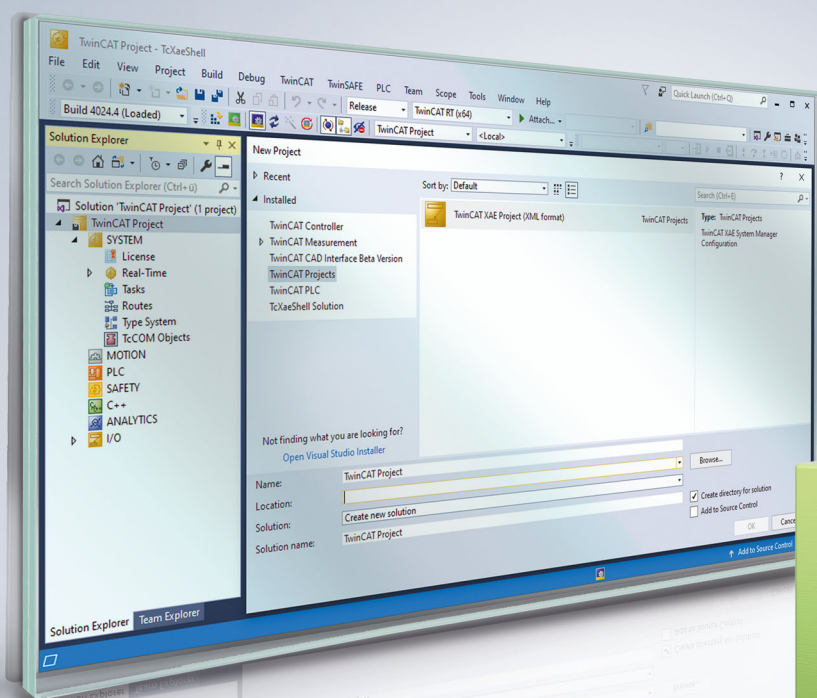


Handbuch | DE

# TwinCAT 3

Grundlagen





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort.....</b>	<b>5</b>
1.1	Hinweise zur Dokumentation .....	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit .....	7
<b>2</b>	<b>Echtzeit .....</b>	<b>8</b>
<b>3</b>	<b>Zielsysteme.....</b>	<b>12</b>
3.1	Routing.....	12
3.1.1	ADS.....	12
3.1.2	AmsNAT .....	173
3.1.3	ADS-over-MQTT .....	177
3.1.4	Secure ADS .....	202
3.2	Ordner und Dateitypen .....	213
3.2.1	Dateien TwinCAT-SPS-Projekt .....	213
3.2.2	Dateien TwinCAT-C++-Projekt.....	218
3.2.3	Dateien TwinCAT-Projekt.....	220
3.2.4	Dateien PLC HMI .....	220
3.2.5	Dateien PLC HMI (Target Visualisierung) .....	221
3.2.6	Dateien PLC HMI Web.....	221
3.3	Maschinen-Update auf Dateiebene.....	222
3.3.1	Übersicht.....	222
3.3.2	SPS-Update durchführen .....	223
3.3.3	C++-Update durchführen .....	223
3.3.4	Gesamtes Maschinen-Update durchführen .....	223
3.3.5	Clonen einer Maschine .....	224
3.4	Programm automatisch starten .....	224
3.5	Korrigierte Zeitstempel .....	225
3.5.1	Übersicht .....	225
3.5.2	Systemvoraussetzungen.....	226
3.5.3	Limitierungen.....	226
3.5.4	Technische Einführung .....	227
3.5.5	Realtime API .....	235
3.5.6	ADS API .....	239
3.5.7	Beispiele.....	240
3.5.8	FAQ.....	243
3.6	TcRTelinstall .....	244
<b>4</b>	<b>Typsystem .....</b>	<b>249</b>
4.1	Projektbasiertes Typsystem .....	249
4.2	Arten von Datentypen .....	249
4.3	Handhabung von Datentypen .....	250
4.4	Verwaltung und Identifizierung von Datentypen.....	252
4.5	Alignment von Datentypen .....	254
4.6	Dateien im Zusammenhang mit dem Typsystem .....	255
<b>5</b>	<b>Support und Service .....</b>	<b>256</b>



# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

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

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

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

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

### Disclaimer

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

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

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

### Marken

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

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

### Patente

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

## 1.2 Zu Ihrer Sicherheit

### Sicherheitsbestimmungen

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

### Haftungsausschluss

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

### Qualifikation des Personals

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

### Signalwörter

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

### Warnungen vor Personenschäden

#### **GEFAHR**

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

#### **WARNUNG**

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

#### **VORSICHT**

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

### Warnung vor Umwelt- oder Sachschäden

#### **HINWEIS**

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

### Information zum Umgang mit dem Produkt



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

## 1.3 Hinweise zur Informationssicherheit

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

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

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

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

## 2 Echtzeit

Entsprechend der Norm DIN 44300 ist die Echtzeit bzw. vielmehr der Echtzeitbetrieb definiert als: „Echtzeitbetrieb ist ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind, derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind.“.

Mit anderen Worten bedeutet dies, dass die Ausgabewerte eines Anwenderprogramms, berechnet basierend auf dem inneren Zustand und den Eingabewerten, innerhalb einer definierten und garantierten Zeit zur Verfügung stehen. Diese definierte Zeit wird auch Zykluszeit genannt.

Das Anwendungsprogramm selbst kann aus mehreren Programmbausteinen bestehen, die wiederum andere Programme, Funktionsbausteine etc. aufrufen (siehe auch Norm IEC 61131-3). Die Programmbausteine können Echtzeit-Tasks zugeordnet werden, welche diese wiederum mit einer zu definierenden Zykluszeit und einer definierten Priorität aufrufen.

Die TwinCAT 3 Echtzeit ist eine Echtzeiterweiterung, welche in der aktuellen TwinCAT 3.1 Version unter den Microsoft Windows Betriebssystemen ab Windows 7 sowie unter TwinCAT/BSD verwendet werden kann. Um den beschriebenen Anforderungen an eine Steuerung von industriellen Prozessen gerecht zu werden, unterstützt die TwinCAT 3 Echtzeit die folgenden Eigenschaften:

- Echtzeitfähiges Scheduling
- Parallele Abarbeitung von Prozessen
- Direkter Hardwarezugriff

Darüber hinaus gehend bietet die TwinCAT 3 Echtzeit auch Multicore-Support, um den immer weiter steigenden Anforderungen an eine performante und flexible/erweiterbare Steuerungsplattform gerecht zu werden. Die verfügbaren Rechenkerne können dabei entweder exklusiv für TwinCAT genutzt werden oder sie werden mit Windows geteilt. Im Folgenden werden die Kerne daher als "isolated" oder "shared" bezeichnet.

### Echtzeitfähiges Scheduling

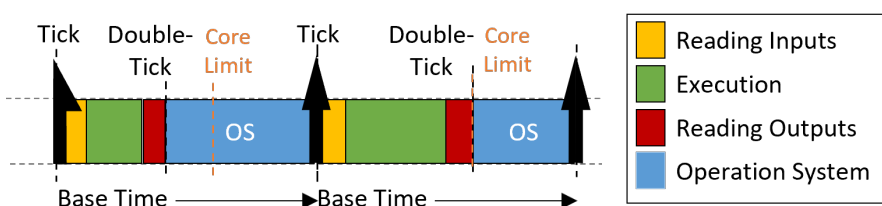
Die TwinCAT 3 Echtzeit arbeitet mit dem Doppeltick-Verfahren. Das bedeutet, dass das Umschalten in den Echtzeitmodus und wieder zurück jeweils von einem Interrupt ausgelöst wird. Der Interrupt beim Umschalten in den Echtzeitmodus startet gleichzeitig auch das Scheduling. Nach einer einstellbaren Zeitdauer, spätestens aber nach 90% der eingestellten Zykluszeit, schaltet TwinCAT auf „shared“-Kernen in den Nicht-Echtzeitmodus zurück, damit das Gastbetriebssystem genügend Rechenzeit erhält, um seinerseits die notwendigen Antwortzeiten für Hardware-Funktionen etc. einzuhalten. Eine Ausnahme bilden hier die isolierten Kerne.

Als Scheduling wird der (System-)Prozess bezeichnet, welcher die Abarbeitungsreihenfolge und den Abarbeitungszeitpunkt der einzelnen Tasks, basierend auf der definierten Zykluszeit und der definierten Priorität bestimmt. Die strenge Einhaltung des Abarbeitungszeitpunkts sorgt dafür, dass die oben beschriebene Einhaltung der Echtzeit gewährleistet wird.

Angestoßen durch einen synchronen Basis-Tick auf allen Echtzeitkernen, wird in der TwinCAT 3 Echtzeit das Scheduling für jeden Echtzeitkern unabhängig berechnet. Damit ist garantiert, dass Echtzeit-Tasks, welche auf verschiedenen Kernen laufen, sich nicht beeinflussen. Sofern dies nicht durch die Verwendung von Verriegelungen explizit im Anwenderprogramm programmiert wurde.

Ein Scheduling, bei dem die Priorität eines Tasks anhand seiner Zykluszeit abgeleitet wird, bezeichnet man auch als Ratenmonotones Scheduling. Die TwinCAT 3 Echtzeit aktiviert automatisch die Option „Automatic Priority Management“. Da dies nicht für jeden Anwendungsfall die beste Lösung ist, können Sie die Prioritäten manuell anpassen.

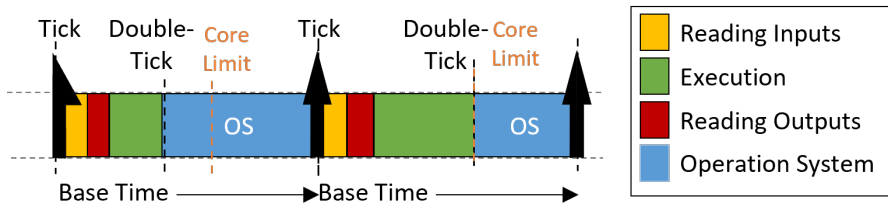
### Beispielhafte Darstellung des Aufrufs einer SPS-Task





In der Abbildung dargestellt sehen Sie den Aufruf einer SPS-Task. Nachdem der Echtzeit-Tick erfolgt ist, wird vom Scheduler die SPS-Task aufgerufen. Diese stellt der SPS-Anwendung die aktuellen Eingangswerte zur Verfügung (Input-Update), danach erfolgt die Abarbeitung des Anwendungsprogramms (Cycle-Update) und abschließend das Schreiben der Ergebnisse auf die Ausgänge (Output-Update). Ist dieses beendet, erfolgt das Umschalten in den Nicht-Echtzeit-Mode (Doppeltick). Wie in der Abbildung zu sehen ist, kann die Ausführungsdauer des Anwenderprogramms variieren, je nachdem welcher Code basierend auf dem inneren Zustand des Programms durchlaufen wird. Somit variiert auch der Zeitpunkt, wann die Ausgänge geschrieben werden. Je nachdem welche Task unter Umständen ein Bussystem treibt, kann dies dazu führen, dass das Absenden der Bustelegramme in gleichem Maße variiert.

**Beispielhafter Aufruf einer Task mit „IO am Task-Beginn“**



Durch die Verwendung der Option „IO am Task-Beginn“ kann die Abarbeitungsreihenfolge innerhalb einer Task dahingehend geändert werden, dass nach dem Lesen der Eingänge direkt die Ausgänge (des vorhergehenden Zyklus) geschrieben werden, bevor die Abarbeitung des Anwendungsprogramms erfolgt. Obwohl das Schreiben der Ausgänge erst einen Zyklus später erfolgt, hat diese Einstellung den Vorteil, dass der Zeitpunkt, wann die Ausgänge auf den Prozess / den Bus geschrieben werden, in jedem Zyklus exakt derselbe ist.

**Präemptives Multitasking**

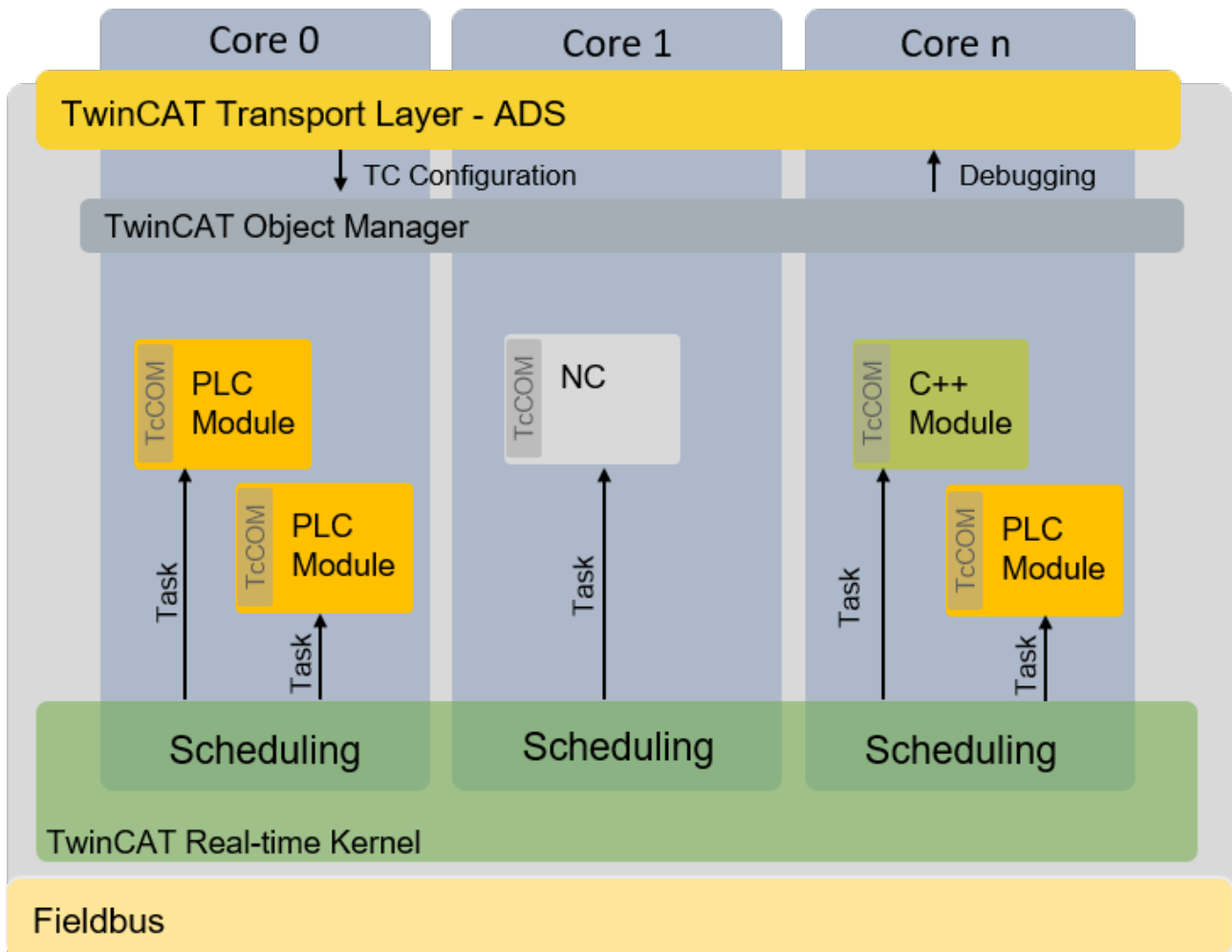
Präemptives Multitasking bedeutet, dass der aktuelle Zustand eines Prozesses (die CPU- und Floatingpoint-Register), bei einer Unterbrechung durch einen Interrupt (z. B. durch höher priorie Prozesse), gesichert und der aktuelle Prozess „schlafen gelegt“ wird. Ist dies passiert, bestimmt der Scheduler, anhand der Prioritäten der Tasks, den (neuen) abzuarbeitenden Prozess. Nachdem der zu unterbrechende Prozess beendet wurde, wird der Prozesskontext wiederhergestellt und der „alte“ Prozess fortgesetzt.

**Direkter Hardwarezugriff**

Um ein deterministisches (reproduzierbares) Echtzeit-Verhalten zu erreichen, benötigt die TwinCAT 3 Echtzeit einen direkten Hardwarezugriff. Damit dies möglich ist, muss die TwinCAT 3 Echtzeit im sogenannten Kernel-Mode von Windows ausgeführt werden. Dadurch ist es u.a. möglich, dass die TwinCAT-Echtzeit direkt auf die Netzwerk-Ports zugreift und Echtzeit-Ethernet-Telegramme (z. B. EtherCAT) versenden und empfangen kann.

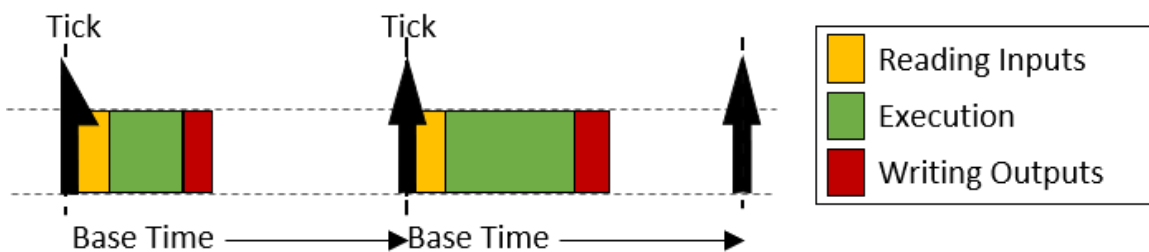
**Schematische Darstellung der TwinCAT 3-Laufzeitumgebung**

Das folgende Bild stellt den Aufbau der TwinCAT 3.1 Laufzeitumgebung (Runtime), bezogen auf das Scheduling, schematisch dar. Die TwinCAT 3 Laufzeitumgebung ermöglicht das Ausführen von Anwendermodulen in Echtzeit. Ein wesentlicher Teil der TwinCAT 3 Laufzeitumgebung ist somit der Echtzeit-Treiber, welcher auf den für TwinCAT aktivierten Kernen ausgeführt wird und dort das Scheduling übernimmt. Letzteres erfolgt auf den einzelnen Kernen unabhängig voneinander.



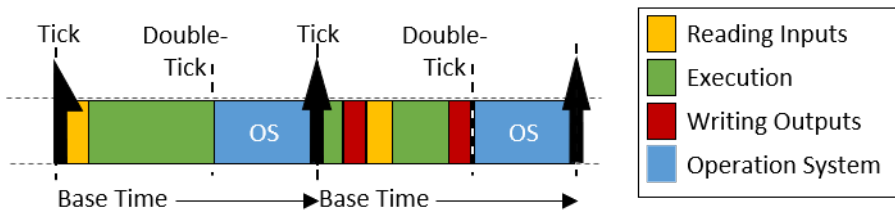
**Isolierte Kerne**

Wie unter [Echtzeit Scheduling \[▶ 8\]](#) beschrieben, verwendet TwinCAT ein Doppeltick-Verfahren, damit zu einem festgelegten Zeitpunkt in den Nicht-Echtzeitmodus zurückgeschaltet wird. Beim Umschalten zwischen Echtzeit-Modus und Nicht-Echtzeit-Modus erfolgt, wie unter [Präemptives Multitasking \[▶ 9\]](#) beschrieben, ein Restaurieren des vorgehenden Prozesszustands. Je nachdem wie intensiv die Echtzeit- und Nicht-Echtzeit-Programme den Speicher und insbesondere den Cache auslasten, braucht das Wiederherstellen Zeit. Um diese zeitlichen Effekte zu beseitigen, erlaubt es die TwinCAT 3.1 Echtzeit, Kerne vom Gastbetriebssystem zu isolieren. Dadurch ist ein Zurückschalten nicht mehr erforderlich, was sowohl in mehr Rechenzeit für das Echtzeit-Anwenderprogramm resultiert als auch in einer besseren Echtzeit-Güte (geringerer Jitter) durch die Vermeidung von zeitlichen Effekten beim Wiederherstellen des „alten“ Prozesszustands.

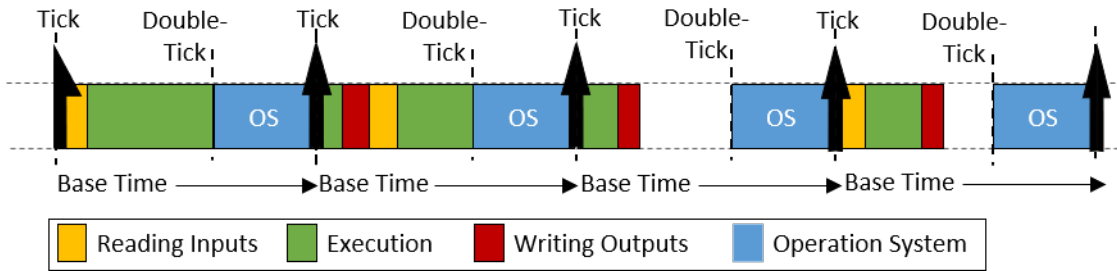


**Verhalten bei Zykluszeitüberschreitung**

Wird die definierte Zykluszeit eines Tasks überschritten, wird im nächsten Zyklus die Abarbeitung des „alten“ Zyklus fortgesetzt. Zudem wird der Überschreitungszähler der Task nach oben gesetzt. Nach der fertigen Abarbeitung des alten / vorangegangenen Zyklus, wird sofort versucht die Taskabarbeitung des aktuellen Zyklus zu starten. Wird diese innerhalb dieses Zyklus fertig gestellt, erfolgt die weitere Abarbeitung wie oben gezeigt.



Wird auch der zweite direkt darauffolgende Zyklus überschritten (wobei es hierbei unerheblich ist, ob es sich noch um die Abarbeitung des 1. Zyklus oder bereits die Abarbeitung des 2. Zyklus handelt), wird die aktuelle Bearbeitung fertig ausgeführt und das nächste Starten der Abarbeitung der Task startet erst zum nächstmöglichen geplanten Zyklusstart. Es gehen hierbei also unter Umständen mehrere Zyklen verloren. Der Überschreitungs-zähler wird auch hierbei entsprechend hochgezählt.

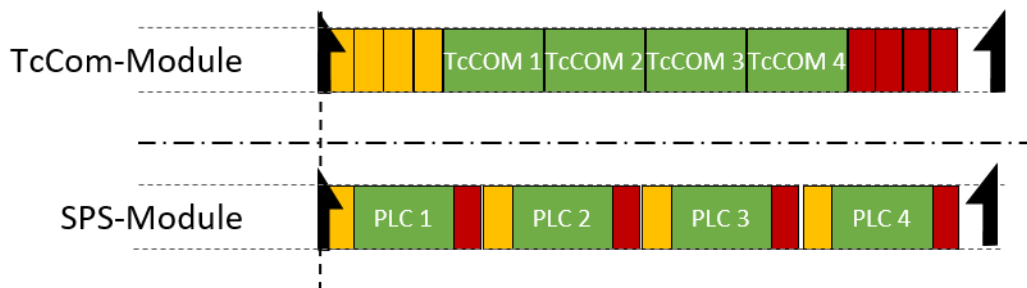


**Unterschiede in der Abarbeitung zwischen SPS- zu „TcCom“-Laufzeitmodulen**

Die Abarbeitungsreihenfolge einer TwinCAT-Task, bezogen auf die Ausführung von Laufzeitmodulen, besteht aus der folgenden Sequenz:

1. Umkopieren der Eingänge auf die Prozessabbilder der von ihr aufgerufenen Laufzeitmodule.
2. Ausführen der Module entsprechend der Sort-Order (in aufsteigender Reihenfolge).
3. Ausgangs-Update, welches die Ausgänge entsprechend bereitstellt. Treibt diese Task einen EtherCAT-Feldbus, wird der Frame während des Ausgangsprozessabbildes bereitgestellt und versendet.
4. Post-Zyklusupdate: Wird u. a. für das Anstoßen des Zyklusupdates verwendet, wenn die Option „IO am Taskbeginn“ aktiv ist.

Werden Laufzeitmodule einer Task hinzugefügt, „melden“ diese sich an den jeweiligen Aufrufen der Task an. Die einzige Ausnahme sind SPS-Laufzeitmodule. Aus Kompatibilitätsgründen zu TwinCAT 2 erfolgt durch die SPS-Laufzeitmodule direkt das Updaten der Ein- und Ausgänge. Der Unterschied zwischen den beiden Verhaltensweisen wird in der folgenden Abbildung dargestellt:



Zu sehen sind jeweils 4 Laufzeitmodule. Standard-TwinCAT 3 –Laufzeitmodule melden sich bei den entsprechenden Methoden-Aufrufen der Task an. Das bedeutet, alle Ein- (gelb) und Ausgangsupdates (rot) werden von der Task angestoßen und erfolgen direkt nacheinander zu Beginn bzw. am Ende der Taskabarbeitung. Kommunizieren zwei dieser Module über ein Mapping miteinander, so erhalten diese die jeweils aktuellen Werte erst im nächsten Zyklus.

Die SPS-Laufzeitmodule führen eigenständig ein Ein- und Ausgangsupdate durch. Kommunizieren zwei SPS-Laufzeiten miteinander, so bekommt das Laufzeitmodul, welches als zweites ausgeführt wird, direkt die aktuellen Werte vom ersten Laufzeitmodul. Somit ist in der Kommunikationsrichtung 1. Laufzeitmodul -> 2. Laufzeitmodul kein Zyklusversatz, in die andere Richtung aber schon.

## 3 Zielsysteme

Als Zielsystem wird jeweils die Steuerung bezeichnet, die gerade mit einer TwinCAT Entwicklungsumgebung (TwinCAT XAE) programmiert wird. In diesem Kapitel sollen wichtige Grundlagen beim Handling von Zielsystemen erläutert werden. Diese werden zudem benötigt, um die darauf aufbauende Dokumentation im Kapitel TE1000 XAE zu verstehen.

Um eine Steuerung programmieren zu können, muss zuerst eine Verbindung zwischen der Entwicklungsumgebung und der Steuerung geschaffen werden. Hierzu sind verschiedene Kanäle möglich. Die einzelnen Möglichkeiten werden im Kapitel [Routing \[► 12\]](#) näher erläutert.

Ist eine Steuerung bereits programmiert und im Feld und es soll ein Update der Maschine erfolgen, ohne dass die Programmierumgebung zum Einsatz kommt, ist es erforderlich zu wissen, welche Dateien und Ordner existieren, wozu sie benötigt werden und wie Sie sie am besten austauschen. Diesen Themen widmen sich die Kapitel [Ordner und Dateitypen \[► 213\]](#) und [Maschinen-Update auf Dateiebene \[► 222\]](#).

Unter Umständen ist es erforderlich, dass bei einem Neustart von TwinCAT auch zusätzliche Programme automatisch gestartet werden (z.B. eine externe HMI). Dies wird im Kapitel [Programm automatisch starten \[► 224\]](#) erläutert.

Arbeiten mehrere Steuerungen in einem Verbund am selben Prozess, ist es bei der Sammlung und Auswertung von Daten erforderlich, die Zeitstempel der einzelnen Steuerungen so zu korrigieren, dass die gesammelten Daten die exakte zeitliche Abfolge einhalten. Um dies zu erreichen, können Sie die Zeitstempel der einzelnen Steuerungen entsprechend korrigieren. Dies beschreibt das Kapitel [Korrigierte Zeitstempel \[► 225\]](#).

