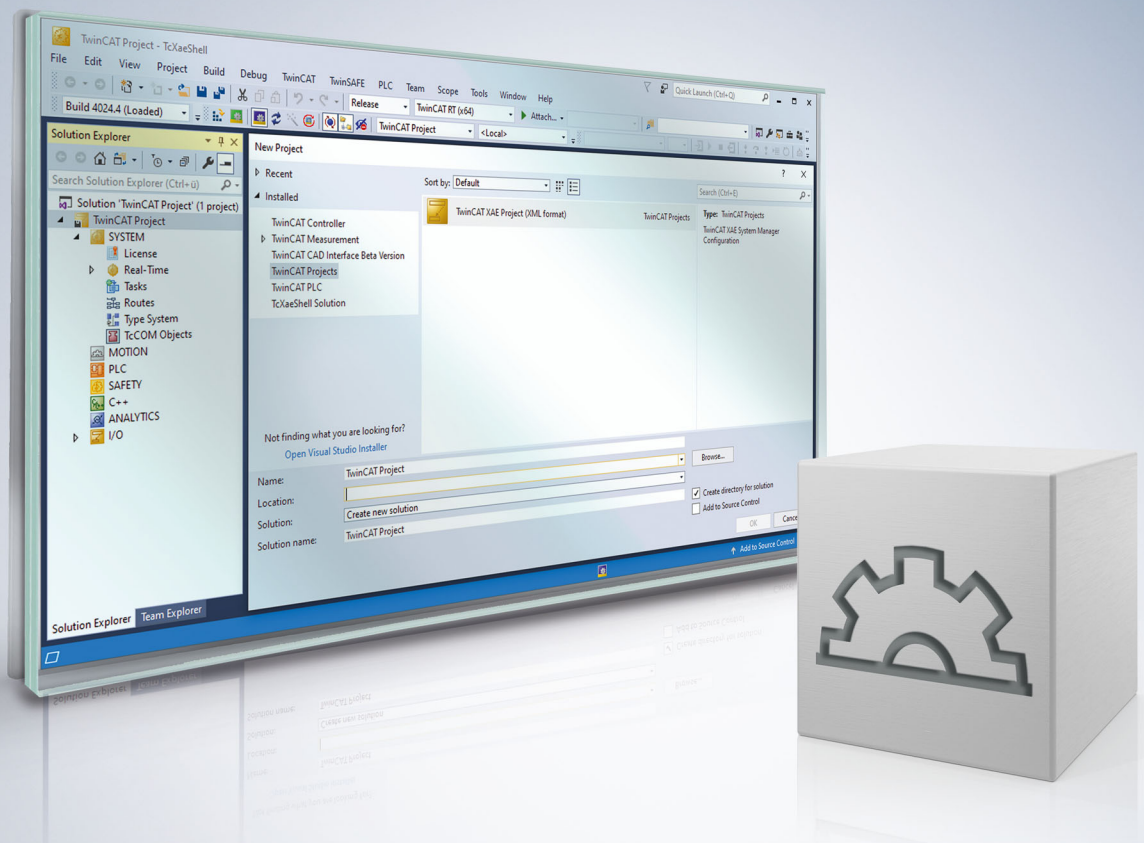
BECKHOFF New Automation Technology

Handbuch | DE

TF5420

TwinCAT 3 | Motion Pick-and-Place



13.08.2025 | Version: 1.15.0

Inhaltsverzeichnis

1 Vorwort ... 5
1.1 Hinweise zur Dokumentation ... 5
1.2 Zu Ihrer Sicherheit ... 6
1.3 Hinweise zur Informationssicherheit ... 7
2 Einführung ... 8
3 Übersicht der neuen Funktionen ... 9
4 Zustandsdiagramm ... 11
5 MC-Gruppe (TF5420 TwinCAT 3 Motion Pick-and-Place) ... 13
5.1 Konfigurieren einer MC-Gruppe ... 13
5.2 MC Group Coordinated Motion ... 18
5.3 MC Group with Pick-and-Place ... 20
6 Räumliche Konfiguration ... 22
6.1 Objekt Coordinate Frame ... 22
6.2 Objekt Conveyor Tracking ... 22
6.3 Conveyor Tracking-Verhalten ... 22
6.4 Objekt Node Connector ... 23
6.5 Konfigurieren eines Node Connectors ... 23
6.6 Konfigurieren für MC_TrackConveyorBelt ... 28
6.7 Hintergrundinformationen ... 35
7 SPS-Bibliotheken ... 38
7.1 Tc3_McCoordinatedMotion ... 38
7.1.1 Funktionsbausteine ... 40
7.1.2 Datentypen ... 78
7.2 Tc3_Mc3Definitions ... 88
7.2.1 Datentypen ... 88
8 Beispiele ... 98
9 Anhang ... 99
9.1 Zyklische Gruppenschnittstelle ... 99
9.1.1 NcToPlc ... 99
9.1.2 PlcToNc ... 100
9.2 Index-Offset-Spezifikation für MC-Gruppenparameter ... 101
9.3 Unterschiede zwischen MC2 und MC3 ... 101
9.4 MC_LREAL/Spezielle Eingangswerte ... 102
9.5 Globale Datentypen ... 102
9.5.1 MC_GROUP_TYPE ... 102
9.5.2 MC_GROUP_STATE ... 102
9.5.3 ST_GroupStatus ... 103
9.5.4 ST_GroupStatusFlags ... 103

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.
Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.
Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.
Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.
Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.
Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Kennzeichnungen führen.

EtherCAT® ist eine eingetragene Marke und patentierte Technologie, lizenziert durch die Beckhoff Automation GmbH, Deutschland.

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.
Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.
Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

Fremdmarken

In dieser Dokumentation können Marken Dritter verwendet werden. Die zugehörigen Markenvermerke finden Sie unter: https://www.beckhoff.com/trademarks.

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

⚠ GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

⚠ WARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

⚠ VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt

Diese Information beinhaltet z. B.:
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem https://www.beckhoff.de/secguide.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter https://www.beckhoff.de/secinfo.

2 Einführung

Das Softwarepaket TF5420 TwinCAT 3 Motion Pick-and-Place wird zusammen mit dem Softwarepaket TF5400 installiert.

Zielsystem

Windows 10

TwinCAT 3 Motion Pick-and-Place

TF5420 TwinCAT 3 Motion Pick-and-Place führt mehrdimensionale Bewegungen aus. Es wurde speziell für die Anforderungen von Pick-and-Place-Anwendungen entwickelt. Es soll auf Bewegungen angewendet werden, bei denen die genaue Bahndynamik in Segmentübergängen nicht so wichtig ist, der Benutzer jedoch möglichst schnell von einem Punkt zu einem anderen gelangen möchte. Alle zugehörigen Funktionsbausteine sind in der Bibliothek *Tc3_McCoordinatedMotion* enthalten.

Zusätzliche Lizenzanforderungen

TF5420 TwinCAT 3 Motion Pick-and-Place benötigt die Lizenz TC1260.

3 Übersicht der neuen Funktionen

Ab TF5400 V3.3.72

- Neu: Im Zustand GroupDisabled sind PTP-Kommandos der Tc2_MC2-Bibliothek (z. B. MC_MoveAbsolute) auf Achsen der Gruppe erlaubt.

Ab TF5400 V3.3.19

- MC-Group Pick-and-Place nicht mehr verfügbar.
- Neu: Funktionsbaustein MC_SetCoordinateTransformPreparation [▶ 55] für die MC-Group Coordinated Motion
- Erfordert TwinCAT V3.1.4024.40 oder höher

Ab TF5400 V3.2.27 für die MC Group Coordinated Motion:

- Neu: Einführung von weiteren dynamischen Randbedingungen (Constraints) für Bahn- und Hilfsachsen.
- Neu: Optional wirkt der Override auch auf die Synchronisierungsphase beim MC_TrackConveyorBelt [▶ 56].
- Optimierungen am MC_TrackConveyorBelt [▶ 56], die einen SAF-Zyklusversatz zwischen Conveyor (Master-) und Slaveachse unterbinden.
- Optimierungen der Fehlerreaktion beim MC_TrackConveyorBelt [▶ 56]. Im Falle eines Laufzeitfehlers des Förderbandes (Master), wird ein aktives MC_MovePath [▶ 71] nicht abgebrochen und eine Fehlerreaktion ist über die SPS anzustoßen.
- Benötigt eine x64-Plattform

Ab TF5400 V3.1.10.64:

- Neu: In einer MC Group Coordinated Motion mit Geo Blending wird ein Blocker, der früh genug ausgelöst wird, bevor er aktiv wird, wird überblendet und ohne Unterbrechung weitergegeben.
- Erfordert TwinCAT V3.1.4024.24 oder höher

Ab TF5400 V3.1.10.1:

- Neu: Gruppentyp MC Group Coordinated Motion.
- Neu: Daten für die MC Group Coordinated Motion in der zyklischen Gruppenschnittstelle [▶ 99].
- Neu: Funktionsbausteine für MC Group Coordinated Motion:
  - MC_BlockerPreparation [▶ 73]
  - MC_ReleaseBlocker [▶ 74]
  - MC_GroupReadBlockerStatus [▶ 76]
  - MC_DwellTimePreparation [▶ 77]
- Neu: Funktionsbaustein MC_GroupHalt [▶ 61] für MC Group Coordinated Motion.
- Neu: Enum MC_TRANSITION_MODE [▶ 85] erweitert um mcTransModeCornerDistance für MC Group Coordinated Motion.
- Neu: Enum MC_CIRC_PATHCHOICE [▶ 83] erweitert um mcCircPathchoiceShortSegment und mcCircPathchoiceLongSegment.
- Erfordert TwinCAT V3.1.4024.7 oder höher

Ab TF5400 V3.1.6.27:

- Neu: Die verbleibende Zeit und Strecke des aktuellen Segments können nun über ADS gelesen werden, siehe Index-Offset-Spezifikation für MC-Gruppenparameter [▶ 101].
- Erfordert TwinCAT V3.1.4022.0 oder höher

Ab TF5400 V3.1.6.3:

- Neu: Funktionsbaustein MC_SetCoordinateTransform [▶ 53].

- Neu: Funktionsbaustein MC_TrackConveyorBelt [▶ 56]

Ab TF5400 V3.1.4.4:

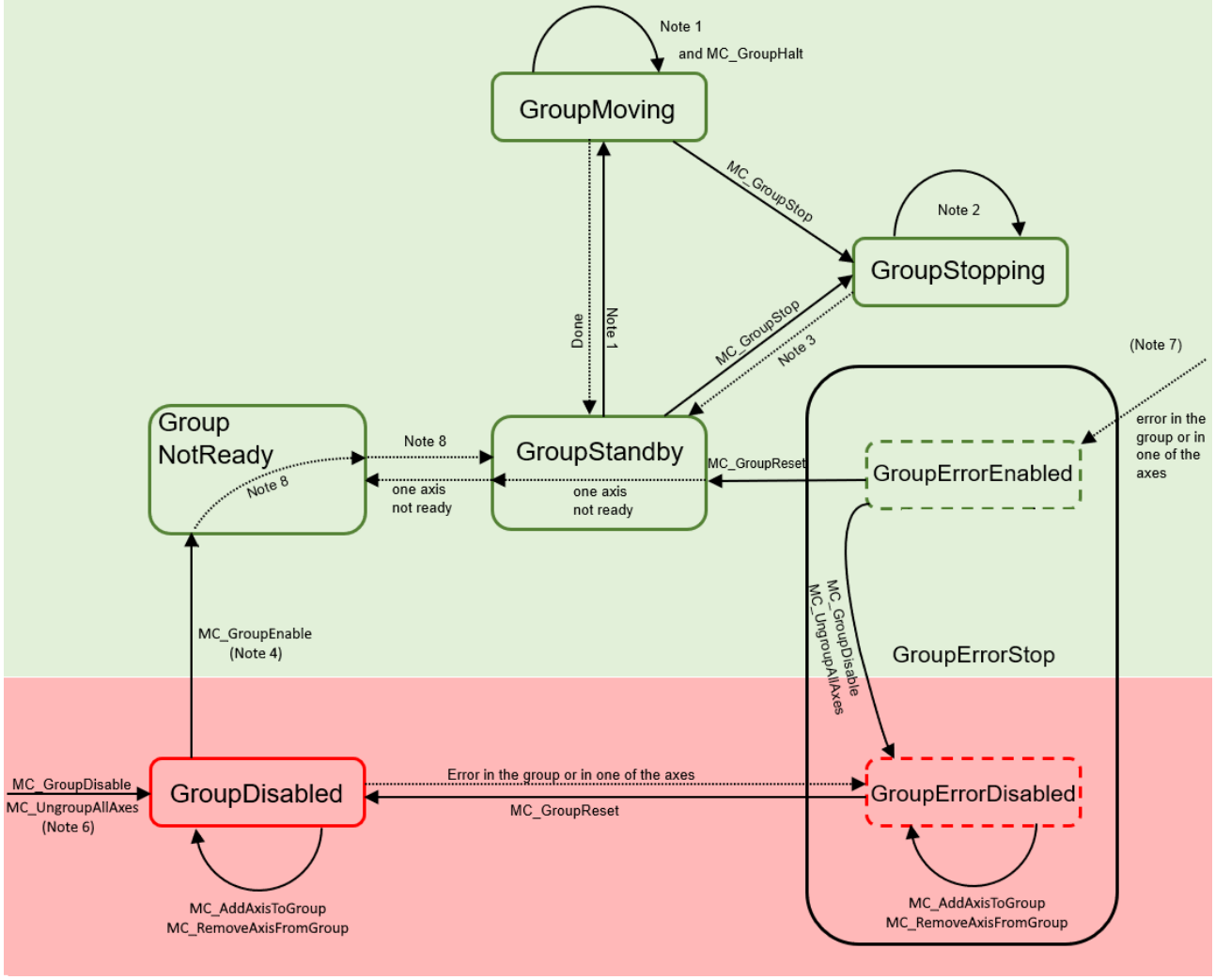
- Neu: Spezieller Eingangswert [▶ 102] `MC_MAXIMUM`.

Ab TF5400 V3.1.2.47:

- Neu: Funktionsbaustein MC_MoveCircularAbsolutePreparation.

4 Zustandsdiagramm

Das Zustandsdiagramm beschreibt den Zustand einer Coordinated-Motion-Gruppe. Die hier beschriebenen Zustände können mit Hilfe des Funktionsbausteins MC_GroupReadStatus aus der SPS ausgelesen werden.



Hinweis	Beschreibung
1	Anwendbar für alle nichtadministrativen (Bewegungs-) Funktionsbausteine.
2	Im Zustand GroupStopping können zwar viele Funktionsbausteine aufgerufen werden, sie werden jedoch nicht ausgeführt. Ausnahmen sind MC_GroupDisable [▶ 43] und MC_UngroupAllAxes [▶ 59], welche den Stopp abbrechen und den Übergang zum Zustand GroupDisabled erzeugen.
3	MC_GroupStop [▶ 63].DONE
4	Die Anzahl der Achsen in der Gruppe (hinzugefügt über MC_AddAxisToGroup [▶ 40]) muss der Achsanzahl der Raumachskonvention plus dem Additional Axes Count entsprechen.
5	-

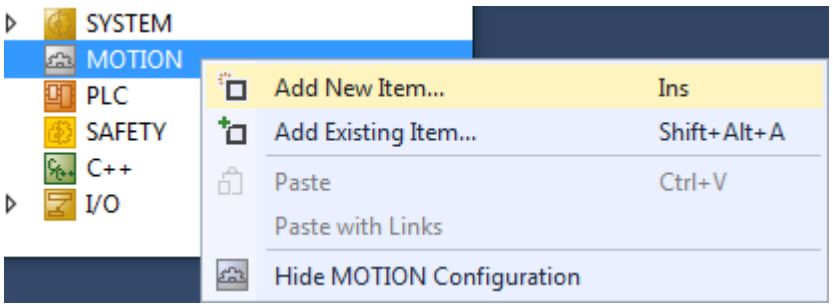
6	MC_GroupDisable kann in allen Zuständen aufgerufen werden und ändert den Zustand in GroupDisabled. Wenn MC_GroupDisable in einem Fehlerzustand aufgerufen wird, ändert sich der Zustand in GroupErrorDisabled.
7	Der Zustandswechsel zu GroupErrorEnabled erfolgt im Achs-/ Gruppenfehlerfall aus jedem Zustand, in dem die Gruppe enabled ist.
8	Der Zustandswechsel erfolgt, wenn "bIsControlLoopClosed" für alle Achsen TRUE ist. "bPositiveDirection"/"bNegativeDirection" müssen nicht aktiviert sein.
9	-
10	MC_GroupReset [▶ 49] hat keine Wirkung, wenn der Zustand ein anderer ist als GroupErrorStop. Um den Zustand GroupErrorStop zu verlassen, muss MC_GroupReset aufgerufen werden.

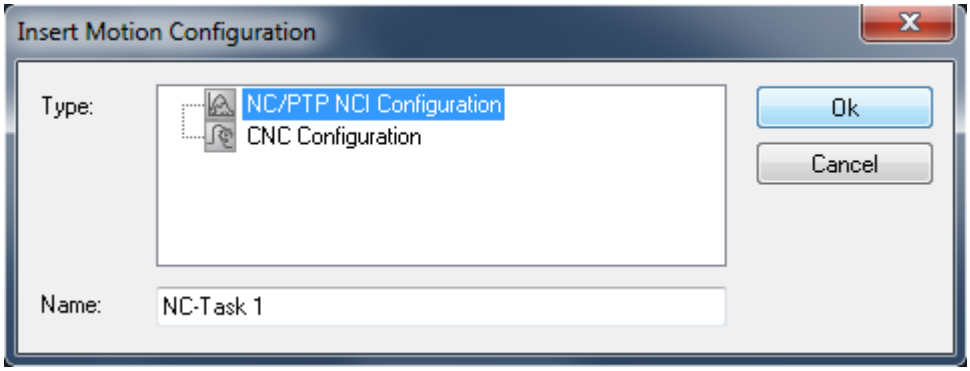
5 MC-Gruppe (TF5420 TwinCAT 3 Motion Pick-and-Place)

5.1 Konfigurieren einer MC-Gruppe

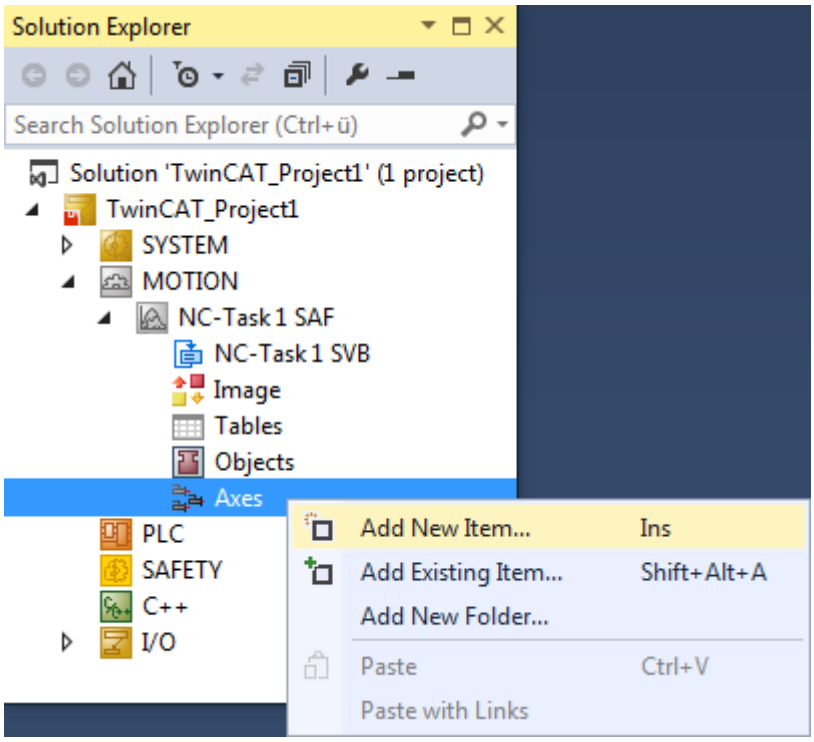
Grundsätzlich gilt die hier beschriebene Konfiguration für alle Motion Objects im Advanced Motion Pack.

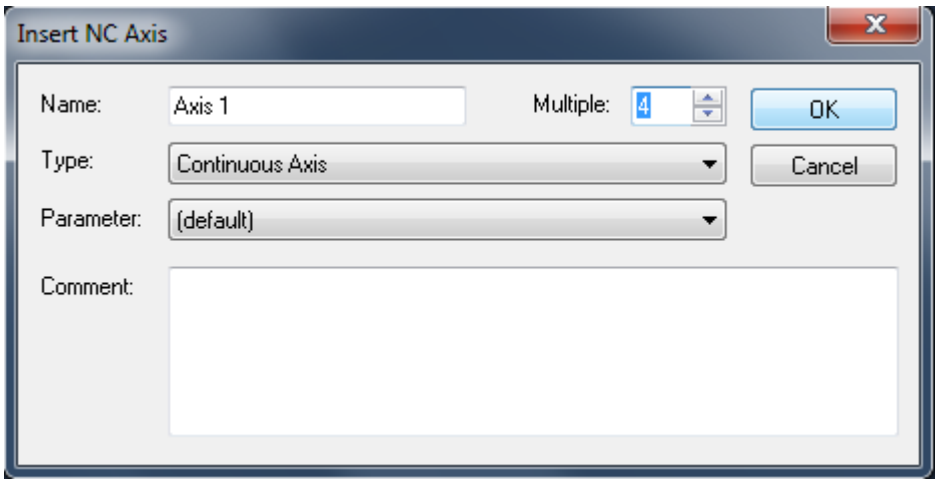
1. Fügen Sie im Bereich Motion eine neue „NC/PTP NCI Configuration“ hinzu.



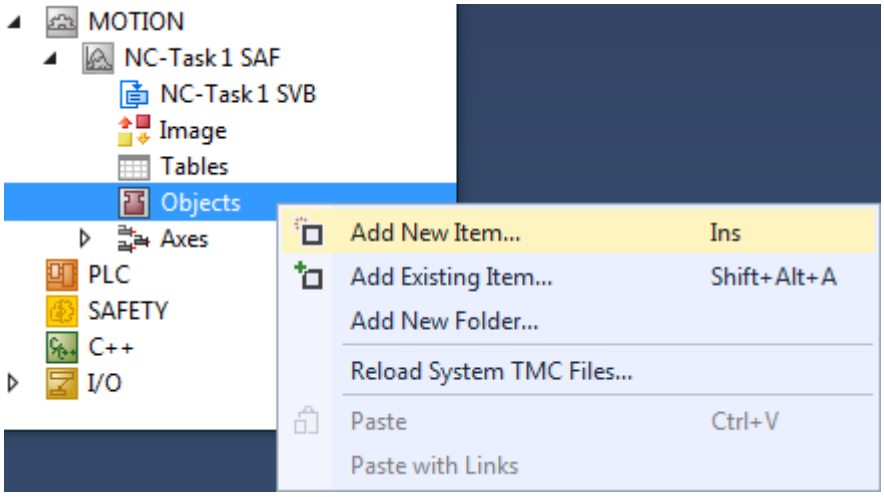


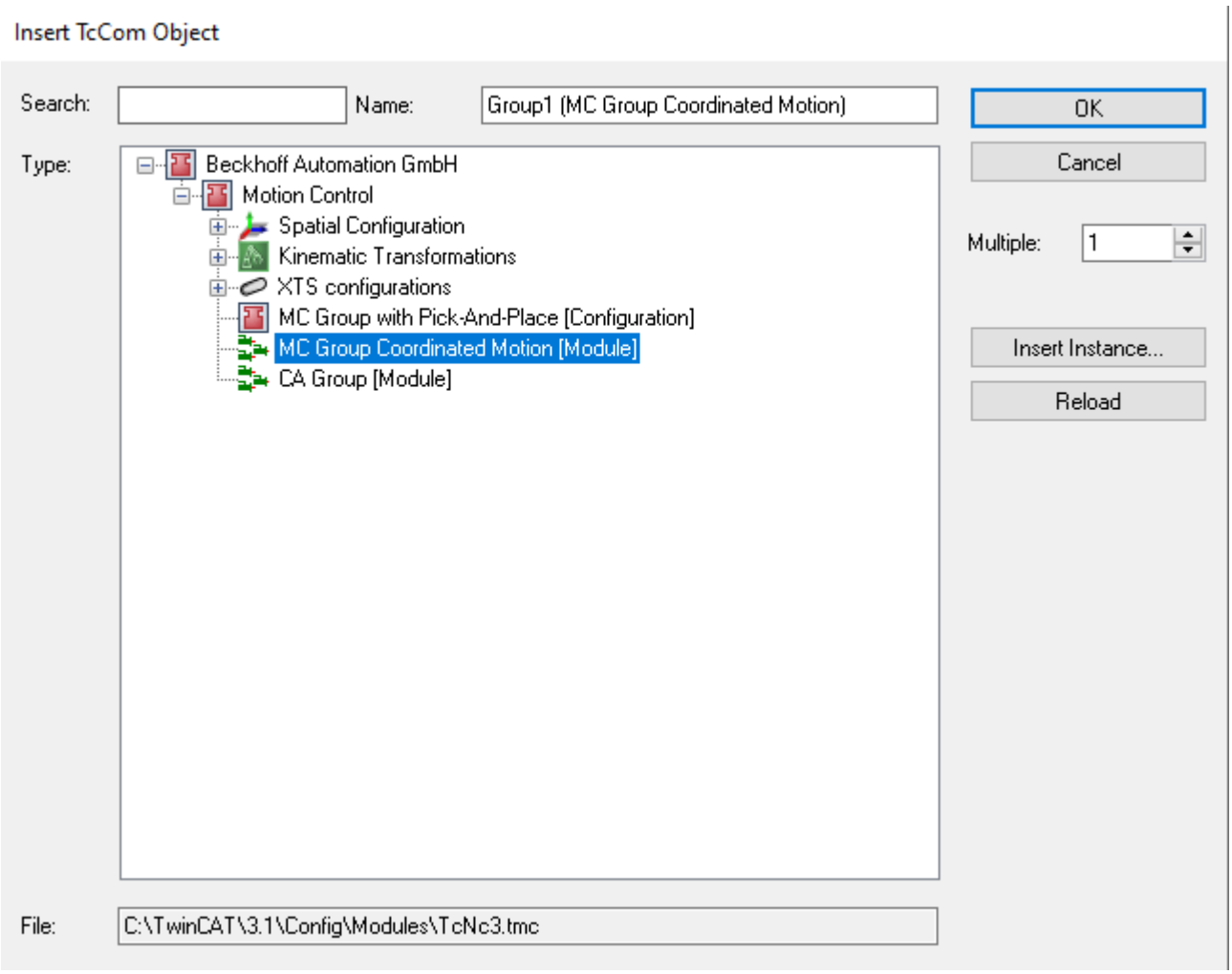
2. Fügen Sie der NC-Konfiguration alle Achsen hinzu.

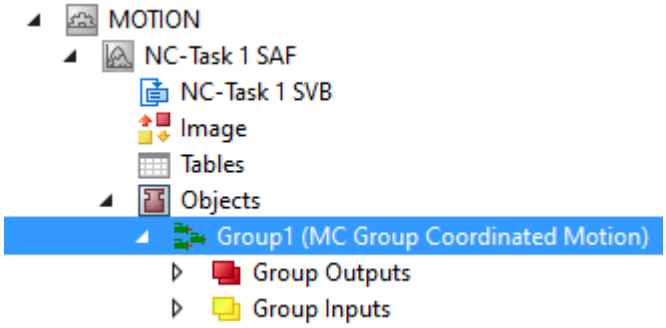




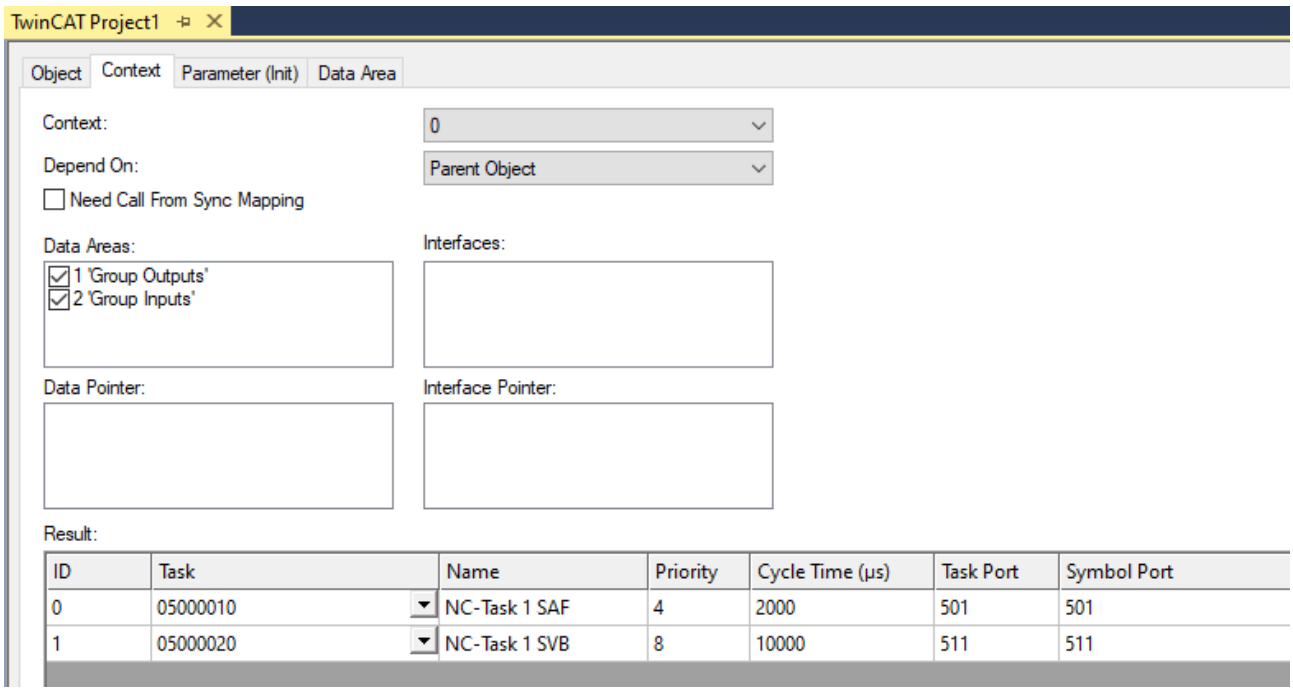
3. Fügen Sie dem Eintrag „Objects“ in der NC-Konfiguration die entsprechende Gruppe hinzu:
   Für koordinierte Bewegung, mehrdimensionale Bewegungen: MC Group Coordinated Motion [▶ 18]
   oder MC Group with Pick-and-Place [▶ 20].







