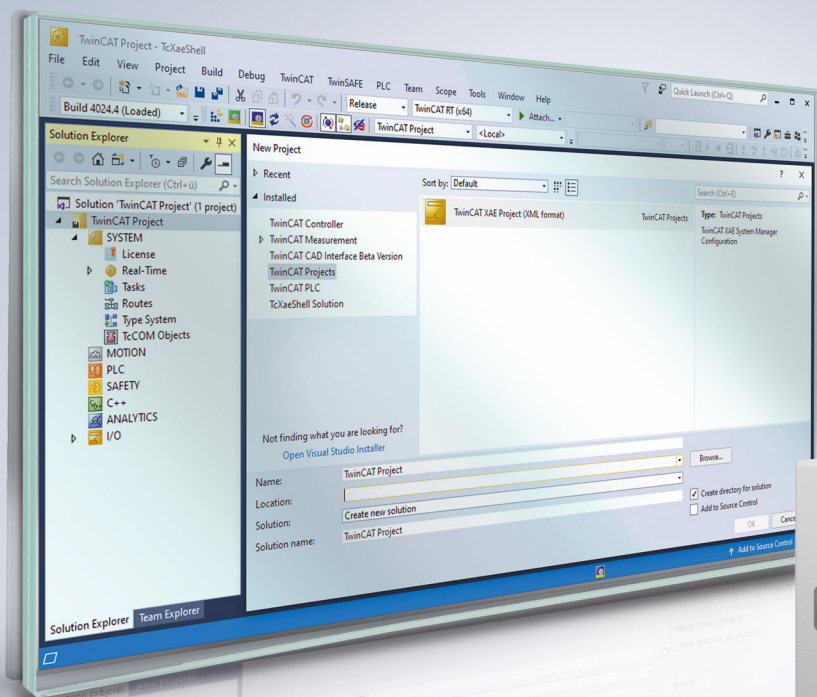


# BECKHOFF New Automation Technology

Handbuch | DE

# TF5890

TwinCAT 3 XPlanar





# Inhaltsverzeichnis

<b>1</b>	<b>Dokumentationshinweise</b>	<b>6</b>
1.1	Disclaimer	6
1.2	Ausgabestände	8
1.3	Dokumentationsumfang	8
1.4	Sicherheit und Einweisung	9
1.4.1	Hinweise zur Informationssicherheit	9
1.5	Symbolerklärung	10
1.5.1	Piktogramme	10
1.6	Beckhoff Services	11
1.6.1	Support-Leistungen	11
1.6.2	Trainingsangebote	11
1.6.3	Service-Leistungen	11
1.6.4	Unternehmenszentrale Deutschland	12
1.6.5	Downloadfinder	12
<b>2</b>	<b>Hinweise zur Informationssicherheit</b>	<b>13</b>
<b>3</b>	<b>Allgemeine Informationen</b>	<b>14</b>
3.1	Kompatibilität	14
3.2	Darstellung der Screenshots	14
<b>4</b>	<b>Installation und Software-Updates</b>	<b>15</b>
4.1	TwinCAT 3   Build 4026	15
4.1.1	Zusätzliche Lizenzanforderungen	15
4.1.2	TwinCAT Package Manager installieren	15
4.1.3	TwinCAT 3   Build 4026 installieren	15
4.1.4	Workloads installieren	16
4.2	TwinCAT 3   4024	17
4.2.1	Zusätzliche Lizenzanforderungen	17
4.2.2	Versionsstand überprüfen	18
4.3	TcCOM-Objekte neu laden	19
4.4	High Level XPlanar Komponenten	20
<b>5</b>	<b>XPlanar Konfigurationsablauf</b>	<b>22</b>
5.1	XPlanar Processing Units	22
5.2	XPlanarPart	23
5.3	XPlanar Tiles	24
5.4	XPlanar Mover	24
5.5	XPlanar Moverkopplung	25
5.6	XPlanar Koordinatensystem	26
5.7	XPlanarMovementArea	27
5.8	XPlanar Mover Parametrierungssätze	28
<b>6</b>	<b>TcCom-Objekte hinzufügen</b>	<b>29</b>
6.1	Processing Unit	29
6.2	Part	30
6.3	Tile	31
6.4	Movement Area	32

6.5	Koordinatensystem .....	33
6.6	Mover .....	34
6.7	Moverkopplung.....	35
6.8	Mover Parametrierung .....	36
<b>7</b>	<b>TcCom-Objekte Basiseinstellungen .....</b>	<b>38</b>
7.1	Tiles.....	39
7.2	Mover .....	39
7.3	MovementArea.....	41
<b>8</b>	<b>TcCOM-Objekte verlinken.....</b>	<b>42</b>
8.1	Mover .....	42
8.1.1	NC3 To HW .....	42
8.1.2	HW To NC3 .....	45
8.2	Tiles.....	47
8.2.1	Drive .....	47
8.2.2	Feedback.....	52
8.2.3	Verlinkungsdetails .....	54
<b>9</b>	<b>Task hinzufügen .....</b>	<b>55</b>
<b>10</b>	<b>Task verlinken .....</b>	<b>56</b>
10.1	Mover .....	56
10.2	Tile .....	56
<b>11</b>	<b>Mover Detection und Mover Identification.....</b>	<b>58</b>
11.1	Detection Mode: AsConfigured .....	58
11.2	Detection Mode: ConfiguredTypesAnyCount.....	60
11.3	Mover Zuordnung.....	62
11.4	Erforderliche Einstellungen .....	63
11.5	Starten der Mover Detection .....	64
11.5.1	Kontrolle der Mover Detection.....	64
11.6	Starten der Mover Identification .....	65
11.6.1	Unterstützte Kombinationen .....	65
11.6.2	Object XPlanarProcessingUnit.....	66
11.6.3	SPS .....	68
<b>12</b>	<b>Moverkopplung.....</b>	<b>69</b>
12.1	Empfohlene Anordnungen bei Kacheln APS4322-0000 .....	69
12.2	Sollwertursprung .....	70
12.3	CPU-Kern.....	72
12.3.1	Moverkopplung 2 x 2 .....	72
12.3.2	Moverkopplung 3 x 2 .....	73
12.4	Submover auswählen.....	74
<b>13</b>	<b>Parametrierungsset auswählen .....</b>	<b>76</b>
<b>14</b>	<b>Mover Parametrierungen .....</b>	<b>77</b>
14.1	General Parametrization .....	77
14.1.1	Flughöhe .....	77
14.1.2	Ohne gemeinsame Referenzfläche.....	78
14.1.3	Mit gemeinsamer Referenzfläche.....	79

14.1.4	Parameter festlegen .....	80
14.2	Observer Parametrization .....	82
14.2.1	Linear axes observer .....	82
14.2.2	Rotational axes observer .....	83
14.2.3	Parameter festlegen .....	84
14.3	Controller Parametrization .....	85
14.3.1	Parameter festlegen .....	86
14.4	Filter Parametrization .....	87
14.4.1	Low Pass 1 .....	87
14.4.2	Low Pass 2 .....	87
14.4.3	Notch .....	87
14.4.4	Bypassed .....	87
14.4.5	Parameter festlegen .....	88
14.5	Inertia Parametrization .....	90
14.5.1	Mover .....	90
14.5.2	Mover und montierte Werkzeug .....	91
14.5.3	Parameter festlegen .....	92
<b>15</b>	<b>SPS-Bibliotheken .....</b>	<b>93</b>
15.1	TwinCAT 3 XPlanar Utility .....	93
15.1.1	Versionsstand überprüfen .....	94
15.1.2	Initialisierung .....	96
15.1.3	PLC-Zugriff .....	99
15.1.4	Visualisierung .....	106
15.1.5	Parameter .....	107

## 1 Dokumentationshinweise

### 1.1 Disclaimer


Beckhoff Produkte werden fortlaufend weiterentwickelt. Wir behalten uns vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten. Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

#### **Dieses Produkt enthält Software.**

Die relevanten rechtlichen Informationen und Lizenzen finden Sie im internen Speicher dieses Produkts oder in den Open-Source-Software-Deklarationen im Downloadbereich des Produkts auf der Beckhoff-Webseite unter

 [www.beckhoff.com/de-de/produkte/](http://www.beckhoff.com/de-de/produkte/)

Sie können auch eine E-Mail an die untenstehende Adresse senden, um weitere Informationen zu erhalten.

 [osscompliance@beckhoff.com](mailto:osscompliance@beckhoff.com)

Durch die Nutzung des Produkts stimmen Sie den relevanten Lizenzen für die im Produkt enthaltene Software zu.

#### 1.1.1 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 Kennzeichen führen.



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

#### 1.1.2 Haftungsbeschränkungen

Die gesamten Komponenten dieses in der Original-Betriebsanleitung beschriebenen Produktes werden je nach Anwendungsbestimmungen in bestimmter Konfiguration von Hardware und Software ausgeliefert. Umbauten und Änderungen der Konfiguration von Hardware oder Software, die über die dokumentierten Möglichkeiten hinausgehen, sind verboten und führen zum Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

#### **Folgendes wird aus der Haftung ausgeschlossen:**

- Nichtbeachtung dieser Dokumentation
- Nichtbestimmungsgemäße Verwendung
- Einsatz von nicht ausgebildetem Fachpersonal
- Verwendung nicht zugelassener Ersatzteile

## 1.1.3 Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland

Die Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet. Zuwiderhandlungen verpflichten zu Schadenersatz.

Wir behalten uns alle Rechte für den Fall der Eintragung der Patente, Gebrauchsmuster und Geschmacksmuster vor.

## 1.1.4 Fremdmarken

In dieser Dokumentation werden Marken und Wortmarken Dritter verwendet. Die zugehörigen Markenvermerke befinden sich unter

 [www.beckhoff.com/trademarks](http://www.beckhoff.com/trademarks)

## 1.2 Ausgabestände

Auf Anfrage erhalten Sie eine Auflistung der Ausgabestände zu Änderungen in der Dokumentation. Senden Sie Ihre Anfrage an:

✉ [support@beckhoff.com](mailto:support@beckhoff.com)

### Dokumentenursprung

Diese Dokumentation ist in deutscher Sprache verfasst. Alle weiteren Sprachen werden von dem deutschen Original abgeleitet.

### Produkteigenschaften

Gültig sind immer die Produkteigenschaften, die in der aktuellen Dokumentation angegeben sind. Weitere Informationen, die auf den Produktseiten der Beckhoff Homepage, in E-Mails oder sonstigen Publikationen angegeben werden, sind nicht maßgeblich.

## 1.3 Dokumentationsumfang

Neben dieser Dokumentation sind folgende Dokumente Bestandteil der Gesamtdokumentation:

### Handbuch | TF5430

🌐 [Direktlink zur Dokumentation TF5430 | TwinCAT 3 Planar Motion](#)

### Original-Betriebsanleitung | XPlanar APS4322

🌐 [Direktlink zur XPlanar APS4322-0000 Original-Betriebsanleitung](#)

### Original-Betriebsanleitung | XPlanar APS42xx

🌐 [Direktlink zur XPlanar APS42xx-1x00 Original-Betriebsanleitung](#)

## 1.4 Sicherheit und Einweisung

Lesen Sie die Inhalte, welche sich auf die von Ihnen durchzuführenden Tätigkeiten mit dem Produkt beziehen. Lesen Sie immer das Kapitel Zu Ihrer Sicherheit in der Dokumentation. Beachten Sie die Warnhinweise in den Kapiteln, sodass Sie ordnungsgemäß und sicher mit dem Produkt umgehen und arbeiten.

### 1.4.1 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 ganzheitlichen 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>.

## 1.5 Symbolerklärung

Für eine übersichtliche Gestaltung werden verschiedene Symbole verwendet:

- ▶ Das Dreieck zeigt eine Handlungsanweisung, die Sie ausführen sollen
- Der Punkt zeigt eine Aufzählung
- [...] Die eckigen Klammern zeigen Querverweise auf andere Textstellen in dem Dokument
- [+] Das Plus-Zeichen in eckiger Klammer zeigt Bestelloptionen und Zubehör

### 1.5.1 Piktogramme

#### HINWEIS

##### Hinweise

Für wichtige Informationen zu dem Produkt werden Hinweise verwendet. Werden diese nicht beachtet, sind mögliche Folgen:

- Funktionsfehler an dem Produkt
- Schäden an dem Produkt
- Schäden an der Umwelt



##### Informationen

Dieses Zeichen zeigt Informationen, Tipps und Hinweise für den Umgang mit dem Produkt oder der Software.



##### Beispiele

Dieses Zeichen zeigt Beispiele für den Umgang mit dem Produkt oder der Software.

## 1.6 Beckhoff Services

Beckhoff und die weltweiten Partnerfirmen bieten einen umfassenden Support und Service.

 [www.beckhoff.com/de-de/support/globale-verfuegbarkeit/](http://www.beckhoff.com/de-de/support/globale-verfuegbarkeit/)

### 1.6.1 Support-Leistungen

Der Beckhoff Support bietet Ihnen technische Beratung bei dem Einsatz einzelner Beckhoff Produkte und Systemplanungen. Die Support-Ingenieure bieten Ihnen kompetente Unterstützung, bei Verständnisfragen ebenso wie bei Inbetriebnahmen.

 +49 5246 963-157

 [support@beckhoff.com](mailto:support@beckhoff.com)

 [www.beckhoff.com/de-de/support/unsere-support-leistungen/](http://www.beckhoff.com/de-de/support/unsere-support-leistungen/)

### 1.6.2 Trainingsangebote

Trainings in Deutschland finden in den Beckhoff Niederlassungen oder nach Rücksprache bei den Kunden vor Ort statt. Beckhoff bietet sowohl Präsenz-Trainings als auch Online-Trainings an.

 +49 5246 963-5000

 [training@beckhoff.com](mailto:training@beckhoff.com)

 [www.beckhoff.com/de-de/support/trainingsangebote/](http://www.beckhoff.com/de-de/support/trainingsangebote/)

### 1.6.3 Service-Leistungen

Die Beckhoff Serviceexperten unterstützen Sie weltweit in allen Bereichen des After Sales Service.


 +49 5246 963-460

 [service@beckhoff.com](mailto:service@beckhoff.com)

 [www.beckhoff.com/de-de/support/unsere-service-leistungen/](http://www.beckhoff.com/de-de/support/unsere-service-leistungen/)

## 1.6.4 Unternehmenszentrale Deutschland

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl, Deutschland

 +49 5246 963-0

 [info@beckhoff.com](mailto:info@beckhoff.com)

 [www.beckhoff.com/de-de/](http://www.beckhoff.com/de-de/)

Eine detaillierte Übersicht über die weltweiten Beckhoff Standorte finden Sie unter:

 [www.beckhoff.com/de-de/unternehmen/globale-praesenz/](http://www.beckhoff.com/de-de/unternehmen/globale-praesenz/)

## 1.6.5 Downloadfinder


Im Downloadfinder finden Sie Konfigurationsdateien, technische Dokumentationen und Applikationsberichte zum Herunterladen.

 [www.beckhoff.com/dokumentationen](http://www.beckhoff.com/dokumentationen)

## 2 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG 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.

Beckhoff empfiehlt folgende Schutzmaßnahmen zur Informationssicherheit und Industrial Security:

 [www.infosys.beckhoff.com](http://www.infosys.beckhoff.com)

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. Weitere Informationen dazu finden Sie unter:

 [www.infosys.beckhoff.com](http://www.infosys.beckhoff.com)

## 3 Allgemeine Informationen

Die vorliegende Bedienungsanleitung hilft Ihnen bei der Inbetriebnahme und Programmierung eines XPlanar Systems.

Die folgenden Kapitel leiten Sie durch die Basiskonfiguration der XPlanar Hardwarekomponenten in *TwinCAT 3*, sodass Sie die XPlanar Mover über die grundlegende MC3 UI-Schnittstelle fahren und in der PLC programmieren können. Das Ziel ist das erste Verfahren der XPlanar Mover und die Überprüfung des Systemkonzepts. Des Weiteren erhalten Sie einen ersten Überblick über alle XPlanar Funktionalitäten der *TF5890 | TwinCAT 3 XPlanar*.



Die Dokumentation der *TF5890 | TwinCAT 3 XPlanar* befindet sich derzeit noch im Aufbau und wird laufend erweitert. Die aktuelle Version dieser Bedienungsanleitung finden Sie unter:

 [www.beckhoff.com/de-de/support/downloadfinder/](http://www.beckhoff.com/de-de/support/downloadfinder/)

### 3.1 Kompatibilität



Zur Inbetriebnahme von XPlanar Systemen mit älteren Software-Versionen ist diese Dokumentation nicht geeignet. Bei Fragen kontaktieren Sie den für Ihre Region zuständigen Produktspezialisten.



Einige der beschriebenen Funktionen sind erst ab *TwinCAT 3 | Build 4026* verfügbar.

Diese Dokumentation bezieht sich auf folgende Softwareversionen:

Software	Version
TF5890   TwinCAT 3 XPlanar	3.20.700.0 oder höher
TF5430   TwinCAT 3 Planar Motion	3.1.6.57 oder höher
TwinCAT 3	3.1.4024.0 oder höher

### 3.2 Darstellung der Screenshots

Für diese Dokumentation wurde die Spracheinstellung Englisch in TwinCAT verwendet. Sämtliche Screenshots werden in dieser Dokumentation exemplarisch in englischer Sprache dargestellt. In den dazugehörigen Handlungsanweisungen werden die Begriffe aus den Screenshots verwendet.

Sämtliche Funktionen und Einstellungen sind unabhängig von der eingestellten Sprache auf Ihrem PC an derselben Stelle in der Software zu finden.

## 4 Installation und Software-Updates



### Softwarestand

Es kann nicht garantiert werden, dass alle hier erklärten Funktionen in TwinCAT 3 | 4024 implementiert sind. Die Entwicklung der *TF5890 | TwinCAT 3 XPlanar* für *TwinCAT 3 | 4024* wurde mit Version **4.2.xxx** beendet. Wenn Sie alle Funktionen nutzen möchten, installieren Sie *TwinCAT 3 | Build 4026* und die neueste Version der *TF5890 | TwinCAT 3 XPlanar*.

Für die Installation der *TF5890 | TwinCAT 3 XPlanar* ist es entscheidend, welche Version von TwinCAT 3 Sie verwenden möchten beziehungsweise welche Version auf Ihrem PC bereits installiert ist. Sie haben die Möglichkeit *TwinCAT 3 Build 4024* oder *TwinCAT 3 Build 4026* zu verwenden.

Beachten Sie die unterschiedlichen Vorgehensweisen bei der Installation der Software:

### TwinCAT 3 | Build 4026

Weitere Informationen dazu finden Sie im Kapitel „TwinCAT 3 | Build 4026“, [Seite 15].

### TwinCAT 3 | Build 4024

Weitere Informationen dazu finden Sie im Kapitel „TwinCAT 3 | 4024“, [Seite 17].

## 4.1 TwinCAT 3 | Build 4026

Zur Installation des Workloads *TF5890 | TwinCAT 3 XPlanar* benötigen Sie:

- TwinCAT Package Manager
- TwinCAT 3 | Build 4026

Außerdem müssen Sie die neueste Version des Workloads *TF5430 | Planar Motion* installieren.

Alle Informationen zur Installation finden Sie im Beckhoff Information System im Bereich *TwinCAT 3 > Installation*

 <https://infosys.beckhoff.com>

### 4.1.1 Zusätzliche Lizenzanforderungen

*TF5890 | TwinCAT 3 XPlanar* benötigt die Lizenz TC1250.

### 4.1.2 TwinCAT Package Manager installieren

Weitere Informationen zur Installation des *TwinCAT Package Managers* finden Sie im Beckhoff Information System:

 [Direktlink Infosys - TwinCAT Package Manager installieren](#)

### 4.1.3 TwinCAT 3 | Build 4026 installieren

Weitere Informationen zur Installation von *TwinCAT 3 | Build 4026* finden Sie im Beckhoff Information System:

 [Direktlink Infosys - TwinCAT 3 | Build 4026 installieren](#)

## 4.1.4 Workloads installieren

Weitere Informationen zur Installation des Workloads *TF5890* | *TwinCAT 3 XPlanar* und *TF5430* | *Planar Motion* finden Sie im Beckhoff Information System:

 [Direktlink Infosys - Workloads installieren](#)

Die Reihenfolge der Installation von den Workloads ist beliebig.

## 4.2 TwinCAT 3 | 4024



### Softwarestand

Es kann nicht garantiert werden, dass alle hier erklärten Funktionen in TwinCAT 3 | 4024 implementiert sind. Die Entwicklung der *TF5890 | TwinCAT 3 XPlanar* für *TwinCAT 3 | 4024* wurde mit Version **4.2.xxx** beendet. Wenn Sie alle Funktionen nutzen möchten, installieren Sie *TwinCAT 3 | Build 4026* und die neueste Version der *TF5890 | TwinCAT 3 XPlanar*.

Bevor Sie ein XPlanar System in Betrieb nehmen können, müssen Sie zunächst die *TF5890 | TwinCAT 3 XPlanar* installieren. Die folgenden Schritte sind auch beim Update auf die neueste Version von *TF5890 | TwinCAT 3 XPlanar* erforderlich.

Wenn Sie bereits eine Softwareversion installiert haben, überprüfen Sie, ob Sie die neueste Version installiert haben. Weitere Informationen finden Sie im Kapitel „Versionsstand überprüfen“, [Seite 18].

Als Voraussetzung für die Installation muss auf Ihrem System eine Version von TwinCAT 3 XAE (TcXaeShell) oder ein kompatibles Microsoft Visual Studio, das mit TwinCAT 3 XAE integriert ist, vorhanden sein.

Die jeweils aktuelle Version der Software finden Sie im Downloadfinder:

 [www.beckhoff.com/de-de/support/downloadfinder/](http://www.beckhoff.com/de-de/support/downloadfinder/)

Zusätzlich zur Software *TF5890 | TwinCAT 3 XPlanar* muss die Software *TF5430 | TwinCAT 3 Planar Motion* ebenfalls auf Ihrem System installiert werden. Weitere Informationen dazu finden Sie in der Dokumentation *TF5430 TwinCAT 3 | Planar Motion*:

 [Direktlink zur Dokumentation TF5430 | TwinCAT 3 Planar Motion](#)

Ob Sie zunächst die Software *TF5430 | TwinCAT 3 Planar Motion* oder die *TF5890 | TwinCAT 3 XPlanar* installieren, ist nicht relevant.

### HINWEIS

#### Als Administrator ausführen

Führen Sie die Installation als Administrator auf Ihrem System aus, um Probleme bei der Installation zu vermeiden.

- ▶ Software **TF5890 | TwinCAT 3 XPlanar** herunterladen und speichern

### 4.2.1 Zusätzliche Lizenzanforderungen

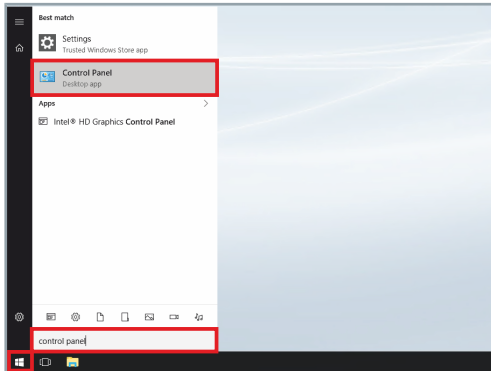
*TF5890 | TwinCAT 3 XPlanar* benötigt die Lizenz *TC1250*.

## 4.2.2 Versionsstand überprüfen

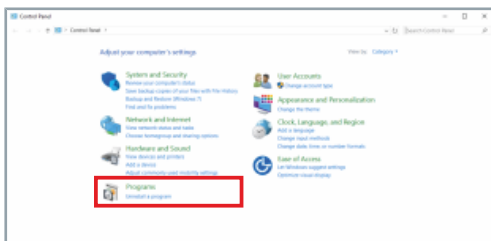


### Betriebssystem beachten

Das Aufrufen des Versionsstandes ist anhängig vom installierten Betriebssystem auf Ihrem Gerät und kann von der hier beschriebenen Variante abweichen.

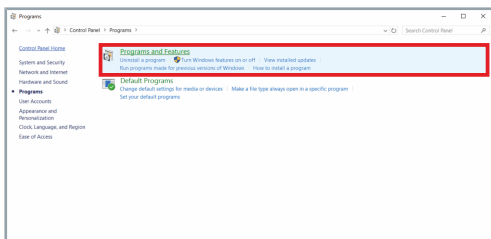


- ▶ Auf **Start** klicken
- ▶ Die Systemsteuerung aufrufen



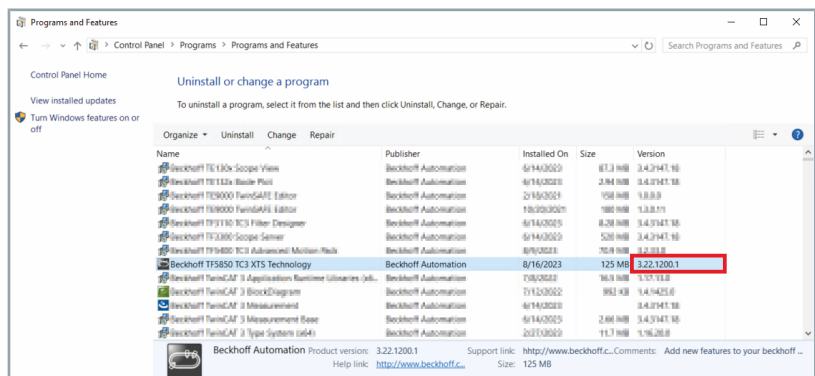
Das Fenster *Control Panel* öffnet sich.

- ▶ Auf **Programs** klicken



Das Fenster *Programs* öffnet sich.

- ▶ Auf **Programs and Features** klicken

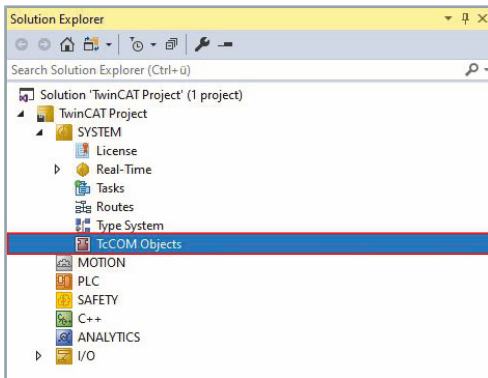


Das Fenster *Programs und Features* öffnet sich.

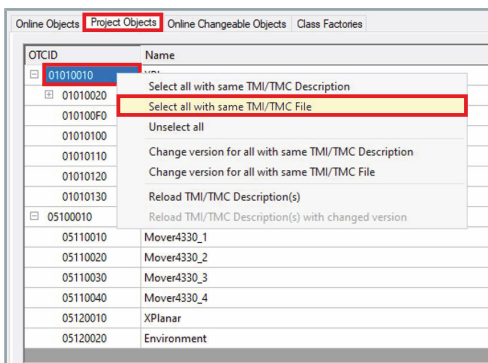
- ▶ Den Versionsstand von Beckhoff **TF5890 TC3 XPlanar Technology** überprüfen

## 4.3 TcCOM-Objekte neu laden

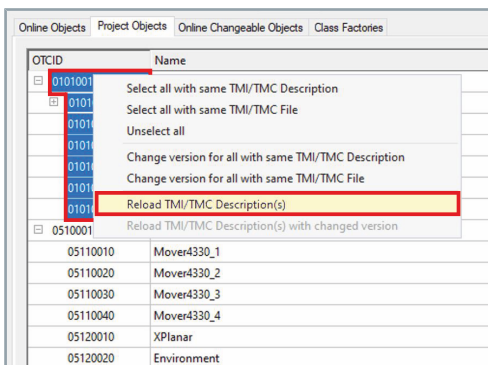
Wenn Sie ein Update der Software *TF5890 | TwinCAT 3 XPlanar* durchgeführt haben, müssen Sie die TcCom-Objekte neu laden.



- ▶ *Solution Explorer* > *TwinCAT Project* > *SYSTEM* ausklappen
- ▶ Doppelklick auf **TcCOM Objects**



- ▶ Im Projektfenster auf den Reiter **Project Objects** klicken
- ▶ Ein Objekt markieren
- ▶ Mit der rechten Maustaste auf das ausgewählte Objekt klicken, um das Kontextmenü aufzurufen
- ▶ Im Kontextmenü auf **Select all with same TMI/TMC file** klicken



- ▶ Mit der rechten Maustaste auf die ausgewählten Objekte klicken, um das Kontextmenü aufzurufen
- ▶ Im Kontextmenü auf **Reload TMI/TMC Description(s)** klicken

OTCID	Name	Version	TMC Filename
01010010	XPlanar	4.4.19.0 (Curr...)	C:\ProgramData\Beckhoff\TwinCAT3.1\Repository\Beckhoff Automation GmbH\TcIoXPlanar\4.4.19.0\TcIoXPlanar.tmc
01010020	Part1	4.4.19.0 (Curr...)	C:\ProgramData\Beckhoff\TwinCAT3.1\Repository\Beckhoff Automation GmbH\TcIoXPlanar\4.4.19.0\TcIoXPlanar.tmc
010100F0	Area1	4.4.19.0 (Curr...)	C:\ProgramData\Beckhoff\TwinCAT3.1\Repository\Beckhoff Automation GmbH\TcIoXPlanar\4.4.19.0\TcIoXPlanar.tmc
01010100	Mover4330_1	4.4.19.0 (Curr...)	C:\ProgramData\Beckhoff\TwinCAT3.1\Repository\Beckhoff Automation GmbH\TcIoXPlanar\4.4.19.0\TcIoXPlanar.tmc
01010110	Mover4330_2	4.4.19.0 (Curr...)	C:\ProgramData\Beckhoff\TwinCAT3.1\Repository\Beckhoff Automation GmbH\TcIoXPlanar\4.4.19.0\TcIoXPlanar.tmc
01010120	Mover4330_3	4.4.19.0 (Curr...)	C:\ProgramData\Beckhoff\TwinCAT3.1\Repository\Beckhoff Automation GmbH\TcIoXPlanar\4.4.19.0\TcIoXPlanar.tmc
01010130	Mover4330_4	4.4.19.0 (Curr...)	C:\ProgramData\Beckhoff\TwinCAT3.1\Repository\Beckhoff Automation GmbH\TcIoXPlanar\4.4.19.0\TcIoXPlanar.tmc
05100010	Context1		
05110010	Mover4330_1		
05110020	Mover4330_2		
05110030	Mover4330_3		
05110040	Mover4330_4		
05120010	XPlanar		
05120020	Environment		

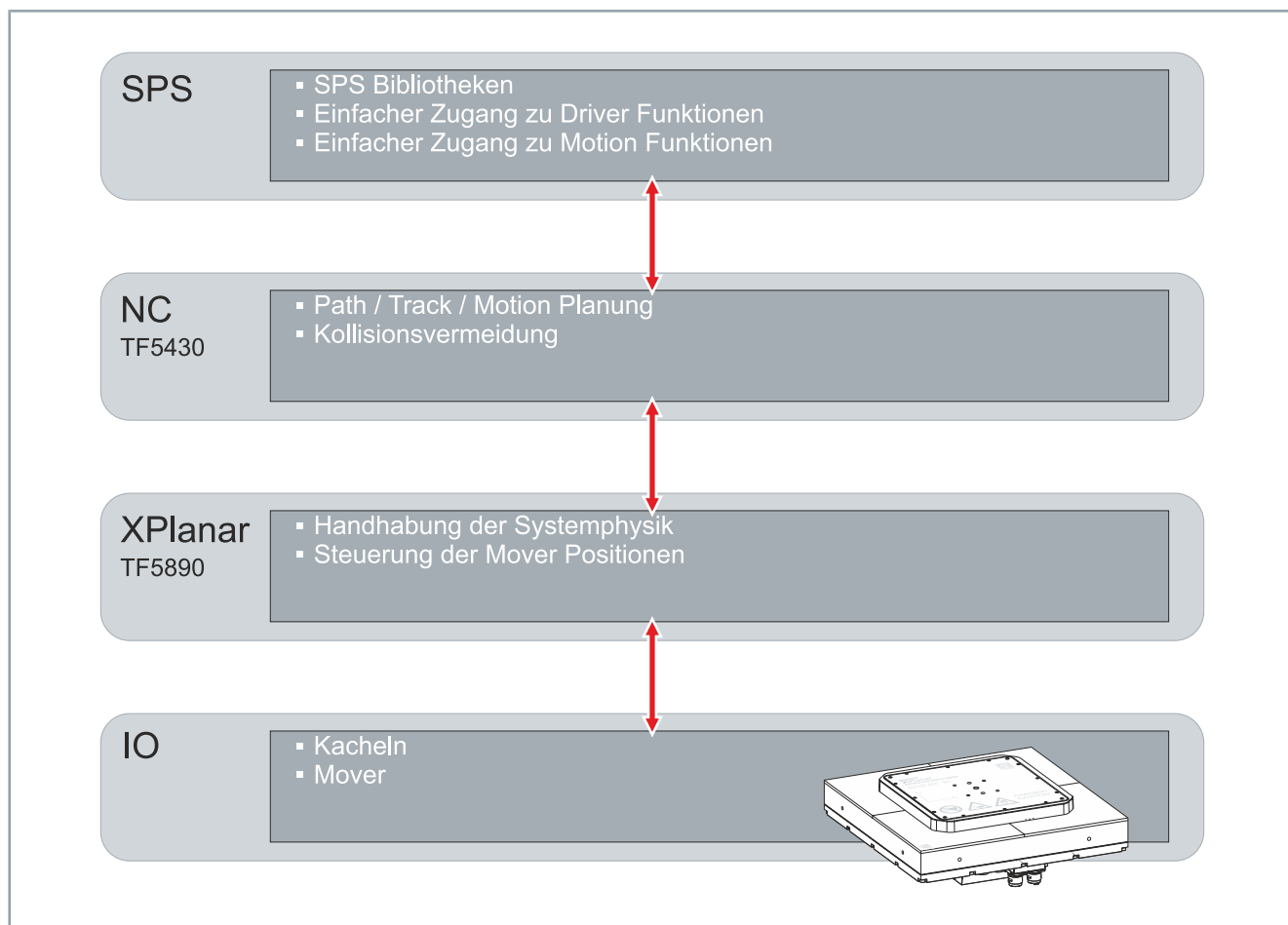
Nachdem die TMI/TMC-Beschreibungen neu geladen wurden, müssen Sie überprüfen, ob alle XPlanar Objekte die korrekte Version des Treibers und die korrekte TMC-Datei verwenden.

## 4.4 High Level XPlanar Komponenten

Das XPlanar-System besteht aus:

- Kacheln
  - *APS4322-0000* oder *APS42xx-1x00*. Die Kacheln enthalten die gesamte erforderliche Leistungselektronik, Messelektronik und Kommunikationselektronik
- Movern
  - *APM4220-0000*, *APM4221-0000*, *APM4330-000x* und *APM4550-0000*
- skalierbarem IPC

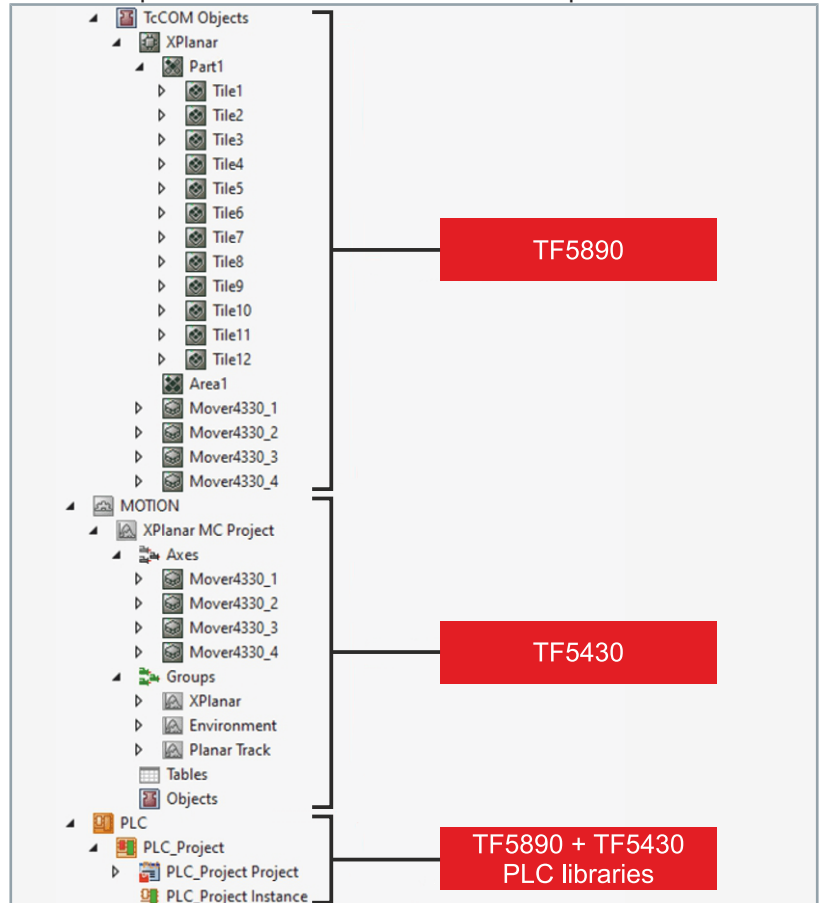
Die Software *TF5890* | *TwinCAT 3 XPlanar* ermöglicht das Anlegen von entsprechenden Objekten im TwinCAT-Konfigurationsbaum, die als Schnittstellen zwischen den Hardwarekomponenten und der Automatisierungssoftware fungieren.



Die XPlanar-Software erfordert die Abstimmung zwischen mehreren Komponenten:

- TcCom-Objekte
  - *TF5890* | *TwinCAT 3 XPlanar*
  - *TF5430* | *TwinCAT 3 Planar Motion*
- SPS
  - Planar Motion und Physics-Bibliotheken, die von der *TF5430* bereitgestellt werden, und den Zugriff auf High-Level-Bewegungsbefehle des XPlanar Systems ermöglichen
  - Die XPlanar Utility-Bibliothek, die von der *TF5890* bereitgestellt wird, und den Zugriff auf die Treiberobjekte und deren Parameter ermöglicht

Diese Komponenten lassen sich im Solution Explorer wiederfinden:

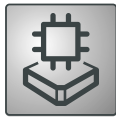


## 5 XPlanar Konfigurationsablauf

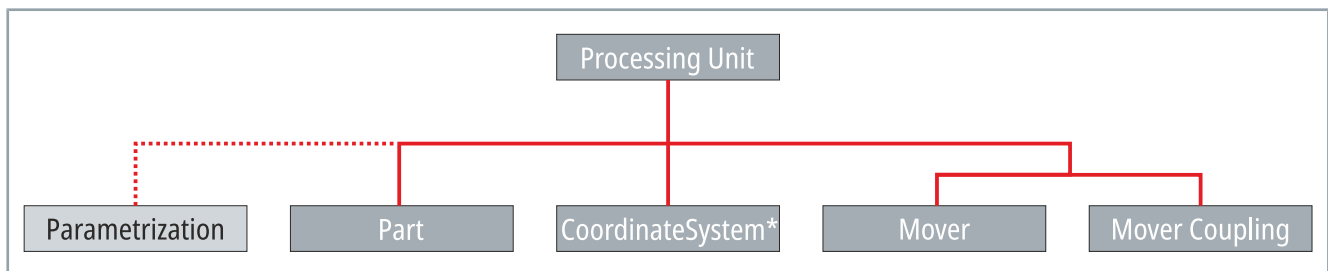
Bevor Sie ein XPlanar System nutzen können, muss es zunächst mit allen Komponenten und allen Einstellungen von *Real Time* und *Distributed Clock* vollständig konfiguriert werden.