### 3.1 Routing

Wie bereits im Kapitel Philosophie beschrieben, besteht TwinCAT 3 aus einer Engineering-Umgebung (XAE) und einer Laufzeitumgebung (XAR). Die Engineering-Umgebung wird verwendet, um die Laufzeitumgebungen im Feld zu konfigurieren und programmieren. Die Laufzeitumgebungen (Zielsysteme) führen die Steuerungsprogramme dann in harter Echtzeit aus. Die Verbindung zwischen beiden Umgebungen, die nicht zwangsweise auf demselben PC/ IPC ausgeführt werden, erfolgt über das ADS-Protokoll (siehe Kapitel [ADS \[► 12\]](#)). Damit eine Engineering-Umgebung mit einem Zielsystem kommunizieren kann, muss eine Route eingetragen werden. Damit wird auf beiden Seiten (Engineering-Umgebung und Laufzeitumgebung) der jeweils andere Teilnehmer als bekannt eingetragen.

Um den aktuellen technischen Trends und Erfordernissen hinsichtlich Sicherheit und Konnektivität Rechnung zu tragen, können Sie die ADS-Verbindung entsprechend absichern bzw. auch über aktuelle Transportprotokolle tunneln. Siehe Kapitel [Secure ADS \[► 202\]](#) bzw. [ADS-over-MQTT \[► 177\]](#).

#### 3.1.1 ADS

##### 3.1.1.1 Einführung ADS

###### Definition ADS

Die Automation Device Specification beschreibt eine geräte- und feldbusunabhängige Schnittstelle, welche die Art des Zugriffs auf ADS-Geräte regelt.

Das ADS-Interface ermöglicht:

- die Kommunikation mit anderen ADS-Geräten
- die Implementierung eines ADS-Gerätes

###### ADS-Kommunikationsteilnehmer

Zur Teilnahme an der ADS-Kommunikation (als ADS-Client oder evtl. als ADS-Server) stehen folgende Software-Objekte zur Verfügung:

- ADS-DLL  
für den Einsatz unter z. B. C/C++
- ADS.NET [▶ 5]-Komponente  
für den Einsatz unter z. B. VB.NET, Visual C#

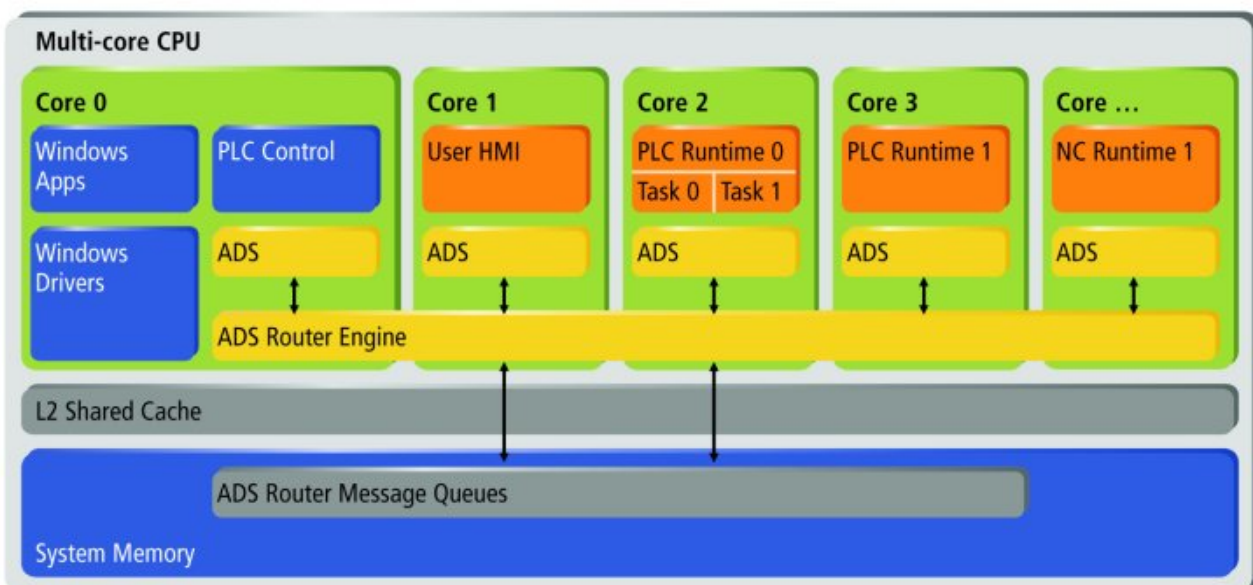
### 3.1.1.2 ADS-Gerätekonzept

Die Systemarchitektur von TwinCAT erlaubt es, die einzelnen Teile der Software (z. B. TwinCAT PLC, TwinCAT NC ...) als eigenständige Geräte zu betrachten: Für jede Aufgabe gibt es ein Softwaremodul („Server“ oder „Client“). Die Server im System sind die ausführenden Arbeits-„Geräte“ in Form von Software, welche in ihrem Betriebsverhalten genau einem Hardwaregerät entsprechen. Man kann daher von „virtuellen“ Geräten in Softwareform sprechen. Die „Clients“ sind Programme, welche die Dienste der „Server“ anfordern, z. B. eine Visualisierung oder auch ein „Programmiergerät“ in Form eines Programms. Auf diese Weise kann TwinCAT wachsen, indem immer neue Server und Clients entstehen, für Aufgaben, wie z. B. Nockenschaltwerk, Oszilloskop, PID-Regler usw.

Der Nachrichtenaustausch zwischen diesen Objekten wird über eine einheitliche ADS-Schnittstelle (**A**utomation **D**evice **S**pecification) vom „Message-Router“ abgewickelt. Dieser verwaltet und verteilt alle Nachrichten im System und über TCP/IP-Verbindungen.

TwinCAT Message-Router existieren auf jedem TwinCAT-Gerät. Somit können alle TwinCAT-Server und Client-Programme Befehle und Daten austauschen, Nachrichten versenden, Statusinformationen übermitteln usw.

Die folgende Abbildung gibt das TwinCAT Gerätekonzept auf der Grundlage von ADS wieder:



### 3.1.1.3 Identifikation ADS-Gerät

Die eindeutige Identifizierung von ADS-Geräten erfolgt über zwei Kennungen:

- PortNr
- NetId

**AMS-Ports**

Die ADS-Geräte an einem TwinCAT-Message-Router werden über eine Nummer, der sogenannten ADS-PortNr, eindeutig identifiziert. Diese ist bei ADS-Geräten fest vorgegeben, während reine ADS-Client-Anwendungen (z. B. ein HMI-System) bei dem ersten Zugriff auf den Message-Router eine variable Port-Nummer zugewiesen bekommen.

Folgende ADS-Port-Nummern sind bereits vergeben:

AMS-Port	Gerät
1	ADS Router
2	AMS Debugger
10	TCom Server
11	TCom Server-Task, RT-Kontext
12	TCom Server, passives Level
20	TwinCAT Debugger
21	TwinCAT Debugger Task
30	Lizenz Server
100	Logger
110	Event Logger
120	Applikation für EtherCAT Geräte
130	Event Logger User Mode (V2)
131	Event Logger Echtzeit (V2)
132	Event Logger Publisher (V2)
200	Ring 0 Echtzeit
290	Ring 0 Trace
300	Ring 0 IO
400	Ring 0 SPS (Legacy)
500	Ring 0 NC
501	Ring 0 NC SAF
511	Ring 0 NC SVB
520	NC-Instanz
550	Ring ISG
600	Ring 0 CNC
700	Ring 0 Zeile
800	Ring 0 TC2 SPS
801	TC2 SPS Laufzeitsystem 1
811	TC2 SPS Laufzeitsystem 2
821	TC2 SPS Laufzeitsystem 3
831	TC2 SPS Laufzeitsystem 4
850	Ring 0 TC3 SPS
851	TC3 SPS Laufzeitsystem 1
852	TC3 SPS Laufzeitsystem 2
853	TC3 SPS Laufzeitsystem 3
854 - ...	TC3 SPS Laufzeitsystem 4 - ...
900	Nockenschaltwerk
950	CAM-Tool
1000-1199	Ring 0 IO Ports
2000	Ring 0 Benutzer
2500	Crestron Server
10000	System Service
10201	TCP/IP Server
10300	System Manager
10400	SMS Server
10500	Modbus Server
10502	AMS Logger
10600	XML Datenserver
10700	Automatische Konfiguration

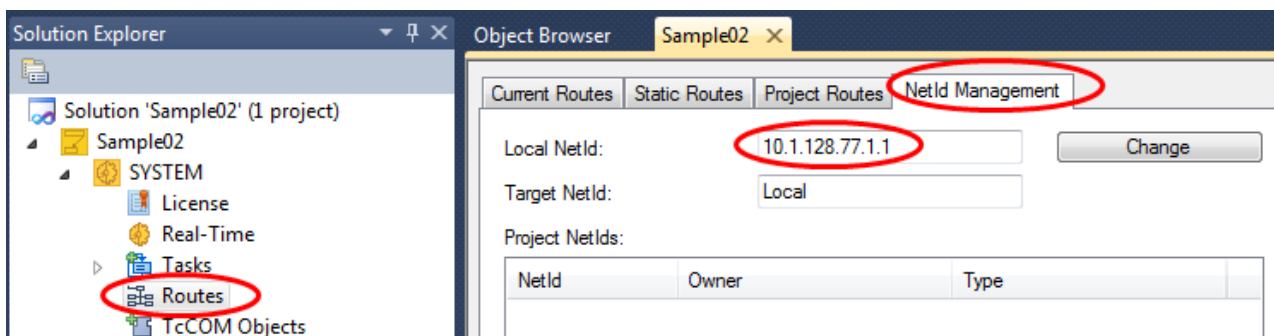
AMS-Port	Gerät
10800	PLC Control
10900	FTP Client
11000	NC Steuerung
11500	NC Interpreter
11600	GST Interpreter
12000	Strecke Steuerung
13000	CAM Steuerung
14000	Scope Server
14100	Condition Monitoring
15000	Sinus CH1
16000	CONTROL NET
17000	OPC Server
17500	OPC Client
18000	Mail Server
19000	Virtueller COM EL60xx
19100	Management Server
19200	Miele@home Server
19300	CP-Link3
19310	Touch lock
19500	Vision Service
19800	HMI Server
21372	Database Server
25000-25999	Reserved port range for ADS servers
25013	FIAS Server
25014	Bang&Olufsen Server
26000 - 26999	Private port range for customers
32768 – 65535	Reserved port range for ADS clients

## AMS-NetId

Jedes TwinCAT Gerät im Netzwerk wird durch die AMS-NetId identifiziert. Die AMS-NetId besteht aus sechs Oktetts, wovon die ersten vier frei gewählt werden können. Die letzten beiden Oktetts (in der Regel .1.1) dienen als Subnetzmaske für Feldbusse oder weitere Geräte. Die AMS-NetId muss für alle Kommunikationspartner eindeutig sein.

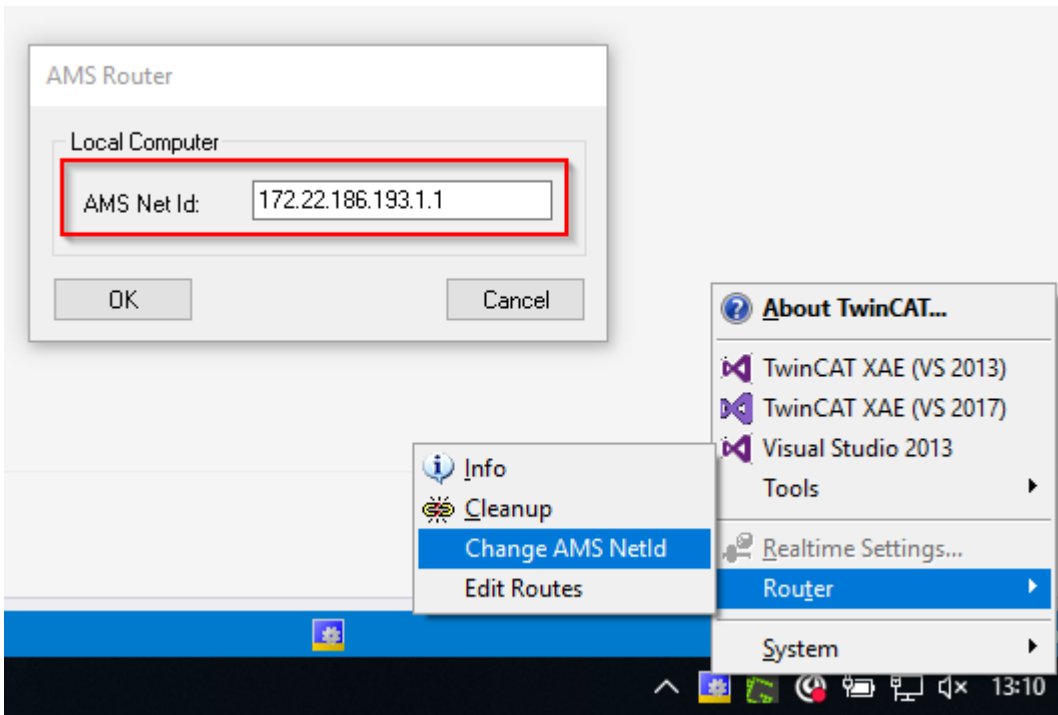
### Konfiguration:

Die AMS-NetId eines lokalen oder fernen TwinCAT-Geräts kann in SYSTEM\Routes\NetId Management eines TC3 Projekts eingestellt werden.



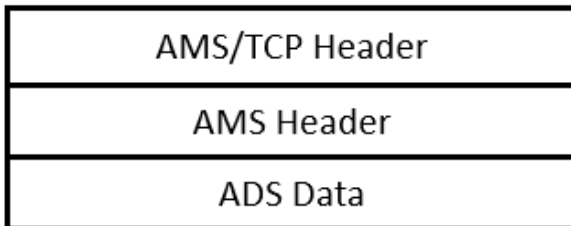
Die AMS-NetId kann alternativ auch lokal über das Systray-Menü von TwinCAT in der Kategorie Router konfiguriert werden. Das Gerät muss nach einer Änderung der AMS-NetId neu gestartet werden.





### 3.1.1.4 AMS/TCP Paket

#### 3.1.1.4.1 Struktur AMS/TCP Paket



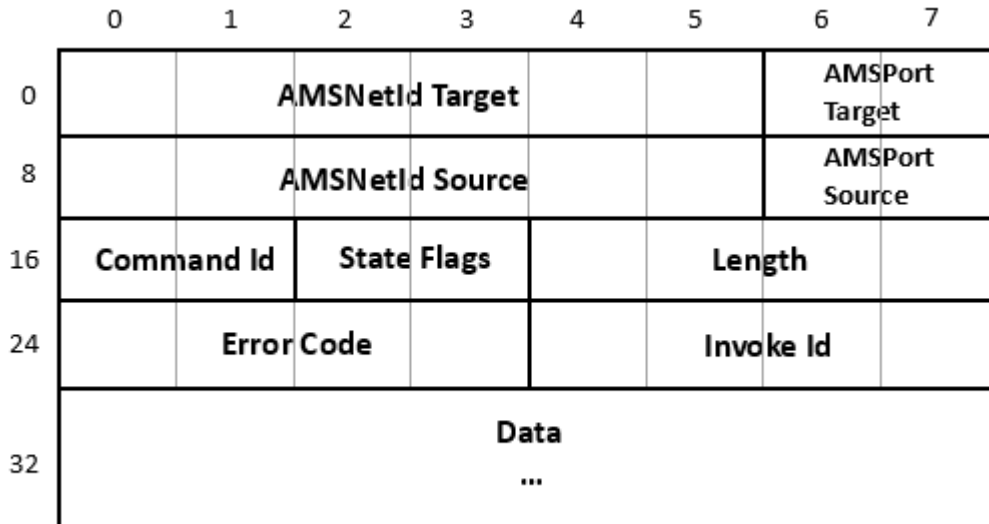
Datenarray	Größe	Beschreibung
AMS/TCP-Header	6 Bytes	Enthält die Länge des Datenpakets.
AMS-Header	32 Bytes	Der AMS/TCP-Header enthält die Adressen von Sender und Empfänger. Darüber hinaus den AMS-Fehlercode, die ADS-Befehls-ID und einige weitere Informationen.
ADS-Daten	n Bytes	Der ADS-Datenbereich enthält die Parameter der einzelnen ADS-Befehle. Die Struktur des Datenarrays hängt vom ADS-Befehl ab. Für einige ADS-Befehle sind keine zusätzlichen Daten erforderlich.

#### 3.1.1.4.2 AMS/TCP-Header



Datenarray	Größe	Beschreibung
reserviert	2 Bytes	Diese Bytes müssen auf 0 gesetzt werden.
Länge	4 Bytes	Dieser Array enthält die Länge des Datenpakets. Er besteht aus dem AMS-Header und den beigefügten ADS-Daten. Die Einheit ist Byte.

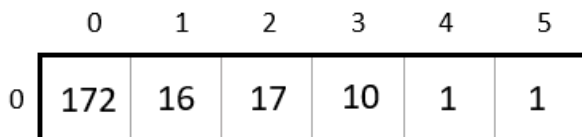
**3.1.1.4.3 AMS-Header**



Datenarray	Größe	Beschreibung
AMSNetID-Ziel	6 Bytes	Dies ist die AMSNetID der Station, für die das Paket bestimmt ist. Anmerkungen siehe nachstehend [▶ 18].
AMSPort-Ziel	2 Bytes	Dies ist der AMSPort der Station, für die das Paket bestimmt ist.
AMSNetID-Quelle	6 Bytes	Diese enthält die AMSNetID der Station, von der das Paket gesendet wurde.
AMSPort-Quelle	2 Bytes	Diese enthält den AMSPort der Station, von der das Paket gesendet wurde.
Befehls-ID	2 Bytes	siehe nachstehend [▶ 19].
State Flags	2 Bytes	siehe nachstehend [▶ 19].
Datenlänge	4 Bytes	Größe des Datenbereichs. Die Einheit ist Byte.
Fehlercode	4 Bytes	AMS-Fehlernummer. Siehe ADS Return-Codes.
Invoke-ID	4 Bytes	Frei verwendbarer 32-Bit-Array. Dieser Array wird normalerweise zum Senden einer ID verwendet. Diese ID ermöglicht die Zuweisung einer erhaltenen Antwort auf eine Anforderung, die zuvor gesandt wurde.
Daten	n Bytes	Datenbereich. Der Datenbereich enthält die Parameter der entsprechenden ADS-Befehle.

**AMS Net ID**

Die AMSNetID besteht aus 6 Bytes und spricht den Sender oder Empfänger an. Eine mögliche AMSNetID wäre z. B. 172.16.17.10.1.1. Die Speicheranordnung in diesem Beispiel ist wie folgt:



Die AMSNetID ist rein logisch und steht normalerweise in keiner Beziehung zu der IP-Adresse. Die AMSNetID wird im Zielsystem konfiguriert. Am PC wird hierfür die TwinCAT Systemsteuerung verwendet. Siehe bei Verwendung einer anderen Hardware die entsprechende Dokumentation für Hinweise zu den Einstellungen der AMS NetID.

**Befehls-ID**

Cmd	Beschreibung
0x0000	Ungültig
0x0001	ADS Read Device Info [▶ 20]
0x0002	ADS-Read [▶ 20]
0x0003	ADS-Write [▶ 21]
0x0004	ADS Read State [▶ 22]
0x0005	ADS Write Control [▶ 22]
0x0006	ADS Add Device Notification [▶ 23]
0x0007	ADS Delete Device Notification [▶ 24]
0x0008	ADS Device Notification [▶ 24]
0x0009	ADS Read Write [▶ 25]

Weitere Befehle sind nicht definiert oder werden intern verwendet. Daher darf die *Command-ID* lediglich die vorstehend aufgezählten Werte enthalten!

**State Flags**

Flag	Beschreibung
0x0001	0: Anforderung / 1: Antwort
0x0004	ADS-Befehl

Das erste Bit gibt Auskunft darüber, ob es sich um eine Anforderung oder eine Antwort handelt. Das dritte Bit muss auf 1 gesetzt werden, um Daten mit ADS-Befehlen auszutauschen. Die anderen Bits sind nicht definiert oder werden zu anderen internen Zwecken verwendet.

Daher müssen die anderen Bits auf 0 gesetzt werden!

Flag	Beschreibung
0x000x	TCP-Protokoll
0x004x	UDP-Protokoll

Bit Nummer 7 gibt an, ob es mit TCP oder UDP übertragen werden muss.

### 3.1.1.4.4 ADS-Befehle

#### 3.1.1.4.4.1 Überblick Befehle

Befehl	Beschreibung
<a href="#">ADS Read Device Info [► 20]</a>	Liest den Namen und die Versionsnummer des ADS-Geräts.
<a href="#">ADS-Read [► 20]</a>	Mit <i>ADS Read</i> können Daten aus einem ADS-Gerät gelesen werden
<a href="#">ADS-Write [► 21]</a>	Mit <i>ADS Write</i> können Daten an ein ADS-Gerät geschrieben werden.
<a href="#">ADS Read State [► 22]</a>	Liest den ADS-Status und den Geräte-Status von dem ADS-Gerät aus.
<a href="#">ADS Write Control [► 22]</a>	Ändert den ADS-Status und den Geräte-Status von einem ADS-Gerät.
<a href="#">ADS Add Device Notification [► 23]</a>	In einem ADS-Gerät wird eine Benachrichtigung erstellt.
<a href="#">ADS Delete Device Notification [► 24]</a>	Eine zuvor definierte Benachrichtigung wird in einem ADS-Gerät gelöscht.
<a href="#">ADS Device Notification [► 24]</a>	Es werden Daten unabhängig von einem ADS-Gerät an einen Client weitergeleitet.
<a href="#">ADS Read Write [► 25]</a>	Mit <i>ADS ReadWrite</i> können Daten an ein ADS-Gerät geschrieben werden. Zudem können Daten von einem ADS-Gerät ausgelesen werden.

#### 3.1.1.4.4.2 ADS Read Device Info

Liest den Namen und die Versionsnummer des ADS-Geräts.

##### Anforderung

Keine zusätzlichen Daten erforderlich

##### Antwort

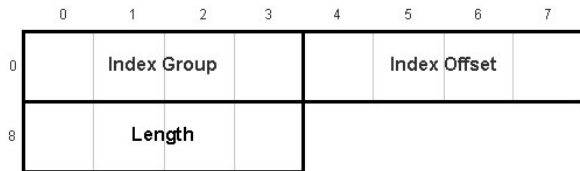
	0	1	2	3	4	5	6	7
0		<b>Result</b>		<b>Major Version</b>	<b>Minor Version</b>	<b>Version Build</b>		
8			<b>Device name</b>					
16								

Datenarray	Größe	Beschreibung
Ergebnis	4 Bytes	ADS-Fehlernummer.
Major Version	1 Byte	Major Versionsnummer
Minor Version	1 Byte	Minor Versionsnummer
Version Build	2 Bytes	Buildnummer
Gerätename	16 Bytes	Name des ADS-Geräts

#### 3.1.1.4.4.3 ADS-Read

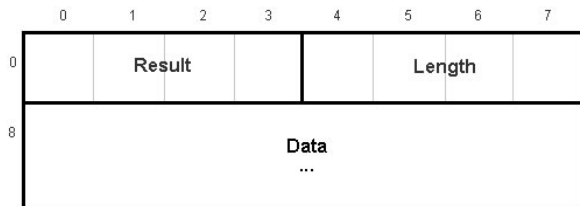
Mit *ADS Read* können Daten aus einem ADS-Gerät gelesen werden. Die Daten werden durch die *Index Group* und den *Index Offset* adressiert

**Anforderung**



Datenarray	Größe	Beschreibung
Index Group	4 Bytes	Indexgruppe der Daten, die gelesen werden sollen.
Index Offset	4 Bytes	Index-Offset der Daten, die gelesen werden sollen.
Länge	4 Bytes	Länge der Daten (in Byte), die gelesen werden sollen.

**Antwort**

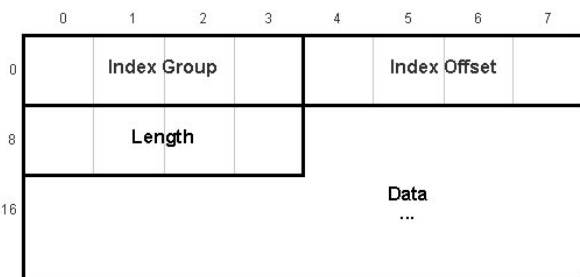


Datenarray	Größe	Beschreibung
Ergebnis	4 Bytes	ADS-Fehlernummer
Länge	4 Bytes	Länge der Daten, die zurückgeführt werden.
Daten	n Bytes	Daten, die zurückgeführt werden.

**3.1.1.4.4 ADS-Write**

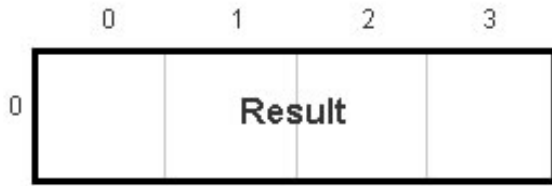
Mit *ADS Write* können Daten an ein ADS-Gerät geschrieben werden. Die Daten werden durch die *Index Group* und den *Index Offset* adressiert

**Anforderung**



Datenarray	Größe	Beschreibung
Index Group	4 Bytes	Indexgruppe, in welche die Daten geschrieben werden sollen.
Index Offset	4 Bytes	Index-Offset, in den die Daten geschrieben werden sollen.
Länge	4 Bytes	Länge der Daten in Bytes, die geschrieben werden
Daten	n Bytes	Daten, die im ADS-Gerät geschrieben werden.

**Antwort**



Datenarray	Größe	Beschreibung
Ergebnis	4 Bytes	ADS-Fehlernummer

**3.1.1.4.4.5 ADS Read State**

Liest den ADS-Status und den Geräte-Status von dem ADS-Gerät aus.