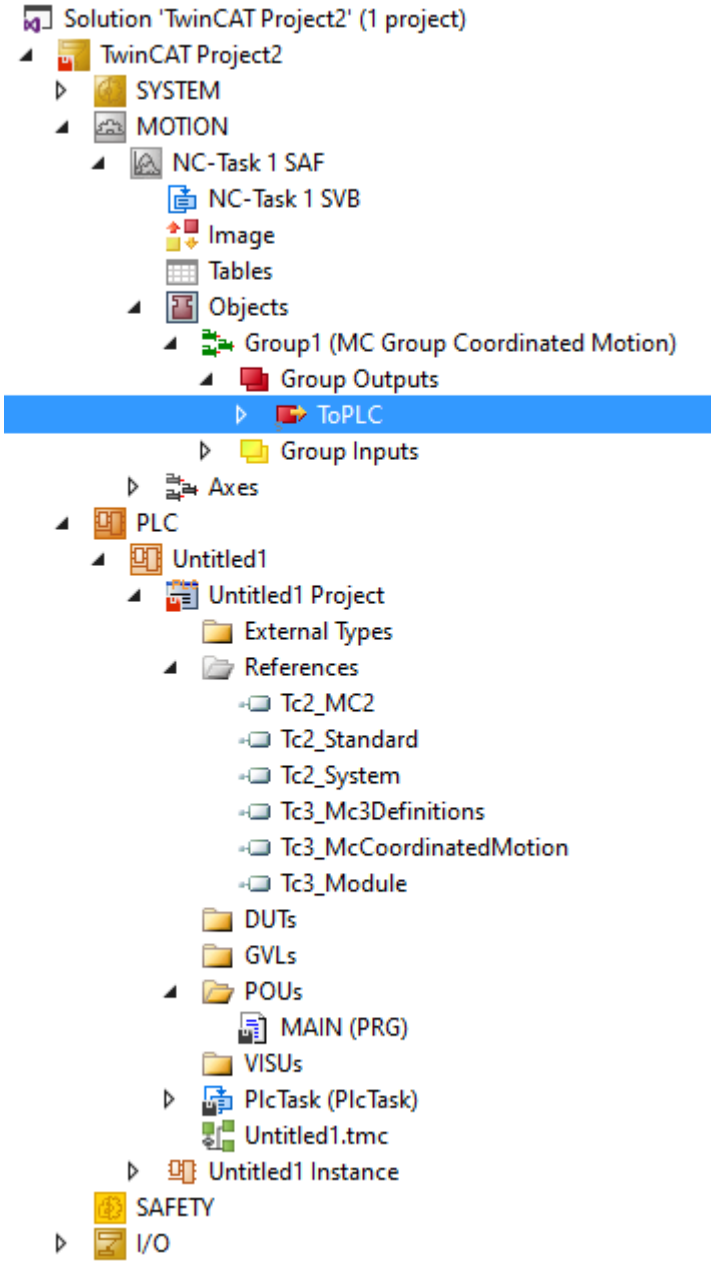
4. Prüfen Sie die Tasks in der Gruppe.
   Kontext-ID 0 muss auf **„NC-Task 1 SAF“** gesetzt sein.
   Kontext-ID 1 muss auf **„NC-Task 1 SVB“** gesetzt sein.

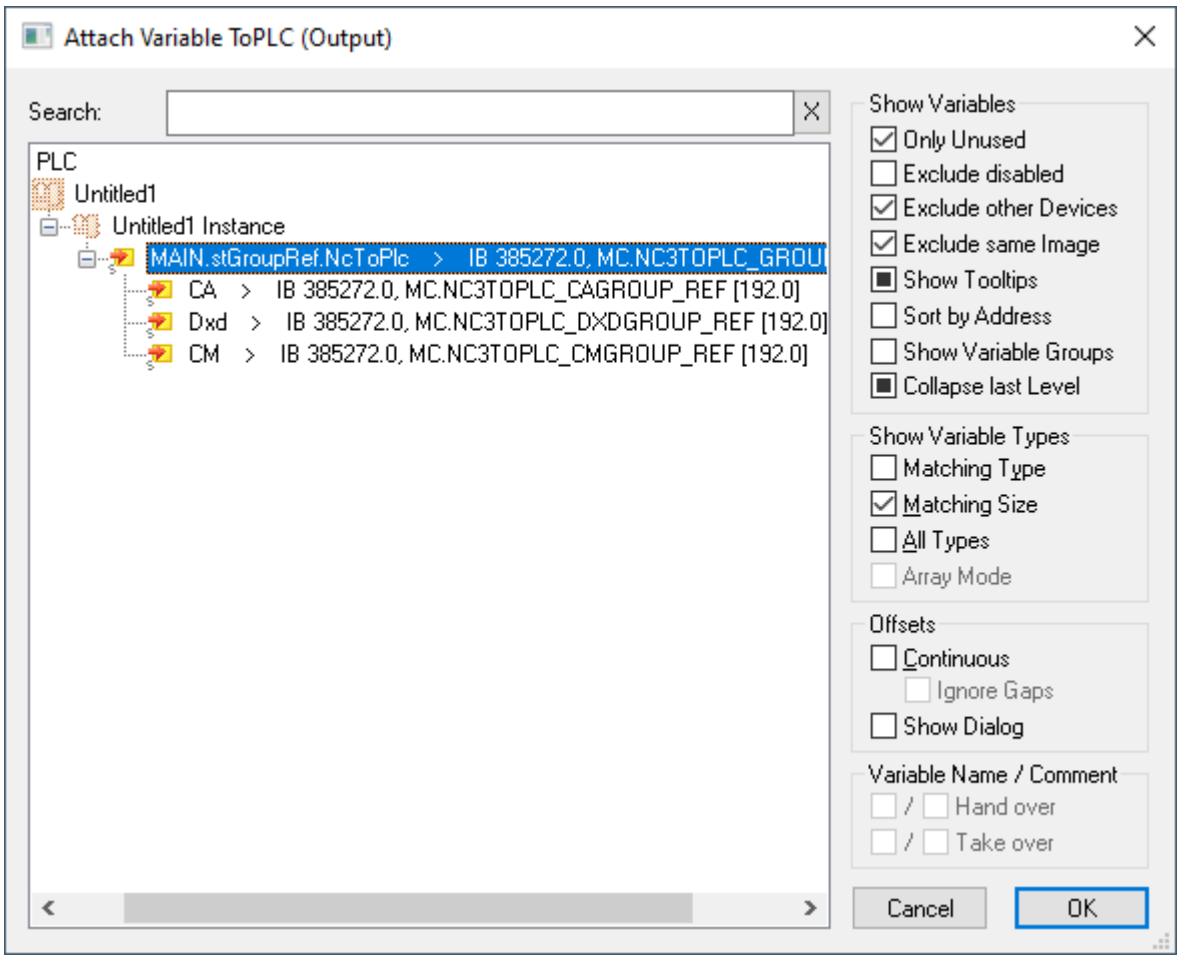


5. Konfigurieren Sie die Gruppenparameter entsprechend der gewünschten Anwendung.
   Für weitere Erläuterungen zu den Gruppenparametern siehe die folgenden Abschnitte.
6. Um die Gruppe aus der SPS zu adressieren, muss eine zyklische Schnittstelle deklariert und mit den I/Os der Gruppe verknüpft werden (siehe SPS-Bibliothek Tc3_McCoordinatedMotion). Um die Achsen zu adressieren und zu aktivieren, muss die Bibliothek „Tc2_Mc2“ dem Projekt hinzugefügt werden.

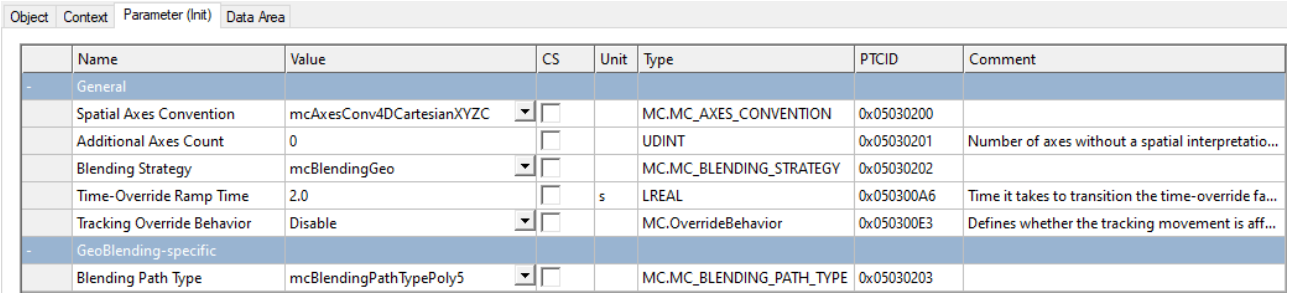
⇨ Eine neue „NC/PTP NCI Configuration“ wurde erstellt.

```
VAR
    stGroupRef : AXES_GROUP_REF;
END_VAR
```


Solution 'TwinCAT Project2' (1 project)
TwinCAT Project2
SYSTEM
MOTION
NC-Task 1 SAF
NC-Task 1 SVB
Image
Tables
Objects
Group1 (MC Group Coordinated Motion)
Group Outputs
ToPLC
Group Inputs
Axes
PLC
Untitled1
Untitled1 Project
External Types
References
Tc2_MC2
Tc2_Standard
Tc2_System
Tc3_Mc3Definitions
Tc3_McCoordinatedMotion
Tc3_Module
DUTs
GVLs
POUs
MAIN (PRG)
VISUs
PlcTask (PlcTask)
Untitled1.tmc
Untitled1 Instance
SAFETY
I/O



5.2 MC Group Coordinated Motion



Object | Context | Parameter (Init) | Data Area

	Name	Value	CS	Unit	Type	PTCID	Comment
-	General						
	Spatial Axes Convention	mcAxesConv4DCartesianXYZC			MC.MC_AXES_CONVENTION	0x05030200	
	Additional Axes Count	0			UDINT	0x05030201	Number of axes without a spatial interpretatio...
	Blending Strategy	mcBlendingGeo			MC.MC_BLENDING_STRATEGY	0x05030202	
	Time-Override Ramp Time	2.0		s	LREAL	0x050300A6	Time it takes to transition the time-override fa...
	Tracking Override Behavior	Disable			MC.OverrideBehavior	0x050300E3	Defines whether the tracking movement is aff...
-	GeoBlending-specific						
	Blending Path Type	mcBlendingPathTypePoly5			MC.MC_BLENDING_PATH_TYPE	0x05030203	

Parameter (Init)

Raumachskonventionen

Es können drei Achskonventionen eingestellt werden.

Die Achskonventionen legen fest, wie die Achsen in der Achsgruppe interpretiert werden. In Kombination mit „Additional Axes Count“ definieren sie die Dimension der Achsgruppe und damit die Anzahl der Achsen, die hinzugefügt werden müssen, sowie die Art und Weise, wie jede der hinzugefügten Achsen interpretiert wird.

Parameter	Wert	Typ	Beschreibung
Spatial Axes Convention	mcAxesConv2DCartesianXY	`MC.MC_AXES_CONVENTION`	Eine 2D-Gruppe bestehend aus `X`, `Y`. Die Reihenfolge der translatorischen Achsen in der Konfiguration bestimmt die Translationsreihenfolge.

Parameter	Wert	Typ	Beschreibung
Spatial Axes Convention	mcAxesConv3DCartesianXYZ	`MC.MC_AXES_CONVENTION`	Eine 3D-Gruppe bestehend aus `X`, `Y`, `Z`. Die Reihenfolge der translatorischen Achsen in der Konfiguration bestimmt die Translationsreihenfolge.
Spatial Axes Convention	mcAxesConv4DCartesianXYZC	`MC.MC_AXES_CONVENTION`	Eine 4D-Gruppe bestehend aus `X`, `Y`, `Z` und einer rotatorischen Achse um `Z` (`C`). Die Reihenfolge der translatorischen und rotatorischen Achsen in der Konfiguration bestimmt die Translations- und Rotationsreihenfolge.

Additional Axes Count

Anzahl der Achsen in der Achsgruppe, die keine geometrische Interpretation haben. Zwischen 0 und 8 Achsen dieser Art können eingefügt werden.

Blending Strategy

Legt die Blending-Strategie fest.

Parameter	Wert	Typ	Beschreibung
Blending Strategy	mcBlendingGeo	`MC.MC_BLENDING_STRATEGY`	Die Blending-Bahn wird geometrisch definiert und dann mit der für die Bahn zulässigen Dynamik ausgeführt.
Blending Strategy	mcBlendingSuperpos	`MC.MC_BLENDING_STRATEGY`	Die Blending-Bahn ergibt sich dynamisch aus der Überlagerung von zwei Segmenten im Blending-Bereich.

Time Override Ramp Time

Rampenzeit für Override-Änderung von 0 % bis 100 %. Das Time-Override wird dem eigentlichen Profil überlagert. Dadurch können sich während der Overrideänderungen in Summe höhere Dynamiken ergeben, als an der Gruppe parametriert wurden.

Tracking Override Behavior

Definiert, ob der Anteil der Bewegung zum Aufsynchronisieren Override behaftet ist oder nicht.

Verfügbar ab TF5400 3.2.27

Parameter	Wert	Typ	Beschreibung
Tracking Override Behavior	Disable	`MC.OverrideBehavior`	Der Gruppenoverride hat keine Auswirkungen auf den Anteil der Bewegung zum Aufsynchronisieren.
Tracking Override Behavior	Enable	`MC.OverrideBehavior`	Der Gruppenoverride wirkt auch auf den Anteil der Bewegung zum Aufsynchronisieren.

GeoBlending-spezifische Parameter

Blending Path Type

Definiert die Geometrie, die für die Blending-Bahn verwendet wird.

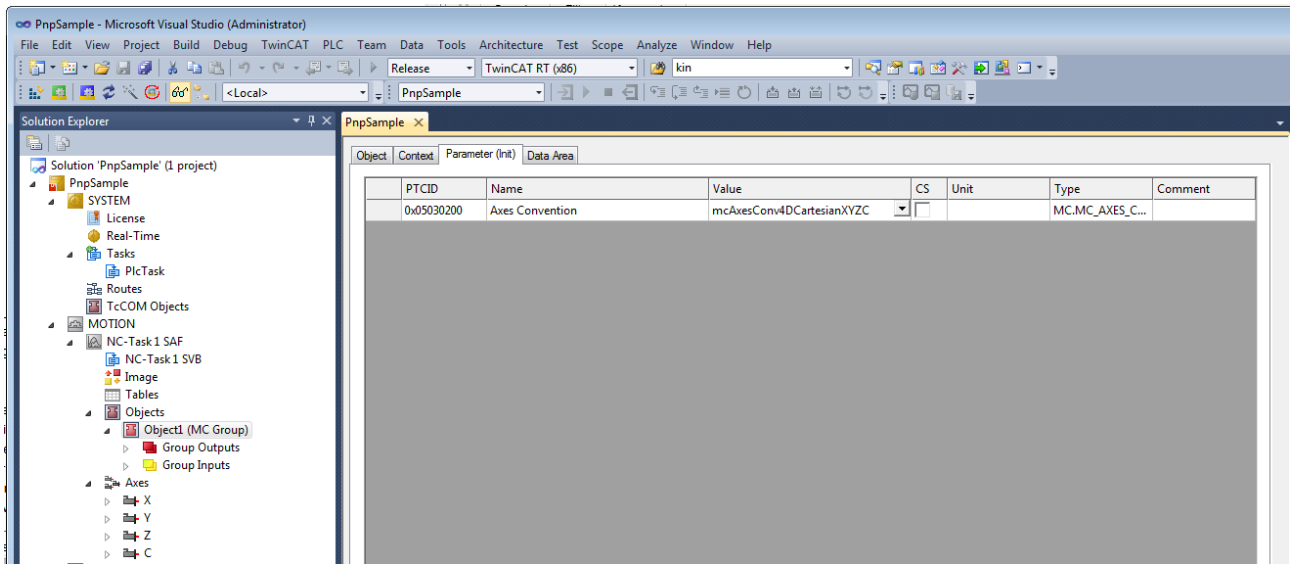
Parameter	Wert	Typ	Beschreibung
Blending Path Type	mcBlendingPathTypeIgnore	`MC.MC_BLENDING_PATH_TYPE`	Kein Blending.
Blending Path Type	mcBlendingPathTypePoly5	`MC.MC_BLENDING_PATH_TYPE`	Die Blending-Bahn verwendet ein Polynom fünften Grades.

5.3 MC Group with Pick-and-Place

MC Group with Pick-and-Place nicht mehr verfügbar

Anstatt der MC Group with Pick-and-Place ist die MC Group Coordinated Motion [▶ 18] zu verwenden. Bis zur TF5400 TwinCAT Advanced Motion Pack Version 3.2.27 können bestehende Projekte mit der MC Group with Pick-and-Place geöffnet und ausgeführt werden.

Die MC-Gruppe verbindet Achsen, um eine mehrdimensionale Bewegung auszuführen.



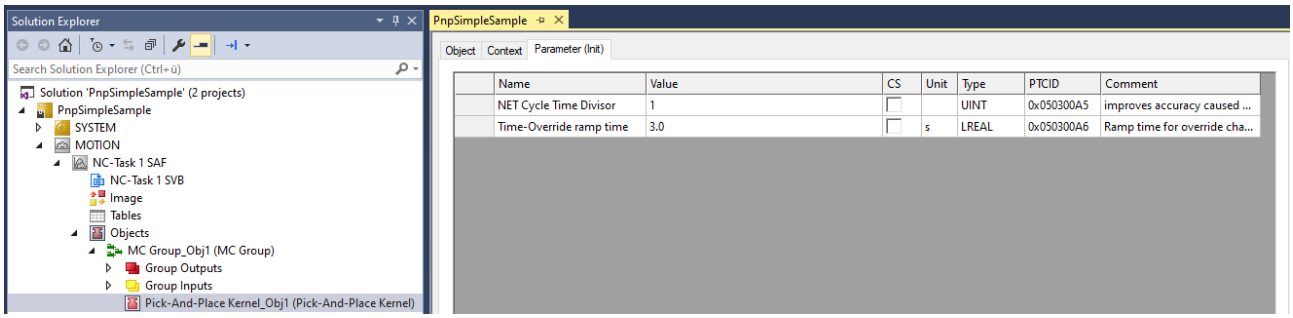
Achskonventionen

Registerkarte: Parameter (Init). Es können drei Achskonventionen eingestellt werden.

Die Achskonventionen legen fest, wie die Achsen in der Achsgruppe interpretiert werden. Sie definieren die Dimension der Achsgruppe und damit die Anzahl der Achsen, die hinzugefügt werden müssen, sowie die Art und Weise, wie jede der hinzugefügten Achsen interpretiert wird.

Parameter	Wert	Typ	Beschreibung
Axes Convention	mcAxesConv2DCartesianXY	`MC.MC_AXES_CONVENTION`	Eine 2D-Gruppe bestehend aus `X`, `Y`. Die Reihenfolge der translatorischen Achsen in der Konfiguration bestimmt die Translationsreihenfolge.
Axes Convention	mcAxesConv3DCartesianXYZ	`MC.MC_AXES_CONVENTION`	Eine 3D-Gruppe bestehend aus `X`, `Y`, `Z`. Die Reihenfolge der translatorischen Achsen in der Konfiguration bestimmt die Translationsreihenfolge.
Axes Convention	mcAxesConv4DCartesianXYZC	`MC.MC_AXES_CONVENTION`	Eine 4D-Gruppe bestehend aus `X`, `Y`, `Z` und einer rotatorischen Achse um `Z` (`C`). Die Reihenfolge der translatorischen und rotatorischen Achsen in der Konfiguration bestimmt die Translations- und Rotationsreihenfolge.

Achsgruppenparameter des Pick-and-Place-Kernels



Registerkarte: Parameter (Init).

Parameter	Einheit	Typ	Beschreibung
NET Cycle Time Divisor		`UINT`	Verbessert die Genauigkeit aufgrund zeitlicher Diskretisierung.
Time Override Ramp Time	`S`	`LREAL`	Rampenzeit für Override-Änderung von 0 % bis 100 %.

Der Pick-and-Place-Sollwertgenerator wurde speziell für die Anforderungen von Pick-and-Place-Anwendungen entwickelt. Er ist für Bewegungen vorgesehen, bei denen die genaue Bahndynamik nicht so wichtig ist, der Benutzer jedoch möglichst schnell von einem Punkt zu einem anderen gelangen möchte. Daher ist es zulässig, dass der Algorithmus innerhalb der Toleranzkugel gegen Beschränkungen der Bahndynamik verstößt. Gegen Beschränkungen von Achsen wird niemals verstoßen.

6 Räumliche Konfiguration

Die räumliche Konfiguration beschreibt die geometrischen Beziehungen zwischen Referenz-Frames. Diese Beziehungen sind vom Typ Translation (Verschiebung) und Rotation (Drehung).

6.1 Objekt Coordinate Frame

Die Objekte Coordinate Frame können für den hierarchischen Aufbau von geometrischen Translations- und Rotationsbeziehungen verwendet werden. Für eine direkte Interpretation sollte die x-Richtung des letzten Elements in der Hierarchie in die Förderrichtung zeigen.

Parameter	Beschreibung	Einheit
Rotation Convention	Konvention für die Berechnung von Rotationen. Der Standard ist DIN9300 Z”Y’X, wobei Rotation 3 der Parameter für Z”, Rotation 2 der Parameter für Y’ und Rotation 1 der Parameter für X ist.	
Definition Direction	Gibt die Richtung an, in der die Verschiebung programmiert wird (aus der Sicht des Referenzsystems oder des MCS).	
Translation X	Translation in Richtung der x-Achse	mm
Translation Y	Translation in Richtung der y-Achse	mm
Translation Z	Translation in Richtung der z-Achse	mm
Rotation 1	Die Rotationsachse wird durch die Rotation Convention definiert.	°
Rotation 2	Die Rotationsachse wird durch die Rotation Convention definiert.	°
Rotation 3	Die Rotationsachse wird durch die Rotation Convention definiert.	°

6.2 Objekt Conveyor Tracking

Ein Conveyor Tracking-Objekt kann verwendet werden, um eine Achsgruppe mit einem Förderband zu synchronisieren. Es wird einem Coordinate Frame-Objekt als Kindelement angefügt. Während der Coordinate Frame die statische Transformation (Translation und/oder Rotation) zum Förderbandsystem beschreibt, übernimmt das Conveyor Tracking-Objekt den dynamischen Teil des Trackings.

Die Conveyor Tracking-Parameter sind in der folgenden Tabelle aufgelistet. Die dynamischen Parameter sind Standardwerte, die verwendet werden, wenn in der Funktionsbausteininstanz MC_TrackConveyorBelt für den entsprechenden Parameter MC_DEFAULT ausgewählt wird.

Parameter	Beschreibung	Einheit
Velocity	Standardgeschwindigkeit für die Synchronisation.	mm s^{-1}
Acceleration	Standardbeschleunigung für die Synchronisation.	mm s^{-2}
Deceleration	Standardverzögerung für die Synchronisation.	mm s^{-2}
Jerk	Standardruck für die Synchronisation.	mm s^{-3}
Default Tracking Behavior	Conveyor Tracking-Verhalten, nachdem `InSync` erreicht worden ist.	
Synchronization Tolerance Distance	Entfernung zum Tracking-Ziel, ab der eine Synchronisation vorliegt (InSync=TRUE). Die Nutzung des Parameters ist insbesondere dann sinnvoll, wenn das Master-Signal verrauscht ist. Dieser Paramter hat nur Einfluss, wenn die MC Group Coordintated Motion genutzt wird.	mm

6.3 Conveyor Tracking-Verhalten

Das Default Tracking Behavior definiert, welche Art von Störung beim Tracking standardmäßig abgelehnt wird. Eine Störung kann ein unerwarteter Impuls oder eine Taktbewegung des Förderers sein.

mcTrackingBehaviorDynLimited

Die Geschwindigkeitssynchronisation mit dem `ConveyorBelt` wird mit Hilfe der gegebenen Parameter für `Acceleration`, `Deceleration` und `Jerk` aufrechterhalten.
Relevant, wenn Störungen nicht genau bekannt sind oder die Störungsdynamik wesentlich ist.
Dynamische Grenzen werden in den Funktionsbaustein `MC_TrackConveyorBelt` eingegeben. Die Werte aus dem Objekt Conveyor Tracking werden verwendet, wenn `MC_Default` in den Funktionsbaustein eingegeben wird. Wenn der Förderer taktet, wird die Reaktion auf die dynamischen Parameter begrenzt. Wird die Einstellung `DynLimited` verwendet, dann wird die Reaktion mit der Ruckgrenze ausgeglichen. Der Ausgang des Funktionsbausteins `MC_TrackConveyorBelt.InSync` gibt an, wenn eine Synchronisation vorliegt.

InSync

Bei Verwendung der Betriebsart `mcTrackingBehaviorDynLimited` kann der Ausgang `InSync = TRUE` verschwinden, wenn die synchronisierte Position verloren gegangen ist. Der Algorithmus bleibt bei der parametrierten Dynamik und versucht eigenständig, zu der synchronisierten Position zurückzukehren. Wenn die synchronisierte Position erreicht ist, erscheint wieder der Ausgang `InSync = TRUE`.

mcTrackingBehaviorStayInSync

Die Geschwindigkeitssynchronisation mit dem `ConveyorBelt` wird mit unbegrenzten Parametern für Acceleration, Deceleration und Jerk aufrechterhalten.
Wenn der Förderer taktet, wird die Tracking-Reaktion nicht begrenzt. Stattdessen hat die Tracking-Reaktion das Ziel, synchronisiert zu bleiben und dem Förderer unbedingt zu folgen. Der Ausgang des Funktionsbausteins `MC_TrackConveyorBelt.InSync` gibt an, wenn eine Synchronisation vorliegt.

InSync

Bei Verwendung der Betriebsart `mcTrackingBehaviorStayInSync` bleibt das Signal `InSync`, sobald es `TRUE` geworden ist, so lange `TRUE`, wie das Kommando aktiv ist.

6.4 Objekt Node Connector

Wie in einer Baumtopologie werden diese geometrischen Translations- und Rotationsbeziehungen durch Zeiger aufgebaut, die ohne Erzeugung von Zyklen von Referenz-Frame zu Referenz-Frame zeigen. Ein solcher Zeiger, der eine Verbindung von einem Referenz-Frame-Knoten zu einem anderen Referenz-Frame-Knoten herstellt, wird als Node Connector bezeichnet. Durch Konfigurieren eines Node Connectors wird festgelegt, von welchem Ursprungsknoten zu welchem Zielknoten die verbundenen Referenzsysteme verwendet werden.

Ein Node Connector ist ein administratives Objekt. Ein Node Connector-Objekt dient dazu, für eine Koordinatentransformation die Zugänglichkeit vom `PLC`-Teil zu ermöglichen.

Parameter	Beschreibung
Start node	Objekt-ID des Anfangspunkts für die Koordinatentransformation. Der Standard-ID-Wert ist `0` und steht für das `WCS`, das Weltkoordinatensystem.
End node	Objekt-ID des Endpunkts der Koordinatentransformation, z. B. ein Punkt auf dem Förderband.

6.5 Konfigurieren eines Node Connectors

Die Konfiguration für `MC_SetCoordinateTransform` wird am Beispiel einer relativ zum `WCS`- oder `MCS`-Koordinatensystem positionierten Palette veranschaulicht.

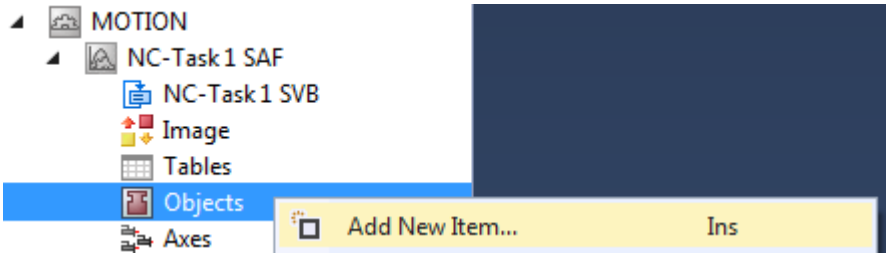
Objekte Node Connector

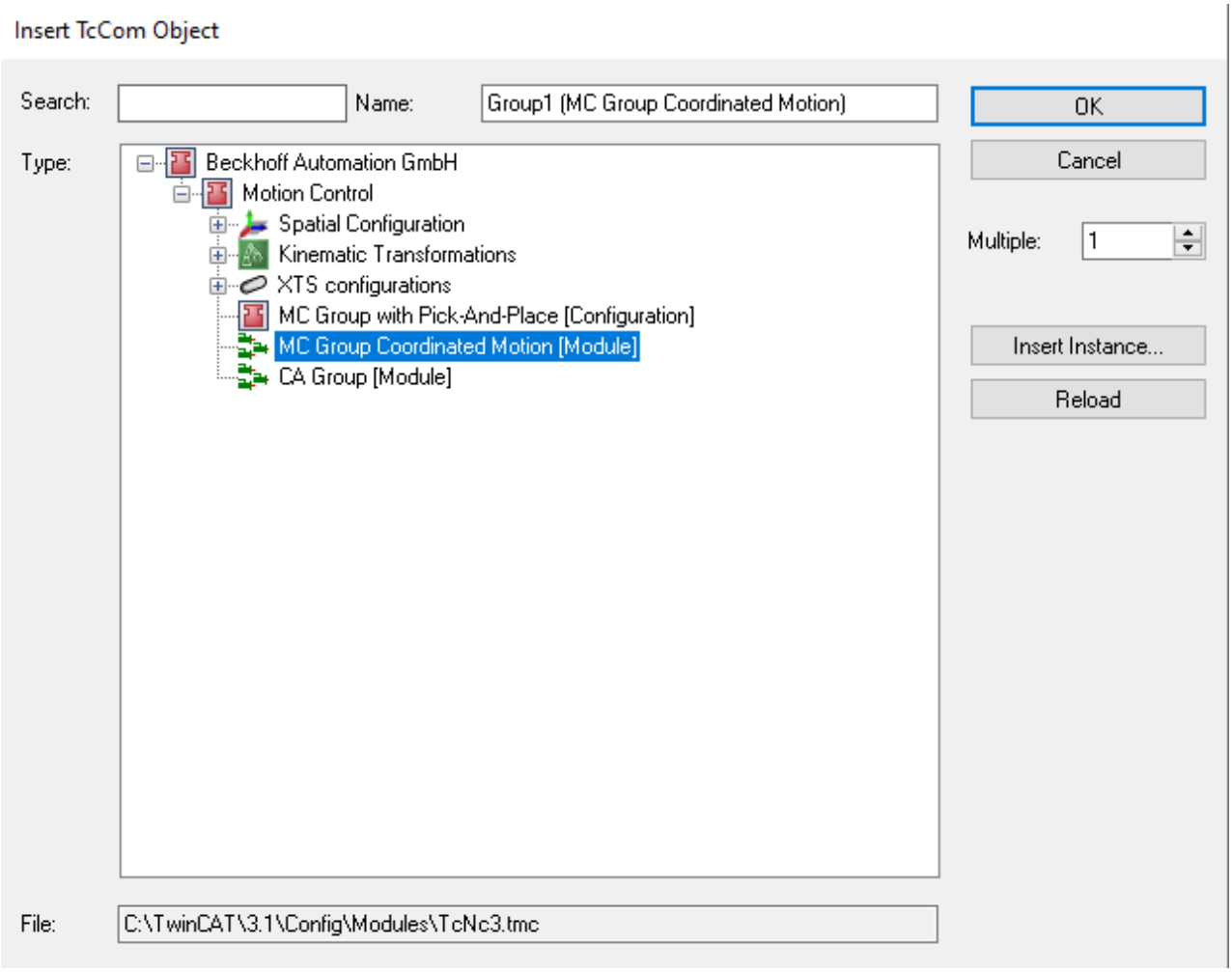
Die Objekte Node Connector werden von MC_SetCoordinateTransform und MC_TrackConveyorBelt verwendet. Anstelle von Coordinate Frames werden die Node Connector-Objekte von der SPS als Vertreter adressiert.

