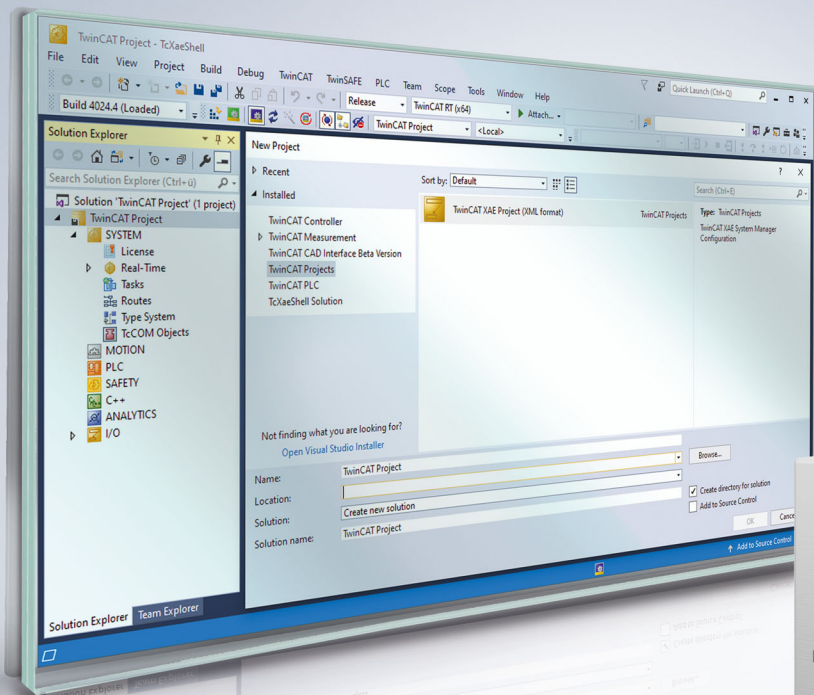


BECKHOFF New Automation Technology

Handbuch | DE

TF5410

TwinCAT 3 | Motion Collision Avoidance



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	Konfigurieren der CA-Gruppe für Collision Avoidance	10
4.1	Geo-Kompensation	15
5	Unterschiede zwischen MC2 und MC3	19
6	CA-Gruppe (TF5410 TwinCAT 3 Collision Avoidance)	20
7	Zustandsdiagramme	24
7.1	Zustandsdiagramm gültig für V3.1.6	24
7.2	Zustandsdiagramm gültig für V3.1.10	25
8	Hintergrundinformationen	27
8.1	Collision Avoidance	27
8.1.1	Grundlagen Collision Avoidance	27
8.1.2	MC_DEFAULT_GAP_CONTROL_MODE	28
8.1.3	MC_GAP_CONTROL_DIRECTION.....	30
8.1.4	MC_GearInPosDefaultDynamicsAfterSync.....	31
8.2	Geo-Kompensation	32
8.3	Track-Management	33
9	SPS-Bibliotheken	34
9.1	Tc3_McCollisionAvoidance	34
9.1.1	Funktionsbausteine	34
9.1.2	Datentypen.....	46
9.2	Tc3_McCompensations	52
9.2.1	Funktionsbausteine	53
9.3	Tc3_McCoordinatedMotion	56
9.3.1	Funktionsbausteine	58
9.3.2	Datentypen.....	94
9.4	Tc3_Mc3Definitions.....	104
9.4.1	Datentypen.....	104
10	Beispiele	113
11	Anhang	114
11.1	Zyklische Gruppenschnittstelle	114
11.1.1	NcToPlc.....	114
11.1.2	PlcToNc.....	115
11.2	MC_LREAL/Spezielle Eingangswerte	115
11.3	Modulo-Positionierung	116

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, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte 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 weiter entwickelt.

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®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® 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 Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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.

Zuwendungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

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

TwinCAT 3 Motion Collision Avoidance ist ein optionales Paket zur Kollisionsvermeidung beim Betrieb mehrerer Achsen mit TwinCAT 3 NC PTP in linearer und/oder translatorischer Abhängigkeit. Der zugrunde liegende Algorithmus sorgt für einen Mindestabstand zur Vorgängerachse. Dadurch ist mit TwinCAT 3 Motion Collision Avoidance eine aktive Kollisionsvermeidung realisierbar, wenn mehrere Motoren sich z. B. eine Schiene teilen. Neben der aktiven Kollisionsvermeidung kann das TF5410 auch genutzt werden, um Achsen kontrolliert aufzustauen, bspw. bei linearen Bewegungen wie dem XTS (eXtended Transport System).

Die Programmierung der Verfahrbefehle aus der SPS erfolgt über die Bibliothek [Tc3_McCollisionAvoidance](#) [► 34], welche an die Bibliothek [Tc2_MC2](#) angelehnt ist und um den Eingang „Gap“ erweitert wurde. Mit TwinCAT 3 Motion Collision Avoidance können so z. B. alle Achsen auf die gleiche Zielposition gestartet werden. Der Algorithmus sorgt dann dafür, dass nur die erste Achse die Zielposition anfährt. Die verbleibenden Achsen halten automatisch ihren Mindestabstand ein und reihen sich auf. So lassen sich ohne weitere Programmierung dynamische Puffer zum Aufstauen von Produkten bilden.

Die administrativen Funktionsbausteine sind in der Bibliothek [Tc3_McCoordinatedMotion](#) [► 56] enthalten.

Zusätzlich ist in TF5410 eine Geo-Kompensation fürs XTS enthalten, womit der Bezugspunkt der Bahndynamik von der XTS-Motorbahn auf den Schwerpunkt des Werkzeuges/Produktes am XTS-Mover verlagert werden kann. In Kurvensegmenten unterscheiden sich die beiden Bahndynamiken, sodass ohne eine Geo-Kompensation unerwartete Kräfte wirken können. In der SPS steht hierfür die Bibliothek [Tc3_McCompensations](#) [► 52] zur Verfügung.

Installation

Das Softwarepaket TF5410 TwinCAT 3 Motion Collision Avoidance wird zusammen mit dem Softwarepaket TF5400 installiert.

Zielsystem

Windows XP oder Windows 7/8/10 ab Plattform-Level 40

Zusätzliche Lizenzanforderungen

TF5410 TwinCAT 3 Motion Collision Avoidance benötigt die Lizenz TC1250.

3 Übersicht der neuen Funktionen

Ab TF5400 V3.2.27:

- Optimierungen am MC_GearInPosCA, die einen SAF-Zyklusversatz zwischen Master- und Slaveachse unterbinden.
- Optimierungen am Gap Controller, wenn die Achse sich schon in der Zielposition befindet und nur der Gap sich ändert. Wird der benachbarte Mover kommandiert, so wirkt der neue Gap.
- Benötigt eine x64-Plattform

Ab TF5400 V3.1.10.63:

- Benötigt TwinCAT V3.1.4024.24 oder höher

Ab TF5400 V3.1.10.30:

- Verhaltensänderung bei der Modulo-Positionierung. Zusätzliche Umdrehungen sind nun über den neuen Parameter `ST_MoveAbsoluteCAOptions.AdditionalTurns` zu kommandieren. Bitte beachten Sie die Hinweise zur [Modulo-Positionierung](#) [► 116].

Ab TF5400 V3.1.10.1:

- Track-Management
- Geändertes Zustandsdiagramm
- Benötigt TwinCAT V3.1.4024.7 oder höher

Ab TF5400 V3.1.6.3:

- Geo-Kompensation

Ab TF5400 V3.1.4.4:

- Neu: Ab TF5400 3.1.4.4 wird `MC_MAXIMUM` als Eingangswert neu unterstützt. Für genauere Informationen schauen Sie sich die Dokumentationen des jeweiligen Funktionsbausteins an.

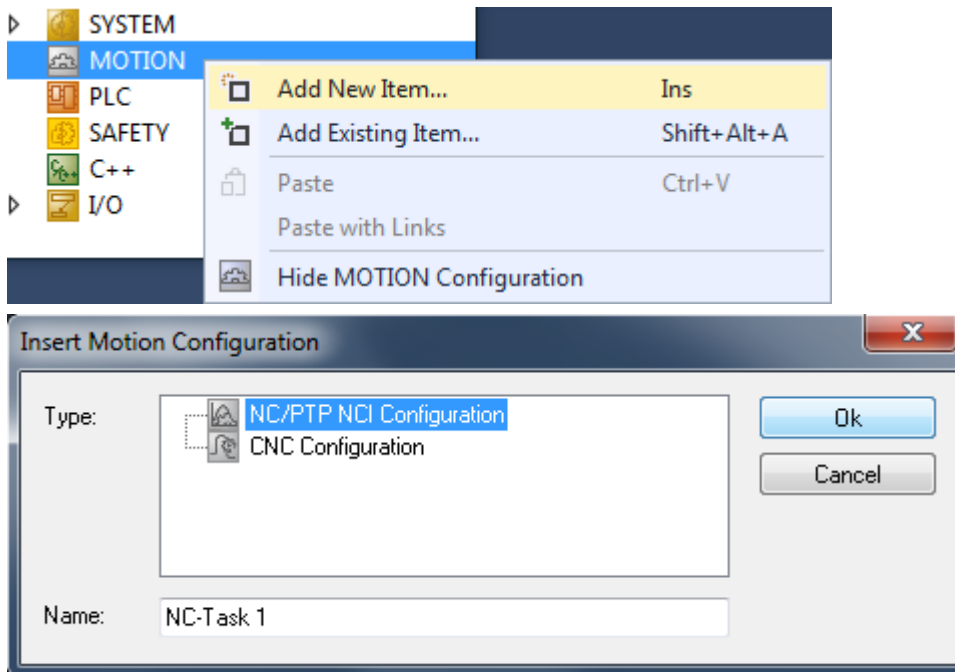
Ab TF5400 V3.1.2.47:

- Neuer Eingang `MC_GAP_CONTROL_MODE` [► 52] an jedem Motion-Funktionsbaustein.
- Neues Flag `MC_GearInPosCAOptions.OverrideSlaveDynamicRestrictions` zur Verbesserung des Verhaltens bei Kopplung an einen Master mit nicht konstanter Geschwindigkeit (z. B. Encoderachse).
- Neuer CA-Gruppenparameter `GapControlModeDirection` legt die Richtung der Abstandsüberwachung fest.

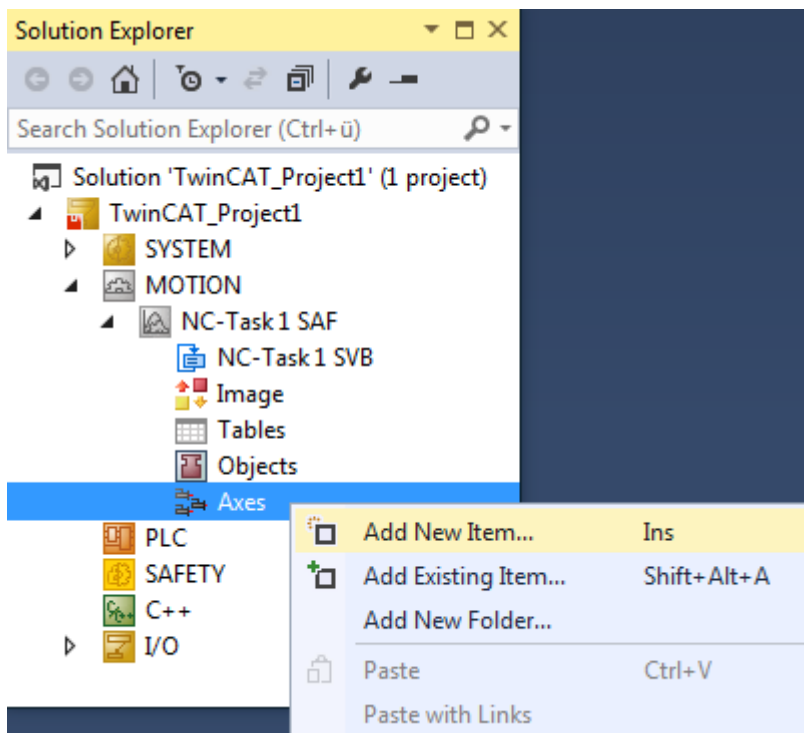
4 Konfigurieren der CA-Gruppe für Collision Avoidance

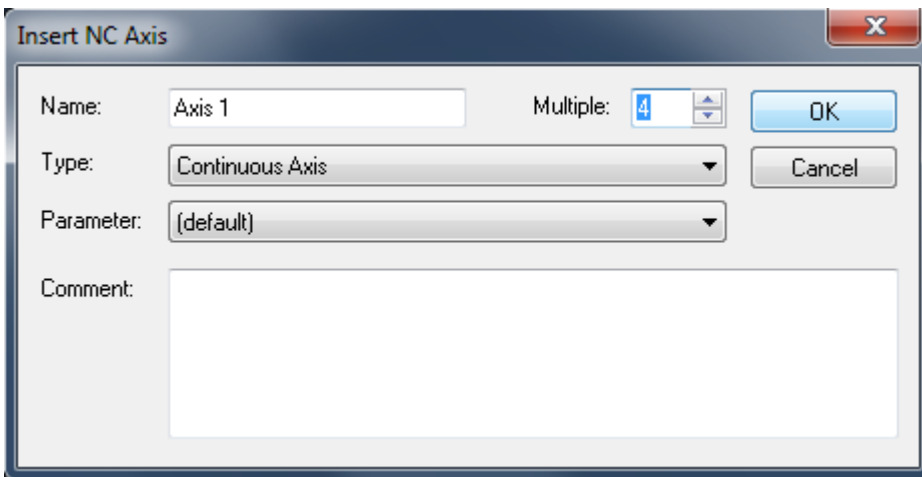
Grundsätzlich gilt die hier beschriebene Konfiguration für alle Motion Objekte im Advanced Motion Pack.

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

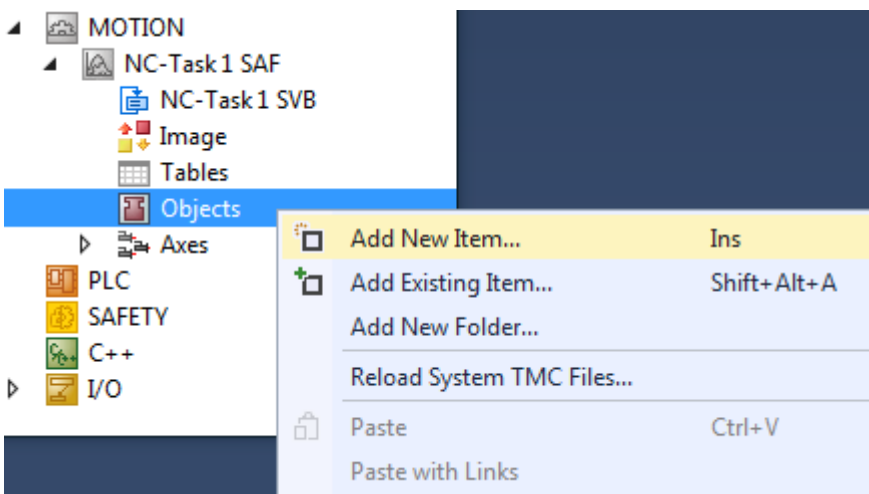


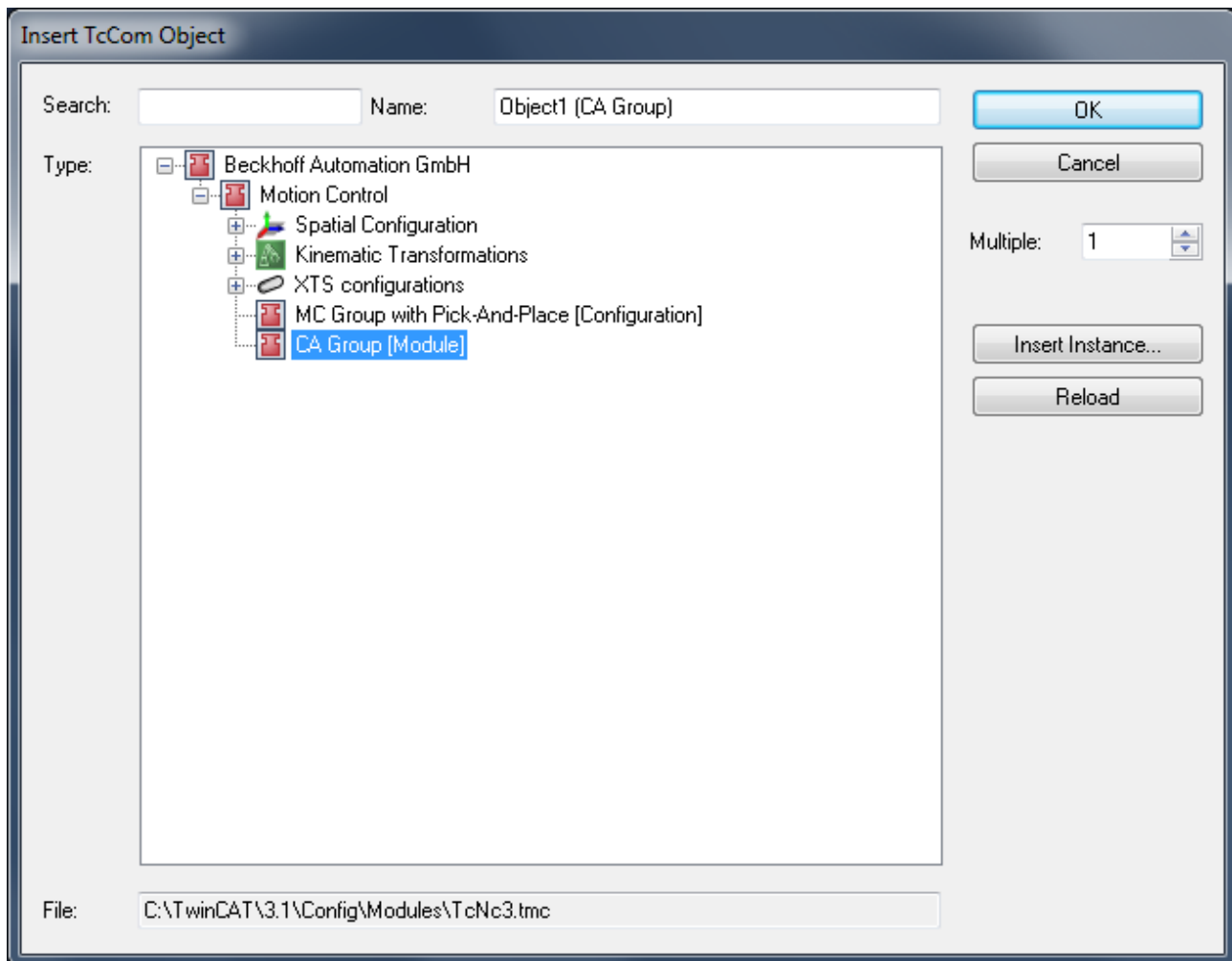
2. Fügen Sie der NC-Konfiguration alle Achsen hinzu.



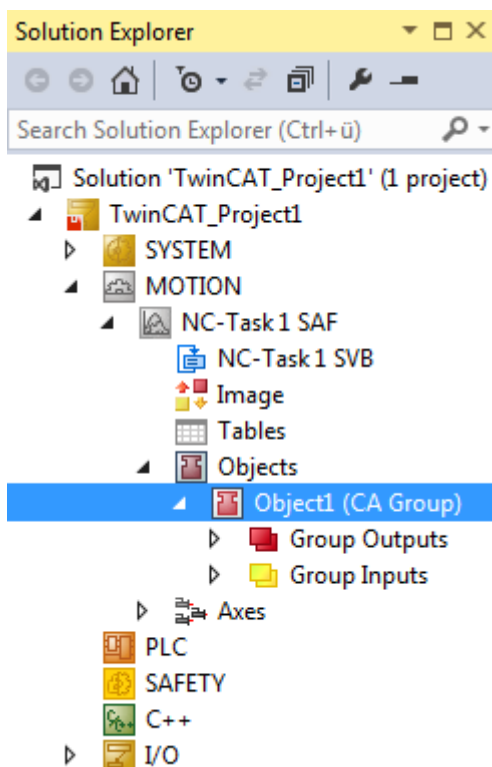


- Fügen Sie dem Eintrag **Objects** in der NC-Konfiguration die entsprechende Gruppe hinzu:
Für koordinierte Bewegung, mehrdimensionale Bewegungen: CA-Gruppe (TF5410 TwinCAT 3 Collision Avoidance) [► 20].





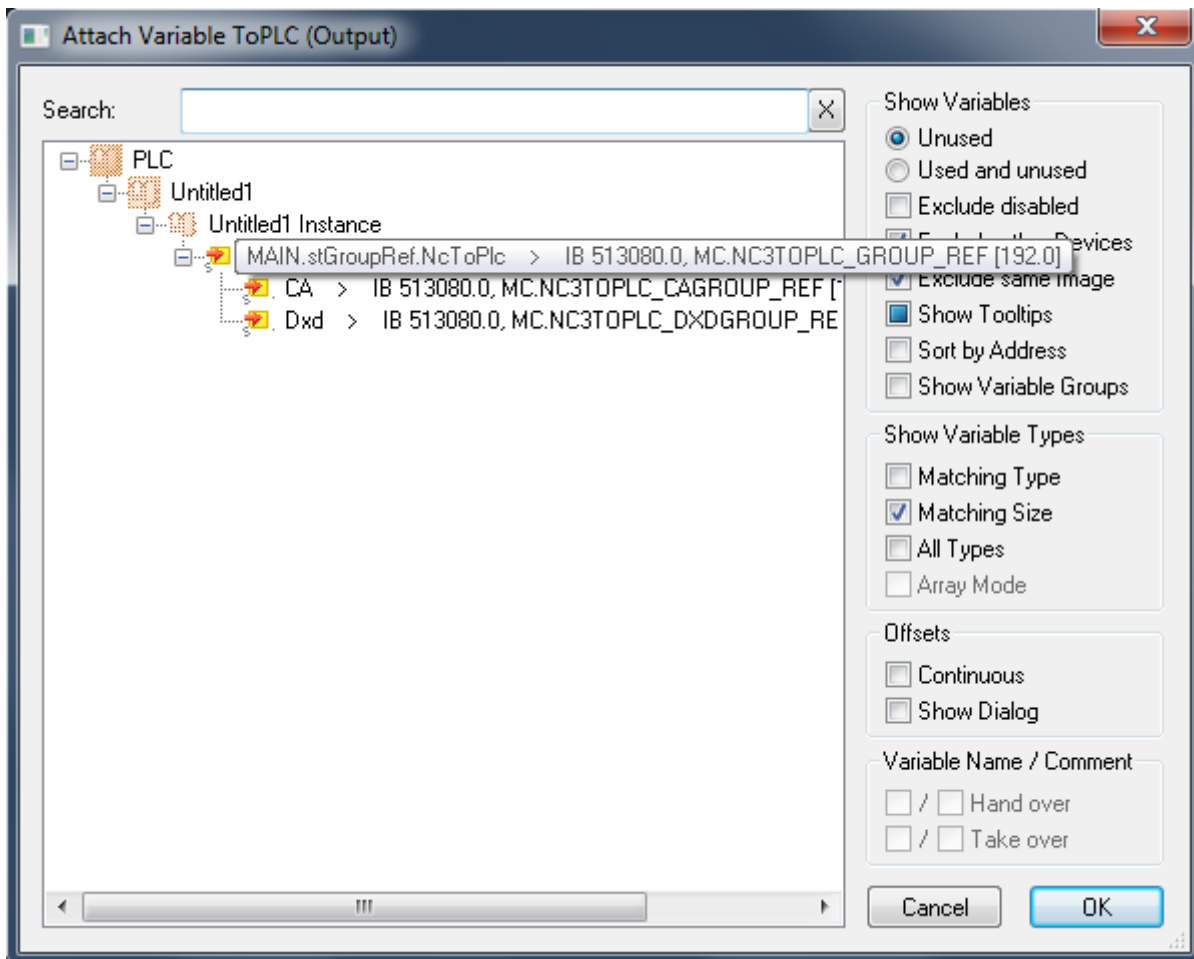
4. Prüfen Sie die Ausführungstask in der Gruppe. Diese muss immer auf "NC-Task 1 SAF" gesetzt sein.



Object	Context	Parameter (Init)
Context:		
Depend On:		
<input type="checkbox"/> Need Call From Sync Mapping		
Data Areas:		
<input checked="" type="checkbox"/> 1 'Group Outputs'		
<input checked="" type="checkbox"/> 2 'Group Inputs'		
Data Pointer:		
Result:		
ID	Task	
0	05000010 'NC-Task 1 SAF'	
	00000000	
	05000020 'NC-Task 1 SVB'	
	05000010 'NC-Task 1 SAF'	
	03000011 'I/O Idle Task'	

