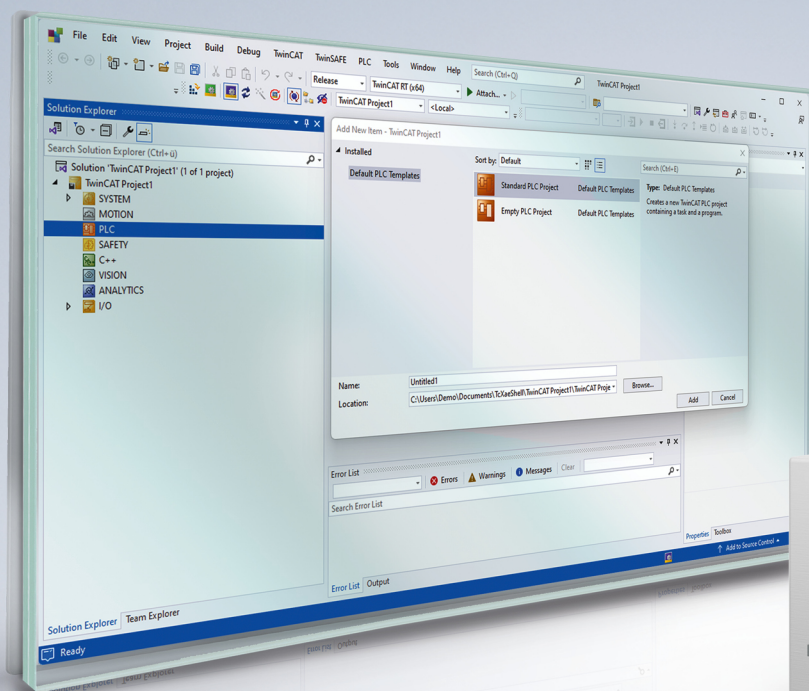


Handbuch | DE

TF55xx

TwinCAT 3 | MC3



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit	7
2	Einführung TwinCAT MC3.....	8
2.1	Installation	8
2.2	Lizenzmodell	9
3	Entwicklungsumgebung.....	12
3.1	SYSTEM	12
3.1.1	Real-Time.....	12
3.1.2	Tasks.....	13
3.1.3	TcCOM Objects.....	13
3.2	MOTION.....	16
3.2.1	MC Project	17
3.2.2	Axes	23
3.2.3	Default Axis	25
3.2.4	Encoder Axis	135
3.2.5	FluidPower Axis	139
3.2.6	Planar Mover.....	190
3.2.7	Groups	190
3.2.8	Tables	190
3.2.9	Objects	194
3.2.10	Erweiterungen	194
3.3	Anhang.....	196
4	Globale Datentypen und MC3 Typen.....	198
4.1	Data Types	198
4.1.1	Enums	198
4.1.2	Structs	209
4.2	Interfaces	216
4.2.1	ITcMcExternalSetpointGenerator.....	216
4.3	Übersicht: MC3-Typen	219
5	PLC Bibliotheken.....	224
5.1	Benannte Konstanten.....	224
5.2	Tc3_Mc3Base	225
5.2.1	Data Types.....	226
5.2.2	Interfaces	229
5.2.3	Function Blocks.....	232
5.3	Tc3_Mc3Ptp	234
5.3.1	Zustandsdiagramm	234
5.3.2	Einfaches Verfahren der Achse über die PLC	236
5.3.3	Data Types.....	239
5.3.4	Interfaces	250
5.3.5	Function Blocks.....	253

5.3.6	Anhang.....	304
5.4	Tc3_Mc3Camming	315
5.4.1	Tc3_Mc3Camming	315
5.4.2	Data Types.....	320
5.4.3	Interfaces	329
5.4.4	Function Blocks.....	330
5.4.5	Anhang.....	347
5.5	Tc3_Mc3FluidPower	348
5.5.1	Data Types.....	348
5.5.2	Function Blocks.....	349
5.6	Beispielprogramme	355
6	Support und Service	357

1 Vorwort

1.1 Hinweise zur Dokumentation

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

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

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

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Kennzeichnungen führen.



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

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

Fremdmarken

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

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.

Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

Haftungsausschluss

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

Qualifikation des Personals

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

Signalwörter

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

Warnungen vor Personenschäden

GEFAHR

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

WARNUNG

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

VORSICHT

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

Warnung vor Umwelt- oder Sachschäden

HINWEIS

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

Information zum Umgang mit dem Produkt



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

1.3 Hinweise zur Informationssicherheit

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

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

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

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

2 Einführung TwinCAT MC3

TF55xx | TwinCAT 3 MC3 ist die neue Motion Control Generation, die sich nahtlos in TwinCAT integriert und damit eine durchgängige und effiziente Automatisierungslösung darstellt.

Vollständige Systemintegration

Die nahtlose Systemintegration von TF55xx | TwinCAT 3 MC3 in das TwinCAT-System bedeutet, dass sich die Motion-Control-Lösung nicht nur parallel zu den anderen TwinCAT-Funktionen auf einem System betreiben lässt, sondern auch mit den anderen TwinCAT-Funktionen interagieren kann.

Multi-core Unterstützung

TF55xx | TwinCAT 3 MC3 lässt sich bedarfsgerecht auf mehrere CPU-Kerne verteilen. Dabei sind synchronisierte Bewegungen über alle verwendeten CPU-Kerne möglich.

Multitasking Unterstützung

Auf einem CPU-Kern können Achsen mit unterschiedlichen Zykluszeiten betrieben werden, genau passend zu den Geschwindigkeiten und Aufgaben der jeweiligen Achsen. So ist eine optimierte Ausnutzung eines CPU-Kerns möglich, da die „schnellste“ Achse nicht zwangsweise den Takt für alle Achsen vorgibt. Auf diese Weise können die Achsen eines Transportbands mit Delta-Picker mit 1 ms Zykluszeit betrieben werden, um ein schnelles Aufnehmen der Werkstücke sicherzustellen, während die Achsen zur Breitenverstellung des Transportbands mit 4 ms Zykluszeit auf demselben CPU-Kern betrieben werden.

Modulare Konfiguration

Funktionserweiterungen mit TcCOM-Objekten und flexible Schnittstellen ermöglichen eine zukunftsorientierte Architektur und eine modulare Projektkonfiguration. Die webbasierte Inbetriebnahmeoberfläche lässt sich Anwendungsspezifisch anpassen.

Hardwareunabhängigkeit

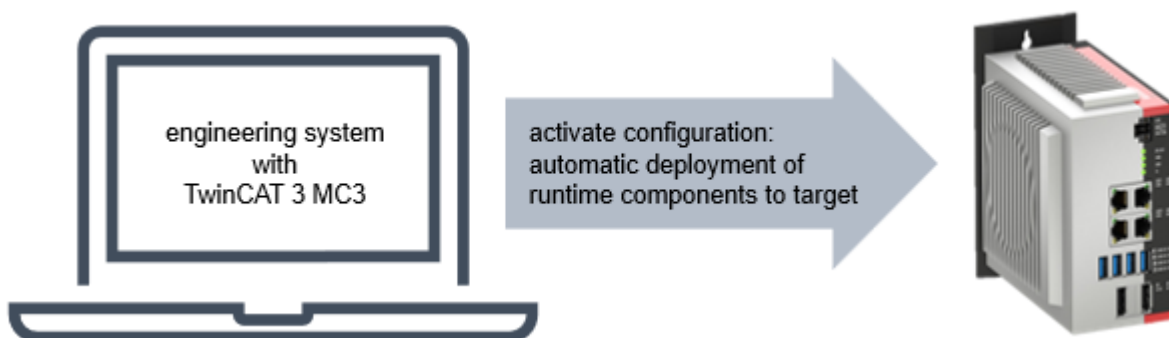
Wie bei der bisherigen Motion-Control-Lösung werden durch Achsobjekte die Hardware und die Programmierung voneinander getrennt. Auf diese Weise entsteht eine Hardwareunabhängigkeit.

Fluid Power (Hydraulik) Unterstützung

TF55xx | TwinCAT 3 MC3 gewährleistet eine gleichermaßen tiefe Integration von elektrischen und hydraulischen Achsen und ermöglicht eine nahtlose Interaktion, präzise Synchronisation und flexible Integration über verschiedene Hardwareplattformen hinweg. Mit der skalierbaren Plattform kann TF5560 | TwinCAT 3 MC3 Fluid Power für eine Vielzahl von Anwendungsfällen eingesetzt werden, von hydraulischen Insellösungen bis hin zu kompletten Maschinensteuerungssystemen.

2.1 Installation

TF55xx | TwinCAT 3 MC3 wird nur auf Engineering-Seite installiert. Auf Runtime-Seite wird eine TwinCAT Standard Installation ab der Version 3.1 Build 4026.23.1 vorausgesetzt. Beim Aktivieren der Konfiguration werden spezifische MC3 Komponenten vom Engineering-System auf das Runtime-System übertragen. Das dem Runtime-System zugrunde liegende Betriebssystem wird dabei automatisch ermittelt und berücksichtigt.



Engineering Installation (Windows)

Ab TwinCAT 3.1 Build 4026 werden die TwinCAT-Produkte über den TwinCAT Package Manager installiert. Eine ausführliche Anleitung zur Installation von Produkten finden Sie im Kapitel [TwinCAT Software verwalten](#) in der [Installationsanleitung TwinCAT 3.1 Build 4026](#).

Eine Basis TwinCAT XAE-Installation erhalten Sie über folgendes Workload:

- TwinCAT.Standard.XAE

Grundlegende MC3 Komponenten werden mit folgendem Workload installiert:

- TF5500.MC3.PTP.XAE

Weitere MC3 Komponenten können zusätzlich über folgende Workloads installiert werden:

- TF5550.MC3.Camming.XAE
- TF5560.MC3.FluidPower.XAE

Darüber hinaus empfiehlt sich die Installation des Drive Manager 2 für die Verwendung und Parametrierung von Beckhoff-Motion-Hardware sowie des TwinCAT Scopes:

- TE5950.DriveManager2.XAE (Engineering) (>= v1.1.114.0 empfohlen)
- TE1300.ScopeViewProfessional.XAE (Engineering)
- TF3300.ScopeServer.XAR (Runtime)

Runtime Installation

Auf Runtime Seite wird nur eine TwinCAT Standard Installation ab der Version 3.1 Build 4026.23.1 vorausgesetzt. Alle notwendigen TwinCAT MC3 Komponenten werden beim Aktivieren der Konfiguration vom Engineering-System aufs Runtime-System übertragen.

Windows XAR Installation über den TwinCAT Package Manager mit dem Workload:

- TwinCAT.Standard.XAR

[Beckhoff RT Linux® Runtime](#) Installation über den Beckhoff Linux Paketserver und folgendem Paket:

- tc31-xar-um

[TwinCAT/BSD Runtime](#) Installation:

- Die TwinCAT Runtime Installation ist in der TwinCAT/BSD-Installation enthalten.

2.2 Lizenzmodell

Das TwinCAT 3 MC3 Lizenzmodell gliedert sich auf in eine Basis Lizenz, Achserweiterungslizenzen und Funktionserweiterungslizenzen.

TF5500 | TwinCAT 3 MC3 Base

Die Basis Lizenz TF5500 | TwinCAT 3 MC3 Base ist Voraussetzung für jede MC3 Achsanwendung.

TF5500 | TwinCAT 3 MC3 Base inkludiert 10 Achslizenzen. Eine Achslizenz gilt zunächst nur für Beckhoff- oder Simulationsachsen. Durch eine MC3-Axes-Pack-Lizenz (TF551x) lässt sich die Achsanzahl flexibel erweitern.

Achsen anderer Hersteller können ebenfalls angesteuert werden. Für diese Fremdashen ist neben einer Achslizenz außerdem eine Multi-Vendor-Lizenz erforderlich. Fünf Multi-Vendor-Lizenzen sind in der MC3-Base-Lizenz inkludiert. Weitere können durch ein zusätzliches TwinCAT 3 MC3 Multi Vendor Pack (TF553x) erworben werden.

So ergeben sich mögliche Konfigurationen aus fünf Beckhoff- bzw. Simulationsachsen und fünf Achsen anderer Hersteller (in Summe 10 Achslizenzen). Entscheidend ist die Summe von maximal 10 Achsen, wobei maximal fünf Achsen anderer Hersteller eingesetzt werden können. Die Kombination von vier

Beckhoff- bzw. Simulationsachsen und sechs Achsen anderer Hersteller ist ohne ein zusätzliches TwinCAT 3 MC3 Multi Vendor Pack und Axes Pack nicht möglich. Das Axes Pack wird benötigt, da ein Multi Vendor Pack immer mindestens ein gleich großes Axes Pack voraussetzt.

In der MC3-Base-Lizenz sind neben Achslizenzen die Funktionalitäten zur Realisierung von Point-to-Point(PTP)-Anwendungen und Kopplungen, wie Fliegende Säge, inkludiert. Weitere Funktionalitäten stehen über ergänzende Produkte (TF5550, TF5560) zur Verfügung.

TF551x | TwinCAT 3 MC3 Axes Packs

TwinCAT 3 MC3 Axes Packs erweitert TF5500 TwinCAT 3 MC3 Base um zusätzliche Achslizenzen.

Je nach TF551x-Lizenz und Plattform kann die Achsanzahl auf 25, 50, 75, 100, 150, 250, 500, 1000 oder 1500 Achsen erweitert werden. Die Achsanzahl aus den TwinCAT 3 MC3 Axes Packs gilt als neue maximale Achsanzahl. Die im TwinCAT 3 MC3 Base inkludierten 10 Achslizenzen (bzw. davon bis zu fünf Achsen anderer Hersteller) sind in der Summe bereits enthalten und werden nicht dazu addiert. Eine Kombination von TwinCAT 3 MC3 Axes Packs ist nicht möglich.

Eine Achslizenz gilt zunächst nur für Beckhoff- oder Simulationsachsen. Auch Achsen anderer Hersteller können angesteuert werden. Für diese Fremdachsen ist neben einer Achslizenz zusätzlich noch eine Multi-Vendor-Lizenz erforderlich. Fünf Multi-Vendor-Lizenzen sind in der MC3-Base-Lizenz inkludiert. Weitere können durch ein zusätzliches MC3 Multi Vendor Pack (TF553x) erworben werden.

TF-Nummer	Name
TF5510	TwinCAT 3 MC3 Axes Pack 25
TF5511	TwinCAT 3 MC3 Axes Pack 50
TF5512	TwinCAT 3 MC3 Axes Pack 75
TF5513	TwinCAT 3 MC3 Axes Pack 100
TF5514	TwinCAT 3 MC3 Axes Pack 150
TF5515	TwinCAT 3 MC3 Axes Pack 250
TF5516	TwinCAT 3 MC3 Axes Pack 500
TF5517	TwinCAT 3 MC3 Axes Pack 1000
TF5518	TwinCAT 3 MC3 Axes Pack 1500



Mehr als 1500 Achsen

Die Lizenz TF5518 | TwinCAT 3 MC3 Axes Pack 1500 stellt nicht die tatsächliche maximal mögliche Achsanzahl dar. Sollten Sie die Ansteuerung von mehr als 1500 Achsen benötigen, wenden Sie sich bitte an Ihren Beckhoff Ansprechpartner.

TF553x | TwinCAT 3 MC3 Multi Vendor Packs

TwinCAT 3 MC3 unterstützt neben der Ansteuerung von Beckhoff Achsen auch die Ansteuerung von Achsen anderer Hersteller.

Für die Achsen anderer Hersteller ist neben einer Achslizenz jeweils eine Multi-Vendor-Lizenz erforderlich. Fünf Multi-Vendor-Lizenzen sind in der MC3-Base-Lizenz inkludiert. Weitere können durch ein zusätzliches TwinCAT 3 MC3 Multi Vendor Pack erworben werden.

Für die erforderlichen Achslizenzen ist ergänzend eine Variante von TF551x | TwinCAT 3 MC3 Axes Pack erforderlich.

TF-Nummer	Name
TF5530	TwinCAT 3 MC3 Multi Vendor Pack 25
TF5531	TwinCAT 3 MC3 Multi Vendor Pack 50
TF5532	TwinCAT 3 MC3 Multi Vendor Pack 75
TF5533	TwinCAT 3 MC3 Multi Vendor Pack 100
TF5534	TwinCAT 3 MC3 Multi Vendor Pack 150

TF5550 und TF5560 Funktionserweiterungslizenzen

TF5500 ermöglicht die Realisierung von PTP-Anwendungen mit einfachen Kopplungen. Mit den nachfolgenden Lizenzen kann der Funktionsumfang erweitert werden.

TF-Nummer	Voraussetzung	Beschreibung
TF5550	TF5500	TwinCAT 3 MC3 Camming Erweiterung um nicht lineare Kopplungen (Kurvenscheiben).
TF5560	TF5500	TwinCAT 3 MC3 Fluid Power Erweiterung um Fluid Power Funktionalitäten.

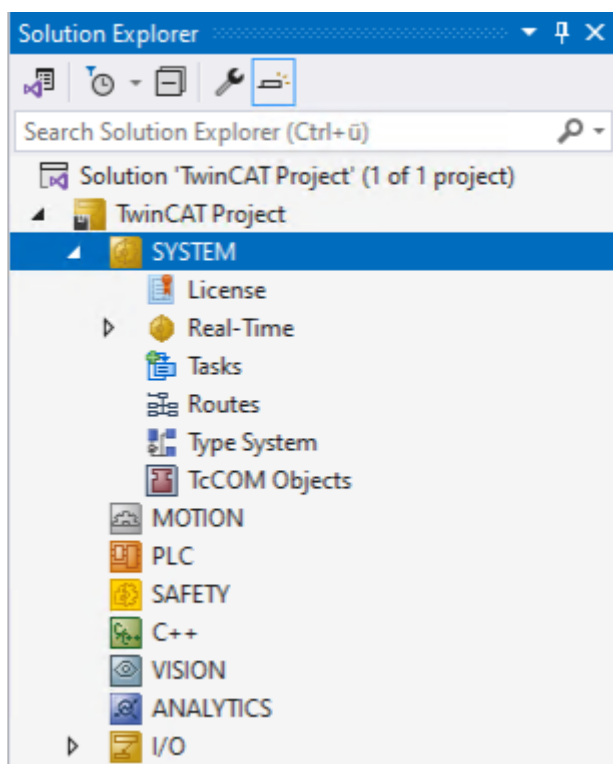
3 Entwicklungsumgebung

Die TwinCAT MC3 ist nahtlos in die TwinCAT Entwicklungsumgebung (XAE) auf Basis des Microsoft Visual Studios integriert. Innerhalb eines TwinCAT-Projekts ist der **MOTION** [► 16]-Knoten der wesentlichste Projektknoten für die TwinCAT MC3. Zudem ist der **SYSTEM** [► 12]-Knoten für allgemeine Einstellungen relevant. Beide Knoten sind nachfolgend im Zusammenhang mit der TwinCAT MC3 beschrieben.

3.1 SYSTEM

Der Knoten SYSTEM im Strukturbaum des TwinCAT-Projekts im Solution Explorer enthält die Systemkonfiguration. Dazu gehören u.a. die Echtzeiteinstellungen, Taskkonfiguration und Lizenzen. In dieser Dokumentation wird auf den Knoten SYSTEM nur in Kombination mit der TwinCAT 3 MC3 eingegangen.

Allgemeine Informationen zum SYSTEM-Knoten, siehe [TE1000 XAE Dokumentation](#).



3.1.1 Real-Time

Die Berechnung der Achsen der TwinCAT MC3 lässt sich bedarfsgerecht auf mehrere CPU-Kerne verteilen. Eine Achse ist einer zyklischen Task zugeordnet. Einer Task können mehrere Achsen zugeteilt sein. Die Zykluszeiten und Priorisierungen können anhand der Aufgabe der Achsen für eine Task eingestellt werden. Dabei sind synchronisierte Bewegungen über alle verwendeten CPU-Kerne möglich.

Object	RT-Core	Base Time (ms)	Cycle Time (ms)	Cycle Ticks	Priority
MC-Task (MET)	Core 0	1 ms	2 ms	2	3
MC-Task (MET)_1	Core 1	1 ms	1 ms	1	4
MC-Task (MPT)	Core 0	1 ms	10 ms	10	8
MC-Task (MPT)_1	Core 1	1 ms	5 ms	5	9
I/O Idle Task	Core 2	1 ms	1 ms	1	11
Job Task	Core 2	1 ms	(none)	0	32

Die Zuordnung von Tasks zu CPU-Kernen erfolgt im TwinCAT-Projekt unter **SYSTEM > Real-Time > Settings**. Um die aktuelle CPU-Kern-Konfiguration des Zielsystems anzuzeigen, klicken Sie auf **Read from Target**. Anschließend können die Tasks auf die entsprechenden Kerne verteilt werden.

Um die CPU-Kern-Konfiguration des Zielsystems zu verändern, klicken Sie auf **Set on Target** und wählen die gewünschte Verteilung von geteilten und isolierten Kernen aus.

Auch auf dem gleichen CPU-Kern können Achsen mit unterschiedlichen Zykluszeiten betrieben werden, indem die Achsen verschiedenen Kontexten [► 19] zugeordnet werden. Die „schnellste“ Achse gibt somit nicht den Takt für alle Achsen vor. Dies ermöglicht eine optimierte Ausnutzung eines CPU-Kerns – bereits für einige wenige Achsen.

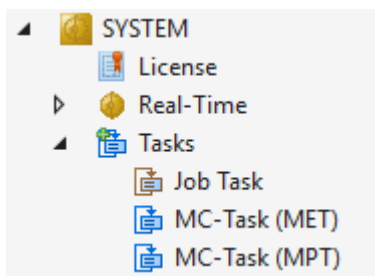
3.1.2 Tasks

Die Konfiguration von Tasks ist nicht spezifisch für TwinCAT Motion, sondern funktioniert für alle TwinCAT Komponenten gleich. Im Folgenden sind die relevanten Einstellungen in Bezug auf TwinCAT Motion beschrieben.

Grundsätzlich werden alle für TwinCAT Motion benötigten zyklischen Tasks automatisch angelegt. Es handelt sich dabei um die folgenden Tasks, die zusammen einen Kontext bilden:

- Motion Execution Task (MET)
- Motion Preparation Task (MPT)
- Job Task

In der MET werden Kommandos verarbeitet und die zyklische Sollwertgenerierung durchgeführt. Ebenso übernimmt diese Task die I/O Kommunikation. In der MPT werden vorbereitende Berechnungen ausgeführt. Die Job Task wird bislang nicht verwendet und ist für zukünftige nicht-zyklische Berechnungen vorgesehen.



Es wird standardmäßig die MET mit 2 ms und die MPT mit 10 ms erstellt. Unter **SYSTEM > Tasks > [Taskname]** können Sie die Einstellungen der Task verändern. MET und MPT müssen auf dem gleichen Kern ausgeführt werden.

Konkret können Achsen, die kleinere Zykluszeiten benötigen, einem Kontext mit einer MET mit beispielsweise 1 ms Zykluszeit zugeordnet werden. Achsen, die eine geringere Taktfrequenz benötigen, können dann einem weiteren Kontext mit einer zweiten MET mit z. B. 4 ms Zykluszeit angehören, die auf dem gleichen oder einem anderen Kern betrieben werden kann. Das Anlegen des Kontext und die Zuordnung der Achsen erfolgt im Context Dialog des Motion Projektes [► 19].

3.1.3 TcCOM Objects

Im Knoten **SYSTEM > TcCOM Objects** unter dem Reiter **Project Objects** sind alle TcCOM-Objekte des Projektes aufgelistet. Unter anderem entspricht eine MC3-Achse oder auch ein Kurvenscheibenobjekt in der MC3 einem TcCOM-Objekt.

In der Spalte **Version** können Sie die Version der jeweiligen Objekte ablesen. Für neue Projekte werden die Versionen der Objekte automatisch auf die neueste Version eingestellt und es ist keine Benutzeraktion erforderlich. Soll ein Projekt, das mit einer älteren Software-Version erstellt wurde, auf eine aktuellere Version umgestellt werden, ist die Einstellung an dieser Stelle möglich. Alternativ kann die Version über die Erweiterung „Motion Versioning“ [► 195] eingestellt werden.

i Unterschiedliche Versionen bei hierarchischen Objekten

Unterschiedliche Versionen zwischen unterlagerten Objekten und ihrem übergeordneten Objekt können zu Fehlern beim Aktivieren des Projekts führen.

- Stellen Sie sicher, dass für das überlagerte Kontext Element und alle unterlagerten Motion Control Objekte die gleiche Version eingestellt und aktiv ist.

3.1.3.1 Versionshandling

Die Pakete der TwinCAT MC3 können gleichzeitig in mehreren Versionen auf Ihrem Engineering-System installiert sein. Dies ist insbesondere der Fall, wenn Ihre Maschinen im Feld mit verschiedenen Software-Versionen ausgeliefert werden. Die nachfolgenden Abschnitte machen Sie mit dem Umgang verschiedener Versionen vertraut.

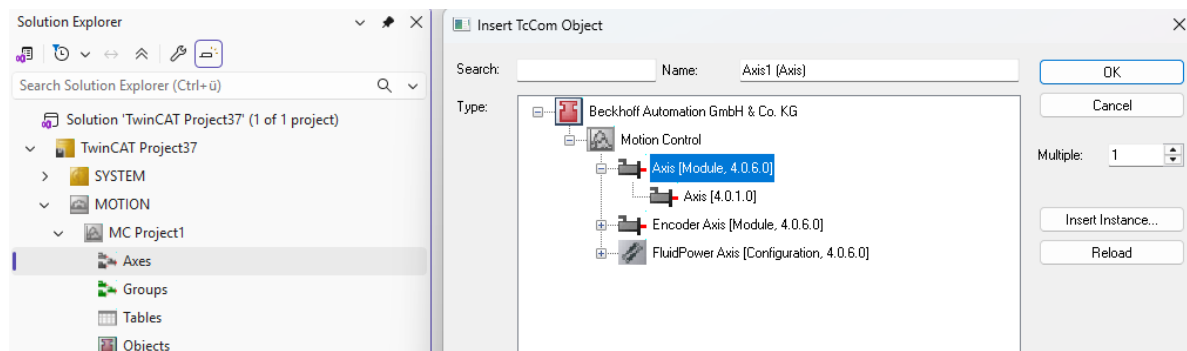
Alternativ kann die Version über die Erweiterung „Motion Versioning“ [► 195] eingestellt werden.

Voraussetzungen

- Ein TwinCAT-Projekt mit MC-Projekt wurde erstellt, siehe [Erstellen eines MC-Projekts \[► 17\]](#).
- Optional ist bereits eine Achse innerhalb des MC-Projekts angelegt. Falls Sie noch nicht damit vertraut sind, eine Achse innerhalb des MC-Projekts anzulegen, schlagen Sie im Kapitel [Erstellen eines Achsobjekts \[► 25\]](#) nach.

Achse in einer bestimmten Version dem MC-Projekt hinzufügen

1. Machen Sie einen Rechtsklick auf **Axes > Add New Item...**
2. Wählen Sie im Dialog **Insert TcCom Object** unter **Type** den Achstyp sowie die Version des gewünschten Objektes aus.
3. Erweitern Sie dazu die Ansicht zum gewünschten Objekt durch einen Klick auf das **+**.
 - ⇒ Es werden alle installierten Versionen angezeigt.
 - ⇒ In der Standardeinstellung wird die neueste (höchste) installierte Version des Objektes verwendet. In diesem Fall muss die Ansicht nicht aufgeklappt werden.



4. Geben Sie abschließend unter **Multiple** die gewünschte Anzahl an. Klicken Sie anschließend den Button **OK**, um die Eingaben zu bestätigen.
 - ⇒ Das Objekt wurde angelegt und befindet sich unterhalb des Axes-Knoten.

Versionseinstellung für Motion-Objekte in bestehenden Projekten vornehmen

i Unterschiedliche Versionen bei hierarchischen Objekten

Unterschiedliche Versionen zwischen unterlagerten Objekten und ihrem übergeordneten Objekt können zu Fehlern beim Aktivieren des Projekts führen.

- Stellen Sie sicher, dass für das überlagerte Kontext Element und alle unterlagerten Motion Control Objekte die gleiche Version eingestellt und aktiv ist.

1. Wählen Sie den Knoten **SYSTEM > TcCOM Objects (1)**.
2. Wechseln Sie in den Reiter **Project Objects (2)**.

⇒ Dort sind alle TcCOM-Objekte des Projektes aufgelistet. Unter anderem entspricht eine MC3-Achse oder auch ein Kurvenscheibenobjekte in der MC3 einem TcCOM-Objekt.

3. Stellen Sie in der Spalte **Version** die entsprechend gewünschte Version eines Objektes ein.

⇒ Die aktuell geladene und aktive Version eines Objektes ist mit **x.x.x.x (Current)**, gegenüber der ausgewählten Version **x.x.x.x**, gekennzeichnet (3).

⇒ Versionen eines Objektes, für das eine neuere Version auf dem Entwicklungssystem installiert ist, werden gelb markiert.

OTCID	Name	Version	TMC Filename
05100010	Context1	4.0.6.0 (Current)	C:\ProgramData...
05110010	Axis1 (Axis)	4.0.1.0 (Current)	C:\ProgramData...
05110020	Axis2 (Axis)	4.0.1.0 (Current)	C:\ProgramData...
05110030	Axis3 (Axis)	4.0.6.0 (Current)	C:\ProgramData...
01010030	SimEncoderDriv...	4.0.6.0 (Current)	C:\ProgramData...
01010040	CoE DS402 Drive	4.0.6.0 (Current)	C:\ProgramData...

4. Sie können eine geänderte Version auf mehrere Objekte übertragen. Wählen Sie **Change version for all with same TMI/TMC Description** um die Version einer Achse für alle anderen Achsen zu übernehmen (Objekte gleichen Typs). Mit der Auswahl **Change version for all with same TMI/TMC File** übernehmen Sie die Version für alle Objekte aus der gleichen TMC-Datei, also z. B. für Achsen und die unterlagerten Antriebs-/Geberobjekte (4).

5. Abschließend muss eine geänderte Version eines Objektes aktiviert werden. Laden Sie dazu das Objekt mit der geänderten Version und der damit verbundenen TMC-Objektbeschreibung neu. Klicken Sie auf **Reload TMI/TMC Description(s) with changed version** (5).

OTCID	Name	Version	TMC Filename
05100010	Context1	4.0.6.0 (Current)	C:\ProgramData...
05110010	Axis1 (Axis)	4.0.6.0	C:\ProgramData...
05110020	Axis2 (Axis)	4.0.6.0	C:\ProgramData...
05110030	Axis3 (Axis)	4.0.6.0	C:\ProgramData...
01010030	SimEncoderDriv...	4.0.6.0	C:\ProgramData...
01010040	CoE DS402 Drive	4.0.6.0	C:\ProgramData...

3.2 MOTION

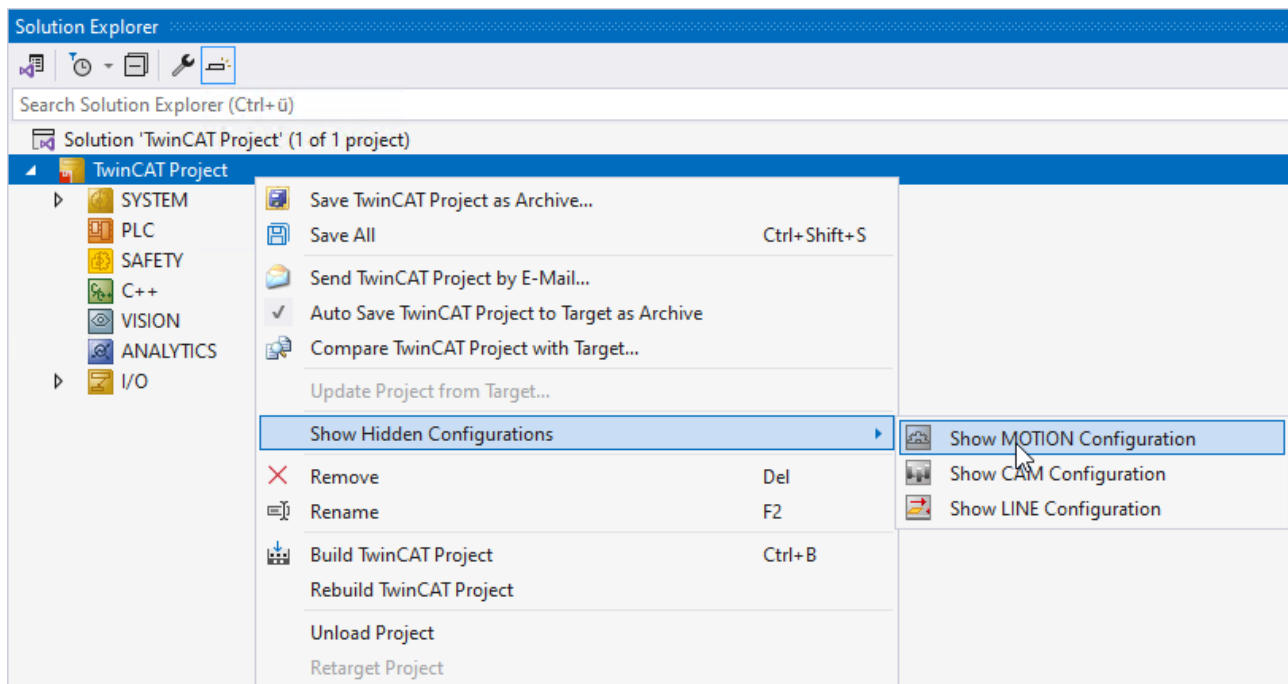
Über den Knoten MOTION im Strukturbaum des TwinCAT-Projekts im Solution Explorer werden die NC2-, MC3- und CNC-Konfigurationen verwaltet.



In dieser Dokumentation wird ausschließlich auf die MC3-Konfiguration eingegangen.

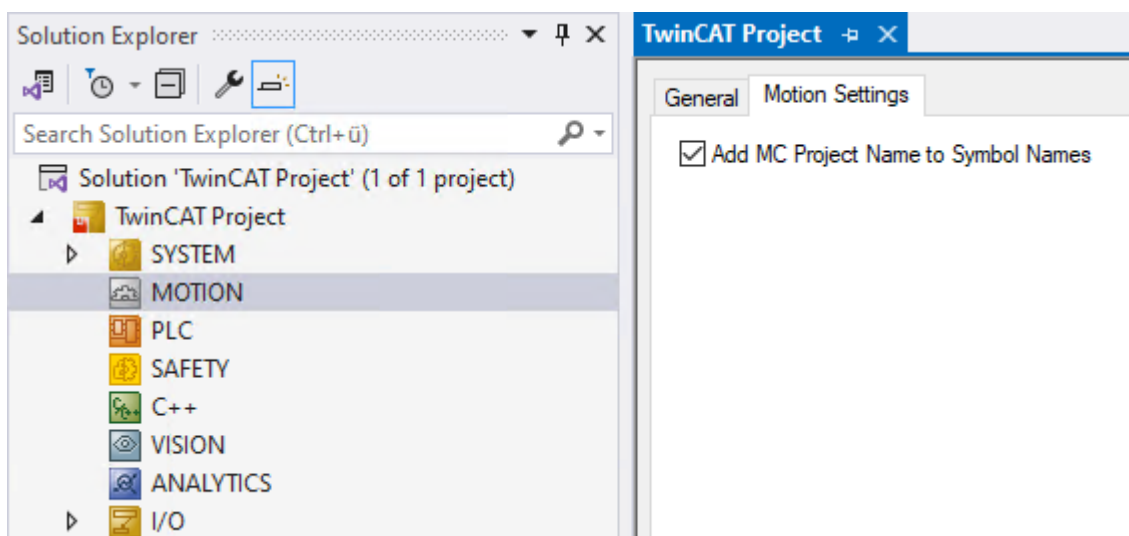
MOTION-Knoten anzeigen

Wenn der MOTION-Knoten im TwinCAT System nicht angezeigt wird, so kann dieser über einen Rechtsklick auf **TwinCAT Project** > **Show Hidden Configurations** > **Show MOTION Configuration** eingeblendet werden.



Registerkarte Motion Settings

Die Registerkarte **Motion Settings** erlaubt Motion-Einstellungen, die alle untergeordneten Projekte (eines Typs) betreffen.



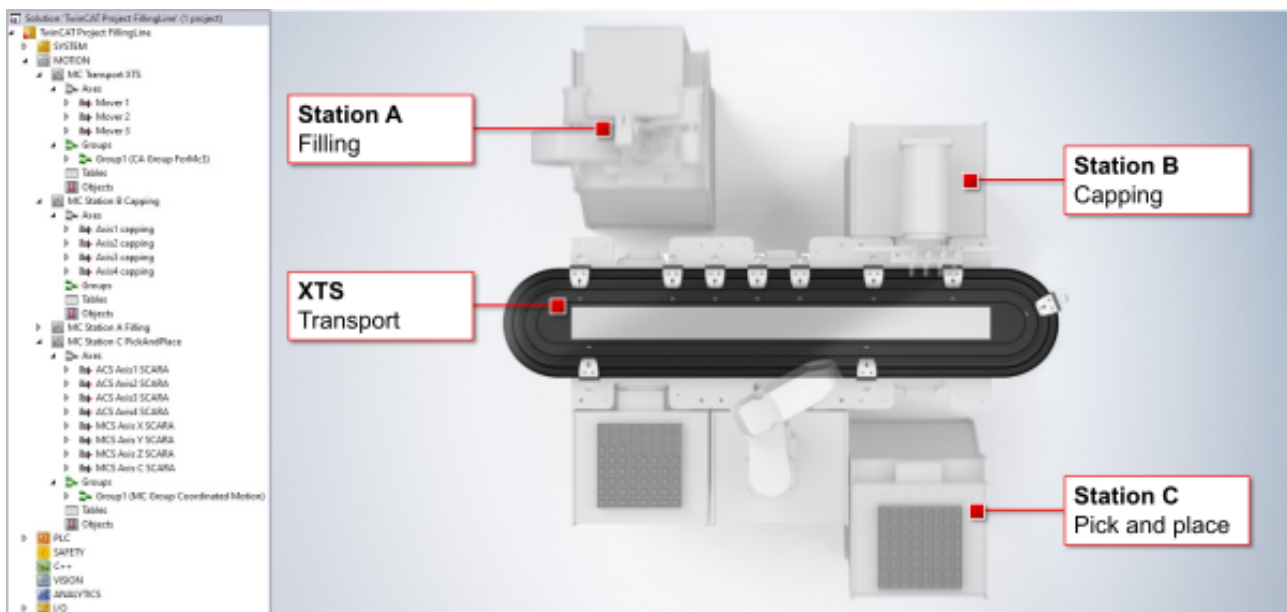
Add MC Project Name to Symbol Names

☐	Standard-Einstellung Befindet sich nur ein MC-Projekt in einem TwinCAT-Projekt, so ist der Name des MC-Projekts nicht Teil von ADS-Symbolnamen. Befinden sich mehrere MC-Projekte in einem TwinCAT-Projekt, so beginnen die ADS-Symbolnamen eines MC-Projekts mit dem Namen des MC Projects, um namensgleiche ADS-Symbole zu vermeiden.
☒	Die ADS-Symbolnamen eines MC-Projekts beginnen mit dem Namen des MC-Projekts.
☐	Die ADS-Symbolnamen eines MC-Projekts enthalten nicht den Namen des MC-Projekts. Bei mehreren MC-Projekts innerhalb eines TwinCAT-Projekts, obliegt es dem Anwender doppelte ADS-Symbolnamen zu verhindern.

3.2.1 MC Project

Ein MC Project ist die Organisationseinheit von MC3 Komponenten, wie Achsobjekten, Gruppen, Tabellen und weiteren Objekten.

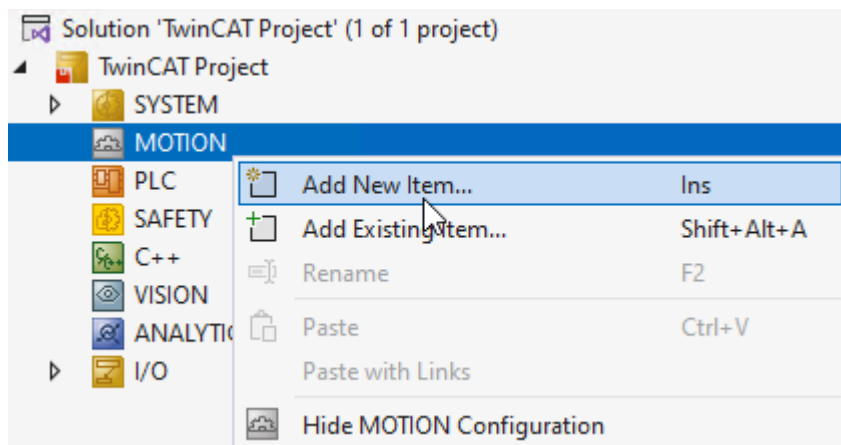
Innerhalb eines TwinCAT-Projekts können mehrere MC-Projekte angelegt werden. Zum Beispiel je ein MC-Projekt je Maschinenmodul, wie in der nachfolgenden Grafik dargestellt wird. Es ist auch weiterhin möglich, alle Achsen über ein MC-Projekt zu verwalten.



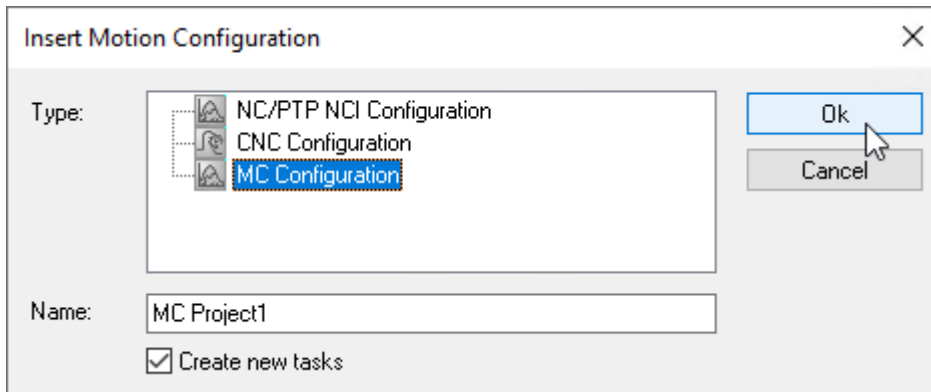
3.2.1.1 MC-Projekt erstellen

✓ Ein TwinCAT-Projekt wurde erstellt.

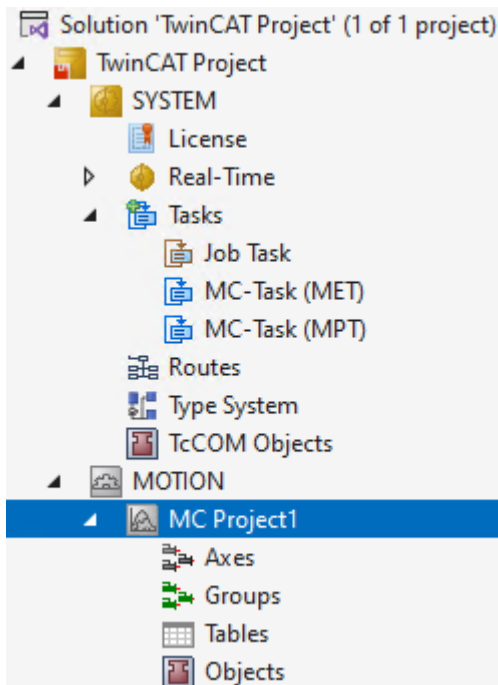
1. Machen Sie im Projektbaum einen Rechtsklick auf den Knoten **MOTION** > **Add New Item...**



- ⇒ Der **Insert Motion Configuration** Dialog öffnet sich.
2. Wählen Sie **MC Configuration** aus.
 3. Vergeben Sie einen Projektnamen.
 4. Bestätigen Sie Ihre Eingaben über den Button **Ok**.



- ⇒ Ein MC-Projekt mit dem gewählten Namen wird angelegt.

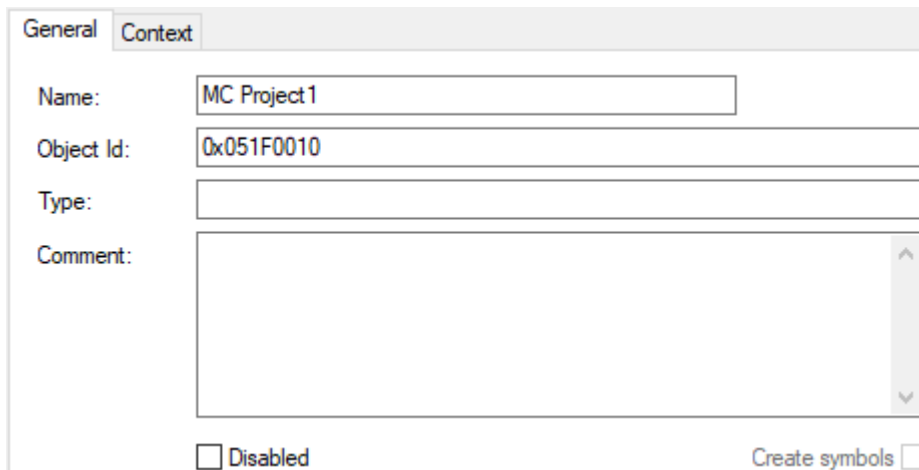


i Option „Create new tasks“ ausgewählt

Für das MC-Projekt wird eine neue MET- und eine MPT-Task angelegt.

- Ist kein neuer Satz von Tasks gewünscht, wählen sie die Option ab. Für das neue MC-Projekt werden die zuerst angelegten Tasks verwendet.

3.2.1.2 Registerkarte General



General Context

Name: MC Project1

Object Id: 0x051F0010

Type:

Comment:

☐ Disabled ☐ Create symbols

Eigenschaft	Beschreibung
Name	Gewählter Name des MC-Projekts. Vergeben Sie keinen Projektnamen doppelt.
Object Id	Automatisch vergebene, innerhalb eines TwinCAT-Projektes eindeutige 32-Bit-Identifikationsnummer.
Type	
Comment	Frei editierbares Feld für eigene Notizen.
Disabled	Option zum Deaktivieren eines MC-Projekts.

3.2.1.3 Registerkarte Context

Bei der TwinCAT MC3 kann die Ausführung der Achsen unterschiedlichen Task-Kontexten zugeordnet werden. Durch die Kontexte ist es möglich, die Ausführung der Achsen auf mehrere CPU-Kerne zu verteilen und unterschiedliche Zykluszeiten für verschiedene Achsen, auch auf einem Kern, vorzugeben. Im MC-Projekt können die Einstellungen in der Registerkarte **Kontext** vorgenommen werden.

Ein Task-Kontext setzt sich immer aus drei Tasks zusammen:

- MET – Motion Execution Task
- MPT – Motion Preparation Task
- Job Task

Die MET- und die MPT-Task eines Kontexts müssen immer demselben Kern zugeordnet sein und beide Tasks dürfen nur in einem Kontext verwendet werden. Im Gegenzug darf eine Job Task mehreren Kontexten zugeordnet werden.

Die Einstellung der Task-Zykluszeiten erfolgt über dem jeweiligen Task-Objekt unter SYSTEM > Tasks [► 13]. Unter SYSTEM > Real-Time [► 12] erfolgt die CPU-Kern Zuweisung.

General Context

Name	Type	Task	Priority	Cycle Ti...	Task Port	RT-CPU	Sort Order	Scheduling
Context1	MET 0	02010030 (MC-Task (MET))	3	2000	521	CPU 0	0 (default=100)	Stable (...)
	MPT 0	02010040 (MC-Task (MPT))	8	10000	531	CPU 0	0 (default=100)	
	MJT 0	02010020 (Job Task)	32			CPU 2	0 (default=100)	
Context2	MET 1	02010050 (MC-Task (MET)_1)	4	1000	522	CPU 1	0 (default=100)	Stable (...)
	MPT 1	02010060 (MC-Task (MPT)_1)	9	5000	532	CPU 1	0 (default=100)	
	MJT 1	02010020 (Job Task)	32			CPU 2	0 (default=100)	

Add Remove ☐ Show Contexts in Tree

Entity	Context
Axis1 (Axis)	Context1
Axis2 (Axis)	Context1
Axis3 (Axis)	Context1
Axis4 (Axis)	Context2
Axis5 (Axis)	Context2

Alternativ kann der Task-Kontext einer Achse unter der Registerkarte **Settings** der Achse eingestellt werden.

Scheduling

Eine MC3-Achse ist ein TcCOM Objekt. Die Grundlagen zur Abarbeitung von TcCOM-Echtzeitmodulen werden in der Dokumentation „Grundlagen“ am Ende des Kapitels [Echtzeit > Unterschiede in der Abarbeitung zwischen SPS- zu „TcCom“-Laufzeitmodulen](#) erläutert. Aufbauend darauf, kann mit den nachfolgenden Einstellmöglichkeiten die interne Ausführungssequenz der Kommandoverarbeitung und Sollwertgenerierung beeinflusst werden.

Stable (Output – Commands – Exec)

Diese Einstellung entspricht dem „IO am Taskbeginn“. Die Kommandoverarbeitung und Sollwertgenerierung werden im „Post-Zyklusupdate“ und damit nach dem „Ausgangs-Update“ ausgeführt. Diese Einstellung ist für einen verringerten Jitter des „Ausgangs-Updates“ zu wählen. Berechnete Sollwerte werden im nächsten Zyklus in das Ausgangsprozessabbild geschrieben.

Fast Reaction (Commands - Exec - Output)

Bei dieser Einstellung werden die Kommandos und die Sollwerte im „Zyklus-Update“ zwischen dem „Eingangs-Update“ und dem „Ausgangs-Update“ verarbeitet bzw. berechnet. Zu beachten ist dabei ein höherer Jitter des „Ausgangs-Update“ gegenüber der Option „Stable“. Berechnete Sollwerte werden direkt im Anschluss an ihre Berechnung (im selben Zyklus) in das Ausgangsprozessabbild geschrieben.



Scheduling bei Verwendung der NC-Task SAF

Bitte beachten Sie, dass die Einstellung für Instanzen, die die NC-Task SAF verwenden, keine Auswirkung hat. In diesem Fall wird Fast Reaction (Commands – Exec – Output) unabhängig von der Parametrisierung verwendet.

3.2.1.3.1 Einen neuen Kontext erstellen

- ✓ Ein TwinCAT-Projekt mit MC-Projekt wurde erstellt, siehe [MC-Projekt erstellen](#) [► 17].
Ein Motion-Projekt kann aus 1 bis n Task-Kontexten bestehen. Auf nachfolgendem Weg kann einem Motion-Projekt ein zusätzlicher Kontext hinzugefügt werden.

1. Klicken Sie im Context-Tab [► 19] des Motion Projektes auf **Add**.

Name	Type	Task	Priority	Cycle Ti...	Task Port	RT-CPU	Sort Order	Scheduling
Context1	MET 0	02010030 (MC-Task (MET))	3	2000	521	CPU 0	0 (default=100)	Stable (...)
	MPT 0	02010040 (MC-Task (MPT))	8	10000	531	CPU 0	0 (default=100)	
	MJT 0	02010020 (Job Task)	32			CPU 2	0 (default=100)	

☐ Show Contexts in Tree

⇒ Es folgt die Frage, ob für den neuen Kontext Tasks erstellt werden sollen. Jeder Kontext benötigt seine eigene MET- und MPT-Task.

2. Wenn im Vorfeld noch keine Tasks für den neuen Kontext angelegt wurden, klicken sie auf den Button **Yes**, um von TwinCAT die Tasks anlegen zu lassen.

TcXaeShell

?

Create new tasks?

⇒ Ein neuer Kontext wird angelegt.

⇒ Wenn neue Tasks erstellt wurden, so sind diese direkt dem Kontext zugewiesen.

Name	Type	Task	Priority	Cycle Ti...	Task Port	RT-CPU	Sort Order	Scheduling
Context1	MET 0	02010030 (MC-Task (MET))	3	2000	521	CPU 0	0 (default=100)	Stable (...)
	MPT 0	02010040 (MC-Task (MPT))	8	10000	531	CPU 0	0 (default=100)	
	MJT 0	02010020 (Job Task)	32			CPU 2	0 (default=100)	
Context2	MET 1	02010070 (MC-Task (MET)_2)	5	2000	523	Default...	0 (default=100)	Stable (...)
	MPT 1	02010080 (MC-Task (MPT)_2)	10	10000	533	Default...	0 (default=100)	
	MJT 1	02010020 (Job Task)	32			CPU 2	0 (default=100)	

☐ Show Contexts in Tree

3. Ordnen Sie dem neuen Kontext entsprechende Tasks zu, falls noch keine Tasks erstellt worden sind.
4. Entsprechend des Einsatzzweck des Kontextes sind die Tasks zu konfigurieren und dem entsprechenden CPU-Kern zuzuordnen.
5. Zudem sind dem Kontext noch Achsen zuzuordnen, siehe Achsen einem Kontext (Context) zuordnen [► 21].

3.2.1.3.2 Achsen einem Kontext (Context) zuordnen

Voraussetzungen

- Ein zusätzlicher Kontext wurde zum MC-Projekt hinzugefügt, siehe Neuen Context erstellen [► 20].
- Es befindet sich mind. eine Achse im MC-Projekt, siehe Achsobjekt erstellen [► 25].

- Jede Achse muss einem Task-Kontext zugeordnet sein. Standardmäßig beinhaltet jedes Motion-Projekt einen Kontext und alle Achsen sind diesem ersten Kontext zugeordnet.
- Befindet sich mehr als ein Kontext im Motion Projekt, so können Achsen dieses Motion-Projekts explizit verschiedenen Kontexten zugeordnet werden. Diese Einstellung ist für einzelne als auch für mehrere Achsen gleichzeitig möglich.

General Context

Name	Type	Task	Priority	Cycle Ti...	Task Port	RT-CPU	Sort Order	Scheduling
Context1	MET 0	02010030 (MC-Task (MET))	3	2000	521	CPU 0	0 (default=100)	Stable (...)
	MPT 0	02010040 (MC-Task (MPT))	8	10000	531	CPU 0	0 (default=100)	
	MJT 0	02010020 (Job Task)	32			CPU 2	0 (default=100)	
Context2	MET 1	02010050 (MC-Task (MET)_1)	4	1000	522	CPU 1	0 (default=100)	Stable (...)
	MPT 1	02010060 (MC-Task (MPT)_1)	9	5000	532	CPU 1	0 (default=100)	
	MJT 1	02010020 (Job Task)	32			CPU 2	0 (default=100)	

Add Remove ☐ Show Contexts in Tree

Entity	Context
Axis1 (Axis)	Context1
Axis2 (Axis)	Context1
Axis3 (Axis)	Context1
Axis4 (Axis)	Context2
Axis5 (Axis)	Context2

Variante 1: Registerkarte Context - Mehrfachauswahl und Rechtsklick

1. Markieren Sie in der Achsliste der Registerkarte **Context** die Achsen, die einem anderen Kontext zugeordnet werden sollen.
2. Machen Sie einen Rechtsklick auf die Auswahl.
⇒ Ein Kontextmenü mit den verfügbaren Kontexten erscheint.

Entity	Context
Axis1 (Axis)	Context1
Axis2 (Axis)	Context1
Axis3 (Axis)	Context1
Axis4 (Axis)	Context1
Axis5 (Axis)	Context1
	Context2

3. Wählen Sie den gewünschten Kontext aus.
⇒ Alle markierten Achsen wurden dem ausgewählten Kontext zugeordnet.

Variante 2: Registerkarte Context - Einzeln auswählen über Combobox

1. Machen Sie in der Achsliste der Registerkarte **Context** einen Doppelklick auf den Kontext der Achse, die geändert werden soll.

⇒ Die Combobox öffnet sich mit einer Auswahl der verfügbaren Kontexte.

Entity	Context
Axis1 (Axis)	Context1
Axis2 (Axis)	Context1
Axis3 (Axis)	Context1
Axis4 (Axis)	Context1
Axis5 (Axis)	Context1
	Context2

2. Wählen Sie den gewünschten Kontext aus.

⇒ Die eine Achse wurde dem ausgewählten Kontext zugeordnet. Sollen mehrere Achsen dem Kontext zugeordnet werden, so ist der Vorgang für alle Achsen zu wiederholen.

Weitere Varianten:

- Über die [Achseinstellungen](#) [► 28].

3.2.2 Axes

Unter **Axes** wird ein Überblick der im MC-Projekt enthaltenen Achsen angezeigt. Es ist möglich, hierüber Achsfreigaben zu setzen und Achsen zu joggen.

Wenn ebenfalls Mover im Projekt enthalten sind, kann über den Tab „Mover“ eine Übersicht aller Mover eingesehen werden. Weitere Informationen darüber sind in der XPlanar Dokumentation hinterlegt.

Es besteht außerdem die Möglichkeit, direkt auf dieser Seite die Achsobjekte mit der PLC und der Hardware (I/O) zu verknüpfen. Alternativ kann die Verknüpfung an den einzelnen Achsen unter [Settings](#) [► 28] erfolgen.

Axes								
Filter axes by name...								
						Link I/O	Link PLC	
Name	Enabled	State	I/O	PLC	Act. Position	Set. Position	Set. Velocity	Error
Axis1 (Axis)	E P N	Disabled		⇒	0.00	0.00	0.00	0.00
Axis2 (Axis)	E P N	Disabled			0.00	0.00	0.00	0.00
Axis3 (Axis)	E P N	Disabled			0.00	0.00	0.00	0.00
Axis4 (Axis)	E P N	Disabled			0.00	0.00	0.00	0.00

> Basic Controls

Enable
 Disable

 Jogging

◀ ◀ ◀ ▶ ▶ ▶ ▶ ▶

 ◀ Negative Positive ▶

Reset

Halt

Bedienung Axes UI

Um Achsen aus dieser Ansicht heraus freizugeben oder zu bewegen, sind die gewünschten Achsen auszuwählen/anzuklicken. Anschließend können über die Side Bar die Achsen angesteuert und bedient werden.

Die Side Bar kann über den Pfeil an der oberen linken Ecke minimiert/eingeklappt sowie auch wieder auf ihre Normalgröße erweitert werden.

Load Links

Über die Schaltfläche **Load Links** werden die aktuellen I/O- und PLC-Verlinkungen abgerufen und in der Tabelle angezeigt.

Ist ein Objekt verlinkt, wird das grüne Symbol  sichtbar.

Link I/O und Link PLC – Linking Manager

Bei einem Klick auf **Link I/O** oder **Link PLC** öffnet sich der Verlinkungsmanager. Mit **Link PLC** können Achs-Objekte mit den Instanzen einer AXIS_REF zur PLC verknüpft (Link) bzw. voneinander entkoppelt werden (Unlink). Über **Link I/O** können Verknüpfungen der Achs-Objekte mit Hardwarekomponenten vorgenommen werden. Die Achsen werden entsprechend der Reihenfolge der Auswahl verknüpft.

Beispiel: in der System-Konfiguration (linke Tabelle) Auswahl Axis2, Axis4, Axis3 und in der PLC-Konfiguration (rechte Tabelle) MAIN.axis[4], MAIN.axis[2], MAIN.axis[3] ergibt Axis2 + MAIN.axis[4], Axis4 + MAIN.axis[2], Axis3 + MAIN.axis[3].

Linking Manager

Manage the Item Links

Filter Axes by name...



All



Unused

Filter Links by name...



All



Unused

Name ↑↓	Current Link	New Link
Axis1 (Axis)		MAIN.axis[1] (Untitled1 Instance)
Axis2 (Axis)		
Axis3 (Axis)		
Axis4 (Axis)		

Link Name ↑↓
MAIN.axis[1] (Untitled1 Instance)
MAIN.axis[2] (Untitled1 Instance)
MAIN.axis[3] (Untitled1 Instance)
MAIN.axis[4] (Untitled1 Instance)

Link

Unlink

Link Changes

Name	New Link
Axis1 (Axis)	MAIN.axis[1] (Untitled1 Instance)

Accept

Cancel

Settings

Unter „Settings“ lassen sich verschiedene Einstellungen für die Motion UI treffen. Über **Save** werden die getätigten Einstellungen projektintern gespeichert.

Settings

Common Storage Polling State

Number Of Digits	2
XAR Interval [ms]	100
XAE Interval [ms]	500
Feedback Duration [s]	3
Severity of Feedback	Warning
Error Code Format	Hex
Show Full Name	☒

Save

Common

Auswahl	Beschreibung
Number Of Digits	Anzahl an anzuzeigenden Nachkommastellen.
XAR Interval [ms]	Legt das Intervall fest, mit dem Daten aus der Laufzeitumgebung ausgelesen werden.
XAE Interval [ms]	Legt das Intervall fest, mit dem Daten aus der Engineering-Umgebung ausgelesen werden.
Feedback Duration [s]	Legt die Zeitdauer fest, für die das Feedback-Fenster (Pop-Up) eingeblendet wird.
Severity of Feedback	Legt fest, ab welchem Level Ereignisse im Feedback-Fenster angezeigt werden.
Error Code Format	Optionen: Hex, Decimal Legt das Format fest, in dem der Fehlercode angezeigt wird (als Hexadezimal- oder Dezimalzahl).
Show Full Name	Vergrößert bei Bedarf die Anzeige des Achsnamen, sodass dieser vollständig angezeigt wird.

Storage

Auswahl	Beschreibung
Layouts and Inputs	Ein Klick auf Clear setzt das Layout auf die Standardansicht zurück und löscht getätigte Eingaben in der UI.
Settings	Ein Klick auf Clear löscht getätigte Einstellungen für diese Oberfläche.

Polling State

Der Polling State gibt zu Diagnosezwecken Daten von der letzten Anfrage der UI an den Motion Treiber zurück.

Auswahl	Beschreibung
Polling State	Idle / Polling / No Connection / Error
Error Message	Fehlermeldung, falls es in der Kommunikation zwischen Motion UI und dem Treiber zu einem Fehler kommt.
Last Fetch Duration	Gesamtdauer der letzten Anfrage der UI an den Motion Treiber.
Last Bundled Response	Letzte Antwort aus dem Treiber an die UI.

Vergrößern/Verkleinern der Ansicht

Die Ansicht lässt sich vergrößern und verkleinern, indem die **[STRG]**-Taste auf der Tastatur gedrückt wird und gleichzeitig das Mausrad bewegt wird.

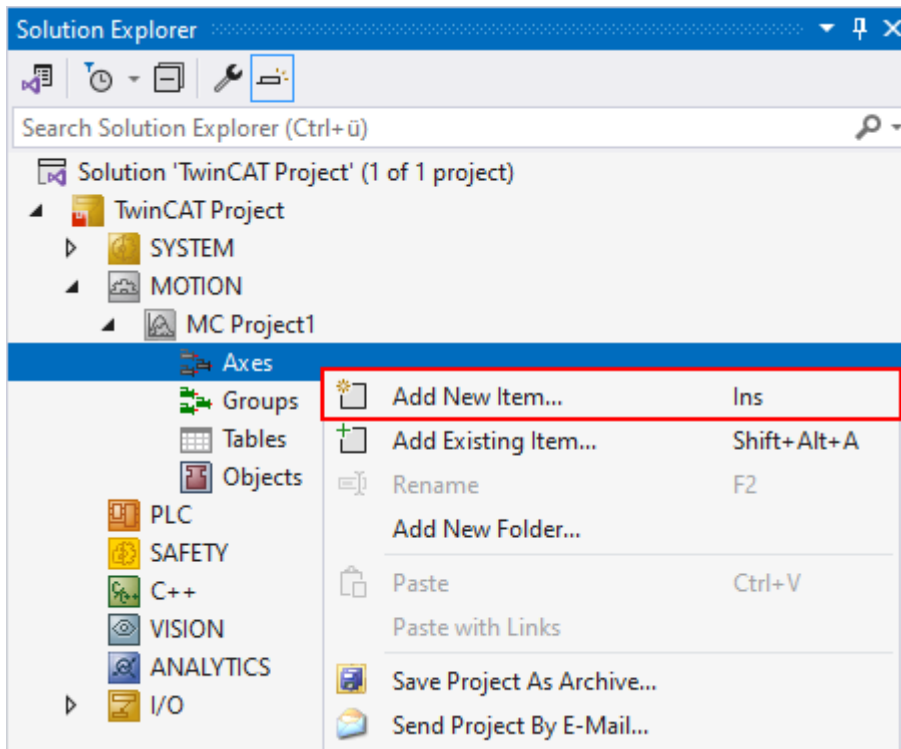
3.2.3 Default Axis

Im folgenden Abschnitt sind Informationen über das Standard-Achsojekt gesammelt. Es werden die verschiedenen Einstellmöglichkeiten und mögliche Erweiterungs-Module vorgestellt.

3.2.3.1 Achsojekt erstellen

- ✓ Ein TwinCAT-Projekt mit MC-Projekt wurde erstellt, siehe [MC-Projekt erstellen](#) [► 17].

1. Machen Sie einen Rechtsklick auf den Knoten im Strukturbaum **Axes** > **Add New Item...**



2. Wählen Sie im Dialog **Insert TcCom Object** in der Auflistung **Type** den Achstyp aus und vergeben Sie unter **Multiple** die gewünschte Anzahl der Achsen.

Axis

Standardachse

Encoder Axis

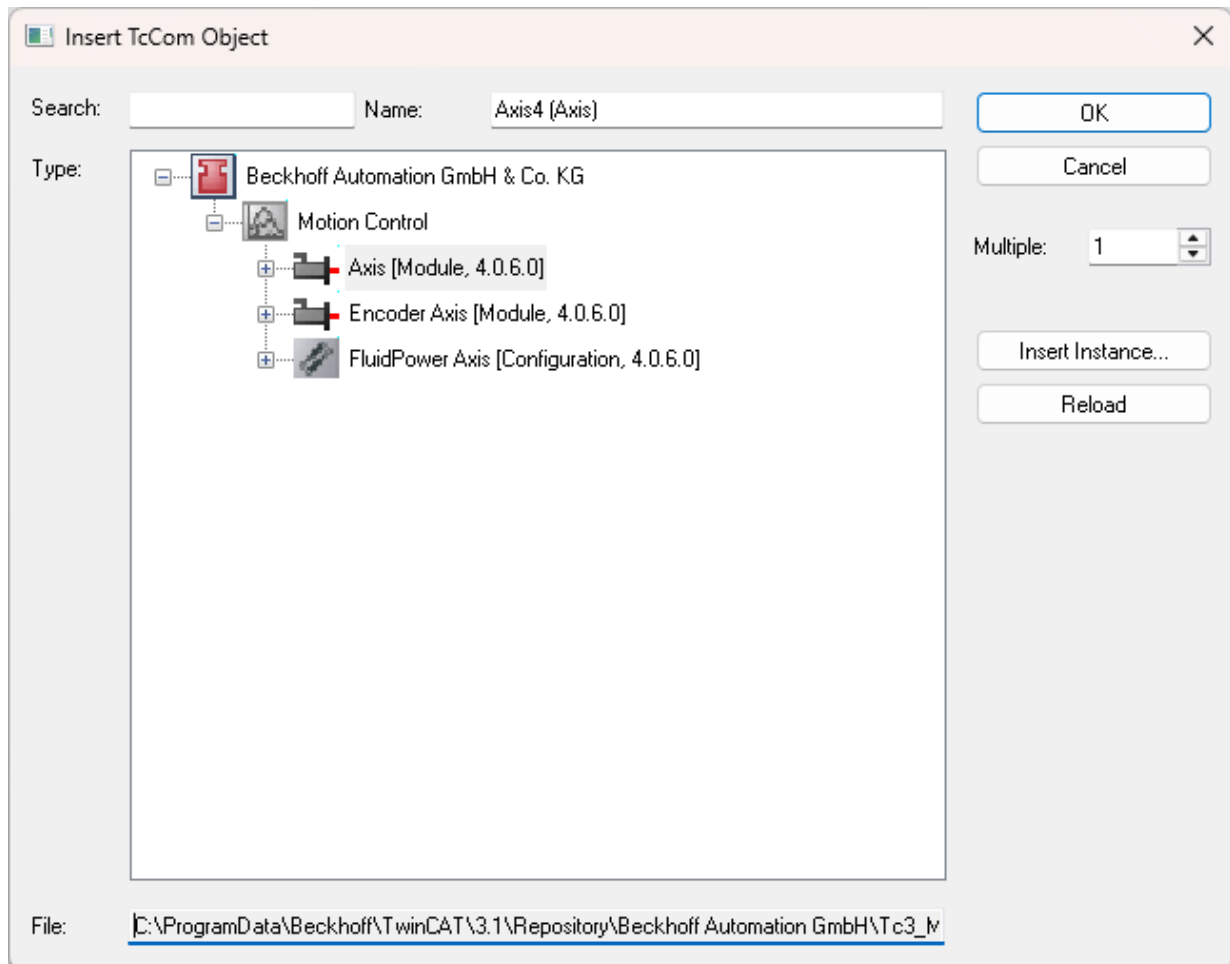
Encoder-Achse

FluidPower Axis

Spezielle Konfiguration der Standardachse und erweitert um zusätzlich Module für die Verwendung von FluidPower.

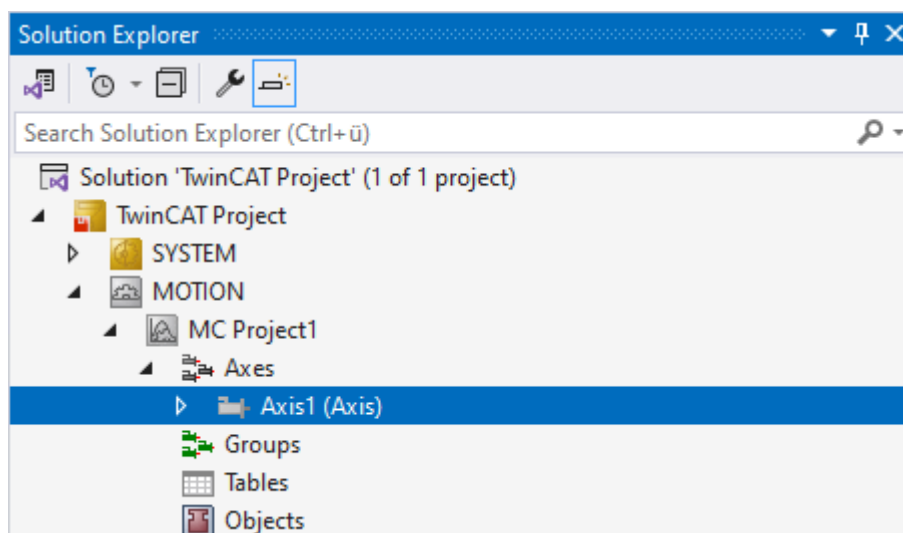
Planar Mover

Zur Verwendung mit Beckhoff XPlanar Mover



3. Bestätigen Sie beides über den Button **OK**.

- ⇒ Die Achse / Der Mover wurde angelegt und befindet sich unterhalb des Axes-Knoten. Bis zur Auswahl des Achstypen handelt es sich automatisch um eine Simulationsachse  bzw. bis zur Verknüpfung mit Hardware um einen Simulationsmover. Der Achstyp kann über die Registerkarte Settings [► 28] des Achsobjektes angepasst werden.



3.2.3.2 Registerkarte Object

Für die Ansicht der Object-Registerkarte gelten die gleichen Eigenschaften wie im Instanzknoten der PLC.

Object	Settings	Init Parameter	Online Parameter	Data Area	Motion UI
Object Id:	0x05110020	☐ Copy TMI to Target			
Object Name:	Axis2 (Axis)	☒ Share TMC Description			
Type Name:	Axis	☐ Auto Reload TMI/TMC			
GUID:	1437754E-F772-4FD4-AD2D-3E16CDF29D72				
Class Id:	050300B8-0000-0000-F000-000000000064				
Class Factory:	Tc3_Mc3Ptp_Core (Beckhoff Automation GmbH Tc3_Mc3Ptp_Core				
TMI/TMC	C:\ProgramData\Beckhoff\TwinCAT\3.1\Repository\Beckhoff Automation GmbH\Tc3_Mc3Ptp_Core\4.0.6.0\Tc3_Mc3Ptp_Core.tmc				
Parent Id:	0x05100010	☒ Keep Unrestored Link Info			
Init Sequence:	PSO				

3.2.3.3 Registerkarte Settings

In der Registerkarte Settings können wesentliche Einstellungen zur Achse vorgenommen werden.

Object	Settings	Init Parameter	Online Parameter	Data Area	Motion UI
Link To I/O...		Drive 2 (AX8108-0000-0107)			
Link To PLC...		MAIN.axis (Untitled1 Instance)			
Axis Type:	CoE DS402 Drive				
Add. Encoder Type:	(none)				
Task Context:	Context1				
☐ Simulation					

Eigenschaft	Beschreibung
Link To I/O...	Der Link-Button öffnet einen Dialog zum Verknüpfen der MC-Achse mit der Antriebshardware unter dem I/O-Knoten. Im Feld rechts daneben wird das verknüpfte Element angezeigt.
Link To PLC...	Der Link-Button öffnet einen Dialog zum Verknüpfen der MC-Achse mit einer PLC-Instanz der AXIS_REF. Im Feld rechts daneben wird das verknüpfte Element angezeigt.
Axis Type	Gibt das Protokoll zur Kommunikation mit der verknüpften Antriebshardware an. Details zu den Achstypen sind unter <u>Drive-Module</u> [► 42] zu finden.
Add. Encoder Type	Option, einen zusätzlichen Encoder zur Achse hinzuzufügen. Details zu den Encodern sind unter <u>Encoder-Module</u> [► 82] zu finden. Beispiel: Zu verwenden, wenn ein Analog Drive ohne eigenes Gebersystem verknüpft wird.
Task Context	Jede Achse muss einem Task-Kontext zugeordnet sein. Besitzt ein <u>MC Project</u> [► 17] mehr als einen Kontext, so kann über die Combobox der Kontext der Achse verändert werden. Das Erstellen und Konfigurieren neuer Kontexte erfolgt über die <u>Registerkarte Context</u> [► 19] des MC Projects.
Simulation	☒  Bei aktivem Simulation Mode wird der Link zum I/O ignoriert und, anstelle des konfigurierten Axis Type, ein Simulationsobjekt (Drive und Encoder) verwendet.

Eigenschaft	Beschreibung		
	☐		Ist der Axis Type (none) ausgewählt, handelt es sich immer um eine Simulationsachse und die Einstellung kann nicht verändert werden.
	☐		Der konfigurierte Axis Type und Link zum I/O wird verwendet.

3.2.3.4 Registerkarte Init Parameter

General

Name	Standardwert	Einheit	Typ
Simulation Mode	TRUE		BOOL
Position Unit	mm	[PosUnit]	MC.EPositionUnit [► 207]
Is Modulo Axis	FALSE		BOOL
Position Modulus	360.0	[PosUnit]	LREAL
Position Modulo Tolerance Window	0.0	[PosUnit]	LREAL

Simulation Mode

Der Simulationsmodus zeigt an, ob die I/O-Verbindung zum Antrieb und Encoder aktiv (**FALSE**) oder simuliert (**TRUE**) sein soll. Der Parameter kann während der Laufzeit geschrieben werden, wenn die Achse ohne Reglerfreigabe ist.

Position Unit

Die Position Unit wird für Anzeigezwecke, z. B. auf der Inbetriebnahmeoberfläche, verwendet. Ein- und Ausgaben an Funktionsbausteinen werden nicht konvertiert.

Is Modulo Axis

Aktivieren Sie diesen Parameter, um für diese Achse Modulo-Funktionalitäten an Funktionsbausteinen zu nutzen. Sollwert und Istwert werden in der zyklischen Schnittstelle als Absolut- und Modulo-Wert berechnet. Um Modulo-Informationen z. B. mit der PLC auszutauschen, ist die Data Area Modulo [► 33] zu aktivieren.

- **Position Modulus**
Modulo-Wert zur Berechnung der Modulo-Position
- **Position Modulo Tolerance Window**
Eine Bewegung entgegen der angegebenen Richtung ist zulässig, sofern die Zielposition innerhalb dieser Positionstoleranz der Startposition des Kommandos liegt. Dieser Parameter hat einen Einfluss auf das Verhalten des MC_MoveModulo [► 278]. Er steht in Wechselwirkung mit der gewählten Einstellung am Eingang Direction des Funktionsbausteins.

Maximum Dynamics

Name	Standardwert	Einheit	Typ
Maximum Velocity	2000.0	[PosUnit]/s	LREAL
Maximum Acceleration	10000.0	[PosUnit]/s ²	LREAL
Maximum Deceleration	10000.0	[PosUnit]/s ²	LREAL
Maximum Jerk	100000.0	[PosUnit]/s ³	LREAL

Maximum Velocity

Maximale Geschwindigkeit, die bei der Profilerstellung berücksichtigt wird. Wird die Konstante #Maximum am Eingang Velocity eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Geschwindigkeit als Grenzwert verwendet.

Maximum Acceleration

Maximale Beschleunigung, die bei der Profilerstellung berücksichtigt wird. Wird die Konstante #Maximum am Eingang Acceleration eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Beschleunigung als Grenzwert verwendet.

Maximum Deceleration

Maximale Verzögerung, die bei der Profilerstellung berücksichtigt wird. Wird die Konstante #Maximum am Eingang Deceleration eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Verzögerung als Grenzwert verwendet.

Maximum Jerk

Maximaler Jerk, der bei der Profilerstellung berücksichtigt wird. Wird die Konstante #Maximum am Eingang Jerk eines Funktionsbausteins angegeben, wird der hier definierte Wert für den Jerk als Grenzwert verwendet.

Default Dynamics

Name	Standardwert	Einheit	Typ
Default Acceleration	2000.0	[PosUnit]/s ²	LREAL
Default Deceleration	2000.0	[PosUnit]/s ²	LREAL
Default Jerk	4000.0	[PosUnit]/s ³	LREAL

Default Acceleration

Dieser Wert wird als Beschleunigung am Eingang Acceleration an Funktionsbausteinen verwendet, sofern er nicht explizit gesetzt ist. Wird die Konstante #Default am Eingang eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Beschleunigung als Grenzwert verwendet.

Default Deceleration

Dieser Wert wird als Verzögerung am Eingang Deceleration an Funktionsbausteinen verwendet, sofern er nicht explizit gesetzt ist. Wird die Konstante #Default am Eingang eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Verzögerung als Grenzwert verwendet.

Default Jerk

Dieser Wert wird als Jerk am Eingang Jerk an Funktionsbausteinen verwendet, sofern er nicht explizit gesetzt ist. Wird die Konstante #Default am Eingang eines Funktionsbausteins angegeben, wird der hier definierte Wert für den Jerk als Grenzwert verwendet.

Manual Motion

Name	Standardwert	Einheit	Typ
Jog Velocity Slow	100.0	[PosUnit]/s	LREAL
Jog Velocity Fast	250.0	[PosUnit]/s	LREAL

Jog Velocity Slow

Diese Zielgeschwindigkeit wird für das langsame Verfahren einer Achse mit Handbedientasten verwendet. Das Verfahren ist mithilfe von MC_Jog sowie über die Inbetriebnahmeoberfläche möglich.

Jog Velocity Fast

Diese Zielgeschwindigkeit wird für das schnelle Verfahren einer Achse mit Handbedientasten verwendet. Das Verfahren ist mithilfe von MC_Jog sowie über die Inbetriebnahmeoberfläche möglich.

Monitoring

Name	Standardwert	Einheit	Typ
Position Limit Enforcement Enabled	FALSE		BOOL
Minimum Position	#Ignore	[PosUnit]	LREAL

Name	Standardwert	Einheit	Typ
Maximum Position	#Ignore	[PosUnit]	LREAL
Position Lag Monitoring Enabled	TRUE		BOOL
Maximum Position Lag Value	5.0	[PosUnit]	LREAL
Maximum Position Lag Filter Time	0.02	s	LREAL
Position Lag Calculation	HardwareProvidedIfAvailable		MC.EPositionLagSource [► 206]

Position Limit Enforcement Enabled

Dieser Parameter gibt an, ob Positionsgrenzen bei der Generierung von Bewegungsprofilen berücksichtigt werden. Wenn der Wert `TRUE` ist, sind die folgenden beiden Parameter einzustellen.

- **Minimum Position**
Minimal zulässige Position für die Achse. Verwenden Sie `#Ignore`, um die Einhaltung der Positionsgrenze in diese Bewegungsrichtung auszuschalten. Es ist nicht möglich, eine Achse außerhalb dieses Limits zu verfahren. Steht eine Achse beim TwinCAT Start außerhalb ihrer Positionslimits, ist eine Bewegung in Richtung des gültigen Positionsbereichs möglich. Kommandos mit einer Zielposition außerhalb dieser Untergrenze werden abgelehnt.
- **Maximum Position**
Maximal zulässige Position für die Achse. Verwenden Sie `#Ignore`, um die Einhaltung der Positionsgrenze in diese Bewegungsrichtung auszuschalten. Es ist nicht möglich, die Achse außerhalb dieses Limits zu verfahren. Steht eine Achse beim TwinCAT Start außerhalb ihrer Positionslimits, ist eine Bewegung in Richtung des gültigen Positionsbereichs möglich. Kommandos mit einer Zielposition außerhalb dieser Untergrenze werden abgelehnt.

Position Lag Monitoring Enabled

Dieser Parameter gibt an, ob der Schleppfehler für die Achse überwacht werden soll. Wenn der Wert `TRUE` ist, werden die folgenden beiden Parameter angezeigt, um Position und Zeit zu parametrieren. Falls die parametrierten Grenzen überschritten werden, wird ein Laufzeitfehler ausgegeben.

Die Berechnung erfolgt wie folgt:

Positions-Schleppfehler = Aktuelle Sollposition - Istposition

- **Maximum Position Lag Value und Maximum Position Lag Filter Time**
`Maximum Position Lag Value` ist der Positions-Schleppfehler, welcher nicht länger als die festgelegte Zeit (`Maximum Position Lag Filter Time`) überschritten werden darf. Andernfalls wird die Achse durch direktes Abschalten unverzüglich gestoppt und ein Fehler ausgegeben.

Position Lag Calculation

Dieser Parameter gibt an, ob der Schleppfehler softwareseitig von der MC-Achse berechnet (`InternallyCalculated`) oder direkt aus der verknüpften Hardware ausgelesen wird (`HardwareProvidedIfAvailable`). Ist `HardwareProvidedIfAvailable` eingestellt aber keine Hardware verfügbar, wird auf den intern berechneten Wert zurückgegriffen. Für eine Simulationsachse werden bei `HardwareProvidedIfAvailable` Daten vom Simulationsdrive abgegriffen.

Error Event Handling

Name	Standardwert	Einheit	Typ
Control Error Setpoint Mode	HardStop		MC.EControlErrorSetpoint Mode [► 201]

Wenn ein Fehler auf der Hardwareseite vorliegt (z. B. im Regelkreis) und die MC keine Kontrolle mehr über die Achse hat, können Sie hier das Verhalten der Sollwertgenerierung einstellen. `HardStop` erzeugt ein sofortiges hartes Anhalten der Achse im Fehlerfall.

Diagnostics

Name	Standardwert	Einheit	Typ
TraceLevel	tlWarning		TcTraceLevel

Legen Sie fest, welche Nachrichten an die TwinCAT-Fehlerliste gesendet werden sollen.

Additional Encoder

Name	Standardwert	Einheit	Typ
Main Position Encoder			OTCID_PositionEncoder
Load Control Type	None		MC.ELoadType ▶ 205

Main Position Encoder

Falls ein zusätzlicher Encoder vorhanden ist, wählen Sie hier die Objekt-ID aus.

Load Control Type

Stellen Sie hier die Art der Last ein, die geregelt werden soll.

3.2.3.5 Registerkarte Online Parameter

Es gibt die folgenden Online-Parameter:

Monitoring

Name	Einheit	Type
Minimum Position Lag	[PosUnit]	LREAL
Maximum Position Lag	[PosUnit]	LREAL
Trigger Minimum/Maximum Position Lag Reset		Trigger Minimum/Maximum Position Lag Reset

Minimum Position Lag

Dieser schreibgeschützte Online-Parameter zeigt den minimalen erkannten Schleppfehler zu Informationszwecken an.

Maximum Position Lag

Dieser schreibgeschützte Online-Parameter zeigt den maximalen erkannten Schleppfehler zu Informationszwecken an.

Trigger Minimum/Maximum Position Lag Reset

Eine steigende Flanke an diesem Parameter Eingang setzt gespeicherte minimale und maximale Schleppfehler zurück.

Additional Error Event Information

Name	Einheit	Type
Originating Error Event Sender Id		OTCID
Original Error Id		HRESULT

Originating Error Event Sender Id

Liegt ein Folgefehler z. B. durch eine Kopplung vor, zeigt dieser Parameter die Objekt-ID der Achse an, von der der Fehler ausging.

Original Error Id

Liegt ein Folgefehler z. B. durch eine Kopplung vor, zeigt dieser Online-Parameter die ursprüngliche Fehlernummer an.

Override

Name	Einheit	Type
Velocity Override Factor		LREAL

Velocity Override Factor

Hier wird der aktuelle Geschwindigkeits-Override der Achse angezeigt.

3.2.3.6 Registerkarte Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

HINWEIS**Nicht verknüpfte CDTs können bei Zugriff zu Floating Point Exception führen**

Vergleiche mit NaN-Werten führen zu einer Exception, welche einen Stopp der Laufzeit und einen möglichen Maschinenschaden nach sich zieht.

Nur wenn das `IsConnected = TRUE` ist, sind valide Daten zu erwarten. Andernfalls besteht die Gefahr eines inkonsistenten oder ungültigen Datensatzes, was zu unabsehbaren Folgen und Fehlverhalten führen kann. Rechenoperationen mit ungültigen Werten können zu Ausnahmefehlern in der Gleitkommaeinheit des Prozessors führen.

- Schalten Sie die Floating Point Exceptions aus, wenn NaN-Werte in der Applikation verwendet und verarbeitet werden sollen.
- Prüfen Sie vor einem Zugriff (z. B. aus der PLC) auf ein CDT das `IsConnected` des CDTs.
- Machen Sie sich mit dem Umgang von benannten Konstanten und NaN Werten vertraut.

PlcToMc (Input)

Enable (default)	Name	Typ	Beschreibung
☒	Std	POINTER TO <u>MC.CDT_PLCTOMC_AXIS_STD</u> [► 215]	Zyklische Standarddatenstruktur zur Freigabe der Achse und setzen des Geschwindigkeitsoverrides.

McToPlc (Output)

Enable (default)	Name	Typ	Beschreibung
☒	Std	POINTER TO <u>MC.CDT_MCTOPLC_AXIS_STD</u> [► 212]	Zyklische Standarddatenstruktur mit der Achs Oid und den wichtigsten Zustandsinformationen.
☒	Set	POINTER TO <u>MC.CDT_MCTOPLC_AXIS_SET</u> [► 211]	Optionale zyklische Datenstruktur mit Informationen zu den aktuellen Sollwerten der Achse. Hinweis Nur wenn <code>IsConnected = TRUE</code> ist, sind valide Daten zu erwarten. Eine Validierung gültiger Daten ist nach folgendem Kapitel sicherzustellen: Konstanten (Named Constants)
☒	Act	POINTER TO <u>MC.CDT_MCTOPLC_AXIS_ACT</u> [► 209]	Optionale zyklische Datenstruktur mit Informationen zu den aktuellen Istwerten der Achse.

Enable (default)	Name	Typ	Beschreibung
			Hinweis Nur wenn IsConnected = TRUE ist, sind valide Daten zu erwarten. Eine Validierung gültiger Daten ist nach folgendem Kapitel sicherzustellen: Konstanten (Named Constants)
☐	Modulo	POINTER TO <u>MC.CDT_MCTOPLC_AXIS_MODULO</u> [► 211]	Optionale zyklische Datenstruktur mit Informationen zur Moduloposition der Achse. Für eine sinnvolle Berechnung der Moduloposition ist in der <u>Achs Parameter (Init)</u> [► 29] der Parameter Is Modulo Axis auf TRUE zu setzen, sowie die Parameter Position Modulus und Position Modulo Tolerance Window entsprechend zu parametrieren. Hinweis Nur wenn IsConnected = TRUE ist, sind valide Daten zu erwarten. Eine Validierung gültiger Daten ist nach folgendem Kapitel sicherzustellen: Konstanten (Named Constants)
☐	LoadControl	POINTER TO <u>MC.CDT_MCTOPLC_AXIS_LOADCONTROL</u> [► 210]	Optionale zyklische Datenstruktur. Hinweis Nur wenn IsConnected = TRUE ist, sind valide Daten zu erwarten. Eine Validierung gültiger Daten ist nach folgendem Kapitel sicherzustellen: Konstanten (Named Constants)
☐	Fluidpower	POINTER TO <u>MC.CDT_MCTOPLC_AXIS_FLUIDPOWER</u> [► 210]	Optionale zyklische Datenstruktur. Hinweis Nur wenn IsConnected = TRUE ist, sind valide Daten zu erwarten. Eine Validierung gültiger Daten ist nach folgendem Kapitel sicherzustellen: Konstanten (Named Constants)

3.2.3.7 Registerkarte Motion UI

In der Registerkarte **Motion UI** ist zu Inbetriebnahme- und Testzwecken eine Inbetriebnahmeoberfläche (Motion UI) für die Achse. Diese stellt Basiskommandos sowie Informationen über die Achse und ihren Zustand bereit. Die Motion UI kann benutzerspezifisch angepasst werden.

In dieser Dokumentation wird die Motion UI in zwei Bereiche unterteilt:

1. Widget Bar mit Statusinformationen und Schnellzugriffen am oberen Rand
2. Karten im Hauptteil

3.2.3.7.1 Einstellmöglichkeiten

In der oberen rechten Ecke befinden sich Bedienfelder, über die die Oberfläche personalisiert und angepasst werden kann.

Icon	Bedeutung
	Über dieses Symbol lassen sich neue Karten dem Hauptteil hinzufügen.
	Über dieses Symbol wird das Menü mit erweiterten Einstellmöglichkeiten geöffnet.

Menü

Im Menü sind verschiedene Aktionen und Einstellmöglichkeiten verfügbar:

Auswahl	Beschreibung
Reconnect	Hierüber wird die Verbindung zwischen der UI und dem anzuzeigenden Objekt neu hergestellt und die Daten neu geladen.
Settings	Hierüber werden die Einstellungen geöffnet.
Open Widget Catalog	Öffnet den Widget Katalog für die Statusleiste.
Save Layout	Speichert das aktuelle Layout der UI. Es muss ein benutzerdefinierter Name vergeben werden. Das gespeicherte Layout ist nur im aktuellen Projekt nutzbar.
Apply Layout	Wendet ein vorhandenes Layout auf die aktuelle Ansicht an. Es sind standardmäßig die folgenden Layouts hinterlegt: • Default, • Modulo, • Fluid Power. Weitere Layouts, die über Save Layout gespeichert wurden, stehen hier ebenso zur Verfügung.
Layout Manager	Öffnet den Layout Manager.

Auswahl	Beschreibung
Reset Layout	Setzt die UI zurück, sodass wieder das Standard-Layout genutzt wird.
Theme	Es kann zwischen den folgenden Themen für die Farbgebung gewählt werden: • System, • Light, • Dark.

Settings

Unter „Settings“ lassen sich verschiedene Einstellungen für die Motion UI treffen. Über **Save** werden die getätigten Einstellungen projektintern gespeichert.

Settings ×

Common Axis UI Storage Polling State

Number Of Digits

XAR Interval [ms]

XAE Interval [ms]

Feedback Duration [s]

Severity of Feedback

Error Code Format

Show Full Name ☒

Save

Common

Auswahl	Beschreibung
Number Of Digits	Anzahl an anzuzeigenden Nachkommastellen.
XAR Interval [ms]	Legt das Intervall fest, mit dem Daten aus der Laufzeitumgebung ausgelesen werden.
XAE Interval [ms]	Legt das Intervall fest, mit dem Daten aus der Engineering-Umgebung ausgelesen werden.
Feedback Duration [s]	Legt die Zeitdauer fest, für die das Feedback-Fenster (Pop-Up) eingeblendet wird.
Severity of Feedback	Legt fest, ab welchem Level Ereignisse im Feedback-Fenster angezeigt werden.
Error Code Format	Optionen: Hex, Decimal Legt das Format fest, in dem der Fehlercode angezeigt wird (als Hexadezimal- oder Dezimalzahl).
Show Full Name	Vergrößert bei Bedarf die Anzeige des Achsnamen, sodass dieser vollständig angezeigt wird.

Axis UI

Auswahl	Beschreibung
Allow Override PLC	Standardmäßig ist diese Option deaktiviert. Dann können Achsfreigaben und Override in der UI nicht verändert werden, solange die Achse aus der PLC angesteuert wird. Wird diese Option aktiviert, kann der Benutzer die Achsfreigaben und den Override in der Benutzeroberfläche überschreiben.

Auswahl	Beschreibung
Allow Keyboard Control	Standardmäßig ist diese Option deaktiviert. In dem Fall können Achsen nicht über die Tastatur bewegt werden. Wird diese Option aktiviert, ist es möglich die Achsen über die Pfeiltasten zu joggen. Über die Leertaste wird ein Halt ausgeführt.

Storage

Auswahl	Beschreibung
Layouts and Inputs	Ein Klick auf Clear setzt das Layout auf die Standardansicht zurück und löscht getätigte Eingaben in der UI.
Settings	Ein Klick auf Clear löscht getätigte Einstellungen für diese Oberfläche.

Polling State

Der Polling State gibt zu Diagnosezwecken Daten von der letzten Anfrage der UI an den Motion Treiber zurück.

Auswahl	Beschreibung
Polling State	Idle / Polling / No Connection / Error
Error Message	Fehlermeldung, falls es in der Kommunikation zwischen Motion UI und dem Treiber zu einem Fehler kommt.
Last Fetch Duration	Gesamtdauer der letzten Anfrage der UI an den Motion Treiber.
Last Bundled Response	Letzte Antwort aus dem Treiber an die UI.

Layout Manager

Der Layout Manager zeigt die eigenen gespeicherten Layouts an und bietet weitere Optionen.

Layout Manager

Manage the UI layouts

Import Layout ↓

Save Layout 

Layout	Action
LayoutMachine1	   
LayoutMachine2	   

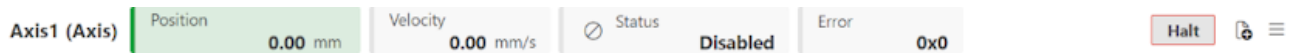
Auswahl	Beschreibung
	Importiert ein bestehendes Layout in das Projekt.
	Speichert das aktuelle Layout der UI projektintern. Es öffnet sich ein weiteres Fenster, um einen Namen zu vergeben. Wird ein bestehender Layout-Name verwendet, wird das bestehende Layout überschrieben.
 	Verändert die Reihenfolge der eigenen Layouts. Mittels der Pfeiltasten kann ein Layout in der Liste nach oben bzw. unten verschoben werden.
	Löscht das jeweilige selbst erstellte Layout.
	Exportiert das selbst erstellte Layout, sodass es auch in anderen Projekten verwendet werden kann. Im Anschluss muss der Speicherort vergeben werden.

Vergrößern/Verkleinern der Ansicht

Die Ansicht lässt sich vergrößern und verkleinern, indem die **[STRG]**-Taste auf der Tastatur gedrückt wird und gleichzeitig das Mausrad bewegt wird.

3.2.3.7.2 Widget Bar

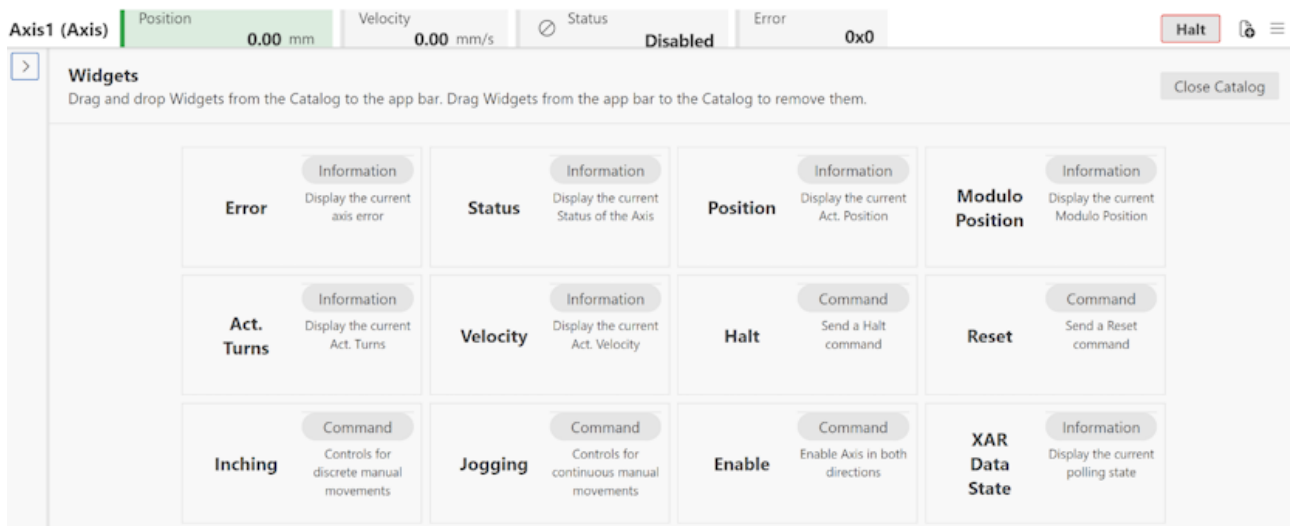
Am oberen Rand der Motion UI ist die Widget Bar platziert. Im Standard-Layout sind dort bereits einige Informationen wie beispielsweise der Achsname, die Achsposition und der Achsstatus enthalten. Außerdem befinden sich rechtsbündig der Halt-Button, um die Achse zu stoppen.



Die Leiste ist in eine linke und rechte Hälfte unterteilt. Ist die Seitenbreite erreicht, wird die linke Hälfte der Statusleiste automatisch um eine weitere Zeile erweitert.

Anpassen der Widget Bar

Es ist möglich, die Widget Bar zu personalisieren, beide Hälften können beliebig befüllt werden. Dazu ist im Menü der Punkt **Open Widget Catalog** auszuwählen. Per Drag-and-Drop können die Widgets der Leiste hinzugefügt sowie entfernt werden.



Die folgenden Widgets sind verfügbar:

Widget	Beschreibung
Error	Zeigt den aktuellen Achsfehler an.
Status	Zeigt den aktuellen Achszustand an.
Position	Zeigt die aktuelle Achsposition (Act Pos) an.
Modulo Position	Zeigt die aktuelle Achsposition im Modulo Bereich an.
Act. Turns	Zeigt die aktuellen Umdrehungen (modulo) an.
Velocity	Zeigt die aktuelle Achsgeschwindigkeit (Act Velo) an.
Halt	Über Halt kann die aktuelle Verfahrbewegung abgebrochen werden und die Achse wechselt in den Stillstand.
Reset	Liegt ein Achsfehler vor, kann dieser mittels Reset zurückgesetzt werden.
Inching	Ist dieses Control eingebunden, lassen sich darüber manuell diskrete (schrittweise) Bewegungen starten.
Jogging	Ist dieses Control eingebunden, lassen sich darüber manuell kontinuierliche Bewegungen starten (= Achse wird gejoggt).
Enable	Über Enable lässt sich die Achse in beide Richtungen freigeben.

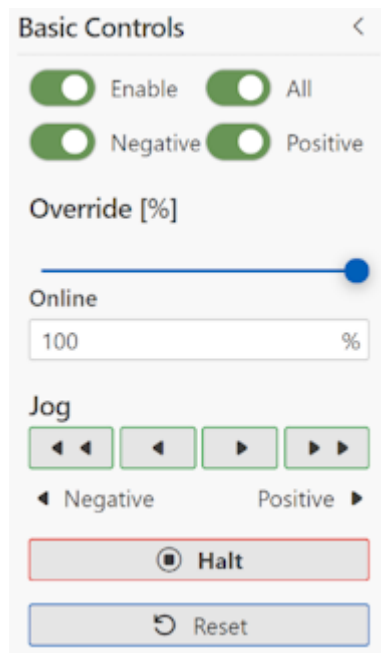
Widget	Beschreibung
XAR Data State	Zeigt den aktuellen Polling State an. Der Polling State gibt an, ob die Motion UI aktuelle Werte erhält (Polling) oder ob die Verbindung zum Treiber unterbrochen ist.

3.2.3.7.3 Side Bar

Am linken Rand der Motion UI ist die Side Bar platziert. Sie enthält die Basic Controls und kann durch

Anklicken des kleinen Pfeils  in der oberen rechten Ecke eingeklappt werden.

Die Side Bar ist nicht konfigurierbar und inhaltlich vorgegeben.



Manuelle Bewegung mittels Tastaturbedienung

Anstelle der Jog-Taster kann ebenfalls die Tastatur genutzt werden. Diese Option muss vom Nutzer zuvor in den Einstellungen aktiviert werden. Folgende Tasten sind den Aktionen zugeordnet:

Aktion	Optionale Tastaturbedienung
Langsames Verfahren per Handbedientasten in positive Richtung	→ (Pfeiltaste nach rechts)
Schnelles Verfahren per Handbedientasten in positive Richtung	↑ (Pfeiltaste nach oben)
Langsames Verfahren per Handbedientasten in negative Richtung	← (Pfeiltaste nach links)
Schnelles Verfahren per Handbedientasten in negative Richtung	↓ (Pfeiltaste nach unten)
Halt	Leertaste

3.2.3.7.4 Karten

Als Karten werden die verschiedenen Dialoge im Hauptteil der Motion UI bezeichnet. Im Standardlayout können hierüber die Achsfreigaben gesetzt sowie einfache Verfahrbewegungen ausgeführt werden. Außerdem sind beispielsweise Informationen über die aktuellen Dynamikwerte gelistet.

Motion Information ...

Cyclic Information

Name	Setpoint	Actpoint
Set/Act (3) ▾		
Position (mm)	99.00	99.00
Velocity (mm/s)	0.00	0.00
Acceleration (mm/s ²)	0.00	0.00
Lag (3) >		
Modulo (2) >		

Lag Information [mm]

Minimal	Maximal
0.00	0.00

Clear Lag

Verschieben und Anpassen der Karten

Die Position der Karten auf der Seite kann per Drag-and-Drop verändert werden. Dazu ist die Kopfzeile/ Titelzeile der Karte anzuklicken und mit gedrückter Maustaste die Position neu zu wählen.

Die Größe der jeweiligen Karte kann ebenfalls per Drag-and-Drop an der unteren rechten Karten-Ecke verändert werden.

Anpassen der Karteninhalte

Die Karten können beliebig benannt und befüllt werden und die Reihenfolge der Inhalte kann manuell angepasst werden. Um in den Bearbeitungsmodus der Karten zu wechseln, ist mittels der drei Punkte am



oberen rechten Rand in das Menü der Karte zu navigieren und **Edit Card** auszuwählen. Ist diese Option ausgewählt, wird der Text im Menü farblich hinterlegt. Die einzelnen Dialoge der Karte können



nun durch die eingeblendeten Pfeile verschoben werden. Außerdem können die einzelnen



Dialoge einfach aus der Karte entfernt werden über das Mülleimer-Symbol.

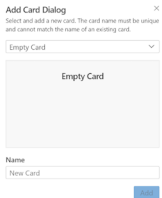


Mittels des im Bearbeitungsmodus sichtbaren Stiftsymbols neben dem Kartennamen kann dieser ebenfalls editiert werden. Um den Bearbeitungsmodus zu verlassen, ist im Menü erneut auf **Edit Card** zu klicken.

Hinzufügen weiterer Karten



Über können weitere Karten in die Motion UI hinzugefügt werden. Der Nutzer kann zwischen einem definierten Standard-Kartentyp oder einer leeren Karte wählen, die er anschließend selbst mit Inhalten füllen kann. Über die Pfeiltasten kann in der Auswahl zwischen den vordefinierten Kartentypen gewechselt werden.



Unter „Name“ kann der Karte vom Nutzer eine Überschrift gegeben werden.

3.2.3.7.5 Eine Achse über die Motion UI bewegen

⚠ GEFAHR

Dieses Beispiel setzt Simulationsachsen voraus – Tödliche bis schwere Verletzungsgefahr bei unsachgemäßer Verwendung von realen Achsen!

Dieses Beispiel erzeugt eine Achsbewegung. Bei der Verwendung von realen Achsen gelten andere Sicherheitsbedingungen. Es besteht eine tödliche bis schwere Verletzungsgefahr durch unbeabsichtigte Bewegung von realen Achsen.

- Achten Sie in jedem Fall darauf, dass weder Sie noch andere durch die Bewegung geschädigt werden könnten, z. B. durch die Einhaltung eines geeigneten Sicherheitsabstandes.
- Führen Sie keine Aktion aus, deren Folgen Sie nicht abschätzen können.

✓ Ein Achsobjekt wurde angelegt,
siehe [Achsobjekt erstellen](#) [► 25].

1. Aktivieren Sie das TwinCAT-Projekt.

2. Geben Sie die Achse über **All** im Enable Dialog frei.

⇒ Die Schieberegler im Enable Dialog sollten nun grün hinterlegt sein. Außerdem sollte der Achsstatus in der Widget Bar grün hinterlegt sein und den Status `Standstill` anzeigen.

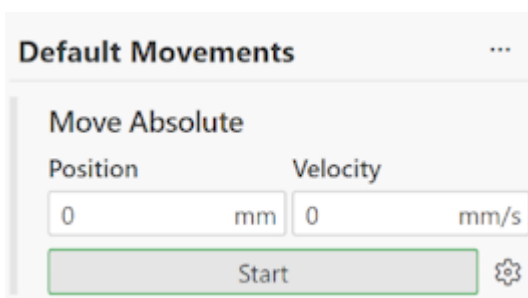
3. Bewegen Sie die Achse, indem Sie eine der Jog-Tasten



in der Sidebar anklicken und gedrückt halten.

⇒ Während Sie den Taster gedrückt halten, sollte der Achsstatus `Discrete Motion` angezeigt werden. Die angezeigte Position verändert sich während der Bewegung und die aktuelle Geschwindigkeit wird ausgegeben.

4. Führen Sie nun eine absolute Verfahrbewegung aus. Dazu ist der Move Absolut Dialog mit Werten zu füllen. Tragen Sie beispielsweise für die **Position** den Wert „1000“ ein und die für die **Geschwindigkeit** den Wert „750“.





5. Um der Bewegung spezifische Dynamiken mitzugeben, können Sie über das Zahnrad-Symbol weitere Einstellungen öffnen und die Parametrierung vornehmen.

⇒ Sind für ein Kommando benutzerdefinierte Parameter hinterlegt, wird das Zahnrad-Symbol grün



eingefärbt:

Über die Schaltfläche **Adopt as default "Init Parameter"** werden die hier definierten Parameter in die Parameterliste der Achse übernommen und überschreiben dort getätigte Einstellungen.

6. Klicken Sie den Button **Start**, um die Bewegung zu starten.

⇒ Der grüne Balken am Dialog gibt an, dass die Bewegung aktiv ist. Während der Bewegung zeigt der Achsstatus **Discrete Motion** an. Die aktuelle Position verändert sich und die Geschwindigkeit wird ebenso ausgegeben.

7. Klicken Sie den Button **Halt**, um die Achse in ihrer Bewegung zu stoppen. Das Verfahrkommando wird damit abgebrochen.

3.2.3.8 Drive-Module

Jede Achse verfügt über ein Simulation Drive [► 43]-Modul. Dieses dient zur Verwendung der Achse ohne reale Antriebshardware. Standardmäßig ist dieses Modul ausgeblendet. Die Konfiguration erfolgt immer direkt über die Achsparametern.

Sobald in den Axis Registerkarte Settings [► 28] ein Axis Typ ausgewählt ist, verfügt die Achse über ein weiteres Drive-Modul, welches als Schnittstelle zur realen Antriebshardware dient. Auch dieses Drive-Modul ist standardmäßig ausgeblendet. Solang es ausgeblendet ist, erfolgt die Parametrierung über die Axis Registerkarte Init Parameter [► 29] und die Konfiguration der zyklischen Schnittstelle zur Hardware über die Axis Registerkarte Data Area [► 33].

Wenn das Drive Modul eingeblendet ist, erfolgt die Parametrierung und Konfiguration über das Drive-Modul.