**Anforderung**

Keine zusätzlichen Daten erforderlich

**Antwort**

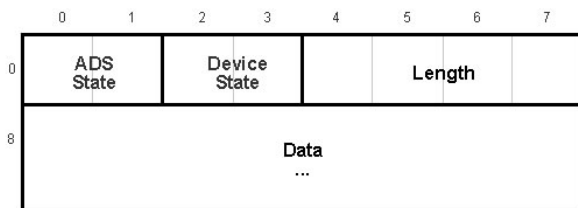


Datenarray	Größe	Beschreibung
Ergebnis	4 Bytes	ADS-Fehlernummer.
ADS State	2 Bytes	ADS-Status (siehe Datentyp ADSSTATE der ADS-DLL).
Device State	2 Bytes	Gerätestatus

**3.1.1.4.4.6 ADS Write Control**

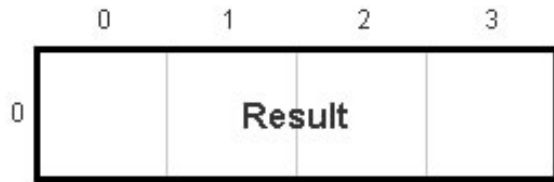
Ändert den ADS-Status und den Geräte-Status von einem ADS-Gerät. Zusätzlich ist es möglich, Daten an das ADS-Gerät zu senden, um weitere Informationen zu übertragen. Diese Daten werden von den aktuellen ADS-Geräten (SPS, NC usw.) nicht analysiert

**Anforderung**



Datenarray	Größe	Beschreibung
ADS State	2 Bytes	Neuer ADS-Status (siehe Datentyp ADSSTATE der ADS-DLL).
Device State	2 Bytes	Neuer Gerätestatus.
Länge	4 Bytes	Länge der Daten in Byte.
Daten	n Bytes	Zusätzliche Daten, die an das ADS-Gerät gesandt werden

Antwort



Datenarray	Größe	Beschreibung
Ergebnis	4 Bytes	ADS-Fehlernummer.

3.1.1.4.4.7 ADS Add Device Notification

In einem ADS-Gerät wird eine Benachrichtigung erstellt.

Hinweis: Wir empfehlen, nicht mehr als 550 Benachrichtigungen je Gerät vorzusehen. Erhöhen Sie andernfalls die Nutzlast, indem sie mit Strukturen arbeiten oder Summenbefehle verwenden.

Anforderung



Datenarray	Größe	Beschreibung
Index Group	4 Bytes	Indexgruppe der Daten, die je Benachrichtigung gesandt werden sollen.
Index Offset	4 Bytes	Index-Offset der Daten, die je Benachrichtigung gesandt werden sollen.
Länge	4 Bytes	Länge der Daten in Byte, die je Benachrichtigung gesandt werden sollen.
Übertragungsmodus	4 Bytes	Siehe Beschreibung der Struktur ADSTRANSMODE an der ADS-DLL.
Max. Verzögerung	4 Bytes	Spätestens nach Ablauf dieser Zeit wird die <i>ADS Device Notification</i> aufgerufen. Die Einheit ist 1 ms.
Zykluszeit	4 Bytes	Der ADS-Server prüft, ob sich der Wert innerhalb dieser Zeitscheibe ändert. Die Einheit ist 1 ms
reserviert	16 Bytes	Muss auf 0 gesetzt werden

Antwort



Datenarray	Größe	Beschreibung
Ergebnis	4 Bytes	ADS-Fehlernummer
Benachrichtigungs-Handle	4 Bytes	Handle der Benachrichtigung

### 3.1.1.4.4.8 ADS Delete Device Notification

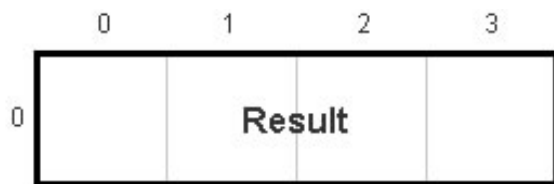
Eine zuvor definierte Benachrichtigung wird in einem ADS-Gerät gelöscht.

#### Anforderung



Datenarray	Größe	Beschreibung
Benachrichtigungs-Handle	4 Bytes	Handle der Benachrichtigung. Das Handle wird durch den ADS-Befehl <i>Add Device Notification</i> erstellt

#### Antwort



Datenarray	Größe	Beschreibung
Ergebnis	4 Bytes	ADS-Fehlernummer

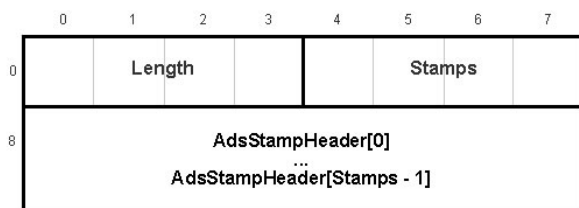
### 3.1.1.4.4.9 ADS Device Notification

Es werden Daten unabhängig von einem ADS-Gerät an einen Client weitergeleitet.

#### Anforderung

Die bei der *Device Notification* übertragenen Daten werden mehrfach ineinander verschachtelt. Der *Notification Stream* enthält einen Array mit Elementen des Typs *AdsStampHeader*. Dieser Array enthält wiederum Elemente des Typs *AdsNotificationSample*.

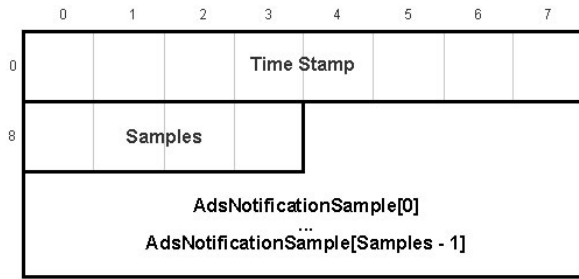
#### AdsNotificationStream



Datenarray	Größe	Beschreibung
Länge	4 Bytes	Größe der Daten in Byte.
Stempel	4 Bytes	Anzahl der Elemente des Typs <a href="#">ADS Device Notification</a> [► 25]
AdsStampHeader	n Bytes	Array mit Elementen des Typs <a href="#">ADS Device Notification</a> [► 25]

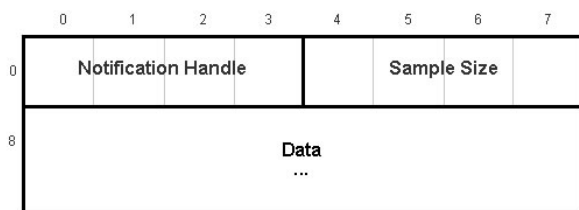


**AdsStampHeader**



Datenarray	Größe	Beschreibung
TimeStamp	8 Bytes	Der Zeitstempel ist nach dem Windows FILETIME-Format codiert. D. h. der Wert enthält die Anzahl der 100-Nanosekunden-Intervalle, die seit dem 1.1.1601 abgelaufen sind. Darüber hinaus ist die lokale Zeitänderung nicht berücksichtigt. Somit erfolgt der Zeitstempel nach der Universal Coordinated Time (UTC).
Beispiele	4 Bytes	Anzahl der Elemente des Typs <a href="#">ADS Device Notification</a> [► 25]
AdsNotificationSample	n Bytes	Array mit Elementen des Typs <a href="#">ADS Device Notification</a> [► 25]

**AdsNotificationSample**



Datenarray	Größe	Beschreibung
Benachrichtigungs-Handle	4 Bytes	Handle der Benachrichtigung.
Samplegröße	4 Bytes	Größe des Datenbereichs in Byte.
Daten	n Bytes	Daten



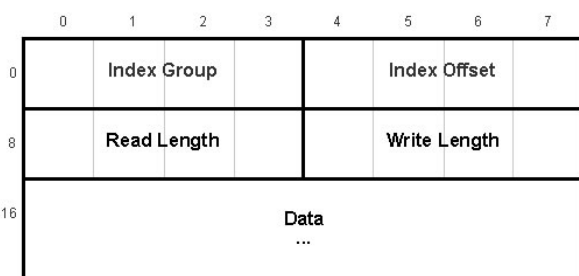
Wenn Ihr Handle ungültig wird, wird einmalig eine Benachrichtigung ohne Daten als Hinweis gesandt.

**3.1.1.4.4.10 ADS Read Write**

Mit *ADS ReadWrite* können Daten an ein ADS-Gerät geschrieben werden. Zudem können Daten von einem ADS-Gerät ausgelesen werden.

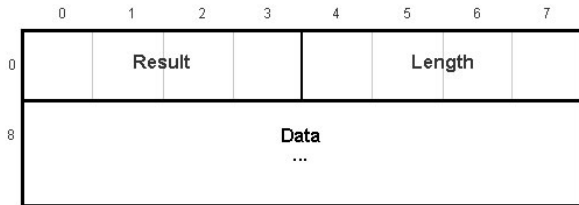
Die lesbaren Daten werden durch die *Index Group* und den *Index Offset* adressiert

**Anforderung**



Datenarray	Größe	Beschreibung
Index Group	4 Bytes	Indexgruppe, in welche die Daten geschrieben werden sollen.
Index Offset	4 Bytes	Index-Offset, in den die Daten geschrieben werden sollen.
Read Length	4 Bytes	Länge der Daten in Byte, die gelesen werden sollen.
Write Length	4 Bytes	Länge der Daten in Byte, die geschrieben werden sollen.
Daten	n Bytes	Daten, die im ADS-Gerät geschrieben werden.

**Antwort**



Datenarray	Größe	Beschreibung
Ergebnis	4 Bytes	ADS-Fehlernummer
Länge	4 Bytes	Länge der Daten, die zurückgeführt werden.
Daten	n Bytes	Daten, die zurückgeführt werden.

**3.1.1.5 Spezifikation für ADS-Geräte**

**3.1.1.5.1 Übersicht**

Die SPS-Software im Rahmen der Beckhoff TwinCAT-Software kann, da es sich um eine reine Software-SPS handelt, als virtuelles Feldgerät (Automation Device) beschrieben werden. Sie stellt daher für andere Kommunikationspartner (z.B. andere virtuelle Feldgeräte oder Windows-Programme) eine Beckhoff-ADS (Automation Device Specification) -Schnittstelle zur Verfügung, über die sie parametrisiert oder abgefragt werden kann. Die Verwendung des ADS standardisiert den Zugriff auf die SPS und reiht sie in die Gruppe der verfügbaren virtuellen Feldgeräte ein.

Die READ und WRITE Operationen auf der SPS-Schnittstelle erfolgen, wie durch ADS festgelegt, über zwei Zahlen: dem Index-Group und dem Index-Offset.

Auf den nächsten Seiten wird die ADS-Schnittstelle der SPS hinsichtlich der Gruppen- und Offsetindizes genauer beschrieben.

**Spezifikationen "Index-Group" der SPS**

Die vier globalen Bereiche eines ADS-Gerätes werden für die SPS als vier Abschnitte in den Index-Groups wie folgt abgebildet:

Index-Group (0x = hex)	Index Group Beschreibung
0x00000000 0x00000FFF	Reserviert
0x00001000	SPS-ADS Parameterbereich
0x00002000	SPS-ADS Zustandsbereich
0x00003000	SPS-ADS Gerätefunktionenbereich
0x00004000	Allgemeine SPS-ADS-Dienste (enthalten Dienste für den Zugriff auf den SPS-Prozessdatenbereich der Merker) [► 27]
0x00006000 0x0000EFFF	Reserviert für SPS-ADS Erweiterung
0x0000F000 0x0000FFFF	Allgemeine TwinCAT ADS-Systemdienste (enthalten Dienste für den Zugriff auf den SPS-Prozessdatenbereich der Ein- und Ausgänge) [► 27]

### 3.1.1.5.2 Spezifikation für allgemeine SPS-Dienste

In dieser Gruppe befinden sich auch Dienste für den Zugriff auf den SPS-Prozessdatenbereich der Merker.

Index Group	Index Offset	Zugriff	Datentyp	Beschreibung	Anmerkung
0x00004020	0x00000000-0x0000FFFF	R/W	UINT8[n]	<b>READ_M - WRITE_M</b> SPS-Memory-Bereich ( %M-Feld ). Offset ist Byteoffset	
0x00004021	0x00000000-0xFFFFFFFF	R/W	UINT8	<b>READ_MX - WRITE_MX</b> SPS-Memory-Bereich ( %MX-Feld ). Das Low-Word des Index-Offsets ist der Byteoffset. Der Index-Offset enthält die Bitadresse, die sich aus $\text{Bytenummer} * 8 + \text{Bitnummer}$ errechnet.	
0x00004025	0x00000000	R	ULONG	<b>PLCADS_IGR_RMSIZE</b> Bytelänge des SPS-Prozessabbildes des Memory-Bereiches	
0x00004030	0x00000000-0xFFFFFFFF	R/W	UINT8	<b>PLCADS_IGR_RWRB</b> Retain-Datenbereich. Offset ist Byteoffset	
0x00004035	0x00000000	R	ULONG	<b>PLCADS_IGR_RRSIZE</b> Bytelänge des Retain-Bereiches	
0x00004040	0x00000000-0xFFFFFFFF	R/W	UINT8	<b>PLCADS_IGR_RWDB</b> Daten-Bereich. Offset ist Byteoffset	
0x00004045	0x00000000	R	ULONG	<b>PLCADS_IGR_RDSIZE</b> Bytelänge des Daten-Bereiches	

### 3.1.1.5.3 Spezifikation für ADS-Systemdienste

Dieser Abschnitt umfasst diejenigen ADS-Dienste, die bei jedem TwinCAT-ADS-Gerät identische Bedeutung und Wirkung haben. In dieser Gruppe befinden sich auch Dienste für den Zugriff auf die SPS-Prozessdaten der Ein- und Ausgänge.

Index Group	Index Offset	Zugriff	Datentyp	Beschreibung
0x0000F003	0x00000000	R/W	W: UINT8[n] R: UINT32	<b>GET_SYMHANDLE_BYNAME</b> Dem in den Write-Daten enthaltene Namen wird ein Handle (Kennwert) zugewiesen und dem Aufrufer als Ergebnis in den Read-Daten zurückgereicht.
0x0000F004	0x00000000	R/W	W: UINT8[n] R: SIZEOF(SYMVAL)	<b>READ_SYMVAL_BYNAME</b>
0x0000F005	0x00000000-0xFFFFFFFF=symHandle	R/W	UINT8[n]	<b>READ / WRITE_SYMVAL_BYHANDLE</b> Den Wert, der durch 'symHdl' identifizierten Variable, lesen oder der Variablen einen Wert zuweisen. Der 'symHdl' muss vorher durch den GET_SYMHANDLE_BYNAME-Dienst ermittelt worden sein.
0x0000F006	0x00000000	W	UINT32	<b>RELEASE_SYMHANDLE</b> Die in den Write-Daten enthaltene Kennzahl (Handle) für eine abzufragende benannte SPS-Variable wird freigegeben.
0x0000F020	0x0001F400-0xFFFFFFFF	R/W	UINT8[n]	<b>READ_I - WRITE_I</b> SPS-Prozessabbild der physikalischen Eingänge( %I-Feld ). Offset ist Byteoffset.
0x0000F021	0x000FA000-0xFFFFFFFF	R/W	UINT8	<b>READ_IX - WRITE_IX</b> SPS-Prozessabbild der physikalischen Eingänge( %IX-Feld ). Der Index-Offset enthält die Bitadresse, die sich aus Basisoffset (0xFA000) + Bytenummer*8+Bitnummer errechnet.
0x0000F025	0x00000000	R	ULONG	<b>ADSIGRP_IOIMAGE_RISIZE</b> Bytelänge des SPS-Prozessabbildes der physikalischen Eingänge.
0x0000F030	0x0003E800-0xFFFFFFFF	R/W	UINT8[n]	<b>READ_Q - WRITE_Q</b> SPS-Prozessabbild der physikalischen Ausgänge( %Q-Feld ). Offset ist Byteoffset.
0x0000F031	0x001F4000-0xFFFFFFFF	R/W	UINT8	<b>READ_QX - WRITE_QX</b> SPS-Prozessabbild der physikalischen Ausgänge( %QX-Feld ). Der Index-Offset enthält die Bitadresse, die sich aus Basisoffset (0x1F4000) + Bytenummer*8+Bitnummer errechnet.
0x0000F035	0x00000000	R	ULONG	<b>ADSIGRP_IOIMAGE_ROSIZE</b> Bytelänge des SPS-Prozessabbildes der physikalischen Ausgänge.

Index Group	Index Offset	Zugriff	Datentyp	Beschreibung
0x0000F080	0x00000000-0xFFFFFFFF= n (Anzahl der internen (Sub-)Befehle)n(max) = 500	R&W	<p>W: (n * ULONG[3]) := <b>IG1, IO1, Len1,</b> <b>IG2, IO2, Len2,</b> ..., <b>IG(n), IO(n), Len(n)</b></p> <p>R: (n * ULONG) + UINT8[Len1] + UINT8[Len2] + ..., + UINT8[Len(n)] := <b>Result1, Result2, ...,</b> <b>Result(n),</b> <b>Data1, Data2, ...,</b> <b>Data(n)</b></p>	<b>ADSIGRP_SUMUP_READ</b> Die Write-Daten enthalten eine Liste von mehreren, separaten AdsReadReq(IG, IO, Len, Data) quasi als "Sammel-Lesebefehl". Dem Aufrufer wird in den Read-Daten das Ergebnis der Sammelanfrage zurückgereicht. Dabei werden zuerst alle Rückgabewerte aufgelistet, anschließend folgen die angefragten Daten.
0x0000F081	0x00000000 - 0xFFFFFFFF= n (Anzahl der internen (Sub-)Befehle)n(max) = 500	R&W	<p>W: (n * ULONG[3]) + UINT8[Len1] + UINT8[Len2] + ..., + UINT8[Len(n)] := <b>IG1, IO1, Len1,</b> <b>IG2, IO2, Len2,</b> ..., <b>IG(n), IO(n), Len(n),</b> <b>Data1, Data2, ...,</b> <b>Data(n)</b></p> <p>R: ULONG[n] := <b>Result1, Result2, ...,</b> <b>Result(n)</b></p>	<b>ADSIGRP_SUMUP_WRITE</b> Die Write-Daten enthalten eine Liste von mehreren, separaten AdsWriteReq(IG, IO, Len, Data) quasi als "Sammel-Schreibbefehl". Dem Aufrufer wird in den Read-Daten das Ergebnis der Sammelanfrage (die Rückgabewerte) zurückgereicht.

Index Group	Index Offset	Zugriff	Datentyp	Beschreibung
0x0000F082	0x00000000 - 0xFFFFFFFF= n (Anzahl der internen (Sub-)Befehle)n(max) = 500	R&W	<p>W: (n * ULONG[4]) + UINT8[WriteLen1] + UINT8[WriteLen2] + ..., + UINT8[WriteLen(n)] : = <b>IG1, IO1, ReadLen1, WriteLen1, IG2, IO2, ReadLen2, WriteLen2, ..., IG(n), IO(n), ReadLen(n), WriteLen(n), WriteData1, WriteData2, ..., WriteData(n)</b></p> <p>R: (n * ULONG[2]) + UINT8[ReturnLen1], + UINT8[ReturnLen2] + ..., + UINT8[ReturnLen(n)] := <b>Result1, ReturnLen1, Result2, ReturnLen2, ..., Result(n), ReturnLen(n), ReadData1, ReadData2, ..., ReadData(n)</b></p>	<b>ADSIGRP_SUMUP_READWRITE</b> Die Write-Daten enthalten eine Liste von mehreren, separaten AdsReadWriteReq(IG, IO, readLen, writeLen, writeData) quasi als "Sammel-Schreib/Lesebefehl". Dem Aufrufer wird in den Read-Daten das Ergebnis der Sammelanfrage zurückgereicht. Dabei werden zuerst alle Rückgabewerte und Return-Längen aufgelistet, anschließend folgen die angefragten Daten.
0x0000F083	0x00000000-0xFFFFFFFF= n (Anzahl der internen (Sub-)Befehle)n(max) = 500	R&W	<p>W: (n * ULONG[3]) := <b>IG1, IO1, Len1, IG2, IO2, Len2, ..., IG(n), IO(n), Len(n)</b></p> <p>R: (n * ULONG) + UINT8[Len1] + UINT8[Len2] + ..., + UINT8[Len(n)] := <b>Result1, Result2, ..., Result(n), Data1, Data2, ..., Data(n)</b></p>	<b>ADSIGRP_SUMUP_READEX</b> Die Write-Daten enthalten eine Liste von mehreren, separaten AdsReadReqEx(IG, IO, Len, Data) quasi als "Sammel-Lesebefehl". Dem Aufrufer wird in den Read-Daten das Ergebnis der Sammelanfrage zurückgereicht. Dabei werden zuerst alle Rückgabewerte aufgelistet, anschließend folgen die angefragten Daten.

Index Group	Index Offset	Zugriff	Datentyp	Beschreibung
0x0000F084	0x00000000-0xFFFFFFFF= n (Anzahl der internen (Sub-)Befehle)n(max) = 500	R&W	<p>W: (n * ULONG[3]) := <b>IG1, IO1, Len1,</b> <b>IG2, IO2, Len2,</b> ..., <b>IG(n), IO(n), Len(n)</b></p> <p>R: (n * ULONG) + UINT8[Len1] + UINT8[Len2] + ..., + UINT8[Len(n)] := <b>Result1, Result2, ...,</b> <b>Result(n),</b> <b>Data1, Data2, ...,</b> <b>Data(n)</b></p>	<p><b>ADSIGRP_SUMUP_READEX2</b> Die Write-Daten enthalten eine Liste von mehreren, separaten AdsReadReqEx2(IG, IO, Len, Data) quasi als "Sammel-Lesebefehl". Dem Aufrufer wird in den Read-Daten das Ergebnis der Sammelanfrage zurückgereicht. Dabei werden zuerst alle Rückgabewerte aufgelistet, anschließend folgen die angefragten Daten.</p>

### 3.1.1.5.4 Spezifikation für NC

Diese Dokumentation beinhaltet alle TC3 spezifischen Änderungen und Neuerungen.

Index-Group (Hex)	Beschreibung	Anmerkung
0x1000	Ring-0-Manager: Parameter [▶ 34]	Optional!
0x1100	Ring-0-Manager: Zustand [▶ 35]	Optional!
0x1200	Ring-0-Manager: Funktionen [▶ 35]	Optional!
0x1300	Ring-0-Manager: zyklische Prozessdaten	Nicht implementiert!
0x2000 + ID	Kanal mit entspr. ID: Parameter [▶ 36]	
0x2100 + ID	Kanal mit entspr. ID: Zustand [▶ 40]	
0x2200 + ID	Kanal mit entspr. ID: Funktionen [▶ 43]	
0x2300 + ID	Kanal mit entspr. ID: zyklische Prozessdaten [▶ 46]	
0x3000 + ID	Gruppe mit entspr. ID: Parameter [▶ 47]	Optional!
0x3100 + ID	Gruppe mit entspr. ID: Zustand [▶ 52]	Optional!
0x3200 + ID	Gruppe mit entspr. ID: Funktionen [▶ 58]	Optional!
0x3300 + ID	Gruppe mit entspr. ID: zyklische Prozessdaten	Nicht implementiert!
0x4000 + ID	Achse mit entspr. ID: Parameter [▶ 64]	
0x4100 + ID	Achse mit entspr. ID: Zustand [▶ 78]	
0x4200 + ID	Achse mit entspr. ID: Funktionen [▶ 89]	
0x4300 + ID	Achse mit entspr. ID: zyklische Prozessdaten [▶ 113]	
0x5000 + ID	Encoder mit entspr. ID: Parameter [▶ 118]	Optional!
0x5100 + ID	Encoder mit entspr. ID: Zustand [▶ 123]	Optional!
0x5200 + ID	Encoder mit entspr. ID: Funktionen [▶ 128]	Optional!
0x5300 + ID	Encoder mit entspr. ID: zyklische Prozessdaten [▶ 131]	Optional!
0x6000 + ID	Regler mit entspr. ID: Parameter [▶ 135]	Optional!
0x6100 + ID	Regler mit entspr. ID: Zustand [▶ 139]	Optional!
0x6200 + ID	Regler mit entspr. ID: Funktionen [▶ 142]	Optional!
0x6300 + ID	Regler mit entspr. ID: zyklische Prozessdaten	Nicht implementiert!
0x7000 + ID	Drive mit entspr. ID: Parameter [▶ 143]	Optional!
0x7100 + ID	Drive mit entspr. ID: Zustand [▶ 147]	Optional!
0x7200 + ID	Drive mit entspr. ID: Funktionen [▶ 149]	Optional!
0x7300 + ID	Drive mit entspr. ID: zyklische Prozessdaten [▶ 150]	Optional!
<b>0x0A000 + ID</b>	Tabellen (n x m) mit entspr. ID: Parameter [▶ 153] 0x0A000+ID für Tabellen-ID [1..255] 0x1A000+ID für Tabellen-ID [256..4095] ... 0xFA000+ID für Tabellen-ID [3840..4095]	Maximalanzahl von Tabellen auf 4095 erweitert (ab TC3.1 B4021)
<b>0x0A100 + ID</b>	Tabellen (n x m) mit entspr. ID: Zustand [▶ 157] 0x0000A100+IDLowByte für Tabellen-ID [1..255] 0x0001A100+IdLowByte für Tabellen-ID [256..4095] ... 0x000FA100+IdLowByte für Tabellen-ID [3840..4095] 0x000nA100+IdLowByte für Tabellen-ID [1..4095] (TabID = n * 256 + IdLowByte)	

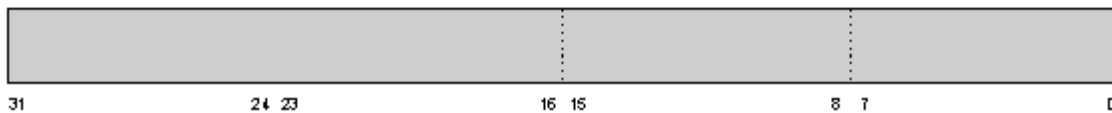


Index-Group (Hex)	Beschreibung	Anmerkung
0x0A200 + ID	Tabellen (n x m) mit entspr. ID: Funktionen [► 158] 0x0000A100+IDLowByte für Tabellen-ID [1..255] 0x0001A100+IDLowByte für Tabellen-ID [256..4095] ... 0x000FA100+IDLowByte für Tabellen-ID [3840..4095] 0x000nA100+IDLowByte für Tabellen-ID [1..4095] (TabID = n * 256 + IdLowByte)	
0x0A300 + ID	Tabellen (n x m) mit entspr. ID: zyklische Prozessdaten 0x0000A100+IDLowByte für Tabellen-ID [1..255] 0x0001A100+IDLowByte für Tabellen-ID [256..4095] ... 0x000FA100+IDLowByte für Tabellen-ID [3840..4095] 0x000nA100+IDLowByte für Tabellen-ID [1..4095] (TabID = n * 256 + IdLowByte)	Nicht implementiert!
0xF000 ... 0xFFFFF	Reservierter Bereich (TwinCAT Systembereich)	
IndexGroup:	IndexOffset:	
0xF081	0x00000000 ... 0xFFFFFFFF (n Elemente)	ADSIGRP_SUMUP_WRITE Das <i>Read-Write-Kommando</i> ist ein Sammelkommando und enthält in den Write-Daten eine Liste von mehreren <i>ADS-Write-Kommandos</i> . Aufbau der Write-Daten: [ IdxGrp(1), IdxOff(1), WriteLen(1), ..., IdxGrp(n), IdxOff(n), WriteLen(n), WriteData(1), ..., WriteData(n) ] Aufbau der Read-Daten: [ Error(1), ..., Error(n) ]
0xF082	0x00000000 ... 0xFFFFFFFF (n Elemente)	ADSIGRP_SUMUP_READWRITE Das <i>Read-Write-Kommando</i> ist ein Sammelkommando und enthält in den Write-Daten eine Liste von mehreren <i>ADS-Read-Write-Kommandos</i> . Aufbau der Write-Daten: [ IdxGrp(1), IdxOff(1), ReadLen(1), WriteLen(1), ..., IdxGrp(n), IdxGrp(n), ReadLen(n), WriteLen(n), WriteData(1), ..., WriteData(n) ] Aufbau der Read-Daten: [ Error(1), ReadLen(1), ..., Error(n), ReadLen(n), ReadData(1), ..., ReadData(n) ]
0xF084	0x00000000 ... 0xFFFFFFFF (n Elemente)	ADSIGRP_SUMUP_READ (READEX2) Das <i>Read-Write-Kommando</i> ist ein Sammelkommando und enthält in den Write-Daten eine Liste von mehreren <i>ADS-Read-Kommandos</i> . Aufbau der Write-Daten: [ IdxGrp(1), IdxOff(1), ReadLen(1), ..., IdxGrp(n), IdxGrp(n), ReadLen(n) ] Aufbau der Read-Daten: [ Error(1), ReadLen(1), ..., Error(n), ReadLen(n), ReadData(1), ..., ReadData(n) ]