Beispiel

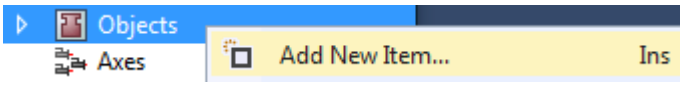
Um eine Koordinatentransformation mit Hilfe von `MC_SetCoordinateTransform` einzuführen:

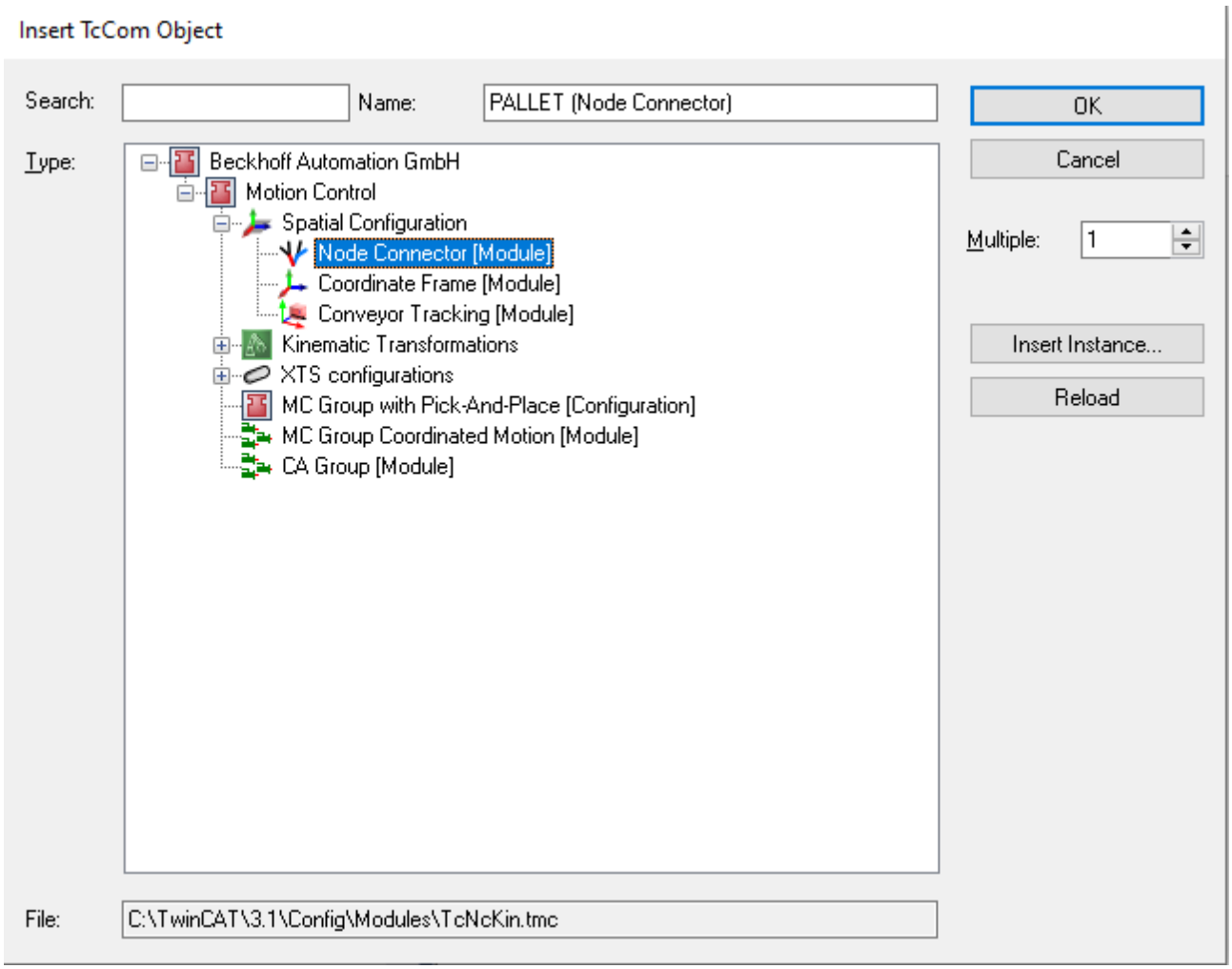
1. Fügen Sie eine `MC`-Gruppe ein.



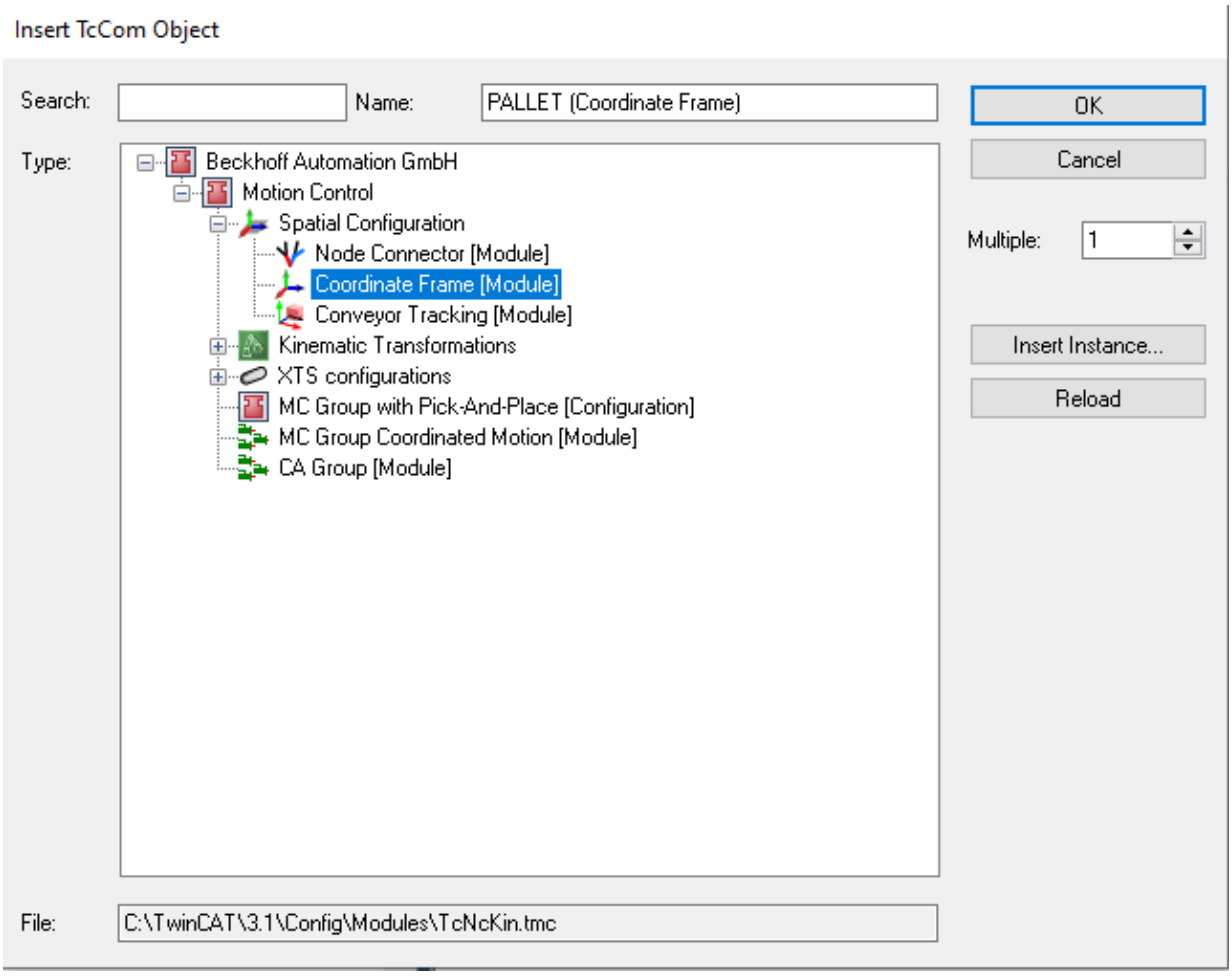


2. Fügen Sie einen Node Connector ein.




Insert TcCom Object
Search:
Name:
PALLET (Node Connector)
OK
Cancel
Type:
Beckhoff Automation GmbH
Motion Control
Spatial Configuration
Node Connector [Module]
Coordinate Frame [Module]
Conveyor Tracking [Module]
Kinematic Transformations
XTS configurations
MC Group with Pick-And-Place [Configuration]
MC Group Coordinated Motion [Module]
CA Group [Module]
Multiple:
1
Insert Instance...
Reload
File:
C:\TwinCAT\3.1\Config\Modules\TcNcKin.tmc

3. Fügen Sie einen Coordinate Frame ein.



4. Geben Sie die entsprechenden Node Connector-Parameter ein – in diesem Beispiel bezieht sich der Endknoten auf die Objekt-ID der Palette.

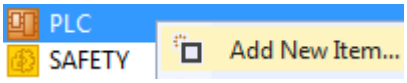
Name	Comment
Configuration	
Start node	First spatial node. Coordinates are interpreted in respect to this reference node. 0x0 represents the world coordinate system (WCS).
End node	Last spatial node. This node is moved in respect to the start node. 0x0 represents the world coordinate system (WCS).

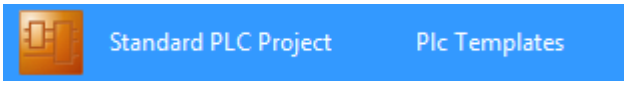
	Name	Value	CS	Unit	Type
-	Configuration				
	Start node	00000000	☐		OTCID
	End node	01010040	☐	PALLET (Coordinate Frame)	OTCID

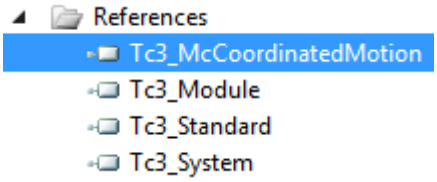
5. Geben Sie die entsprechenden Coordinate Frame-Parameter ein.

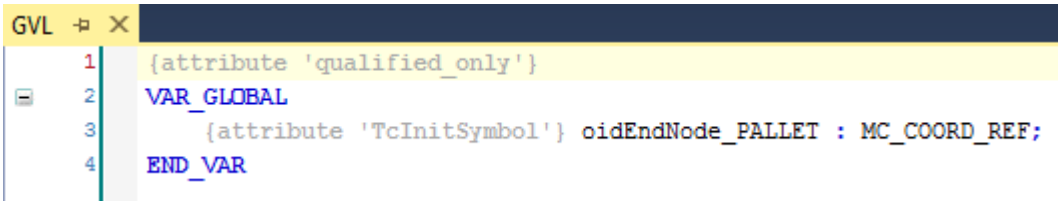
Name	Value	CS	Unit	Type
- Configuration				
Rotation convention	Rotation_Z3Y2X1_DIN9300			MC.CoordInterpretation_SO3
Definition direction	toReference			MC.ReferenceDefDir
- Kinematic				
Translation X	0.0		mm	LREAL
Translation Y	150.0		mm	LREAL
Translation Z	0.0		mm	LREAL
Rotation 1	0.0		°	LREAL
Rotation 2	0.0		°	LREAL
Rotation 3	0.0		°	LREAL

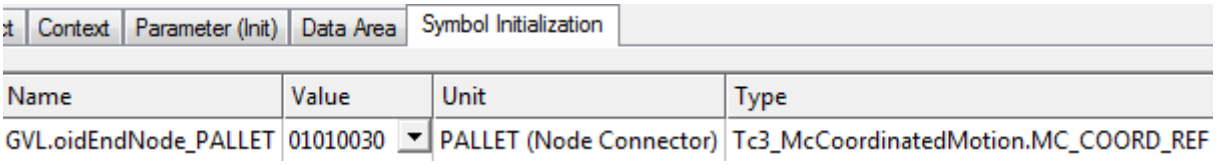
6. Verbinden Sie den eingefügten Node Connector mit der `PLC`.











⇨ Abschließend können Sie den Funktionsbaustein `MC_SetCoordinateTransform` einfügen.

```
MAIN
1  PROGRAM MAIN
2  VAR
3      AxisGroup : AXES_GROUP_REF;
4      MC_SetCoordinateTransform_0: MC_SetCoordinateTransform;
5  END_VAR

1  MC_SetCoordinateTransform_0(
2      AxesGroup:= AxisGroup,
3      Execute:= ,
4      CoordTransform:= GVL.oidEndNode_PALLET,
5      Done=> ,
6      Busy=> ,
7      Active=> ,
8      CommandAborted=> ,
9      Error=> ,
10     ErrorId=> );
```

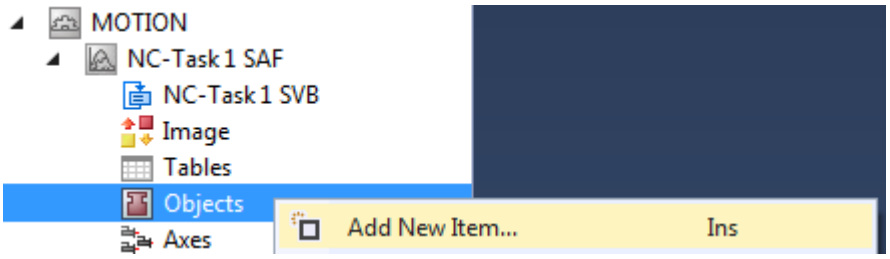
Die Achsgruppe AxisGroup ist mit den Funktionsbausteinen Pick-and-Place verbunden.

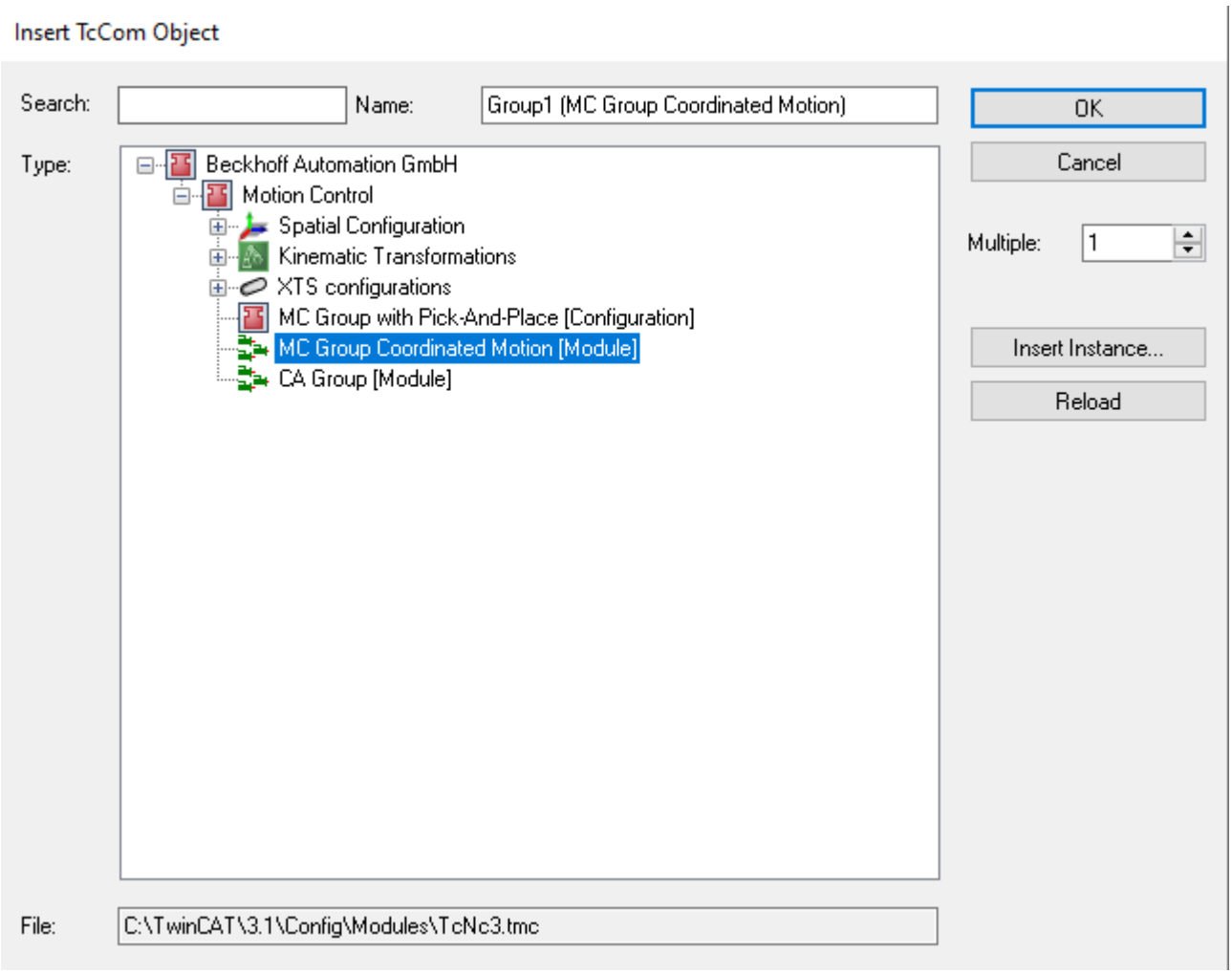
Für Achsbewegungen muss ein Move-Kommando programmiert werden, z. B. MovePath.

6.6 Konfigurieren für MC_TrackConveyorBelt

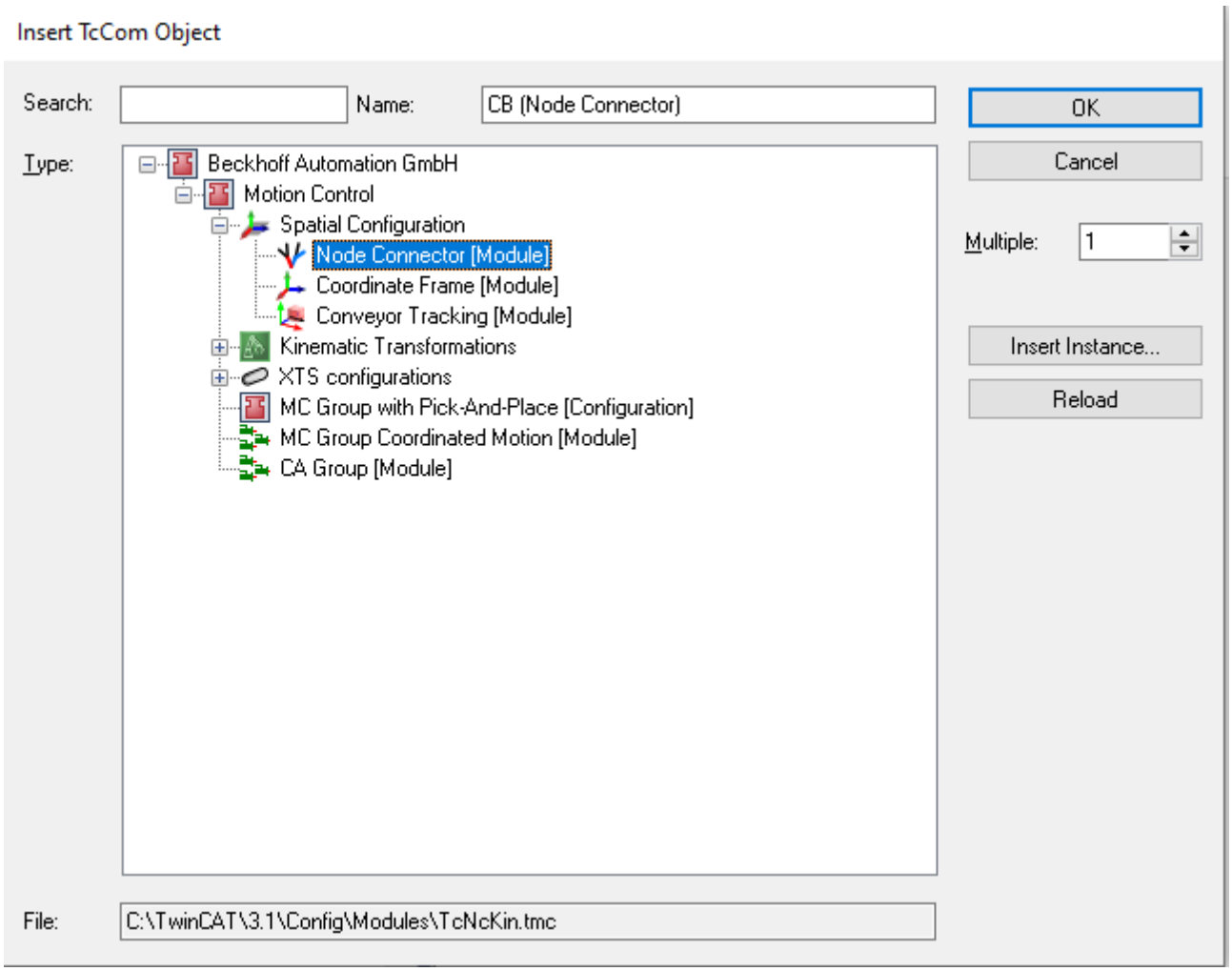
Um ein Förderband mit Hilfe von `MC_TrackConveyorBelt` zu verfolgen:

1. Fügen Sie eine `MC`-Gruppe ein.

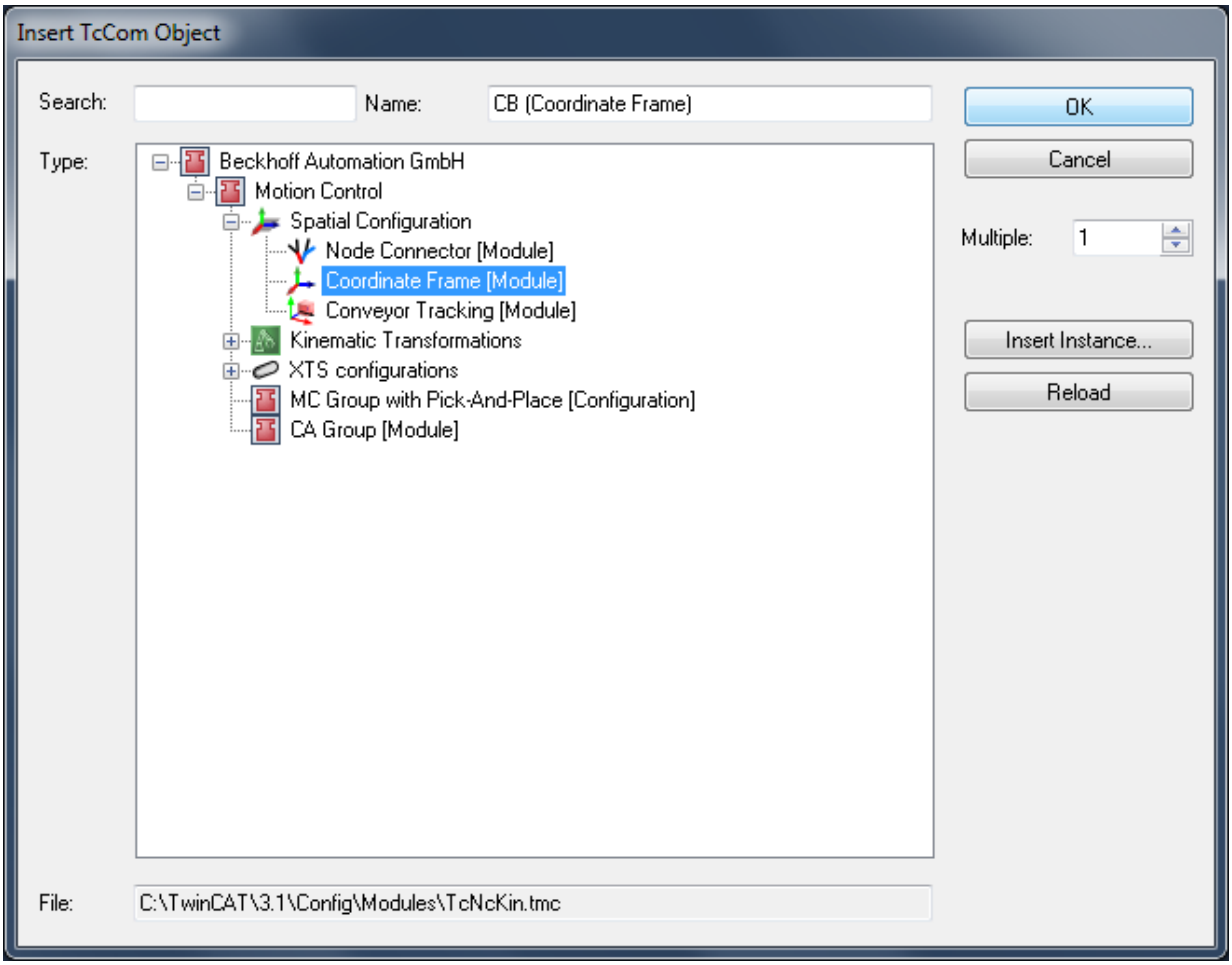




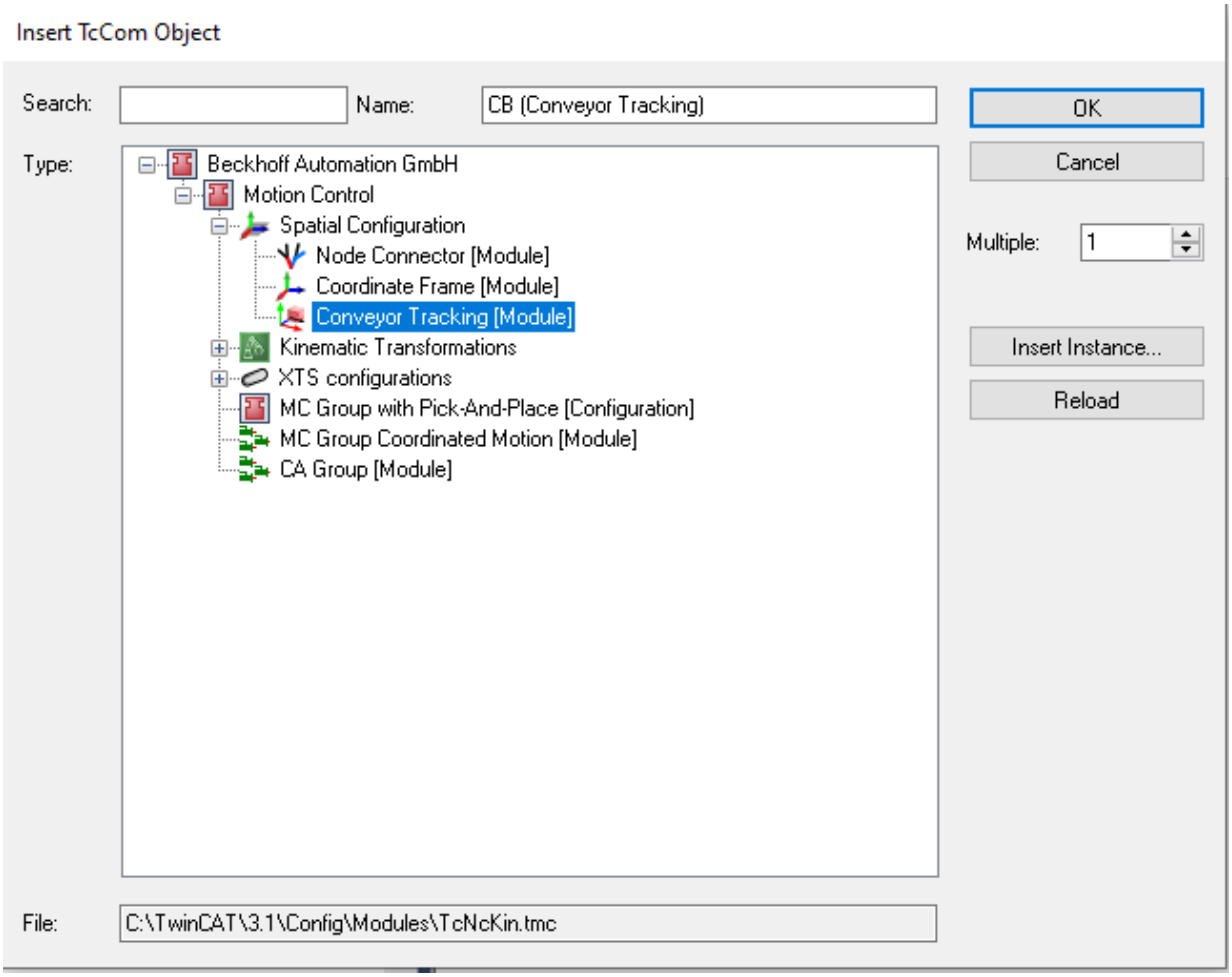
2. Fügen Sie einen Node Connector ein.

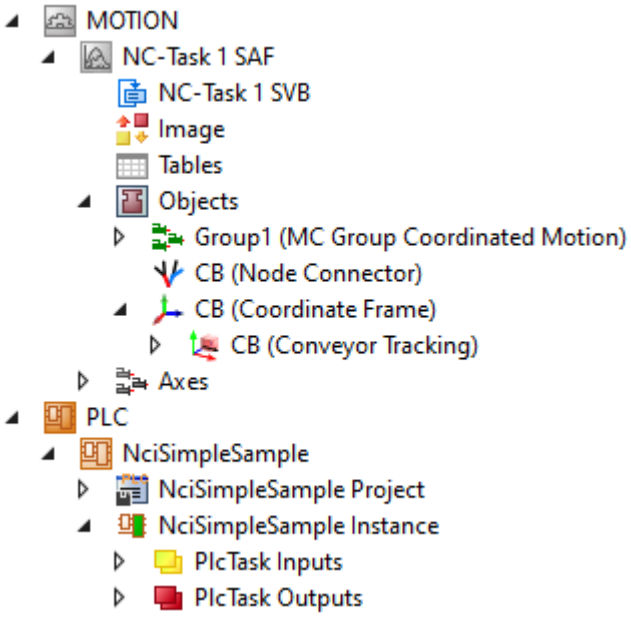

Insert TcCom Object
Search:
Name:
CB (Node Connector)
OK
Cancel
Type:
Beckhoff Automation GmbH
Motion Control
Spatial Configuration
Node Connector [Module]
Coordinate Frame [Module]
Conveyor Tracking [Module]
Kinematic Transformations
XTS configurations
MC Group with Pick-And-Place [Configuration]
MC Group Coordinated Motion [Module]
CA Group [Module]
Multiple:
1
Insert Instance...
Reload
File:
C:\TwinCAT\3.1\Config\Modules\TcNcKin.tmc

3. Fügen Sie einen Coordinate Frame ein.



4. Fügen Sie Conveyor Tracking ein. Als Erstes wurde ein Coordinate Frame erstellt. Als Zweites muss das Objekt Conveyor Tracking dem zuvor erstellten Coordinate Frame als Kindelement hinzugefügt werden.





5. Geben Sie die entsprechenden Node Connector-Parameter ein – der Endknoten bezieht sich auf die Objekt-ID des Conveyor Trackings.

Name	Comment
Configuration	
Start node	First spatial node. Coordinates are interpreted in respect to this reference node. 0x0 represents the world coordinate system (WCS).
End node	Last spatial node. This node is moved in respect to the start node. 0x0 represents the world coordinate system (WCS).

	Name	Value	CS	Unit	Type	PTCID
-	Configuration					
	Start node	00000000			OTCID	0x05010108
	End node	01010050		CB (Conveyor Tracking)	OTCID	0x05010107

6. Geben Sie die entsprechenden Coordinate Frame-Parameter ein.

Name	Comment
Configuration	
Rotation convention	Set the interpretation of the rotation angles.
Definition direction	Set the definition direction.
Kinematic	
Translation X	Translation in x-direction
Translation Y	Translation in y-direction
Translation Z	Translation in z-direction
Rotation 1	Rotation angle 1, interpretation set by rotation convention
Rotation 2	Rotation angle 2, interpretation set by rotation convention
Rotation 3	Rotation angle 3, interpretation set by rotation convention

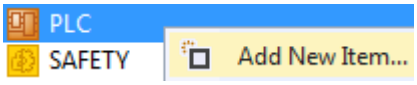
	Name	Value	CS	Unit	Type
-	Configuration				
	Rotation convention	Rotation_Z3Y2X1_DIN9300			MC.CoordInterpretation_SO3
	Definition direction	toReference			MC.ReferenceDefDir
-	Kinematic				
	Translation X	0.0		mm	LREAL
	Translation Y	150.0		mm	LREAL
	Translation Z	0.0		mm	LREAL
	Rotation 1	0.0		°	LREAL
	Rotation 2	0.0		°	LREAL
	Rotation 3	0.0		°	LREAL

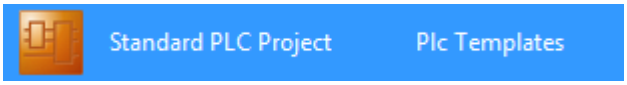
7. Geben Sie die entsprechenden Conveyor Tracking-Parameter ein.