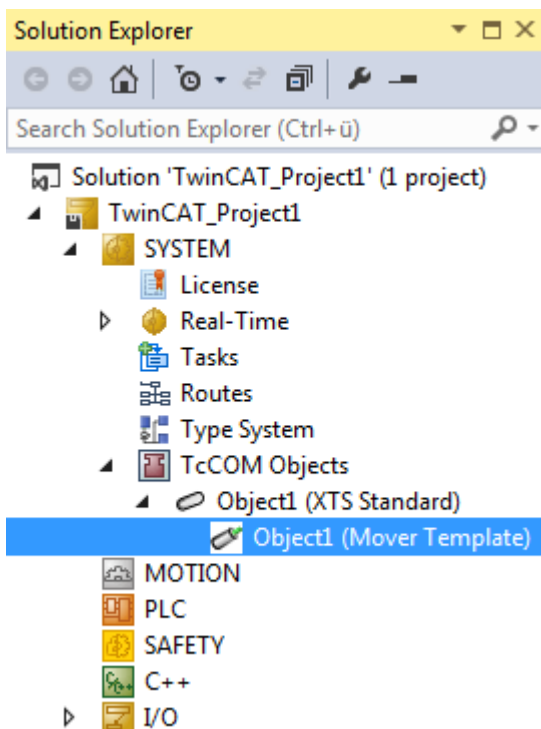
5. Konfigurieren Sie die Gruppenparameter entsprechend der gewünschten Anwendung.
Für weitere Erläuterungen zu den Gruppenparametern siehe [CA-Gruppe \(TF5410 TwinCAT 3 Collision Avoidance\)](#) [► 20].
 6. Um die Gruppe aus der SPS zu adressieren, muss eine zyklische Schnittstelle deklariert und mit den I/Os der Gruppe verknüpft werden (siehe SPS-Bibliothek [Tc3_McCoordinatedMotion](#) [► 56]). 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
```

4.1 Geo-Kompensation

Geometrieinformationen



Für die Geo-Kompensation werden Geometrieinformationen benötigt. Diese Geometrieinformationen werden im Unterbaum *TwinCAT SYSTEM\TcCOM Objects* konfiguriert.

Tabelle eines XTS Standard Objects

The screenshot shows a software window titled 'TwinCAT_Project1'. It contains a tabbed interface with 'Parameter (Init)' selected. Below the tabs is a table with the following data:

	Name	Value	Online	CS	Unit	Type	PTCID	Comment
-	Kinematic							
	Rail length	3000.0		<input type="checkbox"/>	mm	LREAL	0x05010021	Total length of the XTS-rail.
	Rail offset	0.0		<input type="checkbox"/>	mm	LREAL	0x05010080	Offset value applied to adjust the reference position of the rail.

At the bottom of the window, there are two checkboxes: 'Show Online Values' and 'Show Hidden Parameter', both of which are unchecked. To their right are two buttons: 'Expand All' and 'Collapse All'.

Tabelle eines Mover Template Objects

The screenshot shows a software window titled 'TwinCAT_Project1'. It contains a tabbed interface with 'Parameter (Init)' selected. Below the tabs is a table with the following data:

	Name	Value	Online	CS	Unit	Type	PTCID	Comment
-	Kinematic							
	TCP y-displacement	100.0		<input type="checkbox"/>	mm	LREAL	0x05010024	y-displacement of the mover top. The value must be > 0.

At the bottom of the window, there are two checkboxes: 'Show Online Values' and 'Show Hidden Parameter', both of which are unchecked. To their right are two buttons: 'Expand All' and 'Collapse All'.

Das XTS Standard Object beschreibt die Geometrie einer standardmäßigen XTS-Motorbahn. Als Mover Template bezeichnete Objekte definieren jeweils die Geometrie eines einzelnen Mover-Typs, einschließlich einer Verschiebung entlang der y -Verschiebungskomponente. Ein Mover Template wird dem XTS Standard Object hinzugefügt, um die Standard-Geometrieinformationen um die Geometrieinformationen des Movers zu erweitern. Auf ein Mover Template können sich alle Achsen beziehen, die dessen Konfiguration verwenden.

XTS Standard Object



Das XTS Standard Object definiert eine Motorbahn, die die Starterkit-Geometrie einschließlich zwei Kurven von 180 Grad aufweist. Diese Kurven werden durch zwei gerade Bahnen mit gleicher Länge verbunden. Die Länge dieser geraden Bahnen kann während der Konfiguration verändert werden. Dadurch konfiguriert der Parameter **Rail length** des XTS Standard Objects die Gesamtlänge: beide Kurven plus beide Geraden. Für die Positionsinformationen auf der XTS-Motorbahn x kann im XTS Standard Object eine Nullpunktverschiebung (Offset) konfiguriert werden.

Rail length: Gesamtlänge der XTS-Schiene.

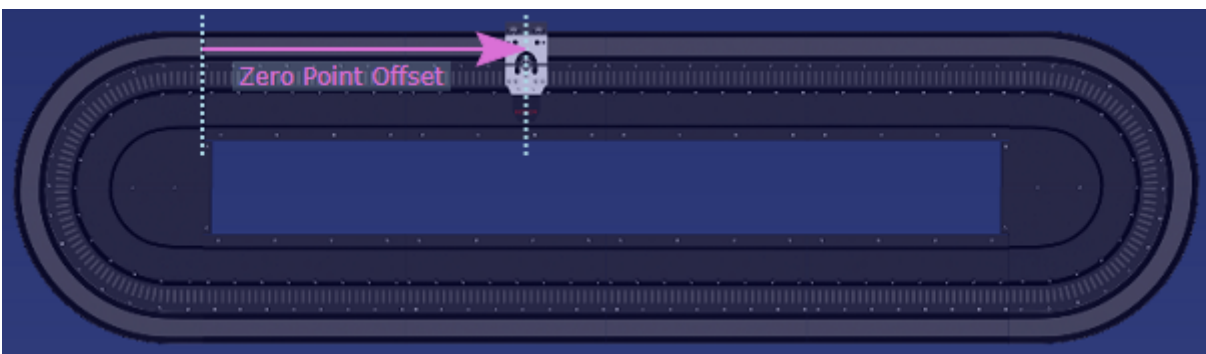
Rail offset: Offset-Wert für die Anpassung der Referenzposition der Schiene. Siehe unten und Abbildung Starterkit-Geometrie.

Rail Offset: Eine Nullpunktverschiebung

Jedes XTS-System umfasst ein Segment, das die Nullposition in x -Richtung festlegt. Die Geo-Kompensation verwendet die Starterkit-Geometrie. Bei der Geo-Kompensation hat das Segment zur Festlegung der Nullposition einen festen Platz. Es handelt sich um das gebogene Element in der oberen linken Ecke vor dem ersten geraden Element.

Um die Nullposition an anderer Stelle festzulegen und von einer anderen Position aus mit der Zählung der x -Koordinate zu beginnen, kann eine Nullpunktverschiebung, der **Rail Offset**, definiert werden.

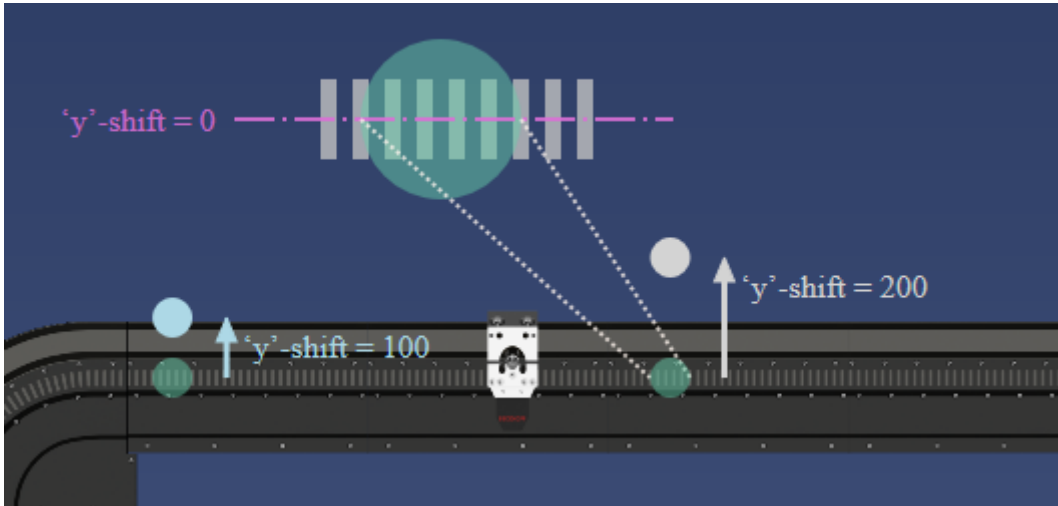
In der Abbildung ist der **Rail Offset** zwischen den beiden gestrichelten Linien dargestellt. Die linke Linie zeigt, wo das Segment zur Festlegung der Nullposition endet. Die gestrichelte Linie rechts und der abgebildete Mover veranschaulichen, wie ein Positionswert von einem Mover interpretiert wird. Die gestrichelte Linie teilt den Mover in zwei Hälften. Der Mover befindet sich an der Nullposition. Für die Festlegung eines **Rail Offset** selbst ist jedoch kein Mover erforderlich.



i Hinweis zur Verfügbarkeit der Nullpunktverschiebung

Derzeit ist die Starterkit-Geometrie die einzige verfügbare Geometrie für die Geo-Kompensation: Zwei Kurven von 180 Grad und zwei gerade Bahnen mit gleicher Länge, die diese Kurven verbinden.

Mover Template Object



Zunächst übernimmt ein Mover Template Object die Geometrieinformationen des XTS Standard Objects. Darüber hinaus beschreibt es eine Mover-Bahngeometrie, d. h. die y -Verschiebung eines bestimmten Mover-Typs. Ein Mover Template kann für verschiedene Mover, die dieselbe Bahngeometrie, d. h. dieselbe Bahn des Werkzeugmittelpunkts (Tool Center Path) aufweisen, wiederverwendet werden. Ein Mover Template kann im Run-Modus aktiviert und deaktiviert werden. Somit kann das Template für einen Mover im Run-Modus geändert werden.

y-Verschiebung des TCP: Konfigurierbare y -Verschiebung für die Steuerung der Bahndynamik. Die y -Verschiebung muss positiv oder gleich null sein. Sie beschreibt für jeden Punkt einer XTS-Motorbahn einen Punkt einer Mover-Bahn, der senkrecht zur Tangente der Motorbahn liegt, die durch diesen Punkt der Motorbahn verläuft. In dieser Richtung mit Orientierung der Geometrie des XTS Standard Objects nach außen wird dieser Punkt der Mover-Bahn um den Wert der y -Verschiebung von der XTS-Motorbahn weg verschoben. Dieser verschobene Punkt wird auch als Tool Center Point (TCP, Werkzeugmittelpunkt) bezeichnet. Zusammen beschreiben die y -Punkte eine Bahn, die als Bahn des Werkzeugmittelpunkts (Tool Center Path) bezeichnet wird.

Auf einer Geraden bilden die Motorspulen ein Zebrastrifen-ähnliches Muster. Wenn diese Gerade in der Mitte dieses Musters liegt und jede Motorspule in eine obere und eine untere Hälfte teilt, weist die y -Verschiebung den Wert null auf dieser Geraden auf (siehe Abbildung). Wenn die y -Verschiebung null ist, erfolgt die Steuerung der Bahndynamik zusammengefasst in der vertikalen Mitte der Motorspulen.

● Hinweis zur Verfügbarkeit der Nullpunktverschiebung

i Derzeit ist die Starterkit-Geometrie die einzige verfügbare Geometrie für die Geo-Kompensation: Zwei Kurven von 180 Grad und zwei gerade Bahnen mit gleicher Länge, die diese Kurven verbinden.

5 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 [► 115]). "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.

6 CA-Gruppe (TF5410 TwinCAT 3 Collision Avoidance)

Die CA-Gruppe verbindet Achsen, um die PTP-Funktionalitäten um Collision Avoidance zu ergänzen.

Dynamik-Werte

- Velocity **Vel**: Geschwindigkeit,
- Acceleration **Acc**: Positive Beschleunigung,
- Deceleration **Dec**: Bremsbeschleunigung, negative Beschleunigung,
- Jerk: Ruck.

Soll-Werte und Grenzen

- Eine Achse fährt Dynamik-Soll-Werte ab. Bei dieser Fahrt geben die maximalen Dynamikwerte die Grenzen für das Dynamik-Profil vor.

CA-Gruppe

- Eine CA-Gruppe bietet Parameter an, mit denen für Dynamik-Werte Default-Werte eingestellt werden können. Diese Default-Werte werden für Standby Gap Control verwendet. Sie werden nicht als Default-Parameter für irgendein Motion-Kommando verwendet, wenn kein Parameter spezifiziert worden ist.

Achse

- In den Achsparametern können die Maximalwerte für die Dynamiklimits eingestellt werden.
- Diese Maximalwerte können sich zum Beispiel aus den physikalischen Eigenschaften (Trägheit, Masse, Maximalstrom, Motorgröße, ..) der Achse oder eines Werkstücks ergeben.

Gap

Nachbarschaft

- Für ein Gap sind zwei oder mehr benachbarte Achsen (Mover) erforderlich.
- Ein Gap liegt jeweils zwischen zwei direkt benachbarten Movern.

Zählrichtung

- Das Gap ist in positiver Zählrichtung vom aktuellen Mover zum direkt vorausfahrenden Mover definiert.
- Diese positive Zählrichtung stimmt mit der Zählrichtung der Sollwert-Generierung überein.

Nachfolger; Vorgänger

- Aktueller Mover: Direkt hinterherfahrender Mover, Nachfolger.
- Direkt vorausfahrender Mover: Vorgänger.

Größe

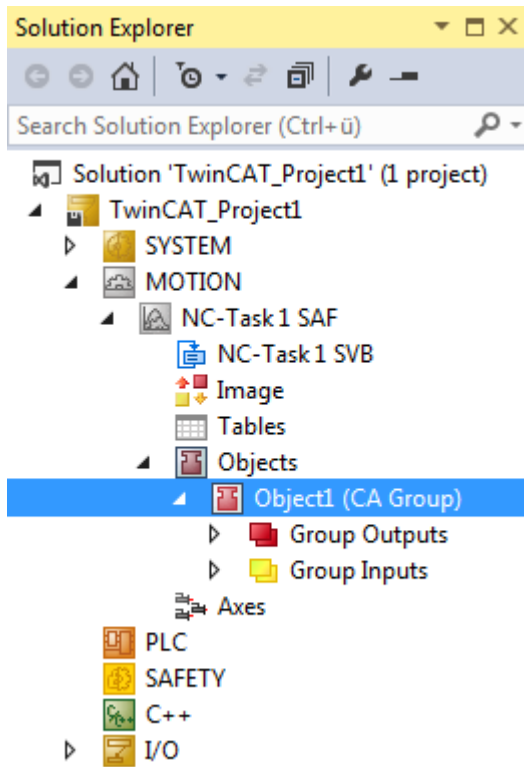
- Größe eines jeweiligen Gaps = (Soll-Position Vorgänger) - (Soll-Position Nachfolger).

Untere Schranke

- Die Gap-Größe wird auf eine untere Schranke geregelt, die nicht unterschritten werden darf.
Für diese untere Schranke stellt der Parameter „Default Gap“ für die ganze CA-Gruppe einen Wert ein, solange für einen aktuellen Mover keine andere Gap-Größe eingestellt wird.
Ein individueller Wert für diese untere Schranke der Gap-Größe kann als Eingangswert an jeden der Motion-Funktionsbausteine angelegt werden: An „MC_MoveAbsoluteCA“, an „MC_MoveRelativeCA“, an „MC_HaltCA“ oder an „MC_GearInPosCA“.

- Gap Control Mode**
 - Der „Gap Control Mode“ mcGapCtrlModeFast regelt im Allgemeinen näher an dieser unteren Schranke der Gap-Größe als der „Gap Control Mode“ mcGapCtrlModeStandard.
- Gap Control Direction**
 - „Gap Control Direction“ mcGapCtrlDirectionPositive: Nur der Nachfolger regelt als einziger Mover die Größe des Gaps, das von Vorgänger und Nachfolger eingeschlossen wird.
 - „Gap Control Direction“ mcGapCtrlDirectionBoth: Beide benachbarte Mover regeln die Größe des von ihnen eingeschlossenen Gaps.

Öffnen des Dialogs „Parameter (Init)“



Wurzelknoten einer CA-Gruppe.

Name	Value	CS	Type	PTCID
Geometry				
Rail Length	3000.0	<input type="checkbox"/>	LREAL	0x05030060
Rail Is Ring	TRUE	<input type="checkbox"/>	BOOL	0x05030066
Gap Control				
Default Gap Control Mode	mcGapCtrlModeStandard	<input type="checkbox"/>	MC.MC_DEFAULT_GAP_CONTROL_MODE	0x050300A0
Gap Control Direction	mcGapCtrlDirectionBoth	<input type="checkbox"/>	MC.MC_GAP_CONTROL_DIRECTION	0x0503009C
Standby Gap Control	FALSE	<input type="checkbox"/>	BOOL	0x050300A1
Default Gap	100.0	<input type="checkbox"/>	LREAL	0x05030062
Default Velocity	150.0	<input type="checkbox"/>	LREAL	0x05030030
Default Acceleration	1000.0	<input type="checkbox"/>	LREAL	0x05030031
Default Deceleration	1000.0	<input type="checkbox"/>	LREAL	0x05030032
Default Jerk	10000.0	<input type="checkbox"/>	LREAL	0x05030033
Miscellaneous				
TraceLevel	tlWarning	<input type="checkbox"/>	TcTraceLevel	0x03002103
Ctx_TaskOid	05000010	<input type="checkbox"/>	OTCID	0x03002060
GearInPosDefaultDynamicsAfterSync	JobDynamics	<input type="checkbox"/>	MC.MC_GearInPosDefaultDynamicsAfterSync	0x05030124

Show Online Values
 Show Hidden Parameter
 Expand All
 Collapse All

Parameter für eine CA-Gruppe.

Die Tabellenspalte „Value“ zeigt den jeweils voreingestellten Parameter-Wert. Die Tabellenspalte „Comment“ enthält kurze Parameter-Beschreibungen.

Parameter	Beschreibung
Geometry	
Rail Length	Länge der Schiene, auf der die Achsen (Mover) montiert sind.
Rail Is Ring	Gibt an, ob die Schienen einen geschlossenen Kreis bilden. In dem Fall (TRUE) ist die Collision Avoidance zwischen dem ersten Mover in der Reihe und dem letzten Mover aktiviert.
Gap Control	
Default Gap Control Mode	Es stehen verschiedene Modi für die Abstandssteuerung zur Verfügung (siehe "MC_DEFAULT_GAP_CONTROL_MODE [▶ 28]").
Gap Control Direction	Für die Regelungs-Richtung des Gaps sind verschiedene Einstellungen verfügbar (siehe Abschnitt „MC_GAP_CONTROL_DIRECTION [▶ 30]“).
Standby Gap Control	Wenn TRUE, ist die Collision Avoidance immer aktiv, auch wenn kein Bewegungskommando an die Achse ausgegeben wurde. Hinweis Die Achsen bewegen sich direkt nach MC_GroupEnable, wenn Standby Gap Control TRUE ist. Wenn der Abstand zwischen zwei Achsen (Mover) kleiner ist als der Default Gap (siehe nächster Parameter), bewegen sich die Achsen so, dass sie den geforderten Abstand erreichen. Diese Bewegung erfolgt unabhängig von jeglichem Bewegungskommando. Dieses Verhalten gilt auch, wenn sich die Achsen nach einem Gruppen-Reset zu nah aneinander befinden.
Default Gap	Dieser Abstand wird für die Standby Gap Control verwendet und wenn bei einem CA-Bewegungskommando kein Abstand festgelegt wurde.
Default Velocity	Diese Geschwindigkeit wird für die Standby Gap Control verwendet, das heißt, wenn kein Bewegungskommando aktiv ist (z. B. direkt nach MC_GroupEnable [▶ 62]). Sie wird nicht als Standardgeschwindigkeit für ein Bewegungskommando verwendet, wenn keine Geschwindigkeit festgelegt wurde.
Default Acceleration	Diese Beschleunigung wird für die Standby Gap Control verwendet, das heißt, wenn kein Bewegungskommando aktiv ist (z. B. direkt nach MC_GroupEnable [▶ 62]). Sie wird nicht als Standardbeschleunigung für ein Bewegungskommando verwendet, wenn keine Beschleunigung festgelegt wurde.
Default Deceleration	Diese Verzögerung wird für die Standby Gap Control verwendet, das heißt wenn kein Bewegungskommando aktiv ist (z.B. direkt nach MC_GroupEnable [▶ 62]). Sie wird nicht als Standardverzögerung für ein Bewegungskommando verwendet, wenn keine Verzögerung festgelegt wurde.
Default Jerk	Dieser Ruck wird für die Standby Gap Control verwendet, das heißt wenn kein Bewegungskommando aktiv ist (z.B. direkt nach MC_GroupEnable [▶ 62]). Er wird nicht als Standardruck für ein Bewegungskommando verwendet, wenn kein Ruck festgelegt wurde.
GearInPosDefaultDynamicsAfterSync (hidden!)	Weist die in MC_GearInPosCA AfterSyncDynamics genutzten Default-Dynamiken aus. Im Default-Zustand ist der Wert „JobDynamics“ gesetzt. Der Parameter ist nicht für ältere Projekte verfügbar (erzeugt mit Versionen < 3.1.10), wird aber intern auf 'MaximumSlaveDynamics' gesetzt.



Nach dem erneuten Laden der TMC-Datei wird "JobDynamics" als Default-Wert gesetzt (siehe MC_GearInPosDefaultDynamicsAfterSync [▶ 31]).

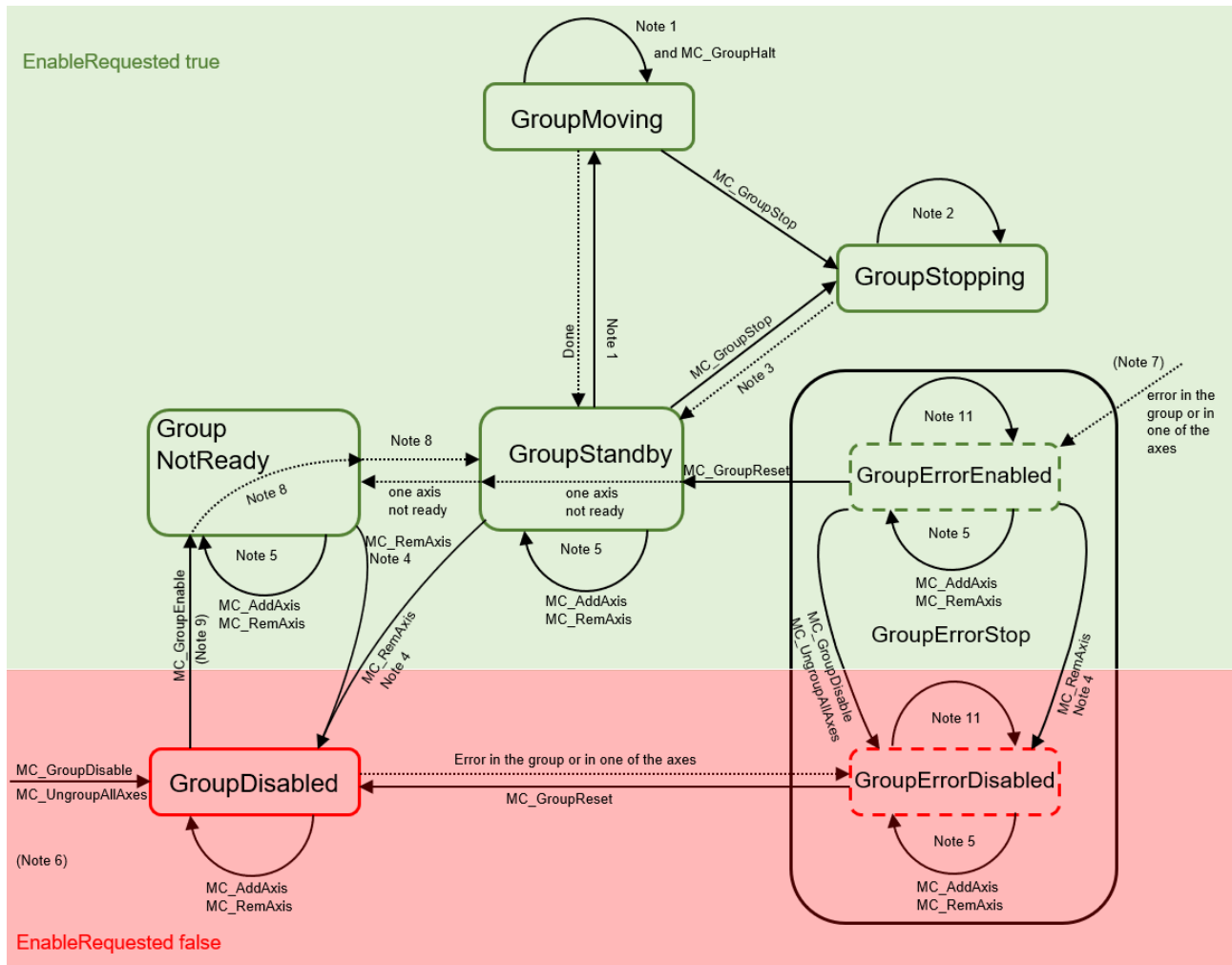
- ✓ Wenn bereits eine NC-Konfiguration hinzugefügt worden ist, dann enthält der MOTION-Teilbaum einen SAF-Task-Teilbaum.
- ✓ Der SAF-Task-Teilbaum enthält den Teilbaum „Objects“.
- ✓ Der Teilbaum „Objects“ kann eine CA-Gruppe enthalten.

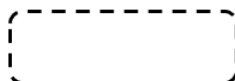
1. Doppelklicken Sie auf den Wurzelknoten der CA-Gruppe, deren Parameter Sie anschauen oder einstellen möchten.
2. Wählen Sie den Karteireiter „Parameter (Init)“ an.
 - ⇒ Der Dialog „Parameter (Init)“ ist geöffnet.
 - ⇒ Er enthält eine Tabelle mit Parametern für die angewählte CA-Gruppe.
 - ⇒ Diese Parameter sind in die Gruppen „Geometry“, „Gap Control“ und ggf. „Misc.“ eingeteilt.

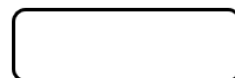
7 Zustandsdiagramme

7.1 Zustandsdiagramm gültig für V3.1.6

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



 : state in the internal state machine, not visible in PLC via MC_ReadGroupStatus

 : state visible in MC_ReadGroupStatus

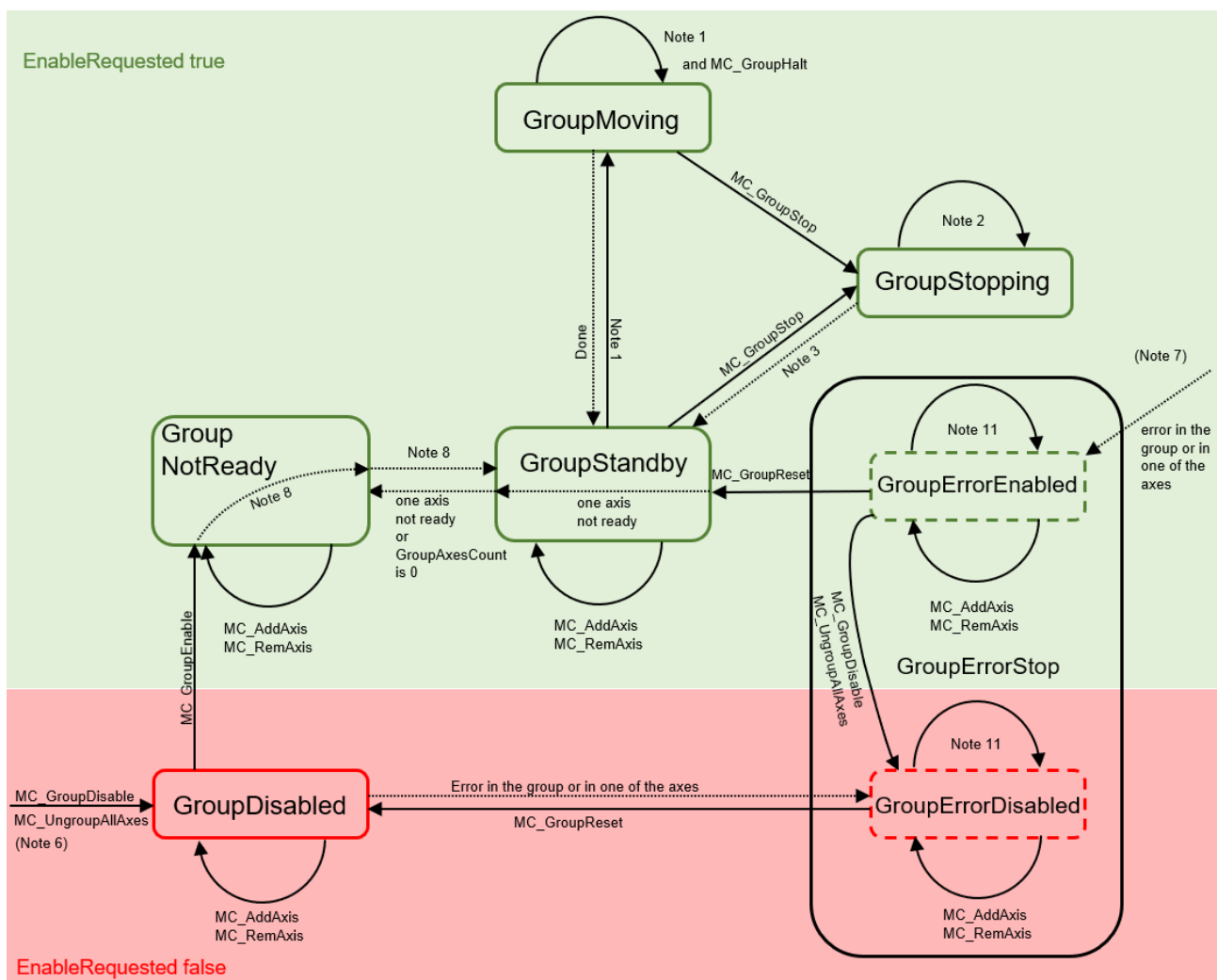
Hinweis Beschreibung

- 1 Anwendbar für alle nichtadministrativen (Bewegungs-) Funktionsbausteine.
- 2 Im Zustand GroupStopping können zwar alle Funktionsbausteine aufgerufen werden, werden jedoch nicht ausgeführt, mit Ausnahme von MC_GroupDisable und MC_UngroupAllAxes, die den Stopp abbrechen und den Übergang zu GroupDisabled erzeugen.
- 3 MC_GroupStop.DONE AND NOT MC_GroupStop.EXECUTE
- 4 Übergang ist anwendbar, wenn die letzte Achse aus der Gruppe entfernt wird.
- 5 Übergang ist anwendbar, während die Gruppe nicht leer ist.

- 6 MC_GroupDisable und MC_UngroupAllAxes können in allen Zuständen ausgegeben werden und ändern den Zustand in GroupDisabled. Wenn sie in einem Fehlerzustand ausgegeben werden, ändert sich der Zustand in GroupErrorDisabled.
- 7 Aus jedem Zustand mit EnableRequested TRUE.
- 8 Wenn "blsControlLoopClosed" für alle Achsen TRUE ist und die Gruppe nicht leer ist. "bPositiveDirection"/"bNegativeDirection" müssen nicht aktiviert sein.
- 9 MC_GroupEnable gibt einen Fehler aus, wenn die Gruppe leer ist.
- 10 MC_GroupReset hat keine Wirkung, wenn der Zustand ein anderer ist als GroupErrorStop.
- 11 In den Fehlerzuständen sind alle administrativen Funktionsbausteine mit Ausnahme von MC_GroupEnable zulässig. Sie können in den Fehlerzuständen jedoch nur Zustandsübergänge erzeugen, z. B. zu GroupErrorDisabled für MC_GroupDisable oder MC_UngroupAllAxes und MC_RemoveAxisFromGroup, wenn die letzte Achse entfernt wird.
- 12 Um den Zustand GroupErrorStop zu verlassen, muss MC_GroupReset aufgerufen werden.

7.2 Zustandsdiagramm gültig für V3.1.10

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





: state in the internal state machine, not visible in PLC via MC_ReadGroupStatus



: state visible in MC_ReadGroupStatus

Hinweis (Note)

Beschreibung

- 1 Anwendbar für alle nichtadministrativen (Bewegungs-) Funktionsbausteine.
- 2 Im Zustand GroupStopping können zwar alle Funktionsbausteine aufgerufen werden, werden jedoch nicht ausgeführt, mit Ausnahme von MC_GroupDisable, die den Stopp abbrechen und den Übergang zu GroupDisabled erzeugen.
- 3 MC_GroupStop.DONE AND NOT MC_GroupStop.EXECUTE
- 4 -
- 5 -
- 6 MC_GroupDisable kann in allen Zuständen ausgegeben werden und ändern den Zustand in GroupDisabled. Wenn sie in einem Fehlerzustand ausgegeben werden, ändert sich der Zustand in GroupErrorDisabled.
- 7 Aus jedem Zustand mit EnableRequested TRUE.
- 8 Wenn "blsControlLoopClosed" für alle Achsen TRUE ist und die Gruppe nicht leer ist. "bPositiveDirection"/"bNegativeDirection" müssen nicht aktiviert sein.
- 9 "blsControlLoopClosed" und die beiden Flags "bPositiveDirection"/"bNegativeDirection" müssen auf den Wert TRUE gesetzt sein.
- 10 -
- 11 In den Fehlerzuständen sind alle administrativen Funktionsbausteine mit Ausnahme von MC_GroupEnable zulässig. Sie können in den Fehlerzuständen jedoch nur Zustandsübergänge erzeugen, z. B. zu GroupErrorDisabled für MC_GroupDisable oder MC_UngroupAllAxes und MC_RemoveAxisFromGroup, wenn die letzte Achse entfernt wird.



Im State GroupMoving dürfen stehende Achsen zu einer **CA Gruppe** hinzugefügt und entfernt werden. Falls versucht wird, eine bewegte Achse einer Gruppe hinzuzufügen oder diese aus der Gruppe zu entfernen, wird das Kommando mit einem Fehler abgelehnt (der Gruppenwechsel mit einer bewegten Achse wird ebenfalls abgelehnt).



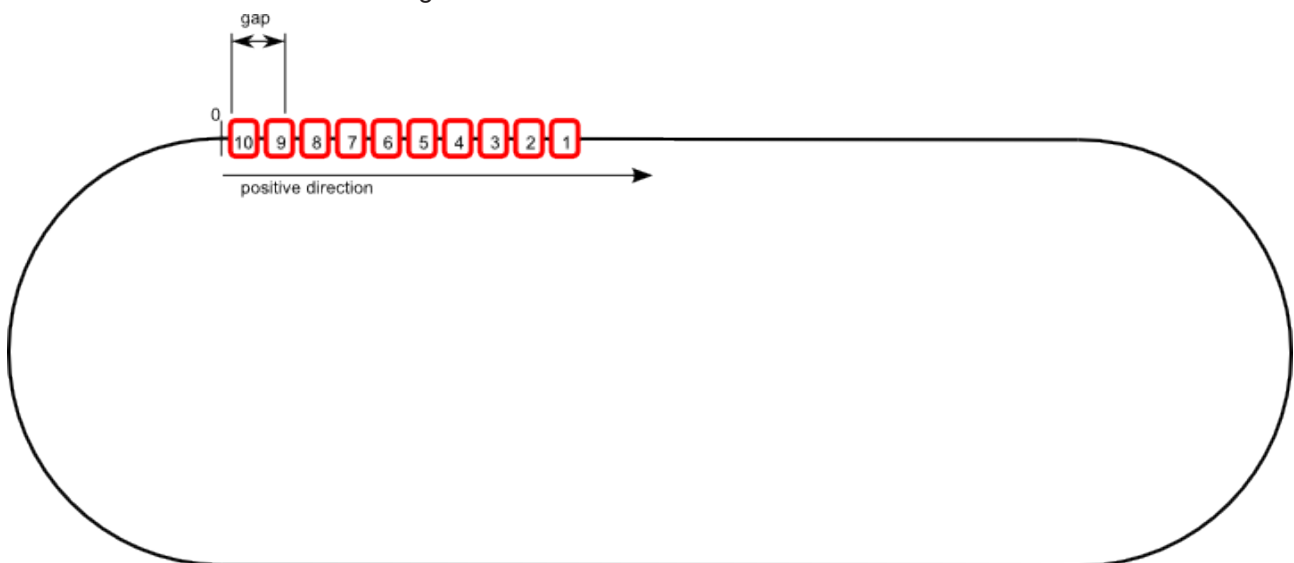
MC_GroupReset hat keine Wirkung, wenn der Zustand ein anderer ist als GroupErrorStop.

8 Hintergrundinformationen

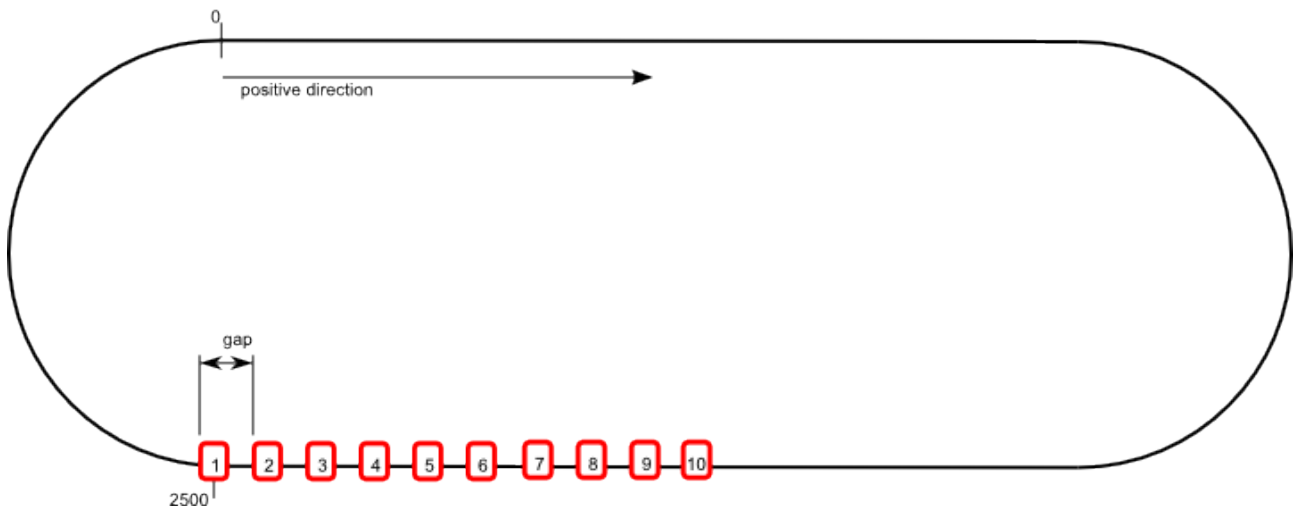
8.1 Collision Avoidance

8.1.1 Grundlagen Collision Avoidance

- ✓ Alle Objekte (CA-Gruppe und alle Achsen) müssen erzeugt, parametrierung und verbunden werden (siehe "Konfiguration", "Parametrierung der CA-Gruppe [▶ 20]").
- ✓ In diesem Beispiel werden die Standardwerte für alle Abstandssteuerungsparameter und 10 Achsen in der Gruppe verwendet. Alle Achsen werden auf einer geschlossenen Schiene (XTS) mit einer Länge von 3000 mm montiert. Die Position der Achsen (Mover) ist beliebig, der in der Gruppe parametrierte Standardabstand wird nicht eingehalten:



1. Alle Achsen müssen der Gruppe hinzugefügt werden (siehe Beispiele in "[MC_AddAxisToGroup \[▶ 58\]](#)").
 - ⇒ Die Reihenfolge der Achsen für die Collision Avoidance wird durch ihre Istposition auf der Schiene bestimmt.
 - ⇒ Wenn die Positionen der Achsen gleich sind (z. B. für Simulationsachsen), ist die Reihenfolge, in der die Achsen der Gruppe hinzugefügt werden, wesentlich. In diesem Fall ist die Achse, die zuletzt hinzugefügt wurde, die erste Achse in der Gruppe.
 - ⇒ Die "IdentInGroup" hat für die bei der Collision Avoidance verwendete Reihenfolge keine Bedeutung.
2. Aktivieren Sie die Gruppe (siehe "[MC_GroupEnable \[▶ 62\]](#)").
 - ⇒ Der GroupState ist nun mcGroupStateStandby (siehe "[MC_GroupReadStatus \[▶ 64\]](#)" oder "Zyklische Gruppenschnittstelle"), der GroupAxesCount ist 10 (siehe "[Zyklische Gruppenschnittstelle \[▶ 114\]](#)").
 - ⇒ Die Position der Achsen (Mover) hat sich nicht geändert, der Abstand wird nach wie vor nicht eingehalten.
3. Geben Sie "[MC_MoveAbsoluteCA \[▶ 34\]](#)" für alle Achsen (Mover) in dieselbe Position (2500 mm) aus.
 - ⇒ Der erste Mover, d. h. der Mover mit der größten absoluten Position, hier Mover 1, erreicht die Zielposition bei 2500 mm. Die anderen Mover reihen sich auf, wobei jeder den Abstand zu seinem Vorgänger einhält. Der Vorgänger des ersten Movers ist der Letzte (da der Gruppenparameter Rail Is Ring auf TRUE gesetzt ist).



8.1.2 MC_DEFAULT_GAP_CONTROL_MODE

Der [Gap Control Mode](#) [► 20] legt das Verhalten der Collision Avoidance fest. Gegenwärtig stehen die folgenden Modi zur Verfügung:

```

TYPE MC_DEFAULT_GAP_CONTROL_MODE :
(
mcGapCtrlModeStandard           := 16#1,
mcGapCtrlModeFast              := 16#2
)
END_TYPE

```

Beispiele

Beispiel mcGapCtrlModeStandard:

- ✓ Konfiguration mit vier Achsen (Mover) in der CA-Gruppe. Die [Rail Length](#) [► 20] beträgt 3000 mm und die Schiene ist geschlossen (z. B. XTS-System).
 - ✓ Die erste Achse in der Reihe (blau) steht an Position 0,0 mm, die übrigen drei Achsen sind dahinter mit einem jeweiligen Abstand von 100 mm aufgereiht.
 - ✓ Der Gap Control Mode wird auf mcGapCtrlModeStandard gesetzt.
1. MC_MoveAbsoluteCA in die Position 3000 mm wird an alle Achsen ausgegeben, der Abstand beträgt 100 mm. Alle Achsen haben dieselbe Dynamik (Geschwindigkeit, Beschleunigung, Verzögerung, Ruck).
- ⇒ Die Achsen fächern während der Beschleunigungsphase charakteristisch aus, so dass eine Kollision während des Bewegungskommandos verhindert wird. Die erste Achse (blau) erreicht die Zielposition, die übrigen Achsen reihen sich nacheinander mit dem konfigurierten [Default Gap](#) [► 20] auf.



Beispiel mcGapCtrlModeFast:

- ✓ Konfiguration mit vier Achsen (Mover) in der CA-Gruppe. Die Rail Length beträgt 3000 mm und die Schiene ist geschlossen (z. B. XTS-System).
 - ✓ Die erste Achse in der Reihe (blau) steht an Position 0,0 mm. Die übrigen drei Achsen sind dahinter mit einem jeweiligen Abstand von 100 mm aufgereiht.
 - ✓ Der Gap Control Mode wird auf mcGapCtrlModeFast gesetzt
1. MC_MoveAbsoluteCA in die Position 3000 mm wird an alle Achsen ausgegeben, der Abstand beträgt 100 mm. Alle Achsen haben dieselbe Dynamik (Geschwindigkeit, Beschleunigung, Verzögerung, Ruck).
- ⇒ Alle Achsen bewegen sich gleichzeitig und mit voller Dynamik. Der Abstand zwischen den Achsen wird nahezu konstant gehalten. Die erste Achse erreicht die Zielposition, die Übrigen reihen sich dahinter auf.



8.1.3 MC_GAP_CONTROL_DIRECTION

Gap Control Direction "mcGapCtrlDirectionPositive"

CA-Gruppe

- Die eingestellte Gap-Regelungs-Richtung gilt für die ganze CA-Gruppe.

Nachfolger

- Geregelt wird jeweils die Größe des Gaps.
- Der Nachfolger regelt als einziger Mover die Größe des Gaps, das von beiden benachbarten Movern eingeschlossen wird.

Gap Control Mode

- Für die Berechnung der Dynamik-Werte eines jeweiligen direkt hinterherfahrenden Movers kann der Gap Control Mode „mcGapCtrlModeStandard“ oder der Gap Control Mode „mcGapCtrlModeFast“ verwendet werden.
- Der Initialisierungs-Parameter Default Gap Control Mode stellt für jeden Nachfolger innerhalb einer CA-Gruppe den gleichen Gap Control Mode als Default-Algorithmus ein.

Individuell

- Sie können den Gap Control Mode mit jedem der Motion-Funktionsbausteine - MC_MoveAbsoluteCA, MC_MoveRelativeCA, MC_HaltCA oder MC_GearInPosCA - individuell für jeweils einen Mover ändern.

Rechenleistung

- Für den Gap Control Mode „mcGapCtrlModeStandard“ wird im Allgemeinen weniger Rechenleistung benötigt als für den Gap Control Mode „mcGapCtrlModeFast“.

Gap Control Direction „mcGapCtrlDirectionBoth“

- Bewegungs-Profile**
 - Erlaubt allgemeinere Bewegungs-Profile, zum Beispiel eine Rückwärts-Bewegung.
- CA-Gruppe**
 - Die eingestellte Gap-Regelungs-Richtung gilt für die ganze CA-Gruppe.
- Nachfolger und Vorgänger**
 - Geregelt wird jeweils die Größe des Gaps.
 - Beide benachbarte Mover, Vorgänger und Nachfolger, regeln die Größe des von ihnen eingeschlossenen Gaps.
- mcGapCtrlMode Standard**
 - Jeder Mover innerhalb einer CA-Gruppe verwendet für die Berechnung seiner Dynamik Werte den gleichen Gap Control Mode: „mcGapCtrlModeStandard“.
 - Der Gap Control Mode „mcGapCtrlModeFast“ kann nicht verwendet werden.



Wenn Sie für eine CA-Gruppe die Gap-Regelungs-Richtung „mcGapCtrlDirectionBoth“ eingestellt haben und wenn Sie mit einem Motion-Funktionsbaustein - MC_MoveAbsoluteCA, MC_MoveRelativeCA, MC_HaltCA oder MC_GearInPosCA - den Gap Control Mode für einen Mover dieser CA-Gruppe auf den Wert „mcGapCtrlModeFast“ stellen, dann wird ein Laufzeit-Fehler ausgelöst.

Korrelierendes Regelungsverhalten

- Nachbarschaft**
 - Geregelt wird die Größe eines Gaps, jeweils zwischen zwei benachbarten Movern.
 - Zwei Gaps sind (direkt) benachbart, wenn der sie trennende Mover gleichzeitig Nachfolger und Vorgänger ist.
- Kette**
 - (Jeweils direkt) Benachbarte Gaps bilden eine (nicht-triviale) Kette.
 - Innerhalb einer Kette korreliert die Regelung der jeweiligen Gaps.
- Gap Control Mode**
 - Der Gap Control Mode beeinflusst die Art und Weise der korrelierenden Regelung.
 - Der Gap Control Mode „mcGapCtrlModeStandard“ erlaubt für eine weichere Regelung Abweichungen vom Soll-Gap-Wert für ein einzelnes Gap und für die Gaps einer Kette.

8.1.4 MC_GearInPosDefaultDynamicsAfterSync

```

TYPE MC_GearInPosDefaultDynamicsAfterSync :
(
  MaximumSlaveDynamics := 16#0,
  JobDynamics := 16#1
);
END_TYPE
    
```

Legt die Default Dynamics fest, die für das Kommando MC_GearInPosCA verwendet werden, nachdem die Slave-Achse zum ersten Mal synchron geworden ist (siehe [ST_GearInPosCAOptions](#) [[▶ 46](#)]).

- MaximumSlaveDynamics:** Die maximale Slave-Achsendynamik (Geschwindigkeit, Beschleunigung, Verzögerung) wird als Standardwert für die AfterSyncDynamics verwendet. Der Ruck ist nicht beschränkt.
- JobDynamics:** Die Job Dynamics (GearInPosCAs Geschwindigkeit, Beschleunigung, Verzögerung und Ruck) wird als Standardwert für die AfterSyncDynamics verwendet.

8.2 Geo-Kompensation

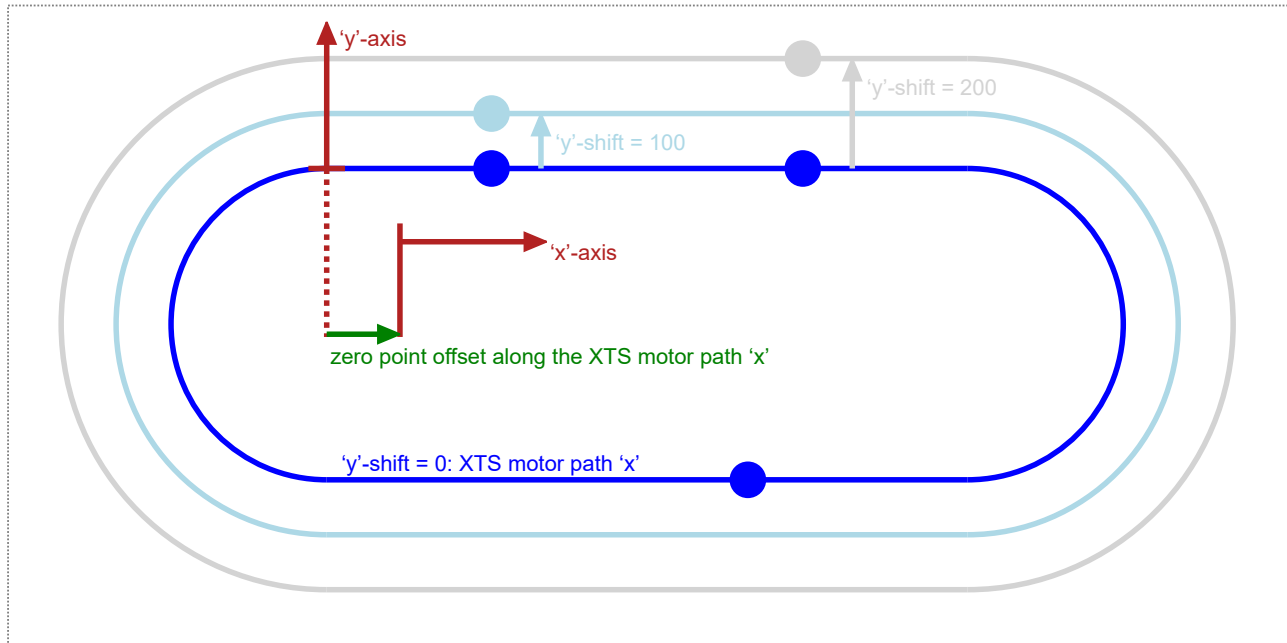


Abb. 1: Starterkit-Geometrie.

Geo-Kompensation: Motivation

Die Geo-Kompensation definiert einen zusätzlichen Freiheitsgrad:

- Eine eindimensionale räumliche Transformation der Steuerung der Bahndynamik.
- Die Positionsbewegungssteuerung bezieht sich immer auf die XTS-Motorbahn.

Als zusätzliche Dimension wird eine y -Achse senkrecht zur Koordinate der XTS-Motorbahn eingeführt. Die Bahndynamik kann für eine vordefinierte Bahn, die auf dieser y -Verschiebungskomponente liegt, gesteuert werden. Diese Bahn kann eine verbesserte Bahndynamik der Mover ermöglichen.

- Die Bahndynamik bezieht sich auf das Geschwindigkeits-, Beschleunigungs-, Verzögerungs- und Ruckverhalten auf einer Bahn.

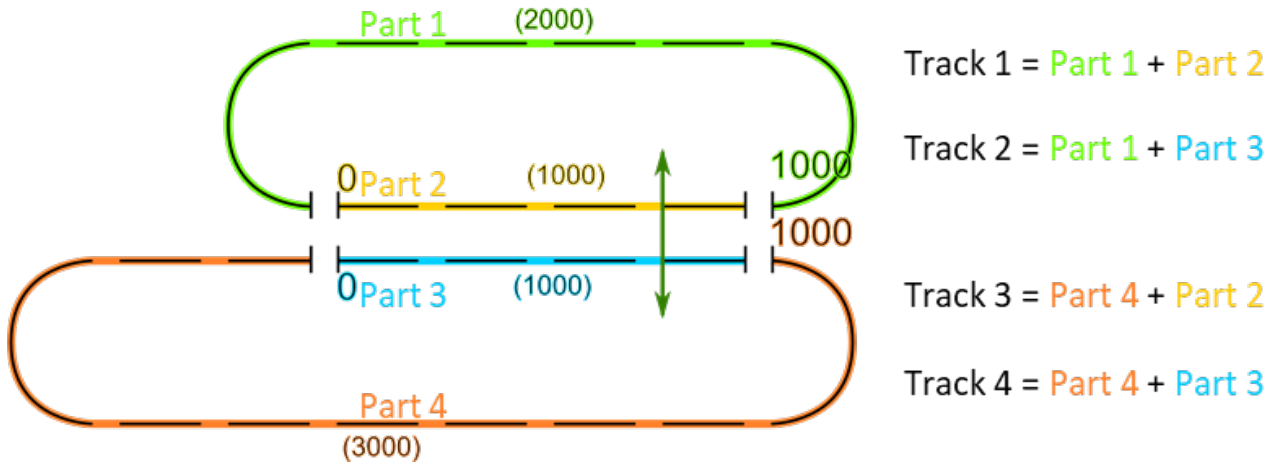
Im Allgemeinen kann der dynamische Bezug auf die XTS-Motorbahn beschränkt bleiben, so dass die Nutzung der Geo-Kompensation als optionale Möglichkeit verbleibt.

Motivationsbeispiel: Schwerpunkt

Bei vielen XTS-Anwendungen werden schwere Werkzeuge oder Produkte auf die Mover montiert bzw. geladen. Daraus ergibt sich, dass der Gesamtschwerpunkt (Mover + Werkzeug + Produkt) nicht zwingend auf der XTS-Motorbahn ist. Auf geraden XTS-Segmenten sind die Geschwindigkeit der XTS-Motorbahn und die Geschwindigkeit der Schwerpunktbahn identisch. Auf gebogenen XTS-Segmenten sind diese Bahngeschwindigkeiten jedoch unterschiedlich. Dieser Unterschied führt zu einer Beschleunigung oder Verzögerung auf der Schwerpunktbahn, während die Geschwindigkeit der XTS-Motorbahn konstant bleibt. Dadurch wirken auf der XTS-Strecke unbeabsichtigte Kräfte, insbesondere bei der Ein- oder Ausfahrt in Kurven. Um einige dieser Kräfte zu vermeiden oder möglichst gering zu halten, könnte der Schwerpunkt mit nahezu konstanter Geschwindigkeit verfahren werden. Dieses Verhalten ist ein Beispiel dafür, was mit der Geo-Kompensation erreicht werden kann: Solange ein Mover und seine Ladung nicht verändert werden, kann die Schwerpunktbahn beschrieben und dynamisch gesteuert werden, indem der XTS-Motorbahn eine radiale Verschiebung hinzugefügt wird. Da diese Verschiebung senkrecht von der XTS-Motorbahn weg zeigt, wird sie als y -Verschiebung bezeichnet.

8.3 Track-Management

Mithilfe des Track-Managements kann ein XTS-Aufbau in einzelne, räumlich getrennte XTS-Parts, unterteilt werden. Diese können nur ein, aber auch beliebig viele aufeinanderfolgende Motormodule umfassen. Einzelne, aneinander angrenzende XTS-Parts können zu sogenannten XTS Tracks zusammengefasst werden. Die Konfiguration der XTS-Parts und XTS-Tracks wird über den XTS-Konfigurator vorgenommen. Im Systemmanager werden die XTS-Parts und XTS-Tracks als TcCOM-Module, mit eindeutiger ObjectID, unterhalb der XTS Processing Unit als Kindknoten eingefügt (siehe XTS-Dokumentation).



Für jede einzelne Achse kann ein Track über den Baustein MC_ActivateTrack mithilfe der ObjectID des XTS-Tracks aktiviert werden. Bei der Aktivierung eines XTS-Tracks muss sich der Mover auf einem XTS-Part befinden, der dem Track zugeordnet ist. Mit der ObjectID 0 lässt sich das absolute Referenzsystem wieder für die einzelne Achse aktivieren. Die aktuellen Soll-Positionen auf den Tracks und Parts können über den Baustein MC_ReadTrackPositions ausgelesen werden.

9 SPS-Bibliotheken

9.1 Tc3_McCollisionAvoidance

Übersicht

Funktionsbaustein	Beschreibung
Motion	
MC_MoveAbsoluteCA [► 34]	Bewegt eine einzelne Achse mit Collision Avoidance in eine absolute Position.
MC_MoveRelativeCA [► 37]	Bewegt eine einzelne Achse mit Collision Avoidance über eine relative Strecke.
MC_HaltCA [► 39]	Hält eine einzelne Achse mit Collision Avoidance an, ohne sie für weitere Bewegungskommandos zu sperren.
MC_GearInPosCA [► 41]	Koppelt eine Slave-Achse mit einem Getriebefaktor und Collision Avoidance an eine Master-Achse.
MC_ReadTrackPositions [► 44]	Liefert die aktuellen XTS Track- und XTS Part-Soll-Positionen mit den zugehörigen Objekt Ids.
MC_ActivateTrack [► 45]	Aktiviert einen Track als Referenzsystem, das anschließend in verschiedenen Motion Bausteinen für die Positionierung benutzt werden kann.

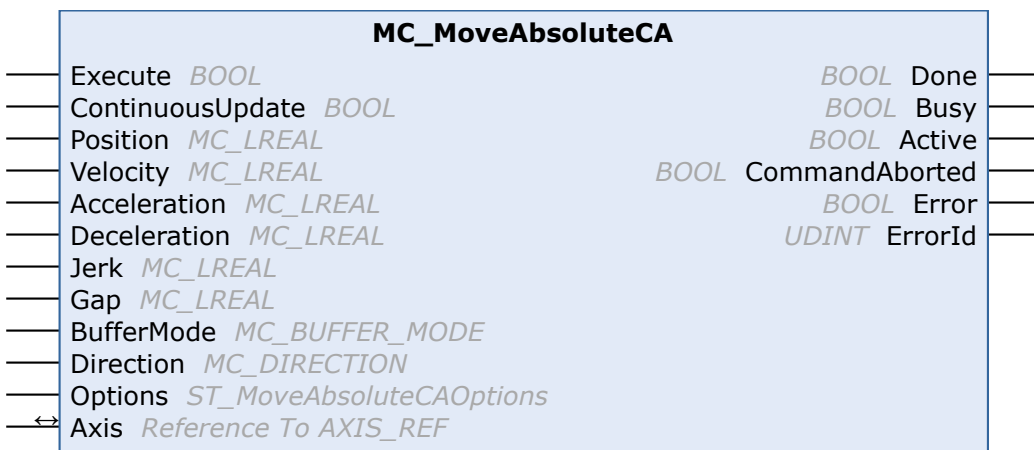
Strukturen und Aufzählungen

Name	Beschreibung
ST_GearInPosCAOptions [► 46]	Optionen für MC_GearInPosCA [► 41] .
ST_MoveAbsoluteCAOptions [► 49]	Optionen für MC_MoveAbsoluteCA [► 34] .
ST_MoveRelativeCAOptions [► 50]	Optionen für MC_MoveRelativeCA [► 37] .
ST_HaltCAOptions [► 51]	Optionen für MC_HaltCA [► 39] .
MC_GAP_CONTROL_MODE [► 52]	Legt den Gap Control Mode auf Ebene des Funktionsbausteins fest.

9.1.1 Funktionsbausteine

9.1.1.1 Motion

9.1.1.1.1 MC_MoveAbsoluteCA



Dieser Funktionsbaustein befiehlt einer einzelnen Achse, sich mit Collision Avoidance in die im Funktionsbaustein festgelegte absolute Position zu bewegen. Die Collision Avoidance hat eine höhere Priorität als das Bewegungskommando. Daher könnte die Achse während der Ausführung des Bewegungskommandos langsamer werden oder warten, um eine Kollision zu vermeiden. Der Funktionsbaustein gibt das Signal `Done` jedoch erst dann aus, wenn die Achse ihre Zielposition erreicht hat.

 **VAR_INPUT**

```

VAR_INPUT
  Execute           : BOOL;
  ContinuousUpdate : BOOL;
  Position          : MC_LREAL := MC_INVALID;
  Velocity          : MC_LREAL := MC_INVALID;
  Acceleration      : MC_LREAL := MC_DEFAULT;
  Deceleration      : MC_LREAL := MC_DEFAULT;
  Jerk              : MC_LREAL := MC_DEFAULT;
  Gap               : MC_LREAL := MC_DEFAULT;
  BufferMode         : MC_BUFFER_MODE := mcAborting;
  Direction          : Tc3_Mc3Definitions.MC_DIRECTION;
  Options           : ST_MoveAbsoluteCAOptions;
END_VAR
    
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
ContinuousUpdate	BOOL	In dieser Version steht die kontinuierliche Aktualisierung nur für den <code>Gap</code> zur Verfügung.
Position	MC_LREAL	<p>Vorgegebene absolute Zielposition für das Kommando.</p> <p>Ab TF5400 V3.1.10.30: Wenn mittels Modulo positioniert wird (<code>Direction ! = mcDirectionNonModulo</code>), muss die Zielposition im <code>Interval[0, ModuloFaktor]</code> liegen. Zusätzliche Umdrehungen werden über den Parameter <code>ST_MoveAbsoluteCAOptions.AdditionalTurns</code> kommandiert.</p> <p>Wenn sich die Zielposition innerhalb des <code>Tolerance Window</code> befindet, so wird die <code>Direction = mcDirectionPositive</code> und <code>Direction = mcDirectionNegative</code> für die Position ohne zusätzliche Umdrehungen ignoriert.</p> <p>Bis TF5400 V3.1.10.14 werden zusätzliche Umdrehungen kommandiert, indem größere Zielpositionen als der <code>ModuloFaktor</code> kommandiert werden.</p> <p>Weitere Details in den Hinweisen zur <u>Modulo-Positionierung</u> [▶ 116].</p>
Velocity	MC_LREAL	Die Geschwindigkeit ist durch die maximale Achsgeschwindigkeit beschränkt. Wird kein Wert eingegeben, dann führt dies zu einem Fehler, weil es keine Standardgeschwindigkeit gibt.
Acceleration	MC_LREAL	Die Beschleunigung ist durch die maximale Achsbeschleunigung beschränkt. Wird kein Wert eingegeben, dann wird die Standardachsbeschleunigung verwendet.
Deceleration	MC_LREAL	Die Verzögerung ist durch die maximale Achsverzögerung beschränkt. Wird kein Wert eingegeben, dann wird die Standardachsverzögerung verwendet.

Name	Typ	Beschreibung
Jerk	MC_LREAL	Wird kein Wert eingegeben, dann wird der Standardachsdruck verwendet. Der maximale Ruck ist nicht beschränkt.
Gap	MC_LREAL	Dieser Wert bestimmt den Mindestabstand zum Vorgänger für die Collision Avoidance. Wird kein Wert eingegeben, dann wird der Standardwert der Gruppe verwendet. Hinweis Bei Verwendung der Geo-Kompensation ist der Abstand besonders zu beachten. Der Mover-Abstand für die Collision Avoidance bezieht sich positionsmäßig und dynamisch immer auf die verschobene Bahngeometrie. Da sich der Abstand bei Verwendung der Geo-Kompensation auf die verschobene Bahn bezieht, können benachbarte Mover in den Kurven kollidieren, wenn er zu gering festgelegt wird. Achten Sie auf ausreichenden Abstand.
BufferMode	MC_BUFFER_MODE	In dieser Version stehen nur <code>mcAborting</code> und <code>mcBuffered</code> zur Verfügung (siehe MC_BUFFER_MODE [▶ 104]).
Direction (verfügbar ab V3.1.10.1)	Tc3_Mc3Definitions.MC_DIRECTION	Definiert die Richtung der Bewegung (default <code>mcDirectionNonModulo</code>), siehe MC_DIRECTION [▶ 108].
Optionen	ST_MoveAbsoluteCAOptions	Weitere Informationen über die verfügbaren Optionen (ab V3.1.2.47) finden Sie in der Dokumentation von ST_MoveAbsoluteCAOptions [▶ 49].

i Die Achse erreicht nicht die Zielgeschwindigkeit, -beschleunigung oder -verzögerung

Die Werte für die Geschwindigkeit, Beschleunigung oder Verzögerung könnten automatisch auf die maximale Achsgeschwindigkeit, -beschleunigung und -verzögerung begrenzt werden. Prüfen Sie die Parameter Maximum Dynamics und Default Dynamics der Achse. Möglich ist auch, dass die Werte von Maximum Dynamics kleiner sind als die Default Dynamics.

