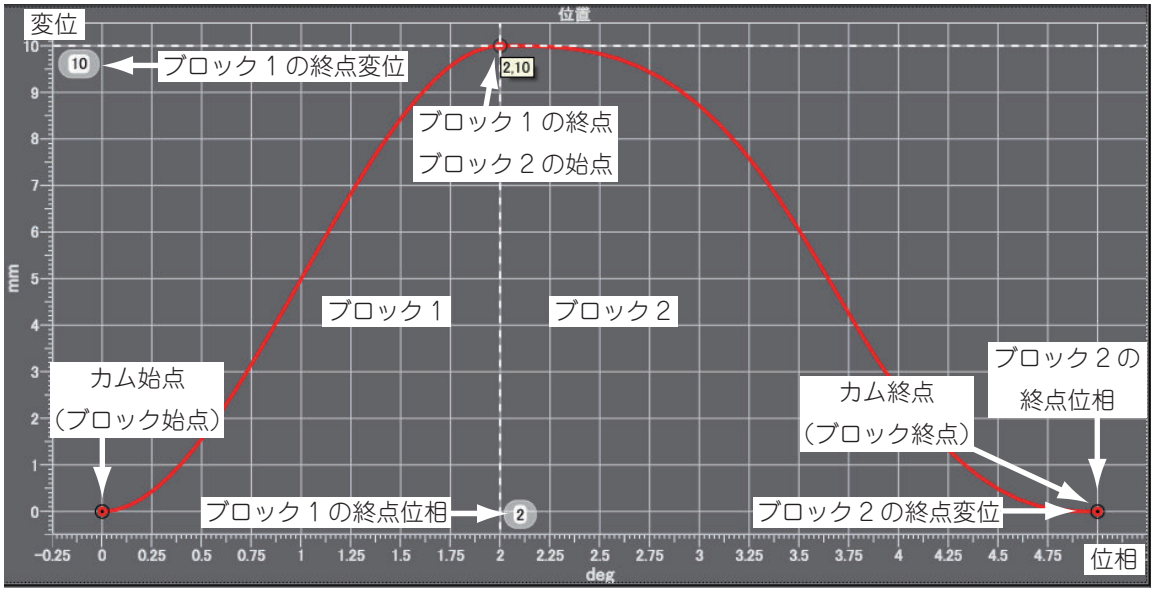
9-2-5 カムテーブル

ここでは、カム動作で使用するカムテーブルについて説明します。

カムテーブルの用語

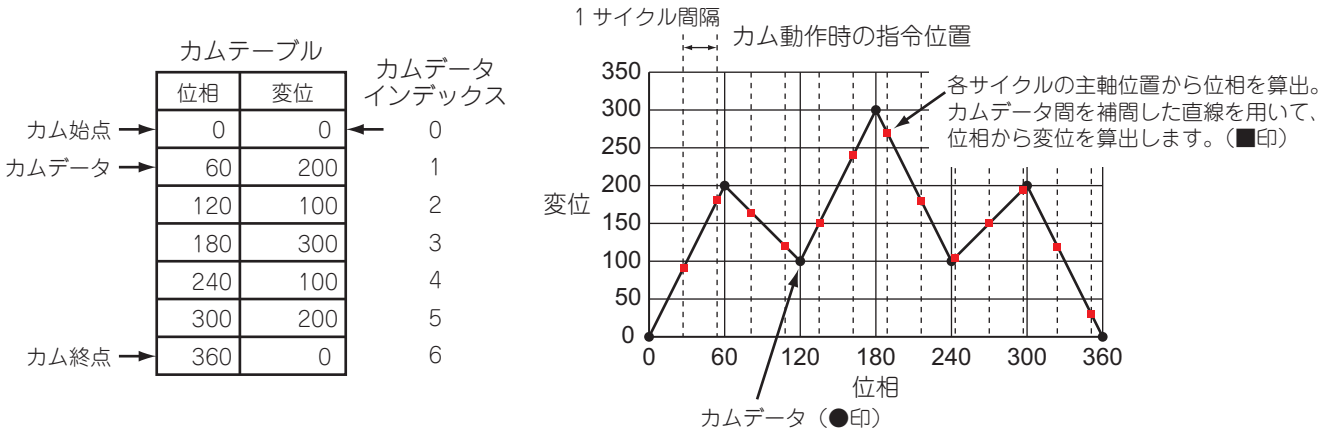
用語	説明
カム動作	1 つの主軸と 1 つの従軸を持ち、カムプロファイル曲線に従って主軸の位相から従軸の変位を求めて動作させる機能です。
カムプロファイル曲線	カム動作を実現するための位相と変位の関係を表す曲線です。 Sysmac Studio のカムエディタで作成します。カムプロファイル曲線を CPU ユニットへダウンロードすると、ユーザプログラムでカムデータ変数として扱うことができます。 CPU ユニットへのダウンロードは、Sysmac Studio の「同期」機能を使用します。
カムブロック	カム曲線を選択できる区間です。 前カムブロックの終点と、自カムブロックの終点の間を表します。
カム曲線	カム特性を表す曲線です。カムブロックごとにカム曲線を選択できます。 指定した点から Sysmac Studio が算出する位相幅、変位幅を用いて、実際のカムプロファイル曲線を Sysmac Studio が作成します。 直線、等加速度、トラペクロイド曲線などがあります。
カムデータ	カム動作を実現するための位相（主軸）と変位（従軸）から構成されるデータです。
カムデータ変数	カムデータを構造体配列化した変数です。
カムテーブル	カムデータのデータテーブルを表します。 位相が昇順でなければ、不正なカムテーブルとみなします。
カム始点	カムデータの先頭のデータを表します。
カム終点	カムデータの有効カムデータの最後を表します。 カム終点 < カムデータ数の場合、カム終点より後ろの位相と変位は「0」です。
カムブロック始点	カムブロックの開始点です。カム動作の開始部では、「カム始点」と同じ点となります。カムプロファイル曲線が連続する場合は「カムブロック終点」と同じ点になります。
カムブロック終点	カムブロックの終了点です。カム動作の終了部では、「カム終点」と同じ点となります。カムプロファイル曲線が連続する場合は「カムブロック始点」と同じ点になります。 「カムブロック終点」は、（横軸、縦軸）＝（位相終点、変位終点）で定義されます。
オリジナルカムデータ	カムエディタで、カムプロファイル曲線を分割して作成したカムデータです。
プログラム変更カムデータ	CPU ユニットの運転中に、ユーザプログラムで変更したカムデータです。
主軸	カム動作の入力となる軸で、［リニアモード］／［ロータリモード］とも指定できます。
従軸	カム動作の出力となる軸で、［リニアモード］／［ロータリモード］とも指定できます。
位相	カムテーブル始点位置からの主軸の相対量です。
変位	主軸追従距離からの従軸の相対量です。
有効カムデータ	カム始点以外で、位相が「0」以外の数値を持つカムデータです。
無効カムデータ	カム始点以外で、位相が「0」のカムデータです。
有効カムデータ数	カムデータの組数を表します。

用語	説明
最大カムデータ数	カムテーブルが持つことができるカムデータの最大組数です。
カムデータインデックス	カムデータが実行される番号です。
カムテーブル開始位置	カム始点（位相＝ 0）に該当する主軸の絶対位置です。
主軸追従距離	従軸がカム動作を開始する主軸位置で絶対位置もしくは相対位置で表します。 相対位置はカム始点位置が基準となります。
開始位置方式指定	主軸追従距離を絶対位置で表すか、相対位置で表すかの指定です。
空きカムデータ	終点より後ろに設定できる位相と変位が「0」のカムデータです。
接続速度	カムプロファイル曲線間を接続する際の速度。指定できない曲線もあります。
接続加速度	カムプロファイル曲線間を接続する際の加速度です。指定できない曲線もあります。
位相間隔幅	カムプロファイル曲線を位相（横軸）で分割する際の幅です。位相間隔幅で曲線を分割した際の各点がカムテーブルのカムデータとなります。カムブロックごとに必ず指定しなければなりません。



カムテーブルとは

MC 機能モジュールでは、主軸の位相と従軸の変位からなる 1 対のデータをカムデータ、複数のカムデータの組み合わせをカムテーブルとして定義しています。
カムテーブルは、Sysmac Studio のカムエディタで作成します。また、ユーザプログラムからカムテーブル内のカムデータを変更することができます。
カムテーブルのカムデータの位相と変位の値は「0.0」の始点からの相対量として表します。
カム動作中は、主軸の位相に隣接する 2 つのカムデータを直線で補間した変位が従軸への指令位置となります。
カムテーブルのカムデータ数を多くすると、軌跡精度が高くなめらかなカムプロファイル曲線になります。



使用上の注意

- カムテーブルのカムデータは、位相の値が昇順になるように作成します。位相が昇順でない場合は、カム動作命令実行時に命令で異常が発生します。
- 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（イベントコード: 54390000 Hex）」が発生します。

カムテーブル仕様

項目	説明
1 カムテーブル当たりの最大カムデータ数	65,535 点
全カムデータのサイズ上限	1,048,560 点 *1
カムテーブル数の上限	640 テーブル *2
カム動作の切り替え	モーション制御命令により別のカム動作に切り替え可能
カムデータの変更	ユーザプログラムで編集可能 カムテーブル生成命令により書き換え可能 *3
カムデータの保存	カムテーブル保存命令により不揮発性メモリに保存可能
カムデータに付随する情報	カムエディタ表示用の情報をダウンロード、アップロード可能 *4
カムデータがメインメモリに反映されるタイミング	• Sysmac Studio からのダウンロード時 • 電源投入時

*1. 1 カムテーブル 65,535 点の場合、最大 16 カムとなります。0.1deg 分解能の場合、1 カムテーブルは 3600 点となり、最大 291 カムとなります。
*2. 合計 10MB 以下としてください。
*3. カムテーブル生成命令は、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することができます。
*4. 「ダウンロード」および「アップロード」は、Sysmac Studio の「同期」機能を使用します。

カムテーブルのデータ型

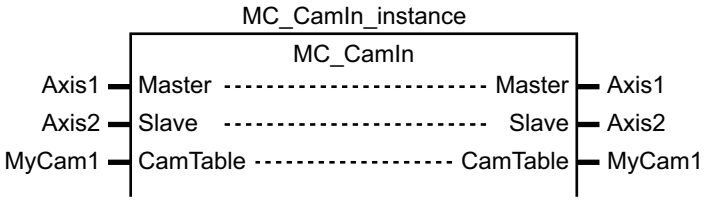
カムテーブルは、カムデータ構造体を配列宣言して使用します。カムデータ構造体は、あらかじめ次のように型宣言されています。

```
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 のカムエディタでカムデータを作成し、カムテーブル名とカムデータ数（配列のサイズ）を指定しなければなりません。
たとえば、カムテーブル名「MyCam1」、カムデータ数を 1000 個にする場合、次のように宣言します。