Index-Group:



Index-Offset:



### 3.1.1.5.4.1 Spezifikation Ring-0-Manager

#### 3.1.1.5.4.1.1 "Index-Offset" Spezifikation für Ring-0-Parameter (Index-Group 0x1000)

Index-Offset (Hex)	Zugriff	Ring-0-Manager	Datentyp	Phys. Einheit	Definitionsbereich	Beschreibung	Anmerkung
0x00000010	Read	every	UINT32	100 ns		Zykluszeit SAF-Task	
0x00000012	Read	every	UINT32	100 ns		Zykluszeit SVB-Task	
0x00000014	Read	every	INT32	ns		Global Time Compensation Shift (SAF-Task)	
0x00000020	Read /Write	every	UINT16	1	0/1	Zyklische Überwachung und Korrektur der NC-Sollwerte auf Datenkonsistenz	

**3.1.1.5.4.1.2 "Index-Offset" Spezifikation für Ring-0-Zustand (Index-Group 0x1100)**

Index-Offset (Hex)	Zugriff	Ring-0-Manager	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	UINT32	1	0, 1...255	Anzahl der Kanäle	
0x00000002	Read	every	UINT32	1	0, 1...255	Anzahl der Gruppen	
0x00000003	Read	every	UINT32	1	0, 1...255	Anzahl der Achsen	
0x00000004	Read	every	UINT32	1	0, 1...255	Anzahl der Encoder	
0x00000005	Read	every	UINT32	1	0, 1...255	Anzahl der Regler	
0x00000006	Read	every	UINT32	1	0, 1...255	Anzahl der Drives	
0x0000000A	Read	every	UINT32	1	0, 1...255	Anzahl der Tabellen (n x m)	
0x00000010	Read	every	UINT32	1		Zykluszeitfehlerzähler SAF-Task (nicht scopebar)	Reserviert!
0x00000014	Read	every	UINT32	1		IO-Zykluszeitfehlerzähler SAF-Task (nicht scopebar)	Reserviert!
0x00000020	Read	every	UINT32	µs		Rechenzeit SAF-Task (nicht scopebar)	Reserviert!
0x00000031	Read	every	UINT32 [n]	1	0, 1...255	Liefert die Kanal-IDs für sämtliche Kanäle im System	
0x00000032	Read	every	UINT32 [n]	1	0, 1...255	Liefert die Gruppen-IDs für sämtliche Gruppen im System	
0x00000033	Read	every	UINT32 [n]	1	0, 1...255	Liefert die Achs-IDs für sämtliche Achsen im System	
0x00000034	Read	every	UINT32 [n]	1	0, 1...255	Liefert die Encoder-IDs für sämtliche Encoder im System	
0x00000035	Read	every	UINT32 [n]	1	0, 1...255	Liefert die Regler-IDs für sämtliche Regler im System	
0x00000036	Read	every	UINT32 [n]	1	0, 1...255	Liefert die Drive-IDs für sämtliche Drives im System	
0x0000003A	Read	every	UINT32 [n]	1	0, 1...255	Liefert die Tabellen-IDs für sämtliche Tabellen im System	
0x000001nn	Read	every	UINT32	1	0, 1...255	Liefert für die Encoder-ID die zugehörige Achs-IDnn = Encoder-ID	Reserviert!
0x000002nn	Read	every	UINT32	1	0, 1...255	Liefert für die Controller-ID die zugehörige Achs-IDnn = Controller-ID	Reserviert!
0x000003nn	Read	every	UINT32	1	0, 1...255	Liefert für die Drive-ID die zugehörige Achs-IDnn = Drive-ID	Reserviert!

**3.1.1.5.4.1.3 "Index-Offset" Spezifikation für Ring-0-Funktionen (Index-Group 0x1200)**

Index-Offset (Hex)	Zugriff	Ring-0-Manager	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000020	Write	every	VOID	1		Clear Zykluszeitfehlerzähler SAF & SVB	Reserviert!

**3.1.1.5.4.2**      **Spezifikation Kanäle****3.1.1.5.4.2.1**      ***"Index-Offset" Spezifikation für Kanalparameter (Index-Group  
0x2000 + ID)***

Index-Offset (Hex)	Zugriff	Kanaltyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	UINT32	1		Kanal-ID	
0x00000002	Read	every	UINT8[30+1]	1		Kanalname	
0x00000003	Read	every	UINT32	1	ENUM	<u>Kanaltyp</u> [ <a href="#">▶ 158</a> ]	
0x00000004	Read	every	UINT32	1	ENUM	<u>Interpretertyp</u> [ <a href="#">▶ 159</a> ]	
0x00000005	Read	every	UINT32	1		ProgrammładepuffergröÙe in Byte	
0x00000006	Read	every	UINT32	1		Programm-Nr laut Job-Liste	
0x00000007	Read/Write	every	UINT32	1	ENUM	Setze <u>Lade-Logmodus</u> [ <a href="#">▶ 159</a> ]	
0x00000008	Read/Write	every	UINT32	1	ENUM	Setze <u>Trace-Modus</u> [ <a href="#">▶ 159</a> ]	
0x00000009	Read/Write	every	UINT32	1		RESERVIERT	
0x0000000A	Read/Write	every	UINT32	1	0/1	Protokolliert alle Feeder-Einträge in einer Log-Datei mit dem Namen "TcNci.log"	
0x0000000B	Read/Write	every	UINT32	1	0/1	Kanalspezifischer Level für NC Logger Messages 0: nur Fehler 1: alle NC-Meldungen	
0x00000010	ReadWrite	every	<b>Write</b> { UINT32 1 0..159 Startindex der M-Fkt. UINT32 1 1..160 Anzahl der zu lesenden M-Fkt. } <b>Read [n]</b> { UINT8 1 0..159 Regelbit-Maske der M-Fkt. INT32[10] 1 -1..159 Nr. der abzulöschenden M-Fkt. }				
0x00000011	Write	Interpolation				Schreibe M-Funktionsbeschreibung	Nur interne Verwendung!
0x00000012	Read/Write	Interpolation	LREAL64	1		Faktor für G70	
0x00000013	Read/Write	Interpolation	LREAL64	1		Faktor für G71	
0x00000014	Write	Interpolation	{ char[32] Benutzersymbol (nullterminiert) char[10] Systemsymbol (nullterminiert) }			Benutzersymbole für Achsen	Noch nicht freigegeben
0x00000015	Read/Write	Interpolation	UINT16 bzw. UINT32	1	0/1Default: FALSE	Aktivierung von Default G-Code	NEU ab TC3.1 B4014
0x00000021	Read	every	UINT32	1		Gruppen-ID (nur eindeutig für 3D- und FIFO-Kanal)	
0x00000031	Read/Write	Interpolation	UINT16	1		Standard-Output-Port des Interpreters	Reservierte Funktion, kein Standard!
0x00000032	Read/Write	Interpolation	UINT16	1	0/1	Cartesian tool offset entry	Reservierte Funktion, kein Standard!

Index-Offset (Hex)	Zugriff	Kanaltyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000040	Read/Write	Interpolation	{			<b>Zieladresse des Interpreter Hooks</b>	Reservierte Funktion, kein Standard!
			char[6]			Ams Net ID	
			UINT16			Port	
			UINT32			Index Group	
			UINT32			Index Offset	
}							
0x00000050	Read/Write	Interpolation	UINT32	1	ENUM	Reaktion, wenn bei der Radiuskorrektur ein Flaschenhals erkannt wird  0: Fehler und Abbruch 1: Hinweis & Behebung 2: Nur Hinweis, ohne Konturanpassung	
0x00000051	Read/Write	Interpolation	UINT32	1	1..24	Look Ahead für die Flaschenhalserkennun g	
0x00000052	Read/Write	Interpolation	UINT32	1	0/1	Fase an/aus	Reservierte Funktion, kein Standard!
0x00000053	Read/Write	Interpolation	UINT32	1		Aktivierung zum Lesen der aktuell wirkenden Interpolationsregeln, Nullpunktverschiebung en & Rotation  0: aus 1: ein	
0x00000054	Read/Write	Interpolation	UINT32	1	0/1	Retrace an/aus	Reservierte Funktion, kein Standard!
0x00000055	Read/Write	Interpolation	UINT32[4]	1		Konfiguration des zyklischen Kanalinterface für UINT32 max. 4 Index Offsets können konfiguriert werden.	
0x00000056	Read/Write	Interpolation	UINT32[4]	1		Konfiguration des zyklischen Kanalinterface für LREAL max. 4 Index Offsets können konfiguriert werden.	
0x00010K0L	Read/Write	every	REAL64	z. B. mm	±MAX REAL64	Wert für Nullpunktverschiebung (NPV)	
					[1..3]	Index der Achse K=1 → X K=2 → Y K=3 → Z	
					[1..0xA]	L=1 → G54F L=2 → G54G L=3 → G55F ...	
0x0002ww00	Read/Write	every	UINT16			Tool-Nummer: Werte für Werkzeugkorrektur	
0x0003ww00	Read/Write	every	UINT16		[1...50]	Tool-Typ: ww = Werkzeug 1...50	
0x0004wwnn	Read/Write	every	REAL64		[1...14]	Parameter: nn = Index 1...14	

Index-Offset (Hex)	Zugriff	Kanaltyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x000500gg	Read/Write	every	REAL64	z. B. mm	$\geq 0$ (Wert) [1...9] (g)	Radius der Toleranzkugel gg = Gruppe des Kanals (Default: 1)	

**3.1.1.5.4.2.2**     *"Index-Offset" Spezifikation für Kanalzustand (Index-Group 0x2100 + ID)*



Index-Offset (Hex)	Zugriff	Kanaltyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	INT32	1	ENUM	Fehlercode Kanal	
0x00000002	Read	every	UINT32	1		Anzahl Gruppen im Kanal	
0x00000003	Read	every	UINT32	1	ENUM	<u>Interpreterstatus</u> [►_159]	Nicht oszilloskopierbar!
0x00000004	Read	every	UINT32	1	ENUM	<u>Interpreter-Betriebsart</u> [►_159] (interpreter/ channel operation mode)	
0x00000005	Read	every	UINT32	1		Aktuell geladenes Programm	
0x00000007	Read	every	UINT8[...]	1		Programmname des aktuell geladenen Programms (100 Zeichen, Null-Terminiert)	Max. 100 Zeichen, nullterminiert
0x00000008	Read	Interpreter	UINT32	1	[0,1]	Interpreter-Simulationsmode 0: off (default) 1: on	Nicht oszilloskopierbar!
0x00000010	Read	Interpreter	UINT32	1		Textindex Falls sich der Interpreter im Aborted-Status befindet, kann hiermit der aktuelle Textindex ausgelesen werden	Nicht oszilloskopierbar!
0x00000011	Read/Write	Interpreter	<b>Write</b>				Nicht oszilloskopierbar!
			UINT32	1		Textindex	
			<b>Read</b>				
			UINT8[...]	1		Zeile des NC-Teilprogramms ab dem Textindex	
0x00000012	Read	Interpreter	{				
			UINT32	1		Aktuelle Anzeige für 1: SAF 2: Interpreter 3: Fehleroffset	
			UINT32	1		Fileoffset	
			UINT8[260]	1		Pfad + Programmname	
			}				
0x00000013	Read	Interpreter	UINT32[18]			Anzeige für aktuell wirkenden G-Code	
0x00000014	Read	Interpreter	{			Ermittelt die aktuell wirkende Nullpunktverschiebung	
			UINT32	1		Satzzähler	
			UINT32			Dummy	
			LREAL[3]	1		Nullpunktverschiebung G54..G57	
			LREAL[3]	1		Nullpunktverschiebung G58	
			LREAL[3]	1		Nullpunktverschiebung G59	
			}				

Index-Offset (Hex)	Zugriff	Kanaltyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000015	Read	Interpreter	{			Ermittelt die aktuell wirkende Rotation	
			UINT32	1		Satzzähler	
			UINT32	1		Dummy	
			LREAL[3]	1		Rotation in Grad von X, Y & Z	
			}				
0x00000016	Read	Interpreter	UINT32	1	[0,1]	Feeder-Info	Nur interne Verwendung! Kein Standard
0x00000100	Read	every	UINT32 [n]	1	[0, 1...255]	Liefert die jeweiligen Achs-IDs im Kanal Anzahl: [1...255] Achs-ID's: [0, 1...255]	Nicht oszilloskopierbar!!

**3.1.1.5.4.2.3**     *"Index-Offset" Spezifikation für Kanalfunktionen (Index-Group  
0x2200 + ID)*

Index-Offset (Hex)	Zugriff	Kanaltyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Write	every	UINT32	1		Lade NC-Programm per Programmnummer	
0x00000002	Write	every	VOID			Starte Interpreter	
0x00000003	Write	every	VOID			RESERVIERT	
0x00000004	Write	every	UINT8[...]			Lade NC-Programm per Programmnamen. Der Standard NC-Pfad muss nicht (darf aber) mit angegeben werden. Auch andere Pfade sind zulässig.	
0x00000005	Write	every	UINT16	ENUM	s. Anhang <a href="#">Interpreter-Betriebsarten</a> [ <a href="#">ID_159</a> ]	Setze Interpreter-Betriebsart (interpreter/channel operation mode)	
0x00000006	Write	Interpreter	UINT8[...]			Setze Pfad für Unterprogramme	
0x00000008	Write	Interpreter	UINT32	1		Interpreter-Simulationsmode 0: off (default) 1: on	Noch nicht freigegeben
0x0000000F	Write	every	VOID			RESERVIERT	
0x00000010	Write	every	VOID			"Reset" Kanal	
0x00000011	Write	every	VOID			"Stopp" Kanal	
0x00000012	Write	every	VOID			"Retry" Kanal (Wiederanlauf Kanal)	
0x00000013	Write	every	VOID			"Skip" Kanal (Überspringe Auftrag/Satz)	
0x00000014/0x00000015	Write	every	{			"Enable Retrace" / "Disable Retrace"	Reservierte Funktion, kein Standard!
			UINT32	1	>0	Feeder-Abarbeitungsrichtung: 1: vorwärts 2: rückwärts	
			UINT32	1	≥ 0	Entry-Index	
			REAL64[3]	mm	±∞	Pos. der Hauptachsen X, Y, Z	
			REAL64[5]	mm	±∞	Pos. der Hilfsachsen Q1, ..., Q5	
			}				
0x00000020	Write	every	VOID			"Save" Nullpunktverschiebung (NPV)	
0x00000021	Write	every	VOID			"Load" Nullpunktverschiebung (NPV)	
0x00000022	Write	every	VOID			"Save" Werkzeugkorrekturen	
0x00000023	Write	every	VOID			"Load" Werkzeugkorrekturen	

Index-Offset (Hex)	Zugriff	Kanaltyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000024	Write	Interpolation	{			Speichert Snapshot des Interpreters in eine angegebene Datei	
			char[32]			Dateiname im TwinCAT\CNC-Verzeichnis	
			UINT32	1	0..1	Maske: 0x1: R-Parameter 0x2: Nullpunktverschiebung 0x4: Werkzeugbeschreibungen	
			}				
0x00000025	Write	Interpolation	{			Liest Snapshot aus einer angegebenen Datei in den Interpreter ein	
			char[32]			Dateiname im TwinCAT\CNC-Verzeichnis	
			UINT32	1	0..1	Maske: 0x1: R-Parameter 0x2: Nullpunktverschiebung 0x4: Werkzeugbeschreibungen	
			}				
0x00000026	Write	Interpolation	VOID			Setzt alle Werkzeugparameter (inkl. Type und Nummer) auf Null	
0x00000027	Write	Interpolation	VOID			Setzt alle Nullpunktverschiebungen auf Null	
0x00000030	Write	every	VOID			Wiederanlauf (Go Ahead) des Interpreters nach programmierten Interpreterstopp	
0x00000040	Write	every	VOID			Triggerevent zum in der NCI	
0x00000041	Write	every				RESERVIERT für Messereignis	
0x00000050	Write	Interpolation	VOID	1		Setzt ExecIdleInfo im Interpreter	Reservierte Funktion, kein Standard!
0x00000051	Write	Interpolation	UINT32	1		Setzt Satzunterdrückungsmaske im Interpreterparameter: SkippingMask	Reservierte Funktion, kein Standard!
0x00000052	Write	Interpolation	UINT32	1		Setzt ItpOperationMode im Interpreterparameter: Maske des OperationModes	Reservierte Funktion, kein Standard!
0x00000053	Write	Interpolation	VOID			Setzt ScanningFlag im NC Device	Reservierte Funktion, kein Standard!

Index-Offset (Hex)	Zugriff	Kanaltyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000054	Write	Interpolation				<b>ScanPosition</b>	Reservierte Funktion, kein Standard!
			double[8]			Position	
0x00000055	Write	Interpolation				Reserviert	
0x00000056	Write	Interpolation	VOID			Setzt Interpreter in Aborted-Status	Reservierte Funktion, kein Standard!
0x00000060	Write	Interpolation	UINT16	1	0..159	Manuelles Zurücksetzen einer schnellen M-Funktion	

### 3.1.1.5.4.2.4 "Index-Offset" Spezifikation für zyklische Kanalprozessdaten (Index-Group 0x2300 + ID)

Index-Offset (Hex)	Zugriff	Kanaltyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000000	Read	every (PLC→NC)	{128 Byte}		STRUCT s. Kanal-Interface	KANAL-STRUKTUR (PLC→NC) Anm.: Größe und Alignment geändert.	Die aktuelle zugehörige PLC Struktur ist: NciChannelFromPlc PLCTONC_NC/ CHANNEL_REF
0x00000001	Read	every	UINT8[...] min. 30 Byte	1		Interpreter-Programmanzeige	Nicht oszilloskopierbar!
0x00000002	Read/Write	every (PLC→NC)	UINT32	%	[0...1000000]	Geschwindigkeitsoverride Kanal (Achsen im Kanal)	1000000 = 100%
0x00000003	Read/Write	every (PLC→NC)	UINT32	%	[0...1000000]	Geschwindigkeitsoverride Spindel	1000000 = 100%
0x00000080	Read	every (NC→PLC)	{160 Byte}		STRUCT s. Kanal-Interface	KANAL-STRUKTUR (NC→PLC) Anm.: Größe und Alignment geändert.	Die aktuelle zugehörige PLC Struktur ist: NciChannelToPlc NCTOPLC_NC/ CHANNEL_REF
0x10000000 +RegIndex	Read/Write	every	REAL64	1	[0...999]	R-Parameter des Interpreters	Nicht oszilloskopierbar!
0x20000001	Read	every	UINT8[...] min. 30 Byte	1	[1...9]	Programmanzeige der Gruppenabarbeitung	Nicht oszilloskopierbar!

**3.1.1.5.4.3 Spezifikation Gruppen****3.1.1.5.4.3.1 *"Index-Offset" Spezifikation für Gruppenparameter (Index-Group 0x3000 + ID)***

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	UINT32	1		Gruppen-ID	
0x00000002	Read	every	UINT8[30+1]	1		Gruppenname	
0x00000003	Read	every	UINT32	1	ENUM	<a href="#">Gruppentyp [► 159]</a>	
0x00000004	Read	every	UINT32	µs		SAF-Zykluszeit Gruppe	
0x00000005	Read	every	UINT32	µs		SVB-Zykluszeit Gruppe	
0x00000006	Read/Write	every	UINT16	1	0/1	Einzelsatz-Betriebsart?	
0x0000000B	Read	every	UINT32	1		Größe der SVB-Tabelle (max. Anzahl von SVB-Einträgen)	
0x0000000C	Read	every	UINT32	1		Größe der SAF-Tabelle (max. Anzahl von SAF-Einträgen)	
0x00000010	Read/Write	every	UINT32	1	[1, 2 ... 32] Default: 1	Interner SAF-Zykluszeit Divisor (dividiert die interne SAF-Zykluszeit um diesen Faktor)	z. B. für DXD-Gruppe
0x00000021	Read	Kanal: every	UINT32	1		Kanal-ID	
0x00000022	Read	Kanal: every	UINT8[30+1]	1		Kanalname	
0x00000023	Read	Kanal: every	UINT32	1	ENUM	<a href="#">Kanaltyp [► 158]</a>	
0x00000024	Read	Kanal: every	UINT32	1	>0	Nummer im Kanal	
0x00000500	Read/Write	DXD-Gruppe	INT32	ENUM	[0, 1]	<a href="#">Kurvengeschwindigkeitsreduktionsmethode [► 159]</a> 0: Coulomb-Scattering 1: Cosinus-Gesetz 2: VeloJump	
0x00000501	Read/Write	DXD-Gruppe	REAL64	1	[0.0...1.0]	Geschwindigkeitsreduktionsfaktor C0-Übergang (stetiger Verlauf, aber weder einmal noch zweimal stetig differenzierbar)	
0x00000502	Read/Write	DXD-Gruppe	REAL64	1	[0.0...1.0]	Geschwindigkeitsreduktionsfaktor C1-Übergang (stetiger Verlauf und einmal stetig differenzierbar)	
0x00000503	Read/Write	DXD-Gruppe	REAL64	Grad	[0.0...180.0]	Kritischer Winkel am Segmentübergang "Low" (muss echt kleiner gleich dem Geschwindigkeitsreduktionswinkel C0 sein)	
0x00000504	Read/Write	DXD-Gruppe	REAL64	Grad	[0.0...180.0]	Kritischer Winkel am Segmentübergang "High" (muss echt kleiner gleich dem Geschwindigkeitsreduktionswinkel C0 sein)	
0x00000505	Read/Write	DXD-Gruppe	REAL64	mm/s	≥ 0	Mindestgeschwindigkeit, die an Segmentübergängen trotz möglicher Geschwindigkeitsreduktion nicht unterschritten werden darf.	Achtung: Parameter wird nicht in der Solution gespeichert und nicht als NC-Boot-Parameter übertragen!



Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000506	Read/Write	DXD-Gruppe	REAL64	z. B. mm	[0.0...1000.0]	Radius der Toleranzkugel für Verschleifungen	Nicht implementiert!
0x00000507	Read/Write	DXD-Gruppe	REAL64	1		Geschwindigkeitsreduktionsfaktor C2-Übergang	
0x00000508	Read/Write	DXD-Gruppe	UINT16	1	0/1	Aktiviert Berechnung der totalen verbleibenden Bahnlänge	NEU ab TC3.1 B4020.40
0x00000509	Read/Write	DXD-Gruppe	UINT16	1	0/1 Default: 1	Allgemeine Aktivierung der Software-Endlagenüberwachung für die Hauptachsen (X, Y, Z) (s. Encoderparameter)	
0x0000050A	Read/Write	DXD-Gruppe	UINT32	1	0/1	NCI Override 0: Bezogen auf interne reduzierte Geschwindigkeit (ohne Iteration) 1: Bezogen auf originale externe (programmierte) Geschwindigkeit 2: Bezogen auf interne reduzierte Geschwindigkeit (0 ... >100%)	
0x0000050C	Read	DXD-Gruppe	UINT32	1	[128 ... 1024] Default: 128	Benutzerdefinierte Maximalanzahl der NCI-SAF-Tabelleneinträge	NEU ab TC3.1 B4014 Boot-Parameter
0x00000510	Read/Write	DXD-Gruppe	REAL64	1	≥ 0	Für Reduktionsmethode VeloJump Reduktionsfaktor für C0-Übergänge: X-Achse	Nicht implementiert!
0x00000511	Read/Write	DXD-Gruppe	REAL64	1	≥ 0	Für Reduktionsmethode VeloJump Reduktionsfaktor für C0-Übergänge: Y-Achse	Nicht implementiert!
0x00000512	Read/Write	DXD-Gruppe	REAL64	1	≥ 0	Für Reduktionsmethode VeloJump Reduktionsfaktor für C0-Übergänge: Z-Achse	Nicht implementiert!
0x00000513	Read/Write	DXD-Gruppe	LREAL64	1	]0.0..1.0[	Verschleifung für Hilfsachsen: Ist die resultierende Bahngeschwindigkeit kleiner als die prog. mal diesem Faktor, so wird ein Genauhalt eingefügt	Noch nicht freigegeben
0x00000514	Read/Write	DXD-Gruppe	UINT32	1	[1 ... 20] Default: 1	Maximale Anzahl von zu übertragenden Kommandos pro NC-Zyklus (von SVB zu SAF)	NEU ab TC3.1 B4020.40

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000604	Read/Write	Encodergruppe	REAL64	z. B. mm/s	[0.0...1000.0]	Geschwindigkeitsfenster bzw. Stillstandsfenster	Base Unit / s
0x00000605	Read/Write	Encodergruppe	REAL64	s	[0.0...60.0]	Filterzeit für Stillstandsfenster in Sekunden	
0x00000606	Read/Write	Encodergruppe	REAL64	s	[0.0...60.0]	Totzeitkompensation Master/Slave-Kopplung ("Winkelvorsteuerung")	
0x00000701	Read	FIFO-Gruppe	UINT32	1	[1...16]	FIFO-Dimension (m = Anzahl der Achsen) Anm.: Die FIFO-Dimension ist auf 16 erhöht worden.	(n x m)-FIFO Boot-Daten!
0x00000702	Read	FIFO-Gruppe	UINT32	1	[1...10000]	FIFO-Größe (Länge) (n = Anzahl der FIFO-Einträge)	(n x m)-FIFO Boot-Daten!
0x00000703	Read	FIFO-Gruppe	UINT32	1	[0, 1, 4]	Interpolationstyp für FIFO-Sollwertgenerator 0: INTERPOLATIONTYPE_LINEAR (Default) 1: INTERPOLATIONTYPE_4POINT 4: INTERPOLATIONTYPE_CUBICSPLINE (with 6 points)	NEU ab TC3.1 B4020
0x00000704	Read/Write	FIFO-Gruppe	UINT32	1	[1, 2]	Override typ für FIFO-Sollwertgenerator Typ 1: OVERRIDETYPE_INSTANTANEOUS (Default) Typ 2: OVERRIDETYPE_PT2	
0x00000705	Read/Write	FIFO-Gruppe	REAL64	s	> 0.0	P-T2-Zeit für Overrideänderung (T1=T2=T0)	
0x00000706	Read/Write	FIFO-Gruppe	REAL64	s	≥ 0.0	Zeitdelta für zwei aufeinanderfolgende FIFO-Einträge (Zeitbasis der FIFO-Einträge)	

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000801	ReadWrite	Kinematik-Gruppe	<b>Write</b>			Berechnung der kinematischen Hintransformation für die Positionen (ACS -> MCS)	
			{				
			REAL64[8]	z. B. Grad	$\pm\infty$	Positionen der ACS-Achsen (Axis Coordinate System), max. Dimension: 8	
			UINT32	1	$\geq 0$	Reserve	
			UINT32	1	$\geq 0$	Reserve	
			}				
			<b>Read</b>				
			{				
			REAL64[8]	z. B. mm	$\pm\infty$	Positionen der MCS-Achsen (Machine Coordinate System), max. Dimension: 8	
			UINT32	1	$\geq 0$	Reserve	
			UINT32	1	$\geq 0$	Reserve	
			}				
0x00000802	ReadWrite	Kinematik-Gruppe	<b>Write</b>			Berechnung der kinematischen Rücktransformation für die Positionen (MCS -> ACS)	
			{				
			REAL64[8]	z. B. mm	$\pm\infty$	Positionen der MCS-Achsen (Machine Coordinate System), max. Dimension: 8	
			UINT32	1	$\geq 0$	Reserve	
			UINT32	1	$\geq 0$	Reserve	
			}				
			<b>Read</b>				
			{				
			REAL64[8]	z. B. Grad	$\pm\infty$	Positionen der ACS-Achsen (Axis Coordinate System), max. Dimension: 8	
			UINT32	1	$\geq 0$	Reserve	
			UINT32	1	$\geq 0$	Reserve	
			}				

**3.1.1.5.4.3.2**     ***"Index-Offset" Spezifikation für Gruppenzustand (Index-Group  
0x3100 + ID)***

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	INT32	1	ENUM	Fehlercode Gruppe	
0x00000002	Read	every	UINT32	1		Anzahl Masterachsen	
0x00000003	Read	every	UINT32	1		Anzahl Slaveachsen	
0x00000004	Read	every	UINT32	1	s. ENUM	SVB-Gruppenstatus (Zustand)	
0x00000005	Read	every	UINT32	1	s. ENUM	SAF-Gruppenstatus (main state)	
0x00000006	Read	every	UINT32	1	s. ENUM	Bewegungszustand (Zustand)	
0x00000007	Read	every	UINT32	1	s. ENUM	SAF-Sub-Gruppenstatus (sub state)	
0x00000008	Read	every	UINT32	1	s. ENUM	Referenzierstatus (Zustand)	
0x00000009	Read	every	UINT32	1	s. ENUM	Koppelstatus (Zustand)	Nicht oszilloskopierbar!
0x0000000A	Read	every	UINT32	1	≥0	Koppeltabellen-Index	Nicht oszilloskopierbar!
0x0000000B	Read	every	UINT32	1	≥0	Aktuelle Anzahl SVB-Einträge/Aufträge	<i>Symbolischer Zugriff: 'SvbEntries' (DXD)</i>
0x0000000C	Read	every	UINT32	1	≥0	Aktuelle Anzahl SAF-Einträge/Aufträge	<i>Symbolischer Zugriff: 'SafEntries' (DXD)</i>
0x0000000D	Read	every	UINT32	1		Aktuelle Satznummer (nur für Interpolationsgruppe aktiv)	<i>Symbolischer Zugriff: 'BlockNumber' (DXD)</i>
0x0000000E	Read	every	UINT32	1	≥0	Aktuelle Anzahl freier SVB-Einträge/Aufträge	Nicht oszilloskopierbar!
0x0000000F	Read	every	UINT32	1	≥0	Aktuelle Anzahl freier SAF-Einträge/Aufträge	Nicht oszilloskopierbar!
0x00000011	Read	every	UINT16	1	0/1	Emergency Stop (E-Stop) aktiv?	Nicht oszilloskopierbar!