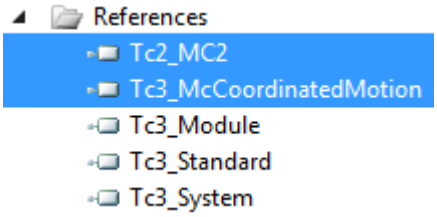
Name	Value	CS	Unit	Type
Velocity	2000.0	☐		LREAL
Acceleration	1500.0	☐		LREAL
Deceleration	1500.0	☐		LREAL
Jerk	25000.0	☐		LREAL
Default Tracking Behavior	mcTrackingBehaviorDynLimited	☐		MC.MC_DEFAULT_TRACKING_BEHAVIOR
Synchronization Tolerance Distance	0.0	☐	mm	LREAL

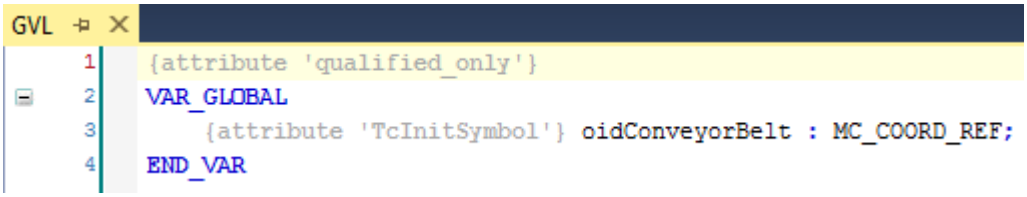
Das Default Tracking Behavior ist das verwendete Standard-Tracking-Verhalten, nachdem `InSync` erstmals erreicht worden ist.

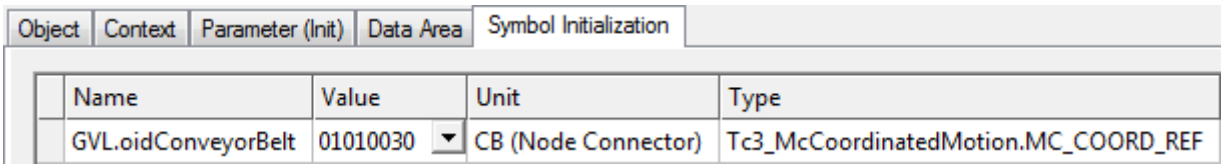
8. Verbinden Sie den Node Connector mit der `PLC`.











⇨ Abschließend können Sie den Funktionsbaustein `MC_TrackConveyorBelt` einfügen.

```
PROGRAM MAIN
VAR
    AxisGroup : AXES_GROUP_REF;
    ConveyorBelt : AXIS_REF;
    MC_TrackConveyorBelt_0: MC_TrackConveyorBelt;
    MasterRefPos: MC_LREAL;
    InitialObjectPos: ARRAY[1..4] OF MC_LREAL;
END_VAR
```

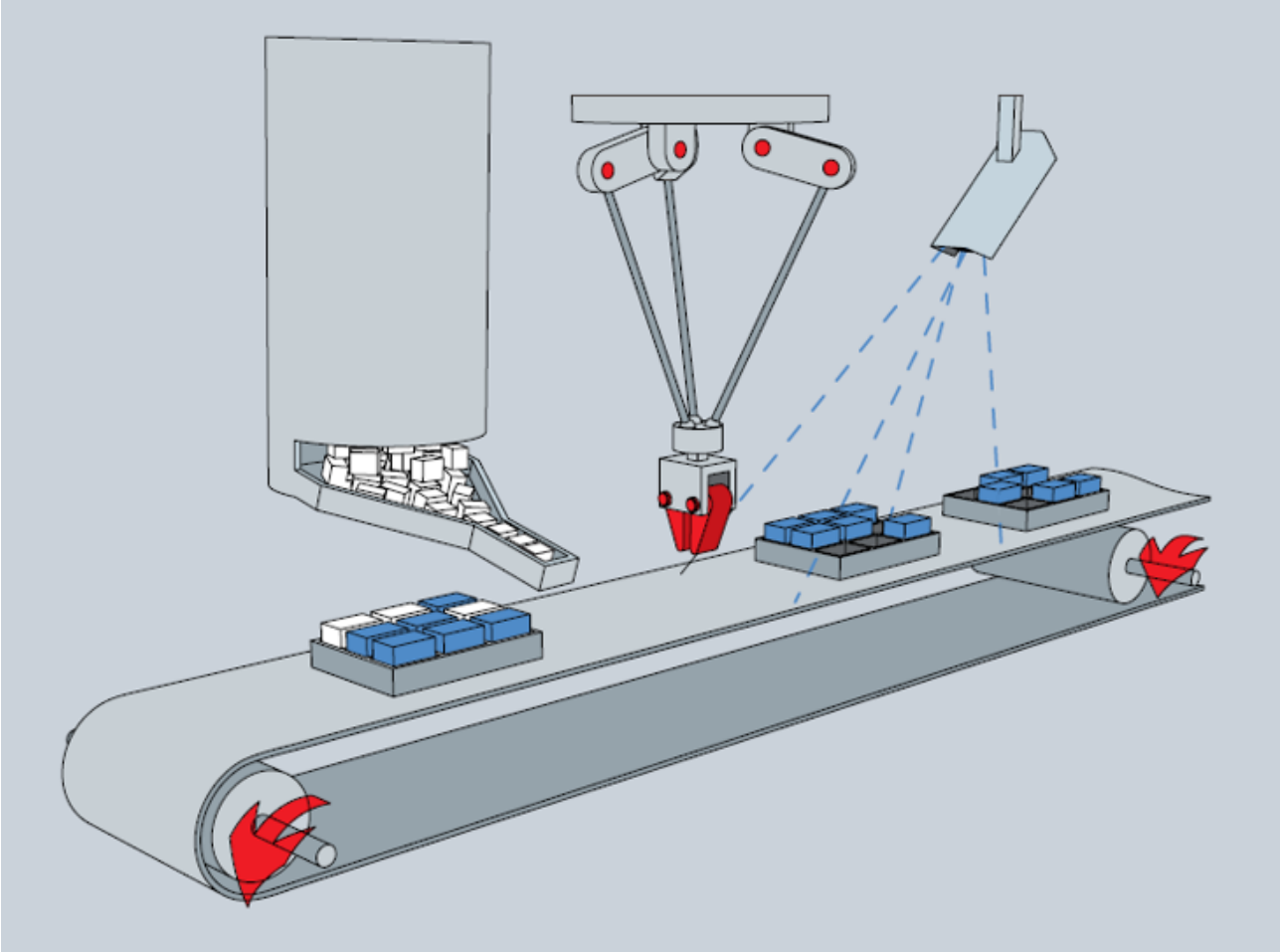
```
MC_TrackConveyorBelt_0(
    AxesGroup:= AxisGroup,
    ConveyorBelt:= ConveyorBelt,
    Execute:= ,
    CoordTransform:= GVL.oidConveyorBelt,
    InitialObjectPos:= ADR(InitialObjectPos),
    InitialObjectPosCount:= SIZEOF(InitialObjectPos)/SIZEOF(InitialObjectPos[1]),
    MasterRefPos:= MasterRefPos,
    Velocity:= MC_DEFAULT,
    Acceleration:= MC_DEFAULT,
    Deceleration:= MC_DEFAULT,
    Jerk:= MC_DEFAULT,
    InSync=> ,
    Busy=> ,
    Active=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorId=> );
```

6.7 Hintergrundinformationen

Koordinatensysteme – Beziehungen

- **WCS**
  Weltkoordinatensystem.
- **MCS**
  Maschinenkoordinatensystem.
- **UCS**
  Benutzerkoordinatensystem.
- **PCS**
  Programmiertes Koordinatensystem. Werkstück.

Pick-and-Place: Vom Lager zum Träger

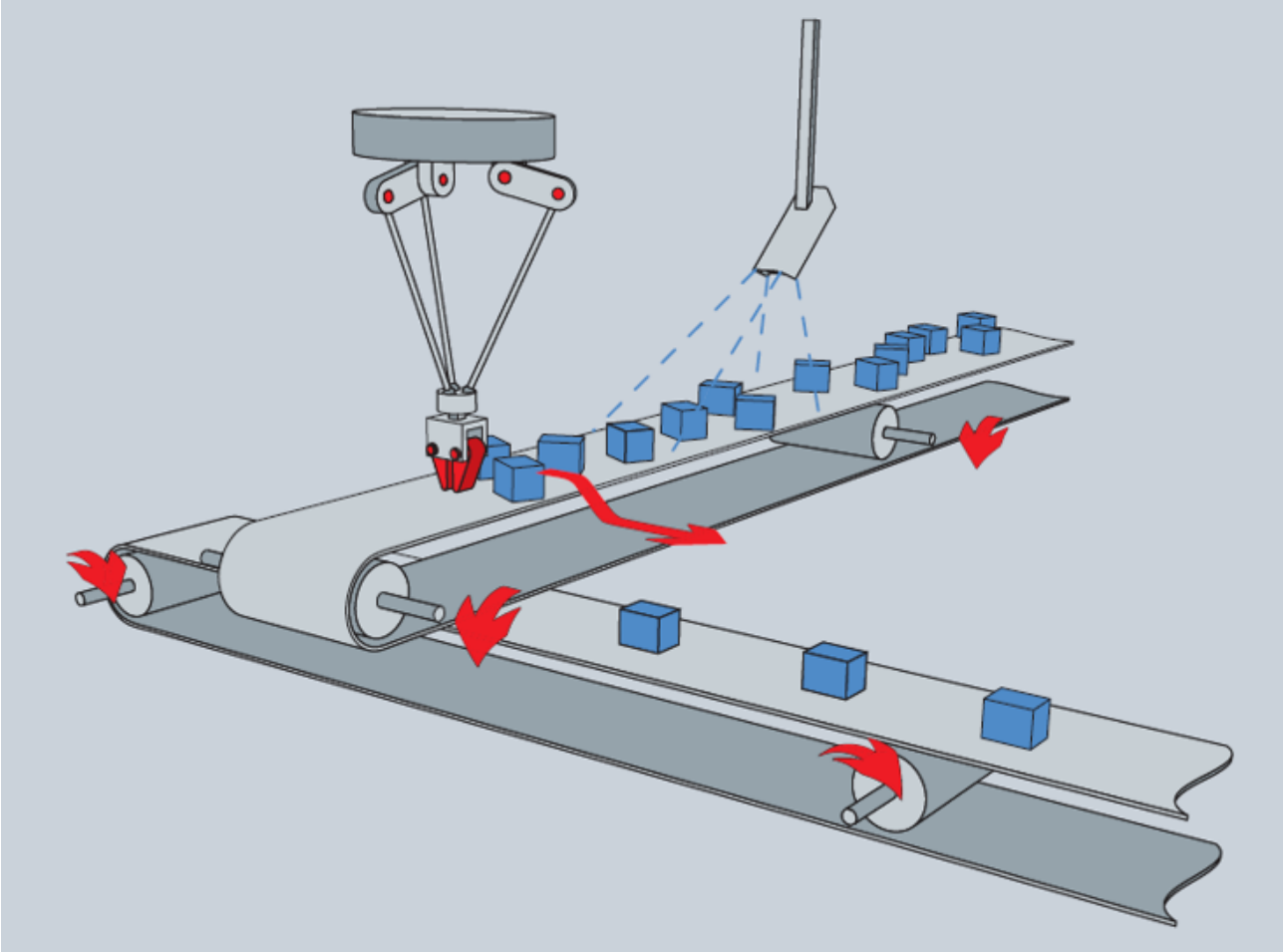


Die Werkstücke müssen vom Lager in die auf dem Förderband laufenden freien Trägerplätze gelegt werden.

Dabei

- wird der Lagerplatz im `WCS` definiert,
- befindet sich der Roboter irgendwo im `WCS`,
- kann der Roboter in seinem `MCS` gesteuert werden,
- befindet sich das Förderband irgendwo im `WCS`,
- kann sich ein Träger auf dem Förderband im `UCS` befinden,
- kann sich ein Werkstück in seinem Träger im `PCS` befinden.

Pick-and-Place: Von Förderband zu Förderband



Die Werkstücke müssen vom oberen Förderband abgenommen und auf das untere Förderband gelegt werden.

Dabei

- befindet sich der Roboter irgendwo im `WCS`,
- kann der Roboter in seinem `MCS` gesteuert werden,
- befindet sich jedes Förderband irgendwo im `WCS`,
- können sich Werkstücke auf dem Förderband in einem `UCS` befinden.

7 SPS-Bibliotheken

7.1 Tc3_McCoordinatedMotion

Die Tc3_McCoordinatedMotion-Bibliothek wird für TF5410 TwinCAT 3 Motion Collision Avoidance und auch für TF5420 TwinCAT 3 Motion Pick-and-Place verwendet.

Übersicht

Funktionsbaustein	Beschreibung	TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
			MC Group with Pick-and-Place	MC Group Coordinated Motion
Administrativ				
MC_AddAxisToGroup [▶ 40]	Fügt eine Achsgruppe hinzu.	✔	✖ (✔ bis v3.2)	✔
MC_GroupDisable [▶ 43]	Deaktiviert eine Achsgruppe.	✔	✖ (✔ bis v3.2)	✔
MC_GroupEnable [▶ 44]	Aktiviert eine Achsgruppe.	✔	✖ (✔ bis v3.2)	✔
MC_GroupReadError [▶ 45]	Liest die Fehler-ID einer Gruppe aus.	✔	✖ (✔ bis v3.2)	✔
MC_GroupReadStatus [▶ 47]	Liest den Gruppenstatus aus.	✔	✖ (✔ bis v3.2)	✔
MC_GroupReset [▶ 49]	Setzt eine Gruppe zurück.	✔	✖ (✔ bis v3.2)	✔
MC_GroupSetOverride [▶ 50]	Setzt den Override einer Gruppe und gibt den Override-Istwert zurück.	✖	✖ (✔ bis v3.2)	✔
MC_RemoveAxisFromGroup [▶ 51]	Entfernt eine Achse aus einer Gruppe.	✔	✖ (✔ bis v3.2)	✔
MC_SetCoordinateTransform [▶ 53]	Aktiviert ein Bezugssystem.	✖	✖ (✔ bis v3.2)	✔
MC_SetCoordinateTransformPreparation [▶ 55]	Fügt einen Wechsel des Bezugssystems in die Liste von Bewegungssegmenten ein.	✖	✖	✔

Funktionsbaustein	Beschreibung	TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
			MC Group with Pick-and-Place	MC Group Coordinated Motion
MC_TrackConveyorBelt [▶ 56]	Hilft bei der Synchronisation der Geschwindigkeit mit einem Objekt, das sich auf einer Geraden durch den Raum bewegt.	✗	✗ (✓ bis v3.2)	✓
MC_UngroupAllAxes [▶ 59]	Deaktiviert eine Gruppe und entfernt alle Achsen.	✓	✗ (✓ bis v3.2)	✓
UDINT_TO_IDENTINGROUP [▶ 60]	Wandelt einen ganzzahligen Wert in IDENT_IN_GROUP_REF um, so dass Achsen ohne besondere Interpretation einer Gruppe hinzugefügt werden können.	✓	✗	✓
Motion				
MC_GroupHalt [▶ 61]	Stoppt eine Gruppe, ohne sie für weitere Bewegungskommandos zu sperren.	✓	✗	✓
MC_GroupStop [▶ 63]	Stoppt eine Gruppe und sperrt sie für weitere Bewegungskommandos.	✓	✗ (✓ bis v3.2)	✓
MC_MoveLinearAbsolutePreparation [▶ 65]	Fügt einer Liste von Bewegungssegmenten eine absolute Linearbewegung hinzu.	✗	✗ (✓ bis v3.2)	✓
MC_MoveCircularAbsolutePreparation [▶ 67]	Fügt einer Liste von Bewegungssegmenten eine absolute Kreisbewegung hinzu.	✗	✗ (✓ bis v3.2)	✓
MC_MovePath [▶ 71]	Führt eine Liste von Bewegungssegmenten aus.	✗	✗ (✓ bis v3.2)	✓
MC_BlockerPreparation [▶ 73]	Hängt einen blockierenden Job an die Liste der Segmente in der Struktur PathData an.	✗	✗	✓
MC_ReleaseBlocker [▶ 74]	Löst einen blockierenden Job auf, der die weitere Ausführung der Bahn blockiert.	✗	✗	✓
MC_GroupReadBlockerStatus [▶ 76]	Liest den aktuellen Blocker-Status.	✗	✗	✓
MC_DwellTimePreparation [▶ 77]	Hängt einen Stillstandsjob mit einer definierten Zeit an die Liste der Segmente in der Struktur PathData an.	✗	✗	✓

Strukturen und Aufzählungen

Funktionsbaustein	Beschreibung	TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
			MC Group with Pick-and-Place	MC Group Coordinated Motion
IDENT_IN_GROUP_REF [▶ 78]	Definiert, wie eine Achse in einer Gruppe interpretiert wird.	✗	✗ (✓ bis v3.2)	✓
MC_CIRC_MODE [▶ 79]	Der Kreismodus definiert, welche Definition zum Programmieren eines Kreises verwendet wird.	✗	✗ (✓ bis v3.2)	✓
MC_CIRC_PATHCHOICE [▶ 83]	Der Datentyp definiert die Drehrichtung eines Kreises.	✗	✗ (✓ bis v3.2)	✓
MC_PATH_DATA_REF [▶ 84]	Stellt die bei MC_MovePath [▶ 71] auszuführende Bahn dar.	✗	✗ (✓ bis v3.2)	✓
ClearPath [▶ 85]	Setzt die durch MC_PATH_DATA_REF [▶ 84] dargestellte Bahn zurück.	✗	✗ (✓ bis v3.2)	✓
MC_TRANSITION_MODE [▶ 85]	Charakterisiert, wie ein Segmentübergang ausgeführt wird.	✗	✗ (✓ bis v3.2)	✓
MC_COORD_REF [▶ 88]	Objekt-ID eines Koordinatensystems.	✗	✗ (✓ bis v3.2)	✓

7.1.1 Funktionsbausteine

7.1.1.1 Administrativ

7.1.1.1.1 MC_AddAxisToGroup

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘ (✔ bis v3.2 einschließlich)	✔

Der Funktionsbaustein MC_AddAxisToGroup fügt einer Gruppe eine Achse hinzu.

Ab V3.1.10.1 können einer **CA-Gruppe** im Gruppenzustand GroupMoving stehende Achsen hinzugefügt und aus ihr entfernt werden. Wenn einer Gruppe eine fahrende Achse hinzugefügt wird, wird das Kommando mit einer Fehlermeldung abgelehnt (eine Änderung des Gruppenzustands mit einer fahrenden Achse wird ebenfalls abgelehnt).

Einer **MC-Gruppe** können nur Achsen im Zustand GroupDisabled bzw. GroupErrorDisabled hinzugefügt werden.

Eingänge

```
VAR_INPUT
    Execute                 : BOOL;
    IdentInGroup            : IDENT_IN_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
IdentInGroup	IDENT_IN_GROUP_REF	Definiert die Interpretation der Achse, die der Gruppe hinzugefügt werden soll. Für mehrdimensionale Bewegungen kann dies die kartesische Interpretation sein. Die globalen Variablen [▶ 78] (z. B. MCS_X) müssen verwendet werden. Für Collision Avoidance muss die Funktion UDINT_TO_IDENTINGROUP [▶ 60] verwendet werden. **Hinweis Die Verwendung ganzzahliger Werte für den Eingang IdentInGroup wird NICHT unterstützt und kann zu Inkompatibilität mit künftigen Releases führen. Bei Verwendung ganzzahliger Werte kann das Projekt unter Umständen nicht mehr gebaut werden. Es wird empfohlen, glo-bale Variablen (z. B. MCS_X) oder die Konvertierungsfunktion UDINT_TO_IDENTINGROUP zu verwenden.**

Ein-/Ausgänge

```
VAR_IN_OUT
    AxesGroup               : AXES_GROUP_REF;
    Axis                    : AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle [▶ 99]).
Axis	AXIS_REF	Referenz auf eine Achse (siehe AXIS_REF).

Ausgänge

```
VAR_OUTPUT
    Done                    : BOOL;
    Busy                    : BOOL;
```

```
    Error                   : BOOL;
    ErrorId                 : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Execute gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn Busy wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted (falls vorhanden) oder Error gesetzt.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Beispiel für TwinCAT 3 Motion Pick-and-Place

Mehrdimensionale Bewegungen

Mehrdimensionale Bewegungen werden nur bei Verwendung von TF5420 eingesetzt.

```
VAR_GLOBAL CONSTANT
cAxesCount            : UINT := 4;
END_VAR
VAR
    stGroupRef            : AXES_GROUP_REF; // link to MC Group
    stAxis                : ARRAY[1..cAxesCount] OF AXIS_REF;
    fbAddAxis             : ARRAY[1..cAxesCount] OF MC_AddAxisToGroup;
    i                     : UINT;
END_VAR
```

```
fbAddAxis[1].IdentInGroup := MCS_X; //X-Axis
fbAddAxis[2].IdentInGroup := MCS_Y; //Y-Axis
fbAddAxis[3].IdentInGroup := MCS_Z; //Z-Axis
fbAddAxis[4].IdentInGroup := MCS_C1;//1st rotation is C-rotation (around Z-Axis)

FOR i:=1 TO cAxesCount DO
    fbAddAxis[i](
    AxesGroup:=stGroupRef,
    Axis := stAxis[i],
    Execute := TRUE);
END_FOR
```

Beispiel für TF5410 TwinCAT 3 Motion Collision Avoidance

PTP mit Collision Avoidance

PTP mit Collision Avoidance wird nur bei Verwendung von TF5410 eingesetzt.

```
VAR_GLOBAL CONSTANT
    cAxesCount            : UDINT:=10;
END_VAR
VAR
    stGroupRef            : AXES_GROUP_REF; // link to CA Group
    stAxis                : ARRAY[1..cAxesCount] OF AXIS_REF;
    fbAddAxis             : ARRAY[1..cAxesCount] OF MC_AddAxisToGroup;
    i                     : UDINT;
END_VAR
```

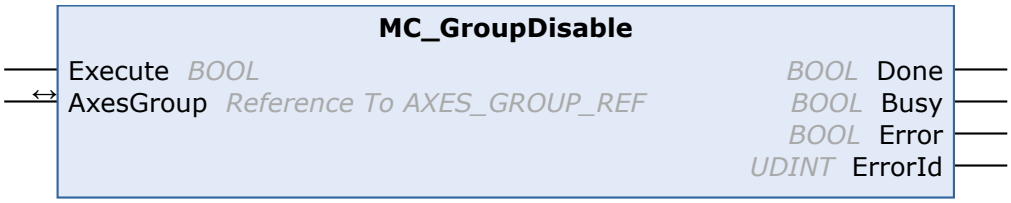
```
FOR i:=1 TO cAxesCount DO
    fbAddAxis[i](
        AxesGroup:=stGroupRef,
        Axis := stAxis[i],
```

```
        IdentInGroup := UDINT_TO_IDENTINGROUP(i),
        Execute := TRUE);
END_FOR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.2 MC_GroupDisable



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘ (✔ bis v3.2 einschließlich)	✔

Der Funktionsbaustein MC_GroupDisable deaktiviert die Gruppe. Nach erfolgreicher Ausführung ändert die Gruppe ihren Zustand in GroupDisabled (siehe Zustandsdiagramme).

Ab TF5400 V3.3.72 sind im Zustand GroupDisabled PTP-Kommandos der Tc2_MC2-Bibliothek (z. B. MC_MoveAbsolute) auf Achsen der MC Group Coordinated Motion erlaubt.

HINWEIS

Die Deaktivierung einer Gruppe in Bewegung hat einen sofortigen Stopp zur Folge.

Beim plötzlichen Stoppen von Achsen werden wahrscheinlich die zulässigen Verzögerungsgrenzen überschritten. Je nach Antriebshardware könnte dies zu Stromspitzen und Laufzeitfehlern führen. Verwenden Sie vor der Ausführung von MC_GroupDisable MC_GroupHalt [▶ 61] oder MC_GroupStop [▶ 63], um diese Situation zu vermeiden.

Eingänge

```
VAR_INPUT
    Execute   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.

Ein-/Ausgänge

```
VAR_IN_OUT
   AxesGroup              : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle).

Ausgänge

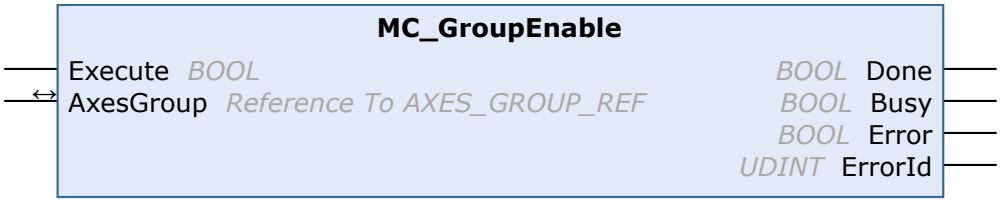
```
VAR_OUTPUT
    Done                    : BOOL;
    Busy                    : BOOL;
    Error                   : BOOL;
    ErrorId                 : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Execute gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn Busy wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted (falls vorhanden) oder Error gesetzt.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.3 MC_GroupEnable



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘ (✔ bis v3.2 einschließlich)	✔

Der Funktionsbaustein MC_GroupEnable aktiviert die Gruppe. Wenn er erfolgreich ist und alle Achsen bereit sind, befindet sich die Gruppe anschließend im Gruppenzustand GroupStandby (siehe Zustandsdiagramme).

Eine **MC-Gruppe** kann nur aktiviert werden, wenn der Gruppe zuvor alle Achsen hinzugefügt wurden.

Eingänge

```
VAR_INPUT
    Execute     : BOOL;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.

Ein-/Ausgänge

```
VAR_IN_OUT
    AxesGroup   : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklisches Gruppen-Interface [▶ 99])

Ausgänge

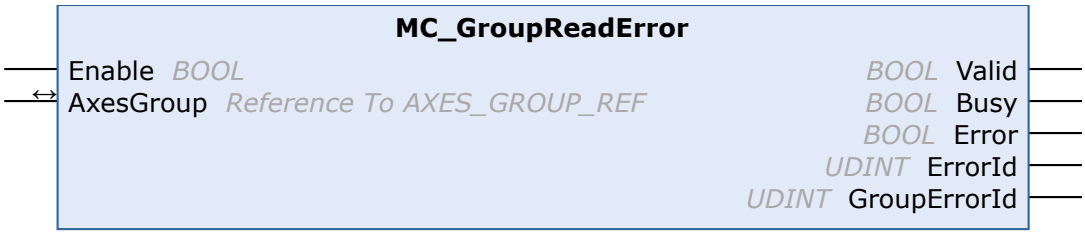
```
VAR_OUTPUT
    Done                    : BOOL;
    Busy                    : BOOL;
    Error                   : BOOL;
    ErrorId                 : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Execute gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn Busy wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted (falls vorhanden) oder Error gesetzt.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.4 MC_GroupReadError



Der Funktionsbaustein MC_GroupReadError gibt den Fehlercode der Gruppe zurück. Er gibt keine Fehler bei Funktionsbausteinen zurück (z. B. ungültige Parametrierung).

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘ (✔ bis v3.2 einschließlich)	✔

Eingänge

```
VAR_INPUT
    Enable      : BOOL;
END_VAR
```

Name	Typ	Beschreibung
Enable	BOOL	Das Kommando wird so lange ausgeführt, wie Enable aktiv ist.

Ein-/Ausgänge

```
VAR_IN_OUT
   AxesGroup              : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle).

Ausgänge

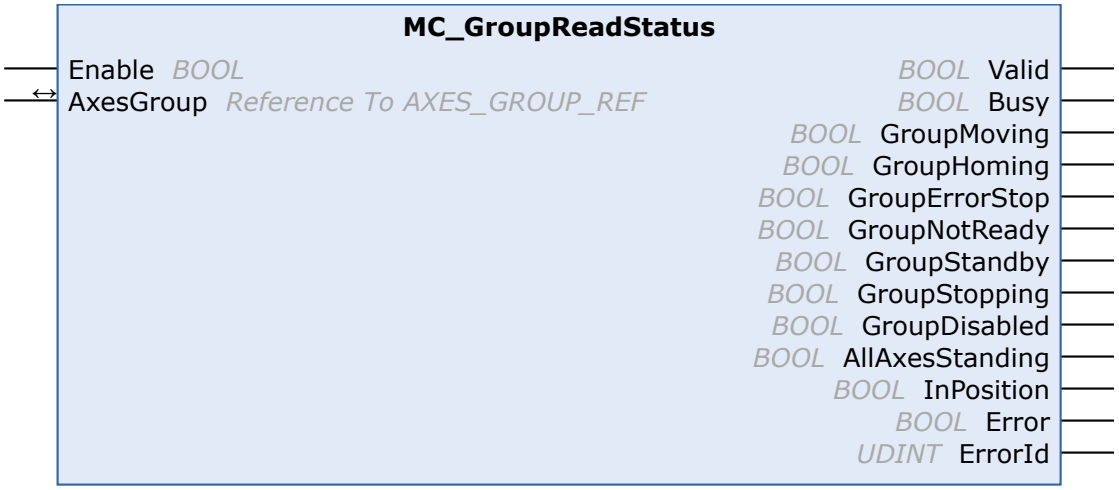
```
VAR_OUTPUT
    Valid        : BOOL;
    Busy         : BOOL;
    Error        : BOOL;
    ErrorId      : UDINT;
    GroupErrorId : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Valid	BOOL	Dieser Ausgang gibt an, dass andere Ausgangswerte bei diesem Funktionsbaustein gültig sind.
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Enable gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.
GroupErrorId	UDINT	Gibt die Fehler-ID der Gruppe zurück (siehe NC-Fehlerdokumentation).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.5 MC_GroupReadStatus



Der Funktionsbaustein MC_GroupReadStatus liest den Zustand einer Achsgruppe aus (siehe Zustandsdiagramme).

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	**MC Group Coordinated Motion**
✔	✘ (✔ bis v3.2 einschließlich)	✔

Eingänge

```
VAR_INPUT
    Enable       : BOOL;
END_VAR
```

Name	Typ	Beschreibung
Enable	BOOL	Das Kommando wird so lange ausgeführt, wie Enable aktiv ist.

Ein-/Ausgänge

```
VAR_IN_OUT
   AxesGroup              : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle).

Ausgänge

```
VAR_OUTPUT
    Valid                  : BOOL;
    Busy                   : BOOL;
    GroupMoving            : BOOL;
    GroupHoming            : BOOL;
    GroupErrorStop         : BOOL;
    GroupNotReady          : BOOL;
    GroupStandby           : BOOL;
    GroupStopping          : BOOL;
    GroupDisabled          : BOOL;
    AllAxesStanding        : BOOL;
    ConstantVelocity       : BOOL; // hidden
    Accelerating           : BOOL; // hidden
    Decelerating           : BOOL; // hidden
```

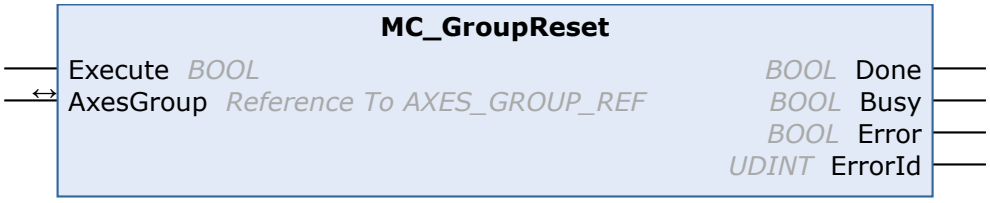
```
    InPosition              : BOOL;
    Error                   : BOOL;
    ErrorId                 : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Valid	BOOL	Dieser Ausgang gibt an, dass andere Ausgangswerte bei diesem Funktionsbaustein gültig sind.
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Enable gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.
GroupMoving	BOOL	Die Gruppe befindet sich im Zustand GroupMoving (siehe Zustandsdiagramme).
GroupHoming	BOOL	Die Gruppe befindet sich im Zustand GroupHoming (siehe Zustandsdiagramme).
GroupErrorStop	BOOL	Die Gruppe befindet sich im Zustand GroupErrorStop (siehe Zustandsdiagramme).
GroupNotReady	BOOL	Die Gruppe befindet sich im Zustand GroupNotReady (siehe Zustandsdiagramme).
GroupStandby	BOOL	Die Gruppe befindet sich im Zustand GroupStandby (siehe Zustandsdiagramme).
GroupStopping	BOOL	Die Gruppe befindet sich im Zustand GroupStopping (siehe Zustandsdiagramme).
GroupDisabled	BOOL	Die Gruppe befindet sich im Zustand GroupDisabled (siehe Zustandsdiagramme).
AllAxesStanding	BOOL	Alle Achsen der Gruppe bewegen sich physikalisch nicht (Geschwindigkeit = 0 und Beschleunigung = 0), unabhängig davon, ob ein Fahrauftrag existiert oder nicht.
ConstantVelocity	BOOL	Nicht unterstützt. Ab TF5400 3.2.27 nicht sichtbar.
Accelerating	BOOL	Nicht unterstützt. Ab TF5400 3.2.27 nicht sichtbar.
Decelerating	BOOL	Nicht unterstützt. Ab TF5400 3.2.27 nicht sichtbar.
InPosition	BOOL	Nicht unterstützt.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.6 MC_GroupReset



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘ (✔ bis v3.2 einschließlich)	✔

Der Funktionsbaustein MC_GroupReset setzt alle internen Fehler einer Gruppe und alle Achsen, die zu der Gruppe gehören, zurück. Wenn die Gruppe aktiviert war, als der Fehler aufgetreten ist, geht die Gruppe in den Zustand GroupStandby über. Wenn die Gruppe deaktiviert war, geht sie in den Zustand GroupDisabled über (siehe Zustandsdiagramme).