Die wichtigsten Schritte bei der Konfiguration eines XPlanar Systems sind folgende:

### 5.1 XPlanar Processing Units



Die *XPlanar Processing Unit* ist das Zentrum des XPlanar Systems. In der *XPlanar Processing Unit* laufen alle erforderlichen Objekte zusammen und werden logisch miteinander verknüpft. Ein XPlanar System kann mehrere *XPlanar Processing Units* enthalten, die als unabhängige Maschinen fungieren.



\* Das Koordinatensystem ersetzt die MovementArea.

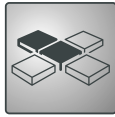
Zu einer *Processing Unit* gehören immer Kacheln, Mover oder Moverkopplungen, Parts und Koordinatensystem. Je nach Bedarf können noch Parametrierungssätze hinzugefügt werden.

Sie können eine oder mehrere *XPlanar Processing Units* anlegen, je nach Anzahl der einzelnen XPlanar Systeme, die in der Maschine verwendet werden. Eine *XPlanar Processing Unit* organisiert verschiedene XPlanar Objekte, die für Kacheln und Mover angelegt werden müssen.

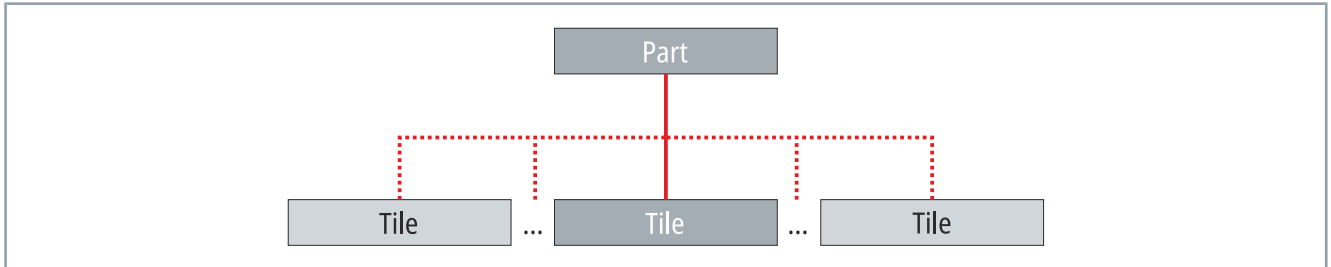


Für weitere Informationen über die benötigten *XPlanar Tasks* für Ihr XPlanar System kontaktieren Sie den für Ihre Region zuständigen Produktspezialisten.

## 5.2 XPlanarPart



Ein *XPlanarPart* ist die mindestens erforderliche Komponente eines XPlanar Systems. Ein *XPlanarPart* besteht aus ein oder mehreren untergeordneten *XPlanarTiles*.



Wenn mehrere Parts verwendet werden, muss ein *XPlanarCoordinateSystem* hinzugefügt werden.

## 5.3 XPlanar Tiles



Ein *XPlanar Tile* ist der kleinste mögliche *XPlanar Part*. Ein *XPlanar Tile* entspricht einer physischen XPlanar Kachel *APS4xxx-xx00*. Ein *XPlanar Tile* ist immer ein untergeordnetes Objekt eines *XPlanar Parts*.



### Simulationsbetrieb oder Normalbetrieb

Für den Simulationsbetrieb des Treibers wird keine Hardware benötigt.

Für den Normalbetrieb muss jedem *XPlanar Tile* eine physische XPlanar Kachel *APS4xxx* zugeordnet werden.

Jedem *XPlanar Tile* muss im Menü Context eine Task zugewiesen werden, um aktive Berechnungen ausführen zu können. Alle *XPlanar Tiles*, die zu einem EtherCAT Master gehören, müssen auf derselben Task laufen. Nur *XPlanar Tiles* verfügen über EtherCAT Kommunikationsschnittstellen.

Ein *XPlanar Tile* kann in einem festgelegten Raster angeordnet werden:

- APS42xx-1x00: 40 mm Raster
- APS4322-0000: 48 mm Raster

Weitere Informationen dazu finden Sie in der XPlanar Original-Betriebsanleitung:



[Direktlink zur XPlanar APS4322-0000 Original-Betriebsanleitung](#)



[Direktlink zur XPlanar APS42xx-1x00 Original-Betriebsanleitung](#)



Die Funktionalität einiger Treiber-Features ist für versetzt angeordnete Kacheln nicht verfügbar.

## 5.4 XPlanar Mover



Die *XPlanar Mover* sind zusammen mit den Kacheln die Hauptkomponenten des XPlanar Systems. Die *XPlanar Mover* fahren auf den Kacheln.

Ein *XPlanar Mover* TcCom-Objekt entspricht einem physischen Mover.



### Simulationsbetrieb oder Normalbetrieb

Für den Simulationsbetrieb des Treibers wird keine Hardware benötigt.

Für den Normalbetrieb muss jedem *XPlanar Mover* ein physischer XPlanar Mover *APM4xx0* zugeordnet werden.

Jedem *XPlanar Mover* muss eine Task zugewiesen werden, um aktive Berechnungen ausführen zu können.

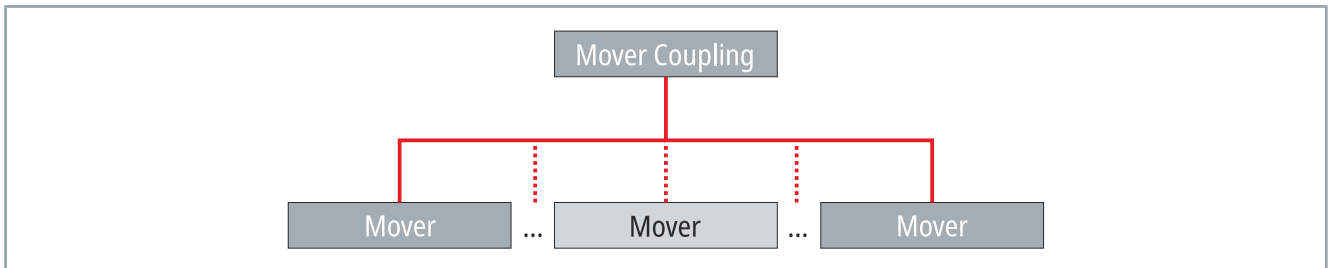
Je nach Konfiguration und Task können bis zu zwei Mover einem CPU-Kern zugewiesen werden, der die aktuelle Position und die Sollposition berechnet.

Die *XPlanar Mover* verfügen über eine Schnittstelle zu externen Positionssollwerten. Über die Schnittstelle werden Fahrbefehle an die *XPlanar Mover* gegeben.

## 5.5 XPlanar Moverkopplung



Ein *XPlanarRigidMoverCoupling* Objekt realisiert die physische, starre Kopplung mehrerer XPlanar Mover. Für ein *XPlanarRigidMoverCoupling* Objekt werden mindestens zwei Mover benötigt. Jeweils zwei Mover einer Kopplung können sich einen Kern teilen. Bei Verwendung von Kacheln *APS4322-0000* empfiehlt Beckhoff, die Submover mit einem Abstand von 240 mm von Mittelpunkt zu Mittelpunkt anzuordnen. Die verwendeten Mover werden von der Software *TF5430 | TwinCAT 3 Planar Motion* und der SPS als ein Moverobjekt *XPlanarRigidMoverCoupling* betrachtet. Zur Steuerung der Moverkopplung dient der konfigurierte Sollwertursprung. Weitere Informationen zu den Moverkopplungen finden Sie im Kapitel „Moverkopplung“, [Seite 69].



## 5.6 XPlanar Koordinatensystem



### Koordinatensystem ersetzt die MovementArea

Das XPlanarCoordinateSystem ersetzt die bisherige XPlanarMovementArea.

Ein Part kann an einer oder mehreren Positionen in einem oder mehreren Koordinatensystemen positioniert werden.

Sobald der Treiber gestartet wird, registriert sich jeder Part mit seiner definierten Position zu dem entsprechenden Koordinatensystem. Anschließend werden automatisch für jedes Koordinatensystem MovementAreas mit allen möglichen Anordnungen der Parts erzeugt.

Wenn keine Koordinatensystem Position für den Part erzeugt werden kann, verwendet der Part seine eigene MovementArea. Deshalb muss kein Koordinatensystem Objekt erzeugt werden, wenn ausschließlich ein Part verwendet wird oder keine Verbindung zwischen verschiedenen Parts gewünscht ist.

Wenn ein Part niemals aus einem Koordinatensystem entfernt werden soll, muss der Parameter *IsFixed* auf TRUE gestellt werden. Dieses Vorgehen reduziert die Anlaufzeit und benötigt weniger Speicherkapazität.

Der Parameter *ActiveIndex* zeigt an, an welcher Position der Part im XPlanarCoordinateSystem zugeordnet ist:

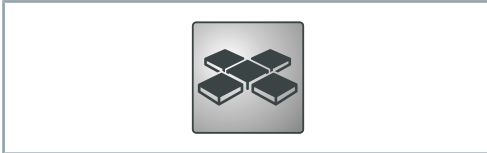
- Standardmäßig wird der Part *ActiveIndex* auf 0 gesetzt. Das bedeutet, dass der Part seine eigene MovementArea verwendet.
- Ist der Part *ActiveIndex* >0, wird die MovementArea aus der Liste ausgewählt. Das entsprechende CoordinateSystem startet mit Index 1.

## 5.7 XPlanarMovementArea

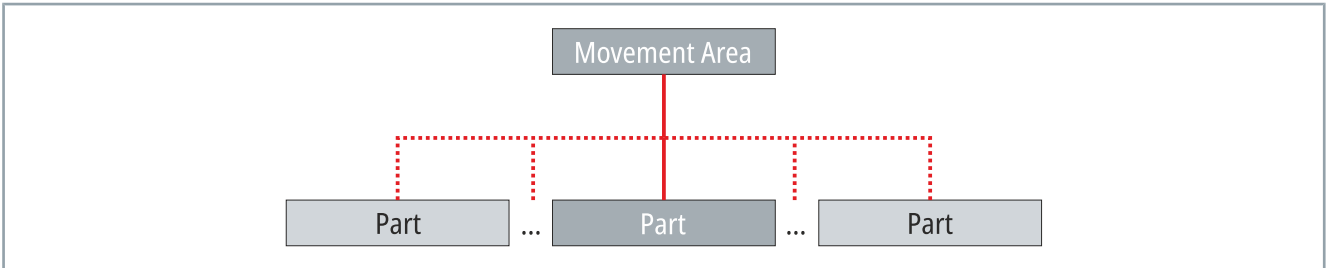


### XPlanarCoordinateSystem wird XPlanarMovermentArea ersetzt

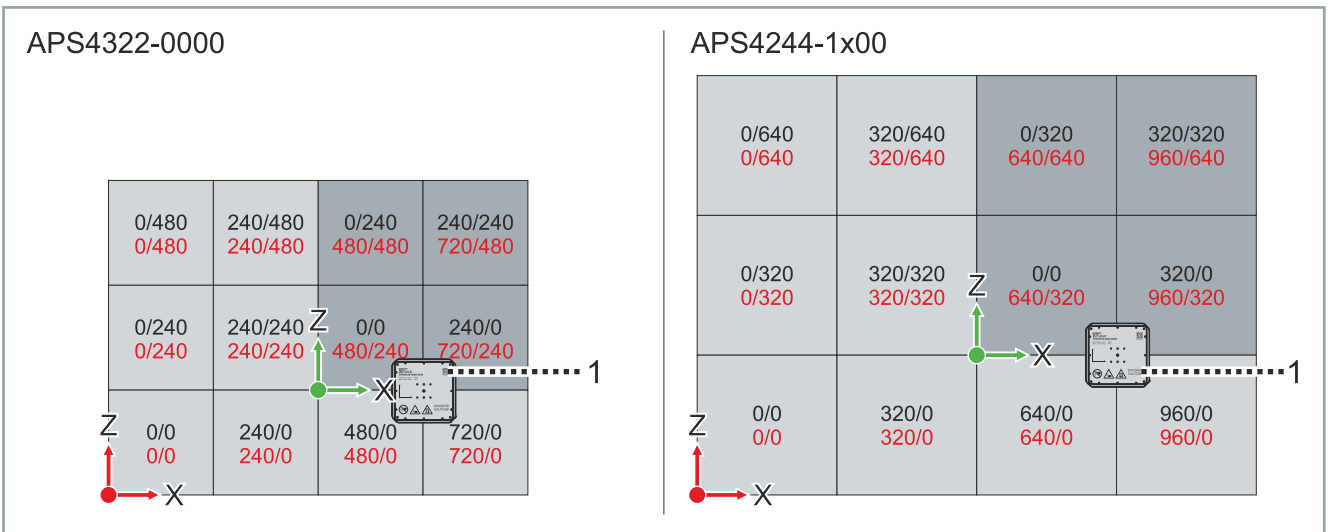
Die *XPlanarMovermentArea* wird schrittweise durch das *XPlanarCoordinateSystem* ersetzt.



Eine *XPlanarMovementArea* besteht aus einem oder mehreren *XPlanarParts*. Jeder *XPlanarPart* hat seinen Ursprung relativ zum Ursprung der *XPlanarMovementArea*.



Ein *XPlanarPart* hat seinen Koordinatenursprung relativ zum Koordinatenursprung der *XPlanarMovement Area*. In der folgenden Grafik sehen Sie eine *XPlanarMovementArea* bestehend aus zwei Parts mit den entsprechenden Koordinaten der *XPlanarParts* und in Bezug auf die Position in der *XPlanarMovementArea*:

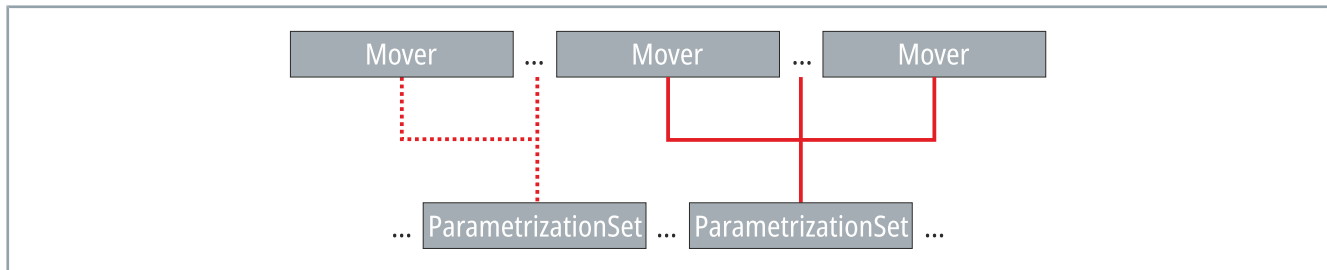


Position	Erläuterung
1	Mover
	XPlanar Part 1
	XPlanar Part 2
+	XPlanar Movement Area
	Koordinatenursprung • Part 1 • Movement Area
	Koordinatenursprung Part 2

## 5.8 XPlanar Mover Parametrierungssätze



Die Mover und Moverkopplungen können auf *XPlanar Mover Parametrisationsets* zugreifen, wenn die in der Software vorhandenen Standardparametrierungen nicht verwendet werden sollen. In den Parametrierungen können sowohl allgemeine Parameter als auch bestimmte Beobachterparameter, Regelungsparameter, Filterparameter und Trägheitsparameter eingestellt werden.



Ein oder mehrere Mover und Moverkopplungen können gleichzeitig auf ein *XPlanar Mover Parametrisationset* zugreifen. Ein Mover kann jedoch immer nur auf einen Parametrierungssatz gleichzeitig zugreifen. Ein dynamischer Wechsel des Parametrierungssatzes ist während des Betriebs möglich.

Bei Änderungen an einem Parametrierungssatz werden diese Änderungen auf alle Mover und Moverkopplungen angewendet, die diesen Parametrierungssatz benutzen.

## 6 TcCom-Objekte hinzufügen



### Manuelles Anlegen oder Verwendung von TwinCAT Tools

Sie haben die Möglichkeit die Hardwarekomponenten und die TcCom-Objekte manuell oder schnell und einfach über das TwinCAT Tool XPlanar Configurator anzulegen. Weitere Informationen dazu Sie im Kapitel .

In diesem Kapitel wird beschrieben, wie Sie Hardwarekomponenten und TcCom-Objekte manuell und ohne die Hilfe von XPlanar Tools anlegen

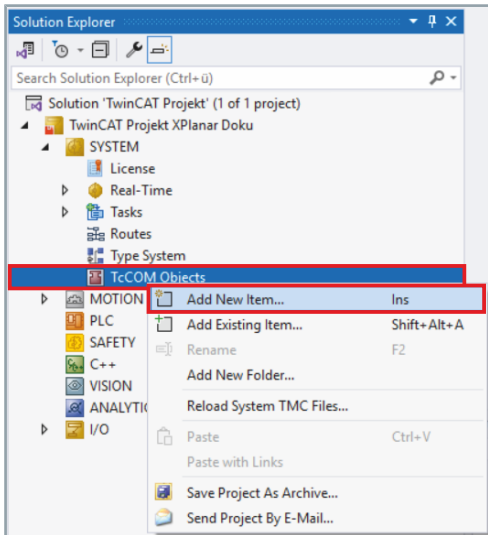
Die TcCom-Objekte müssen bei einer manuellen Konfiguration in der folgenden Reihenfolge hinzugefügt werden:

- Processing Unit
- Part
- Tile
- Movement Area | CoordinateSystem
- Mover | Moverkopplung
- Mover Parametrierung

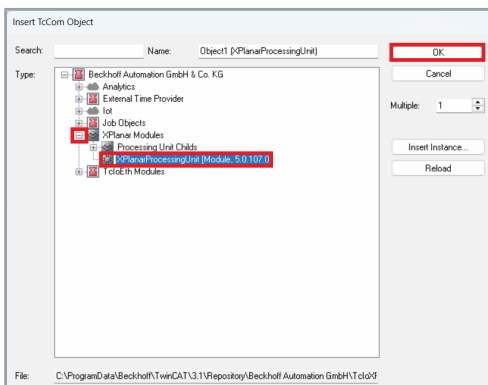


Das Anlegen der TcCom-Objekte wird an einer Beispielkonfiguration mit einem XPlanar System bestehend aus 2 x 3 Kacheln und vier Movern APS4330-0000 gezeigt. Die Abbildungen zeigen entsprechende Einstellungen für diese Beispielkonfiguration.

### 6.1 Processing Unit



- ▶ *Solution-Explorer* > *System* ausklappen
- ▶ Mit der rechten Maustaste auf **TcCOM Objects** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken

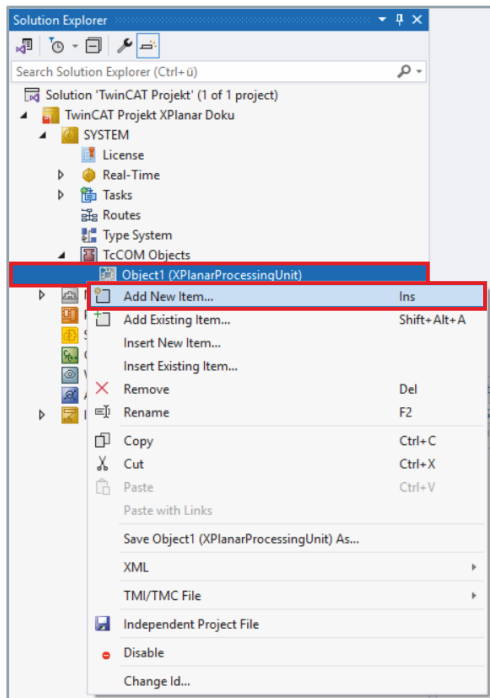


Das Dialogfenster *Insert TcCom Object* öffnet sich.

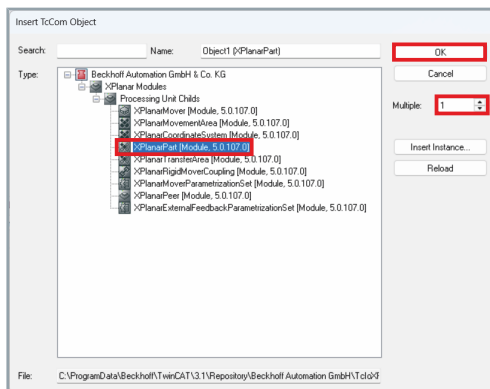
- ▶ Schaltfläche **+** der *XPlanar Modules* anklicken, um die verfügbaren TcCom-Objekte zu sehen
- ▶ **XPlanarProcessingUnit [Module]** auswählen
- ▶ Doppelklick auf die Auswahl oder mit **OK** bestätigen

Eine Processing Unit wird zu den TcCom-Objekten hinzugefügt.

## 6.2 Part



- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* ausklappen
- ▶ Mit der rechten Maustaste auf **Object (XPlanarProcessingUnit)** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken

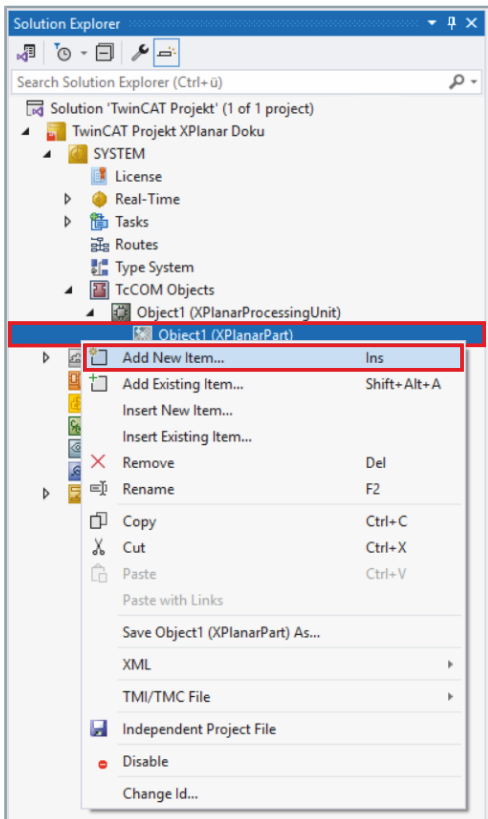


Das Dialogfenster *Insert TcCom Object* öffnet sich.

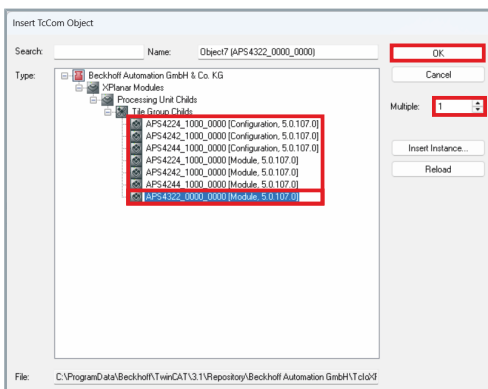
- ▶ Schaltfläche **+** der *XPlanar Modules* anklicken
- ▶ Schaltfläche **+** der *Processing Unit Childs* anklicken, um die verfügbaren TcCom-Objekte zu sehen
- ▶ **XPlanarPart [Module]** auswählen
- ▶ Anzahl der benötigten Parts eingeben
- ▶ Doppelklick auf die Auswahl oder mit **OK** bestätigen

Ein oder mehrere Parts werden zur Processing Unit hinzugefügt.

## 6.3 Tile



- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* ausklappen
- ▶ Mit der rechten Maustaste auf **Object (XPlanarPart)** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken



Das Dialogfenster *Insert TcCom Object* öffnet sich.

- ▶ Schaltfläche **+** der *XPlanar Modules* anklicken
- ▶ Schaltfläche **+** der *Processing Unit Childs* anklicken, um die verfügbaren TcCom-Objekte zu sehen
- ▶ **APS4xxx-x000-0000** auswählen
- ▶ Anzahl der benötigten Tiles eingeben
- ▶ Doppelklick auf die Auswahl oder mit **OK** bestätigen

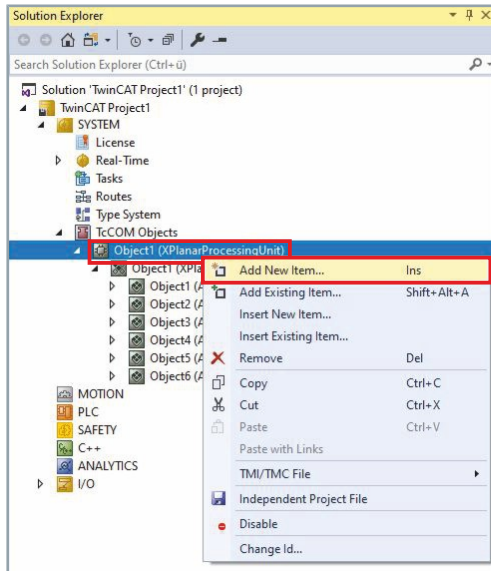
Ein oder mehrere Tiles werden zum XPlanar Part hinzugefügt.

## 6.4 Movement Area

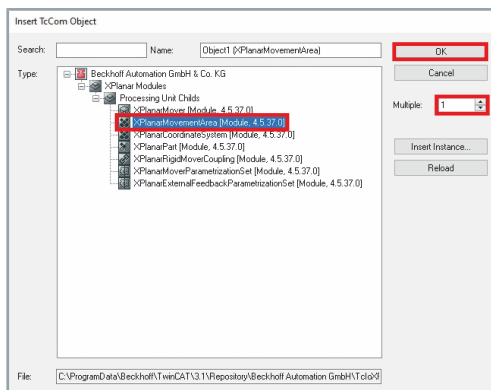


### Koordinatensystem statt MovementArea

Die MovementArea wird langfristig durch das Koordinatensystem ersetzt. Beckhoff empfiehlt, keine neuen MovementAreas anzulegen und stattdessen Koordinatensysteme zu verwenden. Weitere Informationen dazu finden Sie im Kapitel „Koordinatensystem“, [Seite 33].



- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* ausklappen
- ▶ Mit der rechten Maustaste auf **Object (XPlanarProcessingUnit)** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken

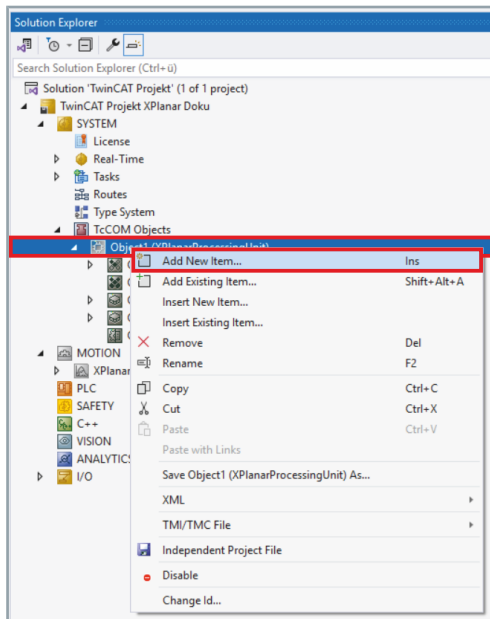


Das Dialogfenster *Insert TcCom Object* öffnet sich.

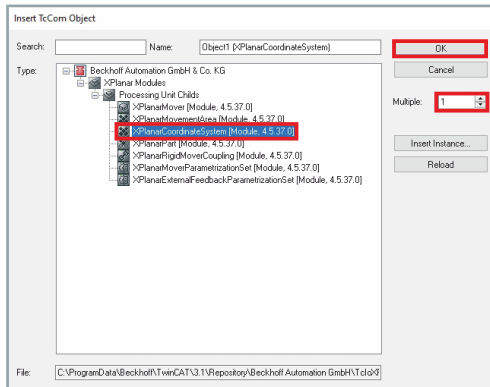
- ▶ Schaltfläche **+** der *XPlanar Modules* anklicken
- ▶ Schaltfläche **+** der *Processing Unit Childs* anklicken, um die verfügbaren TcCom-Objekte zu sehen
- ▶ **XPlanarMovementArea [Module]** auswählen
- ▶ Anzahl der benötigten Movement Areas eingeben
- ▶ Doppelklick auf die Auswahl oder mit **OK** bestätigen

Eine oder mehrere Movement Areas werden zur Processing Unit hinzugefügt.

## 6.5 Koordinatensystem



- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* ausklappen
- ▶ Mit der rechten Maustaste auf **Object (XPlanarProcessingUnit)** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken

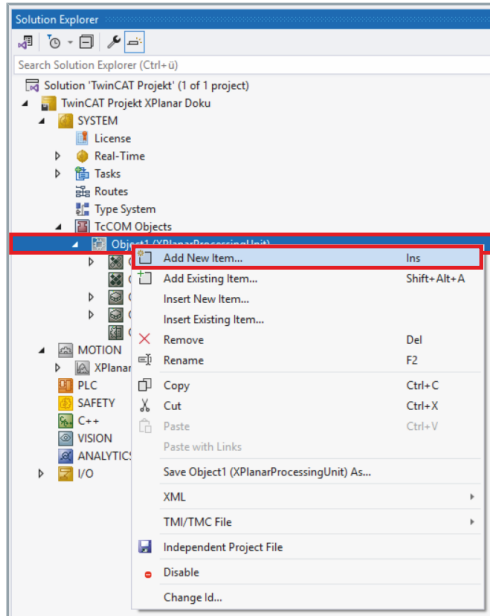


Das Dialogfenster *Insert TcCom Object* öffnet sich.

- ▶ Schaltfläche **+** der *XPlanar Modules* anklicken
- ▶ Schaltfläche **+** der *Processing Unit Childs* anklicken, um die verfügbaren TcCom-Objekte zu sehen
- ▶ **XPlanarCoordinateSystem** auswählen
- ▶ Anzahl der benötigten Koordinatensysteme eingeben
- ▶ Doppelklick auf die Auswahl oder mit **OK** bestätigen

Ein oder mehrere Koordinatensysteme werden zur Processing Unit hinzugefügt.

## 6.6 Mover

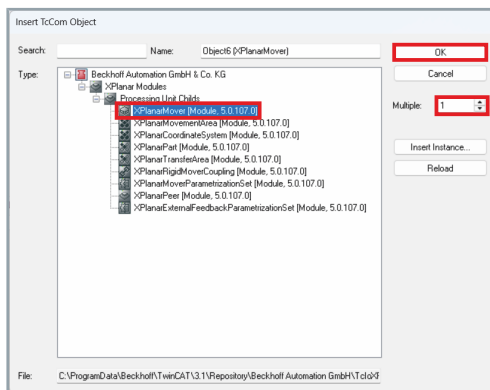


- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* ausklappen
- ▶ Mit der rechten Maustaste auf **Object (XPlanarProcessingUnit)** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken



### Festlegung des Movertyps

Zu den TcCom-Objekten wird zu Beginn ein universelles Mover Objekt hinzugefügt. Die Festlegung des Movertyps erfolgt später im Reiter Parameter (Init). Weitere Informationen zur Festlegung des Movertyps finden Sie im Kapitel „Mover“, [Seite 39].



Das Dialogfenster *Insert TcCom Object* öffnet sich.

- ▶ Schaltfläche **+** der *XPlanar Modules* anklicken
- ▶ Schaltfläche **+** der *Processing Unit Childs* anklicken, um die verfügbaren TcCom-Objekte zu sehen
- ▶ **XPlanarMover [Module]** auswählen
- ▶ Anzahl der benötigten Mover eingeben
- ▶ Doppelklick auf die Auswahl oder mit **OK** bestätigen

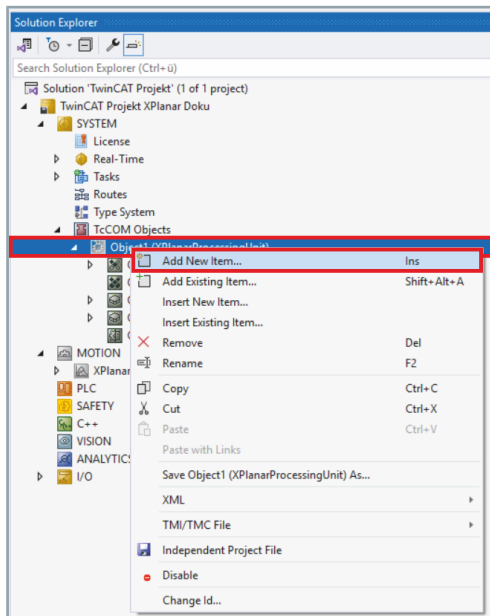
Ein oder mehrere Mover werden zur Processing Unit hinzugefügt.

## 6.7 Moverkopplung

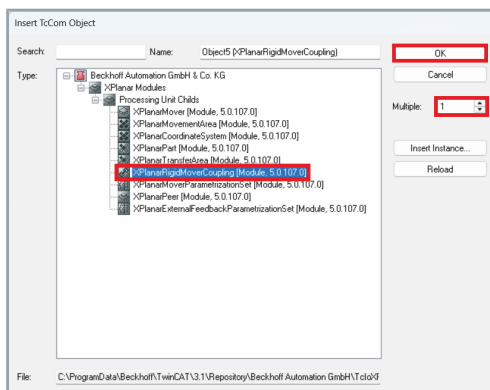


### Moverkopplungen nur bei Bedarf hinzufügen

Die Moverkopplungen müssen Sie nur anlegen, wenn Sie Moverkopplungen auf ihrem System nutzen wollen. Weitere Informationen zu den Moverkopplungen finden Sie im Kapitel „Moverkopplung“, [Seite 69].



- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* ausklappen
- ▶ Mit der rechten Maustaste auf **Object (XPlanarProcessingUnit)** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken



Das Dialogfenster *Insert TcCom Object* öffnet sich.

- ▶ Schaltfläche **+** der *XPlanar Modules* anklicken
- ▶ Schaltfläche **+** der *Processing Unit Childs* anklicken, um die verfügbaren TcCom-Objekte zu sehen
- ▶ **XPlanarRigidMoverCoupling [Module]** auswählen
- ▶ Anzahl der benötigten Moverkopplungen eingeben
- ▶ Doppelklick auf die Auswahl oder mit **OK** bestätigen

Ein oder mehrere Moverkopplungen werden zur Processing Unit hinzugefügt.

## 6.8 Mover Parametrierung



### Parametrierungen nur bei Bedarf hinzufügen

Die Parametrierungen und Parametrierungssätze müssen Sie nur anlegen, wenn Sie nicht die voreingestellten Standardparameter der Mover nutzen möchten. Eine Liste der verfügbaren Parametrierungen finden Sie im Kapitel `XPlanarMoverParametrizationSet`

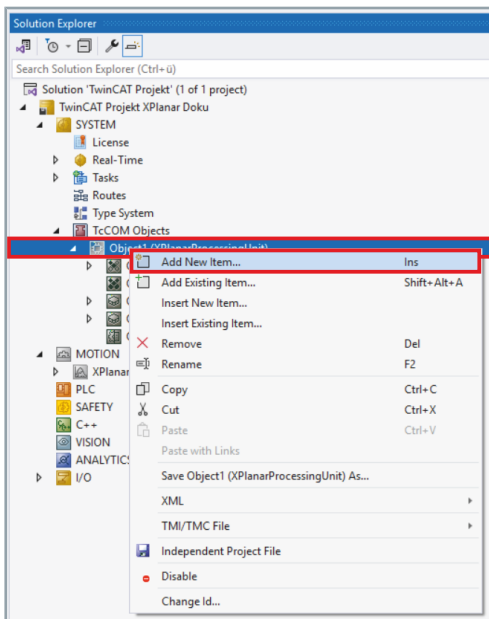
Zusätzlich zu den Hardware-Komponenten können Sie noch ein oder mehrere Parametrierungssätze zur *Processing Unit* hinzufügen.

In einem Parametrierungssatz können Sie die Parameter für einzelne oder mehrere Mover auf dem System ändern und zuweisen, ohne dass die anderen Mover ihre zugewiesenen Parameter verlieren. Außerdem ist der Wechsel zwischen verschiedenen Parametrierungssätzen während des Betriebs möglich. In einem Parametrierungssatz können unterschiedliche Parametrierungen zusammengefasst werden.

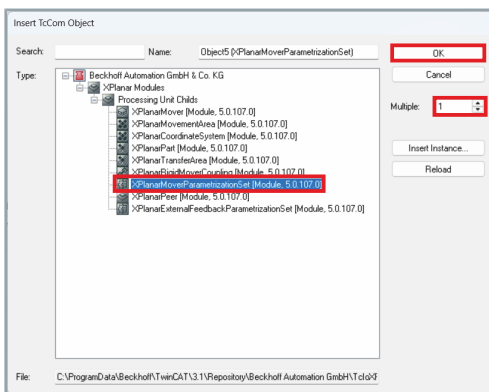


### Parametrierungen nur in Parametrierungssätzen

Um Parametrierungen nutzen zu können, müssen diese in einem Parametrierungssatz angelegt werden. Auch eine einzelne Parametrierung muss einem Parametrierungssatz zugewiesen werden.



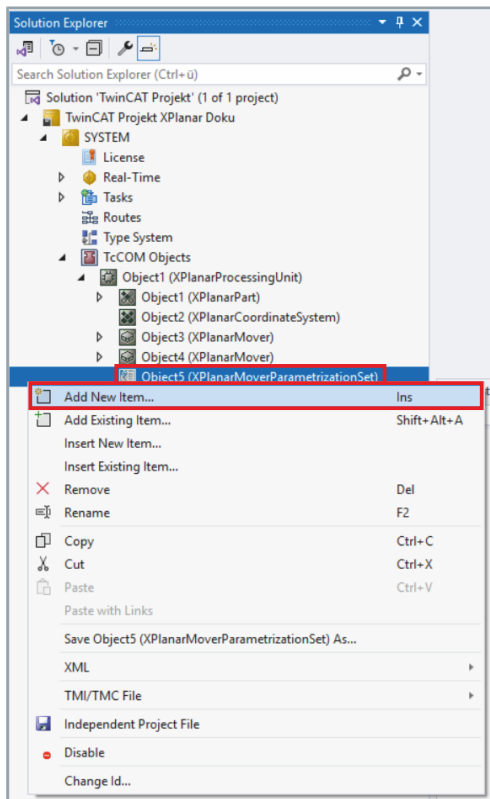
- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* ausklappen
- ▶ Mit der rechten Maustaste auf **Object (XPlanarProcessingUnit)** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken



Das Dialogfenster *Insert TcCom Object* öffnet sich.

- ▶ Schaltfläche **+** der *XPlanar Modules* anklicken
- ▶ Schaltfläche **+** der *Processing Unit Childs* anklicken, um die verfügbaren TcCom-Objekte zu sehen
- ▶ **XPlanarMoverParametrizationSet [Module]** auswählen
- ▶ Anzahl der benötigten Parametrierungssätze eingeben
- ▶ Doppelklick auf die Auswahl oder mit **OK** bestätigen

Ein oder mehrere Parametrierungssätze werden zur *Processing Unit* hinzugefügt.

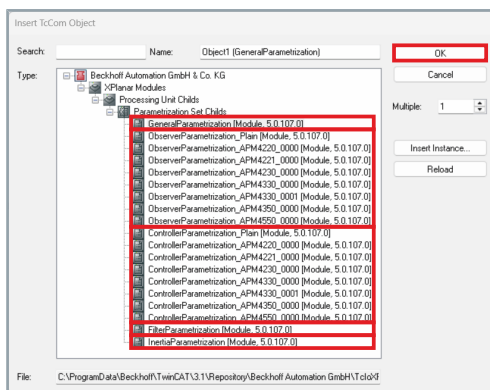


- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* ausklappen
- ▶ Mit der rechten Maustaste auf **Object (XPlanarMoverParametrizationSet)** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken

## HINWEIS

Jeder Parametrierungstyp kann nur einmal in einem Parametrierungssatz enthalten sein.

Es ist nicht möglich in einem Parametrierungssatz beispielsweise eine *ControllerParametrization\_Plain* und eine *ControllerParametrization\_APM4330* zu verwenden, da beide Parametrierungen des gleichen Typs sind.



Das Dialogfenster *Insert TcCom Object* öffnet sich.