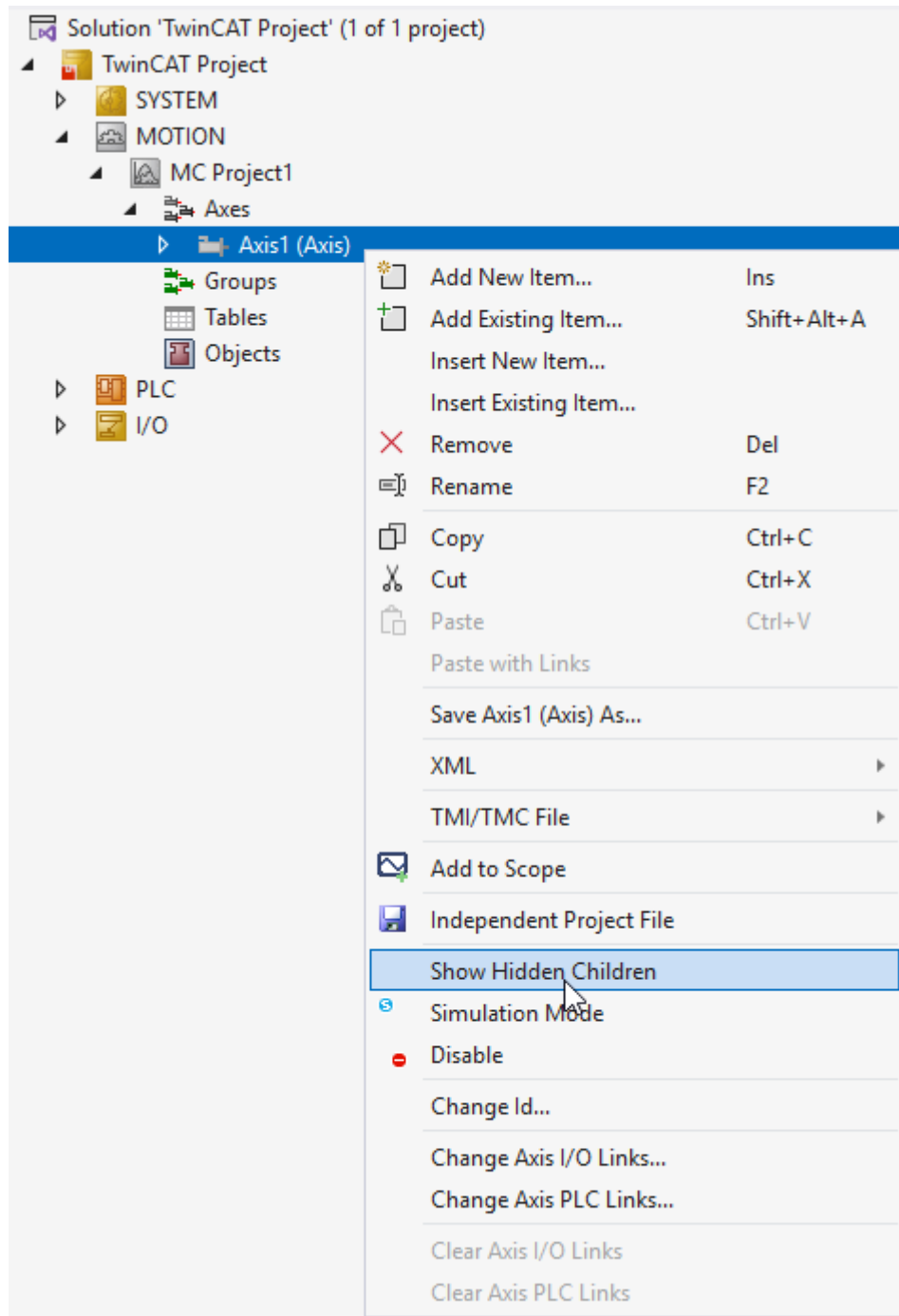


Die verfügbaren Drive-Module finden Sie in den nachfolgenden Kapitelabschnitten.

Drive Module einblenden

✓ Die Drive-Module einer Achse sind standardmäßig ausgeblendet.

1. Machen Sie einen Rechtsklick auf **Axes > Axis > Show Hidden Children**, um diese einzublenden.



3.2.3.8.1 Simulation Drive

Jede Achse verfügt über ein Simulation Drive Modul. Dieses dient der Verwendung der Achse ohne reale Antriebshardware. Standardmäßig ist dieses Modul ausgeblendet. Die Konfiguration erfolgt immer direkt über die Achsparameter.

i Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.8.1.1 Init Parameter

Name	Standardwert	Einheit	Typ
Encoder Mounting Offset	0.0	[PosUnit]	LREAL

Encoder Mounting Offset

`Encoder Mounting Offset` legt den permanenten Korrekturwert für die physikalische Encoderausrichtung fest. Der Wert ist bei der Inbetriebnahme einstellen.

3.2.3.8.1.2 Online Parameter

Scaling

Name	Einheit	Typ
Work Zero Offset	[PosUnit]	LREAL
Active Homing Offset	[PosUnit]	LREAL

Work Zero Offset

Nicht-persistenter Offset, der über den Funktionsbaustein `MC_SetPosition` eingestellt werden kann. Wird `MC_SetPosition` mehrfach ausgeführt, werden die einzelnen Offsets unter Berücksichtigung des Vorzeichens verrechnet.

Der Offset ist zur Laufzeit aktiv und wird beim Neustart von TwinCAT zurückgesetzt.

`Work Zero Offset = Vorgegebene neue Position - Aktuelle Sollposition`

Active Homing Offset

Zeigt den temporären Versatz an, der während der letzten Referenzfahrt berechnet wurde. Dieser Wert wird nicht dauerhaft gespeichert.

3.2.3.8.2 MDP742 Drive

Modul zur Verknüpfung einer MC3-Achse mit einem Antrieb mit dem Antriebsprofil MDP742.

i Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.8.2.1 Init Parameter

General

Name	Standardwert	Einheit	Typ
Position Interpretation	Incremental		MC.EEncoderInterpretation [► 203]
Maximum Absolute Counter Value (Sichtbarkeit abhängig von Position Interpretation)	1048575	INC	UDINT
Position Data Range	FullRange		MC.EEncoderDataRange [► 203]
Minimum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 0 • vorzeichenbehaftet: -2147483648	INC	LINT
Maximum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 4294967295 • vorzeichenbehaftet: 2147483647	INC	LINT
Feedback Velocity Source	HardwareProvidedIfAvailable		MC.EEncoderFeedbackVeloSource [► 203]

Position Interpretation

Der Parameter `Position Interpretation` definiert, nach welcher Semantik die Werte vom Positionsgeber übermittelt und von der MC3-Achse interpretiert werden.

Incremental

Die Einstellung `Incremental` nutzt keine absoluten Eigenschaften des physikalischen Feedbacksystems, kann aber sowohl mit inkrementellen als auch absoluten Gebersystemen verwendet werden. Nach dem Start der Steuerung ist eine Referenzfahrt (Homing) notwendig, um die Ist-Position der Achse zu initialisieren (`IsPositionValid = FALSE`).

AbsoluteFullRange

Die Einstellung `AbsoluteFullRange` wird in der Regel mit Multiturn- oder Singleturn-Absolutwertgebern verwendet. Die MC3-Achse berücksichtigt den gesamten Datenbereich des Datentyps im Mapping zum I/O-Gerät (`FullRange`). Die höchste Anzahl an Inkrementen entspricht der Obergrenzen des Datentyps. Die Größe des Datentyps ist im verwendeten Protokoll spezifiziert.

Zum Zeitpunkt des Einschaltens der Steuerung ist der Positionswert eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`. Kommt es zu einem Überlauf, signalisiert die Achse `IsPositionValid = FALSE`. Eine (erneute) Referenzfahrt ist notwendig, um die Ist-Position der Achse zu initialisieren.

Das Konfigurieren und Einschalten der softwareseitigen Positionsgrenzen der Achse ermöglicht das Wiederherstellen der Achsposition nach einem Neustart der Steuerung. Eine erneute Referenzfahrt ist nicht notwendig. Dies gilt für maximal einen Überlauf der übermittelten Inkremente. Kommt es zu einem weiteren Überlauf, signalisiert die Achse `IsPositionValid = FALSE`.

AbsolutUserDataRange

Es gelten die gleichen Randbedingungen wie für die Einstellung `AbsoluteFullRange`. Zusätzlich wird der Parameter `Maximum Absolute Counter Value` eingeblendet und berücksichtigt:

- `Maximum Absolute Counter Value`
Eindeutige maximale Anzahl an Inkrementen, die das Gebersystem übermittelt (innerhalb der Grenzen des verwendeten Datentyps). Beispielsweise ist der Standardwert auf $2^{20}-1$ eingestellt.

AbsoluteModulo

Die Positionsinkremente werden vom I/O-Gerät mit Modulo Logik übermittelt. Entspricht z. B. eine Umdrehung des Gebersystems dem Modulus der MC3-Achse, ist die Position in diesem Bereich eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`.

Position Data Range

Der Parameter `Position Data Range` definiert den Wertebereich des Gebersystems, der für die Positionsberechnung verwendet wird. Dies bezieht sich auf den Rohwert, der vom Antrieb oder Encoder über die Prozessdaten (PDO-Mapping) empfangen wird und ermöglicht die Erkennung und Handhabung von Überläufen. Der Parameter kann nur zur Konfigurationszeit und nicht zur Laufzeit verändert werden.

FullRange

Der Positionsüberlauf basiert auf den Datentypgrenzen des jeweiligen Gebersystems. Für ein vorzeichenbehaftetes 32-Bit-Gebersystem (z. B. CoE DS402 Drive (AX8000), SoE Drive (AX5000), EtherCAT Encoder) bedeutet dies implizit einen Wertebereich von -2147483648 bis 2147483647. Bei einem vorzeichenlosen 32-Bit-Gebersystem (z. B. MDP742 Drive (ausgewählte ELM72xx), SSI-Encoder) impliziert dies einen Wertebereich von 0 bis 4294967295.

UserDefined

Dieser Parameterwert ermöglicht eine manuelle Eingabe der Wertebereichsgrenzen für die Bestimmung eines Überlaufs. Es werden zwei zusätzliche Parameter zur Eingabe der oberen und unteren Grenze eingeblendet:

- Minimum Counter Value
- Maximum Counter Value

Feedback Velocity Source

Bestimmt die Quelle der Ist-Geschwindigkeit (`AXIS_REF.McToPlc.Act.Velocity`). Die Geschwindigkeit kann entweder von der verwendeten Hardware übernommen oder intern durch Differenzierung des Positions-Istwertes berechnet werden.

HardwareProvidedIfAvailable (Standardwert)

Verwendet die von der Hardware bereitgestellte Geschwindigkeit. Ist kein Eintrag für die Ist-Geschwindigkeit in den Prozessdaten gemappt, wird diese automatisch intern berechnet.

InternallyCalculated

Die Geschwindigkeit wird immer aus der Ableitung der Ist-Positionen berechnet. Ist-Geschwindigkeiten der Hardware werden ignoriert, auch wenn sie verfügbar sind.

HardwareProvided

Die Geschwindigkeit wird immer direkt vom Hardware-Feedback übernommen. Dies setzt eine Verknüpfung der Ist-Geschwindigkeit über die Prozessdaten (PDO-Mapping) voraus. Fehlt das Mapping, führt dies zu einem Fehler beim Aufstarten der Konfiguration.

Scaling

Name	Standardwert	Einheit	Typ
Position Scaling Numerator From Hardware	360	[PosUnit]/INC	LREAL
Position Scaling Denominator From Hardware	1048576.0		LREAL
Encoder Mounting Offset	0.0	[PosUnit]	LREAL
Velocity Scaling Numerator To Hardware	1.0	INC/([PosUnit]/s)	LREAL
Velocity Scaling Denominator To Hardware	1.0		LREAL

Name	Standardwert	Einheit	Typ
Direction Inverted	FALSE		BOOL
Torque Scaling From Hardware	0.1	INC/([TorqueUnit])	LREAL

Der Positions-Skalierungsfaktor ist in einen Zähler und einen Nenner gegliedert. Mit ihm werden die Weginkremente in Achsenpositionen umgerechnet oder eine Benutzereinheit aus Encoderinkrementen berechnet.

Position Scaling Numerator From Hardware

Position Scaling Numerator From Hardware ist der Vorschub, den die Applikation macht, wenn die Abtriebswelle eine Umdrehung gemacht hat.

Position Scaling Denominator From Hardware

Position Scaling Denominator From Hardware ist die Anzahl an Inkrementen, die der Antrieb ausgibt, wenn die Motorwelle eine Umdrehung macht.

Beispiel 1 zum Position Scaling:

Motor ohne Getriebe mit Ritzel mit 100 mm Umfang an AX5000 mit Standardeinstellungen:

- Numerator: 100 mm
- Denominator: 1048576

Beispiel 2 zum Position Scaling:

Motor mit Getriebe mit $i=10$ an Drehteller an AX5000 mit Standardeinstellungen:

- Numerator: $360^\circ / 10 = 36^\circ$
- Denominator: 1048576

Encoder Mounting Offset

Offset, mit dem die Position eines Gebersystem korrigiert bzw. im Maschinenkoordinatensystem ausgerichtet wird und damit den maschinenbezogenen Nullpunkt festlegt. Bei einem Neustart der Steuerung wird dieser Offset wiederhergestellt.

Velocity Scaling Numerator To Hardware

Dieser Wert dient zur Berechnung des Geschwindigkeitssoll- und -istwerts aus der Impulszahl des Encoders.

Velocity Scaling Denominator To Hardware

Dieser Wert dient zur Berechnung des Geschwindigkeitssoll- und -istwerts aus der Impulszahl des Encoders.

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung des Encoders oder Antriebs umkehren:

- `FALSE`: Die Polarität der Achsbewegung stimmt mit der Zählrichtung der Erfassungshardware überein.
- `TRUE`: Die Polarität der Achsbewegung ist invers zur Zählrichtung der Erfassungshardware.

⚠️ WARNUNG

Verletzungsgefahr durch unerwartete Bewegungen

Wenn die Zählrichtung des Encoders und die Motorpolarität nicht miteinander übereinstimmen, führt die Achse unerwartete Bewegungen aus.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Torque Scaling From Hardware

Faktor zur Skalierung des Drehmomentmesswerts auf die gewünschte Einheit.

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0:0		
.netId	0.0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Dead Time Compensation

Die Totzeiten werden bei der EtherCAT-Initialisierung automatisch ermittelt und in der TwinCAT MC3 berücksichtigt. Die ermittelten Werte bedürfen in der Regel keiner manuellen Anpassung. Dennoch kann eine manuelle Konfiguration notwendig sein, um beispielsweise antriebsinterne Totzeiten auszugleichen.

Name	Standardwert	Einheit	Typ
Dead Time Compensation From Hardware	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift From Hardware	0.0	ms	LREAL
Dead Time Compensation To Hardware	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift To Hardware	0.0	ms	LREAL

Dead Time Compensation From Hardware

Stellen Sie ein, wie die kompensierte aktuelle Position berechnet werden soll.

Additional Time Shift From Hardware

Zusätzliche Verschiebung der automatisch ermittelten Totzeit für die Kompensation der aktuellen Position.

Dead Time Compensation To Hardware

Stellen Sie ein, wie die kompensierte Sollposition berechnet werden soll. Die Totzeit wird bei der EtherCAT-Initialisierung automatisch ermittelt.

Additional Time Shift To Hardware

Zusätzliche Verschiebung der automatisch ermittelten Totzeit für die Kompensation der Sollposition.

Actual Position Filter

Name	Standardwert	Einheit	Typ
Filter Type	None		MC.EFilterType [► 204]
Filter Time Position	0.0	s	LREAL
Filter Time Velocity	0.0	s	LREAL

Name	Standardwert	Einheit	Typ
Filter Time Acceleration	0.01	s	LREAL
Filter Time	0.001	s	LREAL

Filter Type

None

Es wird kein Filter angewendet.

PT1-Filter

Es handelt sich um ein Verzögerungsglied erster Ordnung (digitaler Tiefpassfilter, Proportionalglied mit Verzögerung 1. Ordnung) zur Glättung verrauschter Signale (Istwert). Der PT1-Filter wird **unabhängig** auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Filterzeitkonstante. Die Filterzeitkonstanten sind im positiven Wertebereich zu wählen. Der Wert 0.0 deaktiviert die Filterung der entsprechenden Größe. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander.

MovingAverage-Filter

Der Moving-Average-Filter bildet den arithmetischen Mittelwert über ein gleitendes Zeitfenster der letzten N Eingangswerte und ist damit ein einfaches FIR-Tiefpassfilter (mit endlicher Impulsantwort).

Der Moving-Average-Filter wird unabhängig auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Fensterlänge N . Die Fensterlänge ergibt sich aus dem Quotienten der parametrisierten Filterzeit und der Zykluszeit. Es ist eine maximale Fensterlänge von 1000 Datenpunkte möglich. Die Totzeit entspricht der halben Fensterlänge.

Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander. Eine Filterzeit von 0.0 wird der Istwert unmittelbar verarbeitet und nicht gefiltert. Eine große Filterzeit bewirkt eine starke Glättung und erhöht die Totzeit.

KalmanFilter3x1

Der Kalman-Filter 3x1 schätzt aus den fortlaufenden Positions-Istwerten einen Satz von Position (s), Geschwindigkeit (v) und Beschleunigung (a). Dies dient zum einen der Rauschunterdrückung und zum anderen findet die Berechnung von (s,v,a) konsistent, d.h. unter Berücksichtigung der Abhängigkeit der Größen untereinander, statt. 3x1 beschreibt dabei, dass der Filter drei Ausgangsgrößen (s,v,a) und eine Eingangsgröße (s) hat.

Die Konsistenz von (s,v,a) untereinander ist häufig von Vorteil. Insbesondere wird der Kalman-Filter bei einer Kopplung einer Achse auf ein Gebersignal empfohlen.

Der Kalman-Filter 3x1 wird durch die "Filter Time" parametrisiert, welche die Zeitskala beschreibt, die zur Berechnung von (s,v,a) berücksichtigt wird. Die Wahl dieses Parameters stellt einen Kompromiss zwischen Rauschunterdrückung und dynamischer Reaktionsfähigkeit dar. Eine größere Filter Time führt zu einer größeren Rauschunterdrückung bei gleichzeitiger Abnahme der Reaktionsfähigkeit, und umgekehrt (siehe unten Parameter *Filter Time*).

Filter Time Position

Der Parameter ist aktiv, wenn der Filter Type auf PT1 oder MovingAverage eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Position. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Velocity

Der Parameter ist aktiv, wenn der Filter Type auf PT1 oder MovingAverage eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Geschwindigkeit. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Acceleration

Der Parameter ist aktiv, wenn der `Filter Type` auf `PT1` oder `MovingAverage` eingestellt ist. Der Wert `0.0` deaktiviert die Filterung der Beschleunigung. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time

Der Parameter ist aktiv, wenn der `Filter Type` auf `KalmanFilter3x1` eingestellt ist. Der Kalman Filter wird durch eine vereinfachte Parametrierung über eine einzelne Filterzeitkonstante auf die Anwendung angepasst. Je größer die Filterzeitkonstante, desto höher ist der Grad der Glättung.

3.2.3.8.2.2 Online Parameter

Scaling

Name	Einheit	Typ
Work Zero Offset	[PosUnit]	LREAL
Active Homing Offset	[PosUnit]	LREAL

Work Zero Offset

Nicht-persistenter Offset, der über den Funktionsbaustein `MC_SetPosition` eingestellt werden kann. Wird `MC_SetPosition` mehrfach ausgeführt, werden die einzelnen Offsets unter Berücksichtigung des Vorzeichens verrechnet.

Der Offset ist zur Laufzeit aktiv und wird beim Neustart von TwinCAT zurückgesetzt.

`Work Zero Offset = Vorgegebene neue Position - Aktuelle Sollposition`

Active Homing Offset

Zeigt den temporären Versatz an, der während der letzten Referenzfahrt berechnet wurde. Dieser Wert wird nicht dauerhaft gespeichert.

Folgende Online Parameter sind identisch wie für `TouchProbe1` auch für `TouchProbe2` verfügbar. Der Lesbarkeit halber und um Dopplungen zu vermeiden sind diese Parameter hier nur für `TouchProbe1` aufgelistet.

TouchProbe1 Positive Edge Status

Name	Typ
State	<code>MC.ETouchProbeTriggerState</code> [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `#Invalid` signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `0` signalisiert, dass noch kein Ereignis erfasst wurde.

TouchProbe1 Negative Edge Status

Name	Typ
State	<code>MC.ETouchProbeTriggerState</code> [► 208]

Name	Typ
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert #Invalid signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

3.2.3.8.2.3 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

MDP742 Drive / MDP742 Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Statusword	UINT	Status-Informationen
☒	Following error actual value	DINT	Schleppfehler (Lageregelabweichung, d. h. Sollposition minus Istposition) des IO-Drives in Inkrementen.
☒	WcState	BIT	Working Counter
☒	InputToggle	BIT	Pegelwechsel bei Erhalt eines neuen Telegramms.
☒	Position	UDINT	Ist-Position des IO-Drives in Inkrementen.
☐	Velocity actual value	DINT	Ist-Geschwindigkeit des IO-Drives in Inkrementen.
☒	Torque actual value	INT	Drehmoment/Kraft-Istwert des IO-Drives in Inkrementen.
☒	TxPdoState	BIT	Dieses Element gibt an, ob sich die vom Encoder gemessenen Werte in einem ungültigen Zustand befinden.
☒	DcInputShift	DINT	Wird für die Totzeitkompensation des Antriebs verwendet.
☒	DcOutputShift	DINT	Wird für die Totzeitkompensation des Antriebs verwendet.

MDP742 Drive / MDP742 TouchProbe Inputs

In der Standardeinstellung ist dieser Datenbereich deaktiviert.

Enable (default)	Name	Typ	Beschreibung
☒	TP1 Enable	BIT	Messtaster 1 eingeschaltet

Enable (default)	Name	Typ	Beschreibung
☒	TP1 Pos value stored	BIT	Wert für Messtaster 1 bei steigender Flanke gespeichert
☒	TP1 Neg value stored	BIT	Wert für Messtaster 1 bei fallender Flanke gespeichert
☒	TP1 Input	BIT	Digitaler Eingang Messtaster 1
☒	TP2 Enable	BIT	Messtaster 2 eingeschaltet
☒	TP2 Pos value stored	BIT	Wert für Messtaster 2 bei steigender Flanke gespeichert
☒	TP2 Neg value stored	BIT	Wert für Messtaster 2 bei fallender Flanke gespeichert
☒	TP2 Input	BIT	Digitaler Eingang Messtaster 2
☒	TP1 Pos position	UDINT	Positiver Wert von Messtaster 1 Einheit: der angegebene Wert muss mit dem entsprechenden <u>Skalierungsfaktor</u> [► 46] multipliziert werden
☒	TP1 Neg position	UDINT	Negativer Wert von Messtaster 1 Einheit: der angegebene Wert muss mit dem entsprechenden <u>Skalierungsfaktor</u> [► 46] multipliziert werden
☒	TP1 Pos timestamp	UDINT	Zeitstempel der Messtaster 1, während ein Ereignis mit steigender Flanke erkannt wurde.
☒	TP1 Neg timestamp	UDINT	Zeitstempel der Messtaster 1, während ein Ereignis mit fallender Flanke erkannt wurde.
☒	TP2 Pos position	UDINT	Positiver Wert von Messtaster 2 Einheit: der angegebene Wert muss mit dem entsprechenden <u>Skalierungsfaktor</u> [► 46] multipliziert werden
☒	TP2 Neg position	UDINT	Negativer Wert von Messtaster 2 Einheit: der angegebene Wert muss mit dem entsprechenden <u>Skalierungsfaktor</u> [► 46] multipliziert werden
☒	TP2 Pos timestamp	UDINT	Zeitstempel der Messtaster 2, während ein Ereignis mit steigender Flanke erkannt wurde.
☒	TP2 Neg timestamp	UDINT	Zeitstempel der Messtaster 2, während ein Ereignis mit fallender Flanke erkannt wurde.

MDP742 Drive / MDP742 Outputs

Enable (default)	Name	Typ	Beschreibung
☒	Controlword	UINT	Control-Informationen
☒	Target position	UDINT	Aktuelle Sollposition in Inkrementen Die Sollposition der MC3-Achse in physikalischen Einheiten, wie z. B. mm oder Grad, wird von der MC3 mathematisch in einen Inkrementalwert zurückgerechnet und zum Antrieb

Enable (default)	Name	Typ	Beschreibung
			übertragen. Hierbei werden Überläufe des Inkrementalwertes in der Sollposition von der MC3 berücksichtigt.
☐	Target velocity	DINT	Aktuelle Sollgeschwindigkeit in Inkrementen Die Sollgeschwindigkeit der MC3-Achse in physikalischen Einheiten, wie z. B. mm/s oder Grad/s, wird von der MC3 mathematisch in einen Inkrementalwert zurückgerechnet und zum Antrieb übertragen.

MDP742 Drive / MDP742 Messtaster Outputs

In der Standardeinstellung ist dieser Datenbereich deaktiviert.

Enable (default)	Name	Typ	Beschreibung
☒	TP1 Enable	BIT	Messtaster 1 einschalten
☒	TP1 Continuous	BIT	Einmalige oder zyklische Ereignisse 0: erstes Event nach dem Einschalten detektiert 1: fortlaufende Detektierung
☒	TP1 Trigger mode	BIT2	Es wird Eingang 1 getriggert
☒	TP1 Enable pos edge	BIT	Ausschlaggebend für ein Event ist eine steigende Flanke an Messtaster 1
☒	TP1 Enable neg edge	BIT	Ausschlaggebend für ein Event ist eine fallende Flanke an Messtaster 1
☒	TP2 Enable	BIT	Messtaster 2 einschalten
☒	TP2 Continuous	BIT	Einmalige oder zyklische Ereignisse 0: erstes Event nach dem Einschalten detektiert 1: fortlaufende Detektierung
☒	TP2 Trigger mode	BIT2	Es wird Eingang 2 getriggert
☒	TP2 Enable pos edge	BIT	Ausschlaggebend für ein Event ist eine steigende Flanke an Messtaster 2
☒	TP2 Enable neg edge	BIT	Ausschlaggebend für ein Event ist eine fallende Flanke an Messtaster 2

HINWEIS

Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.8.3 CoE DS402 Drive

Modul zur Verknüpfung einer MC3-Achse mit einem Antrieb mit dem Antriebsprofil CoE DS402.

● Ansicht unterlagerter Objekte

i In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.8.3.1 Init Parameter

General

Name	Standardwert	Einheit	Typ
Position Interpretation	Incremental		MC.EEncoderInterpretation [► 203]
Maximum Absolute Counter Value (Sichtbarkeit abhängig von Position Interpretation)	1048575	INC	UDINT
Position Data Range	FullRange		MC.EEncoderDataRange [► 203]
Minimum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 0 • vorzeichenbehaftet: -2147483648	INC	LINT
Maximum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 4294967295 • vorzeichenbehaftet: 2147483647	INC	LINT
Feedback Velocity Source	HardwareProvidedIfAvailable		MC.EEncoderFeedbackVeloSource [► 203]

Position Interpretation

Der Parameter `Position Interpretation` definiert, nach welcher Semantik die Werte vom Positionsgeber übermittelt und von der MC3-Achse interpretiert werden.

`Incremental`

Die Einstellung `Incremental` nutzt keine absoluten Eigenschaften des physikalischen Feedbacksystems, kann aber sowohl mit inkrementellen als auch absoluten Gebersystemen verwendet werden. Nach dem Start der Steuerung ist eine Referenzfahrt (Homing) notwendig, um die Ist-Position der Achse zu initialisieren (`IsPositionValid = FALSE`).

`AbsoluteFullRange`

Die Einstellung `AbsoluteFullRange` wird in der Regel mit Multiturn- oder Singleturn-Absolutwertgebern verwendet. Die MC3-Achse berücksichtigt den gesamten Datenbereich des Datentyps im Mapping zum I/O-Gerät (`FullRange`). Die höchste Anzahl an Inkrementen entspricht der Obergrenzen des Datentyps. Die Größe des Datentyps ist im verwendeten Protokoll spezifiziert.

Zum Zeitpunkt des Einschaltens der Steuerung ist der Positionswert eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`. Kommt es zu einem Überlauf, signalisiert die Achse `IsPositionValid = FALSE`. Eine (erneute) Referenzfahrt ist notwendig, um die Ist-Position der Achse zu initialisieren.

Das Konfigurieren und Einschalten der softwareseitigen Positionsgrenzen der Achse ermöglicht das Wiederherstellen der Achsposition nach einem Neustart der Steuerung. Eine erneute Referenzfahrt ist nicht notwendig. Dies gilt für maximal einen Überlauf der übermittelten Inkremente. Kommt es zu einem weiteren Überlauf, signalisiert die Achse `IsPositionValid = FALSE`.

AbsolutUserDataRange

Es gelten die gleichen Randbedingungen wie für die Einstellung `AbsoluteFullRange`. Zusätzlich wird der Parameter `Maximum Absolute Counter Value` eingeblendet und berücksichtigt:

- `Maximum Absolute Counter Value`
Eindeutige maximale Anzahl an Inkrementen, die das Gebersystem übermittelt (innerhalb der Grenzen des verwendeten Datentyps). Beispielsweise ist der Standardwert auf $2^{20}-1$ eingestellt.

AbsoluteModulo

Die Positionsinkremente werden vom I/O-Gerät mit Modulo Logik übermittelt. Entspricht z. B. eine Umdrehung des Gebersystems dem Modulus der MC3-Achse, ist die Position in diesem Bereich eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`.

Position Data Range

Der Parameter `Position Data Range` definiert den Wertebereich des Gebersystems, der für die Positionsberechnung verwendet wird. Dies bezieht sich auf den Rohwert, der vom Antrieb oder Encoder über die Prozessdaten (PDO-Mapping) empfangen wird und ermöglicht die Erkennung und Handhabung von Überläufen. Der Parameter kann nur zur Konfigurationszeit und nicht zur Laufzeit verändert werden.

FullRange

Der Positionsüberlauf basiert auf den Datentypgrenzen des jeweiligen Gebersystems. Für ein vorzeichenbehaftetes 32-Bit-Gebersystem (z. B. CoE DS402 Drive (AX8000), SoE Drive (AX5000), EtherCAT Encoder) bedeutet dies implizit einen Wertebereich von -2147483648 bis 2147483647. Bei einem vorzeichenlosen 32-Bit-Gebersystem (z. B. MDP742 Drive (ausgewählte ELM72xx), SSI-Encoder) impliziert dies einen Wertebereich von 0 bis 4294967295.

UserDefined

Dieser Parameterwert ermöglicht eine manuelle Eingabe der Wertebereichsgrenzen für die Bestimmung eines Überlaufs. Es werden zwei zusätzliche Parameter zur Eingabe der oberen und unteren Grenze eingeblendet:

- `Minimum Counter Value`
- `Maximum Counter Value`

Feedback Velocity Source

Bestimmt die Quelle der Ist-Geschwindigkeit (`AXIS_REF.McToPlc.Act.Velocity`). Die Geschwindigkeit kann entweder von der verwendeten Hardware übernommen oder intern durch Differenzierung des Positions-Istwertes berechnet werden.

HardwareProvidedIfAvailable (Standardwert)

Verwendet die von der Hardware bereitgestellte Geschwindigkeit. Ist kein Eintrag für die Ist-Geschwindigkeit in den Prozessdaten gemappt, wird diese automatisch intern berechnet.

InternallyCalculated

Die Geschwindigkeit wird immer aus der Ableitung der Ist-Positionen berechnet. Ist-Geschwindigkeiten der Hardware werden ignoriert, auch wenn sie verfügbar sind.

HardwareProvided

Die Geschwindigkeit wird immer direkt vom Hardware-Feedback übernommen. Dies setzt eine Verknüpfung der Ist-Geschwindigkeit über die Prozessdaten (PDO-Mapping) voraus. Fehlt das Mapping, führt dies zu einem Fehler beim Aufstarten der Konfiguration.

Scaling

Name	Standardwert	Einheit	Typ
Position Scaling Numerator From Hardware	360	[PosUnit]/INC	LREAL
Position Scaling Denominator From Hardware	1048576.0		LREAL
Encoder Mounting Offset	0.0	[PosUnit]	LREAL
Velocity Scaling Numerator To Hardware	1.0	INC/([PosUnit]/s)	LREAL
Velocity Scaling Denominator To Hardware	1.0		LREAL
Direction Inverted	FALSE		BOOL
Torque Scaling From Hardware	0.1	INC/([TorqueUnit])	LREAL

Der Positions-Skalierungsfaktor ist in einen Zähler und einen Nenner gegliedert. Mit ihm werden die Weginkremente in Achsenpositionen umgerechnet oder eine Benutzereinheit aus Encoderinkrementen berechnet.

Position Scaling Numerator From Hardware

Position Scaling Numerator From Hardware ist der Vorschub, den die Applikation macht, wenn die Abtriebswelle eine Umdrehung gemacht hat.

Position Scaling Denominator From Hardware

Position Scaling Denominator From Hardware ist die Anzahl an Inkrementen, die der Antrieb ausgibt, wenn die Motorwelle eine Umdrehung macht.

Beispiel 1 zum Position Scaling:

Motor ohne Getriebe mit Ritzel mit 100 mm Umfang an AX5000 mit Standardeinstellungen:

- Numerator: 100 mm
- Denominator: 1048576

Beispiel 2 zum Position Scaling:

Motor mit Getriebe mit $i=10$ an Drehteller an AX5000 mit Standardeinstellungen:

- Numerator: $360^\circ / 10 = 36^\circ$
- Denominator: 1048576

Encoder Mounting Offset

Offset, mit dem die Position eines Gebersystem korrigiert bzw. im Maschinenkoordinatensystem ausgerichtet wird und damit den maschinenbezogenen Nullpunkt festlegt. Bei einem Neustart der Steuerung wird dieser Offset wiederhergestellt.

Velocity Scaling Numerator To Hardware

Dieser Wert dient zur Berechnung des Geschwindigkeitssoll- und -istwerts aus der Impulszahl des Encoders.

Velocity Scaling Denominator To Hardware

Dieser Wert dient zur Berechnung des Geschwindigkeitssoll- und -istwerts aus der Impulszahl des Encoders.

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung des Encoders oder Antriebs umkehren:

- `FALSE`: Die Polarität der Achsbewegung stimmt mit der Zählrichtung der Erfassungshardware überein.
- `TRUE`: Die Polarität der Achsbewegung ist invers zur Zählrichtung der Erfassungshardware.

⚠️ WARNUNG**Verletzungsgefahr durch unerwartete Bewegungen**

Wenn die Zählrichtung des Encoders und die Motorpolarität nicht miteinander übereinstimmen, führt die Achse unerwartete Bewegungen aus.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Torque Scaling From Hardware

Faktor zur Skalierung des Drehmomentmesswerts auf die gewünschte Einheit.

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0:0		
.netId	0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Dead Time Compensation

Die Totzeiten werden bei der EtherCAT-Initialisierung automatisch ermittelt und in der TwinCAT MC3 berücksichtigt. Die ermittelten Werte bedürfen in der Regel keiner manuellen Anpassung. Dennoch kann eine manuelle Konfiguration notwendig sein, um beispielsweise antriebsinterne Totzeiten auszugleichen.

Name	Standardwert	Einheit	Typ
Dead Time Compensation From Hardware	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift From Hardware	0.0	ms	LREAL
Dead Time Compensation To Hardware	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift To Hardware	0.0	ms	LREAL

Dead Time Compensation From Hardware

Stellen Sie ein, wie die kompensierte aktuelle Position berechnet werden soll.

Additional Time Shift From Hardware

Zusätzliche Verschiebung der automatisch ermittelten Totzeit für die Kompensation der aktuellen Position.

Dead Time Compensation To Hardware

Stellen Sie ein, wie die kompensierte Sollposition berechnet werden soll. Die Totzeit wird bei der EtherCAT-Initialisierung automatisch ermittelt.

Additional Time Shift To Hardware

Zusätzliche Verschiebung der automatisch ermittelten Totzeit für die Kompensation der Sollposition.

Touch Probe

Name	Standardwert	Einheit	Typ
TouchProbe1 Source	DriveDefault		MC.ETouchProbeTriggerSource [► 208]
TouchProbe2 Source	DriveDefault		MC.ETouchProbeTriggerSource [► 208]

TouchProbe Source

Auswahl des Eingangs für den Messtaster.

Actual Position Filter

Name	Standardwert	Einheit	Typ
Filter Type	None		MC.EFilterType [► 204]
Filter Time Position	0.0	s	LREAL
Filter Time Velocity	0.0	s	LREAL
Filter Time Acceleration	0.01	s	LREAL
Filter Time	0.001	s	LREAL

Filter Type

None

Es wird kein Filter angewendet.

PT1-Filter

Es handelt sich um ein Verzögerungsglied erster Ordnung (digitaler Tiefpassfilter, Proportionalglied mit Verzögerung 1. Ordnung) zur Glättung verrauschter Signale (Istwert). Der PT1-Filter wird **unabhängig** auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Filterzeitkonstante. Die Filterzeitkonstanten sind im positiven Wertebereich zu wählen. Der Wert 0.0 deaktiviert die Filterung der entsprechenden Größe. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander.

MovingAverage-Filter

Der Moving-Average-Filter bildet den arithmetischen Mittelwert über ein gleitendes Zeitfenster der letzten N Eingangswerte und ist damit ein einfaches FIR-Tiefpassfilter (mit endlicher Impulsantwort).

Der Moving-Average-Filter wird unabhängig auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Fensterlänge N . Die Fensterlänge ergibt sich aus dem Quotienten der parametrisierten Filterzeit und der Zykluszeit. Es ist eine maximale Fensterlänge von 1000 Datenpunkte möglich. Die Totzeit entspricht der halben Fensterlänge.

Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander. Eine Filterzeit von 0.0 wird der Istwert unmittelbar verarbeitet und nicht gefiltert. Eine große Filterzeit bewirkt eine starke Glättung und erhöht die Totzeit.

KalmanFilter3x1

Der Kalman-Filter 3x1 schätzt aus den fortlaufenden Positions-Istwerten einen Satz von Position (s), Geschwindigkeit (v) und Beschleunigung (a). Dies dient zum einen der Rauschunterdrückung und zum anderen findet die Berechnung von (s,v,a) konsistent, d.h. unter Berücksichtigung der Abhängigkeit der Größen untereinander, statt. 3x1 beschreibt dabei, dass der Filter drei Ausgangsgrößen (s,v,a) und eine

Eingangsgröße (s) hat.

Die Konsistenz von (s,v,a) untereinander ist häufig von Vorteil. Insbesondere wird der Kalman-Filter bei einer Kopplung einer Achse auf ein Gebersignal empfohlen.

Der Kalman-Filter 3x1 wird durch die "Filter Time" parametrisiert, welche die Zeitskala beschreibt, die zur Berechnung von (s,v,a) berücksichtigt wird. Die Wahl dieses Parameters stellt einen Kompromiss zwischen Rauschunterdrückung und dynamischer Reaktionsfähigkeit dar. Eine größere Filter Time führt zu einer größeren Rauschunterdrückung bei gleichzeitiger Abnahme der Reaktionsfähigkeit, und umgekehrt (siehe unten Parameter *Filter Time*).

Filter Time Position

Der Parameter ist aktiv, wenn der *Filter Type* auf *PT1* oder *MovingAverage* eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Position. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Velocity

Der Parameter ist aktiv, wenn der *Filter Type* auf *PT1* oder *MovingAverage* eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Geschwindigkeit. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Acceleration

Der Parameter ist aktiv, wenn der *Filter Type* auf *PT1* oder *MovingAverage* eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Beschleunigung. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time

Der Parameter ist aktiv, wenn der *Filter Type* auf *KalmanFilter3x1* eingestellt ist. Der Kalman Filter wird durch eine vereinfachte Parametrierung über eine einzelne Filterzeitkonstante auf die Anwendung angepasst. Je größer die Filterzeitkonstante, desto höher ist der Grad der Glättung.

3.2.3.8.3.2 Online Parameter

Scaling

Name	Einheit	Typ
Work Zero Offset	[PosUnit]	LREAL
Active Homing Offset	[PosUnit]	LREAL

Work Zero Offset

Nicht-persistenter Offset, der über den Funktionsbaustein MC_SetPosition eingestellt werden kann. Wird MC_SetPosition mehrfach ausgeführt, werden die einzelnen Offsets unter Berücksichtigung des Vorzeichens verrechnet.

Der Offset ist zur Laufzeit aktiv und wird beim Neustart von TwinCAT zurückgesetzt.

Work Zero Offset = Vorgegebene neue Position - Aktuelle Sollposition

Active Homing Offset

Zeigt den temporären Versatz an, der während der letzten Referenzfahrt berechnet wurde. Dieser Wert wird nicht dauerhaft gespeichert.

Folgende Online Parameter sind identisch wie für TouchProbe1 auch für TouchProbe2 verfügbar. Der Lesbarkeit halber und um Dopplungen zu vermeiden sind diese Parameter hier nur für TouchProbe1 aufgelistet.

TouchProbe1 Positive Edge Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert #Invalid signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

TouchProbe1 Negative Edge Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert #Invalid signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

3.2.3.8.3.3 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

CoE DS402 Drive /CoE Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Statusword	WORD	Status-Informationen
☒	Following error actual value	DINT	Schleppfehler (Lageregelabweichung, d. h. Sollposition minus Istposition) des IO-Drives in Inkrementen.
☒	Position actual value invalid	BIT	Dieses Element gibt an, ob der vom Positionsgeber gemessene Wert ungültig ist.

Enable (default)	Name	Typ	Beschreibung
☒	WcState	BIT	Working Counter
☒	InputToggle	BIT	Pegelwechsel bei Erhalt eines neuen Telegramms.
☒	Position actual value	DINT	Ist-Position des IO-Drives in Inkrementen.
☐	Velocity actual value	DINT	Ist-Geschwindigkeit des IO-Drives in Inkrementen.
☒	Torque/force actual value	INT	Drehmoment/Kraft-Istwert des IO-Drives in Inkrementen.
☒	DcInputShift	DINT	Wird für die Totzeitkompensation des Antriebs verwendet.
☒	DcOutputShift	DINT	Wird für die Totzeitkompensation des Antriebs verwendet.

CoE DS402 Drive /DS402 Touchprobe Inputs

In der Standardeinstellung ist dieser Datenbereich deaktiviert.

Enable (default)	Name	Typ	Beschreibung
☒	Touch probe status	UINT	Dieses Element zeigt den Status der Messtaster-Funktion an.
☒	Touch probe 1 positive edge	DINT	Dieses Element liefert den Positionswert des Messtasters 1 bei der steigenden Flanke.
☒	Touch probe 1 negative edge	DINT	Dieses Element liefert den Positionswert des Messtasters 1 bei fallender Flanke.
☒	Touch probe time stamp 1 positive edge	UDINT	Dieses Element liefert den Zeitstempel, während ein Ereignis mit steigender Flanke an Messtasters 1 erkannt wurde.
☒	Touch probe time stamp 1 negative edge	UDINT	Dieses Element liefert den Zeitstempel, während ein Ereignis mit fallender Flanke an Messtasters 1 erkannt wurde.
☒	Touch probe 2 positive edge	DINT	Dieses Element liefert den Positionswert des Messtasters 2 bei der steigenden Flanke.
☒	Touch probe 2 negative edge	DINT	Dieses Element liefert den Positionswert des Messtasters 2 bei fallender Flanke.
☒	Touch probe time stamp 2 positive edge	UDINT	Dieses Element liefert den Zeitstempel, während ein Ereignis mit steigender Flanke an Messtasters 2 erkannt wurde.
☒	Touch probe time stamp 2 negative edge	UDINT	Dieses Element liefert den Zeitstempel, während ein Ereignis mit fallender Flanke an Messtasters 2 erkannt wurde.

CoE DS402 Drive /CoE Outputs

Enable (default)	Name	Typ	Beschreibung
☒	Controlword	WORD	Control-Informationen
☒	Target position	DINT	Aktuelle Sollposition in Inkrementen Die Sollposition der MC3-Achse in physikalischen Einheiten, wie z. B. mm oder Grad, wird von der MC3

Enable (default)	Name	Typ	Beschreibung
			mathematisch in einen Inkrementalwert zurückgerechnet und zum Antrieb übertragen. Hierbei werden Überläufe des Inkrementalwertes in der Sollposition von der MC3 berücksichtigt.
☐	Target velocity	DINT	Aktuelle Sollgeschwindigkeit in Inkrementen Die Sollgeschwindigkeit der MC3-Achse in physikalischen Einheiten, wie z. B. mm/s oder Grad/s, wird von der MC3 mathematisch in einen Inkrementalwert zurückgerechnet und zum Antrieb übertragen.

CoE DS402 Drive /DS402 TouchProbe Outputs

In der Standardeinstellung ist dieser Datenbereich deaktiviert.

Enable (default)	Name	Typ	Beschreibung
☒	Touch probe control	UINT	Dieses Element enthält die Konfiguration der Messtester-Funktion.

HINWEIS

Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.8.4 SoE Drive

Modul zur Verknüpfung einer MC3-Achse mit einem Antrieb mit dem Antriebsprofil SoE.



Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.8.4.1 Init Parameter

General

Name	Standardwert	Einheit	Typ
Position Interpretation	Incremental		MC.EEncoderInterpretation [► 203]

Name	Standardwert	Einheit	Typ
Maximum Absolute Counter Value (Sichtbarkeit abhängig von Position Interpretation)	1048575	INC	UDINT
Position Data Range	FullRange		MC.EEncoderDataRange [► 203]
Minimum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 0 • vorzeichenbehaftet: -2147483648	INC	LINT
Maximum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 4294967295 • vorzeichenbehaftet: 2147483647	INC	LINT
Feedback Velocity Source	HardwareProvidedIfAvailable		MC.EEncoderFeedbackVeloSource [► 203]

Position Interpretation

Der Parameter `Position Interpretation` definiert, nach welcher Semantik die Werte vom Positionsgeber übermittelt und von der MC3-Achse interpretiert werden.

Incremental

Die Einstellung `Incremental` nutzt keine absoluten Eigenschaften des physikalischen Feedbacksystems, kann aber sowohl mit inkrementellen als auch absoluten Gebersystemen verwendet werden. Nach dem Start der Steuerung ist eine Referenzfahrt (Homing) notwendig, um die Ist-Position der Achse zu initialisieren (`IsPositionValid = FALSE`).

AbsoluteFullRange

Die Einstellung `AbsoluteFullRange` wird in der Regel mit Multiturn- oder Singleturn-Absolutwertgebern verwendet. Die MC3-Achse berücksichtigt den gesamten Datenbereich des Datentyps im Mapping zum I/O-Gerät (`FullRange`). Die höchste Anzahl an Inkrementen entspricht der Obergrenzen des Datentyps. Die Größe des Datentyps ist im verwendeten Protokoll spezifiziert.

Zum Zeitpunkt des Einschaltens der Steuerung ist der Positionswert eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`. Kommt es zu einem Überlauf, signalisiert die Achse `IsPositionValid = FALSE`. Eine (erneute) Referenzfahrt ist notwendig, um die Ist-Position der Achse zu initialisieren.

Das Konfigurieren und Einschalten der softwareseitigen Positionsgrenzen der Achse ermöglicht das Wiederherstellen der Achsposition nach einem Neustart der Steuerung. Eine erneute Referenzfahrt ist nicht notwendig. Dies gilt für maximal einen Überlauf der übermittelten Inkremente. Kommt es zu einem weiteren Überlauf, signalisiert die Achse `IsPositionValid = FALSE`.

AbsolutUserDataRange

Es gelten die gleichen Randbedingungen wie für die Einstellung `AbsoluteFullRange`. Zusätzlich wird der Parameter `Maximum Absolute Counter Value` eingeblendet und berücksichtigt:

- `Maximum Absolute Counter Value`
Eindeutige maximale Anzahl an Inkrementen, die das Gebersystem übermittelt (innerhalb der Grenzen des verwendeten Datentyps). Beispielsweise ist der Standardwert auf $2^{20}-1$ eingestellt.

AbsoluteModulo

Die Positionsinkremente werden vom I/O-Gerät mit Modulo Logik übermittelt. Entspricht z. B. eine Umdrehung des Gebersystems dem Modulus der MC3-Achse, ist die Position in diesem Bereich eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`.

Position Data Range

Der Parameter `Position Data Range` definiert den Wertebereich des Gebersystems, der für die Positionsrechnung verwendet wird. Dies bezieht sich auf den Rohwert, der vom Antrieb oder Encoder über die Prozessdaten (PDO-Mapping) empfangen wird und ermöglicht die Erkennung und Handhabung von Überläufen. Der Parameter kann nur zur Konfigurationszeit und nicht zur Laufzeit verändert werden.

FullRange

Der Positionsüberlauf basiert auf den Datentypgrenzen des jeweiligen Gebersystems. Für ein vorzeichenbehaftetes 32-Bit-Gebersystem (z. B. CoE DS402 Drive (AX8000), SoE Drive (AX5000), EtherCAT Encoder) bedeutet dies implizit einen Wertebereich von -2147483648 bis 2147483647. Bei einem vorzeichenlosen 32-Bit-Gebersystem (z. B. MDP742 Drive (ausgewählte ELM72xx), SSI-Encoder) impliziert dies einen Wertebereich von 0 bis 4294967295.

UserDefined

Dieser Parameterwert ermöglicht eine manuelle Eingabe der Wertebereichsgrenzen für die Bestimmung eines Überlaufs. Es werden zwei zusätzliche Parameter zur Eingabe der oberen und unteren Grenze eingeblendet:

- Minimum Counter Value
- Maximum Counter Value

Feedback Velocity Source

Bestimmt die Quelle der Ist-Geschwindigkeit (`AXIS_REF.McToPlc.Act.Velocity`). Die Geschwindigkeit kann entweder von der verwendeten Hardware übernommen oder intern durch Differenzierung des Positions-Istwertes berechnet werden.

HardwareProvidedIfAvailable (Standardwert)

Verwendet die von der Hardware bereitgestellte Geschwindigkeit. Ist kein Eintrag für die Ist-Geschwindigkeit in den Prozessdaten gemappt, wird diese automatisch intern berechnet.

InternallyCalculated

Die Geschwindigkeit wird immer aus der Ableitung der Ist-Positionen berechnet. Ist-Geschwindigkeiten der Hardware werden ignoriert, auch wenn sie verfügbar sind.

HardwareProvided

Die Geschwindigkeit wird immer direkt vom Hardware-Feedback übernommen. Dies setzt eine Verknüpfung der Ist-Geschwindigkeit über die Prozessdaten (PDO-Mapping) voraus. Fehlt das Mapping, führt dies zu einem Fehler beim Aufstarten der Konfiguration.

Scaling

Name	Standardwert	Einheit	Typ
Position Scaling Numerator From Hardware	360	[PosUnit]/INC	LREAL
Position Scaling Denominator From Hardware	1048576.0		LREAL
Encoder Mounting Offset	0.0	[PosUnit]	LREAL
Velocity Scaling Numerator To Hardware	1.0	INC/([PosUnit]/s)	LREAL
Velocity Scaling Denominator To Hardware	1.0		LREAL
Direction Inverted	FALSE		BOOL
Torque Scaling From Hardware	0.1	INC/([TorqueUnit])	LREAL

Der Positions-Skalierungsfaktor ist in einen Zähler und einen Nenner gegliedert. Mit ihm werden die Weginkremente in Achsenpositionen umgerechnet oder eine Benutzereinheit aus Encoderinkrementen berechnet.

Position Scaling Numerator From Hardware

Position Scaling Numerator From Hardware ist der Vorschub, den die Applikation macht, wenn die Abtriebswelle eine Umdrehung gemacht hat.

Position Scaling Denominator From Hardware

Position Scaling Denominator From Hardware ist die Anzahl an Inkrementen, die der Antrieb ausgibt, wenn die Motorwelle eine Umdrehung macht.

Beispiel 1 zum Position Scaling:

Motor ohne Getriebe mit Ritzel mit 100 mm Umfang an AX5000 mit Standardeinstellungen:

- Numerator: 100 mm
- Denominator: 1048576

Beispiel 2 zum Position Scaling:

Motor mit Getriebe mit $i=10$ an Drehteller an AX5000 mit Standardeinstellungen:

- Numerator: $360^\circ / 10 = 36^\circ$
- Denominator: 1048576

Encoder Mounting Offset

Offset, mit dem die Position eines Gebersystem korrigiert bzw. im Maschinenkoordinatensystem ausgerichtet wird und damit den maschinenbezogenen Nullpunkt festlegt. Bei einem Neustart der Steuerung wird dieser Offset wiederhergestellt.

Velocity Scaling Numerator To Hardware

Dieser Wert dient zur Berechnung des Geschwindigkeitssoll- und -istwerts aus der Impulszahl des Encoders.

Velocity Scaling Denominator To Hardware

Dieser Wert dient zur Berechnung des Geschwindigkeitssoll- und -istwerts aus der Impulszahl des Encoders.

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung des Encoders oder Antriebs umkehren:

- `FALSE`: Die Polarität der Achsbewegung stimmt mit der Zählrichtung der Erfassungshardware überein.
- `TRUE`: Die Polarität der Achsbewegung ist invers zur Zählrichtung der Erfassungshardware.

⚠️ WARNUNG

Verletzungsgefahr durch unerwartete Bewegungen

Wenn die Zählrichtung des Encoders und die Motorpolarität nicht miteinander übereinstimmen, führt die Achse unerwartete Bewegungen aus.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Torque Scaling From Hardware

Faktor zur Skalierung des Drehmomentmesswerts auf die gewünschte Einheit.

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0.0:0		
.netId	0.0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Dead Time Compensation

Voraussetzung für die hochgenaue Umrechnung von Positionen in Zeiten und umgekehrt ist eine exakte Totzeitkompensation der Achsen.

Name	Standardwert	Einheit	Typ
Dead Time Compensation From Hardware	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift From Hardware	0.0	ms	LREAL
Dead Time Compensation	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift To Hardware	0.0	ms	LREAL

Dead Time Compensation From Hardware

Legen Sie fest, wie die kompensierte Istposition berechnet werden soll. Die Totzeit wird während der EtherCAT-Initialisierung automatisch ermittelt.

Additional Time Shift From Hardware

Zeitdauer zur Anpassung der (additiven), automatisch ermittelten Totzeit für die Kompensation der Istposition.

Dead Time Compensation

Legen Sie fest, wie die kompensierte Sollposition berechnet werden soll. Die Totzeit wird während der EtherCAT-Initialisierung automatisch ermittelt.

Additional Time Shift To Hardware

Zeitraum zur Anpassung der (additiv) automatisch ermittelten Totzeit für die Kompensation der Sollposition.

Actual Position Filter

Name	Standardwert	Einheit	Typ
Filter Type	None		MC.EFilterType [► 204]
Filter Time Position	0.0	s	LREAL
Filter Time Velocity	0.0	s	LREAL
Filter Time Acceleration	0.01	s	LREAL
Filter Time	0.001	s	LREAL

Filter Type

None

Es wird kein Filter angewendet.

PT1-Filter

Es handelt sich um ein Verzögerungsglied erster Ordnung (digitaler Tiefpassfilter, Proportionalglied mit Verzögerung 1. Ordnung) zur Glättung verrauschter Signale (Istwert). Der PT1-Filter wird **unabhängig** auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Filterzeitkonstante. Die Filterzeitkonstanten sind im positiven Wertebereich zu wählen. Der Wert 0.0 deaktiviert die Filterung der entsprechenden Größe. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung. Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander.

MovingAverage-Filter

Der Moving-Average-Filter bildet den arithmetischen Mittelwert über ein gleitendes Zeitfenster der letzten N Eingangswerte und ist damit ein einfaches FIR-Tiefpassfilter (mit endlicher Impulsantwort). Der Moving-Average-Filter wird unabhängig auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Fensterlänge N . Die Fensterlänge ergibt sich aus dem Quotienten der parametrisierten Filterzeit und der Zykluszeit. Es ist eine maximale Fensterlänge von 1000 Datenpunkte möglich. Die Totzeit entspricht der halben Fensterlänge. Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander. Eine Filterzeit von 0.0 wird der Istwert unmittelbar verarbeitet und nicht gefiltert. Eine große Filterzeit bewirkt eine starke Glättung und erhöht die Totzeit.

KalmanFilter3x1

Der Kalman-Filter 3x1 schätzt aus den fortlaufenden Positions-Istwerten einen Satz von Position (s), Geschwindigkeit (v) und Beschleunigung (a). Dies dient zum einen der Rauschunterdrückung und zum anderen findet die Berechnung von (s,v,a) konsistent, d.h. unter Berücksichtigung der Abhängigkeit der Größen untereinander, statt. 3x1 beschreibt dabei, dass der Filter drei Ausgangsgrößen (s,v,a) und eine Eingangsgröße (s) hat.

Die Konsistenz von (s,v,a) untereinander ist häufig von Vorteil. Insbesondere wird der Kalman-Filter bei einer Kopplung einer Achse auf ein Gebersignal empfohlen.

Der Kalman-Filter 3x1 wird durch die "Filter Time" parametrisiert, welche die Zeitskala beschreibt, die zur Berechnung von (s,v,a) berücksichtigt wird. Die Wahl dieses Parameters stellt einen Kompromiss zwischen Rauschunterdrückung und dynamischer Reaktionsfähigkeit dar. Eine größere Filter Time führt zu einer größeren Rauschunterdrückung bei gleichzeitiger Abnahme der Reaktionsfähigkeit, und umgekehrt (siehe unten Parameter *Filter Time*).

Filter Time Position

Der Parameter ist aktiv, wenn der Filter Type auf PT1 oder MovingAverage eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Position. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Velocity

Der Parameter ist aktiv, wenn der Filter Type auf PT1 oder MovingAverage eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Geschwindigkeit. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Acceleration

Der Parameter ist aktiv, wenn der Filter Type auf PT1 oder MovingAverage eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Beschleunigung. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time

Der Parameter ist aktiv, wenn der Filter Type auf KalmanFilter3x1 eingestellt ist. Der Kalman Filter wird durch eine vereinfachte Parametrierung über eine einzelne Filterzeitkonstante auf die Anwendung angepasst. Je größer die Filterzeitkonstante, desto höher ist der Grad der Glättung.

3.2.3.8.4.2 Online Parameter

Scaling

Name	Einheit	Typ
Work Zero Offset	[PosUnit]	LREAL
Active Homing Offset	[PosUnit]	LREAL

Work Zero Offset

Nicht-persistenter Offset, der über den Funktionsbaustein MC_SetPosition eingestellt werden kann. Wird MC_SetPosition mehrfach ausgeführt, werden die einzelnen Offsets unter Berücksichtigung des Vorzeichens verrechnet.

Der Offset ist zur Laufzeit aktiv und wird beim Neustart von TwinCAT zurückgesetzt.

$\text{Work Zero Offset} = \text{Vorgegebene neue Position} - \text{Aktuelle Sollposition}$

Active Homing Offset

Zeigt den temporären Versatz an, der während der letzten Referenzfahrt berechnet wurde. Dieser Wert wird nicht dauerhaft gespeichert.

TouchProbe1 Positive Edge Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert #Invalid signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

TouchProbe1 Negative Edge Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert #Invalid signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

3.2.3.8.4.3 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

SoE Drive / SoE Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Drive status word	UINT	Status-Informationen
☒	Position feedback value 1	DINT	Ist-Position des IO-Drives in Inkrementen.
☒	Following distance	DINT	Schleppfehler (Lageregelabweichung, d. h. Sollposition minus Istposition) des IO-Drives in Inkrementen.
☐	Velocity feedback value 1	DINT	Ist-Geschwindigkeit des IO-Drives in Inkrementen.
☒	Torque feedback value	INT	Drehmoment/Kraft-Istwert des IO-Drives in Inkrementen.
☒	WcState	BIT	Working Counter
☒	InputToggle	BIT	Pegelwechsel bei Erhalt eines neuen Telegramms.
☒	DcInputShift	DINT	Wird für die Totzeitkompensation des Antriebs verwendet.
☒	DcOutputShift	DINT	Wird für die Totzeitkompensation des Antriebs verwendet.

SoE Drive / SoE Touchprobe Inputs

In der Standardeinstellung ist dieser Datenbereich deaktiviert.

Enable (default)	Name	Typ	Beschreibung
☒	Probe value 1 positive edge	DINT	Dieses Element liefert den Positionswert des Messtasters 1 bei der steigenden Flanke.
☒	Probe value 1 negative edge	DINT	Dieses Element liefert den Positionswert der Messtasters 1 bei fallender Flanke.
☒	Probe 1 positive latched	UINT	Ereignis bei steigender Flanke für Messtaster 1 detektiert.
☒	Probe 1 negative latched	UINT	Ereignis bei fallender Flanke für Messtaster 1 detektiert.
☒	Probe 1 time positive edge	UDINT	Dieses Element liefert den Zeitstempel, während ein Ereignis mit steigender Flanke an Messtasters 1 erkannt wurde.
☒	Probe 1 time negative edge	UDINT	Dieses Element liefert den Zeitstempel, während ein Ereignis mit fallender Flanke an Messtasters 1 erkannt wurde.

SoE Drive / SoE Outputs

Enable (default)	Name	Typ	Beschreibung
☒	Master control word	WORD	Control-Informationen
☒	Position command value	DINT	Aktuelle Soll-Position in Inkrementen Die Soll-Position der MC3-Achse in physikalischen Einheiten, wie z. B. mm oder Grad, wird von der MC3 mathematisch in einen Inkrementalwert zurückgerechnet und zum Antrieb übertragen. Hierbei werden Überläufe des Inkrementalwertes in der Sollposition von der MC3 berücksichtigt.
☐	Velocity command value	DINT	Aktuelle Soll-Geschwindigkeit in Inkrementen Die Soll-Geschwindigkeit der MC3-Achse in physikalischen Einheiten, wie z. B. mm/s oder Grad/s, wird von der MC3 mathematisch in einen Inkrementalwert zurückgerechnet und zum Antrieb übertragen.

SoE Drive / SoE Touchprobe Outputs

In der Standardeinstellung ist dieser Datenbereich deaktiviert.

Enable (default)	Name	Typ	Beschreibung
☒	Probe 1 enable	UINT	Messtaster 1 einschalten.

HINWEIS**Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät**

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.8.5 CoE DS408 Drive (hydraulic)

Modul zur Verknüpfung einer MC3-Achse mit einem Antrieb mit dem Antriebsprofil CoE DS408 (hydraulic).

● Ansicht unterlagerter Objekte

i In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.8.5.1 Init Parameter

Scaling

Name	Standardwert	Einheit	Typ
Velocity Scaling Numerator To Hardware	1.0	INC/([PosUnit]/s)	LREAL
Velocity Scaling Denominator To Hardware	1.0		LREAL
Direction Inverted	FALSE		BOOL
Output Zero Offset	0.0	%	LREAL

Velocity Scaling Numerator To Hardware

Velocity Scaling Numerator To Hardware wird verwendet, um den Geschwindigkeitssoll- und -istwert aus den Encoderumdrehungen zu berechnen.

Velocity Scaling Denominator To Hardware

Velocity Scaling Denominator To Hardware wird verwendet, um den Geschwindigkeitssoll- und -istwert aus der Geberzahl zu berechnen.

Invert Direction

Invert Direction gibt an, ob die Geschwindigkeitswerte von und zur Hardware invertiert werden sollen.

Output Zero Offset

Output Zero Offset gibt den Ausgang bei Geschwindigkeit Null im Verhältnis (Prozent) zum maximalen Ausgangsbereich an (+/-32767 für Analogantrieb).

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0.0:0		
.netId	0.0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Dead Time Compensation

Voraussetzung für die hochgenaue Umrechnung von Positionen in Zeiten und umgekehrt ist eine exakte Totzeitkompensation der Achsen.

Name	Standardwert	Einheit	Typ
Dead Time Compensation	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]

Name	Standardwert	Einheit	Typ
Additional Time Shift To Hardware	0.0	ms	LREAL

Dead Time Compensation

Stellen Sie ein, wie die kompensierte Soll- und Istposition berechnet werden soll. Die Totzeit wird bei der EtherCAT-Initialisierung automatisch ermittelt.

Additional Time Shift To Hardware

Zusätzliche Verschiebung der automatisch ermittelten Totzeit für die Kompensation der Sollposition.

3.2.3.8.5.2 Online Parameter

Name	Einheit	Typ
Spool Set Position		LREAL
Spool Act Position		LREAL

Spool Set Position

Sollwert der Spulenposition ((-)1,0 = volle Leistung)

Spool Act Position

Aktueller Wert der Spulenposition ((-)1,0 = volle Leistung)

3.2.3.8.5.3 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

CoE DS408 Drive (hydraulic) / DS408 Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Status word	UINT	
☒	Valve Act Spool Position	INT	
☒	WcState	BIT	
☒	InputToggle	BIT	

CoE DS408 Drive (hydraulic) / DS408 Outputs

Enable (default)	Name	Typ	Beschreibung
☒	Control word	UINT	Control-Informationen
☒	Valve Set Spool Position	INT	

HINWEIS**Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät**

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.8.6 Analog Drive

Modul zur Verknüpfung einer MC3-Achse mit einem analogen Antrieb.

**Ansicht unterlagerter Objekte**

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.8.6.1 Init Parameter**Scaling**

Name	Standardwert	Einheit	Typ
Velocity Scaling Numerator To Hardware	1.0	INC/([PosUnit]/s)	LREAL
Velocity Scaling Denominator To Hardware	1.0		LREAL
Direction Inverted	FALSE		BOOL
Output Zero Offset	0.0	%	LREAL

Velocity Scaling Numerator To Hardware

`Velocity Scaling Numerator To Hardware` wird verwendet, um den analogen Ausgangswert auf Basis des Geschwindigkeitssollwerts zu berechnen.

Velocity Scaling Denominator To Hardware

`Velocity Scaling Denominator To Hardware` wird verwendet, um den analogen Ausgangswert auf Basis des Geschwindigkeitssollwerts zu berechnen.

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung des Encoders oder Antriebs umkehren:

- FALSE: Die Polarität der Achsbewegung stimmt mit der Zählrichtung der Erfassungshardware überein.
- TRUE: Die Polarität der Achsbewegung ist invers zur Zählrichtung der Erfassungshardware.

⚠️ WARNUNG**Verletzungsgefahr durch unerwartete Bewegungen**

Wenn die Zählrichtung des Encoders und die Motorpolarität nicht miteinander übereinstimmen, führt die Achse unerwartete Bewegungen aus.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Output Zero Offset

Output Zero Offset gibt den Ausgang bei Geschwindigkeit Null im Verhältnis (Prozent) zum maximalen Ausgangsbereich an (+-32767 für Analogantrieb).

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0:0		
.netId	0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Dead Time Compensation

Voraussetzung für die hochgenaue Umrechnung von Positionen in Zeiten und umgekehrt ist eine exakte Totzeitkompensation der Achsen.

Name	Standardwert	Einheit	Typ
Dead Time Compensation	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift To Hardware	0.0	ms	LREAL

Dead Time Compensation

Stellen Sie ein, wie die kompensierte Soll- und Istposition berechnet werden soll. Die Totzeit wird bei der EtherCAT-Initialisierung automatisch ermittelt.

Additional Time Shift To Hardware

Zusätzliche Verschiebung der automatisch ermittelten Totzeit für die Kompensation der Sollposition.

3.2.3.8.6.2 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

Analog Drive / Analog Inputs

Enable (default)	Name	Typ	Beschreibung
☒	WcState	BIT	Working Counter

Analog Drive / Analog Outputs

Enable (default)	Name	Typ	Beschreibung
☒	Output	INT	

HINWEIS**Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät**

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.8.7 PulseTrain Drive

Modul zur Verknüpfung einer MC3-Achse mit einem Antrieb mit dem Antriebsprofil PulseTrain.

● Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.8.7.1 Init Parameter**Scaling**

Name	Standardwert	Einheit	Typ
Position Scaling Numerator From Hardware	1.0	[PosUnit]/INC	LREAL
Position Scaling Denominator From Hardware	1.0		LREAL
Direction Inverted	FALSE		BOOL
Encoder Mounting Offset	0.0	[PosUnit]	LREAL

Der Positions-Skalierungsfaktor ist in einen Zähler und einen Nenner gegliedert. Mit ihm werden die Weginkremente in Achsenpositionen umgerechnet oder eine Benutzereinheit aus Encoderinkrementen berechnet.

Position Scaling Numerator From Hardware

Position Scaling Numerator From Hardware ist der Vorschub, den die Applikation macht, wenn die Abtriebswelle eine Umdrehung gemacht hat.

Position Scaling Denominator From Hardware

Position Scaling Denominator From Hardware ist die Anzahl an Inkrementen, die der Antrieb ausgibt, wenn die Motorwelle eine Umdrehung macht.

Beispiel 1 zum Position Scaling:

Motor ohne Getriebe mit Ritzel mit 100 mm Umfang an AX5000 mit Standardeinstellungen:

- Numerator: 100 mm
- Denominator: 1048576

Beispiel 2 zum Position Scaling:

Motor mit Getriebe mit $i=10$ an Drehteller an AX5000 mit Standardeinstellungen:

- Numerator: $360^\circ / 10 = 36^\circ$
- Denominator: 1048576

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung des Encoders oder Antriebs umkehren:

- `FALSE`: Die Polarität der Achsbewegung stimmt mit der Zählrichtung der Erfassungshardware überein.
- `TRUE`: Die Polarität der Achsbewegung ist invers zur Zählrichtung der Erfassungshardware.

WARNUNG

Verletzungsgefahr durch unerwartete Bewegungen

Wenn die Zählrichtung des Encoders und die Motorpolarität nicht miteinander übereinstimmen, führt die Achse unerwartete Bewegungen aus.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Encoder Mounting Offset

Offset, mit dem die Position eines Gebersystem korrigiert bzw. im Maschinenkoordinatensystem ausgerichtet wird und damit den maschinenbezogenen Nullpunkt festlegt. Bei einem Neustart der Steuerung wird dieser Offset wiederhergestellt.

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0.0:0		
.netId	0.0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Dead Time Compensation

Voraussetzung für die hochgenaue Umrechnung von Positionen in Zeiten und umgekehrt ist eine exakte Totzeitkompensation der Achsen.

Name	Standardwert	Einheit	Typ
Dead Time Compensation	None		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift To Hardware	0.0	ms	LREAL

Dead Time Compensation

Legen Sie fest, wie der kompensierte Sollwert und der Istwert berechnet werden sollen. Die Totzeit wird während der EtherCAT-Initialisierung automatisch ermittelt.

Dead Time Compensation To Hardware

Zeitraum zur Anpassung der automatisch ermittelten Totzeit für die Sollwertkompensation.

Actual Position Filter

Name	Standardwert	Einheit	Typ
Filter Type	None		MC.EFilterType [► 204]
Filter Time Position	0.0	s	LREAL
Filter Time Velocity	0.0	s	LREAL
Filter Time Acceleration	0.01	s	LREAL
Filter Time	0.001	s	LREAL

Filter Type

None

Es wird kein Filter angewendet.

PT1-Filter

Es handelt sich um ein Verzögerungsglied erster Ordnung (digitaler Tiefpassfilter, Proportionalglied mit Verzögerung 1. Ordnung) zur Glättung verrauschter Signale (Istwert). Der PT1-Filter wird **unabhängig** auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Filterzeitkonstante. Die Filterzeitkonstanten sind im positiven Wertebereich zu wählen. Der Wert 0.0 deaktiviert die Filterung der entsprechenden Größe. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander.

MovingAverage-Filter

Der Moving-Average-Filter bildet den arithmetischen Mittelwert über ein gleitendes Zeitfenster der letzten N Eingangswerte und ist damit ein einfaches FIR-Tiefpassfilter (mit endlicher Impulsantwort).

Der Moving-Average-Filter wird unabhängig auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Fensterlänge N . Die Fensterlänge ergibt sich aus dem Quotienten der parametrisierten Filterzeit und der Zykluszeit. Es ist eine maximale Fensterlänge von 1000 Datenpunkte möglich. Die Totzeit entspricht der halben Fensterlänge.

Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander. Eine Filterzeit von 0.0 wird der Istwert unmittelbar verarbeitet und nicht gefiltert. Eine große Filterzeit bewirkt eine starke Glättung und erhöht die Totzeit.

KalmanFilter3x1

Der Kalman-Filter 3x1 schätzt aus den fortlaufenden Positions-Istwerten einen Satz von Position (s), Geschwindigkeit (v) und Beschleunigung (a). Dies dient zum einen der Rauschunterdrückung und zum anderen findet die Berechnung von (s,v,a) konsistent, d.h. unter Berücksichtigung der Abhängigkeit der

Größen untereinander, statt. 3x1 beschreibt dabei, dass der Filter drei Ausgangsgrößen (s,v,a) und eine Eingangsgröße (s) hat.

Die Konsistenz von (s,v,a) untereinander ist häufig von Vorteil. Insbesondere wird der Kalman-Filter bei einer Kopplung einer Achse auf ein Gebersignal empfohlen.

Der Kalman-Filter 3x1 wird durch die "Filter Time" parametrisiert, welche die Zeitskala beschreibt, die zur Berechnung von (s,v,a) berücksichtigt wird. Die Wahl dieses Parameters stellt einen Kompromiss zwischen Rauschunterdrückung und dynamischer Reaktionsfähigkeit dar. Eine größere Filter Time führt zu einer größeren Rauschunterdrückung bei gleichzeitiger Abnahme der Reaktionsfähigkeit, und umgekehrt (siehe unten Parameter *Filter Time*).

Filter Time Position

Der Parameter ist aktiv, wenn der *Filter Type* auf *PT1* oder *MovingAverage* eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Position. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Velocity

Der Parameter ist aktiv, wenn der *Filter Type* auf *PT1* oder *MovingAverage* eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Geschwindigkeit. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Acceleration

Der Parameter ist aktiv, wenn der *Filter Type* auf *PT1* oder *MovingAverage* eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Beschleunigung. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time

Der Parameter ist aktiv, wenn der *Filter Type* auf *KalmanFilter3x1* eingestellt ist. Der Kalman Filter wird durch eine vereinfachte Parametrierung über eine einzelne Filterzeitkonstante auf die Anwendung angepasst. Je größer die Filterzeitkonstante, desto höher ist der Grad der Glättung.

3.2.3.8.7.2 Online Parameter

Scaling

Name	Einheit	Typ
Work Zero Offset	[PosUnit]	LREAL
Active Homing Offset	[PosUnit]	LREAL

Work Zero Offset

Nicht-persistenter Offset, der über den Funktionsbaustein MC_SetPosition eingestellt werden kann. Wird MC_SetPosition mehrfach ausgeführt, werden die einzelnen Offsets unter Berücksichtigung des Vorzeichens verrechnet.

Der Offset ist zur Laufzeit aktiv und wird beim Neustart von TwinCAT zurückgesetzt.

Work Zero Offset = Vorgegebene neue Position - Aktuelle Sollposition

Active Homing Offset

Zeigt den temporären Versatz an, der während der letzten Referenzfahrt berechnet wurde. Dieser Wert wird nicht dauerhaft gespeichert.

3.2.3.8.7.3 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

PulseTrain Drive / PulseTrain Inputs

Enable (default)	Name	Typ	Beschreibung
☒	WcState	BIT	
☒	InputToggle	BIT	
☒	Counter value	UDINT	
☒	ENC Sync error	BIT	
☒	ENC TxPDO State	BIT	
☒	ENC TxPDO Toggle	BIT	
☒	DcInputShift	DINT	
☒	DcOutputShift	DINT	

PulseTrain Drive / PulseTrain Outputs

Enable (default)	Name	Typ	Beschreibung
☒	PTO Automatic direction	BIT	
☒	Target counter value	UDINT	

HINWEIS

Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.8.8 Pulse Width Current Drive (MDP250) (both channels)

Modul zur Verknüpfung einer MC3-Achse mit einem Antrieb mit dem Antriebsprofil Pulse Width Current Drive (MDP250) (both channels).

i Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.8.8.1 Init Parameter

Scaling

Name	Standardwert	Einheit	Typ
Velocity Scaling Numerator To Hardware	1.0	INC/([PosUnit]/s)	LREAL
Velocity Scaling Denominator To Hardware	1.0		LREAL
Direction Inverted	FALSE		BOOL
Output Zero Offset	0.0	%	LREAL

Velocity Scaling Numerator To Hardware

Velocity Scaling Numerator To Hardware wird verwendet, um den analogen Ausgangswert auf Basis des Geschwindigkeitssollwerts zu berechnen.

Velocity Scaling Denominator To Hardware

Velocity Scaling Denominator To Hardware wird verwendet, um den analogen Ausgangswert auf Basis des Geschwindigkeitssollwerts zu berechnen.

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung des Encoders oder Antriebs umkehren:

- FALSE: Die Polarität der Achsbewegung stimmt mit der Zählrichtung der Erfassungshardware überein.
- TRUE: Die Polarität der Achsbewegung ist invers zur Zählrichtung der Erfassungshardware.

WARNUNG

Verletzungsgefahr durch unerwartete Bewegungen

Wenn die Zählrichtung des Encoders und die Motorpolarität nicht miteinander übereinstimmen, führt die Achse unerwartete Bewegungen aus.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Output Zero Offset

Output Zero Offset gibt den Ausgang bei Geschwindigkeit Null im Verhältnis (Prozent) zum maximalen Ausgangsbereich an (+/-32767 für Analogantrieb).

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0:0		
.netId	0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Dead Time Compensation

Voraussetzung für die hochgenaue Umrechnung von Positionen in Zeiten und umgekehrt ist eine exakte Totzeitkompensation der Achsen.

Name	Standardwert	Einheit	Typ
Dead Time Compensation	None		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift To Hardware	0.0	ms	LREAL

Dead Time Compensation

Legen Sie fest, wie der kompensierte Sollwert und der Istwert berechnet werden sollen. Die Totzeit wird während der EtherCAT-Initialisierung automatisch ermittelt.

Dead Time Compensation To Hardware

Zeitraum zur Anpassung der automatisch ermittelten Totzeit für die Sollwertkompensation.

3.2.3.8.8.2 Online Parameter**Functionality**

Name	Einheit	Typ
Dither Enabled		BOOL

Aktiviert das Dither-Signal des Antriebs, um Hysterese- und Haftreibungseffekte zu verringern.

3.2.3.8.8.3 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

Pulse Width Current Drive (MDP250) (both channels) / PWM Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Channel 1 Ready to enable	BIT	
☒	Channel 1 Warning	BIT	
☒	Channel 1 Error	BIT	
☒	Channel 2 Ready to enable	BIT	
☒	Channel 2 Warning	BIT	
☒	Channel 2 Error	BIT	
☒	WcState	BIT	
☒	InputToggle	BIT	

Pulse Width Current Drive (MDP250) (both channels) / PWM Outputs

Enable (default)	Name	Typ	Beschreibung
☒	Channel 1 Enable dithering	BIT	
☒	Channel 1 Enable	BIT	
☒	Channel 1 Reset	BIT	
☒	Channel 1 PWM Output	INT	
☒	Channel 2 Enable dithering	BIT	
☒	Channel 2 Enable	BIT	
☒	Channel 2 Reset	BIT	
☒	Channel 2 PWM Output	INT	

HINWEIS**Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät**

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.9 Encoder-Module

Im folgenden Abschnitt sind Informationen über die verfügbaren Encoder Module enthalten.

3.2.3.9.1 EtherCAT Encoder

Modul zur Verknüpfung einer MC3-Achse mit einem EtherCAT Encoder.

● Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.9.1.1 Init Parameter**General**

Name	Standardwert	Einheit	Typ
Position Interpretation	Incremental		MC.EEncoderInterpretation [► 203]
Maximum Absolute Counter Value	1048575	INC	UDINT

Name	Standardwert	Einheit	Typ
(Sichtbarkeit abhängig von Position Interpretation)			
Position Data Range	FullRange		MC.EEncoderDataRange [► 203]
Minimum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 0 • vorzeichenbehaftet: -2147483648	INC	LINT
Maximum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 4294967295 • vorzeichenbehaftet: 2147483647	INC	LINT
Feedback Velocity Source	HardwareProvidedIfAvailable		MC.EEncoderFeedbackVeloSource [► 203]

Position Interpretation

Der Parameter `Position Interpretation` definiert, nach welcher Semantik die Werte vom Positionsgeber übermittelt und von der MC3-Achse interpretiert werden.

Incremental

Die Einstellung `Incremental` nutzt keine absoluten Eigenschaften des physikalischen Feedbacksystems, kann aber sowohl mit inkrementellen als auch absoluten Gebersystemen verwendet werden. Nach dem Start der Steuerung ist eine Referenzfahrt (Homing) notwendig, um die Ist-Position der Achse zu initialisieren (`IsPositionValid = FALSE`).

AbsoluteFullRange

Die Einstellung `AbsoluteFullRange` wird in der Regel mit Multiturn- oder Singleturn-Absolutwertgebern verwendet. Die MC3-Achse berücksichtigt den gesamten Datenbereich des Datentyps im Mapping zum I/O-Gerät (FullRange). Die höchste Anzahl an Inkrementen entspricht der Obergrenzen des Datentyps. Die Größe des Datentyps ist im verwendeten Protokoll spezifiziert.

Zum Zeitpunkt des Einschaltens der Steuerung ist der Positionswert eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`. Kommt es zu einem Überlauf, signalisiert die Achse `IsPositionValid = FALSE`. Eine (erneute) Referenzfahrt ist notwendig, um die Ist-Position der Achse zu initialisieren.

Das Konfigurieren und Einschalten der softwareseitigen Positionsgrenzen der Achse ermöglicht das Wiederherstellen der Achsposition nach einem Neustart der Steuerung. Eine erneute Referenzfahrt ist nicht notwendig. Dies gilt für maximal einen Überlauf der übermittelten Inkremente. Kommt es zu einem weiteren Überlauf, signalisiert die Achse `IsPositionValid = FALSE`.

AbsolutUserDataRange

Es gelten die gleichen Randbedingungen wie für die Einstellung `AbsoluteFullRange`. Zusätzlich wird der Parameter `Maximum Absolute Counter Value` eingeblendet und berücksichtigt:

- `Maximum Absolute Counter Value`
Eindeutige maximale Anzahl an Inkrementen, die das Gebersystem übermittelt (innerhalb der Grenzen des verwendeten Datentyps). Beispielsweise ist der Standardwert auf $2^{20}-1$ eingestellt.

AbsoluteModulo

Die Positionsinkremente werden vom I/O-Gerät mit Modulo Logik übermittelt. Entspricht z. B. eine Umdrehung des Gebersystems dem Modulus der MC3-Achse, ist die Position in diesem Bereich eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`.

Position Data Range

Der Parameter `Position Data Range` definiert den Wertebereich des Gebersystems, der für die Positionsberechnung verwendet wird. Dies bezieht sich auf den Rohwert, der vom Antrieb oder Encoder über die Prozessdaten (PDO-Mapping) empfangen wird und ermöglicht die Erkennung und Handhabung von Überläufen. Der Parameter kann nur zur Konfigurationszeit und nicht zur Laufzeit verändert werden.

FullRange

Der Positionsüberlauf basiert auf den Datentypgrenzen des jeweiligen Gebersystems. Für ein vorzeichenbehaftetes 32-Bit-Gebersystem (z. B. CoE DS402 Drive (AX8000), SoE Drive (AX5000), EtherCAT Encoder) bedeutet dies implizit einen Wertebereich von -2147483648 bis 2147483647. Bei einem vorzeichenlosen 32-Bit-Gebersystem (z. B. MDP742 Drive (ausgewählte ELM72xx), SSI-Encoder) impliziert dies einen Wertebereich von 0 bis 4294967295.

UserDefined

Dieser Parameterwert ermöglicht eine manuelle Eingabe der Wertebereichsgrenzen für die Bestimmung eines Überlaufs. Es werden zwei zusätzliche Parameter zur Eingabe der oberen und unteren Grenze eingeblendet:

- Minimum Counter Value
- Maximum Counter Value

Feedback Velocity Source

Bestimmt die Quelle der Ist-Geschwindigkeit (`AXIS_REF.McToPlc.Act.Velocity`). Die Geschwindigkeit kann entweder von der verwendeten Hardware übernommen oder intern durch Differenzierung des Positions-Istwertes berechnet werden.

HardwareProvidedIfAvailable (Standardwert)

Verwendet die von der Hardware bereitgestellte Geschwindigkeit. Ist kein Eintrag für die Ist-Geschwindigkeit in den Prozessdaten gemappt, wird diese automatisch intern berechnet.

InternallyCalculated

Die Geschwindigkeit wird immer aus der Ableitung der Ist-Positionen berechnet. Ist-Geschwindigkeiten der Hardware werden ignoriert, auch wenn sie verfügbar sind.

HardwareProvided

Die Geschwindigkeit wird immer direkt vom Hardware-Feedback übernommen. Dies setzt eine Verknüpfung der Ist-Geschwindigkeit über die Prozessdaten (PDO-Mapping) voraus. Fehlt das Mapping, führt dies zu einem Fehler beim Aufstarten der Konfiguration.

Scaling

Name	Standardwert	Einheit	Typ
Position Scaling Numerator From Hardware	360	[PosUnit]/INC	LREAL
Position Scaling Denominator From Hardware	1048576.0		LREAL
Encoder Mounting Offset	0.0	[PosUnit]	LREAL
Direction Inverted	FALSE		BOOL
Velocity Scaling Numerator From Hardware	1.0	[PosUnit]/(INC*s)	LREAL
Velocity Scaling Denominator From Hardware	1.0		LREAL

Der Positions-Skalierungsfaktor ist in einen Zähler und einen Nenner gegliedert. Mit ihm werden die Weginkremente in Achsenpositionen umgerechnet oder eine Benutzereinheit aus Encoderinkrementen berechnet.

Position Scaling Numerator From Hardware

Position Scaling Numerator From Hardware ist der Vorschub, den die Applikation macht, wenn die Abtriebswelle eine Umdrehung gemacht hat.

Position Scaling Denominator From Hardware

Position Scaling Denominator From Hardware ist die Anzahl an Inkrementen, die der Antrieb ausgibt, wenn die Motorwelle eine Umdrehung macht.

Beispiel 1 zum Position Scaling:

Motor ohne Getriebe mit Ritzel mit 100 mm Umfang an AX5000 mit Standardeinstellungen:

- Numerator: 100 mm
- Denominator: 1048576

Beispiel 2 zum Position Scaling:

Motor mit Getriebe mit $i=10$ an Drehteller an AX5000 mit Standardeinstellungen:

- Numerator: $360^\circ / 10 = 36^\circ$
- Denominator: 1048576

Encoder Mounting Offset

Offset, mit dem die Position eines Gebersystem korrigiert bzw. im Maschinenkoordinatensystem ausgerichtet wird und damit den maschinenbezogenen Nullpunkt festlegt. Bei einem Neustart der Steuerung wird dieser Offset wiederhergestellt.

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung des Encoders oder Antriebs umkehren:

- `FALSE`: Die Polarität der Achsbewegung stimmt mit der Zählrichtung der Erfassungshardware überein.
- `TRUE`: Die Polarität der Achsbewegung ist invers zur Zählrichtung der Erfassungshardware.

WARNUNG

Verletzungsgefahr durch unerwartete Bewegungen

Wenn die Zählrichtung des Encoders und die Motorpolarität nicht miteinander übereinstimmen, führt die Achse unerwartete Bewegungen aus.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Velocity Scaling Numerator From Hardware

Wird verwendet, um den tatsächlichen Geschwindigkeitswert aus den Encoder-Impulsen zu berechnen.

Velocity Scaling Denominator From Hardware

Wird verwendet, um den tatsächlichen Geschwindigkeitswert aus den Encoder-Impulsen zu berechnen.

Dead Time Compensation

Voraussetzung für die hochgenaue Umrechnung von Positionen in Zeiten und umgekehrt ist eine exakte Totzeitkompensation der Achsen.

Name	Standardwert	Einheit	Typ
Dead Time Compensation	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]

Name	Standardwert	Einheit	Typ
Additional Time Shift From Hardware	0.0	ms	LREAL

Dead Time Compensation

Stellen Sie ein, wie der kompensierte Soll- und Istwert berechnet werden soll. Die Totzeit wird bei der EtherCAT-Initialisierung automatisch ermittelt.

Additional Time Shift From Hardware

Zusätzliche Verschiebung der automatisch ermittelten Totzeit für die Kompensation der aktuellen Position.

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0.0:0		
.netId	0.0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Status Mask

Name	Standardwert	Einheit	Typ
Status Word Error Mask	0x0008		WORD

Actual Position Filter

Name	Standardwert	Einheit	Typ
Filter Type	None		MC.FilterType [► 204]
Filter Time Position	0.0	s	LREAL
Filter Time Velocity	0.0	s	LREAL
Filter Time Acceleration	0.01	s	LREAL
Filter Time	0.001	s	LREAL

Filter Type

None

Es wird kein Filter angewendet.

PT1-Filter

Es handelt sich um ein Verzögerungsglied erster Ordnung (digitaler Tiefpassfilter, Proportionalglied mit Verzögerung 1. Ordnung) zur Glättung verrauschter Signale (Istwert). Der PT1-Filter wird **unabhängig** auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Filterzeitkonstante. Die Filterzeitkonstanten sind im positiven Wertebereich zu wählen. Der Wert 0.0 deaktiviert die Filterung der entsprechenden Größe. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung. Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander.

MovingAverage-Filter

Der Moving-Average-Filter bildet den arithmetischen Mittelwert über ein gleitendes Zeitfenster der letzten N Eingangswerte und ist damit ein einfaches FIR-Tiefpassfilter (mit endlicher Impulsantwort). Der Moving-Average-Filter wird unabhängig auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Fensterlänge N . Die Fensterlänge ergibt sich aus dem Quotienten der parametrisierten Filterzeit und der Zykluszeit. Es ist eine maximale Fensterlänge von 1000 Datenpunkte möglich. Die Totzeit entspricht der halben Fensterlänge. Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander. Eine Filterzeit von 0.0 wird der Istwert unmittelbar verarbeitet und nicht gefiltert. Eine große Filterzeit bewirkt eine starke Glättung und erhöht die Totzeit.

KalmanFilter3x1

Der Kalman-Filter 3x1 schätzt aus den fortlaufenden Positions-Istwerten einen Satz von Position (s), Geschwindigkeit (v) und Beschleunigung (a). Dies dient zum einen der Rauschunterdrückung und zum anderen findet die Berechnung von (s,v,a) konsistent, d.h. unter Berücksichtigung der Abhängigkeit der Größen untereinander, statt. 3x1 beschreibt dabei, dass der Filter drei Ausgangsgrößen (s,v,a) und eine Eingangsgröße (s) hat. Die Konsistenz von (s,v,a) untereinander ist häufig von Vorteil. Insbesondere wird der Kalman-Filter bei einer Kopplung einer Achse auf ein Gebersignal empfohlen. Der Kalman-Filter 3x1 wird durch die "Filter Time" parametrisiert, welche die Zeitskala beschreibt, die zur Berechnung von (s,v,a) berücksichtigt wird. Die Wahl dieses Parameters stellt einen Kompromiss zwischen Rauschunterdrückung und dynamischer Reaktionsfähigkeit dar. Eine größere Filter Time führt zu einer größeren Rauschunterdrückung bei gleichzeitiger Abnahme der Reaktionsfähigkeit, und umgekehrt (siehe unten Parameter *Filter Time*).

Filter Time Position

Der Parameter ist aktiv, wenn der Filter Type auf PT1 oder MovingAverage eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Position. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Velocity

Der Parameter ist aktiv, wenn der Filter Type auf PT1 oder MovingAverage eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Geschwindigkeit. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Acceleration

Der Parameter ist aktiv, wenn der Filter Type auf PT1 oder MovingAverage eingestellt ist. Der Wert 0.0 deaktiviert die Filterung der Beschleunigung. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time

Der Parameter ist aktiv, wenn der Filter Type auf KalmanFilter3x1 eingestellt ist. Der Kalman Filter wird durch eine vereinfachte Parametrierung über eine einzelne Filterzeitkonstante auf die Anwendung angepasst. Je größer die Filterzeitkonstante, desto höher ist der Grad der Glättung.

3.2.3.9.1.2 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

EtherCAT Encoder Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Position	DINT	
☐	Velocity	DINT	
☒	Status	WORD	
☒	WcState	BIT	
☒	InputToggle	BIT	
☐	DcInputShift	DINT	

3.2.3.9.2 Analog Load Encoder

Modul zur Verknüpfung einer MC3-Achse mit einem Analog Load Encoder.



Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.9.2.1 Init Parameter

General

Name	Standardwert	Einheit	Typ
Load Type	None		MC.ELoadType [► 205]

Load Type

Legt fest, welche Lastgröße dieser Encoder misst (z. B. Force, PressureA, PressureB).

Scaling

Name	Standardwert	Einheit	Typ
Direction Inverted	FALSE		BOOL
Load Scaling Factor Numerator From Hardware	0.001	[[LoadUnit]/INC]	LREAL
Load Scaling Factor Denominator From Hardware	1.0		LREAL
Load Offset	0.0	[LoadUnit]	LREAL

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung des Encoders oder Antriebs umkehren:

- `FALSE`: Die Polarität der Achsbewegung stimmt mit der Zählrichtung der Erfassungshardware überein.
- `TRUE`: Die Polarität der Achsbewegung ist invers zur Zählrichtung der Erfassungshardware.

⚠ WARNUNG

Verletzungsgefahr durch unerwartete Bewegungen

Wenn die Zählrichtung des Encoders und die Motorpolarität nicht miteinander übereinstimmen, führt die Achse unerwartete Bewegungen aus.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Load Scaling Factor Numerator From Hardware

Dieser Wert dient zur Umwandlung von Sensorinkrementen in den tatsächlichen Lastwert.

Load Scaling Factor Denominator From Hardware

Dieser Wert dient zur Umwandlung von Sensorinkrementen in den tatsächlichen Lastwert.

Load Offset

Offset zur Festlegung eines maschinenbezogenen Nullpunkts.

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0:0		
.netId	0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Actual Load Filter

Name	Standardwert	Einheit	Typ
Filter Type	None		MC.FilterType [► 204]
Filter Time Load	0.0	s	LREAL
Filter Time Load Derivative	0.005	s	LREAL
Filter Time Load Second Derivative	0.01	s	LREAL
Filter Time	0.001	s	LREAL

Filter Type

Definiert den Filteralgorithmus. Folgende Typen stehen zur Verfügung:

- None: Es wird kein Filter angewendet.
- PT1: Ein Tiefpassfilter erster Ordnung.
- MovingAverage: Gibt den Mittelwert des Eingangssignals über die Filterzeit aus.
- KalmanFilter3x1: FIR-Tiefpassfilter

Filter Time Load

Steigende positive Werte erhöhen die Glättung und Verzögerung der Istlast; Null bedeutet keine Filterung.

Filter Time Load Derivative

Steigende positive Werte erhöhen die Glättung und Verzögerung der Ableitung der Istlast; Null bedeutet keine Filterung.

Filter Time Load Second Derivative

Steigende positive Werte erhöhen die Glättung und Verzögerung der zweiten Ableitung der Istlast; Null bedeutet keine Filterung.

Filter Time

Filterzeit für den Kalman-Filter. Der Wert wird intern quadriert, um die Kovarianz des Beobachtungsrauschens (R_sigma_s) zu erhalten. Größere Werte erzeugen ein glatteres Ausgangssignal, kleinere Werte folgen dem gemessenen Signal genauer.

3.2.3.9.2 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

Analog Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Underrange	BIT	
☒	Overrange	BIT	
☒	Error	BIT	
☒	Counter value	INT	
☒	WcState	BIT	

3.2.3.9.3 SSI Encoder

Modul zur Verknüpfung einer MC3-Achse mit einem SSI Encoder.

● Ansicht unterlagerter Objekte

i In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.9.3.1 Init Parameter

General

Name	Standardwert	Einheit	Typ
Position Interpretation	Incremental		MC.EEncoderInterpretation [► 203]
Maximum Absolute Counter Value (Sichtbarkeit abhängig von Position Interpretation)	1048575	INC	UDINT
Position Data Range	FullRange		MC.EEncoderDataRange [► 203]
Maximum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 4294967295 • vorzeichenbehaftet: 2147483647	INC	LINT
Minimum Counter Value (Sichtbarkeit abhängig von Position Data Range)	32-Bit-Gebersystem • vorzeichenlos: 0 • vorzeichenbehaftet: -2147483648	INC	LINT

Position Interpretation

Der Parameter `Position Interpretation` definiert, nach welcher Semantik die Werte vom Positionsgeber übermittelt und von der MC3-Achse interpretiert werden.

`Incremental`

Die Einstellung `Incremental` nutzt keine absoluten Eigenschaften des physikalischen Feedbacksystems, kann aber sowohl mit inkrementellen als auch absoluten Gebersystemen verwendet werden. Nach dem Start der Steuerung ist eine Referenzfahrt (Homing) notwendig, um die Ist-Position der Achse zu initialisieren (`IsPositionValid = FALSE`).

`AbsoluteFullRange`

Die Einstellung `AbsoluteFullRange` wird in der Regel mit Multiturn- oder Singleturn-Absolutwertgebern verwendet. Die MC3-Achse berücksichtigt den gesamten Datenbereich des Datentyps im Mapping zum I/O-Gerät (`FullRange`). Die höchste Anzahl an Inkrementen entspricht der Obergrenzen des Datentyps. Die Größe des Datentyps ist im verwendeten Protokoll spezifiziert.

Zum Zeitpunkt des Einschaltens der Steuerung ist der Positionswert eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`. Kommt es zu einem Überlauf, signalisiert die Achse `IsPositionValid = FALSE`. Eine (erneute) Referenzfahrt ist notwendig, um die Ist-Position der Achse zu initialisieren.

Das Konfigurieren und Einschalten der softwareseitigen Positionsgrenzen der Achse ermöglicht das Wiederherstellen der Achsposition nach einem Neustart der Steuerung. Eine erneute Referenzfahrt ist nicht notwendig. Dies gilt für maximal einen Überlauf der übermittelten Inkremente. Kommt es zu einem weiteren Überlauf, signalisiert die Achse `IsPositionValid = FALSE`.

`AbsolutUserDataRange`

Es gelten die gleichen Randbedingungen wie für die Einstellung `AbsoluteFullRange`. Zusätzlich wird der Parameter `Maximum Absolute Counter Value` eingeblendet und berücksichtigt:

- `Maximum Absolute Counter Value`
Eindeutige maximale Anzahl an Inkrementen, die das Gebersystem übermittelt (innerhalb der Grenzen des verwendeten Datentyps). Beispielsweise ist der Standardwert auf $2^{20}-1$ eingestellt.

AbsoluteModulo

Die Positionsinkremente werden vom I/O-Gerät mit Modulo Logik übermittelt. Entspricht z. B. eine Umdrehung des Gebersystems dem Modulus der MC3-Achse, ist die Position in diesem Bereich eindeutig. Die Achse signalisiert `IsPositionValid = TRUE`.

Position Data Range

Der Parameter `Position Data Range` definiert den Wertebereich des Gebersystems, der für die Positionsberechnung verwendet wird. Dies bezieht sich auf den Rohwert, der vom Antrieb oder Encoder über die Prozessdaten (PDO-Mapping) empfangen wird und ermöglicht die Erkennung und Handhabung von Überläufen. Der Parameter kann nur zur Konfigurationszeit und nicht zur Laufzeit verändert werden.

FullRange

Der Positionsüberlauf basiert auf den Datentypgrenzen des jeweiligen Gebersystems. Für ein vorzeichenbehaftetes 32-Bit-Gebersystem (z. B. CoE DS402 Drive (AX8000), SoE Drive (AX5000), EtherCAT Encoder) bedeutet dies implizit einen Wertebereich von -2147483648 bis 2147483647. Bei einem vorzeichenlosen 32-Bit-Gebersystem (z. B. MDP742 Drive (ausgewählte ELM72xx), SSI-Encoder) impliziert dies einen Wertebereich von 0 bis 4294967295.

UserDefined

Dieser Parameterwert ermöglicht eine manuelle Eingabe der Wertebereichsgrenzen für die Bestimmung eines Überlaufs. Es werden zwei zusätzliche Parameter zur Eingabe der oberen und unteren Grenze eingeblendet:

- Minimum Counter Value
- Maximum Counter Value

Scaling

Name	Standardwert	Einheit	Typ
Position Scaling Numerator From Hardware	360	[PosUnit]/INC	LREAL
Position Scaling Denominator From Hardware	1048576.0		LREAL
Encoder Mounting Offset	0.0	[PosUnit]	LREAL
Direction Inverted	FALSE		BOOL

Der Positions-Skalierungsfaktor ist in einen Zähler und einen Nenner gegliedert. Mit ihm werden die Weginkremente in Achsenpositionen umgerechnet oder eine Benutzereinheit aus Encoderinkrementen berechnet.

Position Scaling Numerator From Hardware

Position Scaling Numerator From Hardware ist der Vorschub, den die Applikation macht, wenn die Abtriebswelle eine Umdrehung gemacht hat.

Position Scaling Denominator From Hardware

Position Scaling Denominator From Hardware ist die Anzahl an Inkrementen, die der Antrieb ausgibt, wenn die Motorwelle eine Umdrehung macht.

Beispiel 1 zum Position Scaling:

Motor ohne Getriebe mit Ritzel mit 100 mm Umfang an AX5000 mit Standardeinstellungen:

- Numerator: 100 mm
- Denominator: 1048576

Beispiel 2 zum Position Scaling:

Motor mit Getriebe mit $i=10$ an Drehteller an AX5000 mit Standardeinstellungen:

- Numerator: $360^\circ / 10 = 36^\circ$
- Denominator: 1048576

Encoder Mounting Offset

Offset, mit dem die Position eines Gebersystem korrigiert bzw. im Maschinenkoordinatensystem ausgerichtet wird und damit den maschinenbezogenen Nullpunkt festlegt. Bei einem Neustart der Steuerung wird dieser Offset wiederhergestellt.

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung des Encoders oder Antriebs umkehren:

- `FALSE`: Die Polarität der Achsbewegung stimmt mit der Zählrichtung der Erfassungshardware überein.
- `TRUE`: Die Polarität der Achsbewegung ist invers zur Zählrichtung der Erfassungshardware.

WARNUNG

Verletzungsgefahr durch unerwartete Bewegungen

Wenn die Zählrichtung des Encoders und die Motorpolarität nicht miteinander übereinstimmen, führt die Achse unerwartete Bewegungen aus.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Dead Time Compensation

Voraussetzung für die hochgenaue Umrechnung von Positionen in Zeiten und umgekehrt ist eine exakte Totzeitkompensation der Achsen.

Name	Standardwert	Einheit	Typ
Dead Time Compensation	NonlinearShift		MC.EDeadTimeCompensationMode [► 202]
Additional Time Shift From Hardware	0.0	ms	LREAL

Dead Time Compensation

Stellen Sie ein, wie der kompensierte Soll- und Istwert berechnet werden soll. Die Totzeit wird bei der EtherCAT-Initialisierung automatisch ermittelt.

Additional Time Shift From Hardware

Zusätzliche Verschiebung der automatisch ermittelten Totzeit für die Kompensation der aktuellen Position.

Connection

Name	Standardwert	Einheit	Typ
AMS Address	0.0.0.0.0:0		
.netId	0.0.0.0.0		AMSNETID
.port	0x0000		WORD
Channel Number	0		UDINT
EtherCAT Slave Oid	00000000		OTCID

AMS Address

Informationen über die Adresse des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

Channel Number

Informationen über Kanalnummer des zugeordneten Hardware-Laufwerks (nur Lesezugriff), es ist keine manuelle Konfiguration erforderlich. Sie wird für Kommunikationszwecke verwendet.

EtherCAT Slave Oid

Objekt ID des EtherCAT Slaves.

Actual Position Filter

Name	Standardwert	Einheit	Typ
Filter Type	None		MC.FilterType [► 204]
Filter Time Position	0.0	s	LREAL
Filter Time Velocity	0.0	s	LREAL
Filter Time Acceleration	0.01	s	LREAL
Filter Time	0.001	s	LREAL

Filter Type

None

Es wird kein Filter angewendet.

PT1-Filter

Es handelt sich um ein Verzögerungsglied erster Ordnung (digitaler Tiefpassfilter, Proportionalglied mit Verzögerung 1. Ordnung) zur Glättung verrauschter Signale (Istwert). Der PT1-Filter wird **unabhängig** auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Filterzeitkonstante. Die Filterzeitkonstanten sind im positiven Wertebereich zu wählen. Der Wert 0.0 deaktiviert die Filterung der entsprechenden Größe. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander.

MovingAverage-Filter

Der Moving-Average-Filter bildet den arithmetischen Mittelwert über ein gleitendes Zeitfenster der letzten N Eingangswerte und ist damit ein einfaches FIR-Tiefpassfilter (mit endlicher Impulsantwort).

Der Moving-Average-Filter wird unabhängig auf die drei Istwerte Position, Geschwindigkeit und Beschleunigung angewendet. Jede Größe besitzt eine eigene Fensterlänge N . Die Fensterlänge ergibt sich aus dem Quotienten der parametrisierten Filterzeit und der Zykluszeit. Es ist eine maximale Fensterlänge von 1000 Datenpunkte möglich. Die Totzeit entspricht der halben Fensterlänge.

Da die Größen getrennt gefiltert werden, sind die Ausgangswerte nicht zwangsläufig konsistent zueinander. Eine Filterzeit von 0.0 wird der Istwert unmittelbar verarbeitet und nicht gefiltert. Eine große Filterzeit bewirkt eine starke Glättung und erhöht die Totzeit.

KalmanFilter3x1

Der Kalman-Filter 3x1 schätzt aus den fortlaufenden Positions-Istwerten einen Satz von Position (s), Geschwindigkeit (v) und Beschleunigung (a). Dies dient zum einen der Rauschunterdrückung und zum anderen findet die Berechnung von (s,v,a) konsistent, d.h. unter Berücksichtigung der Abhängigkeit der Größen untereinander, statt. 3x1 beschreibt dabei, dass der Filter drei Ausgangsgrößen (s,v,a) und eine Eingangsgröße (s) hat.

Die Konsistenz von (s,v,a) untereinander ist häufig von Vorteil. Insbesondere wird der Kalman-Filter bei einer Kopplung einer Achse auf ein Gebersignal empfohlen.

Der Kalman-Filter 3x1 wird durch die "Filter Time" parametrisiert, welche die Zeitskala beschreibt, die zur Berechnung von (s,v,a) berücksichtigt wird. Die Wahl dieses Parameters stellt einen Kompromiss zwischen Rauschunterdrückung und dynamischer Reaktionsfähigkeit dar. Eine größere Filter Time führt zu einer größeren Rauschunterdrückung bei gleichzeitiger Abnahme der Reaktionsfähigkeit, und umgekehrt (siehe unten Parameter *Filter Time*).

Filter Time Position

Der Parameter ist aktiv, wenn der `Filter Type` auf `PT1` oder `MovingAverage` eingestellt ist. Der Wert `0.0` deaktiviert die Filterung der Position. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Velocity

Der Parameter ist aktiv, wenn der `Filter Type` auf `PT1` oder `MovingAverage` eingestellt ist. Der Wert `0.0` deaktiviert die Filterung der Geschwindigkeit. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time Acceleration

Der Parameter ist aktiv, wenn der `Filter Type` auf `PT1` oder `MovingAverage` eingestellt ist. Der Wert `0.0` deaktiviert die Filterung der Beschleunigung. Die Filterzeitkonstante ist im positiven Wertebereich zu wählen. Je größer die Filterzeitkonstante, desto stärker wird der Istwert geglättet und desto größer ist die Verzögerung.

Filter Time

Der Parameter ist aktiv, wenn der `Filter Type` auf `KalmanFilter3x1` eingestellt ist. Der Kalman Filter wird durch eine vereinfachte Parametrierung über eine einzelne Filterzeitkonstante auf die Anwendung angepasst. Je größer die Filterzeitkonstante, desto höher ist der Grad der Glättung.

3.2.3.9.3.2 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

SSI Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Counter Value	UDINT	
☒	Data Error	BIT	
☒	Frame Error	BIT	
☒	Power Failure	BIT	
☒	Sync Error	BIT	
☒	TxPDO State	BIT	
☒	TxPDO Toggle	BIT	
☒	WcState	BIT	
☒	InputToggle	BIT	
☐	DcInputShift	DINT	

3.2.3.10 Controller-Module

Im folgenden Abschnitt sind Informationen über die verfügbaren Controller-Module enthalten.

3.2.3.10.1 Load Controller



Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.10.1.1 Init Parameter

General

Name	Standardwert	Einheit	Typ
Control Value Type	Velocity		MC.EControlValueType [▶ 201]

Control Value Type

Legt den Typ der Ausgangsvariablen des Reglers fest. Zum Beispiel bedeutet *Velocity*, dass der Reglerausgang eine Geschwindigkeit ist.

Control Parameter

Name	Standardwert	Einheit	Typ
Output Inverted	FALSE		BOOL
Proportional Gain (Kp)	0.0	[ControlVariableUnit]/ [LoadUnit]	LREAL
Derivative Gain (Kd)	0.0	[ControlVariableUnit]/ ([LoadUnit]/s)	LREAL
Integral Gain (Ki)	0.0	[ControlVariableUnit]/ ([LoadUnit]*s)	LREAL
Integrator Lag Window Minimum	#Ignore	[LoadUnit]	LREAL
Integrator Lag Window Maximum	#Ignore	[LoadUnit]	LREAL

Output Inverted

Der Parameter *Output Inverted* kann das Vorzeichen der Ausgangsgröße umkehren.

⚠️ WARNUNG

Verletzungsgefahr durch unerwartete Bewegungen

Wenn die Ausgangsgröße des Reglers nicht auf die Regelstrecke abgestimmt ist, kann die Achse eine unerwartete Bewegung ausführen.

- Stellen Sie sicher, dass bei Inbetriebnahme die Drehrichtung des Antriebs korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

Proportional Gain

Die proportionale Verstärkung (Kp) wird mit der aktuellen Regelabweichung multipliziert, was einen Term der Regelgröße ergibt.

Derivative Gain

Die abgeleitete Verstärkung (Kd) wird mit der Ableitung des Fehlers multipliziert, was einen Term der Regelgröße ergibt.

Integral Gain

Die integrale Verstärkung (Ki) wird mit dem Integral der Regelabweichung multipliziert, was zu einem Term der Regelgröße führt.

Integrator Lag Window Minimum

Wenn die tatsächliche Lastverzögerung kleiner ist als das Integrator Lag Window Minimum, ändert sich der Integrator nicht. Der Wert muss größer oder gleich 0,0 und kleiner oder gleich Integrator Lag Window Maximum sein.

Integrator Lag Window Maximum

Wenn die tatsächliche Lastverzögerung größer ist als das Integrator Lag Window Maximum, ändert sich der Integrator nicht. Der Wert muss größer oder gleich 0,0 sein.

Output Limitation

Name	Standardwert	Einheit	Typ
Integrator Minimum Output	#Ignore	[ControlVariableUnit]	LREAL
Integrator Maximum Output	#Ignore	[ControlVariableUnit]	LREAL
Minimum Output	#Ignore	[ControlVariableUnit]	LREAL
Maximum Output	#Ignore	[ControlVariableUnit]	LREAL

Integrator Minimum Output

Minimaler Ausgang des Integrators. Der Wert Integrator Minimum Output muss kleiner oder gleich dem Wert Integrator Maximum Output sein.

Integrator Maximum Output

Maximale Leistung des Integrators. Der Wert Integrator Maximum Output muss größer oder gleich dem Wert Integrator Minimum Output sein.

Minimum Output

Minimaler Ausgang des Controllers. Der Wert Minimum Output muss kleiner oder gleich dem Maximum Output sein.