```
VAR
  (*カムテーブル*)
  MyCam1  :  ARRAY [0..999] OF _sMC_CAM_REF;
END_VAR
```

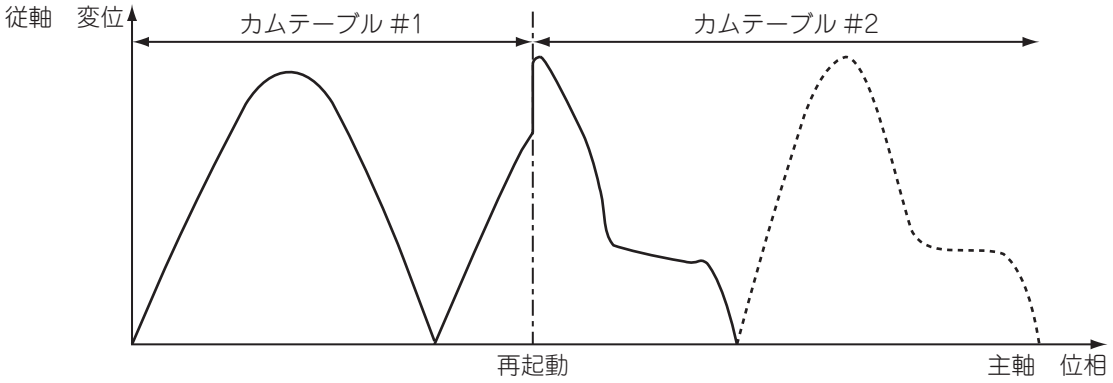
カム動作命令で「MyCam1」を指定する場合、次のように記述します。ここでは、主軸を Axis1、従軸を Axis2 とします。



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。

カムテーブルの切り替え

カム動作中にカム動作命令を再起動することで、カムテーブルを切り替えることができます。
切り替え後は、再起動時に指定したカムテーブルでカム動作を行います。
MC_CamIn（カム動作開始）命令の出力変数「EndOfProfile（終点実行）」および「Index（インデックス）」は切り替えたカムテーブルに従って出力されます。



使用上の注意

- 切り替え先のカムテーブルは、あらかじめ不揮発性メモリに保存しておく必要があります。
- 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。

カムデータの読み書きとカムテーブルの保存

ほかの変数と同様にユーザプログラムからカムデータを読み書きすることができます。
たとえば、カムテーブル名が「MyCam1」の 1 番めの配列要素について、位相は「MyCam1[0].Phase」で、変位は「MyCam1[0].Distance」で指定できます。
ユーザプログラムで書き換えたカムデータは、MC_SaveCamTable（カムテーブル保存）命令により、カムテーブルとして CPU ユニットの不揮発性メモリに保存することができます。

使用上の注意

- カムテーブル保存命令が起動される前に、CPU ユニットの電源が OFF された場合や、本命令による保存が正常完了しなかった場合、または Sysmac Studio からダウンロードを実行すると、書き換えられたカムデータは消滅します。
- CPU ユニット上のユーザプログラムでカムデータを書き換える場合には、書き換えデータが消失しないように注意してください。
- 不揮発性メモリに保存したカムデータは、Sysmac Studio のアップロード機能で取り込むことができます。
- 「ダウンロード」および「アップロード」は、Sysmac Studio の「同期」機能を使用します。

配列指定については、『NJ/NX シリーズ CPU ユニット ユーザーズマニュアル ソフトウェア編 (SBCA-467)』を参照してください。

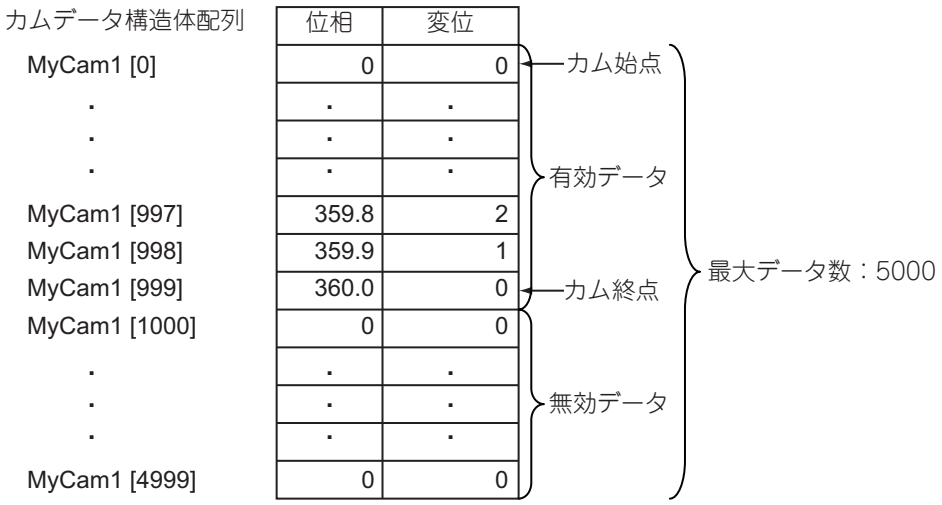
カムテーブル保存命令の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_SaveCamTable（カムテーブル保存）命令を参照してください。

カムテーブルのプロパティ更新と終点インデックス更新

MC 機能モジュールでは、カムテーブルのカム終点を識別する必要があります。
カム動作中にユーザプログラムから有効カムデータ数を変更する書き換えを行った場合は、有効カムデータ数を最新に更新する必要があります。
そのために、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）命令を使用します。
どちらの命令もカムテーブルに対する操作と得られる結果は同じですが、MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令は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。

カム終点とは、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」となるカムデータの 1 つ前に配置されたデータのことです。位相「0」を検出した以降のカムデータは無効となります。

たとえば、次のようなカムテーブルの場合、命令の出力変数「EndPointIndex」（終点インデックス）は「999」、「MaxDataNumber」（最大データ数）は「5000」となります。



使用上の注意

- 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。
- 有効カムデータ数を変更する書き換えを行った後は本命令を実行してください。
  有効カムデータ数を更新しないと、カム動作や MC_CamIn 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。

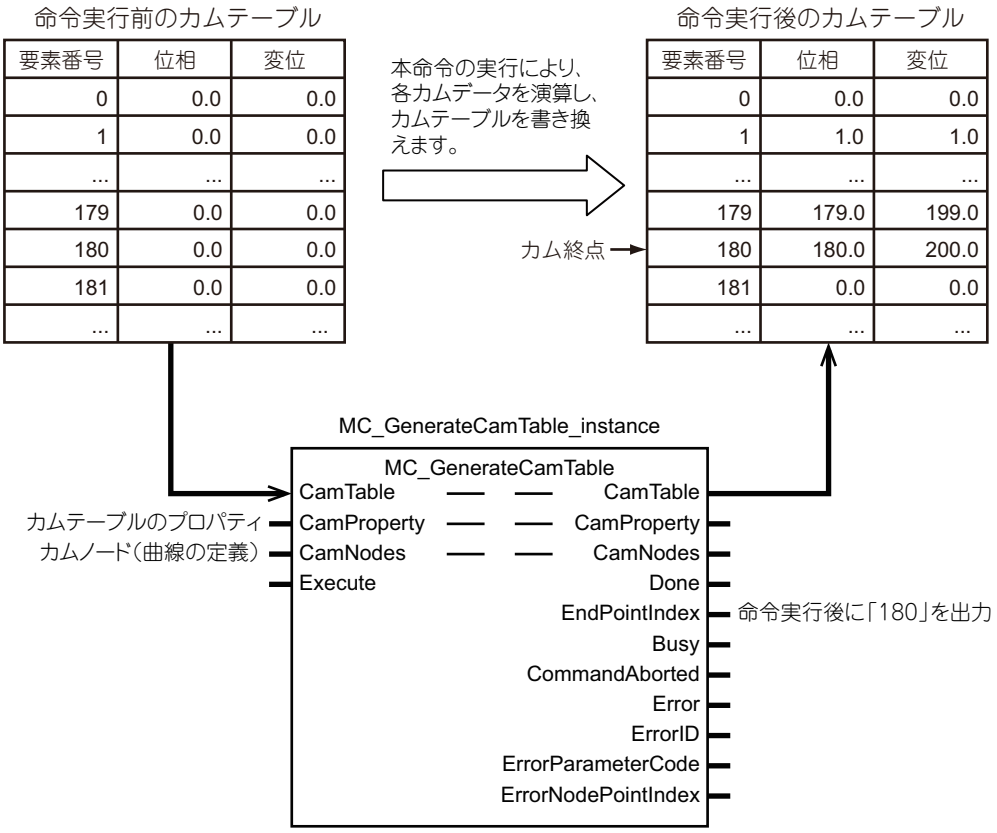
カムテーブルプロパティ更新の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_SetCamTableProperty（カムテーブルプロパティ更新）命令を参照してください。
命令起動から完了までの処理時間の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令を参照してください。

カムテーブルの生成

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は、ユーザプログラム中の MC_GenerateCamTable（カムテーブル生成）命令により、カムテーブルを生成することができます。

MC_GenerateCamTable（カムテーブル生成）命令の CamProperty（カムプロパティ）と CamNodes（カムノード）に指定した値に従ってカムデータの値を演算し、命令の入出力変数 CamTable（カムテーブル）に指定したカムデータ変数を書き換えます。

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。
MC_GenerateCamTable（カムテーブル生成）命令の完了後に 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）命令を実行する必要はありません。



カムデータ変数は、カムデータ構造体_sMC_CAM_REF 型の配列変数で、Sysmac Studio のカムエディタで作成します。
CamProperty（カムプロパティ）には、カムプロパティ変数を指定します。カムプロパティ変数は、カムプロパティ構造体_sMC_CAM_PROPERTY 型の変数で、Sysmac Studio のグローバル変数テーブルでユーザ定義変数として作成するか、または Sysmac Studio のカムデータ設定で作成します。
CamNodes（カムノード）には、カムノード変数を指定します。カムノード変数は、カムノード構造体_sMC_CAM_NODE 型の配列変数で、Sysmac Studio のグローバル変数テーブルでユーザ定義変数として作成するか、または Sysmac Studio のカムデータ設定で作成します。
カムプロパティ変数とカムノード変数を総称して「カム定義変数」と呼びます。

カム定義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。表示器などから都度設定するときは、「非保持」のままでもかまいません。
カム定義変数を Sysmac Studio のカムデータ設定で作成した場合、変数の保持属性は「保持」で固定となります。

MC_GenerateCamTable（カムテーブル生成）命令に指定する値を表示器などから設定することで、Sysmac Studio を使用せずにカムデータ変数を作成したり、カム動作を調整したりできます。
次に、カム動作を調整する手順の概要を説明します。

1 **あらかじめ、次の処理をユーザプログラムで作成しておきます。**
- 表示器で設定したカム定義変数の値をカムテーブル生成命令に指定する処理
- カムテーブル生成命令で生成されたカムデータ変数を表示器にグラフ表示する処理
- EndPointIndex（終点インデックス）の値を表示器に表示する処理

2 **カム定義変数の値を表示器で設定します。**

3 **カムテーブル生成命令を実行します。**

4 **生成されたカムテーブルの形状と終点インデックスの値を表示器で確認します。**

5 **カムテーブルの形状とカムデータ数に問題がなければカム動作を実行します。**

6 **カム動作の結果を確認しカム定義変数の値の変更を検討します。**

7 **2～6 の手順を繰り返します。**

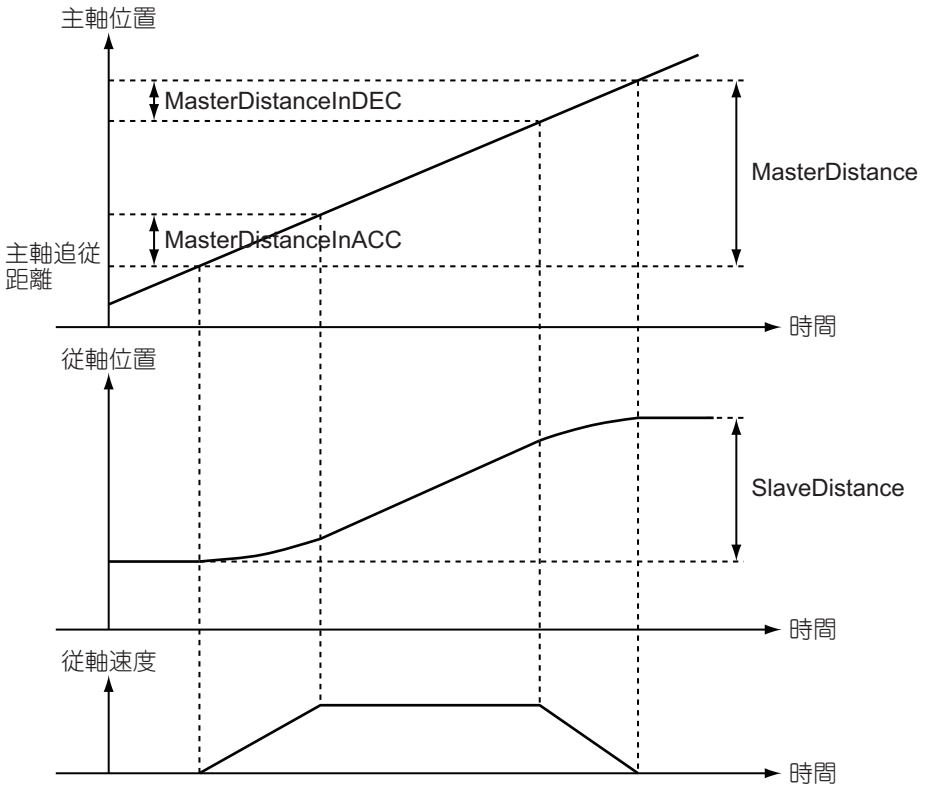
カム定義変数の詳細とカムテーブル生成命令の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_GenerateCamTable（カムテーブル生成）命令を参照してください。

Sysmac Studio でのカム定義変数の作成と転送については、『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。

9-2-6 台形パターンカム動作

指定した従軸が、指定した主軸に同期しながら台形カーブで位置決めを行う機能です。
電子カムの一種ですが、カムエディタで作成したカムテーブルは使用しません。
MC_MoveLink（台形パターンカム）命令の起動で動作を開始します。
動作中の軸を停止するには、MC_Stop（強制停止）命令を使用します。
動作対象の軸を「Slave」（従軸）とし、「Master」（主軸）、「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）、「SlaveDistance」（従軸移動距離）、「MasterStartDistance」（主軸追従距離）を指定します。
主軸には指令位置またはフィードバック位置を指定できます。
同期動作の開始条件として、［命令開始時］、［トリガを検出した時］、［主軸が主軸追従距離に到達した時］のいずれかを指定することができます。

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。
下図の同期開始位置は、同期開始条件が有効になった時の位置を示します。

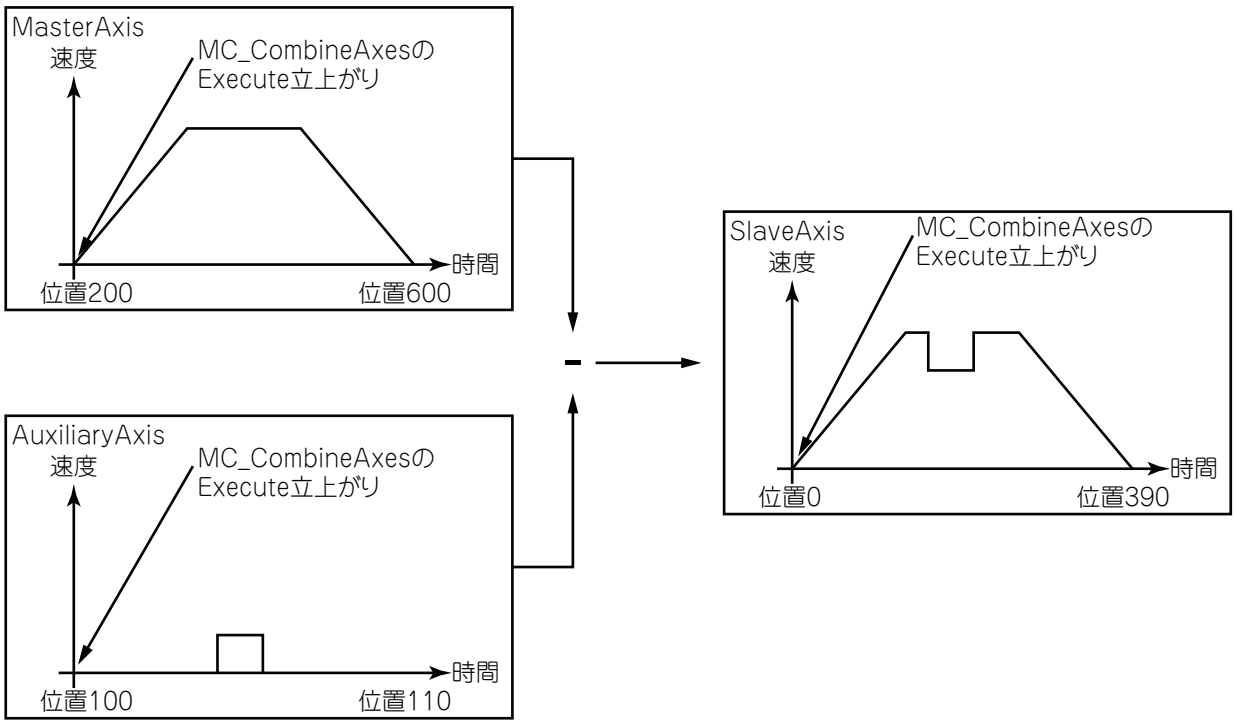


台形パターンカム動作の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_MoveLink（台形パターンカム）命令、および MC_Stop（強制停止）命令を参照してください。

9-2-7　加減算位置決め

2 つの位置を加算または減算した値を、従軸の指令位置とする機能です。
MC_CombineAxes（加減算位置決め）命令の起動で動作を開始します。
動作中の軸を停止するには、MC_Stop（強制停止）命令を使用します。

下図は減算の場合の動作例です。
SlaveAxis（従軸）指令現在位置 = MasterAxis（主軸）指令現在位置 − AuxiliaryAxis（補助軸）指令現在位置

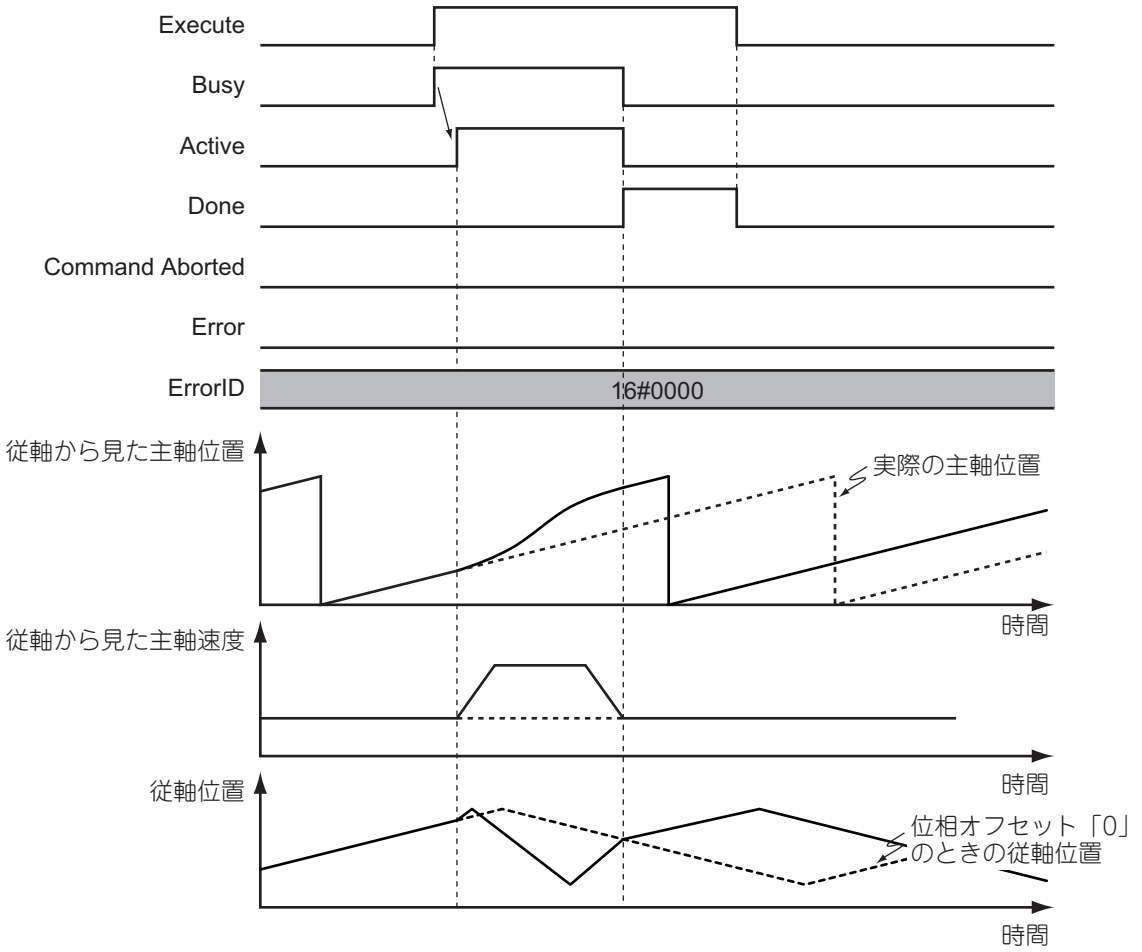


加減算位置決め動作の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_CobineAxes（加減算位置決め）命令、および MC_Stop（強制停止）命令を参照してください。

9-2-8　主軸位相補正

動作中の命令に対して、従軸から見た主軸の位相補正を行う機能です。
従軸から見た補正量は相対量であり、同期状態の間、従軸は主軸の相対量に対して同期します。
MC_Phasing（主軸相対値位相補正）命令を起動することで、同期制御命令に対して位相補正を行うことができます。

MC_Phasing（主軸相対値位相補正）命令は、「位相補正量」、「目標速度」、「加速度」、「減速度」、「ジャーク」を指定できます。



主軸位相補正が可能な同期制御命令、および主軸位相補正機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_Phasing（主軸相対値位相補正）命令を参照してください。

9-2-9　従軸位置補正

同期制御中の従軸に対して位置補正する機能です。
同期制御中の従軸に対して、入力変数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。
MC 機能モジュールが 2 軸に対して同量の移動量を指令しても、メカ的なゆがみなどにより、移動量が異なる場合に補正することができます。
MC_SyncOffsetPosition（サイクリック同期位置オフセット補正）命令を起動することで、同期制御中の従軸の位置補正を周期的に行うことができます。
また、MC_OffsetPosition（位置オフセット補正）命令を起動することで、同期制御中の従軸の位置補正を加減速カーブをつけて行うことができます。

従軸位置補正機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_SyncOffsetPosition（サイクリック同期位置オフセット補正）命令、および MC_OffsetPosition（位置オフセット補正）命令を参照してください。

9-2-10　マルチモーションでの同期制御の実現方法

MC 機能モジュールの標準機能では、マルチモーションで異なるタスクに割り付けている軸間の同期制御命令を実行すると、「主軸指定不正（イベントコード: 54620000 Hex）」が発生します。
しかし、MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令を使用することにより、プライマリ定周期タスクで制御する主軸と定周期タスク（実行優先度 5）で制御する従軸の同期制御が可能となります。

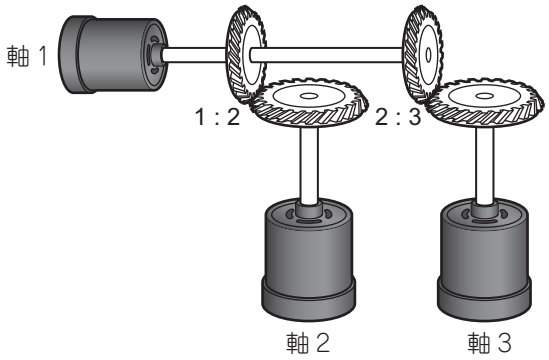
主軸と従軸が同期するカム動作やギア動作は、次の組み合わせで使用可能です。

主軸のタスク	従軸のタスク	
	プライマリ定周期タスク	定周期タスク（実行優先度 5）
プライマリ定周期タスク	モーション制御命令による同期	プライマリ定周期タスクで MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令を実行し、定周期タスク（実行優先度 5）で［仮想主軸］を使用することにより実現可能。
定周期タスク（実行優先度 5）	使用できません。	モーション制御命令による同期

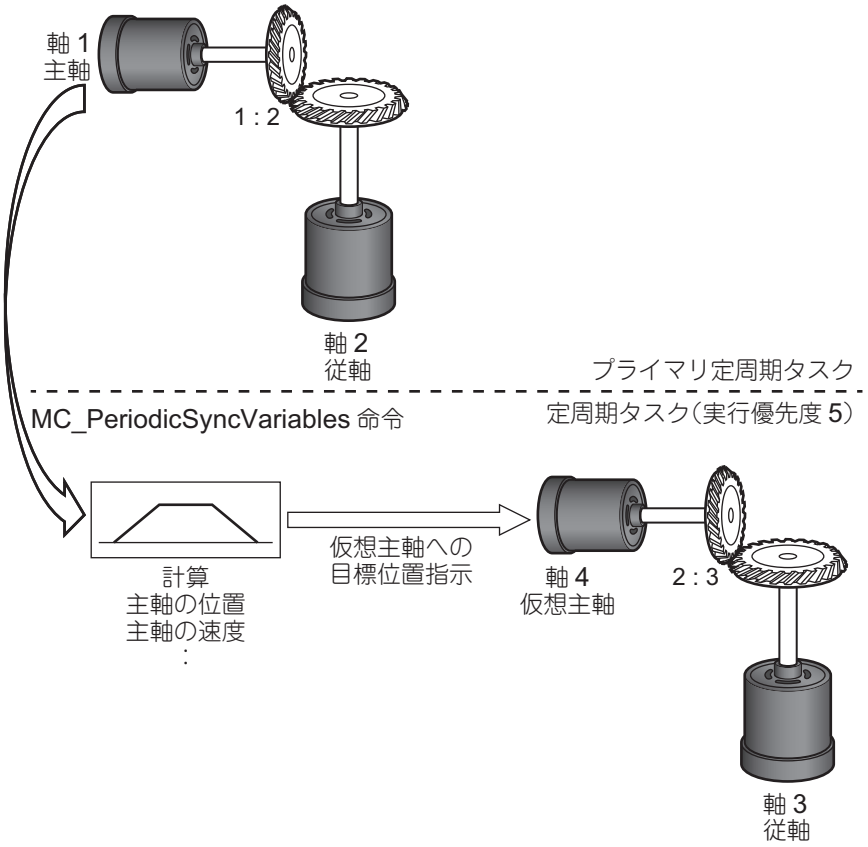
動作例の軸構成

下図に示すように主軸（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。

MC_PeriodicSyncVariables（軸変数タスク間定周期同期）命令の詳細は、『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)』を参照してください。

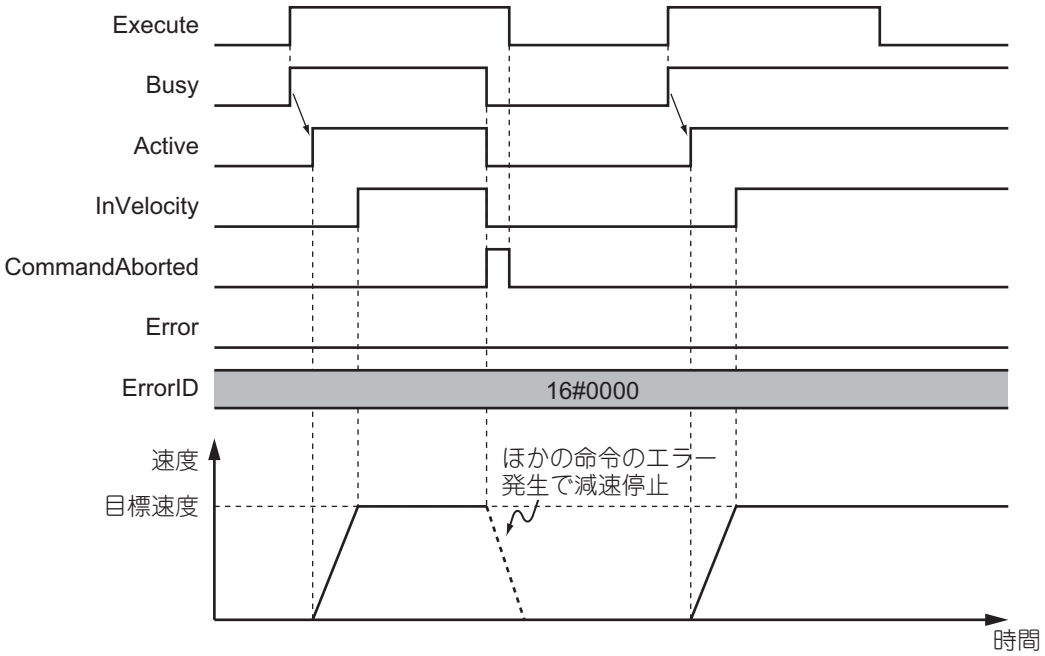
9-3 単軸速度制御

ここでは、単軸の速度制御の動作について説明します。

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

9-3-1 速度制御

速度制御とは、指令した速度で軸を送り続ける機能です。
加速度、減速度およびジャークを指定することもできます。
軸を停止するには、MC_Stop（強制停止）命令を使用するか、ほかの動作命令を起動します。
目標速度に「0」を指定すると、軸は移動しませんが軸の状態は軸動作中となります。
速度制御中に、ほかのモーション制御命令の多重起動で割りこまれた場合は、目標速度に到達した後の状態で動作が切り替わります。



MC 機能モジュールでは、サーボドライバなどを位置制御モードとして指定した目標速度になるように目標位置を指令します。
サーボドライバなどでは位置制御ループが有効なので、外乱などで指令速度より遅くなり位置偏差が大きくなると、位置偏差をなくすための速度変動が発生します。

機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_MoveVelocity（速度制御）命令を参照してください。

9-3-2 サイクリック同期速度制御

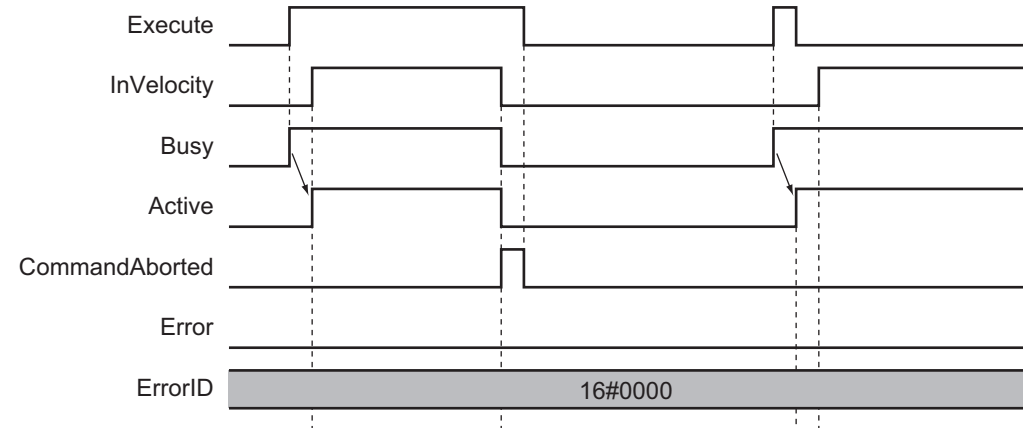
サーボドライバの制御モードを速度制御モードとして、制御周期ごとに指令速度を出力する機能です。

使用上の注意

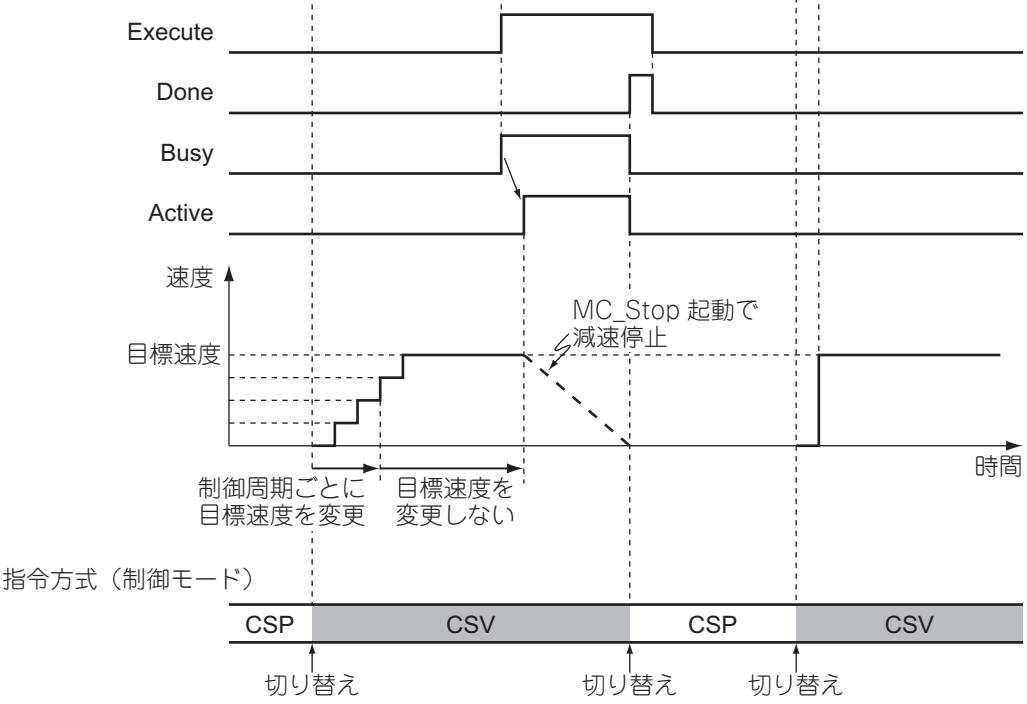
NX シリーズ パルス出力ユニットを使用する場合は、この機能を使用できません。

軸を停止するには、MC_Stop（強制停止）命令を使用するか、ほかの動作命令を起動します。
目標速度に「0」を指定すると、軸は移動しませんが軸の状態は軸動作中となります。

MC_SyncMoveVelocity 命令



MC_Stop 命令



サーボドライバ側は速度制御ループで指令を受けるため、外乱になどで指令速度より遅くなっても位置偏差をなくすための速度変動は発生しません。

機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_SyncMoveVelocity（サイクリック同期速度制御）命令を参照してください。

9-4 単軸トルク制御

トルク制御とは、指令したトルクを発生させ続ける機能です。
Torque（目標トルク）に到達するまでのトルクの変化率を TorqueRamp（トルクランプ）で指定することもできます。

MC 機能モジュールでは、サーボドライバのトルク制御モードを使用しています。
サーボドライバ側は、MC 機能モジュールからのトルク指令値をトルク制御ループで受けて制御しています。
モーション制御命令の入力変数「Velocity」（制限速度）で、サーボドライバ側の「速度制限値」を指定することができます。これにより、トルク制御モードでモータの負荷が軽くなった場合などのモータの高速回転を制限することができます。

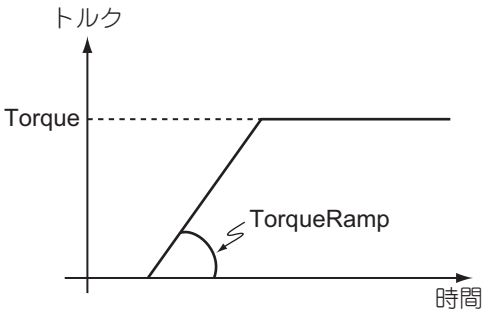
機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_TorqueControl（トルク制御）命令を参照してください。

使用上の注意

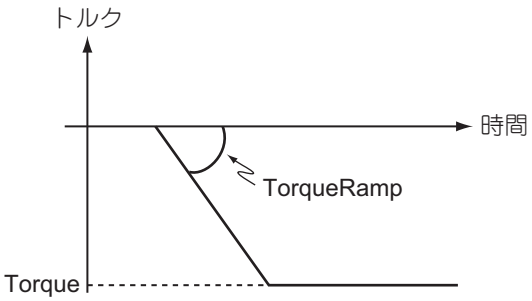
- トルク制御時は、安全のために必ず速度制限値を設定してください。
- NX シリーズ パルス出力ユニットを使用する場合は、この機能を使用できません。

軸を停止するには、MC_Stop（強制停止）命令を使用するか、ほかの動作命令を起動します。
Torque（目標トルク）に「0」を指定すると、軸は移動しませんが軸の状態は軸動作中となります。

● 方向指定 = 正方向の場合



● 方向指定 = 負方向の場合



9-5 単軸制御の共通機能

ここでは、単軸制御の共通機能について説明します。

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

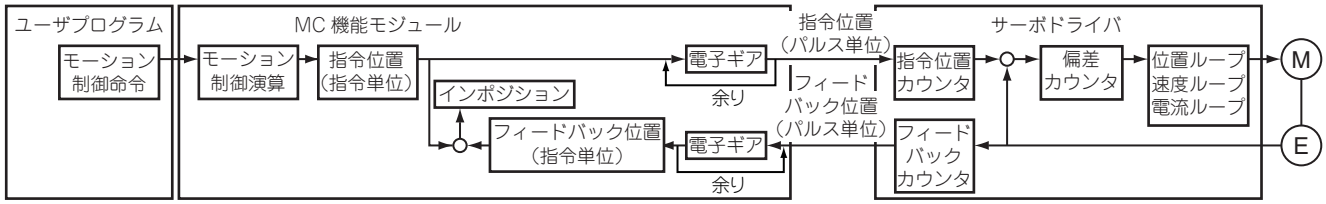
9-5-1 位置

位置の種類

MC 機能モジュールには、つぎの 2 種類の位置があります。

位置種別	定義
指令位置	MC 機能モジュールが、軸を制御するために出力する位置です。
フィードバック位置	サーボドライバやカウンタから入力した実際の位置です。

EtherCAT スレーブサーボドライバの場合の指令位置とフィードバック位置の関係を下図に示します。



指令位置とフィードバック位置は下記の項目を共有しています。

項目	指令位置	フィードバック位置
カウントモード	リニアモードまたはロータリモードを設定します。	指令位置と同じカウントモードになります。
位置の単位	mm, μm, nm, inch, degree, pulse からいずれかを設定します。	指令位置と同じ単位になります。
ソフトウェアリミット	ソフトウェア的な動作可能範囲を設定します。	指令位置と同じ範囲になります。
現在位置変更	現在位置を任意の位置に変更します。	指令位置と同じ値が同時に設定されます。*1
原点確定	原点確定または原点未確定の状態を持ちます。	指令位置と同じ原点状態になります。

*1. 変更前に位置偏差があれば、位置偏差の値を保持した値がフィードバック位置に設定されます。

参考

NX シリーズ 位置インタフェースユニットの場合は、『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。

位置に関する軸パラメータ

パラメータ名称	機能	設定範囲	初期値
インポジション幅	位置決め完了幅を設定します。（単位: 指令単位）	倍精度実数型の正数、0	10

パラメータ名称	機能	設定範囲	初期値
インポジションチェック時間	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のみ位置決め完了のチェックを継続し、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。 （単位: ms）	0～10000	0
ソフトウェアリミット機能	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	0～4	0
正方向ソフトウェアリミット	正方向側のソフトウェアリミット値を設定します。（単位: 指令単位）	倍精度実数型の負数、正数、0	2147483647
負方向ソフトウェアリミット	負方向側のソフトウェアリミット値を設定します。（単位: 指令単位）	倍精度実数型の負数、正数、0	-2147483648
位置偏差オーバー値	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。（単位: 指令単位）	倍精度実数型の正数、0	0
位置偏差警告値	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。（単位: 指令単位）	倍精度実数型の正数、0 （位置偏差オーバー値以下）	0

軸動作の目標位置指定

実際の位置決め動作の位置や距離は、モーション制御命令の入力変数「Position」（目標位置）、「Distance」（移動距離）で指定します。

位置のモニタ

ユーザプログラムから、軸変数を読み取ることで位置をモニタすることができます。
なお、変数名は、_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_AX[0-255].Cmd.Pos	LREAL	指令現在位置	指令位置の現在値です。 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出力します。
_MC_AX[0-255].Act.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です。

9-5-2 速度

速度の種類

MC 機能モジュールで使用する軸の速度には、次の 2 種類があります。

速度種別	定義
指令速度	MC 機能モジュールが、軸を制御するために出力する速度です。
フィードバック速度	サーボドライバやカウンタから入力したフィードバック位置をもとに、MC 機能モジュール内で算出された速度です。*1

*1. PDO マッピングにフィードバック速度（606CHex）をマッピングし、［フィードバック現在速度］機能へ割り付けた場合はこの値を表示しています。

速度の単位

速度の単位は、「指令単位／s」です。
指令単位は位置の表示単位と電子ギアで単位変換された値です。

速度に関する軸パラメータ

パラメータ名称	機能	設定範囲	初期値
最高速度	軸の最高速度を指定します。 軸動作命令で最高速度を超える目標速度が指定された場合は、最高速度で動作します。	倍精度実数型の正数	400,000,000
起動速度 *1	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。 （単位: 指令単位／s）	倍精度実数型の正数	0
ジョグ最高速度	各軸のジョグ最高速度を設定します。*2 最高速度を超えないように設定してください。 （単位: 指令単位／s）	倍精度実数型の正数	1,000,000
速度警告値	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。（単位: %）	0～100	0
フィードバック速度フィルタ時定数	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。（単位: ms） 軸の速度が遅い場合のフィードバック現在速度のばらつきを抑えたい場合に使用します。	0～100	0

*1. 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することができます。
*2. ジョグ最高速度を超える速度指令値を指定した場合は、ジョグ最高速度で指令します。

軸動作の目標速度指定

実際の位置決め動作の速度は、モーション制御命令の入力変数「Velocity」（目標速度）で指定します。

速度のモニタ

ユーザプログラムから、軸変数を読み取ることで速度をモニタすることができます。
なお、変数名は、_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_AX[0-255].Cmd.Vel	LREAL	指令現在速度	指令速度の現在値です。 正方向への移動はプラス符号、負方向への移動はマイナス符号が付きます。

変数名	データ型	名称	機能
_MC_AX[0-255].Act.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です。 正方向への移動はプラス符号、負方向への移動はマイナス符号が付きます。

9-5-3 加速と減速

加速度と減速度の単位

加速度と減速度の単位は「指令単位／s^2」です。
指令単位は位置の表示単位と電子ギアで単位変換された値です。

加速度と減速度に関する軸パラメータ

パラメータ名称	機能	設定範囲	初期値
最大加速度	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。 （単位: 指令単位／s^2）	倍精度実数型の正数、0	0
最大減速度	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。 （単位: 指令単位／s^2）	倍精度実数型の正数、0	0
加減速オーバー	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	0～2	0
加速度警告値	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。（単位: %）	0～100	0
減速度警告値	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。（単位: %）	0～100	0

*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、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44）」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、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44）」を参照してください。

軸動作の加速度と減速度の指定

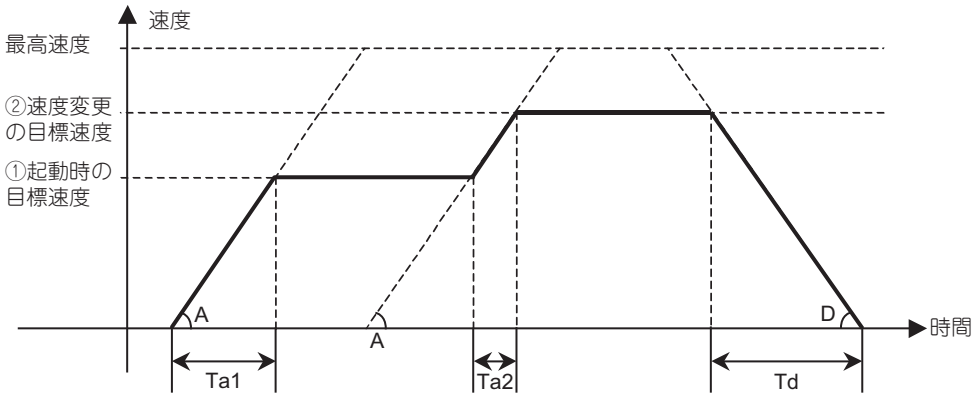
実際の位置決め動作の加速度および減速度は、モーション制御命令の入力変数「Acceleration」（加速度）および「Deceleration」（減速度）で指定します。

加速度と減速度のモニタ

ユーザプログラムからは、軸変数を読み取ることで加速度および減速度をモニタできます。
なお、変数名は、_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_AX[0-255].Cmd.AccDec	LREAL	指令現在加減速度	指令加減速度の現在値です。 加速時はプラス符号、減速時はマイナス符号が付きます。

加減速動作例



【起動時】
Ta1：実際の加速時間
A：加速度

【速度変更時】
Ta2：実際の加速時間
A：加速度

【減速時】
Td：実際の減速時間
D：減速度

指令した移動量が少ない場合、あるいは加減速度が小さい場合は、目標速度に到達しない場合があります。
モーション制御命令の再起動で、変更された新しい加速度（減速度）で動作すると、目標位置を超えてしまう場合は、目標位置に停止可能な加速度あるいは減速度で位置決めを行います。

9-5-4 ジャーク

ジャークは加速度／減速度の変化の比率を指定します。ジャークを指定すると加減速時の速度波形がS字型になるため、機械の衝撃や振動を抑制することができます。

参考

ジャークは、「躍動」や「躍度」あるいは「加加速度」とも呼ばれています。

ジャークの単位

ジャークの単位は「指令単位／s^3」です。
指令単位は位置の表示単位と電子ギアで単位変換された値です。

軸動作のジャークの指定

実際の位置決め動作のジャークは、モーション制御命令の入力変数「Jerk」（ジャーク）で指定します。
加速と減速で共通な値となります。

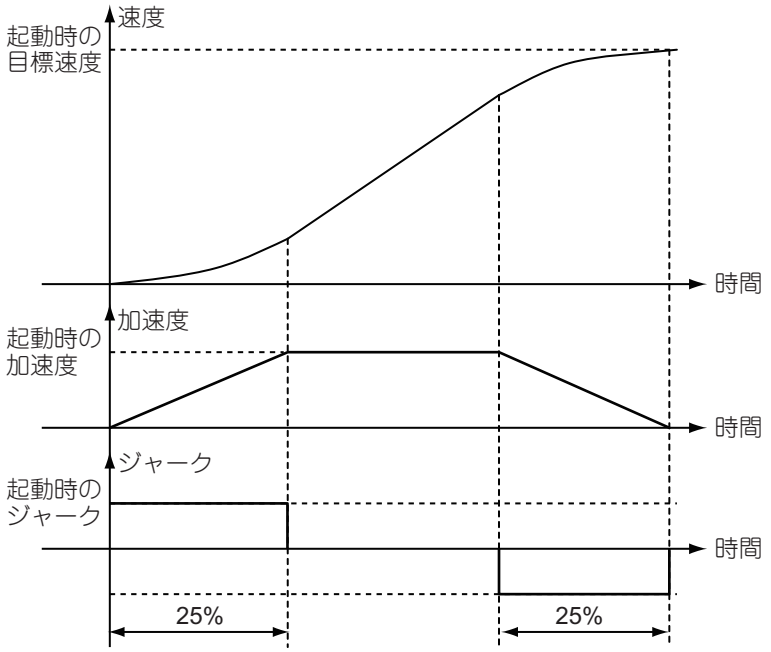
ジャークにはつぎの式で計算した値を設定します。
ジャーク ＝ 加速度／（加速時間×加速している時間のうちジャークを適用する時間の割合／2）

ジャークを適用する時間は、加速開始からの時間と加速終了までの時間に 2 等分されます。そのため、ジャークを適用する時間の割合を 2 で除します。

● ジャークを指定したときの加速制御の例

ジャークが適用される区間の加速度は一定の比率で変化するため、指令速度は S 字型になめらかになります。ジャークが「0」の区間の加速度は固定のため、指令速度は直線状になります。

例: 加速度が 25,000mm/s^2 で、加速時間が 0.1s、ジャークを適用する時間割合が 50%の場合
ジャーク= 25000 / （0.1 × 0.5 / 2） ＝ 1,000,000 （mm/s^3）



ジャークのモニタ

ユーザプログラムからは、軸変数を読み取ることでジャークをモニタできます。
なお、変数名は、_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_AX[0-255].Cmd.Jerk	LREAL	指令現在ジャーク	指令ジャークの現在値です。

9-5-5 動作方向指定

インデックステーブルなどで近回りなどの回転方向を指定して動作させるには、カウントモードをロータリモードに設定します。その上で、絶対位置を指定するモーション制御命令の入力変数「Direction」（方向選択）で動作方向を指定します。

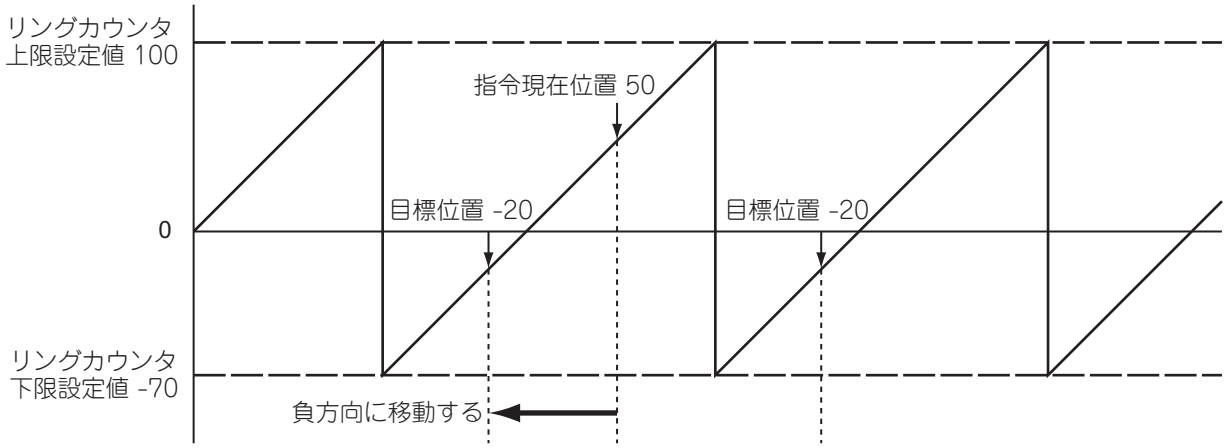
方向選択が［近回り指定］、［正方向指定］、［負方向指定］、［現在方向指定］の場合、目標位置はリングカウンタの 1 回転未満で、リングカウンタの下限値以上、上限値未満の位置を指定できます。
カウントモードがリニアのとき、方向選択は無視され目標位置へ位置決めします。

MC 機能モジュールで指定できる方向選択を下記に示します。

方向選択	動作
近回り指定	指令現在位置と目標位置との距離が短い方向へ向かって動作を開始します。
正方向指定	正方向へ向かって動作を開始します。
負方向指定	負方向へ向かって動作を開始します。
現在方向指定	前回の動作と同じ方向へ向かって動作を開始します。
方向指定なし	リングカウンタの上下限値を通過しない方向へ向かって動作を開始します。 この方向選択では、リングカウンタの上下限値を超えた目標位置を指定できます。そのときの動作は、目標位置と指令現在位置の差分を目標距離とする相対値位置決めです。これにより、リングカウンタの多回転位置へ位置決めできます。

近回り指定の例

指令現在位置が「50」のときに、目標位置「-20」に向かって位置決めする例を示します。

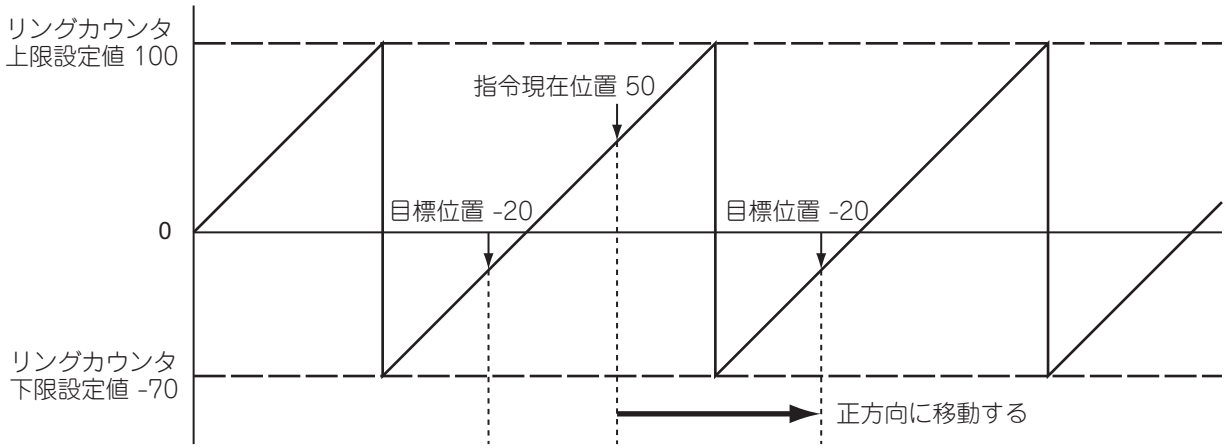


参考

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

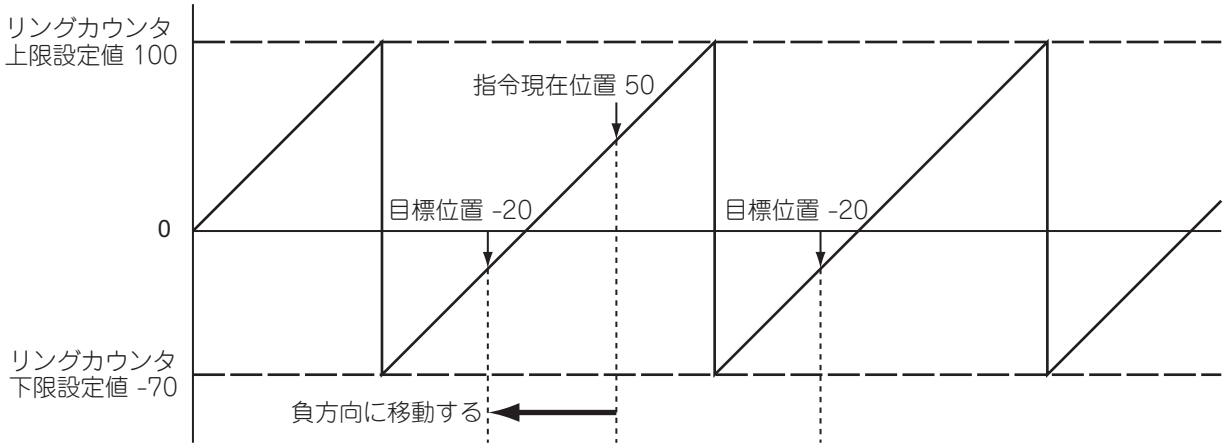
正方向指定の例

指令現在位置が「50」のときに、目標位置「-20」に向かって位置決めする例を示します。



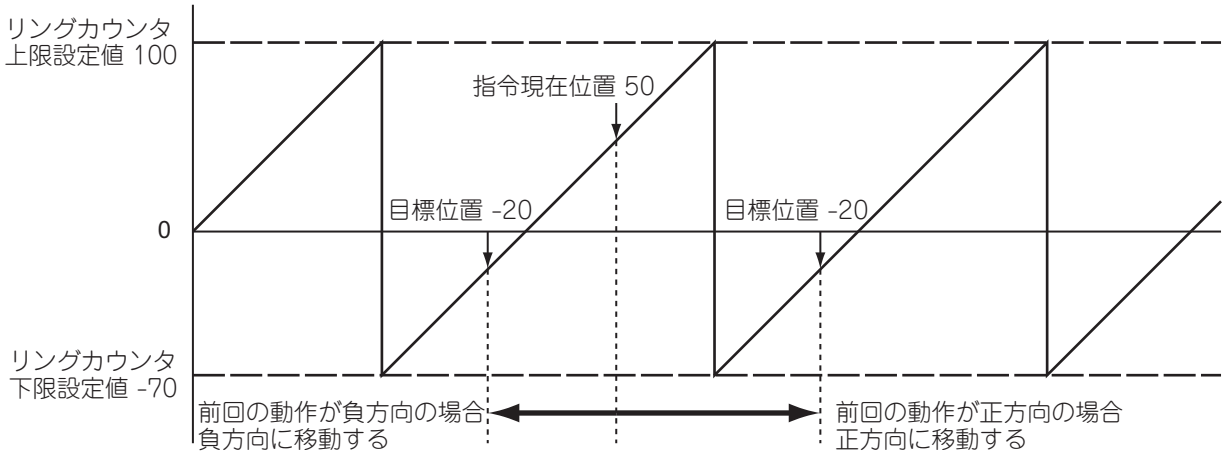
負方向指定の例

指令現在位置が「50」のときに、目標位置「-20」に向かって位置決めする例を示します。



現在方向指定の例

指令現在位置が「50」のときに、目標位置「-20」に向かって位置決めする例を示します。



前回の動作方向は軸変数の「指令方向ステータス」で判別することができます。

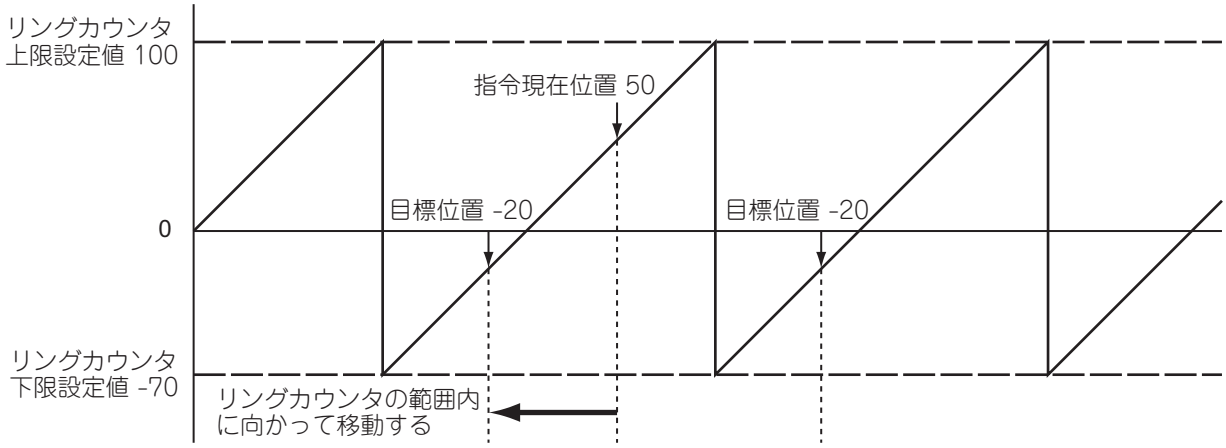
使用上の注意

前回の動作が以下の場合は動作方向に注意してください。

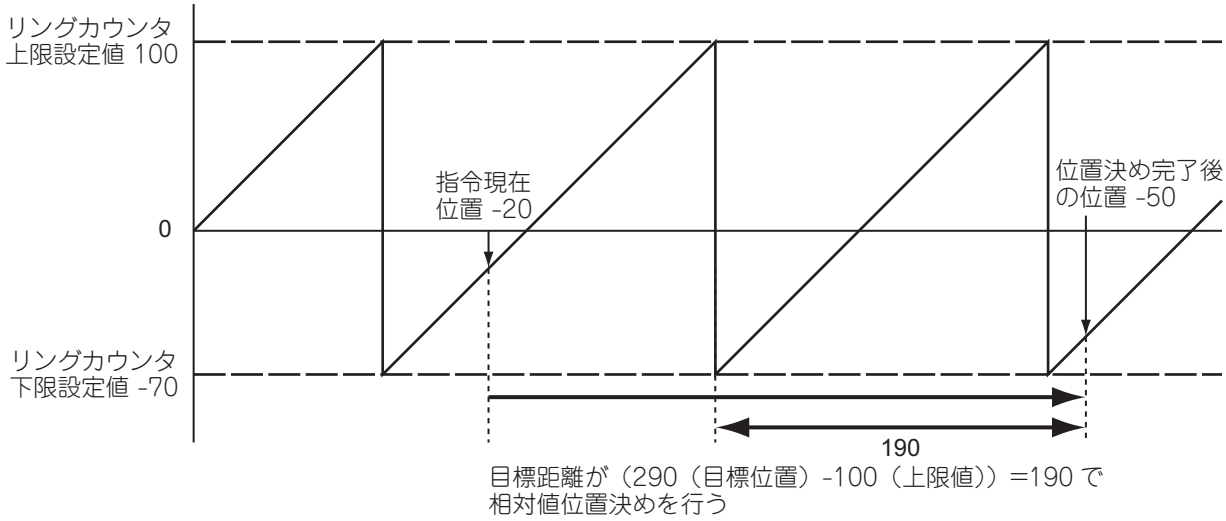
- MC_Home（原点復帰）命令、または MC_HomeWithParameter（パラメータ指定原点復帰）命令で原点入力を検出した位置を通りすぎて反転動作する場合、原点検出方向と反対方向となります。
- MC_Home（原点復帰）命令、または MC_HomeWithParameter（パラメータ指定原点復帰）命令で原点復帰補正値を設定している場合、補正値へ動作する方向となります。
- MC_TouchProbe（外部ラッチ有効）命令で即停止を指定した場合で、ラッチ位置を通りすぎて反転動作する場合。
- MC_MoveFeed（割り込み定寸位置決め）で反転動作する場合。
- MC_ResetFollowingError（偏差カウンタリセット）命令を実行した場は、偏差を「0」にするために与えた指令位置の方向となります。
- 外部入力信号の即停止や限界入力停止で偏差カウンタリセットを指定している場合で、外部入力信号が発生した位置へ反転動作する場合。

方向指定なしの例

指令現在位置が「50」のときに、目標位置「-20」に向かって位置決めする例を示します。



同様に、リングカウンタ上限値が「100」、下限値が「-70」で、指令現在位置が「-20」のときに、目標位置「290」に向かって位置決めする例を示します。



9-5-6　モーション制御命令の再起動

単軸での動作中にモーション制御命令の同じインスタンスの入力変数を変更し、再起動した場合について説明します。

再起動で変更可能な入力変数には、「Position」（目標位置）、「Distance」（移動距離）、「Velocity」（目標速度）、「Acceleration」（加速度）、「Deceleration」（減速度）、「Torque」（目標トルク）などがあります。

変更できない入力変数を変更して再起動すると命令で異常が発生します。

モーション制御命令の多重起動によりバッファリングされている命令に対して再起動すると、バッファ中の命令の入力変数が変更されます。

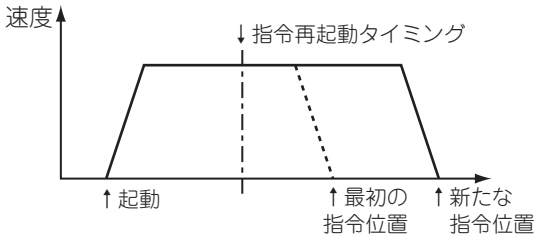
変更可能な入力変数の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の各命令を参照してください。

目標位置の変更

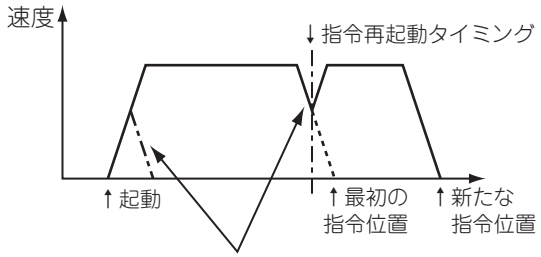
再起動で目標位置を変更する場合、変更した目標位置や変更のタイミングによって動作が異なります。目標位置の変更により動作方向が反転する場合の動作は、軸パラメータの［反転時動作］で［反転時減速停止］または［反転時即停止］から選択することができます。

● 新たな指令値により反転しない場合

等速動作中の再起動



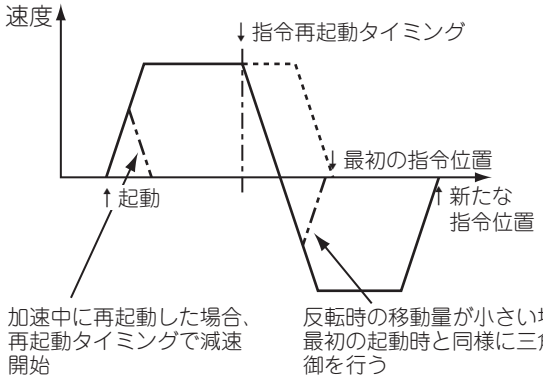
加減速動作中の再起動



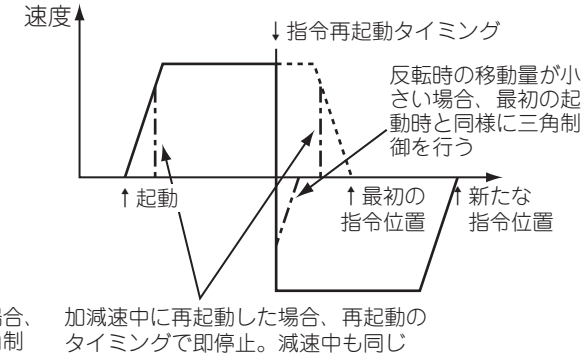
三角制御、もしくは減速中に再起動した場合は、
再度目標速度までの加速に移行します。
このとき、目標速度まで加速しない場合もあります。

● 新たな指令値により反転する場合

反転時減速停止設定



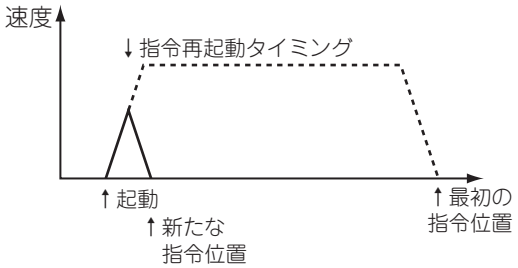
反転時即停止設定



● 三角制御のパターン

目標位置の変更により移動距離が短くなると、下図のような三角制御になる場合があります。

反転しない場合



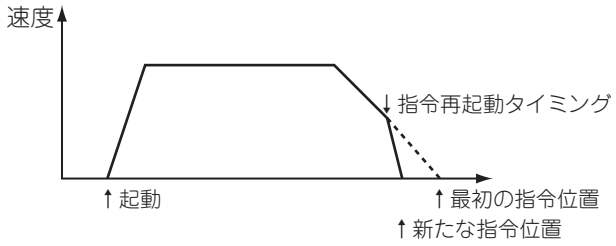
● 減速オーバーのパターン

次の場合は目標位置への停止を優先するため、指定した減速度を超えた減速度となります。

減速度が軸パラメータの［最大減速度］を超過する場合は、軸パラメータ設定の［加減速オーバー］で設定した動作となります。

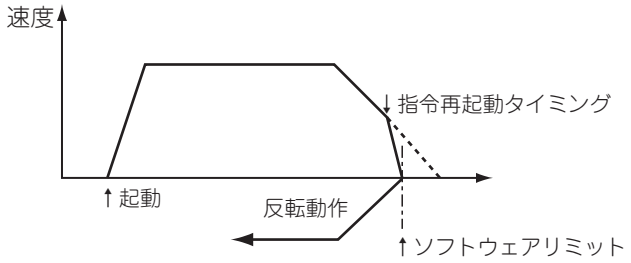
- 動作が反転しない場合で、指定の減速度だと目標位置を通り過ぎてしまう場合は、以下の動作となります。

反転しない場合



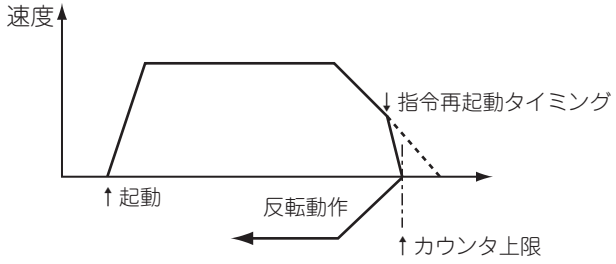
- 動作が反転する場合で、減速停止だとソフトウェアリミットをオーバーする場合は、以下の動作となります。

反転しない場合



- 動作が反転する場合で、減速停止だと指令現在位置がオーバーフロー、またはアンダフローする場合は、以下の動作となります。

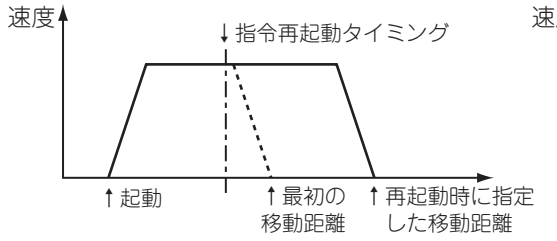
反転しない場合



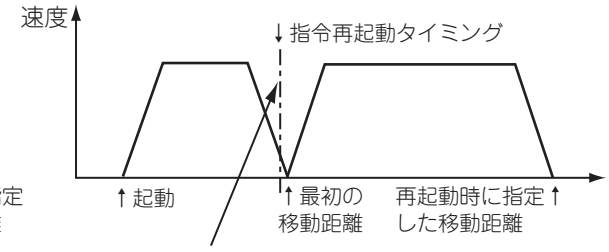
移動距離の変更

MC_MoveRelative（相対値位置決め）命令で、移動距離を変更して再起動を行った場合でも、移動を開始した位置を基準として変更した移動距離で位置決めします。
ただし、位置決め完了の直前に再起動すると、新たな命令の起動として移動する場合があります。

動作中の再起動



位置決め完了直前での再起動



位置決め完了直前で再起動すると、最初の移動距離で位置決めした位置を基準として、再起動時に指定した移動距離を移動する場合があります。

使用上の注意

移動距離を変更する再起動は、位置決め完了の直前に行わないでください。

目標速度の変更

目標速度については、加速中、等速中および三角制御での加速中のときに動作が変更されます。
減速中のときも変更を受け付けていますが、実際の動作には影響しません。

加速度の変更

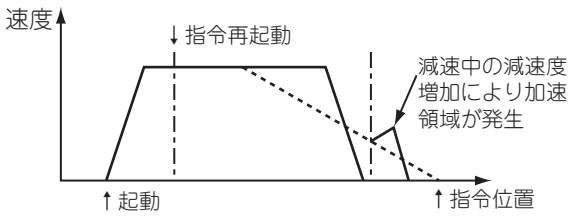
加速度については、加速中および三角制御での加速中のときに動作が変更されます。
等速中のときに変更すると、オーバーライドなどで加速する場合に適用されます。
減速中のときも変更を受け付けていますが、実際の動作には影響しません。

減速度の変更

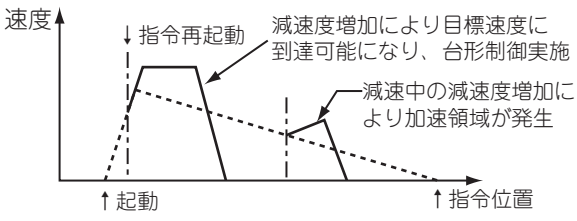
減速度については、加速中、等速中、減速中、三角制御中および減速オーバー制御中に変更されます。
変更された減速度だと目標位置を通り過ぎてしまう場合、目標位置への停止を優先するため、指定した減速度を超えた減速度となります。

● 減速度増加パターン

台形制御

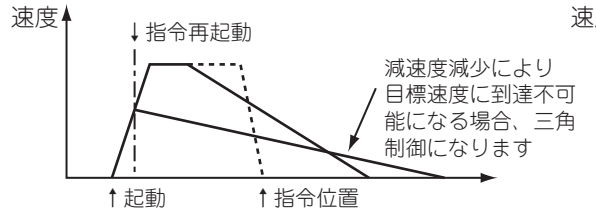


三角制御→台形制御

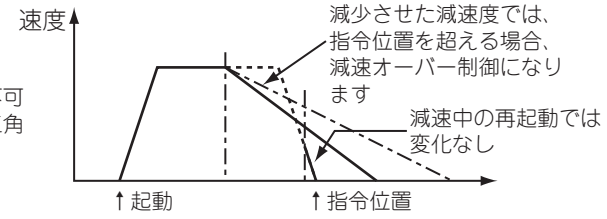


● **減速度減少パターン**

台形制御・三角制御



減速オーバー制御

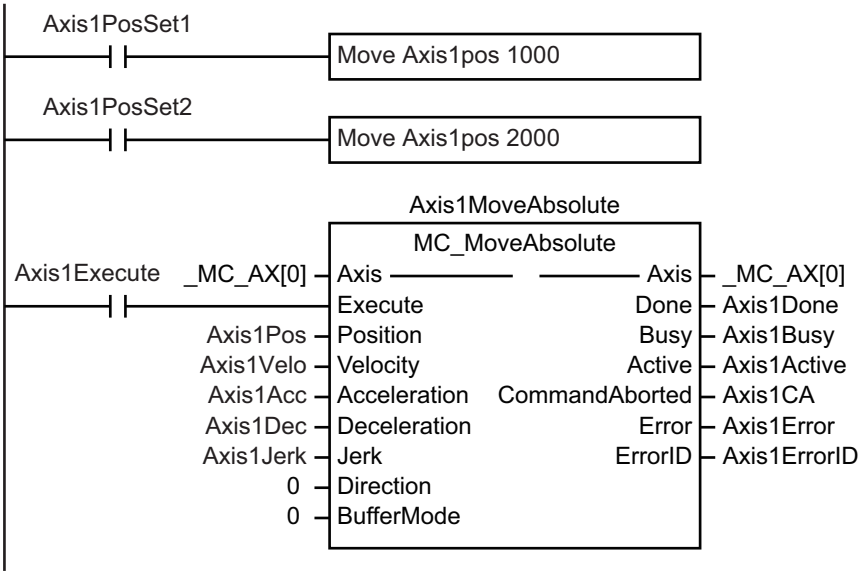


トルク指令の変更

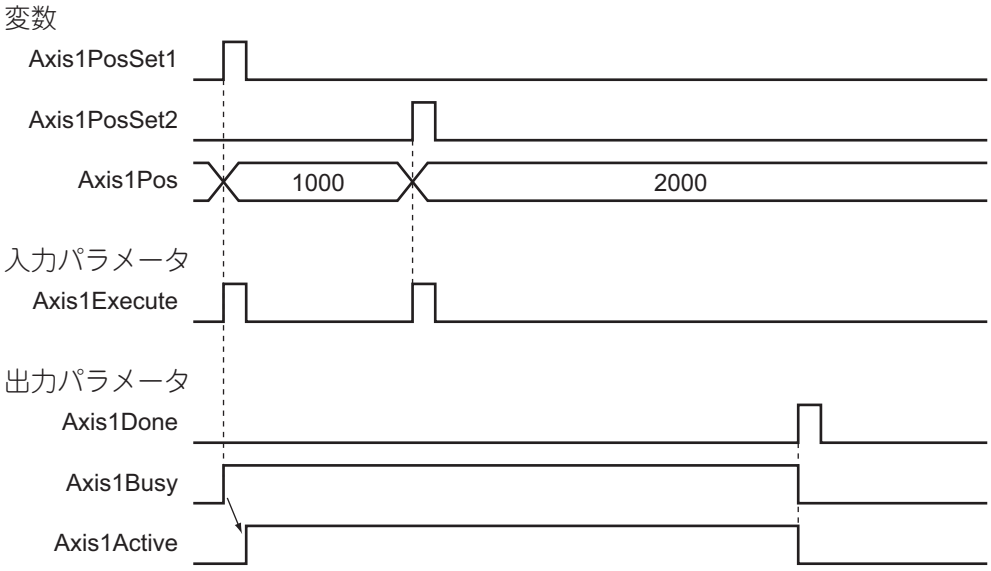
モーション制御命令を再起動したときに、トルクランプの指定に沿ってトルク指令値が変更されます。

再起動のプログラム例

絶対値位置決めで、目標位置を「1000」から「2000」に変更する例です。
目標位置への入力パラメータには、変数「Axis1Pos」を使用しています。
MOV 命令で目標位置を「1000」に設定し、Axis1Execute を FALSE→TRUE にすると位置決めを開始します。
動作中に目標位置を「2000」に設定し、Axis1Execute を再度 FALSE→TRUE にすると目標位置「2000」への位置決め動作に変更されます。



● **タイミングチャート**



使用上の注意

変更しない入力変数は、再起動する前の値を保持してください。

9-5-7　モーション制御命令の多重起動（バッファモード）

軸の動作中に別のモーション制御命令を起動することができます。

PLCopen®技術仕様書では、バッファモードとして定義されていますが、MC 機能モジュールでは「多重起動」として説明します。

多重起動を使用すると、複数のモーション制御命令の動作を停止することなく連続的に実行することができます。

MC 機能モジュールでは多重起動について、次の用語を使用しています。

用語		意味
本書	PLCopen®	
現命令	以前の FB	多重起動を行う直前に動作していたモーション制御命令
バッファリング命令	次の FB	軸の動作中に起動されて実行待ち状態のモーション制御命令
中継速度	ブレンディング（混合）	ブレンディング（混合）指定において、現命令で指定された目標位置まで動作する際に使用する指令速度のこと。

モーション制御命令の入力変数「BufferMode」（バッファモード選択）の指定により、以下の多重起動モードを選択できます。

これらのモードのおもな違いは、多重起動された命令の実行開始タイミングと中継速度です。

多重起動モード	動作説明
アボーティング（中断）	現命令が中断され、多重起動したモーション制御命令が実行されます。
バッファード（待機）	現命令の動作が正常終了したのち、バッファリング命令が実行されます。
ブレンディング（混合）	現命令の目標位置まで到達したのち、バッファリング命令が実行されます。 このモードでは、現命令とバッファ命令の間で停止しません。 現命令が目標位置に到達するときの中継速度を以下から選択できます。
ブレンディングロウ（低速度）	現命令とバッファリング命令の目標速度のうち低い方を中継速度とします。
ブレンディングプレビアス（前速度）	現命令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（次速度）	バッファリング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（高速度）	現命令とバッファリング命令の目標速度のうち高い方を中継速度とします。

多重起動された命令は、バッファード（待機）とブレンディング（混合）では、MC 機能モジュール内でバッファされ、指定した多重起動モードのタイミングと中継速度で実行されます。
このバッファは、軸ごとに 1 つあります。
アボーティング（中断）では、多重起動された命令が直ちに実行されるため、命令はバッファされません。

使用上の注意

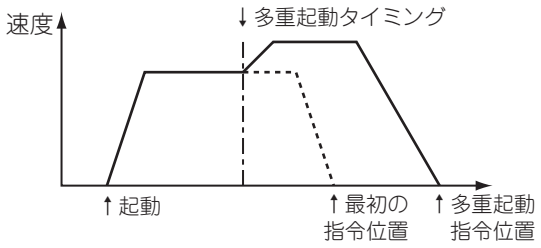
- 1 軸でバッファできる多重起動は 1 つだけです。2 つ以上の命令を多重起動すると、命令で異常が発生します。
- 単軸制御で動作している軸に対して、多軸協調制御命令（軸グループ命令）を多重起動することはできません。また、多軸協調制御（軸グループ命令）で動作している軸に対して、単軸制御命令（軸命令）を多重起動することはできません。これらの多重起動を行った場合、命令で異常が発生します。

アボーティング（中断）

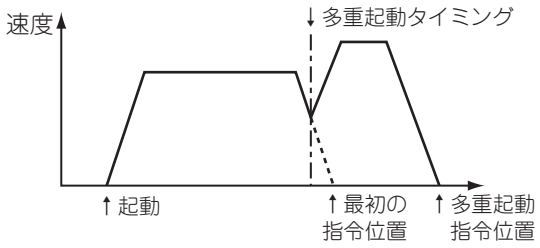
バッファを行わないデフォルトのモードです。
実行中の指令は直ちに中断され、多重起動された新しい命令が実行されます。
アボーティング（中断）は、単軸制御と同期制御のモーション制御命令で多重起動できます。

● 多重起動指令位置により反転しない場合

等速動作中の多重起動

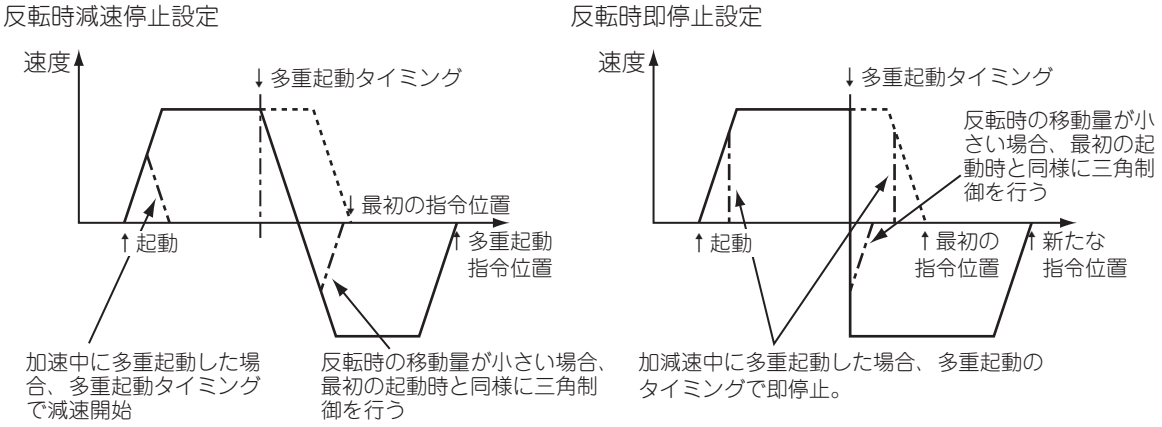


加減速動作中の多重起動



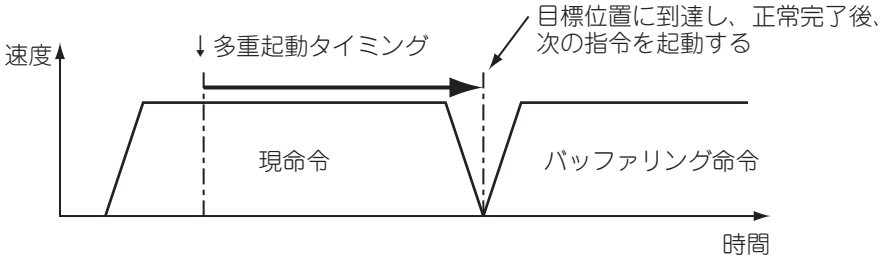
三角制御、もしくは減速中に多重起動した場合は、
バッファリング命令の目標速度までの加速に移行します。
このとき、目標速度まで加速しない場合もあります。

● 多重起動指令位置により反転する場合



バッファード（待機）

多重起動された命令は、実行中の現命令の動作が完了するまでバッファ（待機）されます。
現命令の動作が正常完了したのち、バッファリング命令が実行されます。



ブレンディング（混合）

多重起動された命令は、現命令の目標位置に到達するまでバッファ（待機）されます。
現命令の目標位置に到達したのちバッファ命令が実行されますが、このときに停止することはなく、入力変数「BufferMode」（バッファモード選択）で指定した速度で、2 つの動作が連続します。
相対値移動の場合、最終的な到達位置は 2 の命令の合計値となります。
絶対値移動では 2 つめの命令の目標位置です。
軸パラメータの［最大加速度］あるいは［最大減速度］だと目標位置を通り過ぎてしまう場合の動作は、軸パラメータの［加減速オーバー］で以下から設定できます。

- 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）
- 加減速度を急激にする
- 異常として停止する

使用上の注意

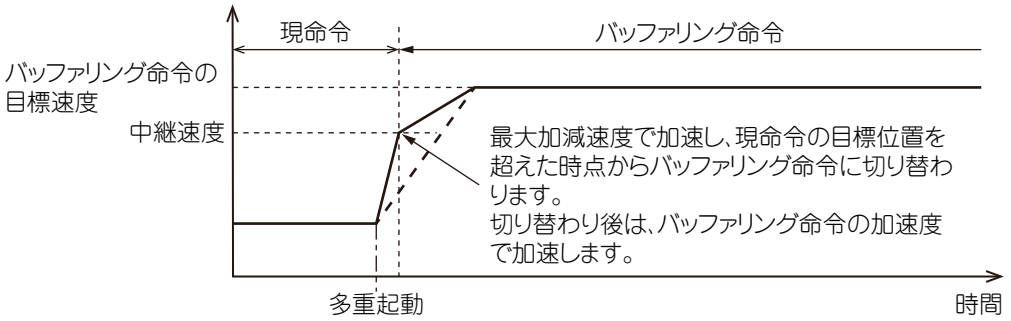
- ブレンディング（混合）は、単軸制御と同期制御の組み合わせでは実行できません。

バージョン関連情報

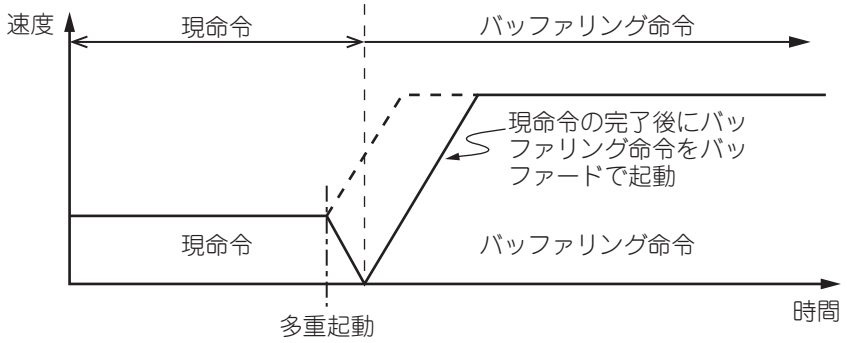
ユニットバージョン Ver.1.10 以降の CPU ユニットでは、[加減速度を急激にする（ブレンディングをバッファードに切り替える）] を設定しても、ブレンディングをバッファードに切り替えません。この場合、最大加減速度で加減速し、ブレンディングを継続します。
また、[異常として停止する] を設定しても、異常として停止しません。この場合も、最大加減速度で加減速し、ブレンディングを継続します。

● 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）

- ユニットバージョン Ver.1.10 以降の CPU ユニットの場合は、次の動作となります。
  また、[異常として停止する] に設定しても同様の動作となります。
  この例では、BufferMode をブレンディングネクストに設定しています。

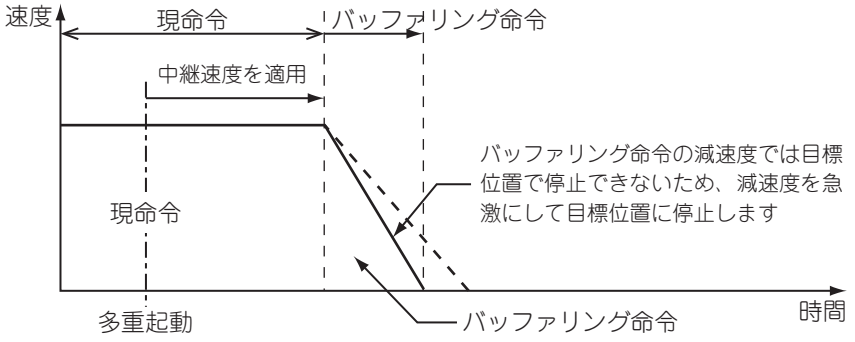


- ユニットバージョン Ver.1.09 以前の CPU ユニットの場合は、次の動作となります。

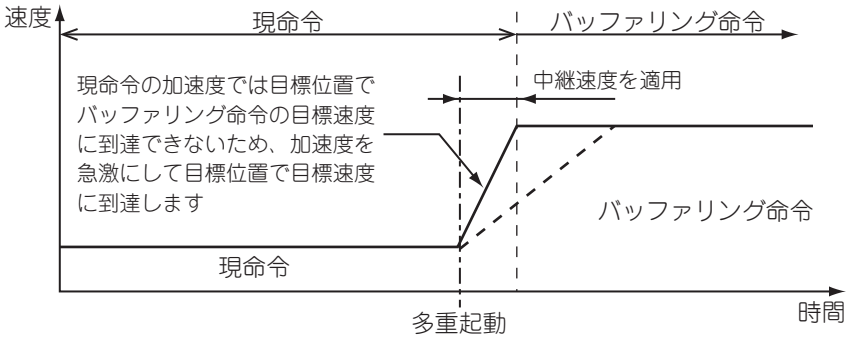


● 加減速度を急激にする

- 「BufferMode」（バッファモード選択）をブレンディングプレビアスに設定した場合は、次の動作となります。



- 「BufferMode」（バッファモード選択）をブレンディングネクストに設定した場合は、次の動作となります。



ブレンディングロウ（低速度）

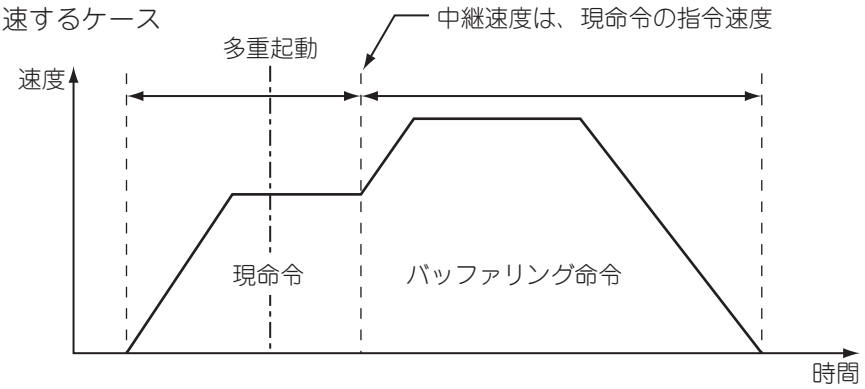
現命令の目標位置で、現命令とバッファリング命令の低い方の目標速度になるように動作します。

ブレンディングプレビアス（前速度）

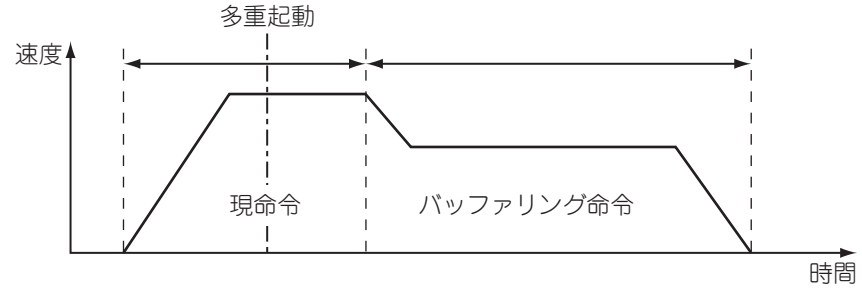
現命令の目標位置までは現命令の目標速度で動作します。
目標位置に到達したのちは、バッファリング命令の目標速度に加減速して動作します。

● 動作方向が反転しない場合

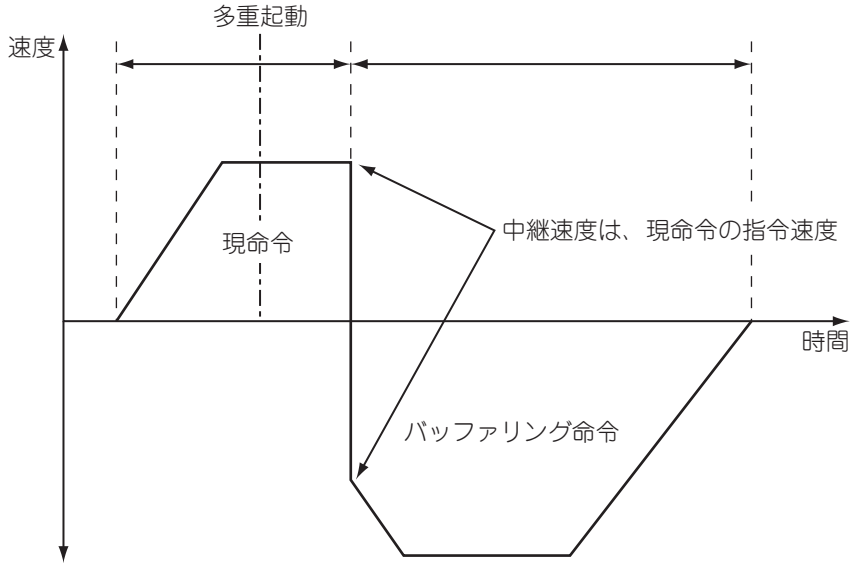
加速するケース



減速するケース



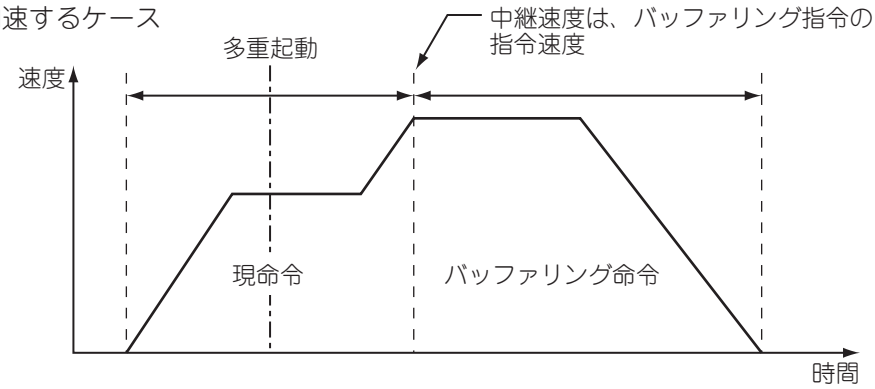
● **動作方向が反転する場合**



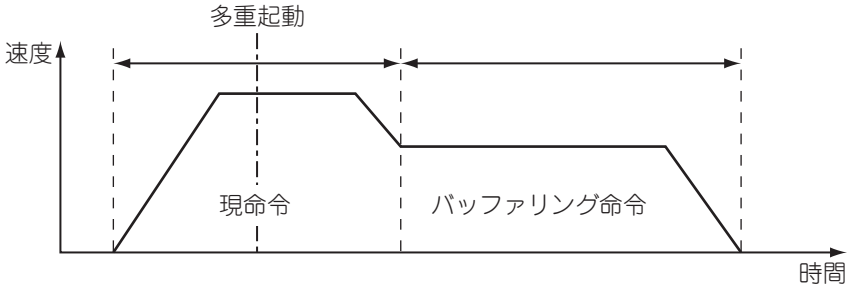
ブレンディングネクスト（次速度）

現命令の目標位置でバッファリング命令の目標速度になるように動作します。

加速するケース



減速するケース



ブレンディングハイ（高速度）

現命令の目標位置で、現命令とバッファリング命令の高い方の目標速度になるように動作します。

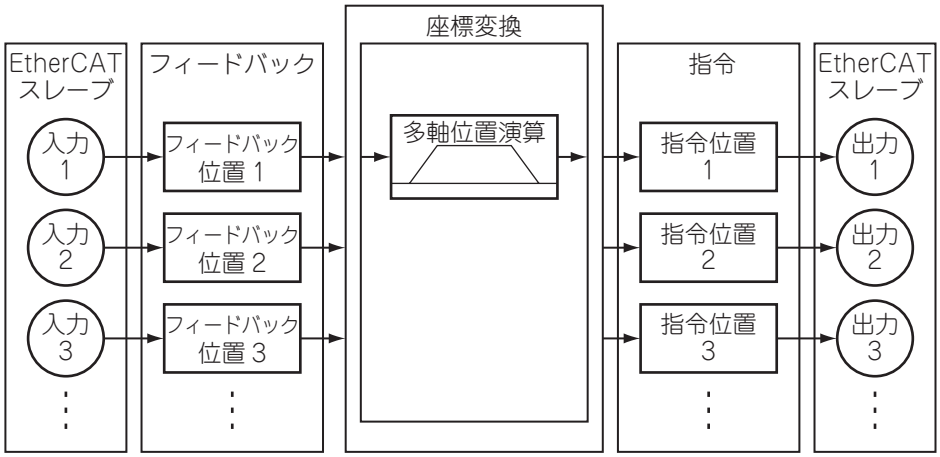
9-6 多軸協調制御

ここでは、多軸の協調制御動作について説明します。
MC 機能モジュールでは、あらかじめ Sysmac Studio で軸グループの設定を行っておくことで、多軸での補間制御を行うことができます。

9-6-1 動作概要

多軸協調制御とは、工具先端などの制御対象の軌跡を制御するために、関連する複数の軸をグループ化して協調動作させる機能です。
MC 機能モジュールでは、協調動作を行う軸の組み合わせを軸グループとして Sysmac Studio により設定します。
ユーザプログラムからは各軸をサーボ ON したのち、実際に多軸協調制御を行う軸グループを有効にします。
多軸協調制御は軸グループに属する軸の協調動作が目的なので、有効な軸グループの軸に対して、単軸動作のモーション制御命令を起動することはできません。また、軸グループに属する軸で異常が発生した場合は、軸グループに属する他の軸は軸グループパラメータの［軸グループ停止方法］の設定で停止します。

MC 機能モジュールでは、2 軸～4 軸までの直線補間と 2 軸の円弧補間を行うことができます。



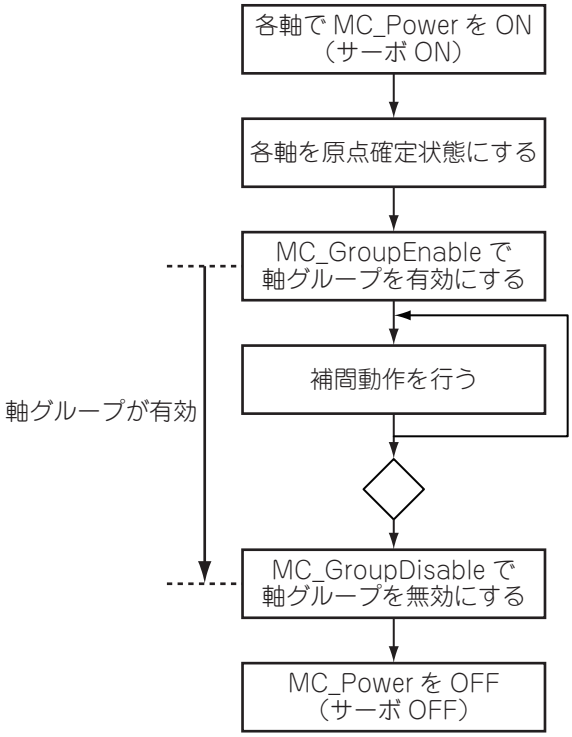
参考

動作中に軸の組み合わせを変更して補間制御を行う必要がある装置では、あらかじめ組み合わせを変更する軸を含む複数の軸グループを Sysmac Studio で作成しておきます。その上で、動作中にユーザプログラムで有効にする軸グループを指定することにより、実現することができます。
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）を使用することで、無効にしている軸グループの構成軸を書き換えることができます。

軸グループの詳細については、「3-3 軸グループ (P.3-21)」を参照してください。

軸グループ有効／無効

軸グループを有効にするには、MC_GroupEnable（軸グループ有効）命令で有効にする軸グループを指定します。無効の状態で軸グループ命令を起動すると命令で異常が発生し起動できません。
有効状態の軸グループを無効にするには、MC_GroupDisable（軸グループ無効）命令で無効にする軸グループを指定します。
軸グループの動作中に軸グループ無効命令を起動すると軸グループに属する軸は、軸パラメータ設定の最大減速度で減速停止します。



軸グループ有効／無効機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_GroupEnable（軸グループ有効）命令、および MC_GroupDisable（軸グループ無効）命令を参照してください。

軸グループ構成軸書込

MC_ChangeAxesInGroup（軸グループ構成軸書込）命令により、無効にしている軸グループの構成軸を一時的に書き換える機能です。軸グループが有効な場合は、MC_GroupDisable（軸グループ無効）命令で無効にしてから構成軸を書き換えてください。
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することができます。

使用上の注意

MC_ChangeAxesInGroup（軸グループ構成軸書込）による変更は CPU ユニット内蔵の不揮発性メモリには保存されません。電源の再投入または Sysmac Studio からのダウンロードにより、不揮発性メモリのパラメータ設定内容に戻ります。

軸グループ構成軸書込の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_ChangeAxesInGroup（軸グループ構成軸書込）命令を参照してください。

軸グループ位置取得

MC_GroupReadPosition（軸グループ位置取得）命令により、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する機能です。
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することができます。

軸グループ位置取得機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_GroupReadPosition（軸グループ位置取得）命令を参照してください。

軸グループエラーリセット

軸グループで異常が発生した場合、異常の原因を取り除いたのちに MC_GroupReset（軸グループエラーリセット）命令で、異常を解除します。

軸グループエラーリセットの詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_GroupReset（軸グループエラーリセット）命令を参照してください。

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

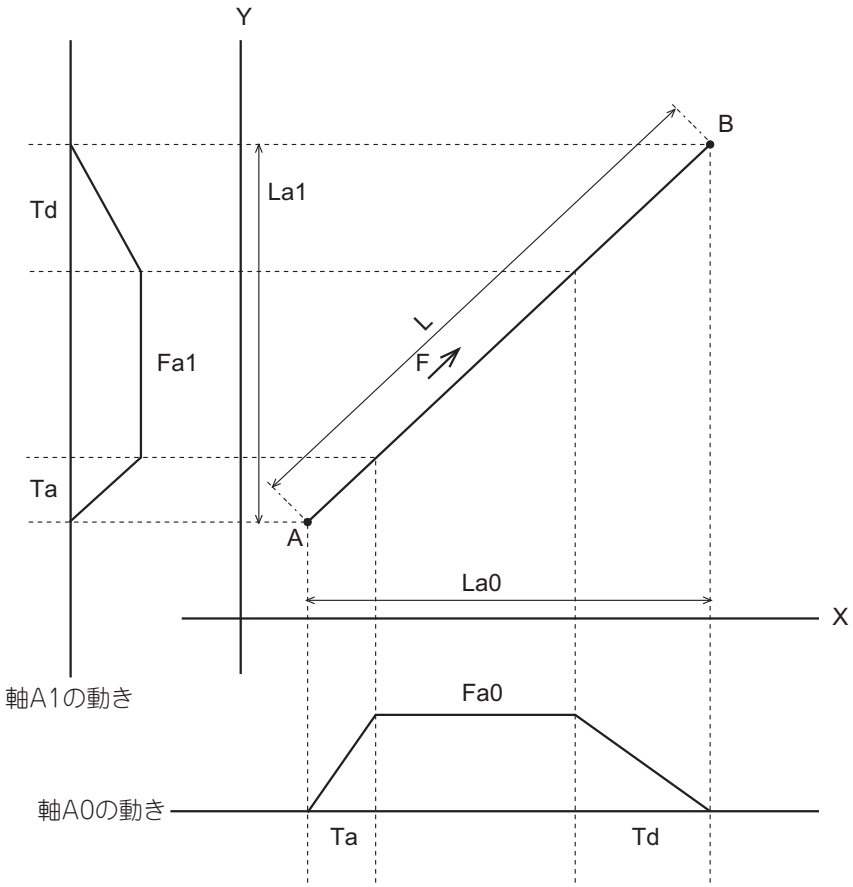
9-6-2　直線補間

直線補間は、軸グループの軸 A0～軸 A3 の論理軸のうち、2 軸～4 軸を使用して始点と終点を直線で位置決めします。
絶対値での位置決めと相対値での位置決めが可能で、補間速度、補間加速度、補間減速度およびジャークを指定することができます。

MC 機能モジュールでは、以下の 3 種類の直線補間命令があります。

- MC_MoveLinear（直線補間）
  入力変数「MoveMode」（移動方法選択）の指定により、絶対値での直線補間と相対値での直線補間を選択できます。MC 機能モジュール独自の命令です。
- MC_MoveLinearAbsolute（絶対値直線補間）
  絶対値での直線補間を行います。PLCopen®技術仕様書で定義されている命令です。
- MC_MoveLinearRelative（相対値直線補間）
  相対値での直線補間を行います。PLCopen®技術仕様書で定義されている命令です。

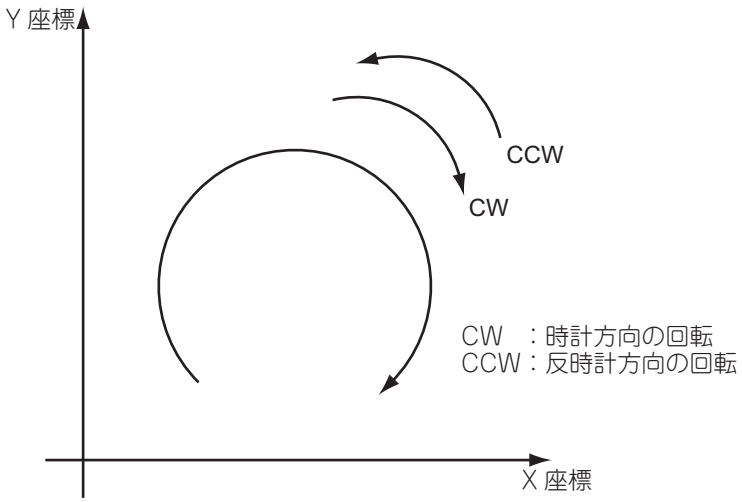
2 軸を使用した A 点から B 点までの直線補間は、次の図のようになります。



直線補間の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_MoveLinear（直線補間）命令、MC_MoveLinearAbsolute（絶対値直線補間）命令、および MC_MoveLinearRelative（相対値直線補間）命令を参照してください。

9-6-3 円弧補間

円弧補間は、軸グループの軸 A0～軸 A3 の論理軸のうち、2 軸を使用して 2 次元平面上で円弧を描いて位置決めします。
絶対値での位置決めと相対値での位置決めが可能で、円弧補間モード、経路の方向、補間速度、補間加速度、補間減速度および 2 軸の合成値のジャークを指定することができます。



MC 機能モジュールでは、入力変数「CircMode」（円弧補間モード）により、以下の 3 種類から円弧補間の方式を指定することができます。
- 通過点指定
- 中心点指定
- 半径指定

使用上の注意

円弧補間で使用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、命令で異常が発生します。

9-6-4　軸グループサイクリック同期位置制御

この機能は軸グループの各軸に指定した目標位置を周期的に出力します。
任意の軌跡で動作させたい場合などに、ユーザプログラムで演算した目標位置を絶対位置で指定します。
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することができます。

軸グループサイクリック同期位置制御機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_GroupSyncMoveAbsolute（軸グループサイクリック同期絶対位置制御）命令を参照してください。

9-6-5　多軸協調制御での停止

軸グループの動作を停止する機能には、ユーザプログラムのモーション制御命令による停止機能と、異常の発生などによる停止があります。

モーション制御命令による停止

軸グループ動作の動作停止には、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。

機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_GroupStop（軸グループ強制停止）命令、MC_GroupImmediateStop（軸グループ即停止）命令を参照してください。

- **MC_GroupStop（軸グループ強制停止）命令**

  軸グループでの直線補間や円弧補間に対して減速度とジャークを指定し軌跡上で減速停止させることができます。
  減速度に「0」を指定すると即時に停止する指令をサーボドライバなどへ送信します。
  本命令による減速停止中と入力変数「Execute」（起動）が TRUE の間は、ほかの動作指令を受け付けません。

- **MC_GroupImmediateStop（軸グループ即停止）命令**

  軸グループのすべての軸を即停止させることができます。即停止の方法は各軸の軸パラメータ［即停止入力停止方法］に従います。
  MC_GroupStop（軸グループ強制停止）命令による減速停止中状態の軸グループに対しても実行可能です。

異常の発生などによる停止

● 軸グループ動作中の異常発生による停止

軸グループ動作中に軸グループのいずれかの構成軸で減速停止となる異常が発生した場合、補間の軌跡上で補間減速停止します。このときの補間減速度は、制御中の命令の減速度の指定に従います。
また、いずれかの構成軸で即停止となる異常が発生した場合、軸グループのそのほかの構成軸は、軸グループパラメータ設定の［軸グループ停止方法］の設定で停止します。

軸グループの停止方法は、下記から選択することができます。
- 即停止
- 各軸を減速停止（各軸の最大減速度）
- 即停止し、同時にサーボ OFF を実行

● モーション制御周期オーバーの発生による停止

モーション制御の処理が 2 周期以内で完了せずモーション制御周期オーバーが発生したときは、すべての軸が即停止します。

使用上の注意

形 NX701 CPU ユニットを使用してマルチモーションで動作させている場合に、いずれか一方のタスクでモーション制御周期オーバーが発生したときは、両タスクのすべての軸が即停止します。
マルチモーションは、「A-6-2 モーション制御機能 (P.A-29)」を参照してください。

● MC 試運転機能の開始による停止

Sysmac Studio の MC 試運転機能を開始したとき、各軸は［最大減速度］で減速停止します。

● CPU ユニットのモード切り替えによる停止

CPU ユニットのモードを切り替えたとき、各軸は［最大減速度］で減速停止します。

参考

- 軸グループでの動作中に、MC_GroupDisable（軸グループ無効）命令を起動すると各軸は最大減速度で減速停止します。
- 軸グループでの動作中に、MC_Stop（強制停止）命令を起動すると、軸と軸グループは異常となり、軸グループ動作は補間減速停止します。このときの補間減速度は、制御中の命令の減速度の指定に従います。
- 軸グループ動作中に MC_Power（運転可）命令の入力変数「Enable」（有効）を FALSE にすると、MC 機能モジュールは、その軸の指令値を即停止しサーボ OFF を実行します。
  サーボ OFF 時のサーボドライバなどの動作は、サーボドライバなどの設定に従います。
  軸グループのそのほかの軸は、軸グループパラメータ設定の［軸グループ停止方法］の設定で停止します。軸グループは異常になります。
- 運転モードからプログラムモードへ切り替えたときは、動作中のモーション制御命令は中断されます。このとき、命令の出力変数「CommandAborted」（実行中断）は FALSE のままです。プログラムモードへ切り替わり後もサーボ ON/OFF 状態を維持しています。
- 運転モードからプログラムモードへ切り替えて減速停止しているときに、運転モードに戻すと、モーション制御命令の出力変数「CommandAborted」（実行中断）が TRUE となります。
- MC_SaveCamTable（カムテーブル保存）命令で保存中の場合、保存処理は継続されます。
- MC_GenerateCamTable（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、生成処理は継続されます。

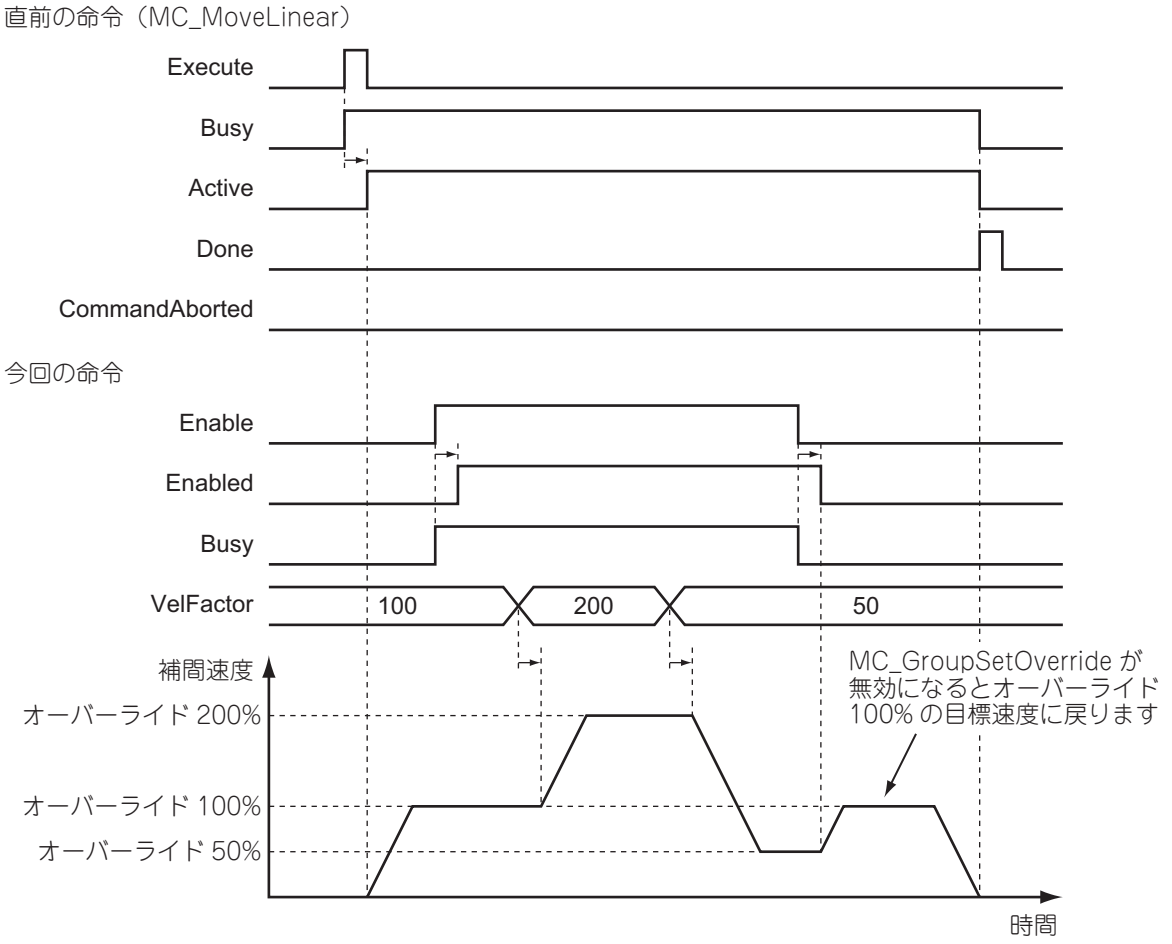
9-6-6　多軸協調制御でのオーバーライド

軸グループでの多軸協調制御では、MC_GroupSetOverride（軸グループオーバーライド値設定）命令により、起動中の補間動作に対するオーバーライド値を設定できます。
オーバーライド値は、補間の目標速度に対する比率で、0～500%を設定できます。
補間の目標速度に対して 0%を設定すると、速度「0」で停止したまま動作状態を継続します。

オーバーライドが有効の間、設定したオーバーライド値が取り込まれます。オーバーライドを無効にすると、オーバーライド値は 100%に戻ります。
オーバーライド値で変更後の値が、最高補間速度をオーバーする場合は、軸グループパラメータの［最高補間速度］になります。

● MC_MoveLinear（直線補間）命令へのオーバーライド

MC_MoveLinear（直線補間）命令に、オーバーライド命令を使用した場合のタイミングチャートの例を下記に示します。



機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編（SBCE-434）』の MC_GroupSetOverride（軸グループオーバーライド値設定）命令を参照してください。

9-7 多軸協調制御の共通機能

ここでは、多軸協調制御の共通機能について説明します。

9-7-1 多軸協調制御での速度

多軸協調制御では、軌跡上での補間速度で速度を指定します。
単位は単軸の場合と同じ「指令単位／s」を使用します。

速度の種類

MC 機能モジュールで扱う軸グループの補間速度は、以下の 1 種類です。

速度種別	定義
指令補間速度	MC 機能モジュールが軸グループを制御するために出力している指令補間速度の現在値です。

速度に関する軸パラメータ

パラメータ名称	機能	設定範囲	初期値
最高補間速度	軌跡の最高補間速度を設定します。 「0」を設定すると補間速度の制限はなくなります。 軸グループ動作命令で最高補間速度を超える補間の目標速度が指定された場合は、最高補間速度で動作します。	倍精度実数型の正数、0	800,000,000
補間速度警告値	補間速度警告を出力するための最高補間速度に対する割合を設定します。 「0」を設定すると補間速度警告は出力しません。 （単位: %）	0～100	0

軸動作の目標速度指定

実際の位置決め動作の補間速度は、モーション制御命令の入力変数「Velocity」（目標速度）で指定します。

速度のモニタ

ユーザプログラムから、軸グループ変数を読み取ることで補間速度をモニタできます。
なお、変数名は、_MC_GRP[*]を例に説明していますが、_MC1_GRP[*]、_MC2_GRP[*]についても同様です。

変数名	データ型	名称	機能
_MC_GRP[0-63].Cmd.Vel	LREAL	指令補間速度	指令補間速度の現在値です。 正方向への移動はプラス符号、負方向への移動はマイナス符号が付きます。

9-7-2 多軸協調制御での加速と減速

多軸協調制御では、軌跡上での補間加速度と補間減速度で制御します。

単位は単軸の場合と同じ「指令単位／s^2」を使用します。

補間加速度と補間減速度に関する軸パラメータ

パラメータ名称	機能	設定範囲	初期値
最大補間加速度	軌跡の最大補間加速度を設定します。 「0」を設定すると補間加速度の制限はなくなります。 （単位: 指令単位／s^2）	倍精度実数型の正数、0	0
最大補間減速度	軌跡の最大補間減速度を設定します。 「0」を設定すると補間減速度の制限はなくなります。 （単位: 指令単位／s^2）	倍精度実数型の正数、0	0
補間加減速オーバー	軸グループの加減速制御で、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補間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	0～2	0
補間加速度警告値	補間加速度警告を出力するための最高補間加速度に対する割合を設定します。 「0」を設定すると補間加速度警告は出力しません。 （単位: %）	0～100	0
補間減速度警告値	補間減速度警告を出力するための最高補間減速度に対する割合を設定します。 「0」を設定すると補間減速度警告は出力しません。 （単位: %）	0～100	0

*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、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44）」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、「9-5-7 モーション制御命令の多重起動（バッファモード）（P.9-44）」を参照してください。

軸グループ動作の補間加速度と補間減速度の指定

実際の位置決め動作の補間加速度および補間減速度は、モーション制御命令の入力変数「Acceleration」（加速度）および「Deceleration」（減速度）で指定します。

補間加速度と補間減速度のモニタ

ユーザプログラムから、軸グループ変数を読み取ることで補間加速度および補間減速度をモニタできます。

なお、変数名は、_MC_GRP[*]を例に説明していますが、_MC1_GRP[*]、_MC2_GRP[*]についても同様です。

変数名	データ型	名称	機能
_MC_GRP[0-63].Cmd.AccDec	LREAL	指令補間加減速度	指令補間加減速度の現在値です。 加速時はプラス符号、減速時はマイナス符号が付きます。

9-7-3　多軸協調制御でのジャーク

多軸協調制御での Jerk（ジャーク）は、補間軌跡上での補間加減速を S 字型になめらかにして機械の衝撃や振動を抑制する機能です。

単位は単軸の場合と同じ「指令単位／s^3」を使用します。

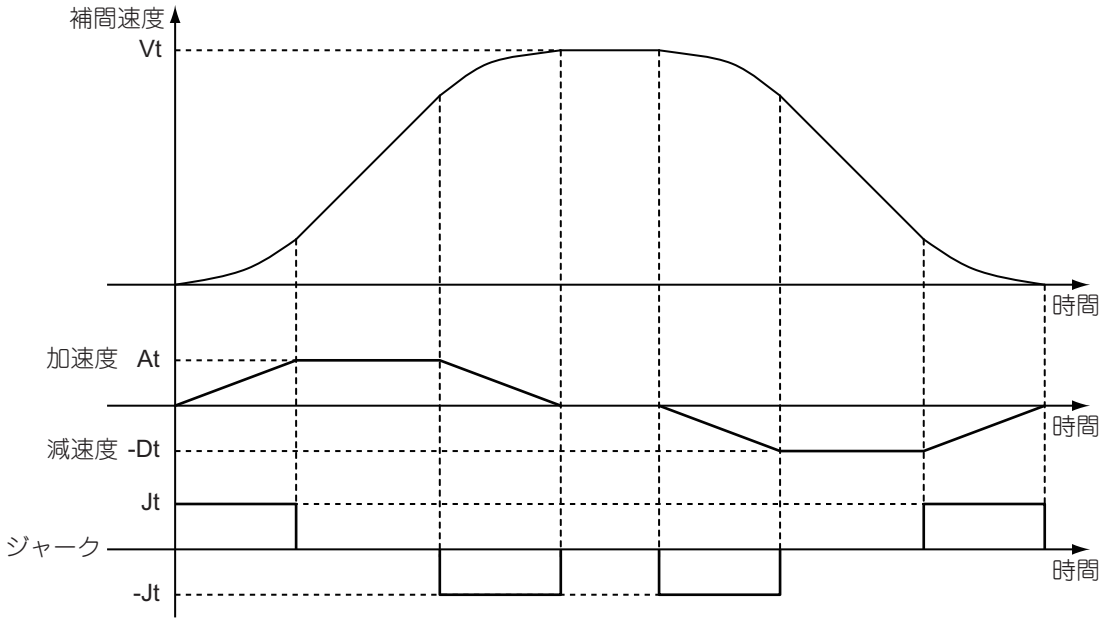
軸グループ動作のジャークの指定

実際の補間動作のジャークは、モーション制御命令の入力変数「Jerk」（ジャーク）で指定します。

Jerk（ジャーク）が「0」以外の場合の例

ジャークが適用される区間の加減速度は一定の比率で変化するため、指令補間速度は S 字型になめらかになります。

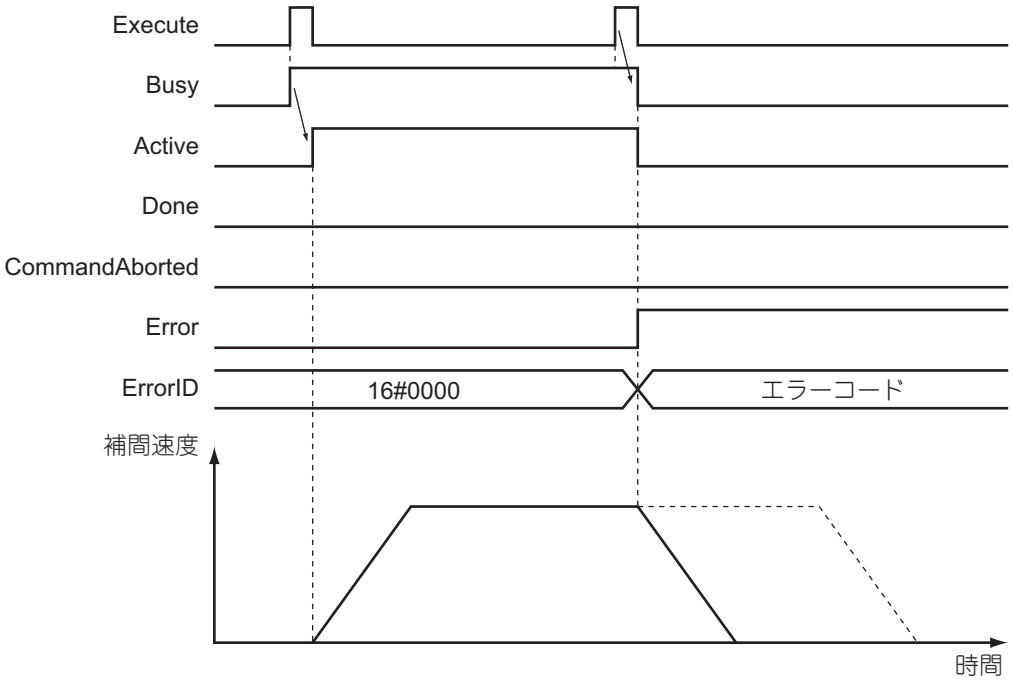
ジャークが「0」の区間の補間加速度は固定のため、指令補間速度は直線状になります。



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

9-7-4　多軸協調制御のモーション制御命令の再起動

直線補間または円弧補間命令の再起動を行った場合、命令で異常が発生します。



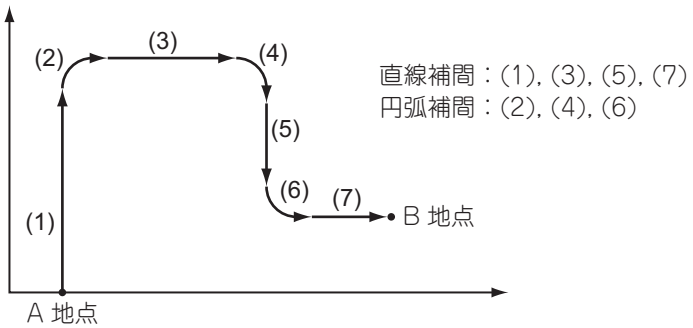
MC_GroupStop（軸グループ強制停止）命令の再起動では、減速度を変更することはできますが、ジャークを変更することはできません。

MC_GroupReset（軸グループエラーリセット）命令の再起動を行った場合、再起動の指令は無視されて異常解除処理を継続します。

各モーション制御命令での再起動の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の各命令を参照してください。

9-7-5 多軸協調制御のモーション制御命令の多重起動（バッファモード）

軸動作と同様に軸グループでの多軸協調制御を多重起動することができます。
多軸協調制御で多重起動を使用すると、複数の直線や円弧が連続する軌跡制御を実行することができます。



モーション制御命令の入力変数「BufferMode」（バッファモード選択）の指定により、単軸のときとおなじ多重起動モードを選択できます。
軸グループの命令バッファは、軸グループごとに動作中のものが 1 つと多重起動用の 7 つで、合計 8 つあります。
軸動作の命令と軸グループ動作の命令では、互いに多重起動することはできません。

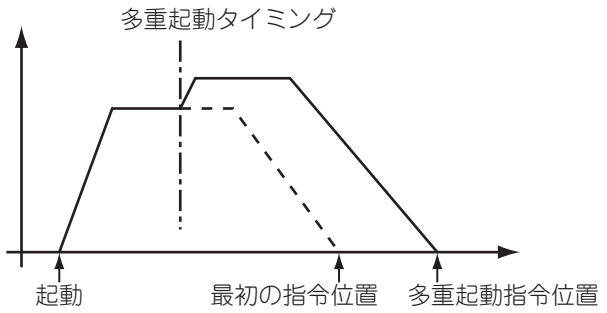
使用上の注意

- 1 つの軸グループでバッファできる多重起動は 7 つまです。8 つ以上の命令を多重起動すると、命令で異常が発生します。
- 単軸制御で動作している軸に対して、多軸協調制御命令（軸グループ命令）を多重起動することはできません。また、多軸協調制御（軸グループ命令）で動作している軸に対して、単軸制御命令（軸命令）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った場合、命令で異常が発生します。

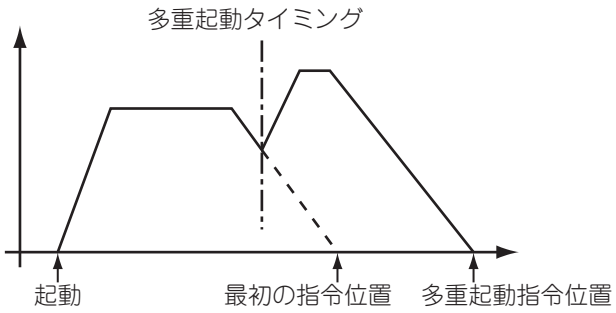
アボーティング（中断）

バッファを行わないデフォルトのモードです。
実行中の命令は直ちに中断され、多重起動された新しい命令が実行されます。
入力変数「BufferMode」（バッファモード選択）がないモーション制御命令の多重起動は、アボーティング（中断）として動作します。
多重起動された時点の補間速度を起点として多重起動された命令の動作が開始されます。
アボーティング（中断）では、同期制御を含む単軸制御と、軸グループ制御の組み合わせでは実行できません。
単軸動作中の軸に軸グループ動作を起動すると多重起動時に命令で異常が発生します。
軸グループ動作中に単軸動作を起動すると軸と軸グループの両方で異常となり停止します。

等速動作中の多重起動

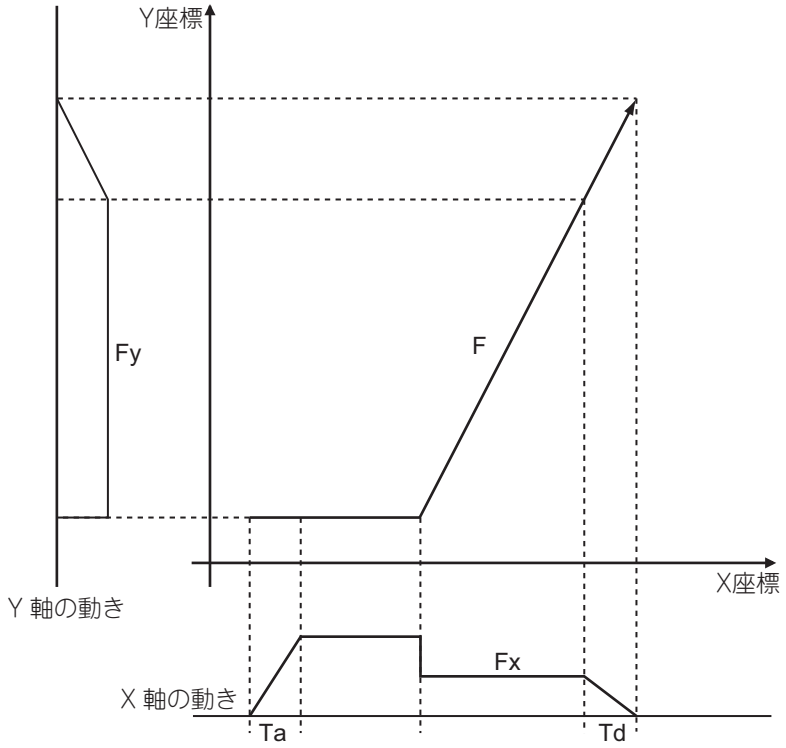


加減速動作中の多重起動



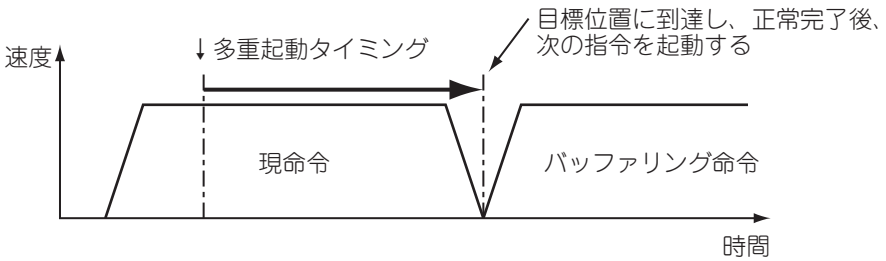
軸グループでの多重起動では、補間速度が連続するように行われます。移動量が「0」となる軸を含む命令と連続する場合、各軸の速度変化は不連続となります。

● 2 軸直交座標での補間速度と各軸の速度の例



バッファード（待機）

多重起動された命令は、実行中の動作が完了するまでバッファ（待機）されます。
現命令の動作が正常完了したのち、バッファリング命令が実行されます。



ブレンディング（混合）

軸グループでのブレンディング（混合）も、単軸動作のブレンディング（混合）と同様の動作となります。
多重起動された命令は、現命令の目標位置に到達するまでバッファ（待機）されます。
現命令の目標位置に到達したのちに、バッファリング命令が実行されます。このときに到達位置で停止することなく、入力変数「BufferMode」（バッファモード選択）で指定した補間速度で 2 つの動作が連続します。
バッファ命令の加速度、あるいは減速度だと目標位置を通り過ぎてしまう場合の動作は、軸グループパラメータの［補間加減速オーバー］で以下の中から選択できます。

- 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）
- 加減速度を急激にする

・異常として停止する

バージョン関連情報

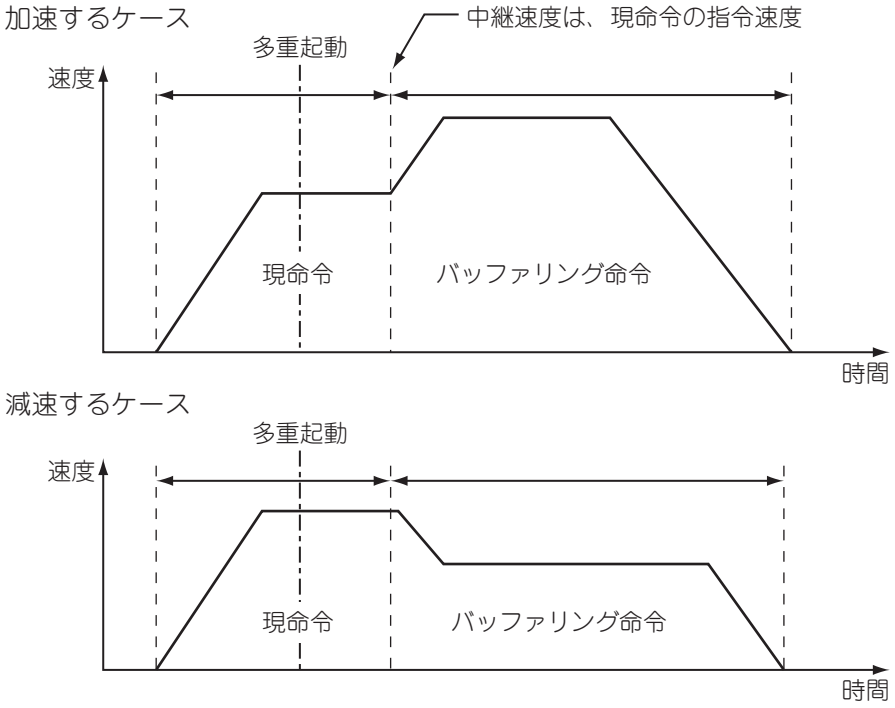
- ユニットバージョン Ver.1.10 以降の CPU ユニットでは、[加減速度を急激にする（ブレンディングをバッファードに切り替える）] を設定しても、ブレンディングをバッファードに切り替えません。この場合、最大加減速度で加減速し、ブレンディングを継続します。
  また、[異常として停止する] を設定しても、異常として停止しません。この場合も、最大加減速度で加減速し、ブレンディングを継続します。
- ユニットバージョン Ver.1.09 以前の CPU ユニットを使用する場合は、次の点に注意してください。
  多軸協調制御でのブレンディング（混合）で、プロファイル演算の結果、現命令の実行時間が 4 制御周期未満となった場合は、バッファード（待機）の動作となります。このとき、監視情報の「ブレンディングの中継速度到達までの移動量不足通知」が発生します。
  ユニットバージョン Ver.1.10 以降の CPU ユニットを使用する場合は、この制約はありません。

● ブレンディングロウ（低速度）

現命令の目標位置で、現命令とバッファリング命令の低い方の目標速度になるように動作します。

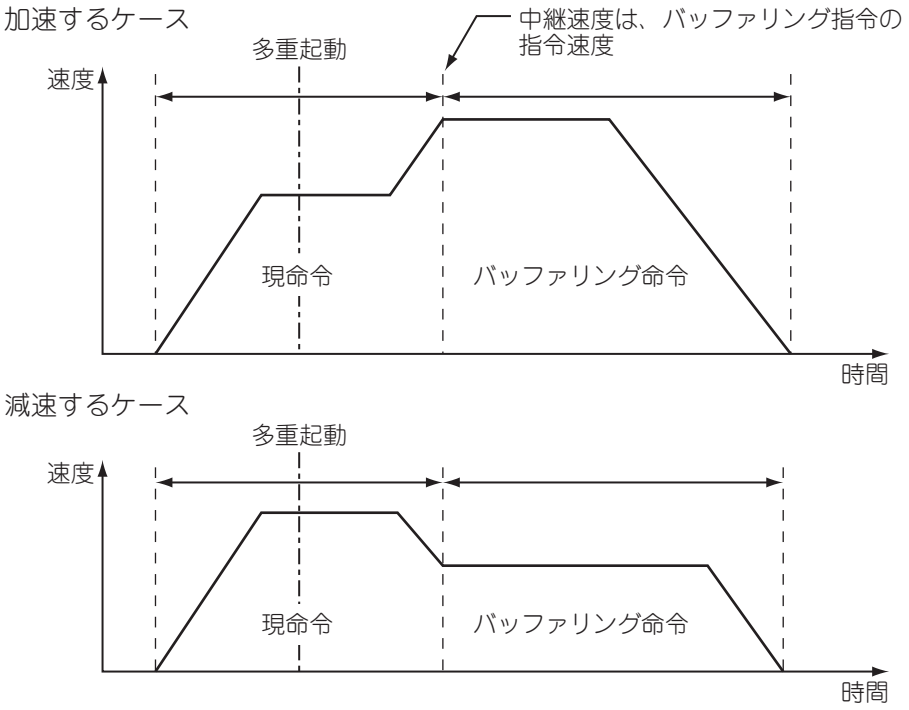
● ブレンディングプレビアス（前速度）

現命令の目標位置までは現命令の目標速度で動作します。目標位置に到達したのちは、バッファリング命令の目標速度に加減速して動作します。



● ブレンディングネクスト（次速度）

現命令の目標位置でバッファリング命令の目標速度になるように動作します。



● **ブレンディングハイ（高速度）**

現命令の目標位置で、現命令とバッファリング命令の高い方の目標速度になるように動作します。

トランジションモード

軸グループの多重起動では、補間軌跡の向きが変わることで装置やワークに衝撃が発生する場合があります。この衝撃を緩和する手段として、命令間の連結動作方法を、モーション制御命令の入力変数「TransitionMode」（トランジションモード）で指定できます。

MC 機能モジュールで選択できるトランジションモードは以下のようになります。

No.	トランジションモード	説明
0	トランジション無効 （_mcTMNone）	トランジションのための処理を行いません（デフォルト）。 衝撃は緩和されませんが動作時間は短くなります。
10	コーナースーパーインポーズ （_mcTMCornerSuperimposed）	現命令の減速とバッファ命令の加速を重畳します。 補間軌跡の線速度を一定にすることができます。

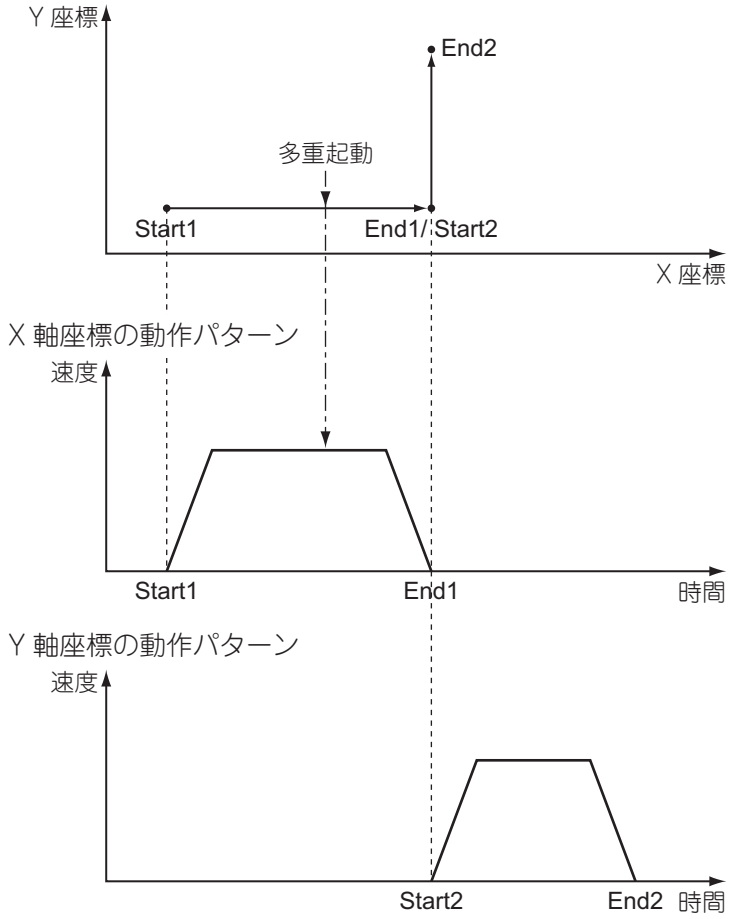
参考

PLCopen®の技術仕様書では、No.0～9 が定義されています。No.10 は MC 機能モジュール独自の仕様です。

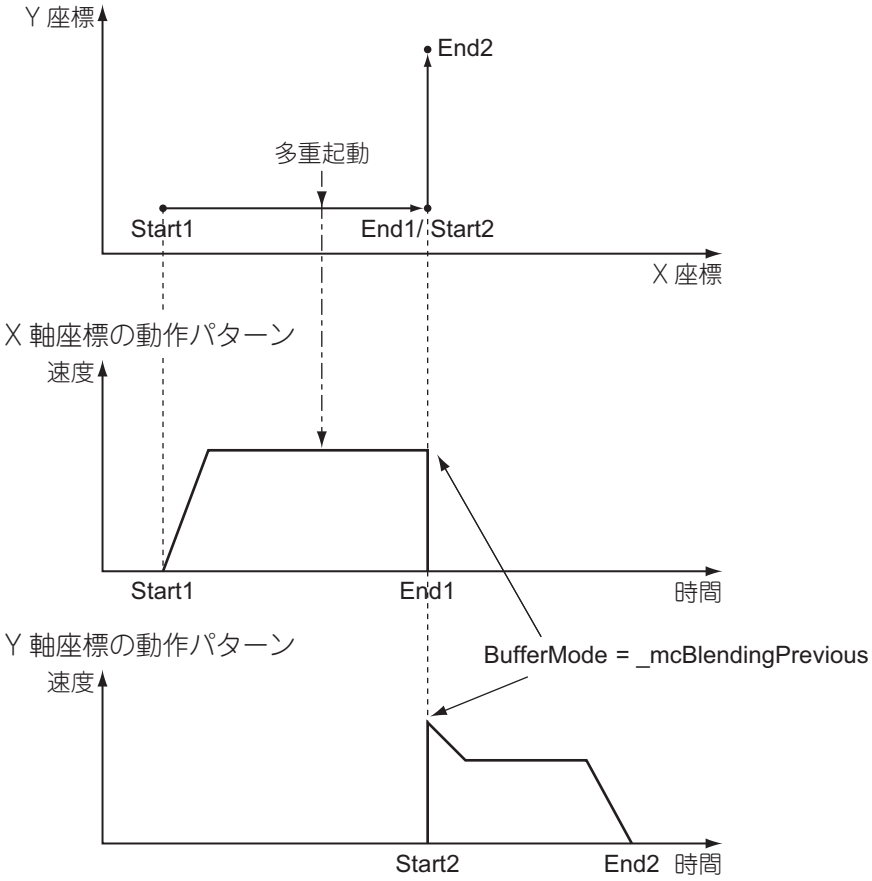
● **トランジション無効（0: _mcTMNone）の場合**

2 つの位置をつなぐための処理は行われません。

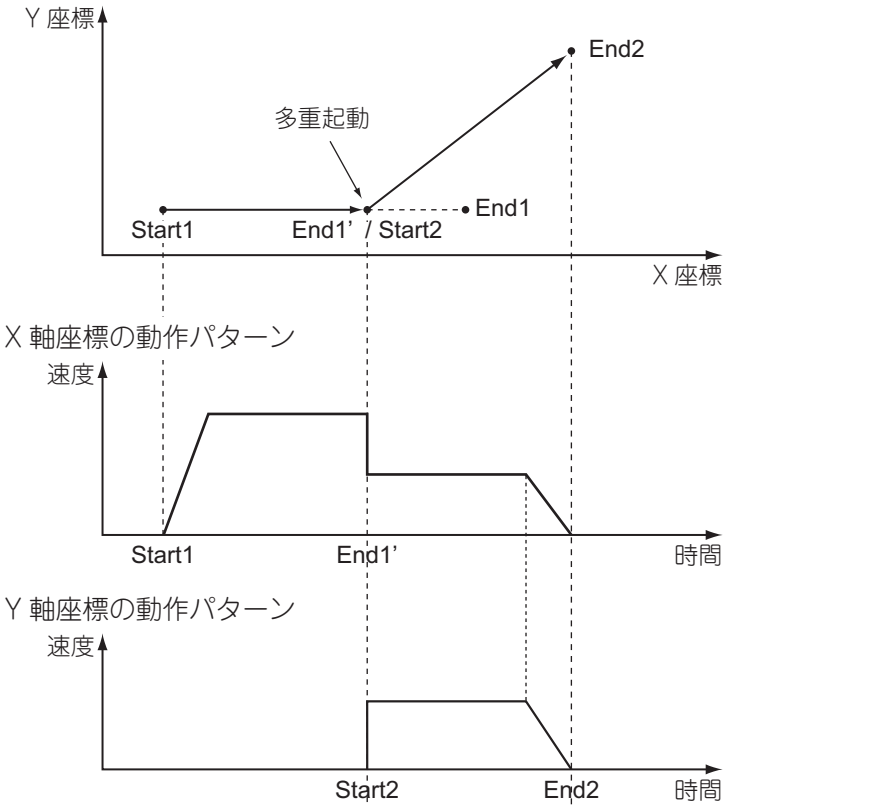
・［BufferMode］（バッファモード選択）が［1: _mcBuffered］の場合は、次のように位置 End1 まで移動して停止したのちに、位置 End2 へ移動します。



・［BufferMode］（バッファモード選択）がブレンディングの場合は、次のように位置 End1 まで移動したのちに、位置 End2 へ移動します。

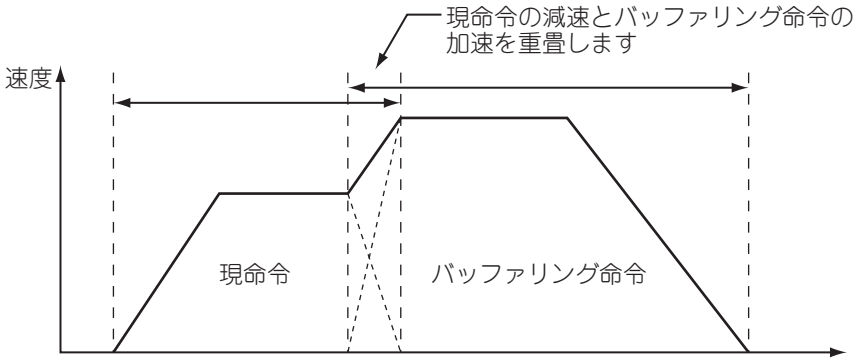


・［BufferMode］（バッファモード選択）が［0: _mcAborting］の場合は、次のように多重起動した位置 End1'から位置 End2 へ移動します。



● **コーナースーパーインポーズ（10: _mcTMCornerSuperimposed）の場合**

現命令の減速とバッファリング命令の加速を重畳します。
バッファ命令の加速度の指定にかかわらず、現命令の減速度による減速時間と同じ時間で動作します。
重畳される区間では、ジャークの指定にかかわらずジャークなしとなります。



［_mcTMCornerSuperimposed］の場合、モーション制御命令の完了を示す出力変数「Done」（完了）は重畳の区間が終了するタイミングで TRUE になります。

参考

以下の 2 つの条件を同時に満たすと、軌跡の線速度が一定となります。
- 現命令とバッファリング命令の目標速度を同じ値にする。
- 現命令の減速度とバッファリング命令の加速度を同じ値にする。

トランジションモードと多重起動の組み合わせ

TransitionMode（トランジションモード）と BufferMode（バッファモード選択）による命令多重起動の組み合わせです。

○: 動作可 –: 異常となり停止

トランジションモード	バッファモード					
	アボーティング	バッファード	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
トランジション無効（_mcTMNone）	○	○	○	○	○	○
コーナースーパーインポーズ *1（_mcTMCornerSuperimposed）	–	–	○	○	○	○

*1. コーナースーパーインポーズの場合は、ブレンディングの種類に関係なく、現命令の減速とバッファリング命令の加速が重畳されます。

9-8 そのほかの機能

ここでは、MC 機能モジュールにあるそのほかの機能について説明します。
NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

9-8-1 現在位置変更機能

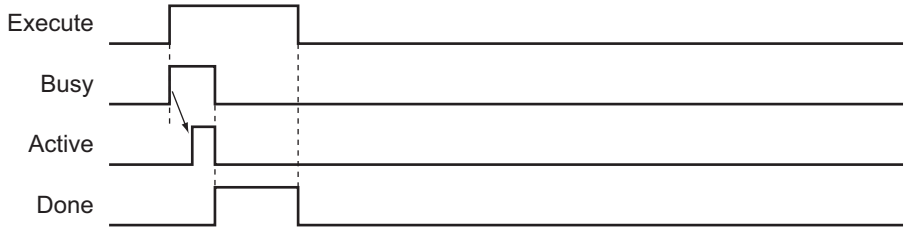
サーボ軸の指令現在位置を任意の値に変更し、フィードバック現在位置は指令現在位置との位置偏差の値を維持した値に変更する機能です。
エンコーダ軸の場合は、フィードバック現在位置を変更します。
MC_SetPosition（現在位置変更）命令を使用し、変更したい現在位置を指定します。

軸の動作中に現在位置を変更することができます。
絶対値への位置決め実行中の場合は、変更した絶対座標上の目標位置へ位置決めします。
相対値での位置決め実行中の場合、移動する距離は変更しません。

使用上の注意

- ［カウントモード］が［ロータリモード］の場合、リングカウンタ範囲外の位置を指定すると命令で異常が発生します。
- 現在位置変更が完了すると原点未確定になるので、下記の機能や命令は使用できなくなります。
  ソフトウェアリミット機能
  高速原点復帰
  補間命令（直線補間と円弧補間）

● 軸が停止中に実行した場合のタイミングチャート



参考

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、「原点プリセット」を指定すると原点確定状態で現在位置を変更できます。

MC_SetPosition（現在位置変更）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 モーション編(SBCE-434)』を参照してください。

9-8-2 トルク制限機能

サーボドライバのトルク制限機能の有効／無効の切り替えとトルク制限値の設定を行うことで、出力トルクを制限する機能です。
正方向へのトルク制限と負方向へのトルク制限を、個別に指定することができます。

機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_SetTorqueLimit（トルク制限）命令を参照してください。

使用上の注意

NX シリーズ パルス出力ユニットを使用する場合は、この機能を使用できません。

9-8-3 ラッチ機能

センサ入力などのトリガ信号の発生を基点とした位置の制御を行うことを目的とし、軸の位置をトリガ信号の発生により記録（ラッチ）する機能です。
軸ごとに同時に 2 点のトリガ信号を設定できます。
MC_TouchProbe（外部ラッチ有効）命令を使用し、ラッチしたい軸に対して「トリガ入力条件」、「ウィンドウ有効」、「停止方法選択」を指定することができます。
トリガにはサーボドライバに接続された信号のほか、ユーザプログラムで扱える変数も指定できます。
ラッチ機能を中断するには、MC_AbortTrigger（外部ラッチ無効）命令を使用します。
ラッチ機能は、オムロン製サーボドライバ 1S シリーズなどラッチ機能（Touch probe）に対応しているサーボドライバや、エンコーダ入力ターミナル 形 GX-EC0211/EC0241 で使用できます。

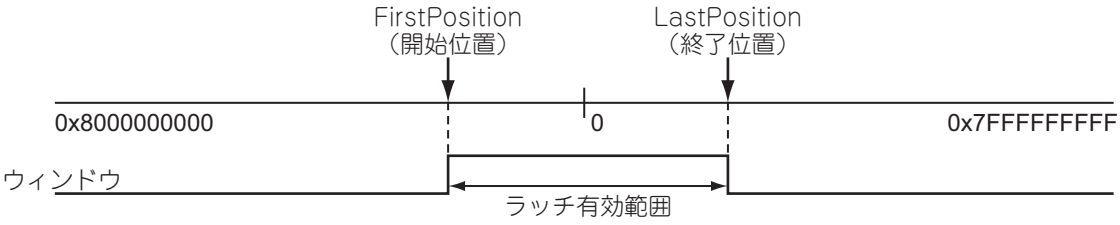
WindowOnly（ウィンドウ有効）を使用することにより、始点と終点の範囲内のみでトリガ信号を検出します。

ラッチ機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_TouchProbe（外部ラッチ有効）命令、および MC_AbortTrigger（外部ラッチ無効）命令を参照してください。

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

● リニアモード

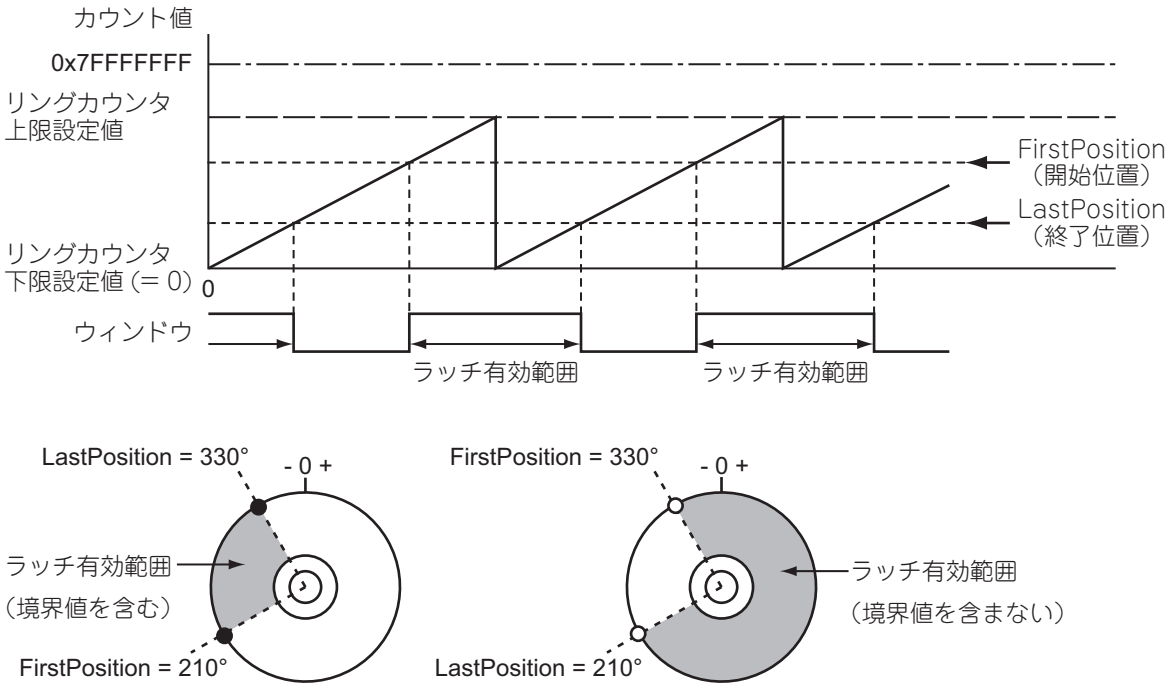
- FirstPosition（開始位置） ≦ LastPosition（終了位置）のときだけ可能です。
- FirstPosition（開始位置） ＞ LastPosition（終了位置）と指定した場合は命令で異常となります。
- リニアモードの位置範囲を超えて指定した場合は命令で異常となります。



● 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。
  後者の場合はリングカウンタ下限設定値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命令で異常となります。

	First Position ≦ Last Position	First Position ＞ Last Position
有効範囲	FirstPosition～LastPosition	LastPosition～FirstPosition



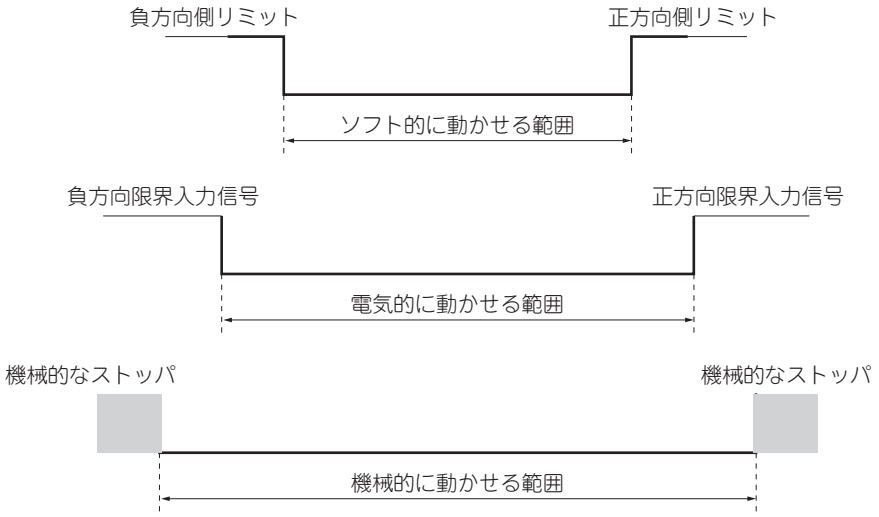
9-8-4　ゾーン機能

指定された範囲（ゾーン）内に、軸の指令位置、あるいはフィードバック位置が存在することを判定する機能です。
MC_ZoneSwitch（ゾーン監視）命令を使用し、判定する範囲の開始位置と終了位置を指定します。
軸の位置が指定したゾーン内に入ると、ゾーン監視命令の出力変数「InZone」（ゾーン内）が TRUE になります。
1 軸で複数のゾーンを指定することができ、ゾーンを重複することもできます。

ゾーン機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編 (SBCE-434)』の MC_ZoneSwitch（ゾーン監視）命令を参照してください。

9-8-5　ソフトウェアリミット機能

ハードウェアの限界入力信号入力とは別に、MC 機能モジュールのソフトウェアにより現在位置を監視する機能です。
ソフトウェアリミットで監視する範囲は、軸パラメータの［正方向ソフトウェアリミット］と［負方向ソフトウェアリミット］を設定します。
通常の位置決め機能では、このソフトウェアリミットの範囲内で動作が可能です。
ソフトウェアリミットを設定することにより、ユーザプログラムのミスや誤操作による機器の破損を防止することができます。



ソフトウェアリミット機能に関する軸パラメータ

パラメータ名称	機能	設定範囲	初期値
ソフトウェアリミット機能	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 *1 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 *1 4: フィードバック位置に対して有効で即停止	0～4	0
正方向ソフトウェアリミット	正方向のソフトウェアリミット値を設定します。 単位は指令単位です。	倍精度実数型の負数、正数、0 *2	2147483647
負方向ソフトウェアリミット	負方向のソフトウェアリミット値を設定します。 単位は指令単位です。		-2147483648

*1. 入力変数に「Deceleration」（減速度）を持つ命令を実行中に、現在位置がソフトウェアリミット範囲外に出た場合は、「Deceleration」（減速度）で減速停止します。
入力変数に「Deceleration」（減速度）を持たない命令を実行中に、現在位置がソフトウェアリミット範囲外に出た場合は、軸パラメータの最大減速度で減速停止します。
*2. パルス単位に変換した値が符号付き整数型 40 ビットの範囲で設定可能です。

上記の軸パラメータは、Sysmac Studio の軸設定、または MC_Write（MC 設定書込）命令、MC_WriteAxisParameter（軸パラメータ書込）命令で設定することができます。
軸や軸グループの動作中に設定値を変更すると、次の動作開始時に変更した設定が有効になります。
軸の動作状態やモーション制御命令により、ソフトウェアリミットは、モーション命令起動時と軸が動作している場合に機能します。

MC 設定書込命令および軸パラメータ書込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 モーション編(SBCE-434)』の MC_Write（MC 設定書込）命令および MC_WriteAxisParameter（軸パラメータ書込）命令を参照してください。

● モーション命令起動時

- 現在位置がソフトウェアリミット範囲内の場合は、目標位置がソフトウェアリミット範囲外の場合に命令で異常が発生します。
- 現在位置がソフトウェアリミット範囲外の場合は、ソフトウェアリミット範囲内へ向かう移動が可能です。このときの目標位置はソフトウェアリミット範囲内の必要はありません。

使用上の注意

軸指令の命令は、目標位置をソフトウェアリミット範囲外にして起動しないでください。

● 軸が動作している場合

位置決め動作中、同期動作中、連続動作中、多軸協調動作中の軸が動作している場合は、以下の動作となります。

- ソフトウェアリミット機能が［指令位置に対して有効］の場合は、指令位置が範囲内から範囲外へ出たときに軸の異常が発生します。
- ソフトウェアリミット機能が［フィードバック位置に対して有効］の場合は、フィードバック位置が範囲内から範囲外へ出たときに軸の異常が発生します。

参考

ソフトウェアリミット機能は、［カウントモード］が［リニアモード］で原点確定状態のときに有効にすることができます。
次の場合は軸パラメータの設定にかかわらず、ソフトウェアリミット機能は無効です。
- ［カウントモード］が［ロータリモード］の場合
- 原点が未確定状態の場合
- 原点復帰動作中の場合

9-8-6 位置偏差監視機能

軸の指令位置とフィードバック位置の差分が位置偏差です。MC 機能モジュールはモーション制御周期ごとに位置偏差を監視しています。
位置偏差の値が軸パラメータの［位置偏差オーバー値］を超えた場合、軽度フォールトレベルの「位置偏差オーバー」が発生します。また、［位置偏差警告値］を超えた場合、監視情報の「位置偏差警告」が発生します。原点復帰で押し当て動作の実行中は、位置偏差監視は無効となります。

● 位置偏差監視機能に関する軸パラメータ

軸パラメータの設定により、位置偏差監視のチェック値を設定できます。［位置偏差警告値］は［位置偏差オーバー値］以下になるように設定してください。
これらの軸パラメータは Sysmac Studio により設定します。

パラメータ名称	機能	設定範囲	初期値
位置偏差オーバー値	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。（単位: 指令単位）	倍精度実数型の正数、0	0
位置偏差警告値	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。（単位: 指令単位）	倍精度実数型の正数、0（位置偏差オーバー値以下）	0

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

9-8-7 偏差カウンタリセット

偏差カウンタリセットは、位置偏差を「0」にする機能です。
ユーザプログラムで偏差カウンタリセットを行うには、MC_ResetFollowingError（偏差カウンタリセット）命令を使用します。
偏差カウンタリセット命令は単軸の位置決め中、および原点復帰中に使用可能です。

軸動作中に偏差カウンタリセットを実行すると、動作中のモーション制御命令は動作を中断し指令位置をフィードバック位置と同じ値に変更します。
偏差カウンタリセットを実行しても原点確定状態は変化しません。

偏差カウンタリセット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 モーション編(SBCE-434)』の MC_ResetFollowingError（偏差カウンタリセット）命令を参照してください。

● 偏差カウンタリセットに関する軸パラメータ

軸パラメータの設定により、即停止時、限界入力停止時、および原点復帰完了時に偏差カウンタリセットすることを選択できます。
これらの軸パラメータは Sysmac Studio により設定します。

パラメータ名称	機能	設定範囲	初期値
即停止入力停止方法	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。 0: 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	0, 2, 3	0
限界入力停止方法	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: 指令値を即停止 1: 指令値を減速停止 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	0～3	0

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

9-8-8 軸間偏差監視機能

2 軸間の指令位置、あるいはフィードバック位置の偏差量を監視する機能です。
MC_AxesObserve（軸間偏差監視）命令を使用し、監視対象の 2 軸と軸間偏差許容値を指定します。
許容値を超えると、軸間偏差監視命令の出力変数「Invalid」（偏差オーバー）が TRUE になります。

ガントリ制御など 2 軸で同じ動作をする装置では、軸間監視機能を使用することで、軸間の偏差が過大となった場合の処理を自由にプログラムすることができます。

使用上の注意

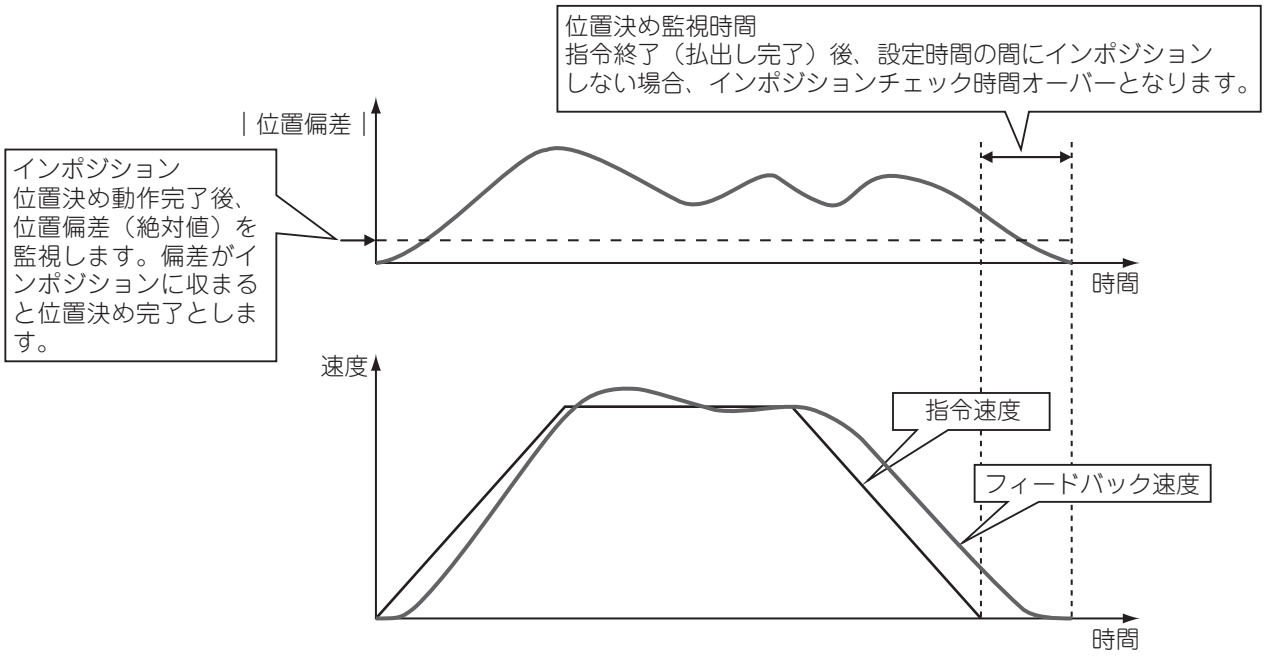
軸間偏差許容値を超えても MC 機能モジュールは異常とはなりません。
出力変数「Invalid」（偏差オーバー）により、ユーザプログラムで軸動作を停止させるなどのプログラム処理を行ってください。

軸間偏差監視機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_AxesObserve（軸間偏差監視）命令を参照してください。

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

9-8-9　インポジションチェック

位置決め動作や原点復帰動作時に、実際のフィードバック現在位置が目標位置に対して指定した範囲内に到達したかをチェックする機能です。
目標位置の指令出力完了後に、目標位置とフィードバック現在位置の差が［インポジション幅］以下になれば位置決め完了となります。
［インポジションチェック時間］以内に位置決め完了しなければ、命令で異常が発生します。



● インポジションチェックに関する軸パラメータ

軸パラメータにより、位置決め完了のチェック条件を設定できます。
確実に位置決め完了してから次の動作を開始させたい場合には、インポジションチェック時間を設定します。

パラメータ名称	機能	設定範囲	初期値
インポジション幅	位置決め完了幅を設定します。（単位: 指令単位）	倍精度実数型の正数、0	10
インポジションチェック時間	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のみ位置決め完了のチェックを継続し、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。（単位: ms）	0～10000	0

上記の軸パラメータは、Sysmac Studio の軸設定、または MC_Write（MC 設定書込）命令、または MC_WriteAxisParameter（軸パラメータ書込）命令で設定することができます。

参考

- インポジションチェックは、MC 機能モジュール側で処理しています。サーボドライバ側の機能は使用していません。
- 少しでも早く動作させたい場合には、位置決め完了を待たずに次の動作に移行させるためにインポジションチェック時間を設定しません。
- CPU ユニットの電源を再投入した場合や、Sysmac Studio の「同期」機能でプログラムをダウンロードした場合、Sysmac Studio で設定した値に戻ります。
  MC_Write（MC 設定書込）命令および MC_WriteAxisParameter（軸パラメータ書込）命令は、一時的にインポジションチェック時間を変更する場合に使用します。

● インポジションチェックに関するモニタ

ユーザプログラムからは、軸変数を読み取ることで位置決め完了をモニタできます。
なお、変数名は、_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_AX[0-255].Details.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。*1 Idle と InPosWaiting は排他的で、同時には TRUE になりません。
_MC_AX[0-255].Details.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。

*1. 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。

軸グループでの位置決めは、軸グループ変数を読み取ることで位置決め完了をモニタできます。
なお、変数名は、_MC_GRP[*]を例に説明していますが、_MC1_GRP[*]、_MC2_GRP[*]についても同様です。

変数名	データ型	名称	機能
_MC_GRP[0-63].Details.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。*1 Idle と InPosWaiting は排他的で、同時には TRUE になりません。
_MC_GRP[0-63].Details.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。*2 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。

*1. 速度「0」での動作も演算している状態に含まれます。
*2. 軸グループ内の全構成軸が、各軸パラメータのインポジション幅の範囲内に入ると FALSE となります。

MC 設定書込命令および軸パラメータ書込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 モーション編(SBCE-434)』の MC_Write（MC 設定書込）命令および MC_WriteAxisParameter（軸パラメータ書込）命令を参照してください。

NX シリーズ パルス出力ユニットを使用する場合の相違については、『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。

9-8-10 軸使用変更機能

MC_ChangeAxisUse（軸使用変更）命令により、軸パラメータの［軸使用］を一時的に切り替える機能です。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。
［軸使用］が［未使用軸（使用軸に切替可）］で、［軸種別］が［サーボ軸］または［仮想サーボ軸］の場合、軸グループに割り付けることができます。
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することができます。

使用上の注意

- ［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。
- 軸パラメータの［軸使用］が［未使用軸（使用軸に切替不可）］に設定されている場合、軸グループに割り付けることはできません。

機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_ChangeAxisUse（軸使用変更）命令を参照してください。

MC_ChangeAxisUse（軸使用変更）命令の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。

9-8-11 デジタルカムスイッチ有効機能

MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令により、軸の位置に応じてデジタル出力を ON または OFF します。
命令の入力変数「ValueSource」の設定により、加減速度を加味することも可能です。
この機能は、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のタイミングで、指定したデジタル出力を ON/OFF します。
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することができます。

使用上の注意

本命令は、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に対して有効です。
使用可能な NX ユニットは、形 NX-EC0□□□、形 NX-ECS□□□で、変化時刻取得機能が動作している必要があります。

デジタルカムスイッチ有効機能の詳細については、『NJ/NX シリーズ コマンドリファレンスマニュアル モーション編(SBCE-434)』の MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令を参照してください。

NX_AryDOutTimeStamp 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。

タイムスタンプ方式対応のデジタル出力ユニットについては、『NX シリーズ デジタル I/O ユニット ユーザーズマニュアル（SBCA-407)』を参照してください。

変化時刻取得機能およびタイムスタンプについては、『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。

9-8-12　任意座標系の 3D モーションモニタ表示

スカラロボットや垂直多関節ロボットなど、直交座標系以外の座標系をユーザプログラムで実現している場合に、ロボットの手先などの軌跡を Sysmac Studio で 3D 表示する機能です。
_sMC_POSITION_REF 型のユーザ定義変数を作成し、3D モーションモニタ表示で表示します。

この機能は、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することができます。

● _sMC_POSITION_REF 型

_sMC_POSITION_REF 型のメンバは以下です。

メンバ	データ型	名称
CommandPosition	ARRAY [0..5] OF LREAL	指令現在位置
ActualPosition	ARRAY [0..5] OF LREAL	フィードバック現在位置

各メンバについて説明します。

メンバ	説明
ユーザ定義変数.CommandPosition[0]	指令現在位置の X 軸成分です。 ユーザプログラムで生成する指令現在位置の X 軸上の位置を示すユーザ定義変数を代入します。
ユーザ定義変数.CommandPosition[1]	指令現在位置の Y 軸成分です。 ユーザプログラムで生成する指令現在位置の Y 軸上の位置を示すユーザ定義変数を代入します。
ユーザ定義変数.CommandPosition[2]	指令現在位置の Z 軸成分です。 ユーザプログラムで生成する指令現在位置の Z 軸上の位置を示すユーザ定義変数を代入します。
ユーザ定義変数.CommandPosition[3]～[5]	未使用
ユーザ定義変数.ActualPosition[0]	フィードバック現在位置の X 軸成分です。 ユーザプログラムで扱うフィードバック現在位置の X 軸上の位置を示すユーザ定義変数を代入します。
ユーザ定義変数.ActualPosition[1]	フィードバック現在位置の Y 軸成分です。 ユーザプログラムで扱うフィードバック現在位置の Y 軸上の位置を示すユーザ定義変数を代入します。
ユーザ定義変数.ActualPosition[2]	フィードバック現在位置の Z 軸成分です。 ユーザプログラムで扱うフィードバック現在位置の Z 軸上の位置を示すユーザ定義変数を代入します。
ユーザ定義変数.ActualPosition[3]～[5]	未使用

各メンバにはユーザ定義変数を代入します。次に例を示します。

名称	データ型	説明
3D_position	_sMC_POSITION_REF	3D 表示するためのユーザ定義変数
MCS_Cmd_TransX	LREAL	ユーザプログラムで生成する指令現在位置の X 軸上の位置を示すユーザ定義変数
MCS_Cmd_TransY	LREAL	ユーザプログラムで生成する指令現在位置の Y 軸上の位置を示すユーザ定義変数
MCS_Cmd_TransZ	LREAL	ユーザプログラムで生成する指令現在位置の Z 軸上の位置を示すユーザ定義変数
MCS_Act_TransX	LREAL	ユーザプログラムで扱うフィードバック現在位置の X 軸上の位置を示すユーザ定義変数
MCS_Act_TransY	LREAL	ユーザプログラムで扱うフィードバック現在位置の Y 軸上の位置を示すユーザ定義変数
MCS_Act_TransZ	LREAL	ユーザプログラムで扱うフィードバック現在位置の Z 軸上の位置を示すユーザ定義変数