- ▶ Schaltfläche **+** der *XPlanar Modules* anklicken
- ▶ Schaltfläche **+** der *Processing Unit Childs* anklicken
- ▶ Schaltfläche **+** der *Parametrization Set Childs* anklicken, um die verfügbaren Parametrierungen zu sehen

Unterschiedliche Parametrierungen stehen zur Auswahl:

- GeneralParametrization
- ObserverParametrization\_XXXX
- ControllerParametrization\_XXXX
- FilterParametrization
- InertiaParametrization
- ▶ Parametrierung auswählen
- ▶ Doppelklick auf die Auswahl oder mit **OK** bestätigen

Die ausgewählte Parametrierung wird zum Parametrierungssatz hinzugefügt.

- ▶ Weitere Parametrierungen auf dieselbe Weise zum Parametrierungssatz hinzufügen

## 7 TcCom-Objekte Basiseinstellungen

Um das XPlanar System starten zu können, müssen noch einige Parametereinstellungen für die Tiles, Mover und MovementAreas vorgenommen werden.

Für die Parametereinstellungen der Tiles und Mover benötigen Sie die *BTN* | *Beckhoff Traceability Number* der entsprechenden Hardwarekomponenten, die Teil des *BIC* | *Beckhoff Identification Codes* ist. Weitere Informationen finden Sie im Kapitel *Produktübersicht* in den Originalbetriebsanleitungen und auf der Beckhoff Webseite:

### **XPlanar | Originalbetriebsanleitung**

Beschreibung der mechanischen und elektrischen Kenngrößen sowie aller notwendigen Informationen für die Montage des XPlanar Systems.


 [Direktlink zur Dokumentation XPlanar | Originalbetriebsanleitung](#)

### **XPlanar | APS42xx-1x00**

Beschreibung der mechanischen und elektrischen Kenngrößen sowie aller notwendigen Informationen für die Montage des XPlanar Systems.

 [Direktlink zur Dokumentation XPlanar | APS42xx-1x00](#)

### **BIC | Beckhoff Identification Code**

 [Direktlink zu weiteren Informationen zum BIC | Beckhoff Identification Code](#)

## 7.1 Tiles

- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarPart)* ausklappen
- ▶ Doppelklick auf **Object (APS4322)**
- ▶ Im Projektfenster auf den Reiter *Parameter (Init)* klicken

Object	Context	Parameter (Init)	Parameter (Online)	Data Area	Interfaces	Data Pointer		
		Name	Value	CS	Unit	Type	P...	Com...
-		General						
		BTN		<input type="checkbox"/>		STRIN...	0...	8 digit ...
-		TilePositionOnPart	...	<input type="checkbox"/>			0...	Positio...
		.x	0		mm	DINT		
		.y	0		mm	DINT		

- ▶ **BTN** der Kachel eingeben
- ▶ Auf **+** von *TilePositionOnPart* klicken, um die Eingabefelder für die X-Position und die Y-Position der Kachel aufzurufen
- ▶ Werte für **x** und **y** eingeben, die der Position der Kachel im XPlanar Part entsprechen

## 7.2 Mover

- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* ausklappen
- ▶ Doppelklick auf **Object (XPlanarMover)**
- ▶ Im Projektfenster auf den Reiter *Parameter (Init)* klicken

Object	Context	Parameter (Init)	Parameter (Online)	Data Area	Interfaces	Event Classes		
		Name	Value	CS	Unit	Type	PTCID	Comment
-		General						
		MoverType	APM4330	<input type="checkbox"/>		TcloXPlan...	0x03...	The mov...
-		AccessoryIDs	APM4220	<input type="checkbox"/>	1 (Arra...		0x03...	Select if t...
		[0]	APM4221			TcloXPlan...		Select if t...
		BTN	APM4230	<input type="checkbox"/>		STRING(8)	0x03...	8 digit al...
		ParametrizationSetId	APM4330_0001	<input type="checkbox"/>		OTCID	0x03...	If not ass...
		Payload	APM4350					
		SimulationMode	APM4550					
		ForceLimits						
		Controller						
		Methods						
		Feedback						
		PositionOffset						
		ExternalFeedback						
		MoverTransfer						
		MoverCommunication						
		TemperatureCompensation						

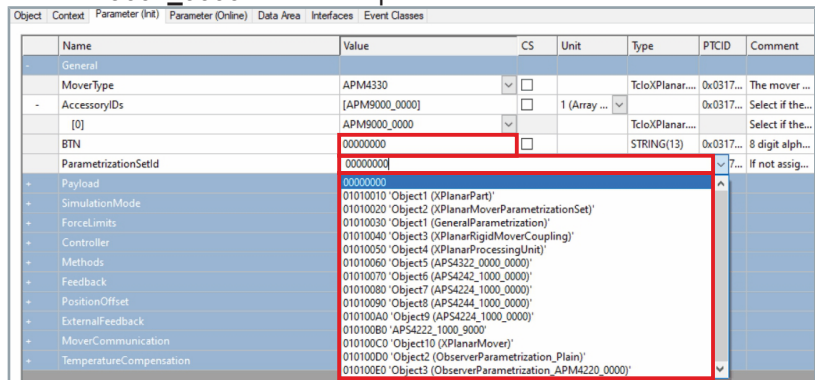
- ▶ **Movertype** im Dropdown-Menü auswählen

Object	Context	Parameter (Init)	Parameter (Online)	Data Area	Interfaces	Event Classes		
		Name	Value	CS	Unit	Type	PTCID	Comment
-		General						
		MoverType	APM4330	<input type="checkbox"/>		TcloXPlanar...	0x0317...	The mover ...
-		AccessoryIDs	[APM9000_0000]	<input type="checkbox"/>	1 (Array ...		0x0317...	Select if the...
		[0]	APM9000_0000	<input type="checkbox"/>		TcloXPlanar...		Select if the...
		BTN	APM9000_0000	<input type="checkbox"/>		STRING(13)	0x0317...	8 digit alph...
		ParametrizationSetId	APM9001_0000	<input type="checkbox"/>		OTCID	0x0317...	If not assig...
		Payload						
		SimulationMode						
		ForceLimits						
		Controller						
		Methods						
		Feedback						
		PositionOffset						
		ExternalFeedback						
		MoverCommunication						
		TemperatureCompensation						

- ▶ **AccessoryIDs** ausklappen

► Bumpertyp im Dropdown-Menü [0] auswählen:

- **APM9000\_0000** für passive Bumper
- **APM9001\_0000** für ID-Bumper



Name	Value	CS	Unit	Type	PTCID	Comment
General						
MoverType	APM4330	<input type="checkbox"/>		TcloXPlanar...	0x0317...	The mover ...
AccessoryIDs	[APM9000_0000]	<input type="checkbox"/>	1 (Array ...)	TcloXPlanar...	0x0317...	Select if the...
[0]	APM9000_0000	<input type="checkbox"/>		TcloXPlanar...	0x0317...	Select if the...
BTN	00000000	<input type="checkbox"/>		STRING(13)	0x0317...	8 digit alph...
ParametrizationSetId	00000000				7...	If not assign...
Payload	00000000					
SimulationMode	01010010 'Object1 (XPlanarPart)'					
ForceLimits	01010020 'Object2 (XPlanarMoverParametrizationSet)'					
Controller	01010030 'Object1 (GeneralParametrization)'					
Methods	01010040 'Object3 (XPlanarRigidMoverCoupling)'					
Feedback	01010050 'Object4 (XPlanarProcessingUnit)'					
PositionOffset	01010060 'Object5 (APS4322_0000_0000)'					
ExternalFeedback	01010070 'Object6 (APS4224_1000_0000)'					
MoverCommunication	01010080 'Object7 (APS4224_1000_0000)'					
TemperatureCompensation	01010090 'Object8 (APS4244_1000_0000)'					
	010100A0 'Object9 (APS4224_1000_0000)'					
	010100B0 'Object5 (APS4322_1000_9000)'					
	010100C0 'Object10 (XPlanarMover)'					
	010100D0 'Object2 (ObserverParametrization_Plain)'					
	010100E0 'Object3 (ObserverParametrization_APM4220_0000)'					

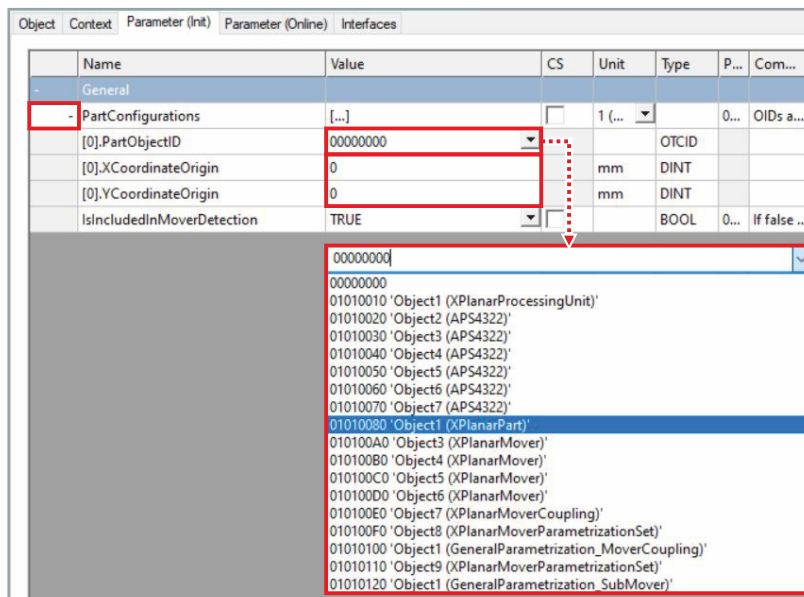
► **BTN** des Movers eingeben

► Bei Bedarf im Dropdown-Menü *ParametrizationSetId* einen Parametrierungssatz auswählen

Wenn die Mover Identification im Prozess nicht erforderlich ist, kann für alle Mover die BTN *00000000* vergeben werden. Die Mover IDs werden in der Reihenfolge zugewiesen, in der sie detektiert werden. Weitere Informationen dazu finden Sie im Kapitel BIC | Beckhoff Identification Code.

## 7.3 MovementArea

- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* ausklappen
- ▶ Doppelklick auf **Object (XPlanarMovementArea)**
- ▶ Im Projektfenster auf den Reiter *Parameter (Init)* klicken



- ▶ Auf + von *PartConfigurations* klicken, um die Eingabefelder aufzurufen
- ▶ **PartObjectID** im Dropdown-Menü auswählen
- ▶ Werte für **XCoordinateOrigin** und **YCoordinateOrigin** eingeben, die dem Positionsursprung des Parts in der MovementArea entsprechen

## 8 TcCOM-Objekte verlinken

Nachdem alle TcCOM-Objekte angelegt wurden, müssen noch einige Verlinkungen erstellt werden, um Berechnungen ausführen zu können.

### 8.1 Mover

Um die Mover steuern zu können, müssen die Sollwerte der Mover berechnet werden. Dazu muss jedes XPlanar Mover Interface zu externen Sollwerten mit einem entsprechenden NC3-Gegenstück verlinkt werden.



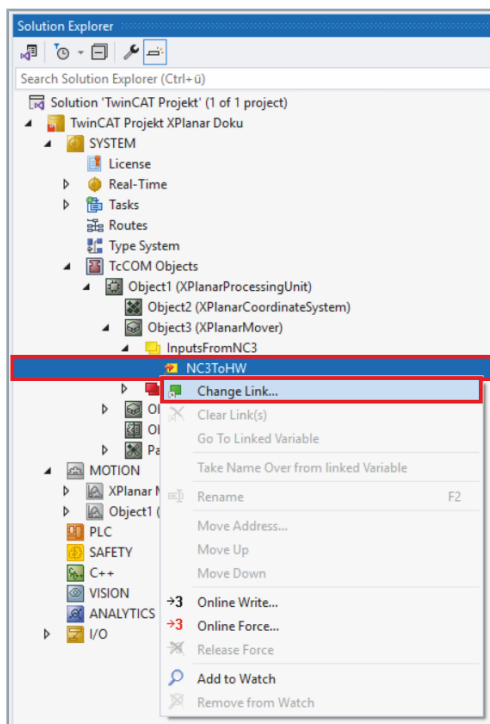
#### Entsprechende Moverobjekte mit TF5430 anlegen

Um die TcCOM-Objekte mit den NC3-Achsen zu verlinken, müssen entsprechende XPlanar Mover Softwareobjekte mit dem TC3 Planar Motion-Softwarepaket TF5430 angelegt werden. Weitere Informationen dazu finden Sie in der Dokumentation TF5430:

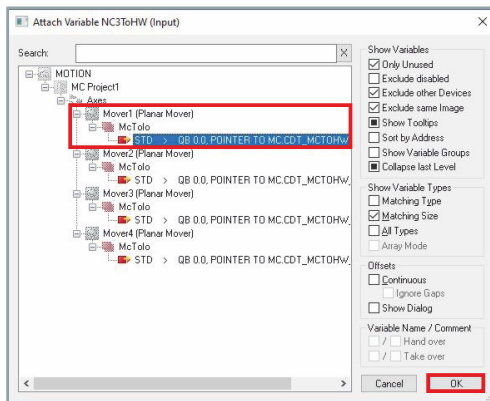
 [Direktlink zur Dokumentation TF5430 | TwinCAT 3 Planar Motion](#)

#### 8.1.1 NC3 To HW

Die Positionssollwerte und Dynamikssollwerte für den Mover werden für die Berechnung bereitgestellt.

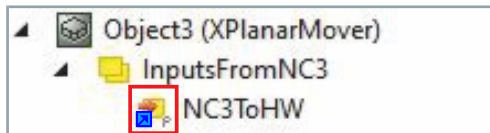


- ▶ *Solution-Explorer > System > TcCOM Objects > Object (XPlanarProcessingUnit) > Object (XPlanarMover) > InputsFromNC3* ausklappen
- ▶ Mit der rechten Maustaste auf **NC3ToHW** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Change Link ...** klicken



Das Dialogfenster *Attach Variable NC3ToHW (Input)* öffnet sich.

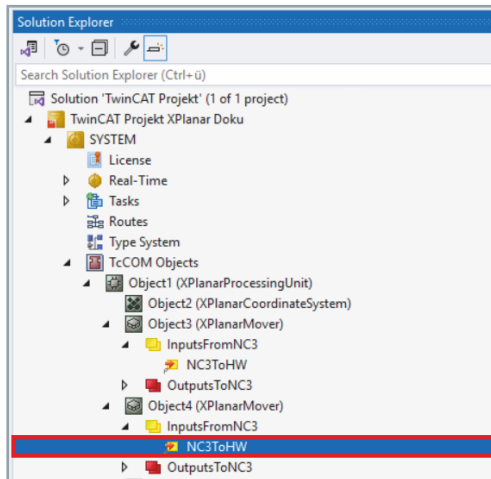
- ▶ Entsprechende *STD* Variable eines Movers auswählen
- ▶ Auswahl mit **OK** bestätigen



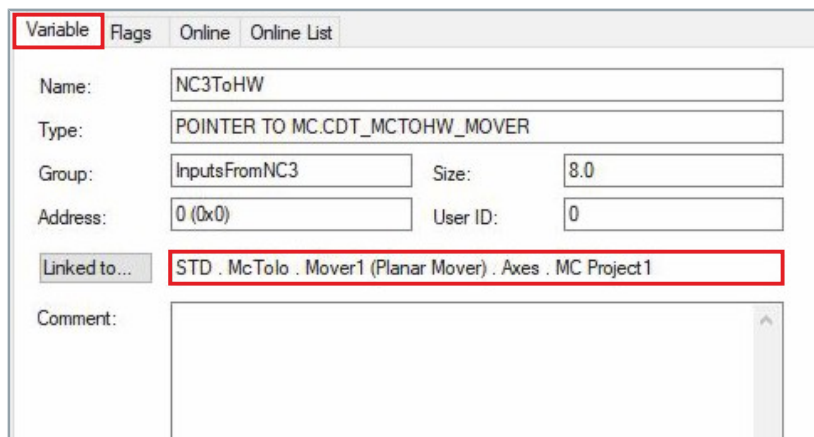
Ein blauer Pfeil an der Variablen zeigt die erfolgreiche Verlinkung an.

- ▶ NC3ToHW-Verlinkung für alle Mover auf dieselbe Weise erstellen

## 8.1.1.1 Verlinkungsdetails NC3ToHW



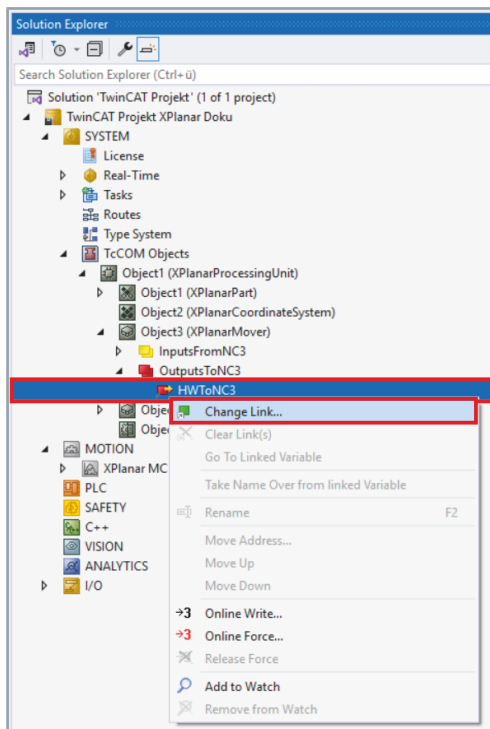
- ▶ *Solution-Explorer > System > TcCOM Objects > Object (XPlanarProcessingUnit) > Object (XPlanarMover) > InputsFromNC3* ausklappen
- ▶ Auf **NC3ToHW** klicken
- ▶ Im Projektfenster auf den Reiter **Variable** klicken



Details der Verlinkung werden unter **Linked to ...** angezeigt.

## 8.1.2 HW To NC3

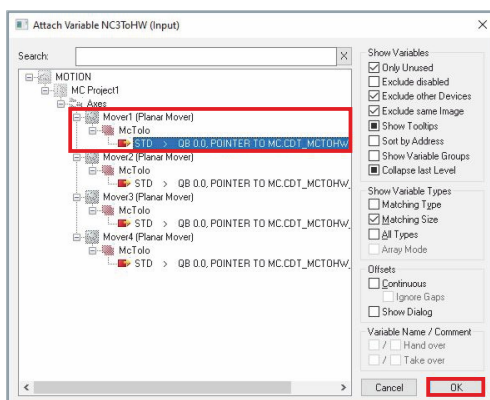
Die Istwerte für Position und Dynamik des Movers werden für die Anwendung bereitgestellt.



- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarMover)* > *OutputsFromNC3* ausklappen
- ▶ Mit der rechten Maustaste auf **HWTtoNC3** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Change Link ...** klicken

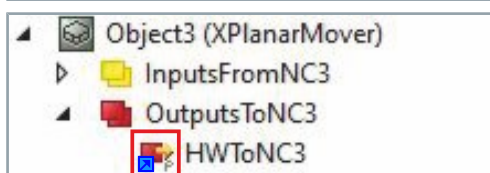
## HINWEIS

Achten Sie bei der HWTtoNC3-Verlinkung darauf, dass Sie denselben Mover auswählen, den Sie bei der NC3ToHW-Verlinkung verwendet haben.



Das Dialogfenster *Attach Variable HWTtoNC3 (Output)* öffnet sich.

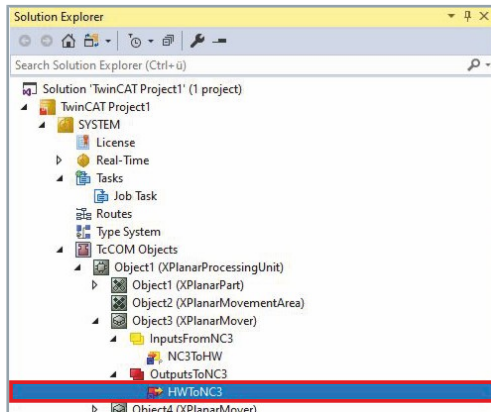
- ▶ Entsprechende *STD* Variable eines Movers auswählen
- ▶ Auswahl mit **OK** bestätigen



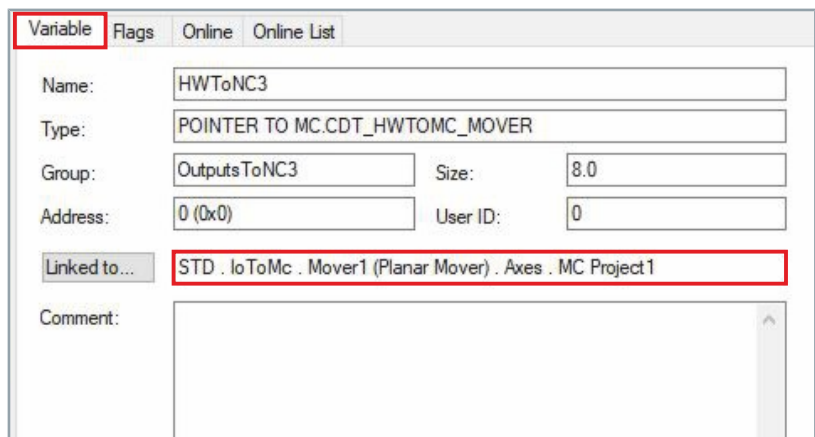
Ein blauer Pfeil an der Variablen zeigt die erfolgreiche Verlinkung an.

- ▶ HWTtoNC3-Verlinkung für alle Mover auf dieselbe Weise erstellen

## 8.1.2.1 Verlinkungsdetails HWToNC3



- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarMover)* > *OutputsFromNC3* ausklappen
- ▶ Auf **HWToNC3** klicken
- ▶ Im Projektfenster auf den Reiter **Variable** anklicken



Details der Verlinkung werden unter **Linked to ...** angezeigt.

## 8.2 Tiles



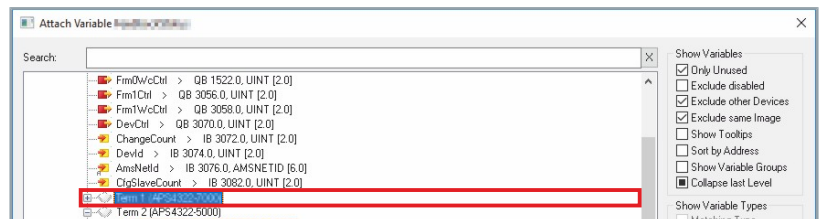
### Simulationsbetrieb oder Normalbetrieb

Für den Simulationsbetrieb des Treibers wird keine Hardware benötigt und die Verlinkung von einem *XPlanar Tile* mit einer physischen XPlanar Kachel *APS4xxx-xx00* ist nicht erforderlich.

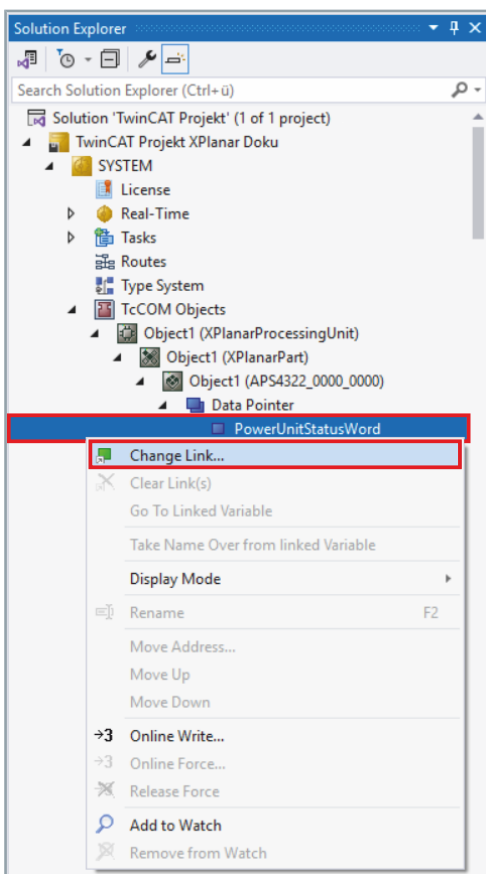
Für den Normalbetrieb muss jedem *XPlanar Tile* eine physische XPlanar Kachel *APS4xxx-xx00* zugeordnet werden.

### 8.2.1 Drive

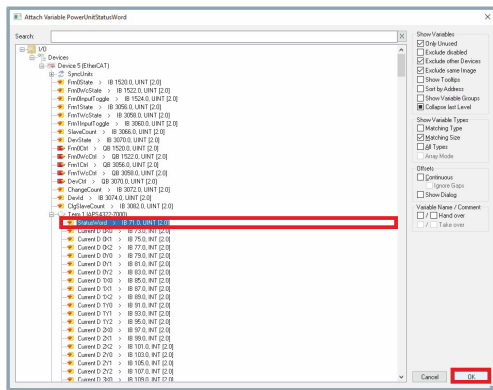
Die Variablen für den Drive-Teil der Kacheln finden Sie im Ordner *Term 1 (APS4322-7000)*:



#### 8.2.1.1 PowerUnitStatusWord

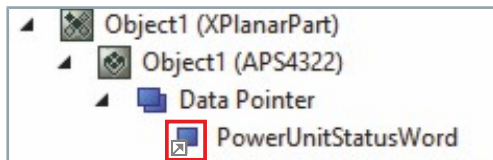


- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarPart)* > *Object (APS4322\_0000\_0000)* > *Data Pointer* ausklappen
- ▶ Mit der rechten Maustaste auf **PowerUnitStatusWord** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Change Link ...** klicken



Das Dialogfenster *Attach Variable PowerUnitStatusWord* öffnet sich.

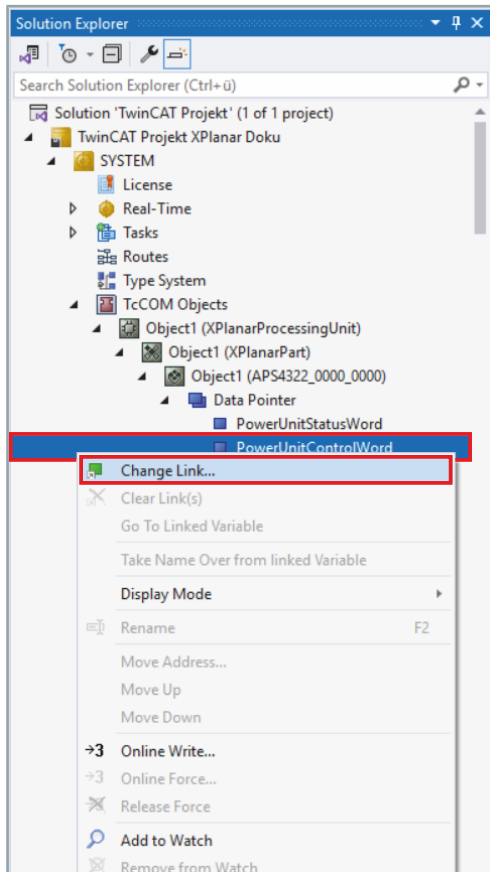
- ▶ **I/O > Devices > Term 1 (APS4322-7000)** ausklappen
- ▶ **StatusWord** auswählen
- ▶ Auswahl mit **OK** bestätigen



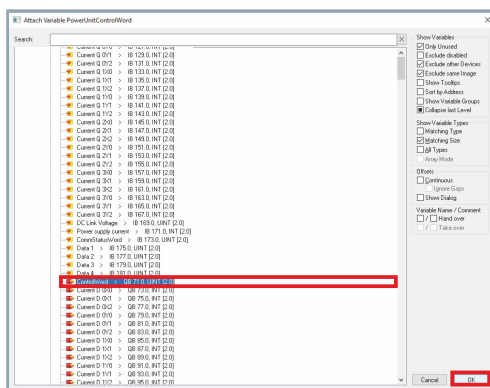
Ein grauer Pfeil an der Variablen zeigt die erfolgreiche Verlinkung an.

Weitere Informationen dazu finden Sie im Kapitel „Verlinkungsdetails“, [Seite 54].

## 8.2.1.2 PowerUnitControlWord

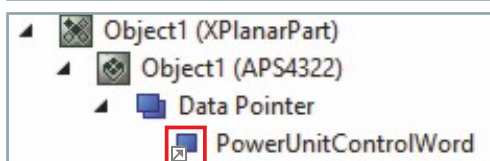


- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (APS4322\_0000\_0000)* > *Data Pointer* ausklappen
- ▶ Mit der rechten Maustaste auf **PowerUnitControlWord** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Change Link ...** klicken



Das Dialogfenster *Attach Variable PowerUnitControlWord* öffnet sich.

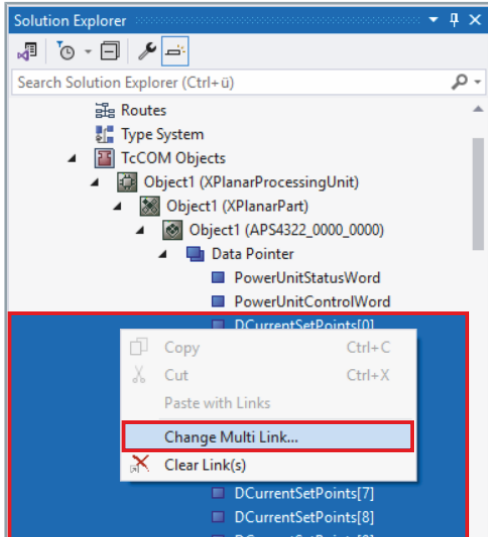
- ▶ *I/O* > *Devices* > *Term 1 (APS4322-7000)* ausklappen
- ▶ **ControlWord** auswählen
- ▶ Auswahl mit **OK** bestätigen



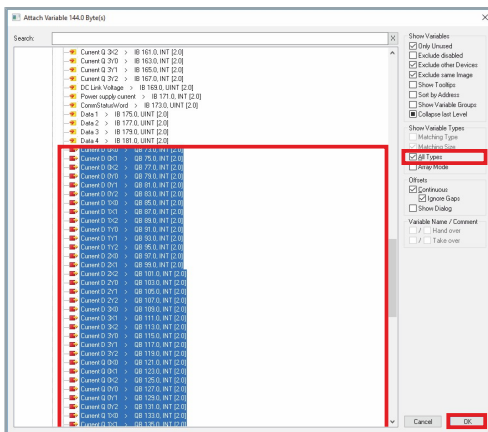
Ein grauer Pfeil an der Variablen zeigt die erfolgreiche Verlinkung an.

Weitere Informationen dazu finden Sie im Kapitel „Verlinkungsdetails“, [Seite 54].

## 8.2.1.3 CurrentSetpoints

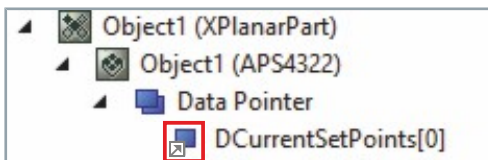


- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (APS4322\_0000\_0000)* > *Data Pointer* ausklappen
- ▶ Alle **DCurrentSetPoints**, **QCurrentSetPoints** und **PhiSetPoints** von oben nach unten auswählen
- ▶ Mit der rechten Maustaste auf die oberste Variable der Markierung klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Change Multi Link ...** klicken



Das Dialogfenster *Attach Variable 144.0 Byte(s)* öffnet sich.

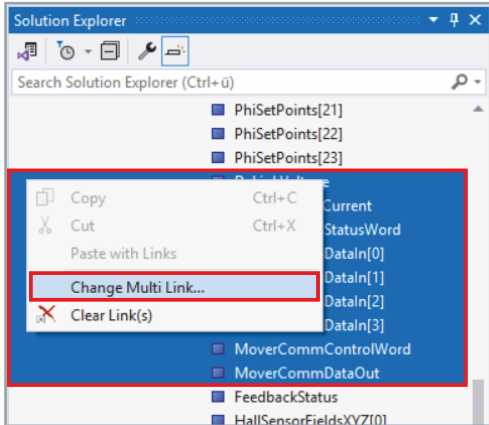
- ▶ *I/O* > *Devices* > *Term 1 (APS4322-7000)* ausklappen
- ▶ Kontrollkästchen bei **All Types** aktivieren
- ▶ Folgende Variablen auswählen:
  - **Current D 0X0** bis **Current D 3Y2**
  - **Current Q 0X0** bis **Current Q 3Y2**
  - **PhE 0X0** bis **PhE 3Y2**
- ▶ Auswahl mit **OK** bestätigen



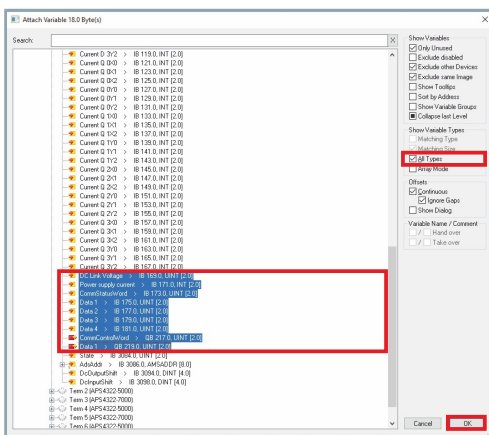
Ein grauer Pfeil an der Variablen zeigt die erfolgreiche Verlinkung an.

Weitere Informationen dazu finden Sie im Kapitel „Verlinkungsdetails“, [Seite 54].

## 8.2.1.4 Diag Data

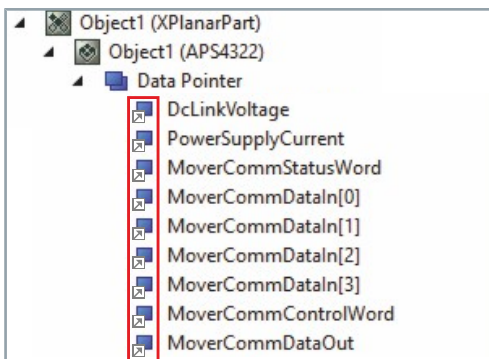


- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlannerProcessingUnit)* > *Object (APS4322\_0000\_0000)* > *Data Pointer* ausklappen
- ▶ Folgende Variablen von oben nach unten auswählen:
  - **DcLinkVoltage**
  - **PowerSupplyCurrent**
  - **MoverCommStatusWord**
  - **MoverCommDataIn[0]** bis **MoverCommDataIn[3]**
  - **MoverCommControlWord**
  - **MoverCommDataOut**
- ▶ Mit der rechten Maustaste auf die oberste Variable der Markierung klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Change Multi Link ...** klicken



Das Dialogfenster *Attach Variable 18.0 Byte(s)* öffnet sich.

- ▶ *I/O* > *Devices* > *Term 1 (APS4322-7000)* ausklappen
- ▶ Kontrollkästchen bei **All Types** aktivieren
- ▶ Folgende Variablen auswählen:
  - **DC Link Voltage**
  - **Power supply current**
  - **CommStatusWord**
  - **Data 1** bis **Data 4**
  - **CommControlWord**
  - **Data 1**
- ▶ Auswahl mit **OK** bestätigen

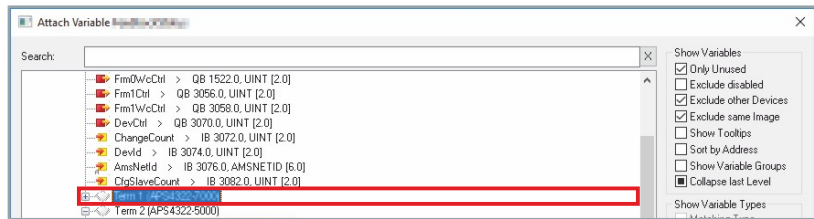


Ein grauer Pfeil an der Variablen zeigt die erfolgreiche Verlinkung an.

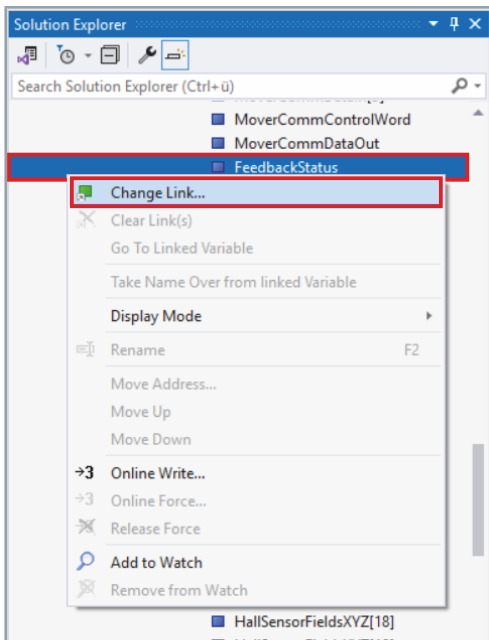
Weitere Informationen dazu finden Sie im Kapitel „Verlinkungsdetails“, [Seite 54].

## 8.2.2 Feedback

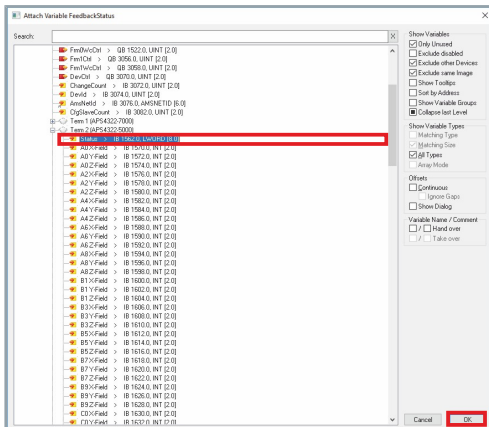
Die Variablen für den Feedback-Teil der Kacheln finden Sie im Ordner *Term 2 (APS4322-5000)*:



### 8.2.2.1 FeedbackStatus

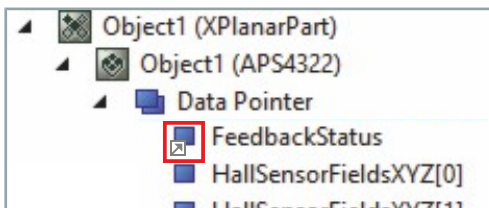


- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlannerProcessingUnit)* > *Object (APS4322\_0000\_0000)* > *Data Pointer* ausklappen
- ▶ Mit der rechten Maustaste auf **FeedbackStatus** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Change Link ...** klicken



Das Dialogfenster *Attach Variable FeedbackStatus* öffnet sich.

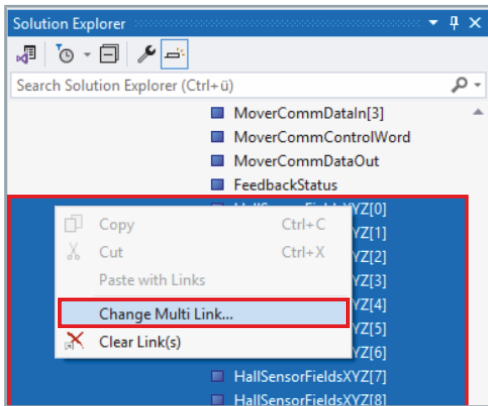
- ▶ *I/O* > *Devices* > *Term 2 (APS4322-5000)* ausklappen
- ▶ **Status** auswählen
- ▶ Auswahl mit **OK** bestätigen



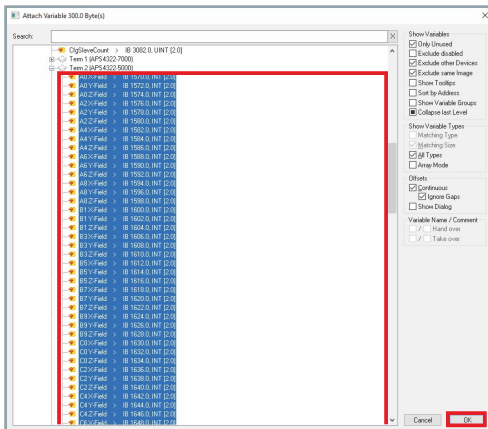
Ein grauer Pfeil an der Variablen zeigt die erfolgreiche Verlinkung an.

Weitere Informationen dazu finden Sie im Kapitel „Verlinkungsdetails“, [Seite 54].