Maximum Output

Maximaler Ausgang des Controllers. Der Wert Maximum Output muss größer oder gleich dem Minimum Output sein.

3.2.3.10.1.2 Online Parameter

Name	Einheit	Typ
Control Variable	[ControlVariableUnit]	LREAL
Control Variable Proportional Part	[ControlVariableUnit]	LREAL
Control Variable Derivative Part	[ControlVariableUnit]	LREAL
Control Variable Integral Part	[ControlVariableUnit]	LREAL

Control Variable

Die Regelgröße zum Erreichen des gewünschten Sollwerts.

Control Variable Proportional Part

Der proportionale Anteil der Regelgröße.

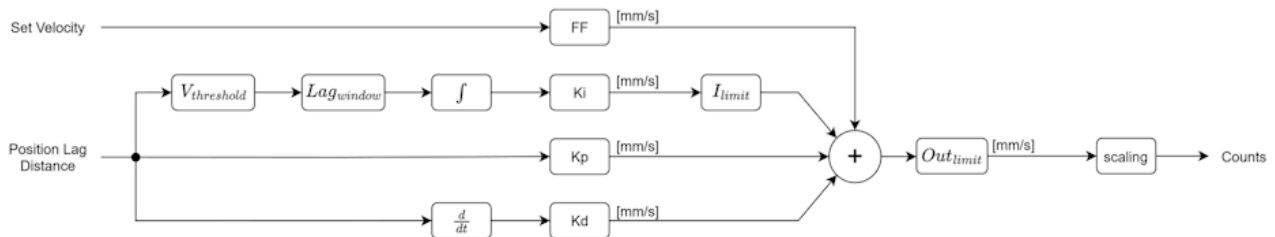
Control Variable Derivative Part

Der abgeleitete Teil der Regelgröße.

Control Variable Integral Part

Der integrale Teil der Steuergröße.

3.2.3.10.2 Position Controller



Bezeichnung	Erläuterung
FF	Feed Forward - Die Sollgröße der Geschwindigkeit wird mit diesem Faktor multipliziert und zur Gesamtausgabe des Reglers addiert.
$V_{\text{threshold}}$	Wenn der Istwert der Geschwindigkeit größer als der Parameter „ Integrator Threshold Maximum Velocity [► 98] “ oder kleiner als der „ Integrator Threshold Minimum Velocity [► 98] “ ist, wird die Regelerkomponente des Integrators nicht mehr angepasst. Der „ Integrator Threshold Minimum Velocity [► 98] “ ist in der Regel ein negativer Wert.
Lag_{window}	Wenn die Größe des Positions-Schleppfehlers kleiner als der Parameter „ Integrator Lag Window Minimum [► 98] “ oder größer als „ Integrator Lag Window Maximum [► 98] “ ist, wird die Regelerkomponente des Integrators nicht mehr angepasst. „ Integrator Lag Window Minimum [► 98] “ und „ Integrator Lag Window Maximum [► 98] “ müssen positiv sein.
I_{Limit}	Limitierung für den Integral-Anteil des Reglers
Out_{Limit}	Limitierung für die Ausgangsgröße des Reglers



Ansicht unterlagter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.10.2.1 Init Parameter

General

Name	Standardwert	Einheit	Typ
Control Value Type	Velocity		MC.EControlValueType [► 201]
Enabled	TRUE		BOOL
Velocity Feed Forward Factor	1.0		LREAL

Control Value Type

Legt den Typ der Ausgangsvariablen des Reglers fest. Zum Beispiel bedeutet `velocity`, dass der Reglerausgang eine Geschwindigkeit ist.

Enabled

`Enabled` zeigt an, ob der Lageregler verwendet wird oder nicht. Der Reglerausgang ist 0,0, wenn er deaktiviert ist.

Velocity Feed Forward Factor

Das Produkt aus Velocity Feed Forward Factor und dem Geschwindigkeitssollwert wird zur Summe von P, I und D addiert und ergibt die Regelgröße.

Control Parameter

Name	Standardwert	Einheit	Typ
Proportional Gain (Kp)	1.0	[ControlVariableUnit] / [PosUnit]	LREAL
Derivative Gain (Kd)	0.0	[ControlVariableUnit] / ([PosUnit] / s)	LREAL
Integral Gain (Ki)	0.0	[ControlVariableUnit] / ([PosUnit] * s)	LREAL
Integrator Threshold Minimum Velocity	#Ignore	[PosUnit]/s	LREAL
Integrator Threshold Maximum Velocity	#Ignore	[PosUnit]/s	LREAL
Integrator Lag Window Minimum	#Ignore	[PosUnit]	LREAL
Integrator Lag Window Maximum	#Ignore	[PosUnit]	LREAL

Proportional Gain (Kp)

Die proportionale Verstärkung (Kp) wird mit der aktuellen Regelabweichung multipliziert, was einen Term der Regelgröße ergibt.

Derivative Gain (Kd)

Die abgeleitete Verstärkung (Kd) wird mit der Ableitung des Fehlers multipliziert, was einen Term der Regelgröße ergibt.

Integral Gain (Ki)

Die integrale Verstärkung (Ki) wird mit dem Integral der Regelabweichung multipliziert, was zu einem Term der Regelgröße führt.

Integrator Threshold Minimum Velocity

Der Integrator ändert sich nicht, wenn die tatsächliche Geschwindigkeit kleiner als die Mindestgeschwindigkeit ist. Die Mindestgeschwindigkeit muss kleiner oder gleich 0,0 sein.

`Integrator Threshold Minimum Velocity ≤ 0,0`

Integrator Threshold Maximum Velocity

Der Integrator ändert sich nicht, wenn die aktuelle Geschwindigkeit größer als die maximale Geschwindigkeit ist. Die Maximalgeschwindigkeit muss größer als 0,0 sein.

`Integrator Threshold Maximum Velocity > 0,0`

Integrator Lag Window Minimum

Der Integrator ändert sich nicht, wenn die tatsächliche Verzögerung außerhalb dieses Intervalls liegt. Der Wert für die minimale Verzögerung des Integrators muss größer als 0,0 sein.

`Integrator Lag Window Minimum > 0,0`

Integrator Lag Window Maximum

Der Integrator ändert sich nicht, wenn die tatsächliche Verzögerung außerhalb dieses Intervalls liegt. Der Wert für die maximale Verzögerung des Integrators muss größer sein als die minimale Verzögerung.

`Integrator Lag Window Maximum > Integrator Lag Window Minimum`

Output Limitation

Name	Standardwert	Einheit	Typ
Integrator Minimum Output	#Ignore	[ControlVariableUnit]	LREAL
Integrator Maximum Output	#Ignore	[ControlVariableUnit]	LREAL
Minimum Output	#Ignore	[ControlVariableUnit]	LREAL
Maximum Output	#Ignore	[ControlVariableUnit]	LREAL

Integrator Minimum Output

Minimaler Ausgang des Integrators. Der Wert `Integrator Minimum Output` muss kleiner oder gleich dem Wert `Integrator Maximum Output` sein.

Integrator Maximum Output

Maximale Leistung des Integrators. Der Wert `Integrator Maximum Output` muss größer oder gleich dem Wert `Integrator Minimum Output` sein.

Minimum Output

Minimaler Ausgang des Controllers. Der Wert `Minimum Output` muss kleiner oder gleich dem `Maximum Output` sein.

Maximum Output

Maximaler Ausgang des Controllers. Der Wert `Maximum Output` muss größer oder gleich dem `Minimum Output` sein.

3.2.3.10.2.2 Online Parameter

Name	Einheit	Typ
Control Value Final Output	[ControlVariableUnit]	LREAL
Control Value	[ControlVariableUnit]	LREAL
Control Variable Proportional Part	[ControlVariableUnit]	LREAL
Control Variable Derivative Part	[ControlVariableUnit]	LREAL
Control Variable Integral Part	[ControlVariableUnit]	LREAL

Control Value Final Output

Summe aus P-Teil, I-Teil, D-Teil und FF-Teil.

Control Value

Die Regelgröße zum Erreichen des gewünschten Sollwerts.

Control Variable Proportional Part

Der proportionale Anteil der Regelgröße.

Control Variable Derivative Part

Der abgeleitete Teil der Regelgröße.

Control Variable Integral Part

Der integrale Teil der Steuergröße.

3.2.3.11 FluidPower-Module

Im Folgenden sind die Bestandteile des FluidPower-Modules beschrieben.

3.2.3.11.1 FluidPower Axis Parameters

Parameter der FluidPower Axis.

● Ansicht unterlagerter Objekte

i In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.11.1.1 Init Parameter

Cylinder Dimensions

Name	Standardwert	Einheit	Typ
Cylinder Design	Differential		MC.ECylinderDesign
Cylinder Bore Diameter	#Ignore	mm	LREAL
Cylinder Rod Diameter	#Ignore	mm	LREAL

Cylinder Design

Der Parameter `Cylinder Design` wählt den Typ des Hydraulikzylinders aus und legt fest, wie Kolben- und Kolbenstangendurchmesser für Kraft- und Flächenberechnungen verwendet werden.

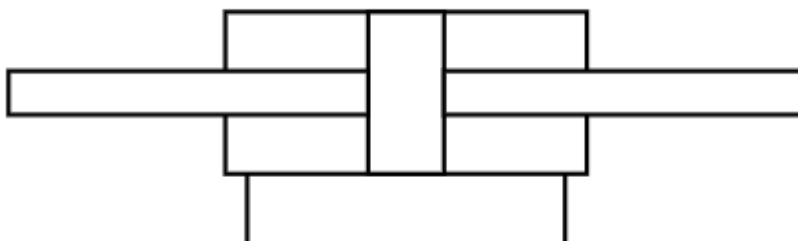
`Differential`

Der Differentialzylinder ist ein doppeltwirkender Hydraulikzylinder mit einseitiger Kolbenstange.



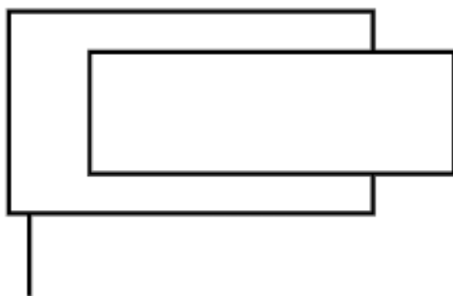
`Synchronous`

Der Synchronzylinder ist ein doppeltwirkender Hydraulikzylinder mit durchgehender Kolbenstange.



`Plunger`

Der Plungerzylinder ist ein einfachwirkender Hydraulikzylinder ohne Kolbenstange.



Cylinder Bore Diameter

Innendurchmesser der Hydraulikzylinderbohrung. Wird zur Berechnung der Kolbenfläche und der Kraft herangezogen.

Cylinder Rod Diameter

Durchmesser der Zylinderstange. Relevant für die Konstruktion von Differential- und Synchronzylindern zur Berechnung der effektiven Kolbenfläche auf der Stangenseite.

3.2.3.11.2 Valve Characteristic Curve

Einstellungen und Parameter an der Valve Characteristic Curve.



Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.11.2.1 Init Parameter

Name	Standardwert	Einheit	Typ
Valve Linearization Enabled	TRUE		BOOL

Dieser Parameter ermöglicht die Linearisierung des Antriebsausgangs, um das nichtlineare Verhalten des Ventils auszugleichen.

3.2.3.11.2.2 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

Outputs

Name	Typ	Beschreibung
McToPlc	OTCID	

3.2.3.12 Touch Probe-Module

Die Verwendung von Touch Probe-Modulen ermöglicht die Erfassung einer Achsposition zum Zeitpunkt eines Trigger-Events. Dies kann ein digitales Signal sein und die Funktion eines Messtasters abbilden. Die Ermittlung eines Zeitstempels, zu dem die Achse eine definierte Position erreicht, ist ebenfalls möglich.

Touch Probe Modul hinzufügen

Mit den nachfolgenden Schritten können *zusätzliche* Touch Probe-Module hinzugefügt werden. Drive-Module, wie z. B. ein DS402 (AX8xxx), MDP742 (ELM72xx) oder Sercos (AX5xxx) verfügen bereits über Touch Probe-Funktionen. Es müssen keine weiteren Touch Probe-Module hinzugefügt werden, lediglich müssen die zyklischen Prozessdaten aktiviert und mit dem entsprechenden I/O-Gerät verknüpft werden (manuell oder durch Konfiguration im Drive Manager). In der PLC kann eine entsprechende `TRIGGER_REF` (abgeleitete Ref-Objekte beachten) verknüpft werden.

1. Machen Sie einen Rechtsklick auf **Axes > Axis > Add New Item...**

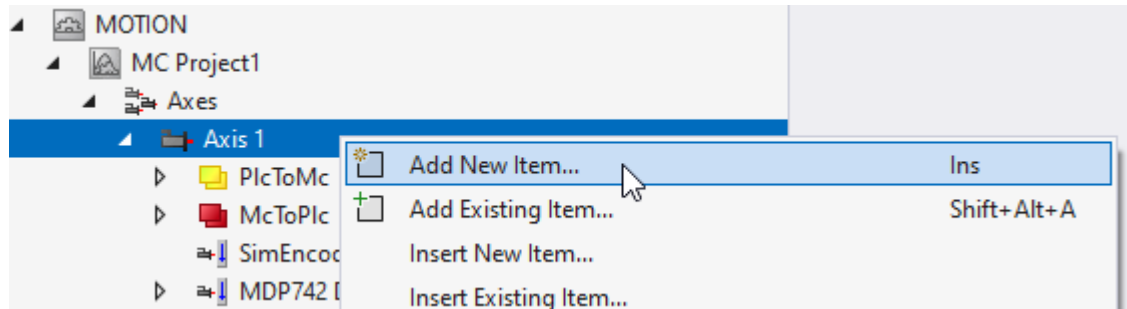
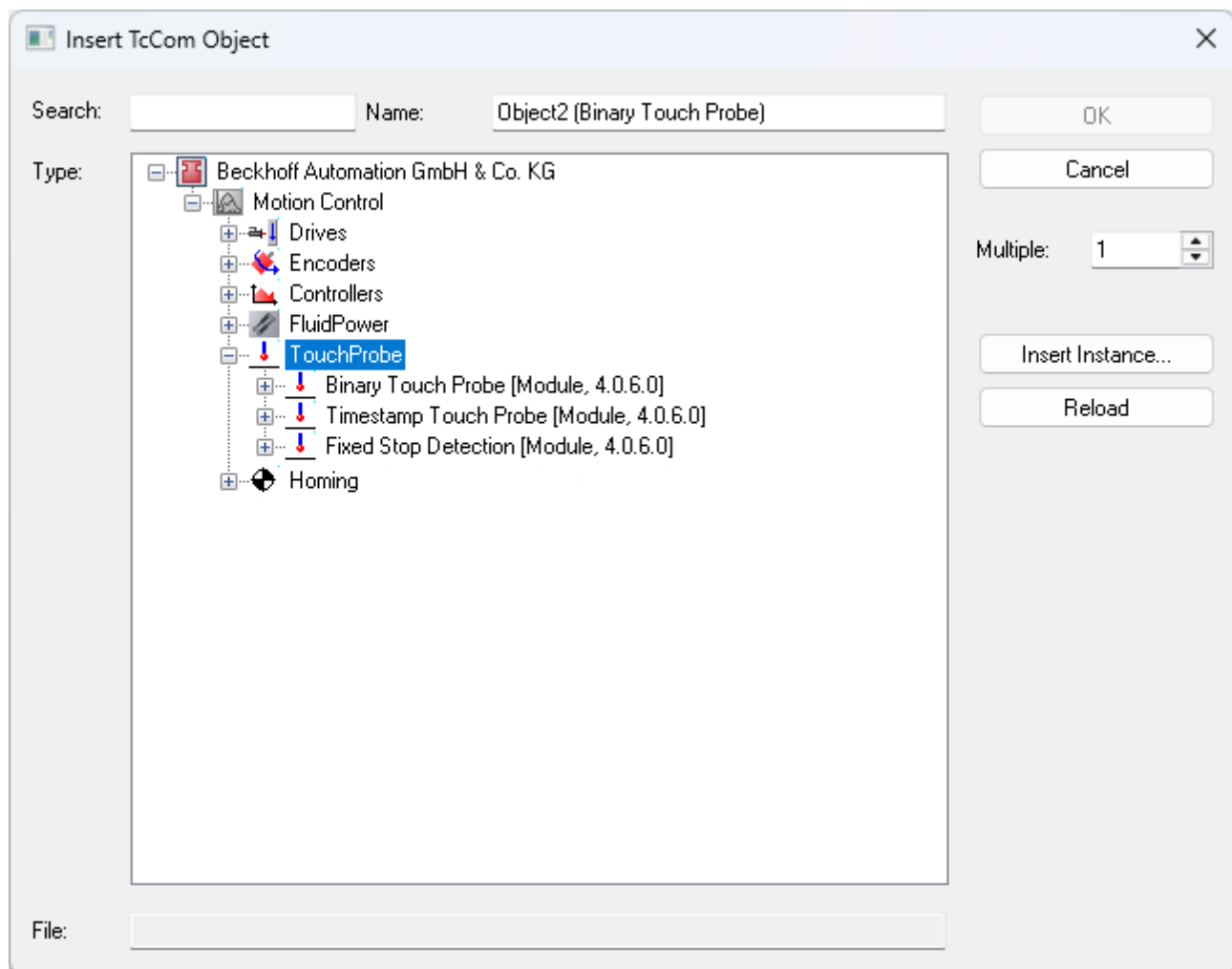


Abb. 1:

2. Navigieren Sie im Dialog **Insert TcCom Object** unter **Type** und der Gruppe **Motion Control** zur Gruppe **TouchProbe**.
3. Klappen Sie diese Gruppe ggf. auf.



4. Wählen Sie ein entsprechendes Touch Probe-Modul aus und schließen den Vorgang mit einem Klick auf den Button **OK** ab. Achten Sie darauf, das Modul in der gleichen Version auszuwählen, die für bereits vorhandene Motion-Control-Objekte in der Solution verwendet wird.

3.2.3.12.1 Binary Touch Probe

Mit dem Modul Binary Touch Probe wird eine Achsposition zum Zeitpunkt eines digitalen Signals (Messtasterfunktion) erfasst. Die Position wird üblicherweise nicht direkt in der PLC-Umgebung, sondern über ein externes Hardware-Signal erfasst. Die Auswertung des Signaleingangs erfolgt im Kontext der MET-Task [► 19] der Achse.

● Ansicht unterlagerter Objekte

I In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.12.1.1 Init Parameter

Binary Touch Probe

Name	Standardwert	Einheit	Typ
Input Inverted	FALSE		BOOL

Invert Input

Ist der Wert des Parameters `Input Inverted` auf `TRUE` gesetzt, invertiert die interne Algorithmik der MC die Logik des Eingangs Signal [► 105] für die Auswertung der Touch Probe Einheit.

3.2.3.12.1.2 Online Parameter

Positive Edge Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `#Invalid` signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `0` signalisiert, dass noch kein Ereignis erfasst wurde.

Negative Edge Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `#Invalid` signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

3.2.3.12.1.3 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Signal	BIT	Eingang für das Triggersignal. Er kann beispielsweise mit einer Variablen aus der PLC oder mit einem digitalen Eingang einer I/O-Klemme verknüpft werden.
☐	WcState	BIT	Working Counter

ToPlc

Enable (default)	Name	Typ	Beschreibung
☒	Oid	OTCID	Objekt ID zur Verknüpfung mit einem Touch Probe Objekt in der PLC

HINWEIS**Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät**

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.12.2 Timestamp Touch Probe

Mit dem Modul Timestamp Touch Probe wird eine Achsposition zum Zeitpunkt eines applikativen vorgegebenen Zeitstempels ermittelt.

**Ansicht unterlagerter Objekte**

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.12.2.1 Online Parameter

Touch Probe Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `#Invalid` signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

3.2.3.12.2.2 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Timestamp 32bit	UDINT	Zeitstempel, dessen Wertänderungen in der Touch Probe Algorithmik ausgewertet werden.
☐	Timestamp 64bit	ULINT	Zeitstempel, dessen Wertänderungen in der Touch Probe Algorithmik ausgewertet werden.
☐	WcState	BIT	Working Counter

HINWEIS

Warnung im Fehlerprotokollfenster durch Mehrfachauswahl von Eingangselemente

Sind sowohl der 32-Bit als auch 64-Bit Eingang für den Zeitstempel aktiviert und verknüpft, führt dies zu einer Warnung im Fehlerprotokollfenster.

- Wählen Sie entsprechend der verwendeten Hardware bzw. des zu verknüpfenden Zeitstempels (vor allem gemäß dessen Datentyp) ein Zeitstemple-Element.

ToPlc

Enable (default)	Name	Typ	Beschreibung
☒	Oid	OTCID	Objekt ID zur Verknüpfung mit einem Touch Probe Objekt in der PLC

HINWEIS**Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät**

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.12.3 Fixed Stop Detection

Mit dem Modul Fixed Stop Detection wird eine Achsposition u.a. zum Zeitpunkt des Erreichens eines Drehmoment/Kraft-Schwellwertes ermittelt. Referenzgröße ist der Drehmoment/Kraft-Eingang des Antriebsobjekts der Achse. Die Auswertung des Drehmoments/der Kraft erfolgt im Kontext der MET-Task [► 19] der Achse.

● Ansicht unterlagerter Objekte

i In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.12.3.1 Init Parameter**Detection Thresholds**

Name	Standardwert	Einheit	Typ
Torque/Force Threshold	10.0	%	LREAL
Velocity Threshold	10.0	[PosUnit]/s	LREAL
Lag Threshold	2.0	[PosUnit]	LREAL

Torque/Force Threshold

Schwellwert für den Istwert des Motordrehmoments. Wenn das Drehmoment diesen Wert während der Auswertung überschreitet, dann ist dies ein Maß für die Erkennung eines Festanschlags. Der Wert #Ignore deaktiviert diesen Schwellwert.

Velocity Threshold

Schwellwert für den Istwert der Achsgeschwindigkeit während der Auswertung. Die Kraft-/Drehmomentschwelle wird nur dann ausgewertet, wenn der Wert der Ist-Geschwindigkeit unterhalb dieses Schwellwerts liegt. Der Wert #Ignore deaktiviert diesen Schwellwert.

Lag Threshold

Schwellwert für den Positionsschleppfehler. Wenn der tatsächliche Schleppfehler diesen Wert während der Auswertung überschreitet, dann ist dies ein Maß für die Erkennung eines Festanschlags. Der Wert #Ignore deaktiviert diesen Schwellwert.

● Deaktivierung aller Schwellwerte

i Werden alle Schwellwerte deaktiviert, meldet das Modul sofort ein erkanntes Event. Im Anwendungsfall des Homing wird dies instantan als ein Erreichen des Endanschlags erkannt.

- Geben Sie mindestens für den Schwellwert für Drehmoment/Kraft oder Schleppfehler eine reelle Zahl ein, um eine Erkennung eines Festanschlags auszuführen.



Verhalten einer beschleunigenden Achse mit einem Geschwindigkeitsschwellwert

Beschleunigt die Achse z. B. aus dem Stillstand heraus und die Auswertung des Touch Probe Moduls ist bereits aktiv, liegt die Geschwindigkeit der Achse u. U. noch unterhalb des eingestellten Schwellwerts. In Abhängigkeit zu den weiteren Schwellwerten kann dies dazu führen, dass das Touch Probe Modul ein erkanntes Event meldet. Im Anwendungsfall des Homing kann dies als ein Erreichen des Endanschlags erkannt werden.

- Starten Sie ggf. erst eine Achsbewegung bevor die Touch Probe Einheit aktiviert wird.

3.2.3.12.3.2 Online Parameter

Touch Probe Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `#Invalid` signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

3.2.3.12.3.3 Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

ToPlc

Enable (default)	Name	Typ	Beschreibung
☒	Oid	OTCID	Objekt ID zur Verknüpfung mit einem Touch Probe Objekt in der PLC

HINWEIS

Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

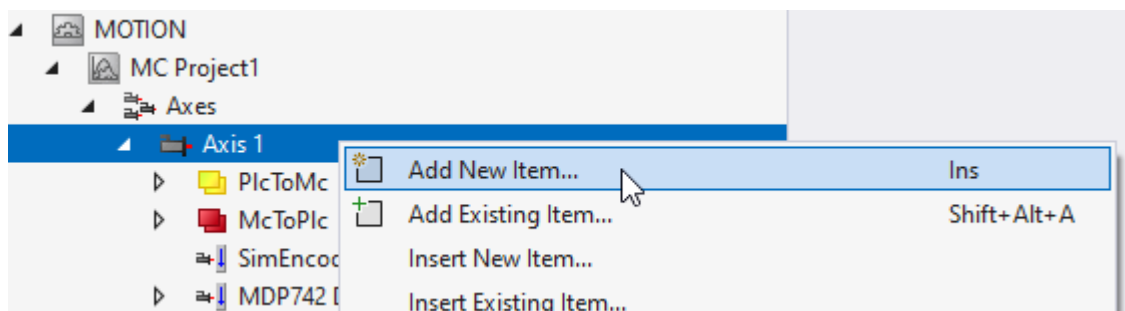
3.2.3.13 Homing-Module

Homing bezeichnet einen Referenzierungsvorgang einer Achse, bei der die korrekte Istposition anhand eines Referenzsignals ermittelt wird. Der Vorgang kann mithilfe der MC durchgeführt werden. Eine erfolgreich referenzierte Achse signalisiert im zyklischen Interface `McToPlc.Std.IsPositionValid = TRUE`.

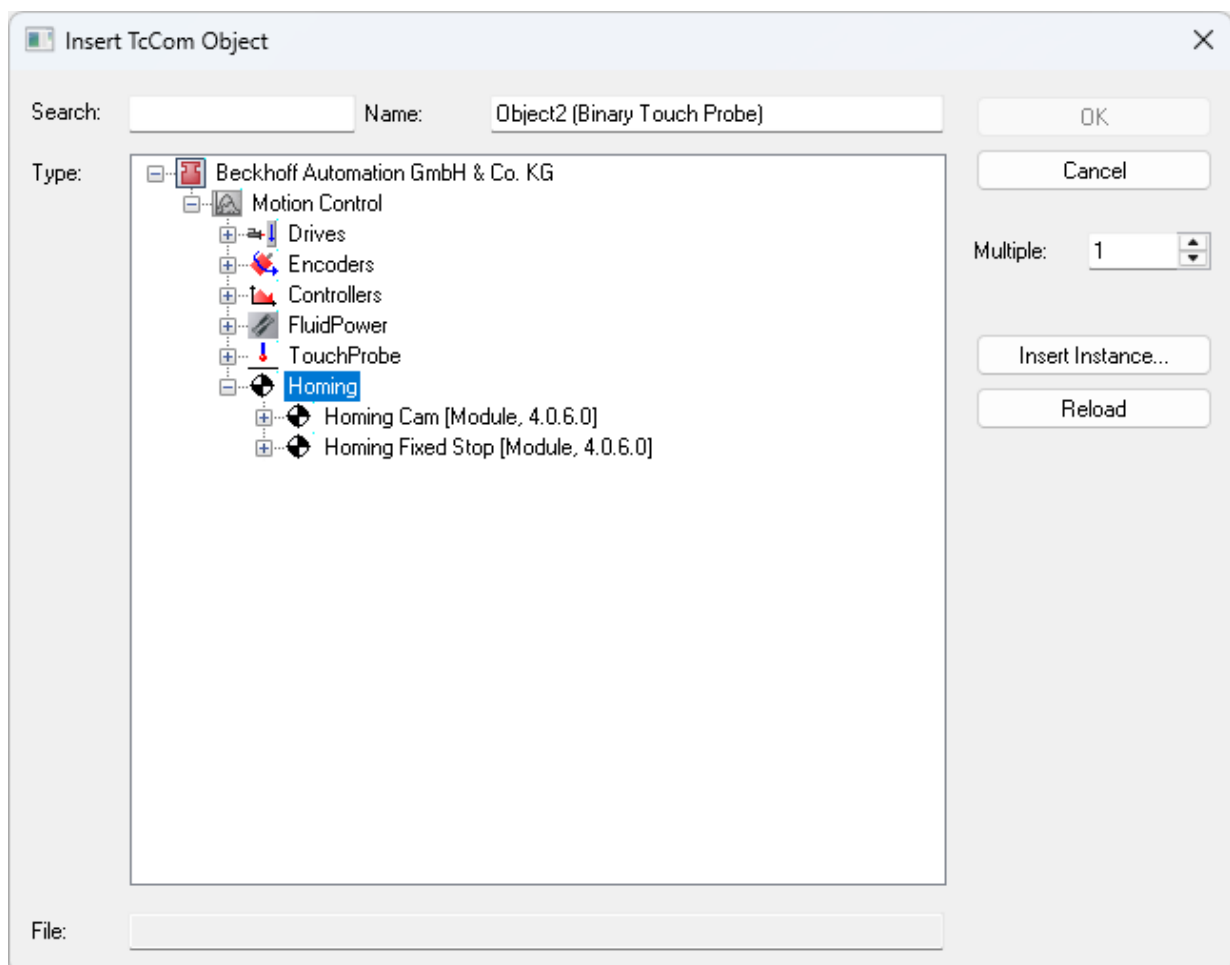
Die grundlegende Konfiguration erfolgt in TcCOM-Modulen. Die Konfiguration umfasst die Phasen einer Initialisierungsfahrt (z. B. Bewegung in Richtung des Referenzsignals oder Bewegung bis ein Synchronereignis eintritt) sowie die zugehörigen Randbedingungen in Form von Objektparametern. Ein Homing-Modul kann optional einer Achse als Kind-Objekt hinzugefügt werden, wenn der Betrieb der Achse eine Initialisierungsfahrt voraussetzt. Pro Achse kann ein Homing-Modul konfiguriert werden.

Homing Modul hinzufügen

1. Machen Sie im MC-Projekt im Strukturbaum einen Rechtsklick auf **Axes > Axis > Add New Item...**



2. Navigieren Sie im Dialog **Insert TcCom Object** unter **Type** und der Gruppe **Motion Control** zur Gruppe **Homing**.
3. Klappen Sie diese Gruppe ggf. auf.

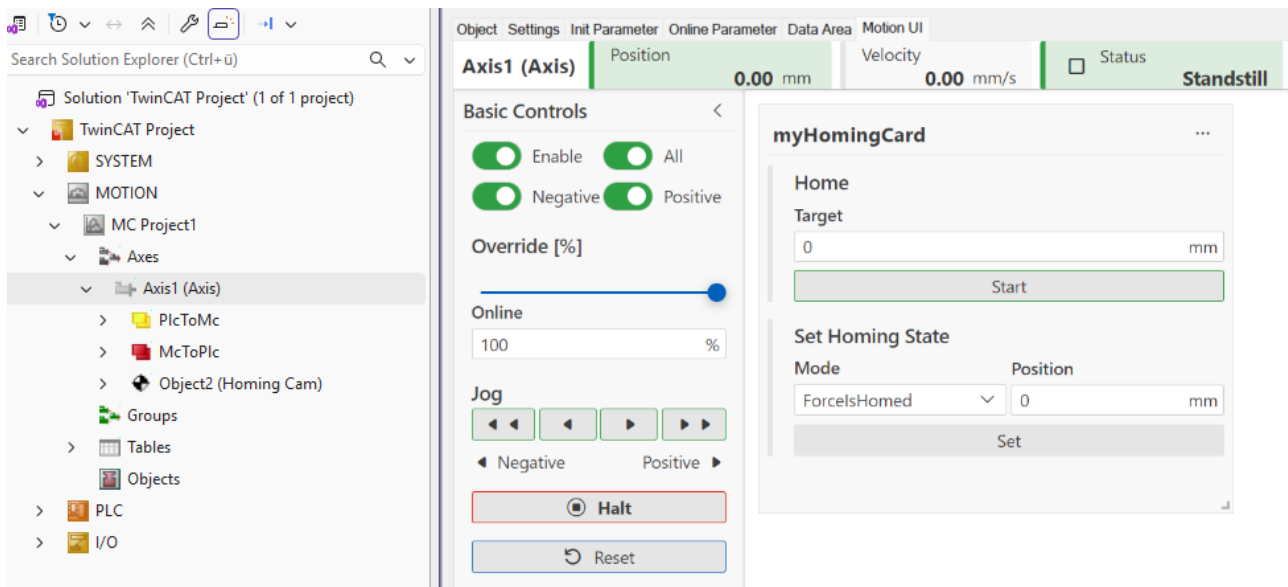


- Wählen Sie ein entsprechendes Homing-Modul aus und schließen den Vorgang mit einem Klick auf den Button **OK** ab.

Achten Sie darauf, das Modul in der gleichen Version auszuwählen, die für bereits vorhandenen Motion-Control-Objekte in der Solution verwendet wird.

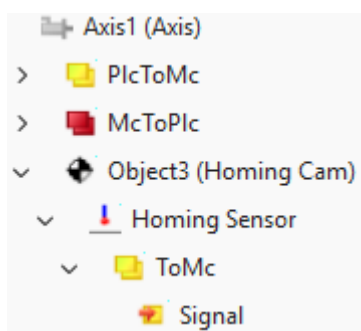
Referenzierfahrt starten

Die im Homing-Modul konfigurierte Sequenz kann mit dem Funktionsbaustein **MC_Home** [► 288] in der PLC gestartet werden. Alternativ ist dies auch über das Home-Control innerhalb einer Karte der Motion UI möglich.



3.2.3.13.1 Homing Cam

Das Objekt Homing Cam ermöglicht einen digitalen Eingang als Referenzsignal für eine Initialisierungsbewegung zu verwenden. Dazu ist dem Objekt ein Binary Touch Probe Objekt [► 104] unterlagert. So ist das Objekt Homing Cam zu verwenden, um beispielsweise das Signal einer digitalen Eingangsklemme oder einen Ausgang der PLC als Referenzsignal zu verknüpfen. Die Signalauswertung findet im Kontext der MET-Task [► 19] statt. Die Initialisierungsfahrt wird im Tab Parameter (Init) [► 111] konfiguriert.



i Ansicht unterlagerter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.13.1 Init Parameter

General

Homing Position

Name	Standardwert	Einheit	Typ
Homing Position	0.0	[PosUnit]	LREAL

Der Parameter `Homing Position` definiert die absolute Referenzposition. Auf diese Position wird die Achse bei einer erfolgreichen Referenzfahrt gesetzt. Der Parameter hat eine Auswirkung, wenn für das Kommando zum Starten einer Referenzfahrt (z. B. `MC_Home`) der Wert `#Default` für den Input der Referenzposition verwendet wird.

Sequence

Ein Referenziervorgang läuft in mehreren Phasen ab. Mit den nachfolgenden Parametern können einzelne Phasen aktiviert und konfiguriert werden.

Move To Cam Enabled

Name	Standardwert	Einheit	Typ
Move To Cam Enabled	FALSE		BOOL
Move To Cam Distance	10.0	[PosUnit]	LREAL
Move To Cam Velocity	30.0	[PosUnit]/s	LREAL

Die Aktivierung des Parameters `Move To Cam Enabled` (Wert `TRUE`) fügt dem Initialisierungsprozess eine Sequenz in Form einer Achsbewegung hinzu. Es wird nach einem Referenzsignal gesucht. Die Achse bewegt sich so lange, bis am Eingang `Signal` [► 105] ein entsprechender Wert detektiert wird. Die Logik des Eingangs ist abhängig vom Parameter `Invert Input` [► 113].

- `Move To Cam Distance`
Wenn innerhalb dieser Distanz (ausgehend von der Startposition des Referenziervorgangs) das erwartete Ereignis am Eingang `Signal` [► 105] nicht erkannt wird, wird der Referenziervorgang abgebrochen. Das Vorzeichen des Wertes beeinflusst die Bewegungsrichtung der Achse.
- `Move To Cam Velocity`
Die Achse bewegt sich während der Sequenz maximal mit der in diesem Parameter eingestellten Geschwindigkeit. Der Wert ist >0.0 einzustellen.

Move From Cam Enabled

Name	Standardwert	Einheit	Typ
Move From Cam Enabled	FALSE		BOOL
Move From Cam Distance	-10.0	[PosUnit]	LINT
Move From Cam Velocity	30.0	[PosUnit]/s	LINT

Die Aktivierung des Parameters `Move From Cam Enabled` (Wert `TRUE`) fügt dem Initialisierungsprozess eine Sequenz in Form einer Achsbewegung hinzu. Die Achse bewegt sich so lange, bis das Referenzsignal am Eingang `Signal` [► 105] (zurück) in den nicht betätigten Zustand wechselt. Die Logik des Eingangs `Signal` [► 105] ist abhängig vom Parameter `Invert Input` [► 113].

- `Move From Cam Distance`
Wenn innerhalb dieser Distanz (ausgehend von der Startposition des Referenziervorgangs) das erwartete Ereignis am Eingang `Signal` nicht erkannt wird, wird der Referenziervorgang abgebrochen. Das Vorzeichen des Wertes beeinflusst die Bewegungsrichtung der Achse.
- `Move From Cam Velocity`
Die Achse bewegt sich während der Sequenz maximal mit der in diesem Parameter eingestellten Geschwindigkeit. Der Wert ist >0.0 einzustellen.

Move To Sync Enabled

Name	Standardwert	Einheit	Typ
Move To Sync Enabled	FALSE		BOOL
Move To Sync Distance	10.0	[PosUnit]	LREAL
Move To Sync Velocity	30.0	[PosUnit]/s	LREAL
Move To Sync Source	ZeroImpulseCoE		MC.EHomingSyncSource [► 205]
Move To Sync Edge	Positive		MC.ETouchProbeTrigger Source [► 208]

Die Aktivierung des Parameters `Move To Sync Enabled` (Wert `TRUE`) fügt dem Initialisierungsprozess eine Sequenz in Form einer Achsbewegung hinzu. Die Achse bewegt sich so lange, bis in der Touch Probe Einheit 1 des Antriebs ein konfiguriertes Ereignis detektiert wird. Die Touch Probe Einheit lässt sich für den Referenziervorgang mit dem Parameter `Move To Sync Source` einstellen.

- `Move To Sync Distance`
Wird innerhalb dieser Distanz (ausgehend von der Startposition des Referenziervorgangs) das erwartete Ereignis in der Touch Probe Einheit 1 nicht erkannt, wird der Referenziervorgang abgebrochen. Das Vorzeichen des Wertes beeinflusst die Bewegungsrichtung der Achse
- `Move To Sync Velocity`
Die Achse bewegt sich während der Sequenz maximal mit der in diesem Parameter eingestellten Geschwindigkeit. Der Wert ist >0.0 einzustellen.
- `Move To Sync Source`
Für die Dauer der Initialisierungsfahrt hat der Parameter einen Einfluss auf die Konfiguration der Touch Probe Einheit 1. Der Homing-Algorithmus wertet den Touch Probe 1 Eingang der Achse bzw. des unterlagerten Antriebsobjekts aus (PDO-Verknüpfung).
 - `HardwareLatch1`
Die Touch Probe Einheit 1 des Antriebs wird ausgewertet.
 - `ZeroImpulseCoE`
Der Nullimpuls des Encoders wird ausgewertet.
 - `HardwareLatch1CoEDriveDefined`
Der auszuwertende Eingang am Antrieb ist im CoE-Objekt `0x60D0:01` konfiguriert.
 - `Move To Sync Edge`
Die Algorithmik zur Auswertung des Sync Ereignisses erwartet eine steigende oder fallende Flanke in der Touch Probe Einheit 1

Move To Initial Position Enabled

Name	Standardwert	Einheit	Typ
Move To Initial Position Enabled	FALSE		BOOL
Move To Initial Position	10.0	[PosUnit]	LREAL
Move To Initial Position Velocity	30.0	[PosUnit]/s	LREAL

Optional kann ein Referenziervorgang mit einer Bewegung auf eine absolute Zielposition abgeschlossen werden, wenn der Parameter `Move To Initial Point Enabled` auf `TRUE` eingestellt wird.

- `Move To Initial Position`
Nach erfolgreicher Referenzierung wird ein Bewegungskommando auf die angegebene absolute Zielposition gestartet. Ist die Konstante `#Default` eingetragen, wird die angegebene `Homing Position` als Positionswert verwendet.
- `Move To Initial Position Velocity`
Die Achse bewegt sich während des abschließenden Bewegungskommandos maximal mit der in diesem Parameter eingestellten Geschwindigkeit. Der Wert ist >0.0 einzustellen.

● **Ansicht unterlagerter Objekte**



In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

Die nachfolgenden Parameter haben ihren Ursprung im unterlagerten Touch Probe Objekt [► 113].

Binary Touch Probe

Name	Standardwert	Einheit	Typ
Input Inverted	FALSE		BOOL

Invert Input

Ist der Wert des Parameters `Input Inverted` auf `TRUE` gesetzt, invertiert die interne Algorithmik der MC die Logik des Eingangs Signal [► 105] für die Auswertung der Touch Probe Einheit.

3.2.3.13.1.2 Online Parameter

Die Parameter (Online) haben ihren Ursprung im unterlagerten Touch Probe Objekt [► 104].

● **Ansicht unterlagerter Objekte**



In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

Positive Edge Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `#Invalid` signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `0` signalisiert, dass noch kein Ereignis erfasst wurde.

Negative Edge Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert #Invalid signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

3.2.3.13.1.3 Data Area

Die Data Areas haben ihren Ursprung im unterlagerten Touch Probe Objekt [► 105].



Ansicht unterlagelter Objekte

In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

Inputs

Enable (default)	Name	Typ	Beschreibung
☒	Signal	BIT	Eingang für das Triggersignal. Er kann beispielsweise mit einer Variablen aus der PLC oder mit einem digitalen Eingang einer I/O-Klemme verknüpft werden.
☐	WcState	BIT	Working Counter

ToPlc

Enable (default)	Name	Typ	Beschreibung
☒	Oid	OTCID	Objekt ID zur Verknüpfung mit einem Touch Probe Objekt in der PLC

HINWEIS

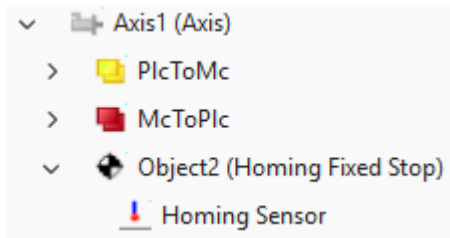
Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.13.2 Homing Fixed Stop

Das Objekt Homing Fixed Stop ermöglicht einen Referenziervorgang mithilfe eines physikalischen Objekts, das die Bewegung der Achse mechanisch blockiert. Dazu ist dem Homing Fixed Stop Objekt ein Fixed Stop Detection Touch Probe Objekt [► 107] unterlagert. So ist das Objekt Homing Fixed Stop zu verwenden, um beispielsweise mithilfe eines Festanschlags die Achse zu referenzieren. Die Signalauswertung findet im Kontext der MET-Task [► 19] statt. Die Initialisierungsfahrt wird im Tab Parameter (Init) [► 115] konfiguriert.



● Ansicht unterlagerter Objekte

i In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

3.2.3.13.2.1 Init Parameter

General

Homing Position

Name	Standardwert	Einheit	Typ
Homing Position	0.0	[PosUnit]	LREAL

Der Parameter `Homing Position` definiert die absolute Referenzposition. Auf diese Position wird die Achse bei einer erfolgreichen Referenzfahrt gesetzt. Der Parameter hat eine Auswirkung, wenn für das Kommando zum Starten einer Referenzfahrt (z. B. `MC_Home`) der Wert `#Default` für den Input der Referenzposition verwendet wird.

Sequence

Ein Referenziervorgang läuft in mehreren Phasen ab. Mit den nachfolgenden Parametern können einzelne Phasen aktiviert und konfiguriert werden.

Move To Fixed Stop Enabled

Name	Standardwert	Einheit	Typ
Move To Fixed Stop Enabled	FALSE		BOOL
Move To Fixed Stop Distance	10.0	[PosUnit]	LREAL
Move To Fixed Stop Velocity	30.0	[PosUnit]/s	LREAL
Torque Limit	30.0	%	LREAL

Die Aktivierung des Parameters `Move To Fixed Stop Enabled` (Wert `TRUE`) fügt dem Initialisierungsprozess eine Sequenz in Form einer Achsbewegung hinzu.

- `Move To Fixed Stop Distance`
Wenn innerhalb dieser Distanz (ausgehend von der Startposition des Referenziervorgangs) der erwartete Kraft-/Drehmoment-Schwellwert nicht erreicht wird, wird der Referenziervorgang abgebrochen. Das Vorzeichen des Wertes beeinflusst die Bewegungsrichtung der Achse.

- **Move To Fixed Stop Velocity**
Die Achse bewegt sich während der Sequenz maximal mit der in diesem Parameter eingestellten Geschwindigkeit. Der Wert ist >0.0 einzustellen.
- **Torque Limit**
Drehmoment-/Kraftbegrenzung während der Bewegung in Richtung des harten Anschlags.

Move From Fixed Stop Enabled

Name	Standardwert	Einheit	Typ
Move From Fixed Stop Enabled	FALSE		BOOL
Move From Fixed Stop Velocity	30.0	[PosUnit]/s	LINT

Die Aktivierung des Parameters **Move From Fixed Stop Enabled** (Wert **TRUE**) fügt dem Initialisierungsprozess eine Sequenz in Form einer Achsbewegung hinzu. Die Achse bewegt sich so lange, bis der erwartete Kraft-/Drehmoment-Schwellwert wieder unterschritten ist.

- **Move From Fixed Stop Velocity**
Die Achse bewegt sich während der Sequenz maximal mit der in diesem Parameter eingestellten Geschwindigkeit. Der Wert ist >0.0 einzustellen.

Move To Sync Enabled

Name	Standardwert	Einheit	Typ
Move To Sync Enabled	FALSE		BOOL
Move To Sync Distance	10.0	[PosUnit]	LREAL
Move To Sync Velocity	30.0	[PosUnit]/s	LREAL
Move To Sync Source	ZeroImpulseCoE		MC.EHomingSyncSource [► 205]
Move To Sync Edge	Positive		MC.ETouchProbeTrigger Source [► 208]

Die Aktivierung des Parameters **Move To Sync Enabled** (Wert **TRUE**) fügt dem Initialisierungsprozess eine Sequenz in Form einer Achsbewegung hinzu. Die Achse bewegt sich so lange, bis in der Touch Probe Einheit 1 des Antriebs ein konfiguriertes Ereignis detektiert wird. Die Touch Probe Einheit lässt sich für den Referenziervorgang mit dem Parameter **Move To Sync Source** einstellen.

- **Move To Sync Distance**
Wird innerhalb dieser Distanz (ausgehend von der Startposition des Referenziervorgangs) das erwartete Ereignis in der Touch Probe Einheit 1 nicht erkannt, wird der Referenziervorgang abgebrochen. Das Vorzeichen des Wertes beeinflusst die Bewegungsrichtung der Achse
- **Move To Sync Velocity**
Die Achse bewegt sich während der Sequenz maximal mit der in diesem Parameter eingestellten Geschwindigkeit. Der Wert ist >0.0 einzustellen.
- **Move To Sync Source**
Für die Dauer der Initialisierungsfahrt hat der Parameter einen Einfluss auf die Konfiguration der Touch Probe Einheit 1. Der Homing-Algorithmus wertet den Touch Probe 1 Eingang der Achse bzw. des unterlagerten Antriebsobjekts aus (PDO-Verknüpfung).
 - **HardwareLatch1**
Die Touch Probe Einheit 1 des Antriebs wird ausgewertet.
 - **ZeroImpulseCoE**
Der Nullimpuls des Encoders wird ausgewertet.
 - **HardwareLatch1CoEDriveDefined**
Der auszuwertende Eingang am Antrieb ist im CoE-Objekt `0x60D0:01` konfiguriert.
 - **Move To Sync Edge**
Die Algorithmik zur Auswertung des Sync Ereignisses erwartet eine steigende oder fallende Flanke in der Touch Probe Einheit 1

Move To Initial Position Enabled

Name	Standardwert	Einheit	Typ
Move To Initial Position Enabled	FALSE		BOOL
Move To Initial Position	10.0	[PosUnit]	LREAL
Move To Initial Position Velocity	30.0	[PosUnit]/s	LREAL

Optional kann ein Referenziervorgang mit einer Bewegung auf eine absolute Zielposition abgeschlossen werden, wenn der Parameter `Move To Initial Point Enabled` auf `TRUE` eingestellt wird.

- **Move To Initial Position**
Nach erfolgreicher Referenzierung wird ein Bewegungskommando auf die angegebene absolute Zielposition gestartet. Ist die Konstante `#Default` eingetragen, wird die angegebene `Homing Position` als Positionswert verwendet.
- **Move To Initial Position Velocity**
Die Achse bewegt sich während des abschließenden Bewegungskommandos maximal mit der in diesem Parameter eingestellten Geschwindigkeit. Der Wert ist >0.0 einzustellen.

● Ansicht unterlagerter Objekte

1 In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

Die nachfolgenden Parameter haben ihren Ursprung im unterlagerten *Touch Probe Objekt* [► 107].

Detection Thresholds

Name	Standardwert	Einheit	Typ
Torque/Force Threshold	10.0	%	LREAL
Velocity Threshold	10.0	[PosUnit]/s	LREAL
Lag Threshold	2.0	[PosUnit]	LREAL

Torque/Force Threshold

Schwellwert für den Istwert des Motordrehmoments. Wenn das Drehmoment diesen Wert während der Auswertung überschreitet, dann ist dies ein Maß für die Erkennung eines Festanschlags. Der Wert `#Ignore` deaktiviert diesen Schwellwert.

Velocity Threshold

Schwellwert für den Istwert der Achsgeschwindigkeit während der Auswertung. Die Kraft-/Drehmomentschwelle wird nur dann ausgewertet, wenn der Wert der Ist-Geschwindigkeit unterhalb dieses Schwellwerts liegt. Der Wert `#Ignore` deaktiviert diesen Schwellwert.

Lag Threshold

Schwellwert für den Positionsschleppfehler. Wenn der tatsächliche Schleppfehler diesen Wert während der Auswertung überschreitet, dann ist dies ein Maß für die Erkennung eines Festanschlags. Der Wert `#Ignore` deaktiviert diesen Schwellwert.

● Deaktivierung aller Schwellwerte

1 Werden alle Schwellwerte deaktiviert, meldet das Modul sofort ein erkanntes Event. Im Anwendungsfall des Homing wird dies instantan als ein Erreichen des Endanschlags erkannt.

- Geben Sie mindestens für den Schwellwert für Drehmoment/Kraft oder Schleppfehler eine reelle Zahl ein, um eine Erkennung eines Festanschlags auszuführen.

● **Verhalten einer beschleunigenden Achse mit einem Geschwindigkeitsschwellwert**

i Beschleunigt die Achse z. B. aus dem Stillstand heraus und die Auswertung des Touch Probe Moduls ist bereits aktiv, liegt die Geschwindigkeit der Achse u. U. noch unterhalb des eingestellten Schwellwerts. In Abhängigkeit zu den weiteren Schwellwerten kann dies dazu führen, dass das Touch Probe Modul ein erkanntes Event meldet. Im Anwendungsfall des Homing kann dies als ein Erreichen des Endanschlags erkannt werden.

- Starten Sie ggf. erst eine Achsbewegung bevor die Touch Probe Einheit aktiviert wird.

3.2.3.13.2.2 Online Parameter

Die Parameter (Online) haben ihren Ursprung im unterlagerten [Touch Probe Objekt \[► 108\]](#).

● **Ansicht unterlagelter Objekte**

i In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

Touch Probe Status

Name	Typ
State	MC.ETouchProbeTriggerState [► 208]
Last Position	LREAL
Last Dc Timestamp	LINT

State

Aktueller Zustand der Touch Probe Einheit.

Last Position

Position, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert `#Invalid` signalisiert, dass noch kein Ereignis erfasst wurde.

Last Dc Timestamp

DC Time in Nanosekunden, an der das konfigurierte Ereignis zuletzt erfasst wurde. Der Wert 0 signalisiert, dass noch kein Ereignis erfasst wurde.

3.2.3.13.2.3 Data Area

Die Data Area haben ihren Ursprung im unterlagerten [Touch Probe Objekt \[► 108\]](#).

● **Ansicht unterlagelter Objekte**

i In Abhängigkeit der Einstellung *Show Hidden Children* im Kontextmenü eines überlagerten Eltern-Objekts, sind unterlagerte Kind-Objekte ausgeblendet. Parameter und Data Areas des Kind-Objekts sind bei deaktivierter Einstellung im Eltern-Objekt sichtbar.

- Um Kind-Objekte einzublenden, wählen sie das **Eltern-Objekt** mit einem **Rechtsklick** aus (Kontextmenü) > und klicken auf **Show Hidden Children**.

ToPlc

Enable (default)	Name	Typ	Beschreibung
☒	Oid	OTCID	Objekt ID zur Verknüpfung mit einem Touch Probe Objekt in der PLC

HINWEIS**Warnung im Fehlerprotokollfenster durch ausgewählte Einträge ohne Verknüpfung zum I/O-Gerät**

Einträge in den Datenbereichen von Antriebs- und Geberobjekten, die zwar freigegeben (Enable = True), aber nicht auf das I/O-Antriebsobjekt verlinkt sind, führen zu einer Warnung im Fehlerprotokollfenster. Fehlt die Verknüpfung für eine zur Laufzeit verwendete Funktion, führt dies zu einem Laufzeitfehler. Beispielsweise kann eine Achse nicht aufgestartet werden, wenn das Statuswort vom I/O-Antriebsobjekt nicht verknüpft ist.

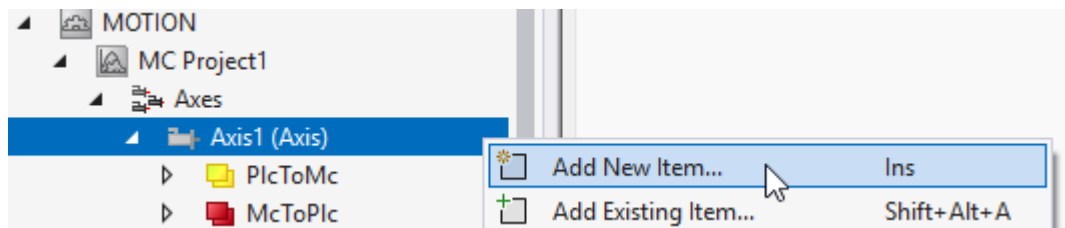
- Verknüpfen Sie benötigte Einträge der Schnittstelle zwischen Achse und I/O-Gerät
- Entfernen Sie die Auswahl des nicht benötigten Eintrags, um keine Warnung zu generieren.

3.2.3.14 Achsobjekt um ein zusätzliches Modul erweitern

Das Achsobjekt kann um ein zusätzliches Modul erweitert werden.

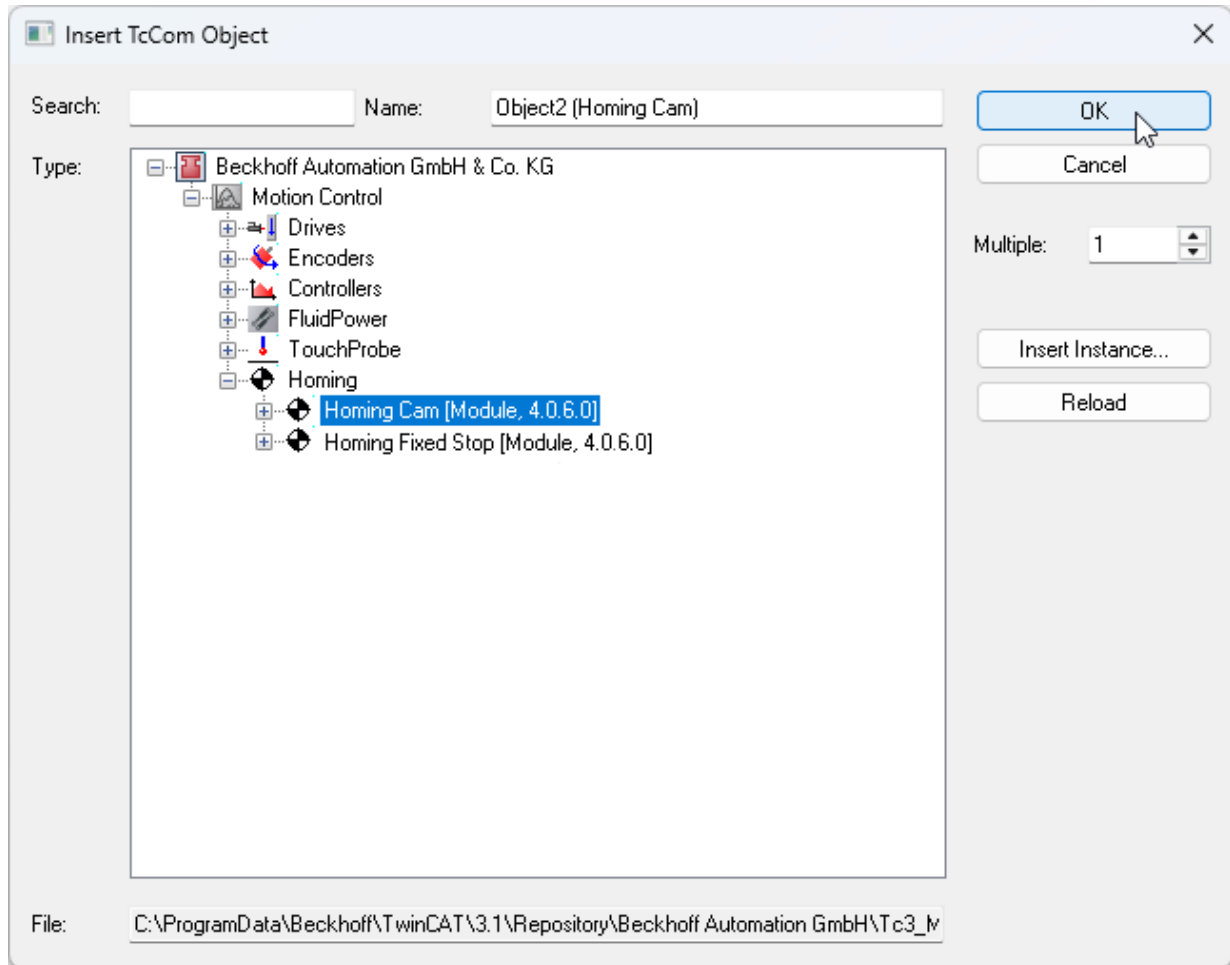
- ✓ Ein TwinCAT-Projekt mit MC-Projekt wurde erstellt, siehe [MC-Projekt erstellen](#) [► 17].
- ✓ Eine MC3-Achse wurde im TwinCAT-Projekt hinzugefügt, siehe [Achsobjekt erstellen](#) [► 25]

1. Machen Sie einen Rechtsklick auf **Axes > Axis > Add New Item...**

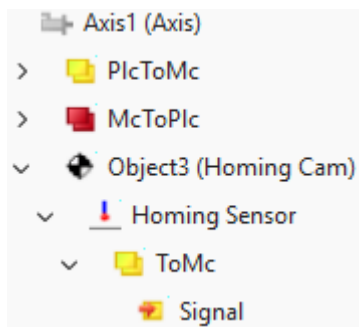


2. Wählen Sie im Dialog **Insert TcCom Object** unter **Motion Control** ein Objekt aus (z. B. **Homing > Homing Cam**).

3. Geben Sie im Eingabefeld **Multiple** die gewünschte Anzahl vor und bestätigen Sie beides über den Button **OK**.



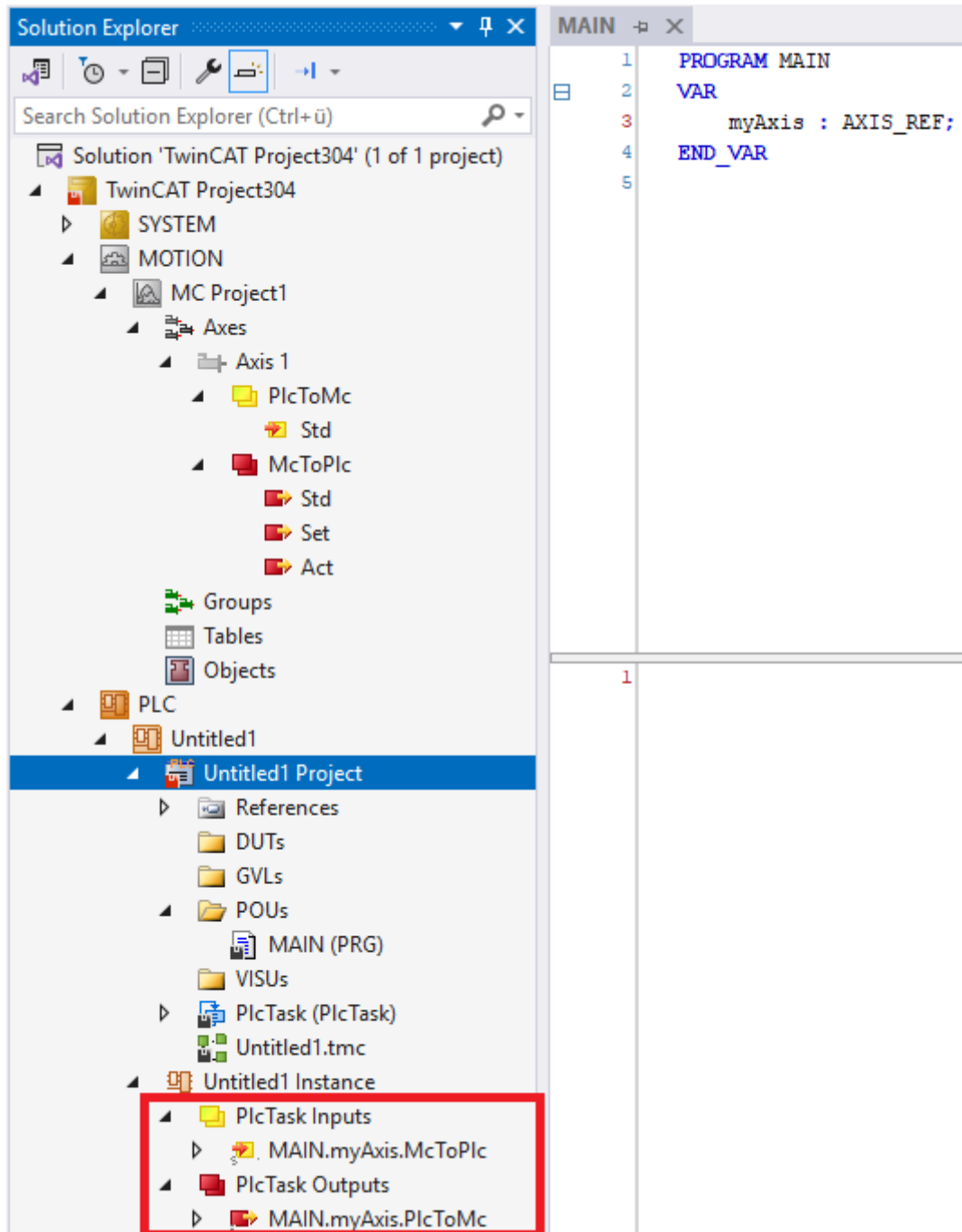
⇒ Ein zusätzliches Modul wurde erfolgreich unter dem Achsobjekt eingefügt.



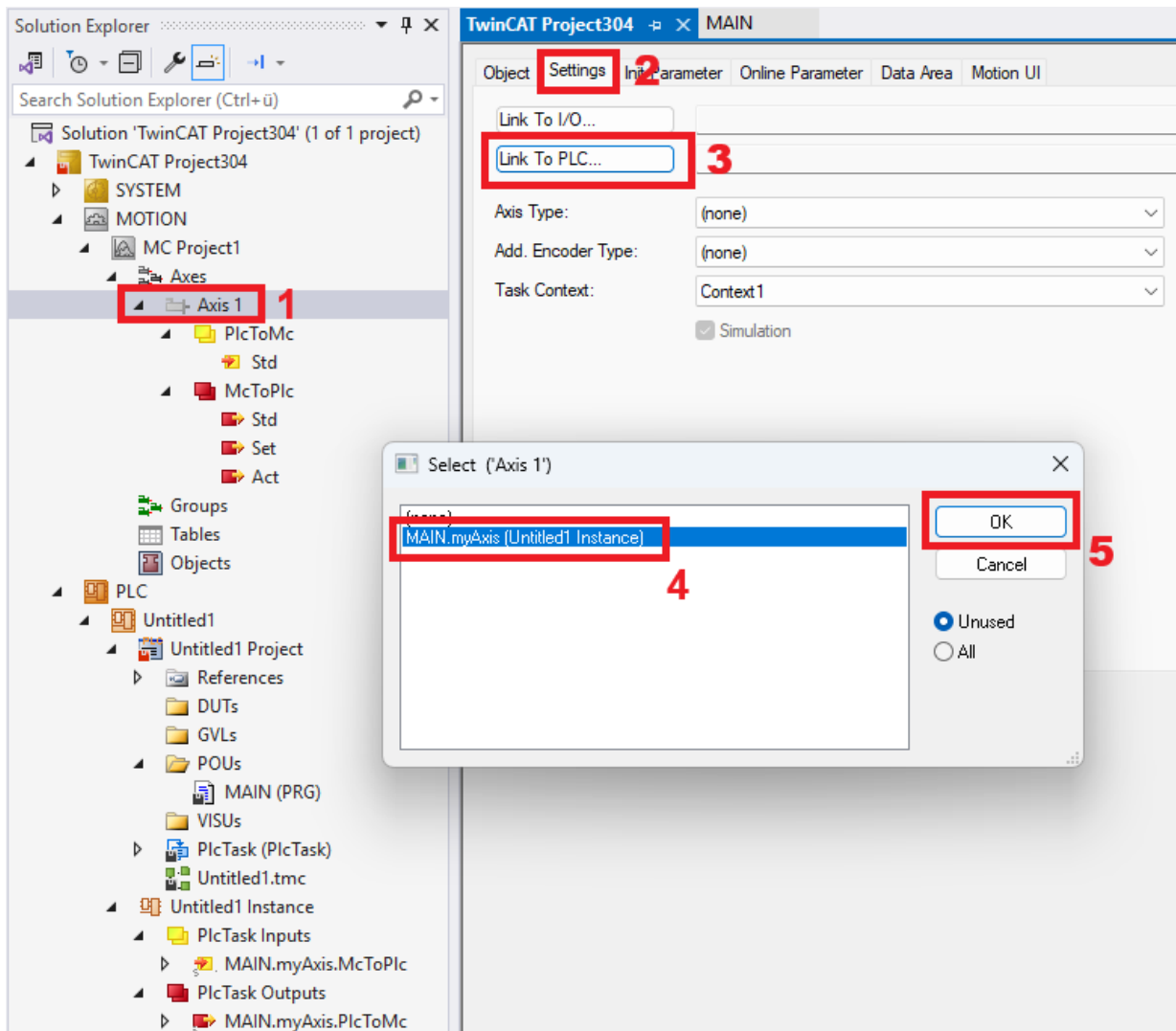
3.2.3.15 PLC mit einem Achsobjekt verknüpfen

- ✓ Ein Achsobjekt wurde angelegt, siehe [Achsobjekt erstellen](#) [► 25].
 - ✓ Eine PLC-Projekt mit einer Instanz des Tc3_Mc3Ptp.AXIS_REF wurde angelegt.
1. Bauen Sie das PLC-Projekt.

- ⇒ Die Instanz `axis` des `Tc3_Mc3Ptp.AXIS_REF` sollte nun unter den PLC-Instanzen im Strukturbaum im Solution Explorer angezeigt werden.

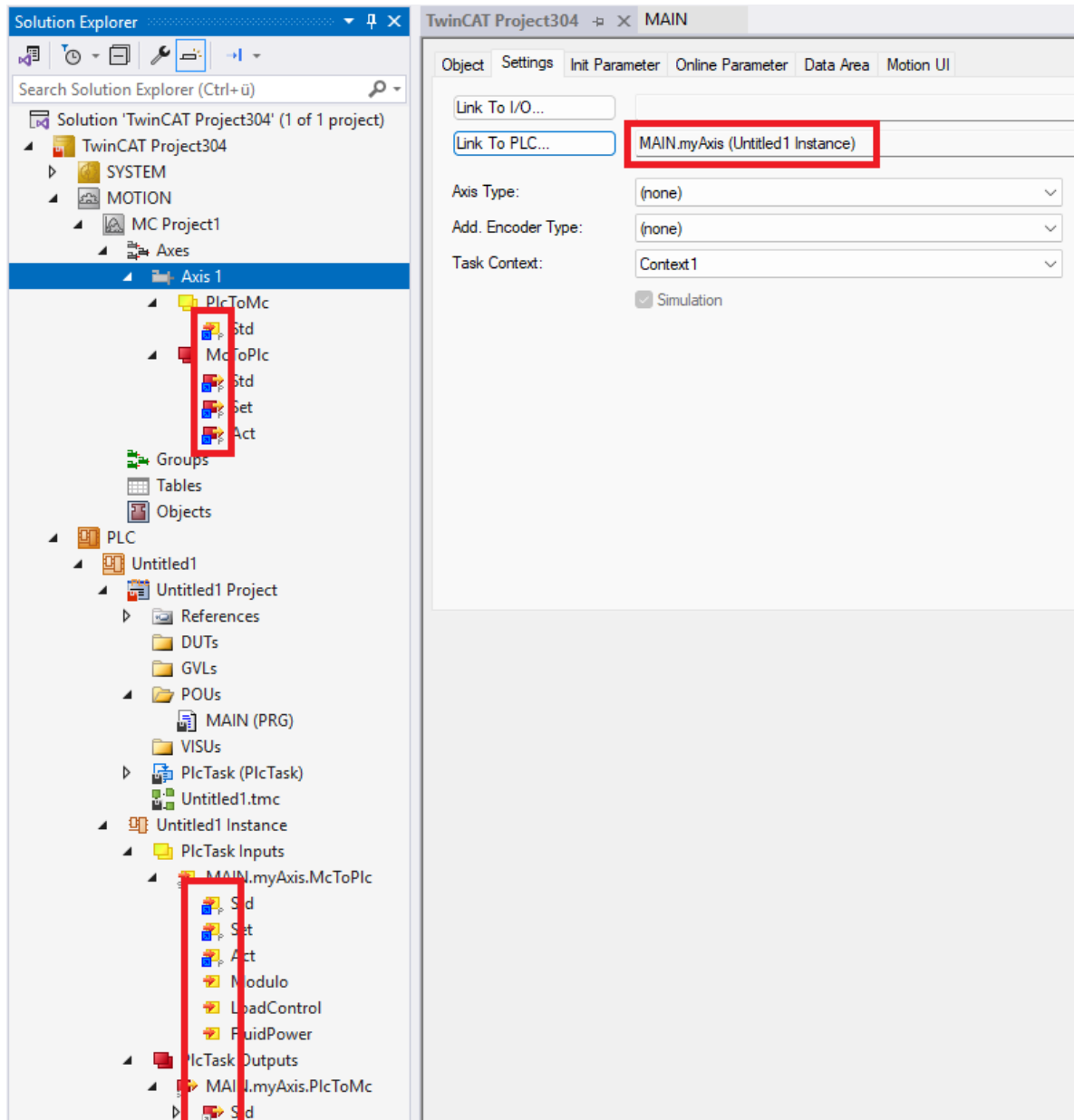


2. Verknüpfen Sie die PLC-Instanz des `Tc3_Mc3Ptp.AXIS_REF` mit der Achsinstanz der MC3 (Schritte 1-5).



- ⇒ Die Achse ist nun mit der PLC verknüpft.
- ⇒ Das Textfeld neben der Schaltfläche Link To PLC... zeigt die verknüpfte Instanz der AXIS_REF in der PLC an.

⇒ In der Konfiguration im Strukturbaum im Solution Explorer ist eine verknüpfte Variable aus dem zyklischen Interface durch ein blaues oder weißes Symbol erkenntlich.

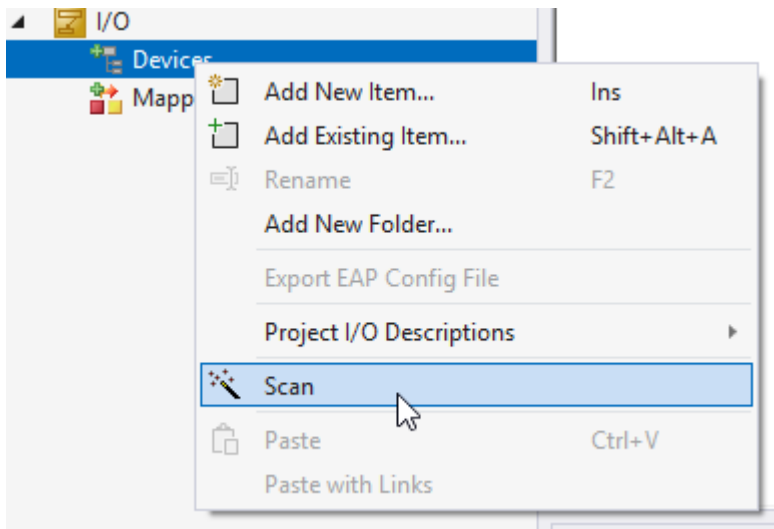


3. Aktivieren Sie das TwinCAT-Projekt.

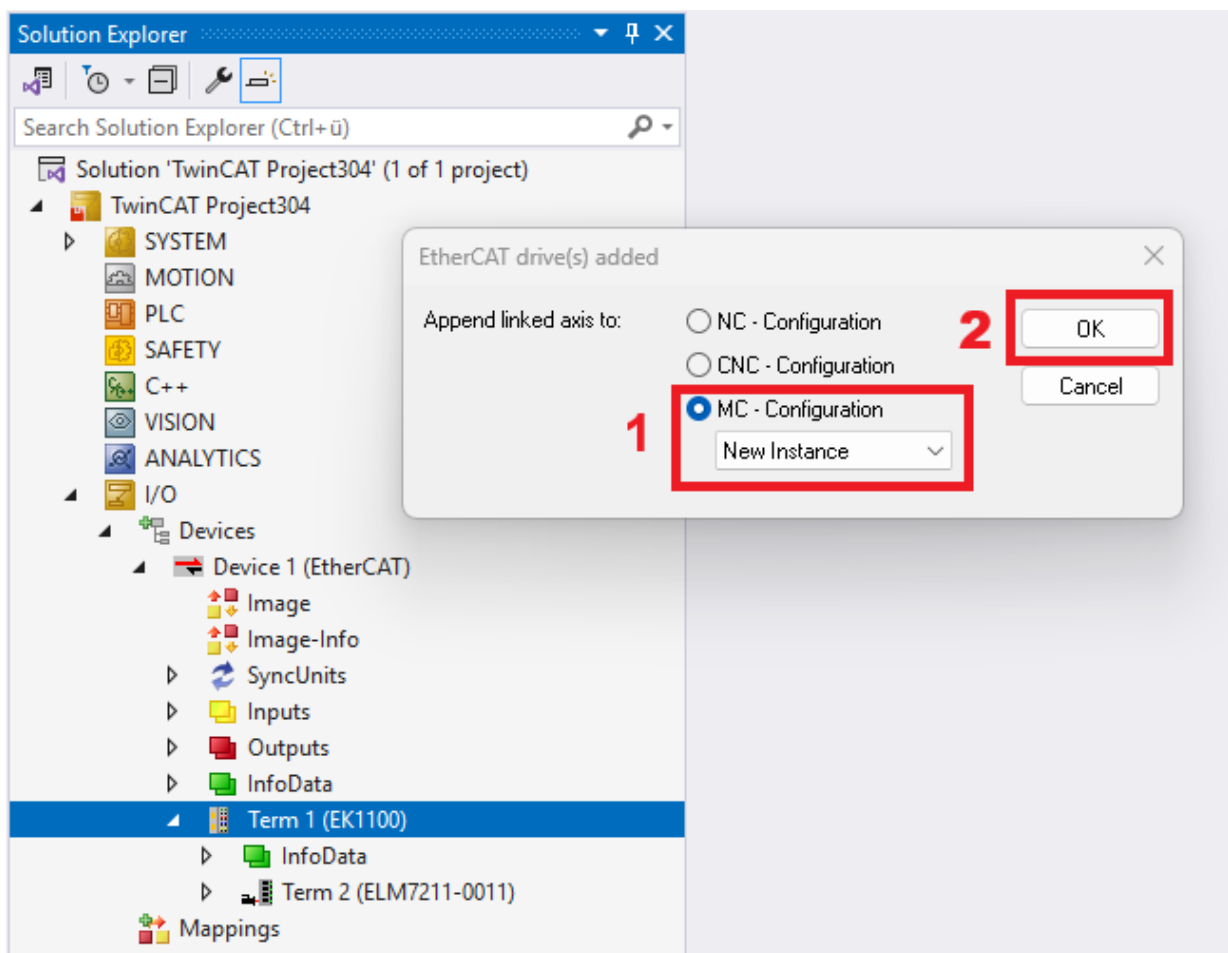
3.2.3.16 Achsobjekt mit Hardware verknüpfen

Automatisches Anlegen und Verknüpfen einer Achse beim Scan der EtherCAT-Konfiguration

1. Scannen Sie die EtherCAT Feldbuskonfiguration: I/O Knoten > Devices > Kontext Menü (Rechtsklick) > **Scan**



2. Wählen Sie im sich öffnenden Dialog „EtherCAT drive(s) added“ die Option **MC – Configuration** (1) aus. Wenn im TwinCAT Projekt bereits eine Motion-Konfiguration angelegt ist, können Sie die gewünschte Instanz im Auswahlménü anwählen. Die Auswahl **New Instance** bewirkt einen neuen Motion-Konfigurationsknoten. Bestätigen Sie den Dialog mit **OK** (2).

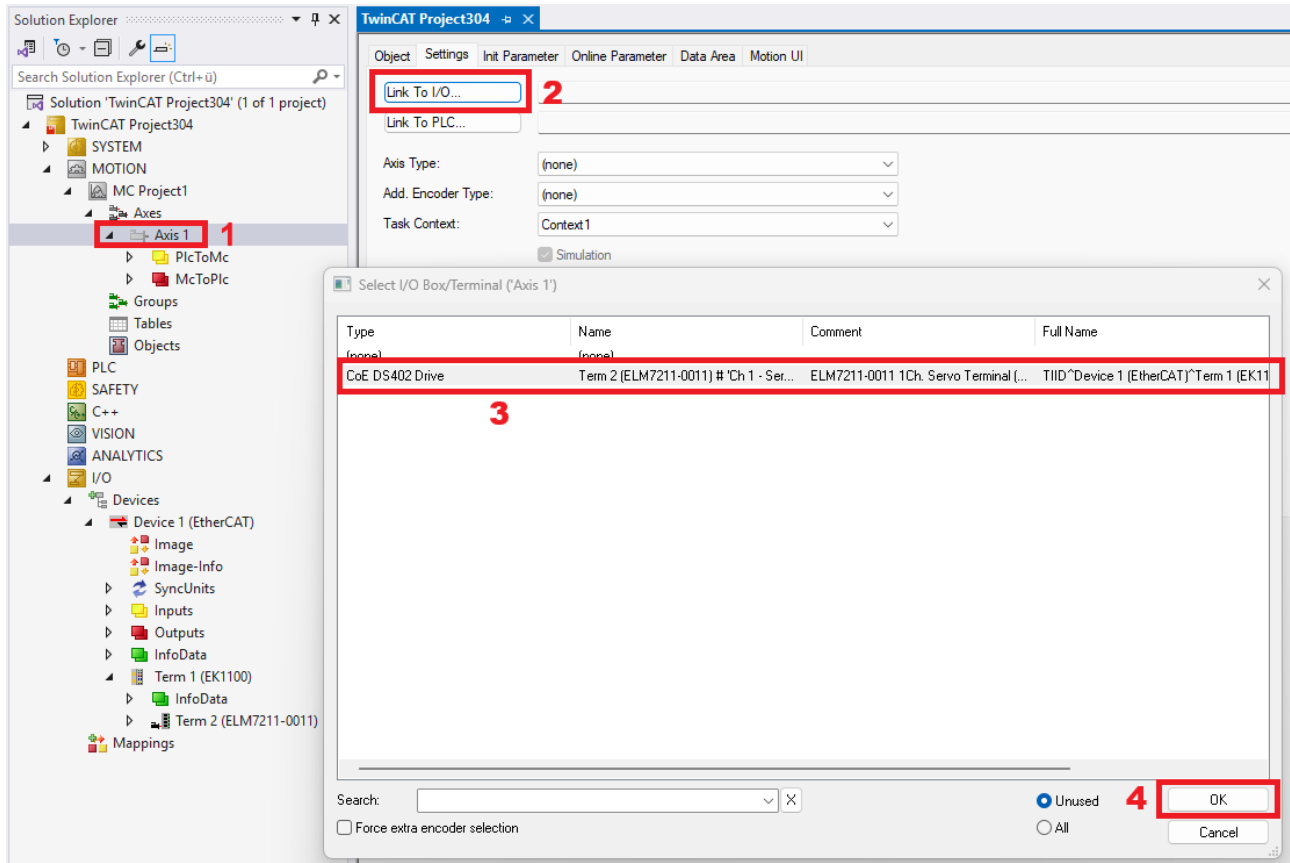


⇒ Im TwinCAT Project im Solution Explorer wird eine neue Motion-Konfiguration sowie eine Achse angelegt. Diese wird automatisch mit der Hardware verknüpft.

Link To I/O – Eine Achse manuell verknüpfen

Vorgehen – Beispiel für EL72xx/ELM72xx, AX5000, AX8000

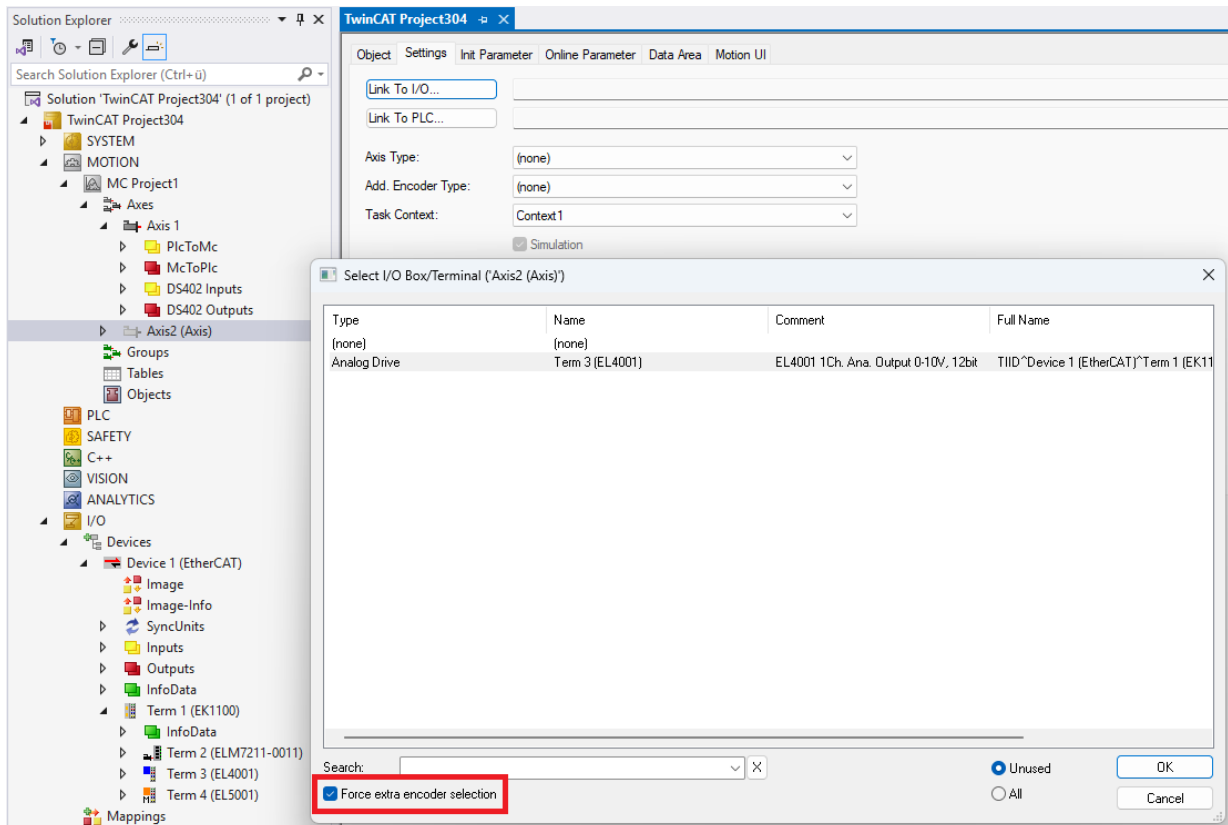
- ✓ Hardwarekonfiguration wurde angelegt.
- ✓ Motion-Konfiguration mit einer MC3-Achse wurde angelegt.
- 3. Wählen Sie die zu verknüpfende Achse (1) > Registerkarte **Settings**
- 4. Klicken Sie auf **Link To I/O...**(2).
- 5. Selektieren Sie die zu verknüpfende Antriebshardware (3).
- 6. Bestätigen Sie die Auswahl mit **OK** (4).



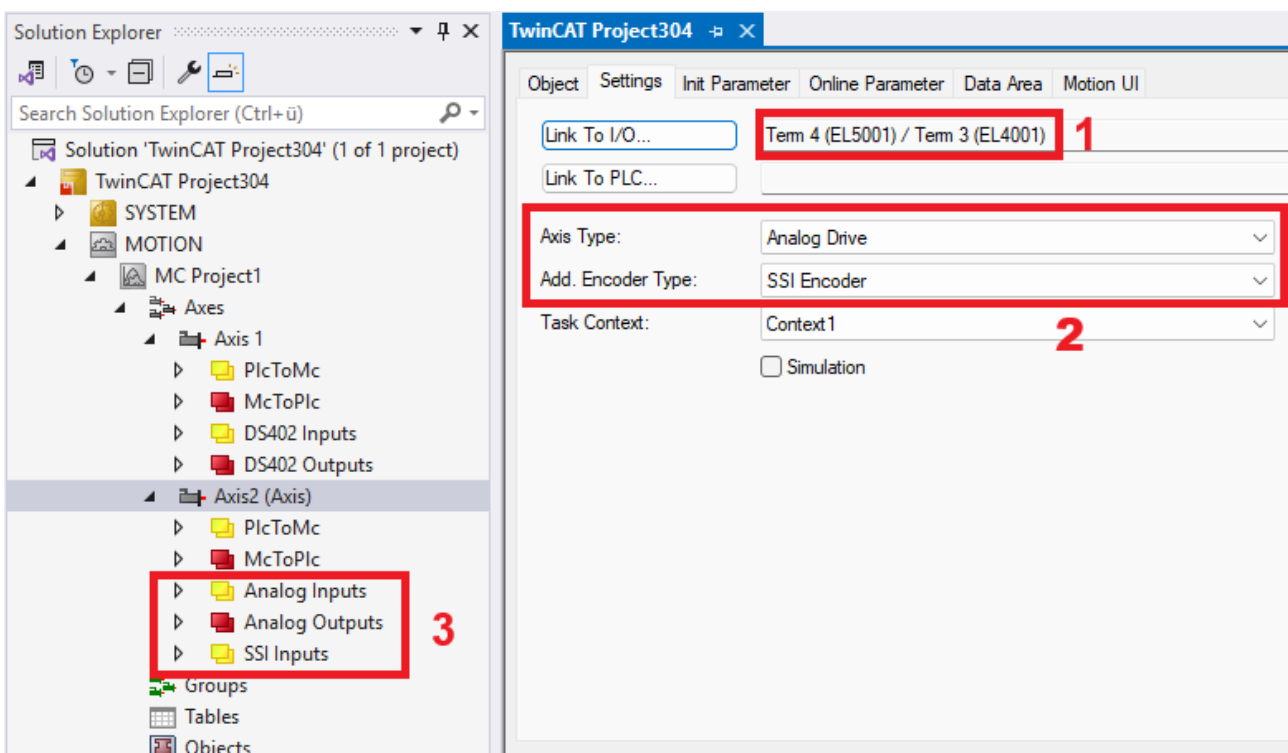
Vorgehen – Beispiel Analog Drive mit zusätzlichem Encoder

- ✓ In diesem Fall gelten zunächst die gleichen Schritte wie im vorherigen Abschnitt. Beachten Sie im Dialog „Select I/O Box/Terminal“ folgende Auswahl:

7. Wählen Sie die Option **Force extra encoder selection**, bevor Sie den Dialog durch **OK** quittieren.



- ⇒ Es öffnet sich ein weiterer Dialog.
- ⇒ Wählen Sie einen Encoder aus und bestätigen Sie die Auswahl mit dem Button **OK**.
- ⇒ Im Textfeld neben der Schaltfläche **Link To I/O...** werden die verknüpften Hardwaregeräte angezeigt (1).
- ⇒ Der Typ der Achse und des zusätzlichen Encoders werden angezeigt (2).
- ⇒ Die zyklischen Schnittstellen zwischen der MC3-Achse und den I/O-Geräten sind sichtbar (3).



3.2.3.17 Von NC2 zu MC3 umsteigen

Diese Anleitung beschreibt Schritt für Schritt, wie eine TwinCAT NC2 Achse (klassisch NC PTP) auf eine MC3-Achse migriert wird.

Voraussetzungen

- TwinCAT 3 Motion Konfiguration mit einer oder mehreren NC2-Achsen
- Abgeschlossene Installation der MC3-Pakete (siehe Kapitel [Installation](#) [► 8])
- Das Anlegen einer Sicherungskopie des originalen Projektes wird empfohlen.

Vorgehen

MC3-Projekte und Achsen anlegen

1. Legen Sie in der bestehenden Projektmappe eine neue MC3-Konfiguration an (siehe Kapitel [MC Projekt erstellen](#) [► 17]).
2. Fügen Sie entsprechend der Anzahl der zu migrierenden Achsen dem MC-Projekt neue Achsobjekte hinzu (siehe Kapitel [Achsobjekt erstellen](#) [► 25]).
3. Vergeben Sie ggf. einen anwendungsspezifischen Namen.
4. Unterscheiden Sie die virtuelle Achse und die reale Achse mit ihrer Verknüpfung zur Hardware:
 - ✓ Virtuelle Achse ohne Verknüpfung zur I/O-Konfiguration:
5. Fahren Sie mit der Migration der Achsparameter fort.
 - ✓ Achse mit Verknüpfung zur I/O-Konfiguration:
6. Entfernen Sie die Verknüpfung der zu migrierenden NC2-Achse zur I/O-Konfiguration.
7. Verknüpfen Sie die MC3-Achse mit dem I/O-Gerät (siehe Kapitel [Achsobjekt mit Hardware verknüpfen](#) [► 124] [Achsobjekt mit Hardware verknüpfen](#)).

Erweiterte Achskonfiguration (optional)

Die Modularität der MC3-Konfiguration ermöglicht eine bessere Übersicht und Strukturierung der Parameter. Ist für die zu migrierende Achse eine Referenzfahrt oder ein Positionsregler in der MC3-Software notwendig, sind zusätzliche Objekte unterhalb der MC3-Achse einzufügen. Gleiches gilt bei Verwendung der Messtaster-Funktion.

Referenzen:

- [Achsobjekt um ein zusätzliches Modul erweitern](#) [► 119]
- [Referenznocken \(Homing-Module\)](#) [► 109]
- [Messtastermodule \(Touch Probe-Module\)](#) [► 102]
- [Reglerobjekte \(Controller-Module\)](#) [► 95]

Achsparameter übertragen

Die Parameter der MC3-Achse sind jeweils in der Registerkarte **Init Parameter** zu finden.

NC2-Parameter	MC3-Parameter	Zu beachten
Simulationsmodus im Kontextmenü des Solution Explorers	Simulationsmodus im Kontextmenü des Solution Explorers und als Achsparameter <i>Simulation Mode</i>	Der Parameter kann für eine MC3-Achse auch während der Laufzeit geschrieben werden. Voraussetzung ist, dass die Achse keine Reglerfreigabe hat.
Unit (Registerkarte „Settings“)	Position Unit (Registerkarte „Init Parameter“)	In der MC3-Auswahl einer Einheit aus einem Aufzählungstypen.
Maximum Dynamics		

NC2-Parameter	MC3-Parameter	Zu beachten
Reference Velocity Default: 2200.0 Maximum Velocity Default: 2000.0	Maximum Velocity Default: 2000.0	Wird die Reference Velocity in der NC2 verwendet, kann für die MC3 das Minimum aus Reference Velocity (NC2) und Maximum Velocity (NC2) gebildet und dieser Wert in die Maximum Velocity (MC3) eingetragen werden.
Maximum Acceleration Default: 15000.0	Maximum Acceleration Default: 10000.0	Änderung des Default Wertes. Wert kann aus bestehender NC2-Achse kopiert werden.
Maximum Deceleration Default: 15000.0	Maximum Deceleration Default: 10000.0	Änderung des Default Wertes. Wert kann aus bestehender NC2-Achse kopiert werden.
Default Dynamics		
Default Acceleration Default: 1500.0	Default Acceleration Default: 2000.0	Änderung des Default Wertes. Wert kann aus bestehender NC2-Achse kopiert werden.
Default Deceleration Default: 1500.0	Default Deceleration Default: 2000.0	Änderung des Default Wertes. Wert kann aus bestehender NC2-Achse kopiert werden.
Default Jerk Default: 2250.0	Default Jerk Default: 4000.0	Änderung des Default Wertes. Wert kann aus bestehender NC2-Achse kopiert werden.
Fast Axis Stop		
Fast Axis Stop Signal Type (optional)	-	
Fast Acceleration (optional)	-	
Fast Deceleration (optional)	-	
Fast Jerk (optional)	-	
Manual Motion and Homing		
Homing Velocity (towards plc cam)	Homing-Objekt [► 109]	Zentrale Parametrierung in einem optionalen Homing-Objekt [► 109] als Kind-Objekt der Achse.
Homing Velocity (off plc cam)	Homing-Objekt [► 109]	Zentrale Parametrierung in einem optionalen Homing-Objekt [► 109] als Kind-Objekt der Achse.
Manual Velocity (Slow) Default: 100.0	Jog Velocity Slow Default: 100.0	Änderung des Parameternamens. Wert kann aus bestehender NC2-Achse kopiert werden.
Manual Velocity (Fast) Default: 600.0	Jog Velocity Fast Default: 250.0	Änderung des Default Wertes und des Parameternamens. Wert kann aus bestehender NC2-Achse kopiert werden.
Jog Increment (Forward)	MC_Jog.Distance	Verfügbar über Funktionsbaustein im Modus „Inching“
Jog Increment (Backward)	MC_Jog.Distance	Verfügbar über Funktionsbaustein im Modus „Inching“
Limit Switches		
Soft Position Limit Minimum Monitoring	Position Limit Enforcement Enabled	Ein allgemeiner Parameter zur Aktivierung der Überwachung

NC2-Parameter		MC3-Parameter	Zu beachten
	Minimum Position	Minimum Position	Der Parameter erscheint, wenn Position Limit Enforcement Enabled = TRUE gesetzt ist. Mit dem Wert #Ignore kann die Überwachung der Softwareendlagen einseitig deaktiviert werden.
Soft Position Limit Maximum Monitoring		Position Limit Enforcement Enabled	Ein allgemeiner Parameter zur Aktivierung der Überwachung
	Maximum Position	Maximum Position	Der Parameter erscheint, wenn Position Limit Enforcement Enabled = TRUE gesetzt ist. Mit dem Wert #Ignore kann die Überwachung der Softwareendlagen einseitig deaktiviert werden.
Monitoring			
Position Lag Monitoring		Position Lag Monitoring Enabled	Änderung des Parameternamens
	Maximum Position Lag Value	Maximum Position Lag Value	Der Parameter erscheint, wenn Position Lag Monitoring Enabled = TRUE gesetzt ist.
	Maximum Position Lag Filter Time	Maximum Position Lag Filter Time	Der Parameter erscheint, wenn Position Lag Monitoring Enabled = TRUE gesetzt ist.
Position Range Monitoring		-	
	Position Range Window	-	
Target Position Monitoring		MC_WaitForStableActualPosition	NC2: MC_MoveAbsolute.Done = TRUE wenn die ActPos entsprechend den Parametern TargetPositionMonitoring und TargetPositionWindow im Fenster ist. MC3: Der Funktionsbaustein kann hinter Bewegungskommandos des Typs DiscreteMotion (PLCopen Zustand) gepuffert werden. MC_MoveAbsolute.Done = TRUE wenn das Sollwertprofil am Ziel ist und ist damit unabhängig vom Target Position Monitoring. Es wird eine klare Trennung der Funktionalitäten geschaffen.
	Target Position Window		
	Target Position Monitoring Time		
In-Target Alarm			
	In-Target Timeout		
Motion Monitoring		-	
	Motion Monitoring Window	-	
	Motion Monitoring Time	-	
Setpoint Generator		-	
Setpoint Generator Type		-	
Velocity Override Type		-	
NCI-Parameter		-	

NC2-Parameter		MC3-Parameter	Zu beachten
Rapid Traverse Velocity (G0)		-	
Velo Jump Factor		-	
Tolerance ball auxiliary axis		-	
Max. position deviation, aux. axis		-	
Other Settings			
Position Correction		-	
	Filter Time Position Correction (P-T1)	-	
Backlash		-	
Error Propagation Mode		Beispiel: <code>MC_CamIn.ReactionToMasterError</code> <code>MC_CamIn.ReactionToSlaveError</code>	Kopplungsbausteine wie ein MC_GearIn oder MC_CamIn haben zusätzliche Eingänge. Kopplungen sind Kommandobasiert. Je Kopplung/Kommando kann eine eigene Fehlerfortpflanzung eingestellt werden.
	Error Propagation Delay	-	Bisherige Fehlerreaktionen und Weiterleitungen können über die Einstellungen an Kopplungsbausteinen und Achsparametern umgesetzt werden. Beispiel: Es tritt ein Fehler in der Slave-Achse auf. Der Fehler darf nicht an die Master-Achse propagiert werden (<code>MC_CamIn.ReactionToSlaveError := NoReaction</code>) und die Slave-Achse muss von der Master-Achse abgekoppelt werden. In der MC3 erfolgt in diesem Fall eine automatische Entkopplung und die Slave-Achse leitet ihre Fehlerreaktion ein. Wenn eine Kontrolle/Regelung der Achse weiterhin möglich ist, kann eine Stopprampe nach einem MC_Reset durch ein benutzerspezifisches Kommando mit gleichen Dynamikgrenzen abgelöst werden (in einem PLC-Zyklus möglich).
Couple slave to actual values if not enabled		Beispiel: <code>MC_CamIn.ReactionToMasterError</code> <code>MC_CamIn.ReactionToSlaveError</code>	Kopplungsbausteine wie ein MC_GearIn oder MC_CamIn haben zusätzliche Eingänge. Kopplungen sind Kommandobasiert. Je Kopplung/Kommando kann eine eigene Fehlerfortpflanzung eingestellt werden.
	Velocity Window	-	
	Filter Time for Velocity Window	-	

NC2-Parameter	MC3-Parameter	Zu beachten
Allow motion commands to slave axis (Default TRUE)	Kein gesonderter Parameter - MC3-Achsen erlauben Bewegungskommandos an die Slave Achse (Default TRUE)	Slave-Achsen können mit einem Bewegungskommando von ihrer Master-Achse abgekoppelt werden (BufferMode := Aborting)
Allow motion commands to external setpoint axis	Kein gesonderter Parameter - MC3-Achsen erlauben Bewegungskommandos an Achsen mit aktivierter externen Sollwertgenerierung (Default TRUE)	
Dead Time Compensation (Delay Velo and Position)	Dead Time Compensation From Hardware	In der Regel werden die Totzeiten während der Initialisierung des EtherCAT Feldbusses automatisch ermittelt und in der MC3 intern verrechnet. Sollte dennoch eine manuelle Korrektur notwendig sein, kann diese in den Parametern Additional Time Shift From Hardware und Additional Time Shift To Hardware eingetragen werden.
	Additional Time Shift from Hardware	
	Dead Time Compensation To Hardware	
	Additional Time Shift to Hardware	
Data Persistence	-	

Drive-Parameter übertragen

Die Parameter der MC3-Achse sind jeweils in der Registerkarte **Init Parameter** zu finden. Dabei ist zu unterscheiden, ob die Funktion *Show Hidden Children* im Kontextmenü der Achse aktiv ist (standardmäßig deaktiviert, Achse > Rechtsklick > Show Hidden Children). Ist die Funktion aktiviert, werden die nachfolgenden Parameter im dann sichtbaren Kind-Objekt der Achse angezeigt. Andernfalls sind alle Parameter zentral über das Achsobjekt abrufbar.

NC2-Parameter	MC3-Parameter	Zu beachten
Output Settings		
Invert Motor Polarity	Drive Type / Scaling > Direction Inverted	Die Invertierung der Motor und Encoder Orientierung ist in der MC3 ein Parameter am das Antriebsobjekt. Aus Sicht der MC ist häufig lediglich ein Hardwareobjekt vorhanden ist (z. B. AX8000). In diesem Fall sind zwei Parameter mit unterschiedlichen Einstellungen nicht sinnvoll, da es zu einem Schleppfehler kommt und der Regelkreis nicht wie erwartet funktioniert. Ein zusätzliches Encoder-TcCom-Objekt als untergeordnetes Element der Achse verfügt über einen eigenen Parameter.
Reference Velocity	-	Siehe Beschreibung der Achsparameter
Position and Velocity Scaling		
Output Scaling Factor (Position)	Drive Type / Scaling > Position Scaling Numerator/Denominator from Hardware	Nicht direkt übertragbar. Siehe MC3-Parameter zur Skalierung der Soll- und Ist-Positionen. Handelt es sich aus Sicht der MC um ein

NC2-Parameter	MC3-Parameter	Zu beachten
		Hardwareobjekt, sind die Parameter unter dem Drive-Objekt zu finden. Andernfalls haben Drive und Encoder eigenständige Skalierungsparameter.
Output Scaling Factor (Velocity)	Drive Type / Scaling > Velocity Scaling Numerator to Hardware	Velocity Scaling Denominator muss bei diesem Vorgehen auf 1 geschrieben werden. Wird verwendet, um die Soll- und Ist-Geschwindigkeiten an den Antrieb bzw. vom Geber zu skalieren.
Minimum Drive Output Limitation [-1.0 ... 1.0]	-	
Maximum Drive Output Limitation [-1.0 ... 1.0]	-	
Output Delay	-	
Torque and Acceleration Scaling		
Input Scaling Factor (Actual Torque)	Drive Type / Scaling > Torque Scaling from Hardware	Wert kann direkt übernommen werden.
Input P-T1 Filter Time (Actual Torque)	Drive Type / Actual Torque Filter / Settings > Filter Time Torque	Eigenständiges TcCOM-Objekt als Kind-Objekt unter Drive Objekt. <i>Show Hidden Children</i> aktivieren, um Filterobjekt hinzuzufügen.
Input P-T1 Filter (Actual Torque Derivative)	Drive Type / Actual Torque Filter / Settings > Filter Time Torque Derivative	Eigenständiges TcCOM-Objekt als Kind-Objekt unter Drive Objekt. <i>Show Hidden Children</i> aktivieren, um Filterobjekt hinzuzufügen.
Output Scaling Factor (Torque Setpoint)	-	
Output Scaling Factor (Torque Offset)	-	
Output Delay (Torque Offset)	-	
Output Scaling Factor (Acceleration)	-	
Output Delay (Acceleration)	-	
Optional Position Command Output Smoothing Filter		
Smoothing Filter Type	-	
Smoothing Filter Time	-	
Smoothing Filter Order (P-Tn only)	-	
Other Settings		
Drive Mode	-	
Drift Compensation (DAC-Offset)	-	
Following Error Calculation	Axis > Monitoring > Position Lag Calculation	Internal – InternallyCalculated Extern - HardwareProvided
Error Tolerance	-	

Encoder-Parameter übertragen

Die Parameter der MC3-Achse sind jeweils in der Registerkarte **Init Parameter** zu finden. Dabei ist zu unterscheiden, ob die Funktion *Show Hidden Children* im Kontextmenü der Achse aktiv ist (standardmäßig deaktiviert, Achse > Rechtsklick > Show Hidden Children). Ist die Funktion aktiviert, werden die nachfolgenden Parameter im dann sichtbaren Kind-Objekt der Achse angezeigt. Andernfalls sind alle Parameter zentral über das Achsobjekt abrufbar.

Aus Sicht der MC ist häufig lediglich ein Hardwareobjekt vorhanden (z. B. AX8000). In diesem Fall ist in der MC3 z. B. für die Drehrichtungsumkehr nur ein Parameter vorgesehen. Ein zusätzliches Encoder-TcCOM-Objekt als untergeordnetes Element der Achse verfügt wiederum über einen eigenen Parameter.

Sind in der nachfolgenden Tabelle zwei Möglichkeiten in der Spalte „MC3-Parameter“ angegeben, gibt es den betreffenden Parameter für ein Drive-Objekt und optional für einen zusätzlichen Encoder.

NC2-Parameter	MC3-Parameter	Zu beachten
Encoder Evaluation		
Invert Encoder Counting Direction	Axis > Drive Type / Scaling > Direction Inverted oder Axis > Encoder Type / Scaling > Direction Inverted	Die Invertierung der Motor und Encoder Orientierung ist in der MC3 <i>ein</i> Parameter am das Antriebsobjekt. Im Fall eines Hardwareobjekts sind zwei Parameter mit unterschiedlichen Einstellungen nicht sinnvoll, da es zu einem Schleppfehler kommt und der Regelkreis nicht wie erwartet funktioniert.
Scaling Factor Numerator	Axis > Drive Type / Scaling > Position Scaling Numerator from Hardware oder Axis > Encoder Type / Scaling > Position Scaling Numerator from Hardware	
Scaling Factor Denominator	Axis > Drive Type / Scaling > Position Scaling Denominator from Hardware oder Axis > Encoder Type / Scaling > Position Scaling Denominator from Hardware	
Position Bias	Axis > Drive Type / Scaling > Encoder Mounting Offset oder Axis > Encoder Type / Scaling > Encoder Mounting Offset	
Modulo Factor	Axis > General > Position Modulus	Parameter der Achse ist sichtbar, sobald der Parameter <code>Is Modulo Axis = TRUE</code> ist.
Tolerance Window for Modulo Start	Axis > General > Position Modulo Tolerance Window	
Encoder Mask (maximum encoder value)	Axis > Drive Type / General > Maximum Counter Value + Minimum Counter Value oder Axis > Encoder Type / General > Maximum Counter Value + Minimum Counter Value	Die beiden MC3-Parameter sind sichtbar, wenn der Parameter <code>Position Data Range</code> auf <code>UserDefined</code> eingestellt ist. Eine Manuelle Prüfung des Wertes ist notwendig. In der MC3 ist der Wertebereich vorzeichenbehaftet. <code>FullRange</code> : bezieht sich auf den maximal darstellbaren Wertebereich des Datentyps, z. B. 0 und 2^{32} für UDINT und -2147483648 bis 2147483647 für DINT. <code>UserDefined</code> : manuelle Eingabe des Wertebereichs für den Wert aus der PDO-Verknüpfung

NC2-Parameter	MC3-Parameter	Zu beachten
Encoder Sub Mask	Axis > Drive Type / General > Maximum Absolute Counter Value oder Axis > Encoder Type / General > Maximum Absolute Counter Value	Die beiden MC3-Parameter sind sichtbar, wenn der Parameter <code>Position Interpretation</code> auf <code>AbsoluteUserDataRange</code> eingestellt ist. Anzahl der eindeutigen Inkremente pro Umdrehung ($2^{20}-1$ für einen Singleturn Absolutwertgeber).
Reference System	Axis > Drive Type / General > Position Interpretation oder Axis > Encoder Type / General > Position Interpretation	- INCREMENTAL entspricht <code>Incremental</code> - INCREMENTAL (singleturn absolute) entspricht <code>Incremental</code> - ABSOLUTE entspricht <code>AbsoluteFullRange</code> - ABSOLUTE MULTITURN RANGE (with single overflow) entspricht <code>AbsoluteFullRange</code> - ABSOLUTE SINGLETURN RANGE (with single overflow) entspricht <code>AbsoluteFullRange</code> - ABSOLUTE (modulo) entspricht <code>AbsoluteModulo</code>
Limit Switches		
Soft Position Limit Minimum Monitoring	Position Limit Enforcement Enabled	Ein allgemeiner Parameter zur Aktivierung der Überwachung
Minimum Position	Minimum Position	Der Parameter erscheint, wenn <code>Position Limit Enforcement Enabled = TRUE</code> gesetzt ist. Mit dem Wert <code>#Ignore</code> kann die Überwachung der Softwareendlagen einseitig deaktiviert werden.
Soft Position Limit Maximum Monitoring	Position Limit Enforcement Enabled	Ein allgemeiner Parameter zur Aktivierung der Überwachung
Maximum Position	Maximum Position	Der Parameter erscheint, wenn <code>Position Limit Enforcement Enabled = TRUE</code> gesetzt ist. Mit dem Wert <code>#Ignore</code> kann die Überwachung der Softwareendlagen einseitig deaktiviert werden.
Filter		
Filter Time for Actual Position (P-T1)	Axis > Drive Type / Actual Position Filter / Settings > Filter Type = PT1 > Filter Time Position oder Axis > Encoder Type / Actual Position Filter / Settings > Filter Type = PT1 > Filter Time Position	Ein zusätzliches TcCOM-Filter-Objekt ist jedem Antriebs- und/oder Geberobjekt standardmäßig unterlagert. Der <code>Filter Typ</code> ist auf PT1 einzustellen. Die weiteren Parameter zur Einstellung der Filterzeiten werden dann sichtbar.
Filter Time for Actual Velocity (P-T1)	Axis > Drive Type / Actual Position Filter / Settings > Filter Type = PT1 > Filter Time Velocity oder	

NC2-Parameter	MC3-Parameter	Zu beachten
	Axis > Encoder Type / Actual Position Filter / Settings > Filter Type = PT1 > Filter Time Velocity	
Filter Time for Actual Acceleration (P-T1)	Axis > Drive Type / Actual Position Filter / Settings > Filter Type = PT1 > Filter Time Acceleration oder Axis > Encoder Type / Actual Position Filter / Settings > Filter Type = PT1 > Filter Time Acceleration	
Homing		
Invert Direction for Homing Sensor Search	Homing-Objekt ▶ 109	Zentrale Parametrierung in einem optionalen Homing-Objekt ▶ 109 als Kind-Objekt der Achse.
Invert Direction for Sync Impulse Search		
Home Position (Calibration Value)		
Reference Mode (Sync condition)		
Homing Sensor Source		

3.2.4 Encoder Axis

Im TwinCAT-Kontext spricht man von einer Encoder-Achse, wenn ein reines Gebersystem als Achs-Objekt ins System integriert werden soll. Einstellungen und Parameter der Encoder Axis werden im folgenden Kapitel beschrieben.