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000110	Read	PTP-Gruppe	{			Interne NC- Informationen (Auflösungen)	Reserviert!
			REAL64	z. B. mm	$\pm \infty$	ExternalEndPosition	
			REAL64	z. B. mm/s	>0	ExternalTargetVelocity	
			REAL64	z. B. mm/s <sup>2</sup>	>0	ExternalAcceleration	
			REAL64	z. B. mm/s <sup>2</sup>	>0	ExternalDeceleration	
			REAL64	z. B. mm/s <sup>3</sup>	>0	ExternalJerk	
			UINT32	1	>0	ExternalOverrideType	
			REAL64	z. B. mm	$\pm \infty$	InternalEndPosition	
			REAL64	z. B. mm/s	>0	InternalTargetVelocity (refers to 100 %)	
			REAL64	%	[0 ... 100]	InternalActualOverride	
			REAL64	z. B. mm/s <sup>2</sup>	>0	InternalAcceleration	
			REAL64	z. B. mm/s <sup>2</sup>	>0	InternalDeceleration	
			REAL64	z. B. mm/s <sup>3</sup>	>0	InternalJerk	
			REAL64	z. B. mm	>0	PositionResolution	
			REAL64	z. B. mm/s	$\geq 0$	VelocityResolution	
			REAL64	z. B. mm/s <sup>2</sup>	$\geq 0$	AccelerationResolutio n	
REAL64	z. B. mm/s	$\geq 0$	VelocityResolutionAtA ccelerationZero				
}							
0x00000500	Read	DXD-Gruppe	REAL64	z. B. mm	$\geq 0$	Bahnrestweg (verbleibende Bogenlänge) auf dem aktuellen Bahnsegment	<i>Symbolischer Zugriff: 'SetPathRemLe ngth'</i>
0x00000501	Read	DXD-Gruppe	REAL64	z. B. mm	$\geq 0$	Abgefahrte Bogenlänge auf dem aktuellen Bahnsegment	<i>Symbolischer Zugriff: 'SetPathLength'</i>
0x00000502	Read	DXD-Gruppe	REAL64	z. B. mm/s	$\geq 0$	Aktuelle Bahn- Sollgeschwindigkeit	<i>Symbolischer Zugriff: 'SetPathVelo'</i>
0x00000503	Read	DXD-Gruppe	REAL64	z. B. mm/s <sup>2</sup>	$\pm \infty$	Aktuelle Bahn- Sollbeschleunigung	<i>Symbolischer Zugriff: 'SetPathAcc'</i>
0x00000504	Read	DXD-Gruppe	REAL64	z. B. mm/s <sup>2</sup>	$\geq 0$	Betrag der aktuellen vektoriellen Sollbeschleunigung	<i>Symbolischer Zugriff: 'SetPathAbsAcc '</i>
0x00000505	Read	DXD-Gruppe	REAL64	z. B. mm/s	$\geq 0$	Maximale Segmentend-Bahn- Sollgeschwindigkeit	<i>Symbolischer Zugriff: 'SetPathVeloEn d'</i>
0x00000506	Read	DXD-Gruppe	REAL64	z. B. mm/s	$\geq 0$	Segmentmaximalbahn sollgeschwindigkeit	<i>Symbolischer Zugriff: 'SetPathVeloMa x'</i>
0x00000507	Read	DXD-Gruppe	REAL64	z. B. mm	$\geq 0$	Aktueller relativer Bremsweg bezogen auf die aktuelle Bogenlänge	<i>Symbolischer Zugriff: 'SetPathStopDi st'</i>
0x00000508	Read	DXD-Gruppe	REAL64	z. B. mm	$\pm \infty$	Sicherheitsabstand = Segmentbogenlänge - aktuelle Bogenlänge - relativer Bremsweg	<i>Symbolischer Zugriff: 'SetPathSecurit yDist'</i>

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsreich	Beschreibung	Anmerkung
0x00000509	Read	DXD-Gruppe	REAL64	1	0/1	Segmentübergang	<i>Symbolischer Zugriff: 'SetPathSegmentChange'</i>
0x0000050A	Read	DXD-Gruppe	REAL64	%	[0 ... 100]	Bahngeschwindigkeits override	<i>Symbolischer Zugriff: 'SetPathOverride'</i>
0x00000511	Read	DXD-Gruppe	REAL64	z. B. mm/s	≥ 0	Betrag der Bahn-Istgeschwindigkeit	<i>Symbolischer Zugriff: 'ActPathAbsVelocity'</i>
0x00000512	Read	DXD-Gruppe	REAL64	z. B. mm/s <sup>2</sup>	± ∞	Bahn-Istbeschleunigung auf aktuellem Segment	<i>Symbolischer Zugriff: 'ActPathAcc'</i>
0x00000513	Read	DXD-Gruppe	REAL64	z. B. mm/s <sup>2</sup>	≥ 0	Betrag der Bahn-Istbeschleunigung auf aktuellem Segment	<i>Symbolischer Zugriff: 'ActPathAbsAcc'</i>
0x00000514	Read	DXD-Gruppe	REAL64	z. B. mm	± ∞	Positionsfehler auf der Bahn in tangentialer Richtung (mit Vorzeichen für Vor- und Nacheilen)	<i>Symbolischer Zugriff: 'PathDiffTangential'</i>
0x00000515	Read	DXD-Gruppe	REAL64	z. B. mm	≥ 0	Positionsfehler auf der Bahn in orthogonaler Richtung	<i>Symbolischer Zugriff: 'PathDiffOrthogonal'</i>
0x00000520	Read	DXD-Gruppe	REAL64	1	≥ 0	Abgefahrte Bogenlänge des aktuellen Segmentes (normiert auf 1.0)	
0x00000521	Read	DXD-Gruppe	REAL64	1	0/1	Teilsegmentwechsel (Radius der Toleranzkugel)	
0x00000522	Read	DXD- Gruppe	REAL64	1	≥ 0	Gesamter Bahnrestweg bis zum letzten Geometrieintrag oder zum nächsten Genauhalt. Bezieht sich auf Gruppenparameter 0x508.	
0x00000523	Read	DXD- Gruppe	REAL64	1	≥ 0	Programmierte Geschwindigkeit des aktuellen Segments	
0x00000524	Read	DXD-Gruppe	REAL64	z. B. mm	≥ 0	Zurückgelegter Bahnweg (Bogenlänge) seit Programmstart	ab TC 3.1 B4022.31 ab TC 3.1 B4024.0
0x00000530	Read	DXD-Gruppe				Aktuelle bzw. letzte MCS-Zielposition der Hauptachsen X, Y und Z	
			REAL64	z. B. mm	± ∞	Zielposition X-Achse	
			REAL64	z. B. mm	± ∞	Zielposition Y-Achse	
			REAL64	z. B. mm	± ∞	Zielposition Z-Achse	
0x00000531	Read	DXD-Gruppe	{			Aktuelle bzw. letzte MCS-Zielposition der Hilfsachsen Q1 bis Q5	
			REAL64[5]	z. B. mm	± ∞	Zielposition der Q1- bis Q5-Achse	
			}				

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsreich	Beschreibung	Anmerkung
0x00000532	Read	DXD-Gruppe	{			Lesen der Bahnlänge, H-Parameter und Entry ID der nächsten 11 Segmente bezogen auf die aktuelle DC-Time	nicht allg. freigegeben
			UINT32			DC Time	
			UINT32			Reserved	
			PreViewTab[11]			11*24 Bytes	
			}				
			PreViewTab				
			{				
			REAL64	z. B. mm		Segmentlänge	
			UINT32	1		Blocknummer	
			UINT32	1		H-Parameter	
			UINT32	1		Entry ID	
UINT32	1		Reserved				
}							
0x0000054n	Read	DXD-Gruppe	REAL64	1	0/1	Innerhalb der Toleranzkugel der Hilfsachse n = 1..5  Nummer der Hilfsachse (nicht Achs-ID)	
0x00000546	Read	DXD-Gruppe	REAL64[8]	z. B. mm	$\pm \infty$	Sollpositions-Array der (3+5) Achsen der 3D-Gruppe	ab TC3.1 B4022.17
0x00000547	Read	DXD-Gruppe	REAL64[8]	z. B. mm	$\pm \infty$	Istpositions-Array der (3+5) Achsen der 3D-Gruppe	ab TC3.1 B4022.17
0x00000548	Read	DXD-Gruppe	REAL64[8]	z. B. mm	$\pm \infty$	Positionsdifferenz (Soll-Ist) bzw. Schleppabstand als Array der (3+5) Achsen der 3D-Gruppe	ab TC3.1 B4022.17
0x00000550	Read	DXD-Gruppe	{			Lesen der Achs-IDs innerhalb der 3D-Gruppe:	
			UINT32	1	[0, 1...255]	X-Achsen ID	
			UINT32	1	[0, 1...255]	Y-Achsen ID	
			UINT32	1	[0, 1...255]	Z-Achsen ID	
}							
0x00000552	Read	DXD-Gruppe FIFO-Gruppe Kinematik-Gruppe	{ UINT32[m] }	1	[0, 1...255]	Achsbelegung der Gruppe:  1. Achs-ID. ..., m.-Achs-ID  m: Dimension der 3D-Gruppe mit Haupt- und Zusatzachsen (X, Y, Z, Q1, Q2, Q3, Q4, Q5) bzw. der FIFO-Gruppe bzw. die ACS-Achsen der Kinematik-Gruppe	



Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000553	Read	Kinematik-Gruppe	{			Lesen der Achsbelegung (IDs) innerhalb der Kinematik-Gruppe:	
			UINT32[8]	1	[0, 1...255]	MCS-Achsen-IDs (Machine Coordinate System)	
			UINT32[8]	1	[0, 1...255]	ACS-Achsen-IDs (Axis Coordinate System)	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve (NEW)	
			}				
0x0000056n	Read	DXD- Gruppe	REAL64	1	± ∞	Aktueller Positionsfehler der Hilfsachse innerhalb der Toleranzkugel (nur sollwertseitig)  Nur für Hilfsachsen n = 1..5  Nummer der Hilfsachse (nicht Achs-ID)	

**3.1.1.5.4.3.3**     ***"Index-Offset" Spezifikation für Gruppenfunktionen (Index-Group  
0x3200 + ID)***

Index-Offset (Hex)	Zugriff	Gruppentyp	Daten- typ	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Write	every	VOID			Reset Gruppe	
0x00000002	Write	every	VOID			Stop Gruppe	
0x00000003	Write	every	VOID			Clear Gruppe (Buffer/ Auftrag)	
0x00000004	Write	PTP-Gruppe, 3D-Gruppe	{			Emergency Stop (E-Stop) (Notstopp mit geregelter Rampe)	
			REAL64	z. B. mm/s <sup>2</sup>	≥ 0.0	Verzögerung (muss größer gleich der Originalverzögerung sein)	
			REAL64	z. B. mm/s <sup>3</sup>	≥ 0.0	Ruck (muss größer gleich dem Originalruck sein)	
			}				
0x00000005	Write	PTP-Gruppe	{			Parametrierbarer Stopp (mit geregelter Rampe)	Reservierte Funktion, kein Standard!
			REAL64	z. B. mm/s <sup>2</sup>	≥ 0.0	Verzögerung	
			REAL64	z. B. mm/s <sup>3</sup>	≥ 0.0	Ruck	
			}				
0x00000006	Write	PTP-Gruppe, 3D-Gruppe	VOID			Weiterfahren ("Step on") nach Emergency-Stop (E-Stop)	
0x00000050	Write	PTP-Gruppe 3D-Gruppe	{			Achsbelegung der Gruppe:	
			UINT32	1	[0, 1...255]	X-Achsen-ID	
			UINT32	1	[0, 1...255]	Y-Achsen-ID	
			UINT32	1	[0, 1...255]	Z-Achsen-ID	
			}				
0x00000051	Write	PTP-Gruppe 3D-Gruppe FIFO-Gruppe	{			Achsbelegung der Gruppe:	
			UINT32	1	[1...255]	Achsen-ID	
			UINT32	1	[0 ... (m-1)]	Platzindex der Achse in der Gruppe m: Gruppen-Dimension (PTP: 1; DXD: 3, FIFO: 16)	
			}				
0x00000052	Write	3D-Gruppe FIFO-Gruppe	{ UINT32[m] }	1	[0, 1...255]	Achsbelegung der Gruppe: 1. Achs-ID, ... , m.-Achs-ID m: Dimension der 3D-Gruppe (X, Y, Z, Q1, Q2, Q3, Q4, Q5) bzw. FIFO-Gruppe	
0x00000053	Write	3D-Gruppe FIFO-Gruppe Kinematik-Gruppe	VOID			Auflösen der 3D-, FIFO- oder Kinematik-Achsbelegung und Rückführung der Achsen in ihre persönlichen PTP-Gruppen	

Index-Offset (Hex)	Zugriff	Gruppentyp	Daten- typ	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000054	Write	Kinematik-Gruppe	{			Achsbelegung der Kinematik-Gruppe:	
			UINT32[8]	1	[0, 1...255]	MCS-Achsen-IDs (Machine Coordinate System)	
			UINT32[8]	1	[0, 1...255]	ACS-Achsen-IDs (Axis Coordinate System)	
			UINT32	1	≥ 0	Reserviert	
			UINT32	1	≥ 0	Reserviert (NEU)	
			}				
0x00000060	ReadWrite	3D-Gruppe		1		Internes "Feed Group" Kommando ("Feeder")	Internes Kommando!
0x00000061	ReadWrite	3D-Gruppe		1		Internes "Feed Group" Kommando ("Feeder")	Internes Kommando!
0x00000110	Write	1D-Gruppe	VOID			Referenziere 1D-Gruppe ("Eichen")	
0x00000111	Write	1D-Gruppe	{			Neue Endposition 1D-Gruppe	
			UINT32	ENUM	s. Anhang	Endpositionstyp [►_161] (s. Anhang)	
			REAL64	z. B. mm	±∞	Neue Endposition (Zielposition)	
			}				
0x0000011A	Write	1D-Gruppe	{			Setze Istposition 1D-Gruppe	<b>Vorsicht bei Benutzung!</b>
			UINT32	ENUM	s. Anhang	Istpositionstyp [►_161] (s. Anhang)	Immer an SAF-Port 501!
			REAL64	z. B. mm	±∞	Istposition für Achse	
			}				
0x0000011B	Write	1D-Gruppe	UINT32	1	0/1	Setze Referenzierflag ("Eichflag")	<b>Vorsicht bei Benutzung!</b>
0x00000120	Write	1D-Gruppe	{			Start 1D-Gruppe (Standard Start):	
			UINT32	ENUM	s. Anhang	Starttyp [►_160] (s. Anhang)	
			REAL64	z. B. mm	±∞	Endposition (Zielposition)	
			REAL64	mm/s	≥ 0.0	Geforderte Geschwindigkeit	
			}				
0x00000121	Write	1D-Gruppe (SERVO)	{			Start 1D-Gruppe (Erweiterter Start):	
			UINT32	ENUM	s. Anhang	Starttyp [►_160] (s. Anhang)	
			REAL64	z. B. mm	±∞	Endposition (Zielposition)	
			REAL64	mm/s	≥ 0.0	Geforderte Geschwindigkeit	
			UINT32	1	0/1	Standardbeschleunigung?	
			REAL64	mm/s^2	≥ 0.0	Beschleunigung	
			UINT32	1	0/1	Standardverzögerung?	
			REAL64	mm/s^2	≥ 0.0	Verzögerung	
			UINT32	1	0/1	Standardruck?	
			REAL64	mm/s^3	≥ 0.0	Ruck	
			}				

Index-Offset (Hex)	Zugriff	Gruppentyp	Daten- typ	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000122	Write	1D-Gruppe(MW-SERVO)	{			Start 1D-Gruppe (Spezieller Start):	Reservierte Startfunktion, kein Standard!
			UINT32	ENUM	s. Anhang	Starttyp [► 160] (s. Anhang)	
			REAL64	z. B. mm	$\pm\infty$	Endposition (Zielposition)	
			REAL64	mm/s	$\geq 0.0$	Geforderte Anfangsgeschwindigkeit	
			REAL64	z. B. mm	$\pm\infty$	Position, für neues Geschwindigkeitsniveau	
			REAL64	mm/s	$\geq 0.0$	Neues Endgeschwindigkeitsniveau	
			UINT32	1	0/1	Standardbeschleunigung?	
			REAL64	mm/s <sup>2</sup>	$\geq 0.0$	Beschleunigung	
			UINT32	1	0/1	Standardverzögerung?	
			REAL64	mm/s <sup>2</sup>	$\geq 0.0$	Verzögerung	
			UINT32	1	0/1	Standardruck?	
			REAL64	mm/s <sup>3</sup>	$\geq 0.0$	Ruck	
			}				
0x00000126	Write	1D-Gruppe	{			Start Drive-Output:	
			UINT32	ENUM	s. Anhang	Ausgabetyyp [► 168] (s. Anhang)	
			REAL64	z. B. %	$\pm\infty$	Geforderter Ausgabewert (z. B. %)	
			}				
0x00000127	Write	1D-Gruppe	VOID			Stop Drive-Output	
0x00000128	Write	1D-Gruppe	{			Änderung/Wechsel des Drive-Outputs:	
			UINT32	ENUM	s. Anhang	Ausgabetyyp [► 168] (s. Anhang)	
			REAL64	z. B. %	$\pm\infty$	Geforderter Ausgabewert (z. B. %)	
			}				
0x00000130	Write	1D-Gruppe (SERVO)	{			1D-Streckenkompensation (SERVO):	
			UINT32	ENUM	s. Anhang	Kompensationstyp [► 162] (s. Anhang)	
			REAL64	mm/s/s	$\geq 0.0$	Max. Beschleunigungserhöhung	
			REAL64	mm/s/s	$\geq 0.0$	Max. Verzögerungserhöhung	
			REAL64	mm/s	$\geq 0.0$	Max. Erhöhungsgeschwindigkeit	
			REAL64	mm/s	$\geq 0.0$	Grundgeschwindigkeit des Prozesses	
			REAL64	z. B. mm	$\pm\infty$	Ausgleichende Wegdifferenz	
			REAL64	z. B. mm	$\geq 0.0$	Weglänge für Kompensation	
			}				
0x00000131	Write	1D-Gruppe SERVO	VOID			Stop Streckenkompensation (SERVO)	

Index-Offset (Hex)	Zugriff	Gruppentyp	Daten- typ	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000140 (0x00n00140)	Write	Master/Slave- Kopplung: 1D- Gruppe (SERVO)	{			Master/Slave Kopplung (SERVO):	Erweiterung für "Fliegende Säge"! Winkel >0.0 und <= 90.0 Grad (Parallelsäge: 90.0 Grad)
			UINT32	ENUM	s. Anhang	Slavetyp/ Kopplungstyp [▶ 162] (s. Anhang)	
			UINT32	1	[1...255]	Achs-ID der Masterachse/Gruppe	
			UINT32	1	[0...8]	Subindex n der Masterachse (Default- Wert: 0)	
			UINT32	1	[0...8]	Subindex n der Slaveachse (Default- Wert: 0)	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Getriebefaktor FlySawVelo: Reserve FlySaw: Abs. Synchronposition Master [mm]	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve FlySawVelo: Reserve FlySawPos: Abs. Synchronposition Slave [mm]	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve FlySawVelo: Neigungswinkel in [GRAD] FlySawPos: Neigungswinkel in [GRAD]	
REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve FlySawVelo: Getriebefaktor FlySawPos: Getriebefaktor				
			}				
0x00000141	Write	Master/Slave- Entkopplung: 1D- Gruppe(SERV O)	VOID			Master/Slave- Entkopplung (SERVO)	
0x00000142	Write	Master/Slave- Parameter 1D- Gruppe(SERV O)	{			Änderung der Kopplungsparameter (SERVO):	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Getriebefaktor	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve	
			}				
0x00000144	Write	Slave-Stop 1D- Gruppe(SERV O)	VOID			Stopp der "Fliegende Säge" (SERVO)	Nur für "Fliegende Säge"
0x00000149	Write	Slave-Tabellen 1D-Gruppe (SERVO)	REAL64	1	±∞	Setzen der Slave- Tabellenskalierung einer Solo- Tabellenkopplung (SERVO)	Nur für Solo- Tabellenslave

Index-Offset (Hex)	Zugriff	Gruppentyp	Daten- typ	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000150	Write	1D-Gruppe	VOID			Deaktiviere komplette 1D-Gruppe/Achse (Disable)	
0x00000151	Write	1D-Gruppe	VOID			Aktiviere komplette 1D-Gruppe / Achse (Enable)	
0x00000160	Write	1D-Gruppe	VOID			Deaktiviere Drive-Output der 1D-Gruppe (Disable)	
0x00000161	Write	1D-Gruppe	VOID			Aktiviere Drive-Output der 1D-Gruppe (Enable)	
0x00000362	Write	Eil/Schleich-Gruppe	UINT16	1	0/1	Feststellbremse lösen ? 0: automatische Ansteuerung (Default) 1: zwingend immer gelöst!	
0x00000701	Write	FIFO-Gruppe	VOID			Start FIFO-Gruppe (FIFO-Tabelle muss zuvor gefüllt worden sein)	(n*m)-FIFO
0x00000710	Write	FIFO-Gruppe	{ REAL64[x*m]}	z. B. mm	$\pm\infty$	Schreiben von x FIFO Einträgen (Zeilen): (x*m)-Werte (eine oder mehrere Zeilen) n: FIFO-Länge (Zeilenanzahl) m: FIFO-Dimension (Spaltenanzahl) Wertebereich x: [1 ... n]	Nur zeilenweise möglich! (ganzzahliges vielfaches)
0x00000711	Write	FIFO-Gruppe	{ REAL64[x*m]}	z. B. mm	$\pm\infty$	Überschreiben der letzten x FIFO Einträge (Zeilen): (x*m)-Werte (eine oder mehrere Zeilen) n: FIFO-Länge (Zeilenanzahl) m: FIFO-Dimension (Spaltenanzahl) Wertebereich x: [1 ... n]	Nur zeilenweise möglich! (ganzzahliges vielfaches)
0x00000801	Write	Kinematik-Gruppe	VOID			Start Kinematik-Gruppe	Reservierte Funktion, kein Standard!