VAR_IN_OUT

```
VAR_IN_OUT
  Axis          : AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
Axis	AXIS_REF	Referenz auf eine Achse (siehe AXIS_REF).

VAR_OUTPUT

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

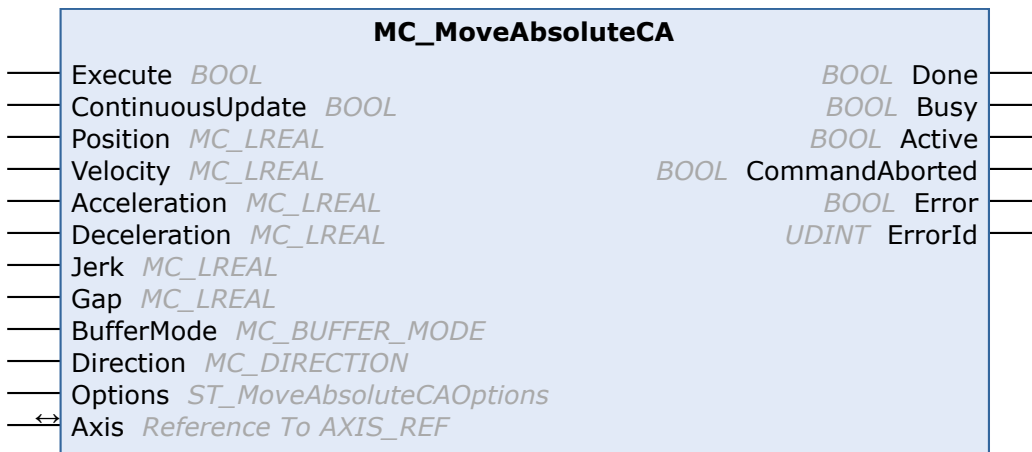
Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando erfolgreich ausgeführt worden ist.

Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird TRUE , wenn das Kommando mit <code>Execute</code> gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn <code>Busy</code> wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge <code>Done</code> , <code>CommandAborted</code> oder <code>Error</code> gesetzt.
Active	BOOL	Wenn <code>Active</code> TRUE ist, steuert der Funktionsbaustein 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.
ErrorId	UDINT	Enthält den befehlspezifischen 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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.1.1.2 MC_MoveRelativeCA



Dieser Funktionsbaustein befiehlt einer einzelnen Achse, sich mit Collision Avoidance über die im Funktionsbaustein festgelegte relative Strecke zu bewegen. Die Collision Avoidance hat eine höhere Priorität als das Bewegungskommando. Während der Ausführung des Bewegungskommandos könnte die Achse langsamer werden oder warten, um eine Kollision zu vermeiden. Der Funktionsbaustein gibt das Signal `Done` jedoch erst dann aus, wenn die Achse die geforderte Strecke zurückgelegt hat.

VAR_INPUT

```

VAR_INPUT
  Execute           : BOOL;
  ContinuousUpdate : BOOL;
  Distance          : MC_LREAL := MC_INVALID;
  Velocity          : MC_LREAL := MC_INVALID;
  Acceleration      : MC_LREAL := MC_DEFAULT;
  Deceleration      : MC_LREAL := MC_DEFAULT;
  
```

```

Jerk          : MC_LREAL := MC_DEFAULT;
Gap          : MC_LREAL := MC_DEFAULT;
BufferMode   : MC_BUFFER_MODE := mcAborting;
Options      : ST_MoveRelativeCAOptions;
END_VAR

```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
ContinuousUpdate	BOOL	In dieser Version steht die kontinuierliche Aktualisierung nur für den <code>Gap</code> zur Verfügung.
Distance	MC_LREAL	Vorgegebene relative Strecke für das Kommando.
Velocity	MC_LREAL	Die Geschwindigkeit ist durch die maximale Geschwindigkeit der Achse beschränkt. Wird kein Wert eingegeben, dann führt dies zu einem Fehler, weil es keine Standardgeschwindigkeit gibt.
Acceleration	MC_LREAL	Die Beschleunigung ist durch die maximale Achsbeschleunigung beschränkt. Wird kein Wert eingegeben, dann wird die Standardachsbeschleunigung verwendet.
Deceleration	MC_LREAL	Die Verzögerung ist durch die maximale Achsverzögerung beschränkt. Wird kein Wert eingegeben, dann wird die Standardachsverzögerung verwendet.
Jerk	MC_LREAL	Wird kein Wert eingegeben, dann wird der Standardachsdruck verwendet. Der maximale Ruck ist nicht beschränkt.
Gap	MC_LREAL	Dieser Wert bestimmt den Mindestabstand zum Vorgänger für die Collision Avoidance. Wird kein Wert eingegeben, dann wird der Standardwert der Gruppe verwendet. Hinweis Bei Verwendung der Geo-Kompensation ist der Abstand besonders zu beachten. Der Mover-Abstand für die Collision Avoidance bezieht sich positionsmäßig und dynamisch immer auf die verschobene Bahngeometrie. Da sich der Abstand bei Verwendung der Geo-Kompensation auf die verschobene Bahn bezieht, können benachbarte Mover in den Kurven kollidieren, wenn er zu gering festgelegt wird. Achten Sie auf ausreichenden Abstand.
BufferMode	MC_BUFFER_MODE	In dieser Version stehen nur <code>mcAborting</code> und <code>mcBuffered</code> zur Verfügung (siehe MC_BUFFER_MODE [► 104]).
Optionen	ST_MoveRelativeCAOptions	Weitere Informationen über die verfügbaren Optionen (ab V3.1.2.47) finden Sie in der Dokumentation von ST_MoveRelativeCAOptions [► 50] .

i Die Achse erreicht nicht die Zielgeschwindigkeit, -beschleunigung oder -verzögerung

Die Werte für die Geschwindigkeit, Beschleunigung oder Verzögerung könnten automatisch auf die maximale Achsgeschwindigkeit, -beschleunigung und -verzögerung begrenzt werden. Prüfen Sie die Parameter Maximum Dynamics und Default Dynamics der Achse. Möglich ist auch, dass die Werte von Maximum Dynamics kleiner sind als die Default Dynamics.

VAR_IN_OUT

```

VAR_IN_OUT
  Axis          : AXIS_REF;
END_VAR

```

Name	Typ	Beschreibung
Axis	AXIS_REF	Referenz auf eine Achse (siehe AXIS_REF).

VAR_OUTPUT

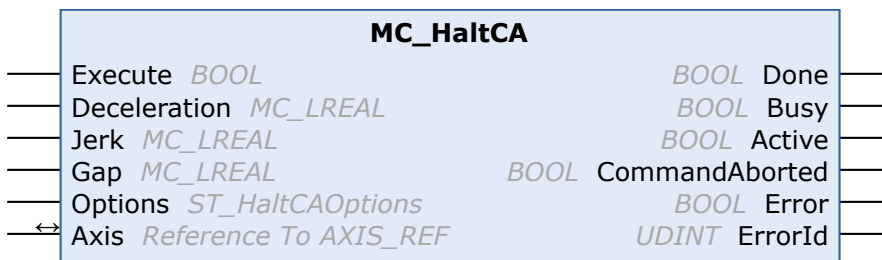
```
VAR_OUTPUT
  Done           : BOOL;
  Busy           : BOOL;
  Active         : BOOL;
  CommandAborted : BOOL;
  Error          : BOOL;
  ErrorId        : UDINT;
END_VAR
```

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 <code>Execute</code> gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn <code>Busy</code> wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge <code>Done</code> , <code>CommandAborted</code> oder <code>Error</code> gesetzt.
Active	BOOL	Wenn <code>Active</code> TRUE ist, steuert der Funktionsbaustein 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.
ErrorId	UDINT	Enthält den befehlspezifischen 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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.1.1.3 MC_HaltCA



Dieser Funktionsbaustein befiehlt einer einzelnen Achse, mit Collision Avoidance anzuhalten.

VAR_INPUT

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

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
Deceleration	MC_LREAL	Die Verzögerung ist durch die maximale Achsverzögerung beschränkt. Wird kein Wert eingegeben, dann wird die Standardachsverzögerung verwendet.
Jerk	MC_LREAL	Wird kein Wert eingegeben, dann wird der Standardachsruck verwendet. Der Ruck ist nicht beschränkt.
Gap	MC_LREAL	Dieser Wert bestimmt den Mindestabstand zum Vorgänger für die Collision Avoidance. Wird kein Wert eingegeben, dann wird der Standardwert der Gruppe verwendet. Bei Verwendung der Geo-Kompensation ist der Abstand besonders zu beachten. Da sich der Abstand bei Verwendung der Geo-Kompensation auf die verschobene Bahn bezieht, können benachbarte Mover in den Kurven kollidieren, wenn er zu gering festgelegt wird.
Options	ST_HaltCA Options	Weitere Informationen über die verfügbaren Optionen (ab V3.1.2.47) finden Sie in der Dokumentation von ST_HaltCAOptions [► 51].

i Die Achse hält nicht schnell genug an

Die gegebene Verzögerung könnte automatisch auf die maximale Achsverzögerung beschränkt werden. Prüfen Sie die Parameter Maximum Dynamics und Default Dynamics der Achse. Möglich ist auch, dass die Werte von Maximum Dynamics unterhalb der Default Dynamics liegen.

VAR_IN_OUT

```
VAR_IN_OUT
  Axis          : AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
Axis	AXIS_REF	Referenz auf eine Achse (siehe AXIS_REF).

VAR_OUTPUT

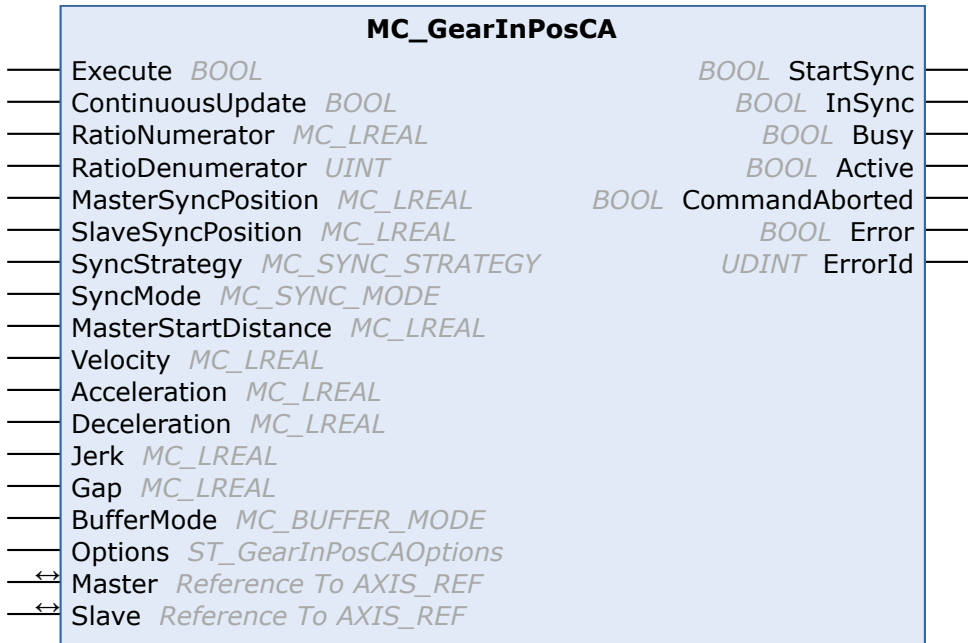
```
VAR_OUTPUT
  Done          : BOOL;
  Busy          : BOOL;
  Active        : BOOL;
  CommandAborted : BOOL;
  Error         : BOOL;
  ErrorId       : UDINT;
END_VAR
```

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 FAL-SE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge Done, CommandAborted oder Error gesetzt.
Active	BOOL	Wenn Active TRUE ist, steuert der Funktionsbaustein 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.
ErrorId	UDINT	Enthält den befehlspezifischen 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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.1.1.4 MC_GearInPosCA



Dieser Funktionsbaustein koppelt eine Slave-Achse an eine Master-Achse. Die gesetzten Werte bilden immer die Quelle für die Master-Werte. Die Collision Avoidance hat eine höhere Priorität als die Achskopplung. Die Slave-Achse kann entkoppelt werden, indem ein Bewegungskommando unter Verwendung des Puffermodus BufferMode mcAborting gesendet wird.

Optimierungen bzgl. MC_GearInPosCA ab TF5400 v3.2.27

- Optimierungen am MC_GearInPosCA, die einen SAF-Zyklusversatz zwischen Master- und Slaveachse unterbinden.
- Optimierungen am Gap Controller, wenn die Achse sich schon in der Zielposition befindet und nur der Gap sich ändert. Wird der benachbarte Mover kommandiert, so wirkt der neue Gap.

 **VAR_INPUT**

```

VAR_INPUT
  Execute           : BOOL;
  ContinuousUpdate : BOOL;
  RatioNumerator   : MC_LREAL := 1;
  RatioDenominator : UINT := 1;
  MasterSyncPosition : MC_LREAL := MC_INVALID;
  SlaveSyncPosition : MC_LREAL := MC_INVALID;
  SyncStrategy      : MC_SYNC_STRATEGY := mcSyncStrategyLate;
  SyncMode          : MC_SYNC_MODE;
  MasterStartDistance : MC_LREAL := MC_IGNORE;
  Velocity           : MC_LREAL := MC_INVALID;
  Acceleration       : MC_LREAL := MC_DEFAULT;
  Deceleration       : MC_LREAL := MC_DEFAULT;
  Jerk               : MC_LREAL := MC_DEFAULT;
  Gap                : MC_LREAL := MC_DEFAULT;
  BufferMode          : MC_BUFFER_MODE := mcAborting;
  Options            : ST_GearInPosCAOptions;
END_VAR
    
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
ContinuousUpdate	BOOL	In dieser Version steht die kontinuierliche Aktualisierung nur für den Gap zur Verfügung.
RatioNumerator	MC_LREAL	Ein Getriebefaktor kann gesetzt werden, indem an den Eingängen <code>RatioNumerator</code> und <code>RatioDenominator</code> ganzzahlige Werte eingegeben werden oder indem für den <code>RatioNumerator</code> ein Dezimalwert eingegeben und der <code>RatioDenominator</code> unverändert gelassen wird (der Standardwert ist 1). Der Getriebefaktor wird als Faktor eingegeben, z. B. bedeutet der Wert 0.8, dass die Slave-Geschwindigkeit $0.8 * \text{master axis velocity}$ (oder 80 % der Geschwindigkeit der Master-Achse) beträgt. Der Wert für den Faktor ist nicht beschränkt, er könnte größer als 1.0 oder negativ sein.
RatioDenominator	UINT	Nenner des Getriebefaktors.
MasterSyncPosition	MC_LREAL	Dieser Eingang weist den Typ <code>LREAL</code> auf. Position des Masters, an der der Slave <code>InSync</code> ist und den richtigen Getriebefaktor aufweist.
SlaveSyncPosition	MC_LREAL	Dieser Eingang weist den Typ <code>LREAL</code> auf. Position des Slaves, an der er mit dem richtigen Getriebefaktor <code>InSync</code> ist.
SyncStrategy	MC_SYNC_STRATEGY	Definiert die Strategie, die der Slave für die Synchronisation verwendet (siehe MC_SYNC_STRATEGY [► 109]). Die default Strategie ist <code>mcSyncStrategyLate</code> .
SyncMode (verfügbar ab V3.1.10.1)	MC_SYNC_MODE	Definiert die Richtung, wie die <code>SlaveSyncPosition</code> interpretiert werden soll, siehe MC_SYNC_MODE [► 109] .
MasterStartDistance	MC_LREAL	Wenn ein positiver Wert gesetzt wird, beginnt die Slave-Achse erst mit der Synchronisation, wenn die Master-Position größer oder gleich ist ($\text{MasterSyncPosition} - \text{MasterStartDistance}$). Wenn ein negativer Wert gesetzt wird, beginnt die Synchronisation erst, wenn die Master-Position kleiner oder gleich ist ($\text{MasterSyncPosition} - \text{MasterStartDistance}$). Wenn <code>MasterStartDistance</code> nicht gesetzt wird, beginnt der Slave mit der Synchronisation, so-bald der Funktionsbaustein das Signal <code>Active</code> gibt. Das genaue Verhalten der Slave-Achse während der Synchronisationsphase hängt von der <code>SyncStrategy</code> ab.
Velocity	MC_LREAL	Maximale Geschwindigkeit der Slave-Achse während der Synchronisationsphase. Die Geschwindigkeit ist durch die maximale Geschwindigkeit der Slave-Achse beschränkt. Wird kein Wert eingegeben, dann führt dies zu einem Fehler, weil es keine Standardgeschwindigkeit gibt.
Acceleration	MC_LREAL	Maximale Beschleunigung der Slave-Achse während der Synchronisationsphase. Die Beschleunigung ist durch die maximale Beschleunigung der Slave-Achse beschränkt. Wird kein Wert eingegeben, dann wird die Standardbeschleunigung der Slave-Achse verwendet.
Deceleration	MC_LREAL	Maximale Verzögerung der Slave-Achse während der Synchronisationsphase. Die Verzögerung ist durch die maximale Verzögerung der Slave-Achse beschränkt. Wird kein Wert eingegeben, dann wird die Standardverzögerung der Slave-Achse verwendet.
Jerk	MC_LREAL	Maximaler Ruck der Achse während der Synchronisationsphase. Wird kein Wert eingegeben, dann wird der Standardruck der Slave-Achse verwendet. Der Jerk ist nicht beschränkt.

Name	Typ	Beschreibung
Gap	MC_LREAL	Dieser Wert bestimmt den Mindestabstand zum Vorgänger für die Collision Avoidance. Wird kein Wert eingegeben, dann wird der Standardwert der Gruppe verwendet. Hinweis Bei Verwendung der Geo-Kompensation ist der Abstand besonders zu beachten. Der Mover-Abstand für die Collision Avoidance bezieht sich positionsmäßig und dynamisch immer auf die verschobene Bahngeometrie. Da sich der Abstand bei Verwendung der Geo-Kompensation auf die verschobene Bahn bezieht, können benachbarte Mover in den Kurven kollidieren, wenn er zu gering festgelegt wird. Achten Sie auf ausreichenden Abstand.
BufferMode	MC_BUFFER_MODE	In dieser Version stehen nur <code>mcAborting</code> und <code>mcBuffered</code> zur Verfügung (siehe MC_BUFFER_MODE [▶ 104]).
Optionen	ST_GearInPosCAOptions	Die <code>Options</code> können verwendet werden, um das Synchronisationsprofil des Slaves ergänzend zur <code>SyncStrategy</code> zu beeinflussen (ab V3.1.2.47) (siehe ST_GearInPosCAOptions [▶ 46]).

i Die Achse erreicht nicht die Zielgeschwindigkeit, -beschleunigung oder -verzögerung

Die Werte für die Geschwindigkeit, Beschleunigung oder Verzögerung könnten automatisch auf die maximale Achsgeschwindigkeit, -beschleunigung und -verzögerung begrenzt werden. Prüfen Sie die Parameter Maximum Dynamics und Default Dynamics der Achse. Möglich ist auch, dass die Werte von Maximum Dynamics kleiner sind als die Default Dynamics.

VAR_IN_OUT

```
VAR_IN_OUT
  Master   : AXIS_REF;
  Slave    : AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
Master	AXIS_REF	Referenz auf eine Achse (siehe AXIS_REF).
Slave	AXIS_REF	Referenz auf eine Achse (siehe AXIS_REF).

VAR_OUTPUT

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

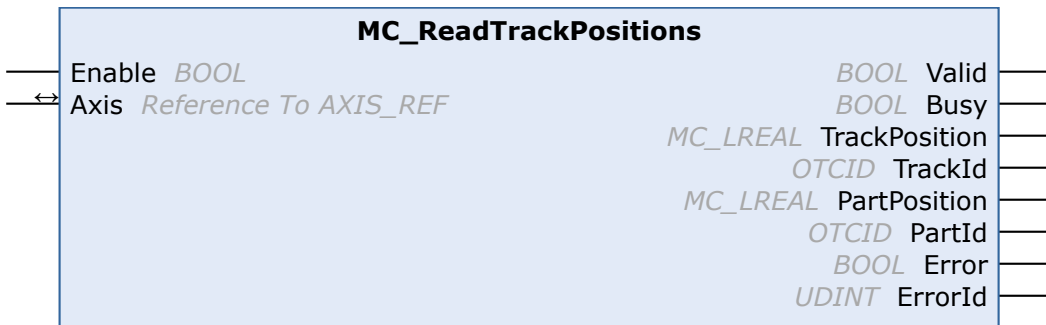
Name	Typ	Beschreibung
StartSync	BOOL	Dieser Ausgang wird gesetzt, sobald der Slave aktiv mit der Synchronisation beginnt, und zu-rückgesetzt, sobald der Slave InSync ist.
InSync	BOOL	Dieser Ausgang wird TRUE, sobald der Slave synchronisiert ist. Ist die Dynamik der Slave-Achse zu gering, um der Bewegung der Master-Achse zu folgen, könnte der Ausgang <code>InSync</code> auf FALSE zurückgesetzt werden, woraufhin die Slave-Achse erneut mit der Synchronisation beginnt.

Name	Typ	Beschreibung
Busy	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando mit <code>Execute</code> gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn <code>Busy</code> wieder <code>FALSE</code> wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge <code>CommandAborted</code> oder <code>Error</code> gesetzt.
Active	BOOL	Wenn <code>Active</code> <code>TRUE</code> ist, steuert der Funktionsbaustein die Achse.
CommandAborted	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando durch ein anderes Kommando unterbrochen worden ist.
Error	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlspezifischen 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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.1.1.5 MC_ReadTrackPositions



Dieser Funktionsbaustein liefert die aktuellen XTS Track- und XTS Part-Soll-Positionen mit den zugehörigen Objekt IDs. Die Achse muss sich in einer CA Gruppe befinden, damit der Baustein gültige Werte liefern kann. Ist kein Track für die Achse aktiviert, werden die aktuellen absoluten Sollwerte, mit der Track-/PartId = 0, zurückgeliefert.

VAR_INPUT

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

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

VAR_IN_OUT

```
VAR_IN_OUT
  Axis : AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
Axis	AXIS_REF	Referenz auf eine Achse (siehe AXIS_REF).

VAR_OUTPUT

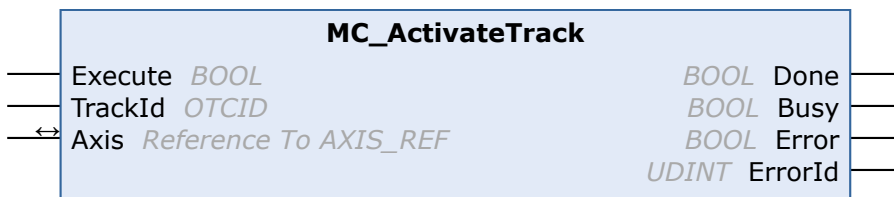
```
VAR_OUTPUT
  Valid      : BOOL;
  Busy       : BOOL;
  TrackPosition : LREAL;
  TrackId     : OTCID;
  PartPosition : LREAL;
  PartId      : OTCID;
  Error       : BOOL;
  ErrorId     : UDINT;
END_VAR
```

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 <code>Execute</code> gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn <code>Busy</code> wieder FALSE wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge <code>Done</code> , <code>CommandAborted</code> oder <code>Error</code> gesetzt.
TrackPosition	LREAL	Position im aktiven Track-Referenzsystem.
TrackId	OTCID	Objekt ID des aktiven Track-Referenzsystems.
PartPosition	LREAL	Position auf dem aktuellen XTS-Part.
PartId	OTCID	Objekt ID des aktuellen XTS-Parts.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlspezifischen 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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.1.1.6 MC_ActivateTrack



Dieser Funktionsbaustein aktiviert einen Track als Referenzsystem, das anschließend in verschiedenen Motion Bausteinen für die Positionierung benutzt werden kann. Das XTS Track Objekt muss unter der XTS Processing Unit angelegt sein und wird dann über die Objekt ID ausgewählt. Die Konfiguration der XTS Tracks findet über den XTS Configurator statt (für weitere Informationen siehe [XTS-Dokumentation](#)). Mit der ObjectID 0 lässt sich das absolute Referenzsystem wieder aktivieren.

VAR_INPUT

```
VAR_INPUT
  Execute : BOOL;
  TrackId : OTCID;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Das Kommando wird durch eine steigende Flanke an diesem Eingang ausgelöst.
TrackId	OTCID	Objekt ID des aktiven Track-Referenzsystems.

VAR_IN_OUT

```
VAR_IN_OUT
  Axis          : AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
Axis	AXIS_REF	Referenz auf eine Achse (siehe AXIS_REF).

VAR_OUTPUT

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

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando mit <code>Execute</code> gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn <code>Busy</code> wieder <code>FALSE</code> wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge <code>Done</code> , <code>CommandAborted</code> oder <code>Error</code> gesetzt.
Error	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlspezifischen 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	Zielpattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4024.7 TF5400 Advanced Motion Pack V3.1.10.1	PC oder CX (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.2 Datentypen

9.1.2.1 ST_GearInPosCAOptions

Die Options können gesetzt werden, um das Synchronisationsprofil des Slaves zu spezialisieren.

```
TYPE ST_GearInPosCAOptions :
STRUCT
  AfterSyncVelocity          : MC_LREAL := MC_DEFAULT;
  AfterSyncAcceleration     : MC_LREAL := MC_DEFAULT;
  AfterSyncDeceleration     : MC_LREAL := MC_DEFAULT;
  AfterSyncJerk             : MC_LREAL := MC_DEFAULT;
  MasterVelocityUndershootAllowed : BOOL := TRUE;
  MasterVelocityOvershootAllowed : BOOL := TRUE;
  MinimalSlavePosition      : MC_LREAL := MC_IGNORE;
  DirectionReversalAllowed  : BOOL := TRUE;
  OverrideSlaveDynamicRestrictions : BOOL := FALSE;
  GapControlMode            : MC_GAP_CONTROL_MODE := mcGapControlModeGroupDefault;
  SlaveSyncPositionReferenceSystem : OTCID := 0;
```

```
DynamicsReferenceSystem : OTCID := 0;
END_STRUCT
END_TYPE
```

Name	Typ	Beschreibung
AfterSyncVelocity (Ab TF5400 V3.1.10.1)	MC_LREAL	Maximale Geschwindigkeit der Slave-Achse, nachdem sich diese zum ersten Mal aufsynchronisiert hat. Die Geschwindigkeit ist begrenzt durch die maximale Geschwindigkeit der Slave-Achse. Wenn es keinen Eingangswert gibt, wird der Default-Wert gesetzt (siehe CA-Group Parameter GearInPosAfterSyncDynamics).
AfterSyncAcceleration (Ab TF5400 V3.1.10.1)	MC_LREAL	Maximale Beschleunigung der Slave-Achse, nachdem sich diese zum ersten Mal aufsynchronisiert hat. Die Beschleunigung ist begrenzt durch die maximale Beschleunigung der Slave-Achse. Wenn es keinen Eingangswert gibt, wird der Default-Wert gesetzt (siehe CA-Group Parameter GearInPosAfterSyncDynamics).
AfterSyncDeceleration (Ab TF5400 V3.1.10.1)	MC_LREAL	Maximale Verzögerung der Slave-Achse, nachdem sich diese zum ersten Mal aufsynchronisiert hat. Die Verzögerung ist begrenzt durch die maximale Verzögerung der Slave-Achse. Wenn es keinen Eingangswert gibt, wird der Default-Wert gesetzt (siehe CA-Group Parameter GearInPosAfterSyncDynamics).
AfterSyncJerk (Ab TF5400 V3.1.10.1)	MC_LREAL	Maximaler Ruck der Slave-Achse, nachdem sich diese zum ersten Mal aufsynchronisiert hat. Der Ruck ist nicht begrenzt durch andere Ruck-Werte (ein maximaler Ruck der Achse kann nicht konfiguriert werden). Wenn es keinen Eingangswert gibt, wird der Default-Wert gesetzt (siehe CA-Group Parameter GearInPosAfterSyncDynamics).
MasterVelocityUndershootAllowed	BOOL	Diese Option wirkt sich nur auf das Synchronisationsprofil aus und hat keine Wirkung, sobald der Slave InSync ist. TRUE: Keine Beschränkungen für das Profil FALSE: Die Slave-Geschwindigkeit während der Synchronisationsphase ist immer größer oder gleich der Master-Geschwindigkeit. Wenn die Slave-Geschwindigkeit zum Zeitpunkt der Kommandoausgabe kleiner ist als die Master-Geschwindigkeit, beschleunigt der Slave mit seiner Synchronisationsdynamik, um die Master-Geschwindigkeit schnellstmöglich zu erreichen. MasterVelocityUndershootAllowed und MasterVelocityOvershootAllowed können nicht beide auf FALSE gesetzt werden.
MasterVelocityOvershootAllowed	BOOL	Diese Option wirkt sich nur auf das Synchronisationsprofil aus und hat keine Wirkung, sobald der Slave InSync ist. TRUE: Keine Beschränkungen für das Profil. FALSE: Die Slave-Geschwindigkeit während der Synchronisationsphase ist immer kleiner oder gleich der Master-Geschwindigkeit. Wenn die Slave-Geschwindigkeit zum Zeitpunkt der Kommandoausgabe größer ist als die Master-Geschwindigkeit, verzögert der Slave mit seiner Synchronisationsdynamik, um die Master-Geschwindigkeit zu erreichen. MasterVelocityUndershootAllowed und MasterVelocityOvershootAllowed können nicht beide auf FALSE gesetzt werden.
MinimalSlavePosition	MC_LREAL	Absolute minimale Position des Slaves während der Synchronisationsphase. Diese Option wirkt sich nur auf das Synchronisationsprofil aus und hat keine Wirkung, sobald der Slave In-Sync ist.

Name	Typ	Beschreibung
DirectionReversalAllowed	BOOL	<p>Diese Option wirkt sich nur auf das Synchronisationsprofil aus und hat keine Wirkung, sobald der Slave InSync ist.</p> <p>TRUE: Keine Beschränkungen für das Profil.</p> <p>FALSE: Die Richtung wird durch das Vorzeichen der Slave-Geschwindigkeit in der SlaveSyncPosition (Getriebefaktor * Master-Geschwindigkeit) bestimmt. Der Slave darf sich nicht in der entgegengesetzten Richtung bewegen.</p>
OverrideSlaveDynamicRestrictions	BOOL	<p>Diese Option wirkt sich nur auf das Synchronisationsprofil aus und hat keine Wirkung, sobald der Slave InSync ist. Sie wirkt sich nur aus, wenn die Strategien mcSyncStrategyLate oder mcSyncStrategySlow verwendet werden.</p> <p>FALSE: Das Synchronisationsprofil wird jedes Mal neu berechnet, wenn sich die Master-Geschwindigkeit ändert. Es kann ein Fehler auftreten, wenn innerhalb der im Funktionsbaustein GearInPosCA festgelegten Dynamikgrenzen kein gültiges Synchronisationsprofil generiert werden kann. Insbesondere ein verrauschtes Master-Signal kann zu einem solchen Fehler führen (z. B. Encoderachse). Des Weiteren kann es eine hohe Last zur Folge haben, wenn sich die Master-Geschwindigkeit häufig ändert, z. B. wenn der Master beschleunigt oder verzögert oder wenn das Master-Signal verrauscht ist.</p> <p>TRUE: Das Synchronisationsprofil wird bei einer Änderung der Geschwindigkeit des Masters nicht notwendigerweise neu berechnet. Stattdessen wird das ursprünglich berechnete Profil gedehnt oder komprimiert. Dadurch werden die oben beschriebenen Fehler (siehe FALSE) vermieden. Dies könnte jedoch zur Verletzung der im Funktionsbaustein GearInPosCA festgelegten Dynamikgrenzen führen (die Maximum Axis Dynamic Limits werden nicht verletzt). Diese Option kann für die Synchronisation auf eine verrauschte Master-Achse (z. B. Encoderachse) verwendet werden und kann zudem die Rechenzeit verkürzen.</p>
GapControlMode	MC_GAP_CONTROL_MODE	<p>Siehe die Beschreibung des Datentyps <code>MC_GAP_CONTROL_MODE</code> [► 52] für weitere Informationen.</p>
SlaveSyncPositionReferenceSystem (Ab TF5400 V3.1.6.03)	OTCID	<ul style="list-style-type: none"> Dieser Eingang weist den Typ <code>OTCID</code> auf und kann sich dadurch auf ein Mover Template beziehen. Für die Positionsreferenz einer synchronisierten Slave-Achse auf die XTS-Motorbahn kann der Eingang <code>SlaveSyncPositionReferenceSystem</code> offen gelassen oder auf den Wert null gesetzt werden, so dass die Kompatibilität mit früheren Versionen dieses Funktionsbausteins erhalten bleibt. Für die Positionsreferenz einer synchronisierten Slave-Achse auf die von einem Mover Template Object definierte Bahn setzen Sie <code>SlaveSyncPositionReferenceSystem</code> auf seine Objekt-ID. Dann wird der Positionseingang für die synchronisierte Slave-Achse entsprechend der verschobenen Bahn interpretiert.
DynamicsReferenceSystem (Ab TF5400 V3.1.6.03)	OTCID	<ul style="list-style-type: none"> Dieser Eingang weist den Typ <code>OTCID</code> auf und kann sich dadurch auf ein Mover Template beziehen. Für die dynamische Referenz auf die XTS-Motorbahn kann der Eingang <code>DynamicsReferenceSystem</code> offen gelassen oder auf den Wert null gesetzt werden, so dass die Kompatibilität mit früheren Versionen dieses Funktionsbausteins erhalten bleibt.

Name	Typ	Beschreibung
		<ul style="list-style-type: none"> Für die dynamische Referenz auf die von einem Mover Template Object definierte Bahn setzen Sie <code>DynamicsReferenceSystem</code> auf seine Objekt-ID. Dann wird die Dynamik des Bewegungsprofils auf die gegebene Bahn beschränkt.



Eine Beschränkung des Synchronisationsprofils könnte die Synchronisation für den Slave unmöglich machen.

Wenn die Synchronisation unmöglich ist, gibt `MC_GearInPosCA` [► 41] einen Fehler aus.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.2.2 ST_MoveAbsoluteCAOptions