## 8.2.2.2 HallSensors

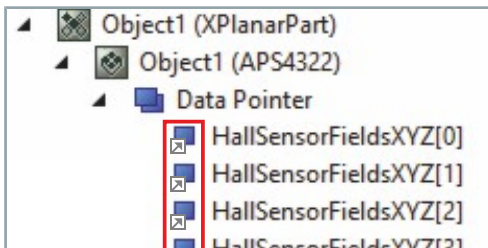


- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (APS4322\_0000\_0000)* > *Data Pointer* ausklappen
- ▶ Alle **HallSensorFieldsXYZ[0]** bis **HallSensorFieldsXYZ[149]** von oben nach unten auswählen
- ▶ Mit der rechten Maustaste auf die oberste Variable der Markierung klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Change Multi Link ...** klicken



Das Dialogfenster *Attach Variable 300.0 Byte(s)* öffnet sich.

- ▶ *I/O* > *Devices* > *Term 2 (APS4322-5000)* ausklappen
- ▶ Kontrollkästchen bei **All Types** aktivieren
- ▶ Alle **A0 X-Field** bis **J9 Z-Field** auswählen
- ▶ Auswahl mit **OK** bestätigen



Ein grauer Pfeil an der Variablen zeigt die erfolgreiche Verlinkung an.

Weitere Informationen dazu finden Sie im Kapitel „Verlinkungsdetails“, [Seite 54].

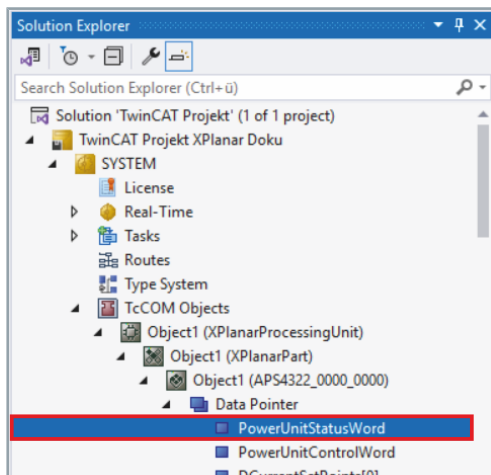
## 8.2.3 Verlinkungsdetails

Nach einer erfolgreichen Verlinkung der *Tiles* TcCom-Objekte mit den physischen Kacheln der I/O wird die Verlinkung durch graue Pfeile an den einzelnen Data Pointer Variablen der TcCOM-Objekte angezeigt. Sie haben nun die Möglichkeit, sich weitere Details zu der Verlinkung anzeigen zu lassen.



### Verlinkungsbeispiel

In diesem Kapitel wird an einer erfolgreichen *PowerUnitStatusWord*-Verlinkung beispielhaft gezeigt, wie Sie sich Details zu der Verlinkung anzeigen lassen können.



- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (APS4322\_0000\_0000)* > *Data Pointer* ausklappen
- ▶ Auf **PowerUnitStatusWord** klicken
- ▶ Im Projektfenster auf den Reiter **Variable** anklicken

Variable	Flags	Online
Name:	PowerUnitStatusWord	
Type:	UINT	
Group:	Data Pointer	Size: 2.0
Address:	0 (0x0)	User ID: 0
Linked to...	StatusWord . APS Inputs . Term 1 (APS4322-7000) . Device 5 (EtherCAT) . I	
Comment:	<div style="border: 1px solid gray; height: 80px;"></div>	
ADS Info:	<div style="border: 1px solid gray; height: 20px;"></div>	
Full Name:	TIRC^TcCOM Objects^Object1 (XPlanarProcessingUnit)^Object1 (XPlanarP	

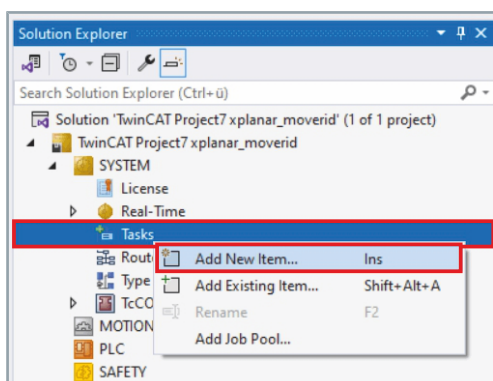
Details der Verlinkung werden unter **Linked to ...** angezeigt.

## 9 Task hinzufügen

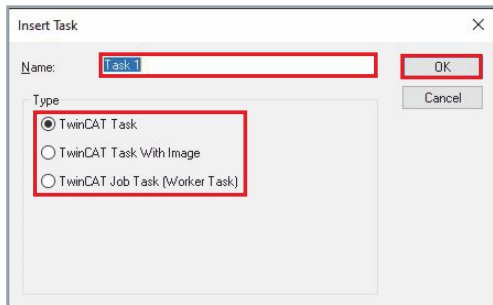
Um aktive Berechnungen ausführen zu können, benötigt jede XPlanar Tile und jeder XPlanar Mover eine Task. Alle XPlanar Tiles, die zu einem EtherCAT Master gehören, müssen auf derselben Task laufen.



Jede Task darf nur auf einem Kern laufen und muss auf 250 µs eingestellt sein.



- ▶ *Solution-Explorer* > *System* ausklappen
- ▶ Mit der rechten Maustaste auf **Tasks** klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Add New Item** klicken



Das Dialogfenster *Insert Task* öffnet sich.

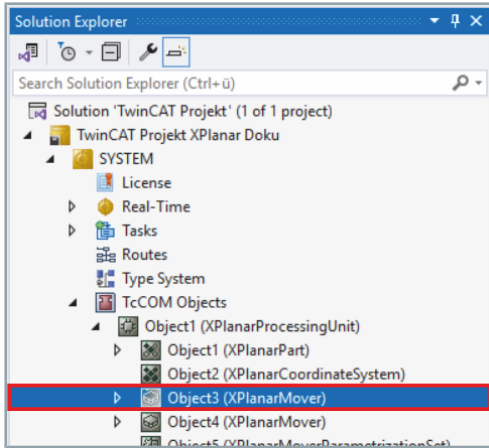
- ▶ **Name** für die Task eingeben
- ▶ Task **Type** auswählen
- ▶ Eingabe und Auswahl mit **OK** bestätigen

Eine entsprechend benannte Task wird zu den Tasks hinzugefügt.

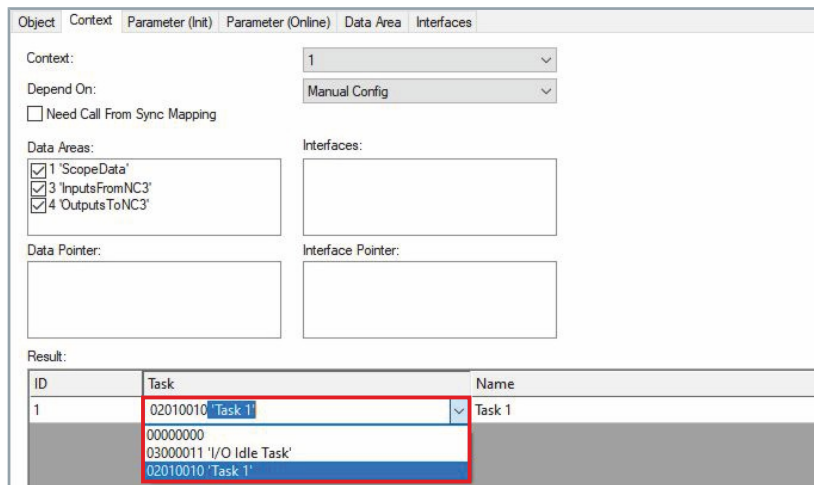
## 10 Task verlinken

Nachdem eine oder mehrere Tasks hinzugefügt wurden, müssen alle XPlanar Tiles und XPlanar Mover mit einer Task verlinkt werden.

### 10.1 Mover

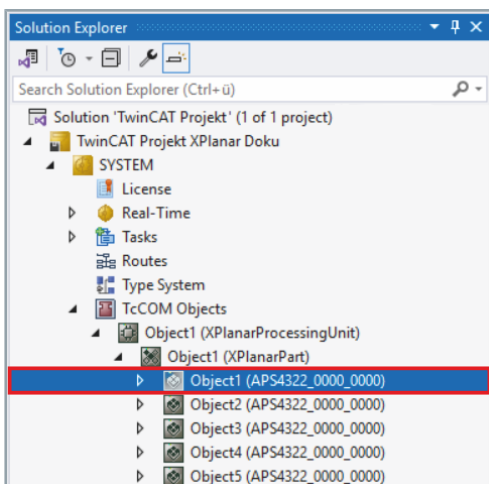


- ▶ *Solution-Explorer* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* ausklappen
- ▶ Doppelklick auf **Object (XPlanar Mover)**
- ▶ Im Projektfenster auf den Reiter **Context** klicken



- ▶ Task im Dropdown-Menü **Task** auswählen

### 10.2 Tile



- ▶ *Solution-Explorer* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarPart)* ausklappen
- ▶ Doppelklick auf **Object (APS4322\_0000\_0000)**
- ▶ Im Projektfenster auf den Reiter **Context** klicken

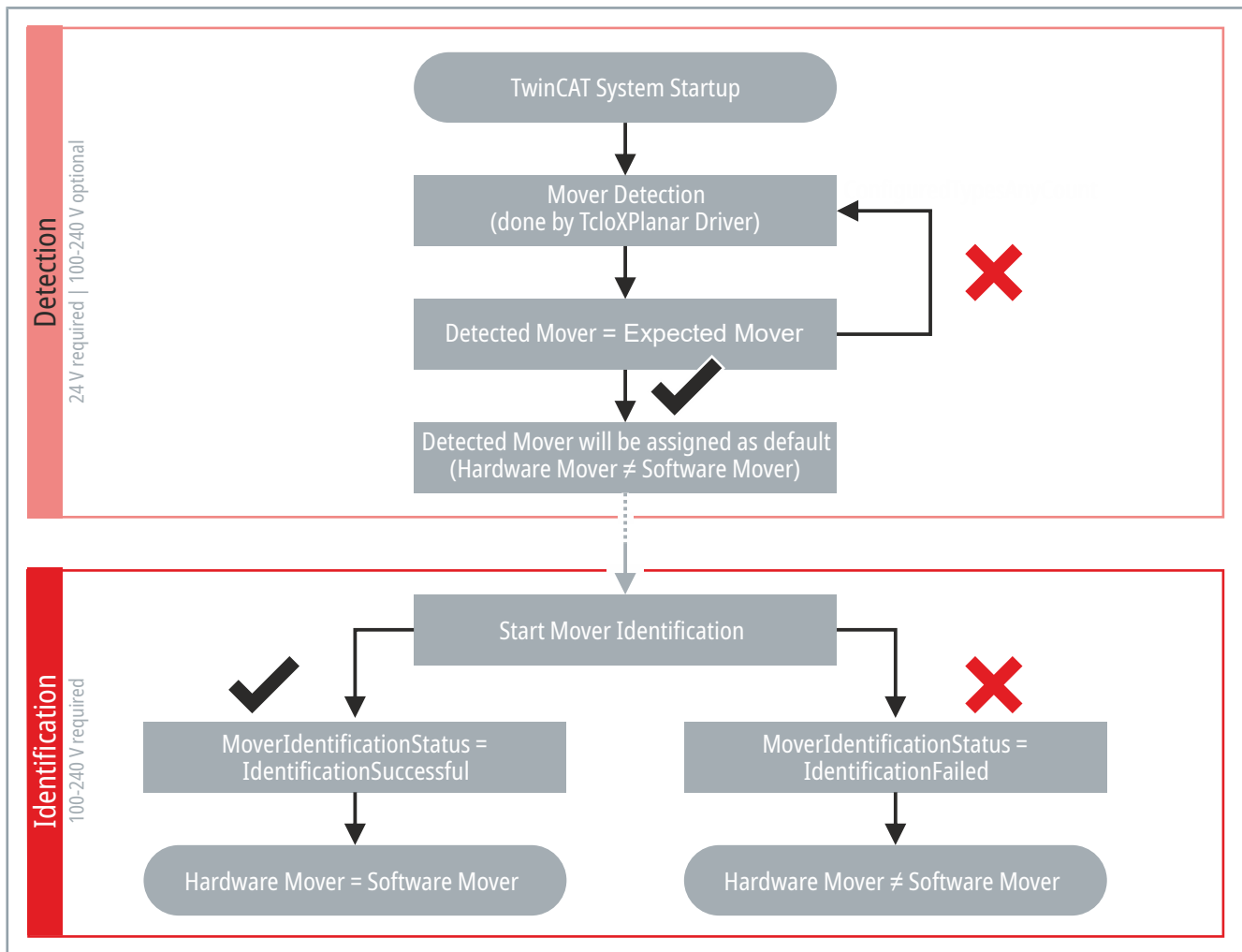
Object	Context	Parameter (Init)	Parameter (Online)	Data Area	Interfaces	Data Pointer
Context:	1					
Depend On:	Manual Config					
<input type="checkbox"/> Need Call From Sync Mapping						
Data Areas:	<input checked="" type="checkbox"/> 1 'ScopeData'		Interfaces:			
Data Pointer:			Interface Pointer:			
Result:						
ID	Task	Name				
1	02010010 'Task 1'	Task 1				
	00000000					
	03000011 '/O Idle Task'					
	02010010 'Task 1'					

► Task im Dropdown-Menü **Task** auswählen

## 11 Mover Detection und Mover Identifikation

Mover, die über einen ID-Bumper verfügen, übermitteln ihre ID bei einer Mover Identification automatisch, sodass eine genaue Zuordnung des Movers auf dem XPlanar System erfolgt. Die Mover Detection ist abhängig von der Auswahl des Parameter *DetectionMode*. Für die Mover Identification sind Movern mit ID-Bumpers erforderlich.

### 11.1 Detection Mode: AsConfigured



Nach dem TwinCAT System Startup wird eine Mover Detection durchgeführt und überprüft, ob die Anzahl der erkannten Mover mit der Anzahl der erwarteten Mover übereinstimmt. Bei der Detection erfolgt ausschließlich eine absolute Positionserkennung der Mover. Die Mover werden anhand aufsteigender Y-Positionen den Softwareachsen zugeordnet. Die Mover Detection wird beendet, wenn die Anzahl der erkannten Mover mit der erwarteten Mover übereinstimmt.

Wenn die Anzahl übereinstimmt und die Lastspannung am XPlanar System anliegt, kann die Mover Identification gestartet werden. Sobald die Mover Identification gestartet wurde, kommuniziert das System automatisch mit den Movern mit ID Bumpers, um eine eindeutige Zuordnung zwischen den XPlanar Movern und den Softwareachsen herzustellen.

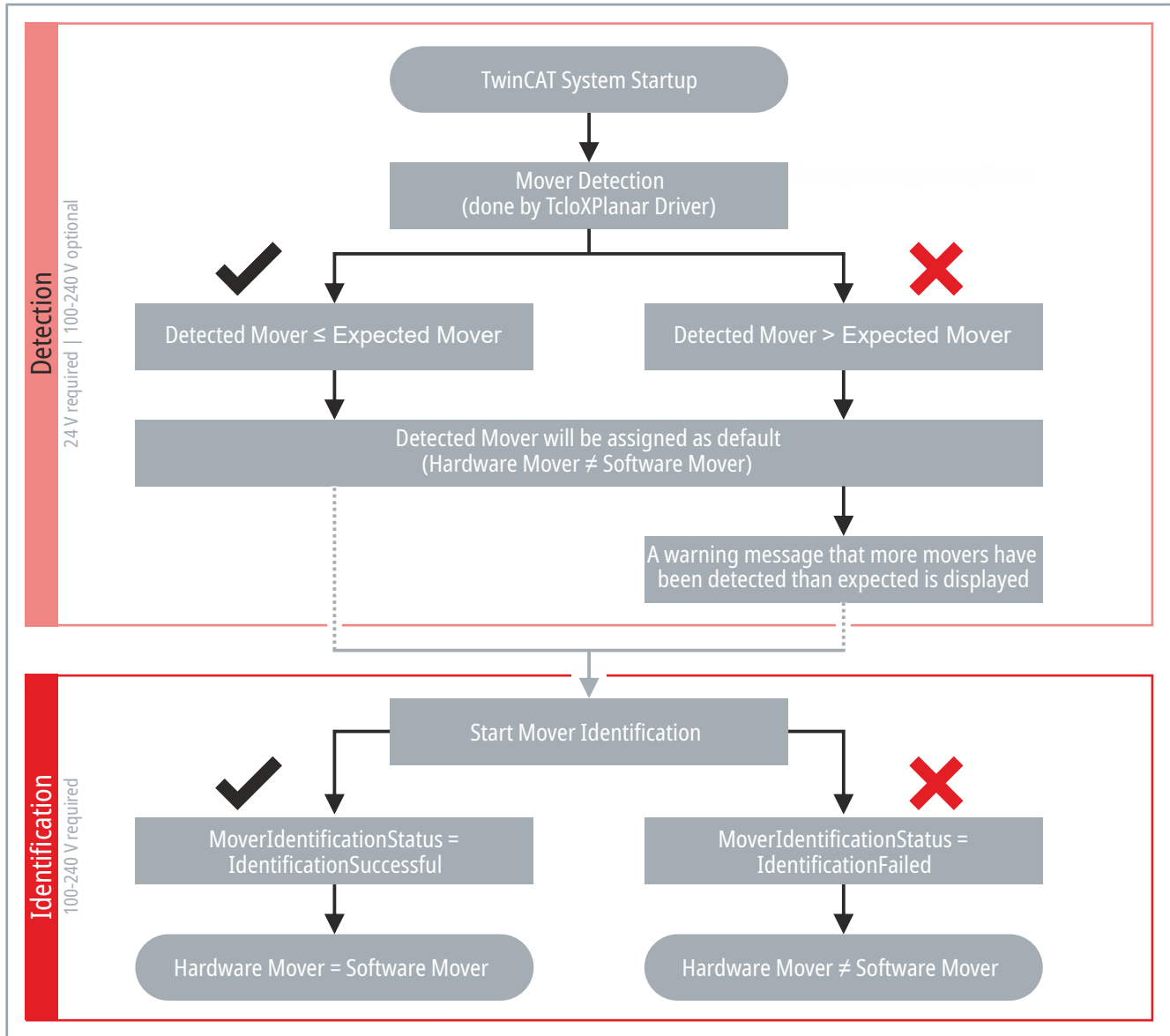
Der aktuelle Status des Kommunikationsprozesses wird über den *CoordinatedMoverCommStatus* dargestellt. Nach erfolgreicher Identification sind die Hardware Mover den Software Movern zugeordnet.

## 11.2 Detection Mode: ConfiguredTypesAnyCount



### Mehr konfigurierte Mover als vorhandene Mover möglich

Der DetectionMode *ConfiguredTypesAnyCount* ermöglicht es, in der Konfiguration Ihres Systems mehr Mover anzulegen als auf dem System vorhanden sind. Nicht vorhandene Mover können zu einem späteren Zeitpunkt ergänzt werden.



Nach dem TwinCAT System Startup wird eine Mover Detection durchgeführt und überprüft, ob die Anzahl der erkannten Mover mit der Anzahl der erwarteten Mover übereinstimmt. Bei der Detection erfolgt ausschließlich eine absolute Positionserkennung der Mover. Die Mover werden anhand aufsteigender Y-Positionen den Softwareachsen zugeordnet. Die Mover Detection läuft einmal durch.

Wenn die Anzahl erkannter Mover größer als die Anzahl erwarteter Mover ist, wird eine Warnmeldung ausgegeben. Zu viele Mover müssen vor der Mover Identification entfernt werden.

## **WARNUNG**

### **Nach Erhalt einer Warnmeldung Moveranzahl korrigieren**

Wenn auf dem System mehr Mover erkannt als erwartet worden sind, müssen Sie die Anzahl der Mover auf dem System entsprechend Ihrer Konfiguration oder die Konfiguration entsprechend der Anzahl der Mover auf dem System anpassen.

*Wenn die Anzahl erkannter und erwarteter Mover nicht übereinstimmt, können Schäden an den Movern und Verletzungen durch Splitter in den Augen die Folge sein.*

- Entfernen Sie die überschüssigen Mover.
- Passen Sie die Konfiguration entsprechend der erkannten, überschüssigen Mover an.

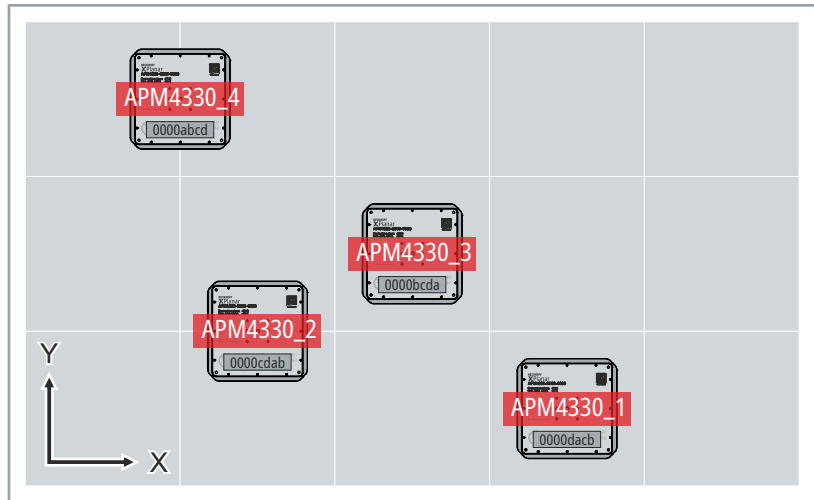
Nach erfolgter Detection, Entfernung überschüssiger Mover und wenn die Lastspannung am XPlanar System anliegt, kann die Mover Identification gestartet werden. Sobald die Mover Identification gestartet wurde, kommuniziert das System automatisch mit den Movern mit ID-Bumpen, um eine eindeutige Zuordnung zwischen den XPlanar Movern und den Softwareachsen herzustellen.

Der aktuelle Status des Kommunikationsprozesses wird über den *CoordinatedMoverCommStatus* dargestellt. Nach erfolgreicher Identification sind die Hardware Mover den Software Movern zugeordnet.

## 11.3 Mover Zuordnung

Nach dem Systemstart und bevor die Mover Identification durchgeführt wurde, stimmen die Software Mover ID und die Hardware Mover ID nicht überein:

### Mover Zuordnung nach dem Systemstart

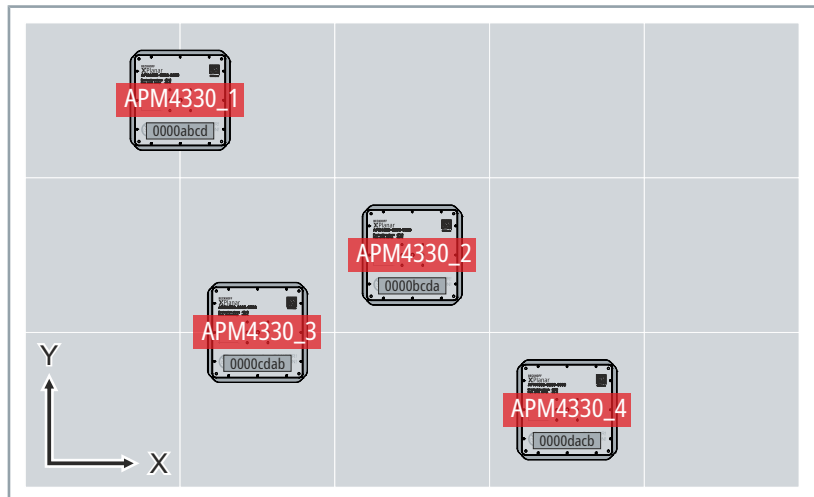


Software Mover	Hardware Mover (BTN)	Position (X/Y)
APM4330_1 (0000abcd) <sup>1)</sup>	≠ 0000dacb	840/120
APM4330_2 (0000bcda) <sup>1)</sup>	≠ 0000cdba	360/240
APM4330_3 (0000cdab) <sup>1)</sup>	≠ 0000bcda	600/360
APM4330_4 (0000dabc) <sup>1)</sup>	≠ 0000abcd	240/600

1) 0000xxxx ist die konfigurierte BTN im Software Objekt

### Mover Zuordnung nach der Mover Identification

Nach einer durchgeführten Mover Identification ist die Zuordnung der Software Mover ID mit der Hardware Mover ID korrekt:



Software Mover	Hardware Mover (BTN)	Position (X/Y)
APM4330_1 (0000abcd) <sup>1)</sup>	= 0000abcd	240/600
APM4330_2 (0000bcda) <sup>1)</sup>	= 0000bcda	600/360
APM4330_3 (0000cdab) <sup>1)</sup>	= 0000cdab	360/240
APM4330_4 (0000dabc) <sup>1)</sup>	= 0000dabc	840/120

1) 0000xxxx ist die konfigurierte BTN im Software Objekt

## 11.4 Erforderliche Einstellungen

Bevor die Mover Detection gestartet werden kann, müssen einige Einstellungen in TwinCAT vorgenommen werden.



### Beispielhafte Darstellung

Im folgenden Beispiel werden die Einstellungen für einen Mover *APM4330-0000* und einem ID-Bumper *AMP9001-0000-4330* gezeigt. Die Abbildungen zeigen entsprechende Einstellungen für diese Beispielkonfiguration.

Name	Value	CS	Unit	Type
MoverType	APM4330	<input type="checkbox"/>		TcIoXPlan...
AccessoryIDs	[APM9000_0000]	<input type="checkbox"/>	1 (Attr...	TcIoXPlan...
[0]	APM9000_0000	<input type="checkbox"/>		STRING(8)
BTN	APM9001_0000	<input type="checkbox"/>		OTCID
ParametrizationSetId		<input type="checkbox"/>		
Payload		<input type="checkbox"/>		
SimulationMode		<input type="checkbox"/>		
ForceLimits		<input type="checkbox"/>		
Controller		<input type="checkbox"/>		
Methods		<input type="checkbox"/>		
Feedback		<input type="checkbox"/>		
PositionOffset		<input type="checkbox"/>		
ExternalFeedback		<input type="checkbox"/>		
MoverTransfer		<input type="checkbox"/>		
MoverCommunication		<input type="checkbox"/>		

► *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* ausklappen

► Auf **Object (XPlanarMover)** klicken

► Im Projektfenster auf den Reiter **Parameter (Init)** klicken

► *General* ausklappen

► *AccessoryIDs* ausklappen

► ID-Bumper **APM9001-0000** im Dropdown-Menü *[0]* auswählen



### BTN des Movers

Jeder Mover benötigt eine eigene *BTN | Beckhoff Traceability Number*. Die BTN des Movers finden Sie auf jedem Mover unter dem *BIC | Beckhoff Identification Code*. Das mehrfache Zuordnen einer Mover BTN führt zu Fehlern bei der Mover Identifictaion.

Weitere Informationen zur BTN finden Sie in der XPlanar Original-Betriebsanleitung oder unter:

🌐 [www.beckhoff.com/bic](http://www.beckhoff.com/bic)

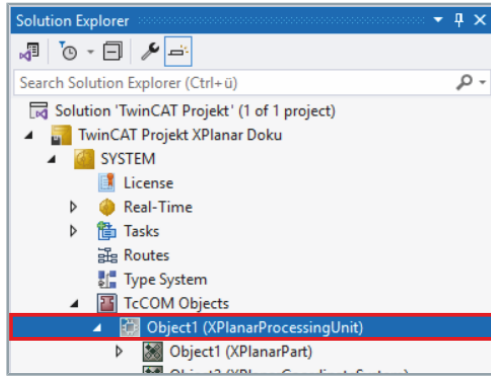
Name	Value	CS	Unit	Type
MoverType	APM4330	<input type="checkbox"/>		TcIoXPlan...
AccessoryIDs	[APM9001_0000]	<input type="checkbox"/>	1 (Attr...	TcIoXPlan...
[0]	APM9001_0000	<input type="checkbox"/>		STRING(8)
BTN	00000000	<input type="checkbox"/>		OTCID
ParametrizationSetId	00000000	<input type="checkbox"/>		
Payload		<input type="checkbox"/>		
SimulationMode		<input type="checkbox"/>		
ForceLimits		<input type="checkbox"/>		
Controller		<input type="checkbox"/>		
Methods		<input type="checkbox"/>		
Feedback		<input type="checkbox"/>		
PositionOffset		<input type="checkbox"/>		
ExternalFeedback		<input type="checkbox"/>		
MoverTransfer		<input type="checkbox"/>		
MoverCommunication		<input type="checkbox"/>		

► In das Eingabefeld *BTN* die BTN eines Hardware Movers eintragen

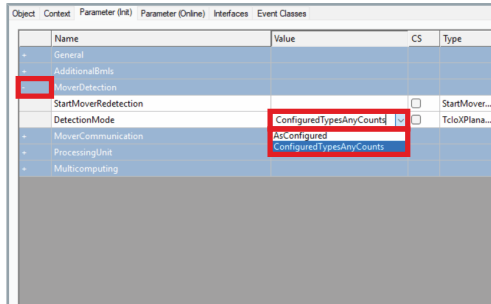
► ID-Bumper und BTN für alle *Object (XPlanerMover)* auf dieselbe Weise konfigurieren

Zur Übernahme der Einstellungen müssen die Konfiguration neu aktiviert und das TwinCAT System neu gestartet werden.

## 11.5 Starten der Mover Detection



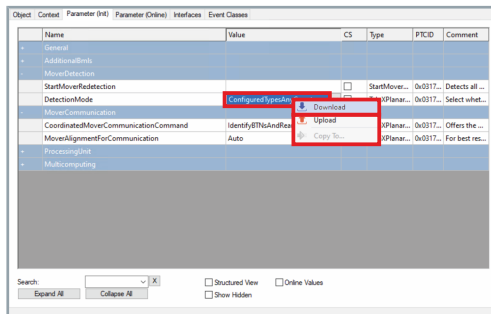
- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* ausklappen
- ▶ Auf **Object (XPlanarProcessingUnit)** klicken



- ▶ Im Projektfenster auf den Reiter **Parameter (Init)** klicken
- ▶ *MoverDetection* ausklappen
- ▶ **AsConfigured** im Dropdown-Menü *DetectionMode* auswählen, wenn die Mover Detection so lange ausgeführt werden soll, bis die Anzahl der erkannten und erwarteten Mover übereinstimmt

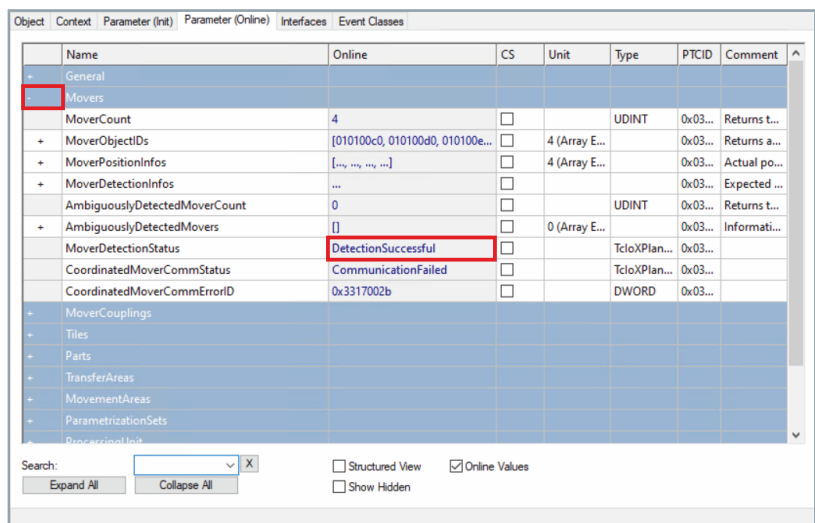
ODER

- ▶ **ConfiguredTypesAnyCounts** im Dropdown-Menü *DetectionMode* auswählen, wenn die Mover Detection einmal durchlaufen soll



- ▶ Mit der rechten Maustaste in das Eingabefeld *DetectionMode* klicken, um das Kontextmenü aufzurufen
- ▶ Im Kontextmenü auf **Download** klicken

### 11.5.1 Kontrolle der Mover Detection



- ▶ Im Projektfenster auf den Reiter **Parameter (Online)** klicken
- ▶ *Movers* ausklappen
- ▶ Status von *MoverDetectionStatus* überprüfen

## 11.6 Starten der Mover Identification


Wenn die ID-Bumper und BTN für alle Mover konfiguriert wurden, kann die Mover Identification gestartet werden. Sie haben zwei Möglichkeiten die Mover Identification zu starten:


- über das *Object XPlanarProcessingUnit* in TwinCAT
- über die SPS.



### ID-Bumper erforderlich

Für die Identification der Mover sind ID-Bumper erforderlich. Passive Bumper *APM9000-0000* können gegen ID-Bumper *APM9001-0000* ausgetauscht werden. Weitere Informationen dazu finden Sie in der XPlanar Original-Betriebsanleitung:

 [Direktlink zur XPlanar APS4322-0000 Original-Betriebsanleitung](#)

 [Direktlink zur XPlanar APS42xx-1x00 Original-Betriebsanleitung](#)

Für folgende Mover sind ID-Bumper *APM9001-0000* verfügbar:

- APM4220-0000
- APM4221-0000
- APM4330-0000
- APM4550-0000

Der Mover *APM4330-0001* verfügt über eine integrierte ID-Funktion.

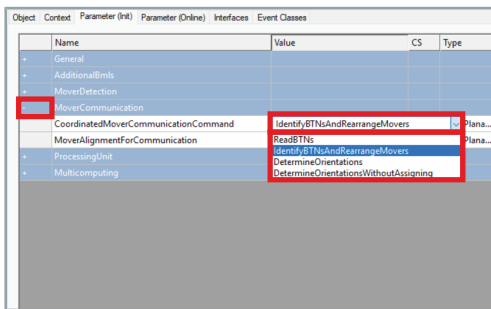
### 11.6.1 Unterstützte Kombinationen

Bisher werden nicht alle Kombinationen von Kacheln und Movern bei der Mover Identification unterstützt.

Mover	Kachel	
	APS4322-0000	APS42xx-0000
APM4220-0000	✓	✓
APM4221-0000	✓	–
APM4330-0000	✓	✓
APM4330-0001	✓	–
APM4550-0000	✓	✓
Moverkopplung	–	–

## 11.6.2 Object XPlanarProcessingUnit

Um die Mover Identification über das *Object (XPlanarProcessingUnit)* in TwinCAT zu starten, muss sich das XPlanar System *Run Mode* befinden und die Lastspannung muss am System anliegen.



- ▶ **MoverCommunication** ausklappen
- ▶ **ReadBTNs** im Dropdown-Menü *CoordinatedMoverCommunicationCommand* auswählen, wenn die BTNs der Mover Objekte zum Einlesen in die SPS unter *XPlanarMover/Parameter(Online)/MoverCommunicationData/IdentifiedMoverBTN* gespeichert werden sollen
- ▶ **IdentifyBTNsAndRearrangeMovers** im Dropdown-Menü *CoordinatedMoverCommunicationCommand* auswählen, wenn Mover Objekte den gefundenen BTNs zugeordnet werden sollen
- ▶ **DetectOrientationsWithoutAssigning** im Dropdown-Menü *CoordinatedMoverCommunicationCommand* auswählen, wenn die Ausrichtung des Movers erkannt und nicht übernommen werden soll

ODER

- ▶ **DetectOrientations** im Dropdown-Menü *CoordinatedMoverCommunicationCommand* auswählen, wenn die Ausrichtung des Movers erkannt und zugeordnet werden soll

### ⚠ WARNUNG

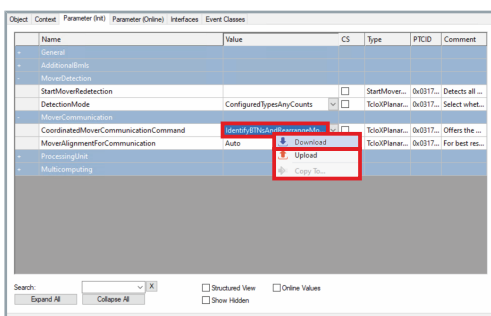
#### Nach Erhalt einer Warnmeldung Moveranzahl korrigieren

Wenn auf dem System mehr Mover erkannt als erwartet worden sind, müssen Sie die Anzahl der Mover auf dem System entsprechend Ihrer Konfiguration oder die Konfiguration entsprechend der Anzahl der Mover auf dem System anpassen.

*Wenn die Anzahl erkannter und erwarteter Mover nicht übereinstimmt, können Schäden an den Movern und Verletzungen durch Splitter in den Augen die Folge sein.*

- Entfernen Sie die überschüssigen Mover.
- Passen Sie die Konfiguration entsprechend der erkannten, überschüssigen Mover an.

Starten Sie die Mover Identification erst, wenn die Anzahl der erkannten Mover kleiner oder gleich der Anzahl der erwarteten Mover ist.



- ▶ Mit der rechten Maustaste in das Eingabefeld *CoordinatedMoverCommunicationCommand* klicken, um das Kontextmenü zu öffnen
- ▶ Im Kontextmenü auf **Download** klicken

## 11.6.2.1 Kontrolle der Mover Identification

Object	Context	Parameter (Init)	Parameter (Online)	Interfaces	Event Classes			
			Online	CS	Unit	Type	PTCID	Comment
+ General								
- Movers								
MoverCount			4	<input type="checkbox"/>		UDINT	0x03...	Returns t...
+ MoverObjectIDs			[010100c0, 010100d0, 010100e...	<input type="checkbox"/>	4 (Array E...		0x03...	Returns a...
+ MoverPositionInfos			[...] ...]	<input type="checkbox"/>	4 (Array E...		0x03...	Actual po...
+ MoverDetectionInfos			...	<input type="checkbox"/>			0x03...	Expected ...
AmbiguouslyDetectedMoverCount			0	<input type="checkbox"/>		UDINT	0x03...	Returns t...
+ AmbiguouslyDetectedMovers			[]	<input type="checkbox"/>	0 (Array E...		0x03...	Informati...
MoverDetectionStatus			DetectionSuccessful	<input type="checkbox"/>		TcIoXPlan...	0x03...	
CoordinatedMoverCommStatus			CommunicationSuccessful	<input type="checkbox"/>		TcIoXPlan...	0x03...	
CoordinatedMoverCommErrorID			0x3317002b	<input type="checkbox"/>		DWORD	0x03...	
+ MoverCouplings								
+ Tiles								
+ Parts								
+ TransferAreas								
+ MovementAreas								
+ ParametrizationSets								
Processional Inst								

Search:  X  Structured View  Online Values  
   Show Hidden

- ▶ Im Projektfenster auf den Reiter **Parameter (Online)** klicken
- ▶ **Movers** ausklappen
- ▶ Status von *CoordinateMoverCommStatus* überprüfen



### Fehler bei der Mover Identification

Wenn die Mover Identification nicht erfolgreich war, kontrollieren Sie die Eingaben der **AccessoryIDs** und **BTN** aller Mover und starten Sie die Mover Identification erneut. Dieser Fehler tritt häufig auf, wenn die Stromversorgung fehlt und das System im Not-Aus-Zustand befindet.

## 11.6.3 SPS

In diesem Kapitel wird beschrieben, wie Sie die Mover Identification über Ihre SPS starten.



### **TF5890 | TwinCAT 3 XPlanar erforderlich**

Voraussetzung für den Start der Mover Identification ist die XPlanar Utility Bibliothek, die mit dem Setup des *TF5890 | TwinCAT 3 XPlanar* erfolgt.