Wird dieser Funktionsbaustein aufgerufen, während kein Fehler vorliegt, dann hat er keine Wirkung.

Eingänge

```
VAR_INPUT
    Execute   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.

Ein-/Ausgänge

```
VAR_IN_OUT
   AxesGroup              : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle).

Ausgänge

```
VAR_OUTPUT
    Done                   : BOOL;
    Busy                   : BOOL;
    Error                  : BOOL;
    ErrorId                : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Execute gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn Busy wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted (falls vorhanden) oder Error gesetzt.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.

Name	Typ	Beschreibung
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.7 MC_GroupSetOverride

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Der Funktionsbaustein MC_GroupSetOverride ändert den Override einer Gruppe. Eine Änderung erfolgt mit einer gewissen Verzögerung. Ein Override-Eingangswert ist zwischen 0 [0 %] und 1 [100 %] gültig. Wird der Wert außerhalb dieses Bereichs festgelegt, dann wird er automatisch auf den jeweiligen Grenzwert gesetzt.

Das Verhalten für Override-Änderungen in Bezug auf die **MC-Gruppe** kann als Achsgruppenparameter festgelegt werden, siehe Time Override Ramp Time.

Eingänge

```
VAR_INPUT
    Enable                  : BOOL;
    VelFactor               : MC_LREAL := 1.0;
END_VAR
```

Name	Typ	Beschreibung
Enable	BOOL	Das Kommando wird so lange ausgeführt, wie Enable aktiv ist.
VelFactor	MC_LREAL	Der Override wird auf diesen Wert gesetzt (Wertebereich zwischen 0 [0 %] und 1 [100 %]).

Ein-/Ausgänge

```
VAR_IN_OUT
   AxesGroup                : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle).

Ausgänge

```
VAR_OUTPUT
    Enabled              : BOOL;
    Busy                 : BOOL;
    Error                : BOOL;
    ErrorId              : UDINT;
    ActualVelFactor      : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Enabled	BOOL	Dieser Ausgang signalisiert, dass der `VelFactor` erfolgreich gesetzt wurde. Der `VelFactor` zeigt den Typ eines Override-Faktors.
Busy	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando mit `Enable` gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt.
Error	BOOL	Dieser Ausgang wird `TRUE`, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.
ActualVelFactor	UDINT	Override, der gegenwärtig in der Gruppe aktiv ist (Wertebereich zwischen 0 [0 %] und 1 [100 %]).

Beispiel

```
VAR
    stGroupRef             : AXES_GROUP_REF;
    fbSetOverride          : MC_GroupSetOverride;
END_VAR
```

```
fbSetOverride(
    AxesGroup:=stGroupRef ,
    Enable:= TRUE ,
    VelFactor:=1.0 , (* 1.0 = 100% *)
);
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.8 MC_RemoveAxisFromGroup

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘ (✔ bis v3.2 einschließlich)	✔

Der Funktionsbaustein MC_RemoveAxisFromGroup entfernt eine Achse aus der Achsgruppe.

Ab TF5400 V3.1.10.1 können einer **CA-Gruppe** im Gruppenzustand GroupMoving stehende Achsen hinzugefügt und aus ihr entfernt werden. Wenn einer Gruppe eine fahrende Achse hinzugefügt wird, wird das Kommando mit einer Fehlermeldung abgelehnt (eine Änderung des Gruppenzustands mit einer fahrenden Achse wird ebenfalls abgelehnt).

Achsen können einer **MC-Gruppe** nur hinzugefügt werden, wenn EnableRequested FALSE ist, z. B. im Zustand GroupDisabled.

Erfolg des Funktionsbausteins

Der Funktionsbaustein gibt immer DONE zurück, wenn die Achse nicht mehr zur Gruppe gehört. Dies bedeutet, dass DONE auch dann zurückgegeben wird, wenn die Achse vor dem Aufruf des Funktionsbausteins nicht in der Gruppe war.

Eingänge

```
VAR_INPUT
    Execute                 : BOOL;
    IdentInGroup            : IDENT_IN_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
IdentInGroup	IDENT_IN_GROUP_REF	Definiert die Interpretation der Achse, die der Gruppe hinzugefügt werden soll. Für mehrdi-mensionale Bewegungen kann dies die kartesische Interpretation sein. Die globalen Variablen (z. B. MCS_X) müssen verwendet werden. Zur Collision Avoidance muss die Funktion U-DINT_TO_IDENTINGROUP verwendet werden.

Verwendung ganzzahliger Werte für den Eingang IdentInGroup

Die Verwendung ganzzahliger Werte für den Eingang IdentInGroup wird NICHT unterstützt und kann zu Inkompatibilität mit künftigen Releases führen. Bei Verwendung ganzzahliger Werte kann das Projekt unter Umständen nicht mehr aufgebaut werden. Es wird empfohlen, globale Variablen [▶ 78] (z. B. MCS_X) oder die Konvertierungsfunktion UDINT_TO_IDENTINGROUP [▶ 60] zu verwenden.

Ein-/Ausgänge

```
VAR_IN_OUT
   AxesGroup                : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle).

Ausgänge

```
VAR_OUTPUT
    Done                    : BOOL;
    Busy                    : BOOL;
    Error                   : BOOL;
    ErrorId                 : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Execute gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn Busy wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted (falls vorhanden) oder Error gesetzt.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.9 MC_SetCoordinateTransform

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Der Funktionsbaustein MC_SetCoordinateTransform aktiviert eine Koordinatentransformation für nachfolgende Bewegungen. Die erfolgreiche Aktivierung wird angezeigt durch `Active` oder `Done`.

Entkoppelt die nachfolgenden Bewegungen von einem Transportsystem (siehe MC_TrackConveyorBelt [▶ 56]).

Nachfolgende Bewegungen (z. B. MC_MovePath [▶ 71]) erfolgen relativ zur Koordinatentransformation.

Anwendungsfall für das Ändern des Referenzsystems

Durch Verwendung von `MC_SetCoordinateTransform` und Ändern des Referenzsystems kann die `MC`-Gruppe entkoppelt werden.

Eingänge

```
VAR_INPUT
    Execute         : BOOL;
    CoordTransform : MC_COORD_REF;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
CoordTransform	MC_COORD_REF	Referenz auf ein Koordinatensystem (siehe MC_COORD_REF [▶ 88]).

Ein-/Ausgänge

```
VAR_IN_OUT
    AxesGroup       : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle [▶ 99]).

Ausgänge

```
VAR_OUTPUT
    Done            : BOOL;
    Busy            : BOOL;
    Active          : BOOL;
    CommandAborted : BOOL;
    Error           : BOOL;
    ErrorId         : UDINT;
END_VAR
```

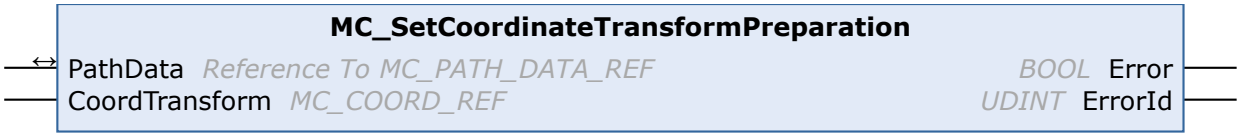
Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando mit `Execute` gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn `Busy` wieder `FALSE` wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge `Done`, `CommandAborted` oder `Error` gesetzt.
Active	BOOL	`Active` gibt an, dass das Kommando ausgeführt wird. `Active` gibt an, dass das Referenzsystem erfolgreich gesetzt wurde (nur MC Coordinated Motion Group). `Active` zeigt eine Verzögerung des Conveyor Trackings an (nur MC Coordinated Motion Group). `Active` wird `FALSE` wenn einer der Ausgänge `Done`, `CommandAborted` oder `Error` auf `TRUE` gesetzt wird. Hinweis: Entsprechend der PLCopen-Definition, wird `Active` zurückgesetzt, wenn `Done` auf `TRUE` gesetzt wird. Im Fall einer unwesentlichen, bis nicht vorhandenen Verzögerung, kann `Active` nur für einen geringfügigen Zeitraum auf `TRUE` gesetzt werden. Sollte im PLC Programm auf `Active` geprüft werden, ist es daher empfehlenswert, zusätzlich auch auf `Done` zu prüfen.

Name	Typ	Beschreibung
CommandAborted	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando durch ein anderes Kommando unterbrochen worden ist.
Error	BOOL	Dieser Ausgang wird `TRUE`, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4022.25 TF5400 Advanced Motion Pack V3.1.6.03	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.10 MC_SetCoordinateTransformPreparation



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗	✓

Der Funktionsbaustein MC_SetCoordinateTransformPreparation fügt in die Tabelle, welche die Segmente einer Bahn enthält, einen Wechsel des Referenzsystems ein.

Eingänge

```
VAR_INPUT
    PathData        : MC_PATH_DATA_REF;
    CoordTransform : MC_COORD_REF;
END_VAR
```

Name	Typ	Beschreibung
PathData	MC_PATH_DATA_REF [▶ 84]	Tabelle, die die Segmente einer Bahn enthält. Die Tabelle wird durch die Funktionsbausteine MC_… Preparation geschrieben und durch MC_MovePath [▶ 71] ausgeführt.
CoordTransform	MC_COORD_REF [▶ 88]	Referenz auf ein Koordinatensystem.

Ausgänge

```
VAR_OUTPUT
    Error   : BOOL;
    ErrorId : UDINT;
END_VAR
```

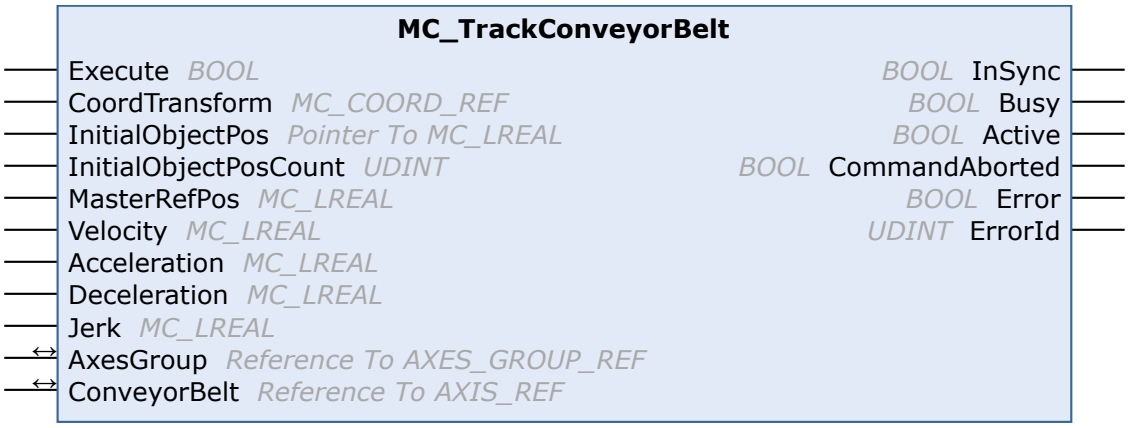
Name	Typ	Beschreibung
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.

Name	Typ	Beschreibung
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliothek
TwinCAT V3.1.4024.40 TF5400 Advanced Motion Pack V3.3.19	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.11 MC_TrackConveyorBelt



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Der Funktionsbaustein Mc_TrackConveyorBelt aktiviert ein in Bewegung befindliches Referenzsystem. Dabei synchronisiert er die `AxesGroup` mit dem `ConveyorBelt` in Bezug auf die Geschwindigkeit.

Die Synchronisation mit einer Position erfordert ein Bewegungskommando.

Somit hilft der Funktionsbaustein bei der Synchronisation mit einem Objekt, das sich auf einer Geraden durch den Raum bewegt. Ein Beispiel sind Produkte, die auf einem Förderband oder einem anderen Transportsystem laufen.

Der Ursprung des Förderbands wird mit einem Koordinatensystem parametriert (`CoordTransform`). `X` ist die Förderrichtung. Die erkannte Objektposition (`InitialObjectPos`) und die entsprechende Touch Probe-Position (`MasterRefPos`) werden in den Funktionsbaustein eingegeben.

Synchronisationsdynamik kann in den Funktionsbaustein eingegeben werden.

Bewegungen, die nach `Active = TRUE` ausgeführt werden, werden mit dem Förderband synchronisiert.

Die Ausführung von `MC_TrackConveyorBelt` mit einer weiteren Instanz bewirkt eine direkte Synchronisation mit einem zweiten Förderband.

Beim Ändern des Referenzsystems kann ein Förderband entkoppelt werden.

Anwendungsfall für das Ändern des Referenzsystems

Durch Verwendung von `MC_TrackConveyorBelt` und Ändern des Referenzsystems kann die `MC`-Gruppe entkoppelt werden. Das Referenzsystem kann mit `MC_SetCoordinateTransform` geändert werden.

Neuheiten und Optimierungen bzgl. MC_TrackConveryorBelt mit TF5400 V3.2.27 für die MC Group Coordinated Motion

- Neu: Optional wirkt der Override auch auf die Synchronisierungsphase beim MC_TrackConveyorBelt. Die Einstellung erfolgt im Parameter „Tracking Override Behavior“ in der MC Group Coordinated Motion [▶ 18].
- Optimierungen am MC_TrackConveyorBelt, die einen SAF-Zyklusversatz zwischen Conveyor (Master-) und Slaveachse unterbinden.
- Optimierungen der Fehlerreaktion beim MC_TrackConveyorBelt. Im Falle eines Laufzeitfehlers des Förderbandes (Master), wird ein aktives MC_MovePath nicht abgebrochen und eine Fehlerreaktion ist über die SPS anzustoßen.

Eingänge

```
VAR_INPUT
    Execute                : BOOL;
    CoordTransform         : MC_COORD_REF;
    InitialObjectPos       : POINTER TO MC_LREAL;
    InitialObjectPosCount  : UDINT;
    MasterRefPos           : MC_LREAL;
    Velocity               : MC_LREAL := MC_DEFAULT;
    Acceleration           : MC_LREAL := MC_DEFAULT;
    Deceleration           : MC_LREAL := MC_DEFAULT;
    Jerk                   : MC_LREAL := MC_DEFAULT;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
CoordTransform	MC_COORD_REF	Referenz auf ein Koordinatensystem (siehe MC_COORD_REF [▶ 88]).
InitialObjectPos	POINTER TO MC_LREAL	Zeiger auf Array `[1..InitialObjectPosCount]`.
InitialObjectPosCount	UDINT	Dimension des Vektors `InitialObjectPos`.
MasterRefPos	MC_LREAL	Touch Probe-Position.
Velocity	MC_LREAL	Die Geschwindigkeit für die Synchronisation. Die Geschwindigkeit muss die Förderbandgeschwindigkeit überschreiten. Die Geschwindigkeit ist nicht durch die maximale Achsgeschwindigkeit beschränkt.
Acceleration	MC_LREAL	Wird im Objekt Conveyor Tracking verwendet. Die Beschleunigung für die Synchronisation. Die Beschleunigung ist nicht durch die maximale Achsbeschleunigung beschränkt. Wird kein Wert eingegeben, dann wird die Standard-Beschleunigung des Conveyor Tracking Objekts verwendet.
Deceleration	MC_LREAL	Wird im Objekt Conveyor Tracking verwendet. Die Verzögerung für die Synchronisation. Die Verzögerung ist nicht durch die maximale Achsverzögerung beschränkt. Wird kein Wert eingegeben, dann wird die Standard-Verzögerung des Conveyor Tracking Objekts verwendet.

Name	Typ	Beschreibung
Jerk	MC_LREAL	Der Ruck für die Synchronisation. Wird kein Wert eingegeben, dann wird der Standard-Ruck des Conveyor Tracking Objekts verwendet. Der maximale Ruck ist nicht beschränkt.

Ein-/Ausgänge

```
VAR_IN_OUT
    AxesGroup              : AXES_GROUP_REF;
    ConveyorBelt           : AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle [▶ 99]).
ConveyorBelt	AXIS_REF	Referenz auf eine Achse. Referenz auf die Fördererachse.

Ausgänge

```
VAR_OUTPUT
    InSync                 : BOOL;
    Busy                   : BOOL;
    Active                 : BOOL;
    CommandAborted         : BOOL;
    Error                  : BOOL;
    ErrorId                : UDINT;
END_VAR
```

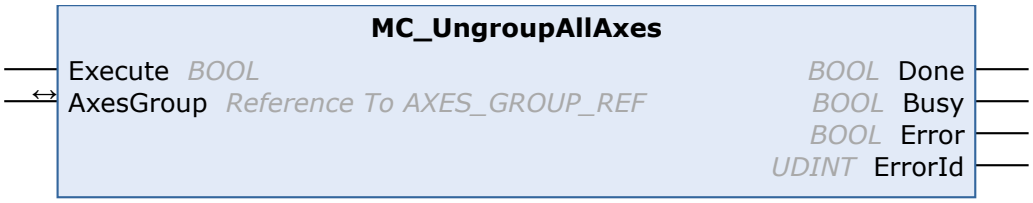
Name	Typ	Beschreibung
InSync	BOOL	Der Ausgang `InSync` wird erstmals `TRUE`, sobald der Slave mit der Geschwindigkeit synchronisiert ist. Ist die Slave-Dynamik zu gering, um der Master-Bewegung zu folgen, könnte der Ausgang `InSync` auf `FALSE` zurückgesetzt werden, woraufhin die Slaveachse erneut mit der Synchronisation beginnt. **Hinweis Geschwindigkeitssynchronisation: Active und InSync - Der Funktionsbaustein MC_TrackConveyorBelt synchronisiert die AxesGroup mit der Geschwindigkeit der ConveyorBelt-Achse. Dabei verwendet der Funktionsbaustein die gege-benen Parameter für Acceleration, Deceleration und Jerk. Wenn diese Synchronisationsbewegung beginnt, wird Active auf TRUE gesetzt. Wenn die Geschwindigkeit des ConveyorBelt erreicht ist, wird InSync auf TRUE gesetzt. Der Synchronisationsstatus wird kontinuierlich überwacht und mit InSync angegeben.** **Hinweis Fördererbewegung, Standard-Tracking-Verhalten und InSync - Nachdem das Ausgangssignal InSync gesetzt wurde, gibt es zwei Optionen, um die Synchronisation aufrechtzuerhalten. mcTrackingBehaviorDynLimited - Dieses Verhalten ist das standardmäßige (MC_Default) Tracking-Verhalten. Die Axes-Group erhält die Geschwindigkeitssynchronisation mit dem ConveyorBelt mit Hilfe der ge-gebenen Parameter für Acceleration, Deceleration und Jerk aufrecht. – mcTrackingBehaviorStayInSync - Die AxesGroup erhält die Geschwindigkeitssynchronisation mit dem ConveyorBelt mit unbegrenzten Parametern für Acceleration, Deceleration und Jerk aufrecht.** **Hinweis Positionssynchronisation: MasterRefPos und InitialObjectPos - Die Funktionsbausteine MC_TrackConveyorBelt und MC_MovePath sollen für eine flexible Synchronisation mit einer beweglichen Zielposition zusammen verwendet werden. Nachdem MC_TrackConveyorBelt.Active auf TRUE gesetzt worden ist, werden Initia-lObjectPos und der Abstand zur MasterRefPos dem nächsten Aufruf von MC_MovePath angehängt. MC_TrackConveyorBelt.InSync = TRUE und MC_MovePath.Done = TRUE geben an, dass die synchronisierte Position erreicht ist.**

Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando mit `Execute` gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn `BUSY` wieder `FALSE` wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge `CommandAborted` oder `Error` gesetzt.
Active	BOOL	Wenn `Active` `TRUE` ist, steuert der Funktionsbaustein die Gruppe.
CommandAborted	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando durch ein anderes Kommando unterbrochen worden ist.
Error	BOOL	Dieser Ausgang wird `TRUE`, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4022.25 TF5400 Advanced Motion Pack V3.1.6.03	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.12 MC_UngroupAllAxes



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘ (✔ bis v3.2 einschließlich)	✔

Der Funktionsbaustein MC_UngroupAllAxes entfernt alle Achsen und deaktiviert die Gruppe. Wenn der Funktionsbaustein erfolgreich ist, befindet sich die Gruppe anschließend im Gruppenzustand GroupDisabled (siehe Zustandsdiagramme).

Eingänge

```
VAR_INPUT
    Execute   : BOOL;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.

Ein-/Ausgänge

```
VAR_IN_OUT
   AxesGroup              : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle).

Ausgänge

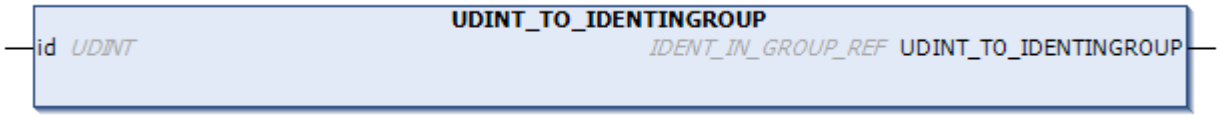
```
VAR_OUTPUT
    Done                  : BOOL;
    Busy                  : BOOL;
    Error                 : BOOL;
    ErrorId               : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Execute gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn Busy wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted (falls vorhanden) oder Error gesetzt.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.1.13 UDINT_TO_IDENTINGROUP



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘	✔

Die Funktion UDINT_TO_IDENTINGROUP ist eine Konvertierungsfunktion, die einen ganzzahligen Wert in IDENT_IN_GROUP_REF umwandelt. Einer CA-Gruppe muss eine PTP-Achse ohne räumliche Interpretation hinzugefügt werden. Diese Konvertierungsfunktion gibt einen gültigen Eingang für MC_AddAxisToGroup [▶ 40] und MC_RemoveAxisFromGroup [▶ 51] zurück. Für Achsen, die für mehrdimensionale Bewegung vorgesehen sind (TF5420), siehe IDENT_IN_GROUP_REF [▶ 78].

Verwendung ganzzahliger Werte für den Eingang IdentInGroup

Die Verwendung ganzzahliger Werte für den Eingang IdentInGroup wird NICHT unterstützt und kann zu Inkompatibilität mit künftigen Releases führen. Bei Verwendung ganzzahliger Werte kann das Projekt unter Umständen nicht mehr aufgebaut werden. Es wird empfohlen, globale Variablen [▶ 78] (z. B. MCS_X) oder die Konvertierungsfunktion UDINT_TO_IDENTINGROUP [▶ 60] zu verwenden.

Rückgabewert

Name	Typ	Beschreibung
UDINT_TO_IDENTINGROUP	IDENT_IN_GROUP_REF [▶ 78]	Wandelt einen ganzzahligen Wert um, so dass eine PTP-Achse einer Bewegungsgruppe hinzugefügt werden kann.

Eingänge

```
VAR_INPUT
    id                  : UDINT;
END_VAR
```

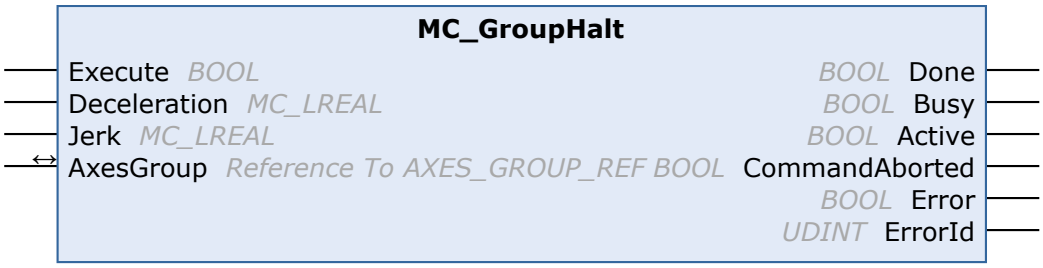
Name	Typ	Beschreibung
id	UDINT	Die eindeutige Kennung, die eine Achse in der Gruppe haben soll. Dies muss nicht die Achs-ID des zyklischen Achsinterfaces sein.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.2 Motion

7.1.1.2.1 MC_GroupHalt



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘	✔

Der Funktionsbaustein MC_GroupHalt stoppt eine Gruppe mit einer definierten Bremsrampe. Anders als bei "MC_GroupStop [▶ 63]" wird die Gruppe nicht für weitere Bewegungskommandos gesperrt. Daher kann die Gruppe durch ein weiteres Kommando während der Bremsrampe oder nach dem Anhalten neu gestartet werden.

⚠ WARNUNG

Eventuell verzögerter Achsen-Stopp

Wenn bei einer CA-Gruppe Standby `Gap Control` aktiv ist und eine Unterschreitung der minimal Gap kommt dazu, wird erst das Gap ausgefahren, bevor die Achsen durch ein `MC_GroupHalt` angehalten werden.

- Stellen Sie sicher, dass Sie das Verhalten von Standby `Gap Control` wirklich benötigen und deaktivieren Sie es gegebenenfalls (Default-Einstellung).
- Verwenden Sie ein MC_GroupStop, anstelle eines MC_GroupHalt, wenn ein unverzögertes Stoppen der Achsen benötigt wird.

HINWEIS

MC_GroupHalt nicht für MC-Group with Pick-and-Place implementiert

Der Funktionsbaustein MC_GroupHalt ist nur für die MC Group Coordinated Motion und für PTP-Bewegungen mit Collision Avoidance (CA-Gruppe) implementiert. Bei Verwendung mit einem anderen Gruppentyp wird das Kommando abgelehnt.

Gilt für die MC_Group: MC_GroupHalt löscht die aktive Koordinatentransformation und löscht alle Jobs in der Warteschlange.

Eingänge

```
VAR_INPUT
    Execute                 : BOOL;
    Deceleration            : MC_LREAL := MC_DEFAULT;
    Jerk                    : MC_LREAL := MC_DEFAULT;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
Deceleration	MC_LREAL	[mm/s²]. Die Verzögerung kann als skalarer Wert (≥1) programmiert werden oder es können "Spezielle Eingangswerte [▶ 102]" verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus.
Jerk	MC_LREAL	[mm/s³]. Der Ruck kann als skalarer Wert (≥100) programmiert werden oder es können "Spezielle Eingangswerte [▶ 102]" verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus. MC_IGNORE führt das Kommando mit unbegrenztem Ruck aus.

Ein-/Ausgänge

```
VAR_IN_OUT
   AxesGroup               : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle).

Ausgänge

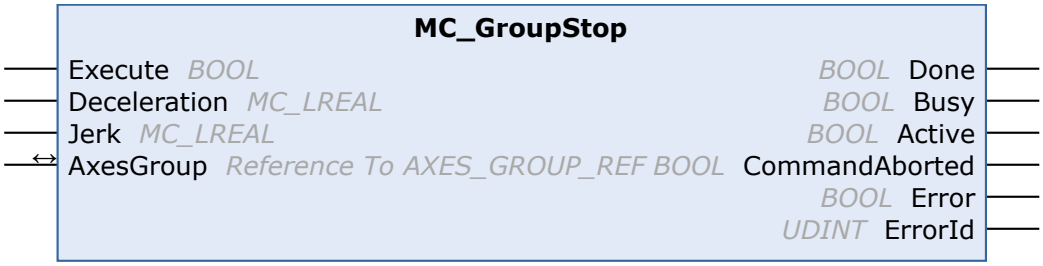
```
VAR_OUTPUT
    Done                    : BOOL;
    Busy                    : BOOL;
    Active                  : BOOL;
    CommandAborted          : BOOL;
    Error                   : BOOL;
    ErrorId                 : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Done	BOOL	Wird TRUE, wenn die Gruppe gestoppt wurde und zum Stillstand gekommen ist. Sobald die Gruppe zum Stillstand gekommen ist, wird der Gruppenzustand zu GroupStandby (siehe Zustandsdiagramme).
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Execute gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn Busy wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted (falls vorhanden) oder Error gesetzt.
Active	BOOL	Active gibt an, dass das Kommando ausgeführt wird. Wenn das Kommando in der Warte-schlange war, wird es aktiv, sobald ein ausgeführtes Kommando abgeschlossen wird.
CommandAborted	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando durch ein anderes Kommando un-terbrochen worden ist.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.2.2 MC_GroupStop



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘ (✔ bis v3.2 einschließlich)	✔

Der Funktionsbaustein MC_GroupStop stoppt die Gruppe und alle zugehörigen Achsen mit einer definierten Bremsrampe und sperrt die Achse für Bewegungskommandos. Während sich die Gruppe im Zustand GroupStopping befindet, kann kein anderer Funktionsbaustein eine Achse der Gruppe bewegen (siehe Zustandsdiagramme).

Die Gruppe kann erst wieder bewegt werden, sobald das Signal *Execute* auf FALSE gesetzt wurde, nachdem die Geschwindigkeit 0 ist.

MC_GroupStop löscht die aktive Koordinatentransformation und löscht alle Jobs in der Warteschlange.

Eingänge

```
VAR_INPUT
    Execute                 : BOOL;
    Deceleration            : MC_LREAL := MC_DEFAULT;
    Jerk                    : MC_LREAL := MC_DEFAULT;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
Deceleration	MC_LREAL	[mm/s²]. Die Verzögerung kann als skalarer Wert (≥1) programmiert werden oder es können "Spezielle Eingangswerte [▶ 102]" verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus.
Jerk	MC_LREAL	[mm/s³]. Der Ruck kann als skalarer Wert (≥100) programmiert werden oder es können "Spezielle Eingangswerte [▶ 102]" verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus. MC_IGNORE führt das Kommando mit unbegrenztem Ruck aus.