3.2.4.1 Registerkarte Settings

In der Registerkarte **Settings** können wesentliche Einstellungen zur Encoder-Achse vorgenommen werden.

Eigenschaft	Beschreibung
Link To I/O...	Der Link-Button öffnet einen Dialog zum Verknüpfen der MC-Encoder-Achse mit der Antriebshardware unter dem I/O-Knoten. Im Feld rechts daneben wird das verknüpfte Element angezeigt.
Link To PLC...	Der Link-Button öffnet einen Dialog zum Verknüpfen der MC-Encoder-Achse mit einer PLC-Instanz der ENCODER_AXIS_REF. Im Feld rechts daneben wird das verknüpfte Element angezeigt.

Eigenschaft	Beschreibung		
Add. Encoder Type	Wählen Sie hier den Encodertyp aus, der verwendet wird. Details zu den Encodern sind unter Encoder-Module [► 82] zu finden. Der Wert wird in der Regel automatisch beim Verknüpfen mit einem Hardware-Element bestimmt (Link To I/O).		
Task Context	Jede Encoder-Achse muss einem Task-Kontext zugeordnet sein. Besitzt ein MC Project [► 17] mehr als einen Kontext, so kann über die Combobox der Kontext der Achse verändert werden. Das Erstellen und Konfigurieren neuer Kontexte erfolgt über die Registerkarte Context [► 19] des MC-Projekts.		
Simulation	☒		Bei aktivem Simulation Mode wird der Link zum I/O ignoriert. Statt des konfigurierten Encoder Typs muss unter Init Parameter > General > Simulation Data Source die Datenquelle für die Simulation ausgewählt werden.
	☐		Ist der Encoder Type (none) ausgewählt, handelt es sich immer um eine Simulation-Encoder-Achse und die Einstellung kann nicht verändert werden.
	☐		Der konfigurierte Encoder Type und Link zum I/O wird verwendet.

3.2.4.2 Registerkarte Init Parameter

General

Name	Standardwert	Einheit	Typ
Simulation Mode	TRUE		BOOL
Simulation Data Source	00000000		OTCID_EncoderAxisSimulationDataSource
Position Unit	mm	[PosUnit]	MC.EPositionUnit
Is Modulo Axis	FALSE		BOOL
Position Modulus	360.0	[PosUnit]	LREAL

Simulation Mode

Der Simulationsmodus zeigt an, ob die I/O-Verbindung zum Antrieb und Encoder aktiv (**FALSE**) oder simuliert (**TRUE**) sein soll. Der Parameter kann während der Laufzeit geschrieben werden, wenn die Achse ohne Reglerfreigabe ist.

Simulation Data Source

Simulation Data Source gibt die Objekt-ID der für die Simulation verwendeten Achse an. Ist der Simulation Mode aktiv und die Simulation Data Source gesetzt, liest die Encoder-Achse die Sollwerte der Simulationsquelle als ihre eigenen Istwerte aus.

Position Unit

Die Position Unit wird für Anzeigezwecke, z. B. auf der Inbetriebnahmeoberfläche, verwendet. Ein- und Ausgaben an Funktionsbausteinen werden nicht konvertiert.

Is Modulo Axis

Aktivieren Sie diesen Parameter, um für diese Achse Modulo-Funktionalitäten an Funktionsbausteinen zu nutzen. Sollwert und Istwert werden in der zyklischen Schnittstelle als Absolut- und Modulo-Wert berechnet. Um Modulo-Informationen z. B. mit der PLC auszutauschen, ist die [Data Area Modulo](#) [► 33] zu aktivieren.

- **Position Modulus**
Modulo-Wert zur Berechnung der Modulo-Position

Diagnostics

Name	Standardwert	Einheit	Typ
TraceLevel	tlWarning		TcTraceLevel

Legen Sie fest, welche Nachrichten an die TwinCAT-Fehlerliste gesendet werden sollen.

3.2.4.3 Registerkarte Online Parameter

Es gibt die folgenden Online-Parameter bei einer Encoder-Achse:

Additional Error Event Information

Name	Einheit	Type
Originating Error Event Sender Id		OTCID
Original Error Id		HRESULT

Originating Error Event Sender Id

Liegt ein Folgefehler z. B. durch eine Kopplung vor, zeigt dieser Parameter die Objekt-ID der Achse an, von der der Fehler ausging.

Original Error Id

Liegt ein Folgefehler z. B. durch eine Kopplung vor, zeigt dieser Online-Parameter die ursprüngliche Fehlernummer an.

3.2.4.4 Registerkarte Data Area

Im Allgemeinen können TcCOM-Objekte Datenbereiche enthalten, die von der Umgebung verwendet werden können (z. B. von einer PLC-Instanz oder vom IO-Bereich von TwinCAT). Bei den TwinCAT MC3 TcCOM-Objekten kann über die Registerkarte Data Area eine Auswahl der zu verwendenden Datenstrukturen erfolgen.

HINWEIS

Nicht verknüpfte CDTs können bei Zugriff zu Floating Point Exception führen

Vergleiche mit NaN-Werten führen zu einer Exception, welche einen Stopp der Laufzeit und einen möglichen Maschinenschaden nach sich zieht.

Nur wenn das `IsConnected = TRUE` ist, sind valide Daten zu erwarten. Andernfalls besteht die Gefahr eines inkonsistenten oder ungültigen Datensatzes, was zu unabsehbaren Folgen und Fehlverhalten führen kann. Rechenoperationen mit ungültigen Werten können zu Ausnahmefehlern in der Gleitkommaeinheit des Prozessors führen.

- Schalten Sie die Floating Point Exceptions aus, wenn NaN-Werte in der Applikation verwendet und verarbeitet werden sollen.
- Prüfen Sie vor einem Zugriff (z. B. aus der PLC) auf ein CDT das `IsConnected` des CDTs.
- Machen Sie sich mit dem Umgang von benannten Konstanten und NaN Werten vertraut.

McToPlc (Output)

Enable (default)	Name	Typ	Beschreibung
☒	Std	POINTER TO MC.CDT_MCTOPLC_ENCODER_AXI S_STD [► 214]	Zyklische Standarddatenstruktur mit der Oid der Encoder-Achse und den wichtigsten Zustandsinformationen.
☒	Act	POINTER TO MC.CDT_MCTOPLC_ENCODER_AXI S_ACT [► 213]	Optionale zyklische Datenstruktur mit Informationen zu den aktuellen Istwerten der Encoder-Achse.

Enable (default)	Name	Typ	Beschreibung
			Hinweis Nur wenn <code>IsConnected = TRUE</code> ist, sind valide Daten zu erwarten. Eine Validierung gültiger Daten ist nach folgendem Kapitel sicherzustellen: Konstanten (Named Constants)
☐	Modulo	POINTER TO <code>MC.CDT_MCTOPLC_ENCODER_AXIS_MODULO</code> [► 213]	Optionale zyklische Datenstruktur mit Informationen zur Moduloposition der Encoder-Achse. Für eine sinnvolle Berechnung der Moduloposition ist im Tab <code>Init Parameter</code> [► 29] der Parameter Is Modulo Axis auf <code>TRUE</code> zu setzen. Hinweis Nur wenn <code>IsConnected = TRUE</code> ist, sind valide Daten zu erwarten. Eine Validierung gültiger Daten ist nach folgendem Kapitel sicherzustellen: Konstanten (Named Constants)

3.2.4.5 Encoder-Achse verwenden

- ✓ Das Hinzufügen eines Motion-Objekts zu einer bestehenden Projektmappe wird im Abschnitt Achsobjekt erstellen [► 25] erläutert.
- 1. Wählen Sie dabei das Modul **Encoder Axis** aus, um mindestens eine Encoder-Achse der Motion-Konfiguration hinzuzufügen.
- 2. Alternativ können Sie eine Encoder-Achse automatisch beim Scan der I/O-Konfiguration über den Strukturbaum des Projekts im Solution Explorer hinzufügen und gleichzeitig verknüpfen.

Option 1: Betrieb im Simulation Mode

Wenn die Encoder-Achse ausschließlich- oder auch während des Betriebs im Simulationsmodus betrieben werden soll, sind die folgenden beiden Parameter zu setzen:

- Init Parameter > Simulation Mode (standardmäßig durch die Entwicklungsumgebung gesetzt; wenn keine Hardware verknüpft ist, kann zur Laufzeit geschrieben werden),
 - Init Parameter > Simulation Data Source.
3. Wählen Sie in Parameter **Simulation Data Source** die Objekt-ID der Achse aus, deren Sollwerte als Istwerte der Encoder-Achse ausgegeben werden sollen.
 4. Legen Sie ggf. eine zusätzliche Simulationsachse an.

Option 2: Betrieb mit Hardware

Wenn die Encoder-Achse mit Hardware verwendet werden soll, ist sie mit einem entsprechenden Gerät in der I/O-Konfiguration zu verknüpfen. Wenn die Encoder-Achse noch nicht automatisch im Strukturbaum im Solution Explorer beim Scan der I/O-Konfiguration verknüpft wurde, können Sie dies entsprechend den Schritten im Abschnitt Achsobjekt mit Hardware verknüpfen [► 124] tun.

Encoder-Achse zum PLC-Programm hinzufügen und verknüpfen

- ✓ Ein PLC-Projekt ist angelegt und die Bibliothek `Tc3_Mc3Ptp` als Referenz hinzugefügt.
- 5. Fügen Sie eine Instanz vom Typ `ENCODER_AXIS_REF` in Ihrem PLC-Programm ein.

Beispiel:

```
PROGRAM MAIN
VAR
    myEncoder : Tc3_Mc3Ptp.ENCODER_AXIS_REF;
END_VAR
```

6. Gehen Sie in ähnlicher Weise zu Abschnitt [PLC mit einem Achsobjekt verknüpfen \[► 120\]](#) vor, um eine Verknüpfung zwischen der PLC und dem Motion-Objekt herzustellen.

⇒ Eine Encoder-Achse ist dem Motion-Knoten im Strukturbaum hinzugefügt worden und mit der PLC verknüpft.

PLC-Programm mit einer Encoder-Achse erstellen

Mit der MC3 ist in der PLC zu beachten, dass zwischen einer klassischen Achse vom Typ `AXIS_REF` und einer Encoder-Achse vom Typ `ENCODER_AXIS_REF` unterschieden wird. Das ermöglicht bereits zum Zeitpunkt der Kompilierung im XAE zu unterscheiden, ob eine programmierte Funktion unterstützt wird (siehe auch [Übersicht: MC3-Typen \[► 219\]](#)). Beispielsweise können Bewegungskommando (`MC_MoveAbsolute`, `MC_Halt`, ...) und ausgewählte administrative Kommandos (`MC_Power`) nicht mit einer Encoder-Achse verwendet werden.



Verwendung nicht unterstützter Funktionsbausteine

Die Programmierung nicht unterstützter Funktionsbausteine führt zu Fehlermeldungen während der Kompilierung.

3.2.5 FluidPower Axis

Eine Hydraulikachse kann sowohl als ventilgeregelter als auch als pumpengeregelter Antrieb ausgeführt werden. Zur Lösung wird in der MC3 eine spezielle Konfiguration der [Standardachse \[► 25\]](#) verwendet. Dabei kann die Hydraulikachse mit mehreren Arten von Wegmesssystemen sowie Stellgliedern arbeiten.

Das Hinzufügen einer Hydraulikachse ist identisch zu dem Prinzip einer elektrischen Standardachse:

- [Achse hinzufügen \[► 140\]](#)

Allgemein verfügt die FluidPower Axis über die Möglichkeit, sowohl die Position als auch die Kraft bzw. den Druck der Hydraulikachse zu regeln. Die Blockschaltbilder sowie eine ergänzende Beschreibung für die jeweilige Regelung kann hier gefunden werden:

- [Positionsregler \[► 150\]](#)
- [Kraft-/Druckregler \[► 151\]](#)

Zur Kraft-/Druckregelung bedarf es der Einbindung der Druck- bzw. Kraftsensoren. Details können dem folgenden Abschnitt entnommen werden:

- [Drucksensor A / B \[► 152\]](#)
- [Virtueller Kraftsensor \[► 154\]](#)

Das Besondere an der in der MC3 integrierten Lösung ist die Möglichkeit der Implementierung und automatischen Aufzeichnung der Systemkennlinie. Diese ist dem Positions- bzw. Kraft-/Druckregler nachgeschaltet, wodurch nichtlineare Eigenschaften des Systems eliminiert werden. Details zur Systemkennlinie können dem folgenden Abschnitt entnommen werden:

- [Ventilkennlinie \[► 155\]](#)

Die FluidPower Axis weist zudem einige Eigenschaften bzw. Parameter auf, die im Speziellen nur bei dieser, nicht aber bei der Standardachse vorzufinden sind:

- [Parameter einer Achse \[► 145\]](#)

Folgende „How to“ beschreiben den Ablauf einer Inbetriebnahme eines:

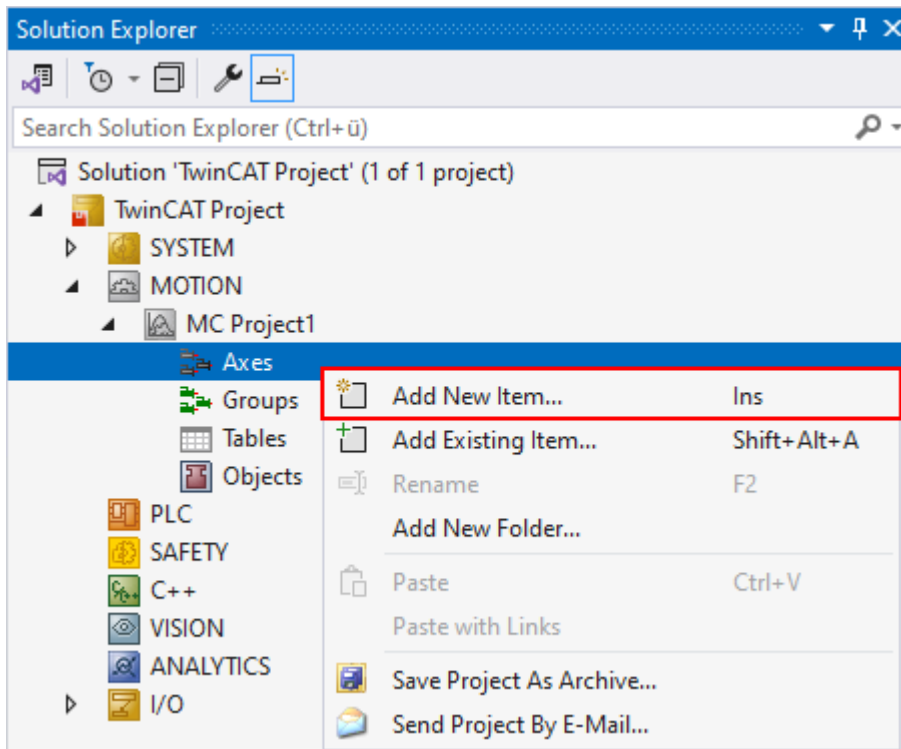
- ventilgeregelten Antriebs,
siehe: [FluidPower-Achse \(ventilgeregelte Achse\) in Betrieb nehmen \[► 159\]](#)
- pumpengeregelten Antriebs,
siehe: [FluidPower-Achse \(pumpengeregelte Achse\) in Betrieb nehmen](#)

3.2.5.1 Achse hinzufügen

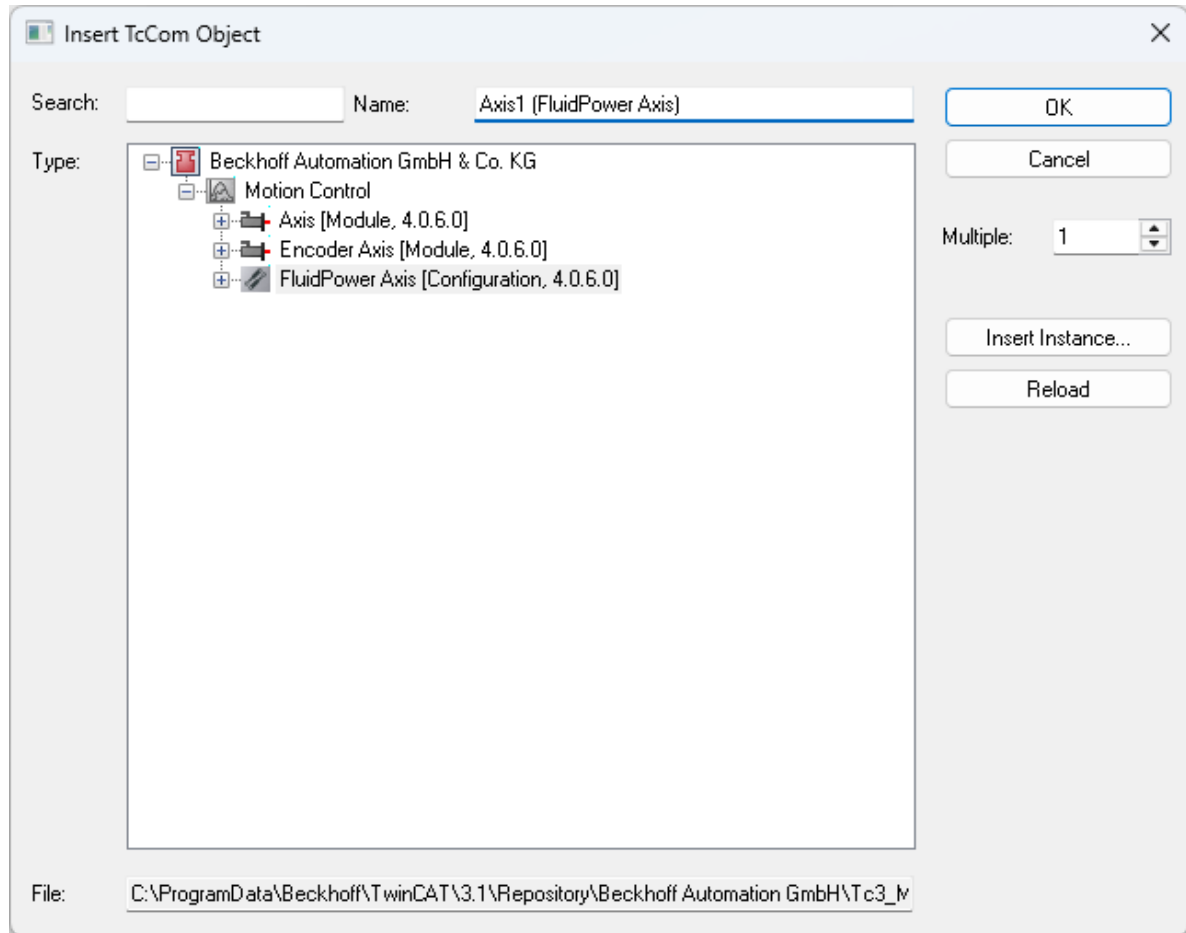
3.2.5.1.1 Achsobjekt erstellen/Achse hinzufügen

- ✓ Ein TwinCAT-Projekt mit MC-Projekt wurde erstellt, siehe [MC-Projekt erstellen](#) [► 17].

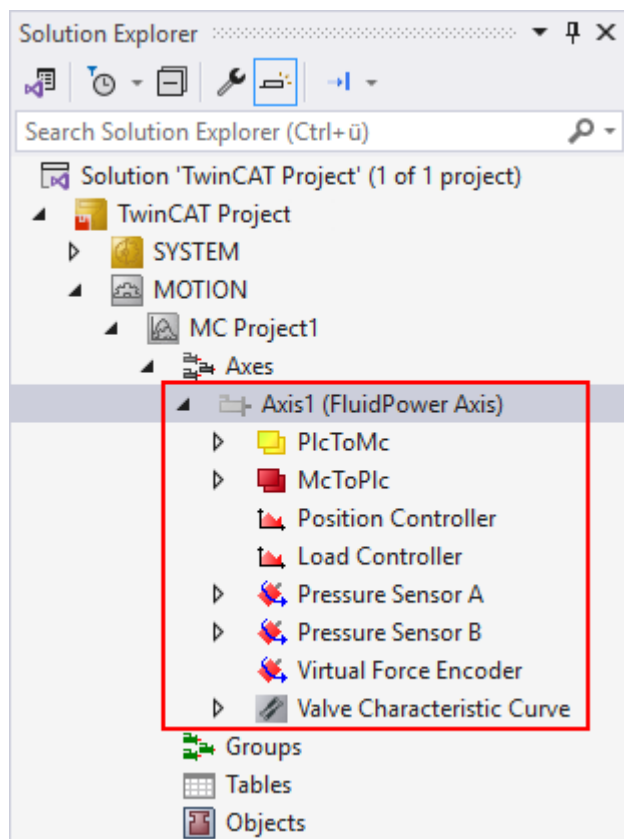
1. Machen Sie einen Rechtsklick auf **Axes > Add New Item...**



- ⇒ Wählen Sie im Dialog **Insert TcCom Objekt** unter **Type** den Achstyp **FluidPower Axis** aus und bestätigen Sie mit **OK**.



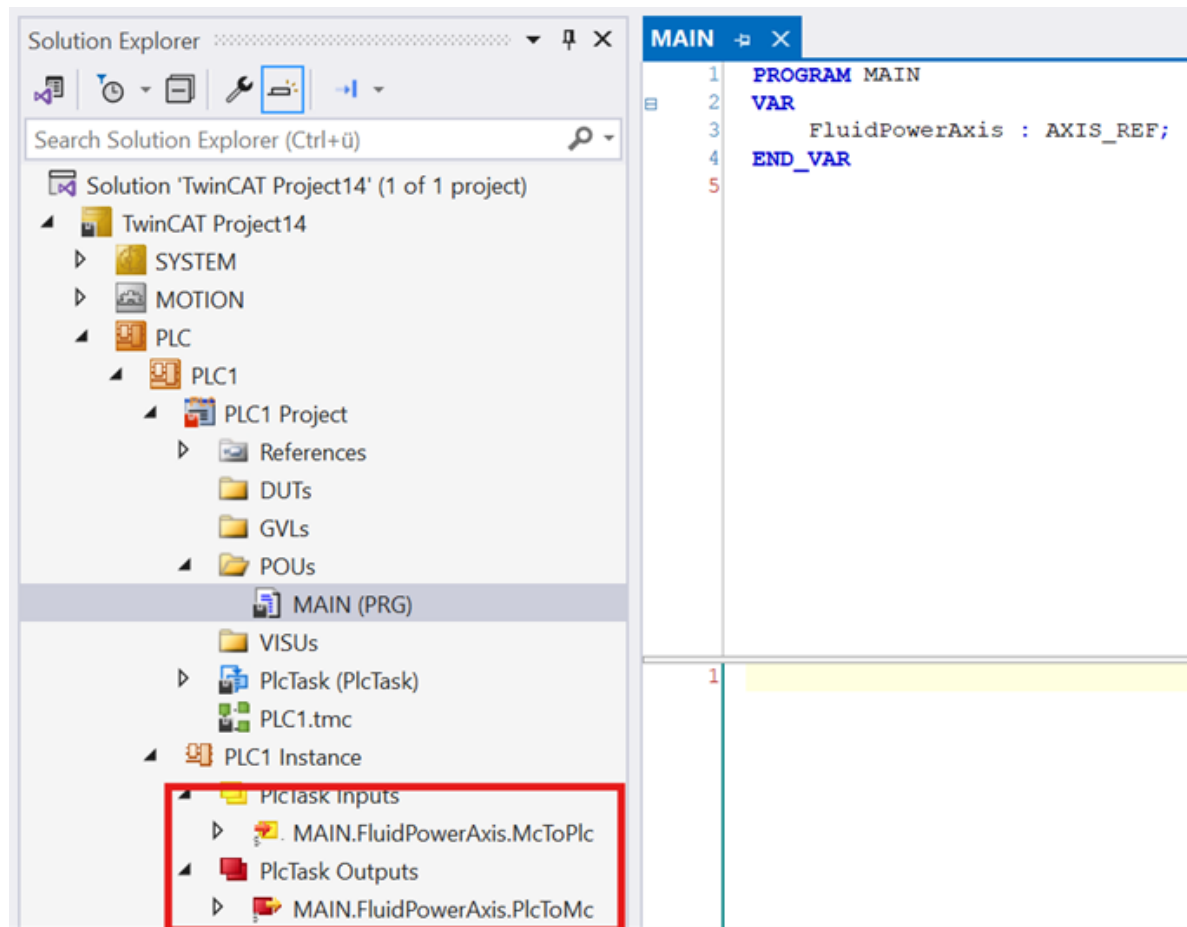
Die FluidPower-Achse wurde mit den benötigten Modulen angelegt und befindet sich unterhalb des Axes-Knoten.



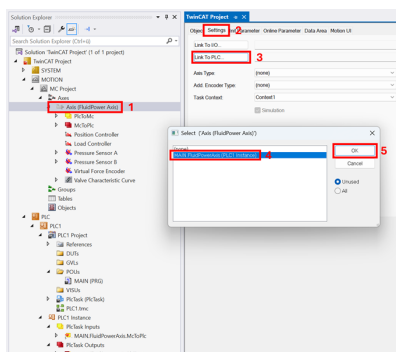
3.2.5.1.2 Achsobjekt mit der PLC verknüpfen

- ✓ Ein Achsobjekt wurde angelegt,
siehe [Achsobjekt erstellen/Achse hinzufügen](#) [► 140].
 - ✓ Ein PLC-Projekt mit einer Instanz des Tc3_Mc3Ptp.AXIS_REF wurde angelegt.
1. Kompilieren Sie das PLC-Projekt.

⇒ Die Instanz `FluidPowerAxis` des `Tc3_Mc3PtP.AXIS_REF` wird nun unter den PLC-Instanzen im Solution Explorer angezeigt.



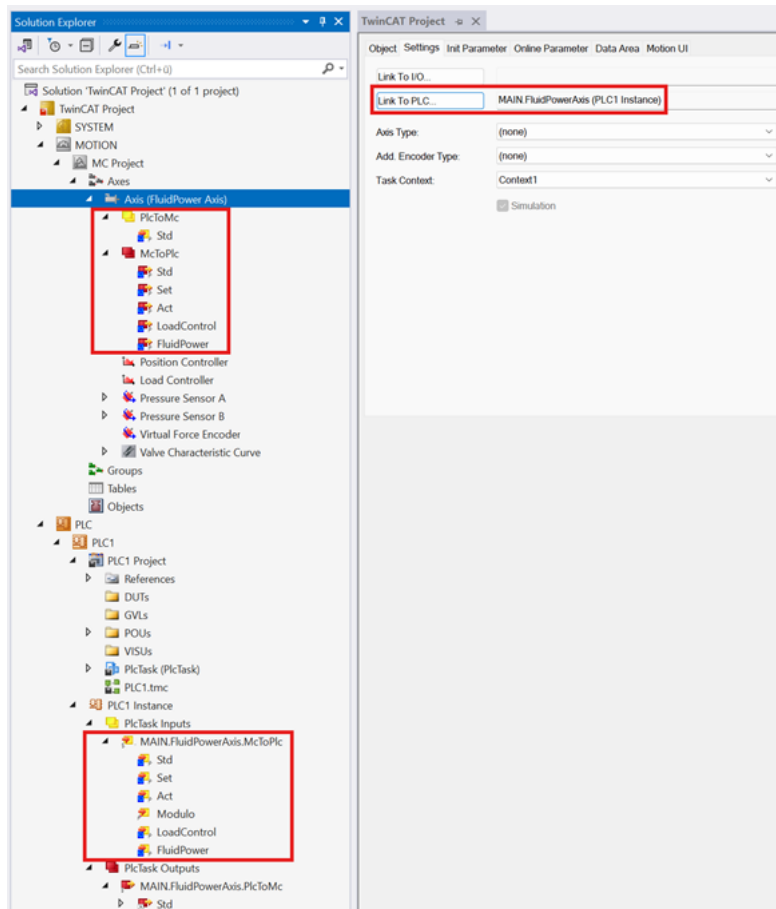
2. Verknüpfen Sie die PLC-Instanz des `Tc3_Mc3Ptp.AXIS_REF` mit der Achsinstanz der MC3 (Schritte 1-5).



⇒ Die Achse ist nun mit der PLC verknüpft.

⇒ Das Textfeld neben der Schaltfläche Link To PLC... zeigt die verknüpfte Instanz der AXIS_REF in der PLC an.

- ⇒ In der Konfiguration des Solution Explorers im Projektbaum ist eine verknüpfte Variable aus dem zyklischen Interface durch ein blaues oder weißes Symbol erkenntlich.



3. Aktivieren Sie das TwinCAT-Projekt.

3.2.5.1.3 Achsobjekt mit der Hardware verknüpfen

Vorgehensweise

- Für eine Anleitung zur Verknüpfung – sowohl das Automatische als auch das Manuelle - eines Achsobjektes mit der Hardware, siehe [Achsobjekt mit Hardware verknüpfen](#) [► 124].
- In den Einstellungen der Achse können der Achstyp, der zusätzliche Encodertyp sowie der Task-Kontext zugeordnet werden. Je nach Auswahl ändert sich die jeweilige Struktur im Achsobjekt.
 - Details zu den Achstypen sind unter [Drive-Module](#) [► 42] zu finden.
 - Details zu den Encodern sind unter [Encoder-Module](#) [► 82] zu finden.
 - Details zum Task-Kontext sind unter [Registerkarte Context](#) [► 19] zu finden.
- In der Hydraulik sind in der Regel folgende Achstypen vorzufinden:
 - [Analog Drive](#) [► 73] – wird genutzt, sobald ein analoges Proportionalventil eingesetzt wird.
 - [CoE DS408 Drive \(hydraulic\)](#) [► 70] – wird genutzt, sobald ein Proportionalventil mit einem DS408-Profil eingesetzt wird.
 - [Pulse Width Current Drive \(MDP250\) \(both channels\)](#) [► 79] – wird genutzt, sobald ein Proportionalventil mit Pulsweitenmodulation (PWM) eingesetzt wird.
- In der Hydraulik sind in der Regel folgende Encodertypen vorzufinden:
 - [EtherCAT Encoder](#) [► 82] – wird genutzt, wenn ein Wegmesssystem mit EtherCAT-Schnittstelle eingesetzt wird.
 - [SSI Encoder](#) [► 90] – wird genutzt, wenn ein Wegmesssystem mit SSI-Schnittstelle eingesetzt wird.

- In der Hydraulik sind in der Regel analoge Sensoren für die Kraft-/Druckregelung vorzufinden, deren Load Type und Skalierung einzustellen sind:
 - [Analog Load Encoder \[► 88\]](#) – wird genutzt, sofern analoge (Spannung/Strom) Drucksensoren genutzt werden.

3.2.5.2 Parameter einer Achse

Allgemein wird zwischen Parametern unterschieden, die nur bei der Initialisierung bzw. auch Online angepasst werden können.

Weiterhin können die Parameter einer FluidPower-Achse in zwei Bereiche unterteilt werden:

- Standardparameter, die jenen einer elektrischen Achse entsprechen
 - Init Parameter siehe: [Registerkarte Init Parameter \[► 145\]](#).
 - Online-Parameter siehe: [Registerkarte Online Parameter \[► 148\]](#).
- Spezifische Parameter einer FluidPower-Achse
 - Init Parameter siehe: [Init Parameter \[► 149\]](#).

3.2.5.2.1 Registerkarte Init Parameter

General

Name	Standardwert	Einheit	Typ
Simulation Mode	TRUE		BOOL
Position Unit	mm	[PosUnit]	MC.EPositionUnit [► 207]
Is Modulo Axis	FALSE		BOOL
Position Modulus	360.0	[PosUnit]	LREAL
Position Modulo Tolerance Window	0.0	[PosUnit]	LREAL

Simulation Mode

Der Simulationsmodus zeigt an, ob die I/O-Verbindung zum Antrieb und Encoder aktiv (**FALSE**) oder simuliert (**TRUE**) sein soll. Der Parameter kann während der Laufzeit geschrieben werden, wenn die Achse ohne Reglerfreigabe ist.

Position Unit

Die Position Unit wird für Anzeigezwecke, z. B. auf der Inbetriebnahmeoberfläche, verwendet. Ein- und Ausgaben an Funktionsbausteinen werden nicht konvertiert.

Is Modulo Axis

Aktivieren Sie diesen Parameter, um für diese Achse Modulo-Funktionalitäten an Funktionsbausteinen zu nutzen. Sollwert und Istwert werden in der zyklischen Schnittstelle als Absolut- und Modulo-Wert berechnet. Um Modulo-Informationen z. B. mit der PLC auszutauschen, ist die [Data Area Modulo \[► 33\]](#) zu aktivieren.

- **Position Modulus**
Modulo-Wert zur Berechnung der Modulo-Position
- **Position Modulo Tolerance Window**
Eine Bewegung entgegen der angegebenen Richtung ist zulässig, sofern die Zielposition innerhalb dieser Positionstoleranz der Startposition des Kommandos liegt. Dieser Parameter hat einen Einfluss auf das Verhalten des [MC_MoveModulo \[► 278\]](#). Er steht in Wechselwirkung mit der gewählten Einstellung am Eingang *Direction* des Funktionsbausteins.

Maximum Dynamics

Name	Standardwert	Einheit	Typ
Maximum Velocity	2000.0	[PosUnit]/s	LREAL

Name	Standardwert	Einheit	Typ
Maximum Acceleration	10000.0	[PosUnit]/s ²	LREAL
Maximum Deceleration	10000.0	[PosUnit]/s ²	LREAL
Maximum Jerk	100000.0	[PosUnit]/s ³	LREAL

Maximum Velocity

Maximale Geschwindigkeit, die bei der Profilerzeugung berücksichtigt wird. Wird die Konstante #Maximum am Eingang Velocity eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Geschwindigkeit als Grenzwert verwendet.

Maximum Acceleration

Maximale Beschleunigung, die bei der Profilerzeugung berücksichtigt wird. Wird die Konstante #Maximum am Eingang Acceleration eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Beschleunigung als Grenzwert verwendet.

Maximum Deceleration

Maximale Verzögerung, die bei der Profilerzeugung berücksichtigt wird. Wird die Konstante #Maximum am Eingang Deceleration eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Verzögerung als Grenzwert verwendet.

Maximum Jerk

Maximaler Jerk, der bei der Profilerzeugung berücksichtigt wird. Wird die Konstante #Maximum am Eingang Jerk eines Funktionsbausteins angegeben, wird der hier definierte Wert für den Jerk als Grenzwert verwendet.

Default Dynamics

Name	Standardwert	Einheit	Typ
Default Acceleration	2000.0	[PosUnit]/s ²	LREAL
Default Deceleration	2000.0	[PosUnit]/s ²	LREAL
Default Jerk	4000.0	[PosUnit]/s ³	LREAL

Default Acceleration

Dieser Wert wird als Beschleunigung am Eingang Acceleration an Funktionsbausteinen verwendet, sofern er nicht explizit gesetzt ist. Wird die Konstante #Default am Eingang eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Beschleunigung als Grenzwert verwendet.

Default Deceleration

Dieser Wert wird als Verzögerung am Eingang Deceleration an Funktionsbausteinen verwendet, sofern er nicht explizit gesetzt ist. Wird die Konstante #Default am Eingang eines Funktionsbausteins angegeben, wird der hier definierte Wert für die Verzögerung als Grenzwert verwendet.

Default Jerk

Dieser Wert wird als Jerk am Eingang Jerk an Funktionsbausteinen verwendet, sofern er nicht explizit gesetzt ist. Wird die Konstante #Default am Eingang eines Funktionsbausteins angegeben, wird der hier definierte Wert für den Jerk als Grenzwert verwendet.

Manual Motion

Name	Standardwert	Einheit	Typ
Jog Velocity Slow	100.0	[PosUnit]/s	LREAL
Jog Velocity Fast	250.0	[PosUnit]/s	LREAL

Jog Velocity Slow

Diese Zielgeschwindigkeit wird für das langsame Verfahren einer Achse mit Handbedientasten verwendet. Das Verfahren ist mithilfe von `MC_Jog` sowie über die Inbetriebnahmeoberfläche möglich.

Jog Velocity Fast

Diese Zielgeschwindigkeit wird für das schnelle Verfahren einer Achse mit Handbedientasten verwendet. Das Verfahren ist mithilfe von `MC_Jog` sowie über die Inbetriebnahmeoberfläche möglich.

Monitoring

Name	Standardwert	Einheit	Typ
Position Limit Enforcement Enabled	FALSE		BOOL
Minimum Position	#Ignore	[PosUnit]	LREAL
Maximum Position	#Ignore	[PosUnit]	LREAL
Position Lag Monitoring Enabled	TRUE		BOOL
Maximum Position Lag Value	5.0	[PosUnit]	LREAL
Maximum Position Lag Filter Time	0.02	s	LREAL
Position Lag Calculation	HardwareProvidedIfAvailable		MC.EPositionLagSource ► 206

Position Limit Enforcement Enabled

Dieser Parameter gibt an, ob Positionsgrenzen bei der Generierung von Bewegungsprofilen berücksichtigt werden. Wenn der Wert `TRUE` ist, sind die folgenden beiden Parameter einzustellen.

- Minimum Position**
 Minimal zulässige Position für die Achse. Verwenden Sie `#Ignore`, um die Einhaltung der Positionsgrenze in diese Bewegungsrichtung auszuschalten. Es ist nicht möglich, eine Achse außerhalb dieses Limits zu verfahren. Steht eine Achse beim TwinCAT Start außerhalb ihrer Positionslimits, ist eine Bewegung in Richtung des gültigen Positionsbereichs möglich. Kommandos mit einer Zielposition außerhalb dieser Untergrenze werden abgelehnt.
- Maximum Position**
 Maximal zulässige Position für die Achse. Verwenden Sie `#Ignore`, um die Einhaltung der Positionsgrenze in diese Bewegungsrichtung auszuschalten. Es ist nicht möglich, die Achse außerhalb dieses Limits zu verfahren. Steht eine Achse beim TwinCAT Start außerhalb ihrer Positionslimits, ist eine Bewegung in Richtung des gültigen Positionsbereichs möglich. Kommandos mit einer Zielposition außerhalb dieser Untergrenze werden abgelehnt.

Position Lag Monitoring Enabled

Dieser Parameter gibt an, ob der Schleppfehler für die Achse überwacht werden soll. Wenn der Wert `TRUE` ist, werden die folgenden beiden Parameter angezeigt, um Position und Zeit zu parametrieren. Falls die parametrisierten Grenzen überschritten werden, wird ein Laufzeitfehler ausgegeben.

Die Berechnung erfolgt wie folgt:

Positions-Schleppfehler = Aktuelle Sollposition - Istposition

- Maximum Position Lag Value und Maximum Position Lag Filter Time**
 Maximum Position Lag Value ist der Positions-Schleppfehler, welcher nicht länger als die festgelegte Zeit (Maximum Position Lag Filter Time) überschritten werden darf. Andernfalls wird die Achse durch direktes Abschalten unverzüglich gestoppt und ein Fehler ausgegeben.

Position Lag Calculation

Dieser Parameter gibt an, ob der Schleppfehler softwareseitig von der MC-Achse berechnet (`InternallyCalculated`) oder direkt aus der verknüpften Hardware ausgelesen wird (`HardwareProvidedIfAvailable`). Ist `HardwareProvidedIfAvailable` eingestellt aber keine Hardware verfügbar, wird auf den intern berechneten Wert zurückgegriffen. Für eine Simulationsachse werden bei `HardwareProvidedIfAvailable` Daten vom Simulationsdrive abgegriffen.

Error Event Handling

Name	Standardwert	Einheit	Typ
Control Error Setpoint Mode	HardStop		MC.EControlErrorSetpoint Mode [► 201]

Wenn ein Fehler auf der Hardwareseite vorliegt (z. B. im Regelkreis) und die MC keine Kontrolle mehr über die Achse hat, können Sie hier das Verhalten der Sollwertgenerierung einstellen. `HardStop` erzeugt ein sofortiges hartes Anhalten der Achse im Fehlerfall.

Diagnostics

Name	Standardwert	Einheit	Typ
TraceLevel	tlWarning		TcTraceLevel

Legen Sie fest, welche Nachrichten an die TwinCAT-Fehlerliste gesendet werden sollen.

Additional Encoder

Name	Standardwert	Einheit	Typ
Main Position Encoder			OTCID_PositionEncoder
Load Control Type	None		MC.ELoadType [► 205]

Main Position Encoder

Falls ein zusätzlicher Encoder vorhanden ist, wählen Sie hier die Objekt-ID aus.

Load Control Type

Stellen Sie hier die Art der Last ein, die geregelt werden soll.

3.2.5.2.2 Registerkarte Online Parameter

Es gibt die folgenden Online-Parameter:

Monitoring

Name	Einheit	Type
Minimum Position Lag	[PosUnit]	LREAL
Maximum Position Lag	[PosUnit]	LREAL
Trigger Minimum/Maximum Position Lag Reset		Trigger Minimum/Maximum Position Lag Reset

Minimum Position Lag

Dieser schreibgeschützte Online-Parameter zeigt den minimalen erkannten Schleppfehler zu Informationszwecken an.

Maximum Position Lag

Dieser schreibgeschützte Online-Parameter zeigt den maximalen erkannten Schleppfehler zu Informationszwecken an.

Trigger Minimum/Maximum Position Lag Reset

Eine steigende Flanke an diesem Parameter Eingang setzt gespeicherte minimale und maximale Schleppfehler zurück.

Additional Error Event Information

Name	Einheit	Type
Originating Error Event Sender Id		OTCID
Original Error Id		HRESULT

Originating Error Event Sender Id

Liegt ein Folgefehler z. B. durch eine Kopplung vor, zeigt dieser Parameter die Objekt-ID der Achse an, von der der Fehler ausging.

Original Error Id

Liegt ein Folgefehler z. B. durch eine Kopplung vor, zeigt dieser Online-Parameter die ursprüngliche Fehlernummer an.

Override

Name	Einheit	Type
Velocity Override Factor		LREAL

Velocity Override Factor

Hier wird der aktuelle Geschwindigkeits-Override der Achse angezeigt.

3.2.5.2.3 Init Parameter

Cylinder Dimensions

Name	Standardwert	Einheit	Typ
Cylinder Design	Differential		MC.ECylinderDesign
Cylinder Bore Diameter	#Ignore	mm	LREAL
Cylinder Rod Diameter	#Ignore	mm	LREAL

Cylinder Design

Der Parameter `Cylinder Design` wählt den Typ des Hydraulikzylinders aus und legt fest, wie Kolben- und Kolbenstangendurchmesser für Kraft- und Flächenberechnungen verwendet werden.

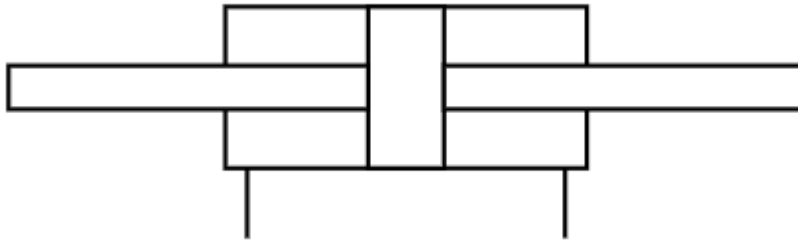
`Differential`

Der Differentialzylinder ist ein doppeltwirkender Hydraulikzylinder mit einseitiger Kolbenstange.



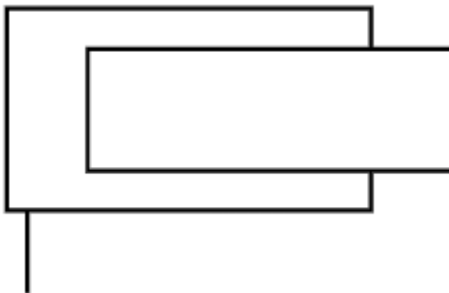
`Synchronous`

Der Synchronzylinder ist ein doppeltwirkender Hydraulikzylinder mit durchgehender Kolbenstange.



Plunger

Der Plungerzylinder ist ein einfachwirkender Hydraulikzylinder ohne Kolbenstange.



Cylinder Bore Diameter

Innendurchmesser der Hydraulikzylinderbohrung. Wird zur Berechnung der Kolbenfläche und der Kraft herangezogen.

Cylinder Rod Diameter

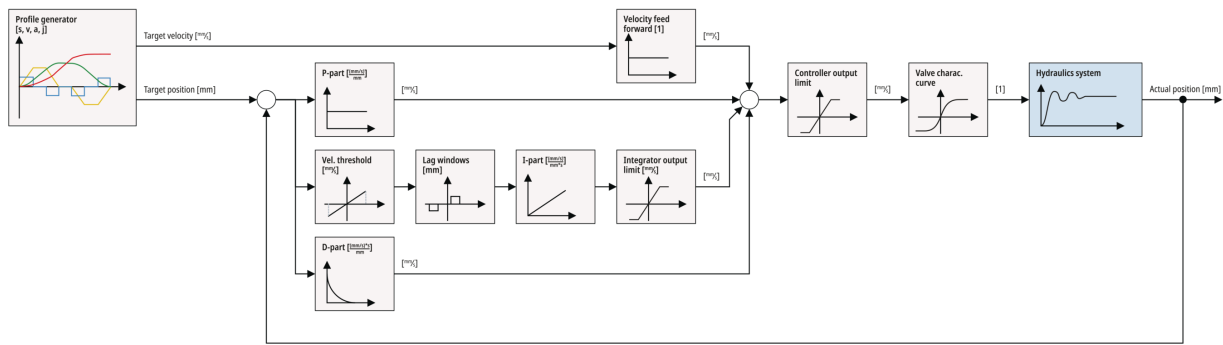
Durchmesser der Zylinderstange. Relevant für die Konstruktion von Differential- und Synchronzylindern zur Berechnung der effektiven Kolbenfläche auf der Stangenseite.

3.2.5.3 Positionsregler

Der Sollwertgenerator basiert auf einem **7-Phasen-Profil** zur definierten Trajektorienvorgabe, welche Beschleunigungs-, Geschwindigkeits- und Verzögerungssollwerte berücksichtigt. Die Regelung erfolgt über einen **klassischen PID-Regler**, wobei der **Integrationsanteil (I)** mit spezifischen Anpassungen versehen ist, um Integrationsüberschwingen, Schwebung und Drift zu vermeiden.

Zur Verbesserung des dynamischen Verhaltens wird eine **Geschwindigkeitsvorsteuerung** implementiert, die den Regler entlastet und die Reaktionszeit auf Sollwertänderungen verkürzt. Im Anschluss erfolgt die **nachgelagerte Reglerlimitierung** sowie die **Einbindung der Ventilkennlinie** als nichtlineare Stellgrößenkompensation. Dadurch werden auch die charakteristischen Durchflussskennwerte des verwendeten Ventils in die Regelung integriert, was eine linearisierte Stellgrößenwirkung und höhere Genauigkeit im gesamten Arbeitsbereich gewährleistet.

Das folgende Blockschaltbild zeigt die grafische Darstellung des Positionsreglers:



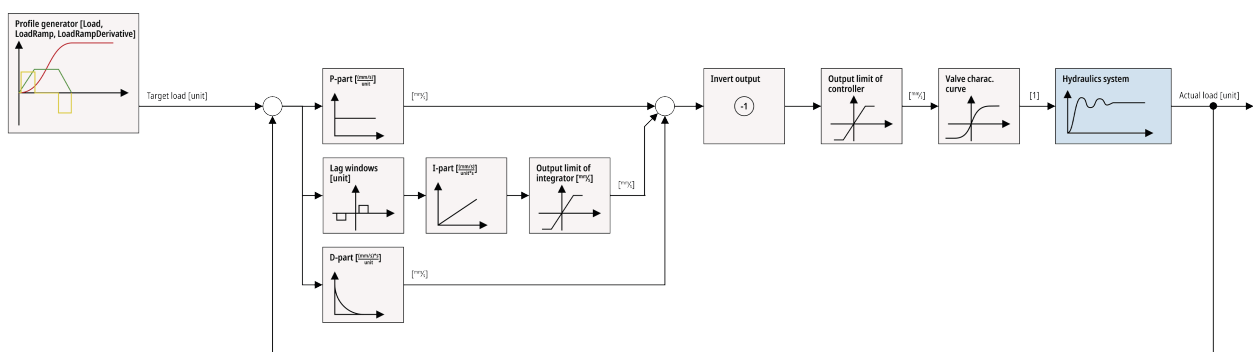
- Erklärungen zu den Regelparametern, siehe [Init Parameter](#) [► 98].
- Erläuterungen zu den Onlineparametern, siehe [Online Parameter](#) [► 100]

3.2.5.4 Kraft-/Druckregler

Der Sollwertgenerator basiert auf einem **5-Phasen-Profil** zur definierten Trajektorienvorgabe, welche Druckrampen- sowie dessen Ableitung (Derivative) berücksichtigt. Die Regelung erfolgt über einen **klassischen PID-Regler**, wobei der **Integrationsanteil (I)** mit spezifischen Anpassungen versehen ist, um Integrationsüberschwingen, Schwebung und Drift zu vermeiden.

Im Anschluss an den Regler erfolgt eine optionale **Reglerinvertierung**, die **nachgelagerte Reglerlimitierung** sowie die **Einbindung der Ventilkennlinie** als nichtlineare Stellgrößenkompensation. Dadurch werden auch die charakteristischen Durchflusskennwerte des verwendeten Ventils in die Regelung integriert, was eine linearisierte Stellgrößenwirkung und höhere Genauigkeit im gesamten Arbeitsbereich gewährleistet.

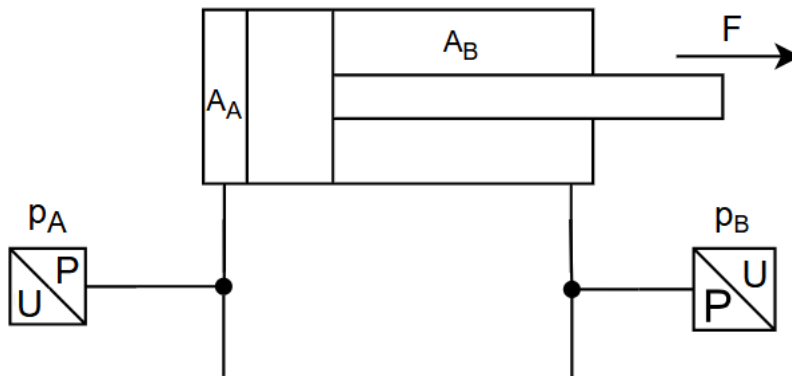
Das Blockschaltbild zeigt die grafische Darstellung des Kraft-/Druckreglers:



- Erklärungen zu den Regelparametern siehe hier: [Init Parameter](#) [► 96]
- Erläuterungen zu den Onlineparametern finden Sie hier: [Online Parameter](#) [► 97]

3.2.5.5 Drucksensor A / B

Zur Erfassung von Drücken innerhalb eines Hydrauliksystems werden Drucksensoren verwendet. Die FluidPower-Achse sieht hierfür die Drucksensoren A und B vor, wobei der Drucksensor A den Druck auf der Kolbenseite und der Sensor B den Druck auf der Stangenseite eines Hydraulikzylinders bestimmt.



Die Parameter zur Konfiguration der Drucksensoren finden Sie hier: [Analog Load Encoder \[► 88\]](#)

3.2.5.5.1 Verknüpfung | Analoge Eingänge

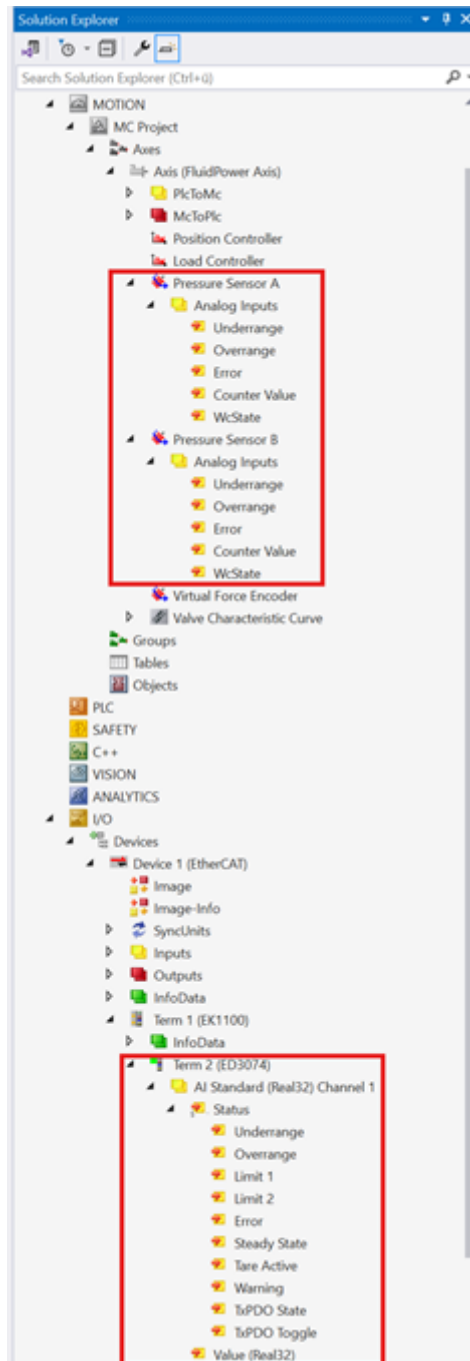
Voraussetzungen

- Eine MC3-FluidPower-Achse wurde im TwinCAT-Projekt hinzugefügt, siehe [Achsojekt erstellen/Achse hinzufügen \[► 140\]](#), oder
- Eine MC3-Achse wurde um ein Analog Load Encoder Objekt erweitert, siehe [Achsojekt um ein zusätzliches Modul erweitern \[► 119\]](#)

Ausgangssituation

- Hardwarekonfiguration wurde angelegt.

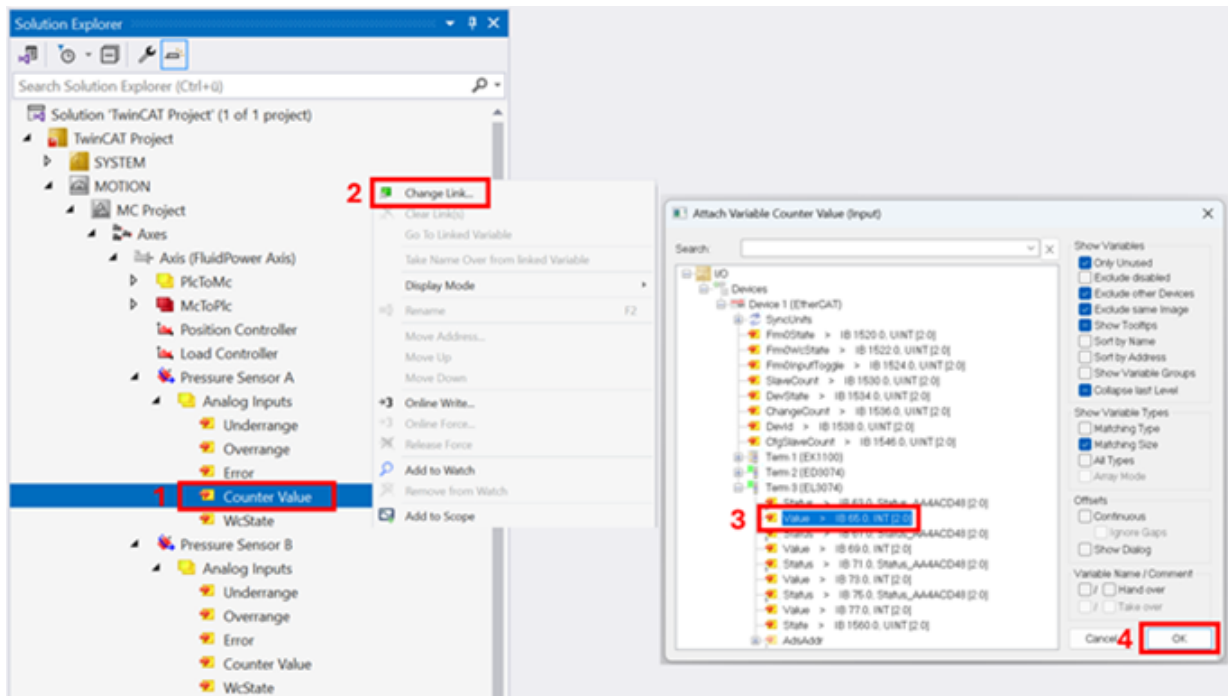
- Drucksensor A/B wurde angelegt.



Vorgehen: Beispiel für EL307x, ED307x

1. Wählen Sie den zu verknüpfenden Link aus.
2. Machen Sie einen Rechtsklick auf **Change Link...**
3. Selektieren Sie den zu verknüpfenden Link der Hardware.

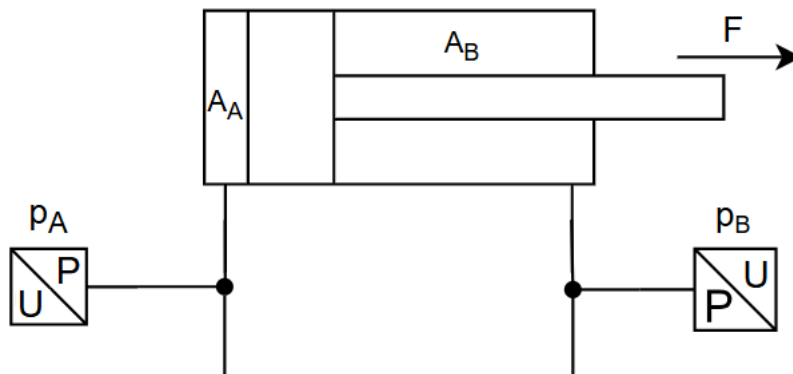
4. Bestätigen Sie die Auswahl mit dem Button **OK**.



3.2.5.6 Virtueller Kraftsensor

Ein **Virtueller Kraftsensor** wird bei Konfiguration einer FluidPower-Achse zur Berechnung der Kraft verwendet.

Die Kraft wird mit Hilfe der parametrisierten Zylinder Daten Init Parameter [► 149] und den zugehörigen Drücken A / B Drucksensor A / B [► 152] berechnet.



Aus den Zylinderflächen A_A / A_B und den Drücken p_A / p_B ergibt sich die Kraft F wie folgt.

$$F = A_A \cdot p_A - A_B \cdot p_B$$

3.2.5.6.1 Virtual Force Encoder - Init Parameter

Scaling

Name	Standardwert	Einheit	Typ
Direction Inverted	FALSE		BOOL

Direction Inverted

Der Parameter `Direction Inverted` kann die Zählrichtung der Kraft wie folgt anpassen:

- `FALSE`: Die Polarität der Kraft stimmt mit der Zählrichtung der Erfassungshardware überein.
- `TRUE`: Die Polarität der Kraft ist invers zur Zählrichtung der Erfassungshardware.

⚠ WARNUNG

Verletzungsgefahr durch unerwartete Bewegungen

Wenn die Zählrichtung des Gebers und die Polarität des Stellgliedes nicht miteinander übereinstimmen, kann die Achse eine unerwartete Bewegung ausführen.

- Stellen Sie sicher, dass bei Inbetriebnahme die Zählrichtung des Gebers korrekt eingestellt ist.
- Halten Sie grundsätzlich einen ausreichenden Sicherheitsabstand.

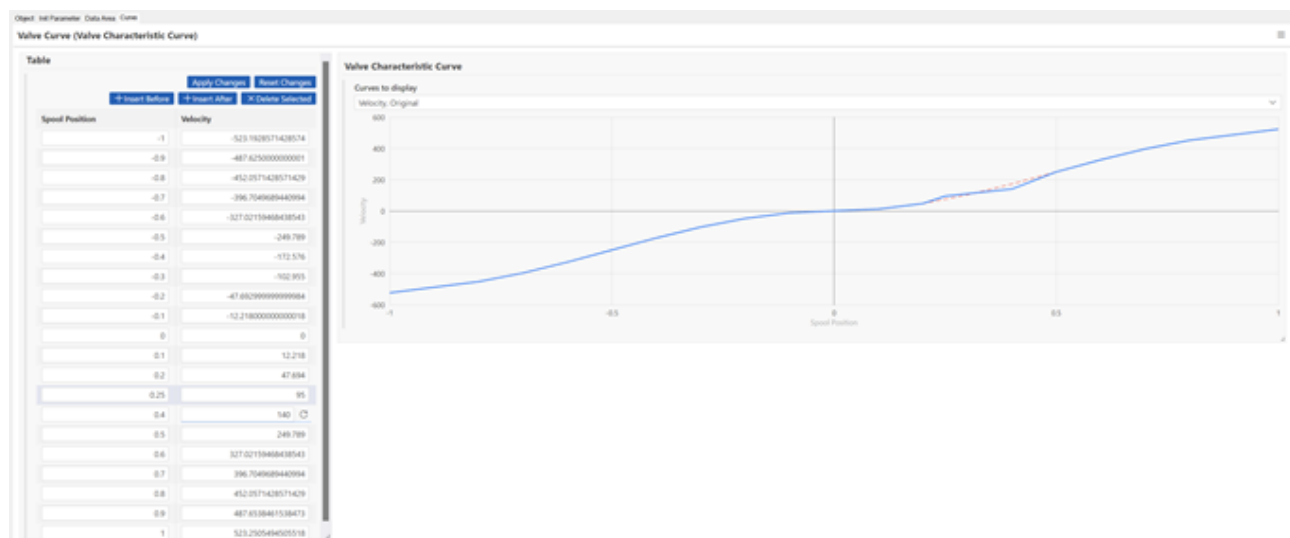
3.2.5.7 Ventilkennlinie

Die MC3-FluidPower-Achse ermöglicht die Implementierung und automatische Aufzeichnung der Ventilkennlinie. Diese ist dem Positions- bzw. Kraft-/Druckregler nachgeschaltet.

Durch die Ermittlung der Kennlinie im bewegten System werden alle nichtlinearen Eigenschaften des Hydrauliksystems erfasst und eliminiert. D. h. die Kennlinie beinhaltet neben der Durchflusskennlinie des Ventils auch die nichtlinearen Einflüsse des Ölflusses durch Hydraulikblöcke, -schläuche und -rohre und die des Hydraulikzylinders.

Curve

Im Tab **Curve** des Ventilkennlinien-Objekts wird eine vorhandene Ventilkennlinie tabellarisch und als Diagramm angezeigt.



Table

In der Karte „Table“ können Änderungen an bereits vorhandenen Punkten der Ventilkennlinie vorgenommen werden. Ebenso können über die Schaltflächen

- Insert Before
- Insert After
- Delete Selected

Punkte hinzugefügt oder entfernt werden.

Änderungen werden mit Hilfe der Schaltflächen

- Apply Changes
- Reset Changes

gespeichert oder zurückgesetzt.

Valve Characteristic Curve

In der Karte „Valve Characteristic Curve“ wird die aktuelle Ventilkennlinie grafisch dargestellt. Die Auswahl „Curves to display“ stellt die folgenden Optionen bereit:

- Velocity
- Original

Per Default ist „Velocity“ aktiv. Mit dieser Auswahl wird die aktuell parametrisierte Kurve angezeigt. Durch die Checkbox „Original“ wird zusätzlich die zuletzt gespeicherte Ventilkennlinie eingeblendet.

Informationen zu den Parametern des Ventilkennlinien-Objekts, siehe [Valve Characteristic Curve \[► 102\]](#).

Erläuterungen zum Menü, siehe [Einstellmöglichkeiten \[► 35\]](#).

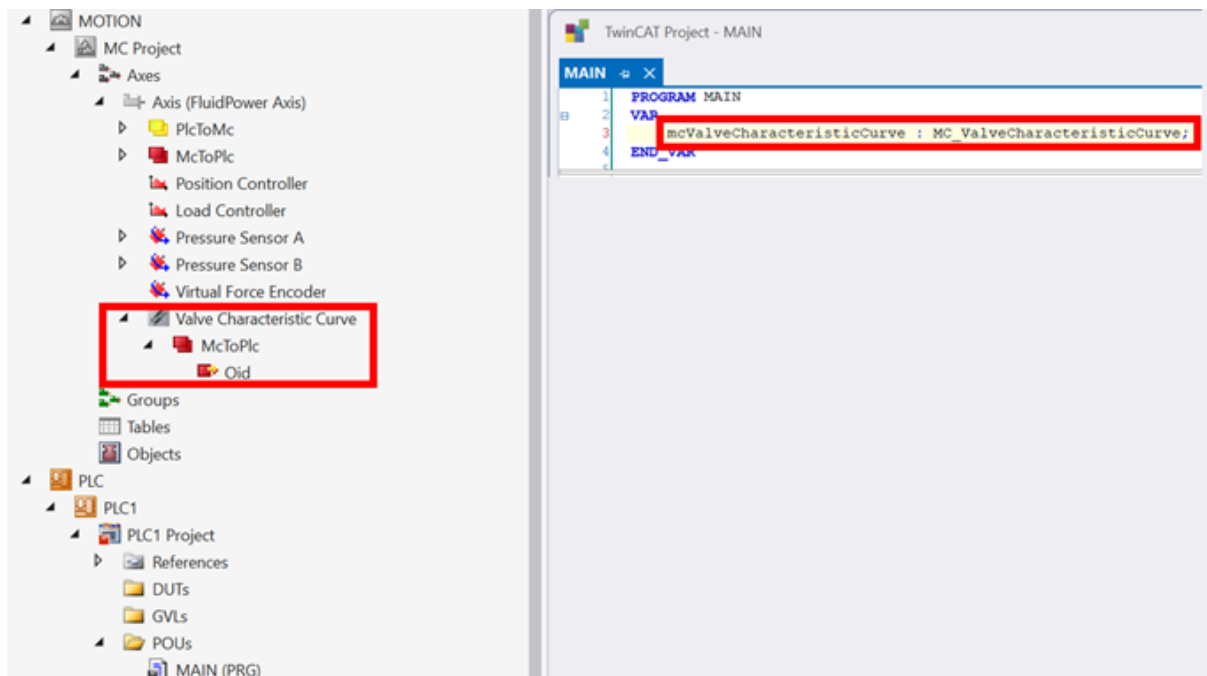
3.2.5.7.1 Verknüpfung MC_ValveCharacteristicCurve

Voraussetzungen

- Eine MC3-FluidPower-Achse wurde im TwinCAT-Projekt hinzugefügt, siehe [Achsojekt erstellen \[► 25\]](#), oder
- Eine MC3-Achse wurde um ein Valve Characteristic Curve Objekt erweitert, siehe [Achsojekt um ein zusätzliches Modul erweitern \[► 119\]](#)
- Ein PLC-Projekt mit einer Instanz des Funktionsbausteins Tc3_Mc3FluidPower.MC_ValveCharacteristicCurve wurde angelegt, siehe MC_ValveCharacteristicCurve

Ausgangssituation

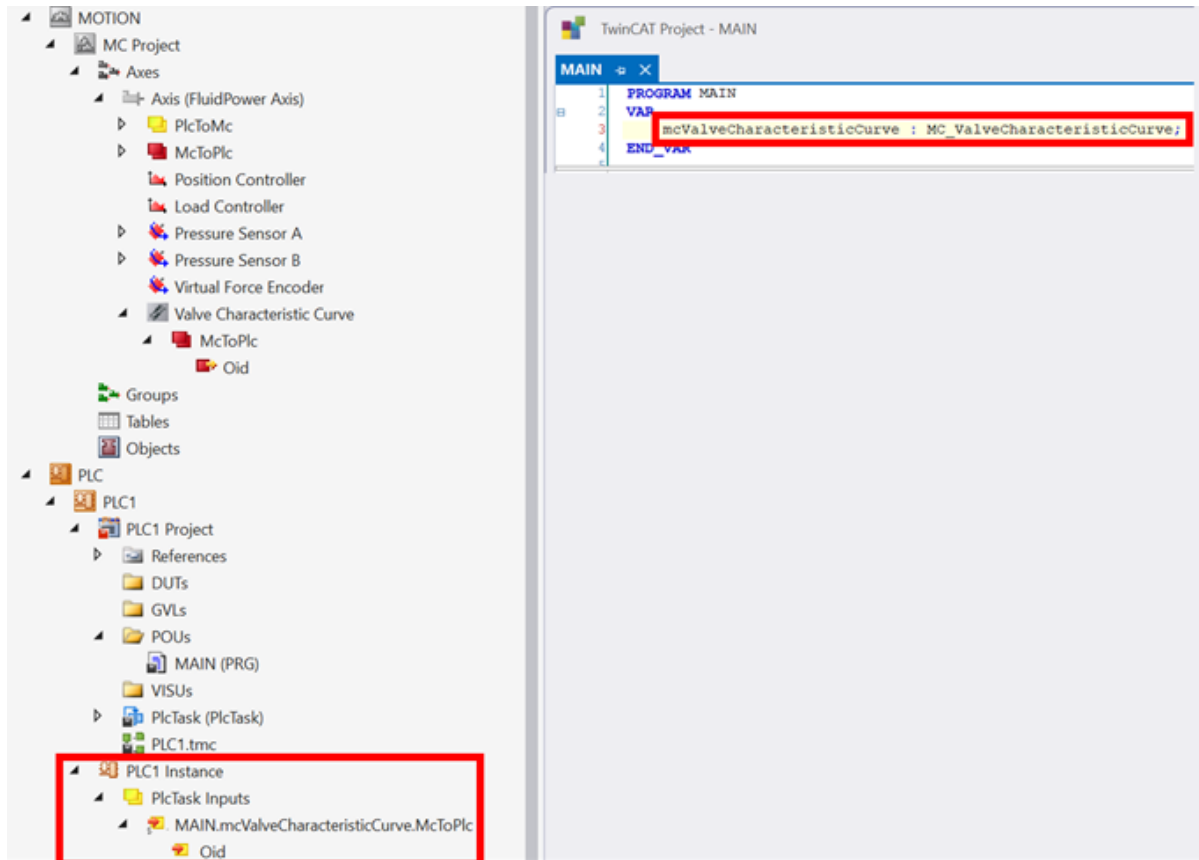
- Valve Characteristic Curve Objekt angelegt
- Instanz von Tc3_Mc3FluidPower.MC_ValveCharacteristicCurve angelegt



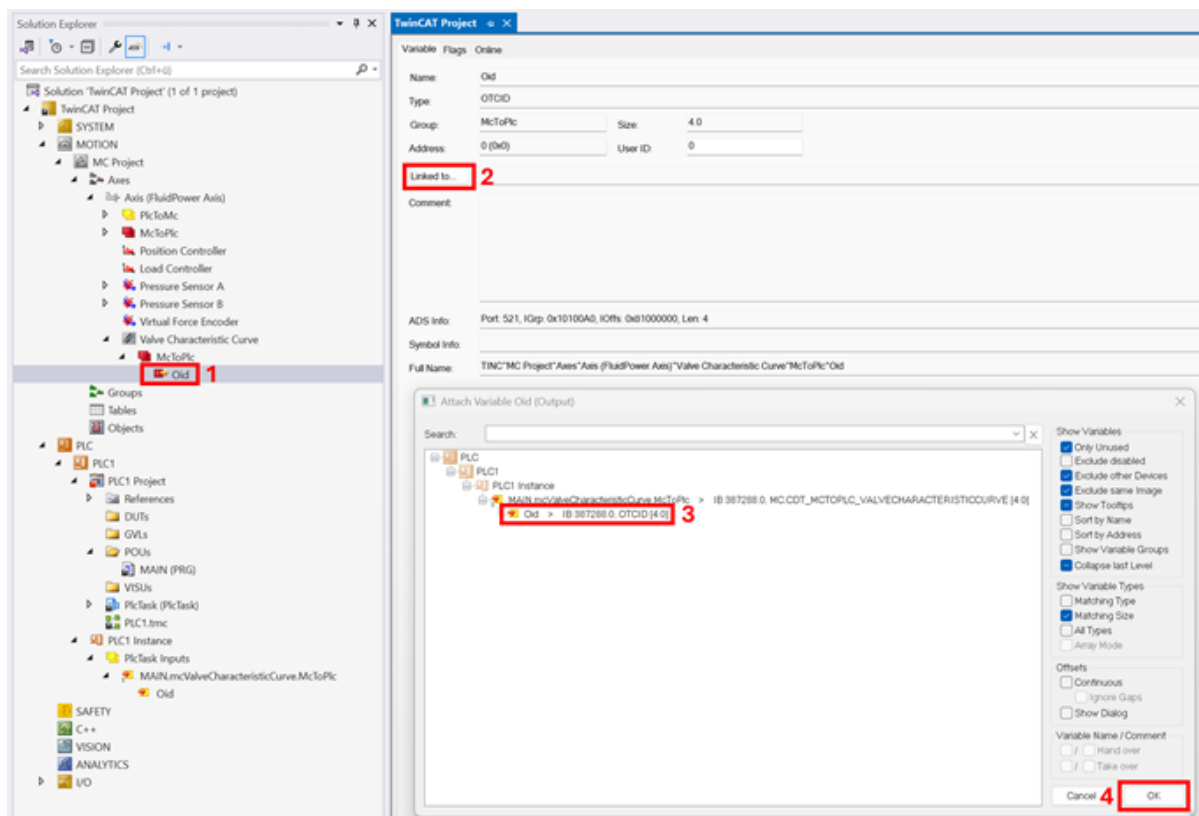
How to

- ## 1. Erstellen Sie das PLC-Projekt

→ Die Instanz des `Tc3_Mc3FluidPower.MC_ValveCharacteristicCurve` wird unter den PLC-Instanzen im Solution Explorer angezeigt.



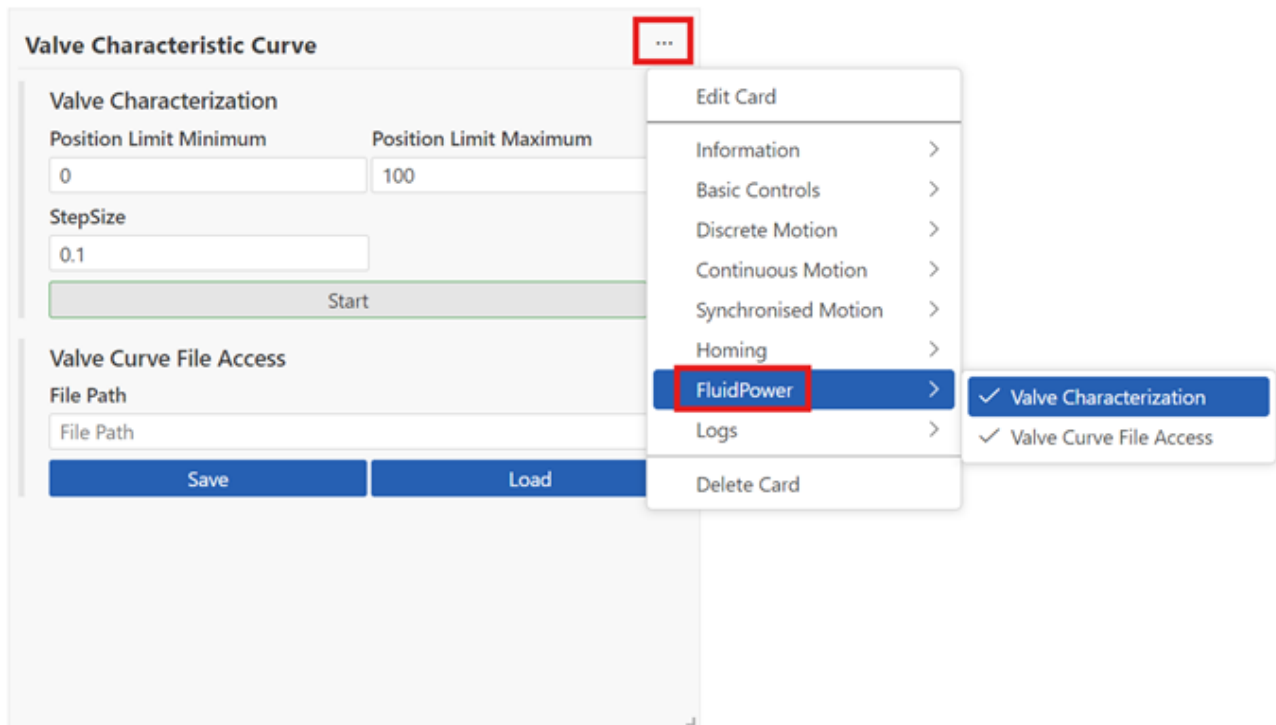
2. Verknüpfen Sie die PLC-Instanz mit der Valve Characteristic Curve-Instanz der MC3.



3. Aktivieren Sie das TwinCAT-Projekt.

3.2.5.7.2 Motion UI

Durch Klick auf „FluidPower“ im Menü der Karten [► 39] des Registerkarte Motion UI [► 34] können auf einer Karte Optionen zur Identifizierung und zum Datei-Handling der Ventilkennlinie eingeblendet werden.



3.2.5.7.2.1 Identifikation der Ventilkennlinie

Voraussetzungen

- Der Achs- und der Encodertyp wurde in den Einstellungen der MC3-Achse parametrisiert, siehe [Registerkarte Settings \[► 28\]](#)
- Die Geschwindigkeits-Vorsteuerung des Positionsreglers muss einen Wert von 1.0 aufweisen, siehe [Init Parameter \[► 98\]](#)

Vorgehensweise

Ist die MC3-Achse konfiguriert, kann die Ventilkennlinie mit Hilfe der Option „Valve Characterization“, ohne die Verwendung eines PLC-Projekts, über das Motion UI identifiziert werden.

The image shows two overlapping windows from the Beckhoff software. The background window is titled 'Valve Characterization' and contains input fields for 'Position Limit Minimum' (50), 'Position Limit Maximum' (950), and 'StepSize' (0.1). Below these fields is a 'Start' button. A red square highlights a gear icon to the right of the 'Start' button. The foreground window is titled 'Options' and contains a checkbox 'Use Options' which is checked. Below this, there are several parameter settings: 'Strategy' (Default), 'Wait Time after Velo Step' (0.02 s), 'Wait Time at Position Limit' (0.5 s), 'Min Travel Distance' (1 mm), 'Output Limit' (1), 'Acceleration' (2000 mm/s²), 'Deceleration' (2000 mm/s²), and 'Jerk' (4000 mm/s³).

1. Setzen Sie die gewünschten Parameter.
2. Ausführen der Identifikation durch Klick auf „Start“.
 - Die Schaltfläche wechselt zu „Halt“.
 - Nach erfolgreicher Identifikation der Ventilkennlinie wechselt die Schaltfläche zurück auf „Start“.
3. Speichern der Kennlinie, siehe [Dateizugriff](#) [► 159]

3.2.5.7.2.2 Dateizugriff

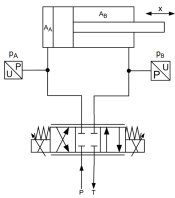
Über die Option „Valve Curve File Access“ kann die Ventilkennlinie in eine Datei gespeichert, bzw. aus einer Datei geladen werden.

The image shows a dialog box titled 'Valve Curve File Access'. It has a section labeled 'File Path' with a text input field containing the placeholder text 'File Path'. Below the input field are two large blue buttons: 'Save' and 'Load'.

- Klick auf „Save“ speichert die aktuelle Ventilkennlinie in einer Datei auf dem Zielsystem.
- Klick auf „Load“ lädt eine bereits vorhandene Ventilkennlinie aus einer Datei auf dem Zielsystem.
- „File Path“ gibt den Datei-Pfad auf dem Zielsystem vor.
 - Ohne Eingabe lautet der Default-Pfad: „C:\ProgramData\Beckhoff\TwinCAT\3.1\Boot\TcCom“.

3.2.5.8 FluidPower-Achse (ventilgeregelte Achse) in Betrieb nehmen

Die folgende Abbildung zeigt ein beispielhaftes Schaltbild eines Differentialzylinders, dessen Position und Druck/Kraft über ein 4/3-Wege-Proportionalventil zu regeln ist.



Im folgenden Abschnitt wird die Inbetriebnahme der Positions- und Druckregelung einer solchen Achse anhand eines Beispiels mit folgenden Komponenten Schritt für Schritt beschrieben.

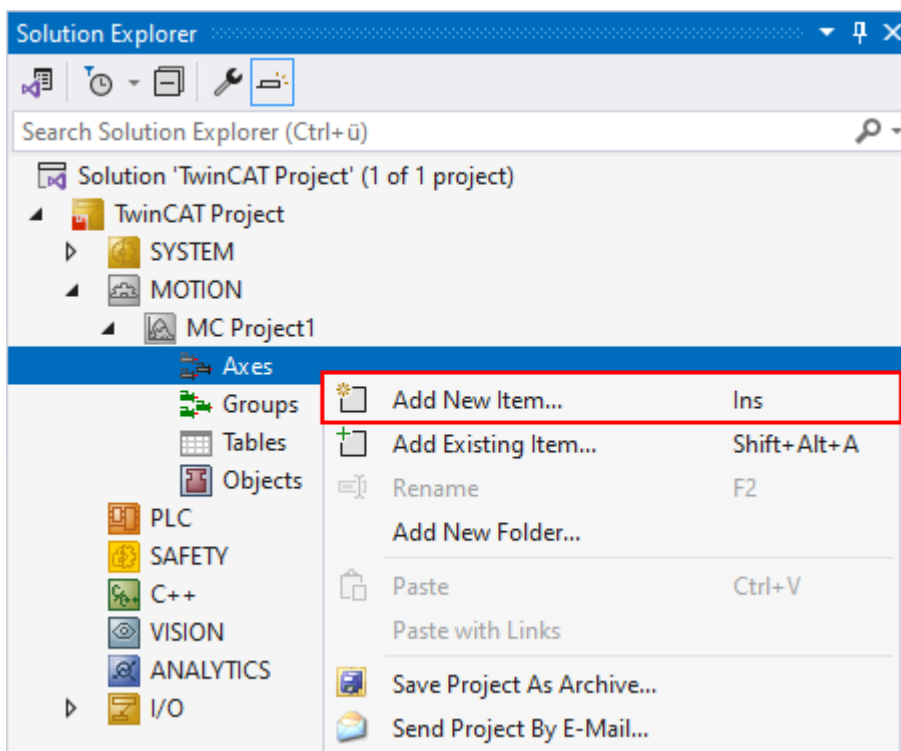
- Hydraulikdifferentialzylinder 50/32 mm
- Absolut SSI-Encoder, 0.1 μm Auflösung
- Analoges 4/3-Wege Proportionalventil mit analoger Ausgangsklemme
- Drucksensoren A/B, 0..400 bar mit analoger Eingangsklemme

Im ersten Schritt wird eine FluidPower-Achse angelegt, siehe [Achse hinzufügen](#) [► 160]

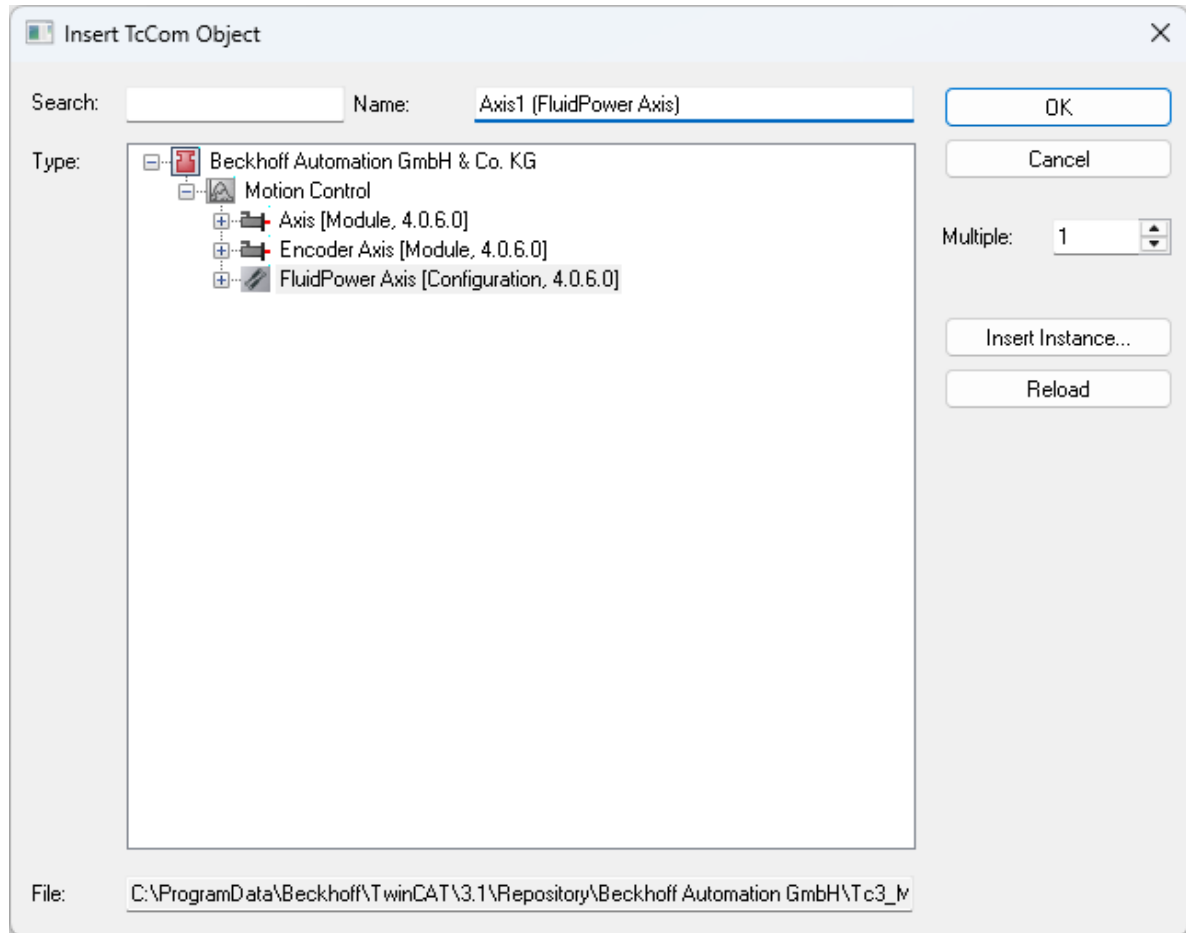
3.2.5.8.1 Achse hinzufügen

- ✓ Ein TwinCAT-Projekt mit MC-Projekt wurde erstellt, siehe [MC-Projekt erstellen](#) [► 17].

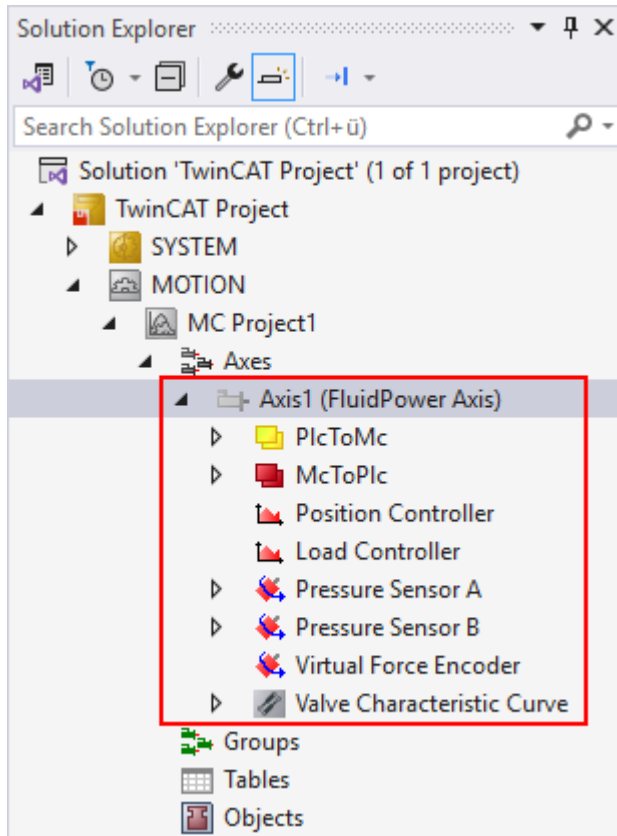
1. Machen Sie einen Rechtsklick auf **Axes > Add New Item...**



- ⇒ Wählen Sie im Dialog **Insert TcCom Objekt** unter **Type** den Achstyp **FluidPower Axis** aus und bestätigen Sie mit **OK**.

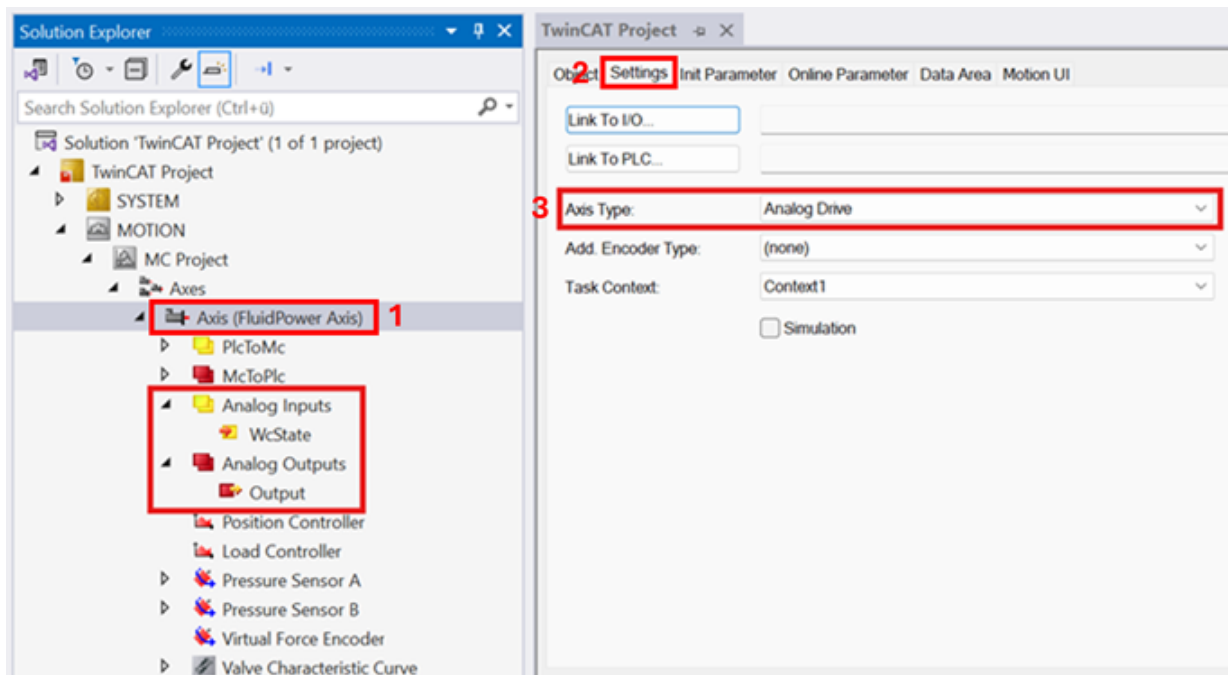


Die FluidPower-Achse wurde mit den benötigten Modulen angelegt und befindet sich unterhalb des Axes-Knoten.



Achs-Typ auswählen

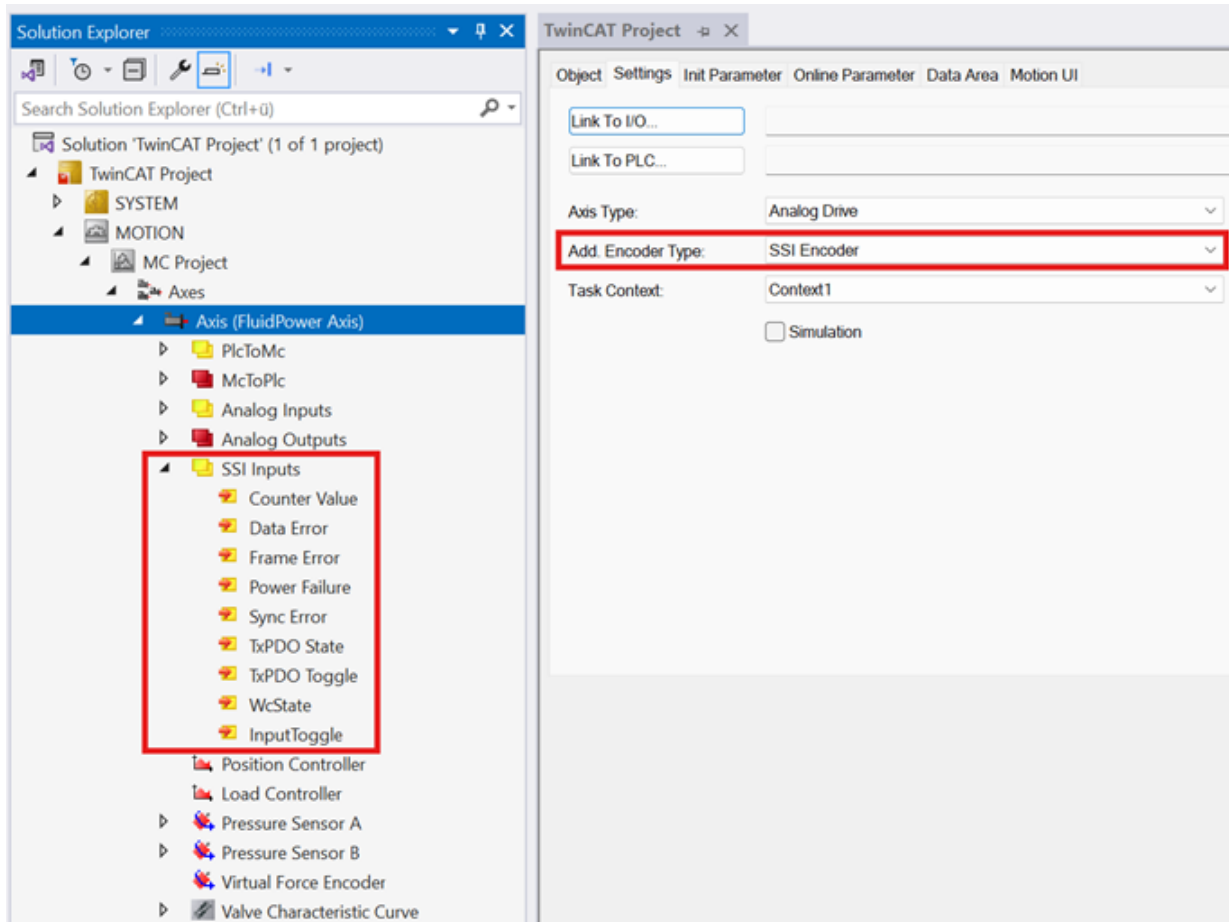
2. Auswahl des Achs Typs über die Registerkarte Settings [► 28] des Achsobjektes.



1. Doppelklick auf das **FluidPower Axis** Objekt
2. Öffnen der Registerkarte **Settings**
3. Auswahl eines **Analog Drive** zur Ansteuerung eines analogen Proportionalventils
→ Die Analog-Module wurden inkl. Data Area unterhalb der FluidPower-Achse angelegt.

Encoder Typ auswählen

3. Auswahl des Encoder Typs **SSI Encoder** über die Registerkarte Settings ► 28] des Achsobjektes



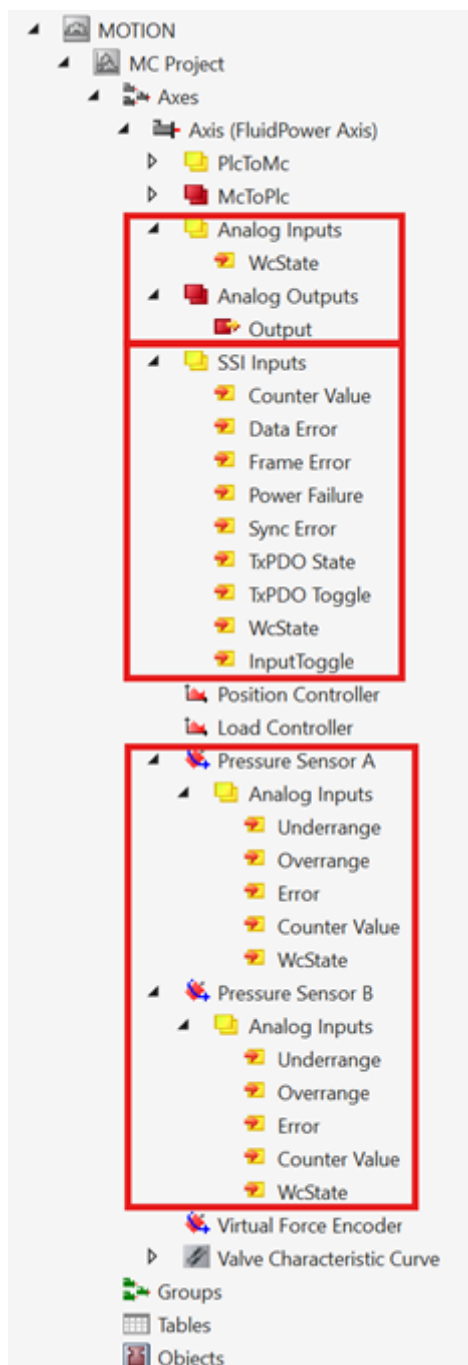
⇒ Das SSI Inputs Modul wurde inkl. Data Area unterhalb der FluidPower-Achse angelegt.

⇒ Im nächsten Schritt werden die Achsobjekte mit der Hardware verknüpft, siehe [Achsobjekte mit Hardware verknüpfen](#) ► 163].

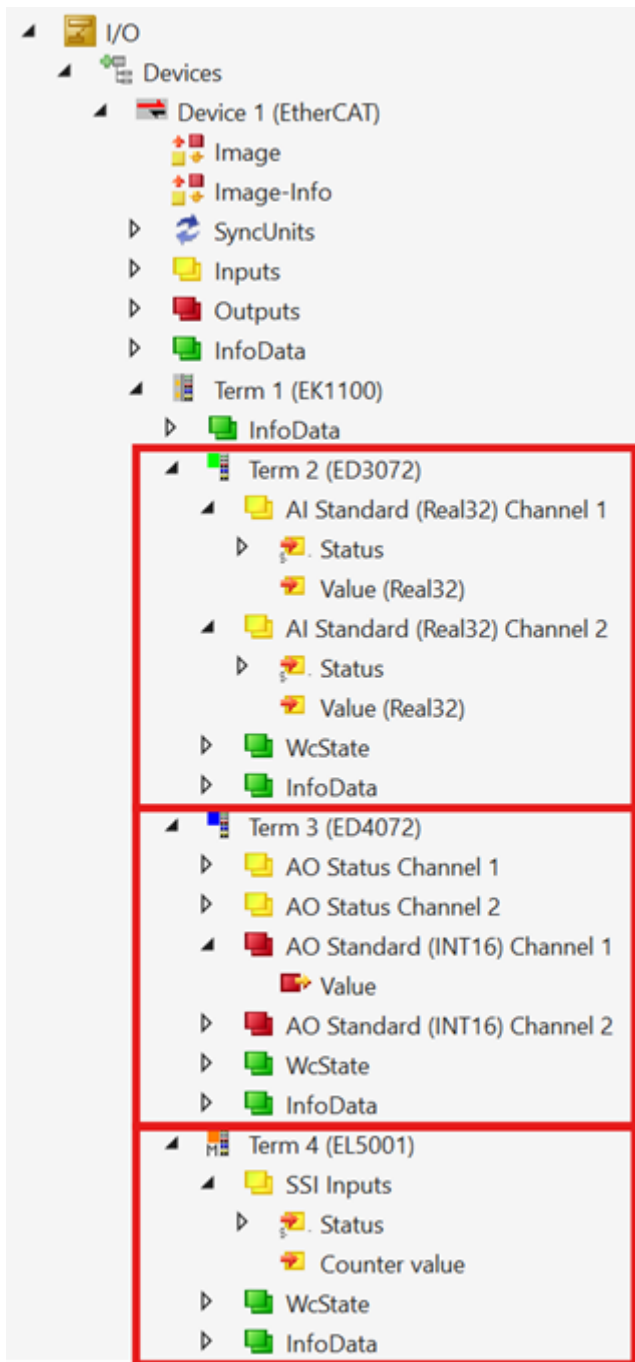
3.2.5.8.2 Achsobjekte mit Hardware verknüpfen

- Eine MC3-Achse wurde erstellt und deren Achs- und Encodertyp konfiguriert, siehe [Achse hinzufügen](#) ► 160].
- Eine Hardwarekonfiguration wurde angelegt, siehe [Anfügen eines IO-Gerätes](#).

- Eine Achskonfiguration wurde angelegt (Drucksensoren A/B, Achstyp Analog Drive, Encodertyp SSI Encoder).

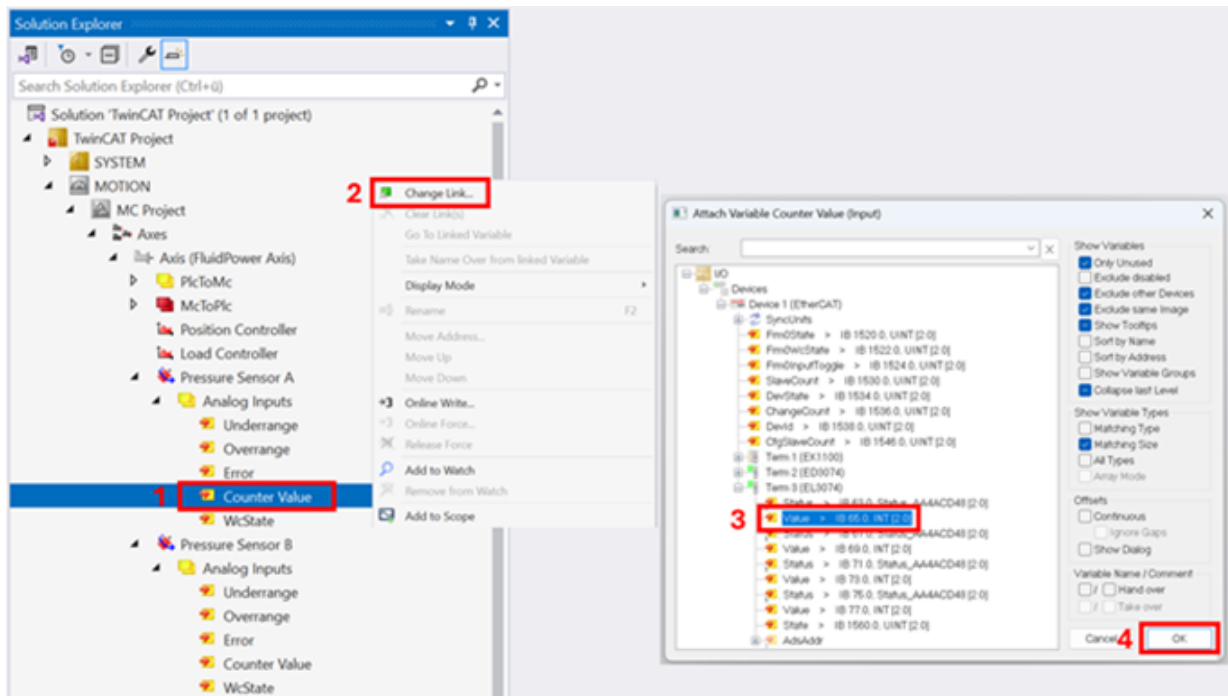


- Eine Hardwarekonfiguration wurde angelegt (Ain-, Aout- und SSI-Klemme).



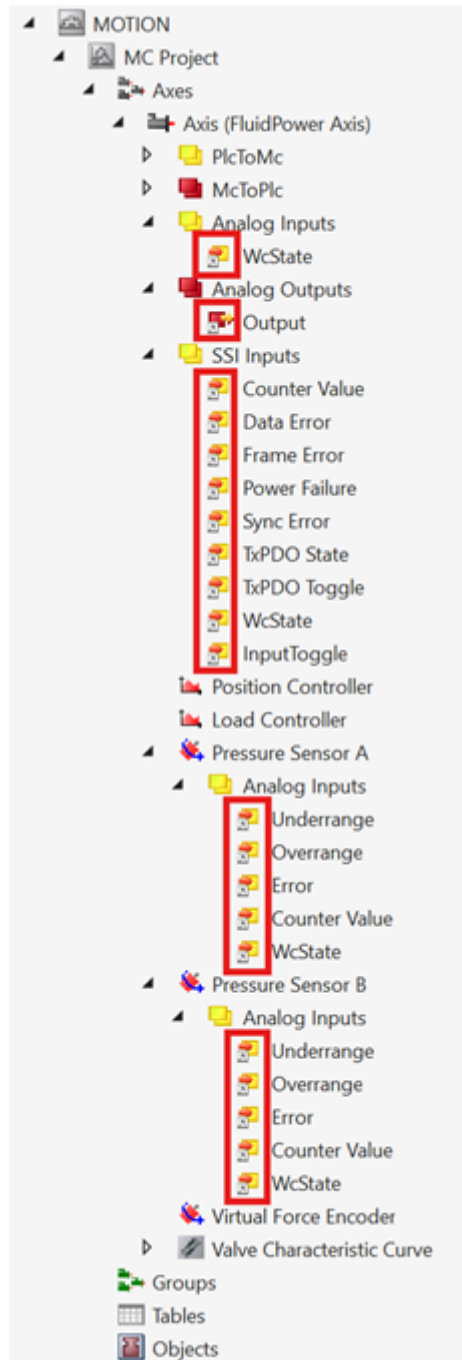
1. Wählen Sie den zu verknüpfenden Link aus der Achskonfiguration,
z. B. Pressure Sensor A → Counter Value zur Verknüpfung des Zähleingangs des Drucksensors A.
2. Machen Sie einen Rechtsklick auf **Change Link...**
3. Selektieren Sie den zu verknüpfenden Link der Hardware,
z. B. EL3074 → Value zur Verknüpfung des Analogeingangs 1.

4. Bestätigen Sie die Auswahl mit **OK**.



5. Wiederholen Sie den Vorgang für alle in der Achs- und Hardwarekonfiguration verfügbaren Links.

⇒ In der Konfiguration des Solution Explorers im Projektbaum sind alle Variablen aus dem zyklischen IO-Interface verknüpft und mit einem weißen Symbol versehen.



6. Aktivieren Sie das TwinCAT-Projekt.

Im nächsten Schritt folgt die Parametrierung des SSI-Encoder, siehe [Wegmesssystem einstellen](#) [► 167].

3.2.5.8.3 Wegmesssystem einstellen

✓ Eine MC3-Achse wurde erstellt und ein SSI-Encoder konfiguriert, siehe [Achse hinzufügen](#) [► 160].

1. Doppelklicken Sie das Objekt **FluidPower Axis**.
2. Öffnen Sie die Registerkarte **Init Parameter**.

3. Parametrieren Sie die Gruppierungen **SSI Encoder** mit Hilfe des Datenblattes des verwendeten Encoders,
z. B. Skalierung der Auflösung 0.1µm.

The screenshot shows the TwinCAT Project environment. On the left, the 'Solution Explorer' displays the project structure, with 'Axis (FluidPower Axis)' selected under the 'MOTION' folder. On the right, the 'TwinCAT Project' window shows the 'Init Parameter' tab, which contains a table of parameters for the SSI Encoder. The table is organized into sections: General, Maximum Dynamics, Default Dynamics, Manual Motion, Monitoring, Error Event Handling, Diagnostics, Additional Encoder, FluidPower Axis Parameters, Analog Drive / Scaling, Analog Drive / Connection, and Analog Drive / Dead Time Compensation. The 'SSI Encoder / General' section is highlighted with a red box, and the 'SSI Encoder / Connection' section is also highlighted with a red box. The 'SSI Encoder / Scaling' section is highlighted with a red box, and the 'SSI Encoder / Dead Time Compensation' section is highlighted with a red box. The 'SSI Encoder / Actual Position Filter / Settings' section is highlighted with a red box.