```
3D_position.CommandPosition[0] := MCS_Cmd_TransX;
3D_position.CommandPosition[1] := MCS_Cmd_TransY;
3D_position.CommandPosition[2] := MCS_Cmd_TransZ;
3D_position.ActualPosition[0] := MCS_Act_TransX;
3D_position.ActualPosition[1] := MCS_Act_TransY;
3D_position.ActualPosition[2] := MCS_Act_TransZ;
```

● 使用手順の概要

1 **_sMC_POSITION_REF 型のユーザ定義変数を作成します。**

2 **作成したユーザ定義変数の各メンバに、3D 表示する「指令現在位置」と「フィードバック現在位置」を示すユーザ定義変数を代入するプログラムを作成します。**

3 **3D 機構モデルリストの種別から［指定座標］を選択します。**
3D 機構モデルパラメータ設定に_sMC_POSITION_REF 型が表示されます。

4 **3D 機構モデルパラメータ設定の設定値欄に、作成したユーザ定義変数を設定します。**

5 **ユーザプログラムを実行します。**

6 **データトレースでトレースを開始し、データのサンプリングを行います。**

7 **データトレース画面で、トレース結果を確認します。**

3D モーションモニタ表示の詳細については、『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。

10

サンプルプログラム

この章では、原点復帰や異常の監視などの基本的な使い方と、絶対値位置決めやカム動作などの軸動作のプログラム例を記載しています。

10-1　サンプルプログラムの概要 10-2
10-1-1　使用機器 10-2
10-1-2　設置と配線 10-2
10-1-3　設定 10-2

10-2　基本プログラム例 10-3
10-2-1　EtherCAT 通信の監視とサーボ ON 10-3
10-2-2　マスタコントロール命令による軸動作のインタロック 10-4
10-2-3　異常の監視と異常解除（単軸動作、同期動作） 10-6
10-2-4　異常の監視と異常解除（多軸協調動作） 10-8
10-2-5　命令異常の監視方法 10-14
10-2-6　異常解除の監視方法 10-15
10-2-7　単軸での動作中における軸停止 10-17
10-2-8　多軸協調動作中での軸グループ停止 10-21
10-2-9　原点復帰と絶対値位置決め 10-28
10-2-10　命令の再起動により目標位置を変更する 10-33
10-2-11　割り込み定寸位置決め 10-38
10-2-12　命令の再起動によりカムテーブルを切り替える 10-43
10-2-13　カムプロファイル曲線を使って同期位置を補正する 10-53
10-2-14　カム動作時に主軸の位相を補正する 10-63
10-2-15　速度制御中に現在位置を変更する 10-72
10-2-16　カムデータ変数の変更とカムテーブルの保存 10-77
10-2-17　軸パラメータの一時書き換え 10-87
10-2-18　カムテーブルの終点インデックス更新 10-90

10-1 サンプルプログラムの概要

ここでは、本章で記載しているサンプルプログラムの共通事項について説明します。

使用上の注意

- 本章のサンプルプログラムは、MC 機能モジュールの機能を使用する部分だけを記載しています。
- 実際の装置のプログラムを作成する場合は、装置の動作に関するインタロックプログラムやほかの機器からの入出力、制御手順等に関するプログラムを作成してください。
- 実際の装置が意図した動作となるように、ユーザプログラムを作成してください。
- 作成したユーザプログラムは、十分な動作確認を行ったのち、本運転に移行してください。
- NX シリーズ 位置インタフェースユニットの使用例は、『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。

10-1-1 使用機器

ここで説明するサンプルプログラムでは、つぎのデバイスの使用を前提としています。

機器	サーボ構成の例
CPU ユニット	形 NJ501-1□00 (Ver.1.0)
電源ユニット	形 NJ-Px3001
サーボドライバ	形 R88D-1SN□□□-ECT
サーボモータ	形 R88M-1□□□□□□□□
エンコーダ入力ターミナル	形 GX-EC0211 (Ver.1.1)

10-1-2 設置と配線

各使用機器の設置と配線については、下記のマニュアルを参照してください。

使用機器	参照マニュアル
CPU ユニットと電源ユニット	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)
サーボドライバ／モータ	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)
エンコーダ入力ターミナル	形 GX シリーズ EtherCAT スレーブ ユーザーズマニュアル (SBCD-350)
EtherCAT 通信の配線	

10-1-3 設定

各種設定については、下記のマニュアルを参照してください。

使用機器	参照マニュアル
コントローラ設定	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)
モーション制御設定	「3-2 軸の設定手順 (P.3-9)」、「5-2 軸パラメータ (P.5-4)」、「A-1 サーボドライバ 1S シリーズとの接続 (P.A-2)」
サーボパラメータ	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)

10-2 基本プログラム例

ここでは、MC 機能モジュールの基本機能に関するユーザプログラム例を示します。

使用上の注意

- 本プログラム例を参考にする場合は、装置の動作条件に合うように適切なインタロックプログラムを追加してください。
- プログラムで使用する変数は Sysmac Studio の［プログラミング］画面で入力してください。

10-2-1 EtherCAT 通信の監視とサーボ ON

対象のサーボドライバと EtherCAT 通信のプロセスデータ通信が確立すれば、MC_Power（運転可）命令によりサーボドライバをサーボ ON します。

プログラムで使用する主な変数

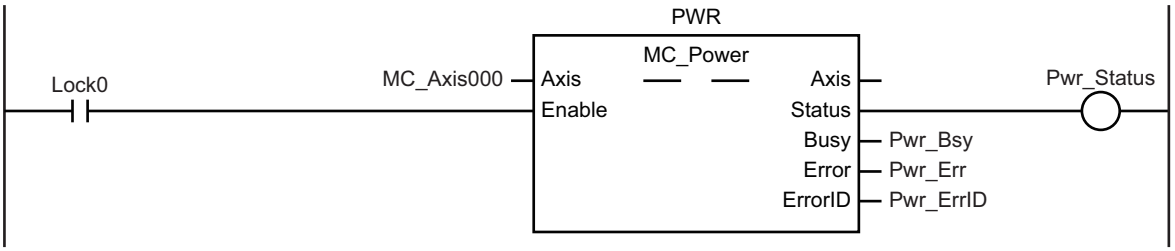
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
MC_Axis000.Cfg.NodeAddress	UINT	−	ノードアドレスです。
_EC_PDSlavTbl[N]	BOOL	FALSE	TRUE のとき、ノードアドレス N の EtherCAT 通信のプロセスデータ通信がオペレーショナル状態であることを表します。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。

ラダー図プログラム

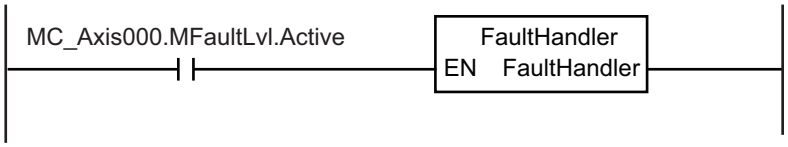
接点 StartPg が TRUE で、サーボドライバがサーボレディ状態か確認します



サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します



軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



ST プログラム

```
// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En:=TRUE;
  ELSE
    Pwr_En:=FALSE;
END_IF;

// 軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせてプログラミングしてください。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// MC_Power
PWR(
  Axis := MC_Axis000,
  Enable := Pwr_En,
  Status => Pwr_Status,
  Busy => Pwr_Bsy,
  Error => Pwr_Err,
  ErrorID => Pwr_ErrID
);
```

10-2-2　マスタコントロール命令による軸動作のインタロック

ラダー図プログラムで MC（マスタコントロール）命令と MCR（マスタコントロールリセット）命令の間に MC_Power（運転可）命令を配置することで、軸動作のインタロック制御を行います。
接点 Mc_On を FALSE にすると、MC 命令と MCR 命令の間に記述した MC_Power（運転可）命令が無効になりサーボ OFF となります。
このとき、実行中の動作命令の出力変数「CommandAborted」（実行中断）が TRUE となり、軸動作は停止します。

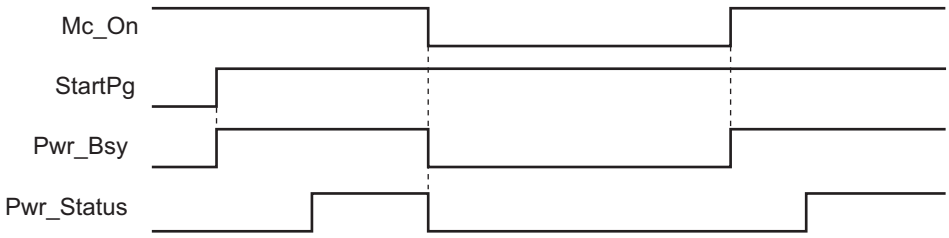
使用上の注意

MC 命令は、ST プログラムでは使用できません。

プログラムで使用する主な変数

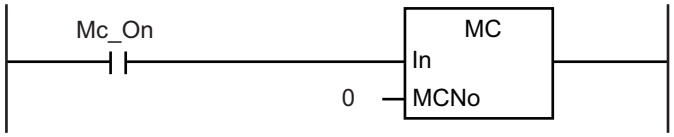
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
Mc_On	BOOL	FALSE	MC 命令の有効および無効を制御します。 このサンプルプログラムでは制御する記述を記載していません。実際のプログラムでは装置の動作にあわせて制御してください。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。

タイミングチャート

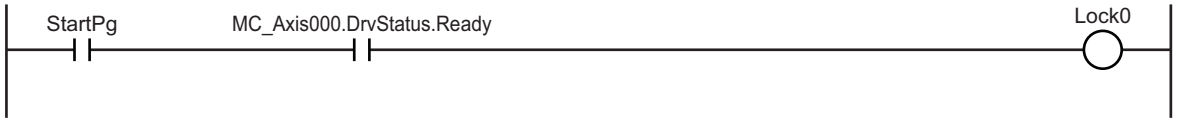


ラダー図プログラム

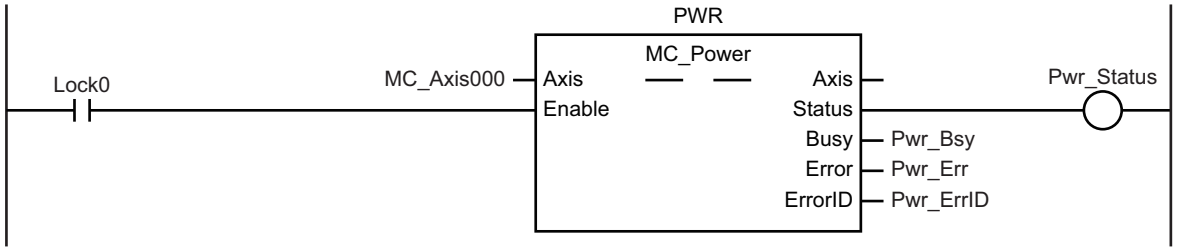
接点 Mc_On が TRUE になれば、マスタコントロールを開始します



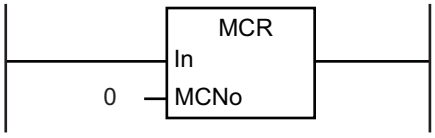
接点 StartPg が TRUE で、サーボドライバがサーボレディ状態か確認します



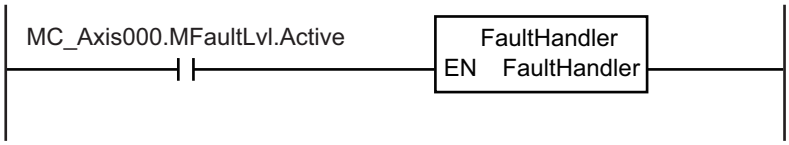
サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します



マスタコントロールを終了します



軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します
異常処理の内容は装置にあわせてプログラミングしてください



10-2-3 異常の監視と異常解除（単軸動作、同期動作）

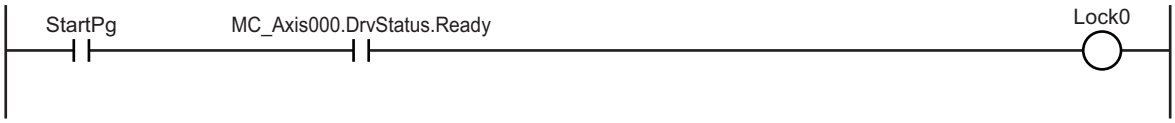
軸変数にある「軸 軽度フォールト発生中」の状態により異常の有無を監視します。
軽度フォールトレベルの異常が発生した場合は、MC_Power（運転可）命令の入力変数「Enable」（有効）が FALSE となりサーボ OFF します。
外部ボタンが ON で指令現在速度がゼロであれば、MC_Reset（軸エラーリセット）命令により異常を解除します。

プログラムで使用する主な変数

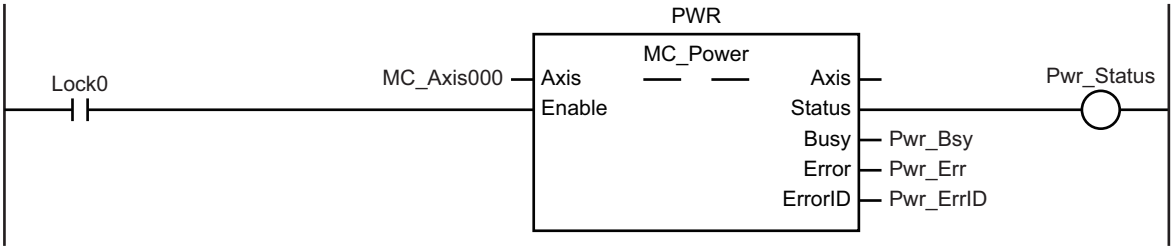
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
MC_Axis000.Status.ErrorStop	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生し、軸が減速停止中または停止中の場合に TRUE となります。
MC_Axis000.Details.Idle	BOOL	FALSE	インポジション待ちのとき以外で、軸 0 の指令現在速度がゼロの場合に TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
ResetON	BOOL	FALSE	異常を解除するための外部ボタンの状態を表します。

ラダー図プログラム

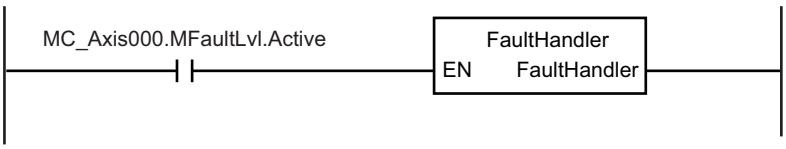
接点 StartPg が TRUE で、サーボドライバがサーボレディ状態か確認します



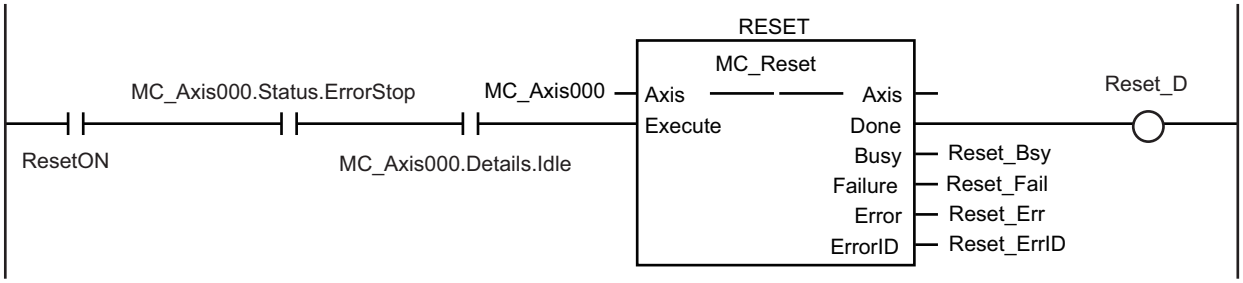
サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します



軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します
異常処理の内容は装置にあわせてプログラミングしてください



外部ボタンの ON により ResetON が TRUE となり、指令現在速度がゼロであれば、異常を解除します



ST プログラム

```
// StartPg が TRUE で、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 0 をサーボ ON にします。
// サーボレディ状態になっていない場合は、軸 0 をサーボ 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 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせてプログラミングしてください。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 外部ボタンの ON により ResetON が TRUE となり、指令現在速度がゼロであれば、異常を解除します。
IF (ResetOn=TRUE)
  AND (MC_Axis000.Status.ErrorStop=TRUE)
  AND (MC_Axis000.Details.Idle=TRUE) THEN
    Reset_Ex := TRUE; // 軽度フォールトの解除
END_IF;
```

```
// MC_Power
PWR(
  Axis := MC_Axis000,
  Enable := Pwr_En,
  Status => Pwr_Status,
  Busy => Pwr_Bsy,
  Error => Pwr_Err,
  ErrorID => Pwr_ErrID
);

// MC_Reset
RESET(
  Axis := MC_Axis000,
  Execute := Reset_Ex,
  Done => Reset_D,
  Busy => Reset_Bsy,
  Failure => Reset_Fai,
  Error => Reset_Err,
  ErrorID => Reset_ErrID
);
```

10-2-4　異常の監視と異常解除（多軸協調動作）

軸変数にある「軸 軽度フォールト発生中」の状態、および軸グループ変数にある「軸グループ 軽度フォールト発生中」の状態により異常の有無を監視します。
軽度フォールトレベルの異常が発生した場合は、MC_GroupDisable（軸グループ無効）命令の入力変数「Execute」（起動）が TRUE となり軸グループを無効にします。
外部ボタンが ON で軸グループの指令現在速度がゼロであれば、MC_GroupReset（軸グループエラーリセット）命令により異常を解除します。

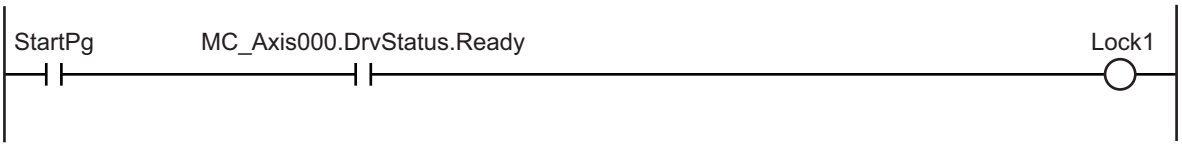
プログラムで使用する主な変数

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.Details.Idle	BOOL	FALSE	インポジション待ちのとき以外で、軸グループ 0 の指令補間速度がゼロの場合に TRUE となります。
MC_Axis000	_sAXIS_REF	－	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 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_Axis000.Details.Homed	BOOL	FALSE	軸 0 が原点確定状態のとき TRUE となります。
MC_Axis001	_sAXIS_REF	－	軸 1 の軸変数です。
MC_Axis001.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.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき TRUE となります。

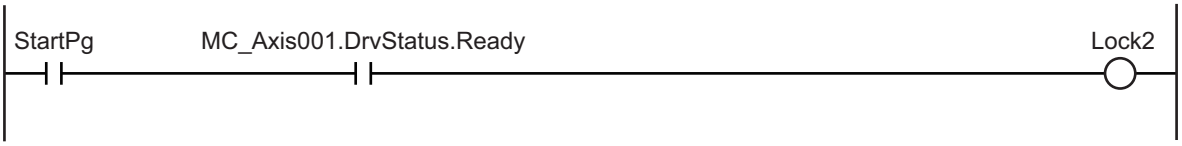
名称	データ型	初期値	コメント
Pwr1_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR1 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR2 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
ResetON	BOOL	FALSE	異常を解除するための外部ボタンの状態を表します。

ラダー図プログラム

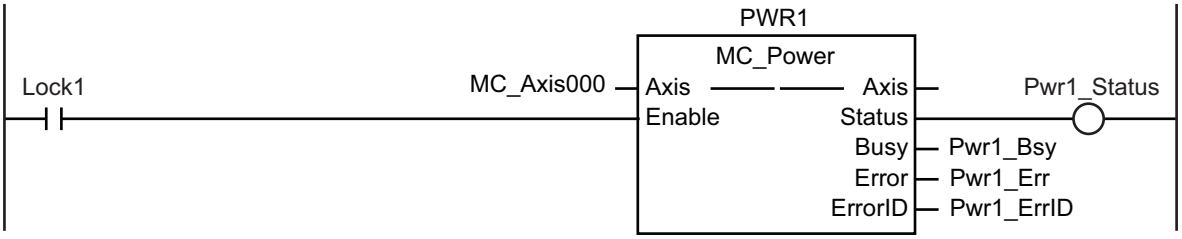
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します



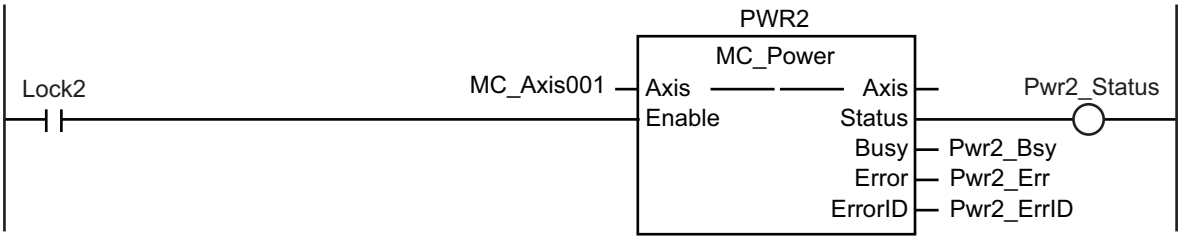
接点 StartPg が TRUE で、軸 1 のサーボドライバがサーボレディ状態か確認します



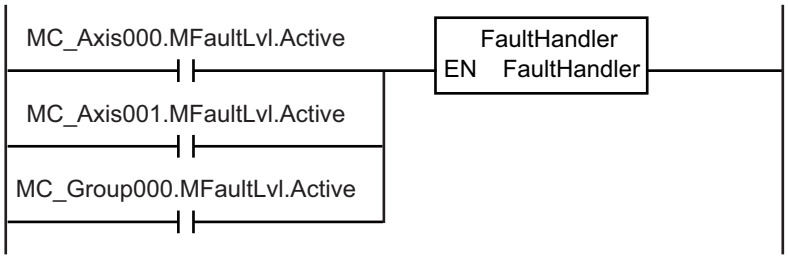
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します



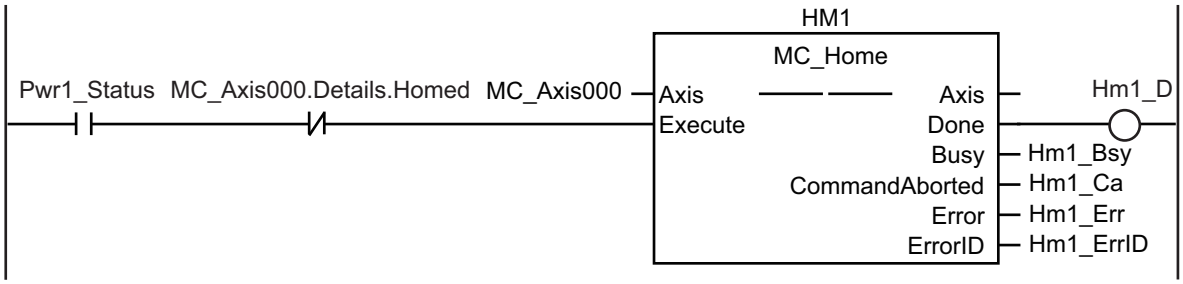
軸 1 のサーボドライバがサーボレディ状態であれば、軸 1 をサーボ ON します



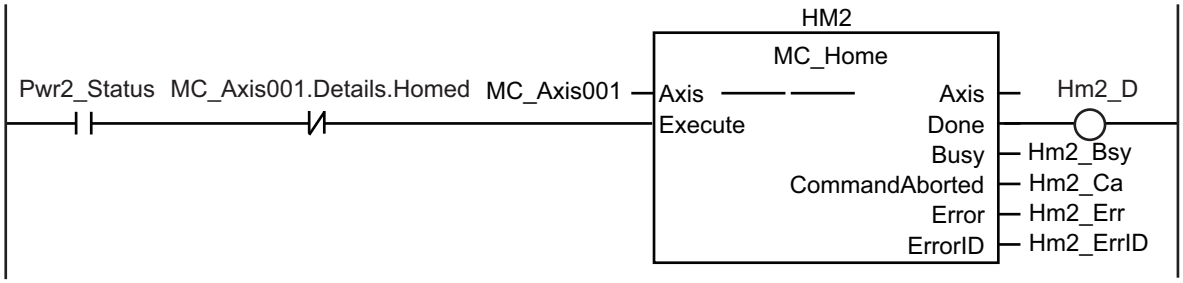
軸グループあるいは構成軸で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



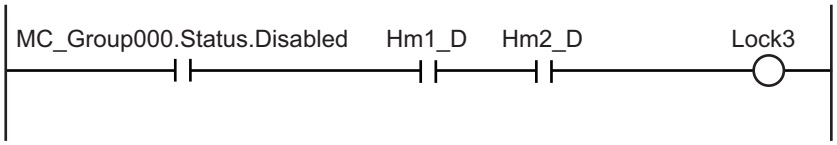
軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します



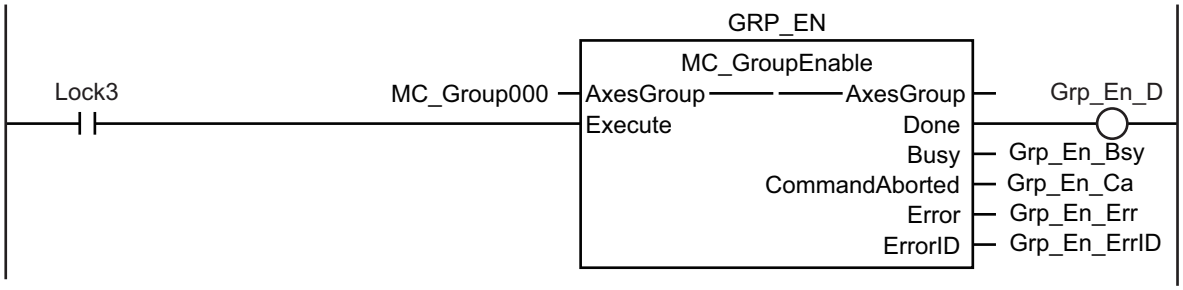
軸 1 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します



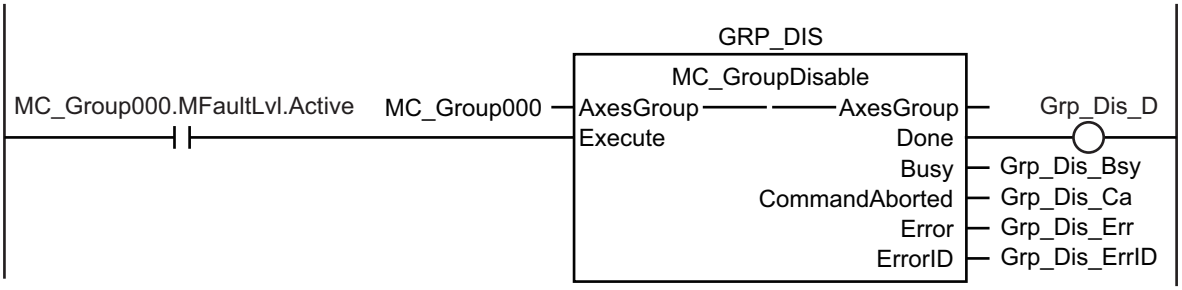
軸グループの無効状態および軸 0 と軸 1 の原点確定を確認します



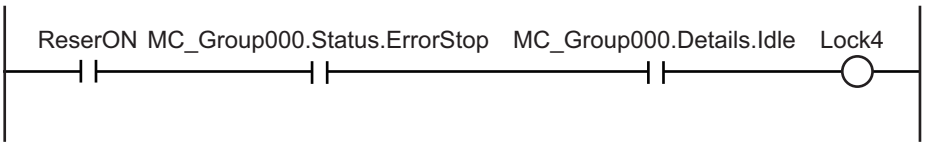
軸 0 と軸 1 が原点確定していれば軸グループを有効にします



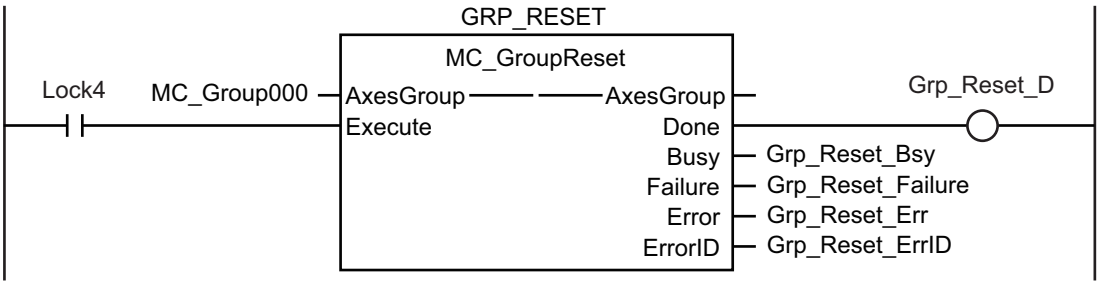
軸グループで軽度フォールトレベルの異常が発生した場合に軸グループを無効にします



外部ボタンの ON による ResetON の状態と軸グループの停止を確認します



ResetON が TRUE で軸グループが停止していれば異常を解除します



ST プログラム

```
// StartPg が TRUE で、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 0 をサーボ ON にします。
// サーボレディ状態になっていない場合は、軸 0 をサーボ OFF にします。
IF (StartPg =TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En :=TRUE; // 軸 0 をサーボ ON します。
  ELSE
    Pwr1_En :=FALSE; // 軸 0 をサーボ OFF します。
END_IF;

//StartPg が TRUE で、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 にします。
// サーボレディ状態になっていない場合は、軸 1 をサーボ OFF にします。
IF (StartPg =TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En :=TRUE; // 軸 1 をサーボ ON します。
  ELSE
    Pwr2_En :=FALSE; // 軸 1 をサーボ OFF します。
END_IF;

// 軸グループあるいは構成軸で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実
```

行します。

```
IF (MC_Axis000.MFaultLvl.Active=TRUE)
  OR (MC_Axis001.MFaultLvl.Active=TRUE)
  OR (MC_Group000.MFaultLvl.Active=TRUE) THEN
    FaultHandler(); // 異常処理の内容は装置にあわせてプログラミングしてください。
END_IF;

// 軸 0 がサーボ ON 状態で原点未確定の場合、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 1 がサーボ ON 状態で原点未確定の場合、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;