**3.1.1.5.4.4**      **Spezifikation Achsen**

**3.1.1.5.4.4.1**      ***"Index-Offset" Spezifikation für Achsparameter (Index-Group  
0x4000 + ID)***



Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n00000	Read	every (Struktur für alle Achsparameter)	{			Allgemeine ACHS- PARAMETER- STRUKTUR (NC/ CNC), beinhaltet auch die Unterelemente wie Encoder, Regler und Drive (s. MC_ReadParameterS et in TcMc2.lib)  Anm.: Größe und Alignment geändert.	Geändert ab TC3
			UINT32	1		Achs-ID	
			STRING[30+1]	1		Achsname	
			UINT32	1	ENUM	Achstyp [ <a href="#">▶_160</a> ]	
			... }	... 	... 	... 1024 Byte (anstatt 512 Byte)	
0x00000001	Read	every	UINT32	1		Achs-ID	
0x00000002	Read	every	STRING[30+1]	1		Achsname	Ab TC3.1 Build 4022.32 oder 4024.6 beliebig viele Zeichen
			UINT8[. . .]				
0x00000003	Read	every	UINT32	1	ENUM	<a href="#">Achstyp [▶_160]</a>	
0x00000004	Read	every	UINT32	µs		Zykluszeit Achse (SAF)	
0x00000005	Read	every	STRING[10+1]	1		Physikalische Einheit	
0x00000006	Read/Write	every	REAL64	z. B. mm/s		Ref.-Geschw. in Nockenrichtung	
0x00000007	Read/Write	every	REAL64	z. B. mm/s		Ref.-Geschw. in Syncrichtung	
0x00000008	Read/Write	every	REAL64	z. B. mm/s		Geschwindigkeit Hand Slow	
0x00000009	Read/Write	every	REAL64	z. B. mm/s		Geschwindigkeit Hand Fast	
0x0000000A	Read/Write	every	REAL64	z. B. mm/s	[0.0...1.0E20]	Geschwindigkeit Eilgang	
0x0000000F	Read/Write	every	UINT16	1	0/1	Positionsbereichsüber- wachung?	
0x00000010	Read/Write	every	REAL64	z. B. mm	[0.0...1.0E6]	Positionsbereichsfenst- er	
0x00000011	Read/Write	every	UINT16	1	0/1	Bewegungsüberwachu- ng?	
0x00000012	Read/Write	every	REAL64	s	[0.0...600]	Bewegungsüberwachu- ngszeit	
0x00000013	Read/Write	every	UINT16	1	0/1	Schleife?	
0x00000014	Read/Write	every	REAL64	z. B. mm		Schleifenweg (±)	
0x00000015	Read/Write	every	UINT16	1	0/1	Zielpositionsüberwach- ung?	
0x00000016	Read/Write	every	REAL64	z. B. mm	[0.0...1.0E6]	Zielpostionsfenster	
0x00000017	Read/Write	every	REAL64	s	[0.0...600]	Zielpositionsüberwach- ungszeit	
0x00000018	Read/Write	every	REAL64	z. B. mm		Pulsweg in pos. Richtung	
0x00000019	Read/Write	every	REAL64	z. B. mm		Pulsweg in neg. Richtung	
0x0000001A	Read/Write	every	UINT32	1	ENUM (≥0)	Fehlersignalisierung/ Fehlerreaktion: 0: sofort (Default) 1: verzögert (z. B. für Master/ Slave-Kopplung)	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000001B	Read/Write	every	REAL64	s	[0...1000]	Fehlerverzögerungszeit (wenn verzögerte Fehlerreaktion angewählt ist)	
0x0000001C	Read/Write	every	UINT16	1	0/1	Slaves über Ist-Werte koppeln wenn nicht betriebsbereit?	
0x0000001D	Read/Write	every	REAL64	z. B. mm/s <sup>2</sup>	[0, 0.01...1.0E10]	Beschleunigung für Übergangprofil für die Umschaltung von Soll- auf Istwerte (Fading der Position):  Default: 0 (hier wird das Minimum der Achsbeschleunigungen verwendet, also MIN(Acc, Dec) )	
0x0000001E	Read/Write	every	UINT32	1	ENUM (≥0)	Fast Axis Stop Signal Type:  Auswahl des Signaltyps durch den ein Fast-Axis-Stopp ausgelöst wird (s. Bit 7 im Drive->nStatus4)  "0 (SignalType_OFF)", "1 (SignalType_RisingEdge)", "2 (SignalType_FallingEdge)", "3 (SignalType_BothEdges)", "4 (SignalType_HighActive)", "5 (SignalType_LowActive)"	
0x00000020	Read/Write	every	UINT16	1	0/1	Bewegungskommandos für Slaveachsen erlauben?  Default: FALSE	
0x00000021	Read/Write	every	UINT16	1	0/1	Bewegungskommandos für Achsen mit aktiver externer Sollwertgenerierung erlauben?  Default: FALSE	
0x00000026	Read/Write	every	UINT32	1		Interpretation der Einheiten (Position, Geschwindigkeit, Zeit)  Bit 0: Geschwindigkeit in x/min statt x/s  Bit 1: Position in tausendstel der Basiseinheit  Bit 2: Modulopositionsanzeige	Siehe Encoder! Bitarray
0x00000027	Read/Write	every	REAL64	z. B. mm/s	[>0...1.0E20]	Maximal erlaubte Fahrgeschwindigkeit	
0x00000028	Read/Write	every	REAL64	z. B. mm	[0.0...1.0E6]	Bewegungsüberwachungsfenster	
0x00000029	Read/Write	every	UINT16	1	0/1	PEH-Zeitüberwachung?	Posi.Ende und Genauhalt
0x0000002A	Read/Write	every	REAL64	s	[0.0...600]	PEH Überwachungszeit	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000002C	Read/Write	every	REAL64	z. B. mm	[-1000.0 ...1000.0]	Lose	
0x00000030	Read	every	UINT16	1	[0,1]	Persistente (dauernd anhaltende) Daten für z. B. Istposition und Referenzierstatus des Encoders?	Boot-Parameter, nicht online änderbar.
0x00000031	Read	every	{ UINT8[6] UINT16 UINT16 } 10 Byte	AmsAdd r: AmsNetI d, AmsPort No. Channel No	1	Lesen der Hardware AMS-Adresse (AMS Net ID und AMS Port No) und der EtherCAT-Kanalnummer (Kommunikationskanal 0,1,2,3...)	
0x00000031	Read	every	{ UINT8[6] UINT16 UINT16 // UINT16 UINT32 UINT32 UINT32 UINT32 UINT32 UINT32 UINT32 UINT32 UINT32 UINT32 UINT32 UINT32[3] } 64 Byte	AmsAdd r: AmsNetI d, AmsPort No. Channel No Reservie rt NcDriveI D NcDriveI ndex NcDrive Type NcEnclD NcEnclI dex NcEncT ype NcAxisI D NcAxisT ype TcDrive ObjectI d TcEncO bjectI d Reservie rt	1	Lesen der Hardware AMS-Adresse (AMS Net ID und Geräte AMS Port No) und der EtherCAT Kanalnummer (Kommunikationskanal 0,1,2,3...)  Ergänzt um Zusatzinformationen der NC wie NcDriveID, NcDriveType (s. Anhang), etc.	NEU ab TC3 DriveObjectI d und EncObjectI d ab NC Build 4437
0x00000033	Read	every	{ UINT16 ApplRequestBit UINT16 ApplRequestTy pe UINT32 ApplCmdNo UINT32 ApplCmdVersio n ... } 1024 Byte	1  1  Nicht impleme ntiert  1	0/1 ≥0 >0 ≥0	Allgemeine APPLICATION REQUEST-STRUKTUR (NC/NCI), z. B. für ApplicationHoming-Request (s. <i>MC_ReadApplicationRequest</i> in <i>TcMc2.lib</i> )  Application Request Types: 0: NONE (IDLE) 1: HOMING	Geändert in TC3
0x00000051	Read	Kanal: every	UINT32			Kanal-ID	
0x00000052	Read	Kanal: every	STRING[30+1]			Kanalname	
0x00000053	Read	Kanal: every	UINT32	1	ENUM	<a href="#">Kanaltyp ▶ 158</a>	
0x00000054	Read	Gruppe: every	UINT32			Gruppen-ID	
0x00000055	Read	Gruppe: every	STRING[30+1]			Gruppenname	
0x00000056	Read	Gruppe: every	UINT32	1	ENUM	<a href="#">Gruppentyp ▶ 159</a>	
0x00000057	Read	every	UINT32			Anzahl der Encoder	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000058	Read	every	UINT32			Anzahl der Regler	
0x00000059	Read	every	UINT32			Anzahl der Drives	
0x0000005A	Read	every	{			Lesen der sämtlicher Unterelemente einer Achse:	
			UINT32[ 9 ]	1	[0, 1...255]	Encoder-IDs der Achse	
			UINT32[ 9 ]	1	[0, 1...255]	Regler-IDs der Achse	
			UINT32[ 9 ]	1	[0, 1...255]	Drive-IDs der Achse	
			} 108 bytes				
0x000000F1	Read/Write	every	REAL64	z. B. mm/s <sup>2</sup>	Default: 1.0E5	Maximal erlaubte Beschleunigung	NEU ab TC 3.2
0x000000F2	Read/Write	every	REAL64	z. B. mm/s <sup>2</sup>	Default: 1.0E6	Maximal erlaubte Verzögerung	NEU ab TC 3.2
0x00000101	Read/Write	Servo	REAL64	z. B. mm/s <sup>2</sup>	[0.01...1.0E20]	Beschleunigung (Default-Datensatz)	
0x00000102	Read/Write	Servo	REAL64	z. B. mm/s <sup>2</sup>	[0.01...1.0E20]	Verzögerung (Default-Datensatz)	
0x00000103	Read/Write	Servo	REAL64	z. B. mm/s <sup>3</sup>	[0.1...1.0E30]	Ruck (Default-Datensatz)	
0x00000104	Read/Write	Servo	REAL64	s	[0.0 ... 1.0] Default: 0.0 s	Verzögerungszeit zwischen Geschwindigkeits- und Positionswerten des Sollwertgenerators in Sekunden	
0x00000105	Read/Write	Servo	UINT32	1	ENUM Default: Typ 1	Override-Typ [► 160] für Geschwindigkeit: 1: Bezogen auf interne reduzierte Geschwindigkeit (ohne Iteration) 2: Bezogen auf originale externe Startgeschwindigkeit (ohne Iteration) 3: Bezogen auf interne reduzierte Geschwindigkeit (Optimierung mittels Iteration) 4: Bezogen auf originale externe Startgeschwindigkeit (Optimierung mittels Iteration)	
0x00000106	Read/Write	Servo	REAL64	1	[0.0 ... 1.0E6] Default: 0.0	Maximal erlaubter Geschwindigkeitssprung für Dynamikreduktion $DV = Faktor * \min(A+, A-) * DT$	
0x00000107	Read/Write	Servo	UINT16	1	[0,1] Default: 1	Aktiviert Beschleunigungs- und Ruckbegrenzung für die Hilfsachse (Q1 bis Q5)	
	Read/Write	Servo	REAL64	z. B. mm	[0.0..1000.0]	Größe des Toleranzballs für die Hilfsachsen	
	Read/Write	Servo	REAL64	z. B. mm	[0.0..10000.0]	Maximal erlaubte Positionsabweichung bei verkleinertem Toleranzball Nur für Hilfsachsen	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000010A	Read/Write	Servo	REAL64	z. B. mm/s^2	[0.01 ... 1.0E20]	Fast Axis Stop: Beschleunigung (s.a. Fast Axis Stop Signal Type)	
0x0000010B	Read/Write	Servo	REAL64	z. B. mm/s^2	[0.01 ... 1.0E20]	Fast Axis Stop: Verzögerung (s.a. Fast Axis Stop Signal Type)	
0x0000010C	Read/Write	Servo	REAL64	z. B. mm/s^3	[0.1 ... 1.0E30]	Fast Axis Stop: Ruck (s.a. Fast Axis Stop Signal Type)	
0x0000010D	Read/Write	Servo	UINT32	1		Index-Offset des Achszustandes, der im zyklischen Interface als „UserData“ übergeben wird.  0x00000000: deaktiviert 0x00010012: Geberposition mit Positionsvorspannung (ohne Positionskorrektur und ohne Totzeitkompensation) 0x00010014: DriveActVelo 0x00010017: MC_SetPosition-Offsets	
0x00000201	Read/Write	Schrittmotor	UINT32	1	ENUM	Betriebsmodus Schrittmotor	
0x00000202	Read/Write	Schrittmotor	REAL64	z. B. mm/STEP	[1.0E-6 ... 1000.0]	Wegskalierung eines Motorschrittes	
0x00000203	Read/Write	Schrittmotor	REAL64	z. B. mm/s	[0.0 ... 1000.0]	Mindestgeschwindigkeit für Geschwindigkeitsprofil	
0x00000204	Read/Write	Schrittmotor	UINT32	1	[0 ... 100]	Anzahl der Schritte pro Frequenz-/ Geschwindigkeitsstufe	
0x00000205	Read/Write	Schrittmotor	UINT32	1		Motormaske als Syncimpuls	Nicht implementiert!
0x00000301	Read/Write	Eil/Schleich	REAL64	z. B. mm	[0.0 ... 100000.0]	Schleichweg in pos.Richtung	
0x00000302	Read/Write	Eil/Schleich	REAL64	z. B. mm	[0.0 ... 100000.0]	Schleichweg in neg. Richtung	
0x00000303	Read/Write	Eil/Schleich	REAL64	z. B. mm	[0.0 ... 100000.0]	Bremsweg in pos. Richtung	
0x00000304	Read/Write	Eil/Schleich	REAL64	z. B. mm	[0.0 ... 100000.0]	Bremsweg in neg. Richtung	
0x00000305	Read/Write	Eil/Schleich	REAL64	s	[0.0 ... 60.0]	Bremsverzög. in pos. Richtung	
0x00000306	Read/Write	Eil/Schleich	REAL64	s	[0.0 ... 60.0]	Bremsverzög. in neg. Richtung	
0x00000307	Read/Write	Eil/Schleich	REAL64	s	[0.0 ... 60.0]	Umschaltzeit Eil auf Schleich	
0x00000308	Read/Write	Eil/Schleich	REAL64	z. B. mm	[0.0 ... 100000.0]	Schleichweg Stop	
0x00000309	Read/Write	Eil/Schleich	REAL64	s	[0.0 ... 60.0]	Verzögerungszeit um Bremse zu lösen	
0x0000030A	Read/Write	Eil/Schleich	REAL64	s	[0.0 ... 60.0]	Pulszeit in pos. Richtung	
0x0000030B	Read/Write	Eil/Schleich	REAL64	s	[0.0 ... 60.0]	Pulszeit in neg. Richtung	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
<b>ENCODER</b>							
0x00n10001	Read	Encoder: every	UINT32	1	[1 ... 255]	Encoder-ID n = 0: Standardencoder der Achsen > 0: n-ter Encoder der Achse (optional)	
0x00n10002	Read	Encoder: every	STRING[30+1]	1	30 Zeichen	Encodername	
0x00n10003	Read	Encoder: every	UINT32	1	ENUM (>0)	Encodertyp [ <a href="#">▶ 164</a> ]	
0x00n10004	Read/Write	Encoder: every	UINT32	1	Byteoffset	Input-Adress-Offset (IO-Input-Image)	Änderung der IO-Adresse
0x00n10005	Read/Write	Encoder: every	UINT32	1	Byteoffset	Output-Adress-Offset (IO-Output-Image)	Änderung der IO-Adresse
0x00n10006	Read/Write	Encoder: every	REAL64	z. B. mm/INC	[1.0E-12 ... 1.0E+30]	Resultierender Skalierungsfaktor (Zähler / Nenner)  Anm.: ab TC3 besteht der Skalierungsfaktor aus zwei Komponenten, Zähler und Nenner (Default: 1.0).	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00n10007	Read/Write	Encoder: every	REAL64	z. B. mm	[±1.0E+9]	Positionsoffset	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00n10008	Read/Write	Encoder: every	UINT16	1	[0,1]	Geberzählrichtung	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00n10009	Read/Write	Encoder: every	REAL64	z. B. mm	[0.001 ... 1.0E+9]	Modulo-Faktor	
0x00n1000A	Read/Write	Encoder: every	UINT32	1	s. ENUM (>0)	Encodermodus [ <a href="#">▶ 165</a> ]	
0x00n1000B	Read/Write	Encoder: every	UINT16	1	0/1	Softend-Min- Überwachung ?	
0x00n1000C	Read/Write	Encoder: every	UINT16	1	0/1	Softend-Max- Überwachung ?	
0x00n1000D	Read/Write	Encoder: every	REAL64	mm		Softendlage Min	
0x00n1000E	Read/Write	Encoder: every	REAL64	mm		Softendlage Max	
0x00n1000F	Read/Write	Encoder: every	UINT32	1	s. ENUM (≥0) im Anhang	Encoder- Auswerterichtung [ <a href="#">▶ 165</a> ] (Freigabe log. Zählrichtung)	
0x00n10010	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filterzeit für Positionswert in Sekunden(P-T1)	
0x00n10011	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filterzeit für Geschwindigkeitswert in Sekunden (P-T1)	
0x00n10012	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filterzeit für Beschleunigungswert in Sekunden (P-T1)	
0x00n10013	Read/Write	Encoder: every	STRING[10+1]	1		Physikalische Einheit	Nicht implementiert!
0x00n10014	Read/Write	Encoder: every	UINT32	1		Interpretation der Einheiten (Position, Geschwindigkeit, Zeit)  Bit 0: Geschwindigkeit in x/min statt x/s  Bit 1: Position in tausendstel der Basiseinheit	Nicht implementiert! Bitarray

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n10015	Read	Encoder: every	UINT32	INC	[0x0... 0xFFFFFFFF]	Geber-Maske (Maximalwert des Geber-Istwertes in Inkrementen  Anm.: Die Geber-Maske darf ein beliebiger Zahlenwert sein (z. B. 3600000) und muss nicht mehr wie in der Vergangenheit einer durchgehende Folge von binären Einsen entsprechen (2 <sup>n</sup> -1).	ReadOnly-Parameter s.a. Param. "Geber-Sub-Maske"
0x00n10016	Read/Write	Encoder: every	UINT16	1	0/1	Istpositionskorrektur (Meßsystemfehlerkorrektur)?	
0x00n10017	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filterzeit für Istpositionskorrektur in Sekunden (P-T1)	
0x00n10019	Read/Write	Encoder: every	UINT32	1	ENUM (>0)	<u>Encoder- Bezugsmaßsystem</u> [► 165]	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00n1001A	Read	Encoder: every	UINT32	1	ENUM (>0)	Encoder- Positionsinitialisierung	Nicht implementiert!
0x00n1001B	Read/Write	Encoder: every	REAL64	z. B. mm	[≥0, Modulo- Faktor/2]	Toleranzfenster für Modulo-Start	
0x00n1001C	Read	Encoder: every	UINT32	1	ENUM (>0)	<u>Encoder-Vorzeichen- Interpretation</u> [► 165] (Datentyp)	
0x00n1001D	Read	Encoder: every	UINT16	1	0/1	Inkremental- oder Absolutencoder ?  0: Inkrementaler Encodertyp 1: Absoluter Encodertyp	
0x00n10023	Read/Write	Encoder: every	REAL64	z. B. mm/INC	[1.0E-12 ... 1.0E+30]	Komponente des Skalierungsfaktors: Zähler  (=> Skalierungsfaktor Zähler / Skalierungsfaktor Nenner)	NEU ab TC3 Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00n10024	Read/Write	Encoder: every	REAL64	1	[1.0E-12 ... 1.0E+30]	Komponente des Skalierungsfaktors: Nenner  (=> Skalierungsfaktor Zähler / Skalierungsfaktor Nenner) Default: 1.0	NEU ab TC3 Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00n10025	Read/Write	Encoder: every	{ REAL64 REAL64 }	z. B. mm/INC 1	[1.0E-12 ... 1.0E+30] [1.0E-12 ... 1.0E+30]	Komponente des Skalierungsfaktors: Zähler  Komponente des Skalierungsfaktors: Nenner  (=> Skalierungsfaktor Zähler / Skalierungsfaktor Nenner)	NEU ab TC3
0x00n10030	Read/Write	Encoder: every	UINT32	1		Internes Encoder-Control-Doppelwort zur Festlegung der Betriebsarten und Eigenschaften	NEU ab TC3

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n10101	Read/Write	E: INC	UINT16	1	[0,1]	Suchrichtung für Ref.nocken invers?	
0x00n10102	Read/Write	E: INC	UINT16	1	[0,1]	Suchrichtung für Syncimpuls invers?	
0x00n10103	Read/Write	E: INC	REAL64	z. B. mm	[±1000000.0]	Referenzposition	
0x00n10104	Read/Write	E: INC	UINT16	1	[0,1]	Abstandsüberwachung zwischen Ref.nocken und Syncimpuls aktiv?	Nicht implementiert!
0x00n10105	Read/Write	E: INC	UINT32	INC	[0 ... 65536]	Mindestabstand Ref.nocken zum Syncimpuls in Inkrementen	Nicht implementiert!
0x00n10106	Read/Write	E: INC	UINT16	1	[0,1]	Externer Syncimpuls?	
0x00n10107	Read/Write	E: INC	UINT32	1	s. ENUM (>0)	<u>Referenziermodus</u> [►_166]	
0x00n10108	Read/Write	E: INC	UINT32	1	[0x0000000F...0xFFFFFFFF]Binärmaske: (2 <sup>n</sup> - 1)	Geber-Sub-Maske (Maximalwert des Absolutbereichs des Geber-Istwertes in Inkrementen)  Wird z. B. verwendet als Referenzmarke für den Referenzier Mode "Software Sync" und für die NC-Retain-Daten ("ABSOLUTE (MODULO)", "INCREMENTAL (SINGLETURN ABSOLUTE)").  Anm.1: Die Geber-Sub-Maske muss kleiner gleich der Geber-Maske sein.  Anm.2: Die Geber-Maske muss ein ganzzahliges Vielfaches der Geber-Sub-Maske sein.  Anm.3: Die Geber-Sub-Maske muss einer durchgehenden Folge von binären Einsen entsprechen (2 <sup>n</sup> -1), z. B. 0x000FFFFFFF.	s.a. Param. "Geber-Maske"
0x00n10110	Read/Write	E: INC (Encodersimulation)	REAL64	1	[0.0 ... 1000000.0]	Skalierung/ Gewichtung des Rauschanteils für Simulationsencoder	
<b>CONTROLLER</b>							
0x00n20001	Read	Regler: every	UINT32	1	[1 ... 255]	Regler ID n = 0: Standardregler der Achsen > 0: n-ter Regler der Achse (optional)	
0x00n20002	Read	Regler: every	STRING[30+1]	1	30 Zeichen	Reglername	
0x00n20003	Read	Regler: every	UINT32	1	ENUM (>0)	<u>Regler-Typ</u> [►_163]	
0x00n2000A	Read/Write	Regler: every		1	ENUM (>0)	Reglermodus	
0x00n2000B	Read/Write	Regler: every	REAL64	%	[0.0 ... 1.0]	Gewichtung der Geschwindigkeitsvorsteuerung (Standardwert: 1.0 = 100 %)	



Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n20010	Read/Write	Regler: every	UINT16	1	0/1	Schleppabstandsüber w. Pos.?	
0x00n20011	Read/Write	Regler: every	UINT16	1	0/1	Schleppabstandsüber w. Geschw.?	
0x00n20012	Read/Write	Regler: every	REAL64	z. B. mm		Max. Schleppabstand Position	
0x00n20013	Read/Write	Regler: every	REAL64	s		Max. Schleppfilterzeit Position	
0x00n20014	Read/Write	Regler: every	REAL64	z. B. mm/s		Max. Schleppabstand Geschw.	
0x00n20015	Read/Write	Regler: every	REAL64	s		Max. Schleppfilterzeit Geschw.	
0x00n20100	Read/Write	P/PID (Pos., (Geschw.))	REAL64	1	[0.0...1.0]	Maximale Ausgabebegrenzung (±) für Regler- Gesamtausgabe	(Standardwert: 0.5 == 50%)
0x00n20102	Read/Write	P/PID (Pos.)	REAL64	z. B. mm/s/ mm	[0.0...1000.0]	Proportionalverstärkun g $k_p$ bzw. $k_v$ Einheit: Base Unit / s / Base Unit	Positionsregelu ng
0x00n20103	Read/Write	PID (Pos.)	REAL64	s	[0.0 ... 60.0]	Nachstellzeit $T_n$	Positionsregelu ng
0x00n20104	Read/Write	PID (Pos.)	REAL64	s	[0.0 ... 60.0]	Vorhaltzeit $T_v$	Positionsregelu ng
0x00n20105	Read/Write	PID (Pos.)	REAL64	s	[0.0 ... 60.0]	Verzögerungszeit $T_d$	Positionsregelu ng
0x00n20106	Read/Write	PP (Pos.)	REAL64	z. B. mm/s/ mm	[0.0...1000.0]	Zusätzliche Proportionalverstärkun g $k_p$ bzw. $k_v$ , die oberhalb einer Grenzgeschwindigkeit in Prozent gilt. Einheit: Base Unit / s / Base Unit	Positionsregelu ng
0x00n20107	Read/Write	PP (Pos.)	REAL64	%	[0.0...1.0]	Schwellgeschwindigke it in Prozent, oberhalb derer die zusätzliche Proportionalverstärkun g $k_p$ bzw. $k_v$ gilt	
0x00n20108	Read/Write	P/PID (Acc.)	REAL64	s	[0.0 ... 100.0]	Proportionalverstärkun g $k_a$	Beschleunigung s- vorsteuerung
0x00n2010D	Read/Write	P/PID	REAL64	mm	[0.0 ... 10000.0]	Totzone ("dead band") für Positionsfehler (Regelabweichung) (für P/PID-Regler mit Geschwindigkeits- oder Momenteninterface)	Reservierte Funktion
0x00n2010F	Read/Write	P/PP/PID (Pos.) Slave-Regelung	REAL64	(mm/s) / mm	[0.0...1000.0]	Slave- Koppeldifferenzregelu ng: Proportionalverstärkun g $k_{cp}$	Slave- Koppeldifferenz regelung
0x00n20110	Read/Write	P (Pos.)	UINT16	1	0/1	Automatischer Offsetabgleich: aktiv/ passiv	
0x00n20111	Read/Write	P (Pos.)	UINT16	1	0/1	Automatischer Offsetabgleich: Halte- Modus	
0x00n20112	Read/Write	P (Pos.)	UINT16	1	0/1	Automatischer Offsetabgleich: Fading-Modus	
0x00n20114	Read/Write	P (Pos.)	REAL64	%	[0.0 ... 1.0]	Automatischer Offsetabgleich: Vorsteuer-Grenze	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n20115	Read/Write	P (Pos.)	REAL64	s	[0.1 ... 60.0]	Automatischer Offsetabgleich: Zeitkonstante	
0x00n20116	Read/Write	PID (Pos.)	REAL64	%	[0.0...1.0]	Maximale Ausgabebeschränkung ( $\pm$ ) für I-Anteil in Prozent (Default-Einstellung: 0.1 = 10 %)	
0x00n20117	Read/Write	PID (Pos.)	REAL64	%	[0.0...1.0]	Maximale Ausgabebeschränkung ( $\pm$ ) für D-Anteil in Prozent (Default-Einstellung: 0.1 = 10 %)	
0x00n20118	Read/Write	PID (Pos.)	UINT16	1	0/1	Abschalten des I-Anteils während eines aktiven Positioniervorganges (sofern I-Anteil aktiv)? (Defaulteinstellung: 0 = FALSE)	
0x00n20120	Read/Write	P/PID (Pos.)	REAL64	s	$\geq 0$	PT-1 Filterwert für Positionsfehler (Pos.-Regeldifferenz)	Reservierte Funktion, kein Standard!
0x00n20202	Read/Write	P/PID (Geschw.)	REAL64	1	[0.0...1000.0]	Proportionalverstärkung $k_p$ bzw. $k_v$	Geschwindigkeitsregelung
0x00n20203	Read/Write	PID (Geschw.)	REAL64	s	[0.0 ... 60.0]	Nachstellzeit $T_n$	Geschwindigkeitsregelung
0x00n20204	Read/Write	PID (Geschw.)	REAL64	s	[0.0 ... 60.0]	Vorhaltzeit $T_v$	Geschwindigkeitsregelung
0x00n20205	Read/Write	PID (Geschw.)	REAL64	s	[0.0 ... 60.0]	Verzögerungszeit $T_d$	Geschwindigkeitsregelung
0x00n20206	Read/Write	PID (Geschw.)	REAL64	%	[0.0...1.0]	Maximale Ausgabebeschränkung ( $\pm$ ) für I-Anteil in Prozent (Defaulteinstellung: 0.1 = 10 %)	Geschwindigkeitsregelung
0x00n20207	Read/Write	PID (Geschw.)	REAL64	%	[0.0...1.0]	Maximale Ausgabebeschränkung ( $\pm$ ) für D-Anteil in Prozent (Defaulteinstellung: 0.1 = 10 %)	Geschwindigkeitsregelung
0x00n2020D	Read/Write	P/PID (Geschw.)	REAL64	mm/s	[0.0 ... 10000.0]	Totzone ("dead band") für Geschwindigkeitsfehler (Regelabweichung) (für P/PID-Regler mit Geschwindigkeits- oder Momenten-Interface)	Reservierte Funktion
0x00n20220	Read/Write	P/PID (Geschw.)	REAL64	s	$\geq 0$	PT-2-Filterwert für Geschwindigkeitsfehler (Geschw.-Regeldifferenz)	Geschwindigkeitsregelung, kein Standard!
0x00n20221	Read/Write	P/PID (Geschw.)	REAL64	s	$\geq 0$	PT-1-Filterwert für Geschwindigkeitsfehler (Geschw.-Regeldifferenz)	Reservierte Funktion, kein Standard!