Name	Value	CS	Unit	Type
SSI Encoder / General				
Position Interpretation	AbsoluteFullRange	☐		MC.EEncoderInterpretation
Position Data Range	FullRange	☐		MC.EEncoderDataRange
SSI Encoder / Scaling				
Position Scaling Numerator From Hardware	0.0001	☐	[PosUnit]/INC	IREAL
Position Scaling Denominator From Hardware	1.0	☐		IREAL
Encoder Mounting Offset	0.0	☐	[PosUnit]	IREAL
Direction Inverted	☐ FALSE	☐		BOOL
SSI Encoder / Connection				
AMS Address	192.168.188.110.2.1:1002	☐		AMSADDR
Channel Number	0	☐		UDINT
EtherCAT Slave Id	03020002	☐	Term 2 (ED30...)	OTCID
SSI Encoder / Dead Time Compensation				
Dead Time Compensation	NonlinearShift	☐		MC.EDeadTimeCompensationMode
Additional Time Shift From Hardware	0.0	☐	ms	IREAL
SSI Encoder / Actual Position Filter / Settings				
Filter Type	None	☐		MC.EFilterType
Filter Time	0.001	☐	s	IREAL

Erläuterungen der Parameter finden Sie hier: [Init Parameter](#) [► 91]

4. Nehmen Sie die weitere Parametrierung des SSI-Encoders in der Klemme unter CoE – Online vor, siehe <https://infosys.beckhoff.com/content/1031/el500x/2247264523.html>

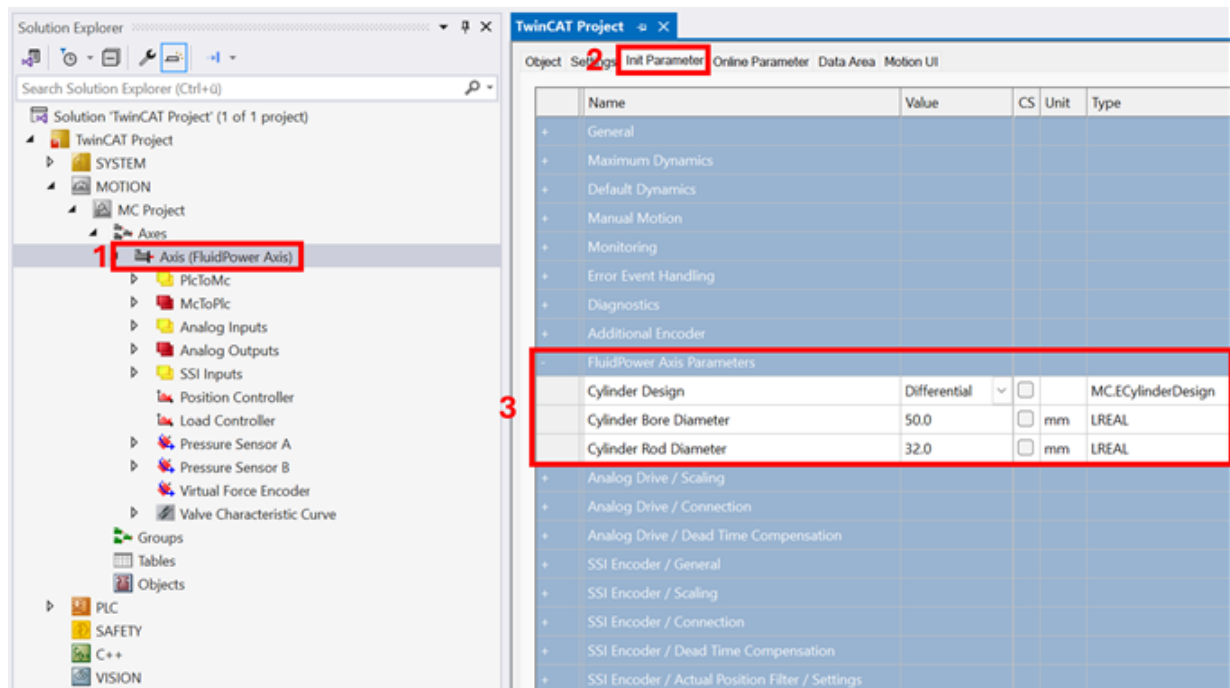
Index	Name	Flags	Value
+ 1C00:0	Sync manager type	RO	> 4 <
+ 1C12:0	RxPDO assign	RW	> 0 <
+ 1C13:0	TxPDO assign	RW	> 1 <
+ 1C33:0	SM input parameter	RO	> 32 <
+ 3101:0	Inputs	RO	> 2 <
+ 4061:0	Feature bits	RO	> 4 <
4066	SSI-coding	RW	Gray code (1)
4067	SSI-baudrate	RW	500 kBaud (3)
4068	SSI-frame type	RW	Multiturn 25 bit (0)
4069	SSI-frame size	RW	0x0019 (25)
406A	Data length	RW	0x0018 (24)
406B	Min. inhibit time[μs]	RW	0x0000 (0)
+ 6010:0	SSI Inputs	RO	> 17 <
- 8010:0	SSI Settings	RW	> 19 <
8010:01	Disable frame error	RW	FALSE
8010:02	Enable power failure bit	RW	FALSE
8010:03	Enable inhibit time	RW	FALSE
8010:04	Enable test mode	RW	FALSE
8010:06	SSI-coding	RW	Gray code (1)
8010:09	SSI-baudrate	RW	500 kBaud (3)
8010:0F	SSI-frame type	RW	Multiturn 25 bit (0)
8010:11	SSI-frame size	RW	0x0019 (25)
8010:12	SSI-data length	RW	0x0018 (24)
8010:13	Min. inhibit time[μs]	RW	0x0000 (0)
+ F000:0	Modular device profile	RO	> 2 <
F008	Code word	RW	0x00000000 (0)
+ F010:0	Module list	RW	> 2 <

Im nächsten Schritt folgt die Parametrierung der Zylinderdaten, siehe [Zylinderdaten einstellen](#) [► 169].

3.2.5.8.4 Zylinderdaten einstellen

- ✓ Eine FluidPower-Achse wurde erstellt, siehe [Achse hinzufügen](#) [► 160].
- 1. Doppelklicken Sie das Objekt **FluidPower Axis**.
- 2. Öffnen Sie die Registerkarte **Init Parameter**.

- Setzen Sie die Zylinderdaten in den Gruppierungen **FluidPower Axis Parameters**, z. B. Differentialzylinder 50/32.

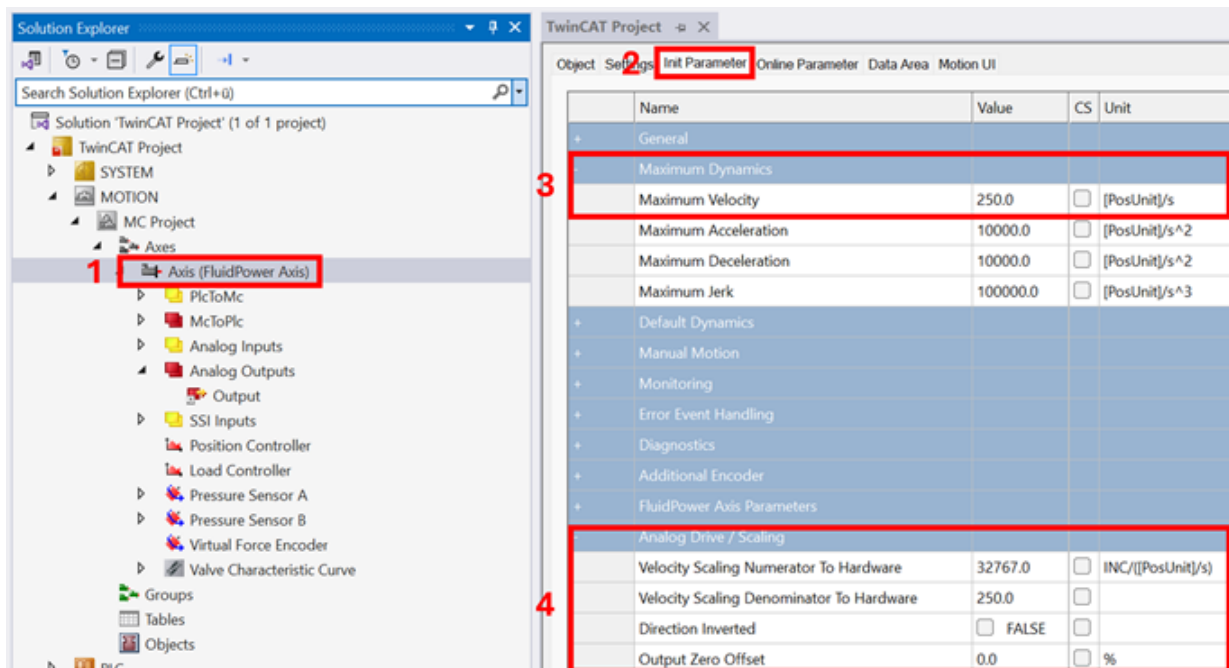


Erläuterungen der Parameter finden Sie hier: [Init Parameter](#) [► 149]

Im nächsten Schritt folgt die Parametrierung des Ausgangssignals, siehe [Ventilausgang einstellen](#) [► 170].

3.2.5.8.5 Ventilausgang einstellen

- ✓ Eine MC3-Achse wurde erstellt und ein Analog Drive als Achstyp konfiguriert, siehe [Achse hinzufügen](#) [► 160].
- Doppelklicken Sie das Objekt **FluidPower Axis**.
 - Öffnen Sie die Registerkarte **Init Parameter**.
 - Setzen Sie die **Maximum Velocity** in den Gruppierungen **Maximum Dynamics**.
Die Maximum Velocity ist die Referenzgeschwindigkeit des Positionsreglers. Sie repräsentiert die maximale Geschwindigkeit, die der Hydraulikzylinder bei 100 % Ventilöffnung erreichen kann. Dieser Wert ist unter Berücksichtigung der Zylinderabmessungen und der Ventilgröße zu berechnen. Ebenso ist dies der Grenzwert der Geschwindigkeit bei der Profilerzeugung.
 - Parametrieren Sie die Gruppierung **Analog Drive / Scaling**,
Velocity Scaling Numerator entspricht in diesem Beispiel den Inkrementen des analogen Ausgangsmoduls und **Velocity Scaling Denominator** der Maximum Velocity.

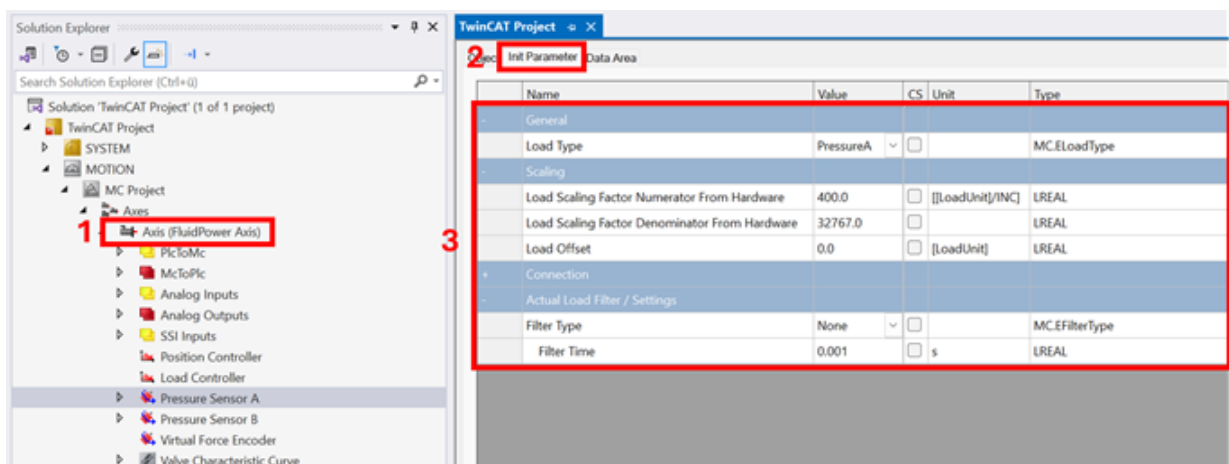


Erläuterungen der Parameter finden Sie hier: [Init Parameter](#) [► 73]

Im nächsten Schritt erfolgt die Parametrierung der Drucksensoren, siehe [Drucksensoren einstellen](#) [► 171].

3.2.5.8.6 Drucksensoren einstellen

- ✓ Eine MC3-FluidPower-Achse wurde erstellt, siehe [Achse hinzufügen](#) [► 160].
- 1. Doppelklicken Sie das Objekt **Pressure Sensor A / B**.
- 2. Öffnen Sie die Registerkarte **Init Parameter**.
- 3. Setzen Sie die Parameter, z. B. Drucksensor pA, 400bar / 32767 Inkremente Analogmodul.



Erläuterungen der Parameter finden Sie hier: [Init Parameter](#) [► 88]

- 4. Aktivieren Sie das TwinCAT-Projekt.
→ Parametrieren Sie **Load Offset** zum Nullpunktabgleich des Sensors bei drucklosem Hydrauliksystem.

Nach erfolgter Parametrierung wird die Achse zur Überprüfung des Drehsinns im Handbetrieb verfahren, siehe [Achse im Handbetrieb mittels Motion UI verfahren](#) [► 172].

3.2.5.8.7 Achse im Handbetrieb mittels Motion UI verfahren

Voraussetzungen

- Eine MC3 FluidPower Achse wurde erstellt und deren Achs- und Encodertyp konfiguriert, siehe [Achse hinzufügen](#) [► 160].
- Die Achsobjekte wurden mit der Hardware verknüpft, siehe [Achsobjekte mit Hardware verknüpfen](#) [► 163].
- Das Wegmesssystem wurde konfiguriert, siehe [Wegmesssystem einstellen](#) [► 167].
- Der Ventilausgang wurde konfiguriert, siehe [Ventilausgang einstellen](#) [► 170].

Limitierung Stellsignal

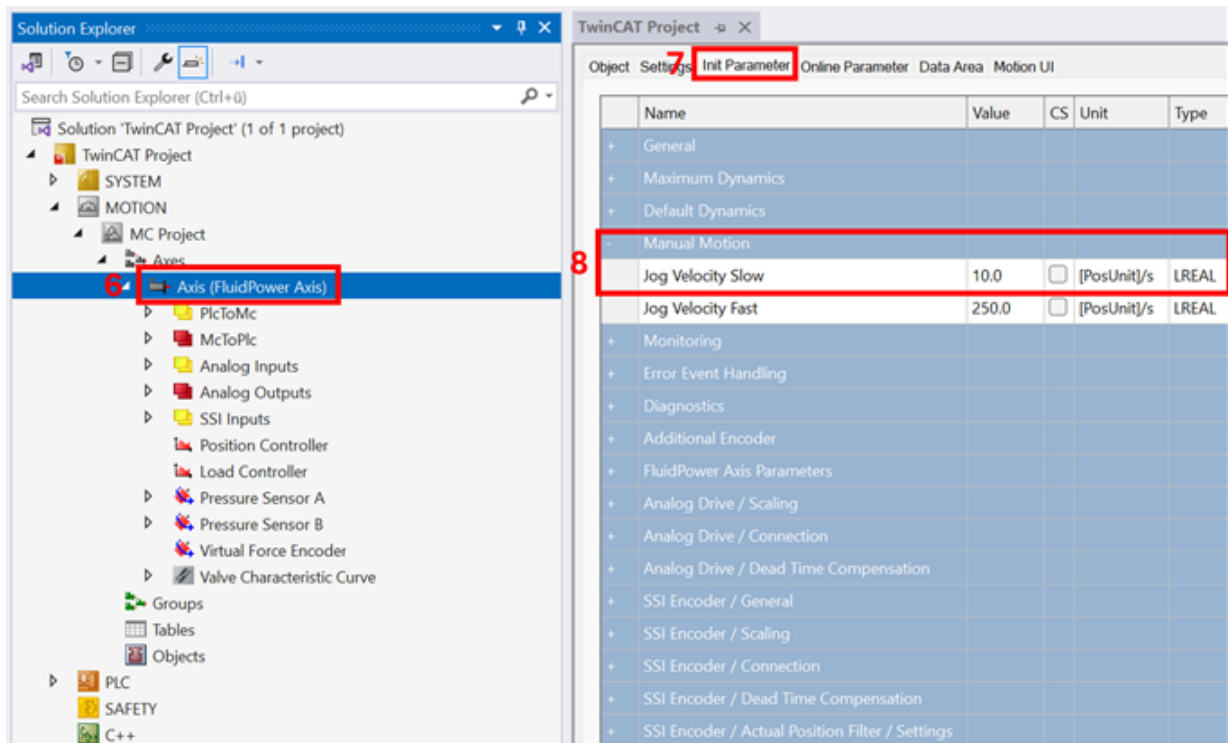
1. Doppelklicken Sie das Objekt **Position Controller**.
2. Öffnen Sie die Registerkarte **Init Parameter**.
3. Setzen des Velocity **Feed Forward Factor** = **1.0** in der Gruppierung **General**
4. Deaktivieren Sie den Positionsregler durch Setzen aller **Regelparameter** = **0.0** in der Gruppierung **Controller Parameter** und **Output Limitation**.
5. Limitieren Sie das Stellsignal via **Minimum / Maximum Output** zur Vermeidung unkontrollierter, schneller Bewegungen.

Name	Value	CS	Unit	Type
General				
Control Value Type	Velocity	☐		MC.EControlValueType
Enabled	☒ TRUE	☐		BOOL
3 Feed Forward Factor	1.0	☐		LREAL
Control Parameter				
Proportional Gain (Kp)	0.0	☐	[ControlValueUnit]/[PosUnit]	LREAL
Derivative Gain (Kd)	0.0	☐	[ControlValueUnit]/([PosUnit]/s)	LREAL
Integral Gain (Ki)	0.0	☐	[ControlValueUnit]/([PosUnit]*s)	LREAL
Integrator Threshold Minimum Velocity	0.0	☐	[PosUnit]/s	LREAL
4 Integrator Threshold Maximum Velocity	0.0	☐	[PosUnit]/s	LREAL
Integrator Lag Window Minimum	0.0	☐	[PosUnit]	LREAL
Integrator Lag Window Maximum	0.0	☐	[PosUnit]	LREAL
Output Limitation				
Integrator Minimum Output	0.0	☐	[ControlValueUnit]	LREAL
Integrator Maximum Output	0.0	☐	[ControlValueUnit]	LREAL
5 Minimum Output	-10	☐	[ControlValueUnit]	LREAL
Maximum Output	10.0	☐	[ControlValueUnit]	LREAL

Handgeschwindigkeit auswählen

6. Doppelklicken Sie das Objekt **Axis**.
7. Öffnen Sie die Registerkarte **Init Parameter**.

8. Reduzieren sie die Handgeschwindigkeit Jog Velocity Slow in der Gruppierung **Manual Motion**.

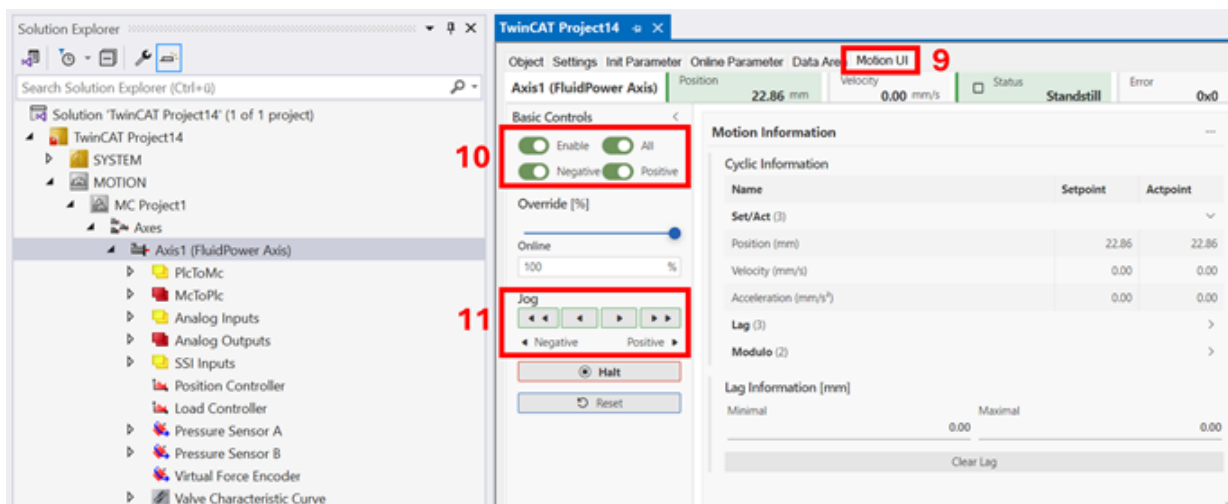


Handbetrieb bewegen

9. Klicken Sie **Motion UI**.

10. Klicken Sie **Enable All** zur Bestromung der Achse.

11. Bewegen Sie die Achse via Klick auf **Jog Slow Positive / Negative**.



→ Überprüfen Sie entsprechend der getriggerten Bewegung die Zählrichtung des Wegmesssystems und die Ansteuerung des Ventils.

→ Bei Fehlverhalten des Wegmesssystems, überprüfen Sie die Verdrahtung, oder setzten Sie die Invertierung (siehe [Wegmesssystem einstellen](#) [► 167]). Wiederholen Sie die Überprüfung.

→ Bei Fehlverhalten des Ventils, überprüfen Sie die Verdrahtung, oder setzten Sie die benötigte Invertierung (siehe [Ventilausgang einstellen](#) [► 170]).

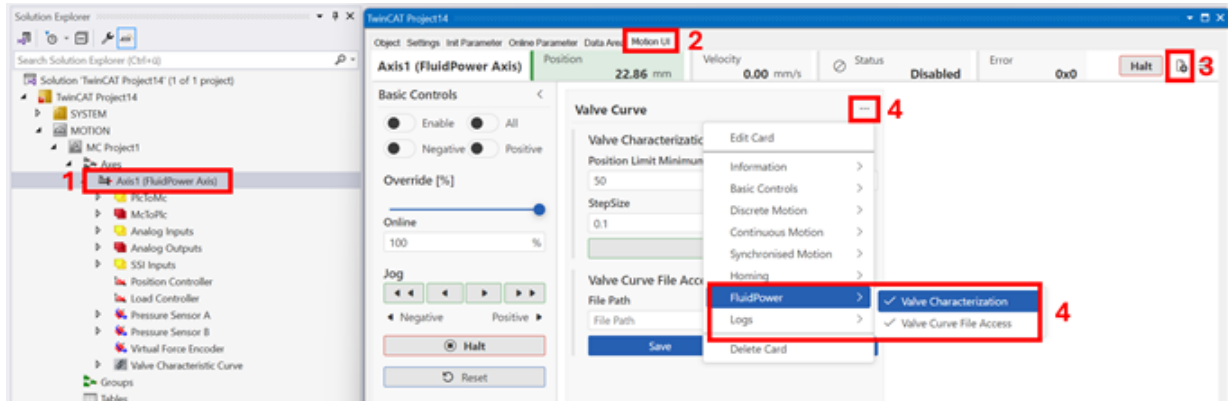
→ Bei starkem Drift der Achse, setzten Sie einen korrigierenden Offset des Ausgangssignals (siehe [Ventilausgang einstellen](#) [► 170]).

Im nächsten Schritt erfolgt die Identifikation der Ventilkennlinie, siehe [Ventilkennlinie vermessen](#) [► 174].

3.2.5.8.8 Ventilkennlinie vermessen

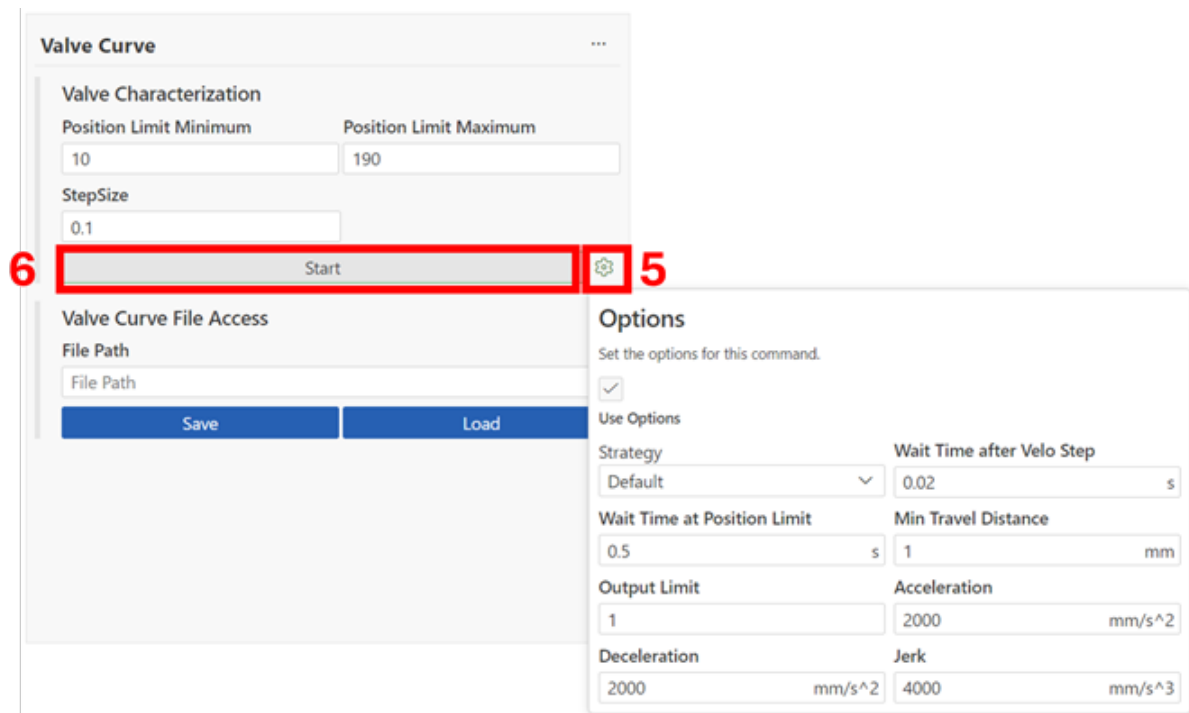
FluidPower-Karte hinzufügen

- ✓ Die MC3-FluidPower-Achse wurde im Handbetrieb verfahren, siehe [Achse im Handbetrieb mittels Motion UI verfahren](#) [► 172].
- 1. Doppelklick auf das **FluidPower Axis** Objekt
- 2. Öffnen der Registerkarte **Motion UI**
- 3. Fügen Sie eine Karte hinzu (siehe [Karten](#) [► 39])
- 4. Fügen Sie der neuen Karte die Optionen **FluidPower** → **Valve Characterization** und **FluidPower** → **Valve Curve File Access** hinzu (siehe [Motion UI](#) [► 158])



Identifikation starten

- 5. Setzen Sie zu Ihrem Hydrauliksystem passende **Limitierungen** (siehe [Identifikation der Ventilkennlinie](#) [► 158])
- 6. Starten Sie die Identifikation der Ventilkennlinie durch Klick auf **Start**



→ Die Schaltfläche „Start“ wechselt zu „Halt“

Valve Curve

Fluid Power

Act Pressure A - Value0.00Act Pressure B - Value0.00

Act Force - ValueInvalid number

Valve Characterization

Position Limit Minimum10Position Limit Maximum190

StepSize0.1

Halt

Valve Curve File Access

File PathFile Path

SaveLoad

→ Nach erfolgreicher Identifikation der Ventilkennlinie wechselt die Schaltfläche zurück auf „Start“.

Ventilkennlinie speichern

7. Klick auf Save zum Speichern der Kennlinie auf dem Zielgerät (siehe [Dateizugriff](#) [► 159])

Valve Curve

Fluid Power

Act Pressure A - Value 0.00 Act Pressure B - Value 0.00

Act Force - Value Invalid number

Valve Characterization

Position Limit Minimum 10 Position Limit Maximum 190

StepSize 0.1

Start

Valve Curve File Access

File Path

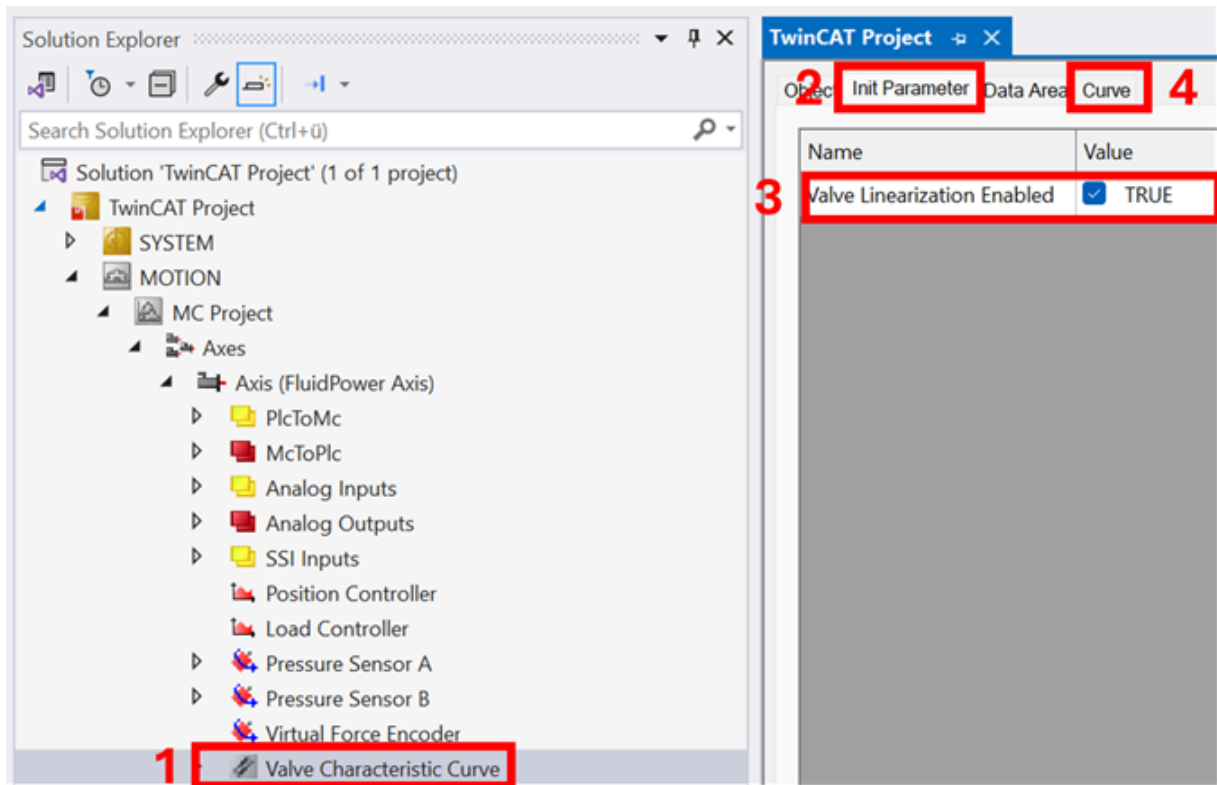
File Path

7 Save Load

Ventilkennlinie aktivieren

8. Doppelklick auf das **Valve Characteristic Curve** Objekt
9. Klick auf **Init Parameter**
10. Aktivieren Sie die Ventilkennlinie

11. Klick auf **Curve** zur Visualisierung und Anpassung der Kennlinie.



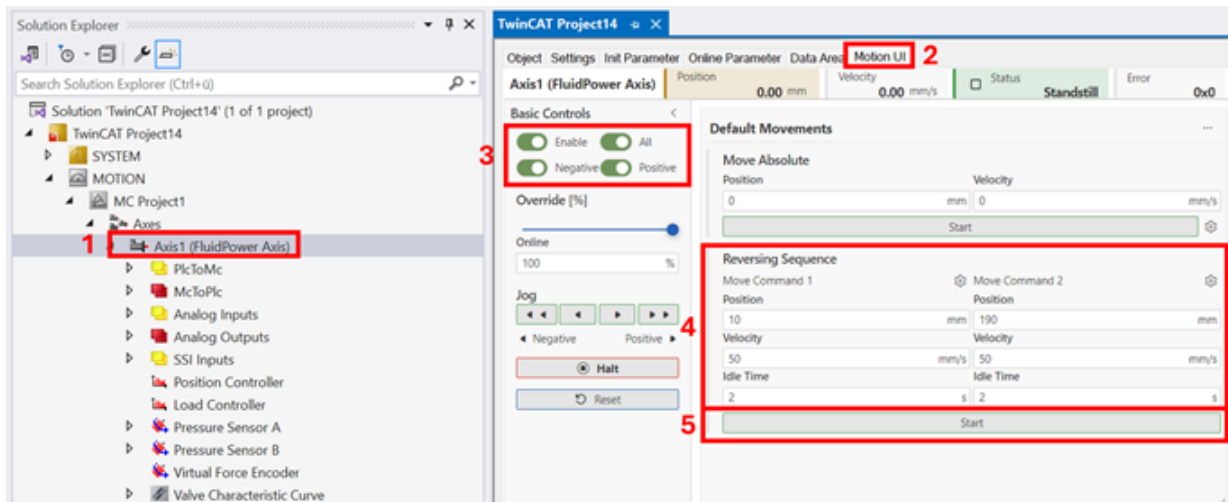
Die Optimierung des Positionsreglers erfolgt im nächsten Schritt, siehe [Regelparameter für Positionsregelung einstellen](#) [► 177].

3.2.5.8.9 Regelparameter für Positionsregelung einstellen

Testbewegung starten

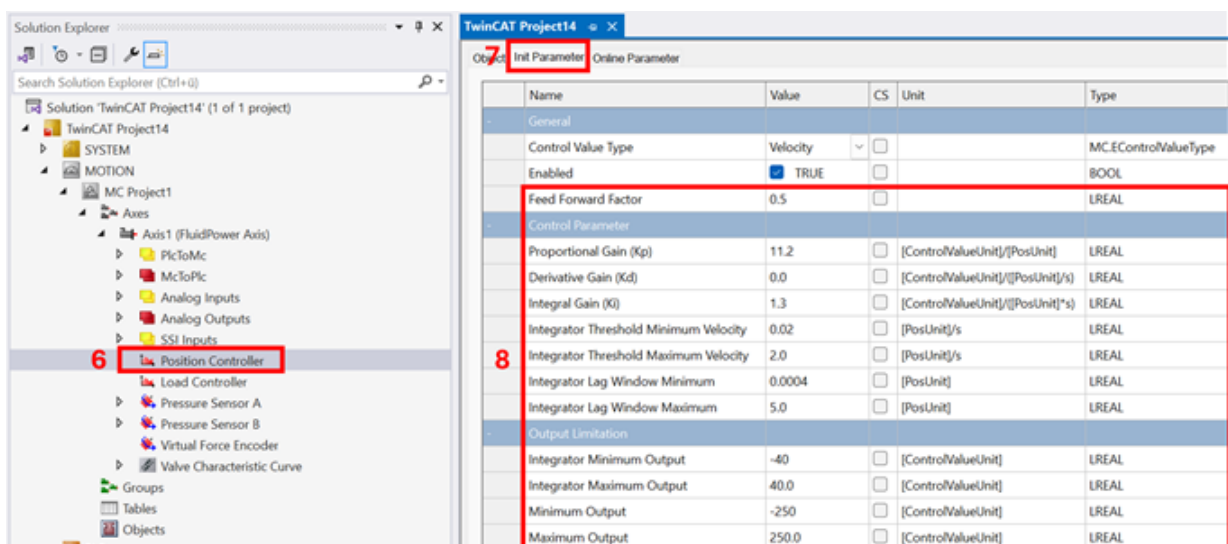
- ✓ Die MC3-FluidPower-Achse wurde im Handbetrieb verfahren, siehe [Achse im Handbetrieb mittels Motion UI verfahren](#) [► 172].
- 1. Doppelklick auf das **Axis** Objekt
- 2. Öffnen der Registerkarte **Motion UI**
- 3. Bestromen der Achse durch Klick auf **Enable All**
- 4. Setzen Sie in der Karte **Default Movements** zu Ihrem Bewegungsprofil passende Parameter des Reversierbetriebs.

5. Start der Bewegung



Regler einstellen

6. Doppelklick auf das **Position Controller** Objekt
7. Öffnen der Registerkarte **Init Parameter**
8. Optimierung der Regelparameter (siehe auch [Init Parameter](#) [► 98])



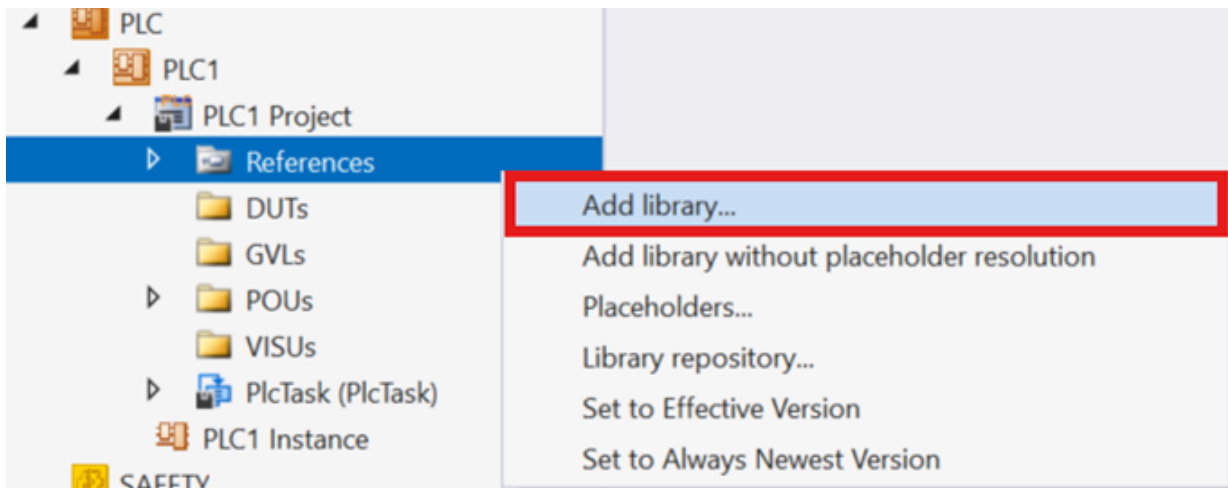
- Ggf. zuvor gesetzte Limitierung des Stellsignals **Minimum / Maximum Output** auf, für das Hydrauliksystem zulässige Werte, erhöhen.
- Setzen **Integral Gain** und **Derivative Gain** = 0.0
- Erhöhen **Proportional Gain** bis das Ausgangssignal beginnt zu oszillieren
- Geringfügige Reduzierung von **Proportional Gain**
- Ggf. Anpassung **Feed Forward Factor** zur Annäherung der Ist- und die Sollgeschwindigkeit
- Setzen **Integral Gain** zur Reduzierung des Restfehlers in der Zielposition
- Begrenzen der Geschwindigkeitsfenster **Integrator Threshold Minimum / Maximum Velocity** des I-Anteils zur Vermeidung einer Aufintegration während der Bewegung und einem daraus resultierenden Überschwingen.
- Begrenzung der Schleppfehlerfenster **Integrator Lag Window Minimum / Maximum** des I-Anteils zur Vermeidung verbleibender Oszillation in der Zielposition.
Empfehlung für **Integrator Lag Window Minimum** ist die 4-fache Auflösung des Wegmesssystems. Das **Integrator Lag Window Maximum** steht in Relation zum verbleibenden Schleppfehler bei reiner P-Regelung.

Im nächsten Schritt folgt die Anbindung der FluidPower-Achse an die PLC, siehe [Achsubjekte mit PLC verknüpfen](#) [► 179]

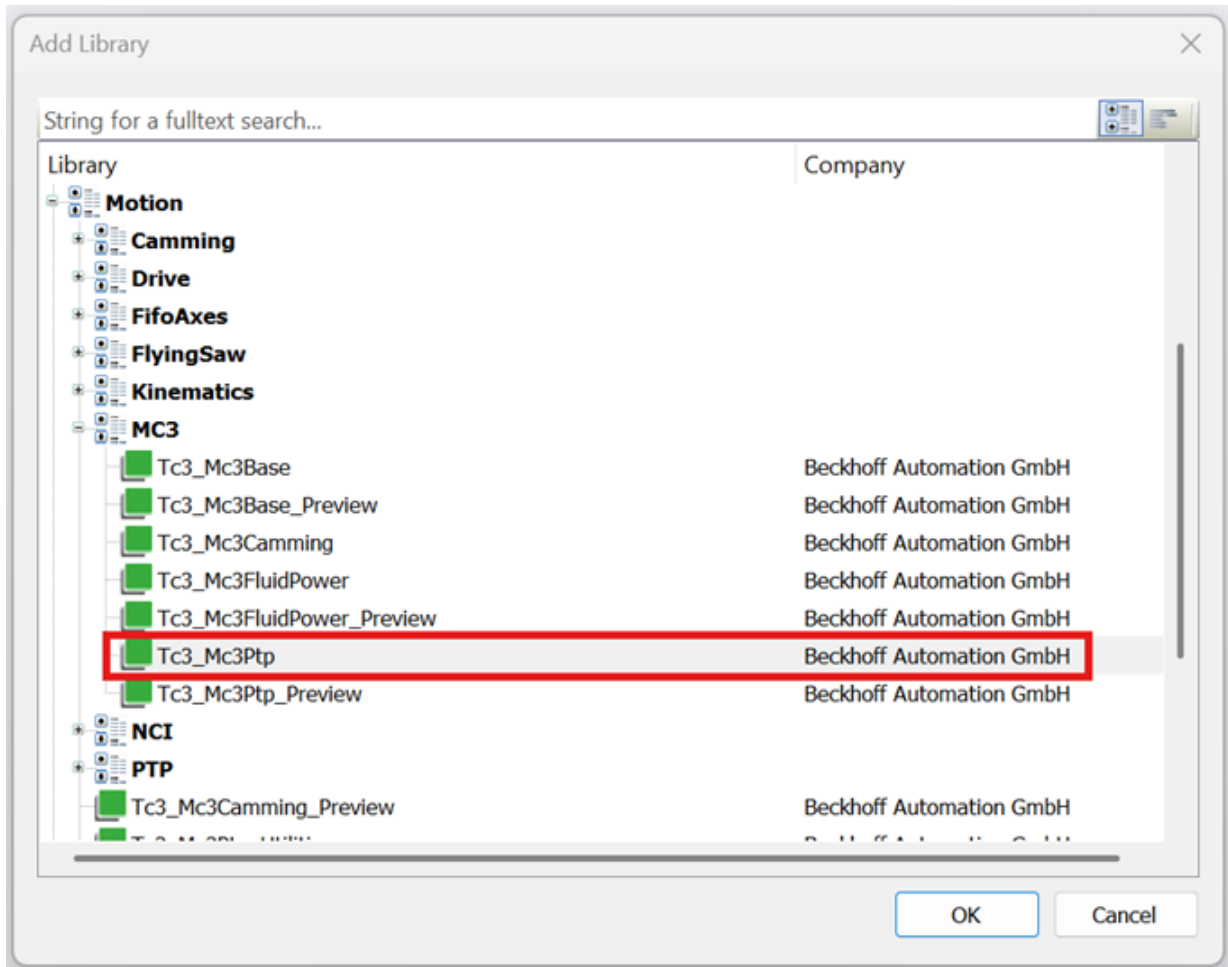
3.2.5.8.10 Achsubjekte mit PLC verknüpfen

PLC-Bibliothek Tc3_Mc3Ptp einfügen

- ✓ Ein FluidPower-Achsubjekt wurde angelegt und der Positionsregler optimiert, siehe [Regelparameter für Positionsregelung einstellen](#) [► 177]
 - ✓ Ein PLC-Projekt wurde angelegt, siehe [Anlegen des Projekts und Auswählen des SPS-Geräts](#)
1. Rechtsklick auf den References-Knoten > **Add library...**

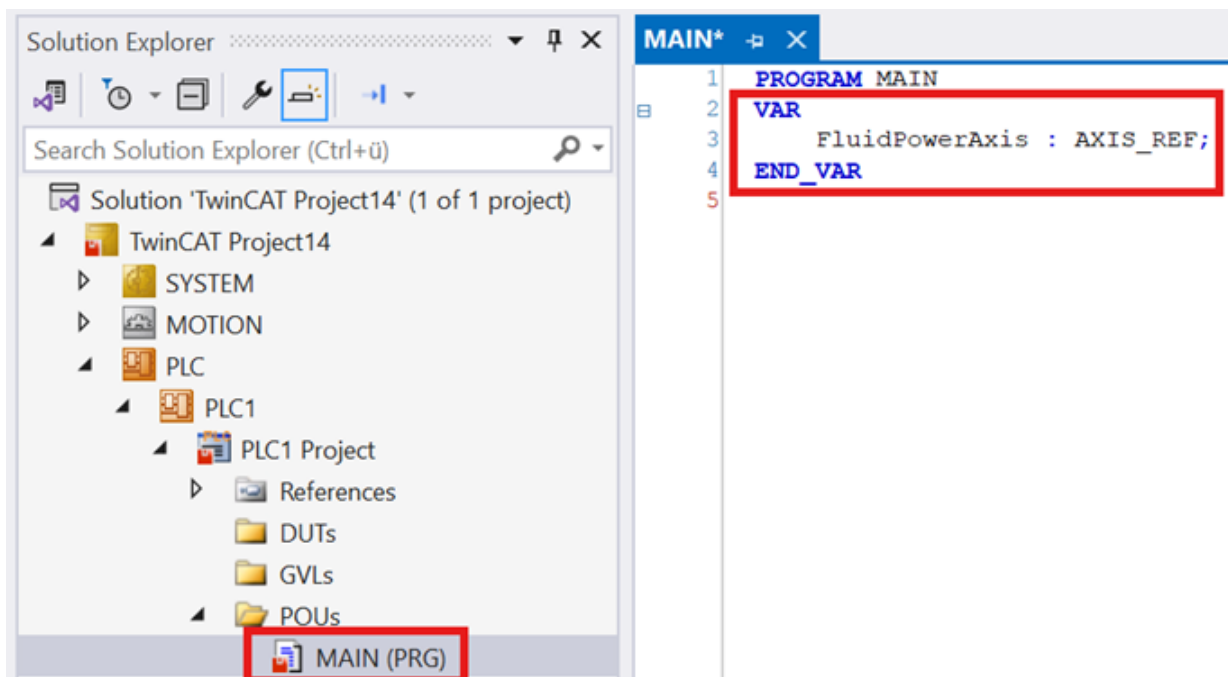


- Im Dialog **Bibliothek hinzufügen** unter **Motion**→**MC3** die Bibliothek **Tc3_Mc3Ptp** auswählen und mit **OK** bestätigen.



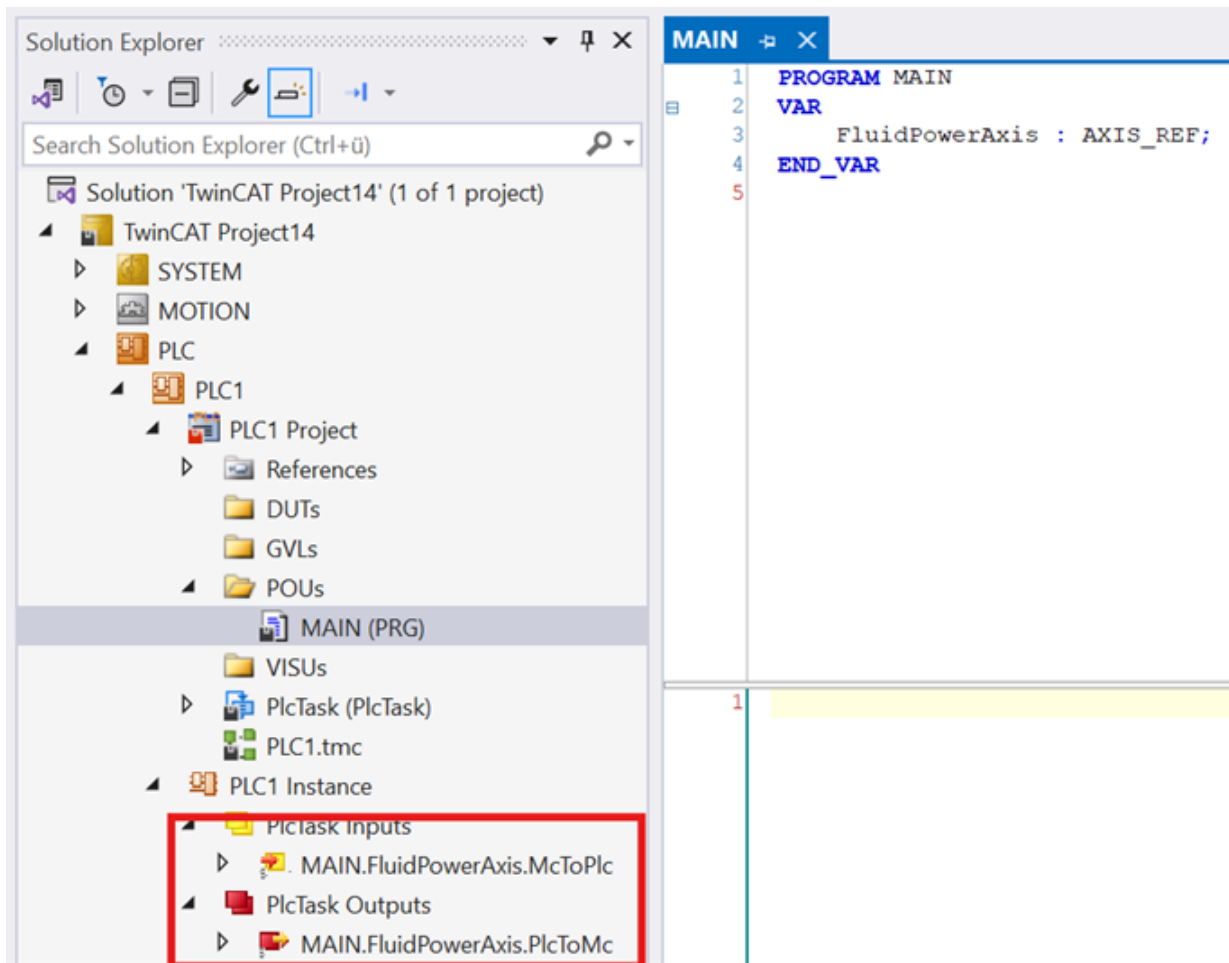
Instanz von **AXIS_REF** anlegen

- Doppelklick auf **MAIN** und anlegen einer Instanz von **Tc3_Mc3Ptp.AXIS_REF**



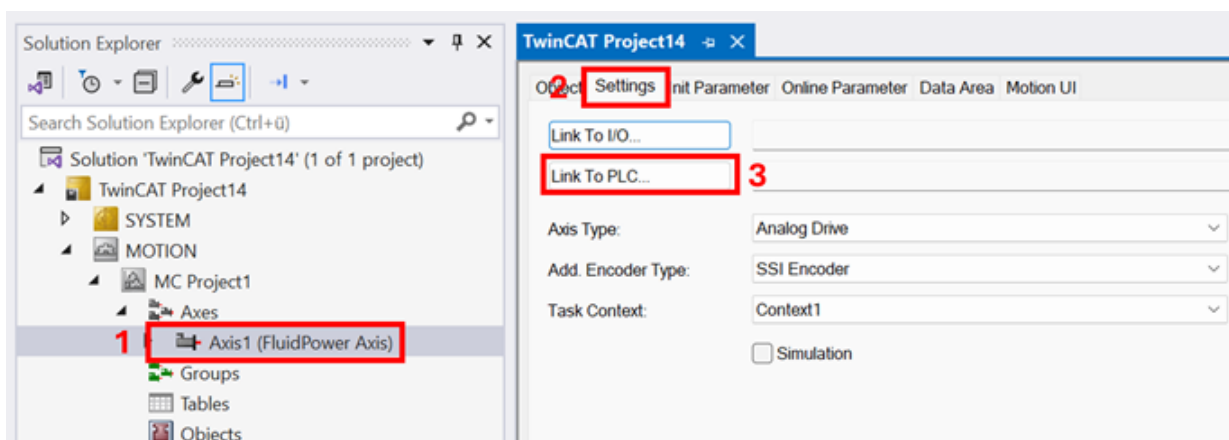
4. Erstellen Sie das PLC-Projekt

→ Die Instanz der `Tc3_Mc3Ptp.AXIS_REF` wird unter den PLC-Instanzen im Solution Explorer angezeigt.

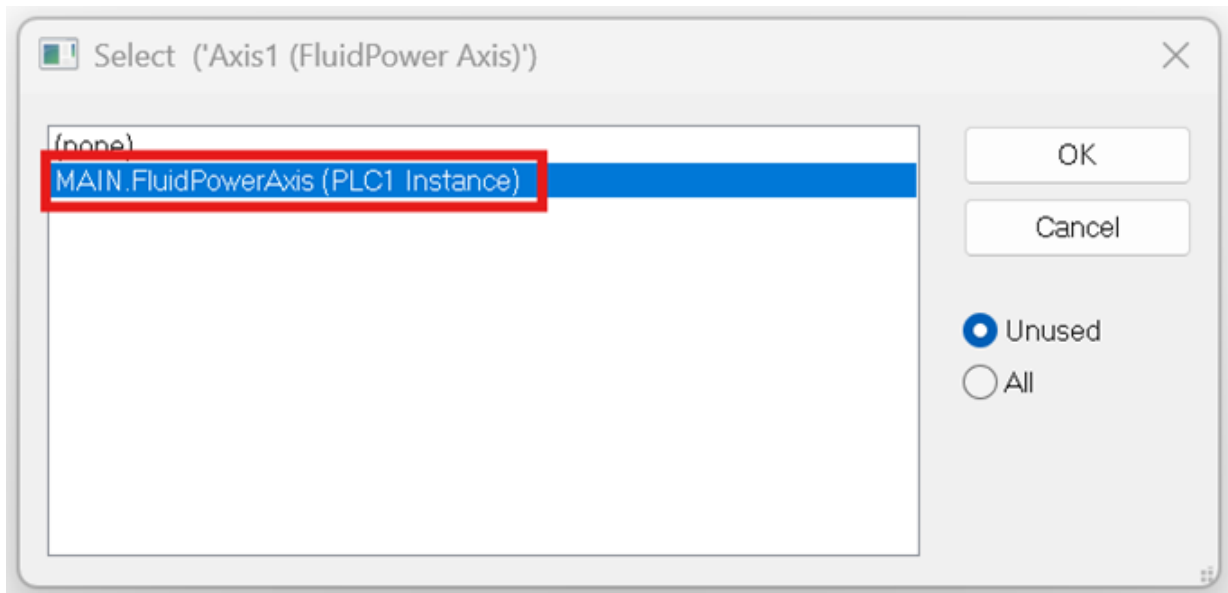


FluidPower Achse mit der PLC verknüpfen

5. Doppelklick auf das **FluidPower Axis** Objekt
6. Öffnen der Registerkarte **Settings**
7. Klick auf **Link To PLC...**



8. Im Dialog **Auswählen** die Instanz der **AXIS_REF** auswählen und mit **OK** bestätigen



TwinCAT-Projekt aktivieren

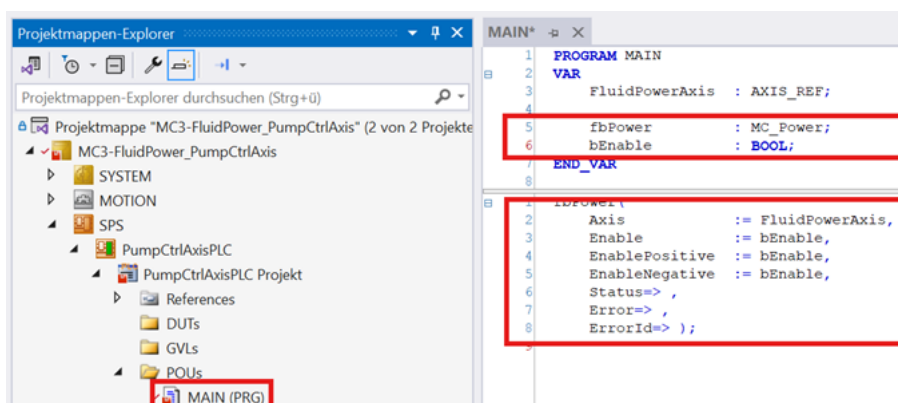
1. Aktivieren Sie das TwinCAT-Projekt

Im nächsten Schritt erfolgt die Bewegung der FluidPower-Achse mittels PLC, siehe [PLC mittels Funktionsbausteinen verfahren](#) [► 182]

3.2.5.8.11 PLC mittels Funktionsbausteinen verfahren

Achse bestromen

- ✓ Ein FluidPower Achsobjekt wurde angelegt und der Positionsregler optimiert, siehe [Regelparameter für Positionsregelung einstellen](#) [► 177]
 - ✓ Das FluidPower Achsobjekt wurde mit der PLC verknüpft, siehe [Achsobjekte mit PLC verknüpfen](#) [► 179]
1. Doppelklick auf **MAIN** und anlegen einer Instanz von **Tc3_Mc3Ptp.MC_Power**, siehe [MC_Power](#)
 2. Implementierung eines zyklischen Aufrufs der angelegten Instanz



```

VAR
    FluidPowerAxis      : AXIS_REF;

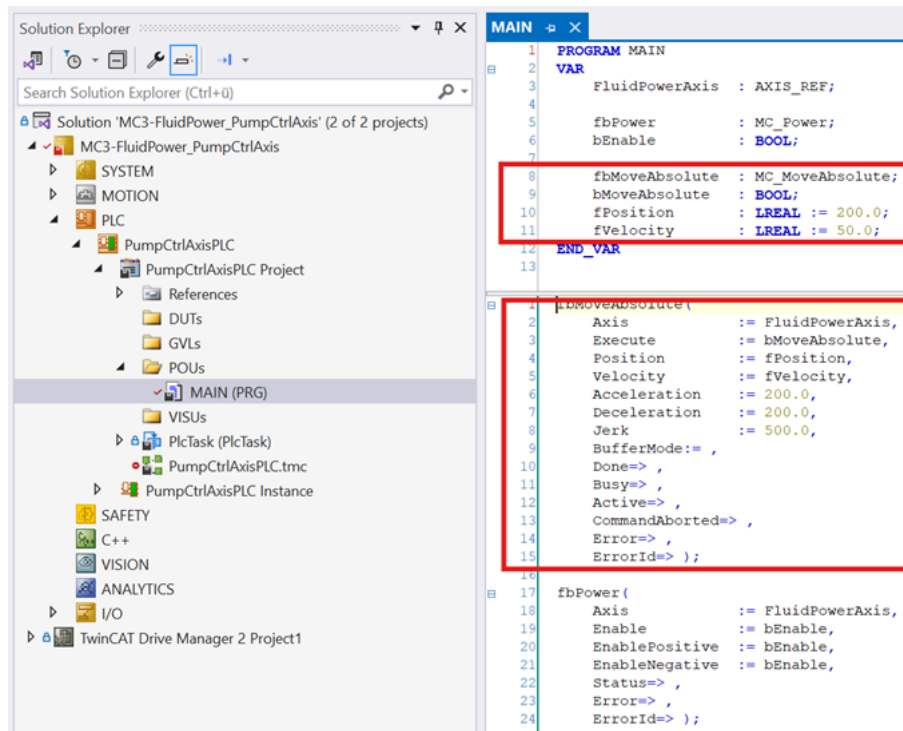
    fbPower              : MC_Power;
    bEnable              : BOOL;
END_VAR

```

```
fbPower(
  Axis          := FluidPowerAxis,
  Enable        := bEnable,
  EnablePositive := bEnable,
  EnableNegative := bEnable,
  Status=> ,
  Error=> ,
  ErrorId=> );
```

Absolute Positionierung

3. Anlegen einer Instanz von **Tc3_Mc3Ptp.MC_MoveAbsolute**, siehe MC_MoveAbsolute
4. Implementierung eines zyklischen Aufrufs der angelegten Instanz



→ Die Achse kann nun mittels PLC bestromt und in die Endlage bewegt werden.

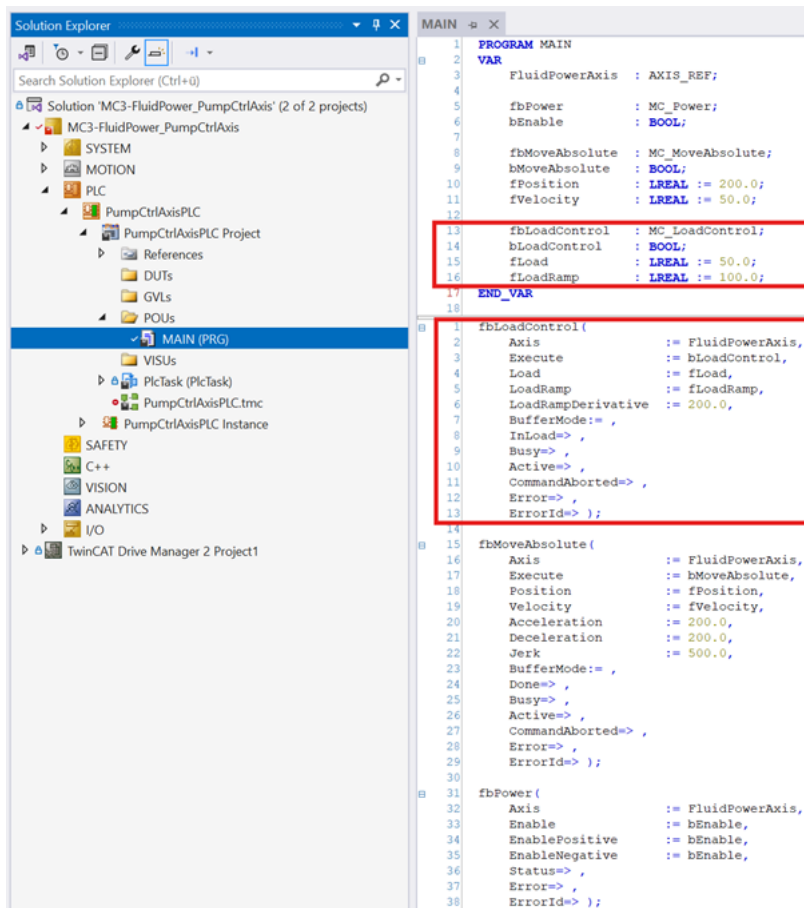
```
fbMoveAbsolute : MC_MoveAbsolute;
bMoveAbsolute  : BOOL;
fPosition      : LREAL := 200.0;
fVelocity      : LREAL := 50.0;
```

```
fbMoveAbsolute(
  Axis          := FluidPowerAxis,
  Execute       := bMoveAbsolute,
  Position      := fPosition,
  Velocity      := fVelocity,
  Acceleration  := 200.0,
  Deceleration  := 200.0,
  Jerk          := 500.0,
  BufferMode:= ,
  Done=> ,
  Busy=> ,
  Active=> ,
  CommandAborted=> ,
  Error=> ,
  ErrorId=> );
```

Druck-/Kraftregelung

5. Anlegen einer Instanz von **Tc3_Mc3Ptp.MC_LoadControl**, siehe MC_LoadControl

6. Implementierung eines zyklischen Aufrufs der angelegten Instanz



→ Die Druck-/Kraftregelung der Achse kann nun mittels PLC angestoßen werden.

```

fbLoadControl : MC_LoadControl;
bLoadControl : BOOL;
fLoad : LREAL := 50.0;
fLoadRamp : LREAL := 100.0;

```

```

fbLoadControl (
  Axis           := FluidPowerAxis,
  Execute        := bLoadControl,
  Load           := fLoad,
  LoadRamp      := fLoadRamp,
  LoadRampDerivative := 200.0,
  BufferMode:= ,
  InLoad=> ,
  Busy=> ,
  Active=> ,
  CommandAborted=> ,
  Error=> ,
  ErrorId=> );

```

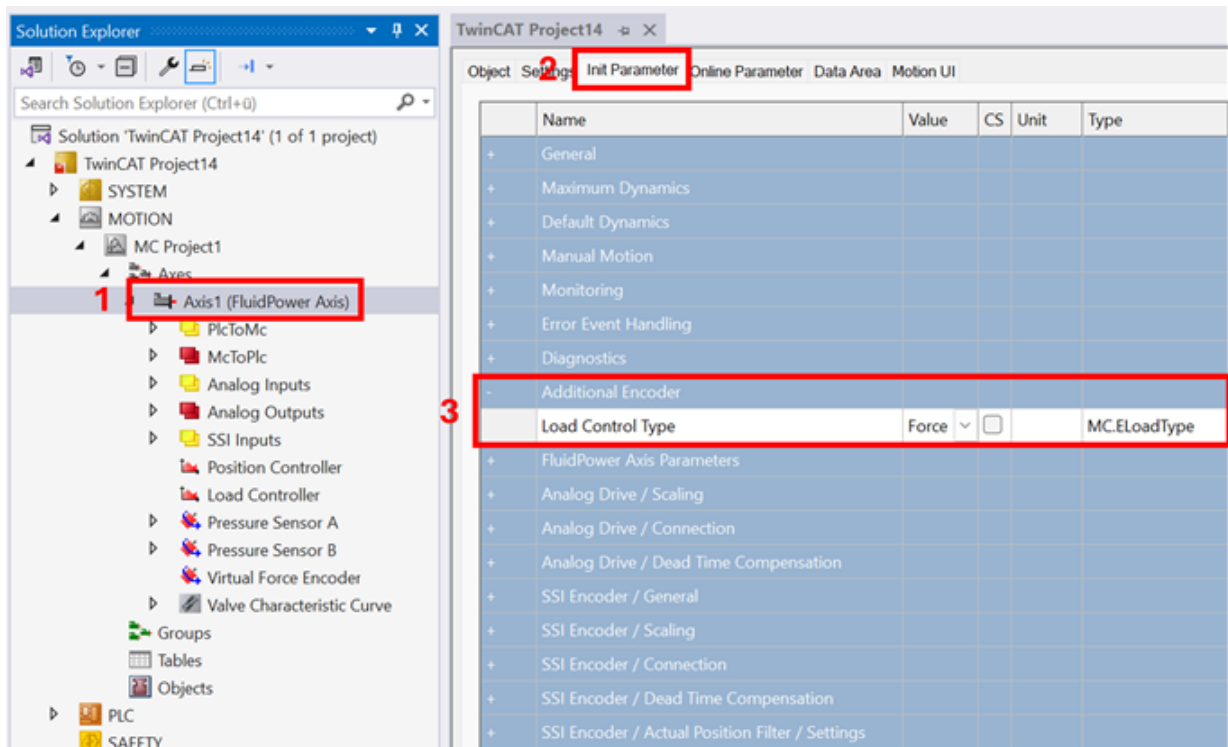
Im nächsten Schritt erfolgt die Optimierung des Druckreglers, siehe [Druckregler einstellen](#) [► 184]

3.2.5.8.12 Druckregler einstellen

Last Typ auswählen

- ✓ Die Druck-/Kraftregelung kann mittels PLC aktiviert werden, siehe [PLC mittels Funktionsbausteinen verfahren](#) [► 182]
- 1. Machen Sie einen Doppelklick auf das Objekt **Axis** (1).
- 2. Öffnen Sie die Registerkarte **Init Parameter** (2).

3. Wählen Sie **Load Control Type** (3) in der Gruppierung **Additional Encoder** aus.



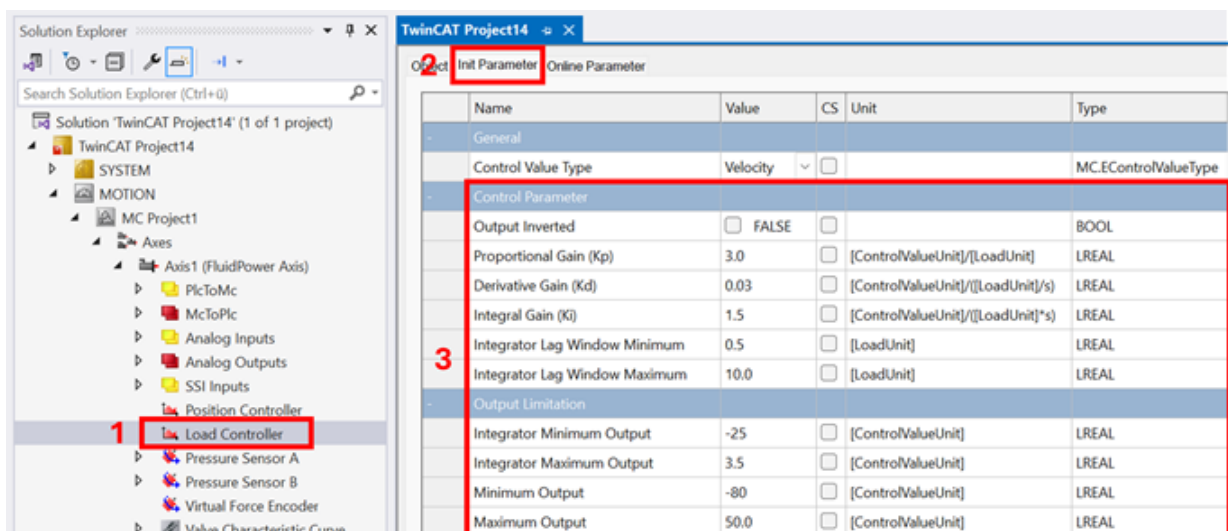
Erläuterungen der Parameter, siehe Values.

Druck-/Kraftregelung mittels PLC starten

1. Bestromen sie die Achse.
2. Positionieren Sie die Achse in die Endlage oder bis zum Kontaktpunkt zur Gegenkraft.
3. Starten Sie die Druck-/Kraftregelung.

Regler einstellen

4. Machen Sie einen Doppelklick auf das Objekt **Load Controller** (1).
5. Öffnen Sie die Registerkarte **Init Parameter** (2).
6. Optimieren Sie die Regelparameter (siehe auch [Init Parameter \[► 96\]](#)).



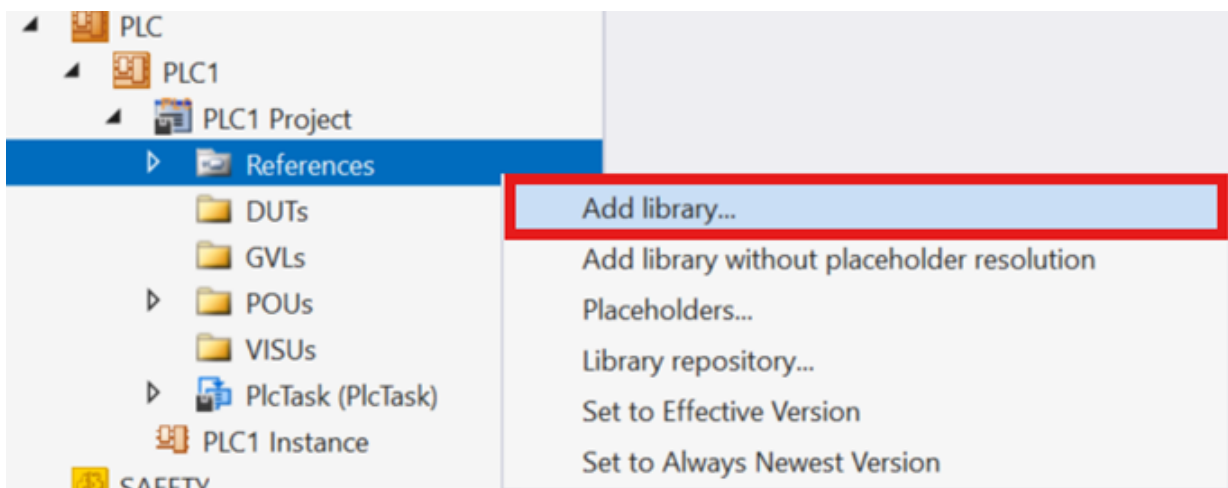
Mögliche Maßnahmen:

- Ggf. das Stellsignal mittels **Output Inverted** invertieren.
- Setzen **Integral Gain** und **Derivative Gain** = 0.0
- Erhöhen **Proportional Gain** bis das Ausgangssignal beginnt zu oszillieren
- Geringfügige Reduzierung von **Proportional Gain**
- Ggf. Anpassung **Derivative Gain** zur Dämpfung des Stellsignals bei Sollwertsprüngen
- Setzen **Integral Gain** zur Reduzierung des Restfehlers in der Ziel-Last
- Begrenzung der Schleppfehlerfenster **Integrator Lag Window Minimum / Maximum** des I-Anteils zur Vermeidung verbleibender Oszillation beim Halten der Ziel-Last.
Empfehlung für **Integrator Lag Window Minimum** ist die 4-fache Auflösung des Druck-/Kraftsensors.
Das **Integrator Lag Window Maximum** steht in Relation zum verbleibenden Schleppfehler bei reiner P-Regelung.

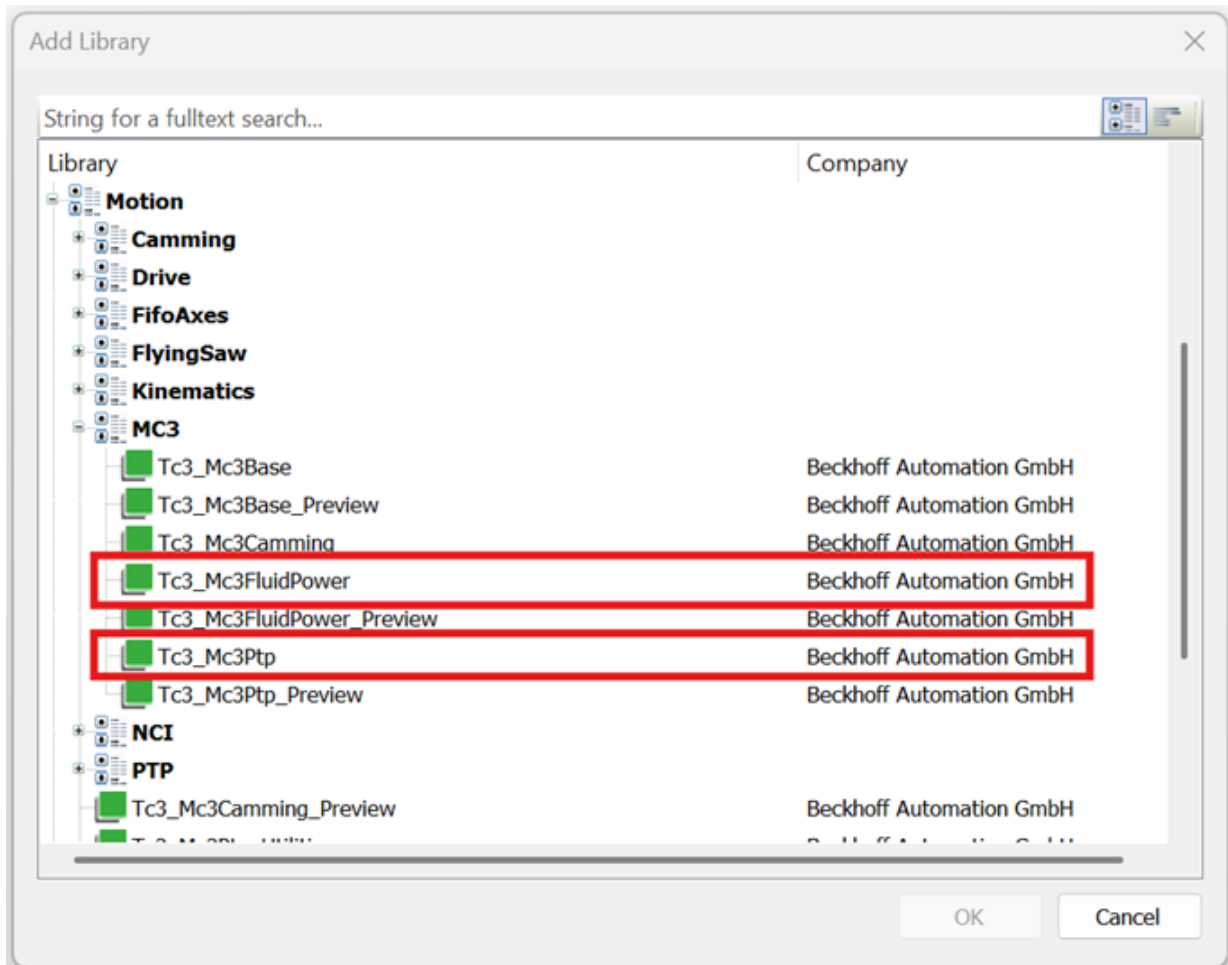
3.2.5.9 Ventilkennlinie via PLC identifizieren**PLC-Bibliothek Tc3_Mc3FluidPower einfügen**

- ✓ Eine MC3-FluidPower-Achse wurde im Handbetrieb verfahren, siehe Achse im Handbetrieb mittels Motion UI verfahren [► 172]

1. Rechtsklick auf den References-Knoten > **Add library...**



2. Im Dialog **Bibliothek hinzufügen** unter **Motion**→**MC3** die Bibliothek **Tc3_Mc3FluidPower** und **Tc3_Mc3Ptp** auswählen und mit **OK** bestätigen.



PLC-Implementierung

1. Anlegen je einer Instanz von
Tc3_Mc3FluidPower.MC_ValveCharacteristicCurve (siehe MC_ValveCharacteristicCurve),
Tc3_Mc3FluidPower.MC_ValveCharacterization (siehe MC_ValveCharacterization),
Tc3_Mc3FluidPower.ValveCharacterizationOptions (siehe ValveCharacterizationOptions)
 und der gewünschten Zahl der Kennliniepunkte **numberOfValveCharacteristicCurvePoints**.
2. Verbinden der Referenz **Tc3_Mc3FluidPower.MC_ValveCharacteristicCurve** mit dem Valve Characteristic Curve Objekt der FluidPower-Achse
3. Bestromen der FluidPower-Achse
4. Setzen der Optionen und Start der Charakterisierung der Ventilkennlinie

5. Nach erfolgreicher Kennlinien-Identifikation, Aktivierung der Kennlinie

```

2  VAR
3      FluidPowerAxis      : AXIS_REF;
4
5      fbPower             : MC_Power;
6
7      fbValveCharacteristicCurve : MC_ValveCharacteristicCurve; // reference to valve characteristic curve object
8      fbValveCharacterization   : MC_ValveCharacterization;      // handling of valve curve characterization
9      stValveCharacterizationOptions : ValveCharacterizationOptions; // options for valve curve characterization
10
11     numberOfValveCharacteristicCurvePoints : UDINT := 41;      // number of curve points, must be an odd number
12     nErrorId                               : UDINT;
13
14     bInit                                   : BOOL;
15     bStart                                 : BOOL;
16     nState                                 : UINT;
17 END_VAR
18
19
20 IF NOT bInit THEN
21     //connect valve characteristic curve object of MC3 axis
22     nErrorId := fbValveCharacteristicCurve.Connect();
23     bInit := fbValveCharacteristicCurve.IsConnected;
24 END_IF
25
26 CASE nState OF
27     0: //enable
28         IF bStart AND fbValveCharacteristicCurve.IsConnected THEN
29             IF NOT fbPower.Status THEN
30                 fbPower.Enable := TRUE;
31                 fbPower.EnableNegative := TRUE;
32                 fbPower.EnablePositive := TRUE;
33             ELSE
34                 bStart := FALSE;
35                 nState := nState + 1;
36             END_IF
37         END_IF
38
39         1: //set valve characterization options
40         stValveCharacterizationOptions.MinimumTravelPerMeasurement := 1.0;
41         stValveCharacterizationOptions.WaitTimeAfterVelocityStep := 0.1;
42         stValveCharacterizationOptions.WaitTimeAtPositionLimit := 0.2;
43         stValveCharacterizationOptions.OutputLimit := 1.0;
44
45         //start characterization
46         fbValveCharacterization.Execute := TRUE;
47         nState := nState + 1;
48
49         2: //enable valve characteristic curve
50         IF NOT fbValveCharacterization.Busy THEN
51             IF fbValveCharacterization.Done THEN
52                 fbValveCharacterization.Execute := FALSE;
53                 fbValveCharacteristicCurve.Enable();
54                 nState := nState + 1;
55             END_IF
56         END_IF
57
58         3: //done
59         nState := 0;
60     END_CASE
61
62 //call FBs
63 fbValveCharacterization(
64     Axis           := FluidPowerAxis,
65     OutputStepsize := 2.0 / TO_LREAL(numberOfValveCharacteristicCurvePoints - 1),
66     PositionLimitMinimum := 50.0,
67     PositionLimitMaximum := 450.0,
68     Options        := stValveCharacterizationOptions);
69
70 fbPower(Axis:= FluidPowerAxis);
71
72
73
74

```

```

VAR
    FluidPowerAxis      : AXIS_REF;

    fbPower             : MC_Power;

    fbValveCharacteristicCurve : MC_ValveCharacteristicCurve; // reference to
valve characteristic curve object
    fbValveCharacterization   : MC_ValveCharacterization;      // handling of valve
curve characterization
    stValveCharacterizationOptions : ValveCharacterizationOptions; // options for valve
curve characterization

    numberOfValveCharacteristicCurvePoints : UDINT := 41;      // number of curve
points, must be an odd number
    nErrorId                               : UDINT;

    bInit                                   : BOOL;
    bStart                                 : BOOL;
    nState                                 : UINT;
END_VAR

IF NOT bInit THEN
    //connect valve characteristic curve object of MC3 axis
    nErrorId := fbValveCharacteristicCurve.Connect();
    bInit := fbValveCharacteristicCurve.IsConnected;
END_IF

```

```

CASE nState OF
  0: //enable
    IF bStart AND fbValveCharacteristicCurve.IsConnected THEN
      IF NOT fbPower.Status THEN
        fbPower.Enable := TRUE;
        fbPower.EnableNegative := TRUE;
        fbPower.EnablePositive := TRUE;
      ELSE
        bStart := FALSE;
        nState := nState + 1;
      END_IF
    END_IF

  1: //set valve characterization options
    stValveCharacterizationOptions.MinimumTravelPerMeasurement := 1.0;
    stValveCharacterizationOptions.WaitTimeAfterVelocityStep := 0.1;
    stValveCharacterizationOptions.WaitTimeAtPositionLimit := 0.2;
    stValveCharacterizationOptions.OutputLimit := 1.0;

    //start characterization
    fbValveCharacterization.Execute := TRUE;
    nState := nState + 1;

  2: //enable valve characteristic curve
    IF NOT fbValveCharacterization.Busy THEN
      IF fbValveCharacterization.Done THEN
        fbValveCharacterization.Execute := FALSE;
        fbValveCharacteristicCurve.Enable();
        nState := nState + 1;
      END_IF
    END_IF

  3: //done
    nState := 0;
END_CASE

//call FBs
fbValveCharacterization(
  Axis                := FluidPowerAxis,
  OutputStepsize      := 2.0 / TO_LREAL(numberOfValveCharacteristicCurvePoints - 1),
  PositionLimitMinimum := 50.0,
  PositionLimitMaximum := 450.0,
  Options              := stValveCharacterizationOptions);

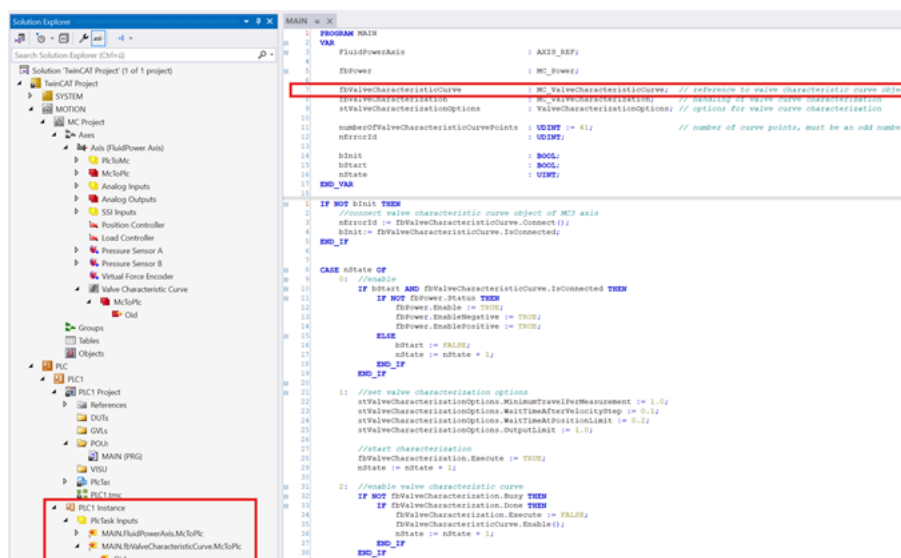
fbPower(Axis:= FluidPowerAxis);

```

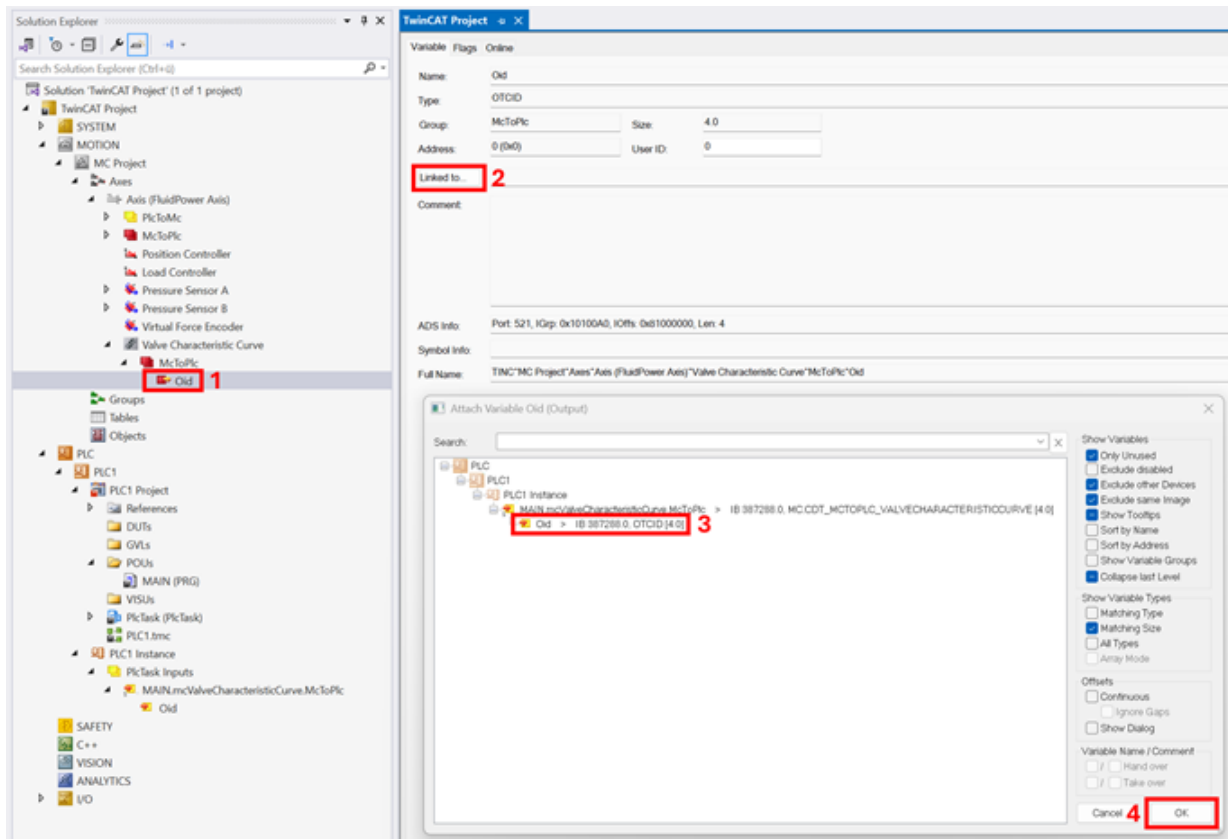
Verknüpfung MC3-Axis/PLC

3. Erstellen Sie das PLC-Projekt

→ Die Instanz des Tc3_Mc3FluidPower.MC_ValveCharacteristicCurve wird unter den PLC-Instanzen im Solution Explorer angezeigt.



4. Verknüpfen Sie die PLC-Instanz mit der Valve Characteristic Curve-Instanz der MC3.



5. Aktivieren Sie das TwinCAT-Projekt.

3.2.6 Planar Mover

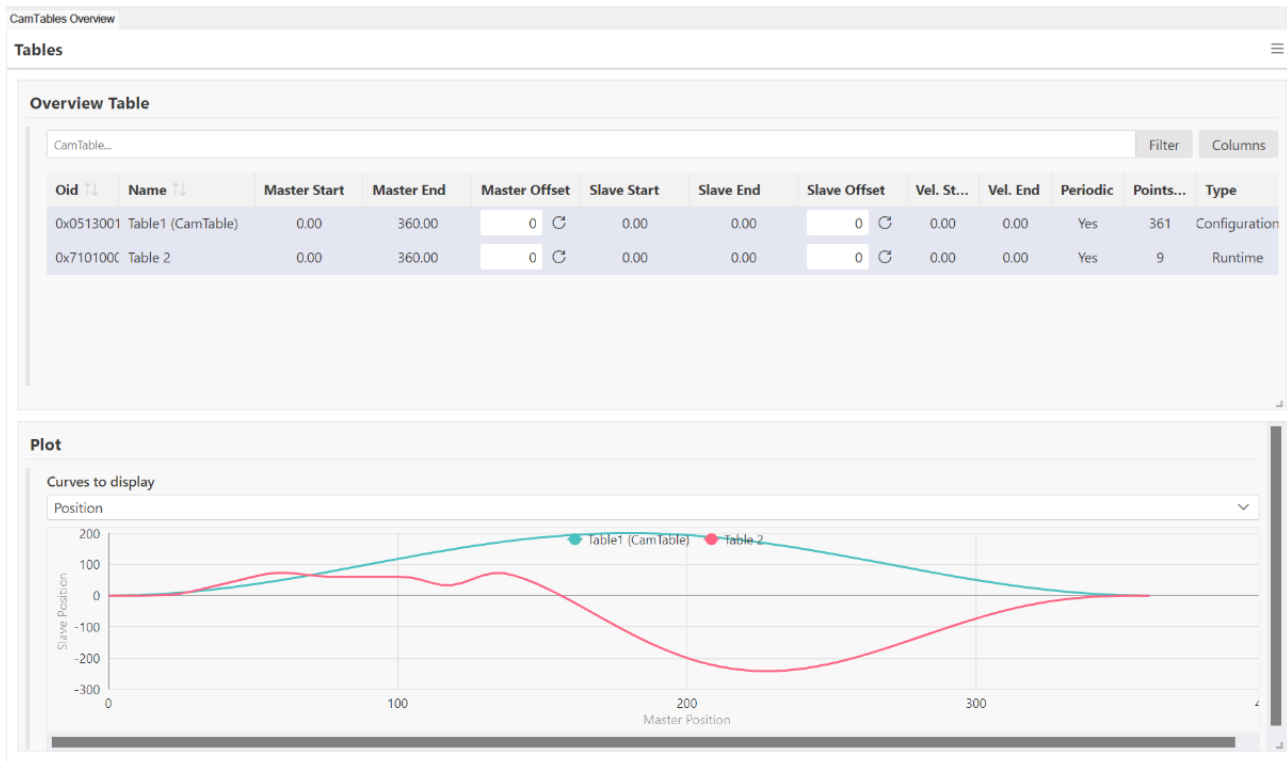
Spezielles Achsobjekt zur Verwendung mit einem Beckhoff XPlanar Mover. Weitere Details sind der [TF5430 | TwinCAT Planar Motion Dokumentation](#) zu entnehmen.

3.2.7 Groups

Aktuell nur in Verbindung mit dem XPlanar zu verwenden, siehe [TF5430 | TwinCAT Planar Motion Dokumentation](#).

3.2.8 Tables

Unter Tables ist ein Überblick über alle vorhandenen Kurvenscheiben zu finden. Wird die jeweilige Tabelle im oberen Bereich der Benutzeroberfläche durch Anklicken selektiert, wird sie in der unteren Hälfte grafisch dargestellt.



Filter

Über den Filter lassen sich die angezeigten Tabellen nach den folgenden beiden Optionen filtern:

- Configuration CamTables,
- Runtime CamTables.

„Configuration CamTables“ beschreibt die Kurvenscheiben, die als Cam-Objekt im Systembaum angelegt sind. „Runtime CamTables“ sind die Tabellen, die nicht als Objekt im Systembaum angelegt sind und zur Laufzeit aus der PLC heraus erstellt werden.

Menü

In der oberen rechten Ecke lässt sich das Menü öffnen.

Im Menü sind verschiedene Aktionen und Einstellmöglichkeiten verfügbar:

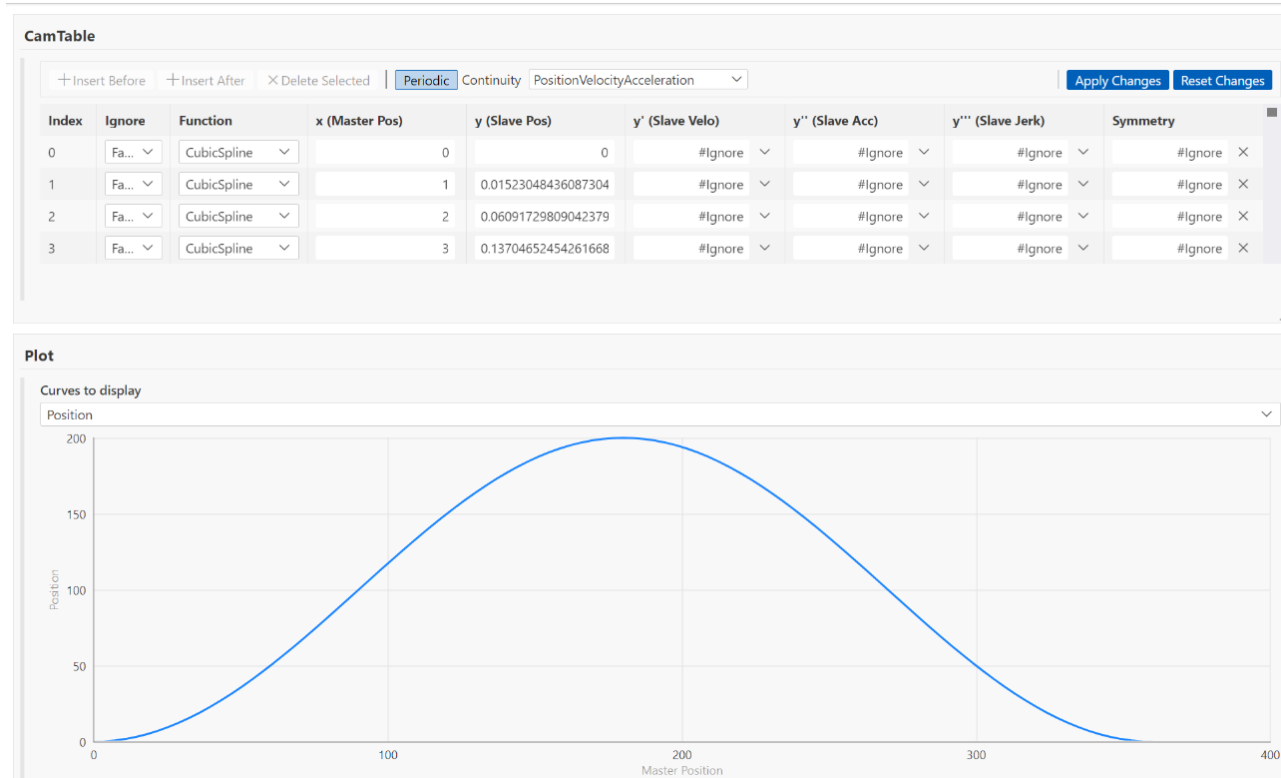
Auswahl	Beschreibung
Reconnect	Hierüber wird die Verbindung zwischen der UI und dem anzuzeigenden Objekt neu hergestellt und die Daten neu geladen.
Settings	Hierüber werden die Einstellungen geöffnet.
Save Layout	Speichert das aktuelle Layout der UI. Es muss ein benutzerdefinierter Name vergeben werden. Das gespeicherte Layout ist nur im aktuellen Projekt nutzbar.
Apply Layout	Wendet ein vorhandenes Layout auf die aktuelle Ansicht an. Es ist standardmäßig das Layout „Default“ hinterlegt. Weitere Layouts, die über „Save Layout“ gespeichert wurden, stehen hier ebenso zur Verfügung.
Layout Manager	Öffnet den Layout Manager.
Theme	Es kann zwischen den folgenden Themen für die Farbgebung gewählt werden: • System,

Auswahl	Beschreibung
	• Light, • Dark.

3.2.8.1 Camming UI

Die Registerkarte **Camming UI** dient der Anzeige sowie dem Editieren der ausgewählten Kurvenscheibe.

Table1 (CamTable)



CamTable

In der Karte „CamTable“ sind die einzelnen Stützpunkte der aktuellen Kurvenscheibe hinterlegt. Es können Änderungen an der Tabelle vorgenommen werden, die direkt in der Karte „Plot“ dargestellt werden. Mit einem Klick auf **Apply Changes** werden die Änderungen gespeichert. **Reset Changes** hingegen setzt die noch nicht gespeicherten Eingaben zurück auf ihren Originalwert. Wird die Checkbox „Periodic“ ausgewählt, wird die Kurvenscheibe als periodische Tabelle übernommen.

Plot

In der Karte „Plot“ wird die aktuelle Kurvenscheibe grafisch dargestellt. Mittels Checkboxes kann zusätzlich zur Position auch die Geschwindigkeits- und Beschleunigungskurve eingeblendet werden. Es besteht außerdem die Möglichkeit über die Checkbox „Show Original“ für Position, Geschwindigkeit und Beschleunigung zusätzlich die Originalkurven einzublenden. Die Originalkurven sind die Kurven, die aktuell aktiv sind. Im Gegensatz dazu sind die dargestellten Verläufe die angepassten Verläufe, die noch nicht gespeichert sind.

Menü

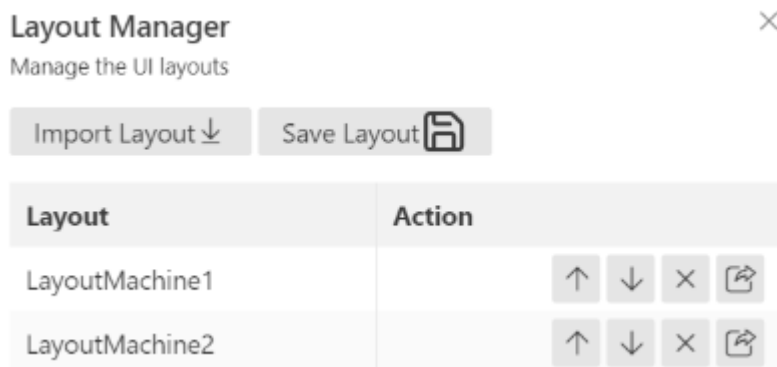
In der oberen rechten Ecke lässt sich das Menü öffnen.

Im Menü sind verschiedene Aktionen und Einstellmöglichkeiten verfügbar:

Auswahl	Beschreibung
Reconnect	Hierüber wird die Verbindung zwischen der UI und dem anzuzeigenden Objekt neu hergestellt und die Daten neu geladen.
Settings	Hierüber werden die Einstellungen geöffnet.
Import	Öffnet den Datei-Manager, um Kurvenscheiben zu importieren.
Export	Öffnet Optionen, um die Kurvenscheibe zu exportieren.
Save Layout	Speichert das aktuelle Layout der UI. Es muss ein benutzerdefinierter Name vergeben werden. Das gespeicherte Layout ist nur im aktuellen Projekt nutzbar.
Apply Layout	Wendet ein vorhandenes Layout auf die aktuelle Ansicht an. Es ist standardmäßig das Layout „Default“ hinterlegt. Weitere Layouts, die über „Save Layout“ gespeichert wurden, stehen hier ebenso zur Verfügung.
Layout Manager	Öffnet den Layout Manager.
Reset Layout	Setzt die UI zurück, sodass wieder das Standard-Layout genutzt wird.
Theme	Es kann zwischen den folgenden Themen für die Farbgebung gewählt werden: • System, • Light, • Dark.

Layout Manager

Der Layout Manager zeigt die eigenen gespeicherten Layouts an und bietet weitere Optionen.



Auswahl	Beschreibung
Import Layout ↓	Importiert ein bestehendes Layout in das Projekt.
Save Layout	Speichert das aktuelle Layout der UI projektintern. Es öffnet sich ein weiteres Fenster, um einen Namen zu vergeben. Wird ein bestehender Layout-Name verwendet, wird das bestehende Layout überschrieben.
	Verändert die Reihenfolge der eigenen Layouts. Mittels der Pfeiltasten kann ein Layout in der Liste nach oben bzw. unten verschoben werden.
	Löscht das jeweilige selbst erstellte Layout.
	Exportiert das selbst erstellte Layout, sodass es auch in anderen Projekten verwendet werden kann. Im Anschluss muss der Speicherort vergeben werden.

Settings

Unter „Settings“ lassen sich verschiedene Einstellungen für die Camming UI treffen. Über den Button **Save** werden die getätigten Einstellungen projektintern gespeichert.

Common

Auswahl	Beschreibung
Number Of Digits	Anzahl an anzuzeigenden Nachkommastellen.
XAR Interval [ms]	Legt das Intervall fest, mit dem Daten aus der Laufzeitumgebung ausgelesen werden.
XAE Interval [ms]	Legt das Intervall fest, mit dem Daten aus der Engineering-Umgebung ausgelesen werden.
Feedback Duration [s]	Legt die Zeitdauer fest, für die das Feedback-Fenster (Pop-Up) eingeblendet wird.
Severity of Feedback	Legt fest, ab welchem Level Ereignisse im Feedback-Fenster angezeigt werden.
Error Code Format	Optionen: Hex, Decimal Legt das Format fest, in dem der Fehlercode angezeigt wird (als Hexadezimal- oder Dezimalzahl).
Show Full Name	Vergrößert bei Bedarf die Anzeige des Achsnamen, sodass dieser vollständig angezeigt wird.

Polling State

Der Polling State gibt zu Diagnosezwecken Daten von der letzten Anfrage der UI an den Motion Treiber zurück.

Auswahl	Beschreibung
Polling State	Idle / Polling / No Connection / Error
Error Message	Fehlermeldung, falls es in der Kommunikation zwischen Motion UI und dem Treiber zu einem Fehler kommt.
Last Fetch Duration	Gesamtdauer der letzten Anfrage der UI an den Motion Treiber.
Last Bundled Response	Letzte Antwort aus dem Treiber an die UI.

Storage

Auswahl	Beschreibung
Layouts and Inputs	Ein Klick auf Clear setzt das Layout auf die Standardansicht zurück und löscht getätigte Eingaben in der UI.
Settings	Ein Klick auf Clear löscht getätigte Einstellungen für diese Oberfläche.

3.2.9 Objects

Aktuell nur in Verbindung mit dem XPlanar zu verwenden, siehe [TF5430 | TwinCAT Planar Motion Dokumentation](#).

3.2.10 Erweiterungen

✓ Motion-Control-Ansichten über die Menüleiste öffnen.

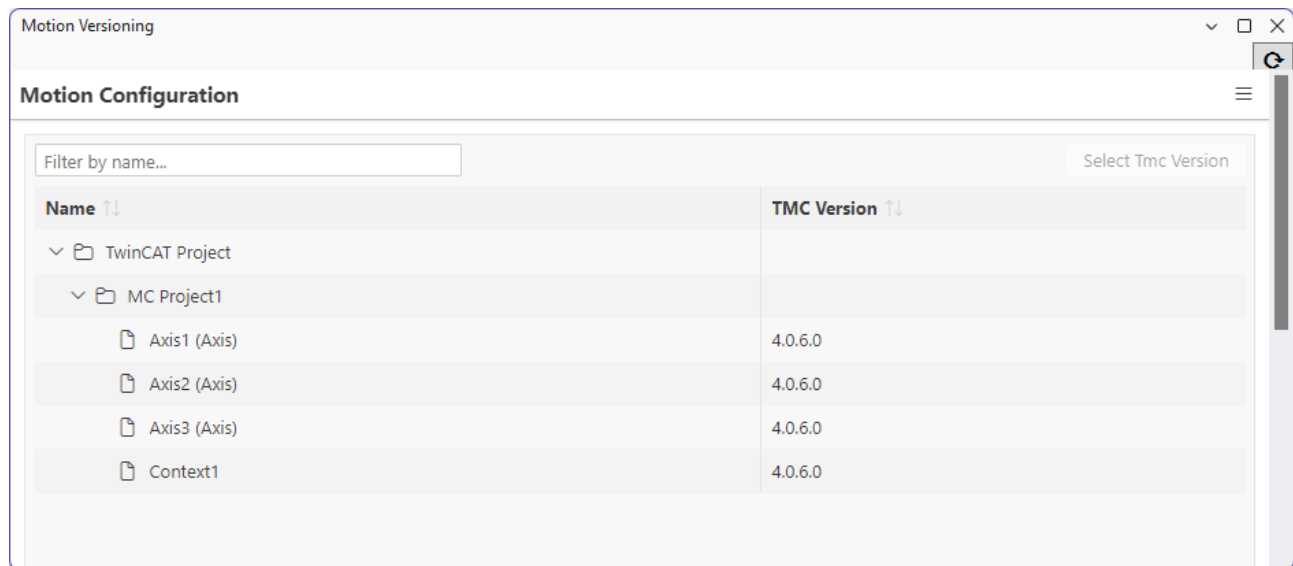
1. TwinCAT > MC3

2. Gewünschte Ansicht auswählen.

⇒ Die folgenden Ansichten stehen zur Verfügung.

Seite	Beschreibung
Motion Versioning	Verwaltung der TMC-Versionen der MC3-Motion-Objekte
Motion Axis View	Detailansicht und Steuerung einer einzelnen Achse
Motion Axes List View	Listenübersicht aller Achsen
Motion Cam Table View	Anzeige der Kurvenscheiben-Tabellen

Motion Versioning



Die Ansicht **Motion Configuration** dient zur Verwaltung und Übersicht aller Motion-Projekte und deren Bestandteile innerhalb eines TwinCAT-Projekts. Sie zeigt die hierarchische Struktur der konfigurierten Achsen, Kurvenscheiben und Kontexte an und ermöglicht die Verwaltung der zugehörigen TMC-Versionen.

Es gibt zwei Bedienelemente:

- **Select TMC Version:** Button oben rechts zum Auswählen/Ändern der Version für markierte Elemente. Er wird aktiv, wenn ein oder mehrere Elemente in der Baumstruktur ausgewählt sind.
- **Filter by name:** Textfeld oben links zum Filtern der Baumstruktur nach Namen. Eingabe eines Suchbegriffs blendet alle nichtzutreffenden Einträge aus.

Baumstruktur (Projektbaum)

Tabellarisch wird der Projektbaum mit allen zugehörigen Motion-Objekten abgebildet.

Spalte	Beschreibung
Name	Hierarchischer Name des Elements (sortierbar)
TMC-Version	Zugewiesene Version des Elements (sortierbar) Die Spalte TMC Version zeigt die Version der TwinCAT-Motion-Komponente, mit der das jeweilige Element konfiguriert wurde. Über Select TMC Version kann die Version bei Bedarf geändert werden, z. B. nach einem Update.

Vorgehen zur Versionsänderung

Option 1 (empfohlenes Vorgehen): Ändern der Versionen für alle Komponenten

Set TMC Version - Axis (1/3)

×

Select a TMC version to apply to Axis elements (3) and their children.

4.0.6.0

▼

☐ Apply to all item types

Cancel

Apply

Um die Version aller untergeordneten Komponenten auf einmal zu aktualisieren, wird das oberste Element in der Hierarchie ausgewählt (z. B. MC Project). Anschließend wird über **Select TMC Version** im Pop-up Fenster die gewünschte Zielversion für die einzelnen Komponenten ausgewählt.

Wird der Haken **Apply to all item types** gesetzt, wird die neue Version mit einer Bestätigung über **Apply** automatisch auf alle darunterliegenden Elemente (Achsen, Kurvenscheiben, Kontexte) übertragen. Das ist besonders vorteilhaft, da nicht jede Komponente einzeln umgestellt werden muss. Besonders nach einem Update können so einfach und schnell alle Komponenten eines Projekts konsistent auf dieselbe Version gebracht werden.

Wird der Haken **Apply to all item types** nicht gesetzt, muss jeweils für alle Objekte vom gleichen Typ (z. B. für alle Achsen) die Versionsnummer ausgewählt und bestätigt werden.

Option 2: Ändern der Versionen für einzelne Komponenten

Es ist technisch möglich, die Version für einzelne Komponenten separat zu ändern. Dazu wird das gewünschte Element (z. B. eine einzelne Achse) in der Baumstruktur ausgewählt und über **Select TMC Version** nur diesem Element eine neue Version zugewiesen.



Verhinderter Projektstart: Verschiedene Versionen innerhalb eines MC-Projekts vermeiden

Eine Verwendung von verschiedenen Versionen innerhalb eines MC-Projekts –also unterschiedliche Versionen für Achsen, Kurvenscheiben und Kontexte– wird ausdrücklich nicht empfohlen. Alle Komponenten eines MC-Projekts sollten stets dieselbe Version verwenden. Unterschiedliche Versionen können dazu führen, dass das Projekt nicht gestartet werden kann.

Motion Axis View

Diese Ansicht entspricht im Wesentlichen der Ansicht in der Registerkarte Registerkarte Motion UI [► 34] der Einzelachse. Vorteil: Diese Seite als Erweiterung kann mehrfach geöffnet werden, sodass verschiedene Einzelachsen in separaten Fenstern parallel betrachtet und bedient werden können.

Motion Axes List View

Diese Ansicht entspricht im Wesentlichen der Achsübersicht Axes [► 23] eines Motion Projekts. Vorteil: Diese Seite als Erweiterung kann mehrfach geöffnet werden, sodass verschiedene Achsübersichten parallel betrachtet und bedient werden können.

Motion Cam Table View

Diese Ansicht entspricht im Wesentlichen der Ansicht in der Registerkarte Camming UI [► 192] der Cam Tabelle. Vorteil: Diese Seite als Erweiterung kann mehrfach geöffnet werden, sodass verschiedene Kurvenscheiben in separaten Fenstern parallel betrachtet werden können.

3.3 Anhang

Wichtige Links

- [MOTION-Knoten anzeigen](#) [► 16]

MC Project

- [MC-Projekt erstellen](#) [► 17]
- [Einen neuen Kontext erstellen](#) [► 20]
- [Achsen einem Kontext \(Context\) zuordnen](#) [► 21]

Axis

- [Achsobjekt erstellen](#) [► 25]
- [Achsobjekt um ein zusätzliches Modul erweitern](#) [► 119]

- [PLC mit einem Achsobjekt verknüpfen \[► 120\]](#)
- [Achsobjekt mit Hardware verknüpfen \[► 124\]](#)
- [Versionshandling \[► 14\]](#)

4 Globale Datentypen und MC3 Typen

Cyclic Data Types (CDTs)

CDTs definieren Strukturen für den zyklischen Austausch zwischen MC3 und PLC. In der [Achregisterkarte Data Area \[► 33\]](#) ist jeweils eingestellt, welche CDTs für eine Achse mit der PLC gemappt und dementsprechend zyklisch mit Daten gefüllt werden.

Gibt es eine Verknüpfung zwischen MC3-Achse und PLC Tc3_Mc3PTP.AXIS_REF so sind immer CDT_PLCTOMC_AXIS_STD und CDT_MCTOPLC_AXIS_STD verknüpft. Alle anderen CDTs sind optional. Bei den optionalen CDTs ist vor jedem Lesen oder Schreiben zunächst das `IsConnected` zu prüfen. Nur wenn `IsConnected = TRUE` ist darf von der PLC auf die weitere Struktur zugegriffen werden.