// 軸グループが無効で軸 0 と軸 1 が原点確定していれば軸グループを有効にします。
IF (MC_Group000.Status.Disabled=TRUE) AND (Hm1_D=TRUE) AND (Hm2_D=TRUE) THEN
  Grp_En_Ex := TRUE;
END_IF;

// 軸グループで軽度フォールトレベルの異常が発生した場合に軸グループを無効にします。
IF MC_Group000.MFaultLvl.Active=TRUE THEN
  Grp_Dis_Ex :=TRUE;
END_IF;

// 外部ボタンの ON による ResetON が TRUE で軸グループが停止していれば異常を解除します。
IF (ResetON=TRUE)
  AND (MC_Group000.Status.ErrorStop=TRUE)
  AND (MC_Group000.Details.Idle=TRUE) THEN
    Grp_Reset_Ex := TRUE;
END_IF;

//MC_Power1
PWR1(
  Axis := MC_Axis000,
  Enable := Pwr1_En,
  Status => Pwr1_Status,
  Busy => Pwr1_Bsy,
  Error => Pwr1_Err,
  ErrorID => Pwr1_ErrID
);

//MC_Power2
PWR2(
  Axis := MC_Axis001,
  Enable := Pwr2_En,
  Status => Pwr2_Status,
  Busy => Pwr2_Bsy,
  Error => Pwr2_Err,
```

```
  ErrorID => Pwr2_ErrID
);

// MC_Home1
HM1(
  Axis := MC_Axis000,
  Execute := Hm1_Ex,
  Done => Hm1_D,
  Busy => Hm1_Bsy,
  CommandAborted => Hm1_Ca,
  Error => Hm1_Err,
  ErrorID => Hm1_ErrID
);

// MC_Home2
HM2(
  Axis := MC_Axis001,
  Execute := Hm2_Ex,
  Done => Hm2_D,
  Busy => Hm2_Bsy,
  CommandAborted => Hm2_Ca,
  Error => Hm2_Err,
  ErrorID => Hm2_ErrID
);

//MC_GroupEnable
GRP_EN(
  AxesGroup := MC_Group000,
  Execute := Grp_En_Ex,
  Done => Grp_En_D,
  Busy => Grp_En_Bsy,
  CommandAborted => Grp_En_Ca,
  Error => Grp_En_Err,
  ErrorID => Grp_En_ErrID
);

//MC_GroupDisable
GRP_DIS(
  AxesGroup := MC_Group000,
  Execute := Grp_Dis_Ex,
  Done => Grp_Dis_D,
  Busy => Grp_Dis_Bsy,
  CommandAborted => Grp_Dis_Ca,
  Error => Grp_Dis_Err,
  ErrorID => Grp_Dis_ErrID
);

//MC_GroupReset
GRP_RESET(
  AxesGroup := MC_Group000,
  Execute := Grp_Reset_Ex,
```

```
  Done => Grp_Reset_D,
  Busy => Grp_Reset_Bsy,
  Failure => Grp_Reset_Fai,
  Error => Grp_Reset_Err,
  ErrorID => Grp_Reset_ErrID
);
```

10-2-5　命令異常の監視方法

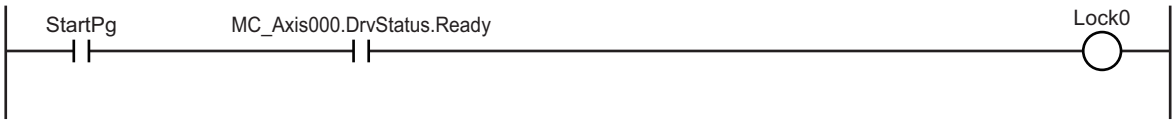
運転可（MC_Power）命令の起動時に異常が発生した場合、以降の処理は実行しません。
本サンプルプログラムでは、以降の処理の許可を変数「UpgOn」で表しています。

プログラムで使用する主な変数

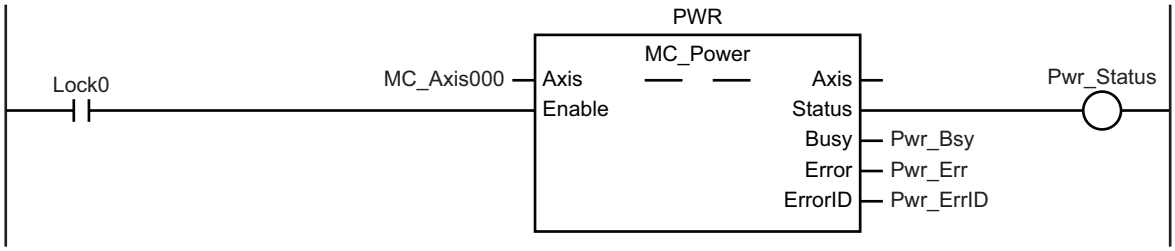
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	－	軸 0 の軸変数です。
Pwr_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
Pwr_Bsy	BOOL	FALSE	MC_Power 命令のインスタンス PWR の出力 Busy に割り付けた変数です。 サーボ ON のときに TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
UpgOn	BOOL	FALSE	TRUE 時に、ユーザプログラムの以降の処理を実行します。

ラダー図プログラム

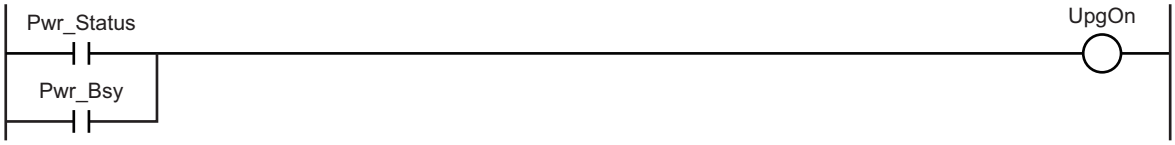
接点 StartPg が TRUE で、サーボドライバがサーボレディ状態か確認します



サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します



MC_Power で異常が発生していないことを確認し、以降の処理を実行します



ST プログラム

```
// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
// サーボレディ状態になっていない場合は、軸 0 をサーボ OFF にします。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En :=TRUE;
  ELSE
    Pwr_En :=FALSE;
END_IF;

IF (Pwr_Status=TRUE) OR (Pwr_Bsy=TRUE) THEN
    UpgOn := TRUE; // 以降の処理を実行します。
  ELSE
    UpgOn := FALSE; // 以降の処理を実行しません。
END_IF;

// MC_Power
PWR(
  Axis := MC_Axis000,
  Enable := Pwr_En,
  Status => Pwr_Status,
  Busy => Pwr_Bsy,
  Error => Pwr_Err,
  ErrorID => Pwr_ErrID
);
```

10-2-6　異常解除の監視方法

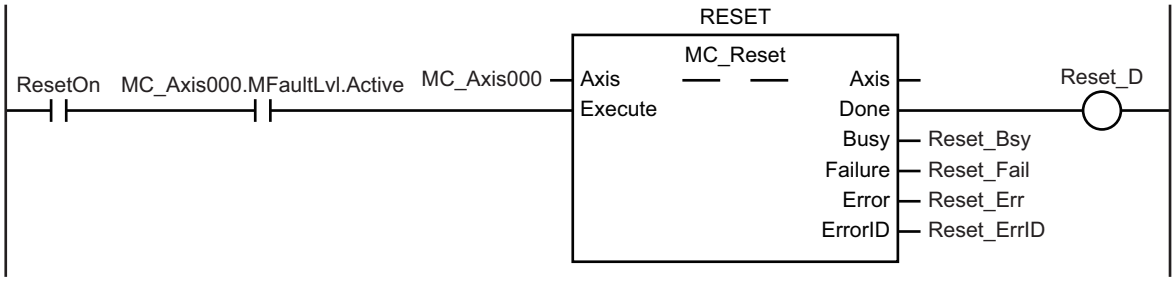
軽度フォールトレベルの異常が発生している状態で外部ボタンを ON すると、軸エラーリセット（MC_Reset）命令を起動します。軸エラーリセット（MC_Reset）命令の出力「Done」が TRUE になるまで、次の正常処理を実行しません。
出力「Failure」が TRUE になれば、軸が減速停止中、あるいは MC 共通異常が発生しています。軸エラーリセット（MC_Reset）命令の「Failure」が TRUE となった要因を取得します。

プログラムで使用する主な変数

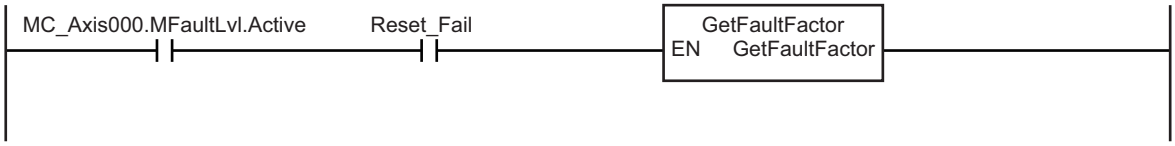
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
ResetON	BOOL	FALSE	異常を解除するための外部ボタンの状態を表します。
GetFaultFactor	−	−	異常要因の取得処理です。 装置にあわせてプログラミングしてください。
RegularProcess	−	−	正常処理です。 装置にあわせてプログラミングしてください。

ラダー図プログラム

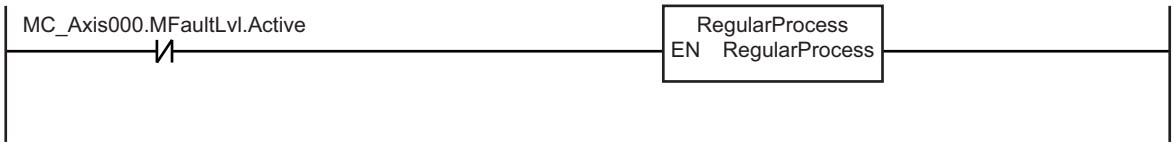
軽度フォールトレベルの異常が発生している状態で外部ボタン（ResetOn）を ON すると、
軸エラーリセット（MC_Reset）命令を実行します



軸エラーリセット（MC_Reset）命令の出力変数「Failure」が TRUE なら、装置の異常要因の取得処理
（GetFaultFactor）を実行します。処理内容は装置にあわせてプログラミングしてください



軽度フォールトレベルの異常が発生していない、あるいは軽度フォールトレベルの異常が解除された場合は、
装置の正常処理（RegularProcess）を実行します。
処理内容は装置にあわせてプログラミングしてください



ST プログラム

```
// 軽度フォールトレベルの異常が発生している状態で外部ボタン(ResetOn)を ON すると、軸エラーリセット(MC_
Reset)命令を実行します。
IF (MC_Axis000.MFaultLvl.Active=TRUE) AND (ResetOn=TRUE) THEN
    Reset_Ex := TRUE; // 軽度フォールトの解除
  ELSE
    Reset_Ex := FALSE;
END_IF;

// 軸エラーリセット(MC_Reset)命令の出力変数「Failure」が TRUE なら、異常要因の取得処理(GetFaultFactor)
を実行します。
// 処理内容は装置にあわせてプログラミングしてください。
IF (MC_Axis000.MFaultLvl.Active=TRUE) AND (Reset_Fail=TRUE) THEN
  GetFaultFactor();
END_IF;

// 軽度フォールトレベルの異常が発生していない、あるいは軽度フォールトレベルの異常が解除された場合は、
装置の正常処理(RegularProcess)を実行します。
// 処理内容は装置にあわせてプログラミングしてください。
IF MC_Axis000.MFaultLvl.Active=FALSE THEN
  RegularProcess();
```

```
END_IF;

// MC_Reset
RESET(
  Axis := MC_Axis000,
  Execute := Reset_Ex,
  Done => Reset_D,
  Busy => Reset_Bsy,
  Failure => Reset_Fail,
  Error => Reset_Err,
  ErrorID => Reset_ErrID
);
```

10-2-7　単軸での動作中における軸停止

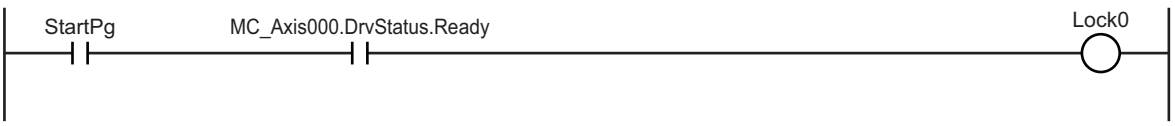
絶対値位置決め（MC_MoveAbsolute）命令を実行中に外部ボタンを ON すると、強制停止（MC_Stop）命令を実行して減速停止させます。軽度フォールトレベルの異常が発生している場合には強制停止（MC_Stop）命令の出力「CommandAborted」が TRUE となります。
このとき、即停止（MC_ImmediateStop）を実行して即停止させます。
また、何らかの理由により強制停止（MC_Stop）命令の出力「Error」が TRUE となった場合も、即停止（MC_ImmediateStop）命令を実行して即停止させます。即停止（MC_ImmediateStop）命令を実行すると、軸はエラー減速停止中になります。

プログラムで使用する主な変数

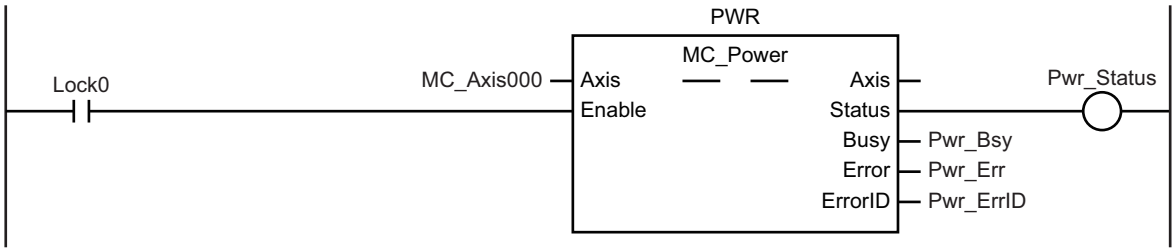
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 0 が原点確定状態のとき TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
Stp_Ca	BOOL	FALSE	MC_Stop のインスタンス STP の出力変数 CommandAborted に割り付けた変数です。
Stp_Err	BOOL	FALSE	MC_Stop のインスタンス STP の出力変数 Error に割り付けた変数です。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
StopOn	BOOL	FALSE	強制停止を行うための外部ボタンの状態です TRUE のとき、MC_Stop による強制停止を行います。

ラダー図プログラム

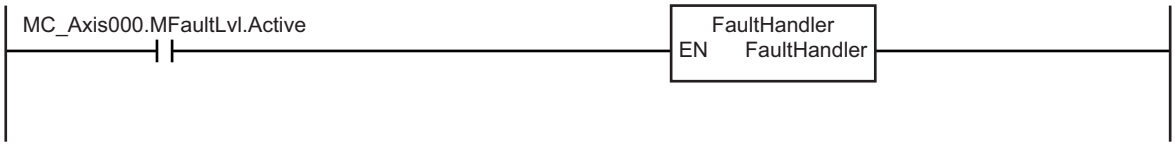
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します



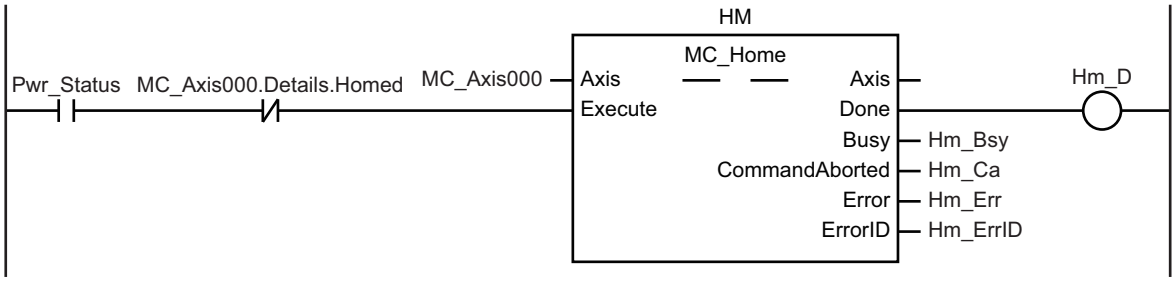
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します



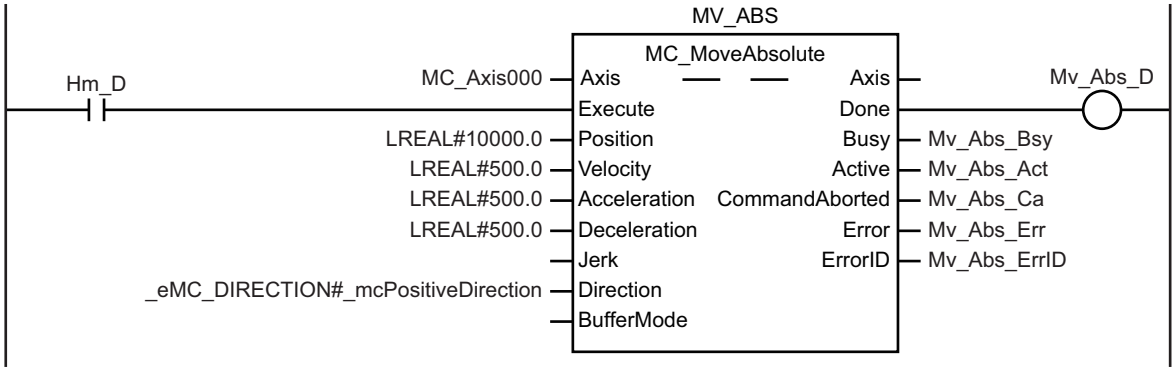
軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します
異常処理の内容は装置にあわせてプログラミングしてください



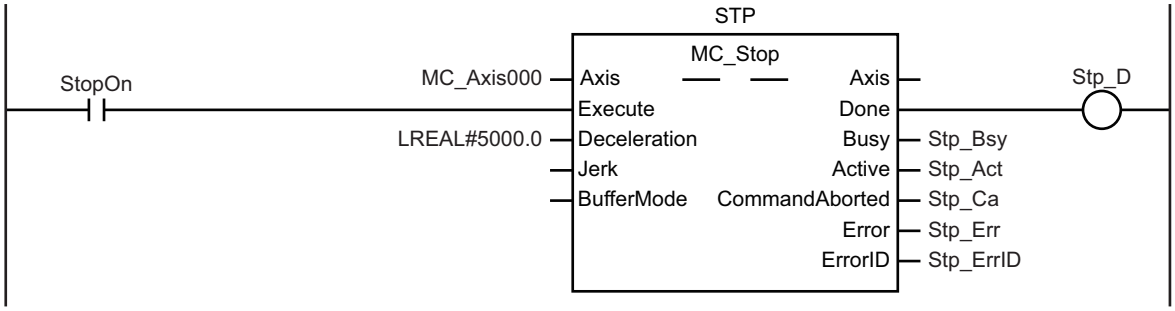
軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します



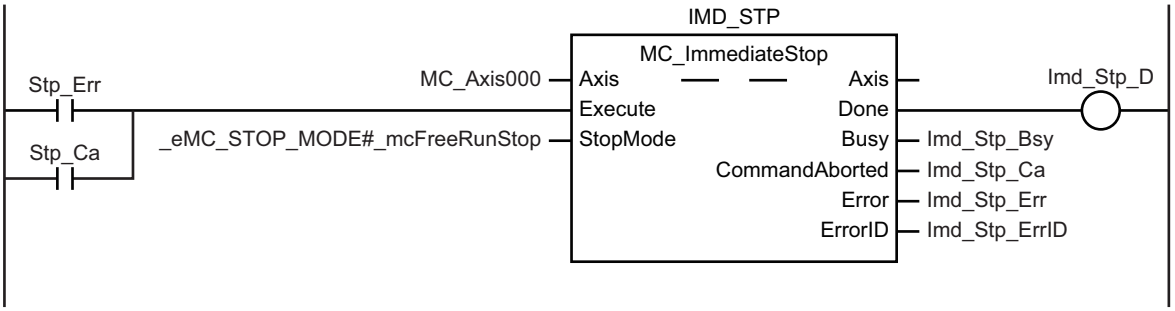
軸 0 の原点復帰が完了すれば、絶対値位置決め (MC_MoveAbsolute) 命令を実行します



StopOn が TRUE の場合に、強制停止 (MC_Stop) 命令を実行します



強制停止(MC_Stop)命令の出力「Error」、あるいは「CommandAborted」が TRUE のとき、
即停止(MC_ImmediateStop)命令を実行し即停止します



ST プログラム

```
// 絶対値位置決めと強制停止の入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag=FALSE THEN
  // 絶対値位置決め(MC_MoveAbsolute)の入力パラメータに値を設定します。
    Mv_Abs_Pos := LREAL#10000.0;
    Mv_Abs_Vel := LREAL#500.0;
    Mv_Abs_Acc := LREAL#500.0;
    Mv_Abs_Dec := LREAL#500.0;
    Mv_Abs_Dir := _eMC_DIRECTION#_mcPositiveDirection;
  // 強制停止(MC_Stop)の入力パラメータに値を設定します。
    Stp_Dec:=LREAL#5000.0;
  // 即停止(MC_ImmediateStop)の入力パラメータに値を設定します。
    Imd_Stp_SM :=_eMC_STOP_MODE#_mcFreeRunStop;
  // 入力パラメータの設定完了フラグを TRUE にします。
    InitFlag := TRUE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En :=TRUE;
  ELSE
    Pwr_En :=FALSE;
END_IF;

// 軸 0 で軽度フォールトレベルの異常が発生した場合、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせてプログラミングしてください。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 軸 0 がサーボ ON 状態で原点未確定の場合、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;

// 原点復帰が完了すれば絶対値位置決めを実行します。
```

```
IF Hm_D=TRUE THEN
  Mv_Abs_Ex := TRUE;
END_IF;

//StopOn が TRUE になれば、強制停止を実行します。
IF StopOn=TRUE THEN
  Stp_Ex :=TRUE;
END_IF;

// 強制停止(MC_Stop)の出力「Error」、あるいは「CommandAborted」が TRUE のとき、即停止(MC_ImmediateStop)
を実行し即停止します。
IF (Stp_Err=TRUE) OR (Stp_Ca=TRUE) THEN
  Imd_Stp_Ex :=TRUE;
END_IF;

//MC_Power
PWR(
  Axis := MC_Axis000,
  Enable := Pwr_En,
  Status => Pwr_Status,
  Busy => Pwr_Bsy,
  Error => Pwr_Err,
  ErrorID => Pwr_ErrID
);

//MC_Home
HM(
  Axis := MC_Axis000,
  Execute := Hm_Ex,
  Done => Hm_D,
  Busy => Hm_Bsy,
  CommandAborted => Hm_Ca,
  Error => Hm_Err,
  ErrorID => Hm_ErrID
);

//MC_MoveAbsolute
MV_ABS(
  Axis := MC_Axis000,
  Execute := Mv_Abs_Ex,
  Position := Mv_Abs_Pos,
  Velocity := Mv_Abs_Vel,
  Acceleration := Mv_Abs_Acc,
  Deceleration := Mv_Abs_Dec,
  Direction := Mv_Abs_Dir,
  Done => Mv_Abs_D,
  Busy => Mv_Abs_Bsy,
  Active => Mv_Abs_Act,
  CommandAborted => Mv_Abs_Ca,
  Error => Mv_Abs_Err,
  ErrorID => Mv_Abs_ErrID
```

```
);

//MC_Stop
STP(
  Axis := MC_Axis000,
  Execute := Stp_Ex,
  Deceleration := Stp_Dec,
  Done => Stp_D,
  Busy => Stp_Bsy,
  Active => Stp_Act,
  CommandAborted => Stp_Ca,
  Error => Stp_Err,
  ErrorID => Stp_ErrID
);

//MC_ImmediateStop
IMD_STP(
  Axis := MC_Axis000,
  Execute := Imd_Stp_Ex,
  StopMode := Imd_Stp_SM,
  Done => Imd_Stp_D,
  Busy => Imd_Stp_Bsy,
  CommandAborted => Imd_Stp_Ca,
  Error => Imd_Stp_Err,
  ErrorID => Imd_Stp_ErrID
);
```

10-2-8　多軸協調動作中での軸グループ停止

絶対値直線補間（MC_MoveLinearAbsolute）命令を実行中に外部ボタンを ON にすると、強制停止（MC_GroupStop）命令を実行して減速停止させます。軽度フォールトレベルの異常が発生している場合は、軸グループ強制停止（MC_GroupStop）の出力「CommandAborted」が TRUE となります。このとき、軸グループ即停止（MC_GroupImmediateStop）を実行して即停止させます。
また、何らかの理由で軸グループ強制停止（MC_GroupStop）の出力「Error」が TRUE となった場合も、軸グループ即停止（MC_GroupImmediateStop）命令を実行して即停止させます。軸グループ即停止（MC_GroupImmediateStop）を実行すると、軸グループはエラー減速停止中となります。

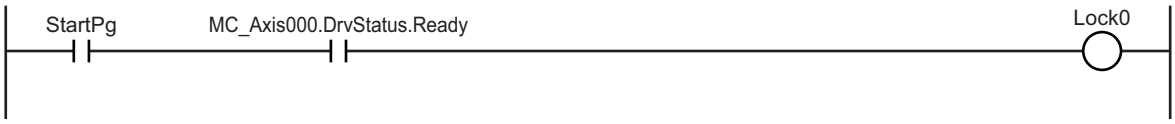
プログラムで使用する主な変数

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.Status.Disabled	BOOL	FALSE	軸グループ 0 が無効のときに TRUE となります。
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_Axis000	_sAXIS_REF	－	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 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_Axis000.Details.Homed	BOOL	FALSE	軸 0 が原点確定状態のとき TRUE となります。
MC_Axis001	_sAXIS_REF	－	軸 1 の軸変数です。

名称	データ型	初期値	コメント
MC_Axis001.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.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき TRUE となります。
Pwr1_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR1 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR2 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
Grp_Stp_Ca	BOOL	FALSE	MC_GroupStop のインスタンス GRP_EN の出力変数 CommandAborted に割り付けた変数です。
Grp_Stp_Err	BOOL	FALSE	MC_GroupStop のインスタンス GRP_EN の出力変数 Error に割り付けた変数です。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
StopOn	BOOL	FALSE	強制停止を行うための外部ボタンの状態です TRUE のとき、MC_GroupStop による強制停止を行います。
InitFlag	BOOL	FALSE	絶対値直線補間命令と軸グループ強制停止の入力パラメータの設定完了を示す変数です。
Grp_En_Ex	BOOL	FALSE	MC_GroupEnable のインスタンス GRP_EN を起動するための変数です。 ST プログラムで使用しています。
Mv_Lin_Abs_Ex	BOOL	FALSE	MC_MoveLinear のインスタンス MV_LIN_ABS を起動するための変数です。 ST プログラムで使用しています。
Grp_Stp_Ex	BOOL	FALSE	MC_GroupStop のインスタンス GRP_STP を起動するための変数です。 ST プログラムで使用しています。
Grp_Imd_Stp_Ex	BOOL	FALSE	MC_GroupImmediateStop のインスタンス GRP_IMD_STP を起動するための変数です。 ST プログラムで使用しています。

ラダー図プログラム

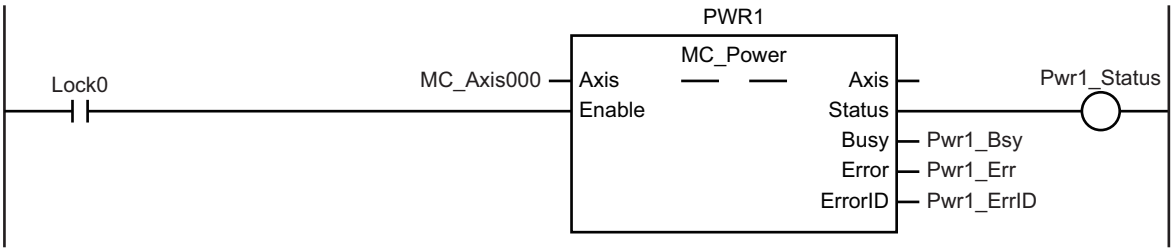
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します



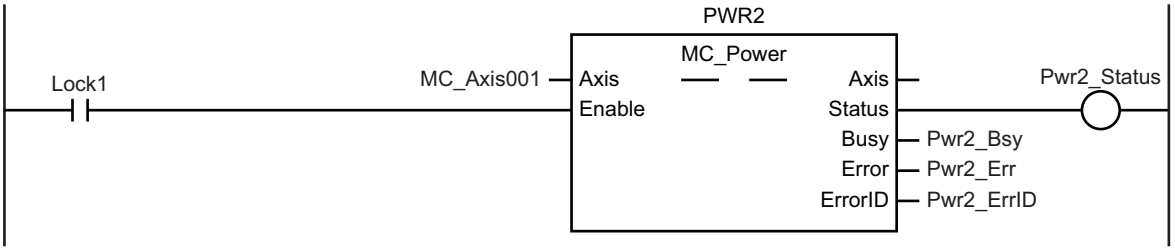
接点 StartPg が TRUE で、軸 1 のサーボドライバがサーボレディ状態か確認します



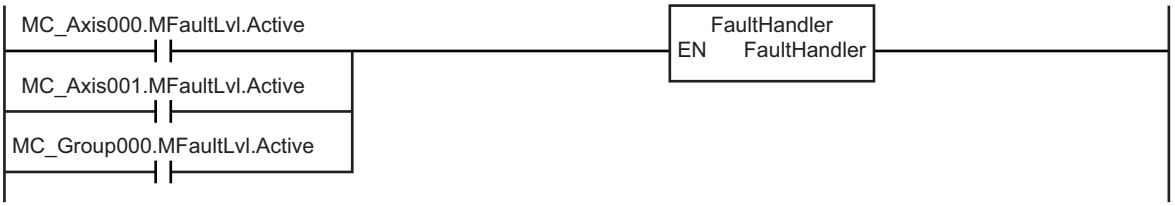
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します



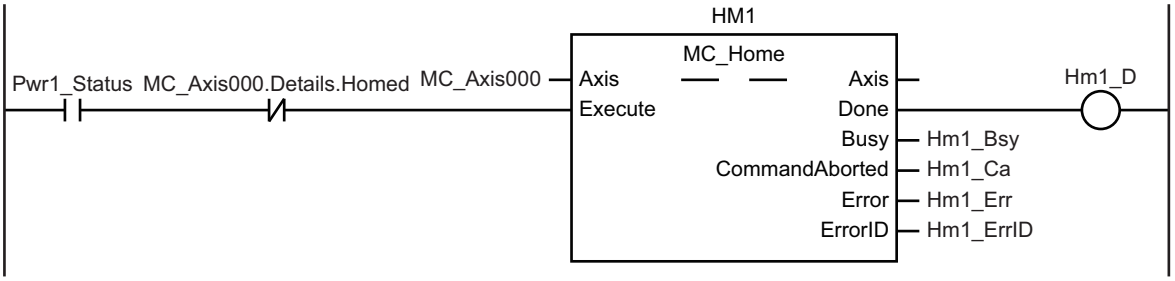
軸 1 のサーボドライバがサーボレディ状態であれば、軸 1 をサーボ ON します



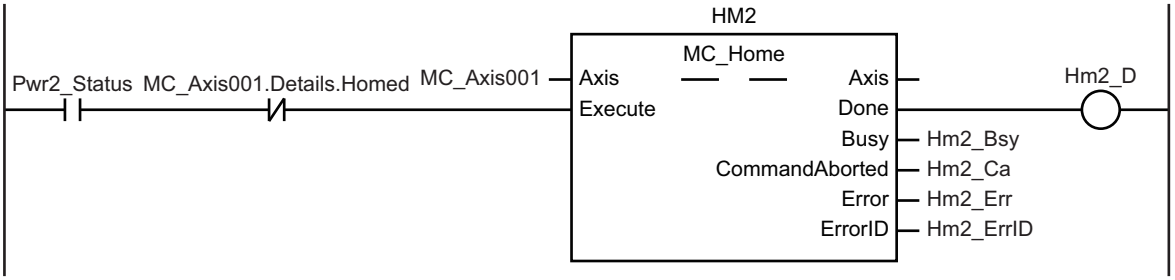
軸グループあるいは構成軸で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。異常処理の内容は装置にあわせてプログラミングしてください



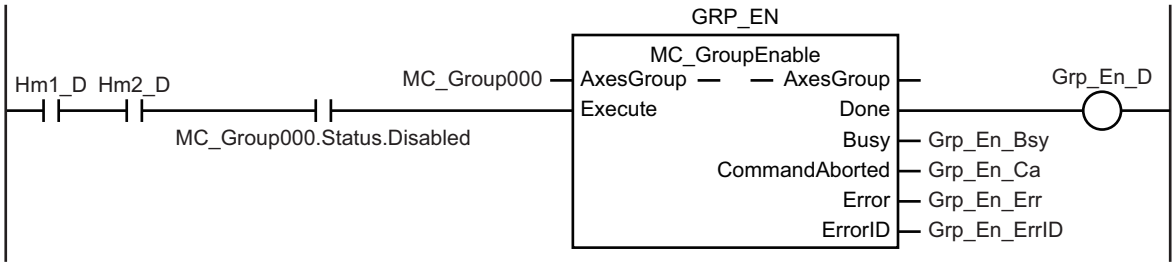
軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します



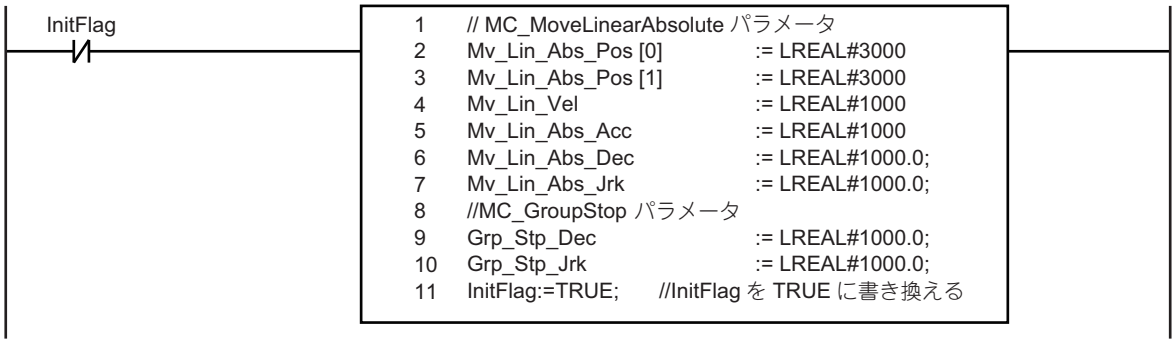
軸 1 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します



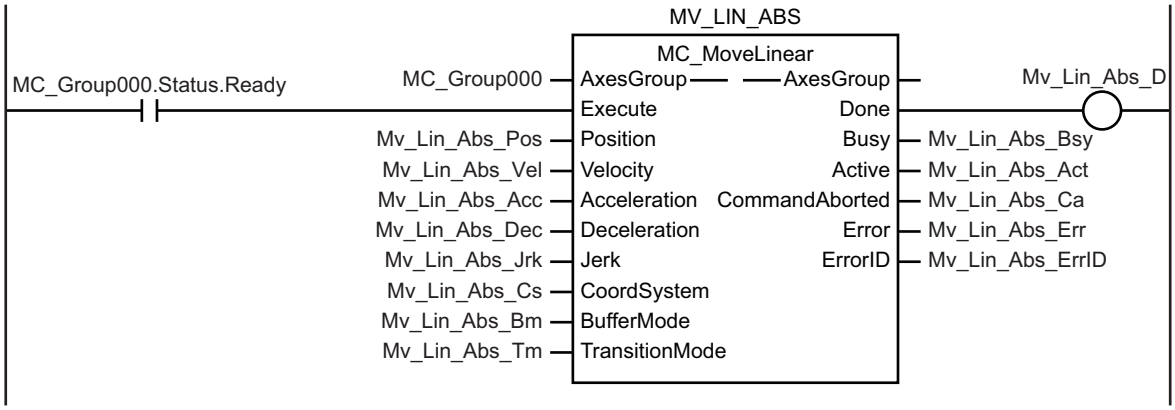
軸 0 と軸 1 が原点確定していれば軸グループを有効にします



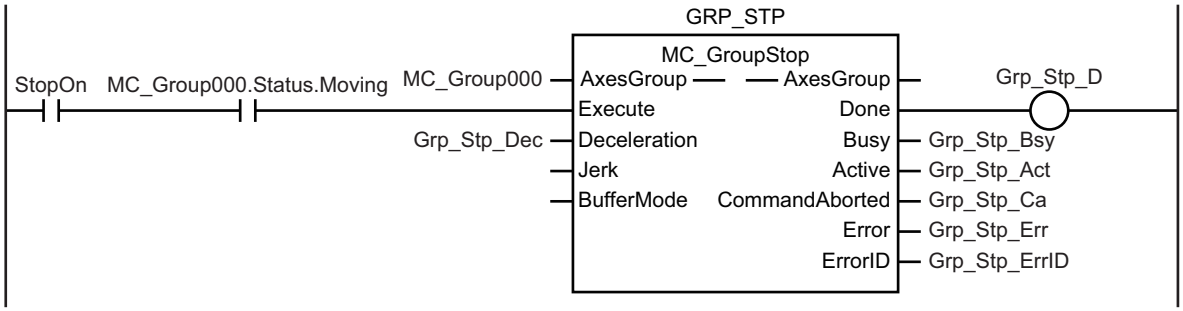
絶対値直線補間と軸グループ強制停止の入力パラメータを設定します



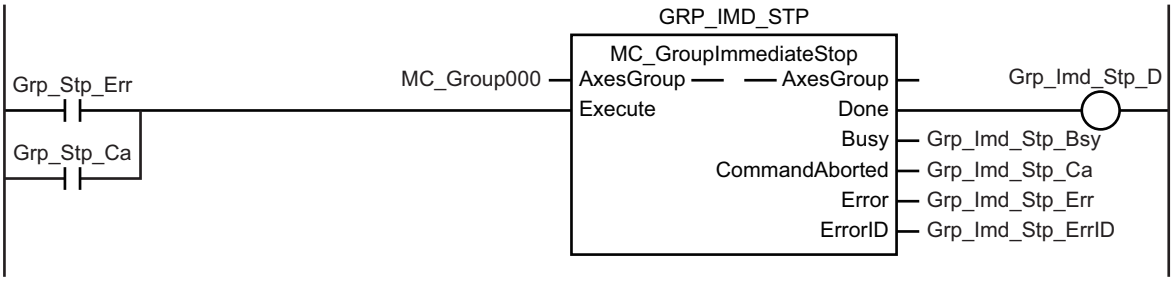
軸グループが有効になったことを確認し、直線補間を実行します



直線補間が動作中に外部ボタン（StopOn）を TRUE にすると、MC_GroupStop を実行し、軸を減速停止します



MC_GroupStop の Error、あるいは CommandAborted が TRUE のとき、MC_GroupImmediateStop を実行し、即停止します



ST プログラム

```
// 絶対値直線補間と軸グループ強制停止の入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag=FALSE THEN
  // 絶対値直線補間(MC_MoveLinearAbsolute)の入力パラメータに値を設定します。
    Mv_Lin_Abs_Pos[0] := LREAL#3000.0;
    Mv_Lin_Abs_Pos[1] := LREAL#3000.0;
    Mv_Lin_Abs_Vel := LREAL#1000.0;
    Mv_Lin_Abs_Acc := LREAL#1000.0;
    Mv_Lin_Abs_Dec := LREAL#1000.0;
    Mv_Lin_Abs_Jrk := LREAL#1000.0;
  // 軸グループ強制停止(MC_GroupStop)の入力パラメータに値を設定します。
    Grp_Stp_Dec := LREAL#1000.0;
    Grp_Stp_Jrk := LREAL#1000.0;
  // 入力パラメータの設定完了フラグを TRUE にします。
    InitFlag := TRUE;
END_IF;

//StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
// サーボレディ状態になっていない場合は、軸 0 をサーボ OFF にします。
IF (StartPg =TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En :=TRUE; // 軸 0 をサーボ ON します。
  ELSE
    Pwr1_En :=FALSE; // 軸 0 をサーボ OFF します。
END_IF;

//StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 1 をサーボ ON します。
// サーボレディ状態になっていない場合は、軸 1 をサーボ OFF にします。
IF (StartPg =TRUE) AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En :=TRUE; // 軸 1 をサーボ ON します。
  ELSE
    Pwr2_En :=FALSE; // 軸 1 をサーボ OFF します。
END_IF;

// 軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせて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 0 がサーボ ON 状態で原点未確定の場合、軸 0 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 1 がサーボ ON 状態で原点未確定の場合、軸 1 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 0 と軸 1 が原点確定したときに軸グループが無効なら、軸グループを有効にします。
IF (MC_Group000.Status.Disabled=TRUE) AND (Hm1_D=TRUE) AND (Hm2_D=TRUE) THEN
  Grp_En_Ex := TRUE;
END_IF;

// 軸グループが有効のとき、絶対値直線補間を実行します。
IF MC_Group000.Status.Ready=TRUE THEN
  Mv_Lin_Abs_Ex :=TRUE;
END_IF;

// 絶対値直線補間を実行中に外部ボタンにより StopOn を TRUE にすれば、強制停止を実行します。
IF (MC_Group000.Status.Moving=TRUE) AND (StopOn=TRUE) THEN
  Grp_Stp_Ex := TRUE;
END_IF;

// 軸グループ強制停止が CommandAborted、あるいは Error になった場合、軸グループを即停止します。
IF (Grp_Stp_Ca=TRUE) OR (Grp_Stp_Err=TRUE) THEN
  Grp_Imd_Stp_Ex :=TRUE;
END_IF;

//MC_Power
PWR1(
  Axis := MC_Axis000,
  Enable := Pwr1_En,
  Status => Pwr1_Status,
  Busy => Pwr1_Bsy,
  Error => Pwr1_Err,
  ErrorID => Pwr1_ErrID
);

PWR2(
  Axis := MC_Axis001,
  Enable := Pwr2_En,
  Status => Pwr2_Status,
  Busy => Pwr2_Bsy,
  Error => Pwr2_Err,
  ErrorID => Pwr2_ErrID
```

```
);

//MC_Home
HM1(
  Axis := MC_Axis000,
  Execute := Hm1_Ex,
  Done => Hm1_D,
  Busy => Hm1_Bsy,
  CommandAborted => Hm1_Ca,
  Error => Hm1_Err,
  ErrorID => Hm1_ErrID
);

HM2(
  Axis := MC_Axis001,
  Execute := Hm2_Ex,
  Done => Hm2_D,
  Busy => Hm2_Bsy,
  CommandAborted => Hm2_Ca,
  Error => Hm2_Err,
  ErrorID => Hm2_ErrID
);

//MC_GroupEnable
GRP_EN(
  AxesGroup := MC_Group000,
  Execute := Grp_En_Ex,
  Done => Grp_En_D,
  Busy => Grp_En_Bsy,
  CommandAborted => Grp_En_Ca,
  Error => Grp_En_Err,
  ErrorID => Grp_En_ErrID
);

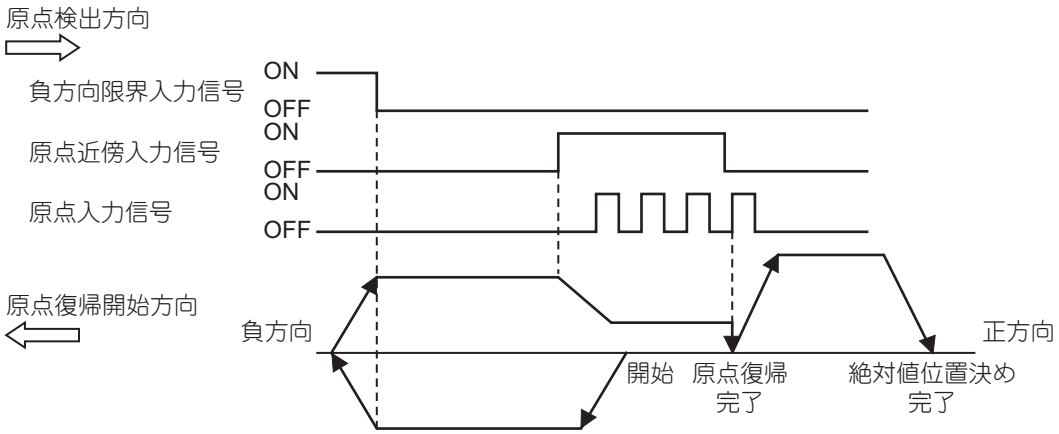
//MC_MoveLinearAbsolute
MV_LIN_ABS(
  AxesGroup := MC_Group000,
  Execute := Mv_Lin_Abs_Ex,
  Position := Mv_Lin_Abs_Pos,
  Velocity := Mv_Lin_Abs_Vel,
  Acceleration := Mv_Lin_Abs_Acc,
  Deceleration := Mv_Lin_Abs_Dec,
  Jerk := Mv_Lin_Abs_Jrk,
  Done => Mv_Lin_Abs_D,
  Busy => Mv_Lin_Abs_Bsy,
  Active => Mv_Lin_Abs_Act,
  CommandAborted => Mv_Lin_Abs_Ca,
  Error => Mv_Lin_Abs_Err,
  ErrorID => Mv_Lin_Abs_ErrID
);
```

```
//MC_GroupStop
GRP_STP(
  AxesGroup := MC_Group000,
  Execute := Grp_Stp_Ex,
  Deceleration := Grp_Stp_Dec,
  Done => Grp_Stp_D,
  Busy => Grp_Stp_Bsy,
  Active => Grp_Stp_Act,
  CommandAborted => Grp_Stp_Ca,
  Error => Grp_Stp_Err,
  ErrorID => Grp_Stp_ErrID
);

//MC_GroupImmediateStop
GRP_IMD_STP(
  AxesGroup := MC_Group000,
  Execute := Grp_Imd_Stp_Ex,
  Done => Grp_Imd_Stp_D,
  Busy => Grp_Imd_Stp_Bsy,
  CommandAborted => Grp_Imd_Stp_Ca,
  Error => Grp_Imd_Stp_Err,
  ErrorID => Grp_Imd_Stp_ErrID
);
```

10-2-9　原点復帰と絶対値位置決め

原点復帰開始位置は、原点近傍入力 ON 上とします。原点復帰は［4: 原点近傍入力 OFF 指定］で行います。原点復帰が正常に完了し原点確定していれば、絶対値位置決めを行います。



原点復帰に関する軸パラメータ設定

設定項目	設定内容	説明
原点復帰動作	4: 原点近傍入力 OFF 指定	原点近傍入力 OFF 指定で原点確定します。
正方向限界入力時動作	1: 反転する・即停止	正方向限界入力を使用しないため初期値のままです。
負方向限界入力時動作	2: 反転する・減速停止	負方向限界入力が有効になると減速停止して反転します。
原点復帰開始方向	2: 負方向	原点復帰を起動すると負方向へ動作開始します。

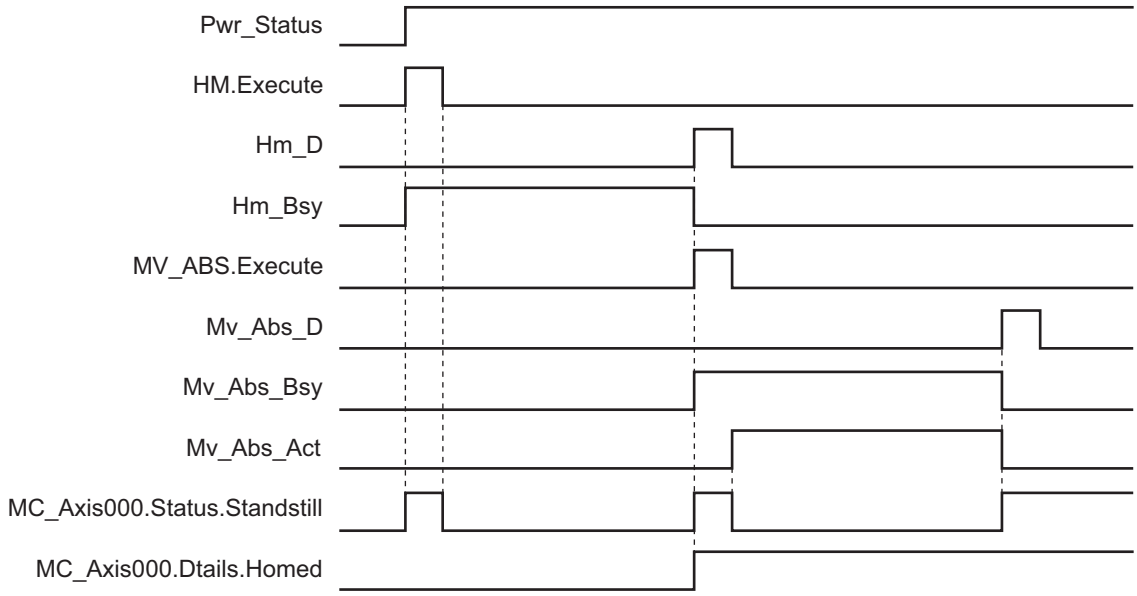
設定項目	設定内容	説明
原点検出方向	1: 正方向	原点を検出するときの動作方向は正方向です。

プログラムで使用する主な変数

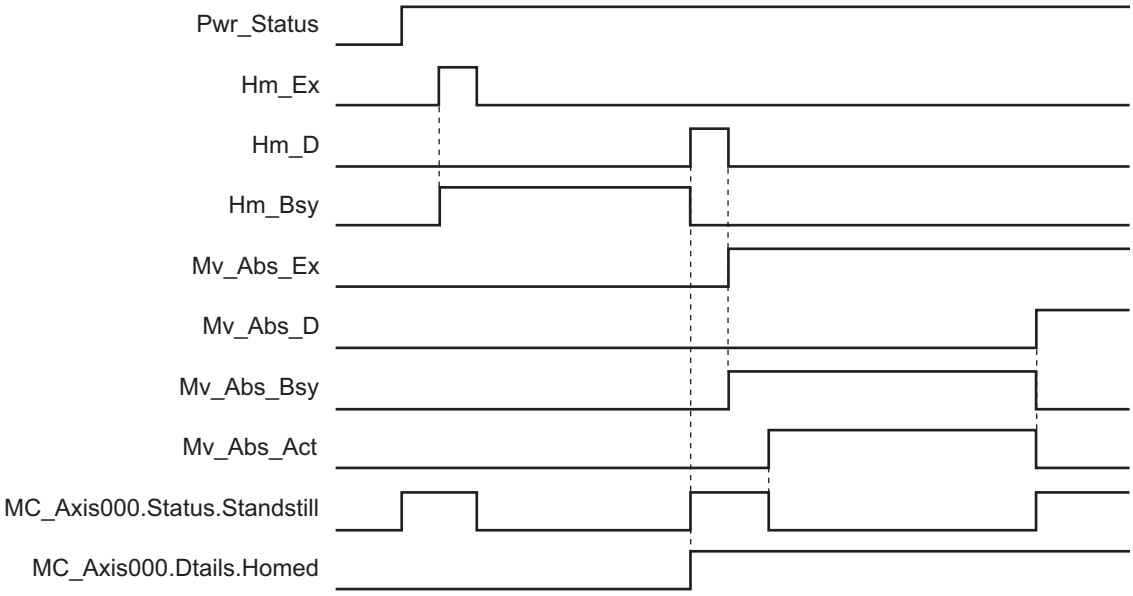
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis000.Status.StandStill	BOOL	FALSE	軸 0 がサーボ ON で停止している状態で TRUE となります。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 0 が原点確定状態のとき TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
Hm_Ex	BOOL	FALSE	MC_Home（原点復帰）命令を起動するための変数です。 ST プログラムで使用しています。
Mv_Abs_Ex	BOOL	FALSE	MC_MoveAbsolute（絶対値位置決め）命令を起動するための変数です。 ST プログラムで使用しています。

タイミングチャート

● ラダー図プログラムの場合

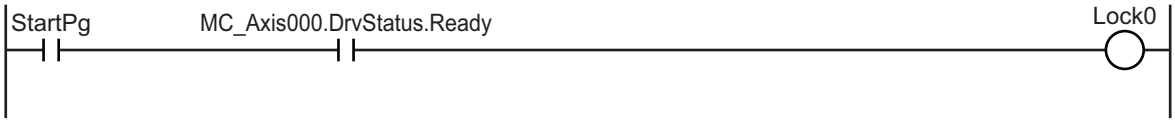


● ST プログラムの場合

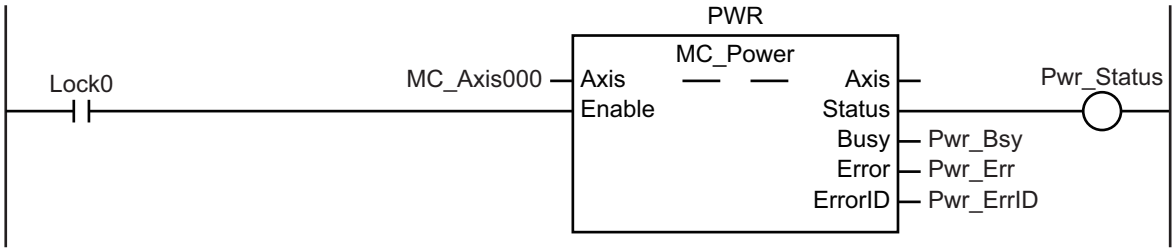


ラダー図プログラム

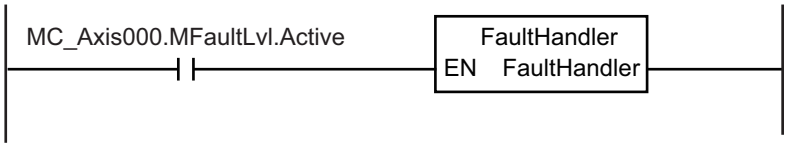
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します



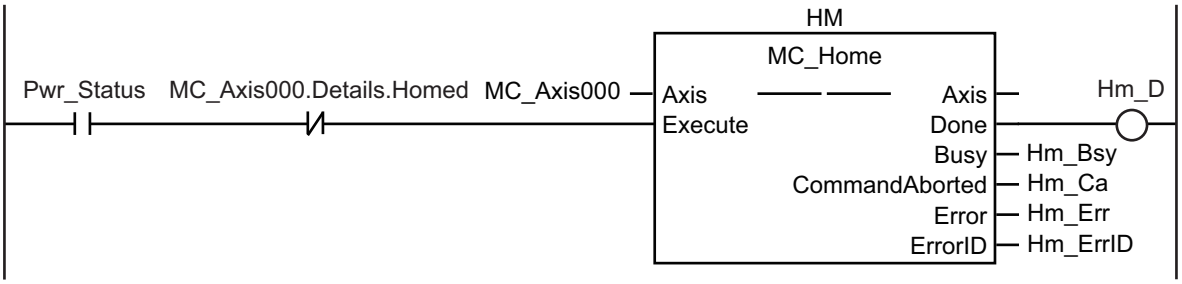
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します



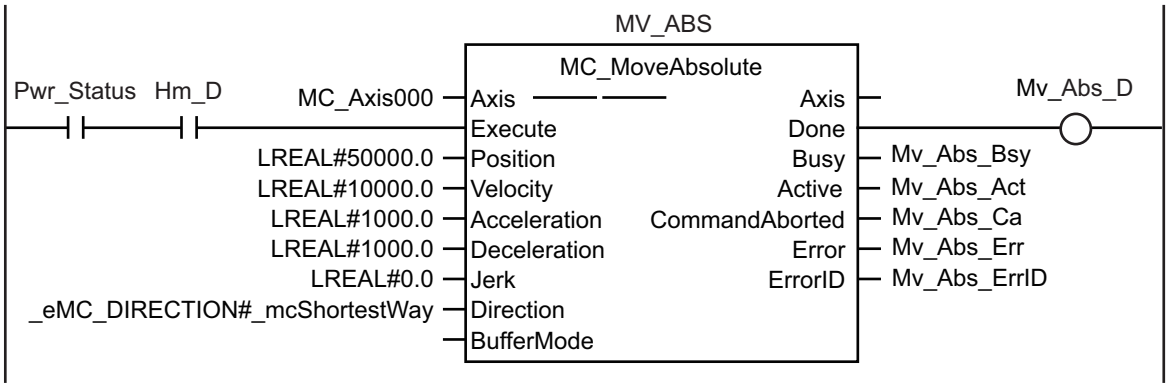
軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します



サーボ ON 状態で原点確定していれば、絶対値位置決めを実行します



ST プログラム

```
// 絶対値位置決めの入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag=FALSE THEN
  // 絶対値位置決め(MC_MoveAbsolute)の入力パラメータに値を設定します。
    Mv_Abs_Pos := LREAL#50000.0;
    Mv_Abs_Vel := LREAL#10000.0;
    Mv_Abs_Acc := LREAL#1000.0;
    Mv_Abs_Dec := LREAL#1000.0;
    Mv_Abs_Dir := _eMC_DIRECTION#_mcShortestWay;
  // 入力パラメータの設定完了フラグを有効にします。
    InitFlag := TRUE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En :=TRUE;
  ELSE
    Pwr_En :=FALSE;
END_IF;

// 軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせてプログラミングしてください。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;
```

```
// 軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。
IF (Pwr_Status=TRUE) AND (MC_Axis000.Details.Homed=FALSE) THEN
    Hm_Ex :=TRUE;
  ELSE
    Hm_Ex :=FALSE;
END_IF;

// サーボ ON 状態で原点確定していれば、絶対値位置決めを実行します。
IF (Pwr_Status=TRUE) AND (Hm_D=TRUE) THEN
  Mv_Abs_Ex :=TRUE;
END_IF;

//MC_Power
PWR(
  Axis := MC_Axis000,
  Enable := Pwr_En,
  Status => Pwr_Status,
  Busy => Pwr_Bsy,
  Error => Pwr_Err,
  ErrorID => Pwr_ErrID
);

//MC_Home
HM(
  Axis := MC_Axis000,
  Execute := Hm_Ex,
  Done => Hm_D,
  Busy => Hm_Bsy,
  CommandAborted => Hm_Ca,
  Error => Hm_Err,
  ErrorID => Hm_ErrID
);

//MC_MoveAbsolute
MV_ABS(
Axis := MC_Axis000,
Execute := Mv_Abs_Ex,
Position := Mv_Abs_Pos,
Velocity := Mv_Abs_Vel,
Acceleration := Mv_Abs_Acc,
Deceleration := Mv_Abs_Dec,
Direction := Mv_Abs_Dir,
Done => Mv_Abs_D,
Busy => Mv_Abs_Bsy,
Active => Mv_Abs_Act,
CommandAborted => Mv_Abs_Ca,
Error => Mv_Abs_Err,
ErrorID => Mv_Abs_ErrID
);
```

10-2-10 命令の再起動により目標位置を変更する

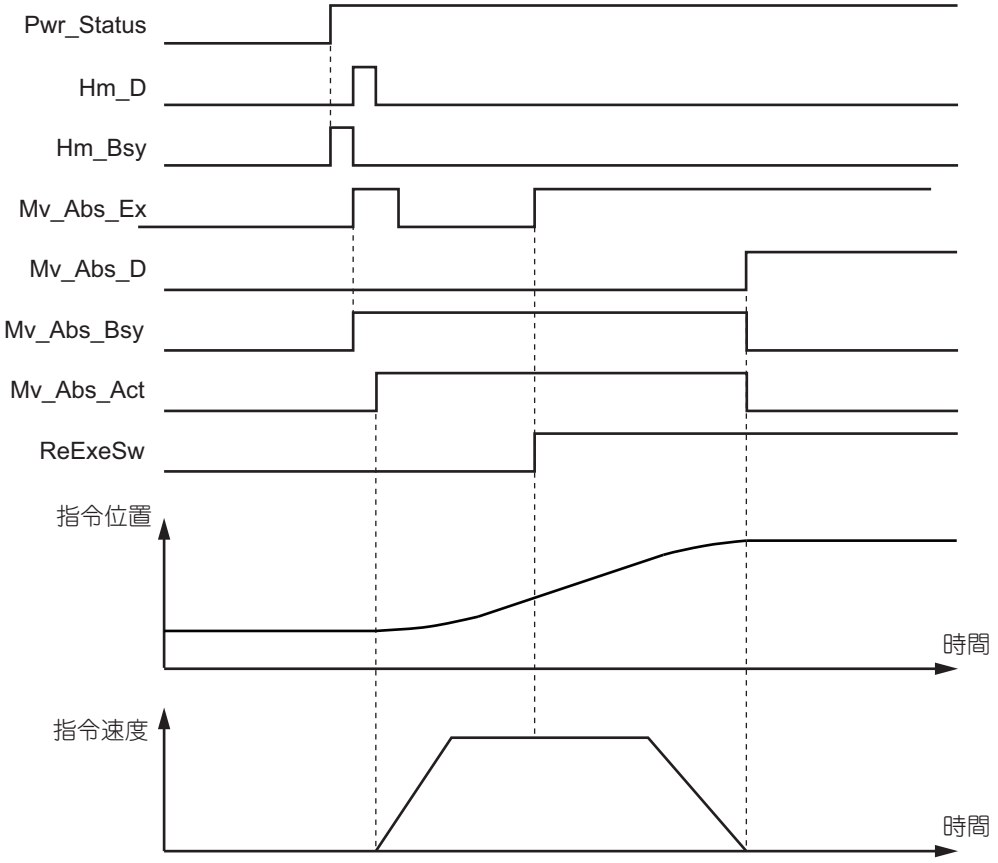
目標位置 1,000 に向けて絶対値位置決め動作を行っている最中に、同じ命令インスタンスを用いて目標位置を 2,000 に変更します。

プログラムで使用する主な変数

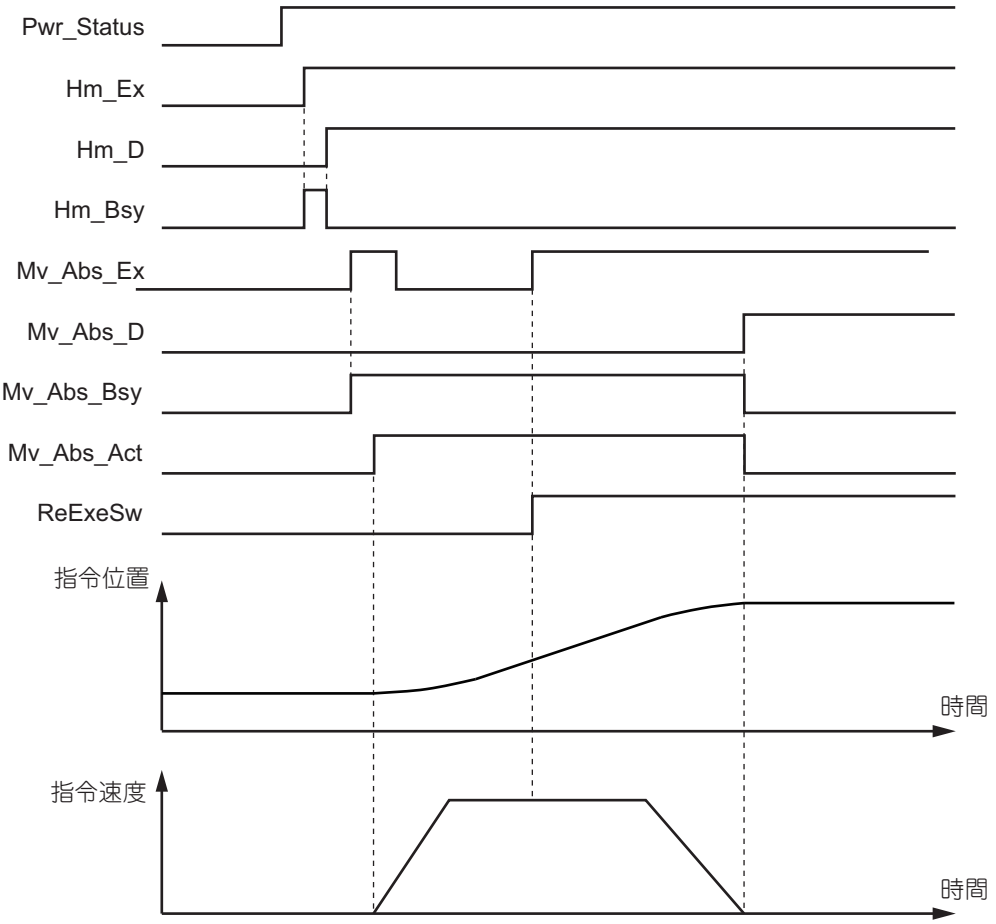
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 0 が原点確定状態のとき TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
ReExeSw	BOOL	FALSE	命令を再起動するための変数です。
Hm_Ex	BOOL	FALSE	MC_Home（原点復帰）命令を起動するための変数です。 ST プログラムで使用しています。
Mv_Abs_Ex	BOOL	FALSE	MC_MoveAbsolute（絶対値位置決め）命令を起動するための変数です。 ST プログラムで使用しています。

タイミングチャート

● ラダー図プログラムの場合



● **ST プログラムの場合**

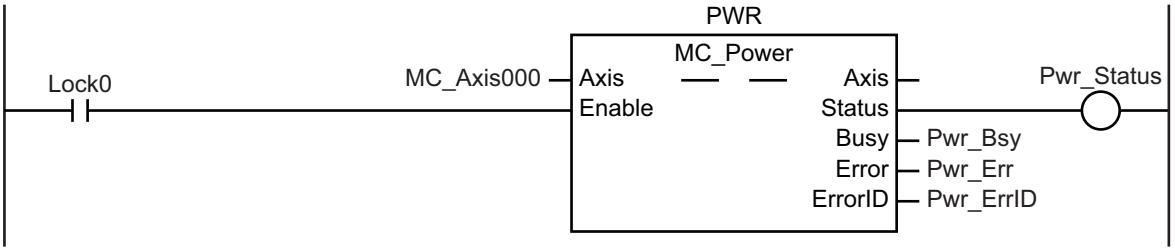


ラダー図プログラム

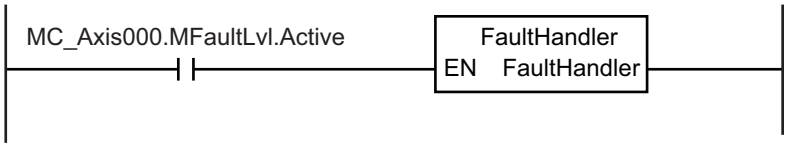
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します



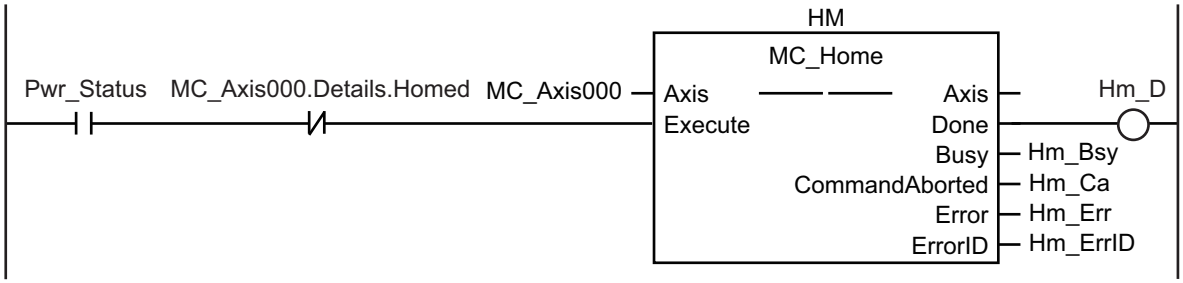
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します



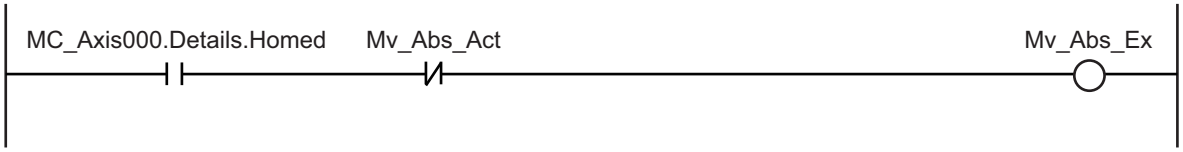
軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します
異常処理の内容は装置にあわせてプログラミングしてください



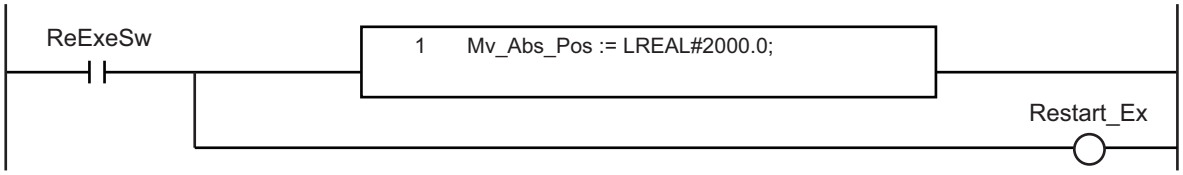
軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



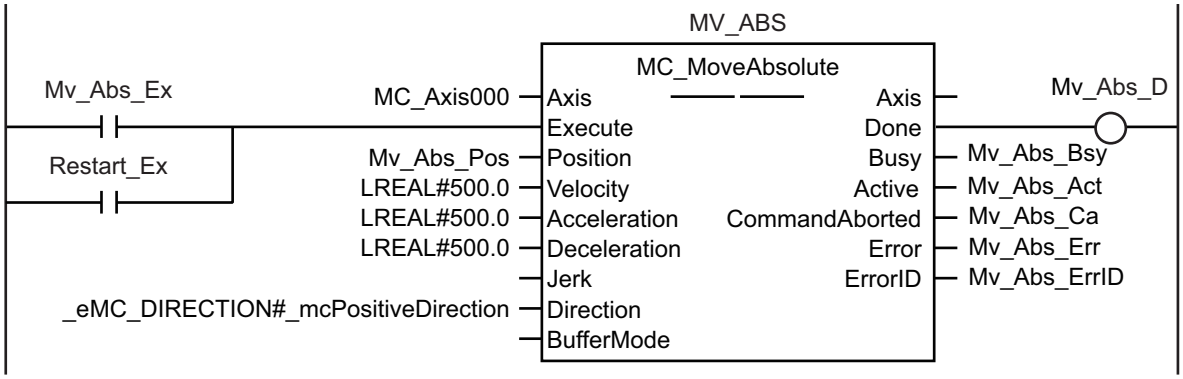
軸 0 が原点確定後、絶対値位置決めが実行されていなければ実行します。



接点 ReExeSw を TRUE にすると、目標位置を 2000 に変更し絶対値位置決めを再起動します



接点 Mv_Abs_Ex の状態により絶対値位置決めを実行します。



ST プログラム

```
// 絶対値位置決めの入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag = FALSE THEN
  //MoveAbsolute パラメータ
    Mv_Abs_Pos := LREAL#1000.0;
    Mv_Abs_Vel := LREAL#500.0;
```

```
    Mv_Abs_Acc := LREAL#500.0;
    Mv_Abs_Dec := LREAL#500.0;
    Mv_Abs_Dir := _eMC_DIRECTION#_mcPositiveDirection;
  // 入力パラメータの設定完了フラグを有効にします。
    InitFlag :=TRUE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En :=TRUE;
  ELSE
    Pwr_En :=FALSE;
END_IF;

// 軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせてプログラミングしてください。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 軸 0 がサーボ ON 状態で原点未確定の場合、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;

// 軸 0 が原点復帰完了後に、絶対値位置決めが実行されていなければ実行します。
IF (Hm_D=TRUE) AND (Mv_Abs_Act=FALSE) THEN
    Mv_Abs_Ex := TRUE;
  ELSE
    Mv_Abs_Ex := FALSE;
END_IF;