```

TYPE ST_MoveAbsoluteCAOptions :
STRUCT
    GapControlMode          : MC_GAP_CONTROL_MODE := mcGapControlModeGroupDefault;
    PositionReferenceSystem : OTCID := 0;
    DynamicsReferenceSystem : OTCID := 0;
    AdditionalTurns         : UDINT := 0;
END_STRUCT
END_TYPE
    
```

Name	Typ	Beschreibung
GapControlMode	MC_GAP_CONTROL_MODE	Siehe die Beschreibung des Datentyps <code>MC_GAP_CONTROL_MODE</code> [► 52] für weitere Informationen.
PositionReferenceSystem (ab TF5400 V3.1.6.03)	OTCID	<ul style="list-style-type: none"> Dieser Eingang weist den Typ <code>OTCID</code> auf und kann sich dadurch auf ein Mover Template beziehen. Für die Positionsreferenz auf die XTS-Motorbahn kann der Eingang <code>PositionReferenceSystem</code> offengelassen oder auf den Wert null gesetzt werden, so dass die Kompatibilität mit früheren Versionen dieses Funktionsbausteins erhalten bleibt. Für die Positionsreferenz auf die von einem Mover Template Object definierte Bahn setzen Sie <code>PositionReferenceSystem</code> auf seine Objekt-ID. Dann wird der Positionseingang entsprechend der verschobenen Bahn interpretiert.
DynamicsReferenceSystem (ab TF5400 V3.1.6.03)	OTCID	<ul style="list-style-type: none"> Dieser Eingang weist den Typ <code>OTCID</code> auf und kann sich dadurch auf ein Mover Template beziehen. Für die dynamische Referenz auf die XTS-Motorbahn kann der Eingang <code>DynamicsReferenceSystem</code> offengelassen oder auf den Wert null gesetzt werden, so dass die Kompatibilität mit früheren Versionen dieses Funktionsbausteins erhalten bleibt. Für die dynamische Referenz auf die von einem Mover Template Object definierte Bahn setzen Sie <code>DynamicsReferenceSystem</code> auf seine Objekt-ID. Dann wird die Dynamik des Bewegungsprofils auf die gegebene Bahn beschränkt.
AdditionalTurns	UDINT	<ul style="list-style-type: none"> Dieser Eingang wird verwendet, um zusätzliche, vollständige Umdrehungen zu kommandieren.

Name	Typ	Beschreibung
(ab TF5400 V3.1.10.30)		<ul style="list-style-type: none"> • <code>AdditionalTurns</code> darf nur verwendet werden (einen Wert > 0 annehmen), wenn: <ul style="list-style-type: none"> ◦ <code>Direction = mcDirectionPositive</code> oder ◦ <code>Direction = mcDirectionNegative</code> ist. • Falls mittels Modulo positioniert wird, d. h. falls <code>Direction</code> einen der drei folgenden Werte annimmt {<code>mcDirectionPositive</code>, <code>mcDirectionNegative</code>, <code>ShortestWay</code>}, muss die <code>TargetPosition</code> ab V3.1.10.30 im Interval <code>[0, ModuloFaktor]</code> liegen. Hier wird vom früheren Verhalten abgewichen. Vor Einführung des Parameters <code>AdditionalTurns</code> wurden zusätzliche Umdrehungen kommandiert, indem größere Zielpositionen als der <code>ModuloFaktor</code> kommandiert wurden. <p>Beispiel: <code>ModuloFaktor = 360</code>, <code>StartPosition = 5</code>; es sollen 2 ganze Umdrehungen kommandiert und nach Position 10 gefahren werden:</p> <ul style="list-style-type: none"> ◦ Bis V3.1.10.14: <code>TargetPosition = 730</code> ◦ Ab V3.1.10.30: <code>TargetPosition = 10</code>, <code>AdditionalTurns = 2</code> • Weitere Details in den Hinweisen zur Modulo-Positionierung [► 116].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.2.3 ST_MoveRelativeCAOptions

```

TYPE ST_MoveRelativeCAOptions :
STRUCT
    GapControlMode          : MC_GAP_CONTROL_MODE := mcGapControlModeGroupDefault;
    PositionReferenceSystem : OTCID := 0;
    DynamicsReferenceSystem : OTCID := 0;
END_STRUCT
END_TYPE

```

Name	Typ	Beschreibung
GapControlMode	MC_GAP_CONTROL_MODE	Siehe die Beschreibung des Datentyps MC GAP CONTROL MODE [► 52] für weitere Informationen.
PositionReferenceSystem (Ab TF5400 V3.1.6.03)	OTCID	<ul style="list-style-type: none"> • Dieser Eingang weist den Typ <code>OTCID</code> auf und kann sich dadurch auf ein Mover Template beziehen. • Für die Positionsreferenz auf die XTS-Motorbahn kann der Eingang <code>PositionReferenceSystem</code> offen gelassen oder auf den Wert null gesetzt werden, so dass die Kompatibilität mit früheren Versionen dieses Funktionsbausteins erhalten bleibt. • Für die Positionsreferenz auf die von einem Mover Template Object definierte Bahn setzen Sie <code>PositionReferenceSystem</code> auf seine Objekt-ID. Dann wird der Positionseingang entsprechend der verschobenen Bahn interpretiert.

Name	Typ	Beschreibung
DynamicsReferenceSystem (Ab TF5400 V3.1.6.03)	OTCID	<ul style="list-style-type: none"> Dieser Eingang weist den Typ <code>OTCID</code> auf und kann sich dadurch auf ein Mover Template beziehen. Für die dynamische Referenz auf die XTS-Motorbahn kann der Eingang <code>DynamicsReferenceSystem</code> offen gelassen oder auf den Wert null gesetzt werden, so dass die Kompatibilität mit früheren Versionen dieses Funktionsbausteins erhalten bleibt. Für die dynamische Referenz auf die von einem Mover Template Object definierte Bahn setzen Sie <code>DynamicsReferenceSystem</code> auf seine Objekt-ID. Dann wird die Dynamik des Bewegungsprofils auf die gegebene Bahn beschränkt.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.2.4 ST_HaltCAOptions

```

TYPE ST_HaltCAOptions :
STRUCT
    GapControlMode          : MC_GAP_CONTROL_MODE := mcGapControlModeGroupDefault;
    DynamicsReferenceSystem : OTCID := 0;
END_STRUCT
END_TYPE
    
```

Name	Typ	Beschreibung
GapControlMode	MC_GAP_CONTROL_MODE	Siehe die Beschreibung des Datentyps <code>MC GAP CONTROL MODE</code> 52 für weitere Informationen.
DynamicsReferenceSystem (Ab TF5400 V3.1.6.03)	OTCID	<ul style="list-style-type: none"> Dieser Eingang weist den Typ <code>OTCID</code> auf und kann sich dadurch auf ein Mover Template beziehen. Für die dynamische Referenz auf die XTS-Motorbahn kann der Eingang <code>DynamicsReferenceSystem</code> offen gelassen oder auf den Wert null gesetzt werden, so dass die Kompatibilität mit früheren Versionen dieses Funktionsbausteins erhalten bleibt. Für die dynamische Referenz auf die von einem Mover Template Object definierte Bahn setzen Sie <code>DynamicsReferenceSystem</code> auf seine Objekt-ID. Dann wird die Dynamik des Bewegungsprofils auf die gegebene Bahn beschränkt.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.1.2.5 MC_GAP_CONTROL_MODE

Der Datentyp MC_GAP_CONTROL_MODE kann verwendet werden, um den Gap Control Mode auf Ebene des Funktionsbausteins festzulegen.

```
TYPE MC_GAP_CONTROL_MODE :
(
  mcGapControlModeGroupDefault := 16#0,
  mcGapControlModeStandard := 16#1,
  mcGapControlModeFast := 16#2
  mcGapControlModeNone := 16#3
) UDINT;
END_TYPE
```

Dieser Datentyp kann nur am FB-Eingang "GapControlMode" verwendet werden, der bei allen Motion-Funktionsbausteinen in [Tc3_McCollisionAvoidance](#) [[▶ 34](#)] vorhanden ist.

Name	Typ	Beschreibung
mcGapControlModeGroupDefault	UDINT	Dieser Wert gibt an, dass der GapControlMode, der in den Gruppenparametern festgelegt wurde, für dieses Bewegungskommando verwendet werden sollte.
mcGapControlModeStandard	UDINT	Siehe die Beschreibung für MC_DEFAULT_GAP_CONTROL_MODE [▶ 28].
mcGapControlModeFast	UDINT	Siehe die Beschreibung für MC_DEFAULT_GAP_CONTROL_MODE [▶ 28].
mcGapControlModeNone	UDINT	Dieser Wert gibt an, dass die Gap Control bei dem Kommando nicht aktiv ist. Nach dem Kommando greift wieder die Standby Gap Control. Mit dem Mode, welcher in der Gruppe eingestellt ist und der Gap-Größe vom letzten gültigen Kommando.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.2 Tc3_McCompensations

Worauf Sollwerte sich beziehen

Sollwerte beziehen sich immer auf die XTS-Motorbahn, denn es ist der Motor, der physikalisch bewegt werden muss. Folglich führt eine Motorbewegung zu einer Zielposition auf der XTS-Motorbahn und damit verbunden auf der Bahn des Werkzeugmittelpunkts.

Auch wenn die Sollwerte für die Bahndynamik auf die XTS-Motorbahn angewendet werden, können sie für die dynamische Steuerung der Bahn des Werkzeugmittelpunkts berechnet werden. Dementsprechend hängt die y -Verschiebung von der gewünschten Anwendung ab und kann für verschiedene Anwendungen unterschiedlich sein. Beispielsweise könnte die Steuerung der Schwerpunktdynamik oder die Verbesserung der Leistung eines auf einem Mover montierten Werkzeugs beabsichtigt werden. Insbesondere kann eine andere Werkzeuggröße ein anderes Mover Template erfordern. Wenn die Bahn für den Werkzeugmittelpunkt ausgewählt wird, werden Sollwerte berechnet, um diesen dynamisch zu steuern.

Koordinatensystem der XTS-Motorbahn

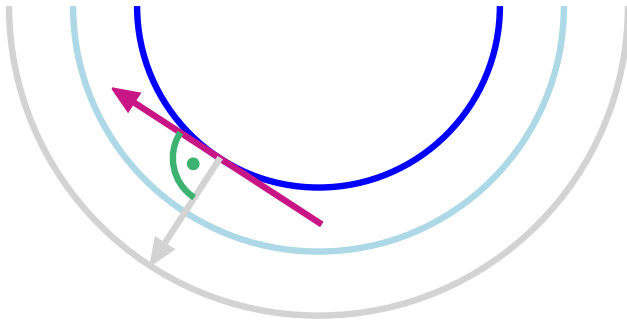
Der Ursprung des Koordinatensystems der Motorbahn des Movers liegt auf der Motorbahn des Movers am im XTS Standard Object definierten Offset x -Wert.

Koordinatensystem des Tool Center Points

Der Ursprung des Koordinatensystems des Tool Center Points liegt am Tool Center Point.

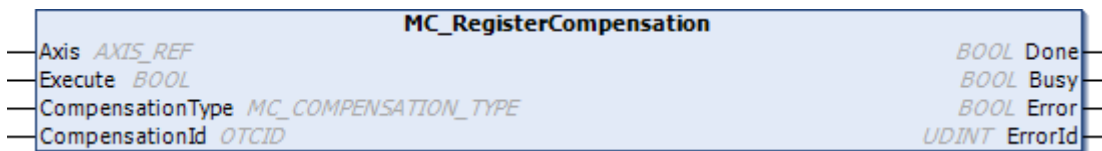
XTS-Motorbahn zum Tool Center Point: Verständnis der Koordinatentransformation

Die Koordinatentransformation von der Motorbahn des Movers zum Tool Center Point bezieht sich immer senkrecht auf die Motorbahn des Movers. Idealerweise und theoretisch weist das Skalarprodukt des Vektors, welcher die Translation der Motorbahn des Movers zum Tool Center Point beschreibt, und des zugehörigen Vektors der Tangente der Motorbahn des Movers, den Wert null auf.



9.2.1 Funktionsbausteine

9.2.1.1 MC_RegisterCompensation



Bezug auf ein Mover Template: Eine Achse bezieht sich über den Funktionsbaustein MC_RegisterCompensation auf ein Mover Template.

- Dieser Funktionsbaustein wählt den Kompensationstyp aus.
- Dieser Funktionsbaustein beeinflusst das Achsverhalten.

VAR_INPUT

```
VAR_INPUT
    Execute          : BOOL;
    CompensationType : MC_COMPENSATION_TYPE;
    CompensationId   : OTCID;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Dieser Funktionsbaustein aktiviert den gewählten Kompensationstyp, wenn eine steigende Flanke an seinem Execute-Eingang ausgelöst wird. Bei Aktivierung werden die Geometrieinformationen für die Geo-Kompensation von den Motion-Funktionsbausteinen, die sich auf die Axis beziehen, berücksichtigt.
Compensation Type	MC_COMPENSATION_TYPE	Wählen Sie mcTypeGeoCompensation für die Geo-Kompensation aus (siehe MC_COMPENSATION_TYPE [► 108]).
CompensationId	OTCID	Dieser Eingang CompensationId weist den Typ OTCID auf und kann sich dadurch auf ein Mover Template beziehen. Der Bezug auf die für die Geo-Kompensation erforderlichen Geometrieinformationen erfolgt über die Objekt-ID CompensationId, die sich auf ein Mover Template bezieht.

VAR_IN_OUT

```
VAR_IN_OUT
    Axis : AXIS_REF;
END_VAR
```

Name	Typ	Beschreibung
Axis	AXIS_REF	Der Eingang Axis weist den Typ <code>AXIS_REF</code> auf und bezieht sich auf eine Achse, z. B. auf einen Mover.

VAR_OUTPUT

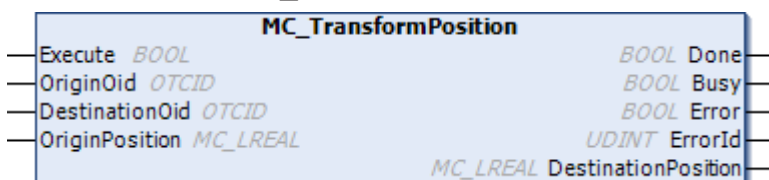
```
VAR_OUTPUT
  Done      : BOOL;
  Busy      : BOOL;
  Error     : BOOL;
  ErrorId   : UDINT;
END_VAR
```

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando mit <code>Execute</code> gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn <code>Busy</code> wieder <code>FALSE</code> wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge <code>Done</code> oder <code>Error</code> gesetzt.
Error	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Kommandos.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4022.25 TF5400 Advanced Motion Pack V3.1.6.03	PC oder CX (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.2.1.2 MC_TransformPosition



- Dieser Funktionsbaustein berechnet eine Koordinatentransformation.
- Eine im Ursprungskordinatensystem gegebene Position wird im Zielkordinatensystem zurückgegeben.
- Ein Ursprungsmover Template Object bezieht sich auf das Ursprungskordinatensystem.
- Ein Zielmover Template Object bezieht sich auf das Zielkordinatensystem.
- Die Objekt-ID 0, `Oid = 0`, bezieht sich auf das absolute Koordinatensystem.
- Das Ursprungsmover Template Object kann sich auf das absolute Koordinatensystem beziehen, und das Zielmover Template Object kann sich auf das Koordinatensystem des Tool Center Points beziehen: Auf diese Weise soll die Berechnung einer Koordinatentransformation vom absoluten Koordinatensystem zum Koordinatensystem des Tool Center Points durchgeführt werden.
- Das Ursprungsmover Template Object kann sich auf das Koordinatensystem des Tool Center Points beziehen, und das Zielmover Template Object kann sich auf das absolute Koordinatensystem beziehen: Es soll also eine Berechnung der Koordinatentransformation vom Koordinatensystem des Tool Center Points in das absolute Koordinatensystem durchgeführt werden.
- Rein informativ: Ohne Auswirkung auf die Sollwerte.

 **VAR_INPUT**

```
VAR_INPUT
    Execute           : BOOL;
    OriginOid         : OTCID;
    DestinationOid    : OTCID;
    OriginPosition    : MC_LREAL;
END_VAR
```

Name	Typ	Beschreibung
Execute	BOOL	Dieser Funktionsbaustein gibt die Zielposition aus, wenn eine steigende Flanke am Eingang <code>Execute</code> ausgelöst wird.
OriginOid	OTCID	Dieser Eingang bezieht sich auf das Ursprungsmover Template Object als Koordinatensys-temreferenz.
DestinationOid	OTCID	Dieser Eingang bezieht sich auf das Ziel-Mover Template Object als Koordinatensystemrefe-renz.
OriginPosition	MC_LREAL	Positionswert im Rahmen des Koordinatensystems, auf das sich das Ursprungsmo-ver Template Object bezieht.

 **VAR_OUTPUT**

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

Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando ausgeführt worden ist und diese Ausführung erfolgreich war.
Busy	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando mit <code>Execute</code> gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn <code>Busy</code> wieder <code>FALSE</code> wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge <code>Done</code> oder <code>Error</code> gesetzt.
Error	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Kommandos.
DestinationPositi on	MC_LREAL	Positionswert im Rahmen des Koordinatensystems, auf das sich das Zielmover Template Object bezieht.

Beispiel

```
VAR
    fbAbsoluteToTcp      : MC_TransformPosition;
    fbTcpToAbsolute     : MC_TransformPosition;
    inputPositionAbsolute : LREAL;
    inputPositionTcp     : LREAL;
    outputPositionTcp    : LREAL;
    outputPositionAbsolute : LREAL;
    oidMoverTemplate     : OTCID;
END_VAR

fbAbsoluteToTcp (
    Execute           := TRUE,
    OriginOid         := 0, //absolute
    DestinationOid    := oidMoverTemplate,
    OriginPosition    := inputPositionAbsolute,
    DestinationPosition => outputPositionTcp
);

fbTcpToAbsolute (
    Execute           := TRUE,
    OriginOid         := oidMoverTemplate,
    DestinationOid    := 0, //absolute
```

```

OriginPosition      := inputPositionTcp,
DestinationPosition => outputPositionAbsolute
);

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4022.25 TF5400 Advanced Motion Pack V3.1.6.07	PC oder CX (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.3 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 [▶ 58]	Fügt eine Achsgruppe hinzu.	✓	✓	✓
MC_GroupDisable [▶ 60]	Deaktiviert eine Achsgruppe.	✓	✓	✓
MC_GroupEnable [▶ 62]	Aktiviert eine Achsgruppe.	✓	✓	✓
MC_GroupReadError [▶ 63]	Liest die Fehler-ID einer Gruppe aus.	✓	✓	✓
MC_GroupReadStatus [▶ 64]	Liest den Gruppenstatus aus.	✓	✓	✓
MC_GroupReset [▶ 66]	Setzt eine Gruppe zurück.	✓	✓	✓
MC_GroupSetOverride [▶ 67]	Setzt den Override einer Gruppe und gibt den Override-Istwert zurück.	✗	✓	✓
MC_RemoveAxisFromGroup [▶ 69]	Entfernt eine Achse aus einer Gruppe.	✓	✓	✓
MC_SetCoordinateTransform [▶ 70]	Aktiviert ein Bezugssystem.	✗	✓	✓
MC_TrackConveyorBelt [▶ 72]	Hilft bei der Synchronisation der Geschwindigkeit mit einem Objekt, das sich auf einer Geraden durch den Raum bewegt.	✗	✓	✓
MC_UngroupAllAxes [▶ 75]	Deaktiviert eine Gruppe und entfernt alle Achsen.	✓	✓	✓
UDINT_TO_IDENTINGROUP P [▶ 76]	Wandelt einen ganzzahligen Wert in IDENT_IN_GROUP_REF	✓	✗	✓

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
	um, so dass Achsen ohne besondere Interpretation einer Gruppe hinzugefügt werden können.			
Motion				
MC_GroupHalt [▶ 77]	Stoppt eine Gruppe, ohne sie für weitere Bewegungskommandos zu sperren.	✔	✘	✔
MC_GroupStop [▶ 79]	Stoppt eine Gruppe und sperrt sie für weitere Bewegungskommandos.	✔	✔	✔
MC_MoveLinearAbsolutePreparation [▶ 81]	Fügt einer Liste von Bewegungssegmenten eine absolute Linearbewegung hinzu.	✘	✔	✔
MC_MoveCircularAbsolutePreparation [▶ 84]	Fügt einer Liste von Bewegungssegmenten eine absolute Kreisbewegung hinzu.	✘	✔	✔
MC_MovePath [▶ 87]	Führt eine Liste von Bewegungssegmenten aus.	✘	✔	✔
MC_BlockerPreparation [▶ 89]	Hängt einen blockierenden Job an die Liste der Segmente in der Struktur PathData an.	✘	✘	✔
MC_ReleaseBlocker [▶ 90]	Löst einen blockierenden Job auf, der die weitere Ausführung der Bahn blockiert.	✘	✘	✔
MC_GroupReadBlockerStatus [▶ 92]	Liest den aktuellen Blocker-Status.	✘	✘	✔
MC_DwellTimePreparation [▶ 93]	Hängt einen Stillstandsjob mit einer definierten Zeit an die Liste der Segmente in der Struktur PathData an.	✘	✘	✔

Strukturen und Aufzählungen

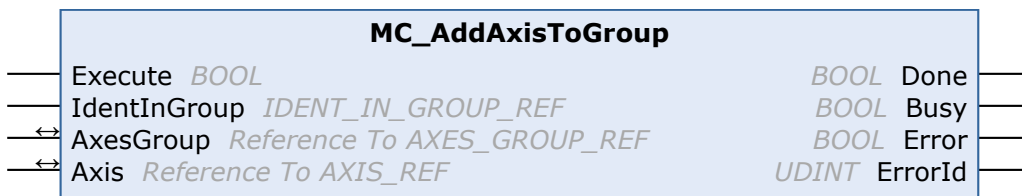
Funktionsbaustein	Beschreibung	TF5410 TwinCAT 3 Motion Collision Avoidance	TF5420 TwinCAT 3 Motion Pick-and-Place	
			MC Group with Pick-and-Place	MC Group Coordinated Motion
IDENT_IN_GROUP_REF [▶ 94]	Definiert, wie eine Achse in einer Gruppe interpretiert wird.	✘	✔	✔

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_CIRC_MODE [► 95]	Der Kreismodus definiert, welche Definition zum Programmieren eines Kreises verwendet wird.	✗	✓	✓
MC_CIRC_PATHCHOICE [► 99]	Der Datentyp definiert die Drehrichtung eines Kreises.	✗	✓	✓
MC_PATH_DATA_REF [► 100]	Stellt die bei MC_MovePath [► 87] auszuführende Bahn dar.	✗	✓	✓
ClearPath [► 101]	Setzt die durch MC_PATH_DATA_REF [► 100] dargestellte Bahn zurück.	✗	✓	✓
MC_TRANSITION_MODE [► 101]	Charakterisiert, wie ein Segmentübergang ausgeführt wird.	✗	✓	✓
MC_COORD_REF [► 104]	Objekt-ID eines Koordinatensystems.	✗	✓	✓

9.3.1 Funktionsbausteine

9.3.1.1 Administrativ

9.3.1.1.1 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
✓	✓	✓

Dieser Funktionsbaustein 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.

 **VAR_INPUT**

```
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 <u>globalen Variablen</u> [▶ 94] (z. B. MCS_X) müssen verwendet werden. Für Collision Avoidance muss die Funktion <u>UDINT TO IDENTINGROUP</u> [▶ 76] 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, globale Variablen (z. B. MCS_X) oder die Konvertierungsfunktion <u>UDINT_TO_IDENTINGROUP</u> zu verwenden.

 **VAR_IN_OUT**

```
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 <u>Zyklische Gruppenschnittstelle</u> [▶ 114]).
Axis	AXIS_REF	Referenz auf eine Achse (siehe <u>AXIS_REF</u>).

 **VAR_OUTPUT**

```
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 befehlspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der <u>ADS-Fehlerdokumentation</u> oder in der <u>NC-Fehlerdokumentation</u> (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

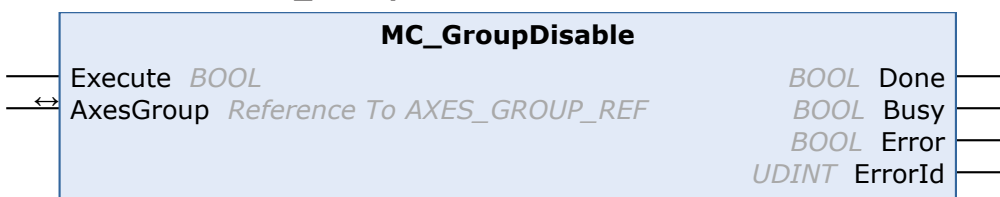
FOR i:=1 TO cAxesCount DO
  fbAddAxis[i](
    AxesGroup:=stGroupRef,
    Axis := stAxis[i],
    IdentInGroup := UDINT_TO_IDENTINGROUP(i),
    Execute := TRUE);
END_FOR

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.2 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

Dieser Funktionsbaustein deaktiviert die Gruppe. Nach erfolgreicher Ausführung ändert die Gruppe ihren Zustand in GroupDisabled (siehe [Zustandsdiagramme](#) [► 24]).

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](#) [► 77] oder [MC_GroupStop](#) [► 79], um diese Situation zu vermeiden.

VAR_INPUT

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

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

VAR_IN_OUT

```
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).

VAR_OUTPUT

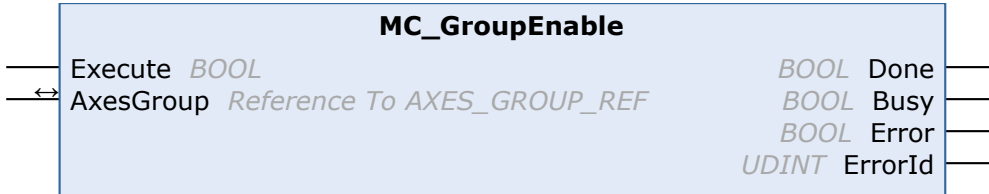
```
VAR_OUTPUT
    Done : BOOL;
    Busy : BOOL;
    Error : BOOL;
    ErrorId : UDINT;
END_VAR
```

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 befehlspezifischen 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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.3 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
✔	✔	✔

Dieser Funktionsbaustein aktiviert die Gruppe. Wenn er erfolgreich ist und alle Achsen bereit sind, befindet sich die Gruppe anschließend im Gruppenzustand GroupStandby (siehe [Zustandsdiagramme \[► 24\]](#)).



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

VAR_INPUT

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

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

VAR_IN_OUT

```
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 [► 114])

VAR_OUTPUT

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

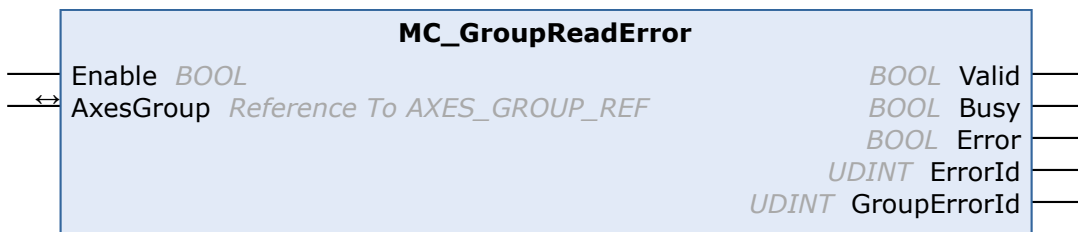
Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando erfolgreich ausgeführt worden ist.