HINWEIS

Nicht verknüpfte CDTs können bei Zugriff zu Floating Point Exception führen

Vergleiche mit NaN-Werten führen zu einer Exception, welche einen Stopp der Laufzeit und einen möglichen Maschinenschaden nach sich zieht.

Nur wenn das `IsConnected = TRUE` ist, sind valide Daten zu erwarten. Andernfalls besteht die Gefahr eines inkonsistenten oder ungültigen Datensatzes, was zu unabsehbaren Folgen und Fehlverhalten führen kann. Rechenoperationen mit ungültigen Werten können zu Ausnahmefehlern in der Gleitkommaeinheit des Prozessors führen.

- Schalten Sie die Floating Point Exceptions aus, wenn NaN-Werte in der Applikation verwendet und verarbeitet werden sollen.
- Prüfen Sie vor einem Zugriff (z. B. aus der PLC) auf ein CDT das `IsConnected` des CDTs.
- Machen Sie sich mit dem Umgang von benannten Konstanten und NaN Werten vertraut.

4.1 Data Types

4.1.1 Enums

4.1.1.1 EAxisLicenseSetting

Syntax

Definition:

```
TYPE EAxisLicenseSetting :
(
    AutoBeckhoff      := 1,
    AutoMultiVendor   := 3,
    Beckhoff           := 5,
    MultiVendor        := 7
) UDINT;
END_TYPE
```

Values

Name	Description
AutoBeckhoff	
AutoMultiVendor	
Beckhoff	
MultiVendor	

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.2 EAxisState

Syntax

Definition:

```

TYPE EAxisState :
(
    Invalid           := 0,
    Disabled          := 1,
    Standstill        := 2,
    Homing            := 3,
    ErrorStop         := 4,
    Stopping          := 5,
    DiscreteMotion    := 6,
    ContinuousMotion  := 7,
    SynchronizedMotion := 8,
    ResetPending      := 10,
    Enabling          := 11,
    Disabling         := 12,
    NotReady          := 13
) UINT;
END_TYPE

```

Values

Name	Description
Invalid	Axis object is not initialized or in an undefined state.
Disabled	Axis is disabled and the drive power stage is off.
Standstill	Axis is enabled and at standstill, ready to accept motion commands.
Homing	Axis is performing a homing procedure to establish a reference position.
ErrorStop	Axis has stopped due to an error. Error must be reset before further motion.
Stopping	Axis is decelerating to standstill during a stop command and remains in this state until Execute for stop command is released.
DiscreteMotion	Axis is executing a point-to-point motion command with a defined target position.
ContinuousMotion	Axis is executing a continuous velocity motion without a defined end position.
SynchronizedMotion	Axis is coupled to a master axis and following synchronized setpoints (e.g. gearing, camming).
ResetPending	An error reset has been requested and is being processed.
Enabling	Axis is in the process of being enabled (drive power stage is starting up).
Disabling	Axis is in the process of being disabled (drive power stage is shutting down).
NotReady	Axis is not ready for operation, e.g. waiting for required sub-modules to initialize or to start up.

Bedeutung der Zustände

Name	Beschreibung
Invalid	- Ungültiger Achszustand - Die Achse ist nicht betriebsbereit. - Mögliche Ursache kann eine fehlende Verknüpfung der AXIS_REF mit einer MC3-Achse sein.
Disabled	Grundzustand einer Achse.

Name	Beschreibung
	- Die Achse ist nicht freigegeben. - Es liegt kein Achsfehler vor.
Standstill	- Die Achse ist freigegeben. - Es liegt kein Kommando an der Achse an.
Homing	- Die Achse ist freigegeben. - Ein Homing der Achse wird durchgeführt (z. B. MC_Home).
ErrorStop	- Ein Achsfehler ist aufgetreten. - Ein Reset ist erforderlich, um den Zustand zu verlassen.
Stopping	- Die Achse ist freigegeben. - Die Achse befindet sich in einer Stopprampe (MC_Stop). - MC_Stop.Execute ist gesetzt und blockiert neue Motion Kommandos.
DiscreteMotion	- Die Achse ist freigegeben. - Die Achse führt eine Punkt-zu-Punkt Bewegung aus (z. B. MC_MoveAbsolute, MC_MoveRelative, MC_MoveModulo, MC_Halt).
ContinuousMotion	- Die Achse ist freigegeben. - Die Achse führt eine kontinuierliche Bewegung aus (z. B. MC_MoveVelocity).
SynchronizedMotion	- Die Achse ist freigegeben. - Die Achse ist an eine andere Achse gekoppelt (z. B. MC_CamIn, MC_GearIn, MC_GearInPos).
ResetPending	Interner Zustand
Enabling	- Erweiterter Achsstatus zur Darstellung von Hardwarezuständen. - Befehlswort zur Freigabe der Endstufe der Antriebshardware ist bereits gesendet oder wird gerade gesendet und die MC3-Achse wartet auf den Zustand "operational" der Antriebshardware
Disabling	- Erweiterter Achsstatus zur Darstellung von Hardwarezuständen. - Befehlswort zur Deaktivierung der Endstufe der Antriebshardware ist bereits gesendet oder wird gerade gesendet und die MC3-Achse wartet auf den Zustand "not operational" der Antriebshardware.
NotReady	- Erweiterter Achsstatus zur Darstellung von Hardwarezuständen. - Die Achse ist nicht betriebsbereit. - Es findet keine aktive Kommunikation zur Hardware statt (z. B. WcState) oder die Zustandsmaschine des Antriebs meldet „not ready“.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.3 EContinuity

Syntax

Definition:

```

TYPE EContinuity :
(
    Position                := 0,
    PositionVelocity        := 1,
    PositionVelocityAcceleration := 2,
    PositionVelocityAccelerationJerk := 3
) UINT;
END_TYPE

```

Values

Name	Description
Position	Position profile is continuous.
PositionVelocity	Position and velocity profiles are continuous.
PositionVelocityAcceleration	Position, velocity and acceleration profiles are continuous.
PositionVelocityAccelerationJerk	Position, velocity, acceleration and jerk profiles are continuous.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.4 EControlErrorSetpointMode**Syntax**

Definition:

```
TYPE EControlErrorSetpointMode :  
(  
    HardStop := 1  
) UDINT;  
END_TYPE
```

Values

Name	Description
HardStop	The dynamics of the axis are set to zero.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.5 EControlValueType**Syntax**

Definition:

```
TYPE EControlValueType :  
(  
    Velocity := 0x1  
) UDINT;  
END_TYPE
```

Values

Name	Description
Velocity	Controller output is a velocity setpoint.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.6 ECylinderDesign

Syntax

Definition:

```
TYPE ECylinderDesign :
(
    Differential := 0,
    Synchronous := 1,
    Plunger      := 2
) UDINT;
END_TYPE
```

Values

Name	Description
Differential	Differential cylinder with different piston areas on extend and retract sides.
Synchronous	Synchronous cylinder with equal piston areas on both sides.
Plunger	Plunger cylinder with a single-acting piston (no rod-side area).

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.7 EDeadTimeCompensationMode

Syntax

Definition:

```
TYPE EDeadTimeCompensationMode :
(
    None                := 0,
    LastVelocity         := 1,
    LastVelocityAcceleration := 2,
    NonlinearShift       := 10
) UDINT;
END_TYPE
```

Values

Name	Description
None	No dead time compensation applied.
LastVelocity	Compensation uses the last known velocity to extrapolate the actual position.
LastVelocityAcceleration	Compensation uses the last known velocity and acceleration for more accurate extrapolation.
NonlinearShift	Compensation applies a non-linear time shift method for improved accuracy in dynamic motion. The history of the last actual values is taken into account.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.8 EEncoderDataRange

Syntax

Definition:

```
TYPE EEncoderDataRange :  
(  
    FullRange    := 1,  
    UserDefined  := 2  
) UDINT;  
END_TYPE
```

Values

Name	Description
FullRange	Position feedback overflow position based on CDT limits.
UserDefined	Manual entry of the overflow position.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.9 EEncoderFeedbackVeloSource

Syntax

Definition:

```
TYPE EEncoderFeedbackVeloSource :  
(  
    HardwareProvidedIfAvailable := 0x0,  
    InternallyCalculated        := 1,  
    HardwareProvided            := 2  
) UDINT;  
END_TYPE
```

Values

Name	Description
HardwareProvidedIfAvailable	Uses hardware-provided velocity feedback if available, otherwise falls back to internal calculation (default).
InternallyCalculated	Velocity is always calculated internally by differentiating the position feedback.
HardwareProvided	Velocity is always taken from hardware feedback. Requires hardware support.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.10 EEncoderInterpretation

Syntax

Definition:

```
TYPE EEncoderInterpretation :  
(  
    AbsoluteFullRange := 0,
```

```

    AbsoluteUserDataRange := 2,
    Incremental            := 4,
    AbsoluteModulo         := 5
) UDINT;
END_TYPE

```

Values

Name	Description
AbsoluteFullRange	Absolute encoder using the full data range defined by the data type of mapping element (default).
AbsoluteUserDataRange	Absolute encoder with user-defined absolute data range and overflow position.
Incremental	Incremental encoder providing relative position changes only. For example, if position recovery after a power cycle is not possible. A homing sequence is necessary.
AbsoluteModulo	Absolute encoder interpreted as modulo position within the configured range if the modulo operating mode is set on drive/encoder device.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.11 EFilterType

Syntax

Definition:

```

TYPE EFilterType :
(
    None           := 0,
    PT1            := 1,
    MovingAverage  := 2,
    KalmanFilter3x1 := 3
) UDINT;
END_TYPE

```

Values

Name	Description
None	No filtering applied.
PT1	First-order low-pass filter.
MovingAverage	Moving average filter that smooths over a configurable number of samples.
KalmanFilter3x1	Kalman filter with 3-state model.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.12 EHomingObjectType

Syntax

Definition:

```

TYPE EHomingObjectType :
(
    Invalid      := 0,
    Cam          := 1,
    FixedStop    := 2
) UDINT;
END_TYPE

```

Values

Name	Description
Invalid	No valid homing object configured.
Cam	Homing uses a reference cam or proximity switch to locate the home position.
FixedStop	Homing drives against a mechanical fixed stop to determine the home position.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.13 EHomingSyncSource

Homing Sync Source

Syntax

Definition:

```

TYPE EHomingSyncSource :
(
    HardwareLatch1           := 0,
    ZeroImpulseCoE           := 1,
    HardwareLatch1CoEDriveDefined := 2
) UDINT;
END_TYPE

```

Values

Name	Description
HardwareLatch1	Synchronization to reference position via hardware latch 1 input on the EtherCAT slave.
ZeroImpulseCoE	Synchronization to reference position via encoder zero impulse (index signal) through CoE drive interface.
HardwareLatch1CoEDriveDefined	Synchronization to reference position via hardware latch 1 with drive-defined CoE configuration.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.14 ELoadType

Indicates the type of the currently selected load of the axis.

Syntax

Definition:

```

TYPE ELoadType :
(
    None           := 0,
    Force          := 2,
    PressureA      := 3,
    PressureB      := 4,
    DirectVelocityOutput := 5
) UDINT;
END_TYPE

```

Values

Name	Description
None	No load control active.
Force	Load control based on force.
PressureA	Load control based on pressure at area A of the cylinder.
PressureB	Load control based on pressure at area B of the cylinder.
DirectVelocityOutput	Velocity setpoint is written directly to the drive output.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.15 EMcTaskInternalSequence

Syntax

Definition:

```

TYPE EMcTaskInternalSequence :
(
    FastReaction := 0,
    Stable       := 1
) UDINT;
END_TYPE

```

Values

Name	Description
FastReaction	Execution order: Commands - Exec - Output. Processes new commands before output update, providing the fastest reaction to new commands.
Stable	Execution order: Output - Commands - Exec. Output update is executed first, then new commands are processed, providing less jitter for output update.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.16 EPositionLagSource

Syntax

Definition:

```

TYPE EPositionLagSource :
(
    HardwareProvidedIfAvailable := 0,
    InternallyCalculated        := 1,

```

```

    HardwareProvided      := 2
) UDINT;
END_TYPE

```

Values

Name	Description
HardwareProvidedIfAvailable	Uses hardware-provided position lag if available, otherwise falls back to internal calculation.
InternallyCalculated	Position lag is always calculated internally from position of setpoint and actual position.
HardwareProvided	Position lag is always taken from hardware. Requires hardware support and mapping.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.17 EPositionUnit

Syntax

Definition:

```

TYPE EPositionUnit :
(
    um    := 0,
    mm    := 1,
    m     := 2,
    in    := 3,
    deg   := 11,
    rad   := 12,
    gon   := 13
) UDINT;
END_TYPE

```

Values

Name	Description
um	Micrometers
mm	Millimeters
m	Meters
in	Inches
deg	Degrees (360 per revolution)
rad	Radians (2*pi per revolution)
gon	Gradians (400 per revolution)

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.18 ETouchProbeTriggerEdge

Touchprobe Trigger Edge

Syntax

Definition:

```

TYPE ETouchProbeTriggerEdge :
(
    Positive := 1,
    Negative := 2
) UDINT;
END_TYPE

```

Values

Name	Description
Positive	Trigger on rising edge of the input signal.
Negative	Trigger on falling edge of the input signal.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.19 ETouchProbeTriggerSource

Selection for CoE Drive touch probe input

Syntax

Definition:

```

TYPE ETouchProbeTriggerSource :
(
    DriveDefault := 0,
    ZeroImpulse := 1,
    DriveDefined := 2
) UDINT;
END_TYPE

```

Values

Name	Description
DriveDefault	Trigger on the touch probe default input (i.e., the digital input 1/2 mapped to TouchProbe1/TouchProbe2 functionality in the drive).
ZeroImpulse	Trigger on the main encoder index signal (encoder Z/ index). Only valid if the drive exposes the index signal to the touch probe function.
DriveDefined	Trigger source is defined by object 0x60D0/0x8001 (DS402/MDP742).

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.1.20 ETouchProbeTriggerState

Touchprobe Trigger State Machine

Syntax

Definition:

```

TYPE ETouchProbeTriggerState :
(
    Init      := 0,
    Armed     := 1,
    Latched   := 2,
    Continuous := 3,

```

```

    Aborted      := 4
) UDINT;
END_TYPE

```

Values

Name	Description
Init	Touch probe is initialized but not yet armed.
Armed	Touch probe is armed and waiting for a trigger event.
Latched	Touch probe has been triggered and the position has been latched.
Continuous	Touch probe is in continuous mode, latching positions on every trigger event.
Aborted	Touch probe operation was aborted before a trigger event occurred.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2 Structs

4.1.2.1 CDT_MCTOPLC_AXIS_ACT

CDT provides current act values.

Syntax

Definition:

```

TYPE CDT_MCTOPLC_AXIS_ACT :
STRUCT
    IsConnected   : BOOL;
    Position      : LREAL;
    Velocity      : LREAL;
    Acceleration  : LREAL;
    PositionLag   : LREAL;
    Torque        : LREAL;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Description
IsConnected	BOOL	TRUE if the data area is connected and data is being received from MC.
Position	LREAL	Current actual position of the axis.
Velocity	LREAL	Current actual velocity of the axis.
Acceleration	LREAL	Current actual acceleration of the axis.
PositionLag	LREAL	Difference between set position and actual position (position following error).
Torque	LREAL	Current actual torque of the axis.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.2 CDT_MCTOPLC_AXIS_FLUIDPOWER

CDT provides fluid power specific setpoint and actpoint.

Syntax

Definition:

```
TYPE CDT_MCTOPLC_AXIS_FLUIDPOWER :
STRUCT
    IsConnected    : BOOL;
    ActPressureA   : LoadData;
    ActPressureB   : LoadData;
    ActForce       : LoadData;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
IsConnected	BOOL	TRUE if the data area is connected and data is being received from MC.
ActPressureA	LoadData [► 215]	PressureA actpoint
ActPressureB	LoadData [► 215]	PressureB actpoint
ActForce	LoadData [► 215]	Force actpoint

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.3 CDT_MCTOPLC_AXIS_LOADCONTROL

CDT provides load setpoint and actpoint.

Syntax

Definition:

```
TYPE CDT_MCTOPLC_AXIS_LOADCONTROL :
STRUCT
    IsConnected    : BOOL;
    IsInLoadMode   : BOOL;
    LoadType      : ELoadType;
    Set            : LoadData;
    Act            : LoadData;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
IsConnected	BOOL	TRUE if the data area is connected and data is being received from MC.
IsInLoadMode	BOOL	True if the axis is producing load setpoints, false otherwise.
LoadType	ELoadType [► 205]	Load selected for load control.
Set	LoadData [► 215]	Current load setpoint.
Act	LoadData [► 215]	Current load actpoint.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.4 CDT_MCTOPLC_AXIS_MODULO

CDT provides information about modulo positioning.

Syntax

Definition:

```
TYPE CDT_MCTOPLC_AXIS_MODULO :  
STRUCT  
    IsConnected : BOOL;  
    SetPosition : LREAL;  
    SetTurns    : LINT;  
    ActPosition : LREAL;  
    ActTurns    : LINT;  
END_STRUCT  
END_TYPE
```

Parameters

Name	Type	Description
IsConnected	BOOL	TRUE if the data area is connected and data is being received from MC.
SetPosition	LREAL	Modulo set position within one modulo period.
SetTurns	LINT	Number of complete modulo turns for the set position.
ActPosition	LREAL	Modulo actual position within one modulo period.
ActTurns	LINT	Number of complete modulo turns for the actual position.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.5 CDT_MCTOPLC_AXIS_SET

CDT provides current set values.

Syntax

Definition:

```
TYPE CDT_MCTOPLC_AXIS_SET :  
STRUCT  
    IsConnected : BOOL;  
    Position     : LREAL;  
    Velocity     : LREAL;  
    Acceleration : LREAL;  
END_STRUCT  
END_TYPE
```

Parameters

Name	Type	Description
IsConnected	BOOL	TRUE if the data area is connected and data is being received from MC.
Position	LREAL	Current position setpoint of the axis.
Velocity	LREAL	Current velocity setpoint of the axis.
Acceleration	LREAL	Current acceleration setpoint of the axis.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.6 CDT_MCTOPLC_AXIS_STD**Syntax**

Definition:

```

TYPE CDT_MCTOPLC_AXIS_STD :
STRUCT
    AxisOid          : OTCID;
    AxisState        : EAxisState;
    IsInControl       : BOOL;
    HasError          : BOOL;
    ErrorId           : UDINT;
    IsPositionValid   : BOOL;
    DcTime            : LINT;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Description
AxisOid	OTCID	Object ID of the axis for identification.
AxisState	EAxisState [► 199]	Current state of the axis state machine.
IsInControl	BOOL	TRUE if the axis controller is actively controlling the drive (control loop closed).
HasError	BOOL	TRUE if the axis is in an error state.
ErrorId	UDINT	Error code of the current axis error (0 if no error).
IsPositionValid	BOOL	TRUE if the reported position values are valid (e.g. after homing or with absolute encoder and drive/encoder signals a valid position).
DcTime	LINT	Distributed clock timestamp of the cyclic data exchange.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.7 CDT_MCTOPLC_CAMTABLE

Syntax

Definition:

```
TYPE CDT_MCTOPLC_CAMTABLE :
STRUCT
    Oid : OTCID;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
Oid	OTCID	Object ID of the cam table instance. Must be mapped to an MC_CamObject instance in the PLC.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.8 CDT_MCTOPLC_ENCODER_AXIS_ACT

CDT provides current actual values from encoder axis.

Syntax

Definition:

```
TYPE CDT_MCTOPLC_ENCODER_AXIS_ACT :
STRUCT
    IsConnected : BOOL;
    Position : LREAL;
    Velocity : LREAL;
    Acceleration : LREAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
IsConnected	BOOL	TRUE if the data area is connected and data is being received from MC.
Position	LREAL	Current actual position of the encoder axis.
Velocity	LREAL	Current actual velocity of the encoder axis.
Acceleration	LREAL	Current actual acceleration of the encoder axis.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.9 CDT_MCTOPLC_ENCODER_AXIS_MODULO

CDT provides modulo positioning information from encoder axis.

Syntax

Definition:

```

TYPE CDT_MCTOPLC_ENCODER_AXIS_MODULO :
STRUCT
    IsConnected : BOOL;
    Position    : LREAL;
    Turns       : LINT;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Description
IsConnected	BOOL	TRUE if the data area is connected and data is being received from MC.
Position	LREAL	Modulo position within one modulo period.
Turns	LINT	Number of complete modulo turns.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.10 CDT_MCTOPLC_ENCODER_AXIS_STD

Standard cyclic data from MC encoder axis to PLC.

Syntax

Definition:

```

TYPE CDT_MCTOPLC_ENCODER_AXIS_STD :
STRUCT
    EncoderAxisOid : OTCID;
    HasError        : BOOL;
    IsPositionValid : BOOL;
    ErrorId         : UDINT;
    DcTime          : LINT;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Description
EncoderAxisOid	OTCID	Object ID of the encoder axis for identification.
HasError	BOOL	TRUE if the encoder axis is in an error state.
IsPositionValid	BOOL	TRUE if the reported position values are valid.
ErrorId	UDINT	Error code of the current encoder axis error (0 if no error).
DcTime	LINT	Distributed clock timestamp of the cyclic data exchange.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.11 CDT_MCTOPLC_VALVECHARACTERISTICCURVE

Syntax

Definition:

```
TYPE CDT_MCTOPLC_VALVECHARACTERISTICCURVE :  
STRUCT  
    Oid : OTCID;  
END_STRUCT  
END_TYPE
```

Parameters

Name	Type	Description
Oid	OTCID	Object ID of the valve characteristic curve instance.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.12 CDT_PLCTOMC_AXIS_STD

Syntax

Definition:

```
TYPE CDT_PLCTOMC_AXIS_STD :  
STRUCT  
    Enable : BOOL;  
    EnablePositive : BOOL;  
    EnableNegative : BOOL;  
END_STRUCT  
END_TYPE
```

Parameters

Name	Type	Description
Enable	BOOL	Enables the axis power stage.
EnablePositive	BOOL	Allows motion in positive direction when TRUE.
EnableNegative	BOOL	Allows motion in negative direction when TRUE.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.1.2.13 LoadData

Syntax

Definition:

```
TYPE LoadData :  
STRUCT  
    Value : LREAL;  
    Derivative : LREAL;  
END_STRUCT  
END_TYPE
```

Parameters

Name	Type	Description
Value	LREAL	Current load value in the configured load unit.
Derivative	LREAL	First time derivative of the load value.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.2 Interfaces

4.2.1 ITcMcExternalSetpointGenerator

Interface for implementing custom external setpoint generators.

Inheritance Hierarchy

```
ITcUnknown
  ITcMcExternalSetpointGenerator
```

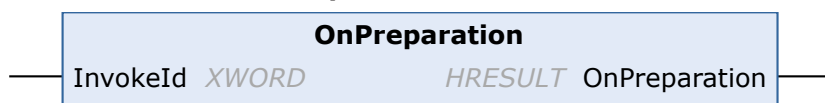
 Methods

Name	Origin	Description
TcAddRef	ITcUnknown	Increments the reference counter.
TcQueryInterface	ITcUnknown	Get a reference to an implemented interface.
TcRelease	ITcUnknown	Decrements the reference counter.
OnPreparation [► 216]	ITcMcExternalSetpointGenerator	Called once when the external setpoint generator is being prepared for activation.
OnStartPointInitialization [► 217]	ITcMcExternalSetpointGenerator	Called once to provide the initial motion state before cyclic setpoint generation begins.
OnCyclicSetpointGeneration [► 218]	ITcMcExternalSetpointGenerator	Called every cycle to compute the next setpoint values for position, velocity, and acceleration.
OnDeactivation [► 218]	ITcMcExternalSetpointGenerator	Called when the external setpoint generator is deactivated/aborted. Use this method to tidy up, e. g. release created objects.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

4.2.1.1 OnPreparation



Called once when the external setpoint generator is being prepared for activation.

Syntax

Definition:

```

METHOD OnPreparation : HRESULT
VAR_INPUT
    InvokeId : XWORD;
END_VAR

```

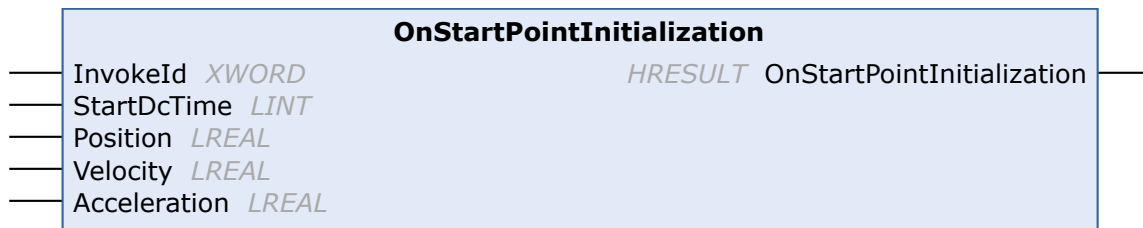
Inputs

Name	Type	Description
Invokeld	XWORD	Unique identifier, the interpretation is user specific.

Return value

HRESULT

4.2.1.2 OnStartPointInitialization



Called once to provide the initial motion state before cyclic setpoint generation begins.

Syntax

Definition:

```

METHOD OnStartPointInitialization : HRESULT
VAR_INPUT
    InvokeId      : XWORD;
    StartDcTime   : LINT;
    Position      : LREAL;
    Velocity      : LREAL;
    Acceleration  : LREAL;
END_VAR

```

Inputs

Name	Type	Description
Invokeld	XWORD	Unique identifier, the interpretation is user specific.
StartDcTime	LINT	DC time at which the generation starts.
Position	LREAL	Start position of the axis.
Velocity	LREAL	Start velocity of the axis.
Acceleration	LREAL	Start acceleration of the axis.

Return value

HRESULT

4.2.1.3 OnCyclicSetpointGeneration



Called every cycle to compute the next setpoint values for position, velocity, and acceleration.

Syntax

Definition:

```
METHOD OnCyclicSetpointGeneration : HRESULT
VAR_INPUT
    InvokeId      : XWORD;
    TimeInGeneration : LREAL;
    Position      : Reference To LREAL;
    Velocity      : Reference To LREAL;
    Acceleration   : Reference To LREAL;
END_VAR
```

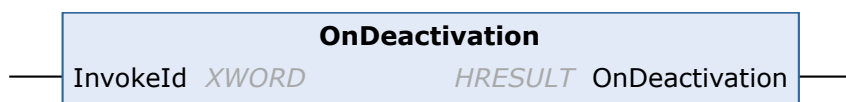
Inputs

Name	Type	Description
InvokeId	XWORD	Unique identifier, the interpretation is user specific.
TimeInGeneration	LREAL	Elapsed time since the start of setpoint generation (input from PLC side of view).
Position	Reference To LREAL	Reference to receive the computed position setpoint. Value must be set during method call (output from PLC side of view).
Velocity	Reference To LREAL	Reference to receive the computed velocity setpoint. Value must be set during method call (output from PLC side of view).
Acceleration	Reference To LREAL	Reference to receive the computed acceleration setpoint. Value must be set during method call (output from PLC side of view).

Return value

HRESULT

4.2.1.4 OnDeactivation



Called when the external setpoint generator is deactivated/aborted. Use this method to tidy up, e. g. release created objects.

Syntax

Definition:

```
METHOD OnDeactivation : HRESULT
VAR_INPUT
    InvokeId : XWORD;
END_VAR
```

 Inputs

Name	Type	Description
Invokeld	XWORD	Unique identifier, the interpretation is user specific.

 Return value

HRESULT

4.3 Übersicht: MC3-Typen

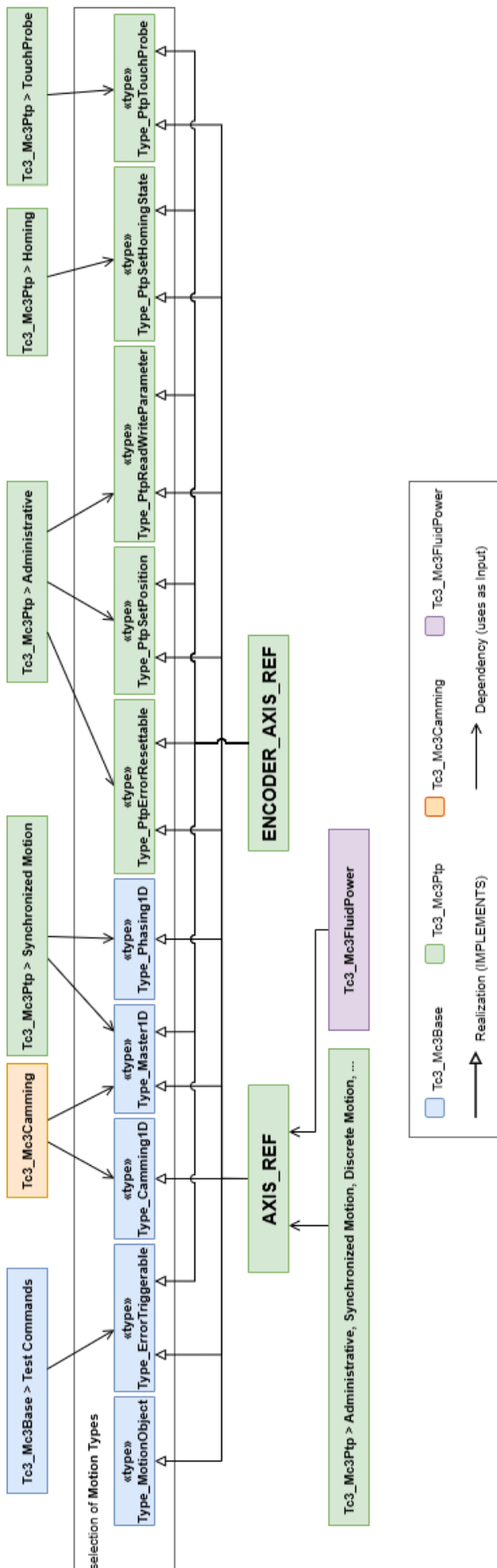
MC3-Funktionsbausteine verwenden typisierte Achsreferenzen, um die erforderlichen Fähigkeiten einer Achse festzulegen. Jeder Typ repräsentiert eine bestimmte Motion-Fähigkeit. Die Fähigkeit kann beispielsweise die Eignung als Kopplungsmaster sein oder die Möglichkeit der Achse, einen Positionssollwert entgegenzunehmen. Jedes Motion-Objekt (`AXIS_REF`, `ENCODER_AXIS_REF`) implementiert nur die Fähigkeiten, die es tatsächlich unterstützt.

Der Compiler prüft, ob das Motion-Objekt den für den Funktionsbaustein erforderlichen Typen implementiert. Ist dies nicht der Fall, lässt sich das Programm nicht kompilieren. Damit wird verhindert, dass Operationen für Motion-Objekte aufgerufen werden, die diese nicht unterstützen. Am Funktionsbaustein werden nur Fähigkeiten abgefragt und nicht das Motion-Objekt selbst. Der Compiler prüft, ob das Motion-Objekt diese Fähigkeit implementiert.

Die Grafik skizziert die Abhängigkeiten und Ursprünge der Typen (`Type_*`) und Motion-Objekte (`AXIS_REF`, `ENCODER_AXIS_REF`). In der Mitte sind die entscheidenden Ebenen zu erkennen:

- Typen: `Type_Master1D` usw., die definieren, was ein Motion-Objekt kann.
- Motion-Objekte: `AXIS_REF` und `ENCODER_AXIS_REF`, die im PLC-Programm deklariert werden.

Die MC3-Funktionsbausteine verwenden die spezialisierten Motion-Objekte oder die Typen direkt.



Type	AXIS_REF	ENCODER_AXIS_REF
Type_PtpErrorResettable	X	X
Type_PtpReadWriteParameter	X	X
Type_PtpSetHomingState	X	X
Type_PtpSetPosition	X	X
Type_PtpTouchProbe	X	X
Type_Camming1D	X	
Type_ErrorTriggerable	X	X
Type_Master1D	X	X
Type_MotionObject	X	
Type_Phasing1D	X	

Verwendung

Die Funktionsbausteine der Tc3_Mc3Base, Tc3_Mc3Ptp, Tc3_Mc3Camming und Tc3_Mc3FluidPower setzen eine Parametrierung mit spezialisierten Typen (Type_*) oder allgemeinere Motion-Objekte (AXIS_REF) voraus. Der oberen Grafik ist zu entnehmen, dass die allgemeineren Motion-Objekte ausgewählte Typen implementieren.

1. Legen Sie in der PLC für eine PTP oder hydraulische Achse wie gewohnt eine Instanz vom Typ `AXIS_REF` an. Die spezialisierten Typen sind nicht für die Verwendung als eigenständige Instanz geeignet.
2. Neben der `AXIS_REF` können weitere Motion-Objekte mit verschiedenen Fähigkeiten instanziiert werden (z. B. ein Positionsgeber als `ENCODER_AXIS_REF`).
 ⇒ Funktionsbausteine können nur mit Motion-Objekten verwendet werden, die die benötigten Fähigkeiten zur Verfügung stellen. Es ist nicht möglich, ein `MC_MoveAbsolute` mit einer `ENCODER_AXIS_REF` auszuführen. Dies führt zu einem Compilerfehler.

Die Benutzung der Typen an z. B. Kopplungsbausteinen ermöglicht die Wiederverwendung der Funktionsbausteine mit verschiedenen Motion-Objekten. Bei einem Funktionsbaustein-Eingang des Typen `Type_Camming1D` ist ersichtlich, dass dieser Parameter ein Objekt erfordert, das als Slave-Objekt für Kurvenscheiben fungieren kann. Eine fehlerhafte Verwendung von Motion-Objekten fällt bereits während der Kompilierung und nicht erst zur Laufzeit des Anwenderprogramms auf.

Die Tabelle zeigt eine Auswahl an Funktionsbausteinen und die verwendeten Motion-Objekte oder Typen.

Funktionsbaustein	Parameter	Erforderlicher Typ	AXIS_REF	ENCODER_AXIS_REF
Basis Kommandos				
MC_Power	Axis	AXIS_REF	X	-
MC_Home	Axis	AXIS_REF	X	-
MC_SetHomingState	Axis	Type_PtpSetHomingState	X	X
MC_Stop	Axis	AXIS_REF	X	-
MC_Halt	Axis	AXIS_REF	X	-
MC_Reset	Axis	Type_PtpErrorResettable	X	X
MC_MoveAbsolute	Axis	AXIS_REF	X	-
MC_MoveRelative	Axis	AXIS_REF	X	-
MC_MoveVelocity	Axis	AXIS_REF	X	-
MC_MoveModulo	Axis	AXIS_REF	X	-
MC_SetOverride	Axis	AXIS_REF	X	-
MC_SetPosition	Axis	Type_PtpSetPosition	X	X
MC_ReadParameter	Axis	Type_PtpReadWriteParameter	X	X
MC_WriteParameter	Axis	Type_PtpReadWriteParameter	X	X

Funktionsbaustein	Parameter	Erforderlicher Typ	AXIS_REF	ENCODER_AXIS_REF
MC_ReadBoolParameter	Axis	Type_PtpReadWriteParameter	X	X
MC_WriteBoolParameter	Axis	Type_PtpReadWriteParameter	X	X
MC_ReadHardwareParameter	Axis	AXIS_REF	X	X
MC_WriteHardwareParameter	Axis	AXIS_REF	X	X
Getriebekopplung				
MC_GearIn	Master	Type_Master1D	X	X
	Slave	AXIS_REF	X	-
MC_GearInPos	Master	Type_Master1D	X	X
	Slave	AXIS_REF	X	-
Kurvenscheiben				
MC_CamIn	Master	Type_Master1D	X	X
	Slave	Type_Camming1D	X	-
MC_CamTableSelect	Slave	Type_Camming1D	X	-
Phasing				
MC_PhasingAbsolute	Master	Type_Master1D	X	X
	Slave	Type_Phasing1D	X	-
MC_PhasingRelative	Master	Type_Master1D	X	X
	Slave	Type_Phasing1D	X	-
Messtaster				
MC_TouchProbe	Axis	Type_PtpTouchProbe	X	X
	TriggerInput	TRIGGER_REF	separater Typ	
MC_AbortTrigger	Axis	Type_PtpTouchProbe	X	X
	TriggerInput	TRIGGER_REF	separater Typ	

Beispiel

Deklaration:

```

VAR
  myAxis1      : AXIS_REF;
  myAxis2      : AXIS_REF;
  myEncoder    : ENCODER_AXIS_REF;
  myGearIn     : MC_GearIn;
  myTypeInstance : Type_Master1D;
END_VAR

```

Implementierung:

```

// valid call
myGearIn(
  Master:= myAxis1,
  Slave:= myAxis2,
  Execute:= TRUE);

// valid call
myGearIn(
  Master:= myEncoder,
  Slave:= myAxis2,
  Execute:= TRUE);

myGearIn(
  Master:= myAxis1,
  Slave:= myEncoder, // Compiler error
  Execute:= TRUE);

myGearIn(
  Master:= myTypeInstance, // FB runtime error 16#8145
  Slave:= myAxis2,
  Execute:= TRUE);

```

Das Kommando MC_GearIn erwartet am Eingang Slave ein Objekt vom Typ AXIS_REF. Somit ist es nicht möglich, eine Encoder-Achse vom Typ ENCODER_AXIS_REF zu verwenden, dies führt zu einem Compilerfehler. Gleiches gilt z. B. für das Kommando MC_CamIn, da ENCODER_AXIS_REF nicht den Typ Type_Camming1D anbietet.

MC3-Typen enthalten keinen Implementierungscode. Sie werden im Solution Explorer nicht für eine Verknüpfung zwischen einer MC3-Achse und einer Instanz in der PLC angeboten. Im letzten Beispiel ist eine Instanz des Typen Type_Master1D dem Eingang Master zugewiesen. Dies führt nicht zu einem Compilerfehler, da der Typ grundsätzlich der richtige ist. Die Logik des Funktionsbaustein erkennt aber eine fehlende Verknüpfung zu einer Achse und quittiert dies mit dem Fehler 16#8145.

Migration NC2 zu MC3

NC2	MC3	Erforderliche Aktion
Axis : Tc2_MC2.AXIS_REF	Axis : Tc3_Mc3Ptp.AXIS_REF	Keine – In der MC3 verwenden die Funktionsblöcke die Motion-Objekte oder die Typen als Input. Da die Motion-Objekte entsprechende Typen anbieten, muss der Input am Funktionsbaustein sowie die Deklaration nicht verändert werden.
Encoder : Tc2_MC2.AXIS_REF	Encoder : Tc3_Mc3Ptp.ENCODER_AXIS_REF	Für Encoder-Achsen ist in der MC3 die eigenständige ENCODER_AXIS_REF zu wählen.

5 PLC Bibliotheken

5.1 Benannte Konstanten

Benannte Konstanten

Für Eingänge an Funktionsbausteinen vom Typ `MC_LREAL` können spezielle Eingangswerte verwendet werden (siehe Kapitel [MC_LREAL/Spezielle Eingangswerte](#)). Die MC3 nutzt ausschließlich den Standard-Datentyp `LREAL` anstatt `MC_LREAL` und unterstützt die Verwendung der so genannten *Benannten Konstanten*.

Benannte Konstanten sind stille NaN-Werte nach IEEE 754, um nicht initialisierte oder ungültige Daten zu signalisieren. Die Mantisse ist in einem Beckhoff-spezifischen Bit-Pattern kodiert. Der Vorteil ist eine erhöhte Lesbarkeit z. B. in der Online-Ansicht. So kann z. B. der bisherige Default-Wert 0 als Platzhalter für die Default-Dynamiken oder den realen Wert 0 interpretiert werden. In diesem Fall sind Fehlinterpretationen oder falsche mathematische Operationen möglich.

Zu den wesentlichen Eigenschaften von NaN-Werten zählen die folgenden Punkte:

- Alle arithmetischen Operationen, die NaN als Eingangsdaten verwenden, liefern wiederum NaN als Ergebnis.
- Alle relationalen Operatoren `=`, `!=`, `>`, `<`, `>=`, `<=` liefern stets den Wert `False`, wenn mindestens einer der Operanden NaN ist.
- Die Standard-C-Funktionen `isnan()` bzw. `_isnan()` oder die SPS-Funktionen `LrealIsNaN()` und `ReallIsNaN()` (Tc2_Uilities Bibliothek) liefern den Wert `True`, wenn das Argument den Wert NaN hat.
- Der Ausdruck `isnan(a)` ist äquivalent dem Ausdruck `!(a == a)` bzw. `NOT(a = a)`.

Die Tatsache, dass NaN-Werte sich bei der Verwendung in weiteren Berechnungen fortpflanzen, hat den Vorteil, dass ungültige Werte nicht übersehen werden können.

⚠ VORSICHT

Fehlfunktionen der Software

NaN-Werte führen zu potenziell gefährlichen Fehlfunktionen der betreffenden Software!

- Verwenden Sie nur ausdrücklich zugelassene NaN-Werte insbesondere als Stellwerte in Funktionen für Motion Control und zur Antriebssteuerung.

Weitere Erläuterungen finden Sie im Kapitel [NaN-Werte](#).

Die Syntax zur Verwendung in der PLC ist `STRING_TO_LREAL(' [Wert] ')` oder die Kurzform `TO_LREAL(' [Wert] ')`. Für die MC3 können die folgenden Konstanten verwendet werden:

Wert	Bedeutung	Beispiel
#Invalid	Ungültiger Wert	Default Eingangswert bei <code>MC_MoveVelocity.Velocity</code> , da dieser Eingang bei der Verwendung des Bausteins beschrieben werden muss.
#Default	Kann verwendet werden, um z. B. die Default-Dynamik der Achse als Eingabe für Funktionsbausteine zu verwenden.	<code>MC_MoveAbsolute.Velocity : = TO_LREAL('#Default');</code>
#Ignore	Kann verwendet werden, um Eingänge oder Randbedingungen zu deaktivieren.	Einseitiges deaktivieren der Softwareendlagen. Ableitungen der Position als Eingabe bzw. Randbedingung für die <code>MotionFunctionPoints</code> (Camming) als zu ignorieren zu definieren.

Wert	Bedeutung	Beispiel
#Maximum	Kann verwendet werden, um z. B. die maximalen Dynamiken der Achse als Eingabe für Funktionsbausteine zu verwenden.	MC_MoveVelocity.Velocity : = TO_LREAL('#Maximum');
#MaximumNegative	Kann verwendet werden, um z. B. die invertierten maximalen Dynamiken der Achse als Eingabe für Funktionsbausteine zu verwenden.	MC_MoveVelocity.Velocity : = TO_LREAL('#MaximumNegative');
#DefaultNegative	Kann verwendet werden, um z. B. die invertierten Default-Dynamiken der Achse als Eingabe für Funktionsbausteine zu verwenden.	MC_MoveVelocity.Velocity : = TO_LREAL('#DefaultNegative');

Die Syntax und Verwendung benannter Konstanten sind mit TwinCAT 3.1 Build 4026.25 optimiert und nachfolgend beschrieben.



Verfügbar ab TwinCAT 3.1 Build 4026.25

Sonderwerte wie z. B. NaN (Not a Number) werden nun als typsichere Konstante bereitgestellt. Über den Namensraum LREAL_CONST erhält man Zugriff auf folgende Variablen:

Konstante	Bedeutung
DEFAULT	Anwendungsspezifischer Standardwert (positiv)
IGNORE	Markerwert „nicht auswerten“
INVALID	Markerwert für ungültige Größe
MAXIMUM	Größter endlicher positiver Wert
DEFAULT_NEGATIVE	Anwendungsspezifischer Standardwert (negativ)
MAXIMUM_NEGATIVE	Größter endlicher negativer Wert
NAN	Not a Number
INFINITY	Positiv unendlich
INFINITY_NEGATIVE	Negativ unendlich

Beispiel: Verwendung

```
VAR
    fNumerator : LREAL;
    fDenominator : LREAL;
    fResult : LREAL
END_VAR
IF fDenominator = 0 THEN
    fDenominator := LREAL_CONST.NAN; // invalid: division by zero
END_IF
fResult := fNumerator / fDenominator;
```

5.2 Tc3_Mc3Base

Die Bibliothek Tc3_Mc3Base enthält Basis-Datentypen, welche übergreifend von den weiteren MC3-Bibliotheken verwendet werden.

5.2.1 Data Types

5.2.1.1 Enums

5.2.1.1.1 EBufferMode

Buffer mode between buffered commands.

Syntax

Definition:

```
TYPE EBufferMode :
(
  Aborting := 0,
  Buffered := 1
) UINT;
END_TYPE
```

Values

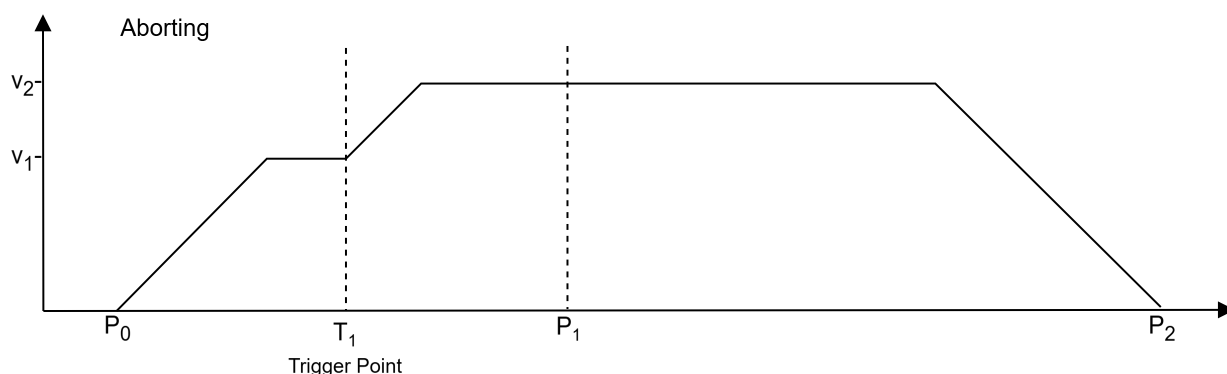
Name	Description
Aborting	The previous command is aborted and a transition to the new command is initiated immediately.
Buffered	The previous command is executed to completion. The new command affects the motion object as soon as the previous movement is 'Done'/'InVelocity'/'InSync'/'InGear'. There is no blending.

Weiterführende Informationen

Der Datentyp `EBufferMode` wird bei Fahrkommandos verwendet, um festzulegen, wie diese das vorherige Fahrkommando ablösen.

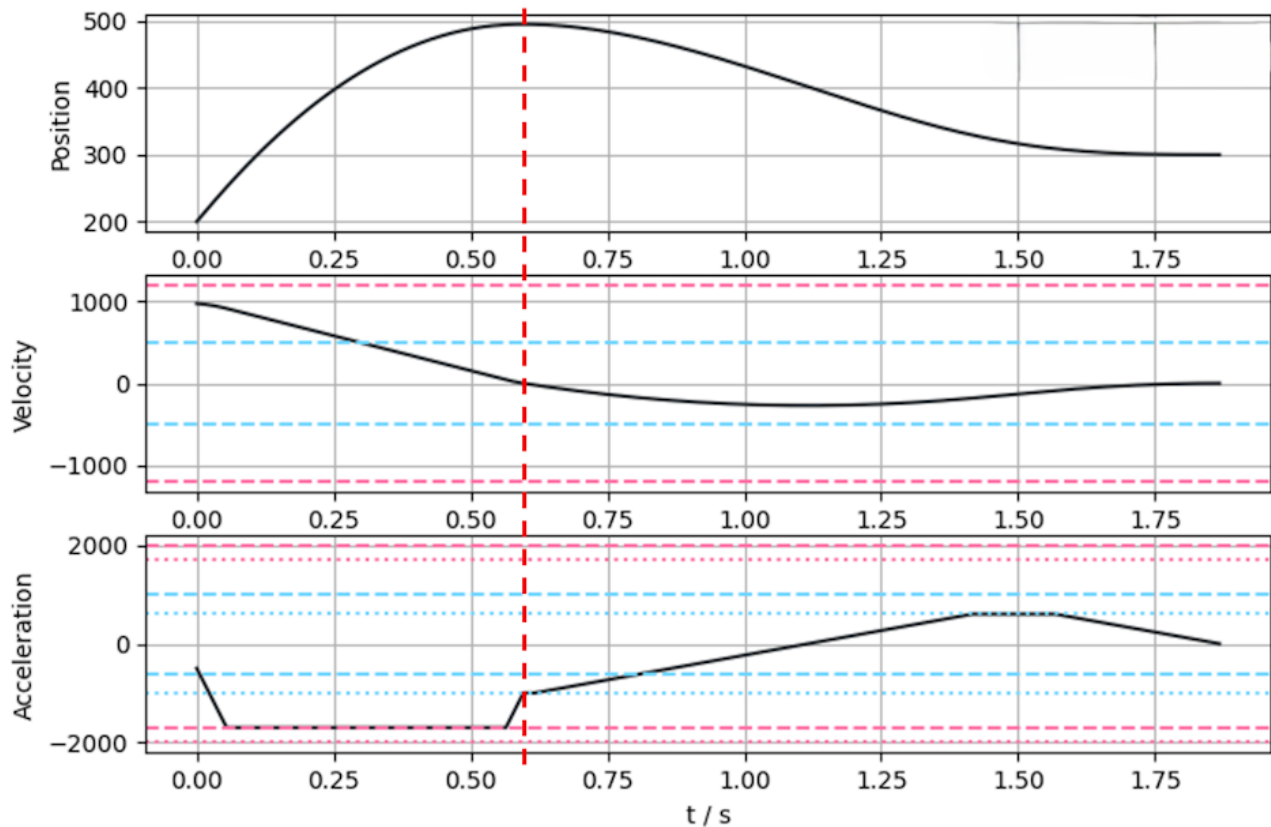
Aborting

Das vorherige Kommando wird sofort abgebrochen und signalisiert `CommandAborted = TRUE`. Sind für das neue Kommando höhere Dynamiken parametrisiert, werden diese instantan als Limit für die Achsbewegung übernommen.



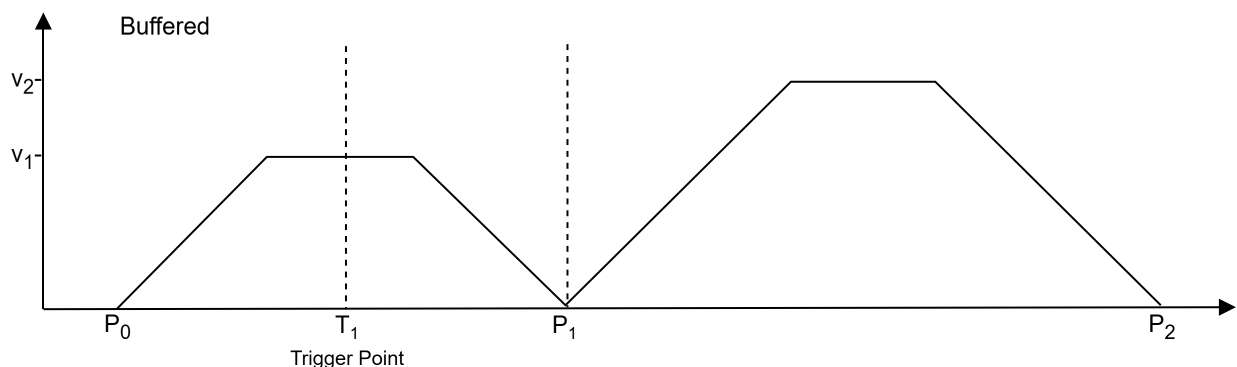
Sind für das neue Kommando geringere Dynamiken parametrisiert, kann es zu einer Übergangsphase kommen. Die Achse wird so schnell wie möglich in den Zielzustand bzw. in den Bereich der neuen Dynamiklimits überführt, wobei Über- oder Unterschwingen unterdrückt werden (z. B. das Überschwingen über die Zielposition des vorherigen Kommandos). Ist dies auch mit den verringerten Dynamiken möglich, werden diese instantan übernommen. Andernfalls wird die Achse mit den höheren Dynamiken des vorherigen Kommandos abgebremst. Diese Bremsphase ist aktiv, bis die Bewegung in Richtung des Zielzustands gerichtet ist.

Die nachstehende Abbildung zeigt eine Umkehrbewegung. Die ursprüngliche Zielposition ist 500, das Aborting-Kommando ist mit geringeren Dynamikparametern und einer Zielposition von 300 parametrisiert. Würde die geringeren Dynamiken instantan übernommen, würde zum einen die ursprüngliche Zielposition von 500 überschritten. Zum anderen würde sich die Achse weiter von ihrem neuen Zielzustand entfernen. Bis zur roten senkrechten Markierung sind die höheren Dynamiken des vorherigen Kommandos aktiv. Aus Sicht des neuen Kommandos wird der Überschwinger in Bezug auf Position 300 und unter Beachtung der Kommandodynamiken so gering wie möglich gehalten. Zudem wird die ursprüngliche Zielposition nicht überschritten.



Buffered

Das vorherige Kommando wird bis zum Ende ausgeführt. Der neue Befehl wirkt sich auf das Bewegungsobjekt aus, sobald die vorherige Bewegung abgeschlossen ist (Done / InVelocity / InSync / InGear). Es erfolgt keine Überblendung.



Versionsinformation

- TwinCAT Standard $\geq v3.1.4026.23.1$
- TF5500 MC3 Base $\geq v4.0.6$

5.2.1.1.2 EReactionToMasterError

Reaction type to master errors in master/slave couplings.

Syntax

Definition:

```
TYPE EReactionToMasterError :
(
    TriggerSlaveError      := 0,
    KeepCouplingWithSetValues := 1
) UINT;
END_TYPE
```

Values

Name	Description
TriggerSlaveError	In case of a master error, a slave error is triggered.
KeepCouplingWithSetValues	In case of a master error the coupling is kept with set values and no slave error is triggered.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.2.1.1.3 EReactionToSlaveError

Reaction type to slave errors in master/slave couplings.

Syntax

Definition:

```
TYPE EReactionToSlaveError :
(
    TriggerMasterError := 0,
    NoReaction          := 1
) UINT;
END_TYPE
```

Values

Name	Description
TriggerMasterError	In case of a slave error a master error is triggered.
NoReaction	No reaction on the master in case of a slave error.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.2.1.1.4 ESyncDirection

Directions in which a slave is allowed to move during synchronizing phase.

Syntax

Definition:

```
TYPE ESyncDirection :
(
    Positive      := 1,
    Negative      := 2,
    Both          := 3,

```

```
TowardsSlaveSyncPosition := 4
)UINT;
END_TYPE
```

Values

Name	Description
Positive	Movement is allowed only in positive direction while synchronizing.
Negative	Movement is allowed only in negative direction while synchronizing.
Both	Movement is allowed in any direction while synchronizing.
TowardsSlaveSyncPosition	Movement is allowed only in direction of the SlaveSyncPosition while synchronizing.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.2.2 Interfaces

5.2.2.1 Types

5.2.2.1.1 Type_Camming1D

Type for a motion control object that can operate as a 1D camming slave object (like slave axis for MC_CamIn, e. g. supported by AXIS_REF).

Inheritance Hierarchy

Type_Camming1D

Methods

Name	Origin	Description
<u>GetCamming1DOid</u> [► 229]	Type_Camming1D	

Verwendung des Typen

- Übersicht: MC3-Typen [► 219]

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.2.2.1.1.1 GetCamming1DOid

GetCamming1DOid

OTCID GetCamming1DOid —

Syntax

Definition:

METHOD GetCamming1DOid : OTCID

 Return value

OTCID

5.2.2.1.2 Type_ErrorTriggerable

Type for a motion control object that supports error triggering (e. g. supported by AXIS_REF).

Inheritance Hierarchy

Type_ErrorTriggerable

 Methods

Name	Origin	Description
GetErrorTriggerableOid [► 230]	Type_ErrorTriggerable	

Verwendung des Typen

- [Übersicht: MC3-Typen \[► 219\]](#)

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.2.2.1.2.1 GetErrorTriggerableOid**GetErrorTriggerableOid***OTCID* GetErrorTriggerableOid**Syntax**

Definition:

METHOD GetErrorTriggerableOid : OTCID

 Return value

OTCID

5.2.2.1.3 Type_Master1D

Type for a motion control object that can operate as a 1D master object (like master axis for MC_GearIn, e. g. supportet by AXIS_REF).

Inheritance Hierarchy

Type_Master1D

 Methods

Name	Origin	Description
GetMaster1DOid [► 231]	Type_Master1D	

Verwendung des Typen

- [Übersicht: MC3-Typen](#) [\[▶ 219\]](#)

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.2.2.1.3.1 **GetMaster1DOid**

GetMaster1DOid

OTCID GetMaster1DOid

Syntax

Definition:

```
METHOD GetMaster1DOid : OTCID
```

 Return value

OTCID

5.2.2.1.4 **Type_MotionObject**

Inheritance Hierarchy

Type_MotionObject

 Methods

Name	Origin	Description
GetMotionObjectOid [▶ 231]	Type_MotionObject	

5.2.2.1.4.1 **GetMotionObjectOid**

GetMotionObjectOid

OTCID GetMotionObjectOid

Syntax

Definition:

```
METHOD GetMotionObjectOid : OTCID
```

 Return value

OTCID

5.2.2.1.5 **Type_Phasing1D**

Type for a motion control object that supports 1D phasing (MC_PhasingAbsolute, e. g. supported by AXIS_REF).

Inheritance Hierarchy

Type_Phasing1D

 Methods

Name	Origin	Description
GetPhasing1DOid [► 232]	Type_Phasing1D	

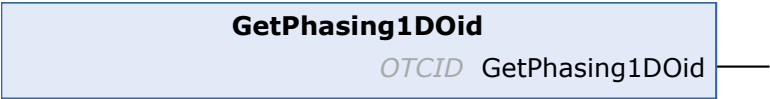
Verwendung des Typen

- [Übersicht: MC3-Typen \[\[► 219\]\(#\)\]](#)

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.2.2.1.5.1 GetPhasing1DOid



Syntax

Definition:

```
METHOD GetPhasing1DOid : OTCID
```

 Return value

OTCID

5.2.3 Function Blocks

5.2.3.1 TestCommands

5.2.3.1.1 MC_TriggerError



This function block triggers an error in the target motion object.

Syntax

Definition:

```
FUNCTION_BLOCK MC_TriggerError
VAR_INPUT
    MotionObject : Type_ErrorTriggerable;
    Execute      : BOOL;
    ErrorCode     : UDINT;
    ControlError : BOOL;
END_VAR
VAR_OUTPUT
    Done : BOOL;
```

```

    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
END_VAR

```

Inputs

Name	Type	Description
MotionObject	Type_ErrorTriggerable [► 230]	Interface pointer to the object on which the error should be triggered (e.g. Axis).
Execute	BOOL	Trigger the command with rising edge.
ErrorCode	UDINT	Error code to be triggered
ControlError	BOOL	Trigger a ControlError (MC has no control over the axis, i.e. hardware or communication fault).

Outputs

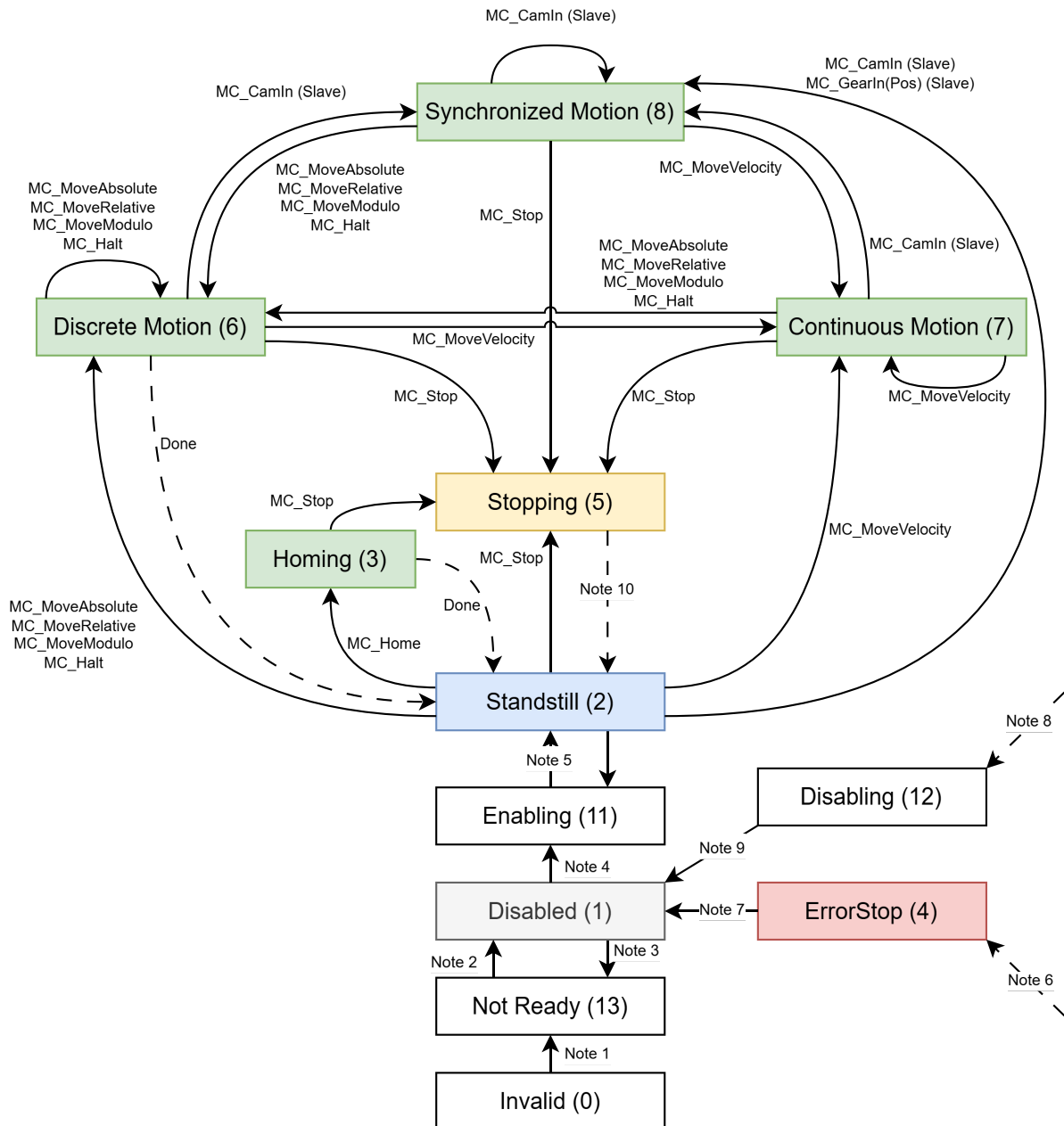
Name	Type	Description
Done	BOOL	Error has been successfully triggered.
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3 Tc3_Mc3Ptp

5.3.1 Zustandsdiagramm



Hinweis	Beschreibung
Note 1	Zustandsübergang während des Startens der XAR und Initialisierung der Achse. Voraussetzung ist eine Verknüpfung zwischen <code>AXIS_REF</code> und des TcCOM-Achsobjekts.
Note 2	Antrieb: <code>WcState = 0</code> , <code>TxPdoState = 0</code> (wenn verfügbar), <code>Position actual value invalid = 0</code> (wenn verfügbar), Statusword signalisiert Bereitschaft für Reglerfreigabe (z. B. <code>ReadyToSwitchOn</code> bei DS402).
Note 3	Der Zustand des Antriebs ändert sich, sodass eine Reglerfreigabe nicht erfolgen kann (z. B. <code>SwitchOnDisabled</code> wenn keine Versorgungsspannung am Leistungsteil).
Note 4	<code>MC_Power.Enable = TRUE</code>
Note 5	<code>MC_Power.Status = TRUE</code> und der Regler des Antriebs ist aktiv (z. B. <code>Operation Enabled</code> bei DS402).

Hinweis	Beschreibung
Note 6	Aus einem beliebigen Zustand, in dem ein Fehler auftritt. Dies betrifft z. B. auch den <code>WcState</code> bzw. <code>Position actual value invalid</code> , wenn diese nach dem Aufstarten einmal valide gemeldet haben und dann diesen Zustand verlieren.
Note 7	<code>MC_Reset</code> (direktes Weiterschalten, wenn <code>MC_Power.Enable = TRUE</code>)
Note 8	Aus einem beliebigen Zustand, wenn <code>MC_Power.Enable = FALSE</code> und die Achse keinen Fehler hat.
Note 9	Der Regler des Antriebs ist nicht mehr aktiv und Statusword meldet Bereitschaft für Reglerfreigabe (z. B. <code>ReadyToSwitchOn</code> bei DS402).
Note 10	<code>MC_Stop.Done = TRUE</code> und <code>MC_Stop.Execute = FALSE</code>

Bewegungskommandos werden grundsätzlich sequenziell abgearbeitet. Alle Kommandos arbeiten in dem beschriebenen Zustandsdiagramm. In der PLC sind die Zustände im Aufzählungstypen `EAxisState` [► 199] enthalten. Die darin hinterlegten numerischen Werte sind an den Zuständen im Diagramm in den Klammern aufgeführt (z. B. „Standstill (2)“).

Die Achse befindet sich immer in einem der definierten Zustände. Jedes Bewegungskommando, das eine Transition verursacht, ändert den Zustand der Achse und damit das Bewegungsprofil. Das Zustandsdiagramm ist eine Abstraktionsebene, die den realen Zustand der Achse, vergleichbar zum Prozessabbild von E/A-Punkten, widerspiegelt. Der Zustand der Achse wechselt sofort mit dem auslösenden Kommando.

Das Zustandsdiagramm bezieht sich auf Einzelachsen. Multi-Achsbausteine, wie `MC_CamIn` oder `MC_GearIn`, beeinflussen die Zustände mehrerer Achsen, die jedoch immer auf einen Einzelachszustand der beteiligten Achsen zurückgeführt werden können. Beispielsweise kann ein Kurvenscheiben-Master im Zustand `Continuous Motion` und der zugehörige Slave im Zustand `Synchronized Motion` sein. Das Ankoppeln eines Slaves hat keinen Einfluss auf den Zustand des Masters.

Der Zustand `Disabled` beschreibt den Grundzustand einer Achse. In diesem Zustand kann die Achse durch keinen Funktionsbaustein bewegt werden, diese ist aber betriebsbereit. Wenn der Funktionsbaustein `MC_Power` mit `Enable = TRUE` aufgerufen wird, wechselt die Achse in den Zustand `Standstill` oder im Fehlerfall in den Zustand `ErrorStop`. Wenn der Funktionsbaustein `MC_Power` mit `Enable = FALSE` aufgerufen wird, wechselt der Zustand nach `Disabled`.

Zweck des Zustands `ErrorStop` ist es, die Achse zu stoppen und anschließend keine weiteren Kommandos anzunehmen, bis ein Reset ausgelöst wurde. Der Zustandsübergang `Error` bezieht sich nur auf tatsächliche Achsfehler und nicht auf Ausführungsfehler eines Funktionsbausteins. Achsfehler können aber auch am Fehlerausgang eines Funktionsbausteins angezeigt werden.

Der Zustand `Stopping` zeigt an, dass sich die Achse in einer Stopprampe befindet. Der Zustand wechselt nach dem vollständigen Stopp zu `Standstill`.

Bewegungskommandos wie `MC_MoveAbsolute`, die aus dem Zustand `Synchronized Motion` herausführen, sind ohne zusätzliche Parametrierung möglich.

Funktionsbausteine, die im Zustandsdiagramm nicht aufgeführt werden, beeinflussen den Zustand der Achse nicht (z. B. `MC_ReadStatus`, `MC_ReadParameter`, `MC_WriteParameter`, ...).

Bedeutung der Zustände

Name	Beschreibung
Invalid	• Ungültiger Achszustand • Die Achse ist nicht betriebsbereit. • Mögliche Ursache kann eine fehlende Verknüpfung der <code>AXIS_REF</code> mit einer MC3-Achse sein.
Disabled	• Grundzustand einer Achse. • Die Achse ist nicht freigegeben. • Es liegt kein Achsfehler vor.
Standstill	• Die Achse ist freigegeben.

Name	Beschreibung
	• Es liegt kein Kommando an der Achse an.
Homing	• Die Achse ist freigegeben. • Ein Homing der Achse wird durchgeführt (z. B. MC_Home).
ErrorStop	• Ein Achsfehler ist aufgetreten. • Ein Reset ist erforderlich, um den Zustand zu verlassen.
Stopping	• Die Achse ist freigegeben. • Die Achse befindet sich in einer Stopprampe (MC_Stop). • MC_Stop.Execute ist gesetzt und blockiert neue Motion Kommandos.
DiscreteMotion	• Die Achse ist freigegeben. • Die Achse führt eine Punkt-zu-Punkt Bewegung aus (z. B. MC_MoveAbsolute, MC_MoveRelative, MC_MoveModulo, MC_Halt).
ContinuousMotion	• Die Achse ist freigegeben. • Die Achse führt eine kontinuierliche Bewegung aus (z. B. MC_MoveVelocity).
SynchronizedMotion	• Die Achse ist freigegeben. • Die Achse ist an eine andere Achse gekoppelt (z. B. MC_CamIn, MC_GearIn, MC_GearInPos).
ResetPending	• Interner Zustand
Enabling	• Erweiterter Achsstatus zur Darstellung von Hardwarezuständen. • Befehlswort zur Freigabe der Endstufe der Antriebshardware ist bereits gesendet oder wird gerade gesendet und die MC3-Achse wartet auf den Zustand "operational" der Antriebshardware
Disabling	• Erweiterter Achsstatus zur Darstellung von Hardwarezuständen. • Befehlswort zur Deaktivierung der Endstufe der Antriebshardware ist bereits gesendet oder wird gerade gesendet und die MC3-Achse wartet auf den Zustand "not operational" der Antriebshardware.
NotReady	• Erweiterter Achsstatus zur Darstellung von Hardwarezuständen. • Die Achse ist nicht betriebsbereit. • Es findet keine aktive Kommunikation zur Hardware statt (z. B. WcState) oder die Zustandsmaschine des Antriebs meldet „not ready“.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.2 Einfaches Verfahren der Achse über die PLC**⚠ GEFAHR****Dieses Beispiel setzt Simulationsachsen voraus – Verletzungsgefahr bei Verwendung von realen Achsen!**

Dieses Beispiel erzeugt eine Achsbewegung. Bei der Verwendung von realen Achsen besteht eine Verletzungsgefahr durch Bewegung von Achsen.

- Achten Sie darauf, dass weder Sie noch andere durch die Bewegung geschädigt werden, z. B. durch die Einhaltung eines geeigneten Sicherheitsabstandes.
- Führen Sie keine Aktion aus, deren Folgen Sie nicht abschätzen können.

Voraussetzungen

- Ein TwinCAT-Projekt mit MC-Projekt wurde erstellt, siehe [MC-Projekt erstellen \[► 17\]](#).

- Eine Simulationsachse wurde im MC-Projekt angelegt, siehe [Achsojekt erstellen](#) [► 25].
- Ein PLC-Projekt wurde im TwinCAT-Projekt angelegt.

PLC Bibliothek einbinden

1. Fügen Sie die Tc3_Mc3Ptp-Bibliothek dem PLC-Projekt hinzu. Hierzu im PLC-Projekt einen Rechtsklick auf **References > Add library...**
 2. Im **Add Library**-Dialog die **Tc3_Mc3Ptp**-Bibliothek auswählen und über **OK** bestätigen.
- ⇒ Die Tc3_Mc3Ptp-Bibliothek wurde dem PLC-Projekt hinzugefügt. Funktionsbausteine und Datentypen der Bibliothek können jetzt verwendet werden.

PLC-Programm schreiben

Alle notwendigen Datentypen und Funktionsbausteine für ein einfaches PLC-Motion-Programm mit der MC3 sind in der Bibliothek Tc3_Mc3Ptp enthalten.

AXIS_REF

Je Achse wird eine Instanz vom Datentyp [AXIS_REF](#) [► 292] benötigt, welche die Schnittstelle zwischen PLC und MC3 ist. Sie enthält alle Informationen zur Achse, die den MC-Funktionsbausteinen als Referenz mitgegeben werden.

MC_Power

Über den Funktionsbaustein [MC_Power](#) [► 253] erfolgt die Freigabe der Achse und deren Bewegungsrichtung.

MC_SetOverride

Über den Funktionsbaustein [MC_SetOverride](#) [► 262] kann der Geschwindigkeits-Override der Achse verändert werden. Standardmäßig ist der Faktor 1 (=100 %) hinterlegt, sodass eine Veränderung des Faktors optional ist.

MC_Reset

Mit dem Funktionsbaustein [MC_Reset](#) [► 261] kann ein Fehler an einer Achse zurückgesetzt werden.

MC_MoveRelative

Mit dem Funktionsbaustein [MC_MoveRelative](#) [► 279] kann die Achse um die definierte Distanz verfahren werden.

Einfache Programmierung

3. Öffnen Sie im PLC-Projekt unter POU's das MAIN(PRG).
4. Fügen Sie folgende Deklarationen ins MAIN(PRG) ein.

```
PROGRAM MAIN
VAR
  axis          : AXIS_REF;
  power         : MC_Power;
  setOverride   : MC_SetOverride;
  moveRelative  : MC_MoveRelative;
  reset         : MC_Reset;
  stop          : MC_Stop;
  enableAxis    : BOOL;
  enableOverride : BOOL;
  executeStop   : BOOL;
  executeReset  : BOOL;
  executeMove   : BOOL;
  velFactor     : LREAL := 1; // adapt override factor if necessary(1 corresponds to 100 %)
  distance      : LREAL := ???; // ToDo: set to a reachable position
```

```

    velocity      : LREAL := ???; // ToDo: set velocity for move
END_VAR

```

5. Fügen Sie folgenden Programmcode ins MAIN(PRG) ein.

```

power(
    Axis      := axis,
    Enable    := enableAxis,
    EnablePositive := enableAxis,
    EnableNegative := enableAxis,
    Status    => ,
    Error     => ,
    ErrorId   => );

setOverride(
    Axis      := axis,
    Enable    := enableOverride,
    VelocityFactor := velFactor,
    Enabled=> ,
    Error=> ,
    ErrorId=> );

stop(
    Axis      := axis,
    Execute   := executeStop, // The command is executed with a positive edge.
    Deceleration := , // If the value is <= 0, the deceleration
                    // parameterized with the last Move command is used.
    Jerk      := , // If the value is <= 0, the jerk parameterized
                    // with the last Move command is used.
    Done      => ,
    Busy      => ,
    Active    => ,
    CommandAborted => ,
    Error     => ,
    ErrorId   => );

reset(
    Axis      := axis,
    Execute   := executeReset, // The command is executed with a positive edge.
    Done      => ,
    Busy      => ,
    Error     => ,
    ErrorId   => );

moveRelative(
    Axis      := axis,
    Execute   := executeMove, // The command is executed with a positive edge.
    Distance  := distance,
    Velocity  := velocity,
    Acceleration := LREAL_CONST.DEFAULT, // If the value is LREAL_CONST.DEFAULT,
                    // the standard acceleration from the axis configuration is used.
    Deceleration := LREAL_CONST.DEFAULT, // If the value is LREAL_CONST.DEFAULT,
                    // the standard deceleration from the axis configuration is used.
    Jerk      := LREAL_CONST.DEFAULT, // If the value is LREAL_CONST.DEFAULT,
                    // the standard jerk from the axis configuration is applied.
    BufferMode  := Tc3_Mc3Base.EBufferMode.Buffered,
    Done      => ,
    Busy      => ,
    Active    => ,
    CommandAborted => ,
    Error     => ,
    ErrorId   => );

```

6. Passen Sie für den Aufruf vom `moveRelative` die Distanz und die Dynamiken entsprechend Ihrer realen Achse an, sollten Sie keine Simulationsachse verwenden.

7. Bauen Sie das PLC-Projekt.

⇒ Die Instanz `axis` des `AXIS_REF` sollte nun unter den PLC-Instanzen im Solution Explorer angezeigt werden.

8. Verknüpfen Sie die PLC-Instanz des `AXIS_REF` mit dem Achsobjekt im MC-Projekt (siehe [Achsobjekt verknüpfen](#) [► 120]).

9. Aktivieren Sie das TwinCAT-Projekt . Loggen Sie die PLC ein und starten Sie diese .

10. Prüfen Sie, dass die beim `moveRelative` die vorgegebene Distanz mit den vorgegebenen Dynamiken angefahren werden kann. Passen Sie diese bei Bedarf an.

11. Wenn Sie sicher sind, dass keine Gefahr durch die beim `moveRelative` vorgegebene Bewegung ausgeht, können Sie der Achse die Reglerfreigabe erteilen, indem Sie die Variable `enableAxis` online auf `TRUE` schreiben.
12. Prüfen Sie, ob die Reglerfreigabe erfolgreich war (`power.Status = TRUE`) und beachten Sie die Hinweise zur Betriebsbereitschaft der Achse [► 254]. Anschließend kann mit `executeMove` das Bewegungskommando aktiviert werden.
 - ⇒ Über `setOverride.Enable := TRUE` kann das kontinuierliche Schreiben des Overrides aktiviert werden. Über `velFactor` wird der Override-Faktor verändert.
 - ⇒ Über `stop.Execute = TRUE` kann bei Bedarf das Bewegungskommando vorzeitig gestoppt werden.
 - ⇒ Im Falle eines Achsfehlers kann dieser über `reset.Execute := TRUE` zurückgesetzt werden.

5.3.3 Data Types

5.3.3.1 Enums

5.3.3.1.1 EAxisParameterId

Parameter IDs for reading and writing axis parameters via `MC_ReadParameter` / `MC_WriteParameter` (`MC_ReadBoolParameter` / `MC_WriteBoolParameter`).

Syntax

Definition:

```

TYPE EAxisParameterId :
(
  MaximumVelocity                := 0x0000000005093009,
  MaximumAcceleration            := 0x000000000509300A,
  MaximumDeceleration            := 0x000000000509300B,
  DefaultAcceleration            := 0x000000000509300C,
  DefaultDeceleration            := 0x000000000509300D,
  DefaultJerk                    := 0x000000000509300E,
  MaximumJerk                    := 0x0000000005093015,
  MinimumPosition                := 0x0000000005093021,
  MaximumPosition                := 0x0000000005093023,
  PositionModulus                := 0x0000000005030107,
  PositionModuloToleranceWindow := 0x0000000005030108,
  MaximumPositionLagValue        := 0x000000000503011B,
  MaximumPositionLagFilterTime   := 0x000000000503011C,
  JogVelocitySlow                := 0x000000000503019E,
  JogVelocityFast                := 0x000000000503019F,
  MinimumPositionLag             := 0x00000000050301A1,
  MaximumPositionLag             := 0x00000000050301A2,
  VelocityOverrideFactor         := 0x0000000005030231,
  SimulationMode                 := 0x00000000050300B1,
  IsModuloAxis                   := 0x0000000005030101,
  PositionLagMonitoringEnabled    := 0x000000000503011A,
  PositionLimitEnforcementEnabled := 0x000000000503015D,
  DriveVelocityScalingNumeratorToHardware := 0x0000000105030122,
  DriveVelocityScalingDenominatorToHardware := 0x00000001050301A7,
  DriveTorqueScalingFromHardware := 0x000000010503014F,
  DriveTimeoutForEnableAndReset  := 0x000000010503D004,
  DriveActiveHomingOffset        := 0x0000000105030224,
  DriveDirectionInverted          := 0x00000001050301B2,
  EncoderPositionScalingNumeratorFromHardware := 0x0000000205030103,
  EncoderPositionScalingDenominatorFromHardware := 0x0000000205030104,
  EncoderMountingOffset          := 0x0000000205030106,
  EncoderWorkZeroOffset          := 0x00000002050301D1,
  EncoderDirectionInverted        := 0x00000002050301B2,
  EncoderIsHomed                 := 0x000000020503D070,
  EncoderIsAbsolutePositionValid := 0x000000020503D071
) ULINT;
END_TYPE

```

Values

Name	Description
MaximumVelocity	Use MC_ReadParameter, MC_WriteParameter
MaximumAcceleration	Use MC_ReadParameter, MC_WriteParameter
MaximumDeceleration	Use MC_ReadParameter, MC_WriteParameter
DefaultAcceleration	Use MC_ReadParameter, MC_WriteParameter
DefaultDeceleration	Use MC_ReadParameter, MC_WriteParameter
DefaultJerk	Use MC_ReadParameter, MC_WriteParameter
MaximumJerk	Use MC_ReadParameter, MC_WriteParameter
MinimumPosition	Use MC_ReadParameter, MC_WriteParameter
MaximumPosition	Use MC_ReadParameter, MC_WriteParameter
PositionModulus	Use MC_ReadParameter, MC_WriteParameter
PositionModuloToleranceWindow	Use MC_ReadParameter, MC_WriteParameter
MaximumPositionLagValue	Use MC_ReadParameter, MC_WriteParameter
MaximumPositionLagFilterTime	Use MC_ReadParameter, MC_WriteParameter
JogVelocitySlow	Use MC_ReadParameter, MC_WriteParameter
JogVelocityFast	Use MC_ReadParameter, MC_WriteParameter
MinimumPositionLag	Use MC_ReadParameter, MC_WriteParameter
MaximumPositionLag	Use MC_ReadParameter, MC_WriteParameter
VelocityOverrideFactor	Use MC_ReadParameter, MC_WriteParameter
SimulationMode	Use MC_ReadBoolParameter, MC_WriteBoolParameter
IsModuloAxis	Use MC_ReadBoolParameter, MC_WriteBoolParameter
PositionLagMonitoringEnabled	Use MC_ReadBoolParameter, MC_WriteBoolParameter
PositionLimitEnforcementEnabled	Use MC_ReadBoolParameter, MC_WriteBoolParameter
DriveVelocityScalingNumeratorToHardware	Use MC_ReadParameter, MC_WriteParameter
DriveVelocityScalingDenominatorToHardware	Use MC_ReadParameter, MC_WriteParameter
DriveTorqueScalingFromHardware	Use MC_ReadParameter, MC_WriteParameter
DriveTimeoutForEnableAndReset	Use MC_ReadParameter, MC_WriteParameter
DriveActiveHomingOffset	Use MC_ReadParameter, MC_WriteParameter
DriveDirectionInverted	Use MC_ReadBoolParameter, MC_WriteBoolParameter
EncoderPositionScalingNumeratorFromHardware	Use MC_ReadParameter, MC_WriteParameter
EncoderPositionScalingDenominatorFromHardware	Use MC_ReadParameter, MC_WriteParameter
EncoderMountingOffset	Use MC_ReadParameter, MC_WriteParameter
EncoderWorkZeroOffset	Use MC_ReadParameter, MC_WriteParameter
EncoderDirectionInverted	Use MC_ReadBoolParameter, MC_WriteBoolParameter
EncoderIsHomed	Use MC_ReadBoolParameter, MC_WriteBoolParameter
EncoderIsAbsolutePositionValid	Use MC_ReadBoolParameter, MC_WriteBoolParameter

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.1.2 ECoeTouchProbeInput

Selection for CoE Drive touch probe input.

Syntax

Definition:

```
TYPE ECoeTouchProbeInput :
(
    DriveDefault      := 0,
    ZeroImpulse       := 1,
    DriveDefined       := 2,
    ConfiguredInAxis  := 255
) UDINT;
END_TYPE
```

Values

Name	Description
DriveDefault	Trigger on the touch probe default input (i.e., the digital input 1/2 mapped to TouchProbe1/TouchProbe2 functionality in the drive).
ZeroImpulse	Trigger on the main encoder index signal (encoder Z/index). Only valid if the drive exposes the index signal to the touch probe function.
DriveDefined	Trigger source is defined by object 0x60D0/0x8001 (DS402/MDP742).
ConfiguredInAxis	The touch probe input source is configured in the axis (TCOM object). The value specified at the CoeDriveSettings input is ignored.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.1.3 EGearInPosSyncState

Synchronization state reported by MC_GearInPos for the PhysicalAxis. Combines whether the slave is synchronized to the master with whether the master has already passed its master sync position. Reports 'Waiting' while the slave cannot move yet due to a sync direction restriction.

Syntax

Definition:

```
TYPE EGearInPosSyncState :
(
    Invalid              := 0,
    Waiting              := 1,
    Synchronizing        := 2,
    SynchronizingAfterMasterSyncPosition := 3,
    InSyncBeforeMasterSyncPosition      := 4,
    InSync                := 5
) UDINT;
END_TYPE
```

Values

Name	Description
Invalid	Default value before the function block has been activated.

Name	Description
Waiting	The slave cannot start to move yet because the master is moving against the restricted sync direction.
Synchronizing	The synchronization is not yet complete and the master has not yet passed its master sync position.
SynchronizingAfterMasterSyncPosition	The synchronization is not yet complete; the master has already passed its master sync position.
InSyncBeforeMasterSyncPosition	The slave is locked in to the master, but the master has not yet passed its master sync position.
InSync	The slave is locked in to the master and the master has passed its master sync position.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.1.4 EHomingMode

Mode selection for functional configuration of MC_SetHomingState command.

Syntax

Definition:

```

TYPE EHomingMode :
(
    ForceIsHomed           := 0,
    ResetIsHomed           := 1,
    ForceIsHomedAtPosition := 2
) UDINT;
END_TYPE

```

Values

Name	Description
ForceIsHomed	Sets the homing state to homed (IsPositionValid = TRUE) without performing any movement or position change.
ResetIsHomed	Resets the homing state to not homed (IsPositionValid = FALSE).
ForceIsHomedAtPosition	Sets the current position to the specified homing position and marks the axis as homed (IsPositionValid = TRUE).

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.1.5 EIncrementalEncoderTouchProbeSource

Selection for incremental encoder touch probe latch source.

Syntax

Definition:

```

TYPE EIncrementalEncoderTouchProbeSource :
(
    LatchC           := 0,
    LatchExternPositive := 1,

```

```

    LatchExternNegative := 2,
    ConfiguredInAxis    := 255
) UDINT;
END_TYPE

```

Values

Name	Description
LatchC	Internal latch (encoder C-track / index signal).
LatchExternPositive	External latch input, capture on rising edge.
LatchExternNegative	External latch input, capture on falling edge.
ConfiguredInAxis	The touch probe latch source is configured in the axis (TCOM object). The value specified at the IncrementalEncoderSettings input is ignored.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.1.6 EJogMode

Selects the jog operation mode.

Syntax

Definition:

```

TYPE EJogMode :
(
    StandardSlow := 1,
    StandardFast := 2,
    Continuous   := 3,
    Inching      := 4
) UDINT;
END_TYPE

```

Values

Name	Description
StandardSlow	Jog at low velocity using the configured slow jog speed.
StandardFast	Jog at high velocity using the configured fast jog speed.
Continuous	Jog continuously at the commanded velocity while the jog input is active.
Inching	Move a fixed incremental distance for each jog activation.

Weiterführende Informationen

Das Enum EJogMode wird in Verbindung mit dem Funktionsbaustein [MC_Jog](#) [► 271] verwendet und dient zur Definition von dessen Betriebsart.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.1.7 EModuloDirection

Direction of travel for modulo positioning.

Syntax

Definition:

```

TYPE EModuloDirection :
(
    ShortestDistance := 0,
    PositiveDirection := 1,
    NegativeDirection := 2
) UDINT;
END_TYPE

```

Values

Name	Description
ShortestDistance	Move in the direction that results in the shortest travel distance.
PositiveDirection	Always move in the positive direction to reach the target.
NegativeDirection	Always move in the negative direction to reach the target.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.1.8 EReadMode

Selects the read operation mode.

Syntax

Definition:

```

TYPE EReadMode :
(
    Once := 1,
    Cyclic := 2
) UDINT;
END_TYPE

```

Values

Name	Description
Once	Read the parameter value once.
Cyclic	Read the parameter value cyclically.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.2 Structs

5.3.3.2.1 AXIS_REF_TOMC

Contains the information of the AXIS_REF passed from PLC to MC.

Syntax

Definition:

```

TYPE AXIS_REF_TOMC :
STRUCT
    Std : CDT_PLCTOMC_AXIS_STD;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Description
Std	CDT_PLCTOMC_AXIS_STD	Standard data from PLC to MC.

Weiterführende Informationen

Die Cyclic Data Types (CDTs) [► 209] sind global definiert und werden in der Data Area [► 33] eines Motion-Objekts konfiguriert.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.2.2 AXIS_REF_TOPLC

Contains the information of the AXIS_REF passed from MC to PLC.

Syntax

Definition:

```

TYPE AXIS_REF_TOPLC :
STRUCT
    Std      : Reference To CDT_MCTOPLC_AXIS_STD;
    Set      : Reference To CDT_MCTOPLC_AXIS_SET;
    Act      : Reference To CDT_MCTOPLC_AXIS_ACT;
    Modulo   : Reference To CDT_MCTOPLC_AXIS_MODULO;
    LoadControl : Reference To CDT_MCTOPLC_AXIS_LOADCONTROL;
    FluidPower : Reference To CDT_MCTOPLC_AXIS_FLUIDPOWER;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Description
Std	Reference To CDT_MCTOPLC_AXIS_STD	Reference to standard data from MC to PLC.
Set	Reference To CDT_MCTOPLC_AXIS_SET	Reference to setpoint data from MC
Act	Reference To CDT_MCTOPLC_AXIS_ACT	Reference to actpoint data from MC
Modulo	Reference To CDT_MCTOPLC_AXIS_MODULO	Reference to modulo positions of a modulo axis
LoadControl	Reference To CDT_MCTOPLC_AXIS_LOADCONTROL	Reference to loadcontrol data from MC
FluidPower	Reference To CDT_MCTOPLC_AXIS_FLUIDPOWER	Reference to fluidpower data from MC

Weiterführende Informationen

Die Cyclic Data Types (CDTs) [► 209] sind global definiert und werden in der Data Area [► 33] eines Motion-Objekts konfiguriert.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.2.3 CoeTouchProbeSettings

CoE drive-specific touch probe source settings.

Syntax

Definition:

```
TYPE CoeTouchProbeSettings :
STRUCT
    TouchProbe1Source : ECoeTouchProbeInput;
    TouchProbe2Source : ECoeTouchProbeInput;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
TouchProbe1Source	ECoeTouchProbeInput [▶ 241]	ConfiguredInAxis	Input source for touch probe 1.
TouchProbe2Source	ECoeTouchProbeInput [▶ 241]	ConfiguredInAxis	Input source for touch probe 2.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.2.4 ENCODER_AXIS_REF_TOPLC

Contains the information of the ENCODER_AXIS_REF passed from MC to PLC.

Syntax

Definition:

```
TYPE ENCODER_AXIS_REF_TOPLC :
STRUCT
    Std      : Reference To CDT_MCTOPLC_ENCODER_AXIS_STD;
    Act      : Reference To CDT_MCTOPLC_ENCODER_AXIS_ACT;
    Modulo   : Reference To CDT_MCTOPLC_ENCODER_AXIS_MODULO;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
Std	Reference To CDT_MCTOPLC_ENCODER_AXIS_STD	Reference to standard data from MC to PLC.
Act	Reference To CDT_MCTOPLC_ENCODER_AXIS_ACT	Reference to actual data from MC.
Modulo	Reference To CDT_MCTOPLC_ENCODER_AXIS_MODULO	Reference to modulo positions from encoder axis.

Weiterführende Informationen

Die Cyclic Data Types (CDTs) [► 209] sind global definiert und werden in der Data Area [► 33] eines Motion-Objekts konfiguriert.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.2.5 FixedStopTouchProbeSettings

Configure up to three thresholds for fixed stop detection (Torque, Velocity, Lag). Enter #Default to use the axis default value, or #Ignore to disable a threshold. Typical detection criteria: combination of high torque, low velocity, and/or increasing lag.

Syntax

Definition:

```
TYPE FixedStopTouchProbeSettings :
STRUCT
    TorqueThreshold    : LREAL;
    VelocityThreshold  : LREAL;
    LagThreshold       : LREAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
TorqueThreshold	LREAL	#Default	Threshold for the actual motor torque. If the torque exceeds this value during evaluation, it contributes to a fixed stop decision. Enter #Ignore to disable this threshold.
VelocityThreshold	LREAL	#Default	Upper limit for the actual axis velocity during evaluation. Fixed stop detection is evaluated only while the absolute velocity is below this value. Enter #Ignore to disable this threshold.
LagThreshold	LREAL	#Default	Threshold for the position following error (command position minus actual position). If the absolute following error exceeds this value during evaluation, it contributes to a fixed stop decision. Enter #Ignore to disable this threshold.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.2.6 IncrementalEncoderTouchProbeSettings

Touch probe settings for incremental encoder terminals.

Syntax

Definition:

```
TYPE IncrementalEncoderTouchProbeSettings :
STRUCT
    Source : EIncrementalEncoderTouchProbeSource;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
Source	EIncrementalEncoderTouchProbeSource ▶ 242	ConfiguredInAxis	Latch source override for position capture.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.2.7 TouchProbeOptions

MC_TouchProbe Options

Syntax

Definition:

```
TYPE TouchProbeOptions :
STRUCT
    WindowFirstPosition : LREAL;
    WindowLastPosition : LREAL;
    PositionsAsModuloValues : BOOL;
    Continuous : BOOL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
WindowFirstPosition	LREAL	#Ignore	Start position from where trigger events are accepted ([PosUnit], included in the position window). It can be left as 'Ignore' if no lower limit is desired. Must have value <= WindowLastPosition otherwise.
WindowLastPosition	LREAL	#Ignore	Stop position up to where trigger events are accepted ([PosUnit], included in the position window). It can be left as 'Ignore' if no upper limit is desired. Must have value >= WindowFirstPosition otherwise.

Name	Type	Default	Description
PositionsAsModuloValues	BOOL	false	Calculate RecordedPosition as modulo value.
Continuous	BOOL	false	Continuous recording mode

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.2.8 TouchProbeUnit

Contains trigger references for positive and negative edge touch probes.

Syntax

Definition:

```

TYPE TouchProbeUnit :
STRUCT
    PositiveTrigger : TRIGGER_REF;
    NegativeTrigger : TRIGGER_REF;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Description
PositiveTrigger	TRIGGER_REF [▶ 249]	Trigger reference for positive edge events.
NegativeTrigger	TRIGGER_REF [▶ 249]	Trigger reference for negative edge events.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.3.2.9 TRIGGER_REF

Syntax

Definition:

```

TYPE TRIGGER_REF :
STRUCT
    Target          : UDINT;
    ProbeId         : UINT;
    Edge            : UDINT;
    ObjId           : PVOID;
    SettingsCoe     : PVOID;
    SettingsFixedStop : PVOID;
    SettingsIncrementalEncoder : PVOID;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Description
Target	UDINT	
Probeld	UINT	
Edge	UDINT	

Name	Type	Description
ObjId	PVOID	
SettingsCoe	PVOID	
SettingsFixedStop	PVOID	
SettingsIncrementalEncoder	PVOID	

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.4 Interfaces

5.3.4.1 Types

5.3.4.1.1 Type_PtpErrorResettable

Type for a motion control object whose errors can be reset (e. g. supported by AXIS_REF).

Inheritance Hierarchy

Type_PtpErrorResettable

Methods

Name	Origin	Description
GetPtpErrorResettableOid [► 250]	Type_PtpErrorResettable	

Verwendung des Typen

- [Übersicht: MC3-Typen](#) [► 219]

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.4.1.1.1 GetPtpErrorResettableOid

GetPtpErrorResettableOid

OTCID GetPtpErrorResettableOid

Syntax

Definition:

```
METHOD GetPtpErrorResettableOid : OTCID
```

Return value

OTCID

5.3.4.1.2 Type_PtpReadWriteParameter

Type for a motion control object that supports parameter read/write access (e.g. supported by AXIS_REF and ENCODER_AXIS_REF).

Inheritance Hierarchy

Type_PtpReadWriteParameter

Methods

Name	Origin	Description
GetPtpReadWriteParameterOid [▶ 251]	Type_PtpReadWriteParameter	

Verwendung des Typen

- [Übersicht: MC3-Typen](#) [[▶ 219](#)]

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.4.1.2.1 GetPtpReadWriteParameterOid

GetPtpReadWriteParameterOid

OTCID GetPtpReadWriteParameterOid

Syntax

Definition:

```
METHOD GetPtpReadWriteParameterOid : OTCID
```

Return value

OTCID

5.3.4.1.3 Type_PtpSetHomingState

Type for a motion control object that supports setting the homing state (e. g. supported by AXIS_REF and ENCODER_AXIS_REF).

Inheritance Hierarchy

Type_PtpSetHomingState

Methods

Name	Origin	Description
GetPtpSetHomingStateOid [▶ 252]	Type_PtpSetHomingState	

Verwendung des Typen

- [Übersicht: MC3-Typen](#) [[▶ 219](#)]

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.4.1.3.1 GetPtpSetHomingStateOid**GetPtpSetHomingStateOid***OTCID* GetPtpSetHomingStateOid**Syntax**

Definition:

METHOD GetPtpSetHomingStateOid : OTCID

 Return value

OTCID

5.3.4.1.4 Type_PtpSetPosition

Type for a motion control object that supports setting position (e. g. supported by AXIS_REF).

Inheritance Hierarchy

Type_PtpSetPosition

 Methods

Name	Origin	Description
<u>GetPtpSetPositionOid</u> [► 252]	Type_PtpSetPosition	

Verwendung des Typen

- Übersicht: MC3-Typen [► 219]

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.4.1.4.1 GetPtpSetPositionOid**GetPtpSetPositionOid***OTCID* GetPtpSetPositionOid**Syntax**

Definition:

METHOD GetPtpSetPositionOid : OTCID

 Return value

OTCID

5.3.4.1.5 Type_PtpTouchProbe

Type for a motion control object that supports touch probe commands (e. g. supported by AXIS_REF and ENCODER_AXIS_REF).

Inheritance Hierarchy

Type_PtpTouchProbe

 Methods

Name	Origin	Description
GetPtpTouchProbeOid [► 253]	Type_PtpTouchProbe	

Verwendung des Typen

- [Übersicht: MC3-Typen](#) [► 219]

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.4.1.5.1 GetPtpTouchProbeOid

GetPtpTouchProbeOid
OTCID GetPtpTouchProbeOid

Syntax

Definition:

```
METHOD GetPtpTouchProbeOid : OTCID
```

 Return value

OTCID

5.3.5 Function Blocks

5.3.5.1 Administrative

5.3.5.1.1 MC_Power

MC_Power

↔ Axis *Reference To AXIS_REF*

Enable *BOOL*

EnablePositive *BOOL*

EnableNegative *BOOL*

BOOL Status

BOOL Error

UDINT ErrorId

This function block controls the power stage (on or off).

Syntax

Definition:

```

FUNCTION_BLOCK MC_Power
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Enable         : BOOL;
    EnablePositive : BOOL;
    EnableNegative : BOOL;
END_VAR
VAR_OUTPUT
    Status         : BOOL;
    Error          : BOOL;
    ErrorId        : UDINT;
END_VAR

```

Inputs

Name	Type	Default	Description
Enable	BOOL		General software enable for the axis
EnablePositive	BOOL	false	Feed enable in positive direction. Only effective if Enable = TRUE
EnableNegative	BOOL	false	Feed enable in negative direction. Only effective if Enable = TRUE

In/Outputs

Name	Type	Description
Axis	Reference To <u>AXIS_REF</u> ID 292	Reference to the axis

Outputs

Name	Type	Description
Status	BOOL	Axis is ready for motion commands.
Error	BOOL	Error occurred within function block or axis already signaling an error.
ErrorId	UDINT	Error identifier of function block or axis

Weiterführende Informationen

Mit dem Funktionsbaustein MC_Power erfolgt die Softwarefreigabe einer Achse. Die Freigabe kann für beide oder nur für eine bestimmte Bewegungsrichtung erfolgen. Durch ein TRUE am Ausgang Status wird die Betriebsbereitschaft der Achse signalisiert.

Abhängig vom Antriebstypen signalisiert Status auch die Betriebsbereitschaft des Antriebs. Insbesondere digitale Antriebe melden die Betriebsbereitschaft zurück, wogegen analog angeschlossene Antriebe ihre Betriebsbereitschaft nicht zurückmelden können. Im letzten Fall signalisiert Status nur die steuerungsseitige Betriebsbereitschaft.

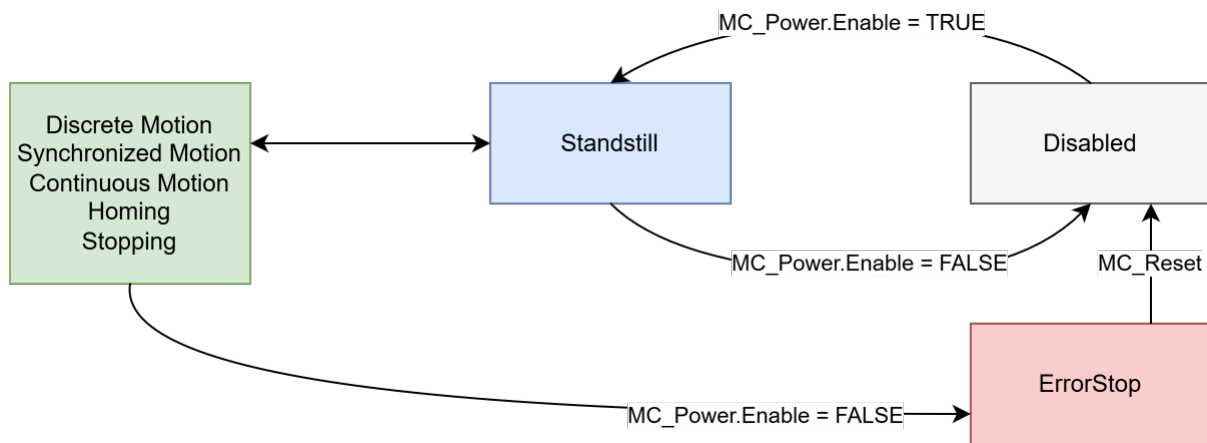


Hardware-Freigabesignal

Zusätzlich zur Software-Freigabe kann es notwendig sein, ein Hardware-Freigabesignal zu schalten, um einen Antrieb freizugeben. Dieses Signal wird nicht durch MC_Power beeinflusst und muss durch die PLC separat geschaltet werden.

Auswirkungen auf den Achszustand

Der MC_Power.Enable wirkt sich folgendermaßen auf den Achszustand aus:



Hinweis zum Umstieg von NC2 zu MC3

Bei den MC3-Achsen ist der Override standardmäßig auf 100% gesetzt. Zum Einstellen des Overrides steht der Funktionsbaustein [MC_SetOverride](#) [► 262] zur Verfügung.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.1.2 MC_ReadActualPosition



This function block reads the current axis actual position.

In case of an invalid actual position value, the Position output is set to 0 and the Valid output is set to FALSE.

Syntax

Definition:

```
FUNCTION_BLOCK MC_ReadActualPosition
VAR_IN_OUT
    Axis      : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Enable    : BOOL;
END_VAR
VAR_OUTPUT
    Valid     : BOOL;
    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
    Position  : LREAL;
END_VAR
```

Inputs

Name	Type	Description
Enable	BOOL	Read the actual position continuously while Enable = TRUE.

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF [► 292]	Reference to the axis

Outputs

Name	Type	Description
Valid	BOOL	Valid output available at function block
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
Position	LREAL	Actual position ([PosUnit])

Weiterführende Informationen

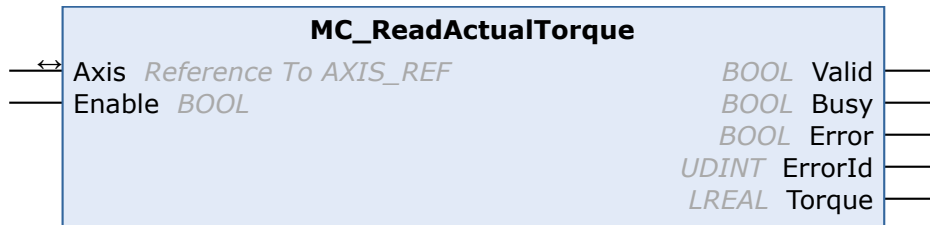
Alternativ kann die Istposition aus der PLC über die `AXIS_REF`-Struktur ausgelesen werden.

```
myAxis.McToPlc.Act.Position
```

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.1.3 MC_ReadActualTorque



This function block reads the current axis actual torque.

In case of an invalid actual torque value, the Torque output is set to 0 and the Valid output is set to FALSE.

Syntax

Definition:

```
FUNCTION_BLOCK MC_ReadActualTorque
VAR_IN_OUT
    Axis      : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Enable    : BOOL;
END_VAR
VAR_OUTPUT
    Valid     : BOOL;
    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
    Torque    : LREAL;
END_VAR
```

🔧 Inputs

Name	Type	Description
Enable	BOOL	Read the actual torque continuously while Enable = TRUE.

🔧 In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF ▶ 292]	Reference to the axis

🔧 Outputs

Name	Type	Description
Valid	BOOL	Valid output available at function block
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
Torque	LREAL	Actual torque ([TorqueUnit])

Weiterführende Informationen

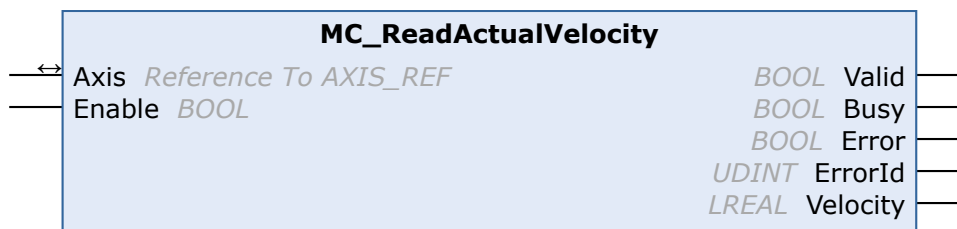
Alternativ kann das Drehmoment aus der PLC über die `AXIS_REF`-Struktur ausgelesen werden.

```
myAxis.McToPlc.Act.Torque
```

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.1.4 MC_ReadActualVelocity



This function block reads the current axis actual velocity.

In case of an invalid actual velocity value, the Velocity output is set to 0 and the Valid output is set to FALSE.

Syntax

Definition:

```
FUNCTION_BLOCK MC_ReadActualVelocity
VAR_IN_OUT
    Axis      : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Enable    : BOOL;
END_VAR
VAR_OUTPUT
    Valid     : BOOL;
    Busy      : BOOL;
    Error     : BOOL;
```

```
ErrorId : UDINT;
Velocity : LREAL;
END_VAR
```

Inputs

Name	Type	Description
Enable	BOOL	Read the actual velocity continuously while Enable = TRUE.

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF [► 292]	Reference to the axis

Outputs

Name	Type	Description
Valid	BOOL	Valid output available at function block
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
Velocity	LREAL	Actual velocity ([PosUnit]/s)

Weiterführende Informationen

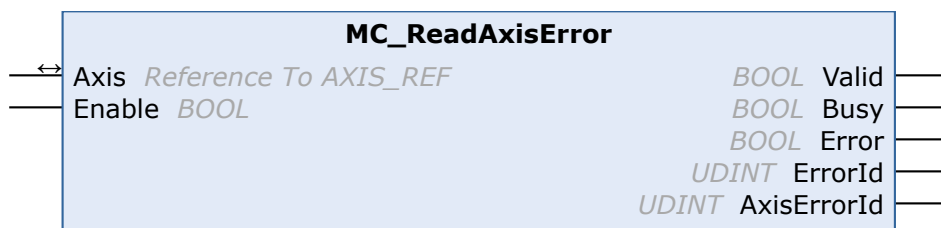
Alternativ kann die Istgeschwindigkeit aus der PLC über die AXIS_REF-Struktur ausgelesen werden.

```
myAxis.McToPlc.Act.Velocity
```

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.1.5 MC_ReadAxisError



This function block reads the current axis error identifier.

Syntax

Definition:

```
FUNCTION_BLOCK MC_ReadAxisError
VAR_IN_OUT
  Axis      : Reference To AXIS_REF;
END_VAR
VAR_INPUT
  Enable     : BOOL;
END_VAR
VAR_OUTPUT
  Valid      : BOOL;
```

```

    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
    AxisErrorId : UDINT;
END_VAR

```

Inputs

Name	Type	Description
Enable	BOOL	Read the axis error identifier continuously while Enable = TRUE.

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF [► 292]	Reference to the axis

Outputs

Name	Type	Description
Valid	BOOL	Valid output available at function block
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
AxisErrorId	UDINT	Axis error identifier

Weiterführende Informationen

Alternativ kann der Achszustand aus der PLC über die [AXIS_REF](#)-Struktur ausgelesen werden.