// 接点 ReExeSw を TRUE にすると、目標位置を 2000 に変更し絶対値位置決めを再起動します。
IF ReExeSw=TRUE THEN
  Mv_Abs_Pos :=LREAL#2000.0;
  Mv_Abs_Ex := TRUE;
END_IF;

//MC_Power
PWR(
  Axis := MC_Axis000,
  Enable := Pwr_En,
  Status => Pwr_Status,
  Busy => Pwr_Bsy,
  Error => Pwr_Err,
  ErrorID => Pwr_ErrID
);

//MC_Home
HM(
```

```
  Axis := MC_Axis000,
  Execute := Hm_Ex,
  Done => Hm_D,
  Busy => Hm_Bsy,
  CommandAborted => Hm_Ca,
  Error => Hm_Err,
  ErrorID => Hm_ErrID
);

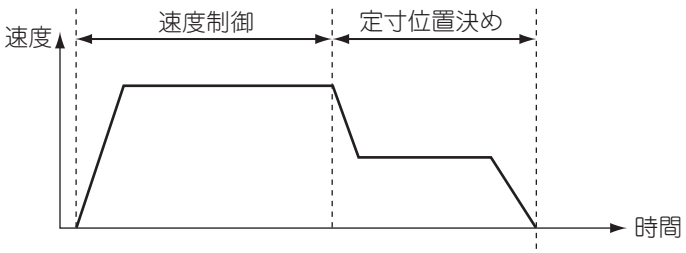
//MC_MoveAbsolute
MV_ABS(
  Axis := MC_Axis000,
  Execute := Mv_Abs_Ex,
  Position := Mv_Abs_Pos,
  Velocity := Mv_Abs_Vel,
  Acceleration := Mv_Abs_Acc,
  Deceleration := Mv_Abs_Dec,
  Direction := Mv_Abs_Dir,
  Done => Mv_Abs_D,
  Busy => Mv_Abs_Bsy,
  Active => Mv_Abs_Act,
  CommandAborted => Mv_Abs_Ca,
  Error => Mv_Abs_Err,
  ErrorID => Mv_Abs_ErrID
);
```

10-2-11 割り込み定寸位置決め

速度制御中に割り込み入力が発生すると、定寸送りに切り替わります。
［ロータリモード］における速度制御のとき、「Direction」（方向選択）では以下のいずれかを指定します。

- _mcPositiveDirection（正方向指定）
- _mcNegativeDirection（負方向指定）
- _mcCurrentDirection（現在方向指定）

このサンプルプログラムでは、_mcCurrentDirection（現在方向指定）を指定します。
割り込み入力の発生前と同じ方向に定寸送りする場合は、入力変数「FeedDistance」（定寸距離）には正数を設定します。割り込み入力の発生前と逆方向に定寸送りする場合には、入力変数「FeedDistance」（定寸距離）に負数を設定します。
たとえば、割り込み入力発生前の動作が負方向で、入力変数「FeedDistance」（定寸距離）に正数を設定すると、負方向に定寸送りを行います。入力変数「FeedDistance」（定寸距離）に負数を設定すると、正方向に定寸送りを行います。



軸パラメータ設定

設定項目	設定内容	説明
カウントモード	ロータリモード	位置のカウントモードをロータリモードに設定します。
リングカウンタ上限設定値	360	リングカウンタの上限値を「360」に設定します。
リングカウンタ下限設定値	0	リングカウンタの下限値を「0」に設定します。
原点復帰動作	原点プリセット	原点プリセットで原点確定します。

プログラムで使用する主な変数

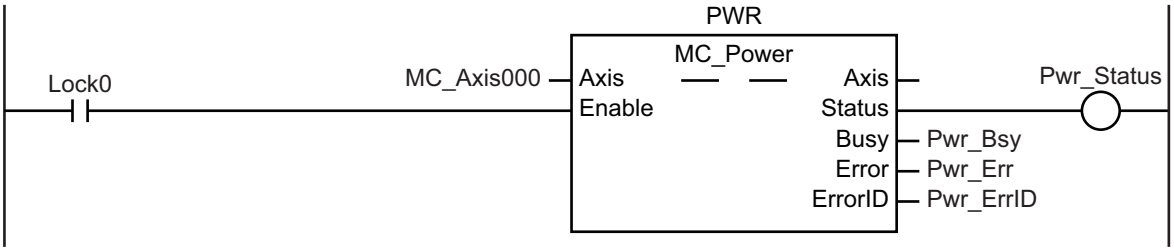
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	－	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 0 が原点確定状態のとき TRUE となります。
Pwr_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
TrigRef	_sTRIGGER_REF	－	割込み入力のトリガ入力条件を指定します。 このサンプルプログラムではサーボドライバのラッチ機能 1 を使用しています。
Hm_Ex	BOOL	FALSE	MC_Home（原点復帰）命令を起動するための変数です。 ST プログラムで使用しています。
Mv_Feed_Ex	BOOL	FALSE	MC_MoveFeed（割り込み定寸位置決め）命令を起動するための変数です。 ST プログラムで使用しています。
InitFlag	BOOL	FALSE	MC_MoveFeed（割り込み定寸位置決め）命令の入力パラメータの設定完了を示す変数です。

ラダー図プログラム

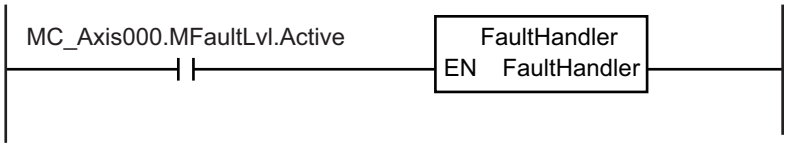
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します。



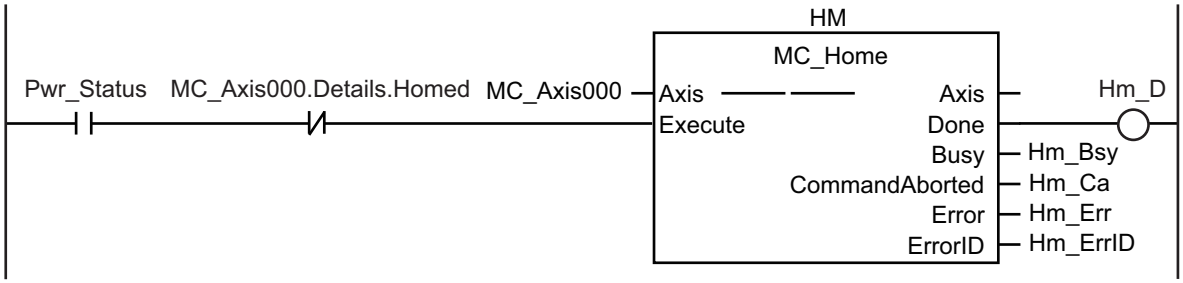
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。



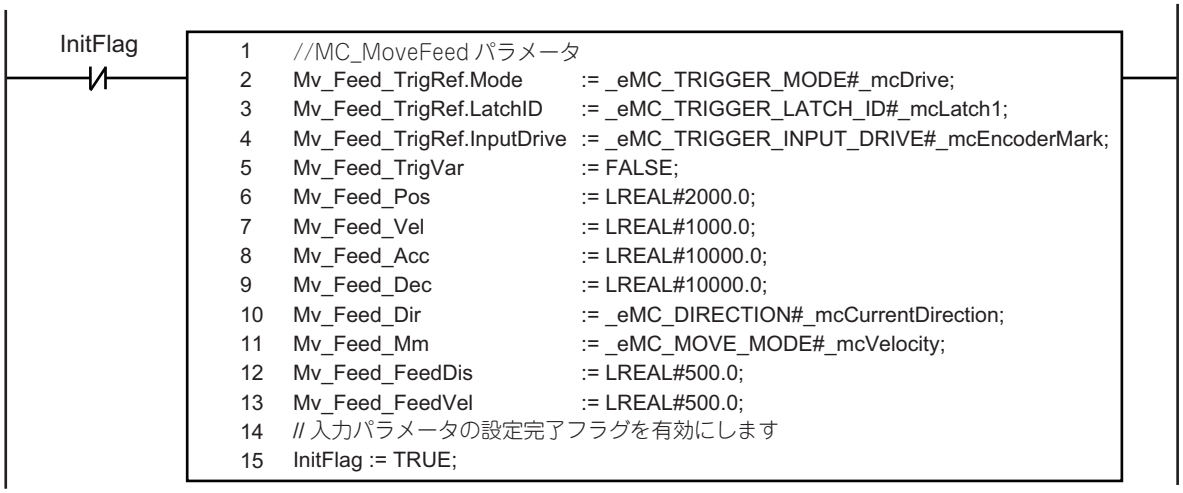
軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



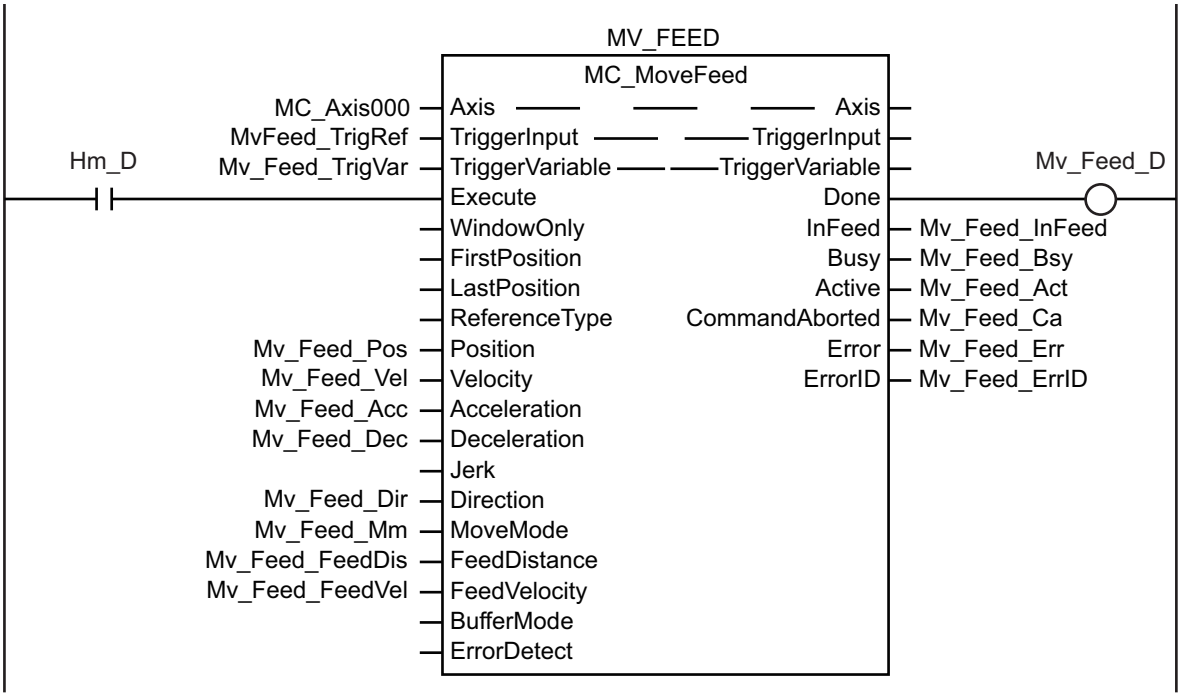
軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



割り込み定寸位置決めの入力パラメータを設定します。



原点復帰が完了すれば割り込み定寸位置決めを実行します。



ST プログラム

```
// 割り込み定寸位置決めの入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag=FALSE THEN
  //MC_MoveFeed 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#_mcEncoderMark;
    Mv_Feed_TrigVar := FALSE;
    Mv_Feed_Pos := LREAL#2000.0;
    Mv_Feed_Vel := LREAL#1000.0;
    Mv_Feed_Acc := LREAL#10000.0;
    Mv_Feed_Dec := LREAL#10000.0;
    Mv_Feed_Dir := _eMC_DIRECTION#_mcCurrentDirection;
    Mv_Feed_Mm := _eMC_MOVE_MODE#_mcVelocity;
    Mv_Feed_FeedDis := LREAL#500.0;
    Mv_Feed_FeedVel := LREAL#500.0;
  // 入力パラメータの設定完了フラグを有効にします。
    InitFlag := TRUE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr_En :=TRUE;
  ELSE
    Pwr_En :=FALSE;
END_IF;
```

```
// 軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせてプログラミングしてください。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 軸 0 がサーボ ON 状態で原点未確定の場合、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;

// 原点確定すれば割り込み定寸位置決めを実行します。
IF Hm_D=TRUE THEN
  Mv_Feed_Ex :=TRUE;
END_IF;

// MC_Power
PWR(
  Axis := MC_Axis000,
  Enable := Pwr_En,
  Status => Pwr_Status,
  Busy => Pwr_Bsy,
  Error => Pwr_Err,
  ErrorID => Pwr_ErrID
);

// MC_Home
HM(
  Axis := MC_Axis000,
  Execute := Hm_Ex,
  Done => Hm_D,
  Busy => Hm_Bsy,
  CommandAborted => Hm_Ca,
  Error => Hm_Err,
  ErrorID => Hm_ErrID
);

//MC_MoveFeed
MV_FEED(
  Axis := MC_Axis000,
  TriggerInput := Mv_Feed_TrigRef,
  TriggerVariable := Mv_Feed_TrigVar,
  Execute := Mv_Feed_Ex,
  Position := Mv_Feed_Pos,
  Velocity := Mv_Feed_Vel,
  Acceleration := Mv_Feed_Acc,
  Deceleration := Mv_Feed_Dec,
  Direction := Mv_Feed_Dir,
  MoveMode := Mv_Feed_Mm,
  FeedDistance := Mv_Feed_FeedDis,
```

```
  FeedVelocity := Mv_Feed_FeedVel,
  Done => Mv_Feed_D,
  InFeed => Mv_Feed_InFeed,
  Busy => Mv_Feed_Bsy,
  Active => Mv_Feed_Act,
  CommandAborted => Mv_Feed_Ca,
  Error => Mv_Feed_Err,
  ErrorID => Mv_Feed_ErrID
);
```

10-2-12 命令の再起動によりカムテーブルを切り替える

カム動作の実行中にカムテーブルを切り替えます。
軸 0 の指令位置が 5,000 以下のときは CamProfile0 を使用し、5,000 を超えると CamTable1 を使用します。

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。

プログラムで使用する主な変数

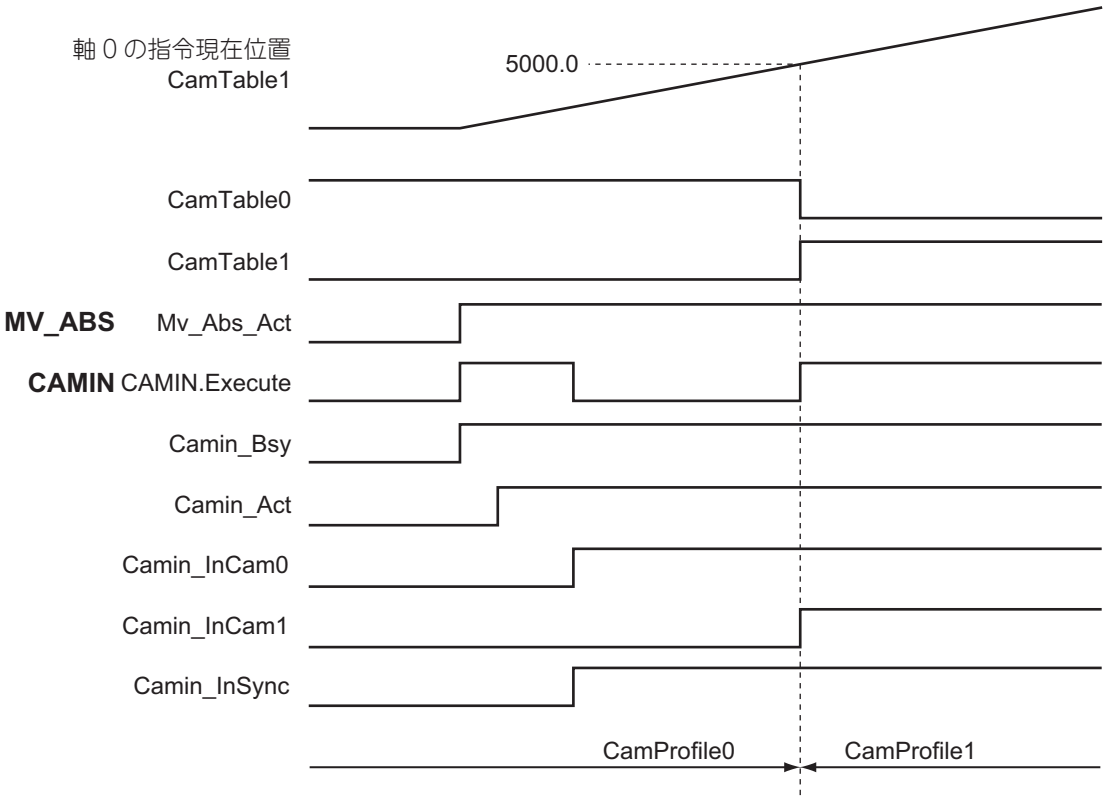
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis001	_sAXIS_REF	−	軸 1 の軸変数です。
CamProfile0	ARRAY[0..100] OF _sMC_CAM_REF	−	カムデータ変数です。*1
CamProfile1	ARRAY[0..10] OF _sMC_CAM_REF	−	カムデータ変数です。*1
Pwr1_S	BOOL	FALSE	MC_Power 命令のインスタンス PWR1 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
Pwr2_S	BOOL	FALSE	MC_Power 命令のインスタンス PWR2 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
CamTable0	BOOL	FALSE	CamProfile0 をカムテーブルとして使用する場合に TRUE にします。
CamTable1	BOOL	FALSE	CamProfile1 をカムテーブルとして使用する場合に TRUE にします。
Camin_InCam0	BOOL	FALSE	MC_CamIn 命令のインスタンス CAMIN の出力 InCam に割り付けた変数です。 CamProfile0 でカム動作中に TRUE となります。 いったん TRUE になると動作終了あるいは中断するまで TRUE を保持します。
Camin_InCam1	BOOL	FALSE	MC_CamIn 命令のインスタンス CAMIN の出力 InCam に割り付けた変数です。 CamProfile1 でカム動作中に TRUE となります。 いったん TRUE になると動作終了あるいは中断するまで TRUE を保持します。

名称	データ型	初期値	コメント
Mv_Abs_Act	BOOL	FALSE	MC_MoveAbsolute 命令のインスタンス MV_ABS の出力 Active に割り付けた変数です。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
Hm1_Ex	BOOL	FALSE	軸 0 で MC_Home（原点復帰）命令を起動するための変数です。
Hm2_Ex	BOOL	FALSE	軸 1 で MC_Home（原点復帰）命令を起動するための変数です。
Mv_Abs_Ex	BOOL	FALSE	MC_MoveAbsolute（絶対値位置決め）命令を起動するための変数です。
Camin_Ex	BOOL	FALSE	MC_CamIn（カム動作開始）命令を起動するための変数です。 ST プログラムで使用しています。

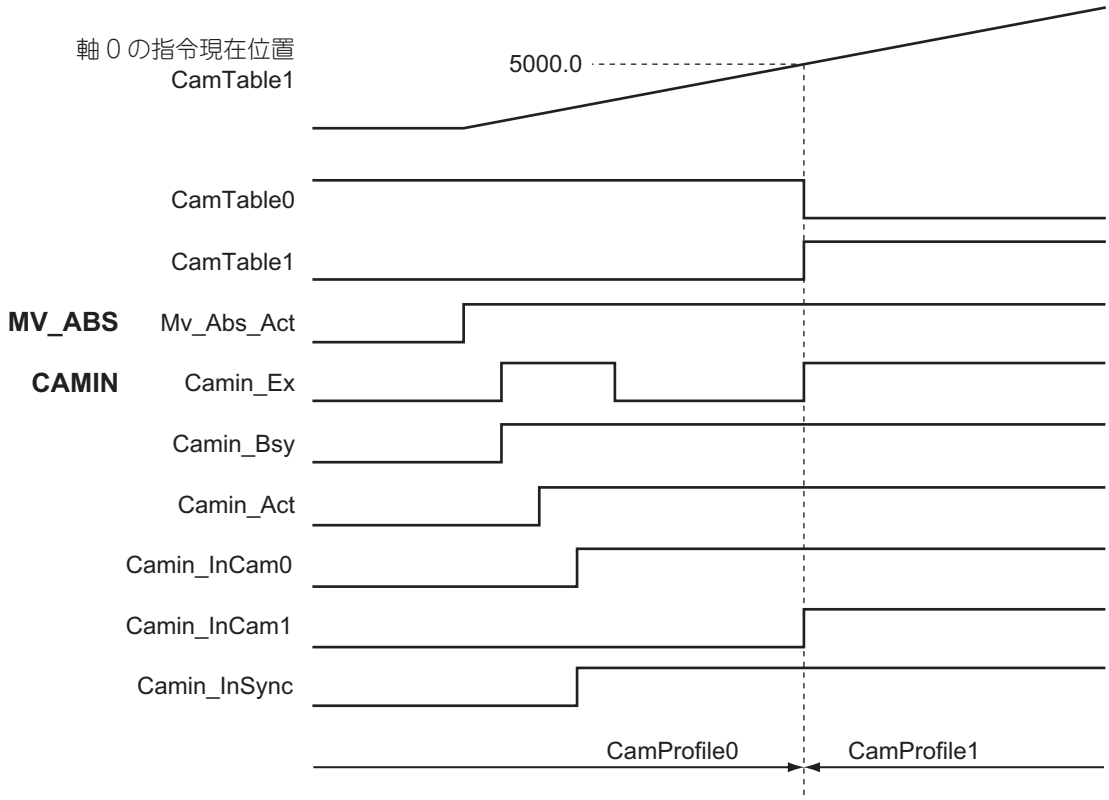
*1. 配列の要素数 ARRAY[0..N]は、Sysmac Studio のカムエディタ機能で設定します。本サンプルプログラムでは[0～109]としています。

タイミングチャート

● ラダー図プログラムの場合



● **ST プログラムの場合**



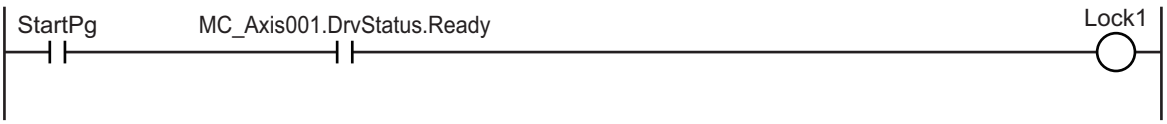
ラダー図プログラム

2 つのカムテーブルを切り替えて使用する場合は、MC_CamIn（カム動作開始）命令の同じ名称のインスタンスを 2 つ配置します。
この場合、それぞれのインスタンスの出力変数「InCam」（カム動作中）には、別の出力パラメータを割り付けます。同じ出力パラメータを割り付けた場合は、異常となります。
このサンプルプログラムでは JMP（ジャンプ）命令により、2 つのインスタンスが同時に実行されないようにしています。

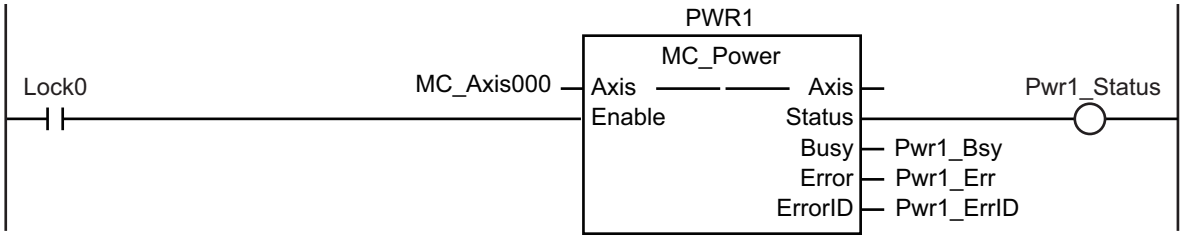
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します。



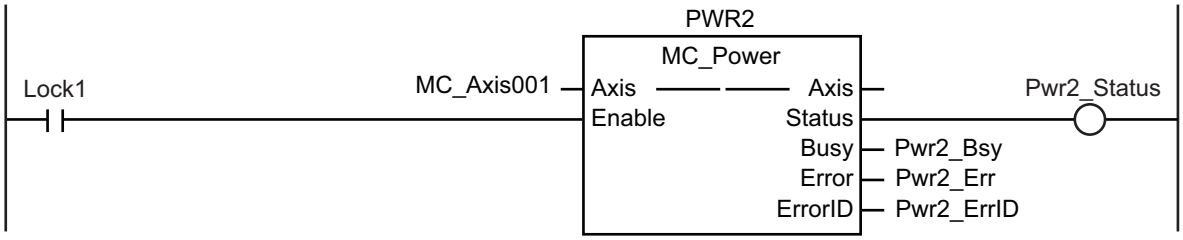
接点 StartPg が TRUE で、軸 1 のサーボドライバがサーボレディ状態か確認します。



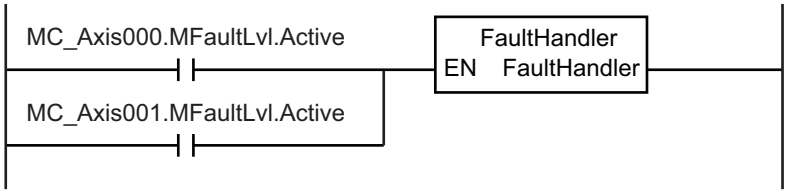
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。



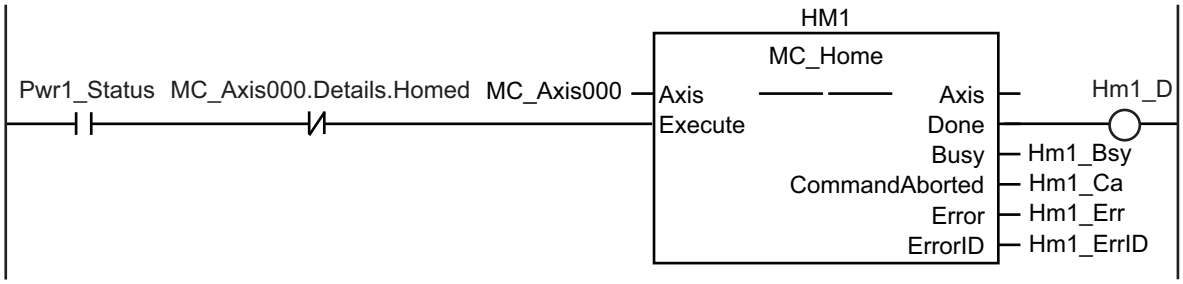
軸 1 のサーボドライバがサーボレディ状態であれば、軸 1 をサーボ ON します。



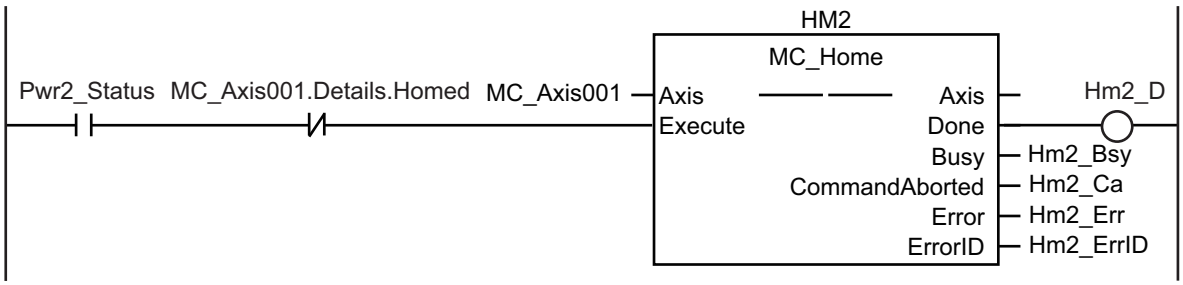
軸 0 または軸 1 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



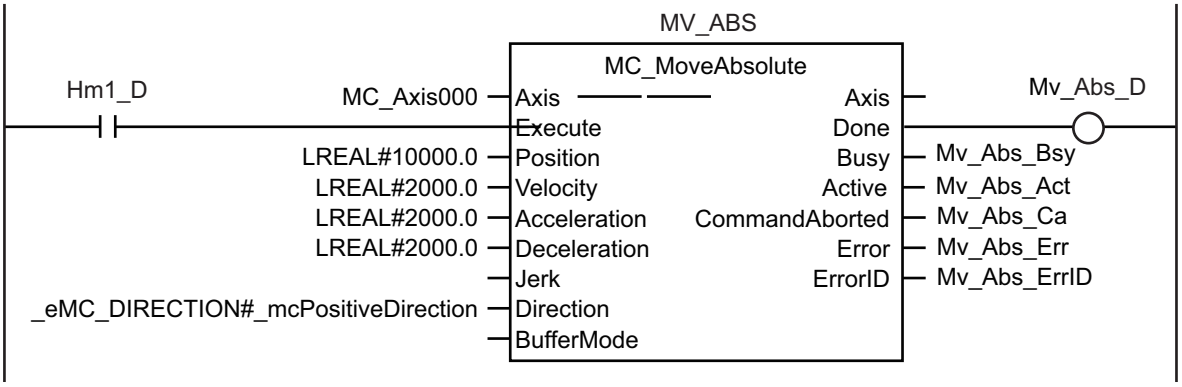
軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



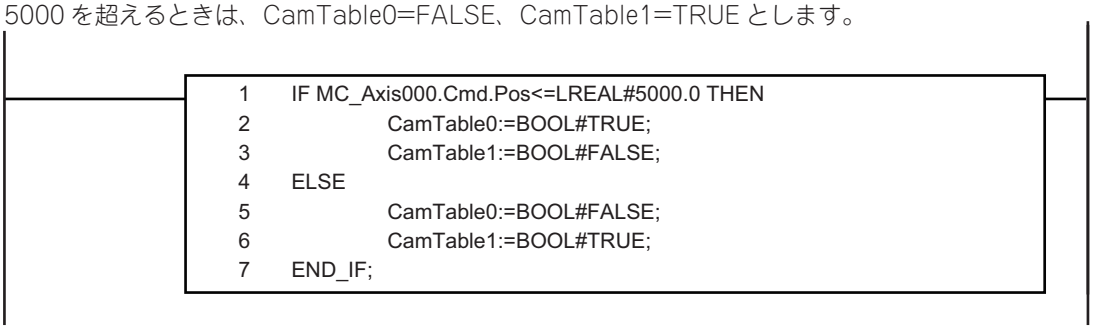
軸 1 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



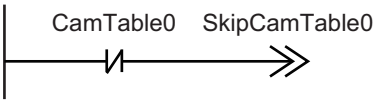
軸 0 の原点復帰が完了すれば、絶対値位置決め (MC_MoveAbsolute) 命令を実行します。

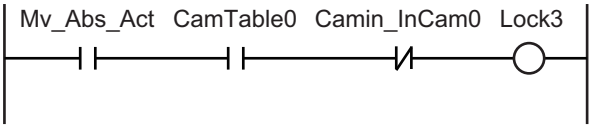


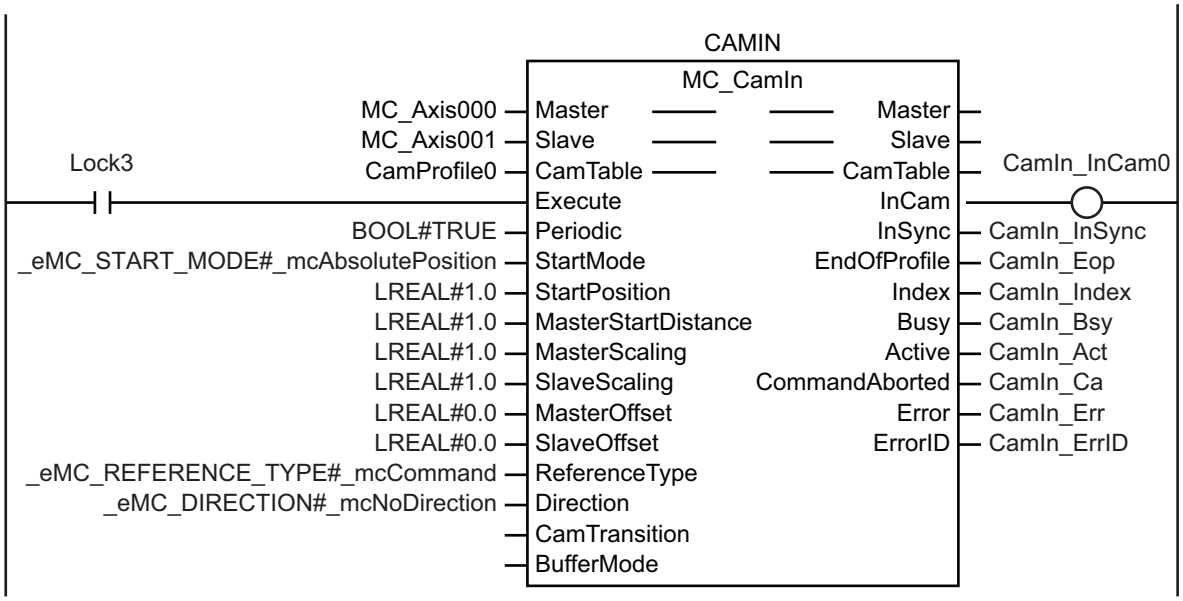
軸 0 の指令位置が 5000 以下のときは、CamTable0=TRUE、CamTable1=FALSE とします。
5000 を超えるときは、CamTable0=FALSE、CamTable1=TRUE とします。



絶対値位置決めが実行中で CamTable0=TRUE の場合は、カムテーブルに CamProfile0 を使用するインスタンスを実行します。
InCam=TRUE のときに、Execute を FALSE とします。

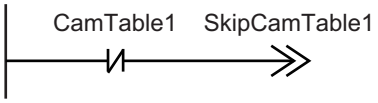


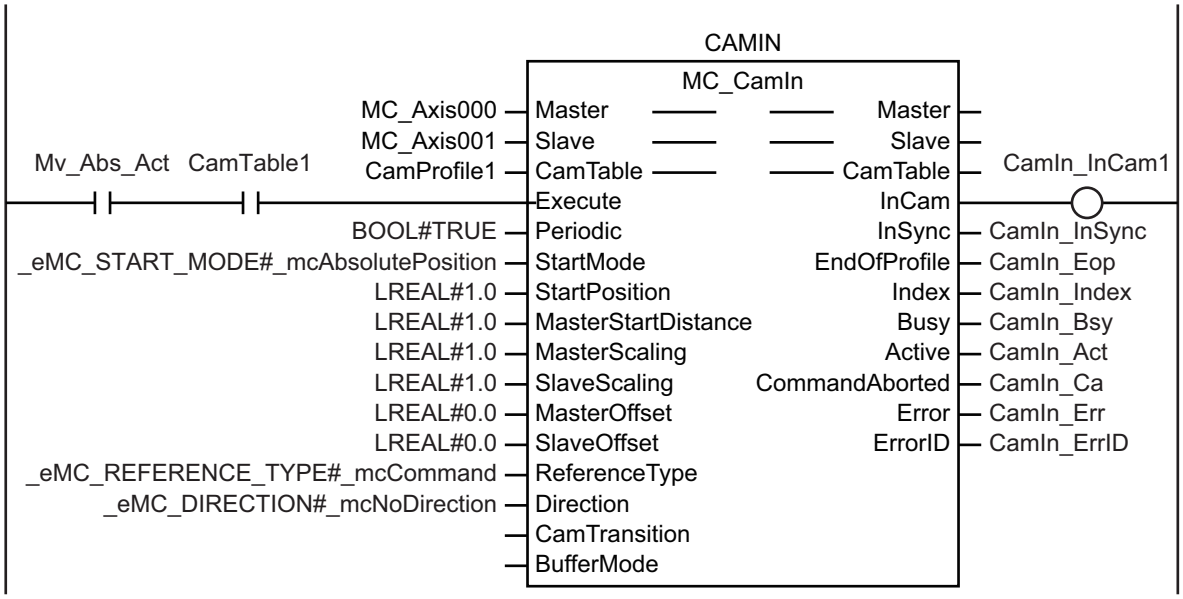




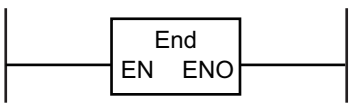
絶対値位置決めが実行中で CamTable1=TRUE の場合は、カムテーブルに CamProfile1 を使用するインスタンスを実行します。

SkipCamTable0





SkipCamTable1



ST プログラム

```
// 絶対値位置決めとカム動作開始の入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag=FALSE THEN
  // 絶対値位置決め(MC_MoveAbsolute)の入力パラメータを設定します。
    Mv_Abs_Pos := LREAL#10000.0;
    Mv_Abs_Vel := LREAL#2000.0;
    Mv_Abs_Acc := LREAL#2000.0;
    Mv_Abs_Dec := LREAL#2000.0;
  // カム動作開始(MC_CamIn)の入力パラメータを設定します。
    Camin_EM := TRUE;
    Camin_StMode := _eMC_START_MODE#_mcAbsolutePosition;
    Camin_StPos := LREAL#1.0;
    Camin_MStDis := LREAL#1.0;
    Camin_MSc := LREAL#1.0;
    Camin_SSc := LREAL#1.0;
    Camin_MO := LREAL#0.0;
    Camin_SO := LREAL#0.0;
    Camin_RT := _eMC_REFERENCE_TYPE#_mcCommand;
    Camin_Dir := _eMC_DIRECTION#_mcNoDirection;
  // カムテーブルの選択
    CamTable0 :=BOOL#TRUE;
    CamTable1 :=BOOL#FALSE;
  // 入力パラメータの設定完了フラグを有効にします。
    InitFlag := TRUE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En :=TRUE;
  ELSE
    Pwr1_En :=FALSE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 1 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En :=TRUE;
  ELSE
    Pwr2_En :=FALSE;
END_IF;

// 軸 0 または軸 1 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせて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 0 がサーボ ON かつ原点未確定の場合、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 1 がサーボ ON かつ原点未確定の場合、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 0 の原点復帰が完了すれば絶対値位置決めを実行します。
IF Hm1_D=TRUE THEN
  Mv_Abs_Ex := TRUE;
END_IF;

// 軸 0 の指令位置が 5000 以下のときは、CamTable0=TRUE、CamTable1=FALSE とします。
//5000 を超えるときは、CamTable0=FALSE、CamTable1=TRUE とします。
IF MC_Axis000.Cmd.Pos<=LREAL#5000.0 THEN
    CamTable0 :=BOOL#TRUE;
    CamTable1 :=BOOL#FALSE;
  ELSE
    CamTable0 :=BOOL#FALSE;
    CamTable1 :=BOOL#TRUE;
END_IF;

// 絶対値位置決めが実行中で CamTable0=TRUE の場合は、カムテーブルに CamProfile0 を使用します。
// インスタンスを実行します。
//InCam=TRUE のときに、Execute を FALSE とします。
IF (Mv_Abs_Act=TRUE)
  AND (CamTable0=TRUE)
  AND (Camin_InCam0=FALSE) THEN
    Camin_Ex := TRUE;
  ELSE
    Camin_Ex := FALSE;
END_IF;

// 絶対値位置決めが実行中で CamTable1=TRUE の場合は、カムテーブルに CamProfile1 を使用します。
// インスタンスを実行します。
IF (Mv_Abs_Act=TRUE) AND (CamTable1=TRUE) THEN
  Camin_Ex := TRUE;
END_IF;

//MC_Camin
IF CamTable0=TRUE THEN
  CAMIN(
    Master := MC_Axis000,
    Slave := MC_Axis001,
    CamTable := CamProfile0,
    Execute := Camin_Ex,
    Periodic := Camin_EM,
    StartMode := Camin_StMode,
    StartPosition := Camin_StPos,
```

```
    MasterStartDistance := Camin_MStDis,
    MasterScaling := Camin_MSc,
    SlaveScaling := Camin_SSc,
    MasterOffset := Camin_MO,
    SlaveOffset := Camin_SO,
    ReferenceType := Camin_RT,
    Direction := Camin_Dir,
    InCam => Camin_InCam0,
    InSync => Camin_InSync,
    EndOfProfile => Camin_EOP,
    Index => Camin_Index,
    Busy => Camin_Bsy,
    Active => Camin_Act,
    CommandAborted => Camin_Ca,
    Error => Camin_Err,
    ErrorID => Camin_ErrID
  );
END_IF;

IF CamTable1=TRUE THEN
  CAMIN(
    Master := MC_Axis000,
    Slave := MC_Axis001,
    CamTable := CamProfile1,
    Execute := Camin_Ex,
    Periodic := Camin_EM,
    StartMode := Camin_StMode,
    StartPosition := Camin_StPos,
    MasterStartDistance := Camin_MStDis,
    MasterScaling := Camin_MSc,
    SlaveScaling := Camin_SSc,
    MasterOffset := Camin_MO,
    SlaveOffset := Camin_SO,
    ReferenceType := Camin_RT,
    Direction := Camin_Dir,
    InCam => Camin_InCam1,
    InSync => Camin_InSync,
    EndOfProfile => Camin_EOP,
    Index => Camin_Index,
    Busy => Camin_Bsy,
    Active => Camin_Act,
    CommandAborted => Camin_Ca,
    Error => Camin_Err,
    ErrorID => Camin_ErrID
  );
END_IF;

// 軸 0 の MC_Power
PWR1(
  Axis := MC_Axis000,
  Enable := Pwr1_En,
```

```
  Status => Pwr1_S,
  Busy => Pwr1_Bsy,
  Error => Pwr1_Err,
  ErrorID => Pwr1_ErrID
);

// 軸 1 の MC_Power
PWR2(
  Axis := MC_Axis001,
  Enable := Pwr2_En,
  Status => Pwr2_S,
  Busy => Pwr2_Bsy,
  Error => Pwr2_Err,
  ErrorID => Pwr2_ErrID
);

// 軸 0 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
);

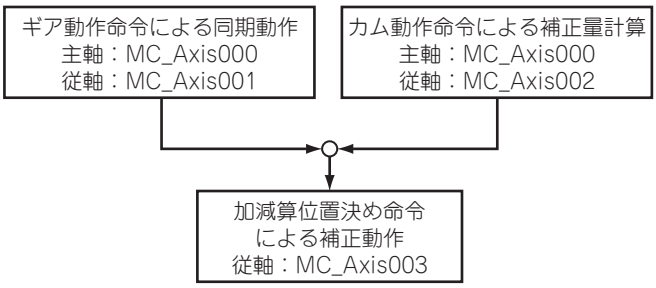
// 軸 1 の MC_Home
HM2(
  Axis := MC_Axis001,
  Execute := Hm2_Ex,
  Done => Hm2_D,
  Busy => Hm2_Bsy,
  CommandAborted => Hm2_Ca,
  Error => Hm2_Err,
  ErrorID => Hm2_ErrID
);

//MC_MoveAbsolute
MV_ABS(
Axis := MC_Axis000,
Execute := Mv_Abs_Ex,
Position := Mv_Abs_Pos,
Velocity := Mv_Abs_Vel,
Acceleration := Mv_Abs_Acc,
Deceleration := Mv_Abs_Dec,
Direction := Mv_Abs_Dir,
Done => Mv_Abs_D,
Busy => Mv_Abs_Bsy,
Active => Mv_Abs_Act,
CommandAborted => Mv_Abs_Ca,
```

```
Error => Mv_Abs_Err,
ErrorID => Mv_Abs_ErrID
);
```

10-2-13 カムプロファイル曲線を使って同期位置を補正する

ギア動作を行う従軸に対し、カムプロファイル曲線を用いて補正を行います。
ギア動作の従軸は仮想サーボ軸の MC_Axis001 で、カム動作の従軸は仮想サーボ軸の MC_Axis002 です。これらの従軸を MC_CombineAxes により加算し、その結果をサーボ軸の MC_Axis003 に出力します。主軸はサーボ軸の MC_Axis000 です。
処理の流れは以下のとおりです。



● 軸種別の設定

軸種別は各軸の軸パラメータでを以下のように設定します。

設定項目	設定内容			
	軸 1	軸 2	軸 3	軸 4
軸変数名	MC_Axis000	MC_Axis001	MC_Axis002	MC_Axis003
軸種別	サーボ軸	仮想サーボ軸	仮想サーボ軸	サーボ軸

プログラムで使用する主な変数

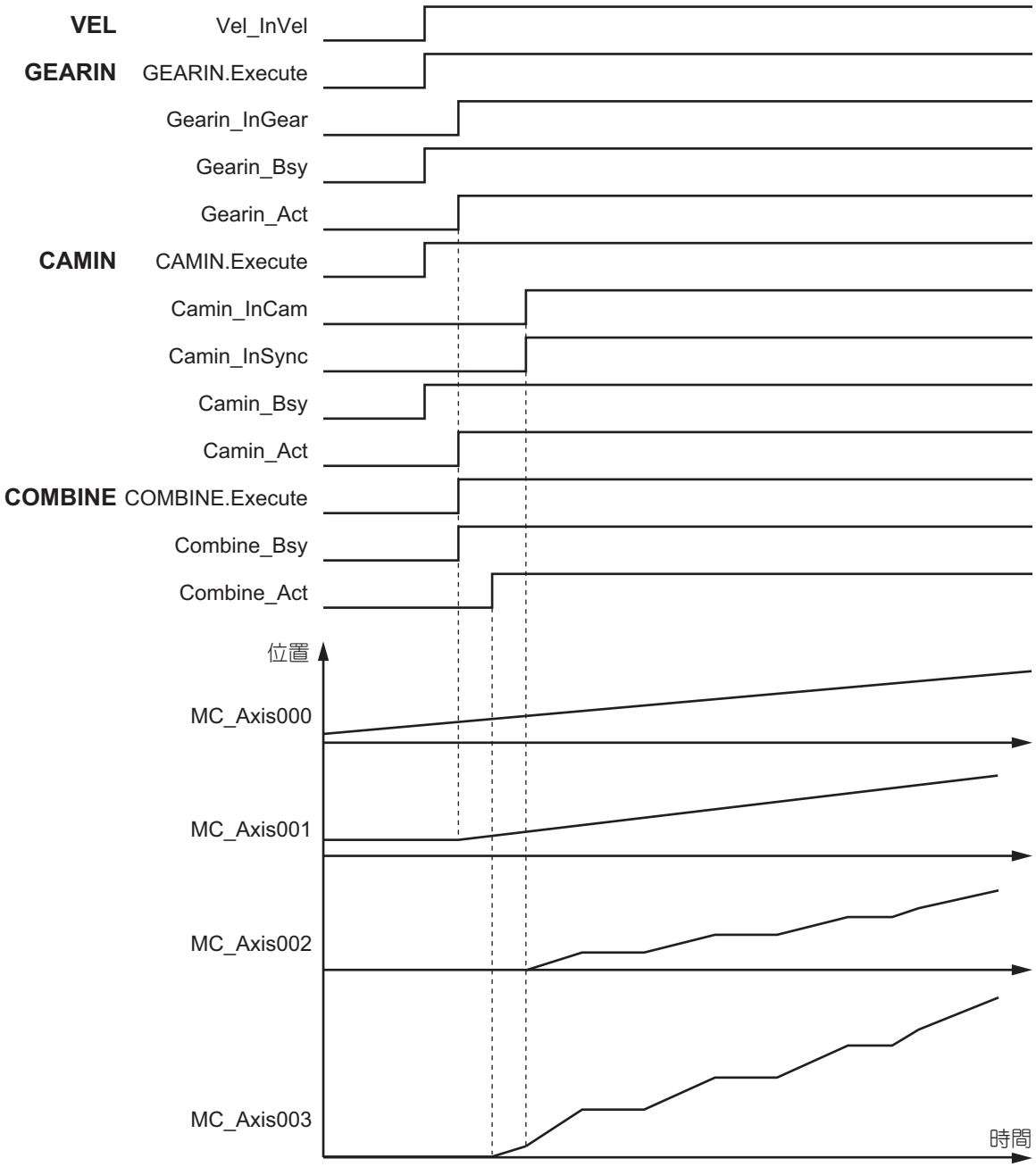
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 0 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
MC_Axis000.Details.Homed	BOOL	FALSE	軸 0 が原点確定状態のとき TRUE となります。
MC_Axis001	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis001.MFaultLvl.Active	BOOL	FALSE	軸 1 で軽度フォールトレベルの異常が発生した場合に、TRUE となります。
MC_Axis002	_sAXIS_REF	−	軸 2 の軸変数です。
MC_Axis002.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_Axis003	_sAXIS_REF	−	軸 3 の軸変数です。
MC_Axis003.MFaultLvl.Active	BOOL	FALSE	軸 3 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_Axis003.Details.Homed	BOOL	FALSE	軸 3 が原点確定状態のとき TRUE となります。
CamProfile0	ARRAY[0..109] OF _sMC_CAM_REF	−	カムデータ変数です。*1
Pwr1_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR1 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。

名称	データ型	初期値	コメント
Pwr4_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR4 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
Vel_InVel	BOOL	FALSE	軸 0 の MC_MoveVelocity が目標速度に到達した場合に、TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
Gearin_Ex	BOOL	FALSE	MC_GearIn（ギア動作開始）命令を起動するための変数です。*2
Camin_Ex	BOOL	FALSE	MC_CamIn（カム動作開始）命令を起動するための変数です。*2
Combine_Ex	BOOL	FALSE	MC_CombineAxes（加減算位置決め）命令を起動するための変数です。*2

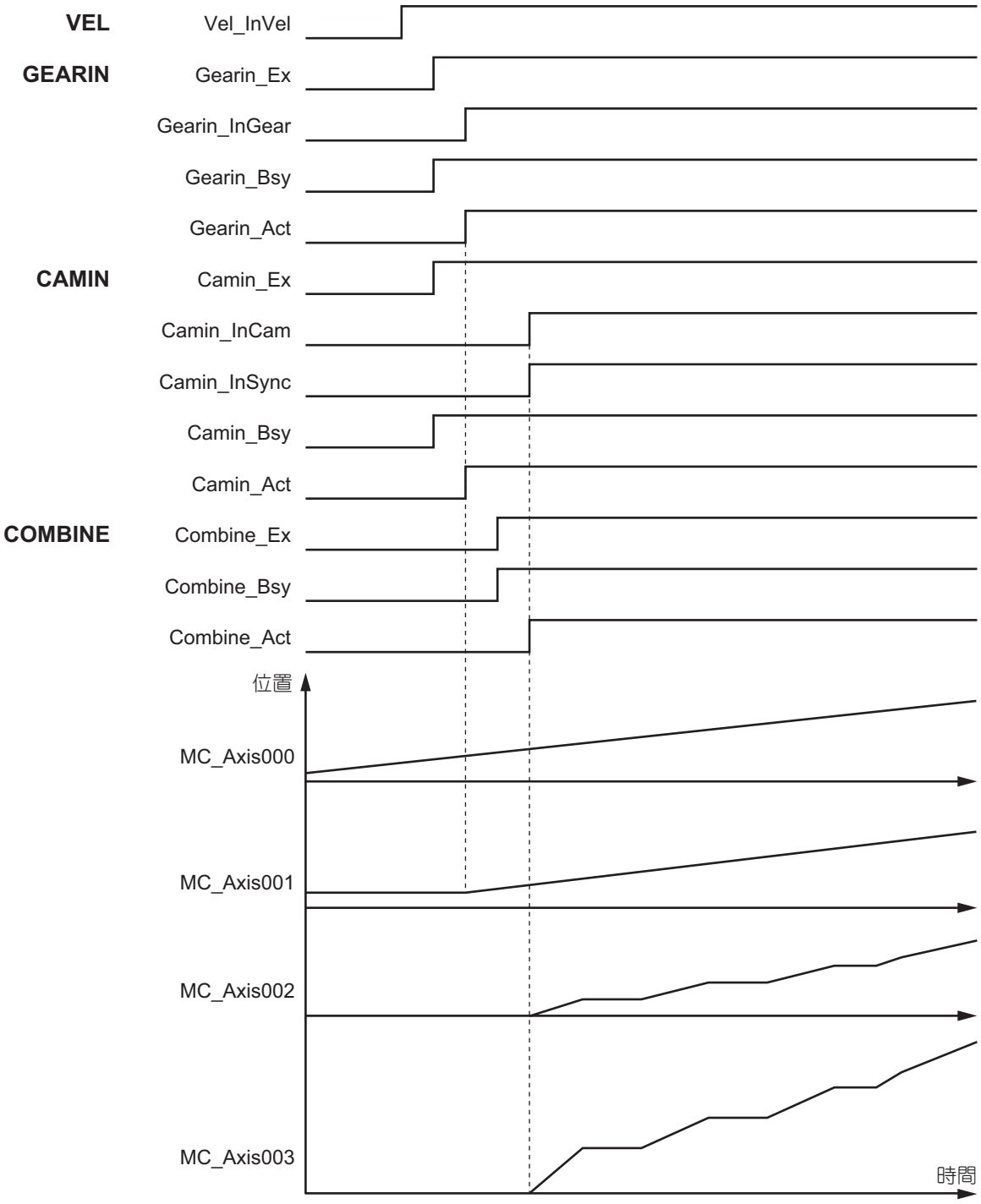
*1. 配列の要素数 ARRAY[0..N]は、Sysmac Studio のカムエディタ機能で設定します。本サンプルプログラムでは[0～109]としています。
*2. ST プログラムで使用しています。

タイミングチャート

● ラダー図プログラムの場合

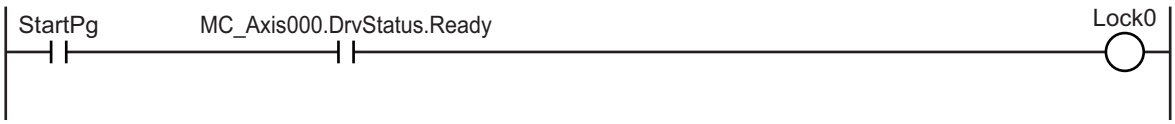


● **ST プログラムの場合**

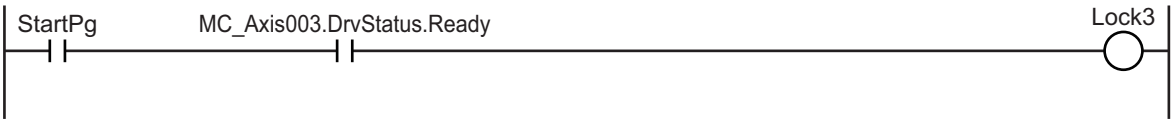


ラダー図プログラム

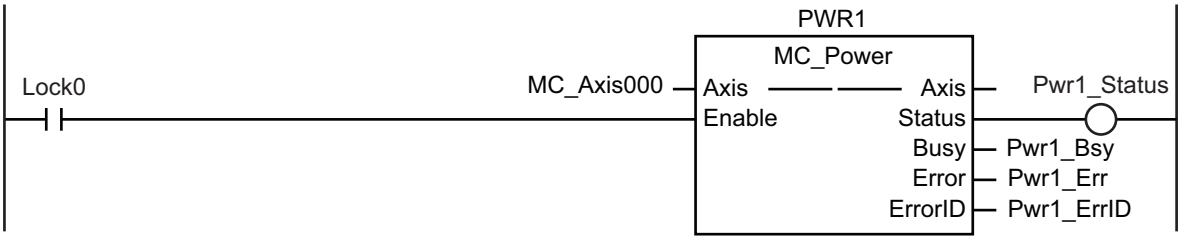
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します。



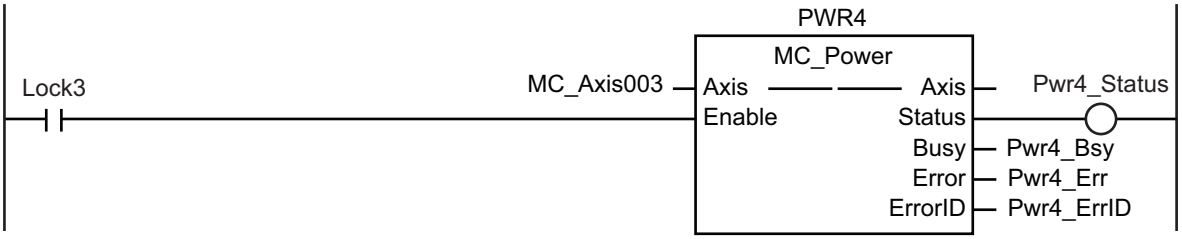
接点 StartPg が TRUE で、軸 3 のサーボドライバがサーボレディ状態か確認します。



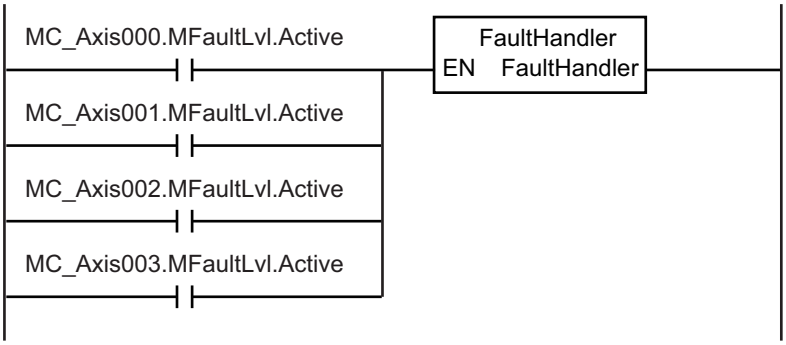
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。



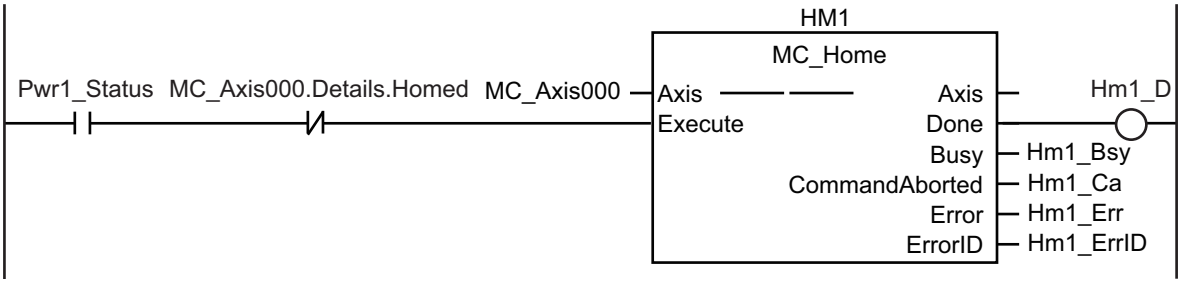
軸 3 のサーボドライバがサーボレディ状態であれば、軸 3 をサーボ ON します。



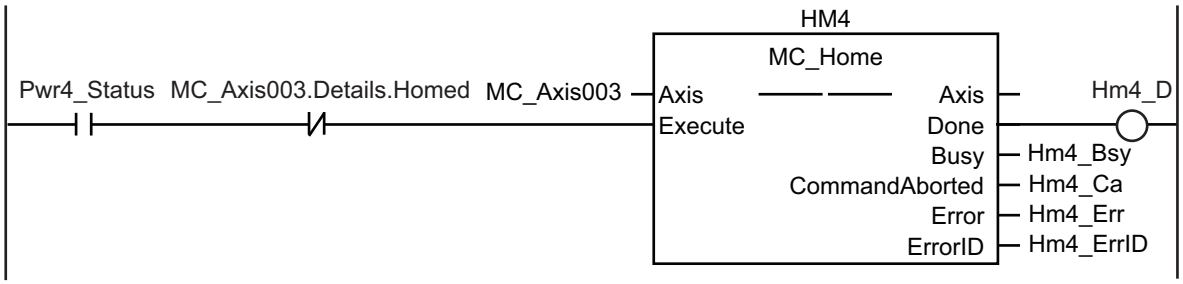
軸グループあるいは構成軸で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



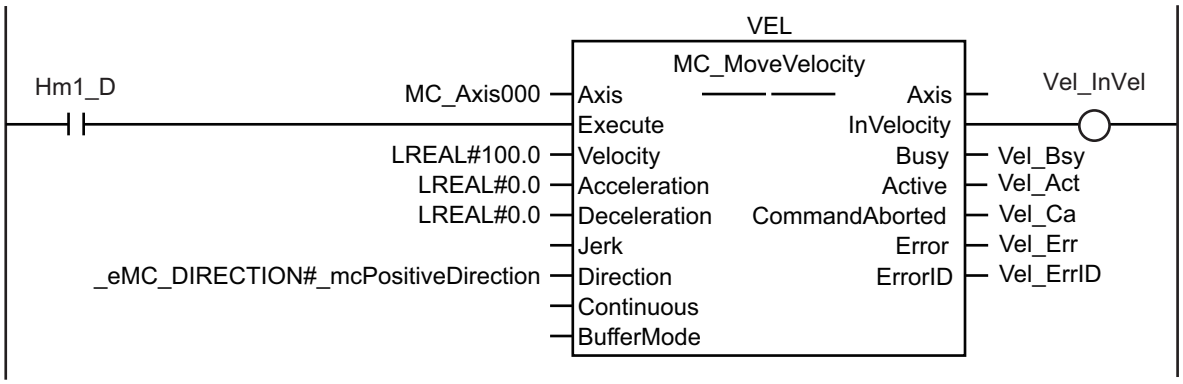
軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



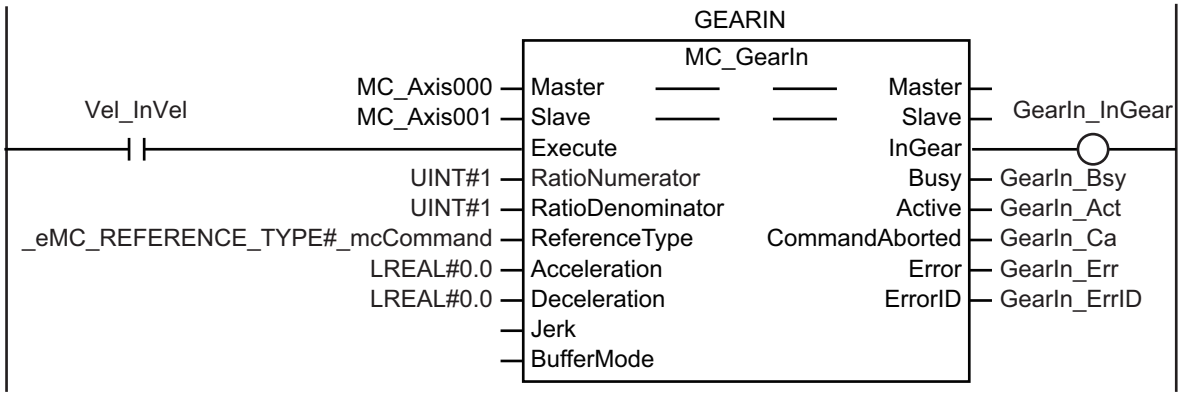
軸 3 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



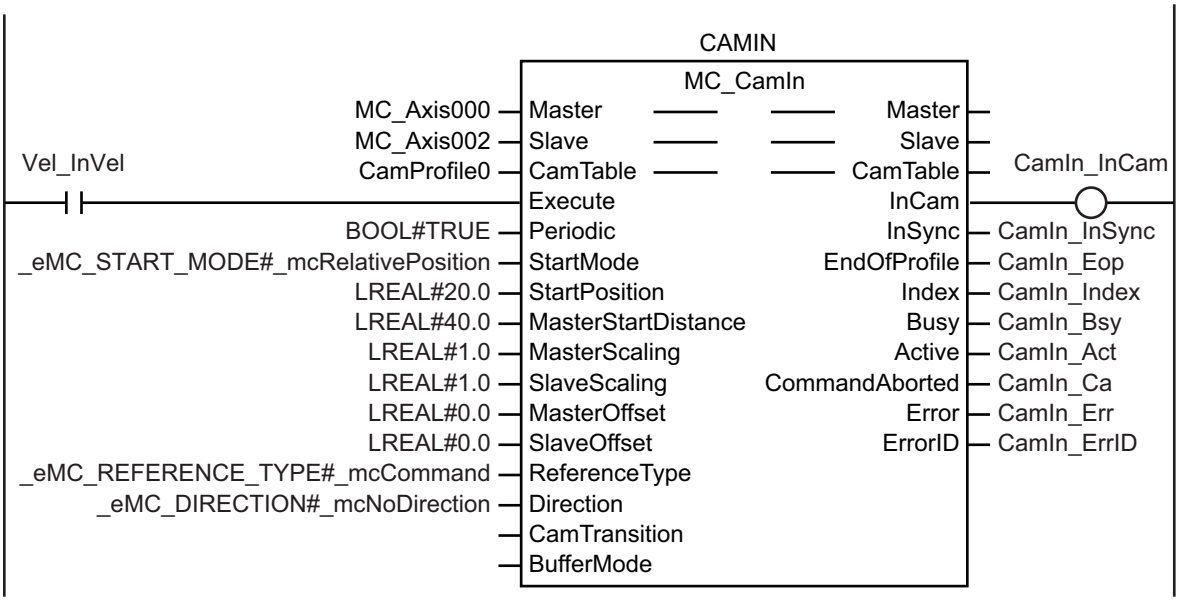
軸 0 の原点復帰が完了すれば速度制御を実行します。



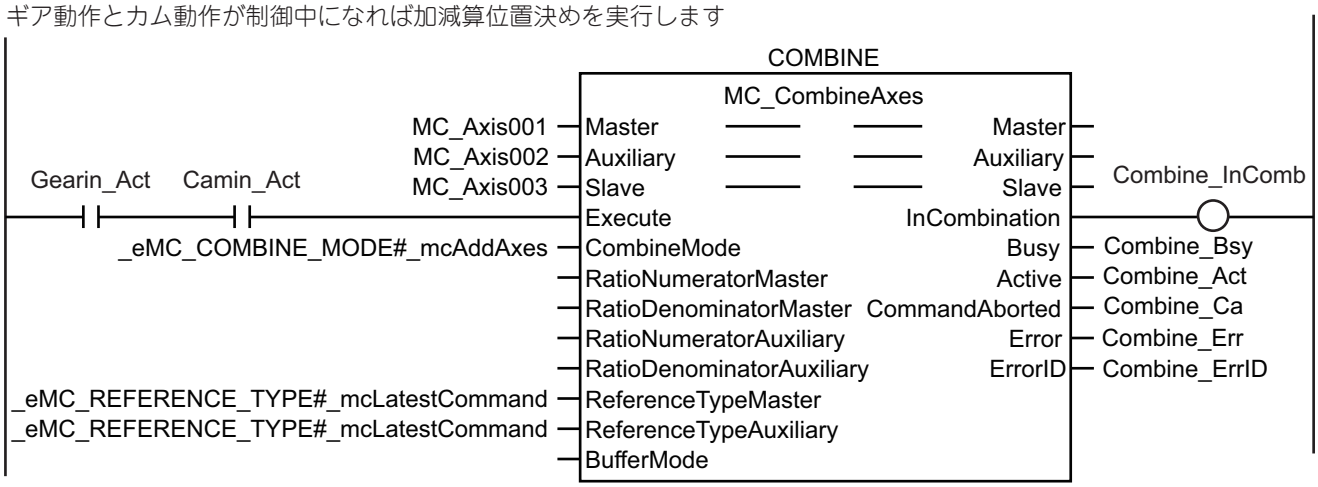
軸 0 が目標速度に到達すればギア動作を実行します。



軸 0 が目標速度に到達すればカム動作を実行します。



ギア動作とカム動作が制御中になれば加減算位置決めを実行します



ST プログラム

```
// 動作命令の入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag=FALSE THEN
  // 速度制御(MC_MoveVelocity)の入力パラメータに値を設定します。
    Vel_Vel := LREAL#100.0;
    Vel_Acc := LREAL#0.0;
    Vel_Dec := LREAL#0.0;
    Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;
  // カム動作開始(MC_CamIn)の入力パラメータに値を設定します。
    Camin_Em := TRUE;
    Camin_Sm := _eMC_START_MODE#_mcRelativePosition;
    Camin_Sp := LREAL#20.0;
    Camin_Msd := LREAL#40.0;
    Camin_Ms := LREAL#1.0;
    Camin_Ss := LREAL#1.0;
    Camin_Mo := LREAL#0.0;
```

```
    Camin_So := LREAL#0.0;
    Camin_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
    Camin_Dir := _eMC_DIRECTION#_mcNoDirection;
  // ギア動作開始(MC_GearIn)の入力パラメータに値を設定します。
    Gearin_RatN := UINT#1;
    Gearin_RatD := UINT#1;
    Gearin_RefTyp := _eMC_REFERENCE_TYPE#_mcCommand;
    Gearin_Acc := LREAL#0.0;
    Gearin_Dec := LREAL#0.0;
  // 加減算位置決め(MC_CombineAxes)の入力パラメータに値を設定します。
    Combine_Cm := _eMC_COMBINE_MODE#_mcAddAxes;
    Combine_RefMas :=_eMC_REFERENCE_TYPE#_mcLatestCommand;
    Combine_RefAux :=_eMC_REFERENCE_TYPE#_mcLatestCommand;
  // 入力パラメータの設定完了フラグを有効にします。
    InitFlag := TRUE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En :=TRUE;
  ELSE
    Pwr1_En :=FALSE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 3 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis003.DrvStatus.Ready=TRUE) THEN
    Pwr4_En :=TRUE;
  ELSE
    Pwr4_En :=FALSE;
END_IF;

// 軸 0～軸 3 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせて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) THEN
    FaultHandler();
END_IF;

// 軸 0 がサーボ ON 状態で原点未確定の場合、軸 0 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 3 がサーボ ON 状態で原点未確定の場合、軸 3 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0の原点復帰が完了すれば速度制御を実行します。
IF Hm1_D=TRUE THEN
  Vel_Ex :=TRUE;
END_IF;

// 軸0が目標速度に到達すればギア動作を実行します。
IF Vel_InVel=TRUE THEN
Gearin_Ex := TRUE;
END_IF;

// 軸0が目標速度に到達すればカム動作を実行します。
IF Vel_InVel=TRUE THEN
  Camin_Ex := TRUE;
END_IF;

// ギア動作とカム動作が制御中になれば加減算位置決めを実行します。
IF (Gearin_Act=TRUE) AND (Camin_Act=TRUE) THEN
  Combine_Ex :=TRUE;
END_IF;

// 軸0のMC_Power
PWR1(
  Axis := MC_Axis000,
  Enable := Pwr1_En,
  Status => Pwr1_Status,
  Busy => Pwr1_Bsy,
  Error => Pwr1_Err,
  ErrorID => Pwr1_ErrID
);

// 軸3のMC_Power
PWR4(
  Axis := MC_Axis003,
  Enable := Pwr4_En,
  Status => Pwr4_Status,
  Busy => Pwr4_Bsy,
  Error => Pwr4_Err,
  ErrorID => Pwr4_ErrID
);

// 軸0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 3 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
);

//MC_MoveVelocity
VEL(
Axis := MC_Axis000,
Execute := Vel_Ex,
Velocity := Vel_Vel,
Acceleration := Vel_Acc,
Deceleration := Vel_Dec,
Direction := Vel_Dir,
InVelocity => Vel_InVel,
Busy => Vel_Bsy,
Active => Vel_Act,
CommandAborted => Vel_Ca,
Error => Vel_Err,
ErrorID => Vel_ErrID
);