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 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 befehlspezifischen 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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.4 MC_GroupReadError



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

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

VAR_INPUT

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

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

VAR_IN_OUT

```
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).

VAR_OUTPUT

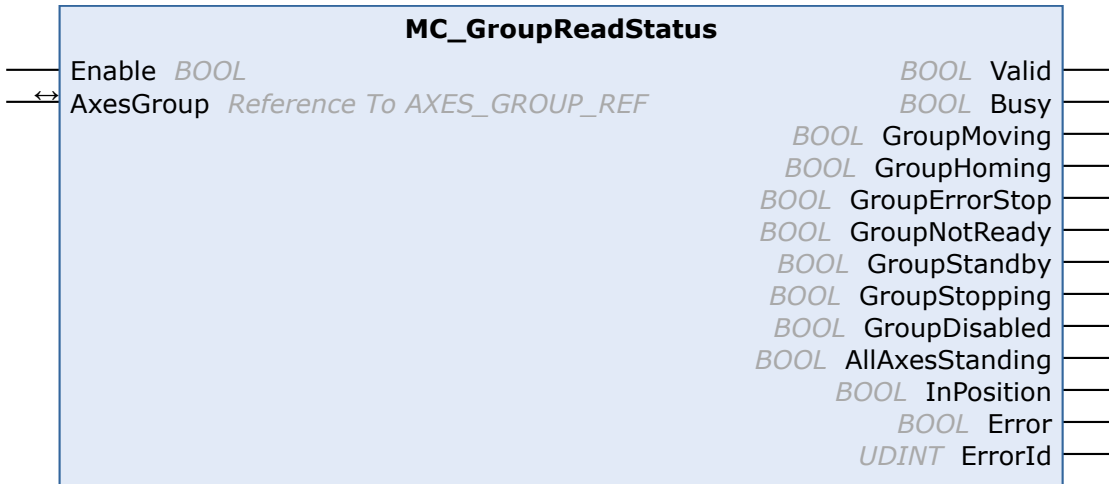
```
VAR_OUTPUT
  Valid      : BOOL;
  Busy       : BOOL;
  Error      : BOOL;
  ErrorId    : UDINT;
  GroupErrorId : UDINT;
END_VAR
```

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 befehlspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der <u>ADS-Fehlerdokumentation</u> oder in der <u>NC-Fehlerdokumentation</u> (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.
GroupErrorId	UDINT	Gibt die Fehler-ID der Gruppe zurück (siehe <u>NC-Fehlerdokumentation</u>).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.5 MC_GroupReadStatus



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

Dieser Funktionsbaustein liest den Zustand einer Achsgruppe aus (siehe Zustandsdiagramme |▶ 24|).

 **VAR_INPUT**

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

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

  **VAR_IN_OUT**

```
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).

 **VAR_OUTPUT**

```
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
```

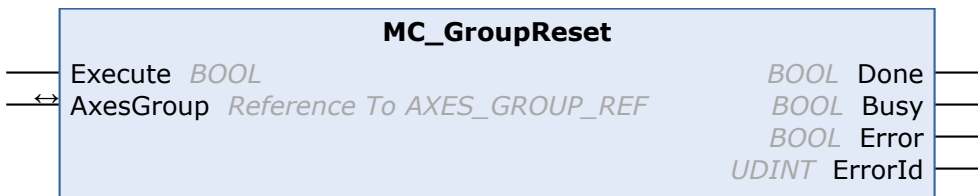
Name	Typ	Beschreibung
Valid	BOOL	Dieser Ausgang gibt an, dass andere Ausgangswerte bei diesem Funktionsbaustein gültig sind.
Busy	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando mit Enable gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt.
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlspezifischen 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 [► 24]).
GroupHoming	BOOL	Die Gruppe befindet sich im Zustand GroupHoming (siehe Zustandsdiagramme [► 24]).
GroupErrorStop	BOOL	Die Gruppe befindet sich im Zustand GroupErrorStop (siehe Zustandsdiagramme [► 24]).
GroupNotReady	BOOL	Die Gruppe befindet sich im Zustand GroupNotReady (siehe Zustandsdiagramme [► 24]).
GroupStandby	BOOL	Die Gruppe befindet sich im Zustand GroupStandby (siehe Zustandsdiagramme [► 24]).

Name	Typ	Beschreibung
GroupStopping	BOOL	Die Gruppe befindet sich im Zustand GroupStopping (siehe Zustandsdiagramme [► 24]).
GroupDisabled	BOOL	Die Gruppe befindet sich im Zustand GroupDisabled (siehe Zustandsdiagramme [► 24]).
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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.6 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
✔	✔	✔

Dieser Funktionsbaustein 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 [► 24]).

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

VAR_INPUT

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

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

 **VAR_IN_OUT**

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

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

 **VAR_OUTPUT**

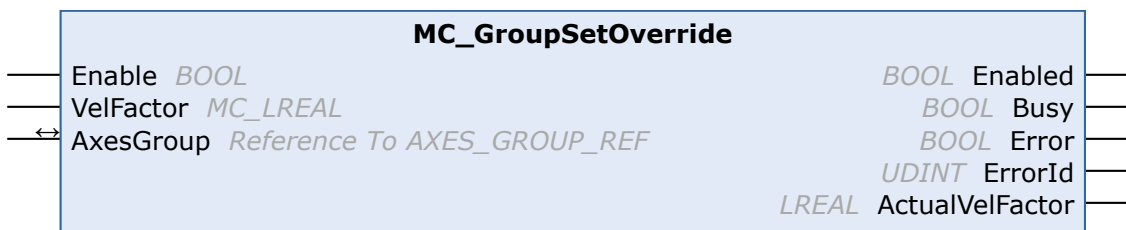
```
VAR_OUTPUT
  Done          : BOOL;
  Busy          : BOOL;
  Error         : BOOL;
  ErrorId       : UDINT;
END_VAR
```




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 befehlspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der <u>ADS-Fehlerdokumentation</u> oder in der <u>NC-Fehlerdokumentation</u> (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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.7 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
		

Dieser 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](#).

VAR_INPUT

```
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 %]).

VAR_IN_OUT

```
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).

VAR_OUTPUT

```
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

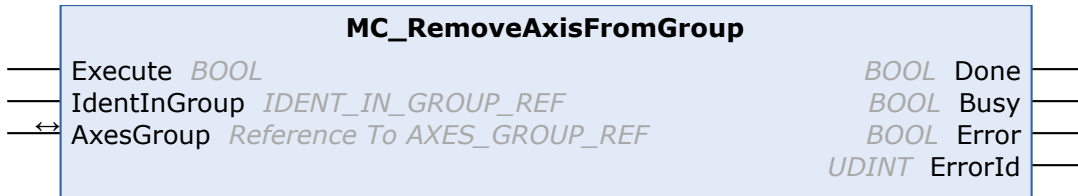
```
VAR
  stGroupRef        : AXES_GROUP_REF;
  fbSetOverride     : MC_GroupSetOverride;
END_VAR

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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.8 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
✔	✔	✔

Dieser Funktionsbaustein entfernt eine Achse aus der Achsgruppe.

i 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).

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

Erfolg des Funktionsbausteins

i 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.

VAR_INPUT

```
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 (z. B. MCS_X) müssen verwendet werden. Zur Collision Avoidance muss die Funktion UDINT_TO_IDENTINGROUP verwendet werden.

i 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 [▶ 94] (z. B. MCS_X) oder die Konvertierungsfunktion UDINT TO IDENTINGROUP [▶ 76] zu verwenden.

VAR_IN_OUT

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

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

VAR_OUTPUT

```
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 befehlspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der <u>ADS-Fehlerdokumentation</u> oder in der <u>NC-Fehlerdokumentation</u> (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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.9 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
		

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](#) [▶ 72]).

Nachfolgende Bewegungen (z. B. [MC_MovePath](#) [▶ 87]) erfolgen relativ zur Koordinatentransformation.

● Anwendungsfall für das Ändern des Referenzsystems

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

 **VAR_INPUT**

```
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 [▶ 104]).

 **VAR_IN_OUT**

```
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 [▶ 114]).

 **VAR_OUTPUT**

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

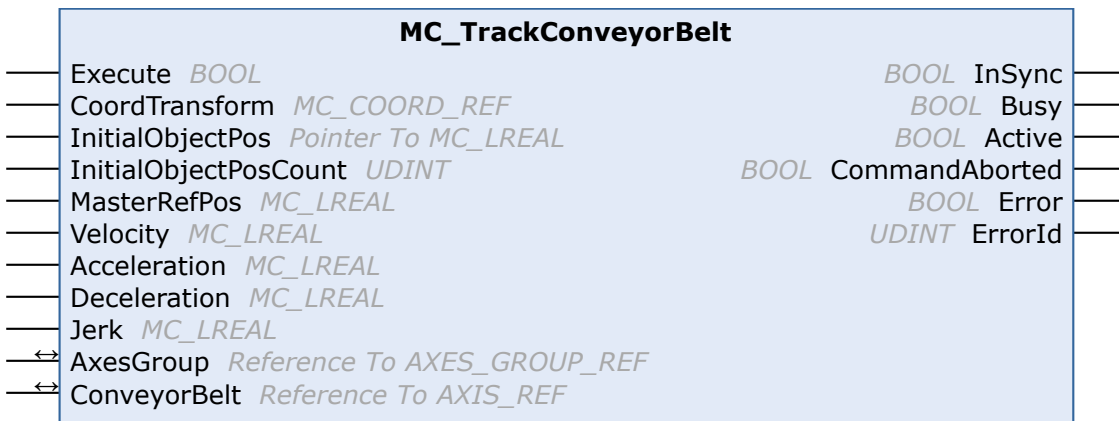
Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando erfolgreich ausgeführt worden ist.
Busy	BOOL	Dieser Ausgang wird <code>TRUE</code> , wenn das Kommando mit <code>Execute</code> gestartet wird, und bleibt es dann so lange, wie der Funktionsbaustein das Kommando ausführt. Wenn <code>Busy</code> wieder <code>FALSE</code> wird, ist der Funktionsbaustein bereit für ein neues Kommando. Gleichzeitig wird einer der Ausgänge <code>Done</code> , <code>CommandAborted</code> oder <code>Error</code> gesetzt.




Name	Typ	Beschreibung
Active	BOOL	<p>Active gibt an, dass das Kommando ausgeführt wird.</p> <p>Active gibt an, dass das Referenzsystem erfolgreich gesetzt wurde (nur MC Coordinated Motion Group).</p> <p>Active zeigt eine Verzögerung des Conveyor Trackings an (nur MC Coordinated Motion Group).</p> <p>Active wird FALSE wenn einer der Ausgänge Done, CommandAborted oder Error auf TRUE gesetzt wird.</p> <p>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.</p>
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 befehlspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der <u>ADS-Fehlerdokumentation</u> oder in der <u>NC-Fehlerdokumentation</u> (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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.10 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
		

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 parametrisiert (`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

i 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_TrackConveyorBelt 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.
- 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.

🚩 VAR_INPUT

```
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 [► 104]).
InitialObjectPos	POINTER TO MC_LREAL	Zeiger auf Array <code>[1..InitialObjectPosCount]</code> .
InitialObjectPosCount	UDINT	Dimension des Vektors <code>InitialObjectPos</code> .
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.

Name	Typ	Beschreibung
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.
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.

VAR_IN_OUT

```
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 [► 114]).
ConveyorBelt	AXIS_REF	Referenz auf eine Achse. Referenz auf die Fördererachse.

VAR_OUTPUT

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

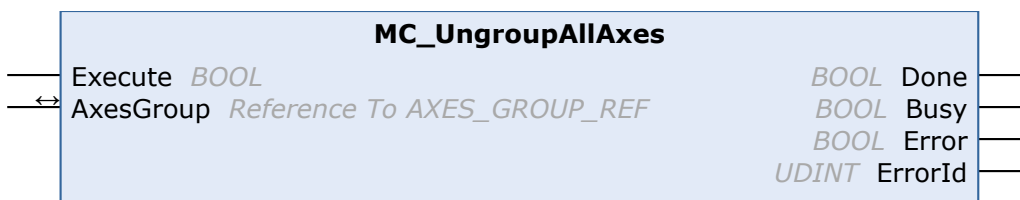
Name	Typ	Beschreibung
InSync	BOOL	<p>Der Ausgang <code>InSync</code> wird erstmals <code>TRUE</code>, sobald der Slave mit der Geschwindigkeit synchronisiert ist. Ist die Slave-Dynamik zu gering, um der Master-Bewegung zu folgen, könnte der Ausgang <code>InSync</code> auf <code>FALSE</code> zurückgesetzt werden, woraufhin die Slaveachse erneut mit der Synchronisation beginnt.</p> <p>Hinweis Geschwindigkeitssynchronisation: <code>Active</code> und <code>InSync</code> - Der Funktionsbaustein <code>MC_TrackConveyorBelt</code> synchronisiert die <code>AxesGroup</code> mit der Geschwindigkeit der <code>ConveyorBelt</code>-Achse. Dabei verwendet der Funktionsbaustein die gegebenen Parameter für <code>Acceleration</code>, <code>Deceleration</code> und <code>Jerk</code>. Wenn diese Synchronisationsbewegung beginnt, wird <code>Active</code> auf <code>TRUE</code> gesetzt. Wenn die Geschwindigkeit des <code>ConveyorBelt</code> erreicht ist, wird <code>InSync</code> auf <code>TRUE</code> gesetzt. Der Synchronisationsstatus wird kontinuierlich überwacht und mit <code>InSync</code> angegeben.</p> <p>Hinweis Fördererbewegung, Standard-Tracking-Verhalten und <code>InSync</code> - Nachdem das Ausgangssignal <code>InSync</code> gesetzt wurde, gibt es zwei Optionen, um die Synchronisation aufrechtzuerhalten. <code>mcTrackingBehaviorDynLimited</code> - Dieses Verhalten ist das standardmäßige</p>

Name	Typ	Beschreibung
		<p>(MC_Default) Tracking-Verhalten. Die Axes-Group erhält die Geschwindigkeitssynchronisation mit dem ConveyorBelt mit Hilfe der gegebenen 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.</p> <p>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-ObjectPos 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.</p>
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 befehlspezifischen 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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.11 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
✔	✔	✔

Dieser Funktionsbaustein entfernt alle Achsen und deaktiviert die Gruppe. Wenn der Funktionsbaustein erfolgreich ist, befindet sich die Gruppe anschließend im Gruppenzustand GroupDisabled (siehe [Zustandsdiagramme \[► 24\]](#)).

VAR_INPUT

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

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

VAR_IN_OUT

```
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).

VAR_OUTPUT

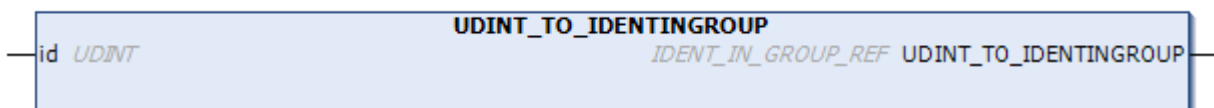
```
VAR_OUTPUT
  Done : BOOL;
  Busy : BOOL;
  Error : BOOL;
  ErrorId : UDINT;
END_VAR
```

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 befehlspezifischen 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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.1.12 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 [▶ 20] muss eine PTP-Achse ohne räumliche Interpretation hinzugefügt werden. Diese Konvertierungsfunktion gibt einen gültigen Eingang für MC_AddAxisToGroup [▶ 58] und MC_RemoveAxisFromGroup [▶ 69] zurück. Für Achsen, die für mehrdimensionale Bewegung vorgesehen sind (TF5420), siehe IDENT_IN_GROUP_REF [▶ 94].

Verwendung ganzzahliger Werte für den Eingang IdentInGroup

I 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 [▶ 94] (z. B. MCS_X) oder die Konvertierungsfunktion UDINT_TO_IDENTINGROUP [▶ 76] zu verwenden.

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.

Rückgabewert

Name	Typ	Beschreibung
UDINT_TO_IDENTINGROUP	IDENT_IN_GROUP_REF [▶ 94]	Wandelt einen ganzzahligen Wert um, so dass eine PTP-Achse einer Bewegungsgruppe hinzugefügt werden kann.




Voraussetzungen

Entwicklungsumgebung	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.2 Motion

9.3.1.2.1 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 [▶ 79]" 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 unverzügliches 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.

VAR_INPUT

```
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/s ²]. Die Verzögerung kann als skalarer Wert (>0) programmiert werden oder es können "Spezielle Eingangswerte [▶ 115]" 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/s ³]. Der Ruck kann als skalarer Wert (>0) programmiert werden oder es können "Spezielle Eingangswerte [▶ 115]" 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.

VAR_IN_OUT

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

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

VAR_OUTPUT

```

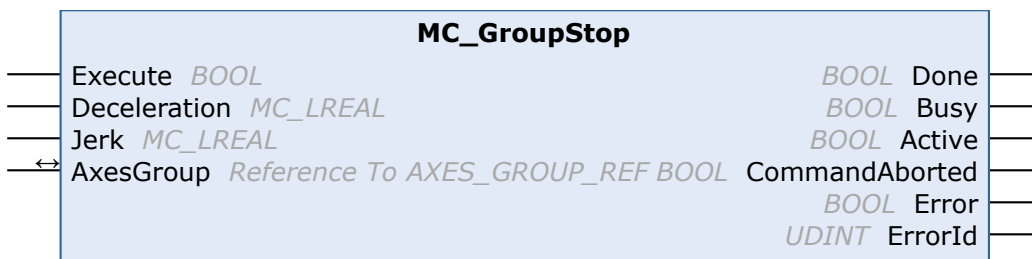
VAR_OUTPUT
  Done           : BOOL;
  Busy           : BOOL;
  Active         : BOOL;
  CommandAborted : BOOL;
  Error          : BOOL;
  ErrorId        : UDINT;
END_VAR
    
```

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 <u>Zustandsdiagramme</u> [► 24]).
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 befehlspezifischen Fehlercode des zuletzt ausgeführten Kommandos. Der Fehlercode kann in der <u>ADS-Fehlerdokumentation</u> oder in der <u>NC-Fehlerdokumentation</u> (Fehlercodes 0x4nnn und 0x8nnn) nachgeschlagen werden.

Voraussetzungen

Entwicklungsumgebung	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.2.2 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
✔	✔	✔

Der Funktionsbaustein 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](#) [► 24]).

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.

VAR_INPUT

```
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/s ²]. Die Verzögerung kann als skalarer Wert (>0) programmiert werden oder es können " Spezielle Eingangswerte [► 115]" 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/s ³]. Der Ruck kann als skalarer Wert (>0) programmiert werden oder es können " Spezielle Eingangswerte [► 115]" 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.

VAR_IN_OUT

```
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).

VAR_OUTPUT

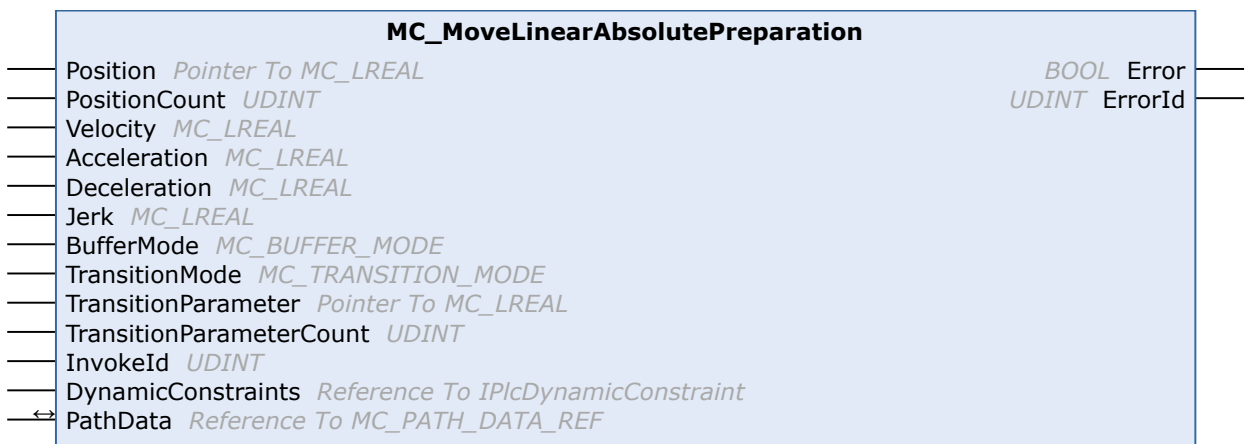
```
VAR_OUTPUT
  Done             : BOOL;
  Busy             : BOOL;
  Active           : BOOL;
  CommandAborted  : BOOL;
  Error            : BOOL;
  ErrorId         : UDINT;
END_VAR
```


Name	Typ	Beschreibung
Done	BOOL	Wird TRUE, wenn die Gruppe gestoppt wurde und zum Stillstand gekommen ist. Während <i>Execute</i> TRUE ist, aber mindestens so lange, bis die Achsen zum Stillstand gekommen sind, bleibt die Gruppe im Zustand <i>GroupStopping</i> . Anschließend befindet sich die Gruppe im Zustand <i>GroupStandby</i> (siehe Zustandsdiagramme [► 24]).
Busy	BOOL	Wird TRUE, wenn das Kommando mit <i>Execute</i> gestartet wird, und bleibt es dann so lange, wie das Kommando ausführt wird. Wenn <i>Busy</i> wieder FALSE wird, ist die Gruppe bereit für ein neues Kommando. Nachdem die Gruppe gestoppt wurde, bleibt <i>Busy</i> TRUE, bis die Gruppe mit <i>Execute</i> =FALSE freigegeben wird.
Active	BOOL	Gibt an, dass der Funktionsbaustein die Gruppe steuert. Nachdem die Gruppe gestoppt wurde, bleibt <i>Active</i> TRUE, bis die Gruppe mit <i>Execute</i> =FALSE freigegeben wird.
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 befehlspezifischen 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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.2.3 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

Der Funktionsbaustein fügt der Tabelle der Segmente in der Struktur PathData eine absolute Linearbewegung hinzu. Nach der Erstellung einer Tabelle kann er über [MC_MovePath \[► 87\]](#) ausgeführt werden. Der Funktionsbaustein MC_MoveLinearAbsolutePreparation kann mehrmals pro Zyklus aufgerufen werden. Maximal 30 Einträge sind pro PathData-Tabelle zulässig.

VAR_INPUT

```
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 [► 115] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus.
Deceleration	MC_LREAL	Maximale Bahnverzögerung für das programmierte Segment. Spezielle Eingangswerte [► 115] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus.
Jerk	MC_LREAL	Bahnruck für das programmierte Segment. Spezielle Eingangswerte [► 115] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. 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 [► 104]).
Transition Mode	MC_TRANSITION_MODE	Definiert den Blending-Modus (siehe MC_TRANSITION_MODE [► 101]).
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 [► 101]).
TransitionParameterCount	UDINT	Anzahl der Blending-Parameter (siehe MC_TRANSITION_MODE [► 101]).
InvokeId	UDINT	Segment-ID für Analysezwecke.

Name	Typ	Beschreibung
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.

 **VAR_IN_OUT**

```
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 [▶ 87] ausgeführt (siehe MC_PATH_DATA_REF [▶ 100]).

● Zurücksetzen einer Tabelle

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

 **VAR_OUTPUT**

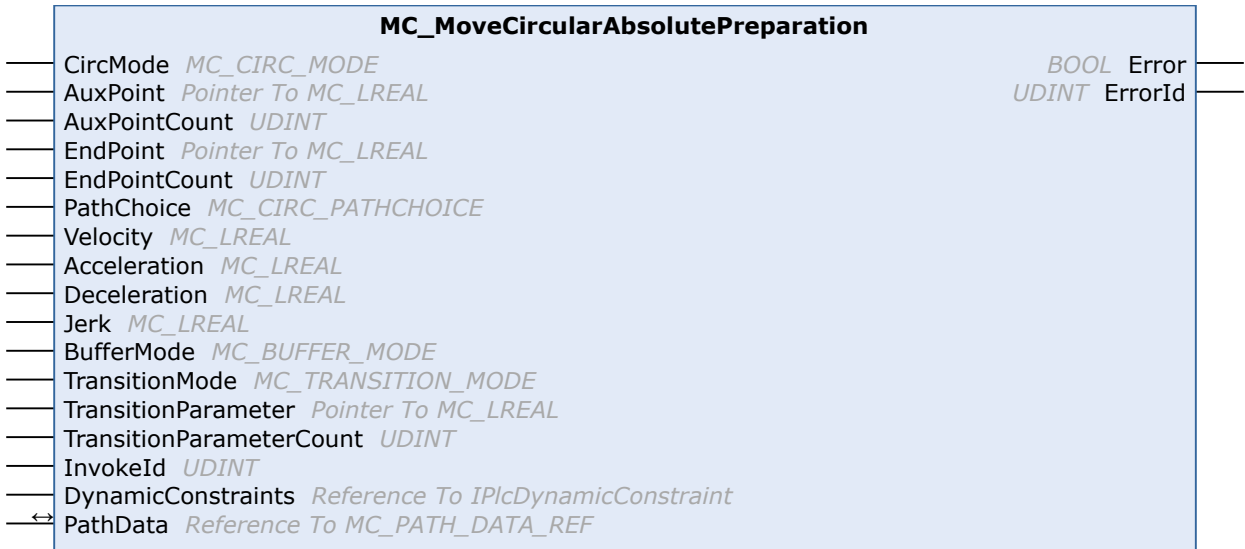
```
VAR_OUTPUT
    Error          : BOOL;
    ErrorId        : UDINT;
END_VAR
```




Name	Typ	Beschreibung
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlspezifischen 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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.2.4 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
		

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

Zurücksetzen einer Tabelle

Eine Tabelle wird bei der Ausführung nicht zurückgesetzt. Zum Zurücksetzen muss die Methode `ClearPath()` von MC_PATH_DATA_REF [\[▶ 100\]](#) aufgerufen werden.

VAR_INPUT

```

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_MOD E	Legt fest, durch welche Kreisdefinition der Kreis programmiert wird. Legt die Bedeutung des Eingangssignals "AuxPoint" fest (siehe MC_CIRC_MODE [▶ 95]).

Name	Typ	Beschreibung
AuxPoint	POINTER TO MC_LREAL	Zeiger auf ein Array [1..AuxPointCount] des AuxPoint-Vektors. Die Interpretation des AuxPoint-Vektors hängt von der Rotationskonvention ab (siehe <u>MC Group Coordinated Motion</u> oder <u>MC Group with Pick-and-Place</u>) und ist immer (x, y, z).
AuxPointCount	UDINT	Dimension des AuxPoint-Vektors. Muss 3 sein. Wenn eine 2D-Rotationskonvention (siehe <u>MC Group Coordinated Motion</u> oder <u>MC Group with Pick-and-Place</u>) verwendet wird, muss der Eingangswert ebenfalls 3 sein. Bei einer 2D-Rotationskonvention und CircMode von <i>mcCircModeBorder</i> oder <i>mcCircModeCenter</i> muss die Komponente, die von der Arbeitsebene unabhängig ist, auf MC_Ignore gesetzt werden (siehe <u>MC LREAL/Spezielle Eingangswerte</u> [► 115]).
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 <u>MC Group Coordinated Motion</u> oder <u>MC Group with Pick-and-Place</u>).
PathChoice	MC_CIRC_PATH CHOICE	Definiert die Drehrichtung in Bezug auf den Normalvektor. Der Eingang wird ignoriert, wenn der Eingang <i>CircMode</i> auf <i>mcCircModeBorder</i> gesetzt ist (siehe <u>MC CIRC PATHCHOICE</u> [► 99]).
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. <u>Spezielle Eingangswerte</u> [► 115] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus.
Deceleration	MC_LREAL	Maximale Bahnverzögerung für das programmierte Segment. <u>Spezielle Eingangswerte</u> [► 115] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. MC_MAXIMUM führt das Kommando mit den Maximalwerten der Achsen aus.
Jerk	MC_LREAL	Bahnruck für das programmierte Segment. <u>Spezielle Eingangswerte</u> [► 115] können verwendet werden. MC_DEFAULT führt das Kommando mit Standardachsenwerten aus. 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 <u>MC BUFFER MODE</u> [► 104]).
Transition Mode	MC_TRANSITION_MODE	Definiert den Blending-Modus (siehe <u>MC TRANSITION MODE</u> [► 101]).
TransitionParameter	POINTER TO MC_LREAL	Zeiger auf Array [1..TransitionParameterCount] der Blending-Parameter. Transitionsparameter definieren das Blending von der letzten programmierten Position (siehe <u>MC TRANSITION MODE</u> [► 101]).
TransitionParameterCount	UDINT	Anzahl der Blending-Parameter.
Invokeld	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.

VAR_IN_OUT

```
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 [▶ 87] ausgeführt (siehe MC_PATH_DATA_REF [▶ 100]).