```
myAxis.McToPlc.Std.AxisState
```

```
myAxis.McToPlc.Std.HasError
```

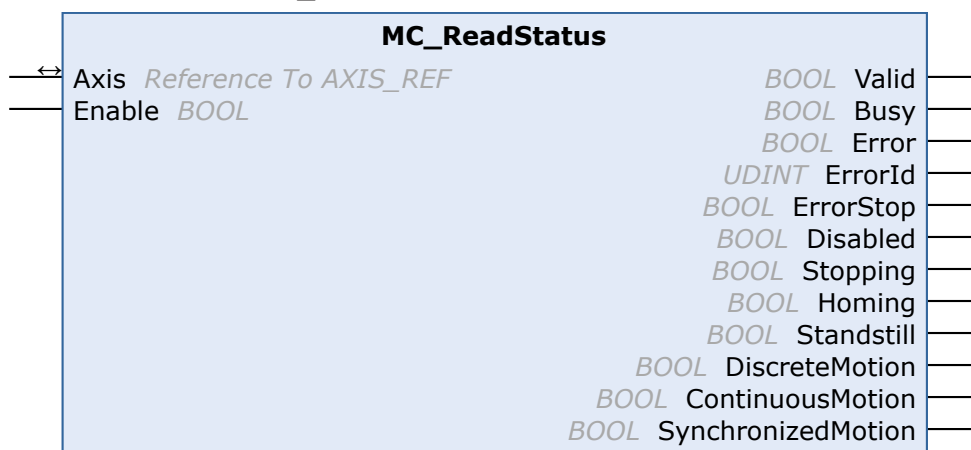
```
myAxis.McToPlc.Std.ErrorId
```

Informationen zu den einzelnen Zuständen sind im Abschnitt [Zustandsdiagramm](#) [► 234] zusammengefasst.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.1.6 MC_ReadStatus



This function block reads the current axis state.

Syntax

Definition:

```
FUNCTION_BLOCK MC_ReadStatus
VAR_IN_OUT
    Axis                : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Enable              : BOOL;
END_VAR
VAR_OUTPUT
    Valid               : BOOL;
    Busy                : BOOL;
    Error               : BOOL;
    ErrorId             : UDINT;
    ErrorStop           : BOOL;
    Disabled            : BOOL;
    Stopping            : BOOL;
    Homing              : BOOL;
    Standstill          : BOOL;
    DiscreteMotion      : BOOL;
    ContinuousMotion    : BOOL;
    SynchronizedMotion : BOOL;
END_VAR
```

Inputs

Name	Type	Description
Enable	BOOL	Read the axis state continuously while Enable = TRUE.

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF [► 292]	Reference to the axis

Outputs

Name	Type	Description
Valid	BOOL	Valid output available at function block
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
ErrorStop	BOOL	Axis is in state ErrorStop.
Disabled	BOOL	Axis is in state Disabled.
Stopping	BOOL	Axis is in state Stopping.
Homing	BOOL	Axis is in state Homing.
Standstill	BOOL	Axis is in state Standstill.
DiscreteMotion	BOOL	Axis is in state DiscreteMotion.
ContinuousMotion	BOOL	Axis is in state ContinuousMotion.
SynchronizedMotion	BOOL	Axis is in state SynchronizedMotion.

Weiterführende Informationen

Alternativ kann der Achszustand aus der PLC über die AXIS_REF-Struktur ausgelesen werden.

```
myAxis.McToPlc.Std.AxisState
```

Informationen zu den einzelnen Zuständen sind im Abschnitt [Zustandsdiagramm](#) [► 234] zusammengefasst.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.1.7 MC_Reset



This function block resets all motion object-related errors and transitions the motion object out of state ErrorStop.

The resulting state is Disabled, if no Enable is requested for the motion object (e.g. via MC_Power).

It results in DiscreteMotion, if the motion object has a closed control loop and a controlled error reaction is still ongoing.

Otherwise, Standstill is the resulting state. This function block does not affect the output of motion function block instances.

Syntax

Definition:

```
FUNCTION_BLOCK MC_Reset
VAR_INPUT
    Axis      : Type_PtpErrorResetable;
    Execute   : BOOL;
END_VAR
VAR_OUTPUT
    Done      : BOOL;
    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
END_VAR
```

Inputs

Name	Type	Description
Axis	Type_PtpErrorResetable [► 250]	Interface pointer to the object on which the command should be performed (e.g. Axis, Linear Mover).
Execute	BOOL	Trigger the command with rising edge.

Outputs

Name	Type	Description
Done	BOOL	Potential axis and/or drive error has been reset. State Standstill, Disabled or DiscreteMotion is reached.
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Hinweise zum Umstieg von NC2 zu MC3

Der MC3 Funktionsbaustein MC_Reset wirkt nur im Fehlerfall.

Der MC3 Baustein MC_Reset setzt einen vorhandenen Schleppfehler der Achse nicht zurück.

Befindet sich auch die verknüpfte Antriebshardware im Fehlerzustand, wird beim MC3 Reset automatisch ein Reset-Kommando an die Hardware gesendet. Ein manueller FB_SoEReset muss nicht ausgeführt werden.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.1.8 MC_SetOverride



This function block sets the velocity override factor that is applied to movement commands of the axis. The override factor does not apply to movement in the synchronized motion state (e.g. movement induced by MC_GearIn, MC_GearInPos, ...).

Syntax

Definition:

```

FUNCTION_BLOCK MC_SetOverride
VAR_IN_OUT
  Axis      : Reference To AXIS_REF;
END_VAR
VAR_INPUT
  Enable     : BOOL;
  VelocityFactor : LREAL;
END_VAR
VAR_OUTPUT
  Enabled    : BOOL;
  Error      : BOOL;
  ErrorId    : UDINT;
END_VAR
  
```

Inputs

Name	Type	Default	Description
Enable	BOOL		Write the velocity override factor continuously while Enable = TRUE.
VelocityFactor	LREAL	1.0	Applies to commanded velocities. Valid range is [0,1]. Use STRING_TO_LREAL('#Ignore') to NOT overwrite the previous value.

In/Outputs

Name	Type	Description
Axis	Reference To <u>AXIS_REF</u> ▶ 292	Reference to the axis

Outputs

Name	Type	Description
Enabled	BOOL	The commanded override is set.
Error	BOOL	Error occurred within function block.

Name	Type	Description
ErrorId	UDINT	Error identifier

Weiterführende Informationen

Die MC3-Achse unterstützt einen Geschwindigkeitsoverride von 0 bis 100%, welcher auf einen Wertebereich von 0 bis 1 normiert ist. Standardmäßig ist der Geschwindigkeitsoverride (*VelocityFactor*) auf 1 gesetzt, was 100% entspricht. Über den Funktionsbaustein MC_SetOverride kann der Geschwindigkeitsoverride innerhalb des zulässigen Wertebereiches angepasst werden.

Der Geschwindigkeitsoverride ermöglicht es, die Geschwindigkeit einer Achse zu beeinflussen ohne die kommandierten Dynamiken (Geschwindigkeit, Beschleunigung und Verzögerung) zu verändern. Dadurch verändert sich auch nicht die zugrunde liegende Bewegungskurve.

In Verbindung mit Kopplungen ist zu beachten, dass nur der Override des Masters berücksichtigt wird.

Hinweis zum Umstieg von NC2 zu MC3

NC2 und MC3 unterstützen einen Geschwindigkeitsoverride von 0 bis 100%.

Bei der NC2 wird der Geschwindigkeitsoverride in Prozent angegeben und ist standardmäßig auf 0% gesetzt.

Hingegen ist bei der MC3 der Wertebereich auf 0 bis 1 normiert und der Standardwert 1, was 100% entspricht.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.1.9 MC_SetPosition



This function block shifts the coordinate system of an axis by manipulating both the setpoint position as well as the actual position of an axis with the same value without any movement caused (Re-calibration with same following error). This can be used for instance for a reference situation.

Syntax

Definition:

```
FUNCTION_BLOCK MC_SetPosition
VAR_INPUT
    Axis          : Type_PtpSetPosition;
    Execute       : BOOL;
    Position      : LREAL;
    Relative      : BOOL;
    ClearPositionLag : BOOL;
END_VAR
VAR_OUTPUT
    Done         : BOOL;
    Busy         : BOOL;
    Error        : BOOL;
    ErrorId      : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
Axis	Type <i>PtpSetPosition</i> [► 252]		Reference to the axis
Execute	BOOL		Trigger the command with rising edge.
Position	LREAL	#Invalid	New position value ([PosUnit]) or distance if Relative = TRUE
Relative	BOOL	false	Shift axis position by value
ClearPositionLag	BOOL		Resets position lag to zero.

Outputs

Name	Type	Description
Done	BOOL	Position has new value.
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Weiterführende Informationen

Mit dem Funktionsbaustein MC_SetPosition wird die aktuelle Achsposition auf einen parametrierbaren Wert gesetzt. Die Achse muss sich hierzu im Stillstand befinden.

Im Standardfall (Eingang *Relative* = FALSE) wird die Istposition auf den parametrisierten absoluten Wert des Eingangs *Position* gesetzt. Im relativen Fall (Eingang *Relative* = TRUE) wird die Istposition um den Wert des Eingangs *Position* relativ verschoben. In beiden Fällen wird die Sollposition der Achse so gesetzt, dass ein eventuell vorhandener Schleppfehler erhalten bleibt. Wenn der Schleppfehler gelöscht werden soll, ist das über den den Eingang *Options.ClearPositionLag* möglich.

Hinweis zum Umstieg von NC2 zu MC3

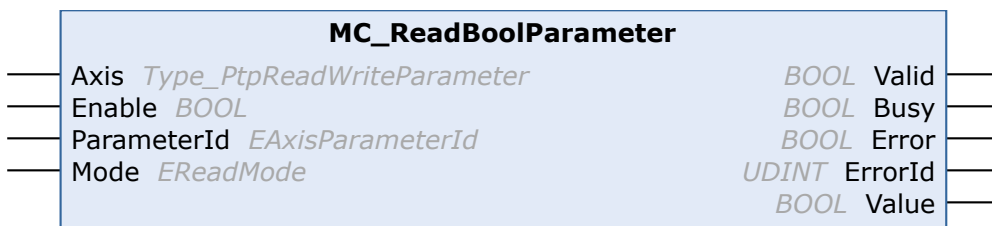
Der Eingang *Relative* entspricht dem Eingang *Mode* des Tc2_MC2.MC_SetPosition Funktionsbausteins.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.2 Administrative/Parameter

5.3.5.2.1 MC_ReadBoolParameter



This function block reads a boolean parameter value from the axis.
The command is typically processed synchronously.

Syntax

Definition:

```

FUNCTION_BLOCK MC_ReadBoolParameter
VAR_INPUT
    Axis      : Type_PtpReadWriteParameter;
    Enable    : BOOL;
    ParameterId : EAxisParameterId;
    Mode      : EReadMode;
END_VAR
VAR_OUTPUT
    Valid     : BOOL;
    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
    Value     : BOOL;
END_VAR

```

Inputs

Name	Type	Default	Description
Axis	Type_PtpReadWriteParameter [► 251]		Reference to the axis
Enable	BOOL		Triggers a read on rising edge (EReadMode.Once) or every cycle while TRUE (EReadMode.Cyclic).
ParameterId	EAxisParameterId [► 239]		Identifier of the parameter to read.
Mode	EReadMode [► 244]	Once	Once: read once at rising edge of Enable. Cyclic: read every cycle while Enable is TRUE.

Outputs

Name	Type	Description
Valid	BOOL	Valid output available at function block.
Busy	BOOL	Function block is not finished. Always FALSE with ReadMode Once.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
Value	BOOL	Last successfully read value of the parameter.

Weiterführende Informationen

Alternativ kann der Achszustand aus der PLC über die AXIS_REF Struktur ausgelesen werden.

```
myAxis.McToPlc.Std.AxisState
```

Informationen zu den einzelnen Zuständen sind im Abschnitt [Zustandsdiagramm \[► 234\]](#) zusammengefasst.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.2.2 MC_ReadHardwareParameter



This function block allows to read parameters from the hardware drive asynchronously via ADS. The function block supports CoE and SoE drives.

Syntax

Definition:

```
FUNCTION_BLOCK MC_ReadHardwareParameter
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    Index         : UINT;
    SubIndex      : USINT;
    DestinationAddress : PVOID;
    DataSize      : UDINT;
END_VAR
VAR_OUTPUT
    Done          : BOOL;
    Busy          : BOOL;
    Active        : BOOL;
    Error         : BOOL;
    ErrorId       : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge.
Index	UINT	0	CoE object index or SoE IDN
SubIndex	USINT	0	CoE sub index or SoE element
DestinationAddress	PVOID	0	Contains the address of the destination buffer.
DataSize	UDINT	0	Contains the size of the destination buffer.

In/Outputs

Name	Type	Description
Axis	Reference To <u>AXIS_REF</u> [► 292]	Reference to the axis

Outputs

Name	Type	Description
Done	BOOL	Value is successfully read from the target.
Busy	BOOL	Function block is not finished.
Active	BOOL	Function block is active.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.2.3 MC_ReadParameter



This function block reads a numeric (LREAL) parameter value from the axis. The command is typically processed synchronously.

Syntax

Definition:

```
FUNCTION_BLOCK MC_ReadParameter
VAR_INPUT
    Axis      : Type_PtpReadWriteParameter;
    Enable    : BOOL;
    ParameterId : EAxisParameterId;
    Mode      : EReadMode;
END_VAR
VAR_OUTPUT
    Valid     : BOOL;
    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
    Value     : LREAL;
END_VAR
```

Inputs

Name	Type	Default	Description
Axis	Type_PtpReadWriteParameter [► 251]		Reference to the axis
Enable	BOOL		Triggers a read on rising edge (EReadMode.Once) or every cycle while TRUE (EReadMode.Cyclic).
ParameterId	EAxisParameterId [► 239]		Identifier of the parameter to read.
Mode	EReadMode [► 244]	Once	Once: read once at rising edge of Enable. Cyclic: read every cycle while Enable is TRUE.

Outputs

Name	Type	Description
Valid	BOOL	Valid output available at function block.
Busy	BOOL	Function block is not finished. Always FALSE with ReadMode Once.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
Value	LREAL	Last successfully read value of the parameter.

Weiterführende Informationen

Alternativ kann der Achszustand aus der PLC über die `AXIS_REF` Struktur ausgelesen werden.

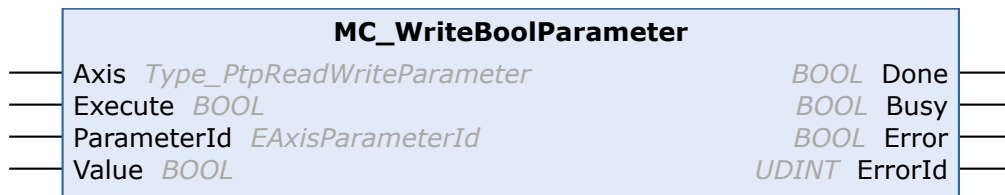
```
myAxis.McToPlc.Std.AxisState
```

Informationen zu den einzelnen Zuständen sind im Abschnitt [Zustandsdiagramm \[► 234\]](#) zusammengefasst.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.2.4 MC_WriteBoolParameter



This function block writes a boolean parameter value to the axis.
The command is typically processed synchronously.

Syntax

Definition:

```
FUNCTION_BLOCK MC_WriteBoolParameter
VAR_INPUT
    Axis      : Type_PtpReadWriteParameter;
    Execute   : BOOL;
    ParameterId : EAxisParameterId;
    Value     : BOOL;
END_VAR
VAR_OUTPUT
    Done      : BOOL;
    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
END_VAR
```

Inputs

Name	Type	Description
Axis	Type_PtpReadWriteParameter [► 251]	Reference to the axis
Execute	BOOL	Trigger the command with rising edge.
ParameterId	EAxisParameterId [► 239]	Identifier of the parameter to write.
Value	BOOL	Value of the parameter to write.

Outputs

Name	Type	Description
Done	BOOL	Parameter value has been successfully written.
Busy	BOOL	Function block is not finished.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.2.5 MC_WriteHardwareParameter

This function block allows to write parameters to the hardware drive asynchronously via ADS. It supports CoE and SoE drives.

Syntax

Definition:

```

FUNCTION_BLOCK MC_WriteHardwareParameter
VAR_IN_OUT
    Axis      : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute    : BOOL;
    Index      : UINT;
    SubIndex   : USINT;
    SourceAddress : PVOID;
    DataSize   : UDINT;
END_VAR
VAR_OUTPUT
    Done       : BOOL;
    Busy       : BOOL;
    Active     : BOOL;
    Error      : BOOL;
    ErrorId    : UDINT;
END_VAR

```

Inputs

Name	Type	Default	Description
Execute	BOOL		Rising edge triggers the write operation.
Index	UINT	0	CoE object index or SoE IDN
SubIndex	USINT	0	CoE sub index or SoE element
SourceAddress	PVOID	0	Contains the address of the data.
DataSize	UDINT	0	Contains the size of the data.

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF ▶ 292	Reference to the axis

Outputs

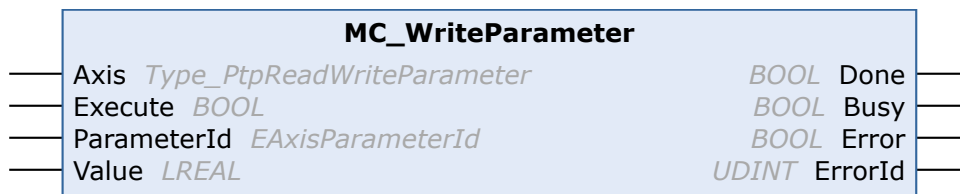
Name	Type	Description
Done	BOOL	Value is successfully sent to the target.
Busy	BOOL	Function block is not finished.

Name	Type	Description
Active	BOOL	Function block is active.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.2.6 MC_WriteParameter



This function block writes a numeric (LREAL) parameter value to the axis. The command is typically processed synchronously.

Syntax

Definition:

```

FUNCTION_BLOCK MC_WriteParameter
VAR_INPUT
    Axis      : Type_PtpReadWriteParameter;
    Execute   : BOOL;
    ParameterId : EAxisParameterId;
    Value     : LREAL;
END_VAR
VAR_OUTPUT
    Done      : BOOL;
    Busy      : BOOL;
    Error     : BOOL;
    ErrorId   : UDINT;
END_VAR
  
```

Inputs

Name	Type	Description
Axis	Type_PtpReadWriteParameter [► 251]	Reference to the axis
Execute	BOOL	Trigger the command with rising edge.
ParameterId	EAxisParameterId [► 239]	Identifier of the parameter to write.
Value	LREAL	Value of the parameter to write.

Outputs

Name	Type	Description
Done	BOOL	Parameter value has been successfully written.
Busy	BOOL	Function block is not finished.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

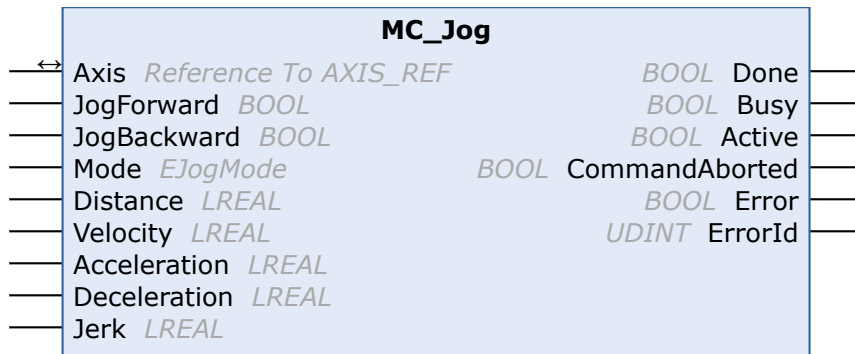
Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1

- TF5500 MC3 Base >= v4.0.6

5.3.5.3 Continuous Motion

5.3.5.3.1 MC_Jog



This function block enables an axis to be moved via manual keys in different modi.

Syntax

Definition:

```
FUNCTION_BLOCK MC_Jog
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    JogForward     : BOOL;
    JogBackward    : BOOL;
    Mode           : EJogMode;
    Distance        : LREAL;
    Velocity        : LREAL;
    Acceleration    : LREAL;
    Deceleration    : LREAL;
    Jerk           : LREAL;
END_VAR
VAR_OUTPUT
    Done           : BOOL;
    Busy           : BOOL;
    Active         : BOOL;
    CommandAborted : BOOL;
    Error          : BOOL;
    ErrorId        : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
JogForward	BOOL		The command is executed with a rising edge, and the axis is moved in positive direction. Depending on the mode, the axis moves as long as the signal remains TRUE, or it stops automatically after a specified distance.
JogBackward	BOOL		The command is executed with a rising edge, and the axis is moved in negative direction. Depending on the mode, the axis moves as long as the signal remains TRUE, or it stops automatically after a specified distance.
Mode	<u>EJogMode</u> [▶ 243]	StandardSlow	Determines the operation mode in which the manual function is executed.
Distance	LREAL	#Ignore	Distance for movements in Mode Inching

Name	Type	Default	Description
Velocity	LREAL	#Ignore	Jog velocity ([PosUnit]/s, positive). Required for Continuous and Inching modes; StandardSlow and StandardFast use axis defaults.
Acceleration	LREAL	#Default	Maximum acceleration ([PosUnit]/s^2, positive)
Deceleration	LREAL	#Default	Maximum deceleration ([PosUnit]/s^2, positive)
Jerk	LREAL	#Default	Maximum jerk ([PosUnit]/s^3, positive)

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF [► 292]	Reference to the axis

Outputs

Name	Type	Description
Done	BOOL	Function block has succeeded.
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Weiterführende Informationen

Die Betriebsart des Funktionsbausteins MC_Jog wird über den Eingang `Mode` vom Typ [E_JogMode](#) [\[► 243\]](#) definiert:

`E_JogMode.StandardSlow`

Die Achse wird so lange verfahren, wie das Signal an einem der Eingänge `JogForward` oder `JogBackward` `TRUE` ist. Dabei wird die im Achsparameter `Jog Velocity Slow` festgelegte Geschwindigkeit sowie die Dynamik entsprechend der Eingänge `Acceleration`, `Deceleration` und `Jerk` verwendet.

Die Funktionsbaustein-Eingänge `Distance` und `Velocity` haben in dieser Betriebsart keinen Einfluss.

`E_JogMode.StandardFast`

Die Achse wird so lange verfahren, wie das Signal an einem der Eingänge `JogForward` oder `JogBackward` `TRUE` ist. Dabei wird die im Achsparameter `Jog Velocity Fast` festgelegte Geschwindigkeit sowie die Dynamik entsprechend der Eingänge `Acceleration`, `Deceleration` und `Jerk` verwendet.

Die Funktionsbaustein-Eingänge `Distance` und `Velocity` haben in dieser Betriebsart keinen Einfluss.

`E_JogMode.Continuous`

Die Achse wird so lange verfahren, wie das Signal an einem der Eingänge `JogForward` oder `JogBackward` `TRUE` ist.

Der Funktionsbaustein-Eingang `Distance` hat in dieser Betriebsart keinen Einfluss.

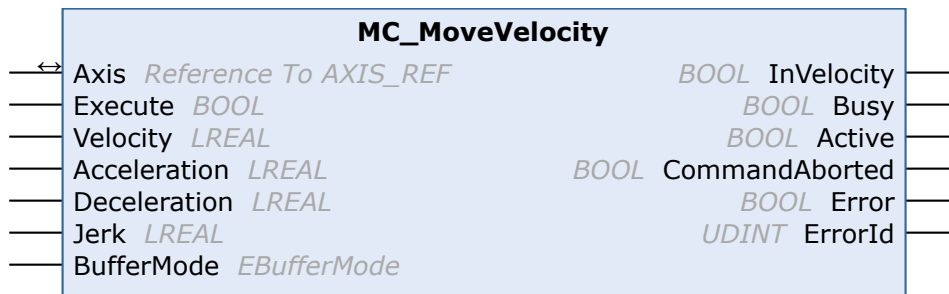
E_JogMode.Inching

Die Achse wird um die am Funktionsbaustein-Eingang *Distance* festgelegte Strecke verfahren, sobald am Eingang *JogForward* oder *JogBackward* eine steigende Flanke detektiert wird.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.3.2 MC_MoveVelocity



This function block commands an endless movement with a specified velocity.

Syntax

Definition:

```
FUNCTION_BLOCK MC_MoveVelocity
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    Velocity      : LREAL;
    Acceleration  : LREAL;
    Deceleration  : LREAL;
    Jerk          : LREAL;
    BufferMode     : EBufferMode;
END_VAR
VAR_OUTPUT
    InVelocity    : BOOL;
    Busy          : BOOL;
    Active        : BOOL;
    CommandAborted : BOOL;
    Error         : BOOL;
    ErrorId       : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge.
Velocity	LREAL	#Invalid	Commanded velocity ([PosUnit]/s, the sign of this parameter determines the direction of movement).
Acceleration	LREAL	#Default	Maximum acceleration ([PosUnit]/s^2, positive)
Deceleration	LREAL	#Default	Maximum deceleration ([PosUnit]/s^2, positive)
Jerk	LREAL	#Default	Maximum jerk ([PosUnit]/s^3, positive)
BufferMode	EBufferMode [► 226]	Aborting	Chronological sequence of the command

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF ▶ 292	Reference to the axis

Outputs

Name	Type	Description
InVelocity	BOOL	Commanded velocity reached
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Weiterführende Informationen

Mit dem Funktionsbaustein MC_MoveVelocity wird eine Endlosfahrt mit vorgegebener Geschwindigkeit und Richtung gestartet. Sobald die vorgegebene konstante Geschwindigkeit erreicht ist, wird der Ausgang `InVelocity` gesetzt und die Funktion des Funktionsbausteins ist abgeschlossen. Es erfolgt keine weitere Überwachung der Bewegung durch den Funktionsbaustein. Wird das Kommando während der Beschleunigungsphase abgebrochen wird der Ausgang `CommandAborted` und im Fehlerfall der Ausgang `Error` gesetzt.

Die Bewegung kann mit einem MC Halt [▶ 274](#) und MC_Stop [▶ 280](#) angehalten oder durch ein anderes Bewegungskommando abgelöst werden.

Wird ein Bewegungskommando mit dem Eingang `BufferMode = Tc3_Mc3Base.EBufferMode.Aborting` auf eine gekoppelte Slave-Achse angewendet, so führt dieses zum automatischen Abkoppeln der Achse und anschließendem Ausführen des Kommandos.

Hinweis zum Umstieg von NC2 zu MC3

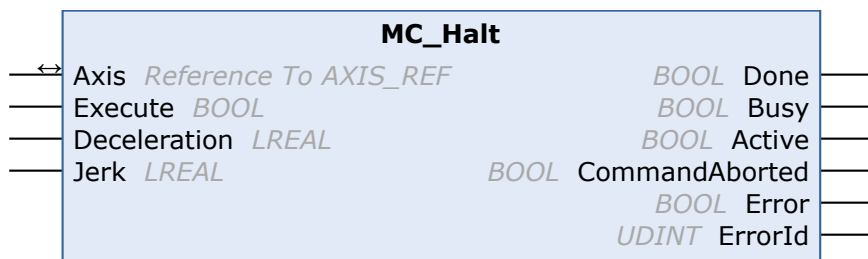
Der MC3 Funktionsbaustein MC_MoveVelocity hat im Gegensatz zum gleichnamigen NC2 Funktionsbaustein einen vorzeichenbehafteten `Velocity`-Eingang. Durch das Vorzeichen wird die Bewegungsrichtung vorgegeben.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.4 Discrete Motion

5.3.5.4.1 MC_Halt



This function block commands a motion stop. The axis is transferred to the state DiscreteMotion, until zero dynamics is reached.

Syntax

Definition:

```
FUNCTION_BLOCK MC_Halt
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    Deceleration  : LREAL;
    Jerk          : LREAL;
END_VAR
VAR_OUTPUT
    Done          : BOOL;
    Busy          : BOOL;
    Active        : BOOL;
    CommandAborted : BOOL;
    Error         : BOOL;
    ErrorId       : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge.
Deceleration	LREAL	#Default	Maximum deceleration ([PosUnit]/s ² , positive)
Jerk	LREAL	#Default	Maximum jerk ([PosUnit]/s ³ , positive)

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF ▶ 292	Reference to the axis

Outputs

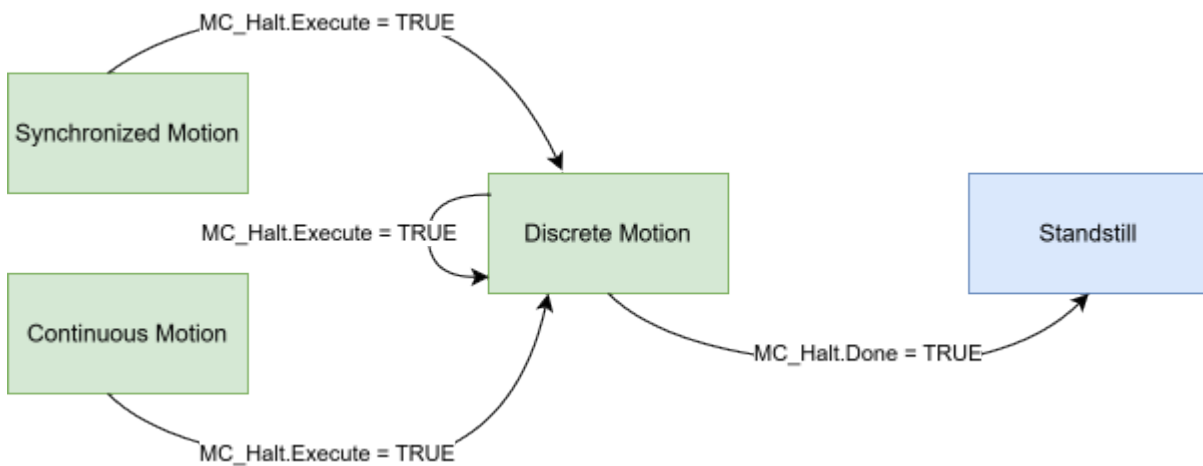
Name	Type	Description
Done	BOOL	Zero dynamics has been reached, axis state is transferred to Standstill.
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Weiterführende Informationen

Mit dem Funktionsbaustein MC_Halt wird eine Achse mit einer definierten Bremsrampe angehalten.

Im Gegensatz zum MC_Stop [▶ 280](#) wird die Achse nicht gegen weitere Bewegungskommandos verriegelt. So kann das MC_Halt-Kommando sowohl während der Bewegungsrampe als auch im Anschluss durch ein anderes Bewegungskommando abgelöst werden.

Wird MC_Halt auf eine gekoppelte Slave-Achse angewendet, so führt dieses zum automatischen Abkoppeln der Achse und anschließendem Ausführen des Halt-Kommandos.

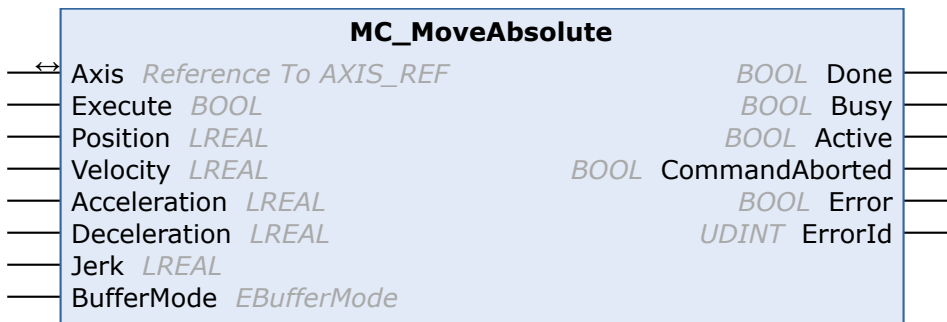
Auswirkungen auf den Achszustand**Hinweis zum Umstieg von NC2 zu MC3**

Der MC3 Funktionsbaustein MC_Halt hat im Gegensatz zum gleichnamigen NC2 Funktionsbaustein keinen Eingang, an dem der Buffermodus vorgegeben werden kann. Ein vorhergehendes Kommando wird bei der MC3 immer direkt vom MC_Halt abgelöst, was intern dem Buffermodus Aborting entspricht.

In der MC3 ist eine Parametrierung der Dynamiken mit dem Wert 0 (implizite Verwendung der zuletzt kommandierten Dynamiken) nicht möglich.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.4.2 MC_MoveAbsolute

This function block commands a motion to a specified absolute position.

Syntax**Definition:**

```

FUNCTION_BLOCK MC_MoveAbsolute
VAR_IN_OUT
  Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
  Execute       : BOOL;
  Position      : LREAL;
  Velocity      : LREAL;
  Acceleration  : LREAL;
  Deceleration  : LREAL;
  Jerk          : LREAL;
  BufferMode     : EBufferMode;
END_VAR
VAR_OUTPUT
  Done          : BOOL;
  Busy          : BOOL;

```

```

Active      : BOOL;
CommandAborted : BOOL;
Error       : BOOL;
ErrorId     : UDINT;
END_VAR

```

Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge.
Position	LREAL	#Invalid	Commanded position ([PosUnit])
Velocity	LREAL	#Invalid	Maximum velocity ([PosUnit]/s, positive)
Acceleration	LREAL	#Default	Maximum acceleration ([PosUnit]/s^2, positive)
Deceleration	LREAL	#Default	Maximum deceleration ([PosUnit]/s^2, positive)
Jerk	LREAL	#Default	Maximum jerk ([PosUnit]/s^3, positive)
BufferMode	EBufferMode [► 226]	Aborting	Chronological sequence of the command

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF [► 292]	Reference to the axis

Outputs

Name	Type	Description
Done	BOOL	Commanded position finally reached
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Weiterführende Informationen

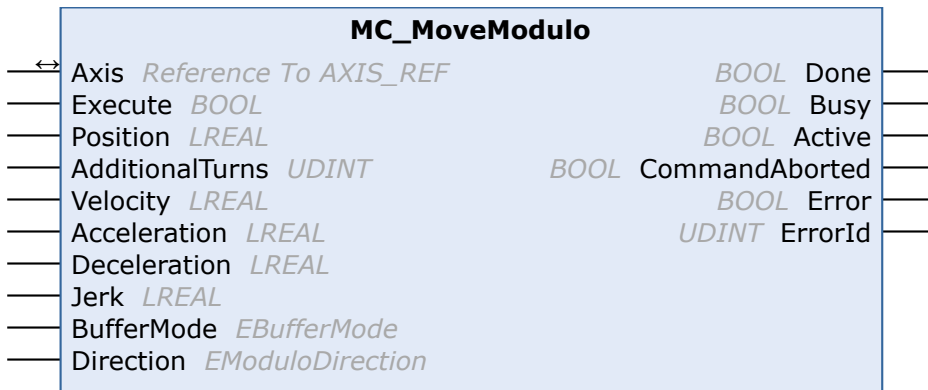
Mit dem Funktionsbaustein MC_MoveAbsolute wird eine Positionierung auf eine absolute Zielposition gestartet und die Achsbewegung über den gesamten Fahrweg überwacht.

Wird ein Bewegungskommando mit dem Eingang `BufferMode = Tc3_Mc3Base.EBufferMode.Aborting` auf eine gekoppelte Slave-Achse angewendet, so führt dieses zum automatischen Abkoppeln der Achse und anschließendem Ausführen des Kommandos.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.4.3 MC_MoveModulo



This function block commands a motion to a specified modulo position.

Syntax

Definition:

```
FUNCTION_BLOCK MC_MoveModulo
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    Position      : LREAL;
    AdditionalTurns : UDINT;
    Velocity      : LREAL;
    Acceleration  : LREAL;
    Deceleration  : LREAL;
    Jerk          : LREAL;
    BufferMode     : EBufferMode;
    Direction     : EModuloDirection;
END_VAR
VAR_OUTPUT
    Done          : BOOL;
    Busy          : BOOL;
    Active        : BOOL;
    CommandAborted : BOOL;
    Error         : BOOL;
    ErrorId       : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge.
Position	LREAL	#Invalid	Commanded modulo position ([PosUnit])
AdditionalTurns	UDINT	0	Number of additional modulo turns in the commanded direction. A value > 0 requires Direction <> EModuloDirection.ShortestDistance.
Velocity	LREAL	#Invalid	Maximum velocity ([PosUnit]/s, positive)
Acceleration	LREAL	#Default	Maximum acceleration ([PosUnit]/s^2, positive)
Deceleration	LREAL	#Default	Maximum deceleration ([PosUnit]/s^2, positive)
Jerk	LREAL	#Default	Maximum jerk ([PosUnit]/s^3, positive)
BufferMode	EBufferMode [► 226]	Aborting	Chronological sequence of the command
Direction	EModuloDirection [► 243]	ShortestDistance	Commanded direction. Must have value <> EModuloDirection.ShortestDistance if AdditionalTurns has value > 0.

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF [► 292]	Reference to the axis

Outputs

Name	Type	Description
Done	BOOL	Commanded position finally reached
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Weiterführende Informationen

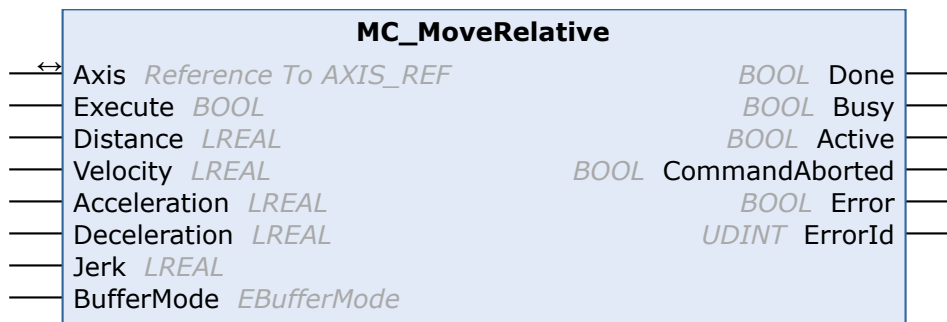
Ist der Eingang Direction auf PositiveDirection oder NegativeDirection gesetzt, so wird auch der Eingang AdditionalTurns berücksichtigt. Über den Eingang AdditionalTurns können zusätzliche Umdrehungen bis zum Erreichen der Zielposition vorgegeben werden.

Weiterführende Informationen zur [Modulo-Positionierung](#) [\[► 309\]](#).

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.4.4 MC_MoveRelative



This function block commands a motion of a specified distance relative to the set position at the time the command gets active.

Syntax

Definition:

```
FUNCTION_BLOCK MC_MoveRelative
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    Distance      : LREAL;
    Velocity      : LREAL;
    Acceleration  : LREAL;
    Deceleration  : LREAL;
    Jerk          : LREAL;
```

```

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

```

Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge.
Distance	LREAL	#Invalid	Relative distance ([PosUnit])
Velocity	LREAL	#Invalid	Maximum velocity ([PosUnit]/s, positive)
Acceleration	LREAL	#Default	Maximum acceleration ([PosUnit]/s^2, positive)
Deceleration	LREAL	#Default	Maximum deceleration ([PosUnit]/s^2, positive)
Jerk	LREAL	#Default	Maximum jerk ([PosUnit]/s^3, positive)
BufferMode	EBufferMode [► 226]	Aborting	Chronological sequence of the command

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF [► 292]	Reference to the axis

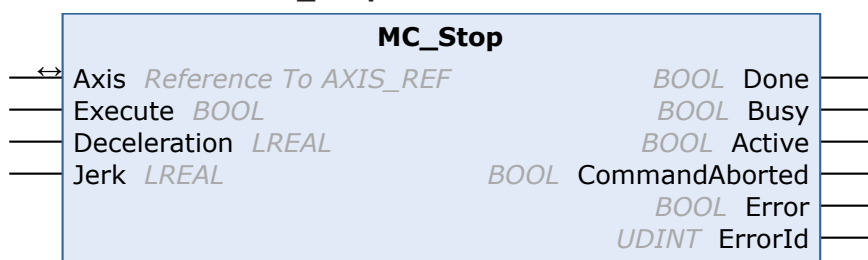
Outputs

Name	Type	Description
Done	BOOL	Commanded distance reached
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.4.5 MC_Stop



This function block commands a motion stop and transfers the axis to the state Stopping. It aborts any ongoing function block execution. While the axis is in state Stopping, no other function block can perform any motion on the same axis. The Done output is set immediately after zero dynamics is reached. The axis remains in the state Stopping as long as Execute is TRUE or zero dynamics is not yet reached. As soon as Done is SET and Execute is FALSE the axis goes to state Standstill.

Syntax

Definition:

```
FUNCTION_BLOCK MC_Stop
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    Deceleration  : LREAL;
    Jerk          : LREAL;
END_VAR
VAR_OUTPUT
    Done          : BOOL;
    Busy          : BOOL;
    Active        : BOOL;
    CommandAborted : BOOL;
    Error         : BOOL;
    ErrorId       : UDINT;
END_VAR
```

🔧 Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge. While this is TRUE, other function blocks will be rejected for the axis.
Deceleration	LREAL	#Default	Maximum deceleration ([PosUnit]/s ² , positive)
Jerk	LREAL	#Default	Maximum jerk ([PosUnit]/s ³ , positive)

🔧 In/Outputs

Name	Type	Description
Axis	Reference To <u>AXIS_REF</u> [▶ 292]	Reference to the axis

🔧 Outputs

Name	Type	Description
Done	BOOL	Zero dynamics has been reached.
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Weiterführende Informationen

Mit dem Funktionsbaustein MC_Stop wird eine Achse mit einer definierten Bremsrampe angehalten und die Achse gegen andere Bewegungskommandos verriegelt. Erst wenn sich die Achse im Stillstand befindet (`MC_Stop.Done = TRUE`) und der Eingang `MC_Stop.Execute = FALSE` ist, wird die Verriegelung der Achse bzgl. anderer Bewegungskommandos aufgehoben.

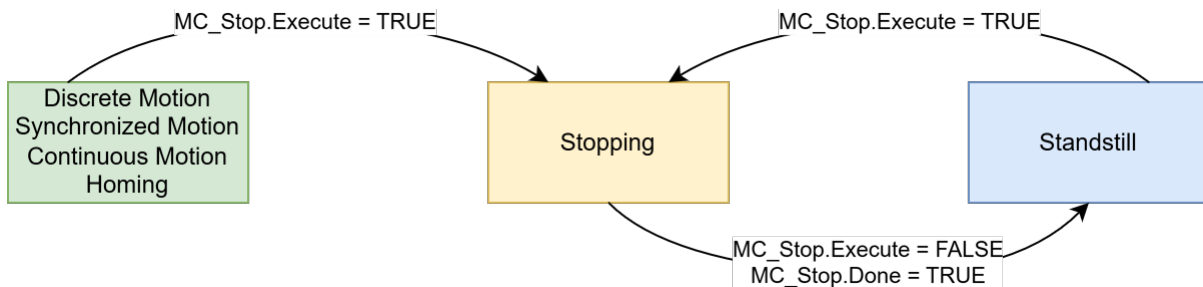
Für die Bremsrampe werden die maximalen Dynamiken von Deceleration und Jerk aus vorherigem Kommando und den am Baustein angegebenen Werten gewählt.

Alternative kann eine Achse mit MC_Halt [► 274] ohne Verriegelung angehalten werden.

Wird MC_Stop auf eine gekoppelte Slave-Achse angewendet, so führt dieses zum automatischen Abkoppeln der Achse und anschließendem Ausführen des Stop-Kommandos.

Auswirkungen auf den Achszustand

Der Funktionsbaustein MC_Stop kann in jedem Bewegungszustand der Achse aufgerufen werden. Die Achse wird dadurch in den Zustand *Stopping* versetzt. Neben dem Anhalten der Achse mit einer definierten Bremsrampe erfolgt im *Stopping*-Zustand eine Verriegelung der Achse gegen andere Bewegungskommandos. Daher ist es auch möglich MC_Stop im Zustand *Standstill* aufzurufen, um eine Verriegelung der Achse zu erzeugen. Erst wenn keine Bewegung der Achse erfolgt und der Eingang `MC_Stop.Execute = FALSE` gesetzt ist, wird die Achse in den Zustand *Standstill* versetzt und dadurch die Verriegelung der Achse wieder aufgehoben.



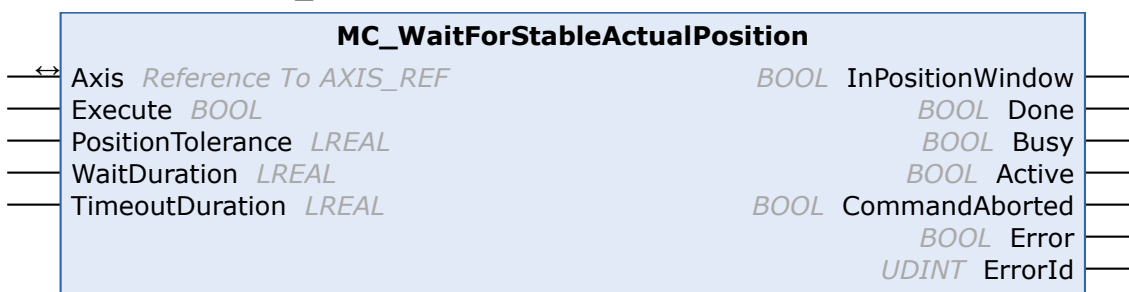
Hinweis zum Umstieg von NC2 zu MC3

In der MC3 ist eine Parametrierung der Dynamiken mit dem Wert 0 (implizite Verwendung der zuletzt kommandierten Dynamiken) nicht möglich.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.4.6 MC_WaitForStableActualPosition



This function block waits for a stable actual position. The command is always buffered and may only be triggered when the previous command ends with zero dynamics.

Syntax

Definition:

```

FUNCTION_BLOCK MC_WaitForStableActualPosition
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    PositionTolerance : LREAL;
    WaitDuration  : LREAL;
    TimeoutDuration : LREAL;
END_VAR
VAR_OUTPUT
    InPositionWindow : BOOL;
    Done             : BOOL;
    Busy            : BOOL;
    Active          : BOOL;
    CommandAborted  : BOOL;
    Error           : BOOL;
    ErrorId         : UDINT;
END_VAR

```

Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge.
PositionTolerance	LREAL	#Invalid	Position tolerance around the set position ([PosUnit], positive). The actual position must remain within this window for the command to succeed.
WaitDuration	LREAL	#Invalid	Duration (in seconds) that the actual position needs to consecutively stay inside the position window. Must have a value >= 0.
TimeoutDuration	LREAL	#Ignore	Duration (in seconds) after which an error is triggered if the actual position is not inside the position window. It can be left as 'Ignore' if no timeout is desired, otherwise must have a value > WaitDuration.

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF [► 292]	Reference to the axis

Outputs

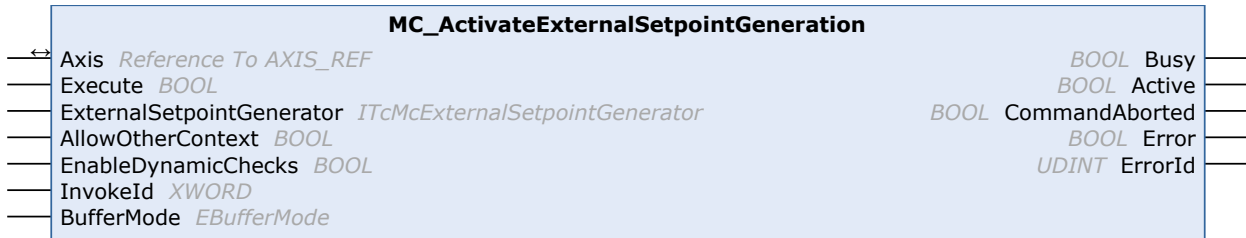
Name	Type	Description
InPositionWindow	BOOL	Actual position is currently inside the position window.
Done	BOOL	Actual position has been consecutively inside the position window for at least WaitDuration.
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis. While Active is TRUE, the set position remains on the final position of the previous command and the set dynamics remain zero.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.5 External Setpoint Generator

5.3.5.5.1 MC_ActivateExternalSetpointGeneration



This function block activates external setpoint generation. Setpoints are generated by a function block, which is called through the ITcMcExternalSetpointGenerator interface from the MC context. Additionally custom startup data can be passed through the generator.

Syntax

Definition:

```
FUNCTION_BLOCK MC_ActivateExternalSetpointGeneration
VAR_IN_OUT
    Axis                : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute              : BOOL;
    ExternalSetpointGenerator : ITcMcExternalSetpointGenerator;
    AllowOtherContext    : BOOL;
    EnableDynamicChecks  : BOOL;
    InvokeId             : XWORD;
    BufferMode            : EBufferMode;
END_VAR
VAR_OUTPUT
    Busy                 : BOOL;
    Active               : BOOL;
    CommandAborted       : BOOL;
    Error                : BOOL;
    ErrorId              : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge.
ExternalSetpointGenerator	ITcMcExternalSetpointGenerator		Function block that implements (cyclic) methods for external setpoint generation
AllowOtherContext	BOOL	false	Allow the calling task to differ from the MC task (MET task). When TRUE, ensure compatible update rates and safe data access in the ITcMcExternalSetpointGenerator methods.
EnableDynamicChecks	BOOL	true	Trigger error on inconsistent setpoint values (position, velocity, acceleration) if EnableDynamicChecks = TRUE.
InvokeId	XWORD	0	Invocation identifier, passed through to the external setpoint generator. Can be used to distinguish multiple activations of the same generator, or to pass custom data (e.g. a pointer).

Name	Type	Default	Description
BufferMode	EBufferMode [► 226]	Aborting	Chronological sequence of the command

In/Outputs

Name	Type	Description
Axis	Reference To <u>AXIS_REF</u> [► 292]	Reference to the axis

Outputs

Name	Type	Description
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Weiterführende Informationen

- [ITcMcExternalSetpointGenerator \[► 216\]](#)
- [Anhang > Externe Sollwertgenerierung \[► 307\]](#)

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.5.2 MC_ExternalSetpointGenerator

This function block provides the API to implement an external setpoint generator from the PLC environment. The included callback methods are called from MC task context. At least the OnCyclicSetpointGeneration method must be implemented.

Requirements for deriving function block:

- The function block must implement ITcMcExternalSetpointGenerator interface.
- The function block and all overridden ITcMcExternalSetpointGenerator methods must have {attribute 'c++_compatible'}.
- The function block must override FB_init.

Notes:

For online change compatibility the function block must have a FB_reinit method, in which MC_ExternalSetpointGenerator.FB_reinit() is called.

Do not call the main FB directly. Only use the available methods.

Methods

Name	Description
FB_reinit [► 286]	
OnPreparation [► 286]	Called once when the external setpoint generator is being prepared for activation.
OnStartPointInitialization [► 287]	Called to initialize the starting point before cyclic setpoint generation begins.

Name	Description
OnCyclicSetpointGeneration [► 287]	Called every MC cycle to compute the next setpoint values. Must be overridden.
OnDeactivation [► 288]	Called when the external setpoint generator is deactivated/aborted.

Weiterführende Informationen

- [ITcMcExternalSetpointGenerator](#) [► 216]
- [Anhang > Externe Sollwertgenerierung](#) [► 307]

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.5.2.1 FB_reinit



Syntax

Definition:

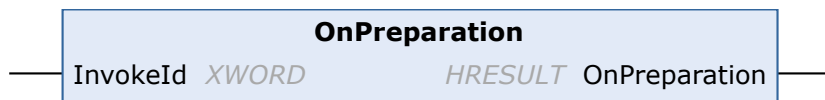
```
METHOD FB_reinit : BOOL
```



Return value

BOOL

5.3.5.5.2.2 OnPreparation



Called once when the external setpoint generator is being prepared for activation.

Syntax

Definition:

```
METHOD OnPreparation : HRESULT
VAR_INPUT
    InvokeId : XWORD;
END_VAR
```



Inputs

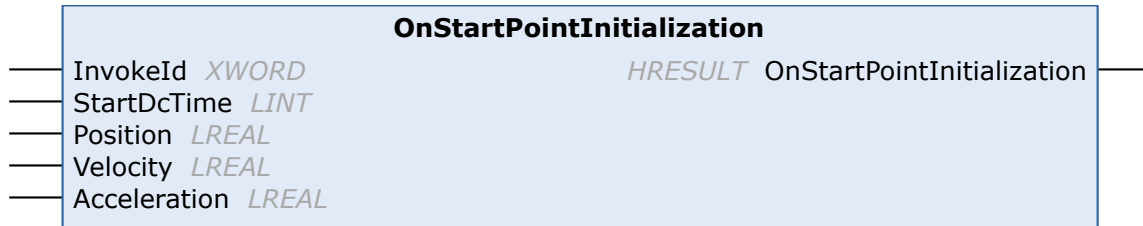
Name	Type	Description
Invokeld	XWORD	



Return value

HRESULT

5.3.5.5.2.3 OnStartPointInitialization



Called to initialize the starting point before cyclic setpoint generation begins.

Syntax

Definition:

```
METHOD OnStartPointInitialization : HRESULT
VAR_INPUT
    InvokeId      : XWORD;
    StartDcTime   : LINT;
    Position      : LREAL;
    Velocity      : LREAL;
    Acceleration  : LREAL;
END_VAR
```

Inputs

Name	Type	Description
Invokeld	XWORD	
StartDcTime	LINT	
Position	LREAL	
Velocity	LREAL	
Acceleration	LREAL	

Return value

HRESULT

5.3.5.5.2.4 OnCyclicSetpointGeneration



Called every MC cycle to compute the next setpoint values. Must be overridden.

Syntax

Definition:

```
METHOD OnCyclicSetpointGeneration : HRESULT
VAR_INPUT
    InvokeId      : XWORD;
    TimeInGeneration : LREAL;
    Position      : Reference To LREAL;
    Velocity      : Reference To LREAL;
    Acceleration  : Reference To LREAL;
END_VAR
```

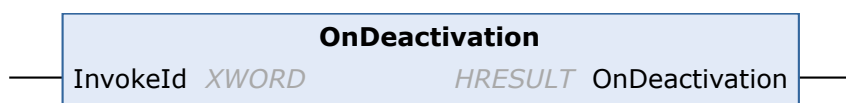
Inputs

Name	Type	Description
Invokeld	XWORD	
TimeInGeneration	LREAL	
Position	Reference To LREAL	
Velocity	Reference To LREAL	
Acceleration	Reference To LREAL	

Return value

HRESULT

5.3.5.5.2.5 OnDeactivation



Called when the external setpoint generator is deactivated/aborted.

Syntax

Definition:

```
METHOD OnDeactivation : HRESULT
VAR_INPUT
    InvokeId : XWORD;
END_VAR
```

Inputs

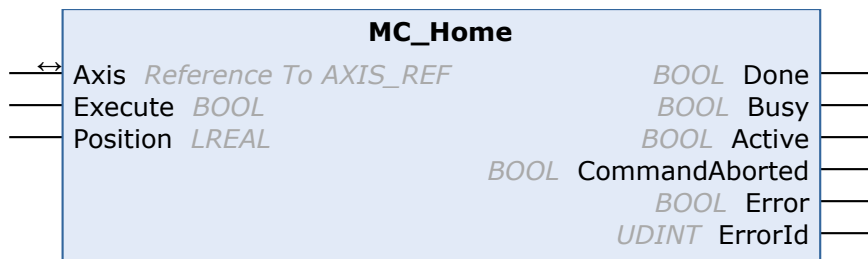
Name	Type	Description
Invokeld	XWORD	

Return value

HRESULT

5.3.5.6 Homing

5.3.5.6.1 MC_Home



This function block commands the axis to perform the homing sequence. The homing sequence is added via the homing TCOM object. The calculated offset is stored in 'Active Homing Offset'. Offsets from MC_SetPosition are discarded.

Syntax

Definition:

```
FUNCTION_BLOCK MC_Home
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    Position      : LREAL;
END_VAR
VAR_OUTPUT
    Done          : BOOL;
    Busy          : BOOL;
    Active        : BOOL;
    CommandAborted : BOOL;
    Error         : BOOL;
    ErrorId       : UDINT;
END_VAR
```

🔧 Inputs

Name	Type	Default	Description
Execute	BOOL		Trigger the command with rising edge.
Position	LREAL	#Default	Absolute position at the reference signal ([PosUnit])

🔧 In/Outputs

Name	Type	Description
Axis	Reference To <u>AXIS_REF</u> ▶ <u>292</u>	Reference to the axis

🔧 Outputs

Name	Type	Description
Done	BOOL	'Search home' sequence succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Auswirkungen auf den Achszustand

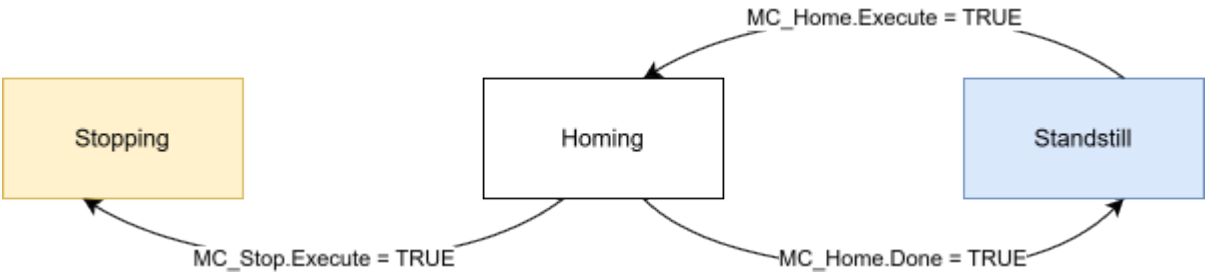


Abb. 2:

Hinweise zum Umstieg von NC2 zu MC3

In der MC3 hat der Baustein MC_Home weniger Eingänge als in der NC2. Der aus der NC2 bekannte Eingang bCalibrationCam ist nun als Eingangsvariable `Signal` am Homing-Objekt enthalten und kann dort direkt verlinkt werden.

- Dokumentation der [Homing Module](#) [► 109]

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.6.2 MC_SetHomingState



This function block allows setting and resetting the `IsPositionValid` flag in the `CDT_MCTOPLC_AXIS_STD`. Optionally, it can also set the Position.

Syntax

Definition:

```

FUNCTION_BLOCK MC_SetHomingState
VAR_INPUT
    Axis      : Type_PtpSetHomingState;
    Execute   : BOOL;
    Position  : LREAL;
    Mode      : EHomingMode;
END_VAR
VAR_OUTPUT
    Done      : BOOL;
    Busy      : BOOL;
    Active     : BOOL;
    CommandAborted : BOOL;
    Error      : BOOL;
    ErrorId    : UDINT;
END_VAR

```

Inputs

Name	Type	Default	Description
Axis	Type_PtpSetHomingState [► 251]		Reference to the axis
Execute	BOOL		Trigger the command with rising edge.
Position	LREAL	#Invalid	New position value ([PosUnit])
Mode	EHomingMode [► 242]	ForcelsHomed	Homing mode selection

Outputs

Name	Type	Description
Done	BOOL	Homing succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.

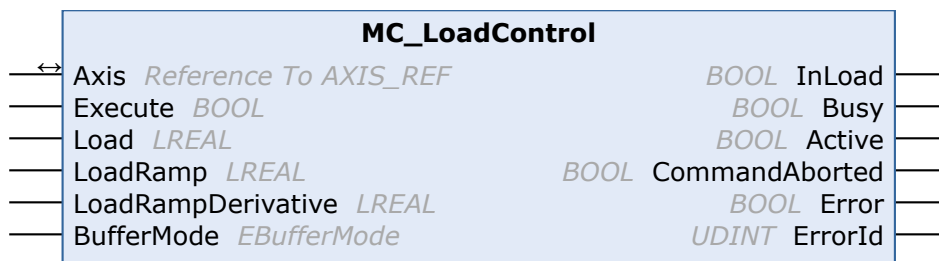
Name	Type	Description
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.7 Load Control

5.3.5.7.1 MC_LoadControl



This function block continuously exerts a torque, force or pressure of the specified value. Positive torque, force and differential pressure is in the positive direction of velocity, pressure is physically unsigned. LoadUnit must match the unit of the value measured by the corresponding sensor.

Syntax

Definition:

```

FUNCTION_BLOCK MC_LoadControl
VAR_IN_OUT
    Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    Load         : LREAL;
    LoadRamp     : LREAL;
    LoadRampDerivative : LREAL;
    BufferMode     : EBufferMode;
END_VAR
VAR_OUTPUT
    InLoad        : BOOL;
    Busy          : BOOL;
    Active        : BOOL;
    CommandAborted : BOOL;
    Error         : BOOL;
    ErrorId       : UDINT;
END_VAR

```

Inputs

Name	Type	Default	Description
Execute	BOOL		Start the motion at rising edge.
Load	LREAL	#Invalid	Commanded load value ([LoadUnit]). It is approached using a defined ramp (LoadRamp).
LoadRamp	LREAL	#Invalid	Maximum load derivative ([LoadUnit/s], positive)

Name	Type	Default	Description
LoadRampDerivative	LREAL	#Ignore	Maximum load ramp derivative ([LoadUnit/s ²], positive). It can be left as 'Ignore' if no ramp derivative limit is desired.
BufferMode	EBufferMode [► 226]	Aborting	Chronological sequence of the command

In/Outputs

Name	Type	Description
Axis	Reference To <u>AXIS_REF</u> [► 292]	Reference to the axis

Outputs

Name	Type	Description
InLoad	BOOL	Commanded load value reached
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.8 Motion Objects

5.3.5.8.1 AXIS_REF



An MC3 axis used with PLCopen motion control function blocks.

Syntax

Definition:

```

FUNCTION_BLOCK AXIS_REF
VAR_INPUT
    PlcToMc : AXIS_REF_TOMC;
    McToPlc : AXIS_REF_TOPLC;
END_VAR

```

Inputs

Name	Type	Description
PlcToMc	<u>AXIS_REF TOMC</u> [► 244]	Data area for cyclic PLC-to-MC data exchange (mapped to MC axis TcCOM object input process image).
McToPlc	<u>AXIS_REF TOPLC</u> [► 245]	Data area for cyclic MC-to-PLC data exchange (mapped to MC axis TcCOM object output process image).

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.8.2 ENCODER_AXIS_REF**ENCODER_AXIS_REF**

McToPlc *ENCODER_AXIS_REF_TOPLC*

An MC3 encoder axis used with PLCopen motion control function blocks.

Syntax

Definition:

```
FUNCTION_BLOCK ENCODER_AXIS_REF
VAR_INPUT
    McToPlc : ENCODER_AXIS_REF_TOPLC;
END_VAR
```

 Inputs

Name	Type	Description
McToPlc	ENCODER_AXIS_REF_TOPLC C [▶ 246]	Data area for cyclic MC-to-PLC data exchange (mapped to MC encoder axis TcCOM object output process image).

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.9 Synchronized Motion**5.3.5.9.1 MC_GearIn****MC_GearIn**

Master	Type_Master1D	BOOL	InGear
Slave	Reference To AXIS_REF	BOOL	Busy
Execute	BOOL	BOOL	Active
RatioNumerator	LREAL	BOOL	CommandAborted
RatioDenominator	UDINT	BOOL	Error
Acceleration	LREAL	UDINT	ErrorId
Deceleration	LREAL	BOOL	MasterHasError
Jerk	LREAL		
BufferMode	EBufferMode		
ReactionToMasterError	EReactionToMasterError		
ReactionToSlaveError	EReactionToSlaveError		

This function block commands a ratio between the velocity of the slave and master axis.

Syntax

Definition:

```
FUNCTION_BLOCK MC_GearIn
VAR_INPUT
    Master      : Type_Master1D;
    Slave       : Reference To AXIS_REF;
    Execute     : BOOL;
```

```

    RatioNumerator      : LREAL;
    RatioDenominator    : UDINT;
    Acceleration         : LREAL;
    Deceleration         : LREAL;
    Jerk                 : LREAL;
    BufferMode            : EBufferMode;
    ReactionToMasterError : EReactionToMasterError;
    ReactionToSlaveError  : EReactionToSlaveError;
END_VAR
VAR_OUTPUT
    InGear              : BOOL;
    Busy                : BOOL;
    Active              : BOOL;
    CommandAborted      : BOOL;
    Error               : BOOL;
    ErrorId             : UDINT;
    MasterHasError       : BOOL;
END_VAR

```

Inputs

Name	Type	Default	Description
Master	Type_Master1D		Reference to the master axis
Slave	Reference To AXIS_REF [► 292]		Reference to the slave axis
Execute	BOOL		Trigger the command with rising edge.
RatioNumerator	LREAL	1.0	Gear ratio numerator
RatioDenominator	UDINT	1	Gear ratio denominator
Acceleration	LREAL	#Default	Maximum slave acceleration ([PosUnit]/s ² , positive)
Deceleration	LREAL	#Default	Maximum slave deceleration ([PosUnit]/s ² , positive)
Jerk	LREAL	#Default	Maximum slave jerk ([PosUnit]/s ³ , positive)
BufferMode	EBufferMode [► 226]	Aborting	Chronological sequence of the command
ReactionToMasterError	EReactionToMasterError	TriggerSlaveError	Reaction if an error occurs in the master while this command is active.
ReactionToSlaveError	EReactionToSlaveError	TriggerMasterError	Reaction if an error occurs in the slave while this command is active.

Outputs

Name	Type	Description
InGear	BOOL	Synchronization established
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
MasterHasError	BOOL	Master is in error state.

Weiterführende Informationen

Die Eingänge Acceleration und Deceleration sind identisch zu belegen. Des Weiteren ist eine Kopplung nur dann möglich, wenn die Slave-Achse sich im Stillstand befindet.

Ergänzungen zu ReactionToSlaveError und ReactionToMasterError

Grundsätzlich gilt bei einem Fehler der Slave-Achse:

- Slave-Achse: Die Kopplung wird aufgelöst und die Reaktion ist abhängig vom Fehlertypen.
- Master-Achse: Die Master-Achse meldet einen Fehler und folgt einer Stopprampe der MC oder führt ihre laufende Bewegung ohne Beeinflussung fort.

Bei einem Fehler der Master-Achse gilt:

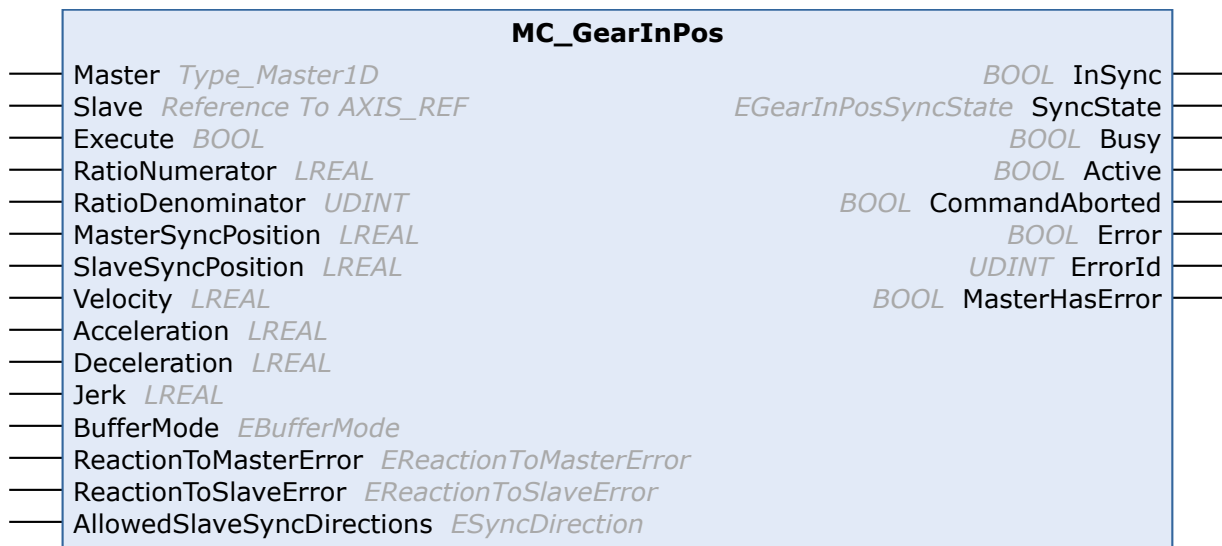
- Master-Achse: Reaktion ist abhängig vom Fehlertypen.
- Slave-Achse: Die Slave-Achse meldet einen Fehler und folgt einer Stopprampe der MC oder den Sollwerten der Master-Achse.

Ausführliche Informationen sind im Kapitel [Fehlerverhalten](#) [► 311] dokumentiert.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.9.2 MC_GearInPos



This function block commands a gear ratio between the position of the slave and master axes from the synchronization point onwards.

Syntax

Definition:

```
FUNCTION_BLOCK MC_GearInPos
VAR_INPUT
    Master          : Type_Master1D;
    Slave           : Reference To AXIS_REF;
    Execute         : BOOL;
    RatioNumerator  : LREAL;
    RatioDenominator : UDINT;
    MasterSyncPosition : LREAL;
    SlaveSyncPosition : LREAL;
    Velocity        : LREAL;
    Acceleration    : LREAL;
    Deceleration    : LREAL;
    Jerk           : LREAL;
    BufferMode       : EBufferMode;
    ReactionToMasterError : EReactionToMasterError;
    ReactionToSlaveError : EReactionToSlaveError;
    AllowedSlaveSyncDirections : ESyncDirection;
END_VAR
VAR_OUTPUT
    InSync : BOOL;
```

```

SyncState      : EGearInPosSyncState;
Busy           : BOOL;
Active         : BOOL;
CommandAborted : BOOL;
Error          : BOOL;
ErrorId        : UDINT;
MasterHasError : BOOL;
END_VAR

```

Inputs

Name	Type	Default	Description
Master	Type_Master1D		Reference to the master axis
Slave	Reference To AXIS_REF [► 292]		Reference to the slave axis
Execute	BOOL		Trigger the command with rising edge.
RatioNumerator	LREAL	1.0	Gear ratio numerator
RatioDenominator	UDINT	1	Gear ratio denominator
MasterSyncPosition	LREAL	#Invalid	Position of master axis where master and slave axis are synchronized.
SlaveSyncPosition	LREAL	#Invalid	Position of slave axis where master and slave axis are synchronized.
Velocity	LREAL	#Maximum	Maximum velocity during synchronization ([PosUnit]/s, positive)
Acceleration	LREAL	#Default	Maximum acceleration during synchronization ([PosUnit]/s^2, positive)
Deceleration	LREAL	#Default	Maximum deceleration during synchronization ([PosUnit]/s^2, positive)
Jerk	LREAL	#Default	Maximum jerk during synchronization ([PosUnit]/s^3, positive)
BufferMode	EBufferMode [► 226]	Aborting	Chronological sequence of the command
ReactionToMasterError	EReactionToMasterError	TriggerSlaveError	Reaction if an error occurs in the master while this command is active.
ReactionToSlaveError	EReactionToSlaveError	TriggerMasterError	Reaction if an error occurs in the slave while this command is active.
AllowedSlaveSyncDirections	ESyncDirection	TowardsSlaveSyncPosition	Allowed directions for establishing initial synchronization

Outputs

Name	Type	Description
InSync	BOOL	True if SyncState == InSync (slave coupled to the master AND master has passed its master sync position).
SyncState	EGearInPosSyncState [► 241]	Synchronization state. See EGearInPosSyncState for details.
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
MasterHasError	BOOL	Master is in error state.

Ergänzungen zu ReactionToSlaveError und ReactionToMasterError

Grundsätzlich gilt bei einem Fehler der Slave-Achse:

- Slave-Achse: Die Kopplung wird aufgelöst und die Reaktion ist abhängig vom Fehlertypen.
- Master-Achse: Die Master-Achse meldet einen Fehler und folgt einer Stopprampe der MC oder führt ihre laufende Bewegung ohne Beeinflussung fort.

Bei einem Fehler der Master-Achse gilt:

- Master-Achse: Reaktion ist abhängig vom Fehlertypen.
- Slave-Achse: Die Slave-Achse meldet einen Fehler und folgt einer Stopprampe der MC oder den Sollwerten der Master-Achse.

Ausführliche Informationen sind im Kapitel [Fehlerverhalten](#) [► 311] dokumentiert.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.9.3 MC_HaltPhasing



This function block commands a phase shift stop in the master position of a synchronized slave axis. It is only valid for camming synchronization. The master axis is not aware of the phase shift.

Syntax

Definition:

```
FUNCTION_BLOCK MC_HaltPhasing
VAR_INPUT
    Master      : Type_Master1D;
    Slave      : Type_Phasing1D;
    Execute     : BOOL;
    Deceleration : LREAL;
    Jerk       : LREAL;
END_VAR
VAR_OUTPUT
    Done       : BOOL;
    Busy       : BOOL;
    Active     : BOOL;
    CommandAborted : BOOL;
    Error      : BOOL;
    ErrorId    : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
Master	Type_Master1D		Reference to the master axis
Slave	Type_Phasing1D		Reference to the slave axis
Execute	BOOL		Trigger the command with rising edge.
Deceleration	LREAL	#Invalid	Maximum deceleration ([PosUnit]/s^2, positive)
Jerk	LREAL	#Invalid	Maximum jerk ([PosUnit]/s^3, positive)

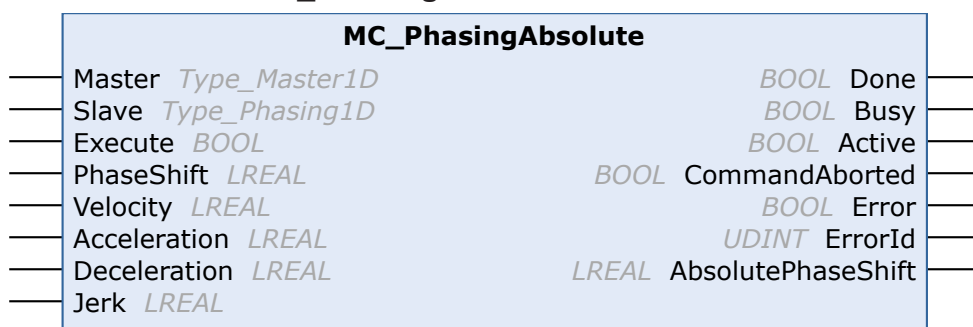
🔌 Outputs

Name	Type	Description
Done	BOOL	Zero dynamics has been reached, axis state is transferred to Standstill.
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.9.4 MC_PhasingAbsolute



This function block commands an absolute phase shift in the master position of a synchronized slave axis. It is only valid for camming synchronization. The master axis is not aware of the phase shift.

Syntax

Definition:

```
FUNCTION_BLOCK MC_PhasingAbsolute
VAR_INPUT
    Master      : Type_Master1D;
    Slave      : Type_Phasing1D;
    Execute     : BOOL;
    PhaseShift  : LREAL;
    Velocity    : LREAL;
    Acceleration : LREAL;
    Deceleration : LREAL;
    Jerk        : LREAL;
END_VAR
VAR_OUTPUT
    Done        : BOOL;
    Busy        : BOOL;
    Active      : BOOL;
    CommandAborted : BOOL;
    Error       : BOOL;
    ErrorId     : UDINT;
    AbsolutePhaseShift : LREAL;
END_VAR
```

🔌 Inputs

Name	Type	Default	Description
Master	Type_Master1D		Reference to the master axis

Name	Type	Default	Description
Slave	Type_Phasing1D		Reference to the slave axis
Execute	BOOL		Trigger the command with rising edge.
PhaseShift	LREAL	#Invalid	Absolute phase difference in master position of the slave axis ([PosUnit])
Velocity	LREAL	#Invalid	Maximum velocity ([PosUnit]/s, positive)
Acceleration	LREAL	#Invalid	Maximum acceleration ([PosUnit]/s^2, positive)
Deceleration	LREAL	#Invalid	Maximum deceleration ([PosUnit]/s^2, positive)
Jerk	LREAL	#Invalid	Maximum jerk ([PosUnit]/s^3, positive)

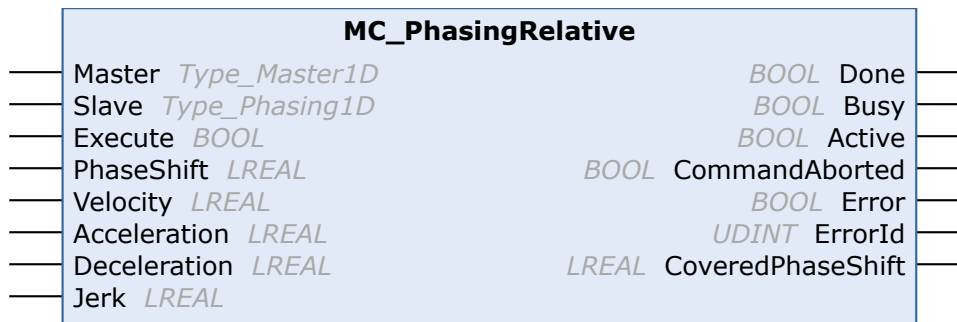
Outputs

Name	Type	Description
Done	BOOL	Commanded phasing reached
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
AbsolutePhaseShift	LREAL	Current absolute phase shift between master and slave.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.9.5 MC_PhasingRelative



This function block commands a relative phase shift in the master position of a synchronized slave axis. It is only valid for camming synchronization. The master axis is not aware of the phase shift.

Syntax

Definition:

```

FUNCTION_BLOCK MC_PhasingRelative
VAR_INPUT
    Master      : Type_Master1D;
    Slave      : Type_Phasing1D;
    Execute     : BOOL;
    PhaseShift  : LREAL;
    Velocity    : LREAL;
    Acceleration : LREAL;
    Deceleration : LREAL;

```

```

    Jerk          : LREAL;
END_VAR
VAR_OUTPUT
    Done          : BOOL;
    Busy          : BOOL;
    Active        : BOOL;
    CommandAborted : BOOL;
    Error         : BOOL;
    ErrorId       : UDINT;
    CoveredPhaseShift : LREAL;
END_VAR

```

Inputs

Name	Type	Default	Description
Master	Type_Master1D		Reference to the master axis
Slave	Type_Phasing1D		Reference to the slave axis
Execute	BOOL		Trigger the command with rising edge.
PhaseShift	LREAL	#Invalid	Additional phase difference in master position of the slave axis ([PosUnit])
Velocity	LREAL	#Invalid	Maximum velocity ([PosUnit]/s, positive)
Acceleration	LREAL	#Invalid	Maximum acceleration ([PosUnit]/s^2, positive)
Deceleration	LREAL	#Invalid	Maximum deceleration ([PosUnit]/s^2, positive)
Jerk	LREAL	#Invalid	Maximum jerk ([PosUnit]/s^3, positive)

Outputs

Name	Type	Description
Done	BOOL	Commanded phasing reached
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
CoveredPhaseShift	LREAL	Accumulated relative phase shift applied so far.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.10 TouchProbe

5.3.5.10.1 BinaryTouchProbeRef

TouchProbeRef for Binary Touch Probes.

Do not call the main FB directly. Only use the available methods.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.10.2 DriveTouchProbeRef

TouchProbeRef for Drive Touch Probe.

Do not call the main FB directly. Only use the available methods.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.10.3 EncoderTouchProbeRef

TouchProbeRef for Encoder Touch Probe.

Do not call the main FB directly. Only use the available methods.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.10.4 FixedStopTouchProbeRef

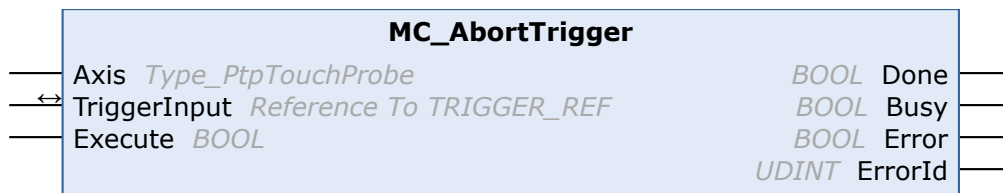
TouchProbeRef for Fixed Stop Detection.

Do not call the main FB directly. Only use the available methods.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.10.5 MC_AbortTrigger



This Function Block is used to abort function blocks which are used to record trigger events (e.g. MC_TouchProbe).

Syntax

Definition:

```
FUNCTION_BLOCK MC_AbortTrigger
VAR_INPUT
    Axis          : Type_PtpTouchProbe;
END_VAR
VAR_IN_OUT
    TriggerInput : Reference To TRIGGER_REF;
END_VAR
VAR_INPUT
    Execute      : BOOL;
END_VAR
VAR_OUTPUT
    Done         : BOOL;
    Busy         : BOOL;
    Error        : BOOL;
    ErrorId      : UDINT;
END_VAR
```

Inputs

Name	Type	Description
Axis	Type_PtpTouchProbe [► 253]	Reference to the axis
Execute	BOOL	Aborts the monitoring of trigger event at rising edge.

In/Outputs

Name	Type	Description
TriggerInput	Reference To TRIGGER_REF [► 249]	Reference to the trigger signal source like DriveTouchProbeRef or BinaryTouchProbeRef

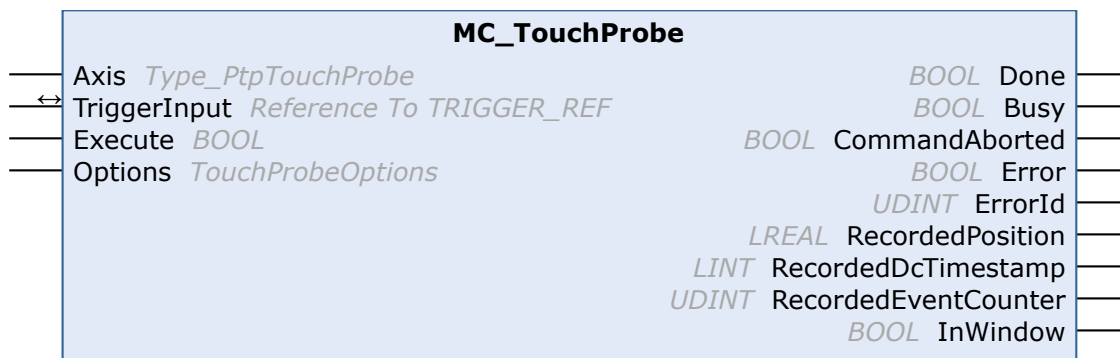
Outputs

Name	Type	Description
Done	BOOL	Abort Trigger succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.10.6 MC_TouchProbe



This function block controls the recording of an axis position at a trigger event. One of WindowFirstPosition or WindowLastPosition can be left as 'Ignore' if no corresponding limit is desired.

Syntax

Definition:

```
FUNCTION_BLOCK MC_TouchProbe
VAR_INPUT
    Axis          : Type_PtpTouchProbe;
END_VAR
VAR_IN_OUT
    TriggerInput  : Reference To TRIGGER_REF;
END_VAR
VAR_INPUT
    Execute       : BOOL;
    Options       : TouchProbeOptions;
END_VAR
VAR_OUTPUT
    Done          : BOOL;
```

```

    Busy                : BOOL;
    CommandAborted      : BOOL;
    Error               : BOOL;
    ErrorId             : UDINT;
    RecordedPosition    : LREAL;
    RecordedDcTimestamp : LINT;
    RecordedEventCounter : UDINT;
    InWindow            : BOOL;
END_VAR

```

Inputs

Name	Type	Description
Axis	Type PtpTouchProbe [► 253]	Reference to the axis
Execute	BOOL	Trigger the command with rising edge.
Options	TouchProbeOptions [► 248]	Options for the touch probe

In/Outputs

Name	Type	Description
TriggerInput	Reference To TRIGGER_REF [► 249]	Reference to the trigger signal source like DriveTouchProbeRef or BinaryTouchProbeRef

Outputs

Name	Type	Description
Done	BOOL	Touch probe process succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.
CommandAborted	BOOL	Command is aborted by another command (MC_AbortTrigger).
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
RecordedPosition	LREAL	Position where trigger event occurred ([PosUnit]).
RecordedDcTimestamp	LINT	DcTime at trigger event
RecordedEventCounter	UDINT	Number of recorded trigger events
InWindow	BOOL	TRUE if the recorded position is within the configured window.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.5.10.7 TimestampTouchProbeRef

TouchProbeRef for Timestamp Touch Probes.

Do not call the main FB directly. Only use the available methods.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5500 MC3 Base >= v4.0.6

5.3.6 Anhang

5.3.6.1 Umstieg NC2 zur MC3

Funktionsbausteine

NC2 Funktionsbaustein	MC3 Funktionsbaustein	Hinweis
MC_Power.Enable_Positive MC_Power.Enable_Negative	MC_Power [► 253].EnablePositive MC_Power [► 253].EnableNegative	Unterstrich entfernt
MC_ExtSetPointGenEnable	MC_ActivateExternalSetpointGeneration [► 284]	Weitere Details zur <u>Externe Sollwertgenerierung</u> [► 307]
MC_ExtSetPointGenDisable	MC_ActivateExternalSetpointGeneration [► 284] mit einem Aborting Kommando ablösen	Beispiel: MC_Halt
MC_ExtSetPointGenFeed	Siehe <u>ITcNcExternalSetpointGeneration</u> [► 284]	
MC_ExtSetPointGenFeedWithTorque	-	
MC_GearInDyn	In Zukunft über MC_GearIn [► 293]	
MC_GearOut		Durch ein Bewegungs- oder Halt/Stop-Kommando auf eine Slave-Achse wird diese vom Master abgekoppelt.
MC_CamOut		Durch ein Bewegungs- oder Halt/Stop-Kommando auf eine Slave-Achse wird diese vom Master abgekoppelt.
MC_Power.Override (Wertebereich in %)	MC_SetOverride [► 262].VelocityFactor (Wertebereich normalisiert auf 1)	Bei der MC3 ist der Override standardmäßig auf 1 (100%) gesetzt. In der Motion UI erfolgt die Eingabe in %.
MC_SetPosition.Mode	MC_SetPosition [► 263].Relative	Umbenennung des Eingangs für bessere Lesbarkeit im Code.
MC_Reset	MC_Reset [► 261]	MC3: Wirkt, wenn sich die Achse im Fehlerzustand befindet. Ein zusätzlicher asynchroner Reset-Befehl für Sercos-Geräte wird ebenfalls ausgelöst und ausgewertet. MC_Reset.Done = TRUE, wenn die Hardware zurückgesetzt wurde (einschließlich zusätzlichem Reset für Sercos) und sich der Achsenstatus auf „Disabled“ oder „Standstill“ geändert hat. Der Schleppfehler wird nicht zurückgesetzt und bleibt trotz Reset-Kommando bestehen.
MC_Home.bCalibrationCam	Siehe Kapitel MC_Home [► 288] und Homing-Module [► 109]	Zentrale Konfiguration, Parametrierung und Verknüpfung mit I/O-Komponenten zentral in Homing Modulen [► 109]

NC2 Funktionsbaustein	MC3 Funktionsbaustein	Hinweis
MC_Home.HomingMode := MC_DefaultHoming	MC_Home [► 288]	MC_Home führt die Standard-Referenzierfahrt, die im Homing-Objekt [► 109] konfiguriert ist, aus.
MC_Home.HomingMode := MC_Direct	MC_SetHomingState [► 290].Mode := EHomingMode.DirectSetPosition	Zusätzlicher Funktionsbaustein neben MC_Home [► 288].
MC_Home.HomingMode := MC_ForceCalibration	MC_SetHomingState [► 290].Mode := EHomingMode [► 242].ForcelsPositionValid	Zusätzlicher Funktionsbaustein neben MC_Home [► 288].
MC_Home.HomingMode := MC_ResetCalibration	MC_SetHomingState [► 290].Mode := EHomingMode [► 242].ResetIsPositionValid	Zusätzlicher Funktionsbaustein neben MC_Home [► 288].
MC_MoveVelocity.InVelocity MC_MoveVelocity.Busy MC_MoveVelocity.Active	MC_MoveVelocity [► 273].InVelocity MC_MoveVelocity [► 273].Busy MC_MoveVelocity [► 273].Active	In der Ausgangssituation wurde das Kommando mit <code>Execute := TRUE</code> abgeschickt und es wird wieder <code>Execute := FALSE</code> gesetzt. NC2: <code>Busy = TRUE</code>, <code>Active = TRUE</code> und <code>InVelocity = TRUE</code> werden mit einer fallenden Flanke am <code>Execute</code> Eingang zurückgesetzt. Jedoch signalisieren die Ausgänge (insbesondere <code>InVelocity</code>) mindestens für einen Zyklus <code>TRUE</code>, sobald die Zielgeschwindigkeit erreicht ist. MC3: <code>Busy = TRUE</code>, <code>Active = TRUE</code> und <code>InVelocity = TRUE</code> bleiben so lange gesetzt, bis das Bewegungskommando durch ein anderes Kommando abgelöst wird, auch wenn <code>Execute</code> bereits auf <code>FALSE</code> gesetzt ist (PLCopen-konform).
MC_GearIn.InGear MC_GearIn.Busy MC_GearIn.Active (ebenso MC_GearInPos)	MC_GearIn [► 293].InGear MC_GearIn [► 293].Busy MC_GearIn [► 293].Active (ebenso MC_GearInPos [► 295])	In der Ausgangssituation wurde das Kommando mit <code>Execute := TRUE</code> abgeschickt und es wird wieder <code>Execute := FALSE</code> gesetzt. NC2: <code>Busy = TRUE</code>, <code>Active = TRUE</code> und <code>InGear = TRUE</code> werden mit einer fallenden Flanke am <code>Execute</code> Eingang zurückgesetzt. Jedoch signalisieren die Ausgänge (insbesondere <code>InGear</code>) mindestens für einen Zyklus <code>TRUE</code>, sobald die Master- und Slave-Achse sich synchron zueinander bewegen. MC3: <code>Busy = TRUE</code>, <code>Active = TRUE</code> und <code>InGear = TRUE</code> bleiben so lange gesetzt, bis das Bewegungskommando durch ein anderes Kommando abgelöst wird,

NC2 Funktionsbaustein	MC3 Funktionsbaustein	Hinweis
		auch wenn <code>Execute</code> bereits auf <code>FALSE</code> gesetzt ist (PLCopen-konform).

Re-Triggern eines MC-Funktionsbausteins am Beispiel MC_MoveAbsolute

```

VAR
    axis_nc2 : TC2_MC2.AXIS_REF;
    axis_mc3 : Tc3_Mc3Ptp.AXIS_REF;

    move_absolute_mc3 : Tc3_Mc3Ptp.MC_MoveAbsolute;
    move_absolute_nc2_1 : Tc2_MC2.MC_MoveAbsolute;
    move_absolute_nc2_2 : Tc2_MC2.MC_MoveAbsolute;
END_VAR

CASE step OF
    0:
        // MC3 init movement
        move_absolute_mc3(Axis := axis_mc3, Position := 111, Execute := TRUE);

        // NC2 init movement
        move_absolute_nc2_1(Axis := axis_nc2, Position := 111, Velocity := 100, Execute := TRUE);

        step := step + 1;

    1:
        // MC3 retrigger same FB instance during one cycle
        move_absolute_mc3 (Axis := axis_mc3, Position := 111, Velocity := 100, Execute := FALSE);
        move_absolute_mc3 (Axis := axis_mc3, Position := 222, Velocity := 100, Execute := TRUE);

        // NC2 use additional FB instance
        move_absolute_nc2_1(Axis := axis_nc2, Position := 111, Velocity := 100, Execute := FALSE);
        move_absolute_nc2_2(Axis := axis_nc2, Position := 222, Velocity := 100, Execute := TRUE);
        step := step + 1;
END_CASE

```

i Re-Triggern eines MC-Funktionsbausteins am Beispiel MC_MoveAbsolute

In der NC2 gilt: Wenn der Eingang `Execute` während der Ausführung des Kommandos erneut getriggert wird, gibt der Funktionsbaustein keinerlei Rückmeldung, führt aber auch kein weiteres Kommando aus.

Für die MC3 gilt diese Einschränkung nicht. Wenn der Eingang `Execute` während der Ausführung des Kommandos erneut getriggert wird, werden die neuen Parameter übernommen und der Baustein entsprechend des Buffermodus erneut ausgeführt. Dies kann innerhalb eines PLC-Zyklus durchgeführt werden. Beispielsweise ist der erste Aufruf mit `Execute := FALSE` (Wechsel von `TRUE` auf `FALSE`) `fbMyMoveAbsolute(Axis := ax, Execute := FALSE, Position := 111)` und direkt folgend der zweite Aufruf derselben Bausteininstanz mit einer neuen Position `fbMyMoveAbsolute(Axis := ax, Execute := TRUE, Position := 222)`.

Status

NC2	MC3	Hinweis
myAxisRef.Status.IoDataInvalid	myAxisRef.McToPlc.Std.AxisState	PLCopen Zustandsdiagramm erweitert (Zustände: Invalid, NotReady, Enabling, Disabling). Beispiel: Die Achse verbleibt im Zustand NotReady, wenn die Antriebshardware nicht bereit ist (siehe Kapitel Zustandsdiagramm [► 234])
myAxisRef.Status.Homed	AxisRef.McToPlc.Std.IsPositionValid	IsPositionValid berücksichtigt z. B. auch den Zustand des Antriebsobjekts der Achse (CoE Antrieb Input: Position actual value invalid). Startet die Achse

NC2	MC3	Hinweis
		u. a. mit einem Absolutwertgeber, hat die Achse eine eindeutige Referenzposition und meldet <code>IsPositionValid := TRUE</code> .
<code>myAxisRef.ReadStatus()</code>	<code>myAxisRef.McToPlc.Std</code>	Statusinformationen können direkt aus dem CDT [► 212] <code>Std</code> abgerufen werden.

Fehler auf Ebene der Antriebshardware oder fehlende Kommunikation

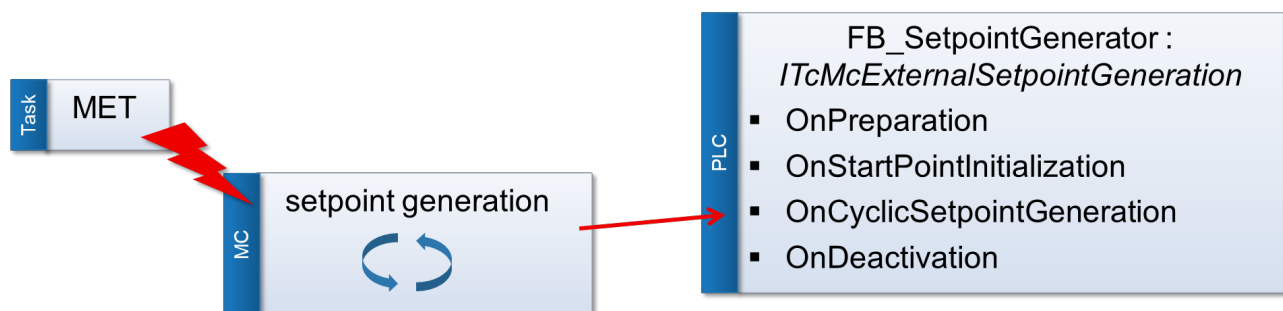
Ausgangssituation ist der Wechsel des TwinCAT-Systems von Config nach Run. Dabei ist die Antriebshardware in einem Fehlerzustand oder die Kommunikation über den EtherCAT-Feldbus ist gestört.

- **NC2:** Ein Achsfehler wird angezeigt, wenn `MC_Power.Enable = TRUE`. Des Weiteren kann ein Kommunikationsproblem z. B. über `myAxisRef.Status.IoDataInvalid` abgefragt werden.
- **MC3:** Wenn die Kommunikation zwischen Achse und Antrieb hergestellt ist, werden der Antriebsstatus und mögliche Fehler direkt ausgewertet und ggf. als Achsfehler angezeigt. Das Zustandsdiagramm der MC3-Achse ist gegenüber der NC2 und PLCopen erweitert (Zustände: Invalid, NotReady, Enabling, Disabling). Über den Achsstatus der MC3 kann zentralisiert der Zustand ausgewertet werden.
 - Beispiel 1: Wenn `MC_Power.Enable := FALSE` und keine Kommunikation mit dem Antrieb möglich ist, verbleibt der Achsstatus im Zustand `myAxisRef.McToPlc.Std.AxisState = EAxisState.NotReady`.
 - Beispiel 2: Wenn `MC_Power.Enable := FALSE` und der Antrieb ist im Fehlerzustand, dann meldet die MC3-Achse `myAxisRef.McToPlc.Std.AxisState = EAxisState.ErrorStop`.

5.3.6.2 Externe Sollwertgenerierung

Die externe Sollwertvorgabe kann genutzt werden, um ein eigenes neues Sollwertprofil zu implementieren, das in der TwinCAT MC3 nicht standardmäßig verfügbar ist. Dieses Profil kann mathematisch sehr einfach oder auch beliebig komplex sein. Bei der externen Sollwertgenerierung werden die Sollwerte in Funktionsblöcken bzw. TcCOM-Modulen generiert, die die Schnittstelle [ITcMcExternalSetpointGenerator \[► 216\]](#) implementieren. Diese wird aus dem MC-Kontext aufgerufen.

Der zentrale Eingang ist `ExternalSetpointGenerator` vom Typ [ITcMcExternalSetpointGenerator \[► 216\]](#). Diese Schnittstelle definiert Methoden, die implementiert werden müssen (z. B. von einem PLC-FB). Diese Methoden werden vom MC-Task-Kontext (MET-Task) über einen Callback-Mechanismus aufgerufen. Dieser Mechanismus ist die Basis für eine Entkopplung von MC-Task und PLC-Task bei der externen Sollwertgenerierung. Dieses grundlegende Funktionsprinzip ist in der nachfolgenden Abbildung dargestellt.



Mit der MC3 ist es möglich, PLC und MC3-Achsen bei der externen Sollwertgenerierung in unterschiedlichen Kontexten mit unterschiedlichen Zykluszeiten auszuführen. Die Update-Rate der Daten, die in den Callback Methoden verwendet werden, wenn `AllowOtherContext` aktiviert ist, ist zu beachten.

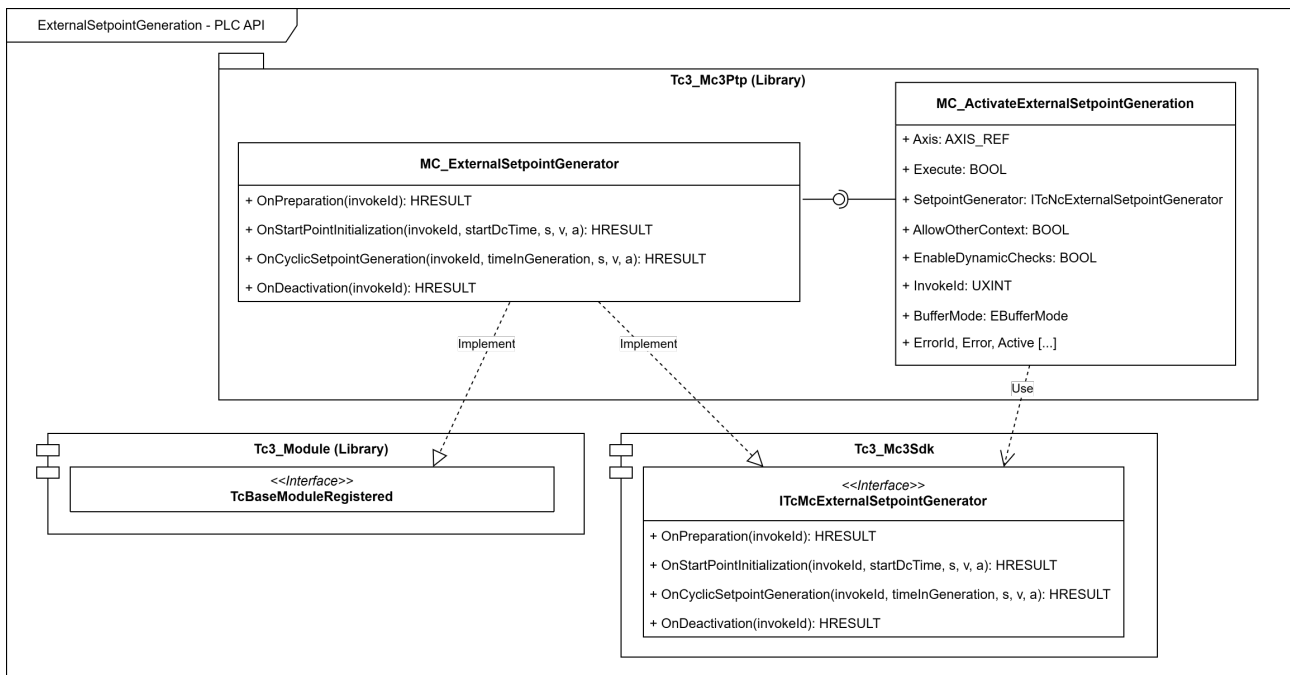
Es ist möglich, dass die MC3 die berechneten Sollwerte auf Konsistenz prüft (vom vorherigen Zyklus zum aktuellen Zyklus unter Berücksichtigung der maximalen dynamischen Parameter der Achse). Dazu muss `EnableDynamicChecks := TRUE` gesetzt werden.

Die Interpretation der Eingangsvariable `InvokeId` vom Typ `XWORD` ist kundenspezifisch. Zum Beispiel kann ein beliebiger numerischer Wert oder die Adresse (Pointer) auf eine kundenspezifische Datenstruktur übergeben werden. Diese `InvokeId` wird jeder, von der Schnittstelle inkludierten Methode, als Input übergeben.

Die enthaltenen Callback-Methoden im Interface `ITcMcExternalSetpointGenerator` [► 216] werden vom MC-Task-Kontext (MET-Task) in der aufgeführten Reihenfolge aufgerufen:

1. `OnPreparation (invokeId : XWORD)`
 - Wird direkt aufgerufen, wenn `MC_ActivateExternalSetpointGeneration` [► 284].`Execute` = `TRUE`
 - Die Methode kann zum Beispiel zur Initialisierung der Sollwertgenerierung verwendet werden.
2. `OnStartPointInitialization (invokeId : XWORD, startDcTime : LINT, position : LREAL, velocity : LREAL, acceleration : LREAL)`
 - Wird aufgerufen, kurz bevor der externe Sollwertgenerator aktiv wird und der nächste Sollwert zum ersten Mal von diesem Generator generiert wird.
 - Die Methode kann zum Beispiel zur Initialisierung der Sollwertgenerierung verwendet werden.
 - Der letzte Sollwert (Position und Dynamik) und der entsprechende Zeitstempel des vorherigen Befehls (zu diesem Zeitpunkt beendet) werden an diese Methode übergeben. Der nächste Sollwert/Zyklus wird durch den externen Sollwertgenerator berechnet.
3. `ABSTRACT OnCyclicSetpointGeneration (invokeId : XWORD, timeInGeneration : LREAL, position : REFERENCE TO LREAL, velocity : REFERENCE TO LREAL, acceleration : REFERENCE TO LREAL)`
 - Diese Methode muss implementiert werden.
 - Die Methode wird zyklisch aufgerufen, wenn `MC_ActivateExternalSetpointGeneration` [► 284].`Active` = `TRUE`.
 - Die Methode wird zur Berechnung des Sollwerts für den angegebenen Zeitstempel verwendet.
 - Die Eingangsvariable `timeInGeneration` beinhaltet die verstrichene Zeit, während die externe Sollwertgenerierung aktiv ist.
 - Position, Geschwindigkeit und Beschleunigung müssen auf die Werte für den aktuellen Zyklus der Sollwertgenerierung gesetzt werden.
4. `OnDeactivation (invokeId : XWORD)`
 - Die Methode ist für die für letzte Schritte vorgesehen, wenn die externe Sollwertgenerierung durch einen anderen Fahrbefehl beendet wird.
 - Sie kann z. B. verwendet werden, um Speicherplatz, der während der externen Sollwertgenerierung allokiert wurde, wieder freizugeben.

Der FB `MC_ExternalSetpointGenerator` [► 285] stellt die API zur Realisierung einer externen Sollwertgenerierung aus der PLC-Umgebung zur Verfügung. Das nachfolgende Komponentendiagramm stellt die Zusammenhänge der Module dar. Der Generator selbst ist ein `TcCOM`-Objekt, das zur Laufzeit angelegt und im TwinCAT-System registriert wird (`TcBaseModuleRegistered`). Methoden zur Behandlung von Online-Change Vorgängen werden von diesem Generator-FB bereits implementiert. Als Basisfunktionalität implementiert er den `ITcMcExternalSetpointGenerator` [► 216] (erweitert `ITcUnknown`).



5.3.6.3 Modulo-Positionierung

Die Modulo-Positionierung kann bei geschlossenen Linearachsen ebenso wie bei rotatorischen Achsen angewendet werden. In TwinCAT nicht zwischen diesen Typen unterschieden. Auch eine Modulo-Achse hat eine fortlaufende absolute Position im Bereich $\pm\infty$. Die Modulo-Position der Achse ist 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 MC3 Modulo Positionierung die absolute Zielposition aus folgenden Parametern gebildet:

- MC3 Achsparametern [► 29]:
 - Position Modulus
 - Position Modulo Tolerance Window
- Tc3_Mc3Ptp.MC_MoveModulo [► 278] Eingangsparametern
 - MC_MoveModulo.Position
 - MC_MoveModulo.AdditionalTurns
 - MC_MoveModulo.Direction

Achs-Parametrierung

Die Modulo-Unterstützung ist an einer MC3-Achse über den Parameter „Is Modulo Axis“ zu aktivieren. Ist der Parameter gesetzt, so können die weiteren Modulo-Parameter gesetzt werden.

	Name	Value	CS	Unit	Type
-	General				
	Simulation mode	☒ TRUE	☐		BOOL
	Position Unit	mm 	☐	[PosUnit]	MC.EPositionUnit
	Is Modulo Axis	☒ TRUE	☐		BOOL
	Position Modulus	360.0	☐	[PosUnit]	LREAL
	Position Modulo Tolerance Window	0.0	☐	[PosUnit]	LREAL

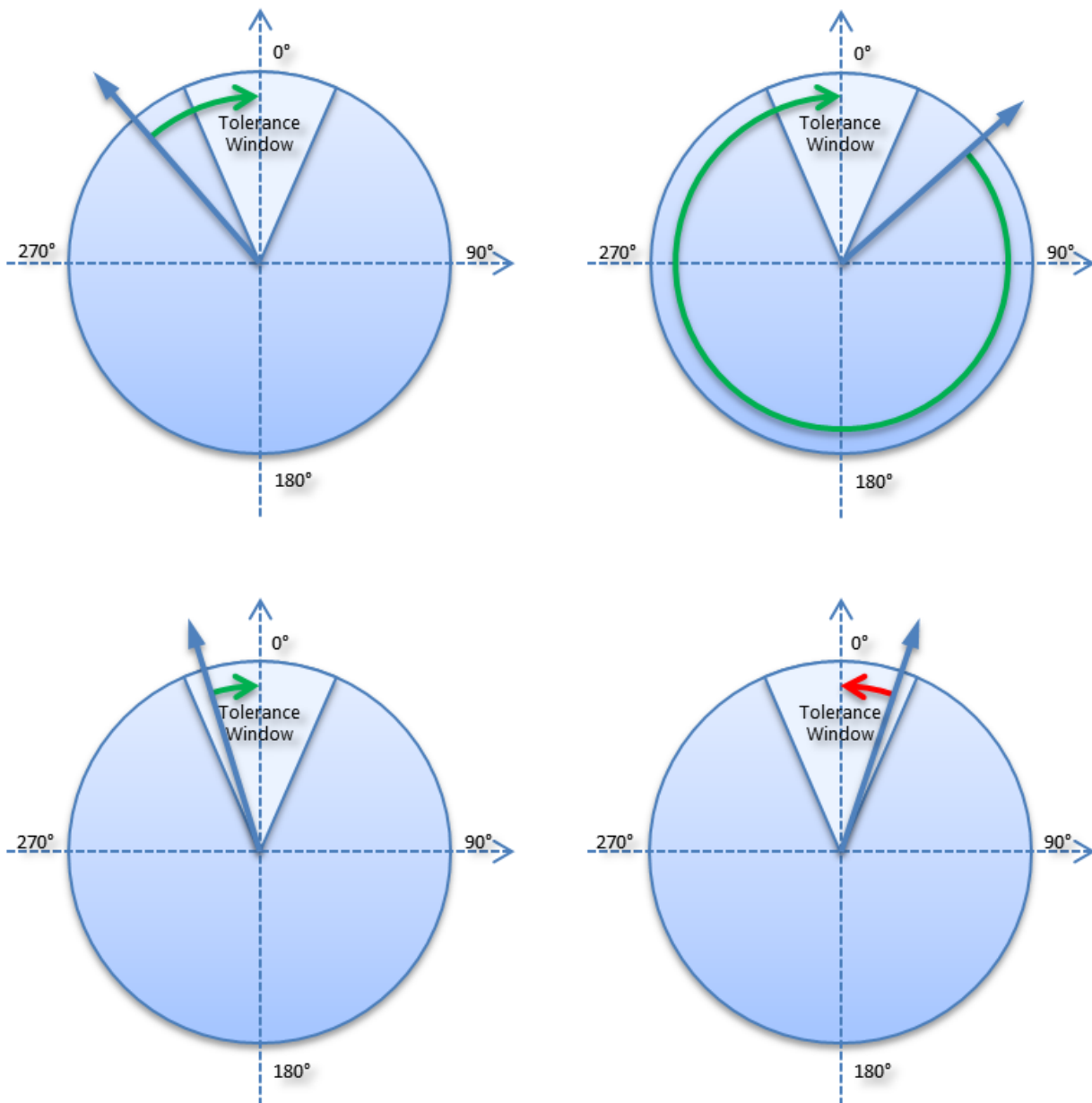
Position Modulus

Der Parameter Position Modulus definiert den Modulo Faktor für die Berechnung von Modulo Umdrehung und Modulo Position.

Position Modulo Tolerance Window

Der Parameter Position Modulo Tolerance Window definiert ein Positionsfenster um die aktuelle Modulo-Sollposition der Achse herum. Das Positionsfenster entspricht dem doppelten angegebenen Wert (Sollposition $\pm$ Positionsfenster).

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 `MC_MoveModulo.Direction = EModuloDirection.PositiveDirection` gewählt wurde. Dies kann insbesondere dann auftreten, wenn die Ist-Position ungenau bestimmt wird. 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 `MC_MoveModulo.Direction = EModuloDirection.ShortestDistance`), also auch entgegen der angegebenen Richtung, angefahren.



Zyklisches Interface

Die Soll- und Ist-Modulo-Position kann mit ins zyklische Interface aufgenommen werden.

Object	Settings	Init Parameter	Online Parameter	Data Area	Motion UI
	Area No	Enable	Name	Type	Size
-	0 (0)	☒	PlcToMc	InputDst	8
		☒	Std	POINTER TO MC.CDT_PLCTOMC_AXIS_STD	8.0 (Offs: 0.0)
-	1 (0)	☒	McToPlc	OutputSrc	32
		☒	Std	POINTER TO MC.CDT_MCTOPLC_AXIS_STD	8.0 (Offs: 0.0)
		☒	Set	POINTER TO MC.CDT_MCTOPLC_AXIS_SET	8.0 (Offs: 8.0)
		☒	Act	POINTER TO MC.CDT_MCTOPLC_AXIS_ACT	8.0 (Offs: 16.0)
		☒	Modulo	POINTER TO MC.CDT_MCTOPLC_AXIS_MODULO	8.0 (Offs: 24.0)
		☐	LoadControl	POINTER TO MC.CDT_MCTOPLC_AXIS_LOADCONTROL	8.0
		☐	FluidPower	POINTER TO MC.CDT_MCTOPLC_AXIS_FLUIDPOWER	8.0

PLC-Programmierung

Die Kommandierung einer MC3-Achse auf eine Modulo-Position erfolgt über den Funktionsbaustein `MC_MoveModulo` [► 278] der Tc3_Mc3Ptp-Bibliothek.

MC_MoveModulo.Position

Bei einer Modulo Positionierung darf die Ziel-Position nur kleiner gleich des parametrierten Position Modulus der Achse sein.

MC_MoveModulo.AdditionalTurns

Sollen vor dem Erreichen der Zielposition eine definierte Anzahl an Umdrehungen erfolgen, so sind diese über den Eingang `AdditionalTurns` vorzugeben.

MC_MoveModulo.Direction

Über den Eingang `Direction` vom Typ `EModuloDirection` aus der Tc3_Mc3Ptp Bibliothek wird die Bewegungsrichtung vorgegeben. Die möglichen Werte sind:

- `ShortestDistance`
- `PositiveDirection`
- `NegativeDirection`

5.3.6.4 Fehlerverhalten

Fehlertyp

Die TwinCAT MC3 unterscheidet zwischen *Execution Error* und *Control Error*.

Bei einem *Execution Error* hat die MC3 weiterhin die Kontrolle über die Achse. Die Kommunikation zur Hardware sowie der Antrieb selbst sind fehlerfrei. Die Sollwertgenerierung ist weiterhin möglich. In diesem Fall ist die Standardreaktion im Fehlerfall eine kontrollierte Stopprampe, mit der für die Achse konfigurierten maximalen Dynamik. Dieser Stopp kann mit `MC_Reset` abgebrochen werden. Nach dem Reset kann im selben Zyklus ein neuer Befehl angestoßen werden.

Bei einem *Control Error* hat die MC3 keine Kontrolle mehr über die Achse, typischerweise aufgrund eines Hardware- oder antriebsseitigen Fehlers. In diesem Zustand wird die Reaktion durch die Parameter der Achse definiert. Derzeit ist ein konsequenter Stopp mit Null-Dynamik möglich (HardStop).

Fehlerfortpflanzung

Die Funktionsbausteine für Kopplungen bieten spezielle Einstellungen zur Steuerung der Fehlerbehandlung zwischen den Achsen. Der Mechanismus ist für MC_GearIn, MC_GearInPos und MC_CamIn identisch.

ReactionToMasterError definiert das Verhalten der Slave-Achse, wenn die Master-Achse in einen Fehlerzustand übergeht. Bei der Einstellung *TriggerSlaveError* löst jeder Master-Fehler sofort auch einen Fehler in der Slave-Achse aus. Zudem wird die Kopplung aufgelöst. Die Slave-Achse folgt dann der Standard-MC3-Reaktion für den spezifischen Fehlertyp (s. o.). Bei der Einstellung *KeepCouplingWithSetValues* bleibt die Kopplung auch dann aktiv, wenn die Master-Achse einen Fehler aufweist. Die Slave-Achse generiert weiterhin Sollwerte aus den Sollwerten der Master-Achse. Dieser Modus ermöglicht ein koordiniertes Anhalten unter Beibehaltung der synchronen Beziehung.

Die umgekehrte Richtung der Propagation wird mit *ReactionToSlaveError* konfiguriert. Bei der Einstellung *TriggerMasterError* leitet die Slave-Achse ihren Fehler an ihre Master-Achse weiter. So geht auch die Master-Achse in ihre Fehlerreaktion über. Bei der Einstellung *NoReaction* bleibt die Master-Achse unbeeinträchtigt. Die Slave-Achse und der Kopplungs-Baustein signalisieren den Fehlerzustand. In beiden Fällen wird die Kopplung für diese Slave-Achse aufgehoben.

Mit Ausnahme des oben beschriebenen expliziten Falls *KeepCouplingWithSetValues* werden die Master- und Slave-Achsen automatisch entkoppelt, wenn auf einer der beiden Seiten ein Fehler auftritt. Nachdem der zugrunde liegende Fehler mit MC_Reset quittiert wurde, muss die Anwendung den gewünschten Kopplungsbefehl erneut ausführen, um den gekoppelten Betrieb fortzusetzen. Anhand der Diagnoseinformationen an dem Funktionsbaustein (*Error* und *ErrorID* sowie *MasterHasError*) kann festgestellt werden, ob der Fehler in der Master- oder der Slave-Achse entstanden ist.

Ein propagierter Fehler ist auf der Empfängerseite immer ein *Execution Error*.

Fehlerfortpflanzung – Beispiele

Grundsätzlich gilt bei einem Fehler der Slave-Achse:

- Slave-Achse: Die Kopplung wird aufgelöst und die Reaktion ist abhängig vom Fehlertypen.
- Master-Achse: Die Master-Achse meldet einen Fehler und folgt einer Stopprampe der MC oder führt ihre laufende Bewegung ohne Beeinflussung fort.

Bei einem Fehler der Master-Achse gilt:

- Master-Achse: Reaktion ist abhängig vom Fehlertypen.
- Slave-Achse: Die Slave-Achse meldet einen Fehler und folgt einer Stopprampe der MC oder den Sollwerten der Master-Achse.

Szenario 1

Einstellungen am Funktionsbaustein und vorliegende Fehlerart:

ReactionToMasterError	ReactionToSlaveError	Error Type
TriggerSlaveError	TriggerMasterError	Control Error

Reaktion auf Fehler:

Master Fehler	Slave Fehler
• Null-Dynamik Master • kontrollierte Stopprampe der Slave-Achse • Kopplung aufgelöst	• Null-Dynamik Slave • kontrollierte Stopprampe der Master-Achse • Kopplung aufgelöst

Szenario 2

Einstellungen am Funktionsbaustein und vorliegende Fehlerart:

ReactionToMasterError	ReactionToSlaveError	Error Type
TriggerSlaveError	NoReaction	Control Error

Reaktion auf Fehler:

Master Fehler	Slave Fehler
• Null-Dynamik Master • kontrollierte Stopprampe der Slave-Achse • Kopplung aufgelöst	• Null-Dynamik Slave • Master-Achse nicht beeinflusst • Kopplung aufgelöst

Szenario 3

Einstellungen am Funktionsbaustein und vorliegende Fehlerart:

ReactionToMasterError	ReactionToSlaveError	Error Type
KeepCouplingWithSetValues	TriggerMasterError	Control Error

Reaktion auf Fehler:

Master Fehler	Slave Fehler
• Null-Dynamik Master • Slave-Achse versucht den Sollwerten der Master-Achse zu folgen • Kopplung beibehalten	• Null-Dynamik Slave • kontrollierte Stopprampe der Master-Achse • Kopplung aufgelöst

Szenario 4

Einstellungen am Funktionsbaustein und vorliegende Fehlerart:

ReactionToMasterError	ReactionToSlaveError	Error Type
KeepCouplingWithSetValues	NoReaction	Control Error

Reaktion auf Fehler:

Master Fehler	Slave Fehler
• Null-Dynamik Master • Slave-Achse versucht den Sollwerten der Master-Achse zu folgen • Kopplung beibehalten	• Null-Dynamik Slave • Master-Achse nicht beeinflusst • Kopplung aufgelöst

Szenario 5

Einstellungen am Funktionsbaustein und vorliegende Fehlerart:

ReactionToMasterError	ReactionToSlaveError	Error Type
TriggerSlaveError	TriggerMasterError	Execution Error

Reaktion auf Fehler:

Master Fehler	Slave Fehler
• kontrollierte Stopprampe der Master-Achse • kontrollierte Stopprampe der Slave-Achse • Kopplung aufgelöst	• kontrollierte Stopprampe der Slave-Achse • kontrollierte Stopprampe der Master-Achse • Kopplung aufgelöst

Szenario 6

Einstellungen am Funktionsbaustein und vorliegende Fehlerart:

ReactionToMasterError	ReactionToSlaveError	Error Type
TriggerSlaveError	NoReaction	Execution Error

Reaktion auf Fehler:

Master Fehler	Slave Fehler
- kontrollierte Stopprampe der Master-Achse - kontrollierte Stopprampe der Slave-Achse - Kopplung aufgelöst	- kontrollierte Stopprampe der Slave-Achse - Master-Achse nicht beeinflusst - Kopplung aufgelöst

Szenario 7

Einstellungen am Funktionsbaustein und vorliegende Fehlerart:

ReactionToMasterError	ReactionToSlaveError	Error Type
KeepCouplingWithSetValues	TriggerMasterError	Execution Error

Reaktion auf Fehler:

Master Fehler	Slave Fehler
- kontrollierte Stopprampe der Master-Achse - Slave-Achse versucht den Sollwerten der Master-Achse zu folgen - Kopplung beibehalten	- kontrollierte Stopprampe der Slave-Achse - kontrollierte Stopprampe der Master-Achse - Kopplung aufgelöst

Szenario 8

Einstellungen am Funktionsbaustein und vorliegende Fehlerart:

ReactionToMasterError	ReactionToSlaveError	Error Type
KeepCouplingWithSetValues	NoReaction	Execution Error

Reaktion auf Fehler:

Master Fehler	Slave Fehler
- kontrollierte Stopprampe der Master-Achse - Slave-Achse versucht den Sollwerten der Master-Achse zu folgen - Kopplung beibehalten	- kontrollierte Stopprampe der Slave-Achse - Master-Achse nicht beeinflusst - Kopplung aufgelöst

5.4 Tc3_Mc3Camming

5.4.1 Tc3_Mc3Camming

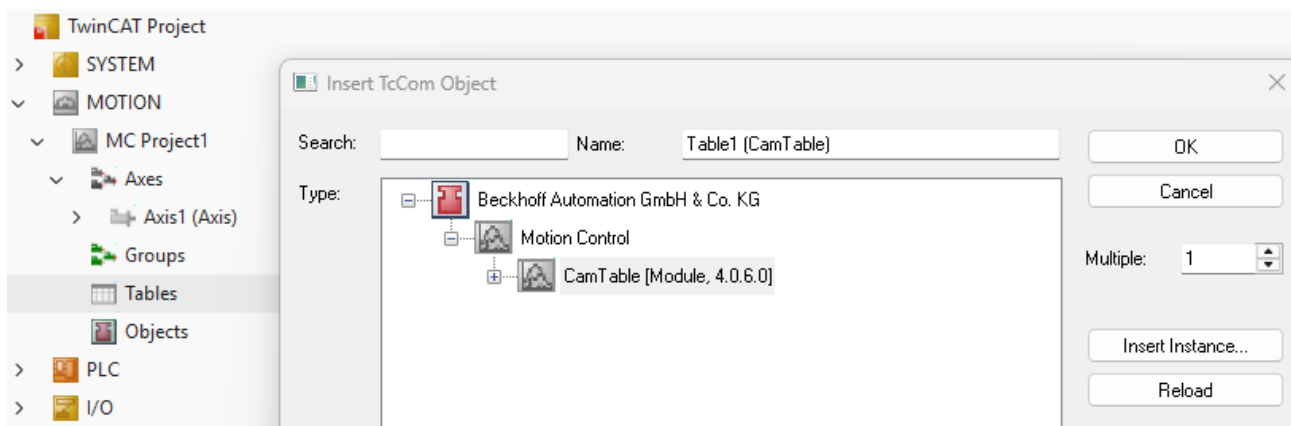
5.4.1.1 Kurvenscheibe erstellen

Die Funktionen und die Handhabung rund um Kurvenscheiben sind bei der MC3 zentral im Funktionsbaustein MC_CamObject [► 339] gebündelt. Zum einen können Sie in der Tc3_Mc3Camming-Bibliothek auf bekannte Funktionsbausteine aus der PLCopen zurückgreifen. Zum anderen finden Sie im MC_CamObject Methoden zur Erstellung und Modifikation von Kurvenscheiben sowie auslesbare Eigenschaften der aktiven Kurvenscheibe.

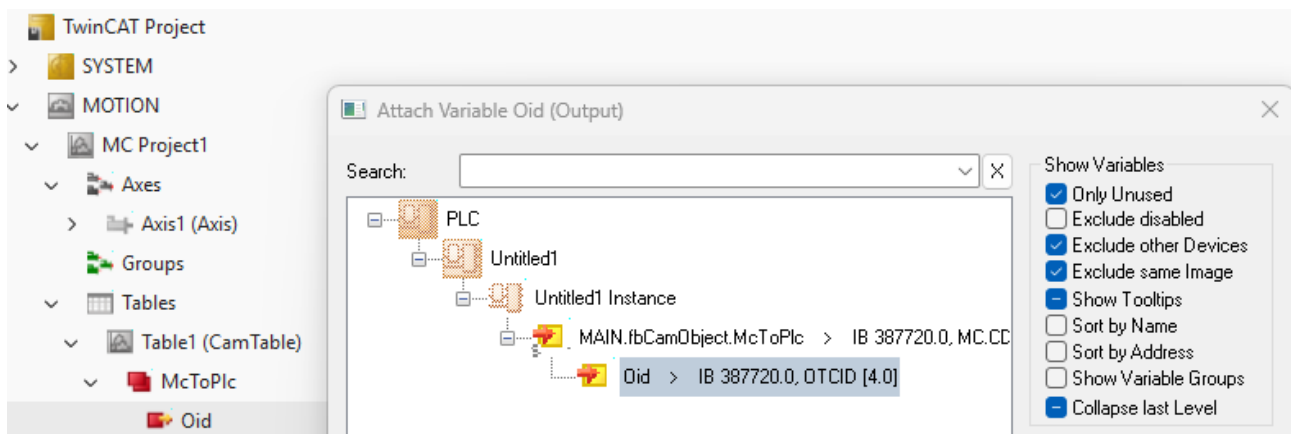
✓ Grundsätzlich kann eine Kurvenscheibe zur Konfigurationszeit im Projekt vorkonfiguriert werden.

1. Öffnen Sie **Axes > Tables**.

2. Machen Sie einen Rechtsklick auf die Tabelle im Strukturbaum und öffnen Sie den Dialog **Insert TcCom Object**.



⇒ Die Objekt-ID des Moduls bzw. TcCOM-Objekts muss in diesem Fall mit einem MC_CamObject in der PLC verknüpft werden.



⇒ Zur Laufzeit erfolgt die Initialisierung dieser Verbindung anschließend über die Methode MC_CamObject.Connect() [► 340].

⇒ Alternativ kann eine Kurvenscheibe auch zur Laufzeit angelegt werden. Durch einen Aufruf der MC_CamObject.Create() [► 341] Methode aus der PLC heraus wird ein TcCOM Modul angelegt. Eine Verknüpfung und weitere Initialisierung sind in diesem Fall nicht notwendig.

Kurvenscheibe mit MotionFunctionPoint als Stützstellen füllen

Nach dem erfolgreichen Erstellen eines Kurvenscheiben-Objekts wird dieses im Anschluss mit den gewünschten Datenpunkten und Funktionen zur Interpolation gefüllt. Die Definition eines einzelnen Datenpunkte ist dabei über die Struktur MotionFunctionPoint gegeben. Mithilfe der Methode MC_CamObject.WriteData() [► 341] kann eine Reihe von Datenpunkten (Array aus MotionFunctionPoints)

synchron in das Kurvenscheiben-Objekt geladen werden. Mit Abschluss dieser Methode ist die grundlegende Konfiguration einer Kurvenscheibe vollzogen und Achsen können über die hinterlegte Vorschrift miteinander gekoppelt werden.