### 11.6.3.1 Variablen Deklaration

```
VAR
  fbEnviroment           :FB_TcIoXPlanarEnviroment;
  ipXpu                  :I_TcIoXPlanarProcessingUnit;
  bStartMoverIdentification :BOOL;
  nStateXPU              :INT;
END_VAR
```

### 11.6.3.2 Beispiel Programmablauf

```
CASE nStateXPU OF
0: // init environment
  IF fbEnvironment.Init(TRUE) THEN
    ipXpu:=fbEnvironment.XpuTcIo(1);
    nStateXPU:=1;
  END_IF

1: //check if MoverDetection is done and the Identification process should be started
  IF bStartMoverIdentification
    AND (ipXpu.GetMoverDetectionInfos().TotalDetected = ipXpu.GetMoverDetectionInfos().TotalExpected)
  THEN
    ipXpu.SetStartMoverIdentification():
    nStateXpu:=2;
  END_IF

2://check and wait if the Identification process has succeded successfully
  IF ipXpu.GetMoverDetectionInfos = MoverIdentificationStatus.IdentificationSuccessful THEN
    //MoverIdentification Successful
    nStateXPU:=3
  END IF
END_CASE
```

## 12 Moverkopplung

Mit einem Rahmen können zwei oder mehrere Mover zu einer Moverkopplung verbunden werden. Für die Moverkopplung und die Submover müssen einige Parameter festgelegt werden, damit die Moverkopplung in *TwinCAT 3* gesteuert werden kann.

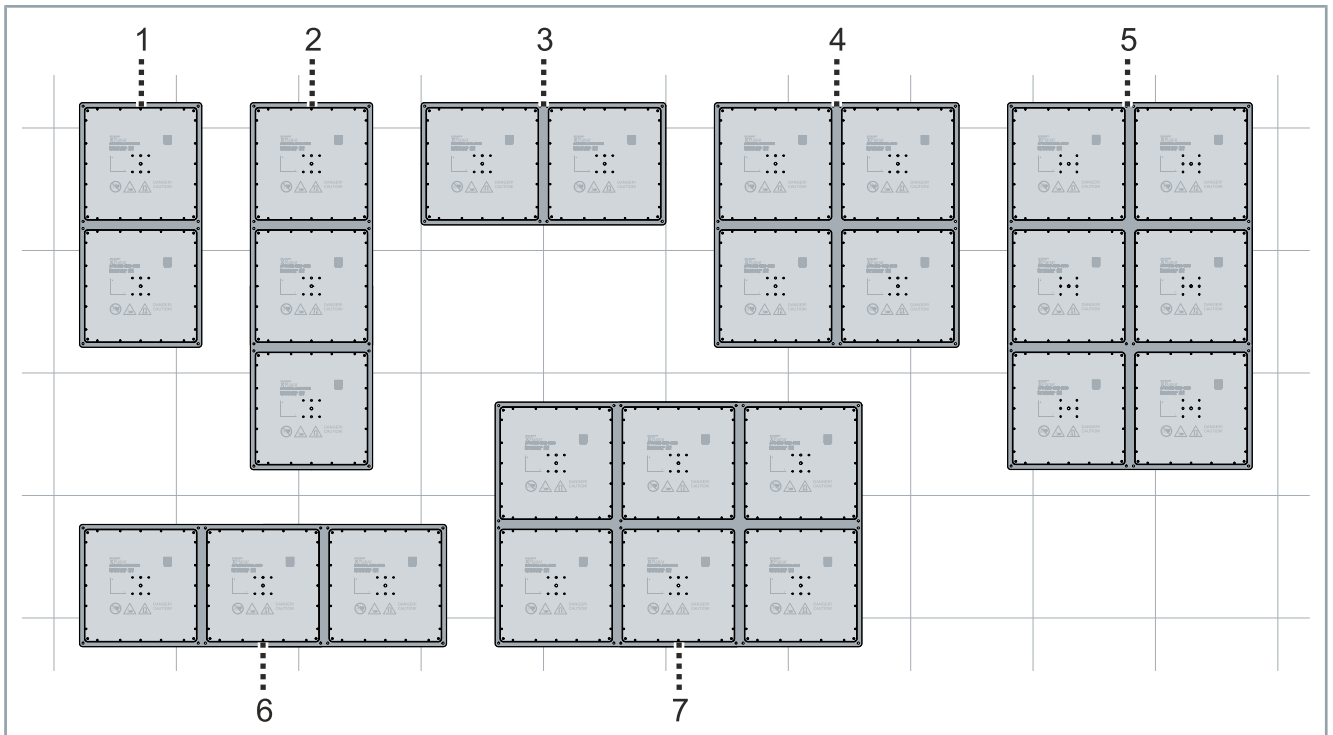


### Mover Identification nur für einzelne Mover

Die Mover Identification ist nur für einzelne Mover mit ID-Bumper möglich. Moverkopplungen können nicht identifiziert werden, da die Mover ohne ID-Bumper in der Moverkopplung montiert werden. Moverkopplungen können nur über die Mover Detection erkannt werden.

### 12.1 Empfohlene Anordnungen bei Kacheln APS4322-0000

Mit einem Rahmen lassen beliebig viele Mover koppeln, solange sie vom IPC gesteuert werden können. Bei Verwendung von Kacheln *APS4322-0000* muss ein Achsabstand von 240 mm von Movermitte zu Movermitte eingehalten werden, um eine nahezu lineare Erhöhung der Nutzlast zu gewährleisten. Die folgende Abbildung zeigt Beispiele für Moverkopplungen mit zwei, vier oder sechs Movern.



Position	Erläuterung
1	1 x 2 Submover
2	1 x 3 Submover
3	2 x 1 Submover
4	2 x 2 Submover
5	2 x 3 Submover
6	3 x 1 Submover
7	3 x 2 Submover

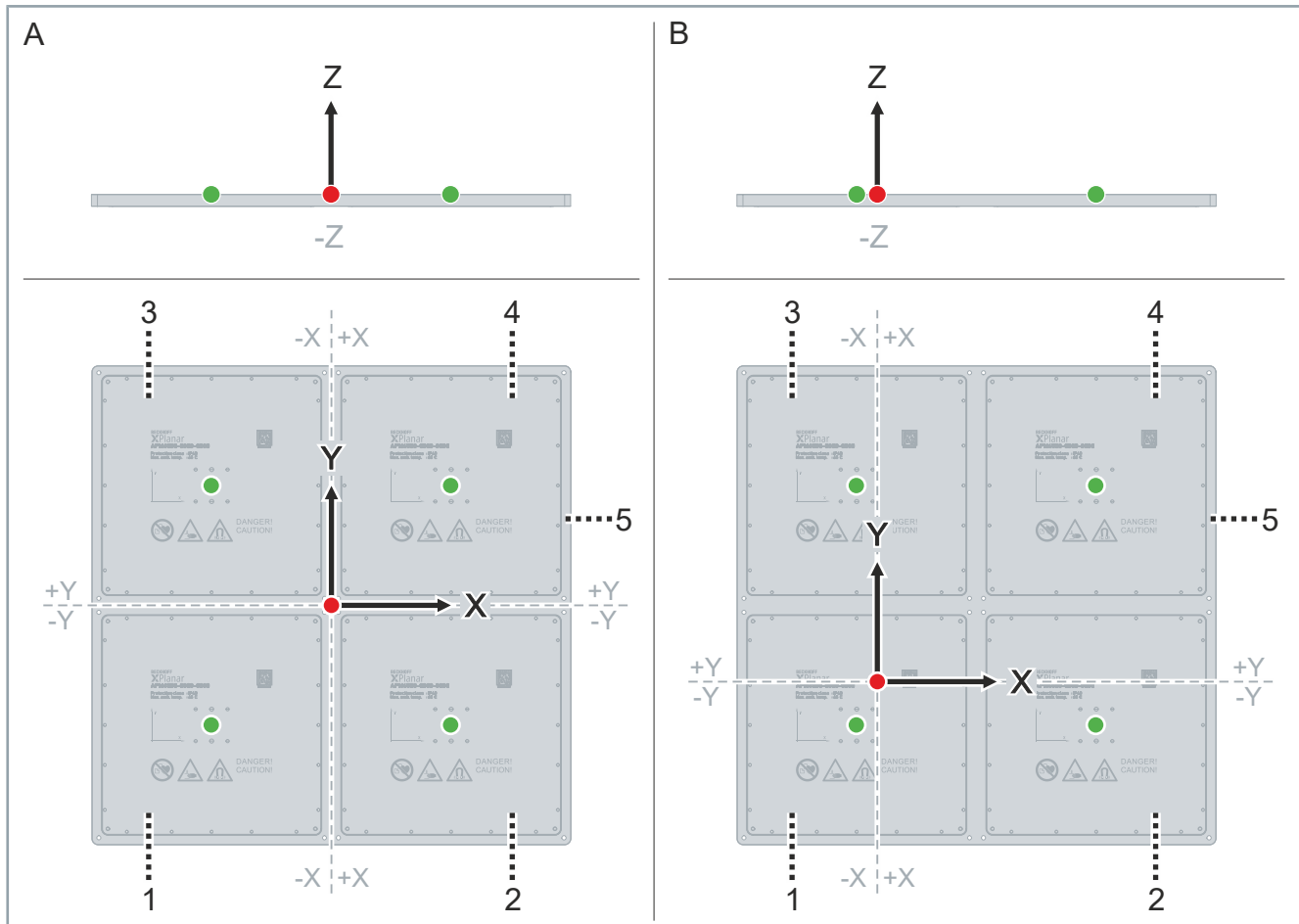
## 12.2 Sollwertursprung

Ausgehend vom Sollwertursprung O müssen die Werte der Sollwertursprünge  $O_X$  der Submover in der X-Achse, Y-Achse und Z-Achse berechnet werden.



### Beispiele für die Positionierung des Sollwertursprungs

Die Positionierung des Sollwertursprungs wird beispielhaft an zwei  $2 \times 2$  Moverkopplungen dargestellt.



Position	Erläuterung
1	Submover 1
2	Submover 2
3	Submover 3
4	Submover 4
5	Rahmen der Moverkopplung
●	Sollwertursprung O der Moverkopplung
●	Sollwertursprung $O_x$ des Submovers, $O_1$ für Submover 1, ...

### Beispiel A

Der Sollwertursprung O befindet sich mittig auf der Oberfläche der Moverkopplung. Die Sollwertursprünge  $O_x$  der Submover haben den gleichen Abstand zum Sollwertursprung O.

### Beispiel B

Der Sollwertursprung O befindet sich nicht in der Mitte der Moverkopplung. Die Sollwertursprünge  $O_x$  der Mover haben unterschiedliche Abstände zum Sollwertursprung O.

Ausgehend vom Sollwertursprung O müssen die Werte der Sollwertursprünge  $O_x$  der Submover in X-Achse, Y-Achse und Z-Achse berechnet werden und unter den Parametern (Init) der *XPlanarRigid-MoverCoupling* eingetragen werden. Die folgenden Tabellen zeigen Werte für die Beispiele A und B der vorherigen Seite mit Movern APS4550-0000.

## X-Achse

Sollwertursprung	O	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>	O <sub>4</sub>
Abstand Beispiel A [mm]	0	-120	120	-120	120
Abstand Beispiel B [mm]	0	-20	220	-20	220

## Y-Achse

Sollwertursprung	O	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>	O <sub>4</sub>
Abstand Beispiel A [mm]	0	-120	-120	120	120
Abstand Beispiel B [mm]	0	-45	-45	195	195

## Z-Achse

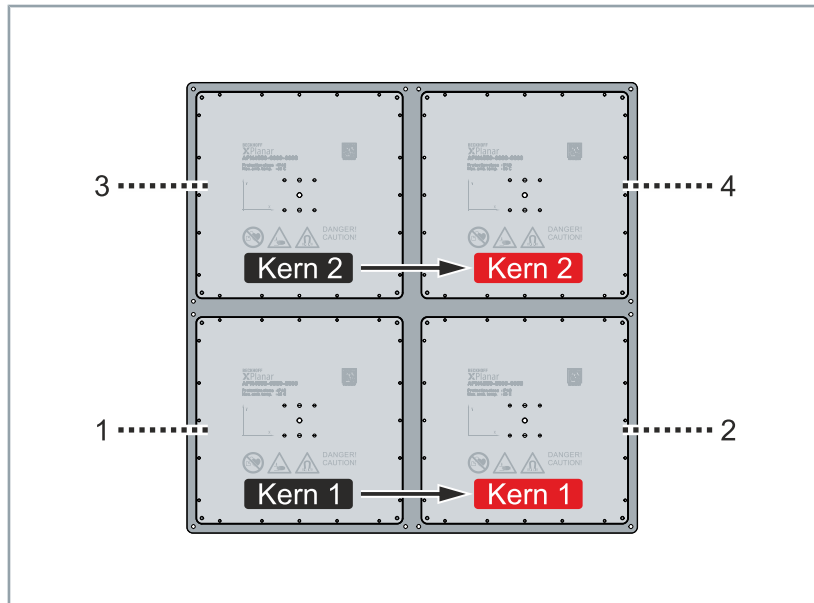
Sollwertursprung	O	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>	O <sub>4</sub>
Abstand Beispiel A [mm]	0	0	0	0	0
Abstand Beispiel B [mm]	0	0	0	0	0

## 12.3 CPU-Kern

### 12.3.1 Moverkopplung 2 x 2

Jeder Mover einer 2 x 2 Moverkopplung benötigt einen eigenen CPU-Kern um Berechnungen in ausreichender Geschwindigkeit durchführen zu können.

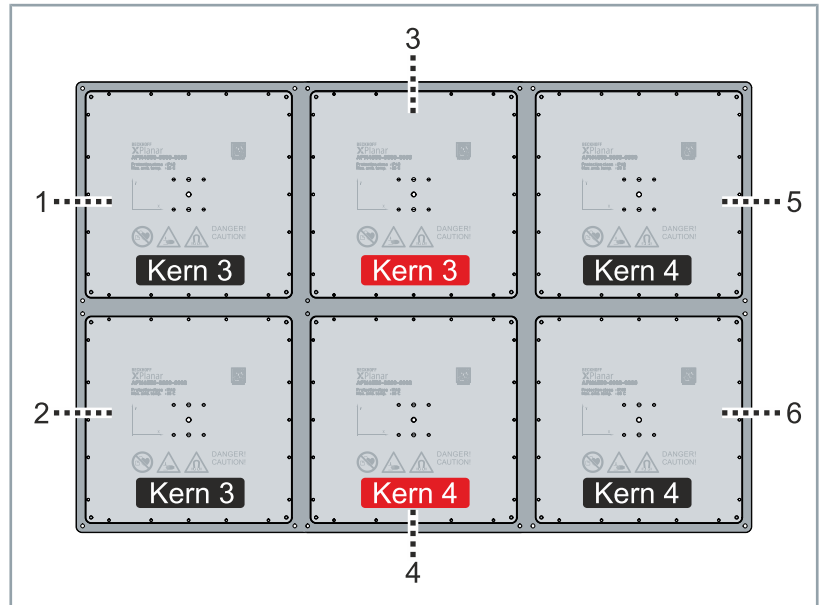
Die für eine Moverkopplung benötigte Task kann sich mit anderen Tasks einen CPU-Kern teilen.



Position	Bezeichnung	CPU-Kern
1	Submover 1	1
2	Submover 2	1
3	Submover 3	2
4	Submover 4	2

## 12.3.2 Moverkopplung 3 x 2

Bei größeren Moverkopplungen von 3 x 2 oder mehr Movern teilen sich jeweils zwei Mover an den Ecken einen CPU-Kern. Da die mittleren Mover weniger Berechnungen durchführen, können sie zu einem der beiden CPU-Kerne der Eckmover hinzugefügt werden.

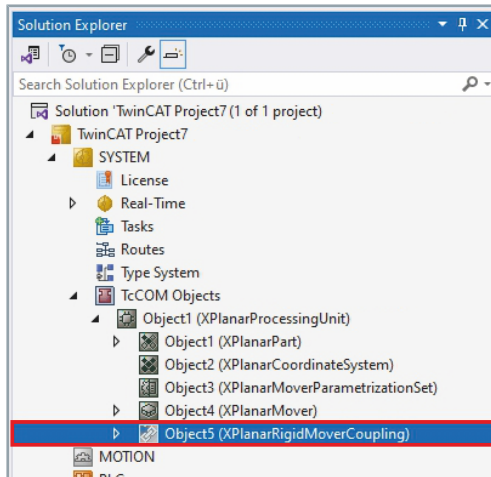


Position	Bezeichnung	CPU-Kern
1	Submover 1	3
2	Submover 2	3
3	Submover 3	3
4	Submover 4	4
5	Submover 5	4
6	Submover 6	4

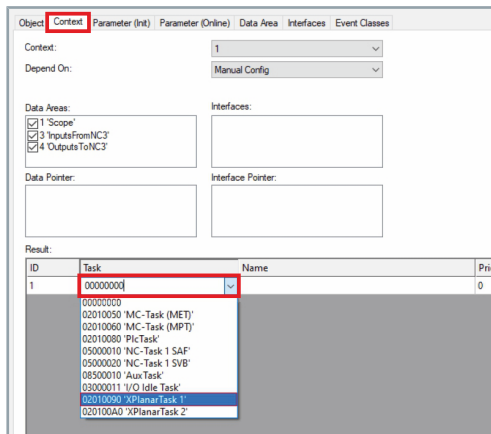
## 12.4 Submover auswählen

Ein TcCom-Objekt *XPlanarMoverCoupling* muss zur Processing Unit hinzugefügt sein, bevor mit der Verknüpfung der Submover begonnen wird. Weitere Informationen dazu finden Sie im Kapitel „Moverkopplung“, [Seite 35].

Die Submover werden automatisch beim Anlegen der *XPlanarRigidMoverCoupling* erstellt.

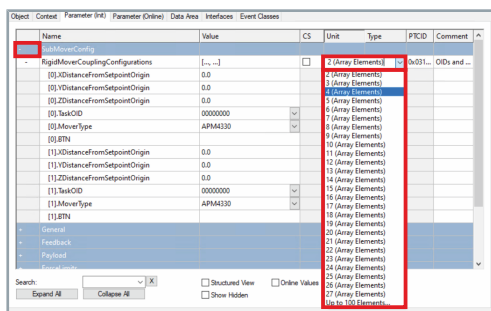


- ▶ *Solution-Explorer* > *System* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* ausklappen
- ▶ Doppelklick auf *Object (XPlanarRigidMoverCoupling)*



- ▶ Im Projektfenster auf den Reiter **Context** klicken
- ▶ Im Dropdown-Menü eine Task auswählen, die als primäre Task für alle Objekte verwendet werden soll

Die ausgewählte Task muss mindestens für einen Submover verwendet werden.



- ▶ Im Projektfenster auf den Reiter **Parameter (Init)** klicken
- ▶ *SubMoverConfig* ausklappen
- ▶ Die Anzahl der Submover im Dropdown-Menü *RigidMoverCouplingConfigurations* auswählen

Für jeden Submover einer Moverkopplung werden folgende Eingabefelder erstellt:

- XDistanceFromSetpointOrigin
- YDistanceFromSetpointOrigin
- ZDistanceFromSetpointOrigin
- TaskOID
- MoverType
- BTN



## Nur ein Movertyp möglich

Für jede Moverkopplung kann nur ein Movertyp ausgewählt werden. Die Auswahl unterschiedlicher Movertypen in einer Moverkopplung ist unzulässig.

Name	Value	CS	Unit	Type	PTCID	Comment
[0] XDistanceFromSetpointOrigin	0.0		mm	LREAL		
[0] YDistanceFromSetpointOrigin	0.0		mm	LREAL		
[0] ZDistanceFromSetpointOrigin	0.0		mm	LREAL		
[0] TaskOID	00000000			OTCID		
[0] MoverType	AP44130			TaskPlan...		
[0] BTN	AP44120		mm	STRINGID		
[1] XDistanceFromSetpointOrigin	0.0		mm	LREAL		
[1] YDistanceFromSetpointOrigin	0.0		mm	LREAL		
[1] ZDistanceFromSetpointOrigin	0.0		mm	LREAL		
[1] TaskOID	00000000			OTCID		
[1] MoverType	AP44130			TaskPlan...		
[1] BTN	AP44130			STRINGID		

► MoverTyp im Dropdown-Menü *MoverType* auswählen

Name	Value	CS	Unit	Type	PTCID	Comment
[0] XDistanceFromSetpointOrigin	0.0		mm	LREAL		
[0] YDistanceFromSetpointOrigin	0.0		mm	LREAL		
[0] ZDistanceFromSetpointOrigin	0.0		mm	LREAL		
[0] TaskOID	00000000			OTCID		
[0] MoverType	AP44130			TaskPlan...		
[0] BTN	AP44120		mm	STRINGID		
[1] XDistanceFromSetpointOrigin	0.0		mm	LREAL		
[1] YDistanceFromSetpointOrigin	0.0		mm	LREAL		
[1] ZDistanceFromSetpointOrigin	0.0		mm	LREAL		
[1] TaskOID	00000000			OTCID		
[1] MoverType	AP44130			TaskPlan...		
[1] BTN	AP44130			STRINGID		

► In die Eingabefelder *XDistanceFromSetpointOrigin*, *YDistanceFromSetpointOrigin* und *ZDistanceFromSetpointOrigin* Werte für die Position des Submovers zum Sollwertursprung eingeben

Weitere Informationen dazu finden Sie im Kapitel „Sollwertursprung“, [Seite 70].

► Eine Task im Dropdown-Menü *TaskOID* auswählen

Mindestens ein Submover muss dieselbe Task verwenden, die für die Kopplung ausgewählt wurde.

► Bei Bedarf in das Eingabefeld *BTN* die BTN des Submovers eingeben

Weitere Informationen zur BTN finden Sie in der XPlanar Original-Betriebsanleitung oder unter:

[www.beckhoff.com/bic](http://www.beckhoff.com/bic)

► Werte für alle Submover auf die gleiche Weise eingeben

Name	Value	CS	Unit	Type	PTCID	Comment
SubMoverConfig						
AccessoriesDis	AP44130_0001		1 (Ara...		0x03...	Select if t...
ParametrizationSetId	00000000			OTCID	0x03...	If not assi...
Feedback	00000000					
Payload	01010010 'RParam'					
ForceMode	01010020 'RParam'					
Controller	01010040 AP44132_P1_1					
ExternalFeedback	01010050 AP44132_P1_3					
SimulationMode	01010060 AP44132_P1_4					
Methods	01010090 AP44132_P1_7					
	010100A0 AP44132_P1_8					
	010100B0 AP44132_P1_9					
	010100C0 Mover430_1					
	010100D0 Mover430_2					
	010100E0 Mover430_3					
	010100F0 Mover430_4					
	02010000 'RParam Task_3'					
	02010070 Job Task					
	02010080 'MC Task (M4T)'					
	02010090 'MC Task (MPT)'					
	03000011 'V0 die Task'					
	05100010 Const11					
	05110010 'Mover430_1'					
	05110020 'Mover430_2'					

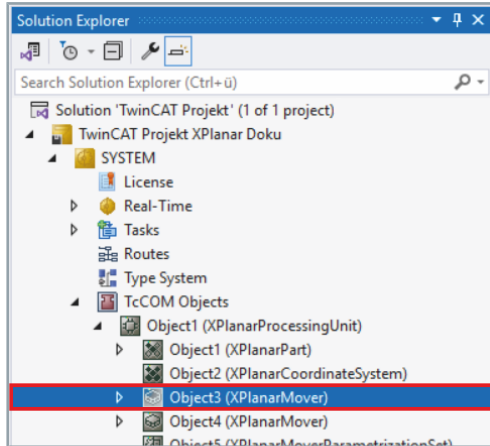
► *General* ausklappen

► Bei Bedarf im Dropdown-Menü *ParametrizationSetId* einen Parametrierungssatz auswählen

Die Moverkopplung ist aktiv, sobald alle Mover im definierten Abstand zueinander bei der Mover Detection gefunden werden.

## 13 Parametrierungsset auswählen

Parametrierungssets sind für sowohl für einzelne oder mehrere Mover als auch für Moverkopplungen zu verwenden. Weitere Informationen zu den Parametrierungssets finden Sie im Kapitel „Mover Parametrierungen“, [Seite 77].



- ▶ *Solution-Explorer* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* ausklappen
- ▶ Doppelklick auf **Object (XPlanarMover)** oder auf **Object (XPlanarRigidMoverCoupling)**

Object	Context	Parameter (Init)	Parameter (Online)	Data Area	Interfaces
-	General				
	MoverType	APM4330			MoverT... 0x... The mo...
	BTN				STRING... 0x... 8 digit a...
+	SimulationModeStartPosition	...			0x... In simul...
+	PayloadInertia	...			0x... Inertia ...
	ParametrizationSetId	00000000			
-	ForceLimits	00000000			
+	ForceLimitMax	01010010 'Object1 (XPlanarProcessingUnit)'			
		01010020 'Object7 (AP54322)'			
		01010030 'Object2 (AP54322)'			
		01010040 'Object3 (AP54322)'			
		01010050 'Object4 (AP54322)'			
		01010060 'Object5 (AP54322)'			
		01010070 'Object6 (AP54322)'			
		01010080 'Object1 (XPlanarPart)'			
		01010090 'Object2 (XPlanarMovementArea)'			
		010100B0 'Object4 (XPlanarMover)'			
		010100C0 'Object5 (XPlanarMover)'			
		010100D0 'Object6 (XPlanarMover)'			
		010100E0 'Object7 (XPlanarMoverCoupling)'			
		010100F0 'Object8 (XPlanarMoverParametrizationSet)'			
		01010100 'Object1 (GeneralParametrization_MoverCoupling)'			
		01010110 'Object9 (XPlanarMoverParametrizationSet)'			
		01010120 'Object10 (XPlanarMoverParametrizationSet)'			
		01010130 'Object11 (XPlanarMoverParametrizationSet)'			
		01010140 'Object1 (GeneralParametrization_SubMover)'			
		01010150 'Object1 (GeneralParametrization_APM4xxx_0000)'			
		01010160 'Object1 (GeneralParametrization_APM4xxx_0000)'			
		01010170 'Object2 (ObserverParametrization_Plain)'			
		03000011 'I/O Idle Task'			

- ▶ Im Projektfenster auf den Reiter **Parameter (Init)** klicken
- ▶ Parametrierungsset im Dropdown-Menü **ParametrizationSetId** auswählen

## 14 Mover Parametrierungen

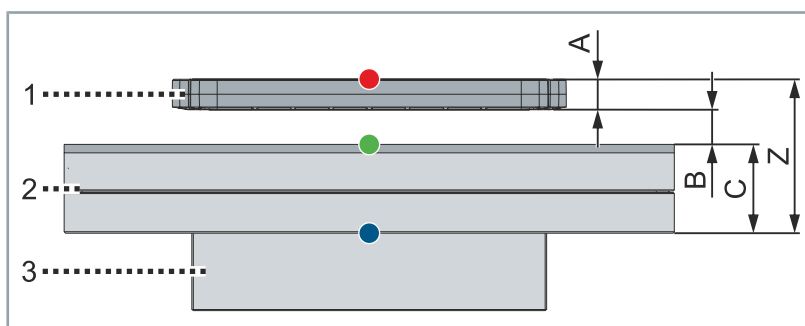
In diesem Kapitel werden die Funktionen der unterschiedlichen Parametrierungen erläutert. Die Parametrierungen werden in Parametrierungssätzen festgelegt und können auf Mover und Moverkopplungen angewendet werden.

Weitere Informationen dazu finden Sie in den Kapiteln „Mover Parametrierung“, [Seite 36], „Parametrierungsset auswählen“, [Seite 76] und Parameter.

### 14.1 General Parametrization

Mit der General Parametrierung werden die Flugeigenschaften der Mover und Moverkopplungen definiert.

#### 14.1.1 Flughöhe



Position	Erläuterung
1	Mover
2	Kachel
3	Abdeckung
●	Moveroberfläche: Feedback und Sollwertursprung des Movers
●	Kacheloberseite
●	Kachelunterseite

Die Flughöhe errechnet sich aus der Höhe des Movers, der Flughöhe des Movers und der Höhe der Kachel:

$$A + B + C = Z$$

Variable	Erläuterung
A	Höhe des Movers
B	Flughöhe des Movers
C	Höhe der Kachel: Grundprofil und Stator
Z	Sollwert Z-Position des Movers: Abstand von der Unterseite der Kachel zur Oberfläche des Movers

Je nach Referenztyp werden unterschiedliche Berechnungen ange­stellt. Entscheidend ist, ob eine feste Flughöhe für alle Mover defi­niert werden soll oder ob eine gemeinsame Oberfläche für alle Mover erreicht werden soll, unabhängig von der Höhe des Movers.

## 14.1.2 Ohne gemeinsame Referenzfläche

Bei dieser Parametereinstellung haben die Mover die gleiche Flug­höhe, unabhängig von der Höhe der Mover. Der Sollwertursprung, der sich auf der Moveroberfläche befindet, liegt bei unterschiedlich hohen Movern nicht auf der gleichen Höhe. Mit folgender Formel lassen sich die Sollwerte der Z-Position für die Mover berechnen:

$$Z_x = A_x + B + C$$



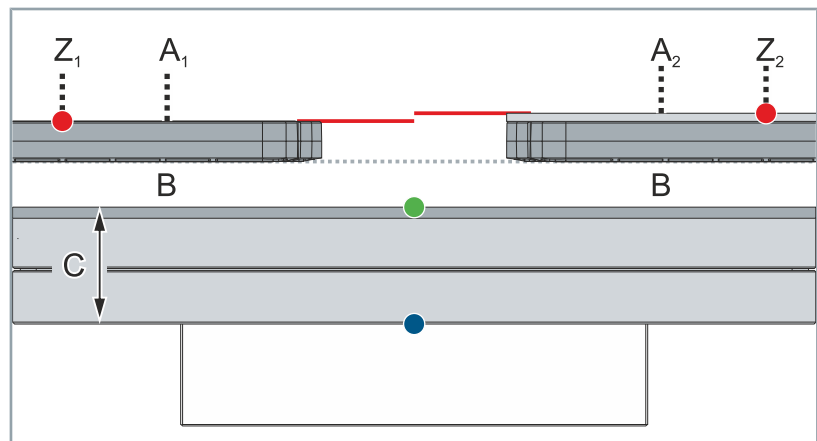
Im folgenden Beispiel werden die Sollwert Z-Positionen für zwei un­terschiedlich hohe Mover berechnet. Mover  $A_1$  ist 12,1 mm hoch und Mover  $A_2$  ist 13,8 mm hoch. Die Kachel C ist 35,2 mm hoch und die Flughöhe B beträgt 2 mm.

	$A_x$	B	$C_x$	$Z_x$
Höhe [mm]	12,1	2	35,2*	49,3*
Höhe [mm]	12,1	2	48,8**	62,9**
Höhe [mm]	13,8	2	35,2*	51*
Höhe [mm]	13,8	2	48,8**	64,6**

\* bei APM4322-0000

\*\* bei APM42xx-1x00

Wird für die Referenz Z-Position keine gemeinsame Referenzfläche angegeben, befinden sich die Moveroberflächen auf unterschiedli­chen Höhen. Der Höhenunterschied zwischen den Moveroberflächen und den Sollwerten für die Z-Position der beiden Mover beträgt 1,7 mm.



## 14.1.3 Mit gemeinsamer Referenzfläche

Bei dieser Parametereinstellung haben die Mover einen gemeinsamen Sollwert für die Z-Position, unabhängig von der Höhe der Mover. Der Sollwertursprung, der sich auf der Moveroberfläche befindet, liegt bei unterschiedlich hohen Movern auf der gleichen Höhe. Die Referenzfläche zum Sollwert der Z-Position ist die Kachelunterseite. Mit folgender Formel lassen sich die Flughöhen für die Mover berechnen:

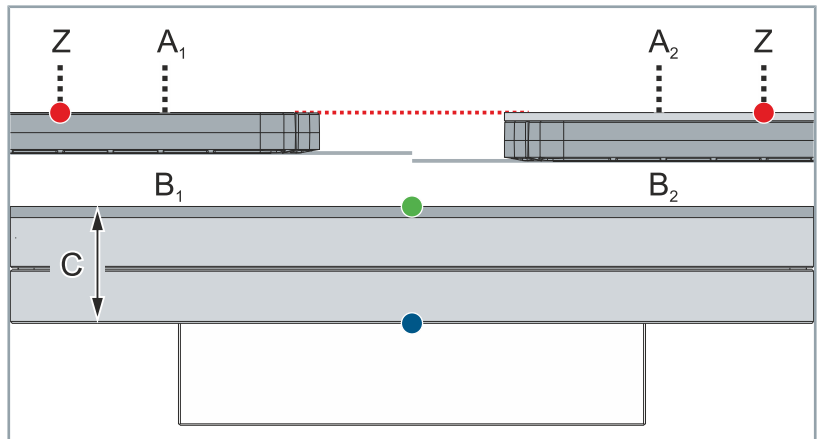
$$Z = A_x + B_x + C \longrightarrow B_x = Z - A_x - C$$



Im folgenden Beispiel werden die Flughöhen für zwei unterschiedlich hohe Mover berechnet. Mover  $A_1$  ist 12,1 mm hoch und Mover  $A_2$  ist 13,8 mm hoch. Die Kachel  $C$  ist 35,2 mm hoch und die Sollwert Z-Position beträgt 50 mm.

	$A_x$	$C$	$Z$	$B_x$
Höhe [mm]	12,1	35,2	50	2,7
Höhe [mm]	13,8	35,2	50	1

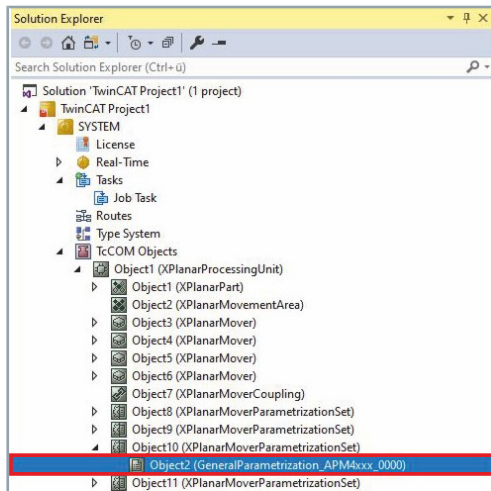
Wird für die Referenz Z-Position eine gemeinsame Referenzfläche angegeben, befinden sich die Moveroberflächen auf gleicher Höhe. Der Flughöhen der beiden Mover unterscheiden sich um 1,7 mm, um die gemeinsame Referenz Z-Position zu erreichen.



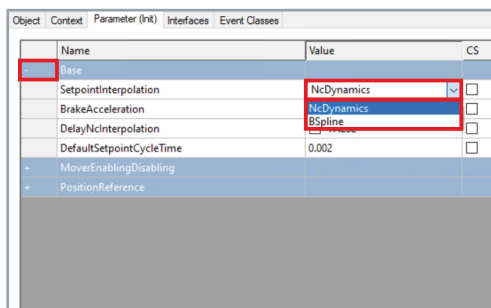
# Mover Parametrierungen

## 14.1.4 Parameter festlegen

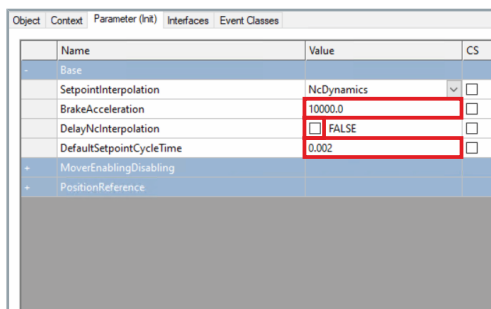
### 14.1.4.1 APM4xxx-0000



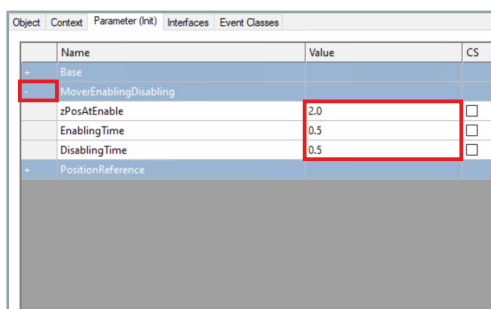
- ▶ *Solution-Explorer* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarMoverParametrizationSet)* ausklappen
- ▶ Doppelklick auf **Object (GeneralParametrization)**



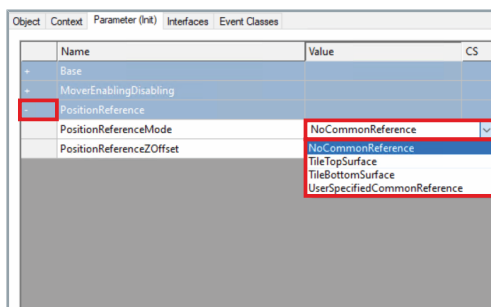
- ▶ Im Projektfenster auf den Reiter **Parameter (Init)** klicken
- ▶ *Base* ausklappen
- ▶ Art der Sollwertinterpolierung im Dropdown-Menü **SetpointInterpolation** auswählen



- ▶ Bei Bedarf Werte für **BrakeAcceleration** und **DefaultSetpointCycleTime** ändern
- ▶ Kontrollkästchen von **DelayNcInterpolation** aktivieren, um *FAL-SE* auszuwählen, wenn die NC-Interpolation um einen NC-Zyklus verzögert werden soll



- ▶ Bei Bedarf Werte für **zPosAtEnable**, **EnablingTime** und **DisablingTime** ändern



- ▶ Z-Referenzposition im Dropdown-Menü **PositionReferenceMode** auswählen

Object	Context	Parameter (Init)	Interfaces	Event Classes
Name	Value	CS		
+	Base			
+	MoverEnablingDisabling			
-	PositionReference			
	PositionReferenceMode	NoCommonReference	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	PositionReferenceZOffset	0.0	<input type="checkbox"/>	<input type="checkbox"/>

Show Online Values  
 Show Hidden Parameter  
  

- Wert für **PositionReferenceZOffset** eingeben

- Zum Anzeigen der *Hidden Parameter* das Kontrollkästchen **Show Hidden Parameter** wählen

Object	Context	Parameter (Init)	Interfaces	Event Classes		
Name	Value	CS	Unit	Type	PIPID	Control
	SetupInterpolation	NoDynamics	<input checked="" type="checkbox"/>		TcolPlanar1	Extern
	BrakeAcceleration	10000.0	<input type="checkbox"/>	mm/s <sup>2</sup>	LREAL	Brake4
	DefaultSetupCycleTime	0.002	<input type="checkbox"/>	s	LREAL	Default
<b>MoverEnablingDisabling</b>						
	zPosAtEnable	2.0	<input type="checkbox"/>	mm	LREAL	When
	zRangeAtEnable	0.1	<input type="checkbox"/>	mm	LREAL	Accept
	TimeToStayAtEnablingHeight	1.0	<input type="checkbox"/>	s	LREAL	Alter
	EnablingForceIncrement	0.0	<input type="checkbox"/>	N/cycle	LREAL	Define
	DisablingForceDecrement	0.0	<input type="checkbox"/>	N/cycle	LREAL	Define
	EnablingTime	0.5	<input type="checkbox"/>	s	LREAL	Define
	DisablingTime	0.5	<input type="checkbox"/>	s	LREAL	Define
<b>PositionReference</b>						
	PositionReferenceMode	NoCommonReference	<input checked="" type="checkbox"/>		TcolPlanar1	NoCo
	PositionReferenceZOffset	0.0	<input type="checkbox"/>	mm	LREAL	Only t
<b>MoverCommunication</b>						
	MoverCommAmplitude	10000	<input type="checkbox"/>		UDINT	Value
	MoverCommThresholdSymul	25	<input type="checkbox"/>		UDINT	Event
	MoverCommUseOneEtherCatBranch	TRUE	<input checked="" type="checkbox"/>		BOOL	Set to
	MoverOrientationDcCurrentFactor	1.0	<input type="checkbox"/>		LREAL	
	MoverOrientationDeviationDiffThres	30.0	<input type="checkbox"/>		LREAL	
	MoverHallSensorRepetitions	32	<input type="checkbox"/>		UDINT	

- Wert für **zRangeAtenable** und **TimeToStayAtEnablingHeight** bei Bedarf ändern
- Wert für **MoverCommAmplitude** und **MoverCommPeriodsPerSymbol** bei Bedarf ändern
- Kontrollkästchen von **MoverCommUseOneEtherCatBranch** aktivieren, um **TRUE** auszuwählen, wenn keine synchronisierten EtherCAT-Ports verfügbar sind

ODER

- Kontrollkästchen von **MoverCommUseOneEtherCatBranch** deaktivieren, um **FALSE** auszuwählen, wenn synchronisierte EtherCAT-Ports verfügbar sind
- Wert für **MoverOrientationDcCurrentFactor**, **MoverOrientationDeviationDiffThreshold** und **MoverHallSensorRepetition** bei Bedarf ändern

## 14.2 Observer Parametrization

Die Observer Parametrierungen werden für die X-Achse und Y-Achse zum Verfahren, für die Z-Achse zum Heben und Senken, für die A-Achse und B-Achse zum Neigen und für die C-Achse zum Drehen eingegeben.