Ein-/Ausgänge

```
VAR_IN_OUT
   AxesGroup               : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklische Gruppenschnittstelle).

Ausgänge

```
VAR_OUTPUT
    Done                    : BOOL;
    Busy                    : BOOL;
    Active                  : BOOL;
    CommandAborted          : BOOL;
    Error                   : BOOL;
    ErrorId                 : UDINT;
END_VAR
```

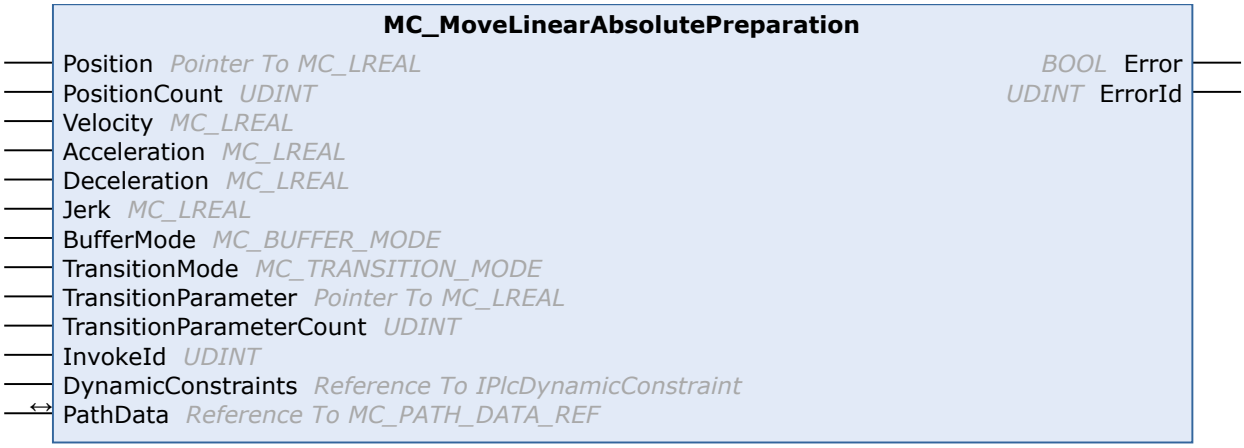
Name	Typ	Beschreibung
Done	BOOL	Wird TRUE, wenn die Gruppe gestoppt wurde und zum Stillstand gekommen ist. Während *Execute* TRUE ist, aber mindestens so lange, bis die Achsen zum Stillstand gekommen sind, bleibt die Gruppe im Zustand GroupStopping. Anschließend befindet sich die Gruppe im Zustand GroupStandby (siehe Zustandsdiagramme).
Busy	BOOL	Wird TRUE, wenn das Kommando mit *Execute* gestartet wird, und bleibt es dann so lange, wie das Kommando ausführt wird. Wenn *Busy* wieder FALSE wird, ist die Gruppe bereit für ein neues Kommando. Nachdem die Gruppe gestoppt wurde, bleibt *Busy* TRUE, bis die Gruppe mit *Execute*=FALSE freigegeben wird.
Active	BOOL	Gibt an, dass der Funktionsbaustein die Gruppe steuert. Nachdem die Gruppe gestoppt wurde, bleibt *Active* TRUE, bis die Gruppe mit Execute=FALSE freigegeben wird.

Name	Typ	Beschreibung
CommandAborted	BOOL	Das Kommando wird abgebrochen, indem MC_Power von mindestens einer Achse der Gruppe deaktiviert wird oder wenn die Gruppe während des Kommandos deaktiviert wird.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.2.3 MC_MoveLinearAbsolutePreparation



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Der Funktionsbaustein MC_MoveLinearAbsolutePreparation fügt der Tabelle der Segmente in der Struktur PathData eine absolute Linearbewegung hinzu. Nach der Erstellung einer Tabelle kann er über MC_MovePath [▶ 71] ausgeführt werden. Der Funktionsbaustein MC_MoveLinearAbsolutePreparation kann mehrmals pro Zyklus aufgerufen werden. Maximal 30 Einträge sind pro PathData-Tabelle zulässig.

Eingänge

```
VAR_INPUT
    Position                    : POINTER TO LREAL;
    PositionCount               : UDINT;
    Velocity                    : MC_LREAL := MC_INVALID;
    Acceleration                : MC_LREAL := MC_DEFAULT;
    Deceleration                : MC_LREAL := MC_DEFAULT;
    Jerk                        : MC_LREAL := MC_DEFAULT;
    BufferMode                  : MC_BUFFER_MODE := mcAborting;
    TransitionMode              : MC_TRANSITION_MODE := mcTransModeNone;
```

```
    TransitionParameter       : POINTER TO LREAL;
    TransitionParameterCount  : UDINT;
    InvokeId                  : UDINT;
    DynamicConstraints        : REFERENCE TO IPlcDynamicConstraint := 0;
END_VAR
```

Name	Typ	Beschreibung
Position	POINTER TO LREAL	Zeiger auf ein Array [1..PositionCount] des Zielpositionsvektors.
PositionCount	UDINT	Dimension des Positionsvektors. Muss der Anzahl der Achsen in der Achskonvention entsprechen (siehe MC Group Coordinated Motion oder MC Group with Pick-and-Place).
Velocity	MC_LREAL	Die maximale Geschwindigkeit für das programmierte Segment. Die Geschwindigkeit muss nicht immer erreicht werden. Die Geschwindigkeit muss >0 gesetzt werden.
Acceleration	MC_LREAL	Maximale Bahnbeschleunigung für das programmierte Segment. Spezielle Eingangswerte [▶ 102] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus. Die Beschleunigung muss ≥1 gesetzt werden.
Deceleration	MC_LREAL	Maximale Bahnverzögerung für das programmierte Segment. Spezielle Eingangswerte [▶ 102] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus. Die Verzögerung muss ≥1 gesetzt werden.
Jerk	MC_LREAL	Bahnruck für das programmierte Segment. Spezielle Eingangswerte [▶ 102] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. Der Ruck muss ≥100 gesetzt werden. **Ab TF5400 V3.2.27:** Bei der MC Group Coordinated Motion wird MC_MAXIMUM unterstützt. Dabei ist MC_MAXIMUM = 100 * MC_DEFAULT.
BufferMode	MC_BUFFER_MODE	Legt fest, wie aufeinanderfolgende Fahraufträge abgearbeitet werden sollen (siehe MC_BUFFER_MODE [▶ 88]).
Transition Mode	MC_TRANSITION_MODE	Definiert den Blending-Modus (siehe MC_TRANSITION_MODE [▶ 85]).
TransitionParameter	POINTER TO LREAL	Zeiger auf Array [1..TransitionParameterCount] der Blending-Parameter. Transitionsparameter definieren das Blending von der letzten programmierten Position (siehe MC_TRANSITION_MODE [▶ 85]).
TransitionParameterCount	UDINT	Anzahl der Blending-Parameter (siehe MC_TRANSITION_MODE [▶ 85]).
InvokeId	UDINT	Segment-ID für Analysezwecke.
DynamicConstraints	REFERENCE TO IPlcDynamicConstraint	**Ab TF5400 V3.2.27, MC Group Coordinated Motion:** Optionaler Eingang um die erlaubten Werte für Geschwindigkeit, Beschleunigung, Verzögerung oder Ruck während der Bewegung weiter zu beschränken. **Hinweis Referenz-Zuweisung:** `<InstanceOfMC_MoveLinearAbsolutePreparation>.DynamicConstraints` **REF=** `<InstanceOfIPlcDynamicConstraint>;`

Ein-/Ausgänge

```
VAR_IN_OUT
    PathData                  : MC_PATH_DATA_REF;
END_VAR
```

Name	Typ	Beschreibung
PathData	MC_PATH_DATA_REF	Tabelle, die die Segmente einer Bahn enthält. Die Tabelle wird durch MC_Move…Preparation geschrieben und durch MC_MovePath [▶ 71] ausgeführt (siehe MC_PATH_DATA_REF [▶ 84]).

Zurücksetzen einer Tabelle

Eine Tabelle wird bei der Ausführung nicht zurückgesetzt. Zum Zurücksetzen muss die Methode `ClearPath()` von MC_PATH_DATA_REF aufgerufen werden.

Ausgänge

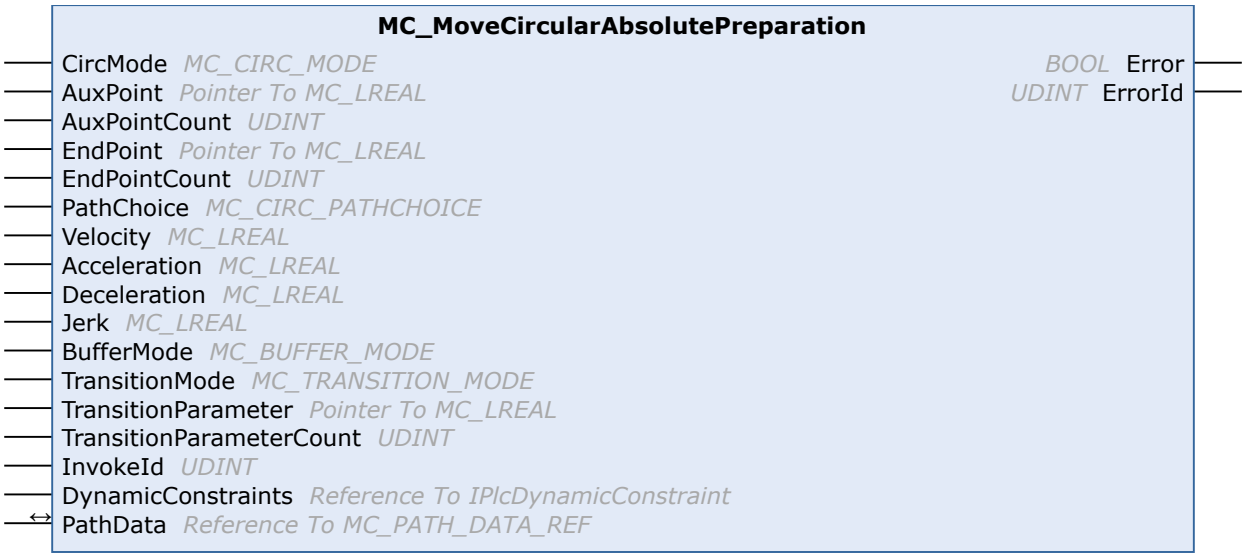
```
VAR_OUTPUT
    Error                       : BOOL;
    ErrorId                     : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.2.4 MC_MoveCircularAbsolutePreparation



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Der Funktionsbaustein MC_MoveCircularAbsolutePreparation fügt der Tabelle der Segmente in der Struktur MC_PATH_DATA_REF [▶ 84] eine absolute Kreisbewegung hinzu. Nach der Erstellung einer Tabelle kann er über MC_MovePath [▶ 71] ausgeführt werden. Der Funktionsbaustein MC_MoveCircularAbsolutePreparation kann mehrmals pro Zyklus aufgerufen werden. Maximal 30 Einträge sind pro MC_Path_DATA_REF-Tabelle zulässig.

Zurücksetzen einer Tabelle

Eine Tabelle wird bei der Ausführung nicht zurückgesetzt. Zum Zurücksetzen muss die Methode ClearPath [▶ 85] von MC_PATH_DATA_REF [▶ 84] aufgerufen werden.

Eingänge

```
VAR_INPUT
    CircMode                 : MC_CIRC_MODE := mcCircModeInvalid;
    AuxPoint                 : POINTER TO MC_LREAL;
    AuxPointCount            : UDINT;
    EndPoint                 : POINTER TO MC_LREAL;
    EndPointCount            : UDINT;
    PathChoice               : MC_CIRC_PATHCHOICE := mcCircPathchoiceCounterClockwise;
    Velocity                 : MC_LREAL := MC_INVALID;
    Acceleration             : MC_LREAL := MC_DEFAULT;
    Deceleration             : MC_LREAL := MC_DEFAULT;
    Jerk                     : MC_LREAL := MC_DEFAULT;
    BufferMode               : MC_BUFFER_MODE := mcAborting;
    TransitionMode           : MC_TRANSITION_MODE := mcTransModeNone;
    TransitionParameter      : POINTER TO MC_LREAL;
    TransitionParameterCount : UDINT;
    InvokeId                 : UDINT;
    DynamicConstraints       : REFERENCE TO IPlcDynamicConstraint := 0;
END_VAR
```

Name	Typ	Beschreibung
CircMode	MC_CIRC_MODE	Legt fest, durch welche Kreisdefinition der Kreis programmiert wird. Legt die Bedeutung des Eingangssignals "AuxPoint" fest (siehe MC_CIRC_MODE [▶ 79]).
AuxPoint	POINTER TO MC_LREAL	Zeiger auf ein Array [1..AuxPointCount] des AuxPoint-Vektors. Die Interpretation des AuxPoint-Vektors hängt von der Raumachskonvention ab (siehe MC Group Coordinated Motion).
AuxPointCount	UDINT	Dimension des AuxPoint-Vektors.
EndPoint	POINTER TO MC_LREAL	Zeiger auf ein Array [1..EndPointCount] des Zielpositionsvektors.
EndPointCount	UDINT	Dimension des EndPoint-Vektors. Muss der Anzahl der Achsen in der Achskonvention entsprechen (siehe MC Group Coordinated Motion oder MC Group with Pick-and-Place).
PathChoice	MC_CIRC_PATHCHOICE	Definiert die Drehrichtung in Bezug auf den Normalvektor. Der Eingang wird ignoriert, wenn der Eingang *CircMode* auf *mcCircModeBorder* gesetzt ist (siehe MC_CIRC_PATHCHOICE [▶ 83]).
Velocity	MC_LREAL	Die maximale Geschwindigkeit für das programmierte Segment. Die Geschwindigkeit muss nicht immer erreicht werden. Die Geschwindigkeit muss >0 gesetzt werden.

Name	Typ	Beschreibung
Acceleration	MC_LREAL	Maximale Bahnbeschleunigung für das programmierte Segment. Spezielle Eingangswerte [▶ 102] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus. Die Beschleunigung muss ≥1 gesetzt werden.
Deceleration	MC_LREAL	Maximale Bahnverzögerung für das programmierte Segment. Spezielle Eingangswerte [▶ 102] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus. Die Verzögerung muss ≥1 gesetzt werden.
Jerk	MC_LREAL	Bahnruck für das programmierte Segment. Spezielle Eingangswerte [▶ 102] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. Der Ruck muss ≥100 gesetzt werden. **Ab TF5400 V3.2.27:** Bei der MC Group Coordinated Motion wird MC_MAXIMUM unterstützt. Dabei ist MC_MAXIMUM = 100 * MC_DEFAULT.
BufferMode	MC_BUFFER_MODE	Legt fest, wie aufeinanderfolgende Fahraufträge abgearbeitet werden sollen (siehe MC_BUFFER_MODE [▶ 88]).
Transition Mode	MC_TRANSITION_MODE	Definiert den Blending-Modus (siehe MC_TRANSITION_MODE [▶ 85]).
TransitionParameter	POINTER TO MC_LREAL	Zeiger auf Array [1..TransitionParameterCount] der Blending-Parameter. Transitionsparameter definieren das Blending von der letzten programmierten Position (siehe MC_TRANSITION_MODE [▶ 85]).
TransitionParameterCount	UDINT	Anzahl der Blending-Parameter.
InvokeId	UDINT	Segment-ID für Analysezwecke.
DynamicContraints	REFERENCE TO IPlcDynamicConstraint	**Ab TF5400 V3.2.27, MC Group Coordinated Motion:** Optionaler Eingang um die erlaubten Werte für Geschwindigkeit, Beschleunigung, Verzögerung oder Ruck während der Bewegung weiter zu beschränken. **Hinweis Referenz-Zuweisung:** `<InstanceOfMC_MoveCircularAbsolutePreparation>.DynamicConstraints` **REF=** `<InstanceOfIPlcDynamicConstraint>;`

Ein-/Ausgänge

```
VAR_IN_OUT
    PathData                    : MC_PATH_DATA_REF;
END_VAR
```

Name	Typ	Beschreibung
PathData	MC_PATH_DATA_REF	Tabelle, die die Segmente einer Bahn enthält. Die Tabelle wird durch MC_Move…Preparation geschrieben und durch MC_MovePath [▶ 71] ausgeführt (siehe MC_PATH_DATA_REF [▶ 84]).

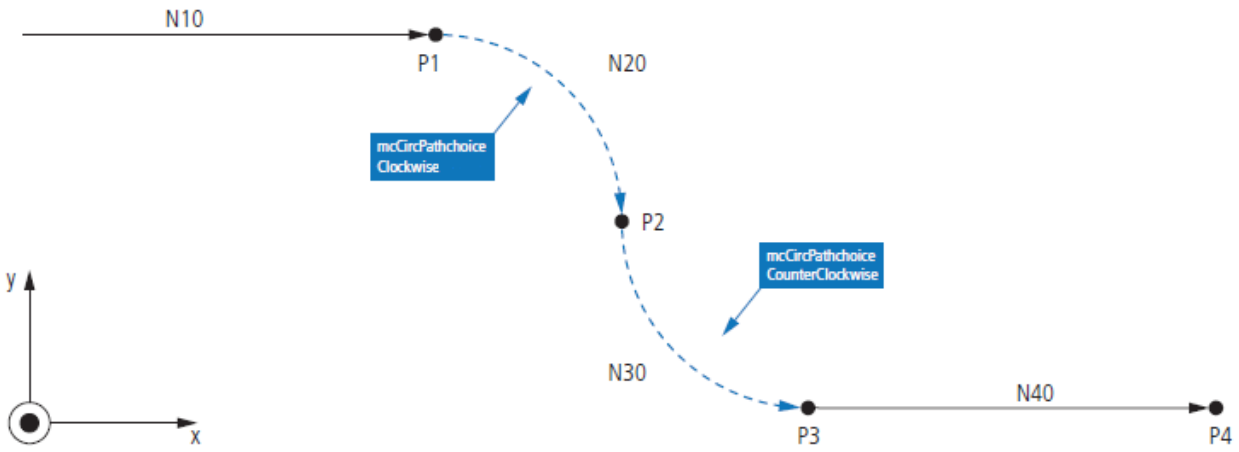
Ausgänge

```
VAR_OUTPUT
    Error                       : BOOL;
    ErrorId                     : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Beispiel Mittelpunktprogrammierung

Angenommen, eine Bahn aus 4 Segmenten wie im Bild dargestellt soll im Modus mcCircModeCenter programmiert werden: Der Benutzer definiert den Mittelpunkt des Kreises als Hilfspunkt ("AuxPoint"). Bei Verwendung von mcCircModeCenter legt der Eingang MC_CIRC_PATHCHOICE [▶ 83] die Drehrichtung fest. Da die Ebene durch das Kreuzprodukt definiert wird, muss für beide Kreissegmente N20 und N30 mcCircPathchoiceCounterClockwise ausgewählt werden.




```
VAR
    Buffer                                  : ARRAY[1..4096] OF BYTE;
    Path                                    : MC_PATH_DATA_REF (ADR(buffer), SIZEOF(buffer));
    fbMoveLinPrep                           : MC_MoveLinearAbsolutePreparation;
    fbMoveCircPrep                          : MC_MoveCircularAbsolutePreparation;

    aTargetPos                              : ARRAY[1..cAxesCount] OF MC_LREAL;
    aCircPos                                : ARRAY[1..cAxesCount] OF MC_LREAL;
    aAuxPoint                               : ARRAY[1..3] OF MC_LREAL;
    aTransitionParam                        : ARRAY[1..2] OF MC_LREAL;
END_VAR
VAR CONSTANT
    cAxesCount                              : UINT:=3;
END_VAR
```

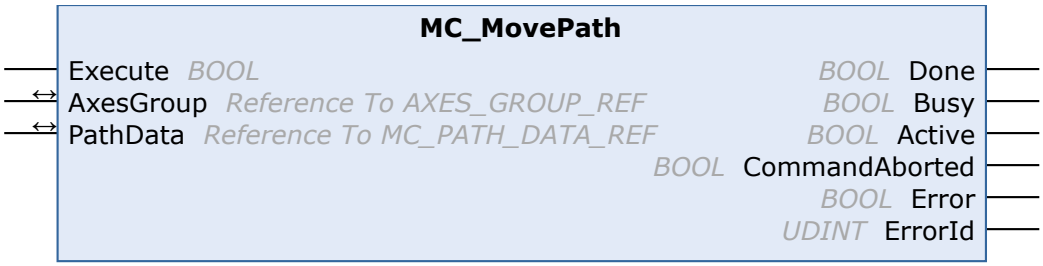
```
    fbMoveLinPrep.Position                  := ADR(aTargetPos);
    fbMoveLinPrep.PositionCount             := cAxesCount;
    fbMoveLinPrep.TransitionParameter       := ADR(aTransitionParam);
    fbMoveLinPrep.TransitionParameterCount  := 2;
    fbMoveLinPrep.BufferMode                := mcBuffered;
    fbMoveLinPrep.TransitionMode            := mcTransModeNone;

    fbMoveCircPrep.EndPoint                 := ADR(aTargetPos);
    fbMoveCircPrep.EndPointCount            := cAxesCount;
    fbMoveCircPrep.AuxPoint                 := ADR(aAuxPoint);
    fbMoveCircPrep.AuxPointCount            := 3;
    fbMoveCircPrep.CircMode                 := mcCircModeCenter;
    fbMoveCircPrep.TransitionParameter      := ADR(aTransitionParam);
    fbMoveCircPrep.TransitionParameterCount := 2;
    fbMoveCircPrep.BufferMode               := mcBuffered;
    fbMoveCircPrep.TransitionMode           := mcTransModeNone;

    aTargetPos[1]                           := 200;
    aTargetPos[2]                           := 0;
    aTargetPos[3]                           := 0;
    aTransitionParam[1]                     := 0;
    aTransitionParam[2]                     := 0;
    fbMoveLinPrep(PathData:= path, Velocity:= 3000, InvokeId:= 10);
```

```
    aTargetPos[1]                           := 300;
    aTargetPos[2]                           := -100;
    aTargetPos[3]                           := 0;
    aAuxPoint[1]                            := 200;
    aAuxPoint[2]                            := -100;
    aAuxPoint[3]                            := 0;
    aTransitionParam[1]                     := 0;
    aTransitionParam[2]                     := 0;
    fbMoveCircPrep(PathData:= path, PathChoice:= mcCircPathchoiceClockwise, Velocity:= 1000, InvokeI
d:= 20);

    aTargetPos[1]                           := 400;
    aTargetPos[2]                           := -200;
    aTargetPos[3]                           := 0;
    aAuxPoint[1]                            := 400;
    aAuxPoint[2]                            := -100;
    aAuxPoint[3]                            := 0;
    aTransitionParam[1]                     := 0;
    aTransitionParam[2]                     := 0;
    fbMoveCircPrep(PathData:= path, PathChoice:= mcCircPathchoiceCounterClockwise, Velocity:= 1000,
InvokeId:= 30);

    aTargetPos[1]                           := 600;
    aTargetPos[2]                           := -200;
    aTargetPos[3]                           := 100;
    aTransitionParam[1]                     := 0;
    aTransitionParam[2]                     := 0;
    fbMoveLinPrep(PathData:= path, Velocity:= 3000, InvokeId:= 40);
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.2.47	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.2.5 MC_MovePath



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Der Funktionsbaustein MC_MovePath führt eine Bewegung aus, die in der Tabelle PathData durch MC_MoveLinearAbsolutePreparation [▶ 65], MC_MoveCircularAbsolutePreparation [▶ 67], MC_BlockerPreparation [▶ 73] und MC_SetCoordinateTransformPreparation [▶ 55] definiert wurden.

Erneutes Auslösen einer FB-Instanz bei noch andauernder Fahrt

Es ist möglich, verschiedene Bewegungskommandos mit einer Instanz dieses Funktionsbausteins auszuführen. Die Ausgänge des Funktionsbausteins bezeichnen jedoch nur das zuletzt ausgeführte Kommando. Der Benutzer verliert die Möglichkeit der Diagnose für die zuvor gesendeten Bewegungskommandos. Das erneute Auslösen eines Funktionsbausteins wird daher nicht empfohlen.

Eingänge

```
VAR_INPUT
    Execute                 : BOOL;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.

Ein-/Ausgänge

```
VAR_IN_OUT
    AxesGroup               : AXES_GROUP_REF;
    PathData                : MC_PATH_DATA_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Gruppe von Achsen (siehe Zyklisches Gruppen-Interface [▶ 99]).
PathData	MC_PATH_DATA_REF [▶ 84]	Tabelle, die die Segmente einer Bahn enthält. Die Tabelle wird durch MC_MoveLinearAbsolutePreparation [▶ 65], MC_MoveCircularAbsolutePreparation [▶ 67], MC_BlockerPreparation [▶ 73] und MC_SetCoordinateTransformPreparation [▶ 55] geschrieben und durch MC_MovePath [▶ 71] ausgeführt.

Ausgänge

```
VAR_OUTPUT
    Done                    : BOOL;
    Busy                    : BOOL;
    Active                  : BOOL;
    CommandAborted          : BOOL;
    Error                   : BOOL;
    ErrorId                 : UDINT;
END_VAR
```

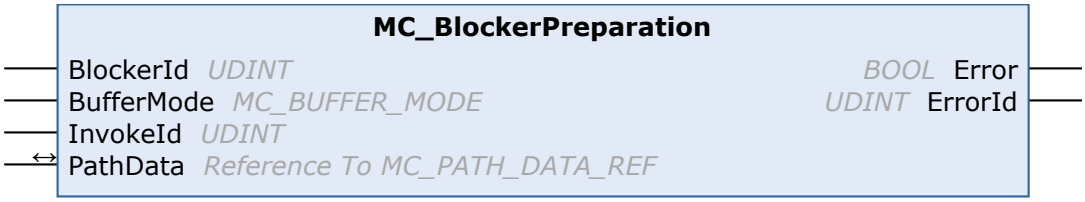
Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando erfolgreich ausgeführt worden ist. Dies bedeutet, dass das letzte durch die Referenzvariable PathData definierte Kommando erfolgreich ausgeführt wurde.
Busy	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando mit Execute gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn Busy wieder `FALSE` wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted (falls vorhanden) oder Error gesetzt.
Active	BOOL	Wenn Active `TRUE` ist, steuert der FB die Achse.
CommandAborted	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando durch ein anderes Kommando unterbrochen worden ist.
Error	BOOL	Dieser Ausgang wird `TRUE`, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.

Name	Typ	Beschreibung
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.2.6 MC_BlockerPreparation



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗	✓

Der Funktionsbaustein MC_BlockerPreparation hängt einen blockierenden Job an die Liste der Segmente in der Struktur PathData an. Die PathData-Tabelle kann über MC_MovePath ausgeführt werden. Der Funktionsbaustein MC_BlockerPreparation kann mehrmals pro Zyklus aufgerufen werden. Maximal 30 Einträge sind pro PathData-Tabelle zulässig.

Ein blockierender Job ist ein Eintrag, der die Ausführung der Bahn aussetzt, bis er mit MC_ReleaseBlocker [▶ 74] aufgelöst wird. Solange der Blocker nicht aufgelöst ist, wird die Ausführung der Bahn an diesem Segment angehalten. Jeder Blocker hat eine Id, so dass die einzelnen Blocker in der SPS unterschieden werden können.

Wenn ein blockierender Job aktiv ist, ist der Gruppenstatus immer noch "moving".
Wenn der Override geändert wird, während der blockierende Job aktiv ist, wird er für den nächsten Fahrjob wirksam.
Wenn ein neuer Job mit BufferMode mcAborting ausgeführt wird, während der blockierende Job aktiv ist, wird der blockierende Job abgebrochen.
Wenn MC_GroupHalt [▶ 61] oder MC_GroupStop [▶ 63] ausgeführt werden, während der blockierende Job aktiv ist, wird die Bahn beendet und der blockierende Job automatisch freigegeben.

Eingänge

```
VAR_INPUT
    BlockerId       : UDINT;
    BufferMode      : MC_BUFFER_MODE   := mcBuffered;
    InvokeId        : UDINT;
END_VAR
```

Name	Typ	Beschreibung
BlockerId	UDINT	Id des Blockers. Kann jeder UDINT >0 sein.

Name	Typ	Beschreibung
BufferMode	MC_BUFFER_MODE	Legt fest, wie aufeinanderfolgende Fahraufträge abgearbeitet werden sollen (siehe MC_BUFFER_MODE [▶ 88]). Hier sind nur mcBuffered und mcAborting erlaubt.
InvokeId	UDINT	Segment-ID für Analysezwecke.

Ein-/Ausgänge

```
VAR_IN_OUT
    PathData        : MC_PATH_DATA_REF;
END_VAR
```

Name	Typ	Beschreibung
PathData	MC_PATH_DATA_REF	Tabelle, die die Segmente einer Bahn enthält. Die Tabelle wird von den Preparation-Bausteinen, wie diesem, geschrieben und von MC_MovePath ausgeführt (siehe MC_PATH_DATA_REF).

Ausgänge

```
VAR_OUTPUT
    Error           : BOOL;
    ErrorId         : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4024.7 TF5400 Advanced Motion Pack V3.1.10.1	PC oder CX (x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.2.7 MC_ReleaseBlocker

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗	✓

Der Funktionsbaustein MC_ReleaseBlocker löst einen blockierenden Job auf, der die weitere Ausführung der Bahn blockiert. Ein blockierender Job wird mit MC_BlockerPreparation [▶ 73] in die Bahn eingefügt.

Mit der Blending-Strategie Superpos, bzw. ab TF5400 3.1.10.63 auch mit der Blending-Strategie GeoBlending, kann die Auflösung des Blockers vor dem Erreichen der Blocker-Position erfolgen. Verschleifungen zwischen Motion-Segmenten, die diesen Blocker umgeben, können ausgeführt werden, wenn diese Segmente dies zulassen und zum Zeitpunkt der Freigabe des blockierenden Jobs noch ausführbar sind.

Eingänge

```
VAR_INPUT
    Execute         : BOOL;
    BlockerId       : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
BlockerId	UDINT	Id des Blockers. Kann jeder UDINT >0 sein.

Ein-/Ausgänge

```
VAR_IN_OUT
    AxesGroup       : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Achsgruppe (siehe Zyklische Gruppenschnittstelle [▶ 99]).

Ausgänge

```
VAR_OUTPUT
    Done            : BOOL;
    Busy            : BOOL;
    Error           : BOOL;
    ErrorId         : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird `TRUE`, wenn das Kommando mit Execute gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn Busy wieder FAL-SE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted (falls vorhanden) oder Error gesetzt.
Error	BOOL	Dieser Ausgang wird `TRUE`, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4024.7 TF5400 Advanced Motion Pack V3.1.10.1	PC oder CX (x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.2.8 MC_GroupReadBlockerStatus

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗	✓

Der Funktionsbaustein MC_GroupReadBlockerStatus liest den aktuellen Blocker-Status.

Eingänge

```
VAR_INPUT
    Enable          : BOOL;
END_VAR
```

Name	Typ	Beschreibung
Enable	BOOL	Aktiviert das Lesen des aktuellen Blocker-Status.

Ein-/Ausgänge

```
VAR_IN_OUT
    AxesGroup       : AXES_GROUP_REF;
END_VAR
```

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Achsgruppe (siehe Zyklische Gruppenschnittstelle [▶ 99]).

Ausgänge

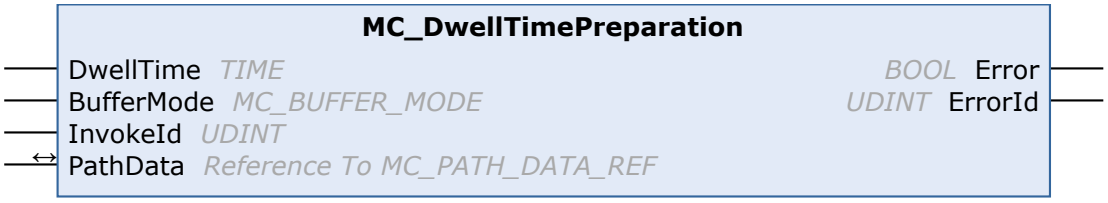
```
VAR_OUTPUT
    Valid           : BOOL;
    Blocked         : BOOL;
    BlockerId       : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Valid	BOOL	Liefert `TRUE` zurück, wenn ein gültiger Gruppentyp verwendet wird. Nur der Gruppentyp MC Group Coordinated Motion ist erlaubt.
Blocked	BOOL	Liefert `TRUE` zurück, wenn ein blockierender Job aktiv ist, d.h. die Ausführung der Bahn an-hält. Liefert `FALSE` zurück, wenn kein blockierender Job aktiv ist.
BlockerId	UDINT	Id des Blockers. Kann jeder UDINT >0 sein.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4024.7 TF5400 Advanced Motion Pack V3.1.10.1	PC oder CX (x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

7.1.1.2.9 MC_DwellTimePreparation



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗	✓

Der Funktionsbaustein MC_DwellTimePreparation hängt einen Stillstandsjob mit einer definierten Zeit an die Tabelle der Segmente in der Struktur PathData an. Die PathData-Tabelle kann über MC_MovePath ausgeführt werden. Der Funktionsbaustein MC_DwellTimePreparation kann mehrmals pro Zyklus aufgerufen werden.

Eingänge

```
VAR_INPUT
    DwellTime       : Time;
    BufferMode      : MC_BUFFER_MODE   := mcBuffered;
    InvokeId        : UDINT;
END_VAR
```

Name	Typ	Beschreibung
DwellTime	Time	Zeit, während der die Bahn mit Geschwindigkeit 0 stillsteht. Jede Zeitspanne >= 0 ist erlaubt. Eine DwellTime gleich Null führt zu einem exakten Stopp, auch wenn die umgebenden Seg-mente einen Übergang mit einer Geschwindigkeit > 0 erlauben würden.
BufferMode	MC_BUFFER_MODE	Legt fest, wie aufeinanderfolgende Fahraufträge abgearbeitet werden sollen (siehe MC_BUFFER_MODE [▶ 88]). Hier sind nur mcBuffered und mcAborting erlaubt.
InvokeId	UDINT	Segment-ID für Analysezwecke.

Ein-/Ausgänge

```
VAR_IN_OUT
    PathData        : MC_PATH_DATA_REF;
END_VAR
```

Name	Typ	Beschreibung
PathData	MC_PATH_DATA_REF	Tabelle, die die Segmente einer Bahn enthält. Die Tabelle wird von den Preparation-Bausteinen, wie diesem, geschrieben und von MC_MovePath ausgeführt (siehe MC_PATH_DATA_REF).

Ausgänge

```
VAR_OUTPUT
    Error           : BOOL;
    ErrorId         : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.