```

PROGRAM MAIN
VAR
  fbCamObject      : MC_CamObject;
  fbCamObject2     : MC_CamObject;
  motionFunctionPoints : ARRAY[0..360] OF MotionFunctionPoint;
  bInit           : BOOL;
  i               : INT;
END_VAR

IF NOT bInit THEN

  fbCamObject.Connect();

  FOR i:=0 TO 360 DO
    motionFunctionPoints[i].Ignored      := FALSE;
    motionFunctionPoints[i].FunctionType := EMotionFunctionType.CubicSpline;
    motionFunctionPoints[i].MasterPosition := i;
    motionFunctionPoints[i].SlavePosition := -(COS(INT_TO_LREAL(i) / 360.0 * 2 * PI) - 1) * 100;
    motionFunctionPoints[i].SlaveVelocity := STRING_TO_LREAL('#Ignore');
    motionFunctionPoints[i].SlaveAcceleration := STRING_TO_LREAL('#Ignore');
    motionFunctionPoints[i].SlaveJerk      := STRING_TO_LREAL('#Ignore');
  END_FOR

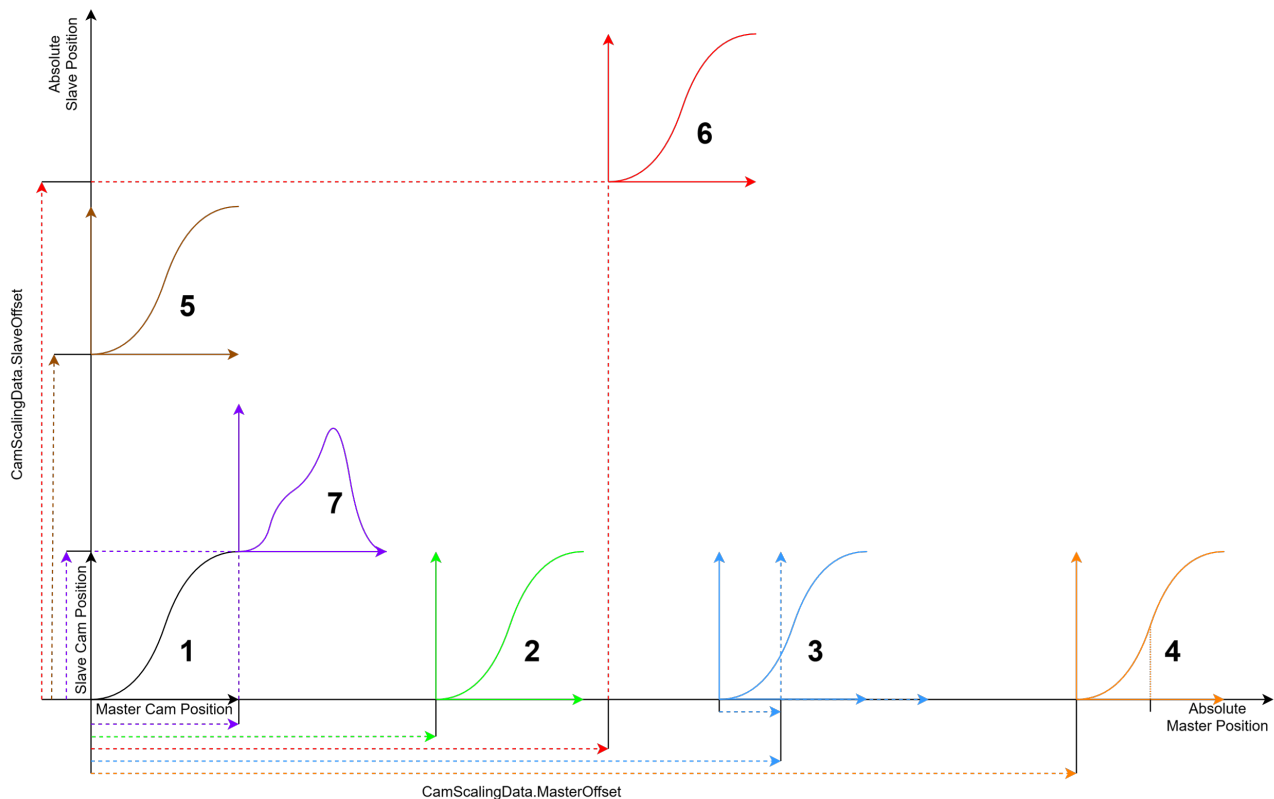
  fbCamObject.WriteData(SourceData := ADR(motionFunctionPoints), NumberOfPoints := 361,
    Periodic:=FALSE);
  fbCamObject2.Create();
  fbCamObject2.CopyFrom(fbCamObject);

  bInit:=TRUE;
END_IF

```

5.4.1.2 CamScalingData: Master-/SlaveOffsetMode verwenden

Beispiele für die Verwendung von ECamOffsetMode.Absolute/Relative in Kombination mit CamActivationMode



1. Kein Offset, instantane Aktivierung/Umschaltung

- MasterOffsetMode = None, SlaveOffsetMode = None, ActivationMode = Instantaneous

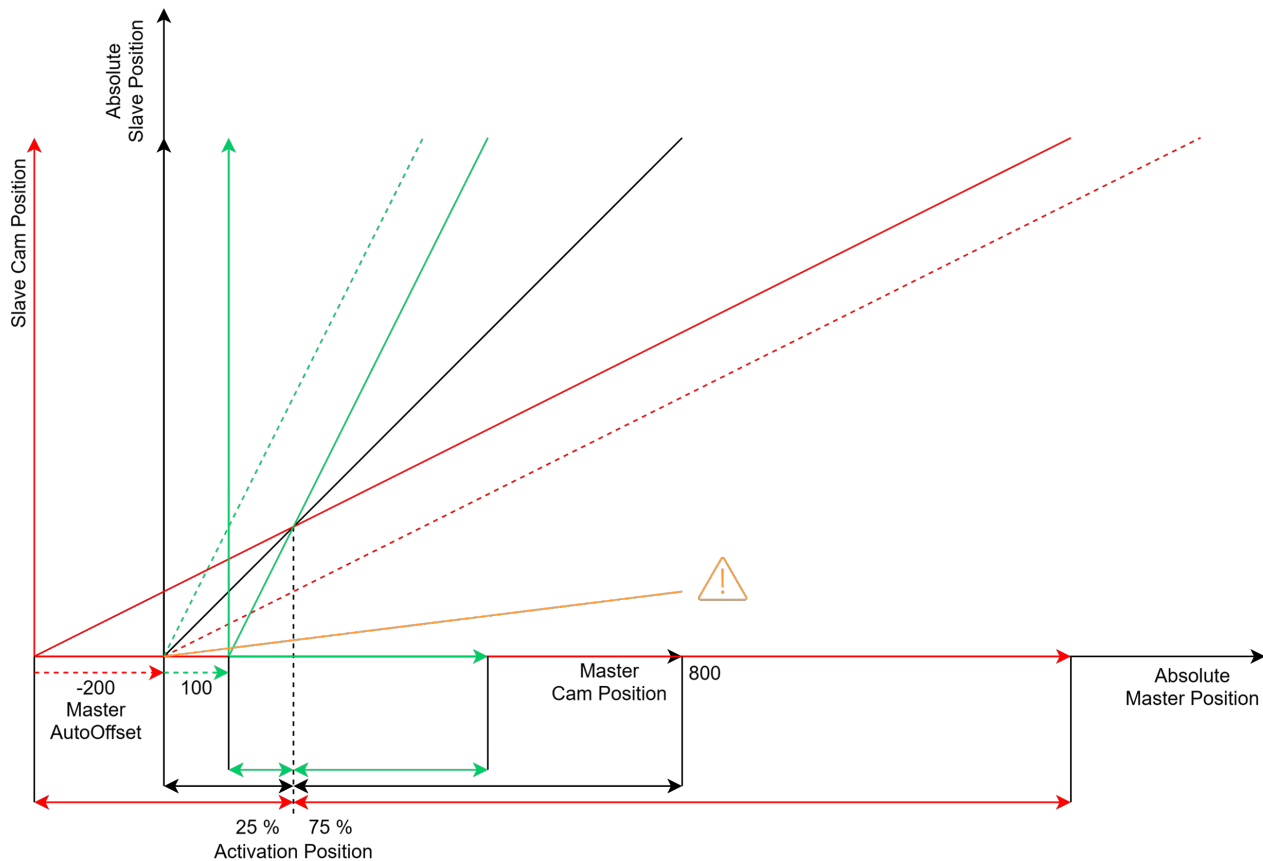
- MasterOffsetMode = None, SlaveOffsetMode = None, ActivationMode = AtMasterCamPosition, ActivationPosition = 0.0
 - MasterOffsetMode = None, SlaveOffsetMode = None, ActivationMode = AtMasterAbsolutePosition, ActivationPosition = 0.0
2. Master-Offset (SlaveOffsetMode = None)
- MasterOffsetMode = Absolute, MasterOffset = 100, ActivationMode = Instantaneous
 - MasterOffsetMode = Absolute, MasterOffset = 100, ActivationMode = AtMasterAbsolutePosition, ActivationPosition = 100
 - MasterOffsetMode = Absolute, MasterOffset = 100, ActivationMode = AtMasterCamPosition, ActivationPosition = 0
 - MasterOffsetMode = Relative, MasterOffset = 0, ActivationMode = AtMasterAbsolutePosition, ActivationPosition = 100
3. Master-Offset relativ
- MasterOffsetMode = Relative, MasterOffset = -10, ActivationMode = AtMasterAbsolutePosition, ActivationPosition = 200
 - An der gestrichelten Linie springt die Slave-Position. Dies ist der Zeitpunkt, zu dem die Kurvenscheibe aktiv wird.
4. Master-Offset mit ActivationPosition (empfohlen, wenn der erste Abschnitt einer Kurvenscheibe nicht betrachtet werden soll)
- MasterOffsetMode = Absolute, MasterOffset = 300, ActivationMode = AtMasterCamPosition, ActivationPosition = 10
 - MasterOffsetMode = Absolute, MasterOffset = 300, ActivationMode = AtMasterAbsolutePosition, ActivationPosition = 310
 - An der gestrichelten Linie springt die Slave-Position. Dies ist der Zeitpunkt, zu dem die Kurvenscheibe aktiv wird.
5. Slave-Offset (MasterOffsetMode = None)
- SlaveOffsetMode = Absolute, SlaveOffset = 100, ActivationMode = Instantaneous
 - SlaveOffsetMode = Relative, SlaveOffset = 100, ActivationMode = Instantaneous
6. Master- und Slave-Offset
- MasterOffsetMode = Absolute, MasterOffset = 150, SlaveOffsetMode = Absolute, SlaveOffset = 150, ActivationMode = Instantaneous
 - MasterOffsetMode = Relative, MasterOffset = 150, SlaveOffsetMode = Relative, SlaveOffset = 150, ActivationMode = Instantaneous
7. Kein Offset, Aktivierung/Umschaltung zum Ende/Start einer Kurvenscheibe
- MasterOffsetMode = None, SlaveOffsetMode = None, ActivationMode = AtBoundary

ECamOffsetMode.AutoOffset verwenden

Der ECamOffsetMode.AutoOffset Modus sorgt für eine automatische Anpassung des Kurvenscheibenoffsets. ECamOffsetMode.AutoOffset kann unabhängig für die Master- und die Slave-Achse einer Kurvenscheibe angewendet werden. Voraussetzung zur Nutzung des AutoOffset ist eine bereits aktive Kurvenscheibe. Somit kann der Modus AutoOffset bei einer Umschaltung oder Skalierung von Kurvenscheiben verwendet werden.

Im Vergleich zu TF5050 TwinCAT 3 NC Camming mit MC_CamScalingMode und MC_StartMode, sind diese Einstellungen in TF5550 TwinCAT 3 MC3 Camming im Aufzählungstypen ECamOffsetMode zusammengefasst. Mit dem ECamOffsetMode.AutoOffset kann ein Sprung in der Position der Master- und/oder Slave-Achse verhindert werden.

Beispiele für die Verwendung von ECamOffsetMode.AutoOffset für die Master-Achse



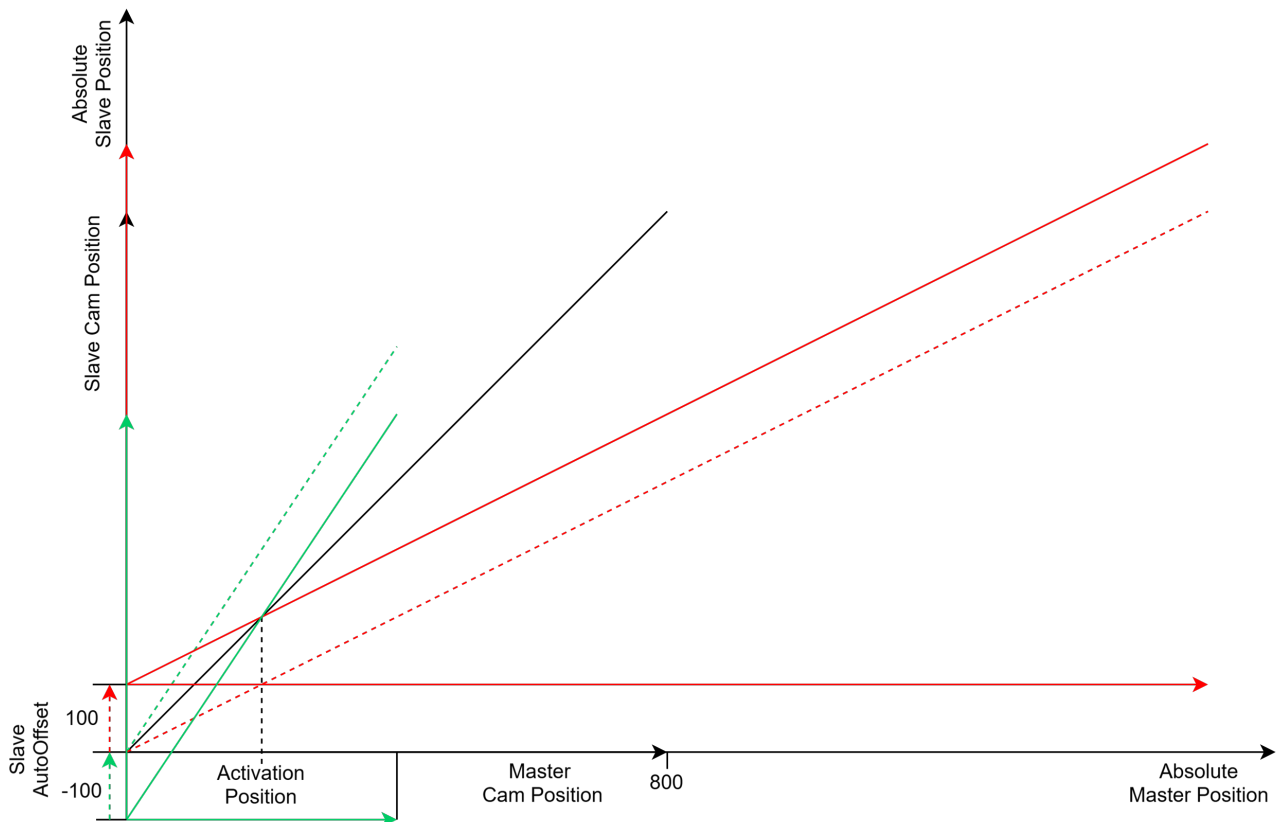
Die Ausgangssituation in der gezeigten Grafik ist die Kurvenscheibe in schwarzer Farbe. Das Ziel ist eine Umschaltung auf die rote, grüne bzw. orangene Kurvenscheibe. Die originale Lage der jeweiligen Kurvenscheiben (ohne Offsetverschiebung) ist mit einer gestrichelten Linie, die resultierende Lage mit einer durchgezogenen Linie in entsprechender Farbe eingezeichnet. Die Umschaltung erfolgt an einer definierten Activation Position (in der Grafik an der Master-Cam-Position 200).

ECamOffsetMode.AutoOffset für die Master-Achse bestimmt den Master-Offset der Kurvenscheibe so, dass die Masterposition innerhalb der Kurvenscheibe beibehalten wird. Bei einer Master-Skalierung bzw. Umschaltung auf eine Kurvenscheibe mit anderem Master-Zyklus bedeutet das, dass die relative (prozentuale) Position vor und nach der Umschaltung identisch ist. In diesem Fall ergibt sich für die rote Kurvenscheibe ein automatischer Master-Offset von -200 und für die grüne Kurvenscheibe ein Offset von 100.

Die rote sowie grüne Kurvenscheibe ermöglichen eine Verschiebung, sodass auch in der Slave-Position keine Unstetigkeiten auftreten.

Im Beispiel der orangefarbenen Kurvenscheibe findet keine Verschiebung bzw. automatische Anpassung des Offsets statt. Die Master-Skalierung bzw. Master-Periode stimmt zwischen der schwarzen und orangefarbenen Kurvenscheibe überein. In diesem Fall wird die Master-Position durch den Modus AutoOffset beibehalten. Zu beachten ist, dass die Slave-Position zum Zeitpunkt der Umschaltung springt.

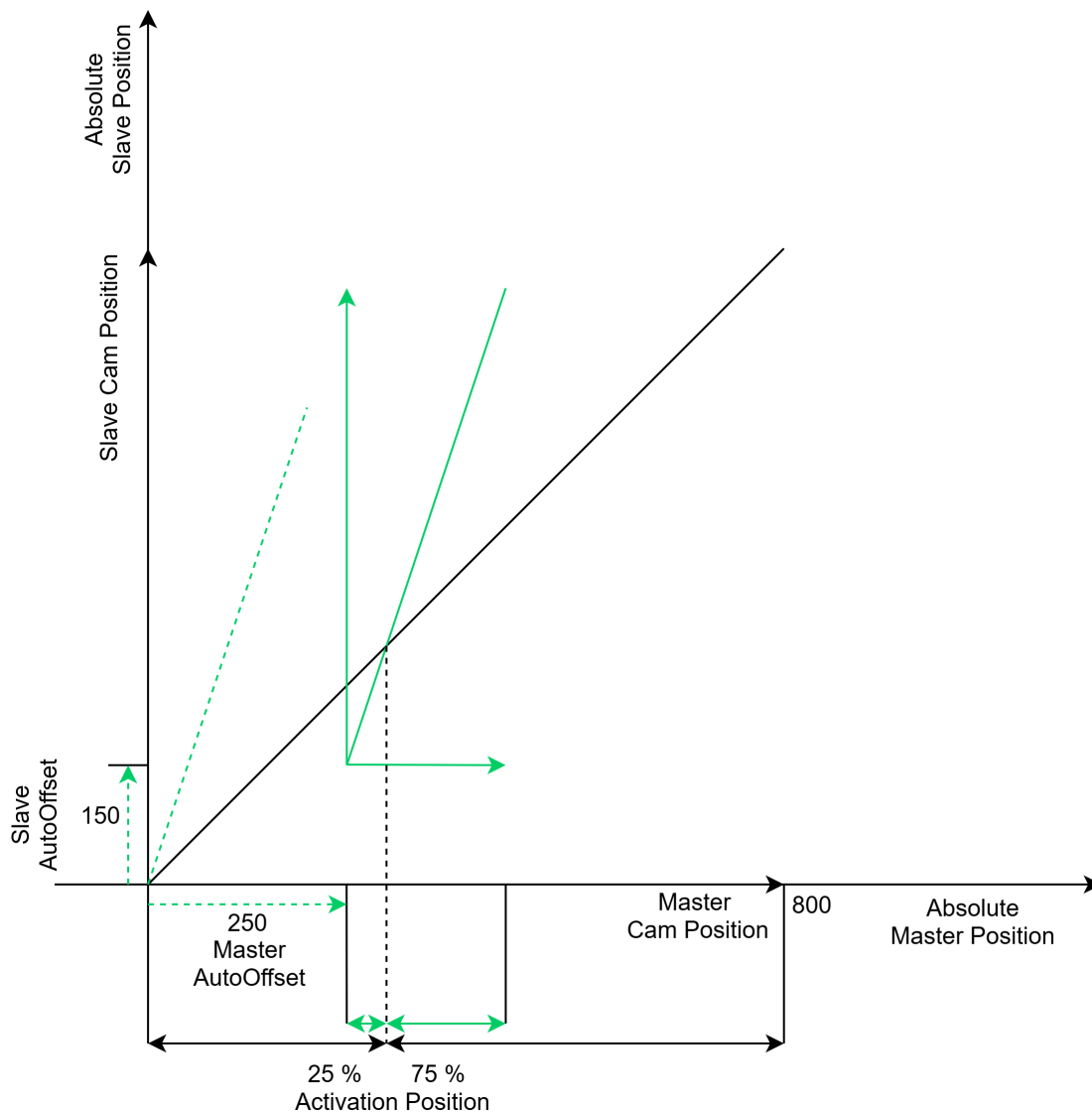
Beispiele für die Verwendung von ECamOffsetMode.AutoOffset für die Slave-Achse



ECamOffsetMode.AutoOffset für die Slave-Achse berechnet den Offset so, dass es durch eine Kurvenscheiben-Umschaltung oder durch eine Skalierung zu keinem Sprung in der Slave-Position kommt. Der Offset der Kurvenscheibe wird also so angepasst, dass die Slave-Position vor und nach der Aktion identisch ist.

Beispiele für die Verwendung von ECamOffsetMode.AutoOffset für die Master- und Slave-Achse

Wird ECamOffsetMode.AutoOffset sowohl für die Master- als auch die Slave-Achse bei einer Kurvenscheibenumschaltung oder Skalierung verwendet, so wird zuerst der Master-Offset und anschließend der Slave-Offset berechnet.



5.4.2 Data Types

5.4.2.1 Enums

5.4.2.1.1 ECamActivationMode

Defines when a cam table activation or exchange takes effect.

Syntax

Definition:

```

TYPE ECamActivationMode :
(
    Instantaneous           := 0,
    AtBoundary              := 1,
    AtMasterCamPosition     := 2,
    AtMasterAbsolutePosition := 3
) UNTIL;
END TYPE

```

Values

Name	Description
Instantaneous	Activation occurs during the next MET cycle.
AtBoundary	Activation occurs when either the start or end of the cam table is crossed.
AtMasterCamPosition	Activation occurs when the master cam position crosses the activation position. Activation position must be within the limits of the cam table.
AtMasterAbsolutePosition	Activation occurs when master axis position crosses the activation position.

Weiterführende Informationen

`ECamActivationMode` bestimmt die Art, wie das betreffende Kommando verarbeitet bzw. aktiv wird. Entweder wird der Zeitpunkt festgelegt oder die Aktivierung erfolgt an einer Position, die durch einen weiteren Eingang spezifiziert wird.

`ECamActivationMode.AtMasterCamPos`

- Eine Kombination mit `CamScalingData.MasterOffsetMode = ECamOffsetMode.Relative` ist möglich, wenn die `ActivationPosition` aus einer bereits aktiven Kurvenscheibenkopplung bestimmt wird.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.2.1.2 ECamOffsetMode

Defines how master and slave offsets are applied during cam activation.

Syntax

Definition:

```

TYPE ECamOffsetMode :
(
    None      := 0,
    Absolute  := 1,
    Relative   := 2,
    AutoOffset := 3
) UINT;
END_TYPE

```

Values

Name	Description
None	Ignores input factors and takes default ones (Offset = 0, Scale = 1).
Absolute	Cam table and axis coordinate systems are deemed to be aligned. Offset and Scale are not affected.
Relative	Offset is calculated such the origin of the cam table coordinate system is aligned to the axis position at the activation time. If the input offset is not 0, it is added. Scale is not affected.
AutoOffset	Input offset is ignored and a new offset is calculated so that no position jump occurs during activation. Scale is not affected.

Weiterführende Informationen

Mit diesem Aufzählungstyp kann angegeben werden, wie ein Positionsoffset für die Skalierung einer Kurvenscheibe zu interpretieren ist.

ECamOffsetMode.Relative:

- Die Ausrichtung des Koordinatensystems der Kurvenscheibe gegenüber den absoluten Master- und Slave-Positionen muss bekannt sein. Eine Kombination mit ECamActivationMode.AtMasterCamPos ist möglich, wenn die ActivationPosition aus einer bereits aktiven Kurvenscheibenkopplung bestimmt wird.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.2.1.3 EMotionFunctionType

Interpolation function type for cam table segments.

Syntax

Definition:

```
TYPE EMotionFunctionType :
(
    Undefined           := 0,
    Linear              := 1,
    Polynomial3         := 2,
    Polynomial5         := 3,
    Polynomial7         := 4,
    Sinusoid            := 5,
    SinusoidCombination := 6,
    InclinedSinusoid    := 7,
    ModifiedSinusoid    := 8,
    HarmonicCombination := 9,
    AccelerationTrapezoid := 10,
    CubicSpline         := 11
) UINT;
END_TYPE
```

Values

Name	Description
Undefined	No function type defined. Segment is not valid.
Linear	Linear interpolation between segment points.
Polynomial3	3rd-order polynomial interpolation.
Polynomial5	5th-order polynomial interpolation.
Polynomial7	7th-order polynomial interpolation.
Sinusoid	Simple sinusoidal motion law.
SinusoidCombination	Combined sinusoidal motion law.
InclinedSinusoid	Bestehorn sinusoidal motion law for minimal peak dynamic loads.
ModifiedSinusoid	Modified sinusoidal motion law.
HarmonicCombination	Harmonic combination motion law.
AccelerationTrapezoid	Trapezoidal acceleration profile motion law.
CubicSpline	Cubic spline interpolation through the segment points.

Weiterführende Informationen

`EMotionFunctionType` gibt an, nach welcher mathematischen Vorschrift zwischen zwei benachbarten Stützstellen interpoliert wird. Anhand der gewählten Dynamiken/Randbedingungen wird automatisch bestimmt, ob es sich um eine Rast nach Rast (RR), Geschwindigkeit nach Geschwindigkeit (VV) usw. Bewegung handelt.

Übertragen des MotionFunctionTypes von NC2 zu MC3

MC_MotionFunctionType[...]	EMotionFunctionType[...]
MOTIONFUNCTYPE_NOTDEF	Undefined
MOTIONFUNCTYPE_POLYNOM1	Linear
MOTIONFUNCTYPE_POLYNOM3	Polynom3
MOTIONFUNCTYPE_POLYNOM5	Polynom5
MOTIONFUNCTYPE_POLYNOM7	Polynom7
MOTIONFUNCTYPE_POLYNOM8	On request
MOTIONFUNCTYPE_POLYNOM9	On request
MOTIONFUNCTYPE_SINUSLINIE	Sinusoid
MOTIONFUNCTYPE_MODSINUSLINIE	ModSinusoid
MOTIONFUNCTYPE_BESTEHOORN	InclinedSinusoid
MOTIONFUNCTYPE_BESCHLTRAPEZ	AccTrapezoid
MOTIONFUNCTYPE_POLYNOM5_MM	Polynom5
MOTIONFUNCTYPE_HARMONIC_KOMBI_RT	HarmonicCombi
MOTIONFUNCTYPE_HARMONIC_KOMBI_TR	HarmonicCombi
MOTIONFUNCTYPE_HARMONIC_KOMBI_VT	HarmonicCombi
MOTIONFUNCTYPE_HARMONIC_KOMBI_TV	HarmonicCombi
MOTIONFUNCTYPE_BESCHLTRAPEZ_RT	AccTrapezoid
MOTIONFUNCTYPE_BESCHLTRAPEZ_TR	AccTrapezoid
MOTIONFUNCTYPE_MODSINUSLINIE_VV	ModSinusoid
MOTIONFUNCTYPE_POLYNOM7_MM	Polynom7
MOTIONFUNCTYPE_BESCHLTRAPEZ_RV	AccTrapezoid
MOTIONFUNCTYPE_BESCHLTRAPEZ_VR	AccTrapezoid
MOTIONFUNCTYPE_POLYNOM3_VV	Polynom3
MOTIONFUNCTYPE_POLYNOM11	OnRequest
MOTIONFUNCTYPE_PEISEKAH11	OnRequest
MOTIONFUNCTYPE_SPLINE	CubicSpline
MOTIONFUNCTYPE_SPLINE_NATURAL	CubicSpline (transferable via boundary conditions)
MOTIONFUNCTYPE_SPLINE_TANGENTIAL	CubicSpline (transferable via boundary conditions)
MOTIONFUNCTYPE_SPLINE_PERIODIC	CubicSpline (transferable via boundary conditions)
MOTIONFUNCTYPE_SPLNE_LINEAR	CubicSpline (transferable via boundary conditions)

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.2.2 Structs

5.4.2.2.1 CamActivationOptions

Options for cam table activation timing and reference.

Syntax

Definition:

```

TYPE CamActivationOptions :
STRUCT
    Mode           : ECamActivationMode;
    Position       : LREAL;
    ReferenceObject : Type_CouplingFunction1D;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
Mode	ECamActivationMode [► 320]	Instantaneous	Activation mode defining when the cam operation takes effect.
Position	LREAL	#Ignore	Master absolute position or master cam position at which the activation occurs. Only relevant for position-based activation modes.
ReferenceObject	Type_CouplingFunction1D [► 329]		Reference cam object for configured the activation options.

Weiterführende Informationen

Diese Struktur enthält die wichtigsten Eigenschaften einer Kurvenscheibe und kann über die Methode MC_CamObject.ReadCharacteristics() ausgelesen werden.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.2.2.2 CamCharacteristics

Computed characteristics of a cam table describing its motion profile properties.

Syntax

Definition:

```

TYPE CamCharacteristics :
STRUCT
    NominalMasterVelocity : LREAL;
    StartPoint            : CamProfilePoint;
    EndPoint              : CamProfilePoint;
    PointAtMinSlavePosition : CamProfilePoint;
    PointAtMinSlaveVelocity : CamProfilePoint;
    PointAtMinSlaveAcceleration : CamProfilePoint;
    PointAtMinSlaveJerk : CamProfilePoint;
    PointAtMaxSlavePosition : CamProfilePoint;
    PointAtMaxSlaveVelocity : CamProfilePoint;
    PointAtMaxSlaveAcceleration : CamProfilePoint;
    PointAtMaxSlaveJerk : CamProfilePoint;
    SlaveMeanAbsoluteVelocity : LREAL;
    SlaveEffectiveAcceleration : LREAL;
    Oid : OTCID;
    NumberOfPoints : UDINT;
    NumberOfActivePoints : UDINT;
    NumberOfActiveSegments : UDINT;
    IsValid : BOOL;
    IsPeriodic : BOOL;
    MinContinuityRequirement : EContinuity;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
NominalMasterVelocity	LREAL	0.0	Nominal master velocity used to convert dimensionless slave values to time-based values.
StartPoint	CamProfilePoint [► 326]		Cam profile point at the start of the table.
EndPoint	CamProfilePoint [► 326]		Cam profile point at the end of the table.
PointAtMinSlavePosition	CamProfilePoint [► 326]		Cam profile point where the slave position reaches its minimum.
PointAtMinSlaveVelocity	CamProfilePoint [► 326]		Cam profile point where the slave velocity reaches its minimum.
PointAtMinSlaveAcceleration	CamProfilePoint [► 326]		Cam profile point where the slave acceleration reaches its minimum.
PointAtMinSlaveJerk	CamProfilePoint [► 326]		Cam profile point where the slave jerk reaches its minimum.
PointAtMaxSlavePosition	CamProfilePoint [► 326]		Cam profile point where the slave position reaches its maximum.
PointAtMaxSlaveVelocity	CamProfilePoint [► 326]		Cam profile point where the slave velocity reaches its maximum.
PointAtMaxSlaveAcceleration	CamProfilePoint [► 326]		Cam profile point where the slave acceleration reaches its maximum.
PointAtMaxSlaveJerk	CamProfilePoint [► 326]		Cam profile point where the slave jerk reaches its maximum.
SlaveMeanAbsoluteVelocity	LREAL	0.0	Mean absolute velocity of the slave over the entire cam table range.
SlaveEffectiveAcceleration	LREAL	0.0	RMS (effective) acceleration of the slave over the entire cam table range.
Oid	OTCID	0	Object ID of the cam table these characteristics belong to.
NumberOfPoints	UDINT	0	Total number of profile points in the cam table.
NumberOfActivePoints	UDINT	0	Number of non-ignored profile points in the cam table.
NumberOfActiveSegments	UDINT	0	Number of active interpolation segments between the profile points.
IsValid	BOOL	false	TRUE if the characteristics have been computed successfully.

Name	Type	Default	Description
IsPeriodic	BOOL	false	TRUE if the cam table is periodic (start and end boundary conditions match).
MinContinuityRequirement	EContinuity	PositionVelocityAcceleration	Minimum continuity level required for smooth operation of the cam table.

Weiterführende Informationen

Diese Struktur enthält die wichtigsten Eigenschaften einer Kurvenscheibe und kann über die Methode MC_CamObject.ReadCharacteristics() ausgelesen werden.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.2.2.3 CamProfilePoint

Describes a point on a cam profile with master position and slave motion state.

Syntax

Definition:

```

TYPE CamProfilePoint :
STRUCT
    MasterPosition      : LREAL;
    SlavePosition       : LREAL;
    SlaveVelocity       : LREAL;
    SlaveAcceleration   : LREAL;
    SlaveJerk           : LREAL;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
MasterPosition	LREAL	#Invalid	Master cam position at this profile point.
SlavePosition	LREAL	#Invalid	Slave cam position at this profile point.
SlaveVelocity	LREAL	#Invalid	Slave velocity at this profile point.
SlaveAcceleration	LREAL	#Invalid	Slave acceleration at this profile point.
SlaveJerk	LREAL	#Invalid	Slave jerk at this profile point.

Weiterführende Informationen

Ein CamProfilePoint enthält Informationen zu den Stützstellen einer Kurvenscheibe. Die Struktur wird für die Charakteristiken einer Kurvenscheibe verwendet (CamCharacteristics), um z. B. Extremwerte auszulesen und eindeutig einer Stützstelle in der Kurvenscheibe zuordnen zu können.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.2.2.4 CamScalingData

Contains scaling and offset factors to be applied in cam table couplings.

MasterCamPos = (MasterPos - MasterOffset) / MasterScaling

SlaveCamPos = CamTable(MasterCamPos)

SlavePos = (SlaveCamPos * SlaveScaling) + SlaveOffset.

Syntax

Definition:

```
TYPE CamScalingData :
STRUCT
    MasterOffsetMode : ECamOffsetMode;
    MasterOffset     : LREAL;
    MasterScaling    : LREAL;
    SlaveOffsetMode  : ECamOffsetMode;
    SlaveOffset      : LREAL;
    SlaveScaling     : LREAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
MasterOffsetMode	ECamOffsetMode [► 321]	None	Defines how the master offset is determined during cam activation.
MasterOffset	LREAL	0.0	Position offset applied to the master input before cam table lookup.
MasterScaling	LREAL	1.0	Scaling factor applied to the master input before cam table lookup.
SlaveOffsetMode	ECamOffsetMode [► 321]	None	Defines how the slave offset is determined during cam activation.
SlaveOffset	LREAL	0.0	Position offset added to the slave output after cam table evaluation.
SlaveScaling	LREAL	1.0	Scaling factor applied to the slave output after cam table evaluation.

Weiterführende Informationen

Die Skalierung einer Kurvenscheibe kann mit den Funktionsbausteinen MC_CamScaling für aktive Kurvenscheibenkopplungen vorgenommen werden. Auch bei einer Änderung der Kopplungssituation über MC_CamAdd, MC_CamIn und MC_CamExchange ist eine Skalierung der neuen Kurvenscheibenkopplung möglich.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.2.2.5 MC_CAM_REF

Reference to a cam table point array for cam table definition.

Syntax

Definition:

```

TYPE MC_CAM_REF :
STRUCT
    PointsAddress : Pointer To MotionFunctionPoint;
    NumberOfPoints : UDINT;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Description
PointsAddress	Pointer To MotionFunctionPoint [► 328]	Pointer to the source array of data points.
NumberOfPoints	UDINT	Number of data points in the source array.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.2.2.6 MotionFunctionPoint

Defines a cam table point with its interpolation function type and boundary conditions.

Syntax

Definition:

```

TYPE MotionFunctionPoint :
STRUCT
    FunctionType : EMotionFunctionType;
    MasterPosition : LREAL;
    SlavePosition : LREAL;
    SlaveVelocity : LREAL;
    SlaveAcceleration : LREAL;
    SlaveJerk : LREAL;
    SymmetryRatio : LREAL;
    Ignored : BOOL;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
FunctionType	EMotionFunctionType [► 322]	Undefined	Interpolation function used for the segment starting at this point.
MasterPosition	LREAL	#Invalid	Master axis position at this point.
SlavePosition	LREAL	#Invalid	Slave axis position at this point.
SlaveVelocity	LREAL	#Ignore	Slave velocity boundary condition at this point. Set to #Ignore for automatic calculation (given by interpolation).
SlaveAcceleration	LREAL	#Ignore	Slave acceleration boundary condition at this point. Set to #Ignore for automatic calculation (given by interpolation).

Name	Type	Default	Description
SlaveJerk	LREAL	#Ignore	Slave jerk boundary condition at this point. Set to #Ignore for automatic calculation (given by interpolation).
SymmetryRatio	LREAL	#Ignore	Normalized inflection point is valid only for rest-to-rest motion functions. This parameter is ignored if it is out of range or set for non-rest-to-rest motion functions.
Ignored	BOOL	false	If TRUE, this point is skipped during cam table compilation.

Weiterführende Informationen

Die Datenstruktur MotionFunctionPoint beschreibt eine Stützstelle einer Kurvenscheibe.

- FunctionType
 - Angabe der mathematischen Funktion, die die Interpolation zwischen zwei Stützstellen bestimmt. Ausgangspunkt für die Interpolation ist die aktuelle Stützstelle.
- SlaveVelo
 - Geschwindigkeit der Slave-Achsen an dieser Stützstelle. Wenn sich die Slave-Geschwindigkeit implizit durch den gewählten FunctionType ergibt, ist der Wert STRING_TO_LREAL('#Ignore') (Default) zu wählen. Ist dennoch ein abweichender Wert angegeben, wird dies beim MC_CamObject.Construct() mit einem Fehler abgelehnt und eine Fehlerbeschreibung in der Error List ausgegeben.
- SlaveAcc
 - Beschleunigung der Slave-Achsen an dieser Stützstelle. Wenn sich die Slave-Beschleunigung implizit durch den gewählten FunctionType ergibt, ist der Wert STRING_TO_LREAL('#Ignore') (Default) zu wählen. Ist dennoch ein abweichender Wert angegeben, wird dies beim MC_CamObject.Construct() mit einem Fehler abgelehnt und eine Fehlerbeschreibung in der Error List ausgegeben.
- SlaveJerk
 - Ruck der Slave-Achsen an dieser Stützstelle. Wenn sich die Slave-Ruck implizit durch den gewählten FunctionType ergibt, ist der Wert STRING_TO_LREAL('#Ignore') (Default) zu wählen. Ist dennoch ein abweichender Wert angegeben, wird dies beim MC_CamObject.Construct() mit einem Fehler abgelehnt und eine Fehlerbeschreibung in der Error List ausgegeben.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.3 Interfaces

5.4.3.1 Types

5.4.3.1.1 Type_CouplingFunction1D

Type for a motion control object that provides a 1D coupling function (e. g. supported by MC_CamObject).

Inheritance Hierarchy

Type_CouplingFunction1D

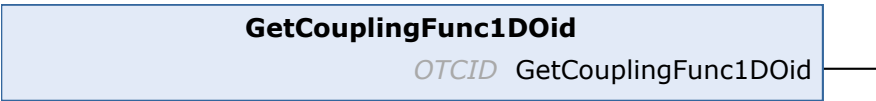
 Methods

Name	Origin	Description
GetCouplingFunc1DOid ▶ 330	Type_CouplingFunction1D	

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.4.3.1.1 GetCouplingFunc1DOid



Syntax

Definition:

```
METHOD GetCouplingFunc1DOid : OTCID
```

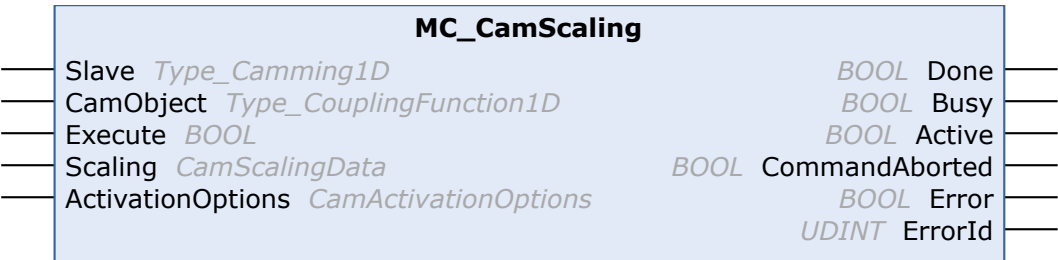
 Return value

OTCID

5.4.4 Function Blocks

5.4.4.1 Administrative

5.4.4.1.1 MC_CamScaling



This function block applies the scaling factor of an existing cam table coupling on a slave. InSync of the existing cam table coupling is not affected by this function block.

Syntax

Definition:

```
FUNCTION_BLOCK MC_CamScaling
VAR_INPUT
    Slave          : Type_Camming1D;
    CamObject      : Type_CouplingFunction1D;
    Execute        : BOOL;
    Scaling        : CamScalingData;
    ActivationOptions : CamActivationOptions;
END_VAR
```

```

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

```

Inputs

Name	Type	Description
Slave	Type_Camming1D	Reference to the slave axis
CamObject	Type_CouplingFunction1D ▶ 329	Reference to the cam object
Execute	BOOL	Apply cam scaling at rising edge
Scaling	CamScalingData ▶ 327	New cam scaling
ActivationOptions	CamActivationOptions ▶ 323	Options controlling how the cam coupling is activated (mode, position, reference object).

Outputs

Name	Type	Description
Done	BOOL	Cam scaling succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Weiterführende Informationen

Mit dem Funktionsbaustein MC_CamScaling kann eine Kurvenscheibenkopplung skaliert werden. Dabei werden nicht die Tabellenrohdaten der Kurvenscheibe beeinflusst, sondern die Skalierung bezieht sich auf eine aktive Master-Slave-Kopplung. Einstellbar sind die Skalierungsfaktoren für Master und Slave und die Offsets zur Verschiebung der Kurvenscheibe im Koordinatensystem.

Optional wirkt die Änderung erst ab einer bestimmten Master-Position, wodurch die Skalierung punktgenau während der Fahrt geändert werden kann.

HINWEIS

Sachschäden durch Veränderung einer aktiven Kurvenscheibenkopplung

Dieser Funktionsbaustein kann auch für eine aktiven Kurvenscheibe verwendet werden. Valide Änderungen werden in die Kurvenscheibe übernommen und können zu Sollwertsprüngen führen. Für das Aktivieren einer Kurvenscheibenskalierung zu einem definierten Zeitpunkt oder an einer definierten Position ist folgende Vorgehensweise empfohlen:

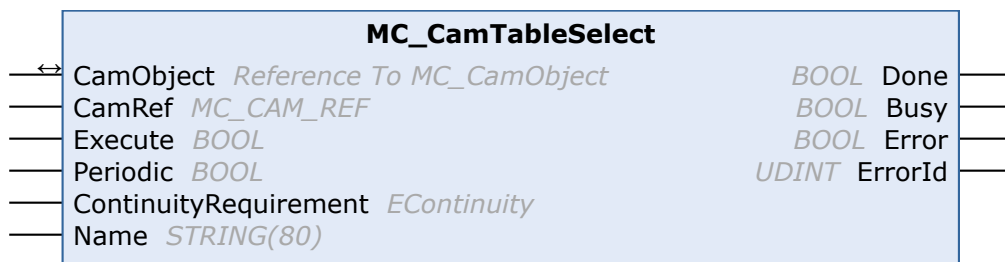
- Verwenden Sie den ECamActivationMode.AtBoundary, ECamActivationMode.AtMasterCamPosition oder ECamActivationMode.AtMasterAbsolutePosition
- Legen Sie alternativ eine Kopie der aktiven Kurvenscheibe an (Methode CopyFrom()) und nehmen Sie die Änderungen in der Kopie vor. Schalten Sie im Anschluss zwischen den Kurvenscheiben um (MC_CamExchange [▶ 335](#)).

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

Required License

TC3 MC3 Camming

5.4.4.1.2 MC_CamTableSelect

This function block selects the cam table. Internally the function block calls CamObject.Connect() if the cam object is linked to an existing cam table object, otherwise CamObject.Create(), and CamObject.Construct() are called based on CamRef data.

Syntax

Definition:

```

FUNCTION_BLOCK MC_CamTableSelect
VAR_IN_OUT
    CamObject      : Reference To MC_CamObject;
END_VAR
VAR_INPUT
    CamRef         : MC_CAM_REF;
    Execute        : BOOL;
    Periodic       : BOOL;
    ContinuityRequirement : EContinuity;
    Name           : STRING(80);
END_VAR
VAR_OUTPUT
    Done          : BOOL;
    Busy          : BOOL;
    Error         : BOOL;
    ErrorId       : UDINT;
END_VAR
  
```

Inputs

Name	Type	Default	Description
CamRef	MC_CAM_REF [► 327]		Data point information.
Execute	BOOL		Start cam select at rising edge.
Periodic	BOOL		Set to TRUE for a periodic cam table.
ContinuityRequirement	EContinuity		Minimum continuity requirement. The higher the requirement, the stricter the point validation.
Name	STRING(80)	""	Only applicable if the cam object is not linked and creation of a new cam table object is required.

In/Outputs

Name	Type	Description
CamObject	Reference To MC_CamObject [► 339]	Reference to the cam object

🚀 Outputs

Name	Type	Description
Done	BOOL	Selection succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Version information

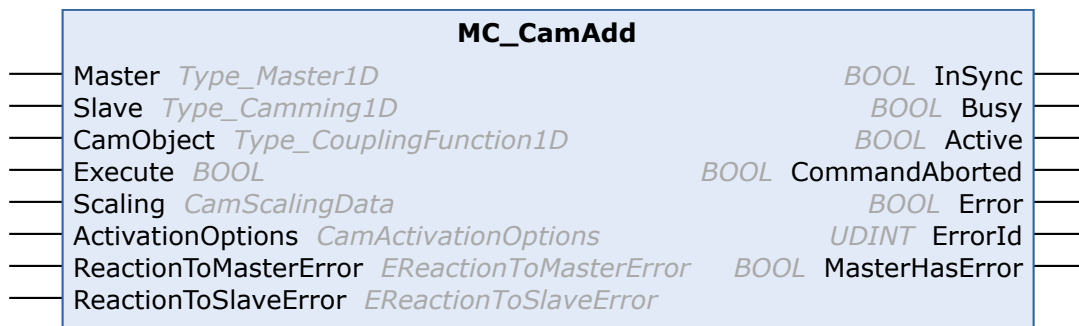
- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

Required License

TC3 MC3 Camming

5.4.4.2 Motion

5.4.4.2.1 MC_CamAdd



This function block adds a cam table coupling on a slave. Generated setpoints from single cam table couplings are internally added.

As MC_CamIn, this function block can also be used to initiate camming-based synchronization.

As this function block is used for multi-camming, some special rules apply:

The same master can be used in multiple cam table couplings. A cam table can only be used once in a slave.

Syntax

Definition:

```
FUNCTION_BLOCK MC_CamAdd
VAR_INPUT
    Master          : Type_Master1D;
    Slave           : Type_Camming1D;
    CamObject       : Type_CouplingFunction1D;
    Execute         : BOOL;
    Scaling         : CamScalingData;
    ActivationOptions : CamActivationOptions;
    ReactionToMasterError : EReactionToMasterError;
    ReactionToSlaveError : EReactionToSlaveError;
END_VAR
VAR_OUTPUT
    InSync          : BOOL;
    Busy            : BOOL;
    Active          : BOOL;
    CommandAborted  : BOOL;
    Error           : BOOL;
    ErrorId         : UDINT;
    MasterHasError  : BOOL;
END_VAR
```

🔧 Inputs

Name	Type	Default	Description
Master	Type_Master1D		Reference to the master axis
Slave	Type_Camming1D		Reference to the slave axis
CamObject	Type_CouplingFunction1D [► 329]		Reference to the cam object
Execute	BOOL		Start cam add at rising edge.
Scaling	CamScalingData [► 327]		Scaling of the new coupling
ActivationOptions	CamActivationOptions [► 323]		Options controlling how the cam coupling is activated (mode, position, reference object).
ReactionToMasterError	EReactionToMasterError	TriggerSlaveError	Reaction if an error occurs in the master while this command is active.
ReactionToSlaveError	EReactionToSlaveError	TriggerMasterError	Reaction if an error occurs in the slave while this command is active.

🔧 Outputs

Name	Type	Description
InSync	BOOL	Synchronization succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
MasterHasError	BOOL	Master is in error state.

Weiterführende Informationen:

Für ActivationOptions.ReferenceObject gelten die gleichen Regeln wie für MC_CamIn.
ActivationOptions.ReferenceObject:

- ReferenceObject ist nicht erforderlich, wenn die Kurvensynchronisation eingeleitet wird (zur Aktivierung wird die Master-Kurventabellenkopplung dieses Funktionsbausteins verwendet).
- Standardmäßig wird die aktive Master-Kurventabellenkopplung des Slaves verwendet, sofern nur eine vorhanden ist (ReferenceObject = 0). Bei mehreren Slaves muss ReferenceObject = CamObject gesetzt werden, um die korrekte Kopplung einzustellen.
- ReferenceObject darf nicht 0 sein, wenn mehr als eine aktive Master-Kurventabellenkopplung vorhanden ist.

MC_CamAdd ermöglicht es, für eine Slave-Achse Kurvenscheibenkopplungen zu mehreren verschiedenen Master-Achsen zu programmieren. Dabei gilt, dass eine Master-Achse in mehreren Kurvenscheibenkopplungen, also für mehrere Slave-Achsen, genutzt werden kann. Ein MC_CamObject kann für dieselbe Slave-Achse nur einmal verwendet werden.

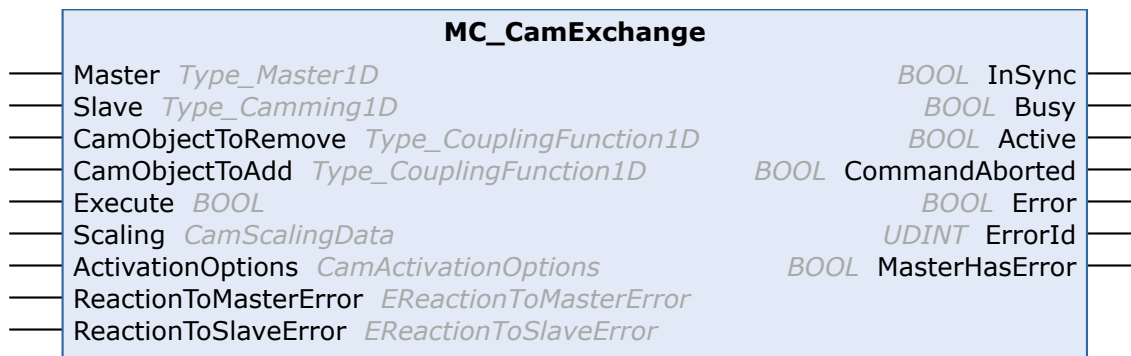
Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

Required License

TC3 MC3 Camming

5.4.4.2.2 MC_CamExchange



This function block exchanges cam table couplings on a slave. The cam table coupling that should be removed has to be active when this function block is executed and the activation condition is met, otherwise an error is triggered. Activation might depend either on the new entered cam table coupling (ActivationOptions.ReferenceObject = ECamActivationRef.NewlyEnteredCamTable) or on the one to be removed (ActivationOptions.ReferenceObject = ECamActivationRef.CamTableAlreadyInUse)

Syntax

Definition:

```
FUNCTION_BLOCK MC_CamExchange
VAR_INPUT
    Master          : Type_Master1D;
    Slave           : Type_Camming1D;
    CamObjectToRemove : Type_CouplingFunction1D;
    CamObjectToAdd   : Type_CouplingFunction1D;
    Execute          : BOOL;
    Scaling           : CamScalingData;
    ActivationOptions : CamActivationOptions;
    ReactionToMasterError : EReactionToMasterError;
    ReactionToSlaveError : EReactionToSlaveError;
END_VAR
VAR_OUTPUT
    InSync          : BOOL;
    Busy             : BOOL;
    Active           : BOOL;
    CommandAborted   : BOOL;
    Error            : BOOL;
    ErrorId          : UDINT;
    MasterHasError    : BOOL;
END_VAR
```

Inputs

Name	Type	Default	Description
Master	Type_Master1D		Reference to the master axis
Slave	Type_Camming1D		Reference to the slave axis
CamObjectTo Remove	<u>Type_CouplingFunction1D</u> [► 329]		Cam object to be removed
CamObjectToA dd	<u>Type_CouplingFunction1D</u> [► 329]		Cam object to be added
Execute	BOOL		Start cam exchange at rising edge.
Scaling	<u>CamScalingData</u> [► 327]		Scaling of the new coupling
ActivationOptio ns	<u>CamActivationOptions</u> [► 323]		Options controlling how the cam coupling is activated (mode, position, reference object).
ReactionToMa sterError	EReactionToMasterError	TriggerSlaveError	Reaction if an error occurs in the master while this command is active.
ReactionToSla veError	EReactionToSlaveError	TriggerMasterError	Reaction if an error occurs in the slave while this command is active.

🔌 Outputs

Name	Type	Description
InSync	BOOL	Synchronization succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
MasterHasError	BOOL	Master is in error state.

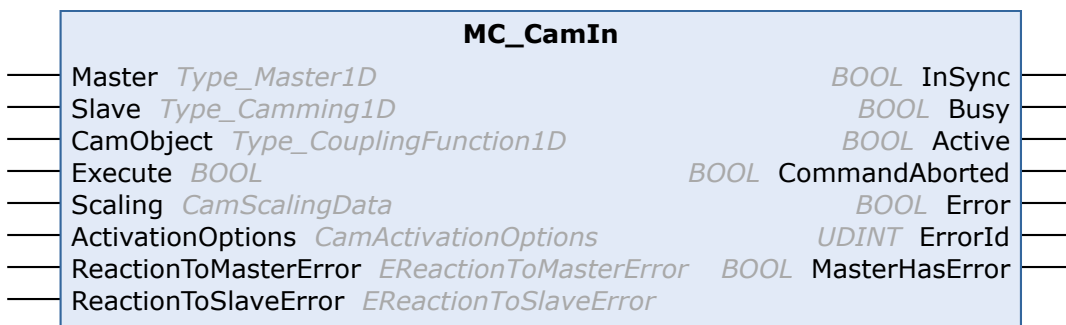
Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

Required License

TC3 MC3 Camming

5.4.4.2.3 MC_CamIn



This function block initiates camming-based synchronization on a slave.

If the slave is already coupled to one or more cam table couplings, it switches to the cam table coupling that is entered in this function block.

Note: Use MC_CamAdd to switch from single-camming to multi-camming. Switch back and forth between single-camming and multi-camming is possible and not limited.

Syntax

Definition:

```
FUNCTION_BLOCK MC_CamIn
VAR_INPUT
    Master          : Type_Master1D;
    Slave           : Type_Camming1D;
    CamObject       : Type_CouplingFunction1D;
    Execute         : BOOL;
    Scaling         : CamScalingData;
    ActivationOptions : CamActivationOptions;
    ReactionToMasterError : EReactionToMasterError;
    ReactionToSlaveError : EReactionToSlaveError;
END_VAR
VAR_OUTPUT
    InSync          : BOOL;
    Busy            : BOOL;
    Active          : BOOL;
    CommandAborted  : BOOL;
    Error           : BOOL;
```

```
ErrorId          : UDINT;
MasterHasError   : BOOL;
END_VAR
```

Inputs

Name	Type	Default	Description
Master	Type_Master1D		Reference to the master axis
Slave	Type_Camming1D		Reference to the slave axis
CamObject	Type_CouplingFunction1D [► 329]		Reference to the cam object
Execute	BOOL		Start cam in at rising edge.
Scaling	CamScalingData [► 327]		Scaling of the new coupling
ActivationOptions	CamActivationOptions [► 323]		Options controlling how the cam coupling is activated (mode, position, reference object).
ReactionToMasterError	EReactionToMasterError	TriggerSlaveError	Reaction if an error occurs in the master while this command is active.
ReactionToSlaveError	EReactionToSlaveError	TriggerMasterError	Reaction if an error occurs in the slave while this command is active.

Outputs

Name	Type	Description
InSync	BOOL	Synchronization succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
MasterHasError	BOOL	Master is in error state.

Weiterführende Informationen

Für ActivationOptions.ReferenceObject gelten die folgenden Regeln:

- ReferenceObject ist nicht erforderlich, wenn die Kurvensynchronisation eingeleitet wird (zur Aktivierung wird die Master-Kurventabellenkopplung dieses Funktionsbausteins verwendet).
- Standardmäßig wird die aktive Master-Kurventabellenkopplung des Slaves verwendet, sofern nur eine vorhanden ist (ReferenceObject = 0). Bei mehreren Slaves muss ReferenceObject = CamObject gesetzt werden, um die korrekte Kopplung einzustellen.
- ReferenceObject darf nicht 0 sein, wenn mehr als eine aktive Master-Kurventabellenkopplung vorhanden ist.

Ergänzungen zu ReactionToSlaveError und ReactionToMasterError

Grundsätzlich gilt bei einem Fehler der Slave-Achse:

- Slave-Achse: Die Kopplung wird aufgelöst und die Reaktion ist abhängig vom Fehlertypen.
- Master-Achse: Die Master-Achse meldet einen Fehler und folgt einer Stopprampe der MC oder führt ihre laufende Bewegung ohne Beeinflussung fort.

Bei einem Fehler der Master-Achse gilt:

- Master-Achse: Reaktion ist abhängig vom Fehlertypen.

- Slave-Achse: Die Slave-Achse meldet einen Fehler und folgt einer Stopprampe der MC oder den Sollwerten der Master-Achse.

Ausführliche Informationen sind im Kapitel [Fehlerverhalten](#) [► 311] dokumentiert.

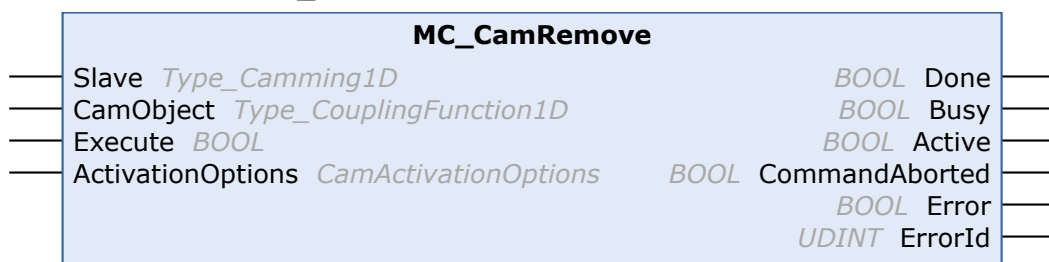
Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

Required License

TC3 MC3 Camming

5.4.4.2.4 MC_CamRemove



This function block removes a cam table coupling from a slave. As a cam table can only be used once in a coupled slave, the master information is not needed. However, the cam table must be in use when this function block is executed and when the activation condition is met, otherwise an error is triggered. If there is no active coupling left and there are no pending commands to add couplings, the camming synchronization ends by itself and axis state changes to Standstill.

Syntax

Definition:

```
FUNCTION_BLOCK MC_CamRemove
VAR_INPUT
    Slave          : Type_Camming1D;
    CamObject      : Type_CouplingFunction1D;
    Execute        : BOOL;
    ActivationOptions : CamActivationOptions;
END_VAR
VAR_OUTPUT
    Done           : BOOL;
    Busy           : BOOL;
    Active         : BOOL;
    CommandAborted : BOOL;
    Error          : BOOL;
    ErrorId        : UDINT;
END_VAR
```

Inputs

Name	Type	Description
Slave	Type_Camming1D	Reference to the slave axis
CamObject	Type_CouplingFunction1D [► 329]	Cam object to be removed (reference to the object)
Execute	BOOL	Start cam remove at rising edge.
ActivationOptions	CamActivationOptions [► 323]	Options controlling how the cam coupling is activated (mode, position, reference object).

 Outputs

Name	Type	Description
Done	BOOL	Removal succeeded
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

Required License

TC3 MC3 Camming

5.4.4.3 Motion Object**5.4.4.3.1 MC_CamObject**

Wrapper that holds a reference to a cam table object and provides methods for setting, accessing and modifying its data.

To use this wrapper, the cam table must be mapped and the method `Connect()` must be called.

Once a reference is successfully stored, it cannot be modified. Subsequent calls to `Connect()` or `Create()` will return an error.

Do not call the main FB directly. Only use the available methods.

 Methods

Name	Description
Connect [► 340]	Stores a reference to an existing cam table object. The cam table must have been previously linked to McToPlc, or the object ID must be passed as parameter.
Create [► 341]	Creates a new cam table object and stores a reference to it.
WriteData [► 341]	Calculates the entire cam table from a new set of data points.
ReadData [► 342]	Reads data points from index StartIndex to StartIndex + NumberOfPoints - 1. The first valid index is 1.
OverwriteData [► 343]	Overwrites data points from index StartIndex to StartIndex + NumberOfPoints - 1 and recalculates the modified cam table segments. The first valid index is 1.
ReadSlaveDynamics [► 343]	Evaluates slave position, velocity and acceleration at a specific master position within the cam table profile.
ComputeMasterPositions [► 344]	Finds the master position(s) for a given slave position.
ReadMasterRange [► 345]	Returns leftmost and rightmost master position of the cam table profile.
ComputeCharacteristics [► 346]	Computes cam table characteristics such as extreme slave dynamics for a given nominal master velocity.
ClearData [► 346]	Removes all stored data. The existing reference remains intact.

Name	Description
CopyFrom [► 346]	Copies data points and cam table segments from another valid MC_CamObject.

Weiterführende Informationen

Das MC_CamObject ist das zentrale Objekt zur Handhabung von Kurvenscheiben. Dies umfasst u. a. Methoden zum Setzen, Auslesen und Modifizieren von Datenpunkten. Um diesen Wrapper zu verwenden, muss zunächst eine Referenz auf eine Kurventabelle über eine der folgenden Optionen abgerufen werden:

1. Verknüpfung mit einer vorhandenen Kurventabelle zur Konfigurationszeit, siehe [Kurvenscheibe erstellen](#) [► 315]:
 - Verknüpfen Sie den Eingang „McToPlc.Oid“ des Cam-Objekts in der PLC mit dem Ausgang „McToPlc.Oid“ einer vorhandenen Kurventabelle im MC3-Motion-Projekt.
 - Rufen Sie in der PLC die Methode [Connect\(\)](#) [► 340] auf, um die Verknüpfung zu initialisieren. Nach der erfolgreichen Verknüpfung kann diese nicht mehr verändert werden. Ein erneutes „Connect()“ führt zu einem Fehler.
2. Erstellen einer neuen Kurventabelle zur Laufzeit in der PLC: Rufen Sie [Create\(\)](#) [► 341] auf, um die Instanziierung eines neuen Kurventabellenobjekts intern anzufordern und eine Referenz auf dieses neu erstellte Objekt zu speichern. Optional kann ein Name als FB-Eingang angegeben werden. Danach ist ein MC_CamObject bereit für die Verwendung. Ein erneutes „Create()“ führt zu einem Fehler.
3. Die Berechnung der Kurvenscheibe anhand gegebener Datenpunkte wird mit der Methode [WriteData\(\)](#) [► 341] ausgeführt.

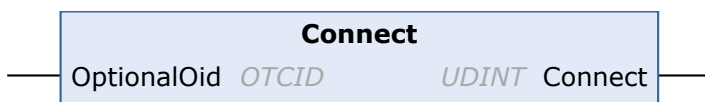
Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5550 MC3 Camming >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

Required License

TC3 MC3 Camming

5.4.4.3.1.1 Connect



Stores a reference to an existing cam table object. The cam table must have been previously linked to McToPlc, or the object ID must be passed as parameter.

Syntax

Definition:

```

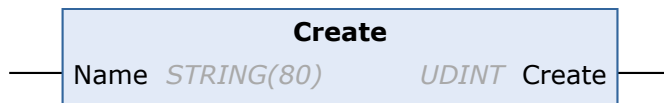
METHOD Connect : UDINT
VAR_INPUT
    OptionalOid : OTCID;
END_VAR
  
```

Inputs

Name	Type	Description
OptionalOid	OTCID	Cam table object ID, used when McToPlc is not linked. If both are set, McToPlc takes priority.

 Return value

UDINT

5.4.4.3.1.2 Create

Creates a new cam table object and stores a reference to it.

Syntax

Definition:

```

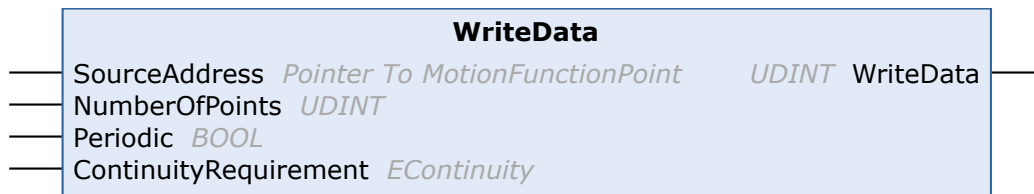
METHOD Create : UDINT
VAR_INPUT
    Name : STRING(80);
END_VAR
  
```

 Inputs

Name	Type	Description
Name	STRING(80)	Optional name for the new cam table.

 Return value

UDINT

5.4.4.3.1.3 WriteData

Calculates the entire cam table from a new set of data points.

Syntax

Definition:

```

METHOD WriteData : UDINT
VAR_INPUT
    SourceAddress      : Pointer To MotionFunctionPoint;
    NumberOfPoints      : UDINT;
    Periodic           : BOOL;
    ContinuityRequirement : EContinuity;
END_VAR
  
```

 Inputs

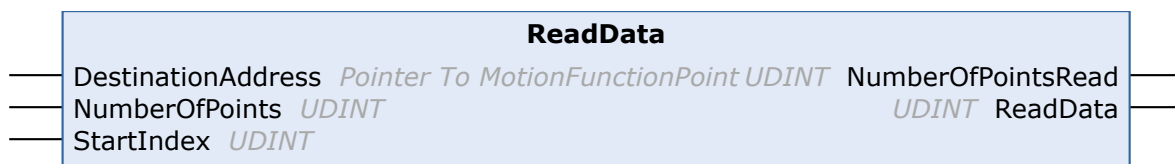
Name	Type	Description
SourceAddress	Pointer To <u>MotionFunctionPoint</u> [► 328]	Pointer to the new set of data points.
NumberOfPoints	UDINT	Number of data points in the source array.

Name	Type	Description
Periodic	BOOL	TRUE for a periodic cam table.
ContinuityRequirement	EContinuity	Minimum continuity requirement. The higher the requirement, the stricter the point validation.

Return value

UDINT

5.4.4.3.1.4 ReadData



Reads data points from index StartIndex to StartIndex + NumberOfPoints - 1. The first valid index is 1.

Syntax

Definition:

```

METHOD ReadData : UDINT
VAR_INPUT
    DestinationAddress : Pointer To MotionFunctionPoint;
    NumberOfPoints      : UDINT;
    StartIndex          : UDINT;
END_VAR
VAR_OUTPUT
    NumberOfPointsRead : UDINT;
END_VAR

```

Inputs

Name	Type	Description
DestinationAddress	Pointer To MotionFunctionPoint [► 328]	Pointer to the destination buffer for the requested data points.
NumberOfPoints	UDINT	Number of points to read.
StartIndex	UDINT	Index of the first point to read.

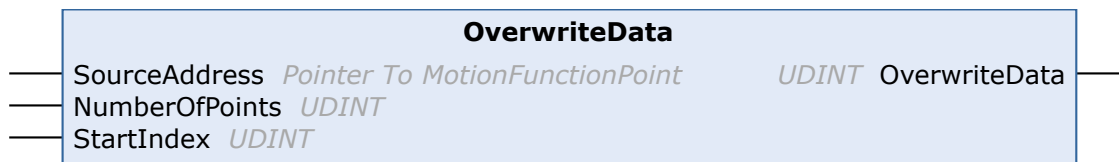
Outputs

Name	Type	Description
NumberOfPointsRead	UDINT	The actual number of points read, which can be less than or equal to the specified NumberOfPoints.

Return value

UDINT

5.4.4.3.1.5 OverwriteData



Overwrites data points from index StartIndex to StartIndex + NumberOfPoints - 1 and recalculates the modified cam table segments. The first valid index is 1.

Syntax

Definition:

```
METHOD OverwriteData : UDINT
VAR_INPUT
    SourceAddress : Pointer To MotionFunctionPoint;
    NumberOfPoints : UDINT;
    StartIndex : UDINT;
END_VAR
```

Inputs

Name	Type	Description
SourceAddress	Pointer To MotionFunctionPoint ▶ 328	Pointer to the new data points to write.
NumberOfPoints	UDINT	Number of points to overwrite.
StartIndex	UDINT	Index of the first point to overwrite.

Return value

UDINT

Weiterführende Informationen

Die Eigenschaft ermöglicht das (Über-) Schreiben eines definierten Datenbereichs der Kurvenscheibe.

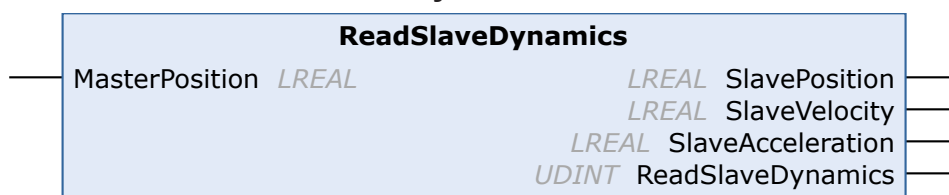
HINWEIS

Sachschäden durch Veränderung einer aktiven Kurvenscheibenkopplung

Diese Methode kann auch bei einer aktiven Kurvenscheibe aufgerufen werden. Valide Änderungen werden synchron in die Kurvenscheibe übernommen und können zu Sollwertsprüngen führen. Für eine Kurvenscheibenumschaltung zu einem definierten Zeitpunkt oder an einer definierten Position werden folgende Schritte empfohlen:

- Legen Sie eine Kopie der aktiven Kurvenscheibe an (Methode CopyFrom()).
- Nehmen Sie die Änderungen in der Kopie vor (Methode OverwriteData()).
- Schalten Sie zwischen den Kurvenscheiben um ([MC_CamExchange ▶ 335](#)).

5.4.4.3.1.6 ReadSlaveDynamics



Evaluates slave position, velocity and acceleration at a specific master position within the cam table profile.

Syntax

Definition:

```
METHOD ReadSlaveDynamics : UDINT
VAR_INPUT
    MasterPosition      : LREAL;
END_VAR
VAR_OUTPUT
    SlavePosition       : LREAL;
    SlaveVelocity       : LREAL;
    SlaveAcceleration   : LREAL;
END_VAR
```

Inputs

Name	Type	Description
MasterPosition	LREAL	Master cam position at which to evaluate the slave dynamics.

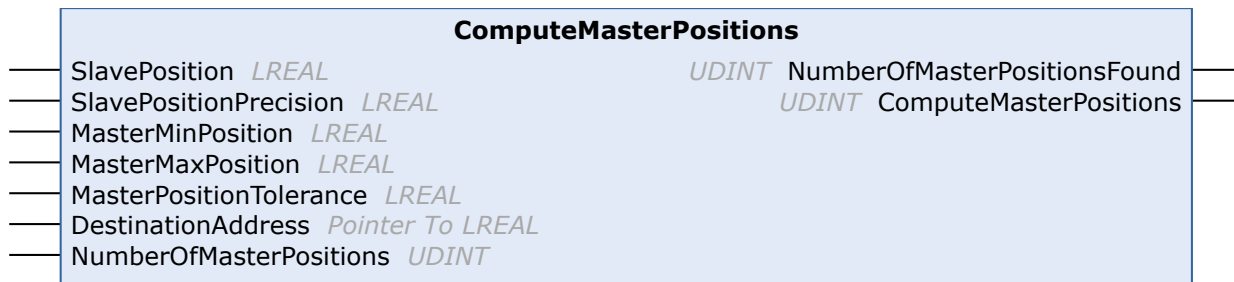
Outputs

Name	Type	Description
SlavePosition	LREAL	Computed slave position at the given master cam position.
SlaveVelocity	LREAL	Computed slave velocity at the given master cam position.
SlaveAcceleration	LREAL	Computed slave acceleration at the given master cam position.

Return value

UDINT

5.4.4.3.1.7 ComputeMasterPositions



Finds the master position(s) for a given slave position.

Syntax

Definition:

```
METHOD ComputeMasterPositions : UDINT
VAR_INPUT
    SlavePosition           : LREAL;
    SlavePositionPrecision  : LREAL;
    MasterMinPosition       : LREAL;
    MasterMaxPosition       : LREAL;
    MasterPositionTolerance : LREAL;
    DestinationAddress       : Pointer To LREAL;
    NumberOfMasterPositions  : UDINT;
END_VAR
VAR_OUTPUT
    NumberOfMasterPositionsFound : UDINT;
END_VAR
```

 Inputs

Name	Type	Description
SlavePosition	LREAL	Position of the slave axis for which the master positions are to be determined.
SlavePositionPrecision	LREAL	Absolute precision for which a slave position at a certain master position is considered equal to the SlavePosition input (must be larger than 0, e.g. 1E-3).
MasterMinPosition	LREAL	Leftmost limit of the cam profile range to be evaluated.
MasterMaxPosition	LREAL	Rightmost limit of the cam profile range to be evaluated.
MasterPositionTolerance	LREAL	Minimum distance between distinct master positions found (must be larger than 0, e.g. 1E-3).
DestinationAddress	Pointer To LREAL	Pointer to the destination array for found master positions.
NumberOfMasterPositions	UDINT	Capacity of the DestinationAddress array.

 Outputs

Name	Type	Description
NumberOfMasterPositionsFound	UDINT	Number of master positions found matching the given slave position and master position range.

 Return value

UDINT

5.4.4.3.1.8 ReadMasterRange

ReadMasterRange	
<i>LREAL</i>	MasterMinPosition
<i>LREAL</i>	MasterMaxPosition
<i>UDINT</i>	ReadMasterRange

Returns leftmost and rightmost master position of the cam table profile.

Syntax

Definition:

```

METHOD ReadMasterRange : UDINT
VAR_OUTPUT
    MasterMinPosition : LREAL;
    MasterMaxPosition : LREAL;
END_VAR

```

 Outputs

Name	Type	Description
MasterMinPosition	LREAL	Leftmost master position of the cam table profile.
MasterMaxPosition	LREAL	Rightmost master position of the cam table profile.

 Return value

UDINT

5.4.4.3.1.9 ComputeCharacteristics**ComputeCharacteristics**

NominalMasterVelocity *LREAL* *UDINT* ComputeCharacteristics
 Destination *Reference To CamCharacteristics*

Computes cam table characteristics such as extreme slave dynamics for a given nominal master velocity.

Syntax

Definition:

```
METHOD ComputeCharacteristics : UDINT
VAR_INPUT
    NominalMasterVelocity : LREAL;
    Destination           : Reference To CamCharacteristics;
END_VAR
```

 Inputs

Name	Type	Description
NominalMasterVelocity	LREAL	Nominal master velocity used for computing absolute slave dynamics.
Destination	Reference To CamCharacteristics [► 324]	Reference to a CamCharacteristics structure to receive the computed results.

 Return value

UDINT

5.4.4.3.1.10 ClearData**ClearData***UDINT* ClearData

Removes all stored data. The existing reference remains intact.

Syntax

Definition:

```
METHOD ClearData : UDINT
```

 Return value

UDINT

5.4.4.3.1.11 CopyFrom**CopyFrom**

Source *Reference To MC_CamObject* *UDINT* CopyFrom

Copies data points and cam table segments from another valid MC_CamObject.

Syntax

Definition:

```
METHOD CopyFrom : UDINT
VAR_INPUT
    Source : Reference To MC_CamObject;
END_VAR
```

Inputs

Name	Type	Description
Source	Reference To MC_CamObject [► 339]	Source MC_CamObject to copy from.

Return value

UDINT

5.4.5 Anhang

Umstieg NC2 zur MC3

Die Funktionalität der Kurvenscheiben verfolgt mit der MC3 einen neuen objektorientierteren Ansatz. Zentrales Element ist das [MC_CamObject \[► 339\]](#).

Funktionsbausteine

NC2 Funktionsbaustein	MC3 Funktionsbaustein	Hinweis
MC_CamIn.InSync MC_CamIn.Busy MC_CamIn.Active	MC_CamIn [► 336].InSync MC_CamIn [► 336].Busy MC_CamIn [► 336].Active	In der Ausgangssituation wurde das Kommando mit <code>Execute := TRUE</code> abgeschickt und es wird wieder <code>Execute := FALSE</code> gesetzt. NC2: <code>Busy = TRUE</code>, <code>Active = TRUE</code> und <code>InSync = TRUE</code> werden mit einer fallenden Flanke am <code>Execute</code> Eingang zurückgesetzt. Jedoch signalisieren die Ausgänge (insbesondere <code>InGear</code>) mindestens für einen Zyklus <code>TRUE</code>, sobald die Master- und Slave-Achse sich synchron zueinander bewegen. MC3: <code>Busy = TRUE</code>, <code>Active = TRUE</code> und <code>InSync = TRUE</code> bleiben so lange gesetzt, bis das Bewegungskommando durch ein anderes Kommando abgelöst wird, auch wenn <code>Execute</code> bereits auf <code>FALSE</code> gesetzt ist (PLCopen-konform).

5.5 Tc3_Mc3FluidPower

5.5.1 Data Types

5.5.1.1 Structs

5.5.1.1.1 ValveCharacteristicPoint

A single point on the valve characteristic curve mapping velocity to spool position.

Syntax

Definition:

```
TYPE ValveCharacteristicPoint :
STRUCT
    Velocity      : LREAL;
    SpoolPosition : LREAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
Velocity	LREAL	#Invalid	Axis velocity corresponding to the given valve spool position.
SpoolPosition	LREAL	#Invalid	Valve spool position corresponding to the given velocity.

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5560 MC3 FluidPower >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

5.5.1.1.2 ValveCharacterizationOptions

Contains additional options for MC_ValveCharacterization

Syntax

Definition:

```
TYPE ValveCharacterizationOptions :
STRUCT
    WaitTimeAfterVelocityStep : LREAL;
    WaitTimeAtPositionLimit   : LREAL;
    MinimumTravelPerMeasurement : LREAL;
    Acceleration               : LREAL;
    Deceleration               : LREAL;
    Jerk                      : LREAL;
    OutputLimit                : LREAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
WaitTimeAfterVelocityStep	LREAL	0.2	Wait time after velocity step ([sec]). Wait time should be large enough

Name	Type	Default	Description
			for velocity oscillations to settle. Must have value ≥ 0.0 s. Default: 0.2.
WaitTimeAtPositionLimit	LREAL	0.5	Wait time at a position limit ([sec]). Wait time should be large enough for the axis to come to a complete stop. Must have value ≥ 0.0 s. Wait time can be extended using the Wait input if it depends on external processes (e.g. switching valves). Default: 0.5.
MinimumTravelPerMeasurement	LREAL	10.0	Minimum position distance per velocity measurement ([PosUnit]). Must have value ≥ 0.0 . Default 10.0.
Acceleration	LREAL	#Default	Maximum acceleration ([PosUnit]/s ² , positive)
Deceleration	LREAL	#Default	Maximum deceleration ([PosUnit]/s ² , positive)
Jerk	LREAL	#Default	Maximum jerk ([PosUnit]/s ³ , positive)
OutputLimit	LREAL	1.0	Maximum valve output during valve characterization. Valid values: [0..1]. Valve characterization may exceed this value by up to OutputStepSize value.

Versionsinformation

- TwinCAT Standard $\geq$ v3.1.4026.23.1
- TF5560 MC3 FluidPower $\geq$ v4.0.6 (inkludiert TF5500 MC3 Base $\geq$ v4.0.6)

5.5.2 Function Blocks

5.5.2.1 Valve Characteristic Curve

5.5.2.1.1 MC_ValveCharacteristicCurve

Wrapper that holds a reference to a valve characteristic curve object and provides methods for setting, accessing and modifying its data.

To use this wrapper, the valve characteristic curve must be mapped and the method Connect() must be called.

Once a reference is successfully stored, it cannot be modified. Subsequent calls to Connect() will return an error.

Do not call the main FB directly. Only use the available methods.

Methods

Name	Description
Connect [► 350]	Stores a reference to an existing valve characteristic curve object. The valve characteristic curve must have been previously linked to McToPlc.
ReadData [► 350]	Reads all stored data points into the given array. Returns the number of data points read.
OverwriteData [► 351]	Overwrites all stored data points with data points from the given array.
ClearData [► 351]	Removes all stored data points. The existing reference remains intact.
CopyFrom [► 352]	Copies all stored data points from another MC_ValveCharacteristicCurve.
Enable [► 352]	Enables valve linearization.
Disable [► 352]	Disables valve linearization.
ComputeVelocityValue [► 353]	Computes velocity value at the given SpoolPosition.
ComputeSpoolPositionValue [► 353]	Computes spool position value at the given Velocity.

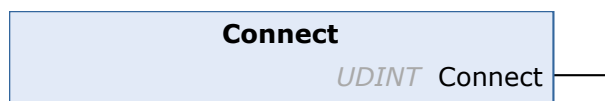
Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5560 MC3 FluidPower >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

Required License

TC3 MC3 Fluid Power

5.5.2.1.1.1 Connect



Stores a reference to an existing valve characteristic curve object. The valve characteristic curve must have been previously linked to McToPlc.

Syntax

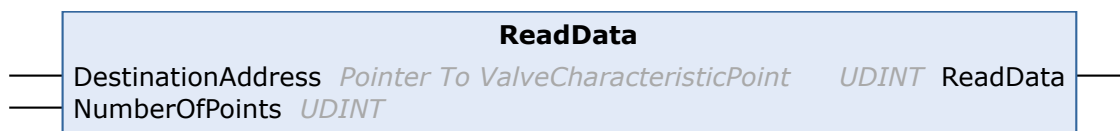
Definition:

```
METHOD Connect : UDINT
```

Return value

UDINT

5.5.2.1.1.2 ReadData



Reads all stored data points into the given array. Returns the number of data points read.

Syntax

Definition:

```

METHOD ReadData : UDINT
VAR_INPUT
    DestinationAddress : Pointer To ValveCharacteristicPoint;
    NumberOfPoints      : UDINT;
END_VAR

```

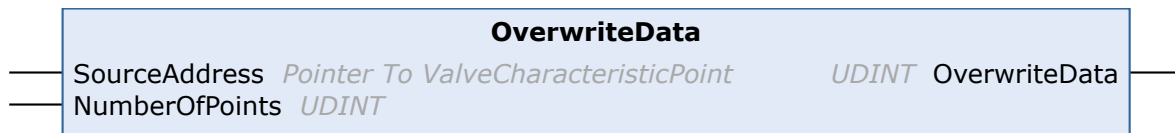
Inputs

Name	Type	Description
DestinationAddress	Pointer To <u>ValveCharacteristicPoint</u> ▶ 348	Pointer to ARRAY[1..NumberOfPoints] of ValveCharacteristicPoint
NumberOfPoints	UDINT	Capacity of the destination array (maximum number of data points to read).

Return value

UDINT

5.5.2.1.1.3 OverwriteData



Overwrites all stored data points with data points from the given array.

Syntax

Definition:

```

METHOD OverwriteData : UDINT
VAR_INPUT
    SourceAddress : Pointer To ValveCharacteristicPoint;
    NumberOfPoints : UDINT;
END_VAR

```

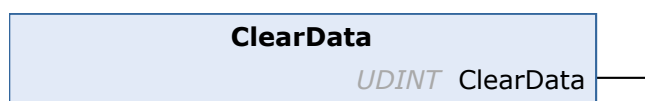
Inputs

Name	Type	Description
SourceAddress	Pointer To <u>ValveCharacteristicPoint</u> ▶ 348	Pointer to ARRAY[1..NumberOfPoints] of ValveCharacteristicPoint
NumberOfPoints	UDINT	Number of points to be written.

Return value

UDINT

5.5.2.1.1.4 ClearData



Removes all stored data points. The existing reference remains intact.

Syntax

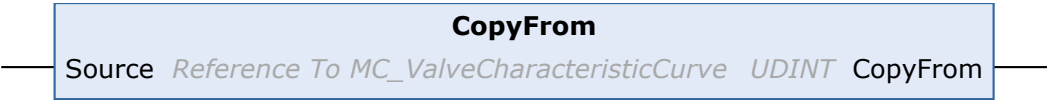
Definition:

```
METHOD ClearData : UDINT
```

 Return value

UDINT

5.5.2.1.1.5 CopyFrom



Copies all stored data points from another MC_ValveCharacteristicCurve.

Syntax

Definition:

```
METHOD CopyFrom : UDINT
VAR_INPUT
    Source : Reference To MC_ValveCharacteristicCurve;
END_VAR
```

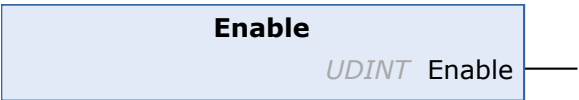
 Inputs

Name	Type	Description
Source	Reference To MC_ValveCharacteristicCurve [► 349]	Source MC_ValveCharacteristicCurve to copy from.

 Return value

UDINT

5.5.2.1.1.6 Enable



Enables valve linearization.

Syntax

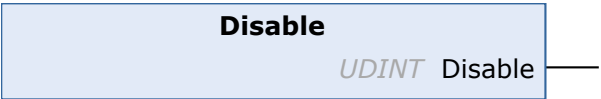
Definition:

```
METHOD Enable : UDINT
```

 Return value

UDINT

5.5.2.1.1.7 Disable



Disables valve linearization.

Syntax

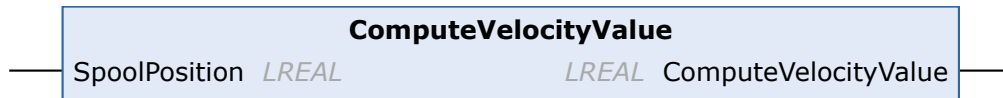
Definition:

```
METHOD Disable : UDINT
```

Return value

UDINT

5.5.2.1.1.8 ComputeVelocityValue



Computes velocity value at the given SpoolPosition.

Syntax

Definition:

```
METHOD ComputeVelocityValue : LREAL
VAR_INPUT
    SpoolPosition : LREAL;
END_VAR
```

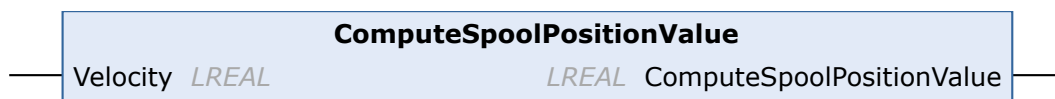
Inputs

Name	Type	Description
SpoolPosition	LREAL	Spool position at which to evaluate the velocity.

Return value

LREAL

5.5.2.1.1.9 ComputeSpoolPositionValue



Computes spool position value at the given Velocity.

Syntax

Definition:

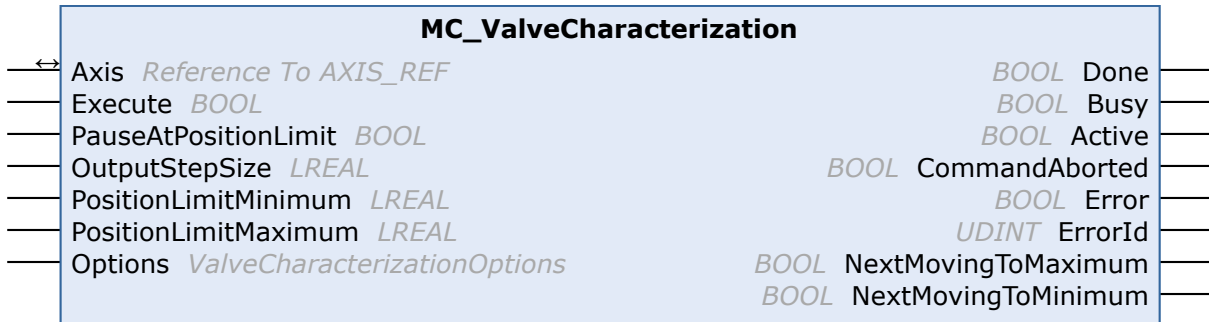
```
METHOD ComputeSpoolPositionValue : LREAL
VAR_INPUT
    Velocity : LREAL;
END_VAR
```

Inputs

Name	Type	Description
Velocity	LREAL	Velocity at which to evaluate the spool position.

 Return value

LREAL

5.5.2.2 Valve Characterization**5.5.2.2.1 MC_ValveCharacterization**

This function block characterizes the valve characteristic curve of the axis through a series of velocity measurements.

For each measurement the function block generates an output value for the drive based on the OutputStepSize input and the drive's actuating value range and measures the resulting actual velocity.

Syntax**Definition:**

```

FUNCTION_BLOCK MC_ValveCharacterization
VAR_IN_OUT
  Axis          : Reference To AXIS_REF;
END_VAR
VAR_INPUT
  Execute       : BOOL;
  PauseAtPositionLimit : BOOL;
  OutputStepSize : LREAL;
  PositionLimitMinimum : LREAL;
  PositionLimitMaximum : LREAL;
  Options       : ValveCharacterizationOptions;
END_VAR
VAR_OUTPUT
  Done          : BOOL;
  Busy          : BOOL;
  Active        : BOOL;
  CommandAborted : BOOL;
  Error         : BOOL;
  ErrorId       : UDINT;
  NextMovingToMaximum : BOOL;
  NextMovingToMinimum : BOOL;
END_VAR

```

 Inputs

Name	Type	Default	Description
Execute	BOOL		Start the valve characterization process at rising edge.
PauseAtPositionLimit	BOOL	false	The upcoming move is delayed while PauseAtPositionLimit = TRUE.
OutputStepSize	LREAL	#Invalid	Size of output steps. Must have value [0..1], where 1 = output value changes by 100% between two measurements (0 = no change)
PositionLimitMinimum	LREAL	#Invalid	Minimum position ([PosUnit]). Must have value < PositionLimitMaximum.

Name	Type	Default	Description
PositionLimitMaximum	LREAL	#Invalid	Maximum position ([PosUnit]). Must have value > PositionLimitMinimum.
Options	ValveCharacterizationOptions [► 348]		Additional function block options

In/Outputs

Name	Type	Description
Axis	Reference To AXIS_REF	Reference to the axis

Outputs

Name	Type	Description
Done	BOOL	Valve characterization finished successfully
Busy	BOOL	Function block is not finished and new output values are to be expected.
Active	BOOL	Function block has active control on the axis.
CommandAborted	BOOL	Command is aborted by another command.
Error	BOOL	Error occurred within function block.
ErrorId	UDINT	Error identifier
NextMovingToMaximum	BOOL	Positive move upcoming
NextMovingToMinimum	BOOL	Negative move upcoming

Versionsinformation

- TwinCAT Standard >= v3.1.4026.23.1
- TF5560 MC3 FluidPower >= v4.0.6 (inkludiert TF5500 MC3 Base >= v4.0.6)

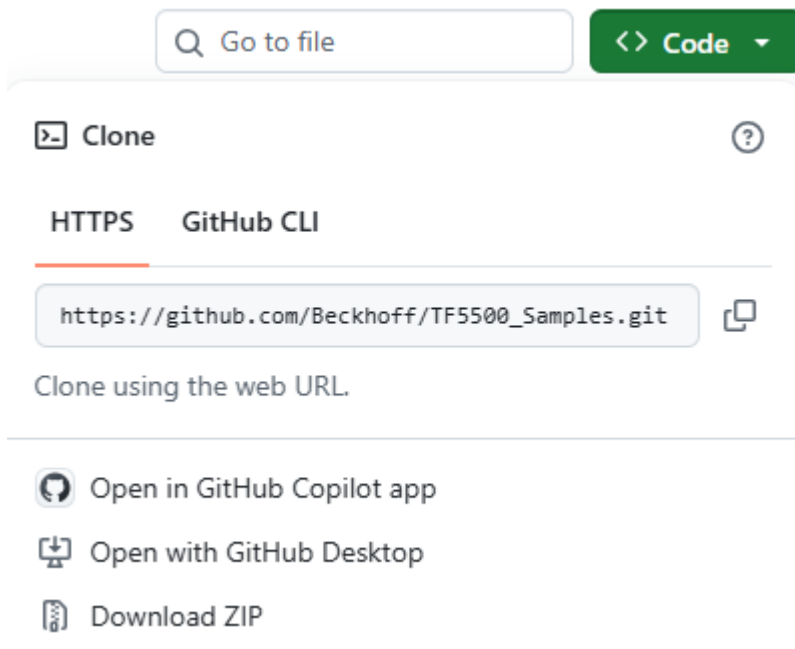
Required License

TC3 MC3 Fluid Power

5.6 Beispielprogramme

Beispielcode und -konfigurationen zu den einzelnen Produktbereichen können über das entsprechende Repository auf GitHub bezogen werden (Links siehe unten).

Sie haben zum Beispiel die Möglichkeit, das Repository zu clonen oder über Klicken von **Download ZIP** ein ZIP-File mit dem Sample herunterzuladen.



TF5500 | TwinCAT 3 MC3 Base

Das Repository zeigt grundlegende MC3-Szenarien wie Gearing, PTP-Bewegungen und Fehlermanagement in gekoppelten Achsen. Es eignet sich, um typische Bewegungsabläufe, Zustandsmaschinen und den praktischen Einsatz zentraler MC3-Bausteine nachzuvollziehen.

- MC3 Version 4.0.6
- https://github.com/Beckhoff/TF5500_Samples

TF5550 | TwinCAT 3 MC3 Camming

Dieses Repository fokussiert Achskopplungen mit Kurvenscheiben mithilfe von Beispielen für MC_CamIn, dem Wechsel von Kurvenscheiben und unterschiedliche Aktivierungsstrategien während der Bewegung. Damit wird klar, wie nichtlineare Kopplungen zwischen Master- und Slave-Achsen in MC3 strukturiert, getestet und umgeschaltet werden.

- MC3 Version 4.0.6
- https://github.com/Beckhoff/TF5550_Samples

TF5560 | TwinCAT 3 MC3 Fluid Power

Hier stehen FluidPower-Anwendungen im Mittelpunkt, insbesondere Ventilcharakterisierung und das Lesen/Schreiben von Kennlinien in CSV-Dateien. Die Beispiele zeigen den Workflow der automatischen Vermessung von Ventilkennlinien.

- MC3 Version 4.0.6
- https://github.com/Beckhoff/TF5560_Samples

6 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Downloadfinder

Unser Downloadfinder beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157

E-Mail: support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460

E-Mail: service@beckhoff.com

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963-0

E-Mail: info@beckhoff.com

Internet: www.beckhoff.com

Trademark statements

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

Third-party trademark statements

Excel, IntelliSense, Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

MAX®, Stratix®, Cyclone®, Altera®, Agilex™, Arria®, Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Sercos is a registered trademark of Sercos International e.V.

Mehr Informationen:
www.beckhoff.com/TF5500

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