Von der Observer Parametrierung stehen Ihnen vier unterschiedliche Varianten zur Auswahl:

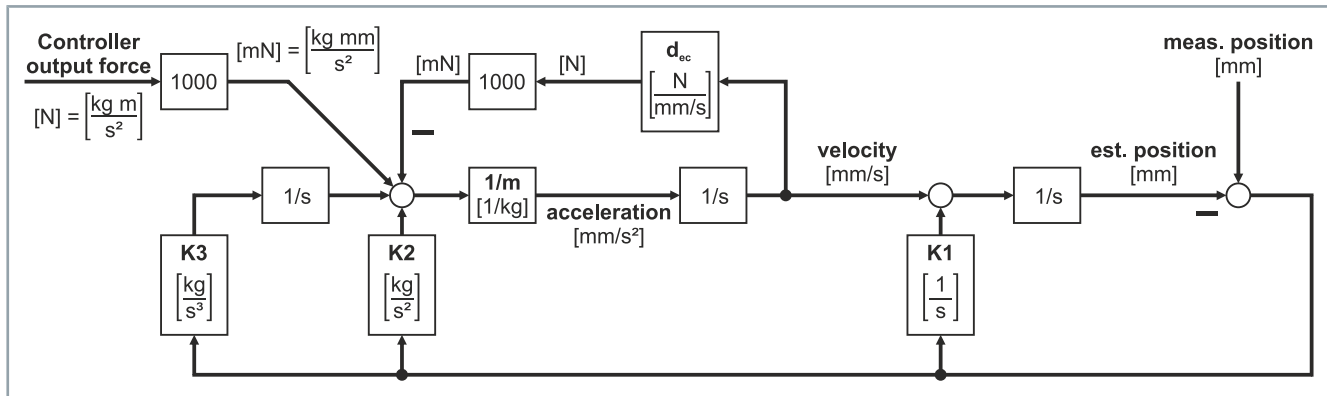
- Plain
- APM4220
- APM4330
- APM4550

Alle Observer Parametrierungen verfügen über die gleichen Funktionen und die gleichen Parameter. Die Plain Parametrierung kann zur Abstimmung für eine Moverkopplung verwendet werden. Die *APM4xxx* Parametrierungen besitzen vordefinierte Werte, die speziell auf den Movertyp voreingestellt sind, und sollten entsprechend dem Movertyp verwendet werden. Die voreingestellten Werte dienen zur Orientierung und können entsprechend Ihrer Anwendung abgestimmt werden.

### 14.2.1 Linear axes observer

Mit einem Observer Parametrierungssatz können die Parameter  $K1$ ,  $K2$ ,  $K3$  und  $d_{ec}$  in der X-Achse, Y-Achse und Z-Achse angepasst werden.

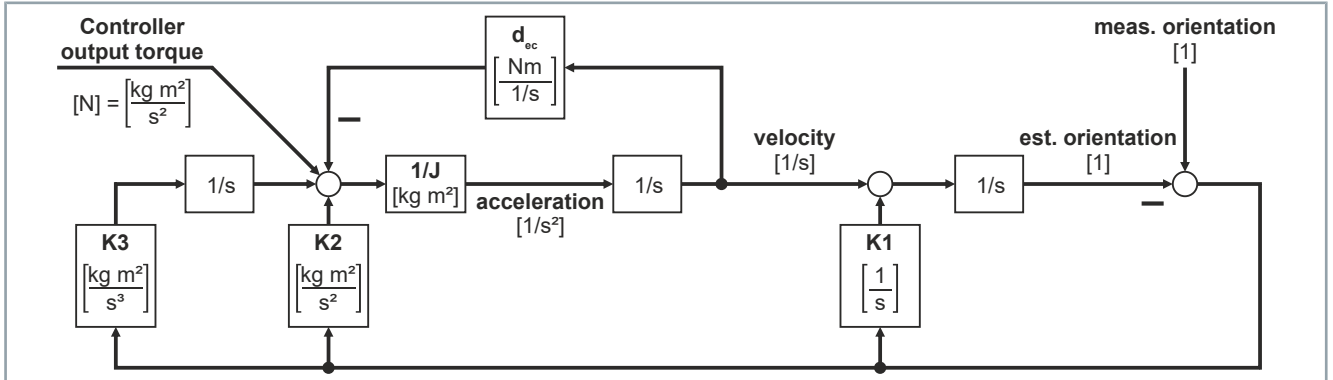
Die Masse ( $m$ ) wird abhängig vom PayloadInertia Parameter eines Moverobjekts oder durch die BasicInertia Parameter eines Inertia Parametrierungssatzes angepasst.



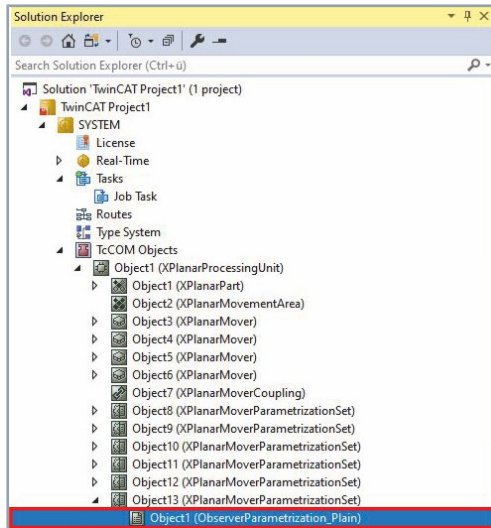
## 14.2.2 Rotational axes observer

Mit einem Observer Parametrierungssatz können die Parameter  $K1$ ,  $K2$ ,  $K3$  und  $d_{ec}$  in den A-Achsen, B-Achsen und C-Achsen angepasst werden.

Die Trägheit ( $J$ ) wird abhängig vom PayloadInertia Parameter eines Moverobjekts oder durch die BasicInertia Parameter eines Inertia Parametrierungssatzes angepasst.



## 14.2.3 Parameter festlegen



- ▶ *Solution-Explorer* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarMoverParametrizationSet)* ausklappen
- ▶ Doppelklick auf **Object (ObserverParametrization\_...)**



### Voreingestellte Werte

Die Observer Parametrierungen für die Mover *APM4220-0000*, *APM4330-0000* und *APM4550-0000* verfügen über voreingestellte Werte für die Parameter *K1*, *K2*, *K3* und *d\_ec*. Bei Bedarf können Sie die moverspezifischen Werte anpassen, sie dienen als Orientierungshilfen bei der Bestimmung Ihrer Werte. Der Parametrierungsatz *Plain* verfügt über keine voreingestellten Werte und kann für eine Moverkopplung verwendet werden.

Object	Context	Parameter (Init)	Interfaces			
Name	Value	CS	Unit	Type	P...	Com...
+ XAxisObserverParameters	...	<input type="checkbox"/>			0...	Obser...
+ YAxisObserverParameters	...	<input type="checkbox"/>			0...	Obser...
+ ZAxisObserverParameters	...	<input type="checkbox"/>			0...	Obser...
+ AAxisObserverParameters	...	<input type="checkbox"/>			0...	Obser...
+ BAxisObserverParameters	...	<input type="checkbox"/>			0...	Obser...
- CAxisObserverParameters	...	<input type="checkbox"/>			0...	Obser...
.K1	0.0		1/s	LREAL		Estima...
.K2	0.0		kg*m^...	LREAL		Estima...
.K3	0.0		kg*m^...	LREAL		Estima...
.d_ec	0.0		N*m/(...	LREAL		Eddy c...

- ▶ Im Projektfenster auf den Reiter **Parameter (Init)** klicken
- ▶ Jeweils auf **+** von *XAxisObserverParameters*, *YAxisObserverParameters*, *ZAxisObserverParameters*, *AAxisObserverParameters*, *BAxisObserverParameters* und *CAxisObserverParameters* klicken, um die Eingabefelder für die Filterparameter der Achsen aufzurufen
- ▶ Werte für **K1**, **K2**, **K3** und **d\_ec** eingeben oder ändern

### 14.3 Controller Parametrization

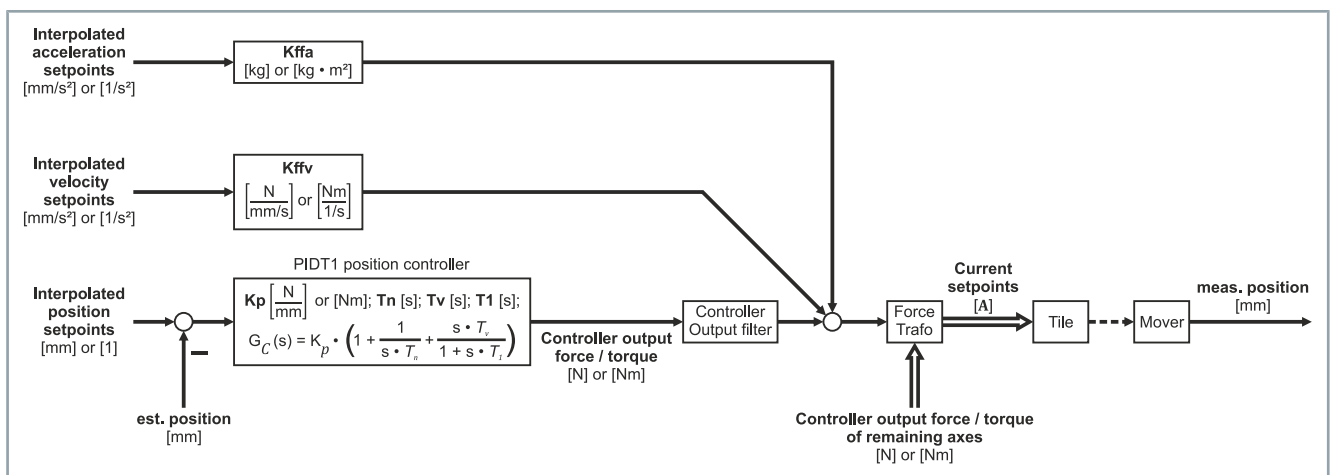
Von den Regelungsparametrierungen stehen Ihnen vier unterschiedliche Varianten zur Auswahl:

- Plain
- APM4220
- APM4330
- APM4550

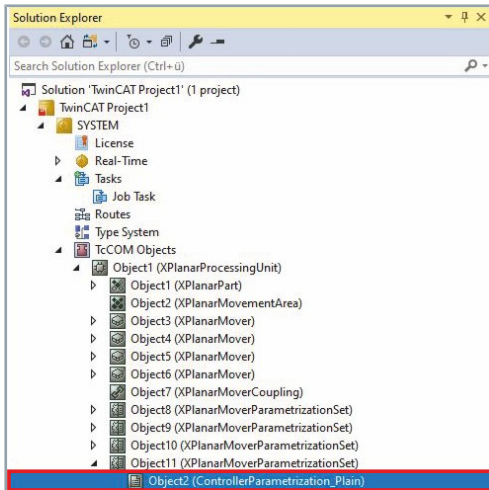
Alle Regelungsparametrierungen verfügen über die gleichen Parameter. Die Plain Parametrierung kann zur Abstimmung für eine Moverkopplung verwendet werden. Die *APM4xxx-000x* Parametrierungen besitzen vordefinierte Werte, die speziell auf den Movertyp voreingestellt sind, und sollten entsprechend dem Movertyp verwendet werden. Die voreingestellten Werte dienen zur Orientierung und können entsprechend Ihrer Anwendung abgestimmt werden.

Sie können die Parameter für die Regler  $K_p$ ,  $T_n$ ,  $T_v$  und  $T_I$  sowie die FeedForward Parameter  $K_{ffa}$  und  $K_{ffv}$  definieren.

Wenn der *AutoAdjust*-Regler aktiviert ist, wird der Parameter  $K_{ffa}$  in Abhängigkeit von der definierten Zuladung angepasst. Ist der *AutoAdjust*-Regler aktiviert, wird der Parameter  $K_p$  in Abhängigkeit der definierten Zuladung angepasst.



## 14.3.1 Parameter festlegen

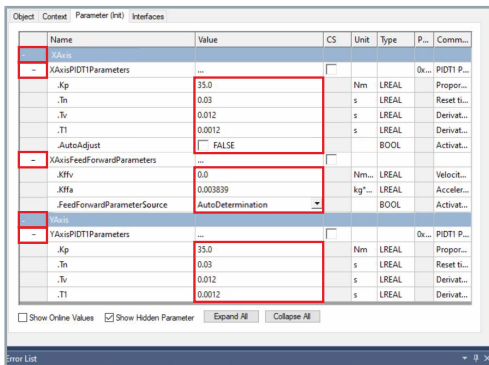


- ▶ *Solution-Explorer* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarMoverParametrizationSet)* ausklappen
- ▶ Doppelklick auf **Object (ControllerParametrization\_...)**
- ▶ Im Projektfenster auf den Reiter **Parameter (Init)** klicken



### Voreingestellte Werte

Die Regelungsparametrierungen für die Mover *APM4220-0000*, *APM4330-0000* und *APM4550-0000* verfügen über voreingestellte Werte für die Parameter *Kp*, *Tn*, *Tv*, *T1*, *Kffa* und *Kffv*. Bei Bedarf können Sie die moverspezifischen Werte anpassen, sie dienen als Orientierungshilfen bei der Bestimmung Ihrer Werte. Der Parametrierungssatz *Plain* verfügt über keine voreingestellten Werte und kann für eine Moverkopplung verwendet werden.



- ▶ Jeweils auf **+** von *XAxis*, *YAxis*, *ZAxis*, *AAxis*, *BAxis* und *CAxis* klicken
- ▶ Jeweils auf **+** *XAxisPIDT1Parameters*, *YAxisPIDT1Parameters*, *ZAxisPIDT1Parameters*, *AAxisPIDT1Parameters*, *BAxisPIDT1Parameters* und *CAxisPIDT1Parameters* ausklappen
- ▶ Werte für **Kp**, **Tn**, **Tv** und **T1** eingeben oder bei Bedarf ändern
- ▶ Kontrollkästchen **FALSE** von *AutoAdjust* aktivieren, um *AutoAdjust* zu aktivieren
- ▶ Jeweils auf **+** von *XAxisFeedForwardParameters*, *YAxisFeedForwardParameters*, *ZAxisFeedForwardParameters*, *AAxisFeedForwardParameters*, *BAxisFeedForwardParameters* und *CAxisFeedForwardParameters* ausklappen
- ▶ Werte für **Kffv** und **Kffa** eingeben oder bei Bedarf ändern
- ▶ **AutoDetermination** im Dropdown-Menü *FeedForwardParameterSource* auswählen

## 14.4 Filter Parametrization



Die Frequenz  $fd$  muss in der Einheit Hz eingegeben werden.

Die *FilterParametrization\_APM4xxx\_0000* kann für jeden Movertyp und Moverkopplung verwendet werden. Vier Filtertypen stehen Ihnen zur Auswahl:

### 14.4.1 Low Pass 1

Für den *Tiefpassfilter 1. Ordnung* muss ein Wert für den Parameter  $fd$  eingegeben werden. Die übrigen Parameter erfordern keine Eingabe. Der *Tiefpassfilter 1. Ordnung* wird durch folgende Formel berechnet:

$$G_f(s) = \frac{1}{1 + s \cdot \tau_d} \quad \text{mit:} \quad \tau = \frac{1}{2\pi \cdot f}$$

### 14.4.2 Low Pass 2

Für den *Tiefpassfilter 2. Ordnung* müssen Werte für den Parameter  $fd$  und  $Dd$  eingegeben werden. Die übrigen Parameter erfordern keine Eingabe. Der *Tiefpassfilter 2. Ordnung* wird durch folgende Formel berechnet:

$$G_f(s) = \frac{1}{\tau_d^2 \cdot s^2 + 2D_d \tau_d \cdot s + 1} \quad \text{mit:} \quad \tau = \frac{1}{2\pi \cdot f}$$

### 14.4.3 Notch

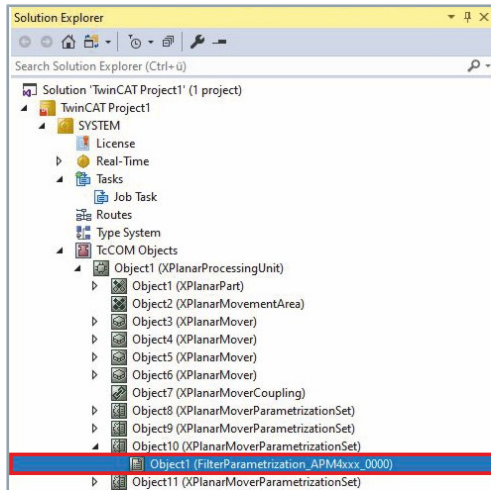
Für den Notch-Filter müssen Werte für alle Parameter eingegeben werden. Der Notch-Filter berechnet sich durch folgende Formel:

$$G_f(s) = \frac{s^2 \tau_n^2 + 2D_n \tau_n \cdot s + 1}{s^2 \tau_d^2 + 2D_d \tau_d \cdot s + 1} \quad \text{mit:} \quad \tau = \frac{1}{2\pi \cdot f}$$

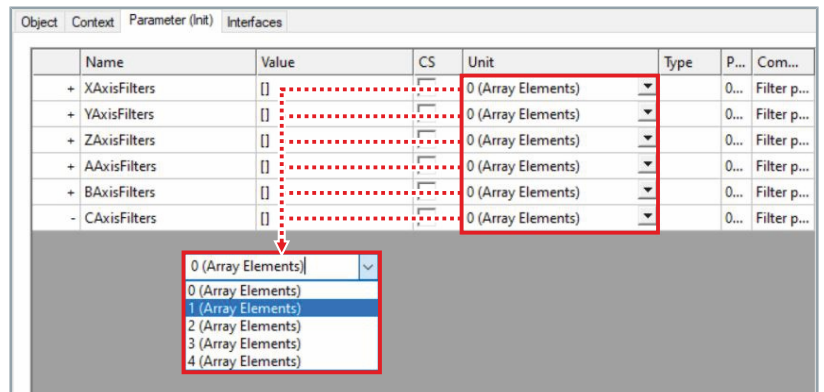
### 14.4.4 Bypassed

Für den Bypassed-Filter müssen keine Werte eingegeben werden. Wenn dieser Filtertyp gewählt wird, ist der Filter inaktiv.

## 14.4.5 Parameter festlegen



- ▶ *Solution-Explorer* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarMoverParametrizationSet)* ausklappen
- ▶ Doppelklick auf **Object (FilterParametrization\_APM-4xxx\_0000)**
- ▶ Im Projektfenster auf den Reiter **Parameter (Init)** klicken



- Im Dropdown-Menü jeweils die Anzahl der **Array Elements** für die Achsen auswählen

Für jedes Array Element werden vier Eingabefelder und drei Dropdown-Menüs zu einer Achse hinzugefügt.



### Mindestens ein Array Element erforderlich

Um Werte in ein Eingabefeld eintragen oder den Filtertyp im Dropdown-Menü auswählen zu können, muss mindestens ein Array Element im Dropdown-Menü der Achsen ausgewählt worden sein. Ohne ein Array Element stehen keine Eingabefelder und Dropdown-Menüs zur Eingabe bereit.



- Auf **+** eines *AxisFilters* klicken, um die Eingabefelder für den Filter aufzurufen
- Art von **Location**, **ZTrafo** und **Type** aus den Dropdown-Menüs auswählen
- Erforderliche Werte für **fd**, **Dd**, **fn** und **Dn** eingeben, je nach ausgewähltem Filtertyp

## 14.5 Inertia Parametrization

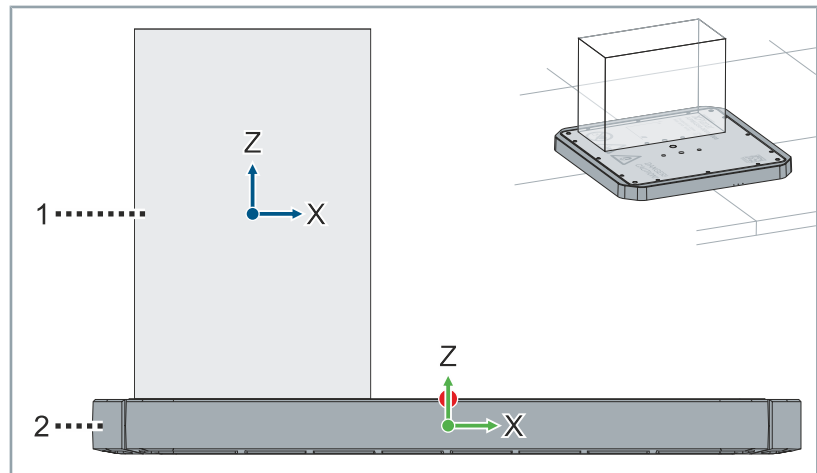
Die Inertia Parametrierung wird als Gesamtträgheit für folgende Objekte definiert:

- Mover mit montiertem Werkzeug
- Moverkopplung, alle Submover mit Rahmen und eventuell montiertem Werkzeug
- Mover mit einer konstanten Nutzlast

Wenn keine Inertia Parametrierung zugewiesen wird oder diese Parametrierung einen Wert von Null hat, werden die standardmäßigen Inertia Parameter des jeweiligen Movertyps verwendet.

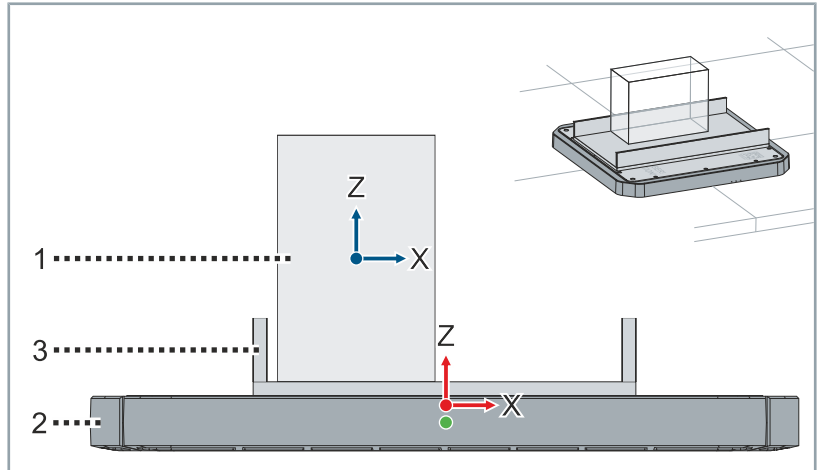
Die Trägheit der Nutzlast wird mit der Trägheit des Movers kombiniert. Dabei wird der Schwerpunkt der Zuladung in Bezug auf den Sollwertursprung des Movers angegeben. Der Sollwertursprung eines Standardmovers befindet sich in der Mitte auf der Oberfläche des Movers. Die Achsen, über die die Trägheit definiert wird, müssen parallel zu den Koordinatenachsen des Movers verlaufen.

### 14.5.1 Mover



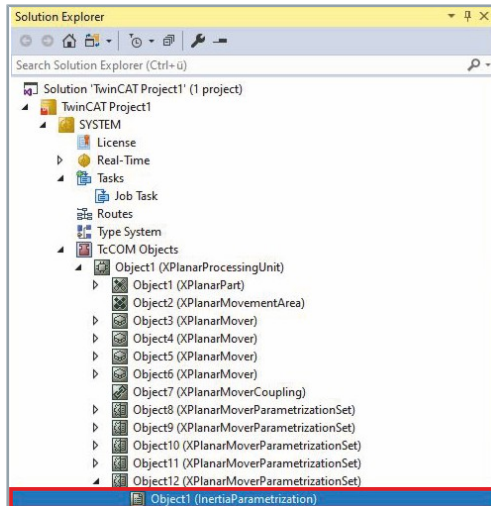
Position	Erläuterung
1	Zuladung
2	Mover
●	Sollwertursprung des Movers
●	Schwerpunkt des Movers
●	Schwerpunkt der Zuladung

## 14.5.2 Mover und montierte Werkzeug



Position	Erläuterung
1	Zuladung
2	Mover
3	Montiertes Werkzeug
●	Schwerpunkt der Grundträgheit
●	Schwerpunkt des Movers
●	Schwerpunkt der Zuladung

## 14.5.3 Parameter festlegen



- ▶ *Solution-Explorer* > *TcCOM Objects* > *Object (XPlanarProcessingUnit)* > *Object (XPlanarMoverParametrizationSet)* ausklappen
- ▶ Doppelklick auf **Object (InertiaParametrization)**

Object	Context	Parameter (Init)	Interfaces
-	BasicInertia	...	<input type="checkbox"/>
	.Mass	0.0	kg LREAL Mass
	.AAxisInertia	0.0	kg*m <sup>2</sup> LREAL A-Axis...
	.BAxisInertia	0.0	kg*m <sup>2</sup> LREAL B-Axis ...
	.CAxisInertia	0.0	kg*m <sup>2</sup> LREAL C-Axis...
	.CentreOfMassX	0.0	mm LREAL X-dista...
	.CentreOfMassY	0.0	mm LREAL Y-dista...
	.CentreOfMassZ	0.0	mm LREAL Z-dista...

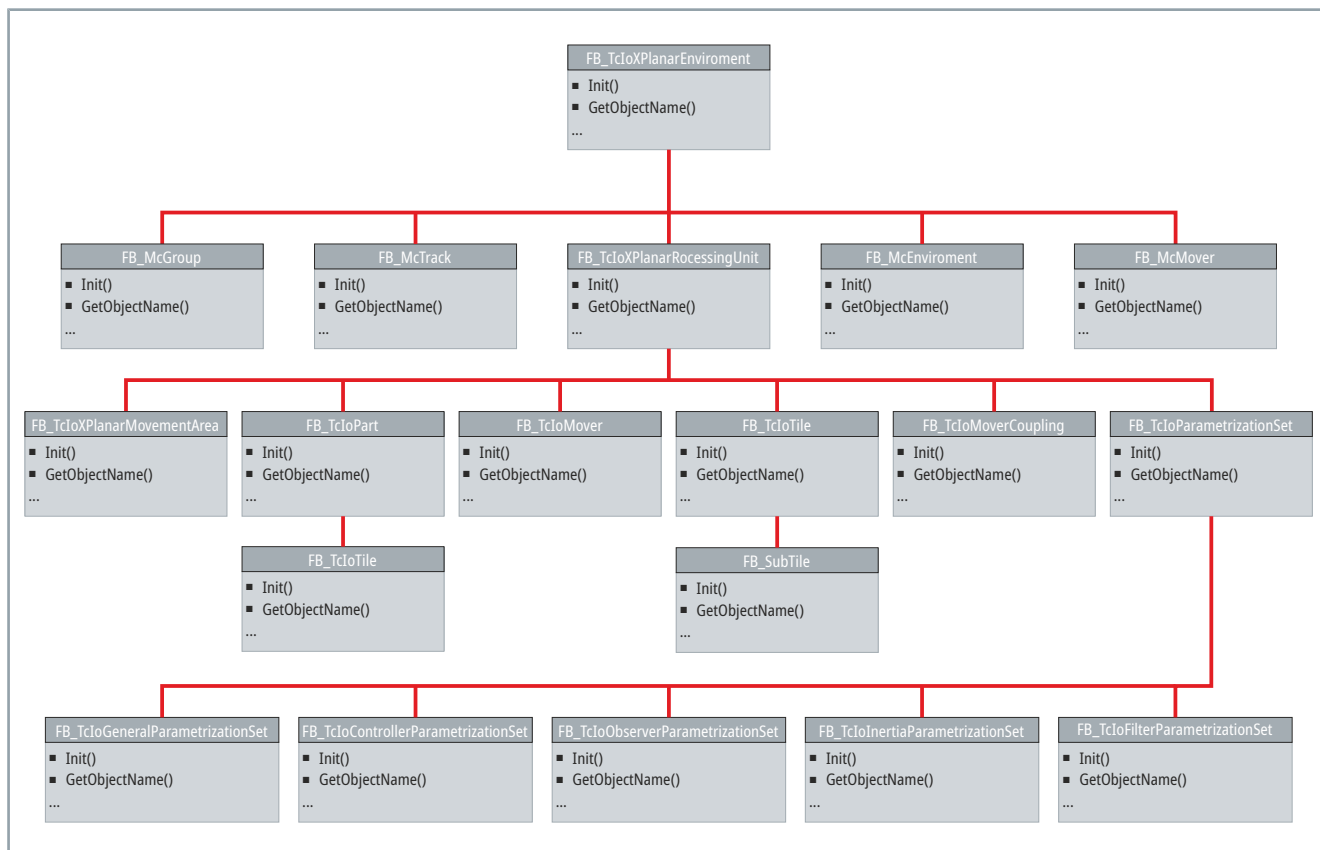
- ▶ Im Projektfenster auf den Reiter **Parameter (Init)** klicken
- ▶ Auf **+** von *BasicInertia* klicken, um die Eingabefelder für die Zuladungsparameter aufzurufen
- ▶ Gewicht der Zuladung über das Feld **Mass** eingeben
- ▶ Trägheitsparameter der Zuladung über die Felder **AAxisInertia**, **BAxisInertia** und **CAxisInertia** eingeben
- ▶ Position des Zuladungsschwerpunkts über die Felder **CentreOfMassX**, **CentreOfMassY** und **CentreOfMassZ** eingeben

## 15 SPS-Bibliotheken

### 15.1 TwinCAT 3 XPlanar Utility

Die TwinCAT 3 XPlanar Utility wird zusammen mit dem TF5890 Software Paket installiert. Die TwinCAT 3 XPlanar Utility ist eine PLC-Bibliothek und wird zum Lesen oder Setzen von Parametern des XPlanar Systems verwendet.

Die folgende Grafik zeigt den Aufbau der TwinCAT 3 XPlanar Utility:



## 15.1.1 Versionsstand überprüfen

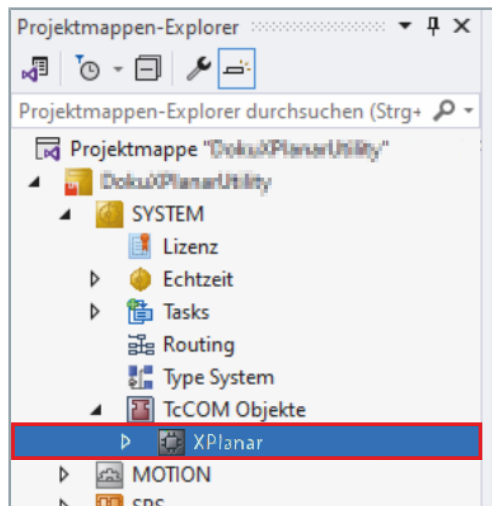
Um die korrekte Funktion zu gewährleisten, stellen Sie sicher, dass die Major-Version und Minor-Version der Tc3\_XPlanarUtility und des XPlanar-Treibers übereinstimmen. Die Tabelle zeigt beispielhaft eine kompatible Konfiguration von Tc3\_XPlanarUtility und XPlanar Treiber:

Produkt	Major	Minor	Revision	Build
Tc3_XPlanarUtility	4	5	13	0
XPlanar TcCOM Objects	4	5	31	0
XPlanar Treiber				

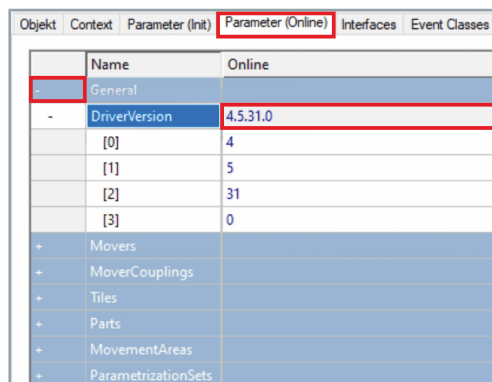
### 15.1.1.1 TcCOM-Objekte

Sie haben zwei Möglichkeiten den Versionsstand der TcCOM-Objekte zu überprüfen:

#### Parameter Online

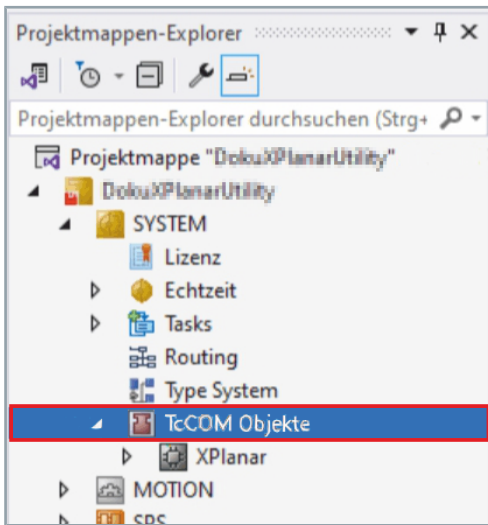


- ▶ *Solution Explorer* > *TwinCAT Project* > *SYSTEM* > *TcCOM Objects* ausklappen
- ▶ Doppelklick auf **XPlanar**

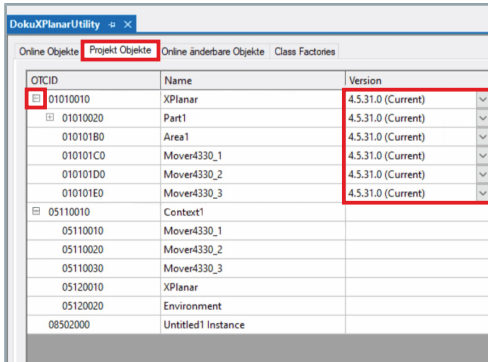


- ▶ Im Projektfenster auf den Reiter **Parameter (Online)** klicken
- ▶ *General* ausklappen
- ▶ Den Versionsstand von *Driver Version* überprüfen

## Projekt Objekte

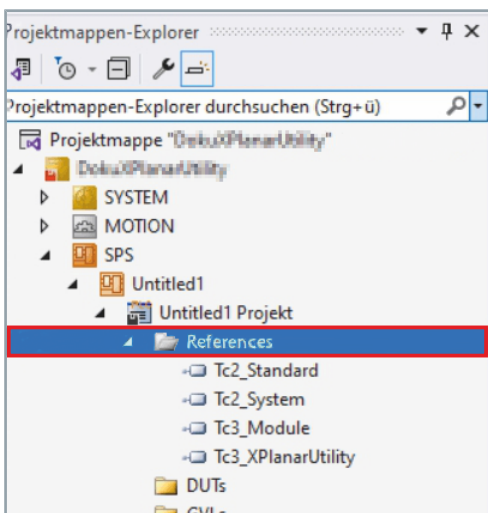


- ▶ *Solution Explorer* > *TwinCAT Project* > *SYSTEM* ausklappen
- ▶ Doppelklick auf **TcCOM Objekte**

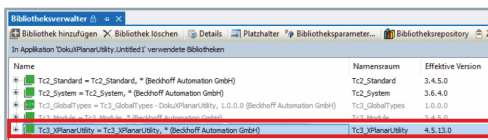


- ▶ Im Projektfenster auf den Reiter **Projects (Online)** klicken
- ▶ 01010010 ausklappen
- ▶ Den Versionsstand überprüfen

## 15.1.1.2 Tc3\_XPlanarUtility



- ▶ *Solution Explorer* > *TwinCAT Project* > *SPS* > *Untitled* > *Untitled Project* ausklappen
- ▶ Doppelklick auf **References**



- ▶ Im Bibliotheksverwalter überprüfen, welche Version der Tc3\_XPlanarUtility verwendet wird

## 15.1.2 Initialisierung

Der Funktionsbaustein *FB\_TcIoXPlanarEnvironment* muss einmal für das XPlanar Projekt instanziiert werden, damit auf alle Parameter der Processing Unit zugegriffen werden kann. Wenn im Projekt die Visualisierung verwendet werden soll, muss zusätzlich der Funktionsbaustein *FB\_XPlanarUtilityVisu* eingerichtet werden.

Der Funktionsbaustein *FB\_TcIoXPlanarEnvironment* muss einmal beim Start initialisiert werden, um Informationen über alle Objekte zu sammeln. Optional können die Statusbits überprüft werden, um ein Feedback des Funktionsblocks zu erhalten. Nach der Initialisierung kann auf alle mit dem *XPlanarEnvironment* unterlagerten Objekten durch Aufruf der entsprechenden Methoden zugegriffen werden. Es gibt keine zyklische Aktualisierung der Daten.

Neben der Verwendung des Funktionsblocks *FB\_TcIoXPlanarEnvironment* haben Sie die Möglichkeit, die Funktionsblöcke der *Tc3\_XPlanarUtility* als Standalone-Variante zu verwenden. Jeder Funktionsblock hat seine eigene Initialisierungsmethode, Sie können diese Methode verwenden, um den Funktionsblock zu initialisieren und den entsprechenden Funktionsblock als Standalone-Variante zu verwenden.

Wenn *FB\_TcIoXPlanarEnvironment* die Eigenschaft *Environment* der *FB\_XPlanarUtilityVisu* zugewiesen wurde, muss die Methode *Cycle()* zyklisch aufgerufen werden.

## 15.1.2.1 Beispiel-Code

```
VAR
    nStateInitEnvironment      : INT;
    fbXplanarEnvironment       : FB_TcIoXPlanarEnvironment;
    fbVisuXPlanarDiag          : FB_XPlanarUtilityVisu;
    ipXpu                       : I_TcIoXplanarProcessingUnit;
END_VAR
CASE nStateInitEnvironment OF
0: // Init Xplanar Environment
    IF fbXplanarEnvironment.Init(TRUE) THEN
        fbXplanarEnvironment.Init(FALSE);
        ipXpu := fbXplanarEnvironment.XpuTcIo(1);
        nStateInitEnvironment := nStateInitEnvironment + 1;
    END_IF

1: // Init Visu
    fbVisuXPlanarDiag.Environment := fbXplanarEnvironment;
    nStateInitEnvironment := nStateInitEnvironment + 1;

ELSE
    fbVisuXPlanarDiag.Cycle();
END_CASE
```

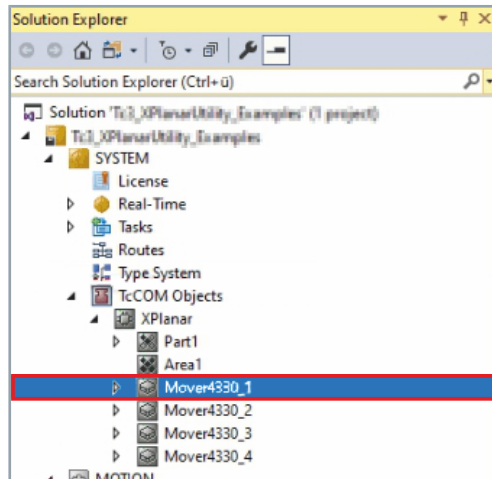
The screenshot displays the Beckhoff Tc3 software interface. On the left, the Solution Explorer shows a project structure with a red box highlighting the 'XPlanar MC Project' under 'MOTION'. The Library Manager in the center shows a list of libraries, with 'Param\_TcloXPlanarEnvironment' selected. On the right, the Library Parameters window shows a table of parameters for 'Param\_TcloXPlanarEnvironment'.