//MC_CamIn
CAMIN(
Master := MC_Axis000,
Slave := MC_Axis002,
CamTable := CamProfile0,
Execute := Camin_Ex,
Periodic := Camin_Em,
StartMode := Camin_Sm,
StartPosition := Camin_Sp,
MasterStartDistance := Camin_Msd,
MasterScaling := Camin_Ms,
SlaveScaling := Camin_Ss,
MasterOffset := Camin_Mo,
SlaveOffset := Camin_So,
ReferenceType := Camin_Rt,
Direction := Camin_Dir,
InCam => Camin_InCam,
InSync => Camin_InSync,
EndOfProfile => Camin_Eop,
Index => Camin_Index,
Busy => Camin_Bsy,
Active => Camin_Act,
CommandAborted => Camin_Ca,
```

```
Error => Camin_Err,
ErrorID => Camin_ErrID
);

//MC_GearIn
GEARIN(
  Master := MC_Axis000,
  Slave := MC_Axis001,
  Execute := Gearin_Ex,
  RatioNumerator := Gearin_RatN,
  RatioDenominator := Gearin_RatD,
  ReferenceType := Gearin_RefTyp,
  Acceleration := Gearin_Acc,
  Deceleration := Gearin_Dec,
  InGear => Gearin_InGear,
  Busy => Gearin_Bsy,
  Active => Gearin_Act,
  CommandAborted => Gearin_Ca,
  Error => Gearin_Err,
  ErrorID => Gearin_ErrID
);

//MC_CombineAxes
COMBINE(
Master := MC_Axis001,
Auxiliary := MC_Axis002,
Slave := MC_Axis003,
Execute := Combine_Ex,
CombineMode := Combine_CM,
ReferenceTypeMaster := Combine_RefMas,
ReferenceTypeAuxiliary := Combine_RefAux,
InCombination => Combine_InComb,
Busy => Combine_Bsy,
Active => Combine_Act,
CommandAborted => Combine_Ca,
Error => Combine_Err,
ErrorID => Combine_ErrID
);
```

10-2-14 カム動作時に主軸の位相を補正する

カム動作の従軸は、速度制御で動作している主軸に同期して動作しています。StartOn を TRUE にすると、MC_Phasing（主軸相対値位相補正）命令により主軸の位相を補正します。従軸は補正された位相に対して同期を行います。

プログラムで使用する主な変数

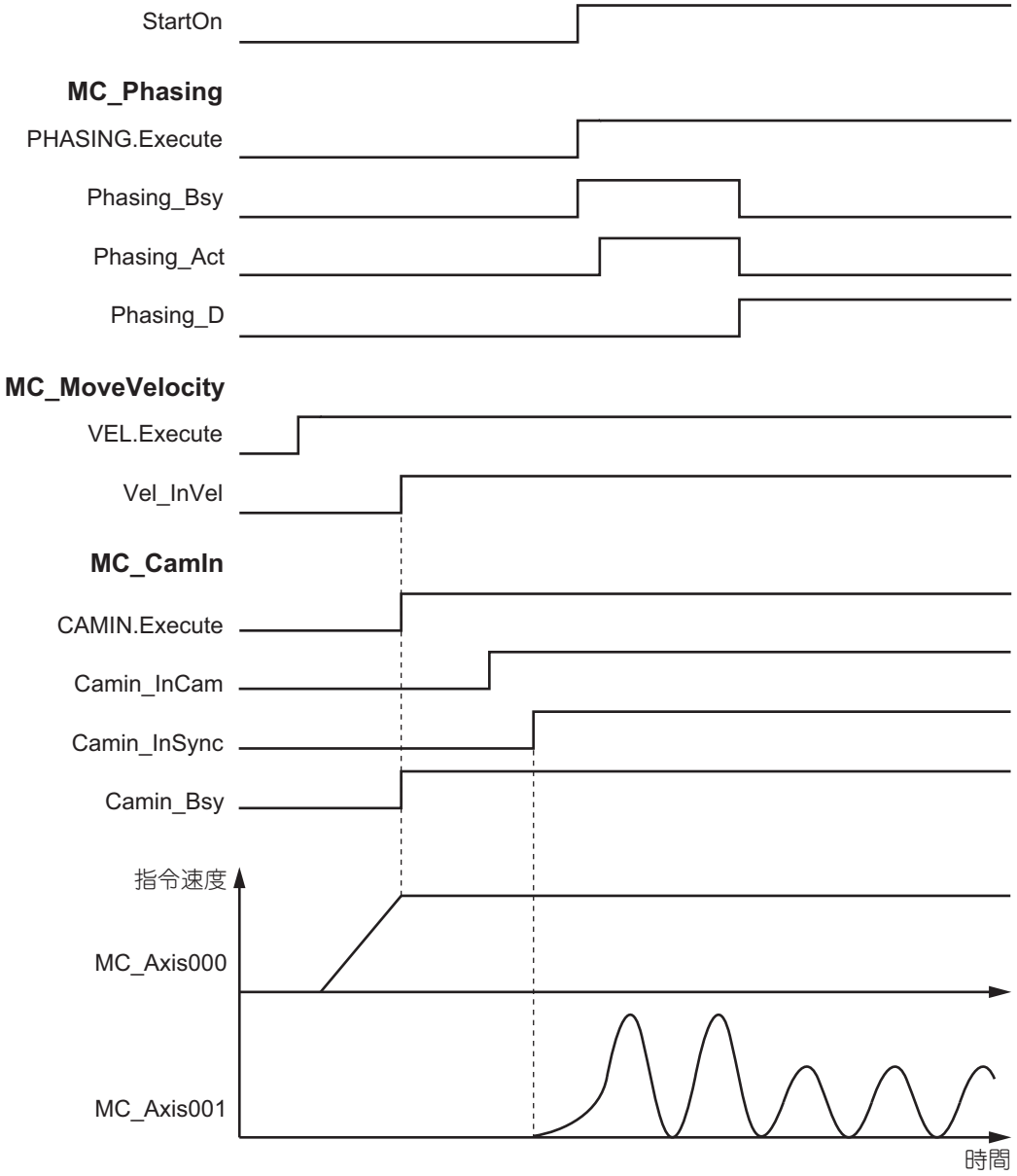
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。

名称	データ型	初期値	コメント
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 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_Axis000.Details.Homed	BOOL	FALSE	軸 0 が原点確定状態のとき TRUE となります。
MC_Axis001	_sAXIS_REF	−	軸 1 の軸変数です。
MC_Axis001.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.Details.Homed	BOOL	FALSE	軸 1 が原点確定状態のとき TRUE となります。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	−	カムデータ変数です。*1
Pwr1_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR1 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR2 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
StartOn	BOOL	FALSE	主軸の位相補正を開始するための変数です。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
Camin_Ex	BOOL	FALSE	MC_CamIn（カム動作開始）命令を起動するための変数です。 ST プログラムで使用しています。
Vel_Ex	BOOL	FALSE	MC_MoveVelocity（速度制御）命令を起動するための変数です。 ST プログラムで使用しています。
Phasing_Ex	BOOL	FALSE	MC_Phasing（主軸相対値位相補正）命令を起動するための変数です。 ST プログラムで使用しています。

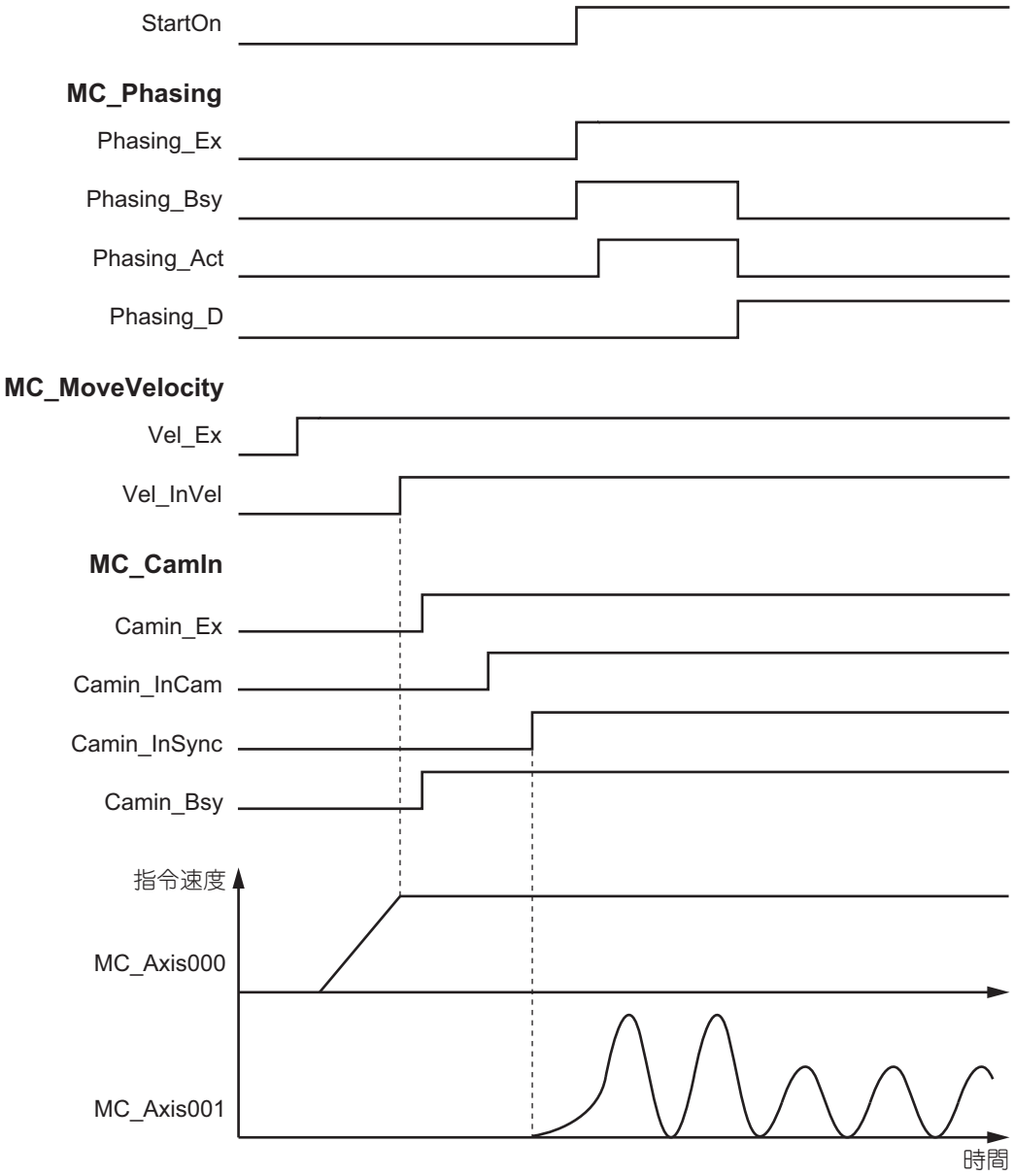
*1. 配列の要素数 ARRAY[0..N]は、Sysmac Studio のカムエディタ機能で設定します。本サンプルプログラムでは[0～360]としています。

タイミングチャート

● ラダー図プログラムの場合

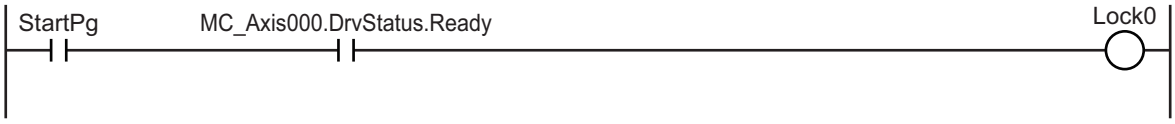


● **ST プログラムの場合**

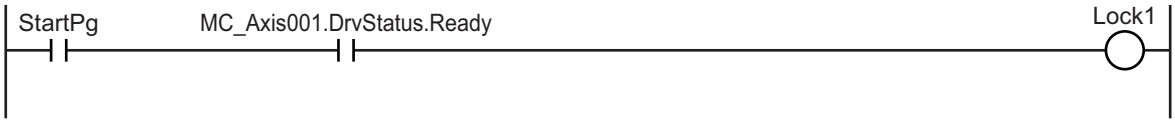


ラダー図プログラム

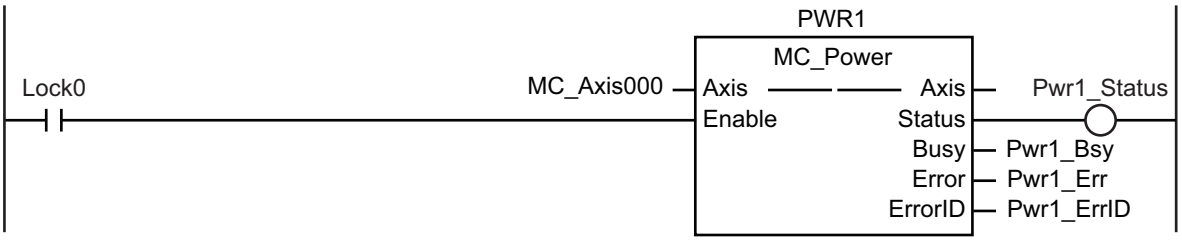
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します。



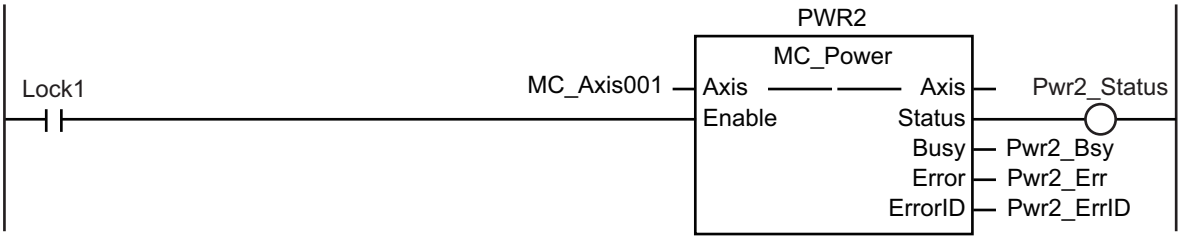
接点 StartPg が TRUE で、軸 1 のサーボドライバがサーボレディ状態か確認します。



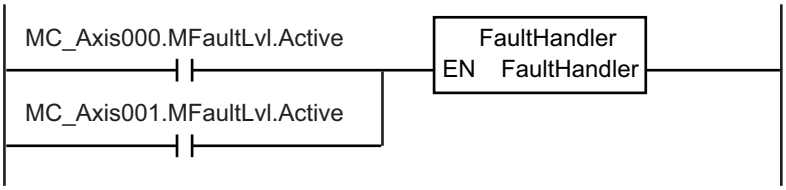
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。



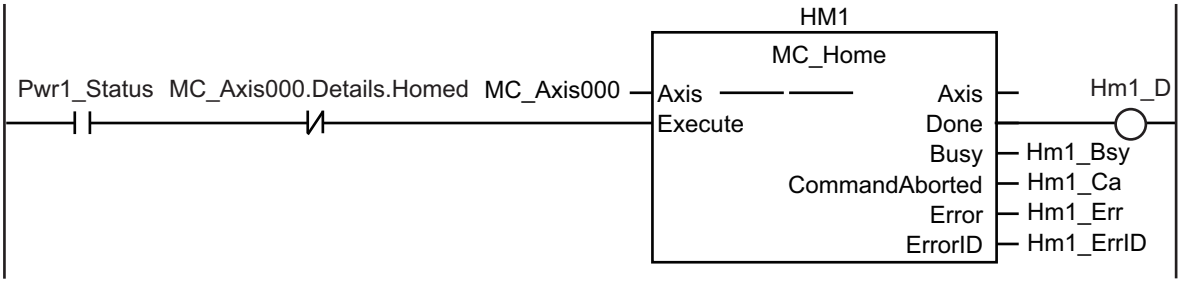
軸 1 のサーボドライバがサーボレディ状態であれば、軸 1 をサーボ ON します。



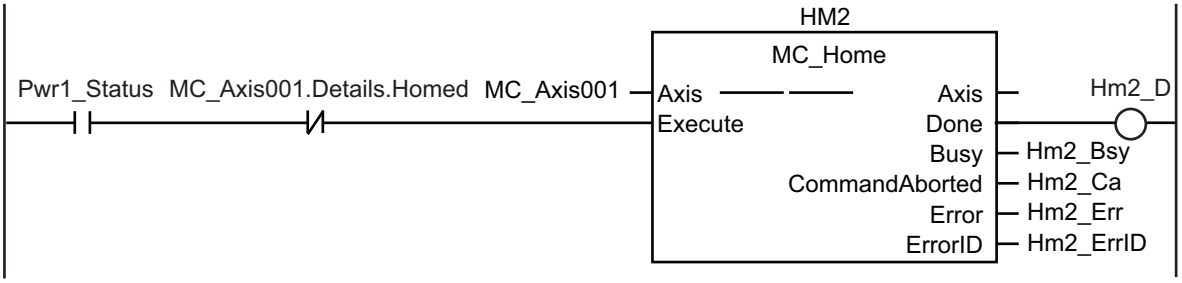
軸グループあるいは構成軸で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



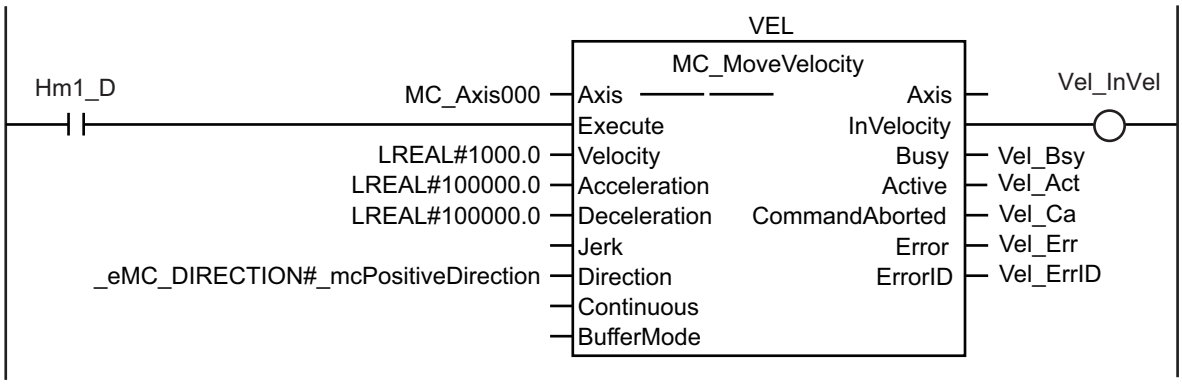
軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



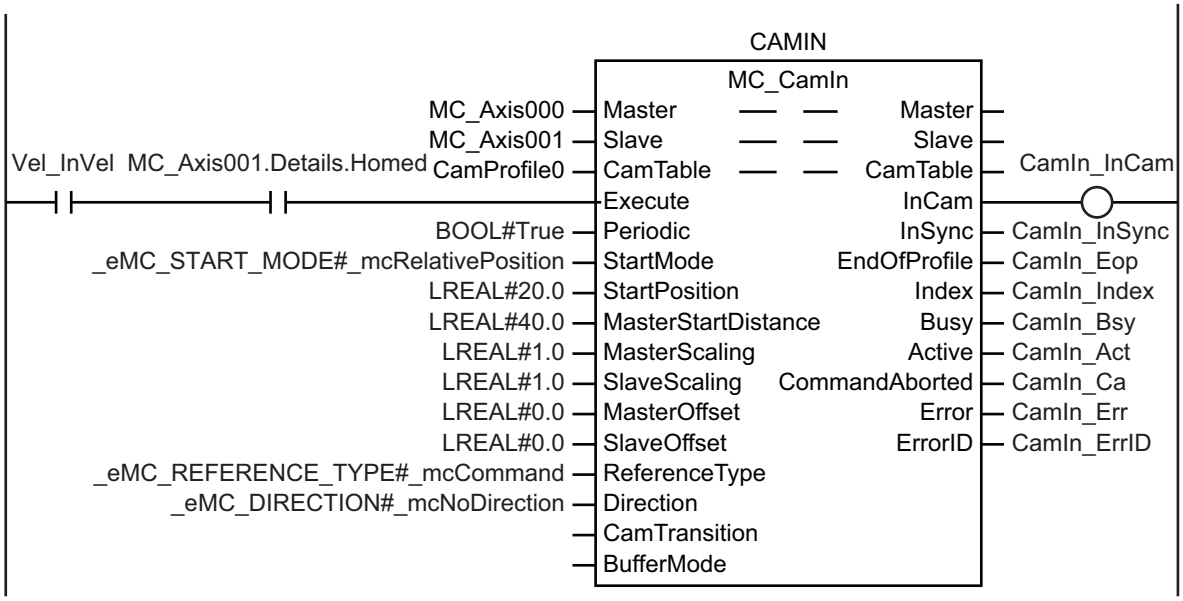
軸 1 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



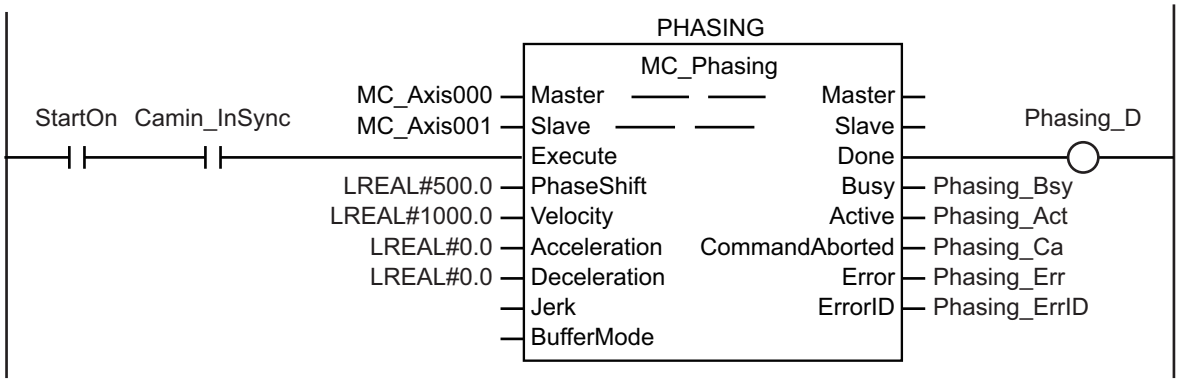
軸 0 の原点復帰が完了すれば速度制御を実行します。



軸 0 が目標速度に到達すればカム動作を実行します。



StartOn が TRUE でカム動作が同期中になれば主軸の位相補正を開始します。



ST プログラム

```
// 動作命令の入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag=FALSE THEN
  // 速度制御(MC_MoveVelocity)の入力パラメータに値を設定します。
    Vel_Vel := LREAL#1000.0;
```

```
    Vel_Acc := LREAL#100000.0;
    Vel_Dec := LREAL#100000.0;
    Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;
  // 主軸相対値位相補正(MC_Phasing)の入力パラメータに値を設定します。
    Phasing_Ps := LREAL#500.0;
    Phasing_Vel := LREAL#1000.0;
    Phasing_Acc := LREAL#0.0;
    Phasing_Dec := LREAL#0.0;
  // カム動作開始(MC_CamIn)の入力パラメータに値を設定します。
    Camin_Em := TRUE;
    Camin_Sm := _eMC_START_MODE#_mcRelativePosition;
    Camin_Sp := LREAL#20.0;
    Camin_Msd := LREAL#40.0;
    Camin_Ms := LREAL#1.0;
    Camin_Ss := LREAL#1.0;
    Camin_Mo := LREAL#0.0;
    Camin_So := LREAL#0.0;
    Camin_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
    Camin_Dir := _eMC_DIRECTION#_mcNoDirection;
  // 入力パラメータの設定完了フラグを有効にします。
    InitFlag := TRUE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis000.DrvStatus.Ready=TRUE) THEN
    Pwr1_En:=TRUE;
  ELSE
    Pwr1_En:=FALSE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 1 をサーボ ON します。
IF (StartPg=TRUE)
  AND (MC_Axis001.DrvStatus.Ready=TRUE) THEN
    Pwr2_En:=TRUE;
  ELSE
    Pwr2_En:=FALSE;
END_IF;

// 軸 0～軸 1 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせて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 0 がサーボ ON 状態で原点未確定の場合、軸 0 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 1 がサーボ ON 状態で原点未確定のとき、軸 1 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 0 の原点復帰が完了すれば、速度制御を実行します。
IF Hm1_D=TRUE THEN
  Vel_Ex :=TRUE;
END_IF;

// 軸 0 が目標速度に到達し、軸 1 が原点確定していればカム動作を実行します。
IF (Vel_InVel=TRUE) AND (MC_Axis001.Details.Homed=TRUE) THEN
  Camin_Ex := TRUE;
END_IF;

// StartOn が TRUE でカム動作が同期中になれば主軸の位相補正を開始します。
IF (StartOn=TRUE) AND (Camin_InSync=TRUE) THEN
  Phasing_Ex :=TRUE;
END_IF;

// 軸 0 の MC_Power
PWR1(
  Axis := MC_Axis000,
  Enable := Pwr1_En,
  Status => Pwr1_Status,
  Busy => Pwr1_Bsy,
  Error => Pwr1_Err,
  ErrorID => Pwr1_ErrID
);

// 軸 1 の MC_Power
PWR2(
  Axis := MC_Axis001,
  Enable := Pwr2_En,
  Status => Pwr2_Status,
  Busy => Pwr2_Bsy,
  Error => Pwr2_Err,
  ErrorID => Pwr2_ErrID
);

// 軸 0 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 1 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_Phasing
PHASING(
Master := MC_Axis000,
Slave := MC_Axis001,
Execute := Phasing_Ex,
PhaseShift := Phasing_Ps,
Velocity := Phasing_Vel,
Acceleration := Phasing_Acc,
Deceleration := Phasing_Dec,
Done => Phasing_D,
Busy => Phasing_Bsy,
Active => Phasing_Act,
CommandAborted => Phasing_Ca,
Error => Phasing_Err,
ErrorID => Phasing_ErrID
);

//MC_CamIn
CAMIN(
Master := MC_Axis000,
Slave := MC_Axis001,
CamTable := CamProfile0,
Execute := Camin_Ex,
Periodic := Camin_Em,
StartMode := Camin_Sm,
```

```
StartPosition := Camin_Sp,
MasterStartDistance := Camin_Msd,
MasterScaling := Camin_Ms,
SlaveScaling := Camin_Ss,
MasterOffset := Camin_Mo,
SlaveOffset := Camin_So,
ReferenceType := Camin_Rt,
Direction := Camin_Dir,
InCam => Camin_InCam,
InSync => Camin_InSync,
EndOfProfile => Camin_Eop,
Index => Camin_Index,
Busy => Camin_Bsy,
Active => Camin_Act,
CommandAborted => Camin_Ca,
Error => Camin_Err,
ErrorID => Camin_ErrID
);
```

10-2-15 速度制御中に現在位置を変更する

速度制御で動作中の軸の指令現在位置とフィードバック現在位置の絶対位置を変更します。

使用上の注意

- 動作中の軸に対して MC_SetPosition（現在位置変更）命令を起動すると、起動から実際に現在位置が変更されるまでの移動距離が誤差となります。
- MC_SetPosition（現在位置変更）命令を起動すると原点未確定となります。

軸パラメータ設定

設定項目	設定内容	説明
カウントモード	ロータリモード	位置のカウントモードをロータリモードに設定します。
リングカウンタ上限設定値	360	リングカウンタの上限値を「360」に設定します。
リングカウンタ下限設定値	0	リングカウンタの下限値を「0」に設定します。
原点復帰動作	原点プリセット	原点プリセットで原点確定します。

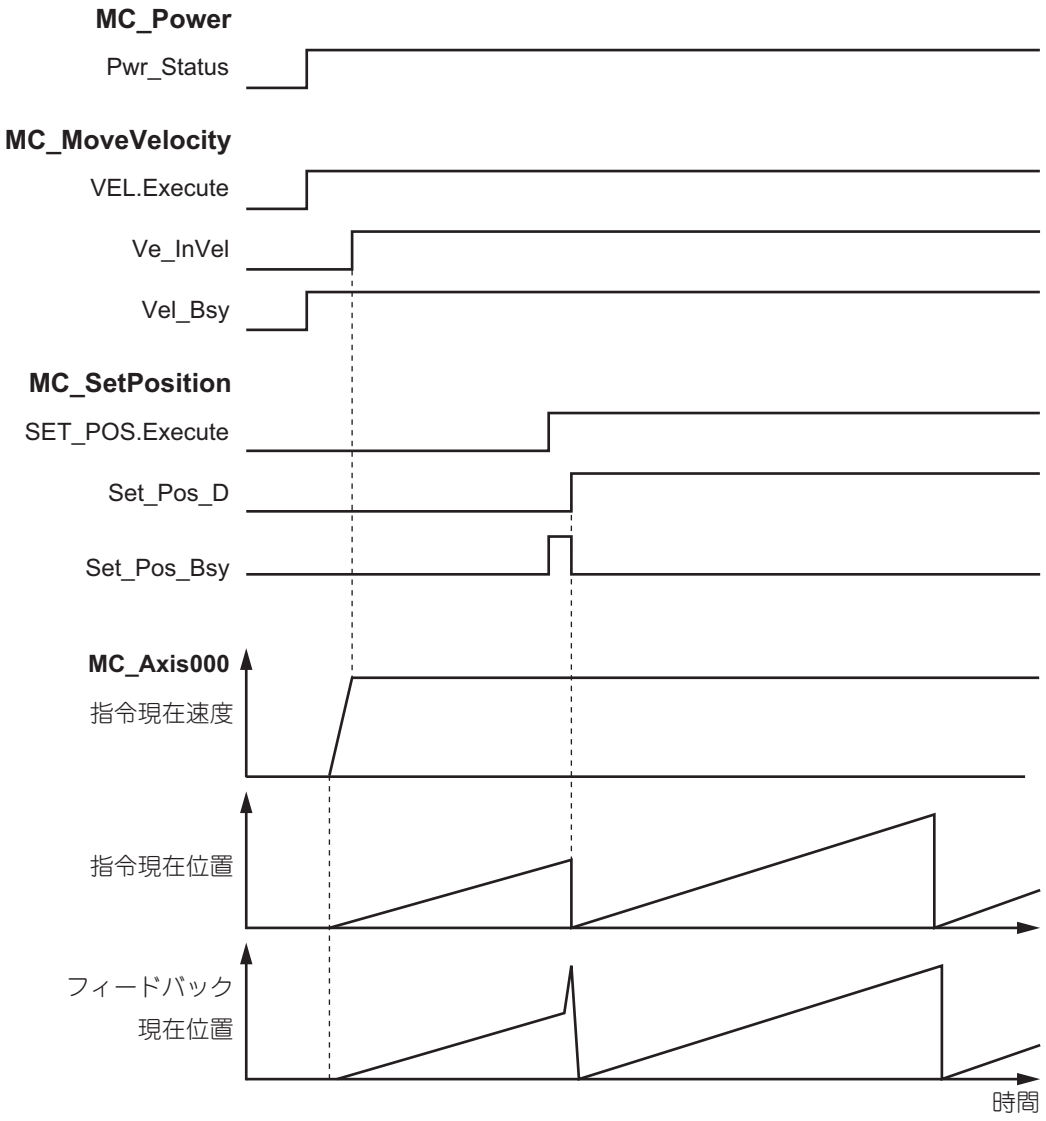
プログラムで使用する主な変数

名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
MC_Axis000.MFaultLvl.Active	BOOL	FALSE	軸 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。
Pwr_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
StartSetPos	BOOL	FALSE	現在位置を変更するための外部ボタンです。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。

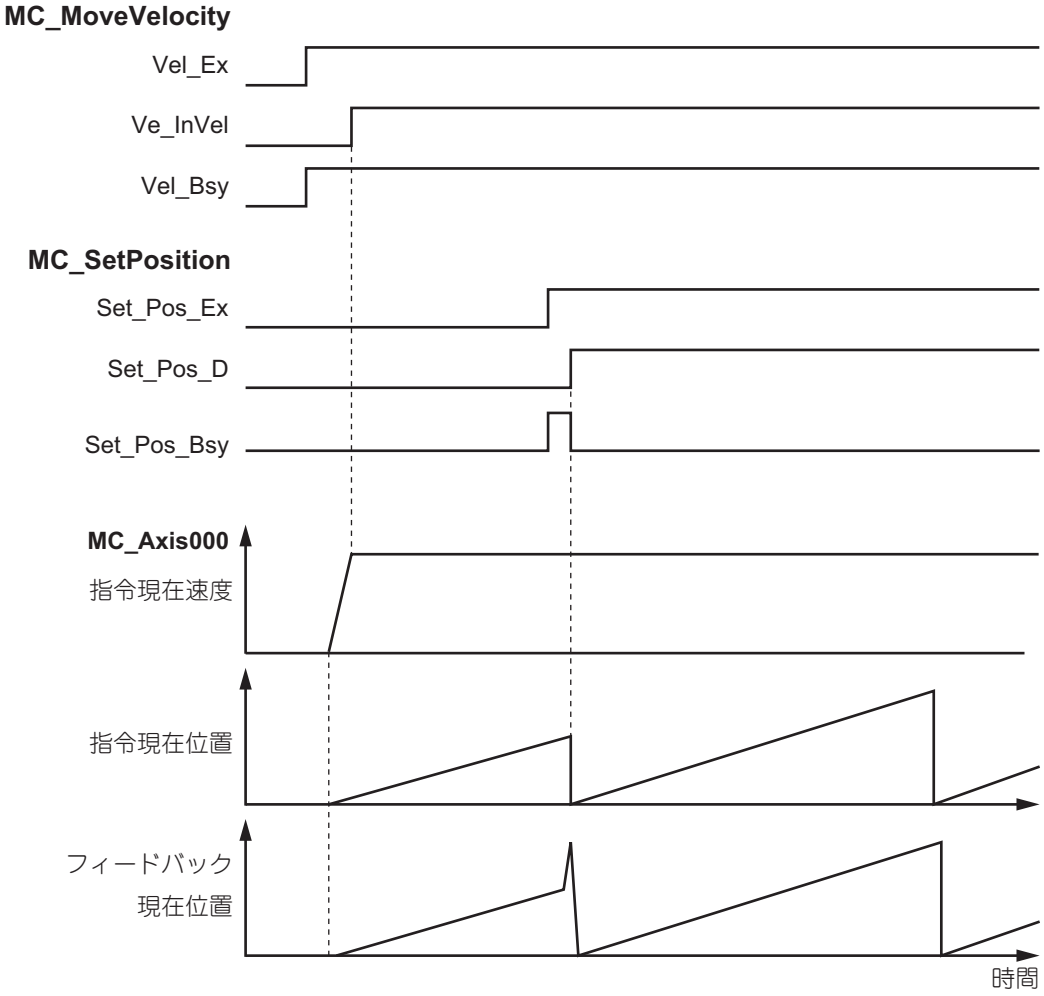
名称	データ型	初期値	コメント
Vel_Ex	BOOL	FALSE	MC_MoveVelocity（速度制御）命令を起動するための変数です。 ST プログラムで使用しています。
SetPos_Ex	BOOL	FALSE	MC_SetPosition（現在位置変更）命令を起動するための変数です。 ST プログラムで使用しています。

タイミングチャート

● ラダー図プログラムの場合

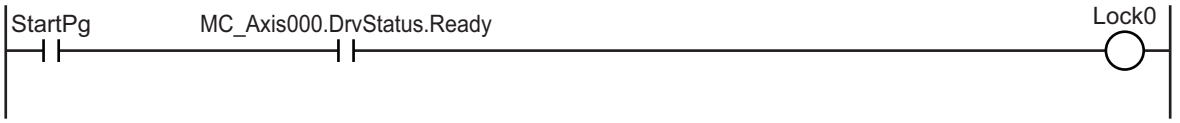


● **ST プログラムの場合**

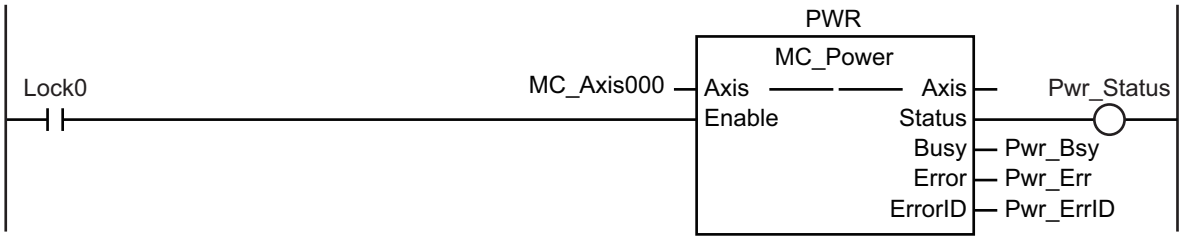


ラダー図プログラム

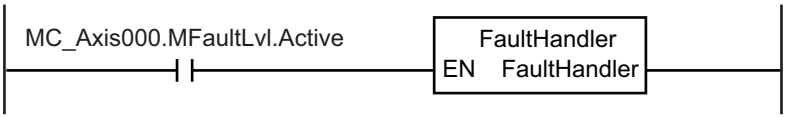
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します。



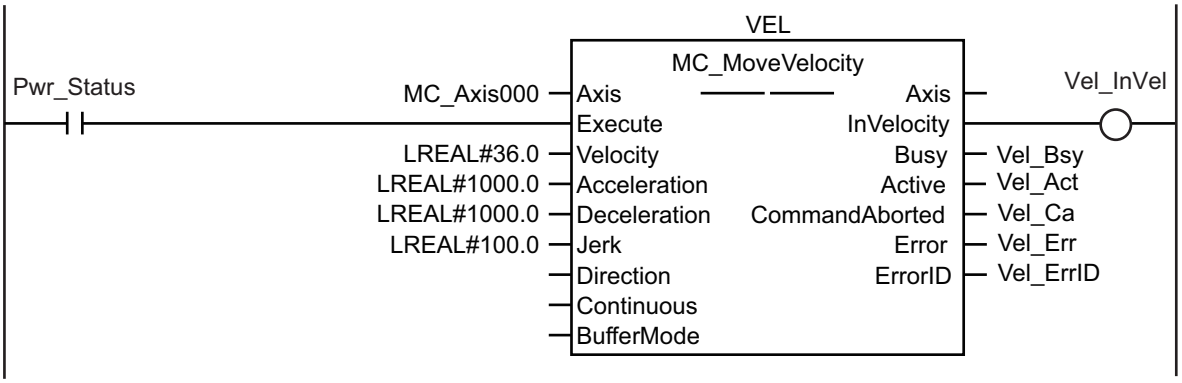
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。



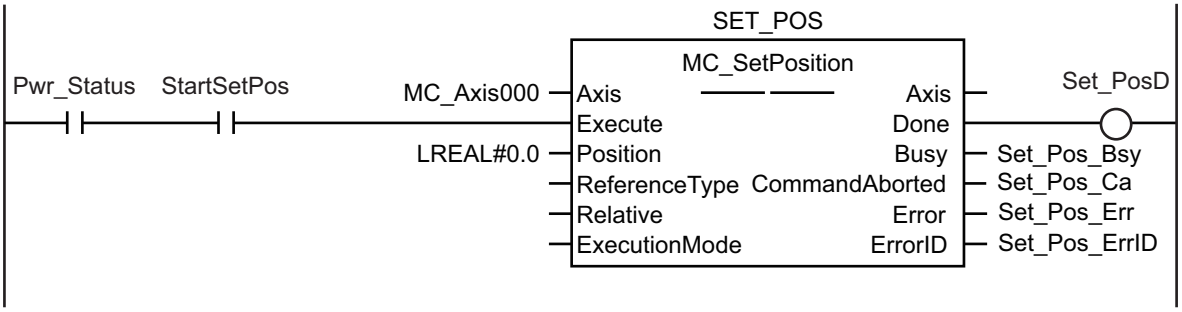
軸 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



軸 0 がサーボ ON すれば速度制御を実行します。



サーボ ON 状態で接点 StartSetPos が TRUE になれば、現在位置変更を実行します。



ST プログラム

```
// 命令の入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag=FALSE THEN
  // 速度制御(MC_MoveVelocity)の入力パラメータに値を設定します。
    Vel_Vel := LREAL#36.0;
    Vel_Acc := LREAL#1000.0;
    Vel_Dec := LREAL#1000.0;
    Vel_Jrk := LREAL#100.0;
  // 現在位置変更(MC_SetPosition)の入力パラメータに値を設定します。
    Set_Pos_Pos := LREAL#0.0;
  // 入力パラメータの設定完了フラグを有効にします。
    InitFlag :=TRUE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON 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 0 で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせてプログラミングしてください。
IF MC_Axis000.MFaultLvl.Active=TRUE THEN
  FaultHandler();
END_IF;

// 軸 0 がサーボ ON すれば軸 0 の速度制御を実行します。
IF Pwr_Status=TRUE THEN
  Vel_Ex :=TRUE;
END_IF;

// サーボ ON 状態で接点 StartSetPos が TRUE になれば、現在位置変更を実行します。
IF (Pwr_Status=TRUE) AND (StartSetPos=TRUE) THEN
  Set_Pos_Ex :=TRUE;
END_IF;

//MC_Power
PWR(
  Axis := MC_Axis000,
  Enable := Pwr_En,
  Status => Pwr_Status,
  Busy => Pwr_Bsy,
  Error => Pwr_Err,
  ErrorID => Pwr_ErrID
);

//MC_MC_MoveVelocity
VEL(
  Axis := MC_Axis000,
  Execute := Vel_Ex,
  Velocity := Vel_Vel,
  Acceleration := Vel_Acc,
  Deceleration := Vel_Dec,
  Jerk := Vel_Jrk,
  InVelocity => Vel_InVel,
  Busy => Vel_Bsy,
  Active => Vel_Act,
  CommandAborted => Vel_Ca,
  Error => Vel_Err,
  ErrorID => Vel_ErrID
);

//MC_SetPosition
SET_POS(
  Axis := MC_Axis000,
  Execute := Set_Pos_Ex,
  Position := Set_Pos_Pos,
```

```
  Done => Set_Pos_D,
  Busy => Set_Pos_Bsy,
  CommandAborted => Set_Pos_Ca,
  Error => Set_Pos_Err,
  ErrorID => Set_Pos_ErrID
);
```

10-2-16 カムデータ変数の変更とカムテーブルの保存

Sysmac Studio のカムエディタで作成したカムデータ変数の値をユーザプログラムで変更します。
位相が 0 度～180 度までは変位を 2 倍にし、181 度～360 度では変位を 0.5 倍にします。

カムデータの変更が完了すれば、MC_SaveCamTable（カムテーブル保存）命令により、CPU ユニットの不揮発性メモリに保存します。
保存が完了すれば、MC_CamIn（カム動作開始）命令によりカム動作を開始します。

使用上の注意

- 位相が昇順でない場合、MC_CamIn（カム動作開始）命令の起動時に異常となります。このサンプルプログラムでは位相の昇順チェックを行っていません。チェックを行う場合は、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。
- CPU ユニットの不揮発性メモリは書き込み回数に制限があります。カムテーブルの保存は必要な場合だけ行ってください。
- MC_SaveCamTable（カムテーブル保存）命令で保存する前に CPU ユニットの電源を OFF すると、ユーザプログラムで変更したカムデータ変数は変更前の内容に戻ります。

プログラムで使用する主な変数

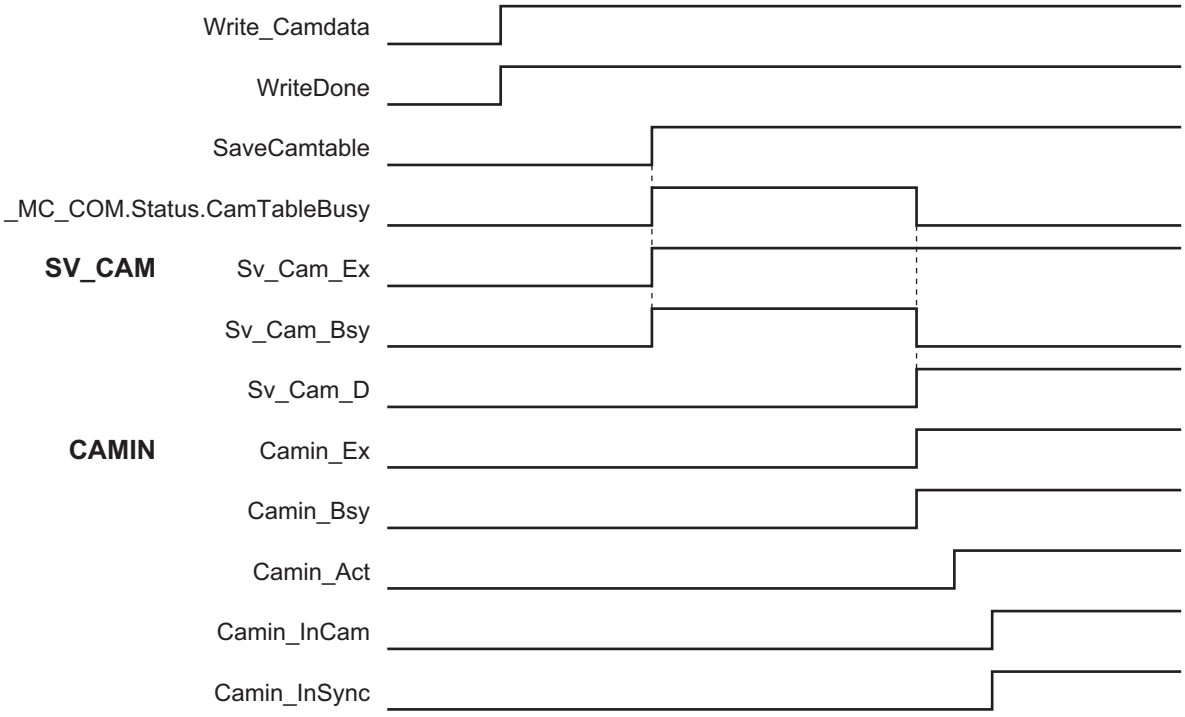
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	－	軸 0 の軸変数です。
MC_Axis001	_sAXIS_REF	－	軸 1 の軸変数です。
CamProfile0	ARRAY[0..360] OF _sMC_CAM_REF	－	カムデータ変数です。*1
Pwr1_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR1 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
Pwr2_Status	BOOL	FALSE	MC_Power 命令のインスタンス PWR2 の出力 Status に割り付けた変数です。 サーボ ON のときに TRUE となります。
StartPg	BOOL	FALSE	この変数が TRUE で EtherCAT 通信のプロセスデータ通信が正常に実行中であれば、サーボ ON します。
WriteCamdata	BOOL	FALSE	カムデータの変更を起動するための変数です。 編集を開始するときに TRUE としてください。
WriteDone	BOOL	FALSE	カムデータの変更が完了したことを表すための変数です。 変更が完了したときに TRUE としてください。
SaveCamtable	BOOL	FALSE	カムテーブル保存を起動するための変数です。
_MC_COM.Status.CamTableBusy	BOOL	FALSE	カムテーブル保存実行中を示すシステム定義変数です。
Sv_Cam_Ex	BOOL	FALSE	MC_SaveCamTable（カムテーブル保存）命令を起動するための変数です。

名称	データ型	初期値	コメント
Camin_Ex	BOOL	FALSE	MC_CamIn（カム動作開始）命令を起動するための変数です。 ST プログラムで使用しています。

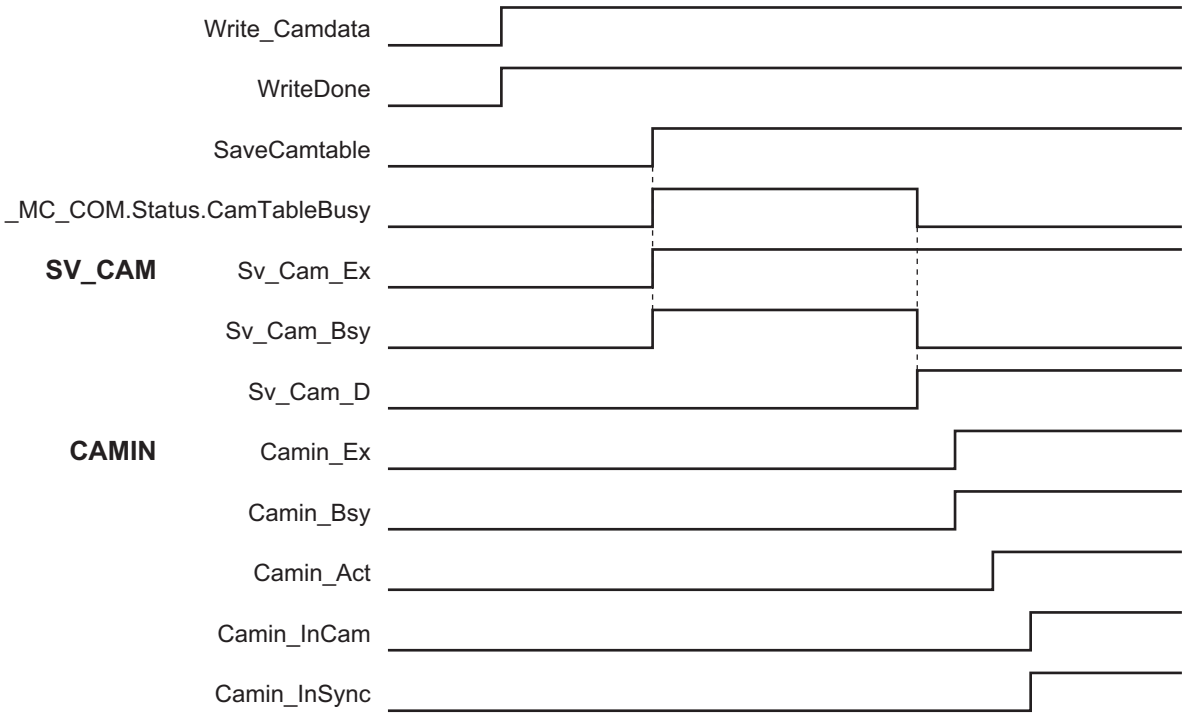
*1. 配列の要素数 ARRAY[0..N]は、Sysmac Studio のカムエディタ機能で設定します。本サンプルプログラムでは[0～360]としています。

タイミングチャート

● ラダー図プログラムの場合

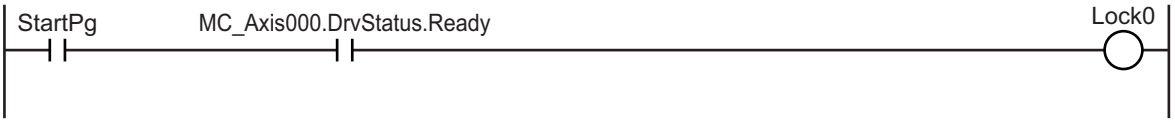


● **ST プログラムの場合**

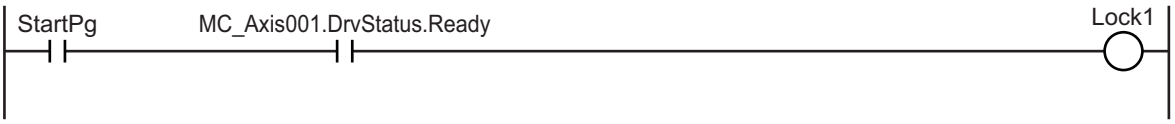


ラダー図プログラム

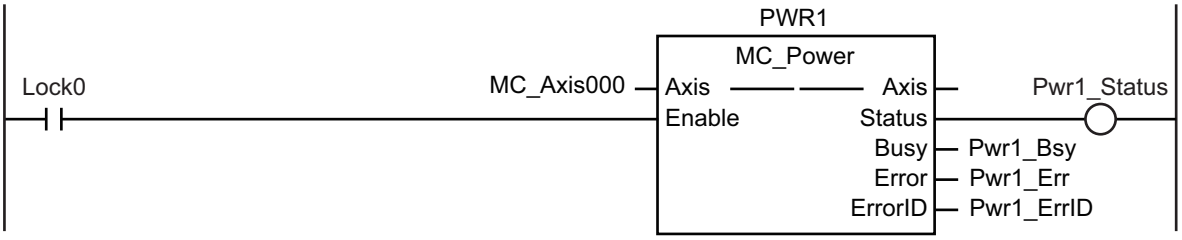
接点 StartPg が TRUE で、軸 0 のサーボドライバがサーボレディ状態か確認します。



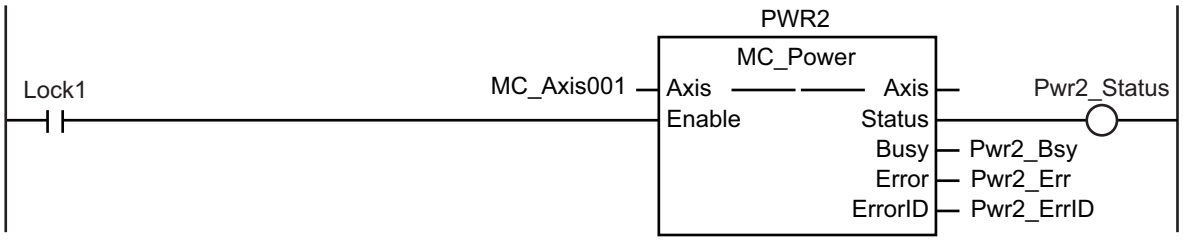
接点 StartPg が TRUE で、軸 1 のサーボドライバがサーボレディ状態か確認します。



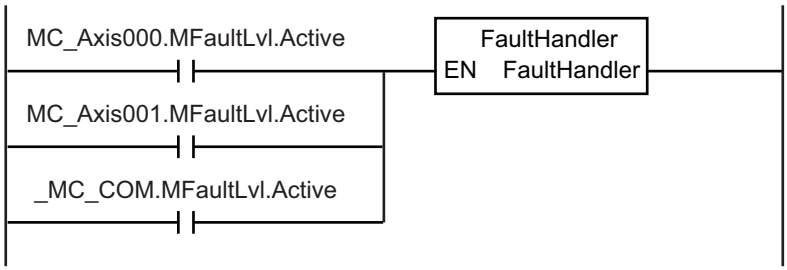
軸 0 のサーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON します。



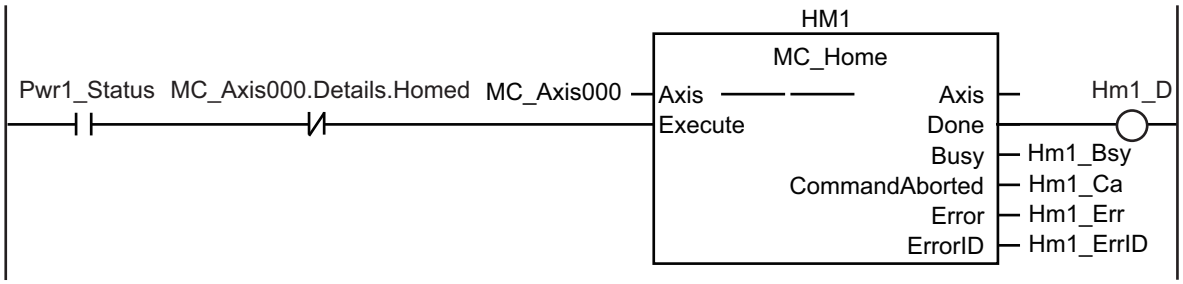
軸 1 のサーボドライバがサーボレディ状態であれば、軸 1 をサーボ ON します。



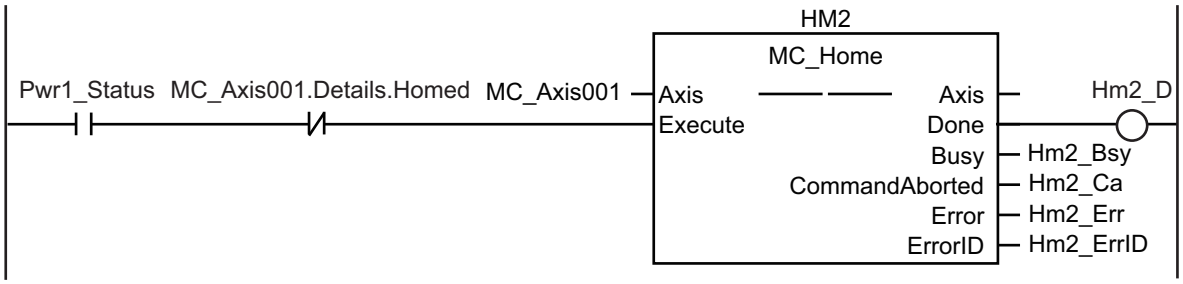
MC 共通あるいは軸で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



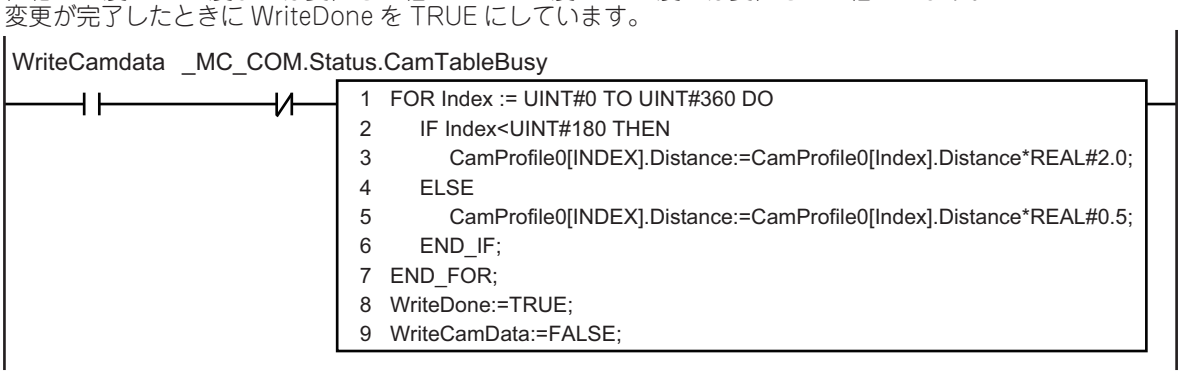
軸 0 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



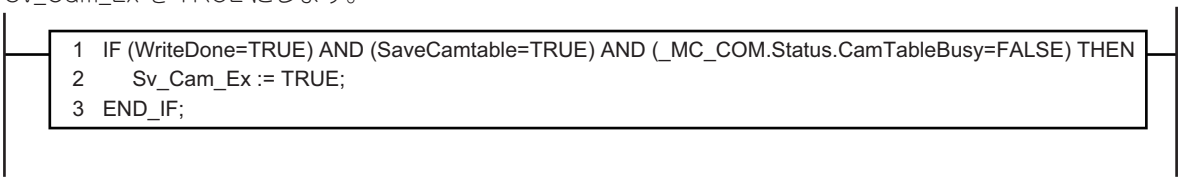
軸 1 がサーボ ON 状態で原点未確定の場合、原点復帰を実行します。



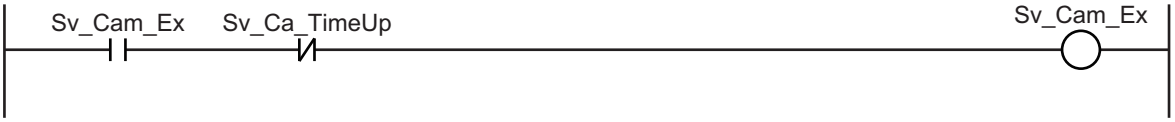
WriteCamData が TRUE でカムテーブルファイル保存が実行中でない場合、カムデータ変数の値を変更します。
位相が 0 度～ 180 度までは変位を 2 倍にし、181 度～ 360 度では変位を 0.5 倍にします。
変更が完了したときに WriteDone を TRUE にしています。



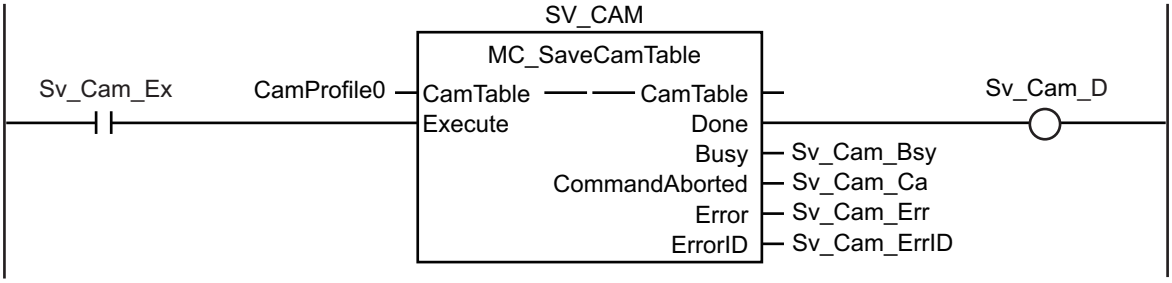
カムデータ変数の変更が完了し、SaveCamtable が TRUE でカムテーブルファイル保存が実行中でない場合、Sv_Cam_Ex を TRUE にします。



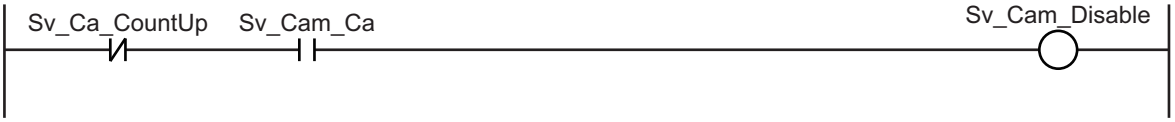
Sv_Ca_TimeUp が TRUE の場合、Sv_Cam_Ex を FALSE にします。
Sv_Cam_Ex が FALSE になれば、Sv_Ca_TimeUp が FALSE になり、Sv_Cam_Ex を TRUE にします。
MC_SaveCamTable（カムテーブル保存）命令を再度実行します。



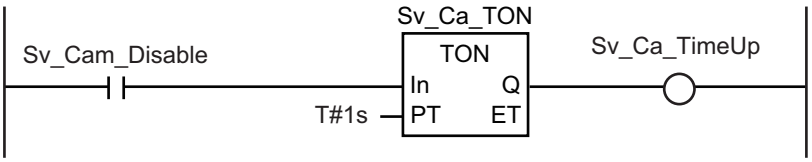
Sv_Cam_Ex が TRUE になれば、MC_SaveCamTable（カムテーブル保存）命令を実行します。



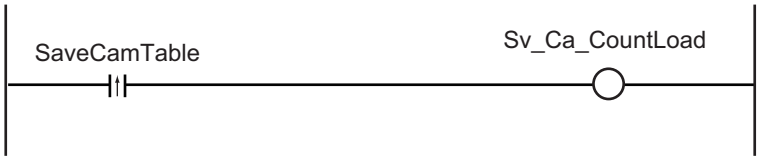
Sv_Ca_CountUp が FALSE で、カムテーブル保存実行不可が発生している場合、Sv_Cam_Disable を TRUE にします。



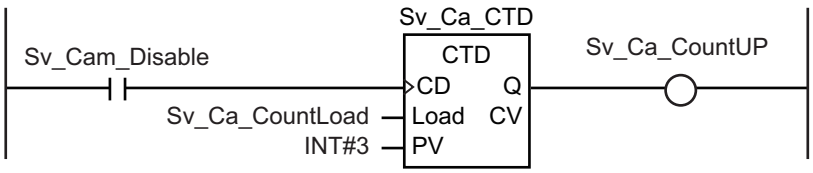
カムテーブル保存実行不可が発生した 1 秒後に、Sv_Ca_TimeUp を TRUE にします。
Sv_Ca_TimeUp が TRUE になれば、Sv_Cam_Ex が FALSE になります。



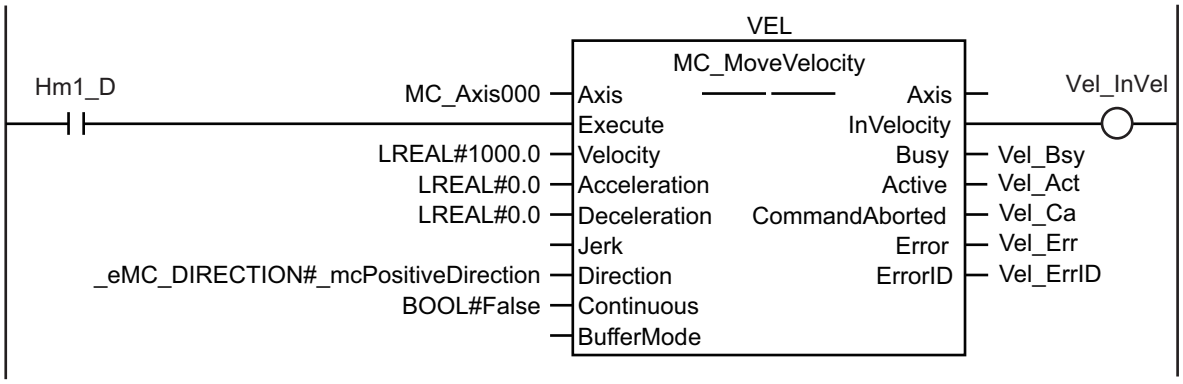
カムテーブル保存実行時に、Sv_Ca_CountLoad を 1 周期の間 TRUE にします。
Sv_Ca_CountLoad が TRUE になれば、リトライカウンタをリセットします。



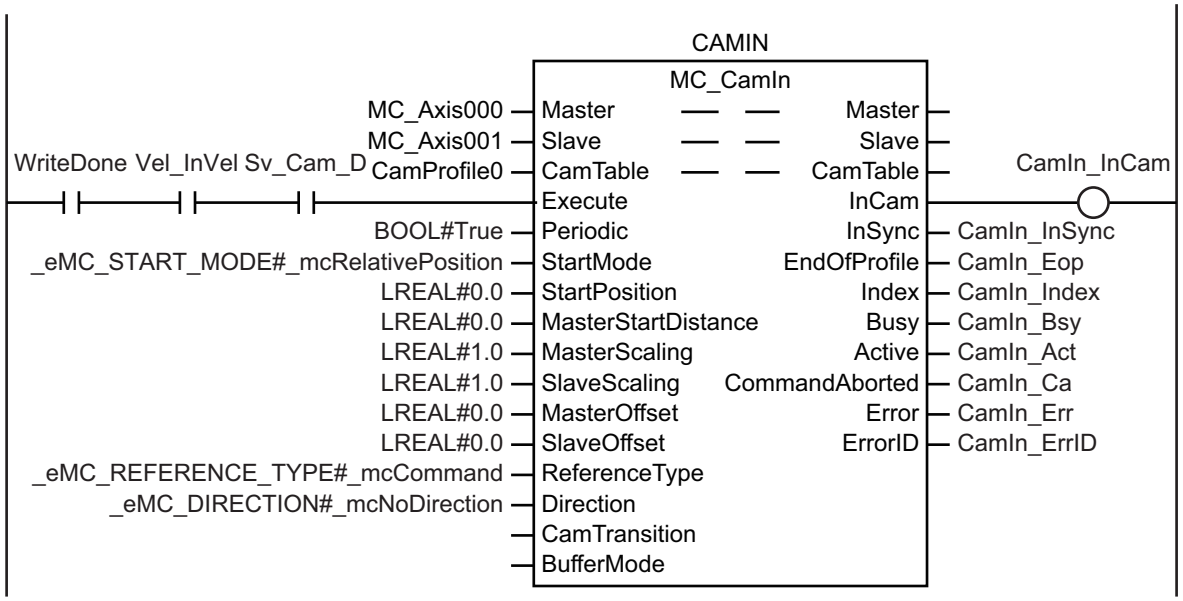
カムテーブル保存実行不可が 3 回発生した場合に、Sv_Ca_CountUp を TRUE にします。
Sv_Ca_CountUp が TRUE になれば、Sv_Cam_Disable を FALSE にします。
MC_SaveCamTable（カムテーブル保存）命令のリトライを終了します。



軸 0 の原点復帰が完了すれば速度制御を実行します。



カムデータ変数の変更とカムテーブルの保存が完了し、軸 0 が目標速度に到達すればカム動作を実行します。



ST プログラム

```
// 命令の入力パラメータが未設定の場合は、目標値などを設定します。
IF InitFlag=FALSE THEN
  // 速度制御(MC_MoveVelocity)の入力パラメータに値を設定します。
    Vel_Vel := LREAL#1000.0;
    Vel_Acc := LREAL#0.0;
    Vel_Dec := LREAL#0.0;
    Vel_Dir := _eMC_DIRECTION#_mcPositiveDirection;
  // カム動作開始(MC_CamIn)の入力パラメータに値を設定します。
    Camin_Em := TRUE;
    Camin_Sm := _eMC_START_MODE#_mcRelativePosition;
    Camin_Sp := LREAL#0.0;
    Camin_Msd := LREAL#0.0;
    Camin_Ms := LREAL#1.0;
    Camin_Ss := LREAL#1.0;
    Camin_Mo := LREAL#0.0;
    Camin_So := LREAL#0.0;
    Camin_Rt := _eMC_REFERENCE_TYPE#_mcCommand;
    Camin_Dir := _eMC_DIRECTION#_mcNoDirection;
  // 入力パラメータの設定完了フラグを有効にします。
    InitFlag: =TRUE;
END_IF;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 0 をサーボ ON 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;

// 接点 StartPg が TRUE で、サーボドライバがサーボレディ状態であれば、軸 1 をサーボ ON 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;

//MC 共通あるいは軸で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
// 異常処理の内容は装置にあわせて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 0 がサーボ ON 状態で原点未確定の場合、軸 0 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 1 がサーボ ON 状態で原点未確定の場合、軸 1 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;

//WriteCamData が TRUE でカムテーブルファイル保存が実行中でない場合、カムデータ変数の値を変更します。
// 位相が 0 度～180 度までは変位を 2 倍にし、181 度～360 度では変位を 0.5 倍にします。
// 変更が完了したときに WriteDone を TRUE にしています。
IF (WriteCamdata=TRUE) AND (_MC_COM.Status.CamTableBusy=FALSE) THEN
  FOR Index := UINT#0 TO UINT#360 DO
    IF Index<UINT#180 THEN
        CamProfile0[Index].Distance:=CamProfile0[Index].Distance*REAL#2.0;
      ELSE
        CamProfile0[Index].Distance:=CamProfile0[Index].Distance*REAL#0.5;
    END_IF;
  END_FOR;
  WriteDone :=TRUE;
  WriteCamdata :=FALSE;
END_IF;

// 軸 0 の原点復帰が完了すれば速度制御を実行します。
IF Hm1_D=TRUE THEN
  Vel_Ex :=TRUE;
END_IF;

// カムデータ変数の変更が完了し、SaveCamtable が TRUE でカムテーブルファイル保存が実行中でない場合、Sv_
Cam_Ex を TRUE にします。
// Sv_Cam_Ex が TRUE になれば、MC_SaveCamTable（カムテーブル保存）命令を実行します。
IF (WriteDone=TRUE)
  AND (SaveCamtable=TRUE)
  AND (_MC_COM.Status.CamTableBusy=FALSE) THEN
    Sv_Cam_Ex := TRUE;
END_IF;

// Sv_Ca_TimeUp が TRUE の場合、Sv_Cam_Ex を FALSE にします。
// Sv_Cam_Ex が FALSE になれば、Sv_Ca_TimeUp が FALSE になり、Sv_Cam_Ex を TRUE にします。
// MC_SaveCamTable（カムテーブル保存）命令を再度実行します。
IF (Sv_Cam_Ex=TRUE) AND (Sv_Ca_TimeUp=FALSE) THEN
    Sv_Cam_Ex := TRUE;
  ELSE
    Sv_Cam_Ex := FALSE;
END_IF;

// Sv_Ca_CountUP が FALSE で、カムテーブル保存実行不可が発生している場合、Sv_Cam_Disable を TRUE にしま
す。
IF (Sv_Ca_CountUP=FALSE) AND (Sv_Cam_Ca=TRUE) THEN
```