Name	Typ	Beschreibung
ErrorId	UDINT	Enthält den befehlsspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der ADS-Fehlerdokumentation oder in der NC-Fehlerdokumentation (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4024.7 TF5400 Advanced Motion Pack V3.1.10.1	PC oder CX (x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

7.1.2 Datentypen

7.1.2.1 IDENT_IN_GROUP_REF

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✓	✓

IDENT_IN_GROUP_REF definiert, wie eine Achse in einer Gruppe interpretiert wird. Für mehrdimensionale Bewegungen können globale Variablen verwendet werden. Für PTP-Collision-Avoidance-Gruppen muss die Funktion UDINT_TO_IDENTINGROUP [▶ 60] aufgerufen werden.

Verwendung ganzzahliger Werte für den Eingang IdentInGroup

Die Verwendung ganzzahliger Werte für den Eingang IdentInGroup wird NICHT unterstützt und kann zu Inkompatibilität mit künftigen Releases führen. Bei Verwendung ganzzahliger Werte kann das Projekt unter Umständen nicht mehr aufgebaut werden. Es wird empfohlen, globale Variablen [▶ 78] (z. B. MCS_X) oder die Konvertierungsfunktion UDINT_TO_IDENTINGROUP [▶ 60] zu verwenden.

Die Konstanten unten definieren Achsen als kartesische Achsen im Maschinenkoordinatensystem (MCS). A bis C definieren die Rotationsachse (C: Rotation um Z; B: Rotation um Y; A: Rotation um X). Die Zahl legt die Rotationsreihenfolge fest. Wenn beispielsweise eine Achse als MCS_C1 und eine andere als MCS_B2 definiert wird, dreht das System zuerst um die Z-Achse und als Zweites um die Y-Achse

```
VAR_GLOBAL
    MCS_X                           : IDENT_IN_GROUP_REF;
    MCS_Y                           : IDENT_IN_GROUP_REF;
    MCS_Z                           : IDENT_IN_GROUP_REF;

    MCS_A1                          : IDENT_IN_GROUP_REF;
    MCS_A2                          : IDENT_IN_GROUP_REF;
    MCS_A3                          : IDENT_IN_GROUP_REF;

    MCS_B1                          : IDENT_IN_GROUP_REF;
    MCS_B2                          : IDENT_IN_GROUP_REF;
    MCS_B3                          : IDENT_IN_GROUP_REF;

    MCS_C1                          : IDENT_IN_GROUP_REF;
    MCS_C2                          : IDENT_IN_GROUP_REF;
    MCS_C3                          : IDENT_IN_GROUP_REF;

//new from TF5400 V3.1.10.1, only compatible with MC Group Coordinated Motion
    ADDAX1                          : IDENT_IN_GROUP_REF;
    ADDAX2                          : IDENT_IN_GROUP_REF;
    ADDAX3                          : IDENT_IN_GROUP_REF;
    ADDAX4                          : IDENT_IN_GROUP_REF;

// new from TF5400 V3.2.27, only compatible with MC Group
```

```
    ADDAX5                          : IDENT_IN_GROUP_REF;
    ADDAX6                          : IDENT_IN_GROUP_REF;
    ADDAX7                          : IDENT_IN_GROUP_REF;
    ADDAX8                          : IDENT_IN_GROUP_REF;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.2.2 MC_CIRC_MODE

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Der Kreismodus bestimmt, welche Kreisdefinition zum Programmieren eines Kreises verwendet wird.

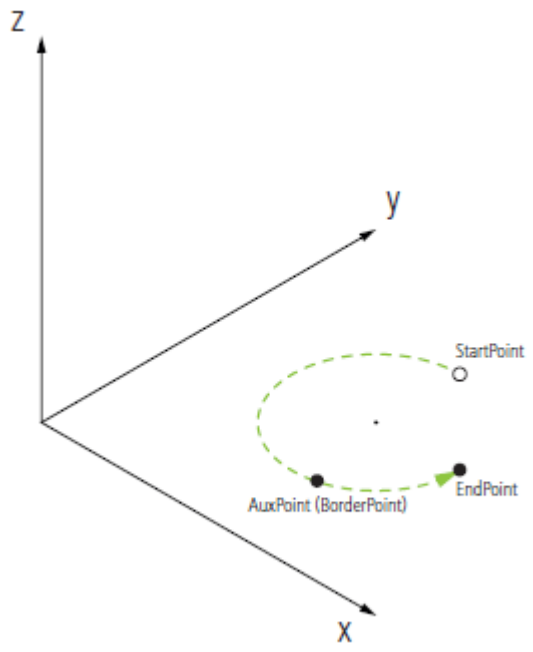
```
TYPE MC_CIRC_MODE :
(
mcCircModeInvalid        := 16#0000,
mcCircModeBorder         := 16#2000,
mcCircModeCenter         := 16#2001,
mcCircModeRadius         := 16#2002
)
END_TYPE
```

mcCircModeInvalid

Gibt Fehler zurück

- Dieser Parameter ist ungültig und führt zu einem Fehler, wenn ein gültiges MC_CIRC_MODE-Argument erforderlich ist.

mcCircModeBorder



StartPoint
- Die Bewegung beginnt am Anfangspunkt „StartPoint“.
- Dieser Punkt ist der Endpunkt des vorhergehenden Move-Kommandos.

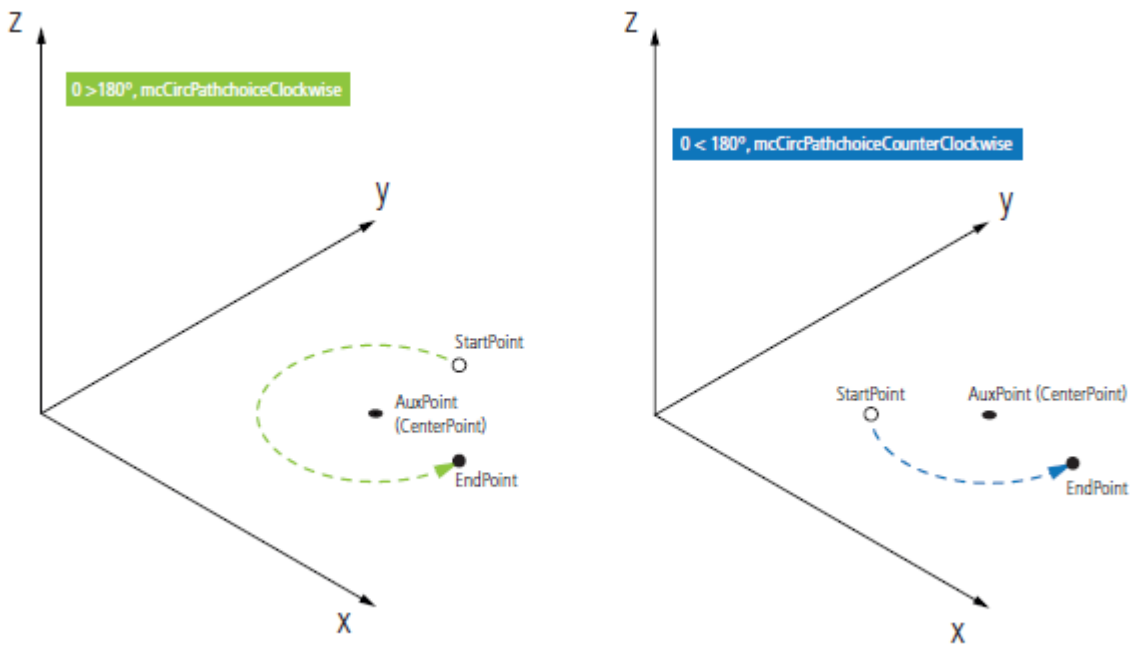
EndPoint
- Der Benutzer konfiguriert den Endpunkt "EndPoint".
- Die kreisförmige Bewegung endet an diesem Punkt.

AuxPoint
- Der Benutzer konfiguriert den Hilfspunkt "AuxPoint".
- Die kreisförmige Bewegung geht durch diesen Punkt.

PathChoice
- Der Eingangsparameter "PathChoice" und der Datentyp "MC_CIRC_PATHCHOICE" werden ignoriert.

Anwendbarkeit
- Der Modus mcCircModeBorder kann nicht verwendet werden, um einen Vollkreis zu beschreiben (d. h. „StartPoint“ gleich „EndPoint“). Dies ist darauf zurückzuführen,dass dabei der Mittelpunkt des Kreises nicht eindeutig wäre.
- Der Modus *mcCircModeBorder* kann nicht verwendet werden, um Bahnen mit mehr als einer vollen Umdrehung des Kreises zu beschreiben.

mcCircModeCenter



StartPoint

- Die Bewegung beginnt am Anfangspunkt „StartPoint“.
- Dieser Punkt ist der Endpunkt des vorhergehenden Fahrbefehls.

EndPoint

- Der Benutzer konfiguriert den Endpunkt "EndPoint".
- Die kreisförmige Bewegung endet an diesem Punkt.

AuxPoint

- Der Benutzer konfiguriert den Hilfspunkt "AuxPoint".
- Für die Kreisbewegung fungiert dieser Hilfspunkt als Kreismittelpunkt.
- Der Mittelpunkt muss den gleichen Abstand zum "StartPoint" und "EndPoint" haben. Wenn die Abstände nur geringfügig voneinander abweichen, wird der Mittelpunkt angepasst. Wenn die Abstände erheblich voneinander abweichen, wird die Kreisbeschreibung nicht akzeptiert.

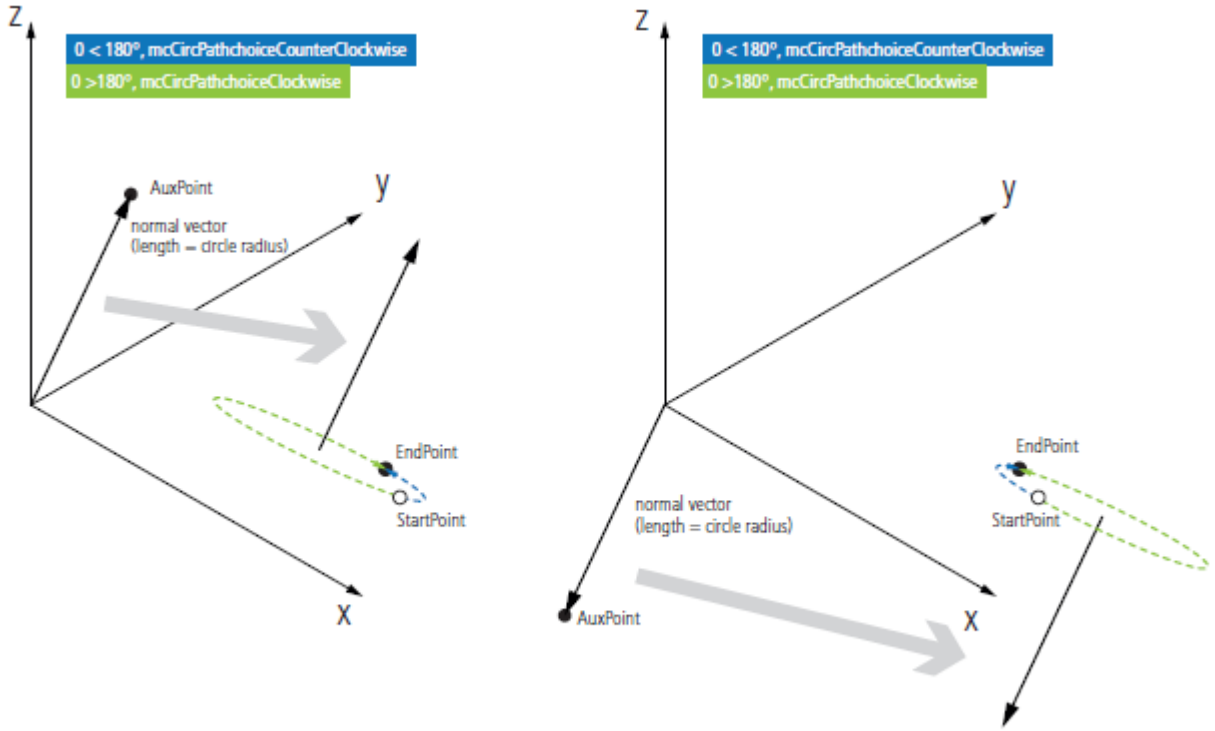
PathChoice

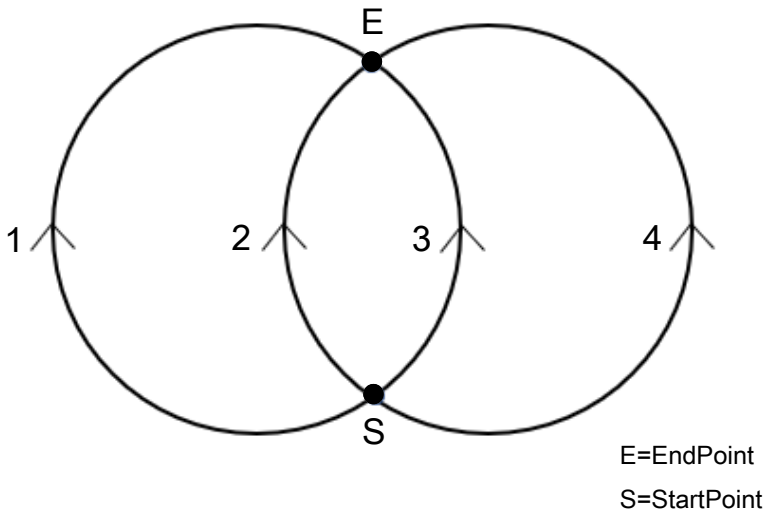
- Es gibt normalerweise zwei mögliche Kreisbögen, die vom Anfangspunkt "StartPoint" zum Endpunkt "EndPoint" durchlaufen werden können. Der "PathChoice"-Parameter macht die beiden eindeutig. Siehe MC_CIRC_PATHCHOICE für weitere Informationen.

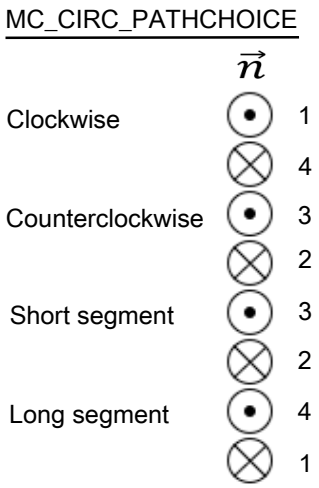
Anwendbarkeit

- Der Modus mcCircModeCenter kann nicht verwendet werden, um einen Halbkreis (d.h. einen Bogen, der einen Winkel von 180° oder sehr nahe daran durchläuft) oder einen Vollkreis (d.h. "StartPoint" ist gleich "EndPoint") zu beschreiben. Dies ist darauf zurückzuführen, dass in diesen Fällen Start-, Mittel- und Endpunkt kollinear wären und somit die Ebene, in der der Kreis liegt, nicht eindeutig wäre.
- Der Modus mcCircModeCenter kann nicht verwendet werden, um Bahnen mit mehr als einer vollen Umdrehung des Kreises zu beschreiben.

mcCircModeRadius







Bilder

- Vier verschiedene Bögen werden durch die Orientierung des Normalenvektors und dem Parameter "PathChoice" unterschieden.

StartPoint

- Die Bewegung beginnt am Anfangspunkt "StartPoint".
- Dieser Punkt ist der Endpunkt des vorhergehenden Move-Kommandos.
- Der zu konstruierende Kreis und seine Ebene enthalten den Anfangspunkt.

AuxPoint

Normalvektor

- Der Benutzer konfiguriert den Parameter "AuxPoint", der in diesem Modus als Normalenvektor der Kreisebene fungiert. Seine Länge gibt den Radius des Kreises an.

EndPoint

- Der Benutzer konfiguriert den Endpunkt "EndPoint".
- Die Bewegung wird an diesem Punkt enden.
- MC-Gruppe nur mit Pick-And-Place: Wenn dieser Punkt außerhalb der durch "StartPoint" und den Normalenvektor definierten Ebene liegt, folgt die Bewegung einer Helix anstelle eines Kreises.

PathChoice und resultierender Bogen

- Die Rechte-Hand-Regel wird für alle "PathChoice"-Werte angewendet, mit Ausnahme von mcCircPathchoiceClockwise, das der Linken-Hand-Regel folgt.
- *mcCircPathchoiceCounterClockwise* und *mcCircPathchoiceShortSegment* beschreiben einen Bogen, der einen Winkel <= 180° abdeckt, *mcCircPathchoiceClockwise* und *mcCircPathchoiceLongSegment* beschreiben einen Bogen, der einen Winkel >= 180° abdeckt.
- Welcher der 4 möglichen Bögen mit einem gegebenen Radius gewählt wird, hängt vom "PathChoice"-Argument und von der Orientierung des Normalenvektors ab. Siehe obige Tabelle für weitere Informationen.

Anwendbarkeit

- Der Modus mcCircModeRadius kann nur zur Beschreibung von Bögen verwendet werden, die einen Winkel < 360° abdecken.
- Die Länge des Normalenvektors (d.h. der Radius des Kreises) muss mindestens die Hälfte der Entfernung zwischen Start- und Endpunkt betragen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.2.47	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.2.3 MC_CIRC_PATHCHOICE

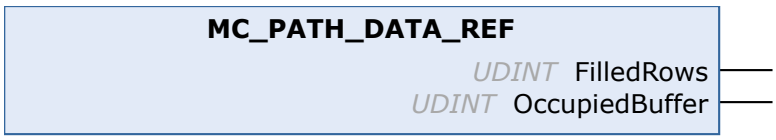
TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Der Datentyp MC_CIRC_PATHCHOICE definiert die Drehrichtung eines Kreises, falls mcCircModeCenter oder mcCircModeRadius aus der Aufzählung MC_CIRC_MODE [▶ 79] ausgewählt wird.

```
TYPE MC_CIRC_PATHCHOICE :
(
    mcCircPathchoiceClockwise          := 16#3000,
    mcCircPathchoiceCounterClockwise   := 16#3001,

//new from TF5400 V3.1.10.1
    mcCircPathchoiceShortSegment       := 16#3002,
    mcCircPathchoiceLongSegment        := 16#3003
);
END_TYPE
```

Name	Typ	Beschreibung
mcCircPathchoiceClockwise	INT	stellt das Kreissegment mit einem Winkel >180° dar.
mcCircPathchoiceCounterClockwise	INT	stellt das Kreissegment mit einem Winkel <180° dar.
mcCircPathchoiceShortSegment	INT	stellt das Kreissegment mit dem kleineren Winkel dar.
mcCircPathchoiceLongSegment	INT	stellt das Kreissegment mit dem größeren Winkel dar.

7.1.2.4 MC_PATH_DATA_REF



TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

MC_PATH_DATA_REF stellt die durch MC_MovePath [▶ 71] auszuführende Bahn dar, wobei die Anzahl der Einträge auf 30 begrenzt ist. Die auszuführende Bahn wird durch MC_MoveLinearAbsolutePreparation [▶ 65], MC_MoveCircularAbsolutePreparation [▶ 67] und MC_BlockerPreparation [▶ 73] geschrieben. Sie wird mit einem Zeiger auf einen benutzerdefinierten Puffer initialisiert. Hierbei kann der Benutzer die Größe der Bahn definieren. Die Initialisierung muss während der Deklaration erfolgen. Die Bahntabelle wird bei der Ausführung nicht zurückgesetzt. Zum Zurücksetzen muss die Methode ClearPath [▶ 85] aufgerufen werden.

Ausgänge

```
VAR_OUTPUT
    FilledRows            : UDINT;
    OccupiedBuffer        : UDINT;
END_VAR
```

Name	Typ	Beschreibung
FilledRows	UDINT	Anzahl der Bahneinträge (z. B. Bahnsegmente).
OccupiedBuffer	UDINT	Belegte Puffergröße in Byte. Durch Analysieren dieses Ausgangs kann der Benutzer prüfen, ob das Ende des definierten Puffers erreicht wird.

Beispiel

Das Beispiel unten zeigt, wie eine Bahnreferenz deklariert wird und wie eine bestehende Bahn zurückgesetzt wird.

```
VAR
    buffer          : ARRAY[1..4096] OF BYTE;
    Path            : MC_PATH_DATA_REF(ADR(buffer), SIZEOF(buffer));
END_VAR
```

```
//delete all segments of path table
Path.ClearPath();
```

Der Datentyp `MC_PATH_DATA_REF` ist Teil der Motion Control (MC)-Bibliothek. Verwenden Sie die Methode `ClearPath()`, um Bahninformationen vom Typ `MC_PATH_DATA_REF` zu löschen und so eine bestehende Bahn zurückzusetzen. Verwenden Sie beim Datentyp `MC_PATH_DATA_REF` nur Motion Control-Funktionen oder Motion Control-Funktionsbausteine. Verwenden Sie beim Datentyp `MC_PATH_DATA_REF` insbesondere keine Speicherfunktionen wie `MEMCMP`, `MEMCPY`, `MEMSET` oder `MEMMOVE`.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.1.2.4.1 ClearPath

ClearPath

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Die Methode ClearPath setzt die durch MC_PATH_DATA_REF dargestellte Bahn zurück. Die Bahntabelle wird bei der Ausführung nicht automatisch zurückgesetzt.

7.1.2.5 MC_TRANSITION_MODE

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Der Transitionsmodus charakterisiert, wie ein Segmentübergang ausgeführt wird.

```
TYPE MC_TRANSITION_MODE :
(
    mcTransModeNone                          := 16#1000,
    mcTransModeStartVelocity                 := 16#1001,
    mcTransModeConstantVelocity              := 16#1002,
    mcTransModeCornerDistance                := 16#1003,
    mcTransModeMaxCornerDeviation            := 16#1004,
    mcTransModeCornerDistanceAdvanced        := 16#100A
);
END_TYPE
```

Die folgende Tabelle zeigt eine Übersicht der implementierten Transitionsmodi und der Anzahl der Parameter, die in TransitionParameterCount definiert werden müssen.

Name	TransitionParameterCount	Beschreibung
mcTransModeNone	Keine Auswirkung	Kein Blending
mcTransModeCornerDistance **nicht kompatibel mit MC Group with Pick-and-Place, verfügbar ab TF5400 V3.1.10.1**	1	Transitionsparameter fungieren als Toleranzkugel, in der die Bahn verlassen werden darf.
mcTransModeCornerDistanceAdvanced	2	TransitionParameter fungieren als Toleranzkugel, in der die Bahn verlassen werden darf.

mcTransModeNone

Es wird kein Blending ausgeführt. Stopp am Segmentübergang.

mcTransModeCornerDistance

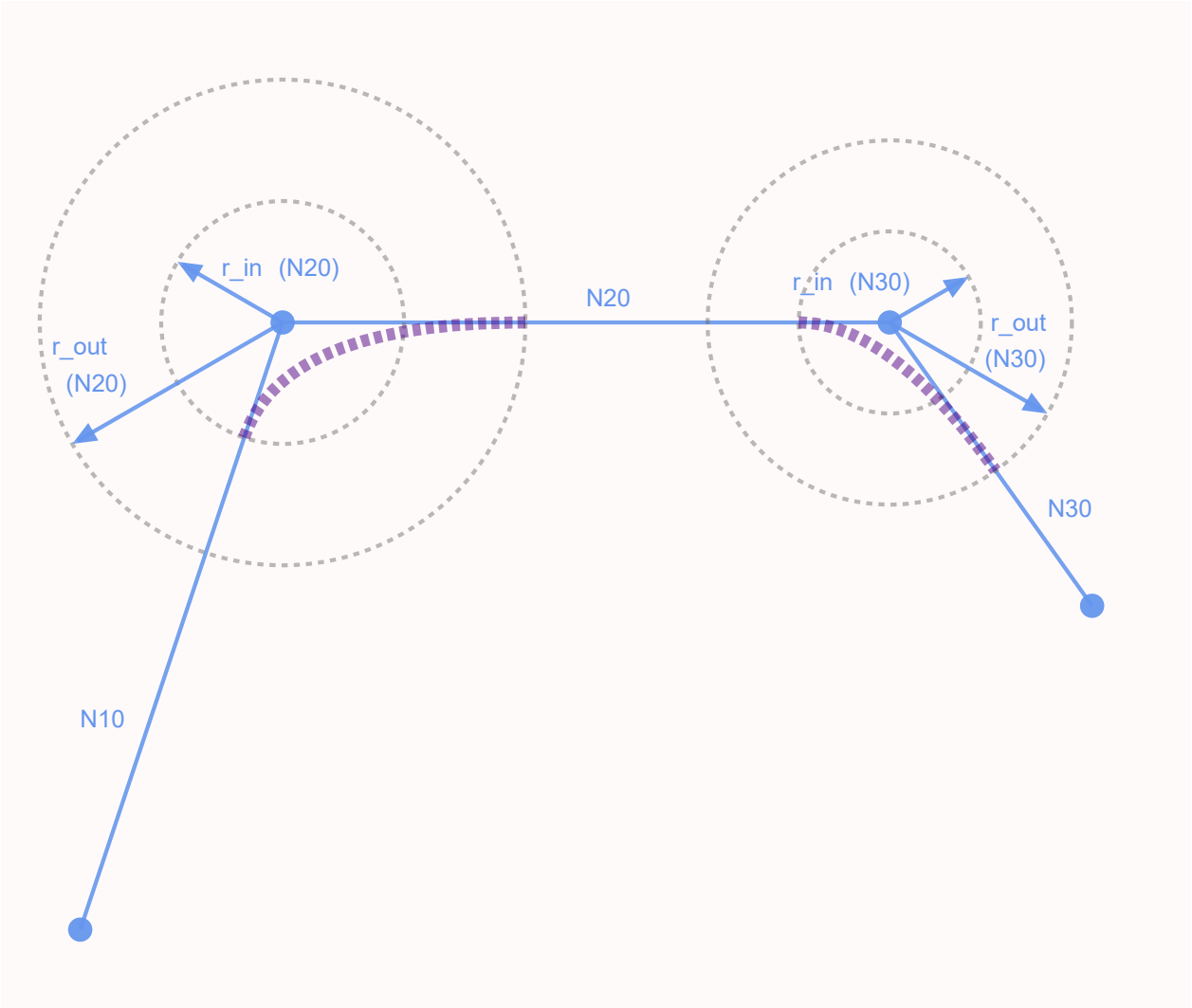
Blending wird zwischen den Segmenten ausgeführt. Die Transitionsparameter fungieren als Toleranzkugel, in der der programmierten Bahn nicht gefolgt wird. Der Parameter beschreibt den Radius am vorherigen und zweiten Segment, bei dem das Blending beginnt und endet.

Dieser Modus ist nur mit MC Group Coordinated Motion kompatibel.

mcTransModeCornerDistanceAdvanced

Blending wird zwischen den Segmenten ausgeführt. Die Transitionsparameter fungieren als Toleranzkugel, in der der programmierten Bahn nicht gefolgt wird. Der erste Parameter beschreibt den Radius am vorherigen Segment, bei dem das Blending beginnt (r_in). Der zweite Parameter beschreibt den Radius am folgenden Segment (r_out), der eine Position definiert, für die das Blending garantiert erfolgt. Der Parameter r_out ist ein Maximalwert. Das Blending kann enden, bevor r_out erreicht ist.

Das Blending (r_in) ist mit der MC Group with Pick-and-Place auf 90 % des vorherigen Segments begrenzt. r_out ist nicht begrenzt.



Empfohlenes Verhältnis der Transitionsparameter für Blending mit MC Group with Pick-and-Place

Die Graphik zeigt eine Planarbewegung im zweidimensionalen Raum. An dieser Bewegung sollen zwei Achsen beteiligt sein. Unter der Annahme, dass die beteiligten Achsen eine ähnliche Dynamik aufweisen, sollte `r_out` mindestens `2 * r_in` betragen.

Kombinationen von Puffermodus und Transitionsmodus

Puffermodus und Transitionsmodus werden nur bei Verwendung von TF5420 kombiniert.

Die folgende Tabelle zeigt die möglichen Kombinationen von Transitionsmodus und Puffermodus und ihre Wirkung.

TM/PM	mcAborting	mcBuffered	mcBlendingPrevious	Sonstige
mcTransModeNone	Das vorherige Kommando wird sofort abgebrochen. Eine neue Bewegung wird gestartet. Die Geschwindigkeit im Übergang ist 0. Diese Kombination ist nur für das 1. Segment einer Bahn zulässig.	Stopp am Ende des vorherigen Kommandos. Anschließend wird das nächste Kommando ausgeführt.	Nicht zulässig	Nicht zulässig
mcTransModeCornerDistance **Neu ab TF5400 V3.1.10.1, nur mit MC Group Coordinated Motion kompatibel**	Blending vom aktiven Segment zum ersten Segment des neuen Kommandos. Der Schnittpunkt der Segmente wird durch die Strecke definiert, die für den Stopp auf dem aktiven Segment benötigt wird. Diese Kombination ist nur für das 1. Segment einer Bahn zulässig.	Nicht zulässig	Blending vom letzten programmierten Kommando zum neuen Kommando	Nicht zulässig
mcTransModeCornerDistanceAdvanced	Blending vom aktiven Segment zum ersten Segment des neuen Kommandos. Der Schnittpunkt der Segmente wird durch die Strecke definiert, die für den Stopp auf dem aktiven Segment benötigt wird. Diese Kombination ist nur für das 1. Segment einer Bahn zulässig.	Nicht zulässig	Blending vom letzten programmierten Kommando zum neuen Kommando	Nicht zulässig
Sonstige	Nicht zulässig	Nicht zulässig	Nicht zulässig	Nicht zulässig

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TF5400 Advanced Motion Pack V3.1.1.17		

7.1.2.6 MC_COORD_REF

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✗	✗ (✓ bis v3.2 einschließlich)	✓

Objekt-ID, die sich auf einen Node Connector bezieht.

7.2 Tc3_Mc3Definitions

Strukturen und Aufzählungen

Name	Beschreibung	TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
			MC Group with Pick-and-Place	MC Group Coordinated Motion
MC_BUFFER_MODE [▶ 88]	Legt fest, wie aufeinanderfolgende Fahrkommandos zu verarbeiten sind.	✓	✓	✓
MC_COMPENSATION_TYPE [▶ 93]	Der Wert definiert den Kompensationstyp.	✓	✗	✗
MC_DIRECTION [▶ 93]	Der Wert bestimmt die Richtung der Bewegung.	✓	✗	✗
MC_SYNC_MODE [▶ 94]	Der Wert definiert die Richtung, in die synchronisiert werden soll.	✓	✗	✗
MC_SYNC_STRATEGY [▶ 94]	Definiert das Synchronisationsprofil der Slave-Achse.	✓	✗	✗

7.2.1 Datentypen

7.2.1.1 MC_BUFFER_MODE

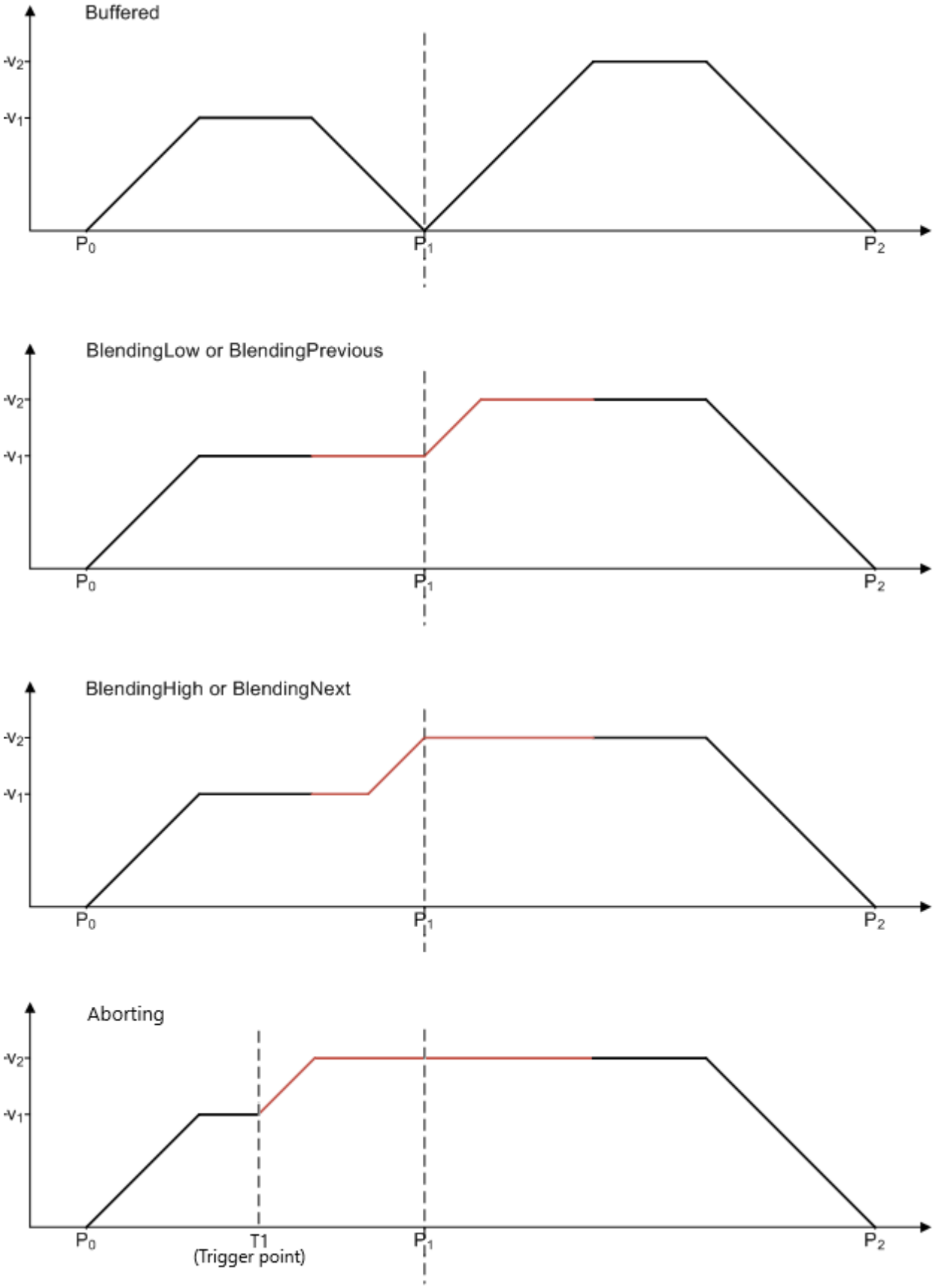
Der Datentyp MC_BUFFER_MODE wird verwendet, um festzulegen, wie aufeinanderfolgende Fahrkommandos abgearbeitet werden sollen. Damit der Puffermodus eine Wirkung hat, sind mindestens zwei Funktionsbausteine erforderlich.