Name	Type	Value (editable)	Comment
MaxProcessingUnits	UDINT	4	
MaxTiles	UDINT	100	
MaxTilesPerPart	UDINT	100	
MaxXPlanarMoversPerXpu	UDINT	40	Number of XPlanar Movers used in this project
MaxMoverParametrizationSets	UDINT	10	
MaxAreas	UDINT	5	Maximum number of Areas used for one Part
MaxParts	UDINT	10	
MaxEtherCatMaster	UINT	10	
MaxTracks	UINT	20	
MaxGroups	UINT	10	
MaxEnvironments	UINT	1	
MaxSubTiles	UINT	3	

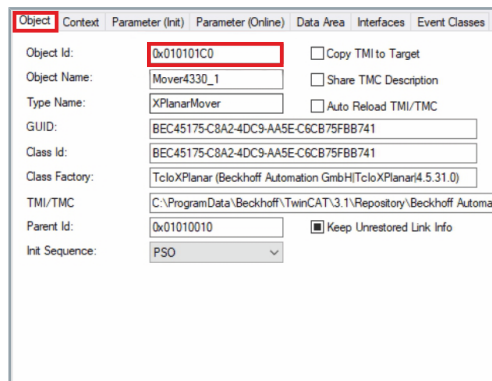
Die Parameterliste *Param\_TcloXPlanarEnvironment* muss an die Konfiguration angepasst werden. Jeder einzelne Parameter muss größer oder mindestens gleich der Konfiguration sein, um sicherzustellen, dass die Initialisierung von *FB\_TcloXPlanarEnvironment* erfolgreich abgeschlossen werden kann.

## 15.1.2.2 Initialisierung von Standalone-Funktionsblöcken

Wenn *FB\_TcIoXPlanarEnvironment* nicht verwendet werden soll, kann jeder untergeordnete Funktionsblock einzeln initialisiert werden. Dafür wird die OTCID des entsprechenden TcCOM-Objekts benötigt.



- ▶ *Solution Explorer* > *SYSTEM* > *TcCOM Objects* > *XPlanar* ausklappen
- ▶ Doppelklick auf einen **Mover4330**



- ▶ Im Projektfenster auf den Reiter **Object** klicken
- ▶ Wert bei *Object Id* ablesen

```

VAR
nState          : UDINT;
fbTcIoMoverStandalone : FB_TcIoXPlanarMover;
END_VAR

CASE nState OF

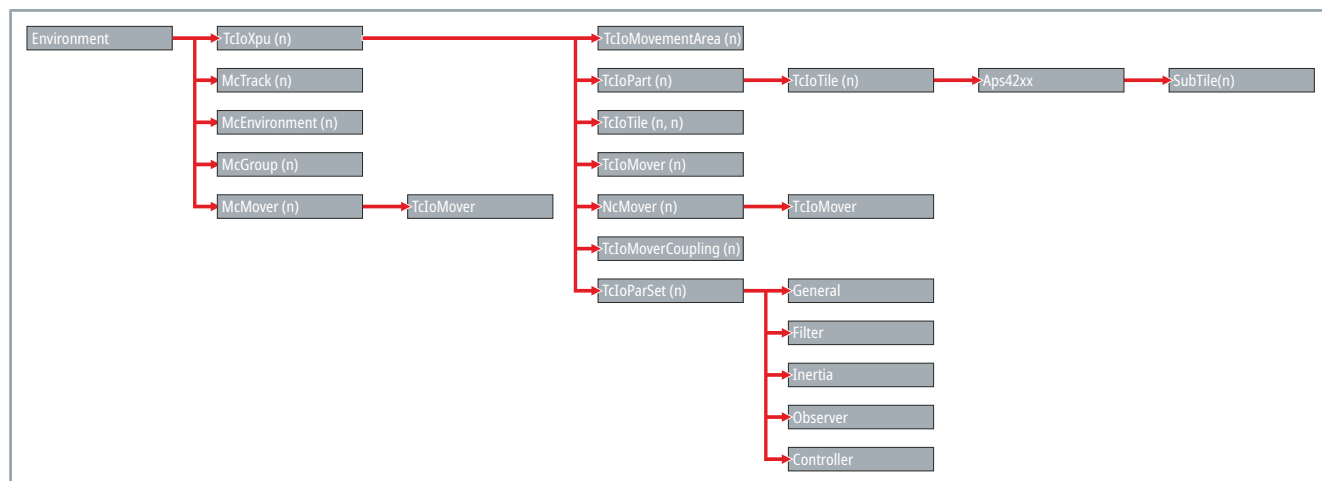
130: // Init stadalone fuction block
    IF fbTcIoMoverStandalone.Init(TRUE,16#010101C0, 1) THEN
        nState := nState + 1;
    END_IF

END_CASE
    
```

## 15.1.3 PLC-Zugriff

### 15.1.3.1 Call Chain

Mit der folgenden Call Chain haben Sie die Möglichkeit auf die wesentlichen Parameter der XPlanar Objekte zuzugreifen. Eine Übersicht zu den Parametern finden Sie im Kapitel „Parameter“, [Seite 107].



### 15.1.3.2 Zugriff auf Parameter in der SPS

In Anlehnung an die Call Chain, können Sie mithilfe der *subitem*-Methoden zum gewünschten Objekt und Parameter navigieren. Der Methodeneingang der *subitem*-Methoden kann als Index, fortlaufende Nummer von 1 bis X, oder als OTCID des Objekts interpretiert werden. Das folgende Beispiel zeigt, wie man auf das TcCOM-Objekt eines Movers zugreifen kann:

```

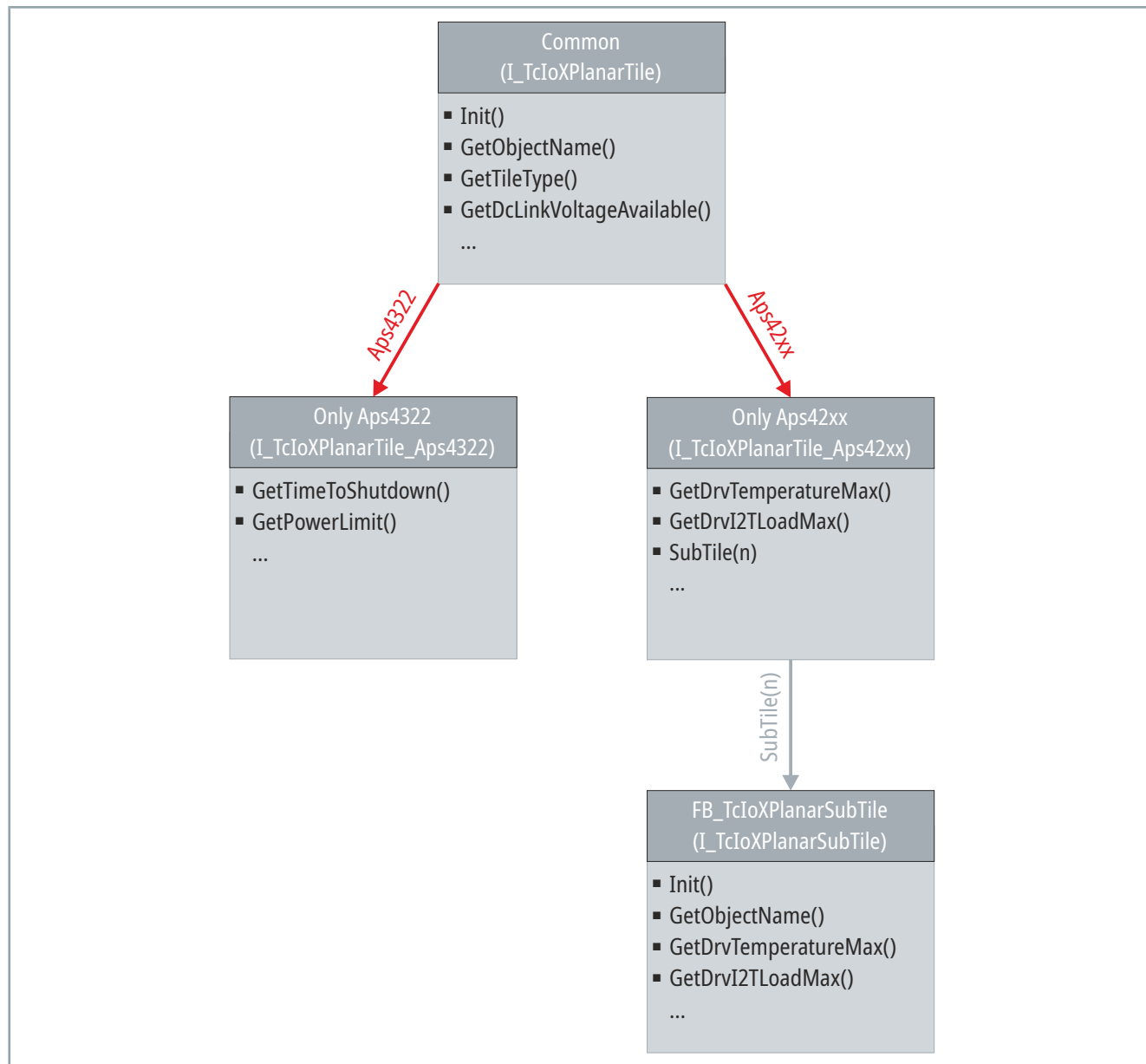
120: // Read object name of mover 1 by index
    sObjectNameIdx := ipXpu.TcIoMover(1).GetObjectName();
    nState := nState + 1;

121: // Read object name of mover 1 by OTCID
    sObjectNameOid := ipXpu.TcIoMover(16#010101C0).GetObjectName();
    nState := nState + 1;
  
```

Beide Aufrufe geben *Mover4330\_1* zurück, da sie auf dasselbe TcCom-Objekt zugreifen.

## Zusätzliche Hinweise für FB-TcIoXPlanarTile

Der unterschiedliche Aufbau der Kacheln AP4322 und APS42xx erfordert einen unterschiedlichen Zugriff auf die jeweiligen TcCOM-Parameter. Der Funktionsbaustein *FB\_TcIoXPlanarTile* ist in einen Abschnitt für APS4322 und in einen Abschnitt für APS42xx aufgeteilt, in denen die entsprechenden Schnittstellen implementiert sind.



Symbol	Erläuterung	Symbol	Erläuterung
↓	Zugriff über Eigenschaften	↓	Zugriff über Methoden

Ein wesentlicher Unterschied zwischen den Kacheln *APS4322* und *APS42xx* besteht darin, dass die Kachel *APS42xx* bis zu drei Unterkacheln besitzen kann: *FB\_TcIoXPlanarSubTile*.

## Zugriff auf TcIo-Parameter

Generell kann mit der SPS auf alle TcIo-Parameter am TcCOM-Objektserver zugegriffen werden. Das folgende Beispiel zeigt, wie man die *MoverParametrizationSet*-OTCID eines Movers lesen/schreiben kann. Der gezeigte SPS-Code stellt den Lesezugriff und Schreibzugriff auf diesen TcIo-Parameter dar.

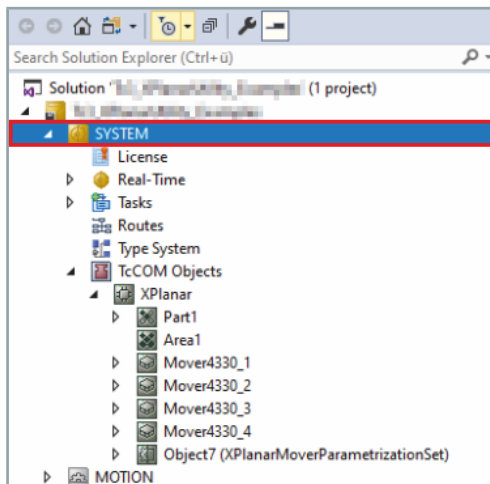
Dieses Prinzip funktioniert für alle Methoden und Parameter in gleicher Weise, entsprechend der Call Chain der *Tc3\_XPlanarUtility*.

```
10: // Read object id of the parametrization set of mover 1
    nParaSetOid := ipXpu.TcIoMover(1).GetParameterizationSetId();
    nState := nState + 1;

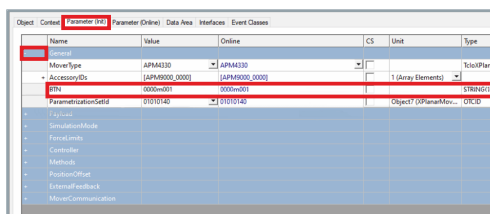
11: // Write object id of a parametrization set to mover 2
    IF ipXpu.TcIoMover(2).SetParameterizationSetId(nParaSetOid) THEN
        nState := nState + 1;
    END_IF
```

**Beispiel Mover Parameter Set**

Der Zugriff auf TcIo-Parameter wird beispielhaft am Auslesen des *MoverParametrizationSet*-OTCID eines Movers gezeigt. Das Aufrufen der Init-Parameter erfolgt für alle TcIo-Parameter auf die gleiche Weise.



- ▶ *Solution Explorer* ausklappen
- ▶ Doppelklick auf **SYSTEM**



- ▶ Auf den Reiter **Parameter (Init)** klicken
- ▶ *General* ausklappen
- ▶ Wert für *ParameterizationSetId* ablesen

## Zugriff auf Scope-Parameter

Die Verwendung der *Tc3\_XPLanarUtility* ermöglicht den Zugriff auf alle Scope-Parameter des XPlanars. Das Aufrufen erfolgt für alle Scope-Parameter auf die gleiche Weise.

```
VAR
    stOutputForce:MoverVector;
END_VAR

20: // Read output force of mover 1 from scope data
    stOutputForce := ipXpu.TcIoMover(1).ScopeAdr.GetOutputForce()^;
    nState := nState + 1;
```

### 15.1.3.3 Beispiele für häufig verwendete Methoden

Für den Zugriff auf verschiedene Parameter in der SPS benötigen Sie diese Aufrufe:

#### Start Mover Redetection (mover detection)

```
0: //Start mover redetection
    ipXpu.SetStartMoverRedetection();
    nState := nState + 1;
```

#### Get Mover Detection Status (mover detection)

```
VAR
    stMoverDetectionStatus:MoverDetectionStatus;
END_VAR

50: // Read mover detection status
    stMoverDetectionStatus := ipXpu.GetMoverDetectionStatus();
    IF stMoverDetectionStatus = MoverDetectionStatus.DetectionSuccessful THEN
        nState := nState + 1;
    END_IF
```

#### Get Mover Detection Infos (mover detection)

```
VAR
    stMoverDetectionInfo:ST_TcIoMoverDetectionInfos;
END_VAR

60: // Read mover detection infos
    stMoverDetectionInfo := ipXpu.GetMoverDetectionInfos();
    IF stMoverDetectionInfo.TotalDetected = stMoverDetectionInfo.TotalDetected
        AND stMoverDetectionInfo.APM4330Detected = stMoverDetectionInfo.APM4330Expected THEN
        nState := nState + 1;
    END_IF
```

#### Mover Detection Setup (mover detection)

```
VAR
    arMoverDetectionSetup:ARRAY[0..4] OF ST_TcIoMoverDetectionSetup;
END_VAR

2: // Adapt mover detection setup
    // Set number of APM4330
    arMoverDetectionSetup[0].MoverType := XPlanarMoverType.APM4330;
    arMoverDetectionSetup[0].ExpectedCount := 4;
    // Set number of APM4220
    arMoverDetectionSetup[1].MoverType := XPlanarMoverType.APM4220;
    arMoverDetectionSetup[1].ExpectedCount := 0;
    // Set number of APM4221
    arMoverDetectionSetup[2].MoverType := XPlanarMoverType.APM4221;
    arMoverDetectionSetup[2].ExpectedCount := 0;
    // Set number of APM4550
    arMoverDetectionSetup[3].MoverType := XPlanarMoverType.APM4550;
    arMoverDetectionSetup[3].ExpectedCount := 0;

    ipXpu.SetMoverDetectionSetup(4,ADR(arMoverDetectionSetup));
    nState := nState - 1;
```

#### Set CoordMoverCommunicationCmd (mover identification)

```
10: // Start coordinated mover communication command
    ipXpu.SetCoordMoverCommunicationCmd(CoordinatedMoverCommCommand.IdentifyAndAssignPossible);
    nState := nState + 1;
```

## Get CoordMoverCommunicationStatus (mover identification)

```

VAR
    stCoordMoverCommStatus:CoordinatedMoverCommStatus;
END_VAR

11: // Check for identification in process
    stCoordMoverCommStatus := ipXpu.GetCoordMoverCommStatus();
    IF stCoordMoverCommStatus = CoordinatedMoverCommStatus.CommunicationInProgress THEN
        nState := nState + 1;
    END_IF

12: // Check result of mover identification
    stCoordMoverCommStatus := ipXpu.GetCoordMoverCommStatus();
    IF stCoordMoverCommStatus = CoordinatedMoverCommStatus.CommunicationSuccessful THEN
        nState := 20;
    ELSIF stCoordMoverCommStatus = CoordinatedMoverCommStatus.CommunicationFailed THEN
        // Error: Identification failed
        nState := 9990;
    END_IF

```

## Get IdentifiedMoverBtn (mover identification)

```

72: // Check IdentifiedMoverBtn
    sBtn := ipXpu.TcIoMover(1).GetIdentifiedMoverBtn();
    nState := nState + 1;

```

## Reading out a single Id-Bumper (mover identification)

Der Code zeigt beispielhaft, wie die Bumper-ID *APM9001-0000-4xxx* eines einzelnen Movers ausgelesen werden kann:

```

70: // Set mover communication command - start reading id-bumper
    IF ipXpu.TcIoMover(1).SetMoverCommunicationCommand(MoverCommunicationCommand.GetMoverBtn) THEN
        nState := nState + 1;
    END_IF

71: // check status
    stMoverCommunicationStatus := ipXpu.TcIoMover(1).GetCommunicationStatus();
    IF stMoverCommunicationStatus = MoverCommunicationStatus.CommunicationSuccessful THEN
        nState := nState + 1;
    END_IF

72: // Check IdentifiedMoverBtn
    sBtn := ipXpu.TcIoMover(1).GetIdentifiedMoverBtn();
    nState := nState + 1;

```

## Switch Controllers on/off

```

80: // Switch controller on/off
    stTcIoControllersOn.x := TRUE;
    stTcIoControllersOn.y := TRUE;
    stTcIoControllersOn.z := FALSE;
    stTcIoControllersOn.a := TRUE;
    stTcIoControllersOn.b := TRUE;
    stTcIoControllersOn.c := TRUE;

    IF ipXpu.TcIoMover(1).SetControllersOn(stTcIoControllersOn) THEN
        nState := nState + 1;
    END_IF

```

## Leaving Movement Area

```

90: // Leave MovementArea
    stEnterOrLeaveMovementareaParameters_XYC.MovementAreaOID := 0; // Set to 0 to remove a mover
    stEnterOrLeaveMovementareaParameters_XYC.Positions.x := 0;
    stEnterOrLeaveMovementareaParameters_XYC.Positions.y := 0;
    stEnterOrLeaveMovementareaParameters_XYC.Positions.C := 0;

    IF ipXpu.TcIoMover(1).SetMovementArea_XYC(stEnterOrLeaveMovementareaParameters_XYC) THEN
        nState := nState + 1;
    END_IF

```

## Enter Movement Area

```
91: // Enter MovementArea
    stEnterOrLeaveMovementareaParameters_XYC.MovementAreaOID := 16#01010020; // Set to OTCID of Part-
Object
    stEnterOrLeaveMovementareaParameters_XYC.Positions.x := 120; // approximate x-position
    stEnterOrLeaveMovementareaParameters_XYC.Positions.y := 120; // approximate y-position
    stEnterOrLeaveMovementareaParameters_XYC.Positions.C := 0; // approximate c-position

    IF ipXpu.TcIoMover(1).SetMovementArea_XYC(stEnterOrLeaveMovementareaParameters_XYC) THEN
        nState := nState + 1;
    END_IF
```

## Get IsDcLinkVoltageAvailable (Aps4322/Aps42xx)

```
100: // Get "IsDcLinkVoltageAvailable"
    bDcLinkVoltageAvailable := ipXpu.TcIoTile(1,1).GetDcLinkAvailable();
    nState := nState + 1;
```

## Get TileType (Aps4322/Aps42xx)

```
110: // Get Tiletype
    stTileType := ipXpu.TcIoTile(1,1).GetTileType();
    nState := nState + 1;
```

## Get GetDrvTemperatureMax (only APS42xx)

```
110: // Get highest temperature of this tile - only available for Aps42xx tiles!
    stAps42xxDrvInfo := ipXpu.TcIoTile(1,1).Aps42xx.GetDrvTemperatureMax();
    stAps42xxDrvInfo.fValue; // Temperature in Deg
    stAps42xxDrvInfo.nSubTileIndex; // Index of the sub-tile with the highest temperature
    stAps42xxDrvInfo.eSensorLocation; // Sensor location on the sub-tile --> north, south, ...
    nState := nState + 1;
```

## Read CoE-Data „General“

```
140: // Read CoE-Data "General" from Aps4322/Aps42xx tiles
    sCoE_Btn := ipXpu.TcIoTile(1,1).CoE_General.GetCoeBtn();
    sCoE_DeviceName := ipXpu.TcIoTile(1,1).CoE_General.GetDeviceName();
    // ...
    // ...
    // Go on like this with all relevant CoE-data ...
    nState := nState + 1;
```

## Read CoE-Data “Diag Data”

```
150: // Read CoE-Data "Diag Data" from Aps4322 tiles
    fTileTemperatureNorth := ipXpu.TcIoTile(1,1).Aps4322.CoE_DiagData.GetTemperatureNorth();
    fTileTemperatureEast := ipXpu.TcIoTile(1,1).Aps4322.CoE_DiagData.GetTemperatureEast();
    fTileTemperatureSouth := ipXpu.TcIoTile(1,1).Aps4322.CoE_DiagData.GetTemperatureSouth();
    fTileTemperatureWest := ipXpu.TcIoTile(1,1).Aps4322.CoE_DiagData.GetTemperatureWest();
    fTileTemperatureCenter := ipXpu.TcIoTile(1,1).Aps4322.CoE_DiagData.GetTemperatureCenter();
    // ...
    // ...
    // Go on like this with all relevant CoE-data ...
    nState := nState + 1;
```

## 15.1.3.4 Veraltete Funktionen

## Auslesen der CoE-Parameter

Diese Funktion ist veraltet und sollte nicht verwendet werden.

**Aktualisierung eines CoE-Objekts**

Die *Tc3\_XPlanarUtility* koordiniert den Zugriff auf CoE-Daten, um Kommunikationsfehler zu vermeiden. Sie können immer nur die Daten eines CoE-Objekts aktualisieren. Parallelzugriffe auf Module innerhalb eines EtherCAT-Masters sind nicht zulässig. Diese Art von Funktionsblöcken und Methoden dürfen nicht in For-Schleifen oder ähnlichem verwendet werden.

```
30: // Update CoE Data - DONT USE IN FOR-LOOP !!!
    IF ipXpu.TcIoTile(1,1).Aps4322.CoE.DiagData.Update() THEN
        fTempl :=ipXpu.TcIoTile(1,1).Aps4322.CoE.DiagData.TemperatureCenter;
        nState := nState + 1;
    END_IF
```

## Zugriff auf alle Daten des CoE-Objekts

Diese Funktion ist veraltet und sollte nicht verwendet werden.

Mit der *Update*-Methode wird immer das gesamte CoE-Objekt ausgelesen. Wenn beispielsweise alle Temperaturen ausgelesen werden sollen, kann das gesamte Objekt mit der Eigenschaft *All* zur Verfügung gestellt werden.

```
31: // Update CoE Data - DO NOT USE IN FOR-LOOP !!!
    IF ipXpu.TcIoTile(1,1).Aps4322.CoE.DiagData.Update() THEN
        stCoeDriveDiagData := ipXpu.TcIoTile(1,1).Aps4322.CoE.DiagData.All;
        nState := nState + 1;
    END_IF
```

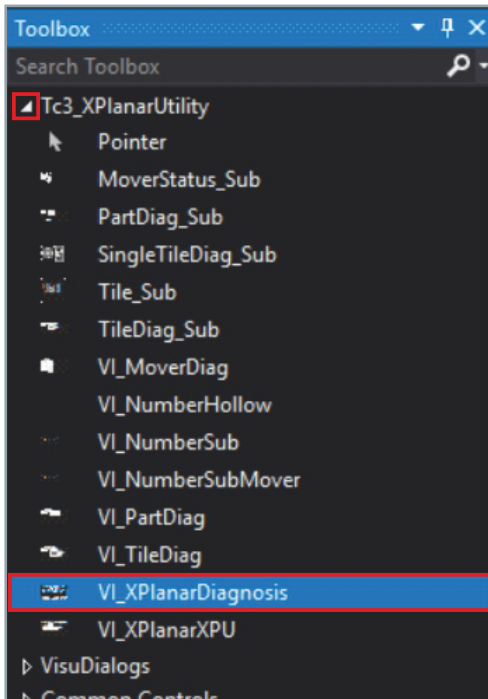
## 15.1.4 Visualisierung

### HINWEIS

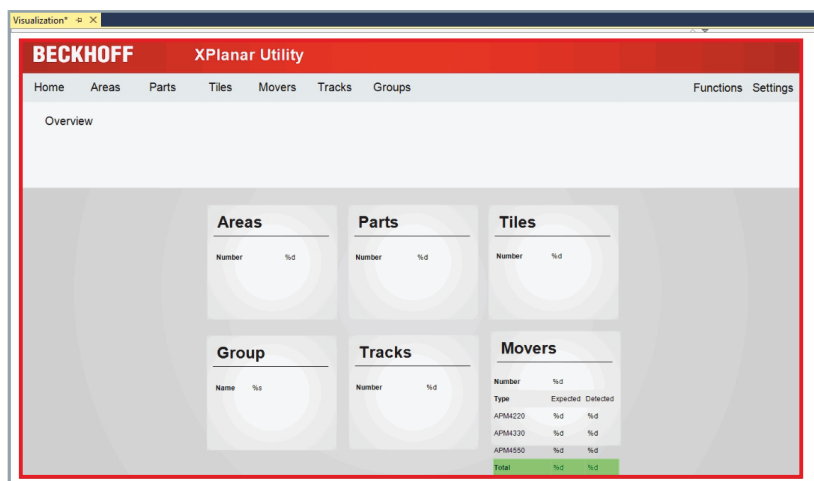
#### Visualisierung nur zur Unterstützung während des Programmierens

Die Visualisierung dient nur zur Unterstützung während des Programmierens eines XPlanar Systems und kann nicht zur Visualisierung in der Anlagenbedienoberfläche genutzt werden.

### 15.1.4.1 Visualisierung zu einem Projekt hinzufügen



- ▶ *Toolbox* > *Tc3\_XPlanarUtility* ausklappen
- ▶ Mit gedrückter Maustaste **VI\_XPlanar\_Diagnostics** in das Projektfenster *Visualization* ziehen



Die Visualisierung wird im Projektfenster *Visualization* dargestellt.

- ▶ Bei Bedarf die Visualisierung für bessere Bedienbarkeit vergrößern
- ▶ Bei Bedarf die Visualisierung im Projektfenster positionieren

## 15.1.5 Parameter

## 15.1.5.1 FB\_TcIoXPlanarEnvironment

## 15.1.5.1.1 Methode

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetXpuOids	–	BOOL	–	Aktualisiert die Objekt-ID-Liste und die Anzahl der Planar Groups, die über <i>P_XpuOids</i> oder <i>P_XpuCount</i> abgerufen werden kann.
GetEnvironmentOids	–	BOOL	–	Aktualisiert die Objekt-ID-Liste und die Anzahl der Planar Groups, die über <i>P_EnvironmentOids</i> oder <i>P_EnvironmentCount</i> abgerufen werden kann.
GetGroupOids	–	BOOL	–	Aktualisiert die Objekt-ID-Liste und die Anzahl der Planar Groups, die über die <i>P_GroupOids</i> oder <i>P_GroupCount</i> abgerufen werden kann.
GetTrackOids	–	BOOL	–	Aktualisiert die Objekt-ID-Liste und die Anzahl der Planar Tracks, die über die Eigenschaft <i>P_TrackOids</i> oder <i>P_TrackCount</i> abgerufen werden kann.
<b>Subitem</b>				
TcloXpu	–	I_TcloXPlanarProcessingUnit	UDINT	Wählt eine Xpu für den weiteren Betrieb durch Eingabe der Nummer oder OTCID dieser Xpu aus.
McEnvironment	–	I_McPlanarEnvironment	UDINT	Wählt eine planare Umgebung für den weiteren Betrieb durch Eingabe der Nummer oder OTCID dieser Umgebung aus.
McGroup	–	I_McPlanarGroup	UDINT	Wählt eine Planar Group für weitere Operationen durch Eingabe der Nummer oder OTCID dieser Group aus.
McTrack	–	I_McPlanarTrack	UDINT	Wählt einen Planar Track für weitere Operationen durch Eingabe der Nummer oder OTCID des Tracks aus.
McMover	–	I_McPlanarMover	UDINT	Wählt einen Planar Mover für weitere Operationen durch Eingabe der Nummer oder OTCID des Mover aus.

## 15.1.5.1.2 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_XpuCount	–	UDINT	–	Gibt die Anzahl der Xpus zurück.
P_XpuOids	–	REFERENCE TO ARRAY [1..Param_TcloXPlanarEnvironment.MaxProcessingUnits] OF OTCID	–	Gibt die Objekt ID-Liste der Xpus zurück.
P_EnvironmentCount	–	UDINT	–	Gibt die Anzahl der Planar Environments zurück.
P_EnvironmentOids	–	REFERENCE TO ARRAY [1..Param_TcloXPlanarEnvironment.MaxEnvironments] OF OTCID	–	Gibt die Objekt ID-Liste des Planar Environments zurück.
P_GroupCount	–	UDINT	–	Gibt die Anzahl der Planar Groups zurück.
P_GroupOids	–	REFERENCE TO ARRAY [1..Param_TcloXPlanarEnvironment.MaxGroups] OF OTCID	–	Gibt die Objekt ID-Liste des Planar Groups zurück.
P_TrackCount	–	UDINT	–	Gibt die Anzahl der Planar Tracks zurück.
P_TrackOids	–	REFERENCE TO ARRAY [1..Param_TcloXPlanarEnvironment.MaxTracks] OF OTCID	–	Gibt die Objekt ID-Liste der Planar Tracks zurück.
P_MoverCount	–	UDINT	–	Gibt die Anzahl der Planar Mover zurück.
P_MoverOids	–	REFERENCE TO ARRAY [1..Param_TcloXPlanarEnvironment.MaxMcPlanarMovers] OF OTCID	–	Gibt die Objekt ID-Liste des Planar Movers zurück.

## 15.1.5.2 FB\_TcIoXPlanarProcessingUnit - I\_TcIoXPlanarProcessingUnit

## 15.1.5.2.1 Methode - Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetOperationMode	General	XPlanarOperation-Mode	–	Gibt die ausgewählte Betriebsart wieder.
GetEtherCatMasterSyncTaskObjectId	General	OTCID	–	Gibt die ID der EtherCAT Master Syncro Aufgabe zurück.
GetDetectionMode	MoverDetection	DetectionMode	–	Gibt den Detection Modus zurück.
SetDetectionMode	MoverDetection	BOOL	DetectionMode	Stellt den Detection Modus für die nächste Mover Detection ein.
SetStartMoverRedetection	MoverDetection	BOOL	–	Startet die Mover-Detection.
SetMoverDetection-Setup	MoverDetection	BOOL	UNIT, PVOID	Stellt das Mover Detection Setup basierend auf der Anzahl und Adresse einer Reihe von Mover Detection Setups ein.
SetCoordMoverCommunicationCmd	MoverDetection	BOOL	Coordinated-MoverComm-Command	Startet die Mover-Detection basierend auf dem ausgewählten Kommunikationsbefehl zur Detection der Mover.

## 15.1.5.2.2 Methode - Parameter (Online)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetDriverVersion	ProcessingUnit	STRING	–	Gibt die XPlanar Treiber Version als STRING zurück.
GetTaskCount	ProcessingUnit	UDNIT	–	Gibt die Anzahl der konfigurierten Tasks zurück.
GetTaskOid	ProcessingUnit	BOOL	–	Aktualisiert die Objekt ID-Liste aller konfigurierten Tasks. Auf die Liste kann über die Eigenschaft P_TaskOids zugegriffen werden.
GetPartCount	Parts	UDNIT	–	Gibt die Anzahl der konfigurierten Parts zurück.
GetPartOid	Parts	BOOL	–	Aktualisiert die Objekt ID-Liste aller konfigurierten Tasks. Auf die Liste kann über die Eigenschaft P_PartOids zugegriffen werden.
GetTileCount	Tiles	UDNIT	–	Gibt die Anzahl der konfigurierten Tiles zurück.
GetTileOid	Tiles	BOOL	–	Aktualisiert die Objekt ID-Liste aller konfigurierten Tasks. Auf die Liste kann über die Eigenschaft P_TileOids zugegriffen werden.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetMovementArea-Count	MovementAreas	UDNIT	–	Gibt die Anzahl der konfigurierten Movement Areas zurück.
GetMovementArea-Oid	MovementAreas	BOOL	–	Aktualisiert die Objekt ID-Liste aller konfigurierten Tasks. Auf die Liste kann über die Eigenschaft P_MovermentAreaOids zugegriffen werden.
GetMoverCount	Mover	UDNIT	–	Gibt die Anzahl aller Mover zurück.
GetMoverOids	Mover	BOOL	–	Aktualisiert die Objekt ID aller Mover.
GetAmbiguouslyDetectedMoverCount	Mover	UDNIT	–	Gibt die Anzahl der nicht eindeutig erkannten Mover zurück.
GetAmbiguouslyDetectedMover	Mover	REFERENCE TO DetectedMoverInfo	–	Gibt die Informationen der nicht eindeutig erkannten Mover zurück.
GetMoverDetection-Infos	Mover	REFERENCE TO ST_TcloMoverDetectionInfos	–	Gibt die Informationen der erkannten Mover zurück.
GetMoverDetection-Status	Mover	REFERENCE TO MoverDetectionStatus	–	Gibt den aktuellen Status der Mover Detection zurück.
GetCoordMover-CommStatus	Mover	REFERENCE TO CoordinatedMoverCommStatus	–	Gibt den aktuellen Status der Mover Detection zurück.
GetCoordMoverCommunicationErrorId	Mover	DWORD	–	Gibt die Fehler ID der letzten fehlerhaften Mover Detection zurück.
GetMoverCoupling-Count	MoverCouplings	UDINT	–	Gibt die Anzahl aller Moverkopplungen zurück. (Starr gekoppelte Mover)
GetMoverCoupling-Oids	MoverCouplings	BOOL	–	Aktualisiert die Objekt ID aller Moverkopplungen. (Starr gekoppelte Mover)
GetProcessingLines-Count	ProcessingLines	UINT	–	Gibt die Anzahl der Processing Lines zurück.
GetProcessingLine-Oids	ProcessingLines	BOOL	–	Aktualisiert die Objekt ID-Liste aller Processing Lines. Auf die Liste kann über die Eigenschaft P_ProcessingLineOids zugegriffen werden.
GetParametrizationSetCount	ParametrizationSets	UINT	–	Gibt die Anzahl der Parametersets zurück.
GetParametrizationSetOids	ParametrizationSets	BOOL	–	Aktualisiert die Objekt ID-Liste aller Parametersets. Auf die Liste kann über die Eigenschaft P_ParametrizationSetOids zugegriffen werden.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	Initializing	BOOL	BOOL	Initialisiert zu Xpu.
InitNc3	Initializing	BOOL	BOOL	Initialisiert die Mc_Mover Objekte. Diese Methode wird einmal in der Init()-Methode aufgerufen. Die Methode sollte nur separat wiederholt werden, wenn sich die Reihenfolge der Mover geändert hat. Die Verlinkung zwischen Tclo_XPlanarMover und Mc_XPlanarMover kann nach einer Mover Identifikation geändert werden.
<b>Subitems</b>				
TcloMover	–	I_TcloXPlanarMover	UDINT	Wählt einen Tclo-Mover für den weiteren Betrieb durch Eingabe der Nummer oder OTCID des ausgewählten Tclo-Movers aus.
TcloMoverCoupling	–	I_TcloXPlanarMover-Coupling	UDINT	Wählt eine Tclo-Moverkopplung für den weiteren Betrieb durch Eingabe der Nummer oder OTCID der ausgewählten Tclo-Moverkopplung aus.
McMover	–	I_MCPlanarMover	UDINT	Wählt einen Mc Mover für den weiteren Betrieb durch Eingabe der Nummer oder OTCID des Mc Mover aus.
TcloMovementArea	–	I_TcloXPlanarMovementArea	UDINT	Wählt eine MovementArea für den weiteren Betrieb durch Eingabe der Nummer oder OTCID der ausgewählten MovementArea aus.
TcloPart	–	I_TcloXPlanarPart	UDINT	Wählt einen Part für den weiteren Betrieb durch Eingabe der Nummer oder OTCID des ausgewählten Parts aus.
TcloParaSet	–	I_TcloMoverParameterizationSet	UDINT	Wählt ein Parameter Set für den weiteren Betrieb durch Eingabe der Nummer oder OTCID des ausgewählten Parameter Sets aus.
TcloTile	–	I_TcloXPlanarTile	UDINT, UDINT	Wählt ein Tile für den weiteren Betrieb durch Eingabe der Nummer oder OTCID des ausgewählten Tiles aus.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
CoETile	–	I_Aps4322Drive	UDINT, UDINT	Wählt ein Tile für den weiteren Betrieb (Zugriff auf den CoE-Parameter des APS4322) durch Eingabe der Nummern des Part und des Tiles oder OTCID des ausgewählten Tiles aus.

## 15.1.5.2.3 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_MoverCouplingOids	–	REFERENCE TO ARRAY [0..Param_TcloXPlanarEnvironment.MaxMoversPerXpu] of OTCID	–	Gibt die Objekt ID aller Moverkopplungen zurück (Starrgekoppelte Mover).
P_MoverOids	–	REFERENCE TO ARRAY [0..Param_TcloXPlanarEnvironment.MaxMoversPerXpu] of OTCID	–	Gibt die Objekt ID aller Mover zurück.
P_PartOids	–	REFERENCE TO ARRAY [0..Param_TcloXPlanarEnvironment.MaxParts] of OTCID	–	Gibt die Objekt ID aller Parts zurück.
P_TaskOids	–	REFERENCE TO ARRAY [1..12] of OTCID	–	Gibt die Objekt ID aller Tasks zurück.
P_TileOids	–	REFERENCE TO ARRAY [0..Param_TcloXPlanarEnvironment.MaxTilesPerPart] of OTCID	–	Gibt die Objekt ID aller Tiles pro Part zurück.
P_MovementAreaOids	–	REFERENCE TO ARRAY [0..Param_TcloXPlanarEnvironment.MaxAreas]	–	Gibt die Objekt ID aller MovementAreas zurück.
P_ParametrizationSetOids	–	REFERENCE TO ARRAY [0..Param_TcloXPlanarEnvironment.MaxMoverParametrizationSets]	–	Gibt die Objekt ID aller Parameter Sets zurück.

## 15.1.5.3 FB\_TcIoXPlanarMovementArea – I\_TcIoXPlanarMovementArea

## 15.1.5.3.1 Methode - Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(31)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetIsIncludedInMoverDetection	–	BOOL	–	Gibt die Information zurück, dass diese Area in die Mover Detection einbezogen ist.
GetPartConfigurations	–	PartConfiguration	UINT	Gibt die Konfiguration des Parts basierend auf einem Index zurück.