```
    Sv_Cam_Disable := TRUE;
  ELSE
    Sv_Cam_Disable := FALSE;
END_IF;

// カムテーブル保存実行不可が発生した 1 秒後に、Sv_Ca_TimeUp を TRUE にします。
// Sv_Ca_TimeUp が TRUE になれば、Sv_Cam_Ex が FALSE になります。
Sv_Ca_TON(
  In := Sv_Cam_Disable,
  PT := T#1s,
  Q => Sv_Ca_TimeUp
);

// カムテーブル保存実行時に、Sv_Ca_CountLoad を 1 周期の間 TRUE にします。
// Sv_Ca_CountLoad が TRUE になれば、リトライカウンタをリセットします。
R_TRIG1(SaveCamtable, Sv_Ca_CountLoad);

// カムテーブル保存実行不可が 3 回発生した場合、Sv_Ca_CountUP を TRUE にします。
// Sv_Ca_CountUP が TRUE になれば、Sv_Cam_Disable を FALSE にします。
// MC_SaveCamTable（カムテーブル保存）命令のリトライを終了します。
Sv_Ca_CTD(
  CD := Sv_Cam_Disable,
  LOAD := Sv_Ca_CountLoad,
  PV := INT#3,
  Q => Sv_Ca_CountUP
);

// カムデータ変数の変更とカムテーブルの保存が完了し、軸 0 が目標速度に到達すればカム動作を実行します。
IF (Vel_InVel=TRUE)
  AND (WriteDone=TRUE)
  AND (Sv_Cam_D=TRUE) THEN
    Camin_Ex :=TRUE;
END_IF;

//MC_SaveCamTable
SV_CAM(
  CamTable :=CamProfile0,
  Execute := Sv_Cam_Ex,
  Done => Sv_Cam_D,
  Busy => Sv_Cam_Bsy,
  CommandAborted => Sv_Cam_Ca,
  Error => Sv_Cam_Err,
  ErrorID => Sv_Cam_ErrID
);

CAMIN(
  Master := MC_Axis000,
  Slave := MC_Axis001,
  CamTable := CamProfile0,
  Execute := Camin_Ex,
  Periodic := Camin_Em,
```

```
  StartMode := Camin_Sm,
  StartPosition := Camin_Sp,
  MasterStartDistance := Camin_Msd,
  MasterScaling := Camin_Ms,
  SlaveScaling := Camin_Ss,
  MasterOffset := Camin_Mo,
  SlaveOffset := Camin_So,
  ReferenceType := Camin_Rt,
  Direction := Camin_Dir,
  InCam => Camin_InCam,
  InSync => Camin_InSync,
  EndOfProfile => Camin_Eop,
  Index => Camin_Index,
  Busy => Camin_Bsy,
  Active => Camin_Act,
  CommandAborted => Camin_Ca,
  Error => Camin_Err,
  ErrorID => Camin_ErrID
);

// 軸 0 の MC_Power
PWR1(
  Axis := MC_Axis000,
  Enable := Pwr1_En,
  Status => Pwr1_Status,
  Busy => Pwr1_Bsy,
  Error => Pwr1_Err,
  ErrorID => Pwr1_ErrID
);

// 軸 1 の MC_Power
PWR2(
  Axis := MC_Axis001,
  Enable := Pwr2_En,
  Status => Pwr2_Status,
  Busy => Pwr2_Bsy,
  Error => Pwr2_Err,
  ErrorID => Pwr2_ErrID
);

// 軸 0 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 1 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
);
```

10-2-17 軸パラメータの一時書き換え

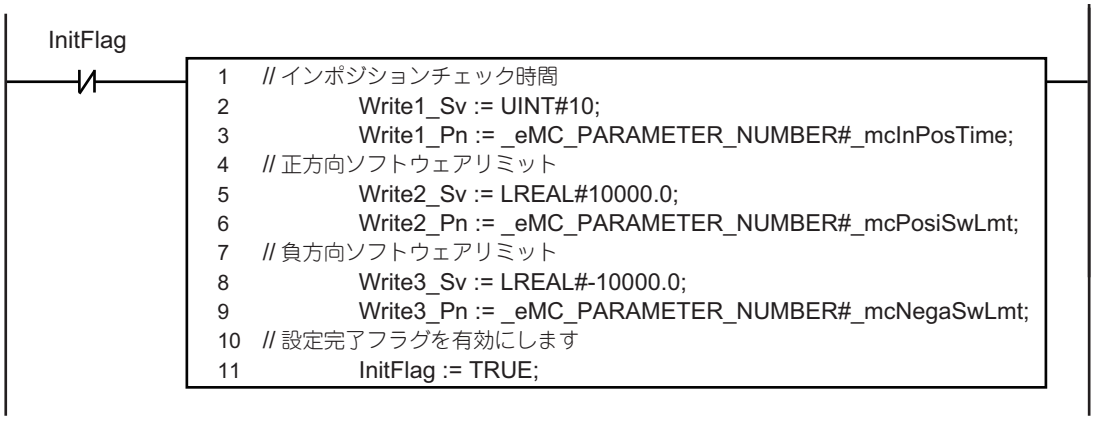
MC 設定書込(MC_Write)命令を使って、インポジションチェック時間、正方向ソフトウェアリミット、負方向ソフトウェアリミットの値を変更します。

プログラムで使用する主な変数

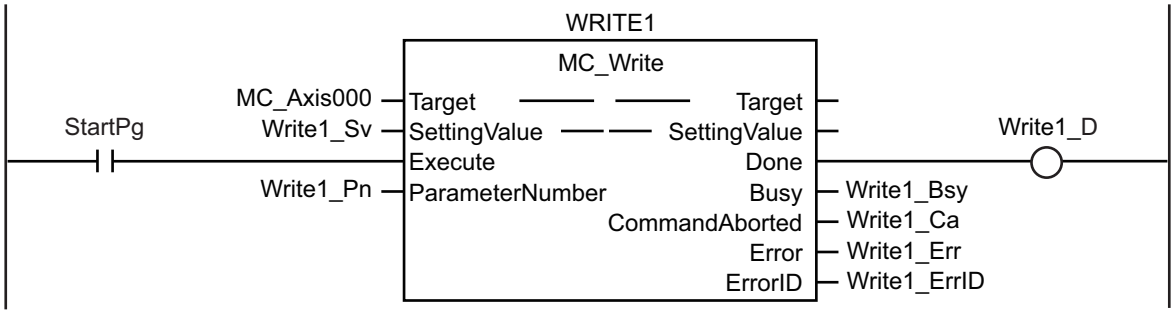
名称	データ型	初期値	コメント
MC_Axis000	_sAXIS_REF	−	軸 0 の軸変数です。
InitFlag	BOOL	FALSE	パラメータの設定状態を示す変数です。 FALSE でパラメータ変更を実行し、完了したときに TRUE にします。
StartPg	BOOL	FALSE	MC 設定書込を実行するための変数です。

ラダー図プログラム

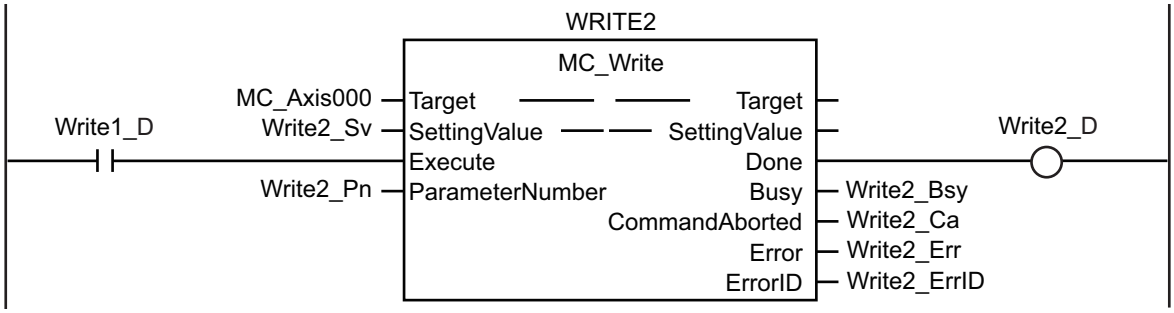
パラメータを設定します。設定が完了すれば、InitFlag を TRUE にします。



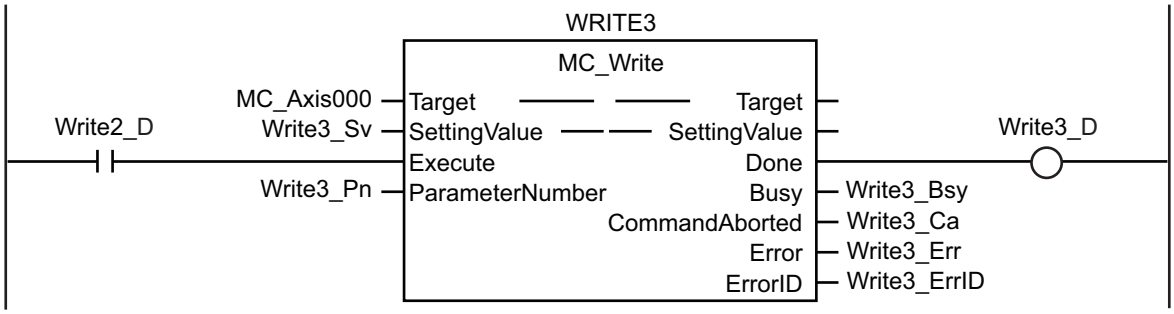
StartPg が TRUE になれば、インポジションチェック時間を変更します。



インポジションチェック時間の設定が完了すれば、正方向ソフトウェアリミットを変更します。



正方向ソフトウェアリミットの設定が完了すれば、負方向ソフトウェアリミットを変更します。



ST プログラム

```
// パラメータを設定します。設定が完了すれば、InitFlag を TRUE にします。
IF InitFlag=FALSE THEN
  // インポジションチェック時間
    Write1_Sv := UINT#10;
    Write1_Pn := _eMC_PARAMETER_NUMBER#_mcInPosTime;
  // 正方向ソフトウェアリミット
    Write2_Sv := LREAL#10000.0;
    Write2_Pn := _eMC_PARAMETER_NUMBER#_mcPosiSwLmt;
  // 負方向ソフトウェアリミット
    Write3_Sv := LREAL#-10000.0;
    Write3_Pn := _eMC_PARAMETER_NUMBER#_mcNegaSwLmt;
  // 入力パラメータの設定完了フラグを有効にします。
    InitFlag := TRUE;
END_IF;

//StartPg が TRUE になれば、インポジションチェック時間を変更します。
IF StartPg =TRUE THEN
  Write1_Ex := TRUE;
END_IF;

// インポジションチェック時間の変更が完了すれば、正方向ソフトウェアリミットを変更します。
IF Write1_D = TRUE THEN
  Write2_Ex := TRUE;
END_IF;

// 正方向ソフトウェアリミットの変更が完了すれば、負方向ソフトウェアリミットを変更します。
IF Write2_D = TRUE THEN
Write3_Ex := TRUE;
END_IF;

// MC_Write
WRITE1(
  Target := MC_Axis000,
  SettingValue := Write1_Sv,
  Execute := Write1_Ex,
  ParameterNumber := Write1_Pn,
  Done => Write1_D,
  Busy => Write1_Bsy,
  CommandAborted => Write1_Ca,
  Error => Write1_Err,
  ErrorID => Write1_ErrID
);

WRITE2(
  Target := MC_Axis000,
  SettingValue := Write2_Sv,
  Execute := Write2_Ex,
  ParameterNumber := Write2_Pn,
```

```
  Done => Write2_D,
  Busy => Write2_Bsy,
  CommandAborted => Write2_Ca,
  Error => Write2_Err,
  ErrorID => Write2_ErrID
);

WRITE3(
  Target := MC_Axis000,
  SettingValue := Write3_Sv,
  Execute := Write3_Ex,
  ParameterNumber := Write3_Pn,
  Done => Write3_D,
  Busy => Write3_Bsy,
  CommandAborted => Write3_Ca,
  Error => Write3_Err,
  ErrorID => Write3_ErrID
);
```

10-2-18 カムテーブルの終点インデックス更新

最大カムデータ数が 110 点で有効データ数が 100 点のカムテーブルについて、有効データ数を 10 点増加し、終点インデックスを更新します。

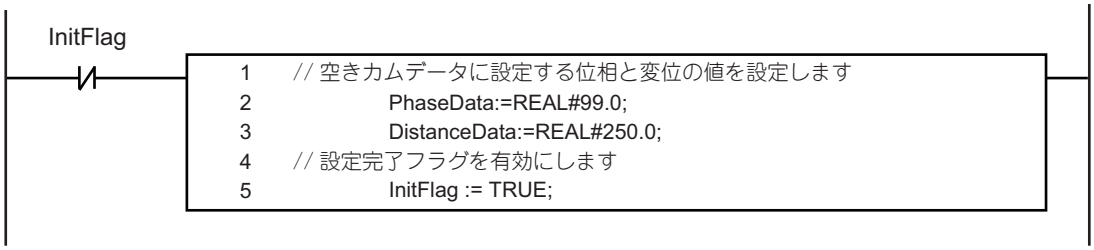
プログラムで使用する主な変数

名称	データ型	初期値	コメント
CamProfile0	ARRAY[0..109] OF _sMC_CAM_REF	−	最大カムデータ数が 110 点のカムデータ変数です。*1 有効カムデータ数は 100 点で、空きカムデータ数は 10 点です。
WriteCamdata	BOOL	FALSE	カムデータの変更を起動するための変数です。 編集を開始するときに TRUE としてください。
WriteDone	BOOL	FALSE	カムデータの変更が完了したことを表すための変数です。 変更が完了したときに TRUE としてください。

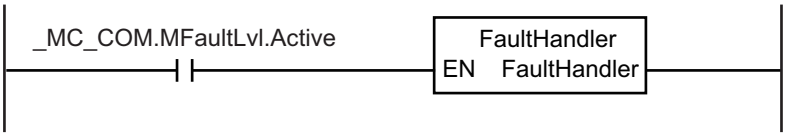
*1. 配列の要素数 ARRAY[0..N]は、Sysmac Studio のカムエディタ機能で設定します。本サンプルプログラムでは[0～109]としています。

ラダー図プログラム

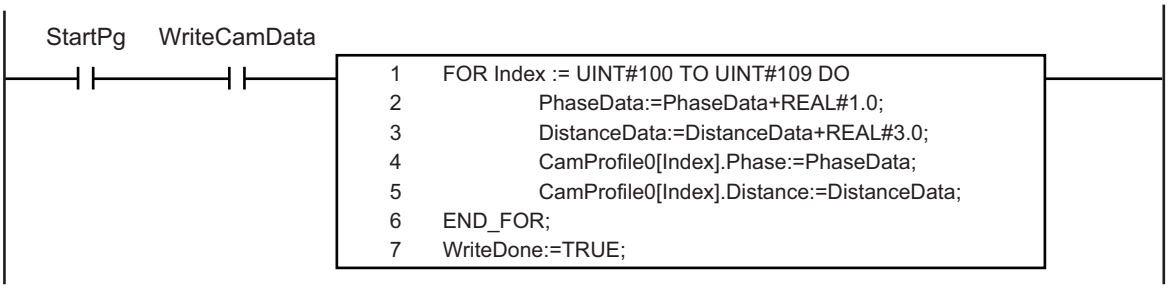
パラメータを設定します。設定が完了すれば、InitFlag を TRUE にします。



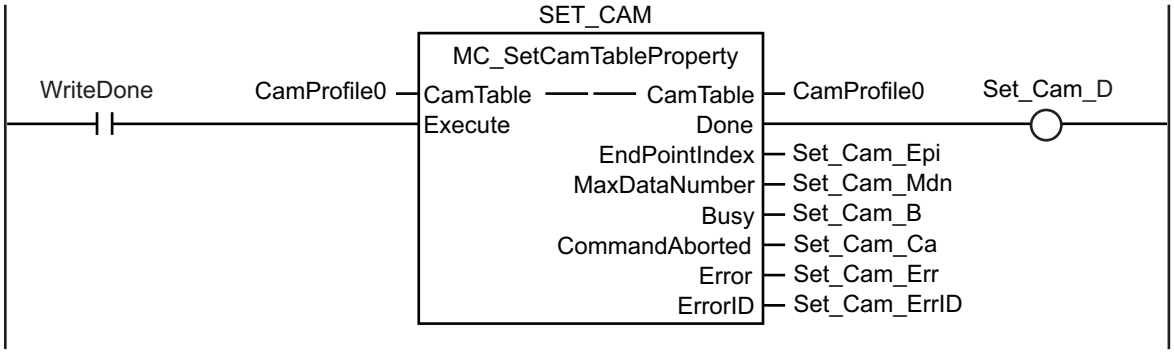
MC 共通で軽度フォールトレベルの異常が発生した場合に、異常処理の FaultHandler を実行します。
異常処理の内容は装置にあわせてプログラミングしてください。



StartPg と WriteCamData が TRUE になれば、カムデータ変数の値を変更します。
CamProfile[100] ～ CamProfile[109] に位相と変位を設定しています。
カムデータ変数の変更が完了したときに WriteDone を TRUE にしています。



カムデータ変数の変更が完了すれば、カムテーブルプロパティ更新を実行します



ST プログラム

```
// パラメータを設定します。設定が完了すれば、InitFlag を TRUE にします。
IF InitFlag=FALSE THEN
  // 空きカムデータに設定する位相と変位の値を設定します。
    PhaseData :=REAL#99.0;
    DistanceData :=REAL#250.0;
  // 設定完了フラグを TRUE にします。
    InitFlag := TRUE;
END_IF;

// MC 共通で軽度フォールトレベルの異常が発生した場合、異常時処理を実行します。
// 異常処理の内容は装置にあわせてプログラミングしてください。
IF _MC_COM.MFaultLvl.Active=TRUE THEN
FaultHandler();
END_IF;

// StartPg と WriteCamData が TRUE になれば、カムデータ変数の値を変更します。
// CamProfile[100]～CamProfile[109]に位相と変位を設定しています。
// カムデータ変数の変更が完了したときに WriteDone を TRUE にしています。
```

```
IF StartPg=TRUE
  AND WriteCamData=TRUE THEN
    FOR Index := UINT#100 TO UINT#109 DO
      PhaseData :=PhaseData+REAL#1.0;
      DistanceData :=DistanceData+REAL#3.0;
      CamProfile0[Index].Phase :=PhaseData;
      CamProfile0[Index].Distance :=DistanceData;
    END_FOR;
    WriteDone :=TRUE;
END_IF;

// カムデータ変数の変更が完了すれば、カムテーブルプロパティ更新を実行します。
IF WriteDone=TRUE THEN
  Set_Cam_Ex := TRUE;
END_IF;