```
TYPE MC_BUFFER_MODE :
(
    mcAborting                          := 16#0,
    mcBuffered                          := 16#1,
    mcBlendingLow                       := 16#12,
    mcBlendingPrevious                  := 16#13,
    mcBlendingNext                      := 16#14,
    mcBlendingHigh                      := 16#15
) UINT;
END_TYPE
```

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✔	✔

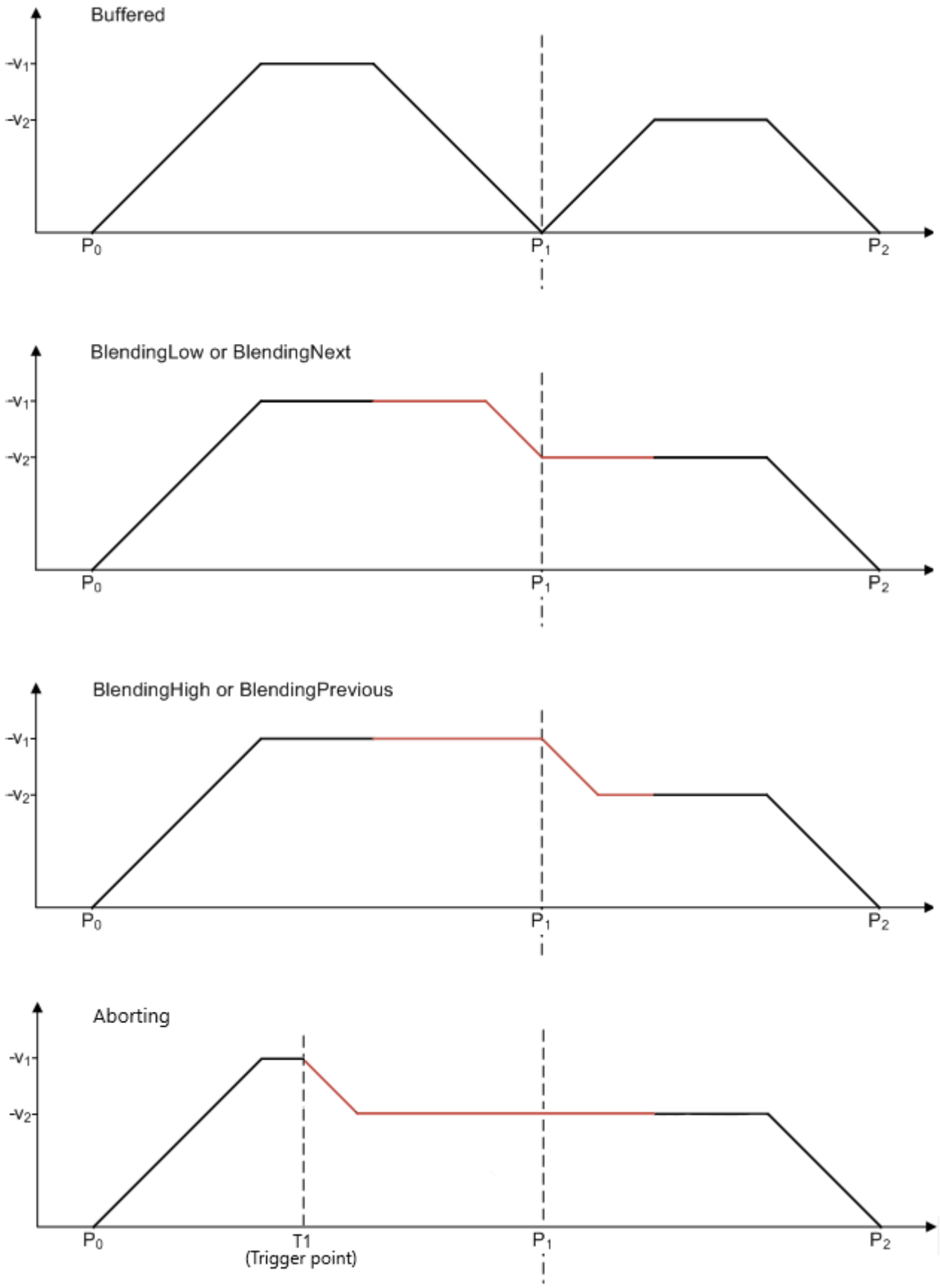
Beispiel:

Im folgenden Beispiel wird ein Move-Kommando verwendet, um eine Gruppe von Position P_0 zu P_1 und dann zu P_2 zu bewegen. Der Referenzpunkt für die verschiedenen Geschwindigkeitsprofile ist immer P_1. Der Modus legt die Geschwindigkeit v_1 oder v_2 an diesem Punkt fest.



Da die Geschwindigkeit des ersten Kommandos niedriger ist als die des zweiten, führen die Modi BlendingLow/BlendingPrevious und BlendingHigh/BlendingNext jeweils zum selben Ergebnis.

Wenn die Geschwindigkeit des zweiten Kommandos niedriger ist als die des ersten, sind die Modi BlendingLow/BlendingPrevious und BlendingHigh/BlendingNext jeweils gleichwertig.



Kombinationen von Puffermodus und Transitionsmodus

Hinweis **Puffermodus und Transitionsmodus werden nur bei Verwendung von TF5420 kombiniert.**

Die folgende Tabelle zeigt die möglichen Kombinationen von Transitionsmodus und Puffermodus und ihre Wirkung.

TM/PM	mcAborting	mcBuffered	mcBlendingPre-vious	Sonstige
mcTransModeNone	Das vorherige Kommando wird sofort abgebrochen. Eine neue Bewegung wird gestartet. Die Geschwindigkeit im Übergang ist 0. Diese Kombination ist nur für das 1. Segment einer Bahn zulässig.	Stopp am Ende des vorherigen Kommandos. Anschließend wird das nächste Kommando ausgeführt.	Nicht zulässig	Nicht zulässig
mcTransModeCornerDistance **Neu ab V3.1.10.1, nur mit MC Group Coordinated Motion kompatibel**	Blending vom aktiven Segment zum ersten Segment des neuen Kommandos. Der Schnittpunkt der Segmente wird durch die Strecke definiert, die für den Stopp auf dem aktiven Segment benötigt wird. Diese Kombination ist nur für das 1. Segment einer Bahn zulässig.	Nicht zulässig	Blending vom letzten programmierten Kommando zum neuen Kommando	Nicht zulässig
mcTransModeCornerDistanceAdvanced	Blending vom aktiven Segment zum ersten Segment des neuen Kommandos. Der Schnittpunkt der Segmente wird durch die Strecke definiert, die für den Stopp auf dem aktiven Segment benötigt wird. Diese Kombination ist nur für das 1. Segment einer Bahn zulässig.	Nicht zulässig	Blending vom letzten programmierten Kommando zum neuen Kommando	Nicht zulässig
Sonstige	Nicht zulässig	Nicht zulässig	Nicht zulässig	Nicht zulässig

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

7.2.1.2 MC_COMPENSATION_TYPE

Der Datentyp MC_COMPENSATION_TYPE wird verwendet, um festzulegen, mit welchem Kompensationstyp verfahren werden soll.

```
TYPE MC_COMPENSATION_TYPE:
(
    mcTypeInvalidCompensation            := 16#0,
    mcTypeGeoCompensation                := 16#1
)UINT;
END_TYPE
```

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘	✘

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.6.07	PC oder CX (x64)	Tc3_McCompensations

7.2.1.3 MC_DIRECTION

```
(* Defines the direction of the movement (e.g. for a modulo axis). *)
TYPE MC_DIRECTION :
(
    mcDirectionNonModulo        := 0, (* Position is interpreted as absolute position. *)
    mcDirectionPositive         := 1, (* Moves in positive direction. *)
    mcDirectionShortestWay      := 2, (* The direction of movement depends on whether the positive
direction of movement or the negative direction of movement is the shortest distance from the target
position. *)
    mcDirectionNegative         := 3 (* Moves in negative
                                        direction. *)
)
END_TYPE
```

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘	✘

MC_DIRECTION wird verwendet, um die Bewegungsrichtung bei der Modulo-Positionierung anzugeben. Die Modulo-Positionierung ist nur bei periodischen Systemen anwendbar. Bei offenen Systemen, wie z. B. offene Tracks, wird nur der Wert `mcDirectionNonModulo` akzeptiert.

mcDirectionNonModulo: Die Position wird immer als absolute Position interpretiert.

mcDirectionPositive: Positive Bewegungsrichtung

mcDirectionNegative: Negative Bewegungsrichtung

mcDirectionShortestWay: Die Bewegungsrichtung hängt davon ab, ob die positive Bewegungsrichtung oder die negative Bewegungsrichtung den kürzesten Abstand zur Zielposition hat.

> In Kombination mit der Tc2_MC2 oder Tc3_Mc3Definitions Bibliothek kann es sein, dass der Datentyp nicht eindeutig aufgelöst werden kann (ambiguous use of name 'MC_Direction'). Dann muss jeweils der Namespace bei Verwendung des Datentyps mit angegeben werden (Tc3_Mc3PlanarMotion.MC_DIRECTION bzw. Tc3_Mc3Definitions.MC_DIRECTION bzw. Tc2_MC2.MC_DIRECTION).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4024.7 TF5400 Advanced Motion Pack V3.1.10.1	PC oder CX (x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

7.2.1.4 MC_SYNC_MODE

```
(* Defines the direction of the synchronization position of modulo axes. *)
TYPE MC_SYNC_MODE :
(
    mcSyncModeNonModulo          := 0, (* SyncSlavePosition is interpreted as absolute position. *)
    mcSyncModePositive           := 1, (* Synchronizes in positive direction. *)
    mcSyncModeNegative           := 3 (* Synchronizes in negative direction. *)
)
END_TYPE
```

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘	✘

Der Wert definiert die Richtung, in die synchronisiert werden soll. Die Angabe des SyncMode ist nur dann wirksam, wenn für die Achse ein Modulo-Koordinatensystem definiert wurde. Dies kann z.B. ein geschlossener XTS Track sein oder eine geschlossene CA-Gruppe. Wenn es nur eine mathematische Lösung gibt, um die Synchronposition zu erreichen, wird der Wert ignoriert.

mcSyncModeNonModulo: Die SlaveSyncPosition wird immer als absolute Position interpretiert.

mcSyncModePositive: Die Slave Achse synchronisiert sich in positive Bewegungsrichtung.

mcSyncModeNegative: Die Slave Achse synchronisiert sich in negativer Bewegungsrichtung.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4024.7 TF5400 Advanced Motion Pack V3.1.10.1	PC oder CX (x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

7.2.1.5 MC_SYNC_STRATEGY

Der Datentyp MC_SYNC_STRATEGY definiert das Synchronisationsprofil des Slaves, z. B. für ein MC_GearInPosCA-Kommando.

```
TYPE MC_SYNC_STRATEGY :
(
    mcSyncStrategyLate               := 16#1,
    mcSyncStrategySlow               := 16#2,
    mcSyncStrategyEarly              := 16#3
)
END_TYPE
```

TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
	MC Group with Pick-and-Place	MC Group Coordinated Motion
✔	✘	✘

Beispiele:

Die Rahmenbedingungen in den folgenden Beispielen sind gleich:

- Die Master-Bewegung ist gleich.
- Die MasterStartDistance ist gleich.
- Die Strecken (MasterSyncPosition – aktuelle Master-Position) und (SlaveSyncPosition – aktuelle Slave-Position) sind in allen drei Beispielen gleich.
- Die Slave-Dynamik ist gleich.
- Konfiguration mit einer Achse in der CA-Gruppe, eine PTP-Achse als Master.
- Ein Bewegungskommando wird an den Master ausgegeben.

Beispiel 1: mcSyncStrategyLate

Der Slave beginnt mit der Synchronisation so spät wie möglich und mit voller Dynamik (gemäß den Eingangswerten Geschwindigkeit, Beschleunigung, Verzögerung, Ruck). Er erreicht die SlaveSyncPosition gerade rechtzeitig mit dem richtigen Getriebefaktor. Der Benutzer muss darauf achten, dass der Master nicht beschleunigt, sobald der Slave StartSync signalisiert, da das Synchronisationsprofil bereits mit der maximalen Slave-Dynamik geplant ist. Der Slave kann nicht gegen seine dynamischen Beschränkungen verstoßen und daher eine Beschleunigung des Masters nicht ausgleichen. Diese Situation führt zu einem Fehler am Funktionsbaustein.

1. Geben Sie das Kommando MC_GearInPosCA an die Achse aus. Das Kommando wird aktiv, während der Master noch beschleunigt.

⇨ Der Slave beginnt mit der Synchronisation so spät wie möglich und mit voller Dynamik und hat die SlaveSyncPosition erreicht, wenn der Master die MasterSyncPosition erreicht hat (schwarzer x-Cursor).

Beispiel 2: mcSyncStrateySlow

Der Slave beginnt seine Synchronisation in Bewegung, sobald der Master (MasterSyncPosition – MasterStartDistance) in der richtigen Richtung passiert, wenn eine MasterStartDist gesetzt wurde, anderenfalls sobald der Funktionsbaustein Active ist. Die Dynamik des Slaves wird so reduziert, dass der

Slave die SlaveSyncPos mit dem richtigen Getriebefaktor gerade rechtzeitig erreicht, wenn der Master die MasterSyncPos erreicht. Der Slave kann eine Beschleunigung des Masters ausgleichen, wenn auch StartSync gesetzt ist, jedoch nur, bis der Slave seine maximale Dynamik erreicht.

1. Geben Sie das Kommando MC_GearInPosCA an die Achse aus. Das Kommando wird aktiv, während der Master noch beschleunigt.

⇨ Der Slave beginnt mit der Synchronisation, sobald MC_GearInPosCA Active ist. Die Dynamik wird so reduziert, dass der Slave die SlaveSyncPosition zur gleichen Zeit erreicht, wie der Master die MasterSyncPosition erreicht (schwarzer x-Cursor).

Eine Synchronisation auf einen stehenden Master kann bei Verwendung von mcSyncStrategySlow zu einer hohen Last führen.

In diesem Fall wird am besten mcSyncStrategyEarly verwendet.

Beispiel 3: mcSyncStrategyEarly

Der Slave beginnt sofort (wenn eine MasterStartDistance gesetzt ist: sofort, nachdem diese zurückgelegt wurde) und mit voller Dynamik mit der Synchronisation. Der Slave signalisiert InSync früher als von der SlaveSyncPosition gefordert, aber es ist dennoch garantiert, dass der geforderte Offset zwischen Master und Slave (MasterSyncPosition – SlaveSyncPosition) mit dem richtigen Getriebefaktor erreicht wird. Diese Strategie kann für die Synchronisation auf einen stehenden Master angewendet werden und ist am besten geeignet, wenn die Geschwindigkeit des Masters nicht konstant ist. Der Slave versucht fortwährend, die Synchronisation durchzuführen. Wenn die Rahmenbedingungen es dem Slave nicht ermöglichen, an der SlaveSyncPosition InSync zu sein, führt dies nicht zu einem Fehler, sondern der Slave versucht kontinuierlich, sich mit dem Master zu synchronisieren.

1. Geben Sie das Kommando MC_GearInPosCA an die Achse aus. Das Kommando wird aktiv, während der Master noch beschleunigt.

⇨ Der Slave beginnt mit der Synchronisation, sobald MC_GearInPosCA Active ist, und mit voller Dynamik. Der Slave ist schnellstmöglich InSync, erreicht die SlaveSyncPosition aber zur gleichen Zeit, wie der Master die MasterSyncPosition erreicht (schwarzer x-Cursor).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x64)	Tc3_McCoordinatedMotion, Tc2_MC2

8 Beispiele

Mehrdimensionale Bewegung

PnpSimpleSample

Download: https://infosys.beckhoff.com/content/1031/TF5420_TC3_Advanced_Pick_And_Place/Resources/9725595531.zip

Beschreibung:
Projekt, das eine einfache Pick-and-Place-Bewegung ausführt.

PnPSimpleSample mit einer zusätzlichen Achse und Verwendung von Blockern

Download: https://infosys.beckhoff.com/content/1031/TF5420_TC3_Advanced_Pick_And_Place/Resources/9725597195.zip

Beschreibung:
Erweiterung des PnpSimpleSample-Projektes um eine zusätzliche Achse und einen blockierenden Job.

9 Anhang

9.1 Zyklische Gruppenschnittstelle

Die zyklische Gruppenschnittstelle sorgt für den zyklischen Datenaustausch zwischen SPS und einem NC-Gruppenobjekt. Die Gruppenschnittstelle beinhaltet die Richtungen NcToPlc [▶ 99] und PlcToNc [▶ 100]. Beide Richtungen sind in allgemeine und gruppenspezifische Daten unterteilt.

AXES_GROUP_REF

```
TYPE AXES_GROUP_REF :
STRUCT
    PlcToNc AT %Q*        : CDT_PLCTOMC_GROUP;
    NcToPlc AT %I*        : CDT_MCTOPLC_GROUP;
END_STRUCT
END_TYPE
```

PlcToNc: PlcToNc [▶ 100] ist eine Datenstruktur, die zyklisch zwischen SPS und NC ausgetauscht wird. Über diese Datenstruktur kommunizieren die MC-Funktionsbausteine mit der Bewegungsgruppe und senden Steuerinformationen von der SPS an die NC. Diese Datenstruktur wird automatisch im Ausgangsprozessabbild der SPS platziert und muss mit dem Eingangsprozessabbild einer Bewegungsgruppe verbunden werden.

NcToPlc: NcToPlc [▶ 99] ist eine Datenstruktur, die zyklisch zwischen SPS und NC ausgetauscht wird. Über diese Datenstruktur kommunizieren die MC-Funktionsbausteine mit der NC und erhalten Statusinformationen von der NC. Diese Datenstruktur wird automatisch im Eingangsprozessabbild der SPS platziert und muss im TwinCAT System Manager mit dem Ausgangsprozessabbild einer NC-Achse verbunden werden.

9.1.1 NcToPlc

Datenstruktur, die zyklisch zwischen NC und PLC ausgetauscht wird. Einige Daten sind gruppenspezifisch.

Name	Typ	Beschreibung
Common		
GroupOID	OTCID	TcCOM-Objekt-ID (OID) dieser Gruppe
GroupType	MC_GROUP_TYPE [▶ 102]	Typ der Gruppe
GroupStatus	ST_GroupStatus [▶ 103]	Statusinformationen zur Gruppe
State	MC_GROUP_STATE [▶ 102]	Zustand der Gruppe, siehe Zustandsdiagramm einer Gruppe.
Flags	ST_GroupStatusFlags [▶ 103]	Zusätzliche Statusinformationen
IsEnableRequested	BIT	Definiert, ob eine Aktivierung oder Deaktivierung einer Gruppe angefordert wird.
AllAxesStanding	BIT	Gibt an, ob alle Achsen physikalisch stillstehen, d. h. die Geschwindigkeit und Beschleunigung bei allen Achsen null ist, unabhängig davon, ob ein Bewegungsbefehl aktiv ist.
InPosition	BIT	Gibt an, ob sich die Gruppe in Position befindet.
GroupErrorId	UDINT	Fehlercode im Falle eines Gruppenfehlers

Name		Typ	Beschreibung
	GroupAxesCount	UDINT	Anzahl der Achsen, die gegenwärtig zu dieser Gruppe gehören (z. B. über MC_AddAxisToGroup hinzugefügt).
	ActOverrideFactor	LREAL	Aktuell verwendeter Overridefaktor
PathVelo		LREAL	Nur MC Group Coordinated Motion: Absolutwert der kartesischen Geschwindigkeit auf der Bahn
InvokeId		UDINT	Nur MC Group Coordinated Motion: Segment-ID für Analysezwecke
IsInBlendingSegment		BIT	Nur MC Group Coordinated Motion: Gibt an, ob ein Blendig-Segment aktiv ist.
RemainingTimeActiveJob		LREAL	Nur MC Group Coordinated Motion: Verbleibende Zeit des aktuellen Segments
RemainingCartesianDistanceActiveJob		LREAL	Nur MC Group Coordinated Motion: Verbleibende Strecke für das aktuelle Segment
ActiveBlockerId		UDINT	Nur MC Group Coordinated Motion: Id des aktiven Blockers.
RemainingTimeToSync		LREAL	Nur MC Group Coordinated Motion: Verbleibende Zeit, bis die Achsgruppe beim Conveyor Tracking mit dem Transportband synchronisiert ist.
RemainingCartesianTimeToSync		LREAL	Nur MC Group Coordinated Motion: Verbleibende Strecke, bis die Achsgruppe beim Conveyor Tracking mit dem Transportband synchronisiert ist.

9.1.2 PlcToNc

Datenstruktur, die zyklisch zwischen PLC und NC ausgetauscht wird.

Name		Typ	Beschreibung
Common			
	OverrideFactor	LREAL	Gewünschter Override-Faktor (1,0 = 100 %, Standardwert ist 1,0)

9.2 Index-Offset-Spezifikation für MC-Gruppenparameter

Port 501: `AMSPORT_R0_NCSAF: UINT := 501;`.
Als Indexgruppe muss die Objekt-ID (OID) der MC-Gruppe genannt werden.

Index Offset (Hex)	Zugriff	Gruppen-typ	Data Ty-pe	Phys. Ein-heit	Definiti-onsbe-reich	Beschrei-bung	Anmerkungen
0x5030084	Lesen	MC-Gruppe	LREAL	mm	>=0	Verbleibende Strecke des aktuellen Segments.	Ab V3.1.6.27, nur verfügbar für MC Group with Pick-and-Place
0x5030085	Lesen	MC-Gruppe	LREAL	s	>=0	Verbleibende Zeit für das aktuelle Segment.	Ab V3.1.6.27, nur verfügbar für MC Group with Pick-and-Place

9.3 Unterschiede zwischen MC2 und MC3

In diesem Kapitel werden die Unterschiede zwischen MC2 und MC3 (eingeführt im TF5400 Advanced Motion Pack) aufgeführt.

Achsen

	MC2	MC3
Maximale Dynamik	Die in der Parametrierung der Achse definierte Geschwindigkeit wird als physikalischer Maximalwert interpretiert. Beschleunigung, Verzögerung und Ruck, die in der Achse festgelegt werden, sind Standardwerte, die nur eine Wirkung haben, wenn keine Dynamik in den FBs festgelegt wird.	Es gibt Maximalwerte für Geschwindigkeit, Beschleunigung, Verzögerung und Ruck, welche die Werte begrenzen, die in den FBs gesetzt werden können. Darüber hinaus kann vom Benutzer am jeweiligen FB-Eingang Standarddynamik ausgewählt werden.

SPS-Bibliothek

	MC2	MC3
Default Werte	Für Dynamikparameter vom Typ LREAL ist "0" der Standardwert. Wenn "0" gesetzt ist, werden die Standardparameter von den Achsen verwendet.	Die Konstante MC_Default wird eingeführt (siehe MC_LREAL/ Spezielle Eingangswerte [▶ 102]). "0" wird nicht als Standardwert interpretiert, sondern als ein normaler Wert, der im Falle von Dynamik ungültig sein kann.
Zeitabstimmung der FB-Ausgänge	Der FB gibt Werte zurück, die am Anfang des SPS-Zyklus gültig waren.	Der FB gibt Werte zurück, die im Moment der Ausführung des SPS-Codes gültig sind. Dies kann zu einer zeitlichen Differenz zwischen zyklischer Schnittstelle und FB-Ausgang führen.
Entkopplung	Ein spezieller Funktionsbaustein kann verwendet werden (z. B. MC_GearOut/MC_CamOut)	Die Slave-Achse wird entkoppelt, indem ein anderes Bewegungskommando mit Buffermode mcAborting gesendet wird.

9.4 MC_LREAL/Spezielle Eingangswerte

Der Datentyp MC_LREAL entspricht dem Datentyp LREAL. Es gibt jedoch einige zusätzliche Werte, die eine spezielle Bedeutung haben.

Wert	Bedeutung	Beispiel
MC_DEFAULT	Der Eingang wird mit dem Standardwert für diesen Eingang ausgeführt.	Beschleunigung, Verzögerung, Ruck für alle Bewegungskommandos
MC_MAXIMUM	Das Kommando wird mit dem Maximalwert für diesen Eingang ausgeführt.	Generell kann ab Softwareversion 3.1.4.4 für spezifische Bewegungskommandos der Wert `MC_MAXIMUM` den Eingängen `Velocity`, `Acceleration`, `Deceleration` und `Jerk` zugewiesen werden. Ausführlichere Informationen finden Sie in der jeweiligen Dokumentation des Funktionsbausteins, zu dem der Eingang gehört, der den Wert `MC_MAXIMUM` erhalten soll.
MC_IGNORE	Der Eingang wird ignoriert.	MC_GearInPosCA.MasterStartDistance
MC_INVALID	Der Eingang muss vom Benutzer gesetzt werden; weder gibt es einen Standard- oder Maximalwert, noch kann der Eingang ignoriert werden.	MC_MoveAbsoluteCA.Position

9.5 Globale Datentypen

9.5.1 MC_GROUP_TYPE

Das Enum `MC.MC_GROUP_TYPE` definiert den Typ einer Gruppe.

Syntax

Definition:

```
TYPE MC_GROUP_TYPE
(
    mcGroupTypeInvalid := 0,
    mcGroupTypeCA      := 1,
    mcGroupTypeCM      := 3
)
END_TYPE
```

Values

Name	Beschreibung
mcGroupTypeInvalid	Keine gültige Gruppe
mcGroupTypeCA	Collision Avoidance Gruppe
mcGroupTypeCM	Coordinated Motion Gruppe [▶ 18]

9.5.2 MC_GROUP_STATE

Das Enum `MC.MC_GROUP_STATE` definiert den Zustand einer Gruppe. Zustandsübergänge, siehe Zustandsdiagramm [▶ 11].

Syntax

Definition

```
TYPE MC_GROUP_STATE
(
    mcGroupStateInvalid   := 0,
    mcGroupStateDisabled  := 1,
    mcGroupStateStandby   := 2
    mcGroupStateMoving    := 3,
    mcGroupStateStopping  := 4,
    mcGroupStateErrorStop := 5,
    mcGroupStateHoming    := 6,
    mcGroupStateNotReady  := 7,
    mcGroupStateSuspended := 8
)
END_TYPE
```

Values

Name	Beschreibung
mcGroupStateInvalid	Kein gültiger Gruppenzustand
mcGroupStateDisabled	Die Gruppe ist deaktiviert.
mcGroupStateStandby	Die Gruppe ist im Standby-Zustand.
mcGroupStateMoving	Die Gruppe führt eine Bewegung aus.
mcGroupStateStopping	Die Gruppe führt einen Stop aus.
mcGroupStateErrorStop	Die Gruppe führt einen Stop aufgrund eines Fehlers aus.
mcGroupStateHoming	
mcGroupStateNotReady	Die Gruppe ist nicht bereit.
mcGroupStateSuspended	

9.5.3 ST_GroupStatus

Die Struktur `ST_GroupStatus` definiert den Zustand einer Gruppe.

Syntax

Definition:

```
TYPE ST_GroupStatus
STRUCT
    State : MC_GROUP_STATE;
    Flags : ST_GroupStatusFlags;
END_STRUCT
END_TYPE
```

Values

Name	Typ	Beschreibung
State	MC_GROUP_STATE [▶ 102]	Zustand der Gruppe
Flags	ST_GroupStatusFlags [▶ 103]	Zusätzliche Statusinformationen

9.5.4 ST_GroupStatusFlags

Die Struktur `ST_GroupStatusFlags` definiert zusätzliche Statusinformationen.

Syntax

Definition:

```
TYPE
STRUCT
    IsEnableRequested : BIT;
    AllAxesStanding   : BIT;
```

```
    InPosition        : BIT;
END_STRUCT
END_TYPE
```

Values

Name	Typ	Beschreibung
IsEnableRequested	BIT	Definiert, ob eine Aktivierung oder Deaktivierung einer Gruppe angefordert wird.
AllAxesStanding	BIT	Gibt an, ob alle Achsen physikalisch stillstehen, d. h. die Geschwindigkeit und Beschleunigung bei allen Achsen null ist, unabhängig davon, ob ein Bewegungsbefehl aktiv ist.
InPosition	BIT	Gibt an, ob sich die Gruppe in Position befindet.

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Mehr Informationen:
www.beckhoff.com/TF5420

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com