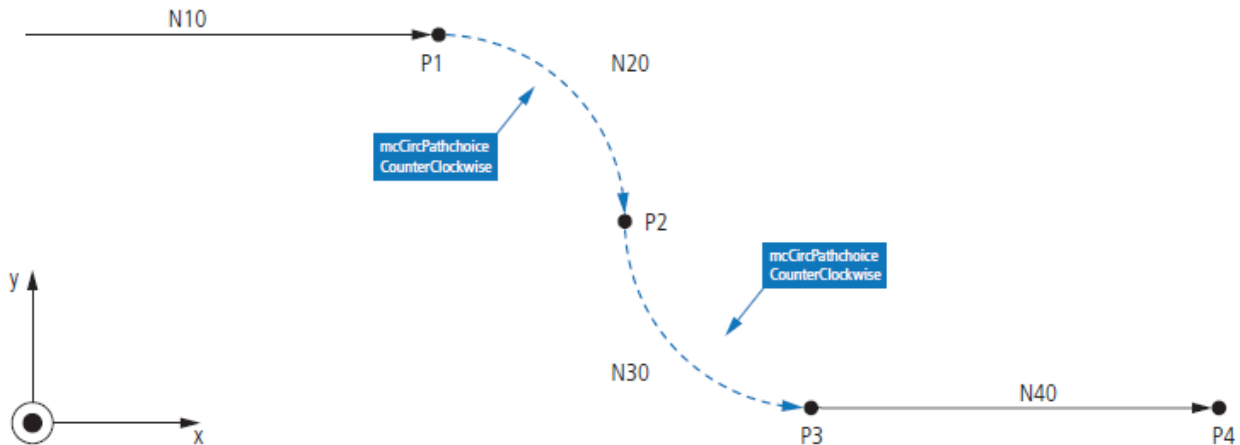
VAR_OUTPUT

```
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 befehlspezifischen 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 [▶ 99] 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:= mcCircPathchoiceCounterClockwise, Velocity:= 1000,
InvokeId:= 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);

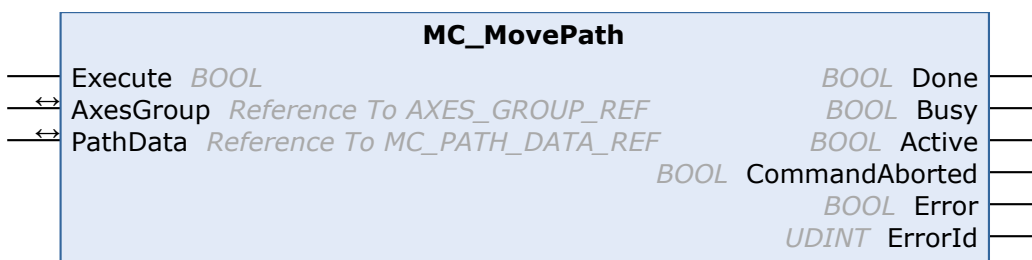
aTargetPos[1] := 600;
aTargetPos[2] := -200;
aTargetPos[3] := 100;
aTransitionParam[1] := 0;
aTransitionParam[2] := 0;
fbMoveLinPrep(PathData:= path, Velocity:= 3000, InvokeId:= 40);




```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.2.47	PC oder CX (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.2.5 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
		

Der Funktionsbaustein MC_MovePath führt eine Bewegung aus, die in der Tabelle PathData durch [MC_MoveLinearAbsolutePreparation \[▶ 81\]](#) und [MC_MoveCircularAbsolutePreparation \[▶ 84\]](#) 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.

VAR_INPUT

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

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

VAR_IN_OUT

```
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 [▶ 114]).
PathData	MC_PATH_DATA_REF	Tabelle, die die Segmente einer Bahn enthält. Die Tabelle wird durch MC_MoveLinearAbsolutePreparation [▶ 81] und MC_MoveCircularAbsolutePreparation [▶ 84] geschrieben und durch MC_MovePath [▶ 87] ausgeführt (siehe MC_PATH_DATA_REF [▶ 100]).

VAR_OUTPUT

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

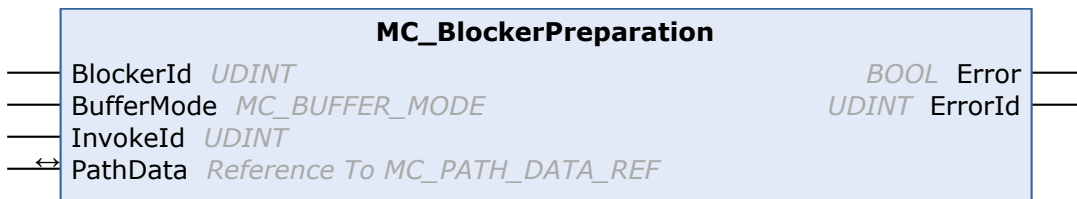
Name	Typ	Beschreibung
Done	BOOL	Dieser Ausgang wird TRUE, wenn das Kommando erfolgreich ausgeführt worden ist. Dies bedeutet, dass das letzte durch die Referenzvariable PathData definierte Kommando erfolgreich ausgeführt wurde.

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 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.
ErrorId	UDINT	Enthält den befehlspezifischen 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	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.2.6 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

Dieser Funktionsbaustein 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](#) [► 90] 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](#) [► 77] oder [MC_GroupStop](#) [► 79] ausgeführt werden, während der blockierende Job aktiv ist, wird die Bahn beendet und der blockierende Job automatisch freigegeben.

VAR_INPUT

```
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.
BufferMode	MC_BUFFER_MODE	Legt fest, wie aufeinanderfolgende Fahraufträge abgearbeitet werden sollen (siehe MC_BUFFER_MODE [► 104]). Hier sind nur mcBuffered und mcAborting erlaubt.
Invokeld	UDINT	Segment-ID für Analysezwecke.

VAR_IN_OUT

```
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).

VAR_OUTPUT

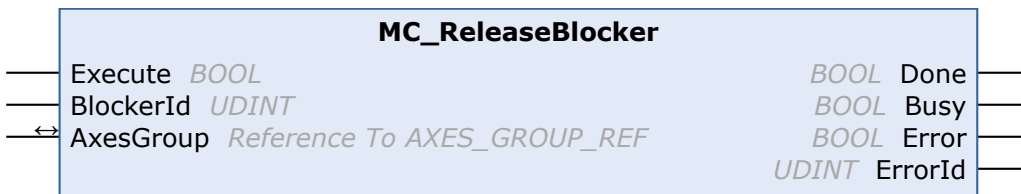
```
VAR_OUTPUT
  Error          : BOOL;
  ErrorId        : UDINT;
END_VAR
```




Name	Typ	Beschreibung
Error	BOOL	Dieser Ausgang wird TRUE, wenn bei der Ausführung des Kommandos ein Fehler aufgetreten ist.
ErrorId	UDINT	Enthält den befehlspezifischen 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	Zielformat	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4024.7 TF5400 Advanced Motion Pack V3.1.10.1	PC oder CX (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.2.7 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
		

Dieser Funktionsbaustein löst einen blockierenden Job auf, der die weitere Ausführung der Bahn blockiert. Ein blockierender Job wird mit [MC BlockerPreparation \[▶ 89\]](#) 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.

 **VAR_INPUT**

```
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.

  **VAR_IN_OUT**

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

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Achsgruppe (siehe Zyklische Gruppenschnittstelle [▶ 114]).

 **VAR_OUTPUT**

```
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 befehlspezifischen 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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.2.8 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
		

Dieser Funktionsbaustein liest den aktuellen Blocker-Status.

 **VAR_INPUT**

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

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

 **VAR_IN_OUT**

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

Name	Typ	Beschreibung
AxesGroup	AXES_GROUP_REF	Referenz auf eine Achsgruppe (siehe Zyklische Gruppenschnittstelle [► 114]).

 **VAR_OUTPUT**

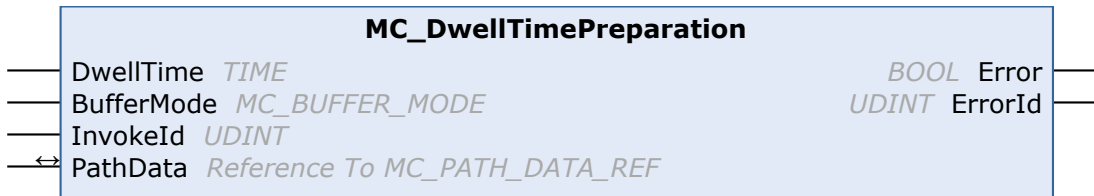
```
VAR_OUTPUT
  Valid       : BOOL;
  Blocked     : BOOL;
  BlockerId   : UDINT;
END_VAR
```

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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.3.1.2.9 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

Dieser Funktionsbaustein 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.

VAR_INPUT

```

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 Segmente einen Übergang mit einer Geschwindigkeit > 0 erlauben würden.
BufferMode	MC_BUFFER_MODE	Legt fest, wie aufeinanderfolgende Fahraufträge abgearbeitet werden sollen (siehe <u>MC_BUFFER_MODE [▶ 104]</u>). Hier sind nur mcBuffered und mcAborting erlaubt.
Invokeld	UDINT	Segment-ID für Analyse Zwecke.

VAR_IN_OUT

```

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 <u>MC_MovePath</u> ausgeführt (siehe <u>MC_PATH_DATA_REF</u>).

VAR_OUTPUT

```
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 befehlspezifischen 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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.3.2 Datentypen

9.3.2.1 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](#) [[▶ 76](#)] aufgerufen werden.

Verwendung ganzzahliger Werte für den Eingang IdentInGroup

I 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](#) [[▶ 94](#)] (z. B. MCS_X) oder die Konvertierungsfunktion [UDINT_TO_IDENTINGROUP](#) [[▶ 76](#)] 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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.2.2 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
		

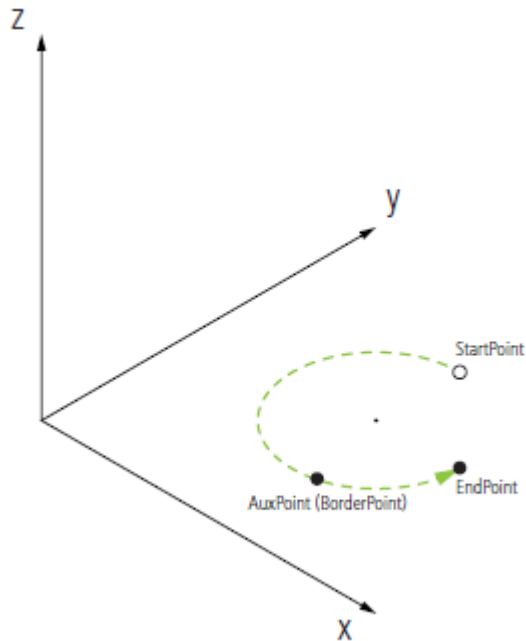
Der Kreismodus bestimmt, welche Kreisdefinition zum Programmieren eines Kreises verwendet wird.

```

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

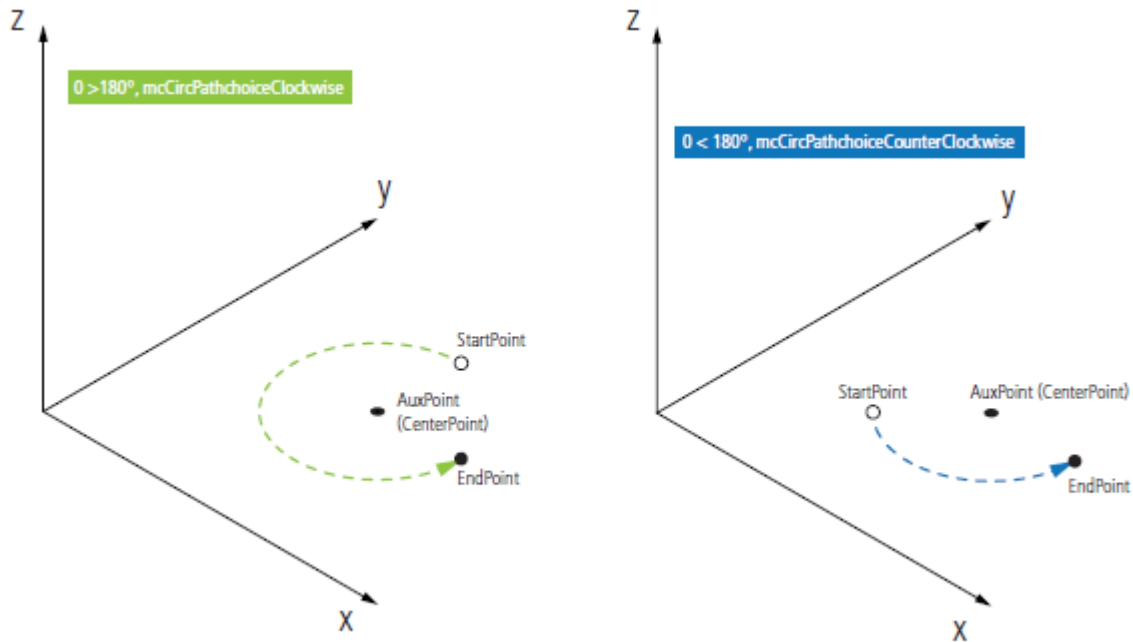
mcCircModeInvalid

- 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.

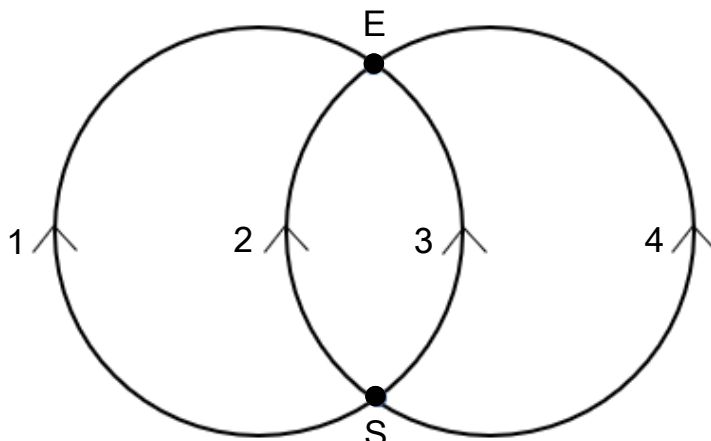
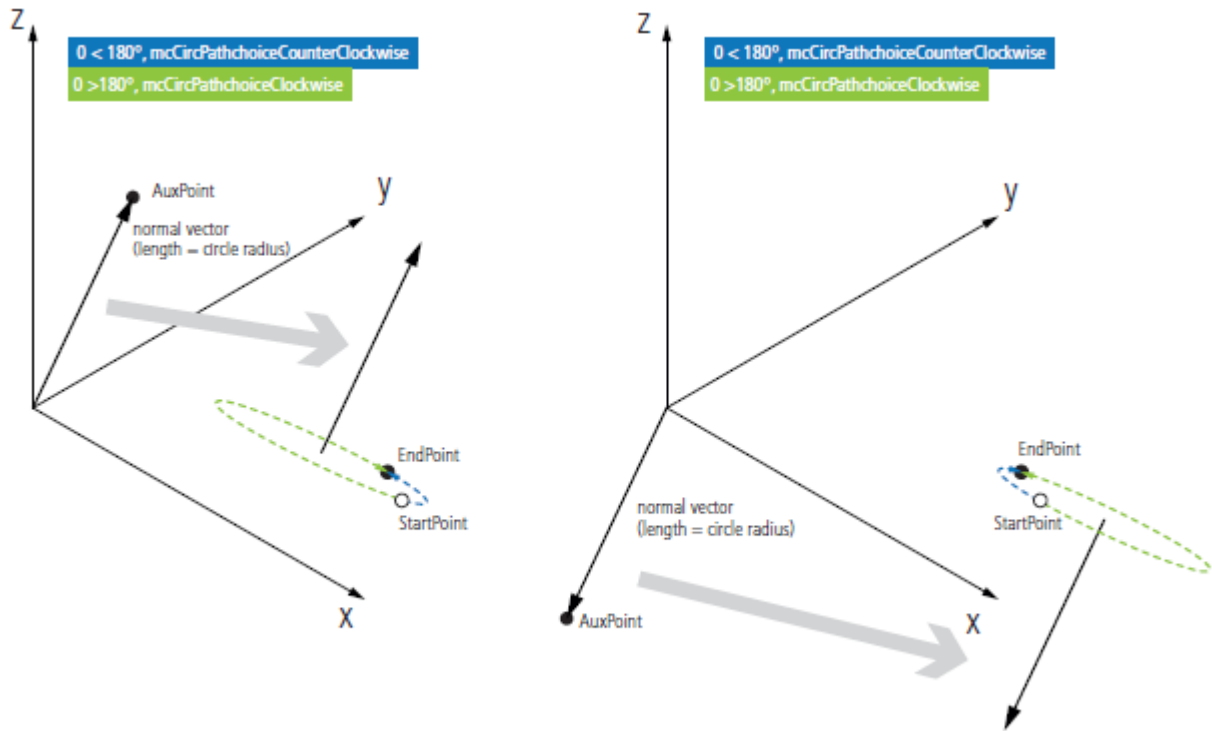
PathChoice

- Es gibt normalerweise zwei mögliche Kreisbögen, die vom Anfangspunkt "Start-Point" 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



E=EndPoint
S=StartPoint

MC_CIRC_PATHCHOICE

	\vec{n}	
Clockwise		1
		4
Counterclockwise		3
		2
Short segment		3
		2
Long segment		4
		1

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 "Start-Point" 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 $\leq 180^\circ$ abdeckt, `mcCircPathchoiceClockwise` und `mcCircPathchoiceLongSegment` beschreiben einen Bogen, der einen Winkel $\geq 180^\circ$ 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 $< 360^\circ$ abdecken.
 - Die Länge des Normalenvektors (d.h. der Radius des Kreises) muss mindestens die Hälfte der Entfernung zwischen Start- und Endpunkt betragen.

Voraussetzungen

Entwicklungsumgebung	Zielpattform	Einzubindende SPS Bibliotheken
TwinCAT V3.1.4018.26 TF5400 Advanced Motion Pack V3.1.2.47	PC oder CX (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.3.2.3 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
		

Der Datentyp `MC_CIRC_PATHCHOICE` definiert die Drehrichtung eines Kreises, falls `mcCircModeCenter` oder `mcCircModeRadius` aus der Aufzählung `MC_CIRC_MODE` [► 95] ausgewählt wird.

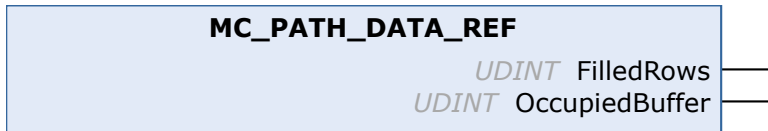
```




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
<code>mcCircPathchoiceClockwise</code>	INT	stellt das Kreissegment mit einem Winkel $>180^\circ$ dar.
<code>mcCircPathchoiceCounterClockwise</code>	INT	stellt das Kreissegment mit einem Winkel $<180^\circ$ dar.
<code>mcCircPathchoiceShortSegment</code>	INT	stellt das Kreissegment mit dem kleineren Winkel dar.
<code>mcCircPathchoiceLongSegment</code>	INT	stellt das Kreissegment mit dem größeren Winkel dar.

9.3.2.4 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
		

MC_PATH_DATA_REF stellt die durch MC_MovePath [► 87] auszuführende Bahn dar, wobei die Anzahl der Einträge auf 30 begrenzt ist. Die auszuführende Bahn wird durch MC_MoveLinearAbsolutePreparation [► 81], MC_MoveCircularAbsolutePreparation [► 84] und MC_BlockerPreparation [► 89] 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 [► 101] aufgerufen werden.

VAR_OUTPUT

```
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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2




9.3.2.4.1 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
		

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

9.3.2.5 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
		

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

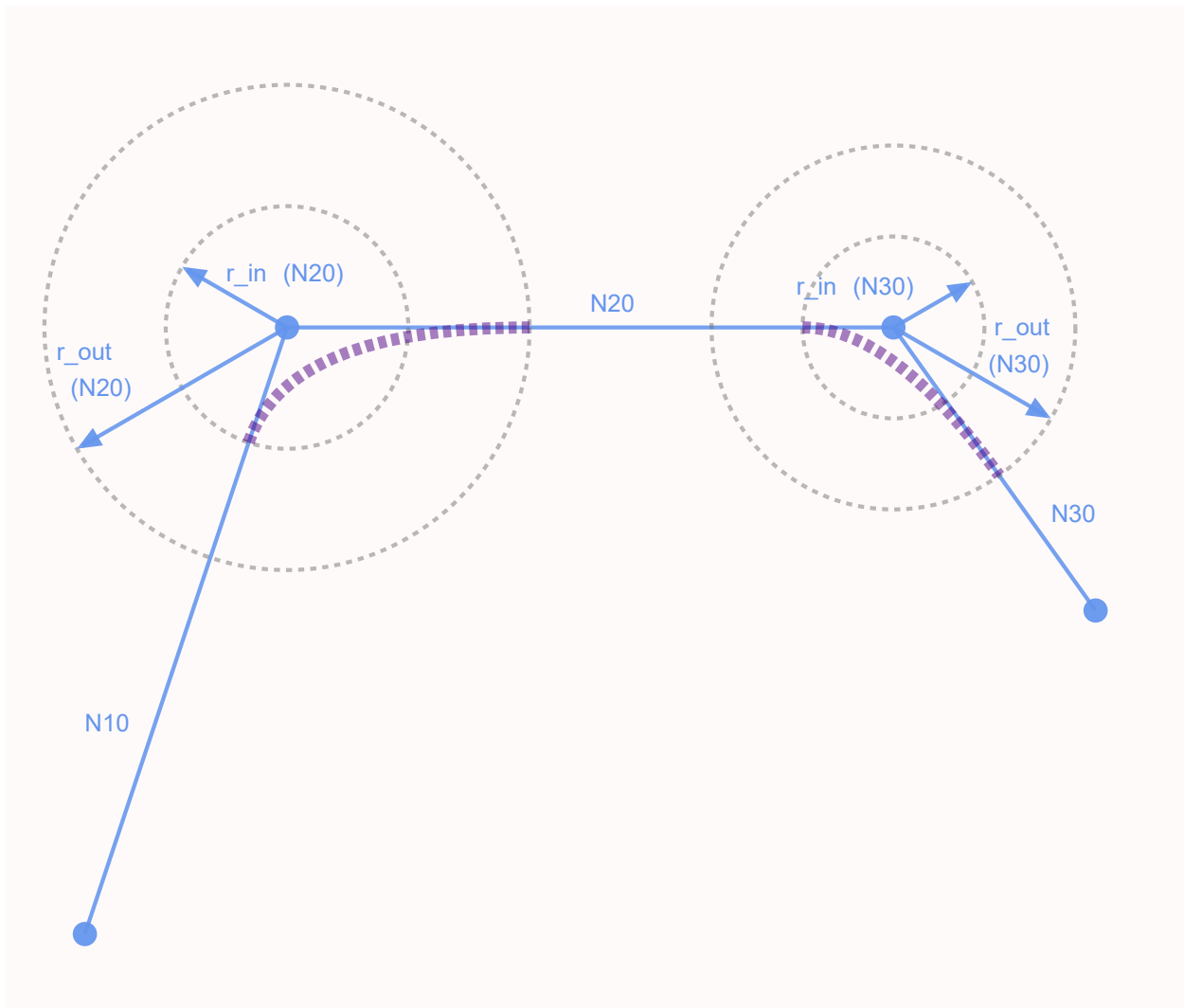
Blending wird zwischen den Segmenten ausgeführt. Die Transitionsparameter fungieren als Toleranzkugel, in der der programmierten Bahn nicht gefolgt wird. Der Parameter beschreibt den Radius am vorherigen und zweiten Segment, bei dem das Blending beginnt und endet.

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.



i 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

i 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 TF5400 Advanced Motion Pack V3.1.1.17	PC oder CX (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2
















9.3.2.6 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
		

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

9.4 Tc3_Mc3Definitions

Strukturen und Aufzählungen

Name	Beschreibung	TF5410 TwinCAT 3 Motion Colli- sion Avoi- dance	TF5420 TwinCAT 3 Motion Pick- and-Place	
			MC Group with Pick- and-Place	MC Group Coordinated Motion
MC_BUFFER_MODE [▶_104]	Legt fest, wie aufeinanderfolgende Fahrkommandos zu verarbeiten sind.			
MC_COMPENSATION_T YPE [▶_108]	Der Wert definiert den Kompensationstyp.			
MC_DIRECTION [▶_108]	Der Wert bestimmt die Richtung der Bewegung.			
MC_SYNC_MODE [▶_109]	Der Wert definiert die Richtung, in die synchronisiert werden soll.			
MC_SYNC_STRATEGY [▶_109]	Definiert das Synchronisationsprofil der Slave-Achse.			

9.4.1 Datentypen




9.4.1.1 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.