## 15.1.5.3.2 Methode – Parameter (Online)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetDetectedMoverCount	–	UDINT	–	Gibt die Anzahl erkannter Mover in dieser Area zurück.
GetDetectedMoverInfo	–	ST_TcIoDetectedMoverInfo	Int	Gibt die Information über einen Mover in dieser Area basierend auf einem Index zurück.
GetMoverCount	–	UDINT	–	Gibt die aktuelle Anzahl der Mover in dieser Area zurück.
GetPartCount	–	UDINT	–	Gibt die Anzahl der Parts in dieser Area zurück.

## 15.1.5.4 FB\_TcIoXPlanarMover – I\_TcIoXPlanarMover

### 15.1.5.4.1 Methode - Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCOM-Objekts zurück.
GetIpMcMover	–	I_McPlanarMover	–	Gibt die Schnittstelle des verknüpften NC-Mover-Objekts zurück.
GetAccessoryId	General	MoverAccessoryID	–	Gibt die konfigurierte Accessoire-ID dieses Movers zurück.
GetAccessoryId-Count	General	UDINT	–	Gibt die Anzahl der konfigurierten Accessoire-IDs zurück, die für diesen Mover konfiguriert sind.
GetBTN	General	STRING(13)	–	Gibt die BTN dieses Movers zurück.
GetMoverType	General	XPlanarMoverType	–	Gibt den MoverTyp zurück.
GetMoverParametrizationSetId	General	OTCID	–	Gibt das Parameter Set zurück, das für diesen Mover konfiguriert ist.
GetTaskOid	General	OTCID	–	Gibt die Task Objekt ID dieses Movers zurück.
SetMoverParametrizationSetOid	General	BOOL	OTCID	Legt eine neue Parameter Set Objekt ID für diesen Mover fest.
SetTaskOid	General	BOOL	OTCID	Legt eine neue Task Objekt ID für diesen Mover fest.
GetPayloadInertia	Payload	ST_TcloPayload	–	Gibt die Nutzlastträgheit dieses Movers zurück.
GetPayloadProductsOfInertia	Payload	ST_TcloPayloadProductsOfInertia	–	Gibt die Trägheitsprodukte der Nutzlast dieses Movers zurück.
SetPayloadInertia	Payload	BOOL	ST_TcloPayload	Legt die Nutzlastträgheit dieses Movers fest.
SetPayloadProductsOfInertia	Payload	BOOL	ST_TcloPayloadProductsOfInertia	Legt die Trägheitsprodukte der Nutzlast dieses Movers fest.
GetSimulationModeStartPosition	SimulationMode	REFERENCE TO SimulationModeStartPosition	–	Gibt die Startposition des Movers im Simulationsmodus zurück.
GetForceLimitMax	ForceLimits	REFERENCE TO Force6D	–	Gibt das maximale Kraftlimit dieses Movers zurück.
GetForceLimitMin	ForceLimits	REFERENCE TO Force6D	–	Gibt das minimale Kraftlimit dieses Movers zurück.
SetForceLimitMax	ForceLimits	BOOL	Force6D	Legt das maximale Kraftlimit dieses Movers fest.
SetForceLimitMin	ForceLimits	BOOL	Force6D	Legt das minimale Kraftlimit dieses Movers fest.
GetControllersOn	Controller	ST_TcloControllersOn	–	Gibt den Status des Controllers dieses Movers zurück.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
SetControllersOn	Controller	BOOL	ST_TcloCon- trollersOn	Legt den Status des Con- trollers für diesen Mover fest.
SetMovementArea	Methods	BOOL	EnterOrLeave- MovementArea- Parameters	Legt die MovementArea Parameter dieses Movers fest.
SetMoverCommuni- cationCommand	Communication	BOOL	MoverCommuni- cationCom- mand	Legt den Mover-Kommuni- kationsbefehl für diesen Mover fest.
GetPositionOffset	Position Offset	MoverVector	–	Gibt das aktuelle Positi- onoffset dieses Movers zurück.
SetPositionOffset	Position Offset	BOOL	MoverVector	Legt ein Positionsoffset für diesen Mover fest.

## 15.1.5.4.2 Methode - Parameter (Online)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetActualPositions	–	MoverPositionInfo	–	Gibt die Positionsinformati- on dieses Movers zurück.
GetNcMoverOid	–	OTCID	–	Gibt die Objekt ID des ver- linkten NC-Mover Objekts zurück.
GetNcPartOid	–	OTCID	–	Gibt die Objekt ID der NC- Part Objekt ID zurück.
GetBumperHwVersi- on	Communication	STRING(13)	–	Gibt die Hardwareversion des montierten ID-Bum- pers zurück. Die ID-Kom- munikation muss zuerst ausgelöst werden.
GetCommErrorId	Communication	DWORD	–	Gibt die ID des Fehlers der zuletzt ausgelösten ID- Kommunikation zurück.
GetSignalToNoise- Ration	Communication	LREAL	–	Gibt das Signal-Rausch- Verhältnis der zuletzt aus- gelösten ID-Kommunikati- on zurück.
GetCommunication- Status	Communication	MoverCommuni- cationStatus	–	Gibt den Kommunikations- status des Movers zurück.
GetCurrentCommuni- cationAttempt	Communication	UDINT	–	Gibt die Anzahl der Kom- munikationsversuche der zuletzt ausgelösten ID- Kommunikation zurück.
GetFileTransferPro- gress	Communication	UDINT	–	Gibt den Fortschritt der ak- tuellen Dateiübertragung zurück - nur für Firmware- Updates des Movers.
GetFirmwareVersion	Communication	MoverFirmwareVersi- on	–	Gibt die Firmware-Version des montierten Bumpers zurück. Die ID-Kommuni- kation muss zuerst ausge- löst werden.
GetIdentifiedBum- perBtn	Communication	STRING(13)	–	Gibt die zuletzt erkannte BTN des montierten Bum- pers zurück. Die ID-Kom- munikation muss zuerst ausgelöst werden.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetIdentifiedMoverBtn	Communication	STRING(13)	–	Gibt die zuletzt erkannte BTN des Movers zurück. Die ID-Kommunikation muss zuerst ausgelöst werden.

## 15.1.5.4.3 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
ScopeAdr	–	I_XPlanarMoverScopeAdr	–	Gibt den Zugriff auf die Scope-Schnittstelle dieses Movers zurück.

## 15.1.5.4.4 I\_TcIoXPlanarMoverScopeAdr

### 15.1.5.4.4.1 Methode

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetActualPosition	–	POINTER TO Mover-Vector	–	Gibt die aktuelle Position aus den Scope-Daten zurück.
GetActualVelocity	–	POINTER TO Mover-Vector	–	Gibt die aktuelle Geschwindigkeit aus den Scope-Daten zurück.
GetActualAcceleration	–	POINTER TO Mover-Vector	–	Gibt die aktuelle Beschleunigung aus den Scope-Daten zurück.
GetInternalPositionSetpoints	–	POINTER TO Mover-Vector	–	Gibt die interpolierten Positionssollwerte aus den Scope-Daten zurück.
GetInternalVelocitySetpoints	–	POINTER TO Mover-Vector	–	Gibt die interpolierten Geschwindigkeitssollwerte aus den Scope-Daten zurück.
GetInternalAccelerationSetpoints	–	POINTER TO Mover-Vector	–	Gibt die interpolierten Beschleunigungssollwerte aus den Scope-Daten zurück.
GetExternalPositionSetpoints	–	POINTER TO Mover-Vector	–	Gibt die externen Positionssollwerte aus den Scope-Daten zurück.
GetExternalVelocitySetpoints	–	POINTER TO Mover-Vector	–	Gibt die externen Geschwindigkeitssollwerte aus den Scope-Daten zurück.
GetExternalAccelerationSetpoints	–	POINTER TO Mover-Vector	–	Gibt die externen Beschleunigungssollwerte aus den Scope-Daten zurück.
GetForceLimit	–	POINTER TO Force6D	–	Gibt das aktuelle Kraftlimit aus den Scope-Daten zurück.
GetForceLimitAdaptionFactor	–	POINTER TO LREAL	–	Gibt den aktuellen Anpassungsfaktor für das Kraftlimit aus den Scope-Daten zurück.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetLimitedOutputForces	–	POINTER TO Mover-Vector	–	Gibt die begrenzte Ausgangskraft aus den Scope-Daten zurück.
GetMovementarea- OID	–	POINTER TO OTCID	–	Gibt die MovementArea Objekt ID aus den Scope-Daten zurück.
GetNcDcTimeStamp	–	POINTER TO ULINT	–	Gibt den NC-DC-Zeitstempel aus den Scope-Daten zurück.
GetOutputForces	–	POINTER TO Mover-Vector	–	Gibt die Ausgangskraft aus den Scope-Daten zurück.
GetPartOID	–	POINTER TO OTCID	–	Gibt die Objekt ID des Parts aus den Scope-Daten zurück.
GetPositionError	–	POINTER TO Mover-Vector	–	Gibt den Positionsfehler aus den Scope-Daten zurück.
GetPositionOnCurrentPart	–	POINTER TO Position2D_LREAL	–	Gibt den Part basierend auf der XY-Position der Scope-Daten zurück.

## 15.1.5.5 FB\_TcIoXPlanarMoverCoupling – I\_TcIoXPlanarMoverCoupling

### 15.1.5.5.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetMoverParametrizationSetId	General	OTCID	–	Gibt das Parameter Set zurück, das für diesen Mover konfiguriert ist.
SetMoverParametrizationSetOid	General	BOOL	OTCID	Legt eine neue Parameter Set Objekt ID für diesen Mover fest.
GetPayloadInertia	Payload	ST_TcloPayload	–	Gibt die Nutzlastträgheit dieses Movers zurück.
GetPayloadProductsOfInertia	Payload	ST_TcloPayloadProductsOfInertia	–	Gibt die Trägheitsprodukte der Nutzlast dieses Movers zurück.
SetPayloadInertia	Payload	BOOL	ST_TcloPayload	Legt die Nutzlastträgheit dieses Movers fest.
SetPayloadProductsOfInertia	Payload	BOOL	ST_TcloPayloadProductsOfInertia	Legt die Trägheitsprodukte der Nutzlast dieses Movers fest.
GetForceLimitMax	Force Limits	REFERENCE TO Force6D	–	Gibt das maximale Kraftlimit dieses Movers zurück.
GetForceLimitMin	Force Limits	REFERENCE TO Force6D	–	Gibt das minimale Kraftlimit dieses Movers zurück.
SetForceLimitMax	Force Limits	BOOL	Force6D	Legt das maximale Kraftlimit dieses Movers fest.
SetForceLimitMin	Force Limits	BOOL	Force6D	Legt das minimale Kraftlimit dieses Movers fest.
GetControllersOn	Controller	ST_TcloControllersOn	–	Gibt den Status des Controllers dieses Movers zurück.
SetControllersOn	Controller	BOOL	ST_TcloControllersOn	Legt den Status des Controllers für diesen Mover fest.
GetPositionOffset	PositionOffset	MoverVector	–	Gibt das aktuelle Positionsoffset dieses Movers zurück.
SetPositionOffset	PositionOffset	BOOL	MoverVector	Legt ein Positionsoffset für diesen Mover fest.

### 15.1.5.5.2 Methode – Parameter (Online)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetActualPositions	–	MoverPositionInfo	–	Gibt die Positionsinformation dieses Movers zurück.
GetNcMoverOid	–	OTCID	–	Gibt die Objekt ID des verlinkten NC-Mover Objekts zurück.
GetSubMoverCount	–	UDINT	–	Gibt die Anzahl der Sub-Mover zurück.
GetSubMoverOids	–	BOOL	–	Liest alle Sub-Mover Objekt IDs.

## 15.1.5.5.3 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_SubMoverOids	–	REFERENCE TO Array [0.. Param_TcIoXPlanarEnvironment.MaxMoversPerXpu] OF OTCID	–	Gibt die Object ID aller Sub-Mover zurück.
ScopeAdr	–	I_XPlanarMoverCouplingScopeAdr	–	Gibt den Zugriff auf die Scope-Schnittstelle dieser Moverkopplung zurück.

## 15.1.5.5.4 I\_TcIoXPlanarMoverCouplingScopeAdr

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetActualPosition	–	POINTER TO Mover-Vector	–	Gibt die aktuelle Position aus den Scope-Daten zurück.
GetActualVelocity	–	POINTER TO Mover-Vector	–	Gibt die aktuelle Geschwindigkeit aus den Scope-Daten zurück.
GetActualAcceleration	–	POINTER TO Mover-Vector	–	Gibt die aktuelle Beschleunigung aus den Scope-Daten zurück.
GetInternalPositionSetpoints	–	POINTER TO Mover-Vector	–	Gibt die interpolierten Sollwerte aus den Scope-Daten zurück.
GetInternalVelocitySetpoints	–	POINTER TO Mover-Vector	–	Gibt die interpolierten Geschwindigkeitssollwerte aus den Scope-Daten zurück.
GetInternalAccelerationSetpoints	–	POINTER TO Mover-Vector	–	Gibt die interpolierten Beschleunigungssollwerte aus den Scope-Daten zurück.
GetExternalPositionSetpoints	–	POINTER TO Mover-Vector	–	Gibt die externen Positionssollwerte aus den Scope-Daten zurück.
GetExternalVelocitySetpoints	–	POINTER TO Mover-Vector	–	Gibt die externen Geschwindigkeitssollwerte aus den Scope-Daten zurück.
GetExternalAccelerationSetpoints	–	POINTER TO Mover-Vector	–	Gibt die externen Beschleunigungssollwerte aus den Scope-Daten zurück.
GetForceLimit	–	POINTER TO Force6D	–	Gibt das aktuelle Kraftlimit aus den Scope-Daten zurück.
GetForceLimitAdaptionFactor	–	POINTER TO LREAL	–	Gibt den aktuellen Anpassungsfaktor für das Kraftlimit aus den Scope-Daten zurück.
GetLimitedOutputForces	–	POINTER TO Mover-Vector	–	Gibt die begrenzte Ausgangskraft aus den Scope-Daten zurück.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetMovementarea- OID	–	POINTER TO OTCID	–	Gibt die MovementArea Objekt ID aus den Scope- Daten zurück.
GetNcDcTimeStamp	–	POINTER TO ULINT	–	Gibt den NC DC-Zeitstempel aus den Scope-Daten zurück
GetOutputForces	–	POINTER TO Mover- Vector	–	Gibt die Ausgangskraft aus den Scope-Daten zurück.
GetPartOID	–	POINTER TO OTCID	–	Gibt die Part Objekt ID aus den Scope-Daten zurück.
GetPositionError	–	POINTER TO Mover- Vector	–	Gibt den Positionsfehler aus den Scope-Daten zu- rück.
GetPositionOnCur- rentPart	–	POINTER TO Positi- on2D_LREAL	–	Gibt den Part basierend auf der XY-Position der Scope-Daten zurück.

## 15.1.5.6 FB\_TcIoXPlanarPart – I\_TcIoXPlanarPart

## 15.1.5.6.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(31)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetActiveIndex	–	UINT	–	Gibt den aktiven Index dieses Parts zurück.
GetCoeControlWord	–	UINT	–	Gibt das Coe-Steuerwort des Parts zurück.
GetPositions	–	PartConfiguration	–	Gibt die auf dem Part basierende XY-Position zurück.
SetStartMoverRedetection	–	BOOL	–	Löst die Mover Erkennung für diesen Part aus.

## 15.1.5.6.2 Methode – Parameter (Online)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetCoeStatusWord	–	UINT	–	Gibt das CoE-Statuswort dieses Parts zurück.
GetMoverDetection-Infos	–	REFERENCE TO PartMoverDetection-Infos	–	Gibt die Informationen zur teilbasierten Mover Detection zurück, basierend auf dem Part.
GetMoverDetectionStatus	–	MoverDetectionStatus	–	Gibt den Status der Mover Detection zurück, basierend auf dem Part.
GetTileCount	–	UDINT	–	Gibt die Anzahl der Tiles des Parts zurück.
GetTileObjectIds	–	OTCID	UINT	Gibt die Objekt ID eines bestimmten Tiles des Parts zurück. Verwenden Sie die Eingabe als Index der angeforderten Objekt ID.
<b>Subitems</b>				
TileCoe	–	I_Aps4322Drive	UINT	Gibt die Coe-Schnittstelle des Tiles unter Verwendung des Eingabeparameters als Index zurück.
TileTclo	–	I_TcloXPlanarTile	UINT	Gibt die Schnittstelle des Tiles unter Verwendung des Eingabeparameters als Index zurück.

## 15.1.5.6.3 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_TileOids	–	REFERENCE TO Array [0...Param_TcIoXPlanarEnvironment.MaxTilesPerPart] OF OTCID	–	Gibt die Objekt ID aller Tiles des Parts zurück.

## 15.1.5.7 FB\_TcIoXPlanarTile – I\_TcIoXPlanarTile

### 15.1.5.7.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(31)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetTaskOid	–	OTCID	–	Gibt die Objekt ID der Task für dieses Tile zurück.
GetTilePositionOnPart	–	ST_TcIoTilePositionOnPart	–	Gibt die Position des Tiles zurück, basierend auf dem Part.

### 15.1.5.7.2 Methode – Parameter (Online)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetActDcLinkVoltage	–	UINT	–	Gibt die aktuelle Zwischenkreisspannung des Tiles zurück.
GetPowerSupplyCurrent	–	INT	–	Gibt die Stromstärke der Stromversorgung des Tiles zurück.
GetCoeTimeToShutdown	–	UDINT	–	Gibt den Zeit-bis-zum-Ab-schalten-Parameter des Tiles zurück.
GetDriveOid	–	OTCID	–	Gibt die Objekt ID des Antriebs des Tiles zurück.
GetDriveStatusWord	–	UINT	–	Gibt das Statuswort des Antriebs des Tiles zurück.
GetFeedbackOid	–	OTCID	–	Gibt die Objekt ID des Feedbacks des Tiles zurück.
GetNetId	–	AMSNETID	–	Gibt die AMSNETID des Tiles zurück.
GetObservedTilePower	–	LREAL	–	Gibt die beobachtete Leistung des Tiles zurück.
GetPowerLimit	–	LREAL	–	Gibt die Leistungsgrenze des Tiles zurück.
GetPowerOverLimitFactor	–	LREAL	–	Gibt den Power-over-Limit-Faktor des Tiles zurück.
GetSlaveAdr	–	UINT	–	Gibt die Slave-Adresse des Tiles zurück.
GetTileType	–	TileTyp	–	Gibt den Type des Tiles zurück.

## 15.1.5.8 FB\_TcIoXPlanarMoverParametrizationSet – I\_TcIoXPlanarMoverParametrizationSet

## 15.1.5.8.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetControllerParamOid	–	OTCID	–	Gibt die Objekt ID des Steuerungsparameters des Mover Parameter Sets zurück.
GetFilterParamOid	–	OTCID	–	Gibt die Objekt ID der Filterparameter des Mover Parameter Sets zurück.
GetGeneralParamOid	–	OTCID	–	Gibt die Objekt ID der allgemeinen Parametrisierung des Mover Parameter Sets zurück.
GetInertiaParamOid	–	OTCID	–	Gibt die Objekt ID der Trägheitsparametrisierung des Mover Parameter Sets zurück.
GetObserverParamOid	–	OTCID	–	Gibt die Objekt ID der Beobachtungsparameter des Mover Parameter Sets zurück.

## 15.1.5.8.2 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Controller	–	I_TcIoControllerParametrization	–	Gibt die Schnittstelle der Steuerungsparameter zurück.
Filter	–	I_TcIoFilterParametrization	–	Gibt die Schnittstelle der Filterparameter zurück.
General	–	I_TcIoGeneralParametrization	–	Gibt die Schnittstelle der allgemeinen Parameter zurück.
Inertia	–	I_TcIoInertiaParametrization	–	Gibt die Schnittstelle der Trägheitsparametrisierung zurück.
Observer	–	I_TcIoObserverParametrization	–	Gibt die Schnittstelle der Beobachtungsparameter zurück.

## 15.1.5.9 FB\_TcIoControllerParametrization – I-TcIoControllerParametrization

## 15.1.5.9.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetAAxisFeedForwardParameters	–	REFERENCE TO RotationalFeedForwardParameters	–	Gibt die Vorsteuerparameter der A-Achse des Parameter Sets zurück.
GetBAxisFeedForwardParameters	–	REFERENCE TO RotationalFeedForwardParameters	–	Gibt die Vorsteuerparameter der B-Achse des Parameter Sets zurück.
GetCAxisFeedForwardParameters	–	REFERENCE TO RotationalFeedForwardParameters	–	Gibt die Vorsteuerparameter der C-Achse des Parameter Sets zurück.
SetAAxisFeedForwardParameters	–	BOOL	REFERENCE TO RotationalFeedForwardParameters	Legt die Vorsteuerparameter des A-Achse des Parameter Sets fest.
SetBAxisFeedForwardParameters	–	BOOL	REFERENCE TO RotationalFeedForwardParameters	Legt die Vorsteuerparameter des B-Achse des Parameter Sets fest.
SetCAxisFeedForwardParameters	–	BOOL	REFERENCE TO RotationalFeedForwardParameters	Legt die Vorsteuerparameter des C-Achse des Parameter Sets fest.
GetAAxisPIDT1Parameters	–	REFERENCE TO RotationalAxisPIDT1Parameters	–	Gibt die PIDT1 Parameter der A-Achse des Parameter Sets zurück.
GetBAxisPIDT1Parameters	–	REFERENCE TO RotationalAxisPIDT1Parameters	–	Gibt die PIDT1 Parameter der B-Achse des Parameter Sets zurück.
GetCAxisPIDT1Parameters	–	REFERENCE TO RotationalAxisPIDT1Parameters	–	Gibt die PIDT1 Parameter der C-Achse des Parameter Sets zurück.
SetAAxisPIDT1Parameters	–	BOOL	REFERENCE TO RotationalAxisPIDT1Parameters	Legt die PIDT1 Parameter der A-Achse des Parameter Sets fest.
SetBAxisPIDT1Parameters	–	BOOL	REFERENCE TO RotationalAxisPIDT1Parameters	Legt die PIDT1 Parameter der B-Achse des Parameter Sets fest.
SetCAxisPIDT1Parameters	–	BOOL	REFERENCE TO RotationalAxisPIDT1Parameters	Legt die PIDT1 Parameter der C-Achse des Parameter Sets fest.
GetXAxisFeedForwardParameters	–	REFERENCE TO LinearAxisFeedForwardParameters	–	Gibt die Vorsteuerparameter der X-Achse des Parameter Sets zurück.
GetYAxisFeedForwardParameters	–	REFERENCE TO LinearAxisFeedForwardParameters	–	Gibt die Vorsteuerparameter der Y-Achse des Parameter Sets zurück.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetZAxisFeedForwardParameters	–	REFERENCE TO LinearAxisFeedForwardParameters	–	Gibt die Vorsteuerparameter der Z-Achse des Parameter Sets zurück.
SetXAxisFeedForwardParameters	–	BOOL	REFERENCE TO LinearAxisFeedForwardParameters	Legt die Vorsteuerparameter der X-Achse des Parameter Sets fest.
SetYAxisFeedForwardParameters	–	BOOL	REFERENCE TO LinearAxisFeedForwardParameters	Legt die Vorsteuerparameter der Y-Achse des Parameter Sets fest.
SetZAxisFeedForwardParameters	–	BOOL	REFERENCE TO LinearAxisFeedForwardParameters	Legt die Vorsteuerparameter der Z-Achse des Parameter Sets fest.
SetXAxisPIDT1Parameters	–	BOOL	REFERENCE TO LinearAxisPIDT1Parameters	Legt die PIDT1 Parameter der X-Achse des Parameter Sets fest.
SetYAxisPIDT1Parameters	–	BOOL	REFERENCE TO LinearAxisPIDT1Parameters	Legt die PIDT1 Parameter der Y-Achse des Parameter Sets fest.
SetZAxisPIDT1Parameters	–	BOOL	REFERENCE TO LinearAxisPIDT1Parameters	Legt die PIDT1 Parameter der Z-Achse des Parameter Sets fest.

#### 15.1.5.9.2 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_GetObjectId	–	OTCID	–	Gibt die Objekt ID dieses TcCom-Objekts zurück.

## 15.1.5.10 FB\_TcIoFilterParametrization – I-TcIoFilterParametrization

## 15.1.5.10.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
SetAAxisFilters	–	BOOL	UINT; PVOID	Legt die Filterparameter für die A-Achse des Parameter Sets fest.  Verwenden Sie den UINT als Index und den PVOID als Adresse einer lokalen Variablen, die in das Filterparameter Set geschrieben werden soll.
SetBAxisFilters	–	BOOL	UINT; PVOID	Legt die Filterparameter für die B-Achse des Parameter Sets fest.  Verwenden Sie den UINT als Index und den PVOID als Adresse einer lokalen Variablen, die in das Filterparameter Set geschrieben werden soll.
SetCAxisFilters	–	BOOL	UINT; PVOID	Legt die Filterparameter für die C-Achse des Parameter Sets fest.  Verwenden Sie den UINT als Index und den PVOID als Adresse einer lokalen Variablen, die in das Filterparameter Set geschrieben werden soll.
SetXAxisFilters	–	BOOL	UINT; PVOID	Legt die Filterparameter für die X-Achse des Parameter Sets fest.  Verwenden Sie den UINT als Index und den PVOID als Adresse einer lokalen Variablen, die in das Filterparameter Set geschrieben werden soll.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
SetYAxisFilters	–	BOOL	UINT; PVOID	Legt die Filterparameter für die Y-Achse des Parameter Sets fest.  Verwenden Sie den UINT als Index und den PVOID als Adresse einer lokalen Variablen, die in das Filterparameter Set geschrieben werden soll.
SetZAxisFilters	–	BOOL	UINT; PVOID	Legt die Filterparameter für die Z-Achse des Parameter Sets fest.  Verwenden Sie den UINT als Index und den PVOID als Adresse einer lokalen Variablen, die in das Filterparameter Set geschrieben werden soll.

## 15.1.5.10.2 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_GetObjectId	–	OTCID	–	Gibt die Objekt ID dieses TcCom-Objekts zurück.

## 15.1.5.11 FB\_TcIoGeneralParametrization – I-TcIoGeneralParametrization

## 15.1.5.11.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetBrakeAcceleration	–	LREAL	–	Gibt die Bremsbeschleunigung zurück.
GetDefaultSetpointCycleTime	–	LREAL	–	Gibt die Standard-Sollwert-Zykluszeit zurück.
GetDisablingForceDecrement	–	LREAL	–	Gibt die Dekremente der Abschaltkraft pro Zyklus zurück.
GetDisablingTime	–	LREAL	–	Gibt die Zeit zurück, die der Mover zum Abschalten hat.
GetEnablingForceIncrement	–	LREAL	–	Gibt die Inkremente der Freigabekraft pro Zyklus zurück.
GetEnablingTime	–	LREAL	–	Gibt die Zeit zurück, die der Mover zur Freigabe hat.
SetDisablingForceDecrement	–	BOOL	LREAL	Legt die Dekremente der Abschaltkraft pro Zyklus fest.
SetDisablingTime	–	BOOL	LREAL	Legt die Zeit fest, die der Mover zum Abschalten hat.
SetEnablingForceIncrement	–	BOOL	LREAL	Legt die Inkremente der Freigabekraft pro Zyklus fest.
SetEnablingTime	–	BOOL	LREAL	Legt die Zeit fest, die der Mover zur Freigabe hat.
GetPositionReferenceMode	–	PositionReferenceMode	–	Gibt den aktuellen Positionsreferenzmodus zurück.
GetPositionReferenceZOffset	–	LREAL	–	Gibt die aktuelle Positions-offset-Referenz für die Z-Achse zurück.
GetSetpointInterpolation	–	SetpointInterpolation	–	Gibt den aktuellen Modus der Sollwertinterpolation zurück.
SetPositionReferenceMode	–	BOOL	PositionReferenceMode	Legt den aktuellen Positionsreferenzmodus fest.
SetPositionReferenceZOffset	–	BOOL	LREAL	Legt die aktuelle Positions-offset-Referenz für die Z-Achse fest.
SetSetpointInterpolation	–	BOOL	SetpointInterpolation	Legt den aktuellen Modus der Sollwertinterpolation fest.
SetBrakeAcceleration	–	BOOL	LREAL	Legt die Bremsbeschleunigung fest.
SetDefaultSetpointCycleTime	–	BOOL	LREAL	Legt die Standard-Sollwert-Zykluszeit fest.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetZPosAtEnable	–	LREAL	–	Gibt die Z-Position zurück, die der Mover bei der Freigabe zu erreichen versucht.
SetZPosAtEnable	–	BOOL	LREAL	Legt die Z-Position fest, die der Mover bei Freigabe zu erreichen versucht.

#### 15.1.5.11.2 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_GetObjectId	–	OTCID	–	Gibt die Objekt ID dieses TcCom-Objekts zurück.

## 15.1.5.12 FB\_TcIoInertiaParametrization – I\_TcIoInertiaParametrization

### 15.1.5.12.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetBasicInertia	–	ST_TcloPayload	–	Gibt die Basisträgheit zurück
GetProductsOfInertiaParameters	–	ST_TcloPayloadProductsOfInertia	–	Gibt das Produkt der Trägheit zurück.
SetBasicInertia	–	BOOL	ST_TcloPayload	Legt die Grundträgheit fest.
SetProductsOfInertiaParameters	–	BOOL	ST_TcloPayloadProductsOfInertia	Legt die Trägheitsprodukte der Zuladung fest.

### 15.1.5.12.2 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_GetObjectId	–	OTCID	–	Gibt die Objekt ID dieses TcCom-Objekts zurück.

## 15.1.5.13 FB\_TcIoObserverParametrization – I\_TcIoObserverParametrization

## 15.1.5.13.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namens dieses TcCom-Objekts zurück.
GetAAxisObserverParameters	–	REFERENCE TO RotationalAxisObserverParameters	–	Gibt die Beobachtungsparameter der A-Achse zurück.
GetBAxisObserverParameters	–	REFERENCE TO RotationalAxisObserverParameters	–	Gibt die Beobachtungsparameter der B-Achse zurück.
GetCAxisObserverParameters	–	REFERENCE TO RotationalAxisObserverParameters	–	Gibt die Beobachtungsparameter der C-Achse zurück.
GetXAxisObserverParameters	–	REFERENCE TO LinearAxisObserverParameters	–	Gibt die Beobachtungsparameter der X-Achse zurück.
GetYAxisObserverParameters	–	REFERENCE TO LinearAxisObserverParameters	–	Gibt die Beobachtungsparameter der Y-Achse zurück.
GetZAxisObserverParameters	–	REFERENCE TO LinearAxisObserverParameters	–	Gibt die Beobachtungsparameter der Z-Achse zurück.
SetAAxisObserverParameters	–	BOOL	REFERENCE TO RotationalAxisObserverParameters	Legt die Beobachtungsparameter der A-Achse fest.
SetBAxisObserverParameters	–	BOOL	REFERENCE TO RotationalAxisObserverParameters	Legt die Beobachtungsparameter der B-Achse fest.
SetCAxisObserverParameters	–	BOOL	REFERENCE TO RotationalAxisObserverParameters	Legt die Beobachtungsparameter der C-Achse fest.
SetXAxisObserverParameters	–	BOOL	REFERENCE TO LinearAxisObserverParameters	Legt die Beobachtungsparameter der X-Achse fest.
SetYAxisObserverParameters	–	BOOL	REFERENCE TO LinearAxisObserverParameters	Legt die Beobachtungsparameter der Y-Achse fest.
SetZAxisObserverParameters	–	BOOL	REFERENCE TO LinearAxisObserverParameters	Legt die Beobachtungsparameter der Z-Achse fest.

## 15.1.5.13.2 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_GetObjectId	–	OTCID	–	Gibt die Objekt ID dieses TcCom-Objekts zurück.

## 15.1.5.14 FB\_McPlanarGroup – I\_McPlanarGroup

### 15.1.5.14.1 Methode

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetGroupState	–	MC_PLA- NAR_STATE	–	Gibt den Status dieser Gruppe zurück.
GetObjectCount	–	UDINT	–	Gibt die Anzahl der Objekte dieser Gruppe zurück.

### 15.1.5.14.2 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_GetObjectId	–	OTCID	–	Gibt die Objekt ID dieses TcCom-Objekts zurück.

## 15.1.5.15 FB\_McPlanarMover – I\_McPlanarMover

## 15.1.5.15.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	General	BOOL	BOOL	Initialisierung
GetObjectName	General	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetCCoordinateModuloTolerance	General	LREAL	–	Gibt die Modulo-Toleranz der C-Achse zurück.
GetCCoordinateModulus	General	LREAL	–	Gibt den Modulus für die C-Achse zurück.
GetMoverHeight	General	LREAL	–	Gibt die Höhe des Movers zurück.
GetMoverWidth	General	LREAL	–	Gibt die Breite des Movers zurück.
GetInitialPosition	Simulation	MoverVector	–	Gibt die Ausgangsposition des Movers zurück.
GetSimulationMode	Simulation	BOOL	–	Gibt den Simulationsmodus des Movers zurück.
GetMaximalAcceleration	Maximum Dynamics	MoverVector	–	Gibt die maximale Beschleunigung des Movers zurück.
GetMaximalDeceleration	Maximum Dynamics	MoverVector	–	Gibt die maximale Verzögerung des Movers zurück.
GetMaximalVelocity	Maximum Dynamics	MoverVector	–	Gibt die maximale Geschwindigkeit des Movers zurück.
GetMaximalJerk	Maximum Dynamics	MoverVector	–	Gibt den maximalen Ruck des Movers zurück.
GetMaximumDynamicA	Maximum Dynamics	DynamicLimits1D	–	Gibt die maximale Dynamik für die A-Achse zurück.
GetMaximumDynamicB	Maximum Dynamics	DynamicLimits1D	–	Gibt die maximale Dynamik für die B-Achse zurück.
GetMaximumDynamicC	Maximum Dynamics	DynamicLimits1D	–	Gibt die maximale Dynamik für die C-Achse zurück.
GetMaximumDynamicXY	Maximum Dynamics	DynamicLimits1D	–	Gibt die maximale Dynamik für die X-Achse und Y-Achse zurück.
GetMaximumDynamicZ	Maximum Dynamics	DynamicLimits1D	–	Gibt die maximale Dynamik für die Z-Achse zurück.
GetDefaultAcceleration	Default Dynamics	MoverVector	–	Gibt die Standardbeschleunigung des Movers zurück.
GetDefaultDeceleration	Default Dynamics	MoverVector	–	Gibt die Standardverzögerung des Movers zurück.
GetDefaultVelocity	Default Dynamics	MoverVector	–	Gibt die Standardgeschwindigkeit des Movers zurück.
GetDefaultJerk	Default Dynamics	MoverVector	–	Gibt den Standardruck des Movers zurück.

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetDefaultDynamicA	Default Dynamics	DynamicLimits1D	–	Gibt die Standarddynamik für die A-Achse zurück.
GetDefaultDynamicB	Default Dynamics	DynamicLimits1D	–	Gibt die Standarddynamik für die B-Achse zurück.
GetDefaultDynamicC	Default Dynamics	DynamicLimits1D	–	Gibt die Standarddynamik für die C-Achse zurück.
GetDefaultDynamicXY	Default Dynamics	DynamicLimits1D	–	Gibt die Standarddynamik für die X-Achse und Y-Achse zurück.
GetDefaultDynamicZ	Default Dynamics	DynamicLimits1D	–	Gibt die Standarddynamik für die Z-Achse zurück.
GetMaximumLagFilterTime	Monitoring	MoverVector	–	Gibt die maximale Verzögerungsfilterzeit zurück.
GetMaximumPositionLagValue	Monitoring	MoverVector	–	Gibt den maximalen Wert der Positionsverzögerung zurück.
GetPositionLagMonitoringEnabled	Monitoring	BOOL	–	Gibt die aktivierte Überwachung der Positionsverzögerung zurück.

## 15.1.5.15.2 Methode – Parameter (Online)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetCommandMode	–	MC_PLANAR_MOVER_COMMAND_MODE	–	Gibt den Befehlsmodus des Movers zurück.
GetExternalSetpositionMode	–	MC_EXTERNAL_SET_POSITION_MODE	–	Gibt den Modus der externen Sollposition des Mover zurück.
GetGroupOid	–	OTCID	–	Gibt die Gruppe zurück, zu der der Mover gehört.
GetMoverState	–	MC_PLANAR_STATE	–	Gibt den aktuellen Status des Movers zurück.
GetPosition	–	MoverVector	–	Gibt die aktuelle Position des Movers zurück.
GetPositionOnTrack	–	ST_MCMoverSetPosOnTrack	–	Gibt die Mover-on-Track-Informationen des Mover zurück.
GetTcloMoverOid	–	OTCID	–	Gibt die Objekt ID des verlinkten Tclo-Movers zurück.

## 15.1.5.15.3 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_GetObjectId	–	OTCID	–	Gibt die Objekt ID dieses TcCom-Objekts zurück.

## 15.1.5.16 FB\_McPlanarTrack – I\_McPlanarTrack

### 15.1.5.16.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetCheckCollision-AgainstStaticObjects	–	BOOL	–	Gibt den Kollision-mit-einem-statischen-Objekt-Parameter zurück.
GetCollisionRangeAtEnd	–	LREAL	–	Gibt den Kollisionsabstand am Ende des Tracks zurück.
GetCollisionRangeAtStart	–	LREAL	–	Gibt den Kollisionsabstand am Anfang des Tracks zurück.
GetCollisionRange-Mode	–	MC_PLA-NAR_TRACK_COLLISION_RANGE_MODE	–	Gibt den Kollisionsabstandsmodus des Tracks zurück.
GetMaximalMover-Height	–	LREAL	–	Gibt die maximale Höhe der Mover des Tracks zurück.
GetMaximalMover-Width	–	LREAL	–	Gibt die maximale Moverbreite der Mover des Tracks zurück.

### 15.1.5.16.2 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetGroupOid	–	OTCID	–	Gibt die Objekt ID der Gruppe zurück, zu der der Track gehört.
GetMoverCountOn-Track	–	UDINT	–	Gibt die Anzahl der Mover des Tracks zurück.
GetMovingMover-Count	–	UDINT	–	Gibt die Anzahl der Mover zurück, die sich auf dem Track bewegen oder sich entlang des Tracks befinden.
GetOperationMode	–	MC_PLA-NAR_TRACK_OPERATION_MODE	–	Gibt die aktuelle Betriebsart des Tracks zurück.
GetTrackLength	–	LREAL	–	Gibt die Gesamtlänge des Tracks zurück.
GetTrackState	–	MC_PLA-NAR_STATE	–	Gibt den Status des Tracks zurück.

### 15.1.5.16.3 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_GetObjectId	–	OTCID	–	Gibt die Objekt ID dieses TcCom-Objekts zurück.

## 15.1.5.17 FB\_McPlanarEnvironment – I\_McPlanarEnvironment

## 15.1.5.17.1 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
Init	–	BOOL	BOOL	Initialisierung
GetObjectName	–	STRING(20)	–	Gibt den Namen dieses TcCom-Objekts zurück.
GetXPlanarXpuOid	–	OTCID	–	Gibt die Objekt ID der XPlanar Processing Unit zurück, die zu dieser Umgebung verlinkt ist.

## 15.1.5.17.2 Methode – Parameter (Init)

Parameter	Gruppe	Return Type	Input Type	Erläuterung
GetBoundaryElementCount	–	UDINT	–	Gibt die Anzahl der Ränder in dieser Umgebung zurück.
GetGroupOid	–	OTCID	–	Gibt die Objekt ID der Gruppe zurück, die zu dieser Umgebung hinzugefügt sind.
GetStatorCount	–	UDINT	–	Gibt die Anzahl der Statoren in dieser Umgebung zurück.

## 15.1.5.17.3 Eigenschaften

Parameter	Gruppe	Return Type	Input Type	Erläuterung
P_GetObjectId	–	OTCID	–	Gibt die Objekt ID des TcCom-Objekts zurück.

## **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.

Mehr Informationen:  
**[www.beckhoff.com/xplanar](http://www.beckhoff.com/xplanar)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