//MC_SetCamTableProperty
SET_CAM(
  CamTable := CamProfile0,
  Execute := Set_Cam_Ex,
  Done => Set_Cam_D,
  EndPointIndex => Set_Cam_Epi,
  MaxDataNumber => Set_Cam_Mdn,
  Busy => Set_Cam_B,
  CommandAborted => Set_Cam_Ca,
  Error => Set_Cam_Err,
  ErrorID => Set_Cam_ErrID
);
```

11

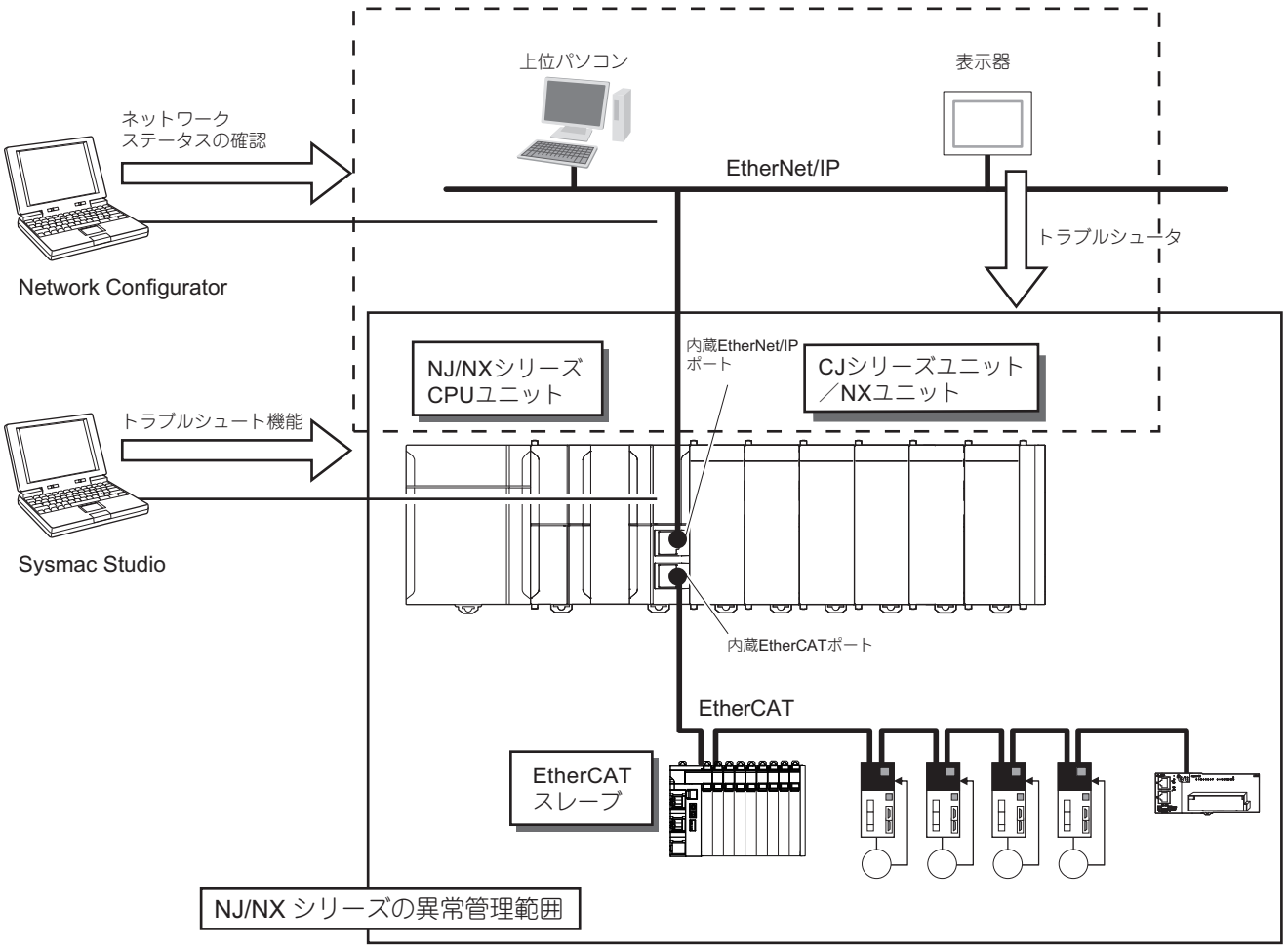
異常の確認と処置

異常の確認方法の概要を説明します。

11-1　異常確認と処置の概要 11-2

11-1 異常確認と処置の概要

NJ/NX シリーズは、コントローラで発生した異常をイベントとして共通の方法で一元的に管理します。
これにより、管理範囲（CPU ユニット、NX ユニット、NX シリーズスレーブターミナル、EtherCAT スレーブ*1、CJ シリーズユニット）の異常に対して、共通の方法で異常の内容と対処方法を確認できます。
*1. Sysmac デバイスであることが必要です。



Sysmac Studio のトラブルシュート機能または表示器のトラブルシュータを使用すると、異常の内容から対処方法までを確認できるため、迅速な対応が可能です。

異常の種類や内容、異常発生時の具体的な対処方法、および NJ/NX シリーズ全体のトラブルシュートについては、『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

オムロン製サーボドライバ 1S シリーズおよび G5 シリーズのオブジェクトの設定値や接続方法を記載しています。

A-1　サーボドライバ 1S シリーズとの接続 A-2
A-1-1　サーボドライバとの配線 A-2
A-1-2　サーボドライバの設定 A-2
A-2　サーボドライバ G5 シリーズとの接続 A-11
A-2-1　サーボドライバとの配線 A-11
A-2-2　サーボドライバの設定 A-11
A-3　エンコーダ入力ターミナルとの接続 A-20
A-3-1　エンコーダ入力ターミナルとの配線 A-20
A-3-2　エンコーダ入力ターミナルの設定 A-20
A-4　NX ユニットとの接続 A-26
A-5　PDS 状態遷移 A-27
A-5-1　PDS 状態制御方式 A-27
A-5-2　主回路電源 OFF 検出 A-28
A-6　用語説明 A-29
A-6-1　NJ/NX シリーズ共通 A-29
A-6-2　モーション制御機能 A-29
A-6-3　EtherCAT 通信 A-31
A-7　バージョン関連情報 A-32
A-8　マニュアル No.変更前の改訂履歴 A-36

A-1　サーボドライバ 1S シリーズとの接続

オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプとの接続について説明します。

A-1-1　サーボドライバとの配線

サーボドライバとは、EtherCAT 通信で接続します。
接続方法については、『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)』を参照してください。

A-1-2　サーボドライバの設定

ここでは、MC 機能モジュールの制御対象であるオムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプと接続した場合のサーボドライバ側の設定概要について説明します。
サーボドライバの詳細については、『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。

外部入力信号の割付

MC 機能モジュールでは、サーボドライバの汎用入力を下記の入力信号として使用します。
- 即停止入力
- 正方向限界入力
- 負方向限界入力
- 原点近傍入力
- 外部ラッチ機能のトリガ信号（ラッチ入力 1／ラッチ入力 2）

● 正方向限界入力／負方向限界入力／原点近傍入力の割付

オムロン製サーボドライバ 1S シリーズの出荷時設定では、各入力信号は下表のように割り付けられています。

信号名称	入力信号	形 R88D-1SN□の配線	形 R88D-1SAN□の配線
即停止入力	サーボドライバの汎用入力 1(IN1)	CN1 コネクタのピン No.12 b 接*1	CN1 コネクタのピン No.5 b 接*1
正方向限界入力	サーボドライバの汎用入力 2(IN2)	CN1 コネクタのピン No.32 b 接*2	CN1 コネクタのピン No.19 b 接*2
負方向限界入力	サーボドライバの汎用入力 3(IN3)	CN1 コネクタのピン No.13 b 接*3	CN1 コネクタのピン No.6 b 接*3
原点近傍入力	サーボドライバの汎用入力 4(IN4)	CN1 コネクタのピン No.33 a 接	CN1 コネクタのピン No.20 a 接

*1.　サーボドライバ側の信号名称は「異常停止入力」です。
*2.　サーボドライバ側の信号名称は「正方向駆動禁止入力」です。
*3.　サーボドライバ側の信号名称は「負方向駆動禁止入力」です。

● 外部ラッチ機能のトリガ信号の割付

オムロン製サーボドライバ 1S シリーズの出荷時設定の場合、下表の入力信号が外部ラッチ機能のトリガ信号に割り付けられています。

MC_TouchProbe 命令の入力変数「TriggerInput（トリガ入力条件）」の設定			外部ラッチ機能のトリガ信号	形 R88D-1SN □の配線	形 R88D-1SAN □の配線
Mode	InputDrive	LatchID			
0: mcDrive	0: mcEncoderMark	−	エンコーダの Z 相	−	−
	1: mcEXT	1: mcLatch1	サーボドライバの汎用入力 7(IN7)	CN1 コネクタのピン No.15 a 接[*1]	CN1 コネクタのピン No.8 a 接[*1]
		2: mcLatch2	サーボドライバの汎用入力 8(IN8)	CN1 コネクタのピン No.35 a 接[*2]	CN1 コネクタのピン No.22 a 接[*2]
1: mcController	−	−	TriggerVariable（トリガ変数）で指定した変数	−	−

*1. サーボドライバ側の信号名称は「外部ラッチ入力 1」です。
*2. サーボドライバ側の信号名称は「外部ラッチ入力 2」です。

バックラッシュ補正機能

MC 機能モジュールはバックラッシュ補正機能がありません。
バックラッシュ補正が必要な場合は、サーボドライバ側の補正機能を使用します。
サーボドライバ側で設定が必要なオブジェクトを下記に示します。

インデックス	サブインデックス	名称	説明
3001Hex	−	機械	−
	02Hex	バックラッシュ補正選択	位置制御時のバックラッシュ補正の有効または無効、補正時の動作方向を設定します。[*1] 0: バックラッシュ補正無効 1: サーボ ON 後、最初の正転側動作時に補正します。 2: サーボ ON 後、最初の逆転側動作時に補正します。
	03Hex	バックラッシュ補正量	位置制御時のバックラッシュ補正量を設定します。
	04Hex	バックラッシュ補正時定数	位置制御時のバックラッシュ補正時定数を設定します。

*1. 出荷時設定は［0: バックラッシュ補正無効］です。

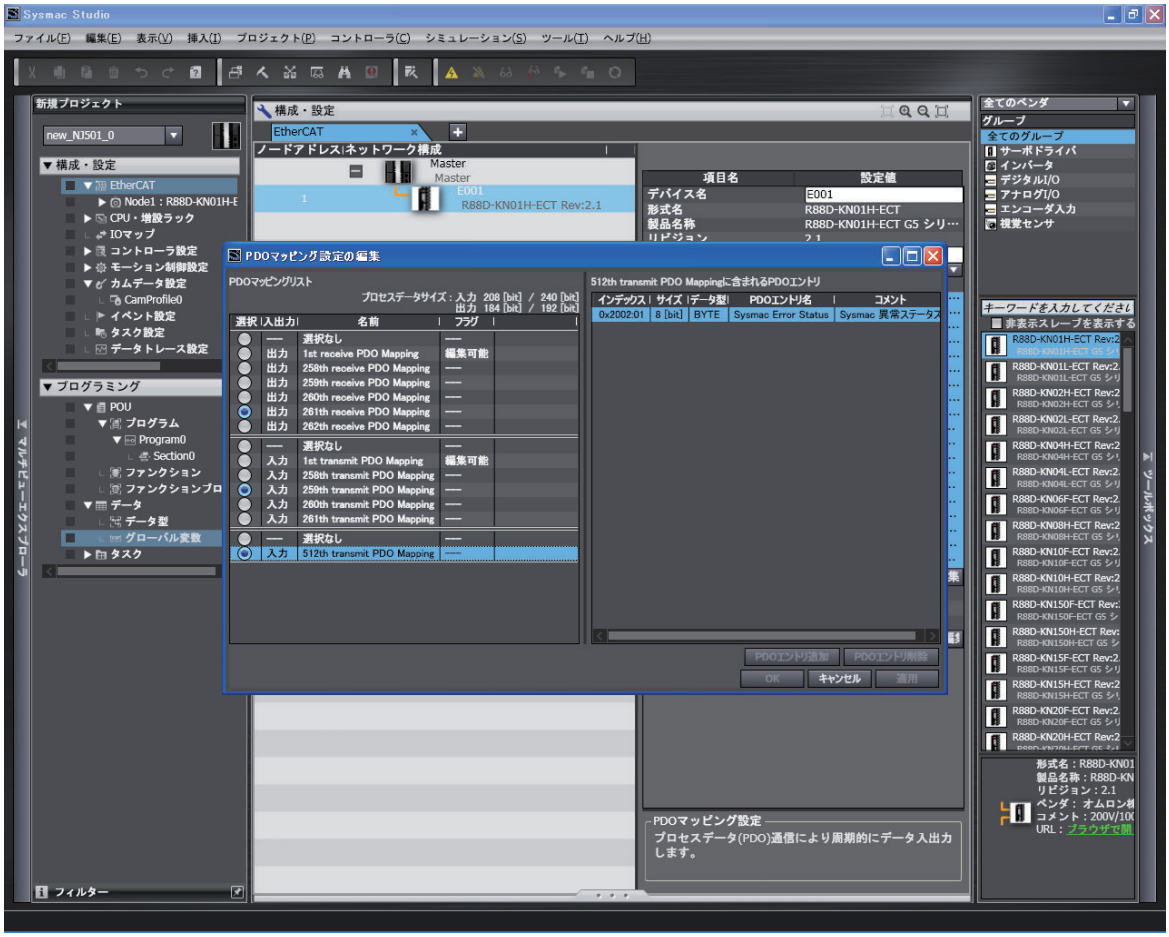
バックラッシュ機能の詳細については、『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル(SBCE-377)』、または『AC サーボモータ／ドライバ 1S シリーズ EtherCAT 通信内蔵タイプ 安全機能対応 ユーザーズマニュアル(SBCE-438)』を参照してください。

PDO マッピング

MC 機能モジュールのサーボ軸として制御するための PDO マッピングについて説明します。

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。

使用する PDO マップは、Sysmac Studio の［EtherCAT］構成編集画面の［PDO マッピング設定の編集］で選択します。



Sysmac Studio のデフォルトでは、下記の PDO マップが選択されています。

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)

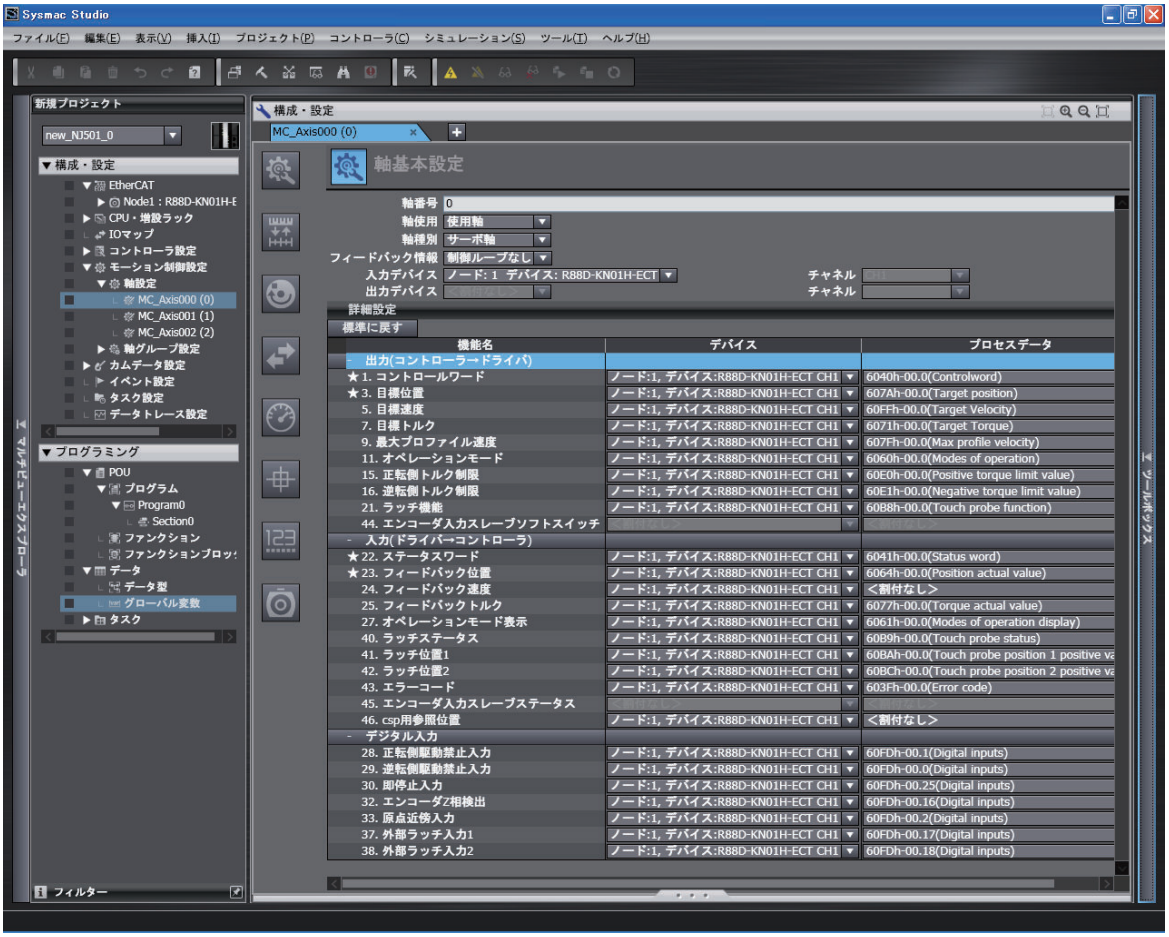
参考

推奨サーボドライバのオムロン製 形 R88D-1SN□□□-ECT、または形 R88D-1SAN□□□-ECT 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。

MC 機能モジュールの機能とプロセスデータの関係

MC 機能モジュールが持つ機能は、プロセスデータのオブジェクトの情報と関係があります。
MC 機能モジュールが使用する EtherCAT スレーブの構成や使用する機能によっては、これらの関係の変更が必要な場合があります。

Sysmac Studio の［軸基本設定］画面の［詳細設定］をクリックすると設定画面が開きます。



参考

推奨サーボドライバのオムロン製 形 R88D-1SN□□□-ECT、または形 R88D-1SAN□□□-ECT の場合、機能とプロセスデータとの関係は、Sysmac Studio のデフォルトのままでも使用することができます。

● 出力（コントローラ→ドライバ）の設定

MC 機能モジュールからサーボドライバへの指令データの設定です。
下記は Sysmac Studio のデフォルト設定です。（O印は必須のオブジェクト）

	機能名	プロセスデータ	説明
O	コントロールワード	6040Hex-00.0 (Controlword)	サーボドライバの状態を制御するためのデータです。 [6040Hex: Controlword] を設定してください。
O	目標位置	607AHex-00.0 (Target position)	位置制御での目標位置です。 [607AHex: Target position] を設定してください。
	目標速度	60FFHex-00.0 (Target velocity)	速度制御での目標速度です。 MC_SyncMoveVelocity（サイクリック同期速度制御）命令など、サイクリック同期速度モードでドライバへ出力する場合に必要です。 通常は［60FFHex: Target velocity］を設定してください。
	目標トルク	6071Hex-00.0 (Target torque)	トルク制御での目標トルクです。 MC_TorqueControl（トルク制御）命令など、サイクリック同期トルクモードでドライバへ出力する場合に必要です。 通常は［6071Hex: Target torque］を設定してください。

機能名	プロセスデータ	説明
最大プロファイル速度	607FHex-00.0 (Max profile velocity)	トルク制御時の速度制限値です。 MC_TorqueControl（トルク制御）命令など、サイクリック同期トルクモードで速度制限する場合に必要です。 通常は［607FHex: Max profile velocity］を設定してください。
オペレーションモード	6060Hex-00.0 (Modes of operation)	制御モードを切り替えるためのデータです。 MC_SyncMoveVelocity（サイクリック同期速度制御）命令や MC_TorqueControl（トルク制御）命令など、サイクリック同期位置モード以外の制御モードへ切り替える場合に必要です。 通常は［6060Hex: Modes of operation］を設定してください。*1
正方向トルク制限値	60E0Hex-00.0 (Positive torque limit value)	正方向側のトルク制限値です。 MC_SetTorqueLimit（トルク制限）命令など、ドライバの出力トルクを制限する場合に必要です。 通常は［60E0Hex: Positive torque limit value］を設定してください。
負方向トルク制限値	60E1Hex-00.0 (Negative torque limit value)	負方向側のトルク制限値です。 MC_SetTorqueLimit（トルク制限）命令など、ドライバの出力トルクを制限する場合に必要です。 通常は［60E1Hex: Negative torque limit value］を設定してください。
ラッチ機能	60B8Hex-00.0 (Touch probe function)	ラッチ機能を制御するためのデータです。 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_TouchProbe（外部ラッチ有効）命令、MC_MoveLink（台形パターンカム）命令など、ラッチ機能を使用する場合に必要です。 通常は［60B8Hex: Touch probe function］を設定してください。

*1. ［6060Hex: Modes of operation］を設定する場合は、［6061Hex: Modes of operation display］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。

使用上の注意

- 設定を変更する場合は、MC 機能モジュールの機能とプロセスデータの組み合わせが、意図したとおりに動作するか確認してください。
- オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプ以外を使用する場合は、オペレーションモード(6060Hex)を必ず設定してください。

● 入力（ドライバ→コントローラ）の設定

サーボドライバから MC 機能モジュールへのステータスデータです。
下記は Sysmac Studio のデフォルト設定です。（O印は必須のオブジェクト）

	機能名	プロセスデータ	説明
O	ステータスワード	6041Hex-00.0 (Statusword)	サーボドライバの状態です。 ［6041Hex: Statusword］を設定してください。
O	現在位置	6064Hex-00.0 (Position actual value)	現在位置の表示です。 ［6064Hex: Position actual value］を設定してください。
	現在速度	設定なし *1	現在速度の表示です。 使用する場合、通常は［606CHex: Velocity actual value］を設定してください。

機能名	プロセスデータ	説明
現在トルク	6077Hex (Torque actual value)	現在のトルク表示です。 MC_TorqueControl（トルク制御）命令など、サイクリック同期トルクモードでドライバへ出力する場合に必要です。 通常は［6077Hex: Torque actual value］を設定してください。
オペレーションモード表示	6061Hex-00.0 (Modes of operation display)	制御モードの表示です。 MC_SyncMoveVelocity（サイクリック同期速度制御）命令や MC_TorqueControl（トルク制御）命令など、サイクリック同期位置モード以外の制御モードへ切り替える場合に必要です。 通常は［6061Hex: Modes of operation display］を設定してください。*2
ラッチステータス	60B9Hex-00.0 (Touch probe status)	ラッチ機能の状態の表示です。 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_TouchProbe（外部ラッチ有効）命令、MC_MoveLink（台形パターンカム）命令など、ラッチ機能を使用する場合に必要です。 通常は［60B9Hex: Touch probe status］を設定してください。
ラッチ 1 位置	60BAHex-00.0 (Touch probe pos1 pos value)	ラッチ機能 1 でラッチした位置です。 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_TouchProbe（外部ラッチ有効）命令、MC_MoveLink（台形パターンカム）命令など、ラッチ機能を使用する場合に必要です。 通常は［60BAHex: Touch probe pos1 pos value］を設定してください。
ラッチ 2 位置	60BCHex-00.0 (Touch probe pos2 pos value)	ラッチ機能 2 でラッチした位置です。 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_TouchProbe（外部ラッチ有効）命令、MC_MoveLink（台形パターンカム）命令など、ラッチ機能を使用する場合に必要です。 通常は［60BCHex: Touch probe pos2 pos value］を設定してください。
エラーコード	603FHex-00.0 (Error code)	ドライバ内の異常コードです。 通常は［603FHex: Error code］を設定してください。
csp 用参照位置	設定なし	csp モード切替時の参照位置です。速度制御モード(CSV)を使用する命令やトルク制御モード(CST)を使用する命令で参照されるデータです。 オムロン製サーボモータ／ドライバ 1S シリーズで使用することができます。*3

*1. 必要な場合は、選択するプロセスデータを PDO にマッピングしたうえで設定してください。606CHex-00.0(Velocity actual value)が標準的な設定です。
*2. ［6061Hex: Modes of operation display］を設定する場合は、［6060Hex: Modes of operation］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。
*3. 使用する場合、［3010Hex-87Hex: Reference Position for csp］を PDO にマッピングしてください。

使用上の注意

- 設定を変更する場合は、MC 機能モジュールの機能とプロセスデータの組み合わせが、意図したとおりに動作するか確認してください。
- オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプ以外を使用する場合は、オペレーションモード表示(6061Hex)を必ず設定してください。
- オムロン製サーボドライバ 1S シリーズを使用し、MC_SyncMoveVelocity（サイクリック同期速度制御）命令で制御モードを切り替える場合は、「csp 用参照位置」をマッピングしてください。

● デジタル入力の設定

MC 機能モジュールが使用するサーボドライバの入力信号です。

機能名	プロセスデータ	説明
正方向駆動禁止入力	60FDHex-00.1 (Digital inputs)	正方向限界入力として使用する信号です。 通常は［60FDHex-00: Digital inputs］の［ビット 1: Positive limit switch］を設定してください。
負方向駆動禁止入力	60FDHex-00.0 (Digital inputs)	負方向限界入力として使用する信号です。 通常は［60FDHex-00: Digital inputs］の［ビット 0: Negative limit switch］を設定してください。
異常停止入力	60FDHex-00.25 (Digital inputs)	即停止入力として使用する信号です。 オムロン製サーボドライバ 1S シリーズは［60FDHex-00: Digital inputs］の［ビット 25: Error Stop Input］を設定してください。
エンコーダ Z 相検出	60FDHex-00.16 (Digital inputs)	Z 相入力の検出状態表示です。 オムロン製サーボモータ／ドライバ 1S シリーズは［60FDHex-00: Digital inputs］の［ビット 16: Encoder Phase Z Detection］を設定してください。
原点近傍入力	60FDHex-00.2 (Digital inputs)	原点近傍入力として使用する信号です。 通常は［60FDHex-00: Digital inputs］の［ビット 2: Home switch］を設定してください。
外部ラッチ入力 1	60FDHex-00.17 (Digital inputs)	外部ラッチ入力 1 として使用する信号の状態表示です。 オムロン製サーボドライバ 1S シリーズは［60FDHex-00: Digital inputs］の［ビット 17: External Latch Input 1］を設定してください。
外部ラッチ入力 2	60FDHex-00.18 (Digital inputs)	外部ラッチ入力 2 として使用する信号の状態表示です。 オムロン製サーボドライバ 1S シリーズは［60FDHex-00: Digital inputs］の［ビット 18: External Latch Input 2］を設定してください。

使用上の注意

設定を変更する場合は、MC 機能モジュールの機能とプロセスデータの組み合わせが、意図したとおりに動作するか確認してください。

レ 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	**(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。

サーボドライバ側のオブジェクトの設定

MC 機能モジュールの制御機能を使用するためのオムロン製サーボドライバ 1S シリーズ側の設定を下記に示します。
サーボドライバの各種機能を使用する場合は、サーボドライバのマニュアルに従って各種機能に関連するオブジェクトも設定してください。

インデックス	サブインデックス	名称	推奨設定値	説明
3001Hex	−	機械	−	サーボ側ギア比は 1:1 とし、ユーザ単位はコントローラ側で設定します。*1
	05Hex	電子ギア比分子	1	
	06Hex	電子ギア比分母	1	
3330Hex	−	トルク制限	−	PCL と NCL は、OFF の状態で［60E0Hex］と［60E1Hex］の値で制限します。
	01Hex	切り替え選択	2	
3A00Hex	−	原点復帰	−	サーボ側のオフセット値は「0」で使用します。
	06Hex	エンコーダ原点位置オフセット	0	
3B10Hex	−	駆動禁止	−	サーボ側は駆動禁止入力を無効にし、コントローラ側で処理します。
	01Hex	有効	0	
3B11Hex	−	ソフトウェアリミット	−	正方向、負方向ともソフトウェアリミットを無効にします。
	01Hex	有効選択	0	

インデックス	サブインデックス	名称	推奨設定値	説明
3B30Hex	－	ラッチ機能 1	－	Touch probe1=外部ラッチ信号 1、Touch probe2=外部ラッチ信号 2 に設定します。
	01Hex	ラッチ 1 トリガ選択	1	
3B31Hex	－	ラッチ機能 2	－	
	01Hex	ラッチ 2 トリガ選択	2	
4020Hex	－	警告カスタマイズ	－	要因消滅で自動的にリセットされる設定にします。
	04Hex	警告保持選択	0	
4510Hex	－	エンコーダ	－	絶対値エンコーダとして使用します。 多回転カウンタオーバーは無視します。
	01Hex	絶対値エンコーダ動作選択	2	
4630Hex	－	正方向駆動禁止	－	正方向駆動禁止を汎用入力 2(IN2)に負論理（b 接点）で割り付けます。
	01Hex	ポート選択	2	
	02Hex	論理選択	1	
4631Hex	－	負方向駆動禁止	－	負方向駆動禁止を汎用入力 3(IN3)に負論理（b 接点）で割り付けます。
	01Hex	ポート選択	3	
	02Hex	論理選択	1	
4632Hex	－	外部ラッチ入力 1	－	外部ラッチ入力 1 を汎用入力 7(IN7)に正論理（a 接点）で割り付けます。
	01Hex	ポート選択	7	
	02Hex	論理選択	0	
4633Hex	－	外部ラッチ入力 2	－	外部ラッチ入力 2 を汎用入力 8(IN8)に正論理（a 接点）で割り付けます。
	01Hex	ポート選択	8	
	02Hex	論理選択	0	
4634Hex	－	原点近傍入力	－	原点近傍入力を汎用入力 4(IN4)に正論理（a 接点）で割り付けます。
	01Hex	ポート選択	4	
	02Hex	論理選択	0	

*1. ユニットバージョン 1.10 以前の CPU ユニットの場合、オムロン製サーボモータ 1S シリーズを最高回転数で動作させることができません。1S シリーズを最高回転数で動作させるには、電子ギア比を 2:1 以上に設定してください。

A-2　サーボドライバ G5 シリーズとの接続

オムロン製サーボドライバ G5 シリーズ EtherCAT 通信内蔵タイプとの接続について説明します。

A-2-1　サーボドライバとの配線

サーボドライバとは、EtherCAT 通信で接続します。
接続方法については、『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)』を参照してください。

A-2-2　サーボドライバの設定

ここでは、MC 機能モジュールの制御対象であるオムロン製サーボドライバ G5 シリーズ EtherCAT 通信内蔵タイプと接続した場合のサーボドライバ側の設定概要について説明します。
サーボドライバの詳細については、『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)』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。

推奨サーボドライバ

MC 機能モジュールのすべての機能を使用できるサーボドライバのユニットバージョンを下表に示します。

メーカ	対象形式	対象ユニットバージョン
オムロン	形 R88D-KN□□□-ECT	Ver.2.1 以降
	形 R88D-KN□□□-ECT-L	Ver.1.1 以降

参考

- 推奨対象外のオムロン製サーボドライバ G5 シリーズ EtherCAT 通信内蔵タイプのユニットバージョンも使用することができます。使用できる機能はサーボドライバ側の仕様により異なります。Sysmac Studio で使用する機能とオブジェクトディクショナリを設定してください。
- オムロン製サーボドライバ G5 シリーズの形 R88D-KN□□□-ECT-R（ユニットバージョン Ver.1.0）は、位置制御 (cyclic synchronous position)にだけ対応しているサーボドライバです。
  速度制御 (cyclic synchronous velocity)あるいは、トルク制御 (cyclic synchronous torque)が不要なアプリケーションでは、使用することができます。
  機能の詳細は、『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-360)』を参照してください。
- オムロン製サーボモータ／ドライバ G5 シリーズのうち、シリンダタイプのユニットバージョン Ver.2.0 以前を使用する場合は、ノードアドレススイッチを「00」以外に設定してください。「00」に設定するとネットワーク構成の異常となります。
- オムロン製サーボドライバ G5 シリーズのユニットバージョンによる相違は、『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。

外部入力信号の割付

MC 機能モジュールでは、サーボドライバの汎用入力を下記の入力信号として使用します。

- 即停止入力
- 正方向限界入力
- 負方向限界入力
- 原点近傍入力
- 外部ラッチ機能のトリガ信号（ラッチ入力 1／ラッチ入力 2）

● 正方向限界入力／負方向限界入力／原点近傍入力の割付

オムロン製サーボドライバ G5 シリーズの出荷時設定では、各入力信号は下表のように割り付けられています。

信号名称	入力信号
即停止入力	サーボドライバの汎用入力 1（IN1: CN1 コネクタのピン No.5 b 接）
正方向限界入力	サーボドライバの汎用入力 2（IN2: CN1 コネクタのピン No.7 b 接）*1
負方向限界入力	サーボドライバの汎用入力 3（IN3: CN1 コネクタのピン No.8 b 接）*2
原点近傍入力	サーボドライバの汎用入力 4（IN4: CN1 コネクタのピン No.9 a 接）

*1. サーボドライバ側の信号名称は「正転側駆動禁止入力」です。
*2. サーボドライバ側の信号名称は「逆転側駆動禁止入力」です。

● 外部ラッチ機能のトリガ信号の割付

オムロン製サーボドライバ G5 シリーズの出荷時設定の場合、下表の入力信号が外部ラッチ機能のトリガ信号に割り付けられています。

MC_TouchProbe 命令の入力変数「TriggerInput」（トリガ入力条件）の設定			外部ラッチ機能のトリガ信号
Mode	InputDrive	LatchID	
0: mcDrive	0: mcEncoderMark	-	エンコーダの Z 相
	1: mcEXT	1: mcLatch1	サーボドライバの汎用入力 7（IN7: CN1 コネクタのピン No.12 a 接）*1
		2: mcLatch2	サーボドライバの汎用入力 6（IN6: CN1 コネクタのピン No.11 a 接）*2
1: mcController	-	-	TriggerVariable（トリガ変数）で指定した変数

*1. サーボドライバ側の信号名称は「外部ラッチ入力 1」です。
*2. サーボドライバ側の信号名称は「外部ラッチ入力 2」です。

バックラッシュ補正機能

MC 機能モジュールはバックラッシュ補正機能がありません。
バックラッシュ補正が必要な場合は、サーボドライバ側の補正機能を使用します。
サーボドライバ側で設定が必要なオブジェクトを下記に示します。

インデックス	名称	説明
3704Hex	バックラッシュ補正選択	位置制御時のバックラッシュ補正の有効／無効ならびに補正時の動作方向を設定します。 出荷時設定は「無効」です。
3705Hex	バックラッシュ補正量	位置制御時のバックラッシュ補正量を設定します。
3706Hex	バックラッシュ補正時定数	位置制御時のバックラッシュ補正時定数を設定します。

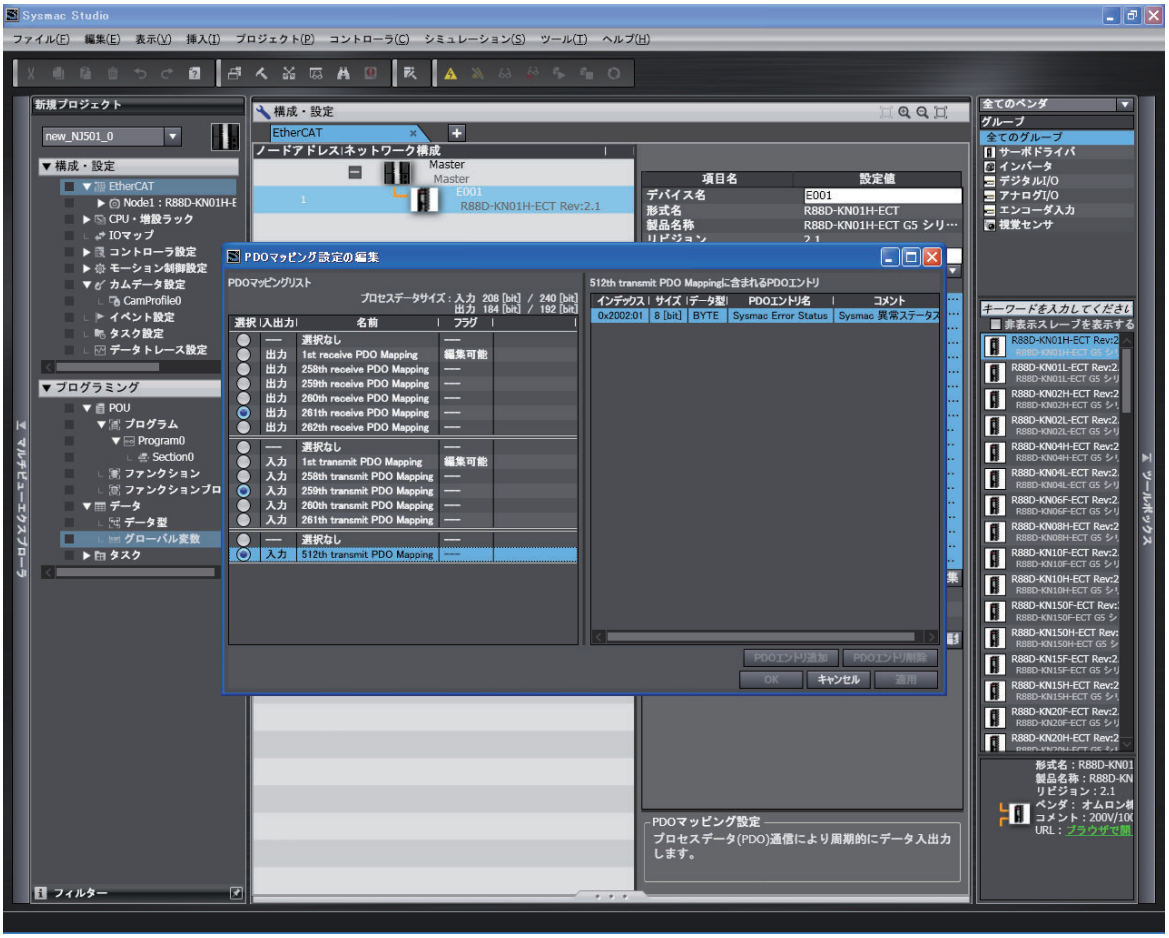
バックラッシュ機能の詳細については、『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵タイプ ユーザーズマニュアル (SBCE-365)』および、『AC サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプ ユーザーズマニュアル (SBCE-366)』を参照してください。

PDO マッピング

MC 機能モジュールのサーボ軸として制御するための PDO マッピングについて説明します。

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。

使用する PDO マップは、Sysmac Studio の［EtherCAT］構成編集画面の［PDO マッピング設定の編集］で選択します。



Sysmac Studio のデフォルトでは、下記の PDO マップが選択されています。

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)、正転側トルク制限値(60E0Hex)、逆転側トルク制限値(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)、フィードバック位置(6064Hex)、フィードバックトルク(6077Hex)、オペレーションモード表示(6061Hex)、ラッチステータス(60B9Hex)、ラッチ位置 1(60BAHex)、ラッチ位置 2(60BCHex)、デジタル入力(60FDHex)

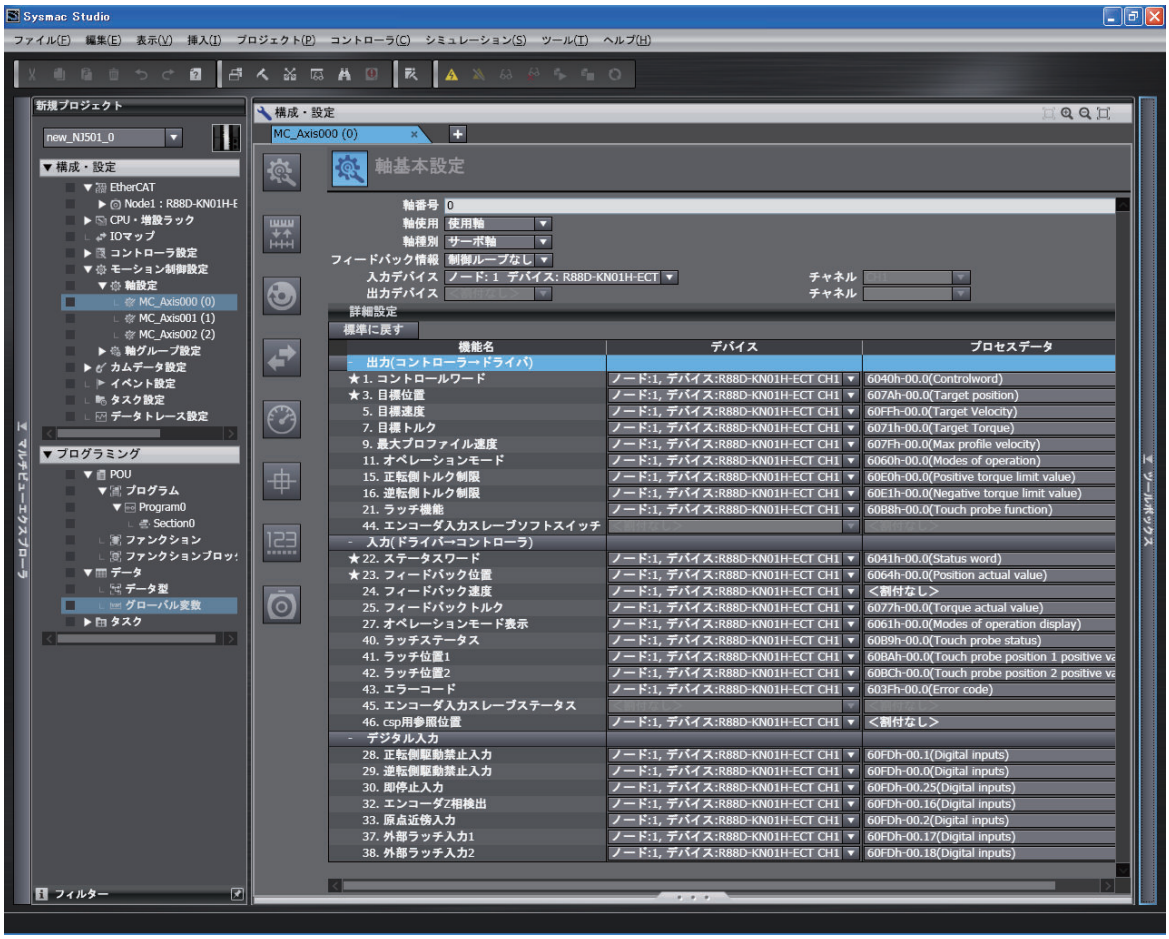
参考

- 推奨サーボドライバのオムロン製 形 R88D-KN□□□-ECT（Ver.2.1 以降）、または形 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。
- オムロン製の形 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。

MC 機能モジュールの機能とプロセスデータの関係

MC 機能モジュールが持つ機能は、プロセスデータのオブジェクトの情報と関係があります。
MC 機能モジュールが使用する EtherCAT スレーブの構成や使用する機能によっては、これらの関係の変更が必要な場合があります。

Sysmac Studio の［軸基本設定］画面の［詳細設定］をクリックすると設定画面が開きます。



参考

推奨サーボドライバのオムロン製 形 R88D-KN□□□-ECT（Ver.2.1 以降）、または形 R88D-KN□□□-ECT-L（Ver.1.1 以降）の場合、機能とプロセスデータとの関係は、Sysmac Studio のデフォルトのままでも使用することができます。

● 出力（コントローラ→ドライバ）の設定

MC 機能モジュールからサーボドライバへの指令データの設定です。

下記は Sysmac Studio のデフォルト設定です。（O印は必須のオブジェクト）

	機能名	プロセスデータ	説明
O	コントロールワード	6040Hex-00.0 (Controlword)	サーボドライバの状態を制御するためのデータです。 [6040Hex: Controlword] を設定してください。
O	目標位置	607AHex-00.0 (Target position)	位置制御での目標位置です。 [607AHex: Target position] を設定してください。
	目標速度	60FFHex-00.0 (Target velocity)	速度制御での目標速度です。 MC_SyncMoveVelocity（サイクリック同期速度制御）命令など、サイクリック同期速度モードでドライバへ出力する場合に必要です。 通常は［60FFHex: Target velocity］を設定してください。
	目標トルク	6071Hex-00.0 (Target torque)	トルク制御での目標トルクです。 MC_TorqueControl（トルク制御）命令など、サイクリック同期トルクモードでドライバへ出力する場合に必要です。 通常は［6071Hex: Target torque］を設定してください。
	最大プロファイル速度	607FHex-00.0 (Max profile velocity)	トルク制御時の速度制限値です。 MC_TorqueControl（トルク制御）命令など、サイクリック同期トルクモードで速度制限する場合に必要です。 通常は［607FHex: Max profile velocity］を設定してください。
	オペレーションモード	6060Hex-00.0 (Modes of operation)	制御モードを切り替えるためのデータです。 MC_SyncMoveVelocity（サイクリック同期速度制御）命令や MC_TorqueControl（トルク制御）命令など、サイクリック同期位置モード以外の制御モードへ切り替える場合に必要です。 通常は［6060Hex: Modes of operation］を設定してください。*1
	正転側トルク制限値	60E0Hex-00.0 (Positive torque limit value)	正方向側のトルク制限値です。 MC_SetTorqueLimit（トルク制限）命令など、ドライバの出力トルクを制限する場合に必要です。 通常は［60E0Hex: Positive torque limit value］を設定してください。
	逆転側トルク制限値	60E1Hex-00.0 (Negative torque limit value)	負方向側のトルク制限値です。 MC_SetTorqueLimit（トルク制限）命令など、ドライバの出力トルクを制限する場合に必要です。 通常は［60E1Hex: Negative torque limit value］を設定してください。
	ラッチ機能	60B8Hex-00.0 (Touch probe function)	ラッチ機能を制御するためのデータです。 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_TouchProbe（外部ラッチ有効）命令、MC_MoveLink（台形パターンカム）命令など、ラッチ機能を使用する場合に必要です。 通常は［60B8Hex: Touch probe function］を設定してください。

*1. ［6060Hex: Modes of operation］を設定する場合は、［6061Hex: Modes of operation display］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。

使用上の注意

- 推奨サーボドライバ以外のオムロン製サーボドライバ G5 シリーズ EtherCAT 通信内蔵タイプのユニットバージョンを接続した場合、対応する機能がない場合があります。
  接続するサーボドライバのマニュアルを確認してください。
- 設定を変更する場合は、MC 機能モジュールの機能とプロセスデータの組み合わせが、意図したとおりに動作するか確認してください。
- オムロン製サーボドライバ 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)を必ず設定してください。
- オムロン製サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵タイプでフルクローズ制御を行う場合、オブジェクトの合計サイズが 12 バイト以下となるように設定してください。

● 入力（ドライバ→コントローラ）の設定

サーボドライバから MC 機能モジュールへのステータスデータです。
下記は Sysmac Studio のデフォルト設定です。（O印は必須のオブジェクト）

	機能名	プロセスデータ	説明
O	ステータスワード	6041Hex-00.0 (Statusword)	サーボドライバの状態です。 [6041Hex: Statusword] を設定してください。
O	フィードバック位置	6064Hex-00.0 (Position actual value)	現在位置の表示です。 [6064Hex: Position actual value] を設定してください。
	フィードバック速度	設定なし *1	現在速度の表示です。 使用する場合、通常は［606CHex: Velocity actual value］を設定してください。
	フィードバックトルク	6077Hex (Torque actual value)	現在のトルク表示です。 MC_TorqueControl（トルク制御）命令など、サイクリック同期トルクモードでドライバへ出力する場合に必要です。 通常は［6077Hex: Torque actual value］を設定してください。
	オペレーションモード表示	6061Hex-00.0 (Modes of operation display)	制御モードの表示です。 MC_SyncMoveVelocity（サイクリック同期速度制御）命令や MC_TorqueControl（トルク制御）命令など、サイクリック同期位置モード以外の制御モードへ切り替える場合に必要です。 通常は［6061Hex: Modes of operation display］を設定してください。*2
	ラッチステータス	60B9Hex-00.0 (Touch probe status)	ラッチ機能の状態の表示です。 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_TouchProbe（外部ラッチ有効）命令、MC_MoveLink（台形パターンカム）命令など、ラッチ機能を使用する場合に必要です。 通常は［60B9Hex: Touch probe status］を設定してください。
	ラッチ位置 1	60BAHex-00.0 (Touch probe pos1 pos value)	ラッチ機能 1 でラッチした位置です。 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_TouchProbe（外部ラッチ有効）命令、MC_MoveLink（台形パターンカム）命令など、ラッチ機能を使用する場合に必要です。 通常は［60BAHex: Touch probe pos1 pos value］を設定してください。

機能名	プロセスデータ	説明
ラッチ位置 2	60BCHex-00.0 (Touch probe pos2 pos value)	ラッチ機能 2 でラッチした位置です。 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_TouchProbe（外部ラッチ有効）命令、MC_MoveLink（台形パターンカム）命令など、ラッチ機能を使用する場合に必要です。 通常は［60BCHex: Touch probe pos2 pos value］を設定してください。
エラーコード	603FHex-00.0 (Error code)	ドライバ内の異常コードです。 通常は［603FHex: Error code］を設定してください。
csp 用参照位置	設定なし	csp モード切替時の参照位置です。 速度制御モード(CSV)を使用する命令やトルク制御モード(CST)を使用する命令で参照されるデータです。 オムロン製サーボモータ／ドライバ G5 シリーズのうち、シリンダタイプのユニットバージョン Ver.2.1 以降で使用することができます。*3 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では、このオブジェクトに対応していません。

*1. 必要な場合は、選択するプロセスデータを PDO にマッピングしたうえで設定してください。［606CHex-00.0（Velocity actual value)］が標準的な設定です。
*2. ［6061Hex: Modes of operation display］を設定する場合は、［6060Hex: Modes of operation］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。
*3. 使用する場合は、［4020Hex: Reference Position for csp］を PDO にマッピングしてください。

使用上の注意

- 推奨サーボドライバ以外のオムロン製サーボドライバ G5 シリーズ EtherCAT 通信内蔵タイプのユニットバージョンを接続した場合、対応する機能がない場合があります。接続するサーボドライバのマニュアルを確認してください。
- 設定を変更する場合は、MC 機能モジュールの機能とプロセスデータの組み合わせが、意図したとおりに動作するか確認してください。
- オムロン製サーボドライバ 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表示(6061Hex)を必ず設定してください。
- オムロン製サーボドライバ G5 シリーズを使用し、MC_SyncMoveVelocity（サイクリック同期速度制御）命令で制御モードを切り替える場合は、「csp 用参照位置」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 シリーズで「csp 用参照位置」を使用する場合、プライマリ周期は 1ms 以上としてください。また、G5 シリーズの電子ギア比は 1:1 としてください。詳細は、『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)』、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。

● デジタル入力の設定

MC 機能モジュールが使用するサーボドライバの入力信号です。

機能名	プロセスデータ	説明
正転側駆動禁止入力	60FDHex-00.1 (Digital inputs)	正方向限界入力として使用する信号です。 通常は［60FDHex-00: Digital inputs］の［ビット 1: Positive limit switch］を設定してください。
逆転側駆動禁止入力	60FDHex-00.0 (Digital inputs)	負方向限界入力として使用する信号です。 通常は［60FDHex-00: Digital inputs］の［ビット 0: Negative limit switch］を設定してください。
即停止入力	60FDHex-00.25 (Digital inputs)	即停止入力として使用する信号です。 オムロン製サーボドライバ G5 シリーズでは［60FDHex-00: Digital inputs］の［ビット 25: Immediate Stop Input］を設定してください。

機能名	プロセスデータ	説明
エンコーダ Z 相検出	60FDHex-00.16 (Digital inputs)	Z 相入力の検出状態表示です。 オムロン製サーボモータ／ドライバ G5 シリーズのうち、シリンダタイプのユニットバージョン Ver.2.1 以降では［60FDHex-00: Digital inputs］の［ビット 16: Encoder Phase Z Detection］を設定してください。 サーボドライバ G5 シリーズのうち、リニアモータタイプではこのオブジェクトに対応していません。
原点近傍入力	60FDHex-00.2 (Digital inputs)	原点近傍入力として使用する信号です。 通常は［60FDHex-00: Digital inputs］の［ビット 2: Home switch］を設定してください。
外部ラッチ入力 1	60FDHex-00.17 (Digital inputs)	外部ラッチ入力 1 として使用する信号の状態表示です。 オムロン製サーボドライバ G5 シリーズでは［60FDHex-00: Digital inputs］の［ビット 17: External Latch Input 1］を設定してください。
外部ラッチ入力 2	60FDHex-00.18 (Digital inputs)	外部ラッチ入力 2 として使用する信号の状態表示です。 オムロン製サーボドライバ G5 シリーズでは［60FDHex-00: Digital inputs］の［ビット 18: External Latch Input 2］を設定してください。

使用上の注意

- 推奨サーボドライバ以外のオムロン製サーボドライバ G5 シリーズ EtherCAT 通信内蔵タイプのユニットバージョンを接続した場合、対応する機能がない場合があります。接続するサーボドライバのマニュアルを確認してください。
- 設定を変更する場合は、MC 機能モジュールの機能とプロセスデータの組み合わせが、意図したとおりに動作するか確認してください。

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	**(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 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。

サーボドライバ側のオブジェクトの設定

MC 機能モジュールの制御機能を使用するためのオムロン製サーボドライバ G5 シリーズ側の設定を下記に示します。
サーボドライバの各種機能を使用する場合は、サーボドライバのマニュアルに従って各種機能に関連するオブジェクトも設定してください。

インデックス	サブインデックス	名称	推奨設定値	説明
3013Hex	00Hex	第 1 外部トルク制限 *1	1388Hex	初期値=500.0%
3015Hex *2	00Hex	絶対値エンコーダ使用時動作切り替え	0002Hex	絶対値として用い、多回転カウンタオーバーを無視します。
3317Hex	00Hex	速度制限選択	0001Hex	トルク制御時の速度制限方法は、[607FHex: Max profile velocity] と [3321Hex: 速度制限値設定] の小さい方の値で制限します。
3323Hex	00Hex	外部フィードバックパルスタイプ選択	0000Hex または 0001Hex, 0002Hex *3	使用する外部スケールタイプの設定です。初期値の 0000Hex は［90° 位相差出力タイプ］です。
3324Hex *2	00Hex	外部フィードバックパルス分周比分子	00000000Hex	モータ 1 回転あたりのエンコーダ分解能を設定します。「0」に設定することにより、自動設定されます。
3401Hex	00Hex	入力信号選択 2	00818181Hex	正転側駆動禁止入力（b 接）
3402Hex	00Hex	入力信号選択 3	00828282Hex	逆転側駆動禁止入力（b 接）
3403Hex	00Hex	入力信号選択 4	00222222Hex	原点近傍入力（a 接）
3404Hex	00Hex	入力信号選択 5	002B2B2BHex	外部ラッチ信号 3（a 接）
3405Hex	00Hex	入力信号選択 6	00212121Hex	外部ラッチ信号 2（a 接）
3406Hex	00Hex	入力信号選択 7	00202020Hex	外部ラッチ信号 1（a 接）
3504Hex	00Hex	駆動禁止入力選択	0001Hex	サーボ側は駆動禁止入力を無効にします。MC 機能モジュール側で行います。
3508Hex	00Hex	不足電圧エラー選択	0001Hex	主電源不足電圧が発生し、停止します。
3521Hex	00Hex	トルク制限選択 *1	0006Hex	正逆とも 2 個ずつの制限値を持ちます。切替は、PCL, NCL で行います。
3522Hex	00Hex	第 2 外部トルク制限 *1	1388Hex	初期値=500.0%
3703Hex	00Hex	トルク制限中出力設定 *1	0001Hex	トルク指令値を除くトルク制限で ON します。
3801Hex	00Hex	ソフトウェアリミット機能	0003Hex	両側ソフトウェアリミット無効にします。
3758Hex	00Hex	ラッチトリガ選択	0100Hex	Touch probe1 =外部ラッチ信号 1、Touch probe2 =外部ラッチ信号 2 にします。
3759Hex	00Hex	警告保持選択	0000Hex	警告要因がなくなると警告は自動的にクリアされます。
607CHex	00Hex	エンコーダ原点位置オフセット	00000000Hex	サーボ側のオフセット値は「0」で使用。
6091Hex	01Hex	電子ギア比分子	00000001Hex	サーボドライバ側のギア比は 1:1 とします。同様の機能は、MC 機能モジュール側で設定します。
	02Hex	電子ギア比分母	00000001Hex	

*1. 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では、「トルク」を「推力」に読み替えてください。
*2. 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では、このオブジェクトに対応していません。
*3. オムロン製サーボモータ／ドライバ G5 シリーズでフルクローズ制御する場合、またはオムロン製サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプの場合、使用する外部スケールタイプに合わせて設定してください。

A-3　エンコーダ入力ターミナルとの接続

オムロン製 GX シリーズ EtherCAT スレーブ エンコーダ入力ターミナルとの接続について説明します。

A-3-1　エンコーダ入力ターミナルとの配線

エンコーダ入力ターミナルとは、EtherCAT 通信で接続します。

接続方法については、『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)』を参照してください。

A-3-2　エンコーダ入力ターミナルの設定

ここでは、MC 機能モジュールの制御対象であるオムロン製 EtherCAT リモート I/O ターミナル GX シリーズ形 GX-EC0211/EC0241 と接続した場合のエンコーダ入力ターミナル側の設定概要について説明します。

エンコーダ入力ターミナルの詳細については、『形 GX シリーズ EtherCAT スレーブ ユーザーズマニュアル (SBCD-350)』を参照してください。

推奨エンコーダ入力ターミナル

MC 機能モジュールのエンコーダ軸として、すべての機能を使用できるエンコーダ入力ターミナルのユニットバージョンを下表に示します。

メーカ	対象形式	対象ユニットバージョン
オムロン	形 GX-EC0211	Ver.1.1 以降
	形 GX-EC0241	Ver.1.1 以降

参考

- エンコーダ軸として使用できる EtherCAT スレーブは、オムロン製の形 GX-EC0211/ EC0241 だけです。
- オムロン製の形 GX-EC0211/ EC0241（ユニットバージョン Ver.1.0）もエンコーダ軸として使用できますが、Sysmac デバイスではありません。
  ユニットバージョン Ver.1.0 を使用する場合、ノードアドレススイッチは「00」以外に設定してください。「00」に設定するとネットワーク構成の異常となります。
  機能の詳細は、『形 GX シリーズ EtherCAT スレーブ ユーザーズマニュアル (SBCD-350)』を参照してください。
  Sysmac デバイスについては、『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。

外部入力信号

エンコーダ軸としてのすべての機能を使用する場合、エンコーダ入力ターミナル側の下記の入力信号を使用します。

- カウンタ A 相
- カウンタ B 相
- カウンタ Z 相

- ラッチ入力(A/B)

カウンタは 2 チャネルあり、チャネルごとに 2 つの外部ラッチがあります。実際に使用する入力信号を配線してください。

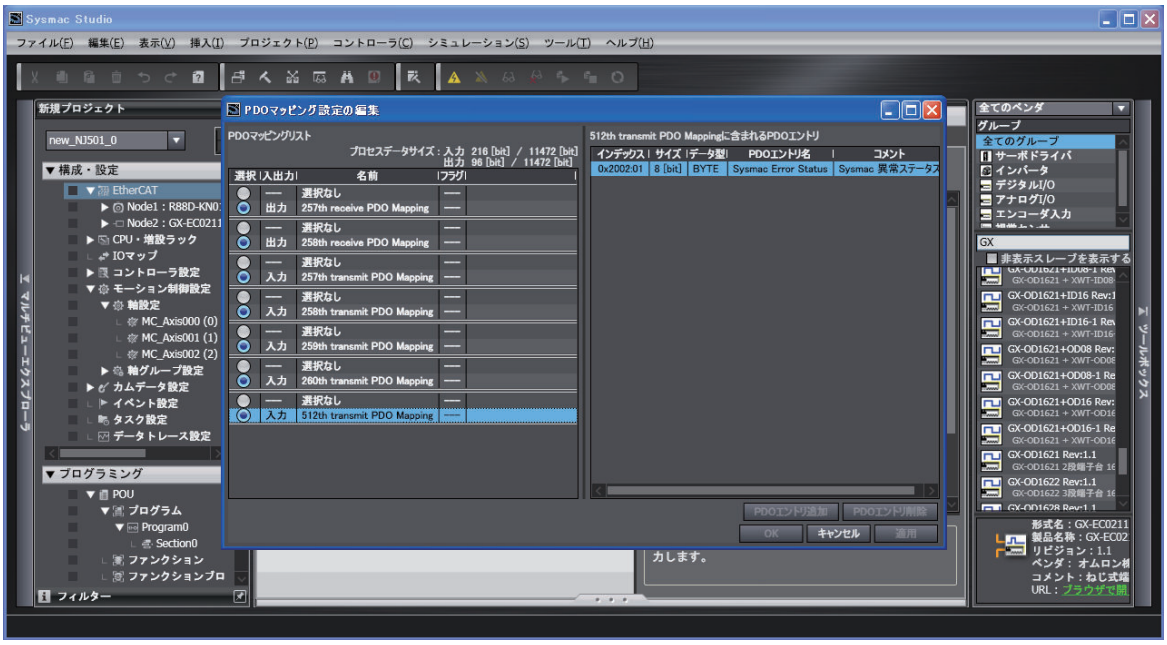
入力信号と配線方法は、『形 GX シリーズ EtherCAT スレーブ ユーザーズマニュアル (SBCD-350)』を参照してください。

PDO マッピング

MC 機能モジュールのエンコーダ軸として制御するための PDO マッピングについて説明します。

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。

使用する PDO マップは、Sysmac Studio の［EtherCAT］構成編集画面の［PDO マッピング設定の編集］で選択します。



Sysmac Studio のデフォルトでは、下記の PDO マップが選択されています。

RxPDO (1700Hex)	チャネル 1 ソフトスイッチ(4020Hex-01Hex)、チャネル 2 ソフトスイッチ(4020Hex-02Hex)
RxPDO (1701Hex)	チャネル 1 プリセット現在値(4011Hex-01Hex)、チャネル 2 プリセット現在値(4011Hex-02Hex)
TxPDO (1B00Hex)	チャネル 1 カウンタ現在値(4010Hex-01Hex)、チャネル 2 カウンタ現在値(4010Hex-02Hex)
TxPDO (1B01Hex)	チャネル 1 ラッチ値 A(4012Hex-01Hex)、チャネル 2 ラッチ値 A(4012Hex-02Hex)
TxPDO (1B02Hex)	チャネル 1 ラッチ値 B(4013Hex-01Hex)、チャネル 2 ラッチ値 B(4013Hex-02Hex)
TxPDO (1B03Hex)	チャネル 1 ステータスビット(4030Hex-01Hex)、チャネル 2 ステータスビット(4030Hex-02Hex)
TxPDO (1BFFHex)	Sysmac 異常ステータス(2002Hex-01Hex)

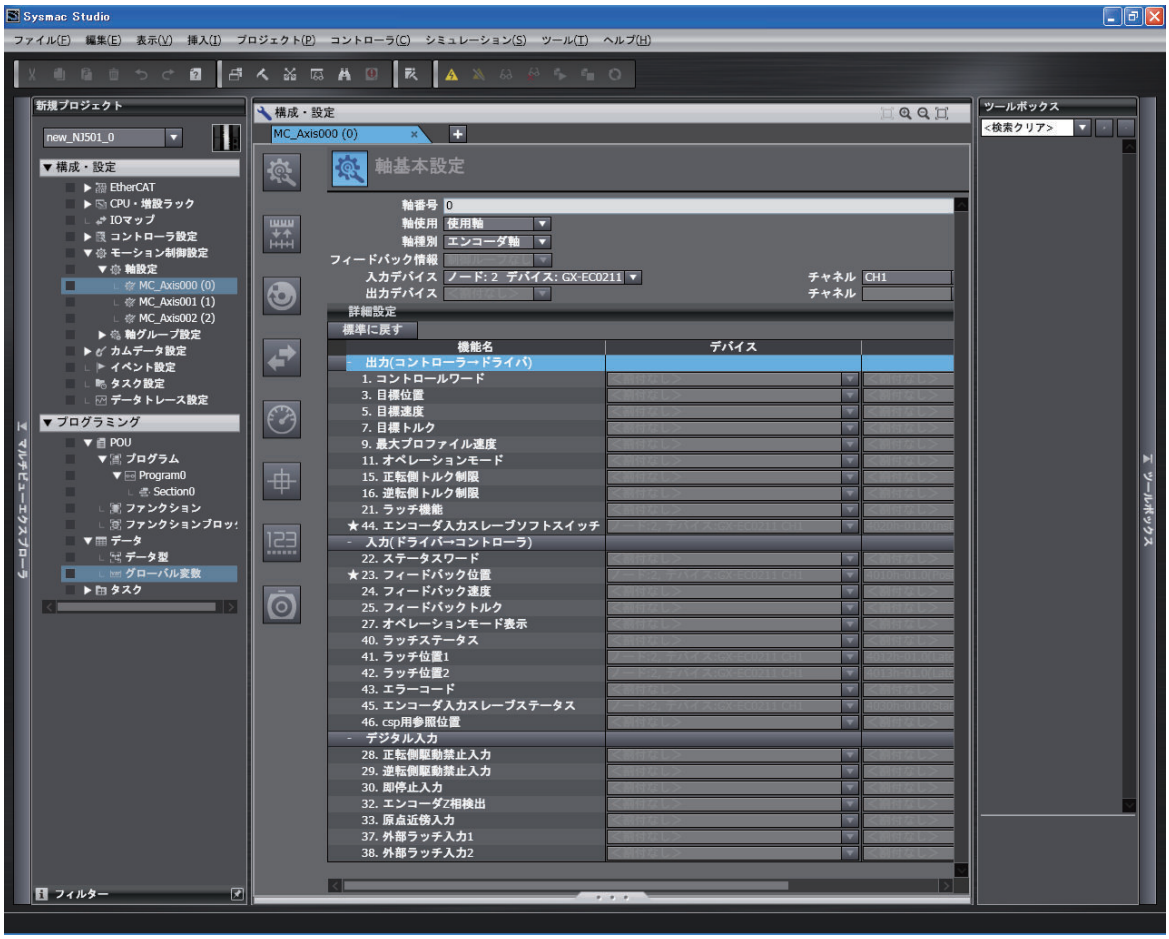
参考

推奨エンコーダ入力ターミナルの形 GX-EC0211/EC0241（ユニットバージョン 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。

MC 機能モジュールの機能とプロセスデータの関係

MC 機能モジュールが持つ機能は、プロセスデータのオブジェクトの情報と関係があります。
MC 機能モジュールが使用する EtherCAT スレーブの構成や使用する機能によっては、これらの関係の変更が必要な場合があります。

Sysmac Studio の［軸基本設定］画面の［詳細設定］をクリックすると設定画面が開きます。



参考

推奨エンコーダ入力ターミナルの形 GX-EC0211/EC0241（ユニットバージョン Ver.1.1 以降）の場合、機能とプロセスデータとの関係は、Sysmac Studio のデフォルトのままでも使用することができます。

● 出力（コントローラ→ドライバ）の設定

MC 機能モジュールからエンコーダ入力ターミナルへの指令データの設定です。
下記は Sysmac Studio のデフォルト設定です。（O印は必須のオブジェクト）

機能名	プロセスデータ		説明
	チャネル 1	チャネル 2	
○ エンコーダ入力スレーブソフトスイッチ	4020Hex-01.0 (Instruction Bits)	4020Hex-02.0 (Instruction Bits)	ソフトスイッチの内容を設定します。 使用するチャネルにより、左記のオブジェクトを設定してください。

使用上の注意

設定を変更する場合は、MC 機能モジュールの機能とプロセスデータの組み合わせが、意図したとおりに動作するか確認してください。

● 入力（ドライバ→コントローラ）の設定

エンコーダ入力ターミナルから MC 機能モジュールへのステータスデータです。
下記は Sysmac Studio のデフォルト設定です。（○印は必須のオブジェクト）

機能名	プロセスデータ		説明
	チャネル 1	チャネル 2	
○ フィードバック位置	4010Hex-01.0 (Position Value)	4010Hex-02.0 (Position Value)	カウンタの現在値が格納されています。 使用するチャネルにより、左記のオブジェクトを設定してください。
ラッチ位置 1	4012Hex-01.0 (Latch Value A)	4012Hex-02.0 (Latch Value A)	ラッチ機能 1 でラッチした位置です。ラッチ A の値が格納されています。 MC_TouchProbe（外部ラッチ有効）命令など、ラッチ機能を使用する場合に必要です。 使用するチャネルにより、左記のオブジェクトを設定してください。
ラッチ位置 2	4013Hex-01.0 (Latch Value B)	4013Hex-02.0 (Latch Value B)	ラッチ機能 2 でラッチした位置です。ラッチ B の値が格納されています。 MC_TouchProbe（外部ラッチ有効）命令など、ラッチ機能を使用する場合に必要です。 使用するチャネルにより、左記のオブジェクトを設定してください。
エンコーダ入力スレーブステータス	4030Hex-01.0 (Status Bits)	4030Hex-02.0 (Status Bits)	ステータスビットが格納されています。 MC_TouchProbe（外部ラッチ有効）命令など、ラッチ機能を使用する場合に必要です。 使用するチャネルにより、左記のオブジェクトを設定してください。

使用上の注意

設定を変更する場合は、MC 機能モジュールの機能とプロセスデータの組み合わせが、意図したとおりに動作するか確認してください。

● デジタル入力の設定

エンコーダ軸として使用する場合、設定は不要です。

エンコーダ入力ターミナル側のオブジェクトの設定

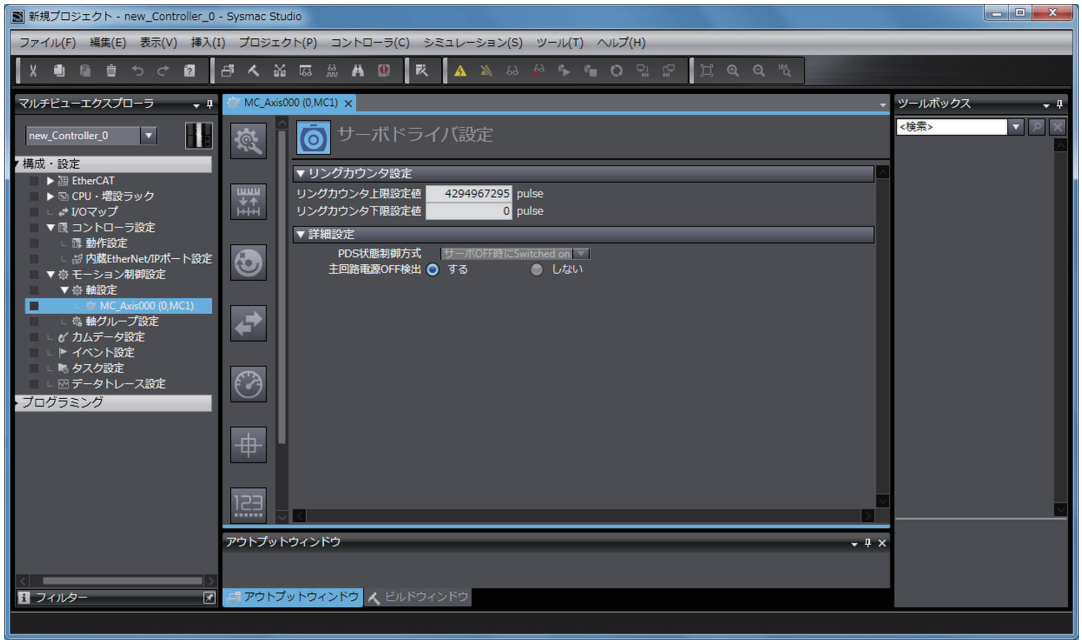
エンコーダ入力ターミナルでは、設定が必要なオブジェクトはありません。

MC 機能モジュールとエンコーダ入力ターミナルのリングカウンタの関係

MC 機能モジュールの軸パラメータのサーボドライバ設定（リングカウンタ上限値、リングカウンタ下限値）と、エンコーダ入力ターミナルのリングカウンタ最大値設定は、一致させておく必要があります。

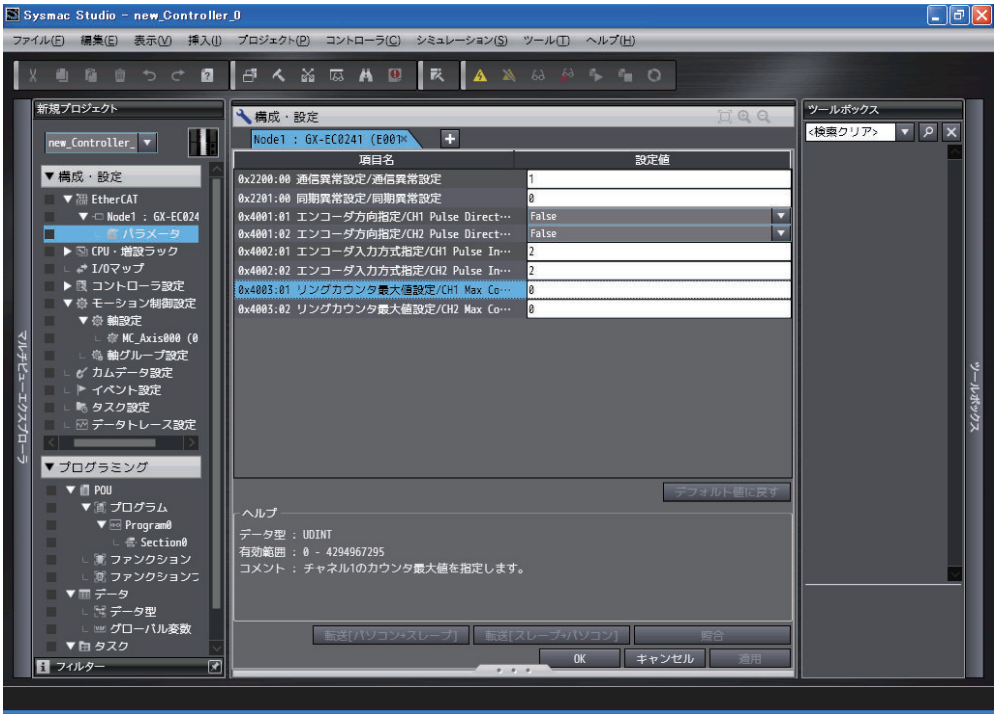
サーボドライバ設定（リングカウンタ上限値、リングカウンタ下限値）は、Sysmac Studio の［サーボドライバ設定］画面で編集します。
設定内容は、以下のとおりです。

パラメータ	意味	設定値
リングカウンタ上限設定値	サーボドライバ、またはエンコーダ入力ターミナルで設定されているリングカウンタ上限値を設定します。	エンコーダ入力ターミナルのリングカウンタ最大値設定と一致させてください。
リングカウンタ下限設定値	サーボドライバ、またはエンコーダ入力ターミナルで設定されているリングカウンタ下限値を設定します。	「0」を設定してください。



エンコーダ入力ターミナルのリングカウンタ最大値設定は、Sysmac Studio の［EtherCAT］編集画面で設定します。
設定内容は、以下のとおりです。

インデックス	オブジェクト名	設定値
0x4003	リングカウンタ最大値設定（Max Count Setting）	MC 機能モジュールの軸パラメータのサーボドライバ設定（リングカウンタ上限設定値）と一致させてください。


Sysmac Studio - new_Controller_0
ファイル(F) 編集(E) 表示(V) 挿入(I) プロジェクト(P) コントローラ(C) シミュレーション(S) ツール(T) ヘルプ(H)
新規プロジェクト
new_Controller_
▼ 構成・設定
EtherCAT
Node1 : GX-EC024
パラメータ
CPU・増設ラック
I/Oマップ
コントローラ設定
モーション制御設定
軸設定
MC_Axis000 (0
軸グループ設定
カムデータ設定
イベント設定
タスク設定
データトレース設定
▼ プログラミング
POU
プログラム
Program0
Section0
ファンクション
ファンクション
データ
データ型
グローバル変数
タスク
フィルター
構成・設定
Node1 : GX-EC0241 (E001
項目名
設定値
0x2200:00 通信異常設定/通信異常設定
0x2201:00 同期異常設定/同期異常設定
0x4001:01 エンコーダ方向指定/CH1 Pulse Direct…
0x4001:02 エンコーダ方向指定/CH2 Pulse Direct…
0x4002:01 エンコーダ入力方式指定/CH1 Pulse In…
0x4002:02 エンコーダ入力方式指定/CH2 Pulse In…
0x4003:01 リングカウンタ最大値設定/CH1 Max Co…
0x4003:02 リングカウンタ最大値設定/CH2 Max Co…
False
False
デフォルト値に戻す
ヘルプ
データ型 : UDINT
有効範囲 : 0 - 4294967295
コメント : チャネル1のカウンタ最大値を指定します。
転送[パソコン→スレーブ]
転送[スレーブ→パソコン]
照合
OK
キャンセル
適用
ツールボックス
<検索クリア>

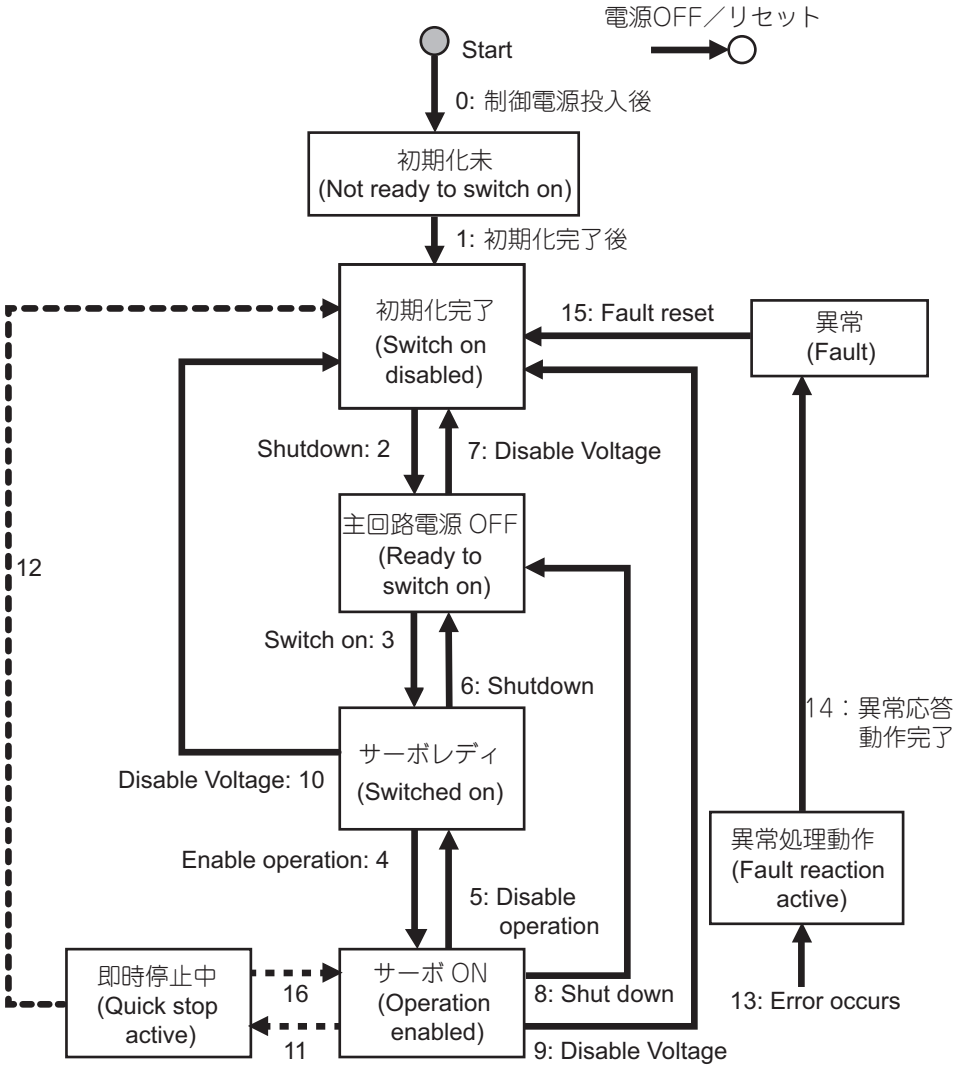
A-4　NX ユニットとの接続

NX シリーズ 位置インタフェースユニットとの接続については、『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-5 PDS 状態遷移

PDS 状態遷移は CiA402 ドライブプロファイルで定義されています。PDS 状態遷移の指令は、プロセスデータのコントロールワード(6040Hex: Controlword)を使用します。実際の PDS 状態は、プロセスデータのステータスワード(6041Hex: Statusword)で確認することができます。

CiA402 ドライブプロファイルで定義された状態遷移図を次に示します。
長方形内はステートを表し、数字は状態制御コマンドを表しています。



A-5-1 PDS 状態制御方式

ここでは、軸パラメータ［PDS 状態制御方式］の設定値と、PDS 状態の関係を説明します。

●［PDS 状態制御方式］を［0］に設定した場合

［PDS 状態制御方式］を［0: サーボ OFF 時に Switeched on］に設定すると、以下のような動作となります。

- サーボドライバとの通信確立後、MC 機能モジュールは、PDS 状態をサーボレディ(Switched on)状態まで自動的に遷移させます。

- サーボレディ(Switched on)状態で、MC_Power（運転可）命令の「Enable」（有効）を TRUE にすると、サーボ ON(Operation enabled)に遷移します。状態遷移図の［4: Enable operation］の遷移となります。
- サーボ ON(Operation enabled)状態で、MC_Power（運転可）命令の「Enable」（有効）を FALSE にすると、サーボレディ(Switched on)状態に遷移します。状態遷移図の［5: Disable operation］の遷移となります。

●［PDS 状態制御方式］を［1］に設定した場合

［PDS 状態制御方式］を［1: サーボ OFF 時に Ready to switch on］に設定すると、以下のような動作となります。

- サーボドライバとの通信確立後、MC 機能モジュールは、PDS 状態を主回路電源 OFF(Ready to switch on)状態まで自動的に遷移させます。
- 主回路電源 OFF(Ready to switch on)状態で、MC_Power（運転可）命令の「Enable」（有効）を TRUE にすると、サーボ ON(Operation enabled)に遷移します。状態遷移図の［3: Switch on］と［4: Enable operation］の遷移を同時に実施します。
- サーボ ON(Operation enabled)状態で、MC_Power（運転可）命令の「Enable」（有効）を FALSE にすると、主回路電源 OFF(Ready to switch on)状態に遷移します。状態遷移図の［5: Disable operation］と［6: Shutdown］の遷移を同時に実施します。

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 ユニットは、［PDS 状態制御方式］が選択可能です。［PDS 状態制御方式］の初期値は、「0」です。
- ユニットバージョン Ver.1.09 以前の CPU ユニットは、［PDS 状態制御方式］が「0」です。

A-5-2　主回路電源 OFF 検出

サーボロック状態で主回路電源 OFF を検出するかどうかを選択することができます。
MC_Power（運転可）命令を実行しても「サーボ主回路電源オフ検出」異常となる場合は、［検出しない］に設定してください。

使用上の注意

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、［検出しない］を選択できません。
［検出しない］を選択し、サーボロック状態でサーボ主回路電源を OFF すると、「サーボ主回路電源オフ検出」異常となります。

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 ユニットは、［主回路電源 OFF 検出］が選択可能です。［主回路電源 OFF 検出］の初期値は、［検出する］です。
- ユニットバージョン Ver.1.09 以前の CPU ユニットは、［主回路電源 OFF 検出］が［検出する］です。

A-6 用語説明

モーション制御に関連する用語を説明します。

A-6-1 NJ/NX シリーズ共通

用語	説明
メインメモリ (main memory)	CPU ユニットが OS、ユーザプログラム実行に使用する CPU ユニット内部のメモリ
定周期タスク (periodic task)	周期ごとに、I/O リフレッシュとユーザプログラム実行をするタスク
プライマリ定周期タスク (primary periodic task)	最も優先度が高いタスク
タスク周期 (task period)	プライマリ定周期タスクおよび定周期タスクの実行間隔
I/O リフレッシュ (I/O Refresh)	あらかじめ定められたメモリエリアの、外部とのサイクリックなデータ交換のこと。
プログラム (program)	ファンクション、ファンクションブロックと並ぶ、3 種類の POU のひとつ。 タスクに割り付けられて動作する。
ユーザプログラム (user program)	複数のプログラムの集合体
インライン ST (Inline ST)	ラダー図プログラムの中に、その一部として ST プログラムを書くこと、または書かれた ST プログラムのこと。
システム定義変数 (system-defined variables)	すべての変数属性がシステムによって定義されており、ユーザはそれを一切変更することができない変数のこと。
グローバル変数 (global variable)	すべての POU（プログラム、ファンクションブロック、ファンクション）から読み書き可能な変数のこと。
ローカル変数 (local variable)	定義された POU 内だけからアクセス可能な変数。 ローカル変数には、内部変数、入力変数、出力変数、入出力変数、外部変数が含まれる。
ダウンロード (download)	Sysmac Studio の「同期」機能により、Sysmac Studio からコントローラにデータを転送すること。
アップロード (upload)	Sysmac Studio の「同期」機能により、コントローラからデータを Sysmac Studio に転送すること。
全停止フォールトレベルのコントローラ異常 (major fault level Controller error)	NJ/NX シリーズ コントローラ全体の制御が不能となる異常のこと。 CPU ユニットは、ユーザプログラムの実行をすぐに停止し、リモート I/O を含むすべてのスレーブ／ユニットに対して負荷を遮断する。
部分停止フォールトレベルのコントローラ異常 (partial fault level Controller error)	NJ/NX シリーズ コントローラの、ある機能モジュール全体の制御が不能となる異常のこと。 部分停止フォールトの発生後も NJ/NX シリーズ CPU ユニットは運転を継続する。
軽度フォールトレベルのコントローラ異常 (minor fault level Controller error)	NJ/NX シリーズ コントローラの、ある機能モジュールの一部の制御が不能となる異常のこと。 軽度フォールトの発生後も NJ/NX シリーズ CPU ユニットは運転を継続する。
監視情報 (observation)	コントローラ情報およびユーザ情報のイベント重要度を示すランクのひとつ。 制御に影響がないレベルの軽微な異常であるが、イベントログに記録してユーザに通知する情報。

A-6-2 モーション制御機能

用語	説明
使用実軸	軸種別が［サーボ軸］または［エンコーダ軸］で、かつ軸使用が［使用軸］に設定された軸

用語	説明
使用仮想軸	軸種別が［仮想サーボ軸］または［仮想エンコーダ軸］で、かつ軸使用が［使用軸］に設定された軸
モーション制御機能モジュール (Motion Control Function Module)	モーション制御を実行するソフトウェア構成のひとつ。 ユーザプログラム内のモーション制御命令から与えられた指令に従って、モーション制御を実行する。(略語: MC 機能モジュール)
モーション制御命令 (motion control instruction)	モーション制御機能を実行するためのファンクションブロック定義の命令。 モーション制御機能モジュールでは、PLCopen®のモーション制御用ファンクションブロックに準拠した命令とモーション制御機能モジュール独自の命令を持つ。
モーション制御軸 (motion control axes)	モーション制御機能のすべての機能が使用可能な軸
単軸位置制御軸 (single-axis position control axes)	モーション制御機能のうち、「単軸位置制御」だけが使用可能な軸
単軸位置制御 (single-axis position control)	1 軸での位置決め制御
単軸速度制御 (single-axis velocity control)	1 軸での速度制御。 MC 機能モジュールでは、サーボドライバに対して速度指令値を与える場合と位置指令値を与える場合がある。
単軸トルク制御 (single-axis torque control)	1 軸でのトルク制御
単軸同期制御 (single-axis synchronized control)	ひとつの主軸に対して 1 軸の従軸が同期する制御。 ギア比で同期するギア動作と、カムテーブルの位相と変位の関係に同期するカム動作がある。
単軸手動制御 (single-axis manual operation)	ジョグ送りなど手動操作のための制御
単軸制御補助 (auxiliary functions for single-axis control)	オーバーライド値設定や異常のリセットなど 1 軸での制御を補助する機能
多軸協調制御 (multi-axes coordinated control)	直線補間や円弧補間など複数軸が関連して動作する制御。 軸グループにより協調する軸を指定する。
多軸協調制御補助 (auxiliary functions for multi-axes coordinated control)	軸グループに対するオーバーライド値設定や異常のリセットなど複数軸での制御を補助する機能
モーション制御パラメータ (motion control parameters)	MC 機能モジュールの各種動作を指定するためのパラメータ。 MC 共通パラメータ、軸パラメータ、軸グループパラメータの総称。
軸パラメータ (axis parameters)	軸単位のパラメータ
軸グループパラメータ (axes group parameters)	軸グループ単位のパラメータ
モーション制御システム変数 (system-defined variables for motion control)	MC 機能モジュールの各種ステータス情報などを表す変数。 MC 共通変数、軸変数、軸グループ変数の総称。
MC 共通変数 (MC common variable)	MC 機能モジュール共通の各種ステータス情報などを構造体で定義したシステム定義変数
軸変数 (axis variable)	軸ごとの各種ステータス情報や一部の軸パラメータ設定情報などを構造体で定義したシステム定義変数
軸グループ変数 (axes group variable)	軸グループごとの各種ステータス情報や一部の軸グループパラメータ設定情報などを構造体で定義したシステム定義変数
原点復帰 (homing)	機械原点を決定すること。 原点決め、原点サーチ、キャリブレーション、Datum とも呼ばれる。
機械原点 (home)	装置ごとに定められた機械の原点。 原点復帰動作時に原点入力信号で決定する原点。
原点 (zero position)	機械原点をもとにユーザプログラムでゼロ位置として扱う原点。 原点位置変更を行わなければ機械原点と同じ位置である。

用語	説明
位置偏差 (following error)	指令現在位置とフィードバック現在位置の差。 位置制御モード以外では指令現在位置がないため位置偏差も存在しない。 偏差カウンタ値、たまりパルスなどとも呼ばれる。
偏差カウンタリセット (following error reset)	位置偏差をゼロにすること。
カムプロファイル曲線 (cam profile curve)	カム動作を実現するための位相と変位の関係を表す曲線。 Sysmac Studio のカムエディタで作成した曲線のこと。
カムデータ (cam data)	カム動作を実現するための位相（主軸）と変位（従軸）から構成されるデータ
カムデータ変数 (cam data variable)	カムデータを構造体配列化した変数のこと。 位相と変位データからなる、構造体型の配列指定の変数である。
カムテーブル (cam table)	カムデータのデータテーブル。 カムエディタで作成したカムプロファイル曲線を CPU ユニットに転送すると、カムテーブルとして CPU ユニット内蔵の不揮発性メモリに保存される。
オーバーライド (override)	作業中に作業者がプログラムされた値を一時的に変更することができる機能
ジャーク (jerk)	加速度／減速度の変化の比率。 ジャークの指定により加減速時の速度波形が S 字型となる。 躍動や躍度あるいは加加速度とも呼ばれる。
マルチモーション (Multi-motion)	プライマリ定周期タスクと定周期タスク（実行優先度 5）の両タスクを使用しモーション制御を並列実行すること。

A-6-3　EtherCAT 通信

用語	説明
CAN application protocol over EtherCAT (CoE)	EtherCAT 上での CAN アプリケーション プロトコル サービス
CAN in Automation (CiA)	上位層プロトコルの開発とサポートを行うユーザおよびメーカによる国際団体
EtherCAT 協会 (ETG) (EtherCAT Technology Group)	EtherCAT をオープン化し普及促進をはかる団体
オブジェクト (Object)	デバイス内の特定の構成体の抽象的表現で、データ、パラメータ、手法の集合体
オブジェクトディクショナリ Object Dictionary)	データ型オブジェクト、通信オブジェクト、アプリケーションオブジェクトの記述が入ったデータ構造
プロセスデータ (Process Data)	測定および制御の目的で周期的あるいは非周期的に伝送指定されるアプリケーションオブジェクトの集合体
プロセスデータオブジェクト (Process Data Object)	1 つ以上のプロセスデータエンティティを持つパラメータをマッピングすることによって記述される構造
サービスデータオブジェクト (Service Data Object)	CoE の非同期メールボックス通信で、すべてのオブジェクトディクショナリを読み書き可能
受信 PDO (Receive PDO)	EtherCAT スレーブで受信するプロセスデータオブジェクト
送信 PDO (Transmit PDO)	EtherCAT スレーブから送信するプロセスデータオブジェクト
デバイスプロファイル (Device Profile)	デバイスクラスのアプリケーション用のパラメータや機能の記述の集合体

A-7　バージョン関連情報

ユニットバージョンごとのサポート機能について説明します。

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。

モーション制御命令のバージョン関連情報の詳細は、『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)』を参照してください。

ユニットバージョン Ver.1.01 で追加されたモーション制御機能

ユニットバージョン Ver.1.01 で追加された機能を使用するときは、Sysmac Studio Ver.1.02 以降を使用してください。

機能	概要
軸グループ構成軸書込	MC_ChangeAxesInGroup（軸グループ構成軸書込）命令により、軸グループパラメータの［構成軸］を一時的に書き換える機能。
軸グループ位置取得	MC_GroupReadPosition（軸グループ位置取得）命令により、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する機能。
軸グループサイクリック同期絶対位置制御	MC_GroupSyncMoveAboslute（軸グループサイクリック同期絶対位置制御）命令により、各軸に指定した目標位置を周期的に出力する機能。
制御対象サーボドライバ	オムロン製サーボモータ／ドライバ G5 シリーズ EtherCAT 通信内蔵リニアモータタイプを追加。

ユニットバージョン Ver.1.02 で追加されたモーション制御機能

ユニットバージョン Ver.1.02 で追加された機能を使用するときは、Sysmac Studio Ver.1.03 以降を使用してください。

ユニットバージョン Ver.1.02 で追加されたモーション制御機能はありませんが、一部の命令の仕様を改善しました。モーション制御命令のバージョン関連情報の詳細は、『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)』を参照してください。

ユニットバージョン Ver.1.03 で追加されたモーション制御機能

ユニットバージョン Ver.1.03 で追加された機能を使用するときは、Sysmac Studio Ver.1.04 以降を使用してください。

機能	概要
サイクリック同期絶対位置決め	MC_SyncMoveAbsolute（サイクリック同期絶対位置決め）命令により、位置制御モードで制御周期ごとに指令位置を出力する機能。
パラメータ指定原点復帰	MC_HomeWithParameter（パラメータ指定原点復帰）命令により、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る機能。

ユニットバージョン Ver.1.04 で追加されたモーション制御機能

ユニットバージョン Ver.1.04 で追加された機能を使用するときは、Sysmac Studio Ver.1.05 以降を使用してください。

機能	概要
軸使用変更	MC_ChangeAxisUse（軸使用変更）命令により、軸パラメータの［軸使用］を一時的に切り替える機能。

ユニットバージョン Ver.1.05 で追加されたモーション制御機能

ユニットバージョン Ver.1.05 で追加された機能を使用するときは、Sysmac Studio Ver.1.06 以降を使用してください。

機能	概要
起動速度	軸動作が開始するときの初速度を設定する機能。
入力信号の論理反転	即停止入力信号、正方向限界入力信号、負方向限界入力信号、原点近傍入力信号の論理を反転する機能。

（注）ユニットバージョン Ver.1.05 以降の CPU ユニットで、NX シリーズ 位置インタフェースユニットを軸に割り付けてモーション制御を行うことができます。

ユニットバージョン Ver.1.06 で追加、変更されたモーション制御仕様

ユニットバージョン Ver.1.06 で追加、変更された性能仕様、機能仕様を使用するときは、Sysmac Studio Ver.1.07 以降を使用してください。

機能	概要
制御軸最大数	NJ301-□□□□の制御軸最大数を 15 軸に拡張。 使用実軸最大数は変更なし。
単軸制御最大数	NJ301-□□□□の単軸制御最大数を 15 軸に拡張。 使用実軸最大数は変更なし。
デジタルカムスイッチ有効	MC_DigitalCamSwitch（デジタルカムスイッチ有効）命令により、軸の位置に応じてデジタル出力を ON または OFF する機能。
タイムスタンプ→軸位置算出	MC_TimeStampToPos（タイムスタンプ→軸位置算出）命令により、指定したタイムスタンプでの軸の位置を算出する機能。
カム動作開始にブレンディングを追加	MC_CamIn（カム動作開始）命令のバッファモード選択にブレンディングを追加。
_sMC_POSITION_REF 型	任意座標系の軌跡を 3D モーションモニタ表示するためのデータ型。

ユニットバージョン Ver.1.08 で追加、変更されたモーション制御機能

ユニットバージョン Ver.1.08 で追加、変更された機能を使用するときは、Sysmac Studio Ver.1.09 以降を使用してください。

機能	概要
カムテーブル生成	MC_GenerateCamTable（カムテーブル生成）命令により、入力パラメータで指定したカムプロパティとカムノードからカムテーブルを生成する機能。
軸パラメータの変更	MC_WriteAxisParameter（軸パラメータ書込）命令および MC_ReadAxisParameter（軸パラメータ読出）命令により、軸パラメータをユーザプログラムから参照・変更する機能。
デバイス変数の割り付け	軸変数に割り付けたスレーブやユニットの I/O ポートにデバイス変数を割り付け可能。*1 詳細は、「2-4-2 EtherCAT マスタ機能とモーション制御機能モジュールの関係 (P.2-19)」を参照してください。

*1. CPU ユニットのユニットバージョンには依存しません。Sysmac Studio Ver.1.09 以降を使用することで割り付け可能となります。

ユニットバージョン Ver.1.09 で追加、変更されたモーション制御機能

ユニットバージョン Ver.1.09 で追加された機能を使用するときは、Sysmac Studio Ver.1.10 以降を使用してください。

ユニットバージョン Ver.1.09 で追加されたモーション制御機能はありませんが、一部の命令の仕様を改善しました。モーション制御命令のバージョン関連情報の詳細は、『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)』を参照してください。

ユニットバージョン Ver.1.10 で追加、変更されたモーション制御機能

ユニットバージョン Ver.1.10 で追加された以下の機能を使用するときは、Sysmac Studio Ver.1.12 以降を使用してください。

機能	概要
従軸位置補正	同期制御中の従軸に対して位置補正する機能。
ブレンディングの動作変更	［加減速オーバー］設定が［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)］、［異常として停止する］の場合でも、最大加減速度で加減速し、ブレンディングを継続します。
絶対値エンコーダの原点確定タイミング	従来の原点確定に加えて、EtherCAT スレーブ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、原点確定します。
プロセスデータ通信が未確立の場合の現在位置	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。
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	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)のマッピング有無の組み合わせにより、動作が異なります。
PDS 状態制御方式の追加	MC_Power 命令でサーボ OFF する時の PDS 状態の遷移先を設定します。
主回路電源 OFF 検出の追加	サーボロック指令時、またはサーボロック状態で主回路電源 OFF を検出するかどうかを選択することができます。

ユニットバージョン Ver.1.10 で追加された以下の機能を使用するときは、Sysmac Studio Ver.1.13 以降を使用してください。

機能	概要
マルチモーションでの同期制御	マルチモーションで異なるタスクに割り付けている軸間の同期制御命令を実行する機能。

ユニットバージョン Ver.1.11 で追加、変更されたモーション制御機能

ユニットバージョン Ver.1.11 で追加された以下の機能を使用するときは、Sysmac Studio Ver.1.15 以降を使用してください。

機能	概要
単位変換設定のパラメータ追加	［減速機使用］、［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］のパラメータを追加。

ユニットバージョン Ver.1.11 で追加された以下の機能を使用するときは、Sysmac Studio Ver.1.16 以降を使用してください。

機能	概要
制御対象サーボドライバ	オムロン製サーボドライバ 1S シリーズ EtherCAT 通信内蔵タイプを追加。

ユニットバージョン Ver.1.13 で追加、変更されたモーション制御機能

ユニットバージョン Ver.1.13 で追加された以下の機能を使用するときは、Sysmac Studio Ver.1.17 以降を使用してください。

機能	概要
制御機能	制御する軸の機能を選択します。
ダウンロード中の I/O デバイスへの指令	ダウンロード開始時に、I/O デバイスへ指令を停止するか、継続するかを選択することにより、MC 機能モジュールの動作が変わります。

ユニットバージョン Ver.1.21/Ver.1.32 で追加されたモーション制御機能

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 以降を使用してください。

CPU ユニット	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

機能	概要
カムモニタ	MC_CamMonitor（カムモニタ）命令により、現在の主軸の位置に基づいたカムテーブルの位相と変位、および現在の従軸の位置と変位の偏差をモニタする機能。
従軸位置補正	MC_OffsetPosition（位置オフセット補正）命令により、同期制御中の従軸に対して、指定した位置オフセットを加減速カーブをつけて加算し、指令位置を補正する機能。
制御対象サーボドライバ	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 安全機能対応を追加。

ユニットバージョン Ver.1.36/Ver.1.69 で追加されたモーション制御機能

以下に示すユニットバージョンの CPU ユニットと Sysmac Studio Ver.1.63 以降を使用してください。

CPU ユニット	ユニットバージョン
形 NX102 CPU ユニット	Ver.1.69 以降
形 NX1P2 CPU ユニット	Ver.1.69 以降
形 NX502 CPU ユニット	Ver.1.69 以降
NJ シリーズ CPU ユニット	Ver.1.69 以降
形 NX701 CPU ユニット	Ver.1.36 以降

機能	概要
カムテーブル終点インデックス更新	MC_SetCamTableProperty（カムテーブルプロパティ更新）命令の代替命令である MC_SetCamTableEndPointIndex（カムテーブル終点インデックス更新）命令を提供します。 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。

A-8　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-433	SBCE-363

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
A	2011 年 7 月	初版
B	2012 年 1 月	• 形 NJ301-□□□□の追加 • CPU ユニット Ver.1.01 へのバージョンアップに伴う改訂
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-□□□□、および NJ シリーズ CPU ユニット形 NJ101-10□□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 1 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 • イベントコードおよび異常に関する説明の『NJ/NX シリーズ トラブルシューティングマニュアル』への移動に伴う改訂 • 誤記訂正

I

索引

索引

E

EtherCAT.....1-2
EtherCAT マスタ機能モジュール.....2-2

M

_MC_AX[*]（軸変数）.....6-22
_MC_AX[*].Act.Pos（フィードバック現在位置）.....6-25
_MC_AX[*].Act.Timestamp（タイムスタンプ）.....6-26
_MC_AX[*].Act.Trq（フィードバック現在トルク）....6-25
_MC_AX[*].Act.Vel（フィードバック現在速度）.....6-25
_MC_AX[*].Cfg.AxEnable（軸使用）.....6-26
_MC_AX[*].Cfg.AxNo（軸番号）.....6-26
_MC_AX[*].Cfg.AxType（軸種別）.....6-26
_MC_AX[*].Cfg.ExecID（実行 ID）.....6-26
_MC_AX[*].Cfg.NodeAddress（ノードアドレス）... 6-26
_MC_AX[*].Cmd.AccDec（指令現在加減速度）.....6-25
_MC_AX[*].Cmd.Jerk（指令現在ジャーク）.....6-25
_MC_AX[*].Cmd.Pos（指令現在位置）.....6-25
_MC_AX[*].Cmd.Trq（指令現在トルク）.....6-25
_MC_AX[*].Cmd.Vel（指令現在速度）.....6-25
_MC_AX[*].Details.Homed（原点確定）.....6-24
_MC_AX[*].Details.Idle（停止中）.....6-24
_MC_AX[*].Details.InHome（原点停止）.....6-24
_MC_AX[*].Details.InPosWaiting（インポジション待ち）.....6-24
_MC_AX[*].Details.VelLimit（指令速度飽和）.....6-24
_MC_AX[*].Dir.Nega（負方向指令中）.....6-24
_MC_AX[*].Dir.Posi（正方向指令中）.....6-24
_MC_AX[*].DrvStatus.CSP（サイクリック同期位置(CSP)モード中）.....6-25
_MC_AX[*].DrvStatus.CST（サイクリック同期トルク(CST)モード中）.....6-25
_MC_AX[*].DrvStatus.CSV（サイクリック同期速度(CSV)モード中）.....6-25
_MC_AX[*].DrvStatus.DrvAlarm（ドライバエラー入力）.....6-24
_MC_AX[*].DrvStatus.DrvWarning（ドライバワーニング入力）.....6-24
_MC_AX[*].DrvStatus.Home（原点入力）.....6-24
_MC_AX[*].DrvStatus.HomeSw（原点近傍入力）.... 6-24
_MC_AX[*].DrvStatus.ILA（ドライバ内部機能制限中）.....6-25
_MC_AX[*].DrvStatus.ImdStop（即停止入力）.....6-24
_MC_AX[*].DrvStatus.Latch1（ラッチ入力 1）.....6-24
_MC_AX[*].DrvStatus.Latch2（ラッチ入力 2）.....6-24
_MC_AX[*].DrvStatus.MainPower（主回路電源）... 6-24
_MC_AX[*].DrvStatus.N_OT（負方向限界入力）..... 6-24
_MC_AX[*].DrvStatus.P_OT（正方向限界入力）..... 6-24
_MC_AX[*].DrvStatus.Ready（サーボレディ）.....6-24
_MC_AX[*].DrvStatus.ServoOn（サーボ ON）.....6-24
_MC_AX[*].MFaultLvl.Active（軸 軽度フォールト発生中）.....6-26
_MC_AX[*].MFaultLvl.Code（軸 軽度フォールトコード）.....6-26
_MC_AX[*].Obsr.Active（軸 監視情報発生中）.....6-26
_MC_AX[*].Obsr.Code（軸 監視情報コード）.....6-26
_MC_AX[*].Scale.CountMode（カウントモード）... 6-26
_MC_AX[*].Scale.Den（モータ 1 回転の移動量）..... 6-26
_MC_AX[*].Scale.MaxPos（現在位置上限値）..... 6-27
_MC_AX[*].Scale.MinPos（現在位置下限値）..... 6-27
_MC_AX[*].Scale.Num（モータ 1 回転のパルス数）. 6-26
_MC_AX[*].Scale.Units（表示単位）..... 6-26
_MC_AX[*].Status.Continuous（連続動作中）..... 6-23
_MC_AX[*].Status.Coordinated（多軸協調動作中） 6-23
_MC_AX[*].Status.Disabled（軸無効）..... 6-23
_MC_AX[*].Status.Discrete（位置決め動作中）..... 6-23
_MC_AX[*].Status.ErrorStop（エラー減速停止中）. 6-23
_MC_AX[*].Status.Homing（原点復帰中）..... 6-23
_MC_AX[*].Status.Ready（軸起動準備完了）..... 6-23
_MC_AX[*].Status.Standstill（停止中）..... 6-23
_MC_AX[*].Status.Stopping（減速停止中）..... 6-23
_MC_AX[*].Status.Synchronized（同期動作中）..... 6-23
_MC_COM（MC 共通変数）..... 6-22
_MC_COM.MFaultLvl.Active（MC 共通 軽度フォールト発生中）..... 6-22
_MC_COM.MFaultLvl.Code（MC 共通 軽度フォールトコード）..... 6-22
_MC_COM.Obsr.Active（MC 共通 監視情報発生中） 6-22
_MC_COM.Obsr.Code（MC 共通 監視情報コード）. 6-22
_MC_COM.PFaultLvl.Active（MC 共通 部分停止フォールト発生中）..... 6-22
_MC_COM.PFaultLvl.Code（MC 共通 部分停止フォールトコード）..... 6-22
_MC_COM.Status.CamTableBusy（カムテーブルファイル保存実行中／待機中）..... 6-22
_MC_COM.Status.GenerateCamBusy（カムテーブル生成実行中）..... 6-22
_MC_COM.Status.RunMode（MC 運転中）..... 6-22
_MC_COM.Status.TestMode（MC 試運転中）..... 6-22
_MC_GRP[*]（軸グループ変数）..... 6-30
_MC_GRP[*].Cfg.ExecID（実行 ID）..... 6-31
_MC_GRP[*].Cfg.GrpEnable（軸グループ使用）..... 6-31
_MC_GRP[*].Cfg.GrpNo（軸グループ番号）..... 6-31
_MC_GRP[*].Cmd.AccDec（指令補間加減速度）..... 6-31
_MC_GRP[*].Cmd.Vel（指令補間速度）..... 6-31
_MC_GRP[*].Details.Idle（停止中）..... 6-30
_MC_GRP[*].Details.InPosWaiting（インポジション待ち）..... 6-31
_MC_GRP[*].Kinematics.Axis[0]（構成軸（軸 A0）） 6-31
_MC_GRP[*].Kinematics.Axis[1]（構成軸（軸 A1）） 6-31
_MC_GRP[*].Kinematics.Axis[2]（構成軸（軸 A2）） 6-31
_MC_GRP[*].Kinematics.Axis[3]（構成軸（軸 A3）） 6-31
_MC_GRP[*].Kinematics.GrpType（軸構成）..... 6-31
_MC_GRP[*].MFaultLvl.Active（軸グループ 軽度フォールト発生中）..... 6-31
_MC_GRP[*].MFaultLvl.Code（軸グループ 軽度フォールトコード）..... 6-31

_MC_GRP[*].Obsr.Active（軸グループ 監視情報発生中）.....6-31
_MC_GRP[*].Obsr.Code（軸グループ 監視情報コード）.....6-31
_MC_GRP[*].Status.Disabled（軸グループ無効）....6-30
_MC_GRP[*].Status.ErrorStop（エラー減速停止中）6-30
_MC_GRP[*].Status.Moving（動作中）.....6-30
_MC_GRP[*].Status.Ready（起動準備完了）.....6-30
_MC_GRP[*].Status.Standby（停止中）.....6-30
_MC_GRP[*].Status.Stopping（減速停止中）.....6-30
MC 運転中.....6-22
MC 共通 監視情報.....6-22
MC 共通 監視情報コード.....6-22
MC 共通 監視情報発生中.....6-22
MC 共通 軽度フォールト.....6-22
MC 共通 軽度フォールトコード.....6-22
MC 共通 軽度フォールト発生中.....6-22
MC 共通ステータス.....6-22
MC 共通 部分停止フォールト.....6-22
MC 共通 部分停止フォールトコード.....6-22
MC 共通 部分停止フォールト発生中.....6-22
MC 共通変数.....6-22
MC 試運転機能.....4-2
MC 試運転中.....6-22
MC パラメータ設定.....5-2

P

PLCopen.....1-2
PLC 機能モジュール.....2-2
Process Data Objects (PDO).....2-19

S

Service Data Objects (SDO).....2-19
S 字.....9-35

あ

空きカムデータ.....9-16
アボーティング（中断）.....9-45, 9-61

い

位相.....9-15
位相間隔幅.....9-16
位置.....9-31
位置決め動作中.....6-23
位置指定ギア動作.....9-13
位置の種類.....9-31
位置偏差オーバー値.....5-23
位置偏差監視機能.....9-72
位置偏差警告値.....5-23
移動距離の変更.....9-41
インポジションチェック.....9-74
インポジションチェック時間.....5-20
インポジション幅.....5-20
インポジション待ち（軸制御ステータス）.....6-24
インポジション待ち（軸グループ制御ステータス）.....6-31

う

運動学変換設定.....6-31

え

エラー減速停止中.....6-23, 6-30
円弧補間.....9-53
エンコーダ軸.....3-2, 5-8, 5-9
エンコーダ種別.....5-24, 5-25

お

オブジェクトディクショナリ.....2-19
オリジナルカムデータ.....9-15
オーバーライド.....9-10, 9-56

か

開始位置方式指定.....9-16
外部ラッチ入力 1.....6-24
外部ラッチ入力 2.....6-24
カウントモード.....5-23, 6-26
加減算位置決め.....9-23
加減速オーバー.....5-19, 9-34
仮想エンコーダ軸.....3-2, 5-8, 5-9
仮想サーボ軸.....3-2, 5-8
加速度警告値.....5-19, 9-34
加速と減速.....9-34
加速度の単位.....9-34
加速度の変更.....9-42
カム曲線.....9-15
カム始点.....9-15
カム終点.....9-15
カムデータ.....9-15
カムデータインデックス.....9-16
カムデータの読み書き.....9-19
カムデータ変数.....9-15
カムテーブル.....9-15, 9-16
　カムテーブル仕様.....9-17
　カムテーブル生成.....9-20
　カムテーブルの切り替え.....9-18
　カムテーブルのデータ型.....9-17
　カムテーブルの保存.....9-19
　カムテーブル名.....6-35
　終点インデックス更新.....9-19
　プロパティ更新.....9-19
　ユーザプログラムでの指定方法.....6-35
カムテーブル開始位置.....9-16
カムテーブル生成実行中.....6-22
カムテーブルファイル保存実行中／待機中.....6-22
カム動作.....9-14, 9-15
カムブロック.....9-15
カムブロック始点.....9-15
カムブロック終点.....9-15
カムプロファイル曲線.....6-32, 9-15
　名前.....6-35

き

ギア動作 9-12
機械原点 8-2
起動準備完了 6-30
起動速度 5-19, 5-21, 9-33
基本データ型 6-20

け

限界入力停止方法 5-22
現在位置下限値 6-27
現在位置上限値 6-27
現在位置変更機能 9-68
現在方向指定 9-37
減速停止中 6-23, 6-30
減速度警告値 5-19, 9-34
減速度の単位 9-34
減速度の変更 9-42
原点位置オフセット 5-27, 8-10
原点位置範囲 5-20
原点確定 6-24
原点決め 8-2
原点近傍入力 6-24
原点近傍入力論理反転 5-23
原点検出方向 5-26, 8-8
原点停止 6-24
原点入力 6-24
原点入力信号 5-26, 8-7
原点入力マスク距離 5-27, 8-9
原点復帰アプローチ速度 5-27, 8-9
原点復帰押し当て時間 5-27, 8-10
原点位置オフセット 5-27, 8-10
原点復帰開始方向 5-26, 8-7
原点復帰加速度 5-27, 8-9
原点復帰減速度 5-27, 8-9
原点復帰ジャーク 5-27, 8-9
原点復帰速度 5-27, 8-9
原点復帰中 6-23
原点復帰動作 5-26, 8-6
原点復帰パラメータ 8-5
原点復帰補正速度 5-27, 8-11
原点復帰補正値 5-27, 8-10
原点未確定 8-2

こ

構成軸 5-31
構成軸（軸 A0） 6-31
構成軸（軸 A1） 6-31
構成軸（軸 A2） 6-31
構成軸（軸 A3） 6-31
構造体型 6-20
高速原点復帰 8-18
コーナースーパーインポーズ 9-64, 9-67

さ

再起動 9-39, 9-59
サイクリック同期位置決め 9-5
サイクリック同期位置(CSP)モード中 6-25
サイクリック同期速度制御 9-28
サイクリック同期速度(CSV)モード中 6-25
サイクリック同期トルク(CST)モード中 6-25
最高速度 5-19, 5-20, 9-33
最高補間速度 5-32
最大加速度 5-19, 9-34
最大カムデータ数 9-16
最大減速度 5-19, 9-34
最大補間加速度 5-32
最大補間減速度 5-32
サービスデータオブジェクト 2-19
サーボ ON 6-24
サーボ軸 3-2, 5-8
サーボドライバ状態 6-24
サーボレディ 6-24

し

試運転機能 4-2
軸変数と軸種別の関係 6-28
軸エラーリセット 9-3
軸 監視情報 6-26
軸 監視情報コード 6-26
軸 監視情報発生中 6-26
軸間偏差監視機能 9-73
軸起動準備完了 6-23
軸基本設定 6-26
軸グループ位置取得 9-52
軸グループエラーリセット 9-52
軸グループ 監視情報 6-31
軸グループ 監視情報コード 6-31
軸グループ 監視情報発生中 6-31
軸グループ基本設定 6-31
軸グループ 軽度フォールト 6-31
軸グループ 軽度フォールトコード 6-31
軸グループ 軽度フォールト発生中 6-31
軸グループ構成軸書込 9-51
軸グループ軸構成 5-30
軸グループ使用 5-31, 6-31
軸グループ指令値 6-31
軸グループステータス 6-30
軸グループ制御ステータス 6-30
軸グループ停止方法 5-33
軸グループの状態
　エラー減速停止中 6-9
　減速停止中 6-9
　停止中 6-9
　動作中 6-9
軸グループパラメータ 3-21
軸グループ番号 5-31, 6-31
軸グループ変数 3-21, 6-30
軸グループ変数名 3-23
軸グループ無効 6-30
軸グループ有効／無効 9-51
軸 軽度フォールト 6-26
軸 軽度フォールトコード 6-26
軸 軽度フォールト発生中 6-26
軸現在値 6-25

軸構成.....5-31, 6-31
軸種別.....5-6, 5-8, 6-26
軸使用.....5-6, 6-26
軸使用変更機能.....9-76
軸指令値.....6-25
軸ステータス.....6-23
軸制御ステータス.....6-24
軸の状態
　位置決め動作中.....6-8
　エラー減速停止中.....6-8
　減速停止中.....6-8
　原点復帰中.....6-8
　軸無効.....6-8
　多軸協調動作中.....6-8
　停止中.....6-8
　同期動作中.....6-8
　連続動作中.....6-8
軸パラメータ.....3-2
軸番号.....5-6, 5-7, 6-26
軸変数.....3-2, 6-22
軸変数名.....3-6
軸無効.....6-23
システム定義変数.....6-19
実行 ID.....6-26, 6-31
ジャーク.....1-2, 9-35
ジャークの単位.....9-35
従軸.....9-15
主回路電源.....6-24
主軸.....9-15
主軸位相補正.....9-24
主軸追従距離.....9-16
出力デバイス.....5-6, 5-10
ジョグ最高速度.....5-19, 9-33
指令位置.....6-17, 9-31
指令現在位置.....6-25
指令現在加減速度.....6-25
指令現在ジャーク.....6-25
指令現在速度.....6-25
指令現在トルク.....6-25
指令速度.....9-33
指令速度飽和.....6-24
指令方向ステータス.....6-24
指令補間加減速度.....6-31
指令補間速度.....6-31

せ

制御機能.....5-6, 5-9
正方向限界入力.....6-24
正方向限界入力時動作.....5-27, 8-8
正方向限界入力論理反転.....5-22
正方向指定.....9-37
正方向指令中.....6-24
正方向ソフトウェアリミット.....5-23
正方向トルク警告値.....5-19
正方向トルク制限上限値.....5-22
接続加速度.....9-16
接続速度.....9-16
絶対値位置決め.....9-4
絶対値エンコーダ
　原点位置オフセット.....8-15
　セットアップ.....8-16
　対象となるサーボモータ.....8-15
　ロータリモードでの使用.....8-16

そ

相対値位置決め.....9-4
即停止入力.....6-24
即停止入力停止方法.....5-22
即停止入力論理反転.....5-22
速度警告値.....5-19, 9-33
速度制御.....9-28
速度の種類.....9-32
速度の単位.....9-33
ソフトウェアリミット機能.....5-23, 9-70
ゾーン.....9-70

た

台形パターンカム動作.....9-22
タイムスタンプ.....6-26
多軸協調制御.....9-50
多軸協調制御での停止.....9-54
多軸協調動作中.....6-23
多重起動.....9-44, 9-60
タスク.....2-5
タスクの周期.....2-10
単位変換.....6-26
単軸制御の共通機能.....9-31
単軸速度制御.....9-28
単軸トルク制御.....9-30

ち

近回り指定.....9-37
中心点補正許容率.....5-33
直線補間.....9-52

て

停止.....9-6
　MC_GroupImmediateStop（軸グループ即停止）命令.....9-54
　MC_GroupStop（軸グループ強制停止）命令.....9-54
　MC_ImmediateStop（即停止）命令.....9-7
　MC_Stop（強制停止）命令.....9-7
　異常の発生などによる停止.....9-8, 9-55
　限界入力.....9-6
　サーボドライバの入力信号.....9-6
　即停止入力.....9-6
　停止方法.....9-9
　　指令値を減速停止.....9-9
　　指令値を即停止.....9-9
　　指令値を即停止し、同時にサーボ OFF.....9-10
　　指令値を即停止し偏差カウンタリセット.....9-10
停止中（軸ステータス）.....6-23
停止中（軸制御ステータス）.....6-24

停止中（軸グループステータス）.....6-30
停止中（軸グループ制御ステータス）.....6-30
停止の優先順位.....9-10
定周期タスク.....2-6
デジタルカムスイッチ有効機能.....9-76
電子ギア比.....5-13
データ型.....6-20

と

同期制御.....9-12
同期動作中.....6-23
動作中.....6-30
動作方向指定.....9-36
ドライバエラー入力.....6-24
ドライバエラーリセット監視時間.....5-22
ドライバ内部機能制限中.....6-25
ドライバワーニング入力.....6-24
トランジション無効.....9-64
トランジションモード.....9-64
トルク指令の変更.....9-43
トルク制限機能.....9-68

に

入力デバイス.....5-6, 5-10
任意座標系の 3D モーションモニタ表示.....9-77

の

ノードアドレス.....6-26

は

派生データ型.....6-20
バッファモード.....9-44, 9-60
バッファード（待機）.....9-46, 9-62
反転時動作.....5-19
バージョン.....22

ひ

表示単位.....5-11, 5-13, 6-26

ふ

フィードバック位置.....6-17, 9-31
フィードバック現在位置.....6-25
フィードバック現在速度.....6-25
フィードバック現在トルク.....6-25
フィードバック速度.....9-33
フィードバック速度フィルタ時定数.....5-20, 9-33
負方向限界入力.....6-24
負方向限界入力時動作.....5-27, 8-8
負方向限界入力論理反転.....5-23
負方向指定.....9-37
負方向指令中.....6-24
負方向ソフトウェアリミット.....5-23
負方向トルク警告値.....5-19
負方向トルク制限上限値.....5-22
プライマリ周期.....2-7, 2-22, 2-23
プライマリ定周期タスク.....2-5
ブレンディング（混合）.....9-46, 9-62
プログラム変更カムデータ.....9-15
プロセスデータオブジェクト.....2-19
プロセスデータ通信周期.....2-22

へ

変位.....9-15
偏差カウンタリセット.....9-72

ほ

方向指定なし.....9-37
補間加減速オーバー.....5-33
補間加速度警告値.....5-33
補間減速度警告値.....5-33
補間速度警告値.....5-33

む

無限長軸.....5-24
無効カムデータ.....9-15

も

目標位置の変更.....9-40
　新たな指令値により反転しない場合.....9-40
　新たな指令値により反転する場合.....9-40
　減速オーバーのパターン.....9-40
　三角制御のパターン.....9-40
目標速度の変更.....9-42
モーション制御.....5-6, 5-8, 5-31
モーション制御機能モジュール.....2-2
モーション制御システム変数
　属性.....6-20
モーション制御周期.....2-22
モーション制御パラメータ設定.....5-2
モーション制御プログラム.....6-2
モーション制御命令
　Enable 型.....6-13
　Enable 型の命令実行タイミングチャート.....6-14
　Execute 型.....6-13
　Execute 型の命令実行タイミングチャート.....6-13
　出力ステータス.....6-11
　出力の排他性.....6-11
　出力変数「Active（制御中）」.....6-12
　出力変数「Busy（実行中）」の動作.....6-12
　出力変数「CommandAborted（実行中断）」の動作6-12
　出力変数「Done（完了）」の動作.....6-12
　入力パラメータ.....6-11
モータ 1 回転の移動量.....5-11, 6-26
モータ 1 回転のパルス数.....5-11, 6-26

ゆ

有限長軸.....5-24
有効カムデータ.....9-15
有効カムデータ数.....9-15

ユーザプログラムでの軸グループの指定方法.....3-21
ユーザプログラムでの軸の指定方法.....3-3

ら

ラッチ機能.....9-69

り

リニアモード.....5-24
リングカウンタ下限設定値..... 5-24, 5-25
リングカウンタ上限設定値..... 5-23, 5-25

れ

列挙型..... 6-20
連続動作中.....6-23

ろ

ロータリモード.....5-24

わ

割り込み定寸位置決め.....9-4

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-433N　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