```

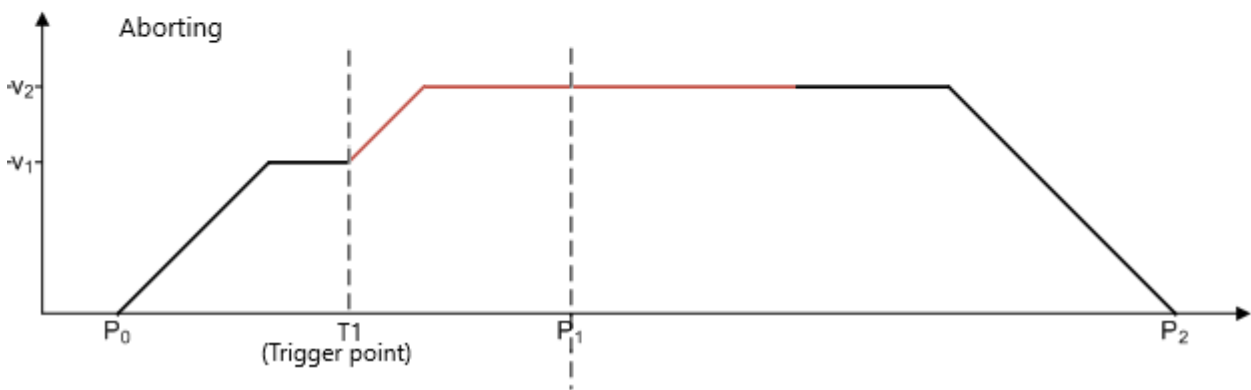
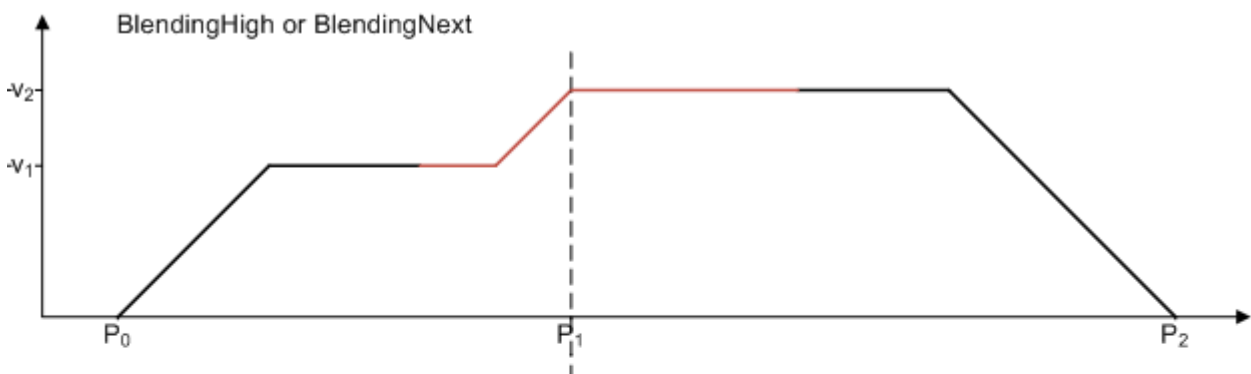
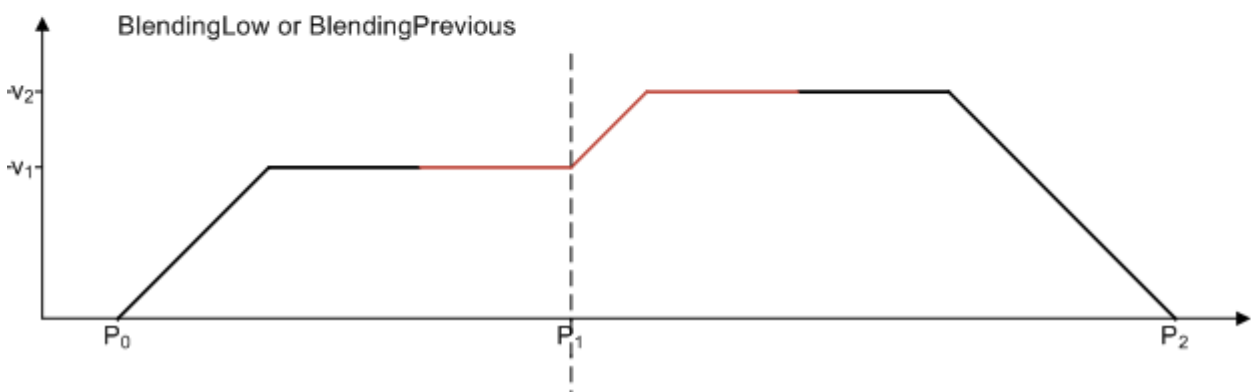
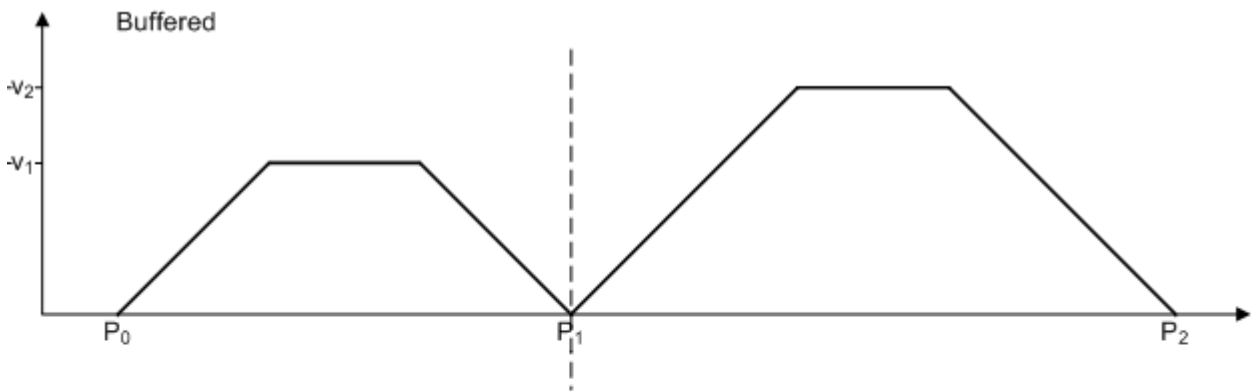
TYPE MC_BUFFER_MODE :
(
  mcAborting           := 16#0,
  mcBuffered           := 16#1,
  mcBlendingLow        := 16#12,
  mcBlendingPrevious   := 16#13,
  mcBlendingNext       := 16#14,
  mcBlendingHigh       := 16#15
) UINT;
END_TYPE

```

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

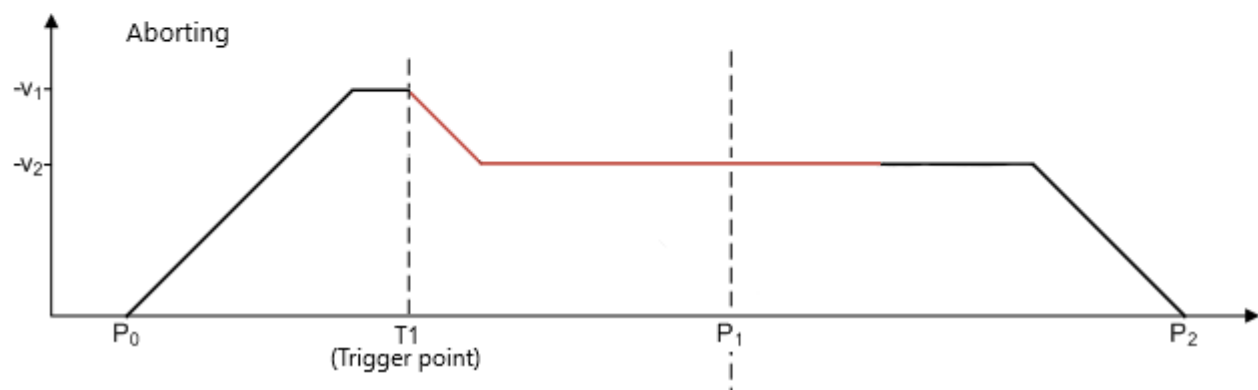
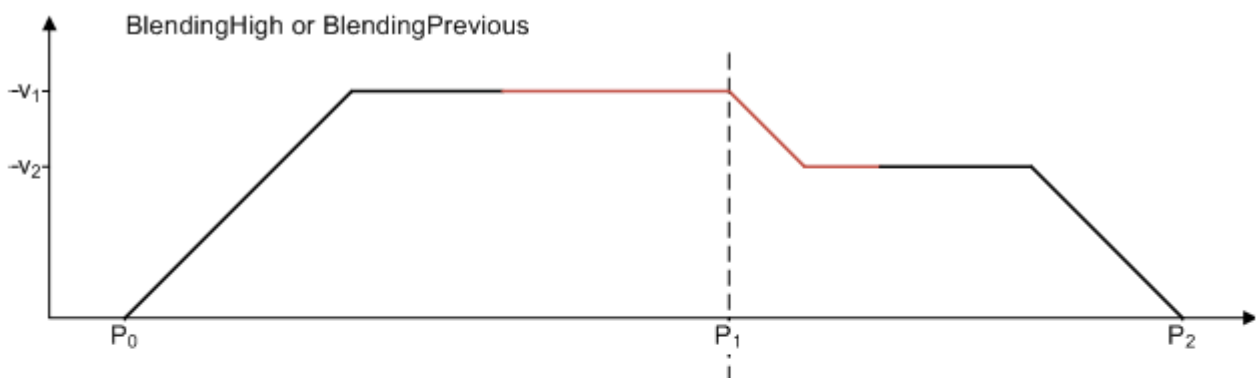
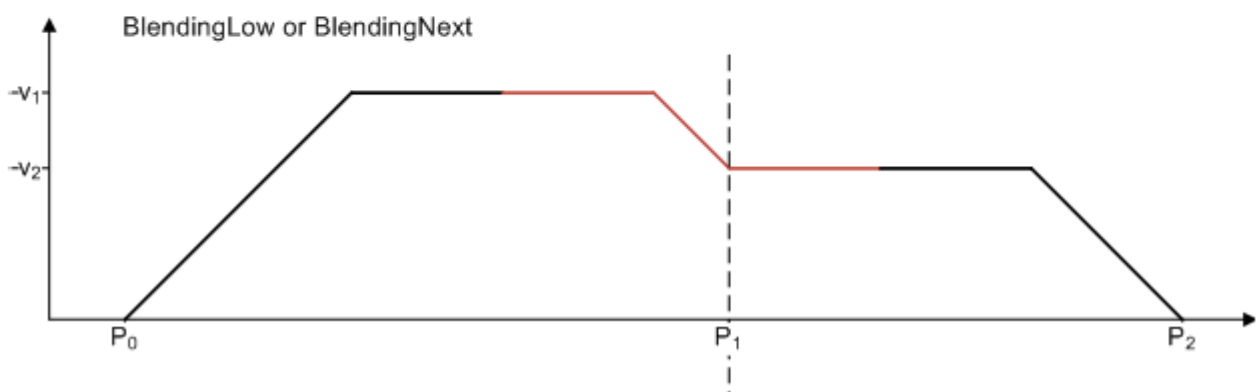
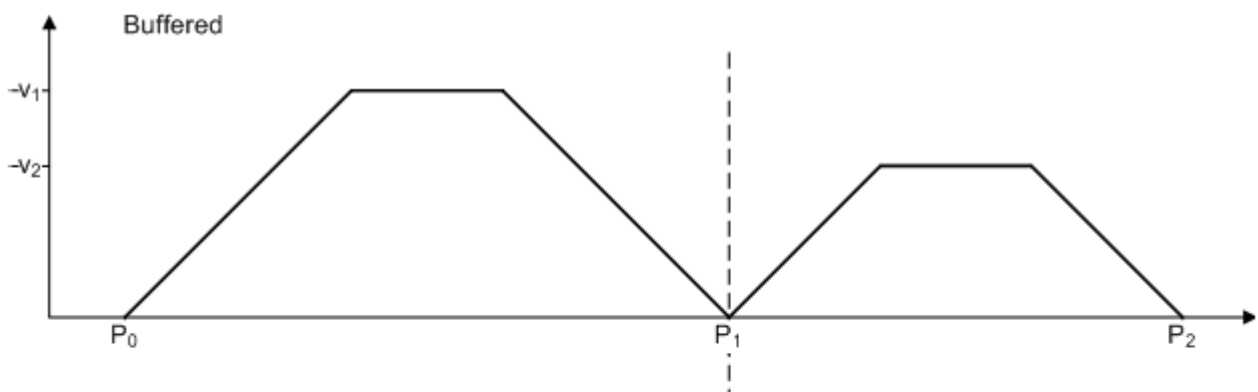
Beispiel:

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	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 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 (x86 oder x64)	Tc3_McCoordinatedMotion, Tc2_MC2

9.4.1.2 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 (x86 oder x64)	Tc3_McCompensations

9.4.1.3 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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.4.1.4 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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

9.4.1.5 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: mcSyncStrategySlow**

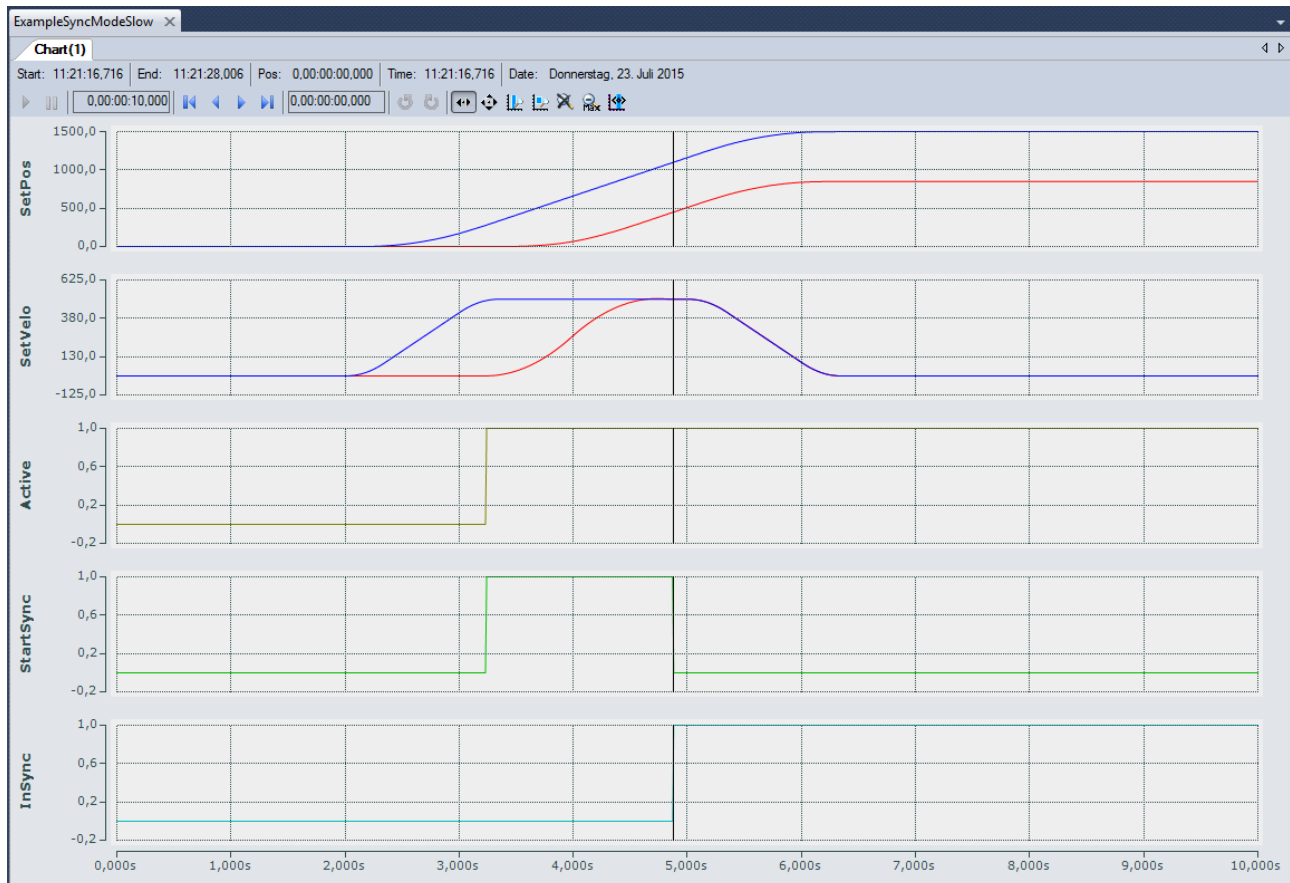
Der Slave beginnt seine Synchronisation in Bewegung, sobald der Master (MasterSyncPosition – MasterStartDistance) in der richtigen Richtung passiert, wenn eine MasterStartDist gesetzt wurde, anderenfalls sobald der Funktionsbaustein Active ist. Die Dynamik des Slaves wird so reduziert, dass der

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).

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

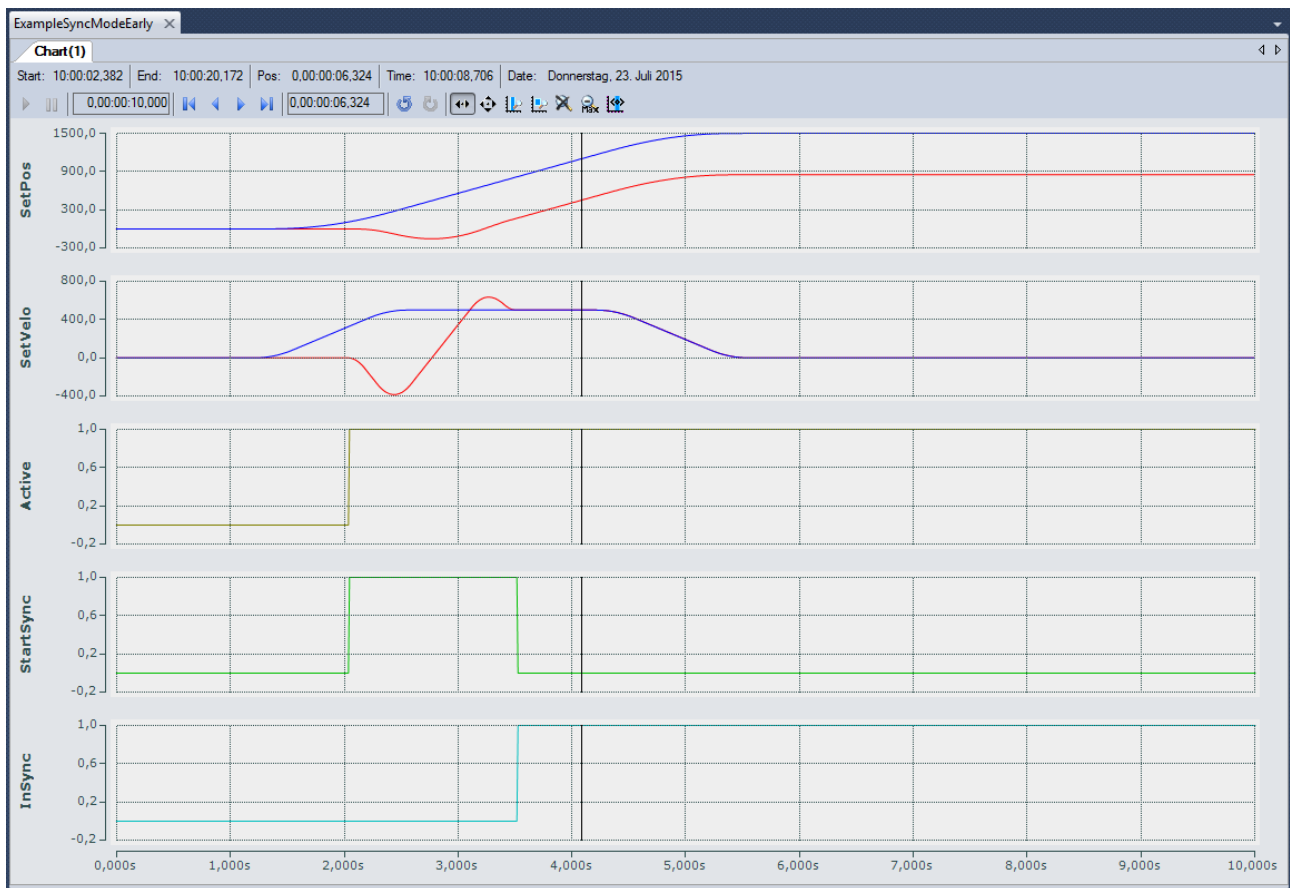
In diesem Fall wird am besten mcSyncStrategyEarly verwendet.



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 (x86 oder x64)	Tc3_McCollisionAvoidance, Tc3_McCoordinatedMotion, Tc2_MC2

10 Beispiele

PTP Collision Avoidance

XTS Demo 1

Download:

https://infosys.beckhoff.com/content/1031/tf5410_tc3_collision_avoidance/Resources/1546301963.zip

Beschreibung:

Projekt für XTS Starterkit (geschlossene Schiene (3000 mm) mit 10 Movern), das [MC_MoveAbsoluteCA \[▶ 34\]](#)-Bewegungen ausführt.

XTS Demo 2

Download:

https://infosys.beckhoff.com/content/1031/tf5410_tc3_collision_avoidance/Resources/1546304267.zip

Beschreibung:

Projekt für XTS Starterkit (geschlossene Schiene (3000 mm) mit 10 Movern), das [MC_GearInPosCA \[▶ 41\]](#)-Bewegungen ausführt.

11 Anhang

11.1 Zyklische Gruppenschnittstelle

Die zyklische Gruppenschnittstelle sorgt für den zyklischen Datenaustausch zwischen SPS und einem NC-Gruppenobjekt. Die Gruppenschnittstelle beinhaltet die Richtungen [NcToPlc \[► 114\]](#) und [PlcToNc \[► 115\]](#). 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 \[► 115\]](#) 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 \[► 114\]](#) 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.

11.1.1 NcToPlc

Die Struktur ist in allgemeine Daten und gruppenspezifische Daten unterteilt.

Allgemein

GroupOID: TcCOM-Objekt-ID (OID) dieser Gruppe.

GroupType: Typ dieser Gruppe: 0 = Ungültig (mcGroupTypeInvalid), 1 = Kollisionsvermeidung (mcGroupTypeCA), 2 = DXD/CNC (mcGroupTypeDxd).

GroupStatus: Enthält Informationen über den Gruppenstatus (siehe [GroupStatus \[► 114\]](#)).

GroupErrorId: Identifizierung aktueller Fehler (0 = kein Fehler).

GroupAxesCount: Anzahl der Achsen, die gegenwärtig zu dieser Gruppe gehören (z. B. über [MC_AddAxisToGroup](#) hinzugefügt).

GroupStatus:

State: Siehe Zustandsdiagramm einer Gruppe.

- 1 = Deaktiviert (mcGroupStateDisabled)
- 2 = Standby (mcGroupStateStandby)
- 3 = In Bewegung (mcGroupStateMoving)
- 4 = Stoppt (mcGroupStateStopping)
- 5 = Fehler Stopp (mcGroupStateErrorStop)
- 6 = Homing (mcGroupStateHoming)
- 7 = Nicht bereit (mcGroupStateNotReady)
- 8 = Ausgesetzt (mcGroupStateSuspended)

Flags: Zusätzliche Statusinformationen.

IsEnableRequested: Definiert, ob eine Aktivierung oder Deaktivierung einer Gruppe angefordert wird.

Dxd (mehrdimensionale Bewegung)

PathVelo: Geschwindigkeit auf der Bahn ohne Richtung.

Invokeld: Segment-ID für Analysezwecke.

CM (MC Group Coordinated Motion)

verfügbar ab V3.1.10.1

PathVelo: Absolutwert der kartesischen Geschwindigkeit auf der Bahn.

Invokeld: Segment-ID für Analysezwecke.

IsInBlendingSegment: Gibt an, ob ein Blendig-Segment aktiv ist.

RemainingTimeActiveJob: Verbleibende Zeit des aktuellen Segments.

RemainingCartesianDistanceActiveJob: Verbleibende Strecke für das aktuelle Segment.

ActiveBlockerId: Id des aktiven Blockers.

verfügbar ab V3.1.10.30

RemainingTimeToSync: Verbleibende Zeit, bis die Achsgruppe beim Conveyor Tracking mit dem Transportband synchronisiert ist.

RemainingCartesianDistanceToSync: Verbleibende Strecke, bis die Achsgruppe beim Conveyor Tracking mit dem Transportband synchronisiert ist.

11.1.2 PlcToNc

Die Struktur ist in allgemeine Daten und gruppenspezifische Daten unterteilt.

Allgemein

OverrideFactor: Gewünschter Override-Faktor (1,0 = 100 %, Standardwert ist 1,0)

11.2 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

Wert	Bedeutung	Beispiel
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

11.3 Modulo-Positionierung

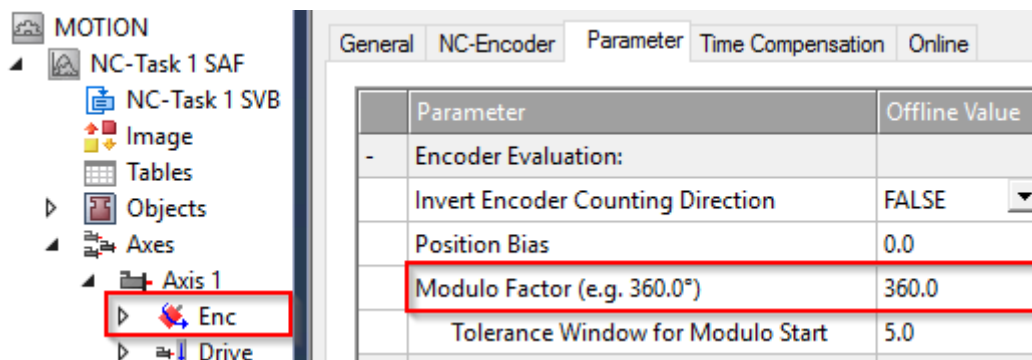
Die Modulo-Positionierung kann bei geschlossenen Linearachsen ebenso wie bei rotatorischen Achsen angewendet werden. TwinCAT unterscheidet nicht zwischen diesen Typen. Auch eine Modulo-Achse hat eine fortlaufende absolute Position im Bereich $\pm\infty$. Die Modulo-Position der Achse ist einfach eine zusätzliche Information zur absoluten Achsposition und die Modulo-Positionierung stellt die gewünschte Zielposition auf eine andere Art dar. Im Gegensatz zur absoluten Positionierung, bei der der Benutzer das Ziel eindeutig vorgibt, wird bei der Modulo Positionierung die absolute Zielposition aus folgenden Parametern gebildet:

- Modulo-Zielposition
- Modulo Factor
- Tolerance Window
- Direction, siehe [MC Direction](#) [► 108]
- (Additional Turns, siehe [Additional Turns in der CA](#) [► 49])

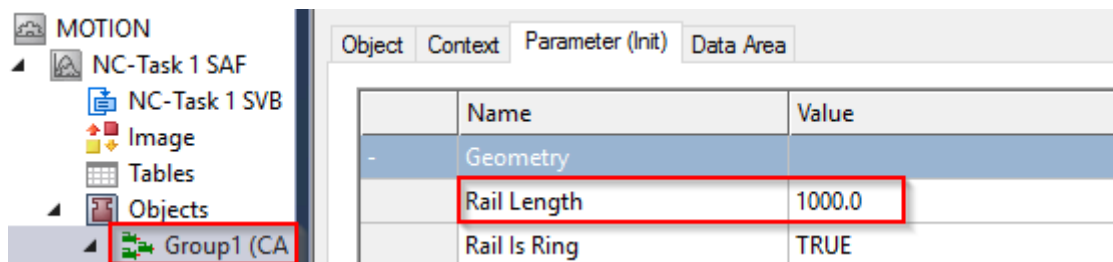
Modulo Factor

Die Modulo-Positionierung bezieht sich grundsätzlich auf einen einstellbaren Modulo Factor, welcher im TwinCAT Engineering eingestellt wird. Hierbei ist die Achse und ihre Verwendung zu beachten, zum Beispiel:

- Wird eine PTP-Achse verwendet, gilt der Modulo Factor des Achs-Encoders, Details in den Hinweisen der Modulo Positionierung einer PTP Achse.



- Wird z.B. eine Mover auf einem XTS-System in einer CA-Gruppe verwendet, gilt die Rail Length, welche in der CA-Gruppe eingestellt ist.



- Wird ein XPlanar Mover genutzt so kann dessen „C-Achse“ Modulo positioniert werden. Hier wird der Modulo Factor als „C coordinated modulus“ in den Init Parametern des XPlanar Mover eingestellt.

Object	Parameter (Init)	Parameter (Online)	Data Area	Settings
	Name			Value
-	General			
	Mover width			155.0
	Mover height			155.0
+	Initial position			...
	C coordinate modulus			360.0
	C coordinate modulo tolerance window			0.0

Modulo Tolerance Window

Das Modulo Tolerance Window definiert ein Positionsfenster um die aktuelle Modulo-Sollposition der Achse herum. Die Fensterbreite entspricht dem doppelten angegebenen Wert (Sollposition ± Toleranzwert) und wird im TwinCAT Engineering vorgegeben:

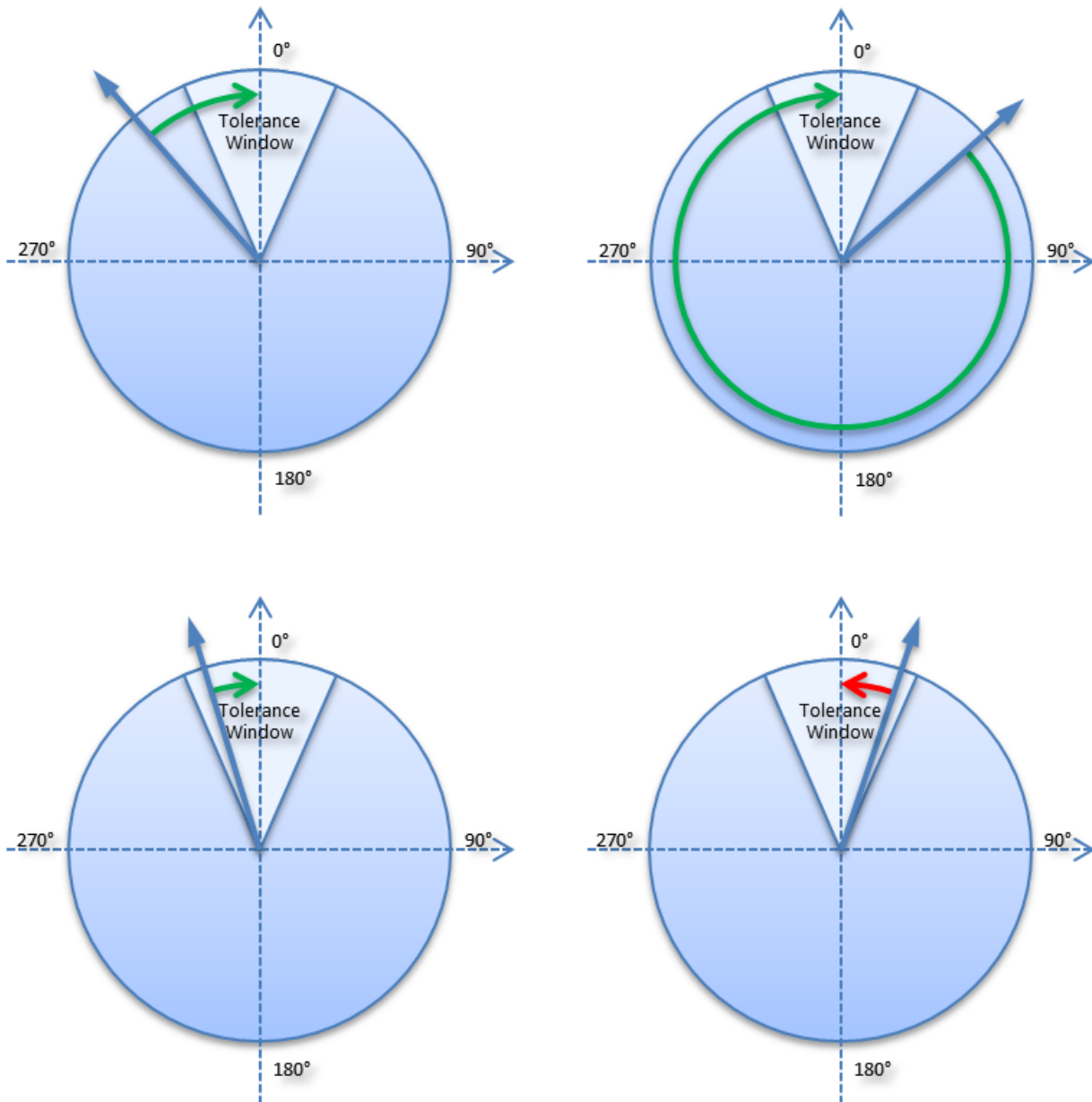
- Bei einer PTP-Achse oder einer Achse in einer CA-Gruppe erfolgt die Definition des Tolerance Window im Achs-Encoder

Parameter	Offline Value
- Encoder Evaluation:	
Invert Encoder Counting Direction	FALSE
Position Bias	0.0
Modulo Factor (e.g. 360.0°)	360.0
Tolerance Window for Modulo Start	5.0

- Bei der C-Achse eines XPlanar Movers wird das Tolerance Window in den Init Parametern des XPlanar Mover definiert.

Object	Parameter (Init)	Parameter (Online)	Data Area	Settings
	Name			Value
-	General			
	Mover width			155.0
	Mover height			155.0
+	Initial position			...
	C coordinate modulus			360.0
	C coordinate modulo tolerance window			0.0

Die Positionierung einer Achse bezieht sich immer auf deren aktuelle Ist-Position. Liegen Ist-Position und Zielposition sehr nah beieinander, kann es passieren, dass unbeabsichtigte Umdrehungen gefahren werden. Z.B. wenn die Ist-Position minimal größer ist als die Zielposition und `Direction = mcDirectionPositive` gewählt wurde. Dies kann insbesondere dann auftreten, wenn die Ist-Position ungenau bestimmt wird (z.B. aufgrund einer fehlerhaften Positionierung durch einen Stall der Achse, oder durch die endliche Auflösung des Encoders). Um dies zu vermeiden, kann ein Toleranzfenster für die Modulo Positionierung festgelegt werden. Falls die Entfernung zwischen Start- und Zielposition kleiner oder gleich dem Toleranzfenster ist, so wird die Zielposition auf kürzestem Weg (wie bei `Direction = mcDirectionShortestWay`), also auch entgegen der angegebenen `Direction`, angefahren.



Beispiele

- Modulo Factor = 100
- Tolerance Window = 1

Parameter Direction	Absolute Startposition	Zielposition	Parameter Additional Turns	Relative Pfad	Absolute Endposition	Modulo Endposition
mcDirectionPositive	110	10	0	0	110	10
mcDirectionPositive	110.9	10	0	-0.9	110	10
mcDirectionPositive	112	10	0	98	110	10
mcDirectionPositive	95	10	0	15	110	10
mcDirectionPositive	110	110	0	ERROR: INVALID TARGET POSITION		
mcDirectionPositive	110	10	3	300	410	10
mcDirectionPositive	110.9	10	3	299.1	410	10
mcDirectionPositive	112	10	3	398	410	10

Parameter Direction	Absolute Startposition	Zielposition	Parameter Additional Turns	Relativer Pfad	Absolute Endposition	Modulo Endposition
mcDirectionPositive	95	10	3	315	410	10
mcDirectionPositive	110	110	3	ERROR: INVALID TARGET POSITION		
mcDirectionNegative	110	10	0	0	110	10
mcDirectionNegative	109.9	10	0	0.1	110	10
mcDirectionNegative	108	10	0	-98	10	10
mcDirectionNegative	95	10	0	-85	10	10
mcDirectionNegative	110	110	0	ERROR: INVALID TARGET POSITION		
mcDirectionNegative	410	10	3	-300	110	10
mcDirectionNegative	409.9	10	3	-299.9	110	10
mcDirectionNegative	408	10	3	-398	10	10
mcDirectionNegative	495	10	3	-385	10	10
mcDirectionNegative	410	110	3	ERROR: INVALID TARGET POSITION		
mcDirectionShortestWay	440	50	0	10	450	50
mcDirectionShortestWay	440	10	0	-30	410	10
mcDirectionShortestWay	440	50	1	ERROR: INVALID ADDITIONAL TURN COUNT		

Weitere Beispiele

Weitere Beispiele ohne den Parameter Additional Turns sind in den [Hinweisen zur Modulo Positionierung](#) einer PTP Achse zu finden.

Mehr Informationen:
www.beckhoff.de/TF5410

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