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n20250	Read/Write	P/PI (Beobachter)	UINT32	1	ENUM (>0)	Beobachtermodus [►_163] für Regelung im Momenten- Interface 0: OFF (default) 1: LUENBERGER	
0x00n20251	Read/Write	P/PI (Beobachter)	REAL64	Nm / A	>0.0	Motor: Drehmomentkonstante $K_T$	
0x00n20252	Read/Write	P/PI (Beobachter)	REAL64	kg m <sup>2</sup>	>0.0	Motor: Trägheitsmoment $J_M$	
0x00n20253	Read/Write	P/PI (Beobachter)	REAL64	Hz	[100.0 ... 2000.0] Default: 500	Bandbreite $f_0$	
0x00n20254	Read/Write	P/PI (Beobachter)	REAL64	1	[0.0 ... 2.0] Default: 1.0	Korrekturfaktor $k_c$	
0x00n20255	Read/Write	P/PI (Beobachter)	REAL64	s	[0.0 ... 0.01] Default: 0.001	Geschwindigkeitsfilter (1. Ordnung): Zeitkonstante T	
0x00n20A03	Read/Write	P/PID (MW)	REAL64	cm <sup>2</sup>	[0.0 ... 1000000]	Zylinderfläche $A_A$ der A-Seite in cm <sup>2</sup>	Reservierte Parameter!
0x00n20A04	Read/Write	P/PID (MW)	REAL64	cm <sup>2</sup>	[0.0 ... 1000000]	Zylinderfläche $A_B$ der B-Seite in cm <sup>2</sup>	Reservierte Parameter!
0x00n20A05	Read/Write	P/PID (MW)	REAL64	cm <sup>3</sup> /s	[0.0 ... 1000000]	Nennvolumenstrom $Q_{nenn}$ in cm <sup>3</sup> /s	Reservierte Parameter!
0x00n20A06	Read/Write	P/PID (MW)	REAL64	bar	[0.0 ... 1000000]	Nenndruck bzw. Ventildruckabfall $P_{nenn}$ in bar	Reservierte Parameter!
0x00n20A07	Read/Write	P/PID (MW)	UINT32	1	[1 ... 255]	Achs-ID für den Systemdruck $P_0$	Reservierte Parameter!
<b>DRIVE:</b>							
0x00n30001	Read	Drive: every	UINT32	1	[1 ... 255]	Drive ID	
0x00n30002	Read	Drive: every	STRING[30+1]	1	30 Zeichen	Drive-Name	
0x00n30003	Read	Drive: every	UINT32	1	ENUM (>0)	Drive-Typ [►_168]	
0x00n30004	Read/Write	Drive: every	UINT32	1	Byteoffset	Input-Adress-Offset (IO-Input-Image)	Änderung der IO-Adresse
0x00n30005	Read/Write	Drive: every	UINT32	1	Byteoffset	Output-Adress-Offset (IO-Output-Image)	Änderung der IO-Adresse
0x00n30006	Read/Write	Drive: every	UINT16	1	[0,1]	Motorpolarität	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00n3000A	Read/Write	Drive: every	UINT32	1	ENUM (≥0)	Drive-Modus	
0x00n3000B	Read/Write	Drive: every	REAL64	%	[-1.0 ... 1.0]	Minimale Ausgabeschränke (Ausgabelimitierung) (Default-Einstellung: -1.0 = -100%)	
0x00n3000C	Read/Write	Drive: every	REAL64	%	[-1.0 ... 1.0]	Maximale Ausgabeschränke (Ausgabelimitierung) (Default-Einstellung: 1.0 = 100%)	
0x00n3000D	Read	Drive: every	UINT32	INC		Maximale Anzahl von Ausgabeinkrementen (Ausgabemaske)	
0x00n30010	Read/Write	Drive: every	UINT32	1		Internes Drive Control Doppelwort zur Festlegung der Antriebs-Betriebsarten	Reserviert!

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n30011	Read/Write	every	UINT32	1	$\geq 5$	Interner Drive-Reset-Zähler (Zeit in NC-Zyklen für Enable und Reset)	Reserviert!
0x00n30101	Read/Write	D: Servo	REAL64	z. B. mm/s	$>0.0$	Bezugsgeschwindigkeit bei Bezugs- bzw. Referenzoutput (Geschwindigkeitsvorteuerung)	
0x00n30102	Read/Write	D: Servo	REAL64	%	[0.0 ... 5.0]	Bezugs- bzw. Referenzoutput in Prozent (Default-Einstellung: 1.0 = 100%)	
0x00n30103	Read	D: Servo	REAL64	z. B. mm/s	$>0.0$	Resultierende Geschwindigkeit bei 100% Output	
0x00n30104	Read/Write	D: Servo	REAL64	z. B. mm/s	$\pm\infty$	Geschwindigkeitsoffset (DAC-Offset) für Driftabgleich (Offsetabgleich) der Achse	
0x00n30105	Read/Write	D: Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	Geschwindigkeitsskalierung (Skalierungsfaktor um auf Wichtung im Antrieb zu reagieren)	Für Sercos, Profi Drive, AX200x, CANopen
0x00n30106	Read/Write	D: Profi Drive DSC	UINT32	0.001 * 1/s	$\geq 0$	Profibus/Profi Drive DSC: Lageregelverstärkung Kpc	Nur für Profi Drive DSC
0x00n30107	Read/Write	D: Profi Drive DSC	REAL64	1	$\geq 0.0$	Profibus/Profi Drive DSC: Skalierung für Berechnung von 'XERR' (Default: 1.0)	Nur für Profi Drive DSC
0x00n30109	Read/Write	D: Servo (Sercos, CANopen)	REAL64	1	[0.0 ... 100000000.0]	Positionsskalierung (Skalierungsfaktor um auf Wichtung im Antrieb zu reagieren)	Für Sercos, CANopen
0x00n3010A	Read/Write	D: Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	Beschleunigungsskalierung (Skalierungsfaktor um auf Wichtung im Antrieb zu reagieren)	Für Sercos, Profi Drive, AX200x, CANopen
0x00n3010B	Read/Write	D: Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	Drehmomentskalierung (rot. Motor) bzw. Kraftskalierung (Linearmotor) (Skalierungsfaktor, um auf Wichtung im Antrieb zu reagieren) für „TorqueOffset“ (additives Moment als Vorsteuerung)	Für Sercos, Profi Drive, AX200x, CANopen
0x00n3010C	Read/Write	D: Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	Drehmomentskalierung (rot. Motor) bzw. Kraftskalierung (Linearmotor) (Skalierungsfaktor, um auf Wichtung im Antrieb zu reagieren) für „SetTorque“ (z.B. MC_TorqueControl) mit Drive OpMode CST)	Für Sercos, Profi Drive, AX200x, CANopen Ab TC3.1 B4024.2
0x00n30120	Read/Write	D: Servo/ Hydraulik/	UINT32	1	$\geq 0$	Tabellen-ID (0: keine Tabelle)	Nur für KL4xxx, M2400, Universal

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n30121	Read/Write	D: Servo/ Hydraulik	UINT32	1	≥ 0	Interpolation-Type 0: Linear 2: Spline	Nur für KL4xxx, M2400, Universal
0x00n30122	Read/Write	Servo/ Hydraulik	REAL64	%	[-1.0 ... 1.0]	Ausgabeoffset in Prozent Anmerkung: Wirkt nach der Kennlinienauswertung!	Nur für KL4xxx, M2400, Universal
0x00n30151	Read/Write	D: Servo / Nichtlinear	REAL64	1	[0.0 ... 100.0]	Quadrantenausgleichs faktor (Verhältnis zwischen I und III Quadr.)	
0x00n30152	Read/Write	D: Servo / Nichtlinear	REAL64	1	[0.01 ... 1.0]	Geschwindigkeitsstütz stelle in Prozent (1.0 = 100 %)	
0x00n30153	Read/Write	D: Servo / Nichtlinear	REAL64	1	[0.01 ... 1.0]	Ausgabe-Stützstelle in Prozent (1.0 = 100 %)	
0x00030301	Read/Write	D: Schrittmotor	UINT8	1		Bit-Maske: Zyklus 1	
0x00030302	Read/Write	D: Schrittmotor	UINT8	1		Bit-Maske: Zyklus 2	
0x00030303	Read/Write	D: Schrittmotor	UINT8	1		Bit-Maske: Zyklus 3	
0x00030304	Read/Write	D: Schrittmotor	UINT8	1		Bit-Maske: Zyklus 4	
0x00030305	Read/Write	D: Schrittmotor	UINT8	1		Bit-Maske: Zyklus 5	
0x00030306	Read/Write	D: Schrittmotor	UINT8	1		Bit-Maske: Zyklus 6	
0x00030307	Read/Write	D: Schrittmotor	UINT8	1		Bit-Maske: Zyklus 7	
0x00030308	Read/Write	D: Schrittmotor	UINT8	1		Bit-Maske: Zyklus 8	
0x00030310	Read/Write	D: Schrittmotor	UINT8	1		Bit-Maske: Haltestrom	

**3.1.1.5.4.4.2**     ***"Index-Offset" Spezifikation für Achsenzustand (Index-Group  
0x4100 + ID)***

Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n00000	Read	every (Online Struktur für Achsdaten)	{			ACHS-ONLINE-STRUKTUR (NC/CNC)	Geändert ab TC3 Nicht oszilloskopierbar! (NCAXISSTAT E_ONLINESTRU CT)
			INT32	1		Fehlerstatus	
			INT32			Reserviert	
			REAL64	z. B. mm		Istposition	
			REAL64	z. B. Grad		Modulo-Istposition	
			REAL64	z. B. mm		Sollposition	
			REAL64	z. B. Grad		Modulo-Sollposition	
			REAL64	z. B. mm/s		Optional: Istgeschwindigkeit	
			REAL64	z. B. mm/s		Sollgeschwindigkeit	
			UINT32	%	0...1000000	Geschwindigkeitsoverride (1000000 == 100%)	
			UINT32			Reserviert	
			REAL64	z. B. mm		Schleppabstand Position	
			REAL64	z. B. mm		PeakHold-Wert für max. neg. Schleppabst. (Pos.)	
			REAL64	z. B. mm		PeakHold-Wert für max. pos. Schleppabst. (Pos.)	
			REAL64	%		Reglerausgabe in Prozent	
			REAL64	%		Gesamtausgabe in Prozent	
			UINT32	1	≥ 0	Achsstatus-Doppelwort	
			UINT32	1	≥ 0	Achssteuer-Doppelwort	
			UINT32	1	≥ 0	Slave Koppelstatus (Zustand)	
			UINT32	1	0; 1,2,3...	Achs-Regelkreis-Index	
			REAL64	z. B. mm/s^2		Istbeschleunigung	
			REAL64	z. B. mm/s^2		Sollbeschleunigung	
			REAL64	z. B. mm/s^3		Sollruck (neu ab TC3.1 B4013)	
			REAL64	z. B. 100% = 1000		Sollmoment bzw. Sollkraft („SetTorque“)	
			REAL64	z. B. 100% = 1000		Istmoment bzw. Istkraft (neu ab TC3.1 B4013)	
			REAL64	z.B. %/s		Sollmomentänderung bzw. Sollkraftänderung (zeitliche Ableitung des Sollmomentes bzw. der Sollkraft) (ab TC3.1 B4024.2)	
			REAL64	z. B. 100% = 1000		Additives Sollmoment bzw. additive Sollkraft („TorqueOffset“) (ab TC3.1 B4024.2)	
... }			256 Byte				
0x00000001	Read	every	UINT32	1		Fehlercode Achsstatus	Symbolischer Zugriff: „ErrState“

Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n00009	Read	every	UINT32	1	≥ 0	Sollzykluszähler (SAF-Timestamp)	
0x00n0000A	Read	every	REAL64	z. B. mm		Sollposition	<i>Symbolischer Zugriff: „SetPos“</i>
0x00n0000B	Read	every	REAL64	z. B. GRAD		Modulo-Sollposition	<i>Symbolischer Zugriff: „SetPosModulo“</i>
0x00n0000C	Read	every	INT32	1		Modulo-Sollumdrehung	
0x00n0000D	Read	every	REAL64	1	[-1.0, 0.0, 1.0]	Sollfahrrichtung	
0x00n0000E	Read	every	REAL64	z. B. mm/s		Sollgeschwindigkeit	<i>Symbolischer Zugriff: „SetVelo“</i>
0x00n0000F	Read	every	REAL64	z. B. mm/s^2		Sollbeschleunigung	<i>Symbolischer Zugriff: „SetAcc“</i>
0x00n00010	Read	every	REAL64	z. B. mm/s^3		Sollruck (zeitliche Ableitung der Sollbeschleunigung)	<i>Symbolischer Zugriff: „SetJerk“</i>
0x00n00011	Read	every	REAL64	z. B. Nm bzw. N, z. B. 100%=1 000		Sollmoment (rot. Motor) bzw. Sollkraft (Linearmotor) ("SetTorque")	NEU ab TC3.1 B4022 <i>Symbolischer Zugriff: „SetTorque“</i>
0x00n00012	Read	every	REAL64	1		Soll-Koppelfaktor (Soll-Getriebeverhältnis)	
0x00n00013	Read	every	REAL64	z. B. mm		Voraussichtliche Zielposition (Target Position)	
0x00n00014	Read	Servo	{			Verbleibende Fahrzeit und Restweg (SERVO):	Immer an SAF-Port 501!
			REAL64	s	≥ 0	Verbleibende Fahrzeit	
			REAL64	z. B. mm	≥ 0	Verbleibender Restweg	
			}				
0x00n00015	Read	every	UINT32	1	≥ 0	Soll-Kommandonummer	<i>Symbolischer Zugriff: „CmdNo“</i>
0x00n00016	Read	Servo	REAL64	s	≥ 0	Positionierzeit des letzten Fahrauftrags (Start → Zielpositionsfenster)	
0x00n00017	Read	Servo	REAL64	%	[0.0...1.0] 1.0=100%	Soll-Overridewert für Geschwindigkeit  Anm.: vorerst nur für FIFO-Gruppe implementiert	NEU ab TC3.1 B4020
0x00000018	ReadWrite	Servo	<b>Write</b>			Lesen der "Stopp Informationen" (Stopp-Weg, Stopp-Zeit)	Immer an SAF-Port 501!
			REAL64	z. B. mm/s^2	≥ 0	Verzögerung für Achs-Stopp	
			REAL64	z. B. mm/s^3	≥ 0	Ruck für Achs-Stopp	
			<b>Read</b>				
			REAL64	z. B. mm	≥ 0	Stopp-Weg	
REAL64	s	≥ 0	Stopp-Zeit				
0x00n0001A	Read	every	REAL64	z. B. mm		Unkorrigierte Sollposition	



Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n0001D	Read	every	REAL64	1	[-1.0, 0.0, 1.0]	Unkorrigierte Sollfahrrichtung	
0x00n0001E	Read	every	REAL64	z. B. mm/s		Unkorrigierte Sollgeschwindigkeit	
0x00n0001F	Read	every	REAL64	z. B. mm/s^2		Unkorrigierte Sollbeschleunigung	
0x00000020	Read	every	UINT32	1	s. ENUM	Koppelstatus (Zustand)	
0x00000021	Read	every	UINT32	1	≥ 0	Koppeltabellen-Index	
0x00000022	Read	Servo Master/ Slave-Kopplung Typ: LINEAR, (&SPECIAL)	{			Lesen der Kopplungsparameter (SERVO):	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Getriebefaktor	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve	
}							
0x00000023	Read	Servo Master/ Slave-Kopplung Typ: LINEAR, (&SPECIAL)	REAL64	1	[±1000000.0]	Lesen des Getriebefaktors (SERVO) Typ: LINEAR	
0x00000024	Read	Servo	UINT32	1	≥ 0	Nummer/Index des aktiven Achsregelkreises (Trippel aus Encoder, Regler und Achsinterfaces)	
0x00000025	Read	Servo	UINT16	1	0/1	Externe Sollwertvorgabe über Achsinterface PLctoNC aktiv?	
0x00000026	Read	Servo Master/ Slave-Kopplung Typ: SYNCHRONIZI NG	REAL64 [64]	1	±∞	Lesen der charakteristischen Kennwerte des Slave-Aufsynchrosierungsprofils Typ: SYNCHRONIZING	Geändert ab TC3
0x00000027	ReadWrite	Servo Master/ Slave-Kopplung Typ: TABULAR, MF	<b>Write</b>			Lesen der "Tabellen-Kopplungsinformationen"	Nur Port 500! Geändert ab TC3
			VOID oder REAL64 oder DWORD, DWORD, REAL64	z. B. mm	±∞	- Keine Daten für die "aktuelle Information" - Optional für eine bestimmte "Master Achsposition" - Für eine bestimmte Tabellen-ID und optionale „Master Achsposition“ (TC 3.1 B4017)	
			<b>Read</b> REAL64 [32]		±∞	Lesen der Struktur für die Tabellen-Kopplungsinformationen [▶ 171]	

Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000028	ReadWrite	Servo Master/ Slave-Kopplung  Typ: MULTICAM (CamAddition)	<b>Write</b>			Lesen der "Multitabellen- Kopplungsinforma- tion" (CamAddition)	Nur Port 500!
			UINT32	1	$\geq 0$	Tabellen-ID auf die sich die Anfrage bezieht	
			<b>Read</b> 96 Byte			Lesen der Struktur für die Multitabellen- Kopplungsinforma- tion [► 171]	
0x00000029	Read	Servo	UINT32	1		verzögerter Fehlercode (Fehlervorwarnung) im Falle einer verzögerten Fehlerreaktion (s. Bit <i>ErrorPropagationDelay</i> <i>ed</i> )	
0x0000002A	Read	Servo	REAL64	z. B. mm	$\pm\infty$	Positionsdifferenz beim Überblenden von Sollpositionen auf Istpositionen (Fading Anteil)	
0x0000002B	Read	Servo	REAL64	z. B. mm/s	$\pm\infty$	Relative Geschwindigkeit beim Überblenden von Sollpositionen auf Istpositionen (Fading Anteil)	
0x0000002C	Read	Servo	REAL64	z. B. mm/s <sup>2</sup>	$\pm\infty$	Relative Beschleunigung beim Überblenden von Sollpositionen auf Istpositionen (Fading Anteil)	
0x0000002D	Read	Servo	UINT32	1	$\geq 0$	Zähler für Initialisierungs- Kommando (InitializeCommandCo unter)	NEU
0x0000002E	Read	Servo	UINT32	1	$\geq 0$	Zähler für Reset- Kommando (ResetCommandCoun ter)	NEU
0x00000030	Read	Servo	REAL64	z. B. Nm/s bzw. N/s	$\pm\infty$	Sollmomentänderung (rot. Motor) bzw. Sollkraftänderung (Linearmotor)  (zeitliche Ableitung des Sollmomentes bzw. der Sollkraft)	NEU ab TC3.1 B4024
0x00000031	Read/Write	Servo	REAL64	z. B. Nm bzw. N,  z. B. 100%=1 000	$\pm\infty$	Additives Sollmoment (rot. Motor) bzw. additive Sollkraft (Linearmotor) für Vorsteuerung. ("TorqueOffset")	Ab TC3.1 B4024.2  <i>Symbolischer Zugriff: „TorqueOffset“</i>
0x00000040	Read	Servo	UINT32	1	$\geq 0$	Zähler für Korrektur der NC-Sollwerte bei Dateninkonsistenz (Aktivierung mit Idx- Group 0x1000 und Idx-Offset 0x0020)	NEU ab TC3.1 B4020
0x00000050	Read	every	UINT32	1		Sollfahrphase (SWGenerator)	Nicht oszilloskopierba r!

Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000051	Read	every	UINT16	1		Ist Achse deaktiviert ?	Nicht oszilloskopierbar!
0x00n00060	Read/Write	every (Online Sollwert - Struktur) 40 Byte	{ REAL64 REAL64 REAL64 REAL64 REAL64 }	z. B. mm z. B. mm/s z. B. mm/s <sup>2</sup> 1 z. B. mm/s <sup>3</sup>	   [-1.0, 0.0, 1.0]	Einfache ACHS-SOLLWERT-STRUKTUR (NC/CNC) Sollposition Sollgeschwindigkeit Sollbeschleunigung/Sollverzögerung Sollfahrtrichtung Sollruck	Nicht oszilloskopierbar!  ab TC 3.1 B4022.30
0x00n00060	Read/Write	every (Online Sollwert - Struktur) 56 Byte	{ REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 }	z. B. mm z. B. mm/s z. B. mm/s <sup>2</sup> 1 z. B. mm/s <sup>3</sup> Nm bzw. N bzw. % Nm/s bzw. N/s bzw. %/s	   [-1.0, 0.0, 1.0]	Erweiterte ACHS-SOLLWERT-STRUKTUR (NC/CNC) Sollposition Sollgeschwindigkeit Sollbeschleunigung/Sollverzögerung Sollfahrtrichtung Sollruck Sollmoment bzw. Sollkraft zeitliche Ableitung des Sollmomentes bzw. der Sollkraft (Rampe)	Nicht oszilloskopierbar!  ab TC 3.1 B4022.29
0x00n00061	Read/Write	every (Online Dynamik Sollwert-Struktur) 32 Byte	{ REAL64 REAL64 REAL64 REAL64 }	z. B. mm/s z. B. mm/s <sup>2</sup> 1 z. B. mm/s <sup>3</sup>	  [-1.0, 0.0, 1.0]	ACHS-DYNAMIK-SOLLWERT STRUKTUR (NC/CNC) Sollgeschwindigkeit Sollbeschleunigung/Sollverzögerung Sollfahrtrichtung Sollruck	ab TC 3.1 B4022.30
0x00n00061	Read/Write	every (Online Dynamik Sollwert-Struktur) 48 Byte	{ REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 }	z. B. mm/s z. B. mm/s <sup>2</sup> 1 z. B. mm/s <sup>3</sup> Nm bzw. N bzw. % Nm/s bzw. N/s bzw. %/s	  [-1.0, 0.0, 1.0]	ACHS-DYNAMIK-SOLLWERT STRUKTUR (NC/CNC) Sollgeschwindigkeit Sollbeschleunigung/Sollverzögerung Sollfahrtrichtung Sollruck Sollmoment bzw. Sollkraft zeitliche Ableitung des Sollmomentes bzw. der Sollkraft (Rampe)	ab TC 3.1 B4022.29

Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n00062	Read/Write	every (Online TORQUE Sollwert-Struktur) 16 Byte	{			TORQUE-SOLLWERT STRUKTUR (NC/CNC)	ab TC 3.1 B4022.30
			REAL64	Nm bzw. N bzw. %		Sollmoment bzw. Sollkraft	
			REAL64	Nm/s bzw. N/s bzw. %/s		zeitliche Ableitung des Sollmomentes bzw. der Sollkraft (Rampe)	
			}				
0x00000063	ReadWrite	only for SERCOS/SoE and CANopen/CoE	<b>Write</b>			Lese aktiven „Drive Operation Mode“	NEU ab TC 3.1 B4022 (NC 4443) Immer an SAF-Port 501!
			UINT32	1		Reserve	
			UINT32	1		Reserve	
			<b>Read</b>				
			INT32	ENUM [▶_169] (s. Anhang)	[0; 1, 2, 3, ...] Special cases: ≥ 100: SoE <0: CoE	Aktuell aktiver „Drive Operation Mode“ (generische Modi)	
			UINT32	1		Reserve	
0x00n10002	Read	every (Encoder)	REAL64	z. B. mm		Istposition (verrechnet mit Istpositionskorrekturwert) n = 0: Standardencoder der Achsen > 0: n-ter Encoder der Achse (optional)	Symbolischer Zugriff: „ActPos“
0x00n10003	Read	every (Encoder)	REAL64	z. B. GRAD		Modulo-Istposition	Symbolischer Zugriff: „ActPosModulo“
0x00n10004	Read	every (Encoder)	INT32	1		Modulo-Istumdrehung	
0x00n10005	Read	every (Encoder)	REAL64	z. B. mm/s		Optional: Istgeschwindigkeit	Symbolischer Zugriff: „ActVelo“
0x00n10006	Read	every (Encoder)	REAL64	z. B. mm/s <sup>2</sup>		Optional: Istbeschleunigung	Symbolischer Zugriff: „ActAcc“
0x00n10007	Read	every (Encoder)	INT32	INC		Geber-Istinkremente	
0x00n10008	Read	every (Encoder)	INT64	INC		Software - Istinkrementalzähler	
0x00n10009	Read	every (Encoder)	UINT16	1	0/1	Referenzierflag ("Eichflag")	
0x00n1000A	Read	every (Encoder)	REAL64	z. B. mm		Istpositionskorrekturwert (Meßsystemfehlerkorrektur)	
0x00n1000B	Read	every (Encoder)	REAL64	z. B. mm		Istposition ohne Istpositionskorrekturwert	Nicht oszilloskopierbar!
0x00n10010	Read	every (Encoder)	REAL64	z. B. mm/s		Istgeschwindigkeit ohne Istpositionskorrekturwert	
0x00n10012	Read	every (Encoder)	REAL64	z. B. mm		Ungefilterte Istposition (verrechnet mit Istpositionskorrekturwert)	

Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsreich	Beschreibung	Anmerkung
0x00n10014	Read	Encoder: SoE, CoE, MDP 742	REAL64	z. B. mm/s		Optional: Antriebs-Istgeschwindigkeit (direkt vom SoE, CoE oder MDP 742 Drive übertragen)	NEU ab TC3.1 B4020.30
0x00n10015	Read	every (Encoder)	REAL64	z. B. mm/s		Optional: Ungefilterte Istgeschwindigkeit	
0x00n10017	Read		REAL64	z. B. mm		Auslesen des MC_SetPosition-Offsets	
0x00n10101	Read	INC (Encoder)	REAL64	z. B. mm		Rücklesen der Positionsdifferenz zwischen Aktivieren und Gültig werden des internen Hardwarelatches	Nicht oszilloskopierbar!
0x00n20001	Read	R: every	INT32	1		Fehlerstatus des Reglers n = 0: Standardregler der Achsen > 0: n-ter Regler der Achse (optional)	
0x00n20002	Read	R: every	REAL64	z. B. mm/s		Reglerausgabe in absoluten Einheiten	<i>Symbolischer Zugriff: „CtrlOutput“</i>
0x00n20003	Read	R: every	REAL64	%		Reglerausgabe in Prozent	Nicht oszilloskopierbar!
0x00n20004	Read	R: every	REAL64	V		Reglerausgabe in Volt	Nicht oszilloskopierbar!
0x00n2000D	Read	R: every	REAL64	z. B. mm		Schleppabstand Position (ohne Totzeitkompensation)	Base Unit
0x00n2000F	Read	R: every	REAL64	z. B. mm		Schleppabstand Position (mit Totzeitkompensation)	<i>Symbolischer Zugriff: „PosDiff“</i>
0x00n20010	Read	R: every	REAL64	z. B. mm		PeakHold-Wert für maximalen negativen Schleppabstand der Position	
0x00n20011	Read	R: every	REAL64	z. B. mm		PeakHold-Wert für minimalen positiven Schleppabstand der Position	
0x00n20012	Read	R: every	REAL64	z. B. mm/s		Schleppabstand Geschwindigkeit	Nicht implementiert!
0x00n20021	Read	R: every	REAL64	z. B. mm		Differenz (Abweichung) der Positionsschleppabstände zwischen Master- und Slaveachse (Master- minus Slaveschleppabstand)	<i>Symbolischer Zugriff: „PosDiffCouple“</i>
0x00n20022	Read	R: every	REAL64	z. B. mm		PeakHold-Wert für maximale negative Differenz der Schleppabstände (Position) zwischen Master- und Slaveachse	Base Unit

Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n20023	Read	R: every	REAL64	z. B. mm		PeakHold-Wert für maximale positive Differenz der Schleppabstände (Position) zwischen Master- und Slaveachse	Base Unit
0x00n20101	Read	R: P/PID (Pos.)	REAL64	z. B. mm/s		P-Anteil des Reglers in absoluten Einheiten	
0x00n20102	Read	R: PID (Pos.)	REAL64	z. B. mm/s		I-Anteil des Reglers in absoluten Einheiten	
0x00n20103	Read	R: PID (Pos.)	REAL64	z. B. mm/s		D-Anteil des Reglers in absoluten Einheiten	
0x00n20104	Read	R: PID (Pos.)	UINT16	1	0/1	Begrenzung des I-Anteils aktiv?	
0x00n20105	Read	R: PID (Pos.)	UINT16	1	0/1	Begrenzung des D-Anteils aktiv?	
0x00n20106	Read	R: PID (Pos.)	UINT16	1	0/1	ARW-Massnahmen des I-Anteils aktiv? ARW: Anti Reset Windup	Nicht implementiert!
0x00n20110	Read	R: PID (Pos.)	REAL64	z. B. mm/s		Beschleunigungsvorsteuerung Yacc des Reglers in absoluten Einheiten  Anmerkung: Funktion abhängig vom Reglertyp!	Beschleunigung s- vorsteuerung
0x00n20111	Read	R: PP (Pos.)	REAL64	mm/s/ mm	≥0	Interne interpolierte Proportionalverstärkung kp bzw. kv	PP-Regler
0x00n20201	Read	R: P,PID (Geschw.)	REAL64	z. B. mm/s		Geschwindigkeitsanteil des Reglers	
0x00n20202	Read	R: P,PID (Geschw.)	REAL64	%		Geschwindigkeitsanteil des Reglers in Prozent	Nicht oszilloskopierbar!
0x00n20203	Read	R: P,PID (Geschw.)	REAL64	V		Geschwindigkeitsanteil des Reglers in Volt	Nicht oszilloskopierbar!
0x00n20201	Read	R: P/PID (Geschw.)	REAL64	z. B. mm/s		P-Anteil des Reglers in absoluten Einheiten	
0x00n20202	Read	R: P/ PID (Geschw.)	REAL64	z. B. mm/s		I-Anteil des Reglers in absoluten Einheiten	
0x00n20203	Read	R: P/ PID (Geschw.)	REAL64	z. B. mm/s		D-Anteil des Reglers in absoluten Einheiten	
0x00n20204	Read	R: P/ PID (Geschw.)	UINT16	1	0/1	Begrenzung des I-Anteils aktiv?	
0x00n20205	Read	R: P/ PID (Geschw.)	UINT16	1	0/1	Begrenzung des D-Anteils aktiv?	
0x00n20206	Read	R: P/ PID (Geschw.)	UINT16	1	0/1	ARW-Massnahmen des I-Anteils aktiv? (ARW: Anti Reset Windup)	
0x00n2020A	Read	R: P/ PID (Geschw.)	REAL64	z. B. mm/s		Gesamteingangsgröße des Geschwindigkeits-Reglers	
0x00n20A00	Read	R: PID (MW)	REAL64	%	[-1.0...1.0]	Verrechnung der Sollgeschwindigkeit (Vorsteuerung)	Reservierte Parameter!
0x00n20A01	Read	R: PID (MW)	REAL64	z. B. mm/s		P-Anteil des Reglers in absoluten Einheiten oder Prozent (je nach Ausgabegewichtung)	Reservierte Parameter!

Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsreich	Beschreibung	Anmerkung
0x00n20A02	Read	R: PID (MW)	REAL64	z. B. mm/s		I-Anteil des Reglers in absoluten Einheiten oder Prozent (je nach Ausgabegewichtung)	Reservierte Parameter!
0x00n20A03	Read	R: PID (MW)	REAL64	z. B. mm/s		D-Anteil des Reglers in absoluten Einheiten oder Prozent (je nach Ausgabegewichtung)	Reservierte Parameter!
0x00n20A04	Read	R: PID (MW)	UINT16	1	0/1	Begrenzung des I-Anteils aktiv?	Reservierte Parameter!
0x00n20A05	Read	R: PID (MW)	UINT16	1	0/1	Begrenzung des D-Anteils aktiv?	Reservierte Parameter!
0x00n20A06	Read	R: PID (MW)	UINT16	1	0/1	ARW-Massnahmen des I-Anteils aktiv? ARW: Anti Reset Windup	Reservierte Parameter!
0x00n20A10	Read	R: PID (MW)	REAL64	z. B. mm/s		Beschleunigungsvorsteuerung Yacc des Reglers in absoluten Einheiten	Reservierte Parameter!
0x00n30001	Read	D: every	INT32	1		Fehlerstatus des Drives	
0x00n30002	Read	D: every	REAL64	z. B. mm/s		Gesamtausgabe in absoluten Einheiten	<i>Symbolischer Zugriff: „DriveOutput“</i>
0x00n30003	Read	D: every	REAL64	%		Gesamtausgabe in Prozent	
0x00n30004	Read	D: every	REAL64	V		Gesamtausgabe in Volt	Nicht oszilloskopierbar!
0x00n30005	Read	D: every	REAL64	z. B. mm/s		PeakHold-Wert für maximale negative Gesamtausgabe	
0x00n30006	Read	D: every	REAL64	z. B. mm/s		PeakHold-Wert für maximale positive Gesamtausgabe	
0x00n30007	Read	D: every	REAL64	z. B. 100%=1000, z. B. Nm bzw. N		Istmoment bzw. Istkraft (typisch 100%=1000)	ab TC3.1 B4022 <i>Symbolischer Zugriff: „ActTorque“</i>
0x00n30008	Read	D: every	REAL64	z. B. Nm/s bzw. N/s	$\pm\infty$	Istmomentänderung bzw. Istkraftänderung (zeitliche Ableitung des Istmomentes bzw. der Istkraft)	ab TC3.1 B4024
0x00n30013	Read	D: every	REAL64	%		Gesamtausgabe in Prozent (nach nichtlinearer Kennlinie!)	
0x00n30014	Read	D: every	REAL64	V		Gesamtausgabe in Volt (nach nichtlinearer Kennlinie!)	Nicht oszilloskopierbar!
0x00n3011A	Read	D: Servo (Sercos, CANopen)	REAL64	z. B. mm		Optionale Ausgabefilterung: Gefilterte Sollposition	NEU Für Sercos, CANopen
0x00n3011E	Read	D: Servo (Sercos, CANopen)	REAL64	z. B. mm/s		Optionale Ausgabefilterung: Gefilterte Sollgeschwindigkeit	NEU Für Sercos, CANopen

Index-Offset (Hex)	Zugriff	Achsentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n3011F	Read	D: Servo (Sercos, CANopen)	REAL64	z. B. mm/s <sup>2</sup>		Optionale Ausgabefilterung: Gefilterte Sollbeschleunigung / Sollverzögerung	NEU Für Sercos, CANopen



**3.1.1.5.4.4.3**     ***"Index-Offset" Spezifikation für Achsfunktionen (Index-Group 0x4200 + ID)***

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Write	every	VOID			Reset Achse	Auch für FIFO-Achsen!
0x00000002	Write	every	VOID			Stop Achse	Auch für FIFO-Achsen!
0x00000003	Write	every	VOID			Clear Achse (Auftrag)	Auch für FIFO-Achsen!
0x00000004	Write	every	{			Emergency Stop (Notstopp mit geregelter Rampe)	Nur für PTP-Achsen!
			REAL64	z. B. mm/s <sup>2</sup>	> 0.0	Verzögerung (muss größer gleich der Originalverzögerung sein!)	
			REAL64	z. B. mm/s <sup>3</sup>	> 0.0	Ruck (muss größer gleich dem Originalruck sein!)	
			}				
0x00000005	Write	PTP-Achse	{			Parametrierbarer Stopp (mit geregelter Rampe)	Nur für PTP-Achsen! Reservierte Funktion, kein Standard!
			REAL64	z. B. mm/s <sup>2</sup>	> 0.0	Verzögerung	
			REAL64	z. B. mm/s <sup>3</sup>	> 0.0	Ruck	
			}				
0x00000009	Write	PTP-Achse	{			Orientierter Stopp (orientierte Endposition)	Nur für PTP-Achsen!
			REAL64	z. B. Grad	≥ 0.0	Modulo-Endposition (Modulo-Zielposition)	
			REAL64	z. B. mm/s <sup>2</sup>	> 0.0	Verzögerung (momentan nicht wirksam)	
			REAL64	z. B. mm/s <sup>3</sup>	> 0.0	Ruck (momentan nicht wirksam)	
}							
0x00000010	Write	every	VOID			Referenziere Achse ("Eichen")	
0x00000011	Write	every	{			Neue Endposition Achse	Geändert ab TC3
			UINT32	ENUM	s. Anhang	Endpositionstyp [► 161] (s. Anhang)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	±∞	Neue Endposition (Zielposition)	
}							
0x00000012	Write	every	{			Neue Endposition und neue Geschwindigkeit Achse	
			UINT32	ENUM	s. Anhang	Kommandotyp [► 161] (s. Anhang)	
			UINT32	ENUM	s. Anhang	Endpositionstyp [► 161] (s. Anhang)	
			REAL64	z. B. mm	±∞	Neue Endposition (Zielposition)	
			REAL64	z. B. mm/s	≥ 0.0	Neue Endgeschwindigkeit (angeforderte Fahrgeschwindigkeit)	
			REAL64	z. B. mm	±∞	Optional: Umschaltposition, ab der neues Fahrprofil aktiviert wird	
}							

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000015	Write	every	{			Neue Dynamikparameter für aktive Positionierung	
			REAL64	z. B. mm/s <sup>2</sup>	> 0.0	Beschleunigung	
			REAL64	z. B. mm/s <sup>2</sup>	> 0.0	Verzögerung	
			REAL64	z. B. mm/s <sup>3</sup>	> 0.0	Optional: Ruck (momentan nicht wirksam)	
			}				
0x00000016	ReadWrite	every SERVO	<b>Write</b> (80 byte)			Universeller Achsstart (UAS): Verschmelzung von Einzelkommandos wie z. B. Achsstart, und Online-Änderungen in Kombination mit "Buffer-Mode" (s. TcMc2.lib)	Immer an SAF-Port 501! Geändert ab TC3
			{				
			UINT32	ENUM	s. Anhang	Starttyp [ <a href="#">▶ 160</a> ] (s. Anhang)	
			UINT32	1	≥ 0	Bitmaske für Überprüfungen und Betriebsarten (Default-Wert: 0)	
			REAL64	z. B. mm	±∞	Endposition (Zielposition)	
			REAL64	z. B. mm/s	≥ 0.0	Geforderte Geschwindigkeit <i>Vrequ</i>	
			REAL64	z. B. mm/s <sup>2</sup>	≥ 0.0	Optional: Beschleunigung	
			REAL64	z. B. mm/s <sup>2</sup>	≥ 0.0	Optional: Verzögerung	
			REAL64	z. B. mm/s <sup>3</sup>	≥ 0.0	Optional: Ruck	
			UINT32	ENUM	s. Anhang	Buffer-Mode [ <a href="#">▶ 161</a> ] (Kommandozwischenspeicher)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	±∞	Optional: Blending-Position (Kommandoüberblendungsposition)	
			REAL64	z. B. mm/s	≥ 0.0	Optional: Segment-Anfangsgeschwindigkeit <i>V<sub>i</sub></i> ( $0 \leq V_i \leq V_{requ}$ )	
			REAL64	z. B. mm/s	≥ 0.0	Optional: Segment-Endgeschwindigkeit <i>V<sub>f</sub></i> ( $0 \leq V_f \leq V_{requ}$ )	
			}				
			<b>Read</b>				
			{				
UINT16	1	≥ 0	Kommando-Nummer (Job-Nummer)				
UINT16	1	≥ 0	Kommando-Status				
}							

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsreich	Beschreibung	Anmerkung	
0x00000017	ReadWrite	SERVO	Write(80 byte)			"Master/Slave Entkopplung" und "Universeller Achsstart (UAS)": Verschmelzung vom Abkoppelkommando einer Slaveachse (IdxOffset: 0x00000041) und nachfolgendem Universellen Achsstart (UAS) (IdxOffset: 0x00000016)	Noch nicht freigegeben!	
			{					
			UINT32	ENUM	s. Anhang		Starttyp [ <a href="#">▶ 160</a> ] (s. Anhang)	
			UINT32	1	$\geq 0$		Bitmaske für Überprüfungen und Betriebsarten (Default-Wert: 0)	
			REAL64	z. B. mm	$\pm\infty$		Endposition (Zielposition)	
			REAL64	z. B. mm/s	$\geq 0.0$		Geforderte Geschwindigkeit <i>Vrequ</i>	
			REAL64	z. B. mm/s <sup>2</sup>	$\geq 0.0$		Beschleunigung	
			REAL64	z. B. mm/s <sup>2</sup>	$\geq 0.0$		Verzögerung	
			REAL64	z. B. mm/s <sup>3</sup>	$\geq 0.0$		Ruck	
			UINT32	ENUM	s. Anhang		Buffer-Mode [ <a href="#">▶ 161</a> ] (Kommandozwischenspeicher)	
			UINT32				Reserve (TC3)	
			REAL64	z. B. mm	$\pm\infty$		Optional: Blending-Position (Kommandoüberblendungsposition)	
			REAL64	z. B. mm/s	$\geq 0.0$		Optional: Segment-Anfangsgeschwindigkeit <i>Vi</i> ( $0 \leq Vi \leq Vrequ$ )	
			REAL64	z. B. mm/s	$\geq 0.0$		Optional: Segment-Endgeschwindigkeit <i>Vf</i> ( $0 \leq Vf \leq Vrequ$ )	
			}					
			<b>Read</b>					
			{					
			UINT16	1	$\geq 0$		Kommando-Nummer (Job-Nummer)	
			UINT16	1	$\geq 0$		Kommando-Status	
			}					
0x00000018	Write	every	VOID			Aufhebung der Achssperre für Bewegungskommandos (TcMc2)		
0x00000019	Write	every	UINT32	1	> 0	Setze externen Achsfehler (Laufzeitfehler)	<b>Vorsicht bei Benutzung!</b>	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n0001A	Write	every	{			Setze Istposition Achse	<b>Vorsicht bei Benutzung!</b> Auch für FIFO-Achsen! Immer an SAF-Port 501! Geändert ab Tc3
			UINT32	ENUM	s. Anhang	Istpositionstyp [► 161] (s. Anhang)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	$\pm\infty$	Istposition für Achsen = 0: Standardencoder der Achse n > 0: n-ter Encoder der Achse (optional)	
			}				
0x00n0001B	Write	every	UINT32	1	0/1	Setze Referenzierflag ("Eichflag") n = 0: Standardencoder der Achse n > 0: n-ter Encoder der Achse (optional)	<b>Vorsicht bei Benutzung!</b> Auch für FIFO-Achsen!
0x00n0001C	Write	SERVO	{			Setze nur Istposition Achse, ohne Manipulation der Sollposition (auch für Slave und bei aktivem Verfahren)	<b>Vorsicht bei Benutzung!</b>
			UINT32	ENUM	s. Anhang	Istpositionstyp [► 161] (s. Anhang)	
			REAL64	z. B. mm	$\pm\infty$	Istposition für Achse n = 0: Standardencoder der Achsen > 0: n-ter Encoder der Achse (optional) <b>Vorsicht bei Benutzung!</b>	
						}	
0x00n0001D	Write	every	{			Antriebseitiges Istwertsetzen der Achse (Positionsinterface und Encoder-Offset von Null vorausgesetzt!) n = 0: Standardencoder der Achse n > 0: n-ter Encoder der Achse (optional)	<b>Vorsicht bei Benutzung!</b> Nur für CANopen!
			UINT32	ENUM	s. Anhang	Istpositionstyp [► 161] (s. Anhang)	
			REAL64	z. B. mm	$\pm\infty$	Istposition für Achse	
						}	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00n0001E	Write	every	{			Fliegendes Setzen eines neuen Encoder-Skalierungsfaktors (in Bewegung der Achse)	<b>Vorsicht bei Benutzung!</b> Immer an SAF-Port 501!  Geändert ab TC3
			UINT16	ENUM	1	Encoder-Skalierungsfaktortyp 1: Absolut 2: Relativ	
			UINT16			ControlWord	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm/ INC	[1.0E-8 ... 100.0]	Neuer Encoder-Skalierungsfaktor n = 0: Standardencoder der Achse n > 0: n-ter Encoder der Achse (optional)	
		}					
0x00n0001F	Write	every	{			Fliegendes Istwertsetzen der Achse (in Bewegung der Achse)	<b>Vorsicht bei Benutzung!</b> Immer an SAF-Port 501!
			UINT32	ENUM		Positionstyp für Fliegendes Istwertsetzen 1: Absolut 2: Relativ	
			UINT32	1		Kontrolldoppelwort für z. B. "Ablöschen des Schleppabstandes"	
			REAL64			Reserve	
			REAL64	z. B. mm	$\pm\infty$	Neue Istposition der Achse	
			UINT32			Reserve	
			UINT32			Reserve	
					}		
0x00000020	Write	every 1D-Start	{			Standard Achsstart:	Geändert ab TC3
			UINT32	ENUM	s. Anhang	Starttyp [► 160] (s. Anhang)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	$\pm\infty$	Endposition (Zielposition)	
			REAL64	z. B. mm/s	$\geq 0.0$	Geforderte Geschwindigkeit	
		}					

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000021	Write	every 1D-Start	{			Erweiterter Achsstart (SERVO):	Geändert ab Tc3
			UINT32	ENUM	s. Anhang	Starttyp [► 160] (s. Anhang)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	$\pm\infty$	Endposition (Zielposition)	
			REAL64	z. B. mm/s	$\geq 0.0$	Geforderte Geschwindigkeit	
			UINT32	0/1	0/1	Standardbeschleunigung?	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm/s <sup>2</sup>	$\geq 0.0$	Beschleunigung	
			UINT32	0/1	0/1	Standardverzögerung?	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm/s <sup>2</sup>	$\geq 0.0$	Verzögerung	
			UINT32	0/1	0/1	Standarddruck?	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm/s <sup>3</sup>	$\geq 0.0$	Ruck	
0x00000022	Write	SERVO(MW)	{			Spezieller Achsstart (SERVO):	Reservierte Startfunktion, kein Standard! Geändert ab TC3
			UINT32	ENUM	s. Anhang	Starttyp [► 160] (s. Anhang)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	$\pm\infty$	Endposition (Zielposition)	
			REAL64	mm/s	$\geq 0.0$	Geforderte Anfangsgeschwindigkeit	
			REAL64	z. B. mm	$\pm\infty$	Position, für neues Geschwindigkeitsniveau	
			REAL64	z. B. mm/s	$\geq 0.0$	Neues Endgeschwindigkeitsniveau	
			UINT32	0/1	0/1	Standardbeschleunigung?	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm/s <sup>2</sup>	$\geq 0.0$	Beschleunigung	
			UINT32	0/1	0/1	Standardverzögerung?	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm/s <sup>2</sup>	$\geq 0.0$	Verzögerung	
			UINT32	0/1	0/1	Standarddruck?	
UINT32			Reserve (TC3)				
REAL64	z. B. mm/s <sup>3</sup>	$\geq 0.0$	Ruck				
			}				

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000023	Write	SERVO	{			Start externe Sollwertvorgabe (Vorgabe durch zyklisches Achsinterface PLCtoNC)	Geändert ab TC3
			UINT32	ENUM	1: Absolut 2: Relativ	Starttyp [ <a href="#">▶ 160</a> ]	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	$\pm\infty$	Neue Endposition (Zielposition) optional!	
			REAL64			Reserve (TC3)	
		}					
0x00000024	Write	SERVO	VOID			Stop/Disable externe Sollwertvorgabe (zykl. Achsinterface PLCtoNC)	
0x00000025	Write	SERVO	{			Start Reversierbetrieb für Positionierung (SERVO):	Geändert ab TC3
			UINT32	ENUM	1	Starttyp [ <a href="#">▶ 160</a> ] (default: 1)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	$\pm\infty$	Endposition 1 (Zielposition)	
			REAL64	z. B. mm	$\pm\infty$	Endposition 2 (Zielposition)	
			REAL64	0/1	0/1	Geforderte Geschwindigkeit	
			REAL64	s	$\geq 0.0$	Pausenzeit (Idle time)	
		}					
0x00000026	Write	every	{			Start-Drive-Output	Geändert ab TC3
			UINT32	ENUM	s. Anhang	Ausgabetyyp [ <a href="#">▶ 168</a> ] (s. Anhang)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. %	$\pm\infty$	Geforderter Ausgabewert (z. B. %)	
		}					
0x00000027	Write	every	VOID			Stop-Drive-Output	
0x00000028	Write	every	{			Änderung/Wechsel des Drive-Outputs:	
			UINT32	ENUM	s. Anhang	Ausgabetyyp [ <a href="#">▶ 168</a> ] (s. Anhang)	
			REAL64	z. B. %	$\pm\infty$	Geforderter Ausgabewert (z. B. %)	
		}					
0x00000029	Write	every	VOID			Aktuellen Override-Wert instantan übernehmen und bis zur nächsten Overrideänderung einfrieren!	Reservierte Funktion, kein Standard!
0x0000002A	Write	every	{ 32 bytes }			Calculate and set encoder offset	Reservierte Funktion, kein Standard!
0x0000002B	ReadWrite	every	WriteData: s. 'UAS' ReadData: s. 'UAS'			Stop external setpoint generator and continuous endless motion ('UAS': Universal axis start)	Reservierte Funktion, kein Standard!
0x0000002C	Write	every	UINT32		$\geq 0$	Setze "Homing State" (zur internen Verwendung)	Neu ab TC3



Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000002D	ReadWrite	Servo	Write			Schaltet eine NC-gesteuerte Achse in den "Cyclic Synchronous Torque Mode" (CST) und setzt dafür einen Drehmoment-Sollwert.	<b>Gefahr bei Benutzung!</b> (* siehe Tabellenende)
			{				
			UINT32			Drehmoment-Achsen-Starttyp: 0x3001: Absolute 0x3002: Relative	
			UINT32	1 (bit array)		Interne Steuermaske (Bit-Array): 00000000_00000001 (Bit 0): Manuelles Drehmoment zur Initialisierung verwenden. 10000000_00000000 (Bit 31): Update/Refresh-Parameter für aktuelles Kommando im 'ContinuousUpdate'-Modus (fTorqueRamp, fVelocityLimitHigh, fVelocityLimitLow), cmd no nicht erhöhen.	
			UINT32	0/1	0/1	Mode: 0: Default (discrete) 1: ContinuousUpdate	
			UINT32	ENUM	siehe Anhang	Buffer-Mode [ <a href="#">▶ 161</a> ] nur ABORTING möglich	
			REAL64	Nm oder %	[0.0 ... 1.0E10]	Drehmoment-Zielwert (signed value)	
			REAL64	Nm/s oder %/s	[0.0 ... 1.0E10]	Drehmoment-Änderungsgeschwindigkeit	
			REAL64	z. B. mm/s	[0.0 ... 1.0E10]	Geschwindigkeitsbegrenzung hoch 'VelocityLimitHigh' muss größer oder gleich sein als 'VelocityLimitLow' (beide Werte können negativ sein).	
			REAL64	z. B. mm/s	[0.0 ... 1.0E10]	Geschwindigkeitsbegrenzung niedrig	
			REAL64	z. B. mm/s <sup>2</sup>	[0.0 ... 1.0E10]	Beschleunigung	
			REAL64	z. B. mm/s <sup>2</sup>	[0.0 ... 1.0E10]	Verzögerung	
			REAL64	Nm oder %	[0.0 ... 1.0E10]	Optional: Manueller Drehmoment-Startwert (sync value)	
			}				
			Read				
			{				
UINT16	1	>=0	Kommando-Nummer (Job-Nummer)				
UINT16	1	>=0	Kommando-Status				
}							
0x0000002E						Reserviert	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000002F						Reserviert	
0x00000030	Write	SERVO	{			Start Streckenkompensation (SERVO)	Betrifft nur ältere TwinCAT 2 Systeme
			UINT32	ENUM	s. Anhang	Kompensationstyp [►_162] (s. Anhang)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm/ s <sup>2</sup>	≥ 0.0	Max. Beschleunigungserhö- hung	
			REAL64	z. B. mm/ s <sup>2</sup>	≥ 0.0	Max. Verzögerungserhö- hung	
			REAL64	z. B. mm/s	> 0.0	Max. Erhöhungsgeschwindi- gkeit	
			REAL64	z. B. mm/s	> 0.0	Grundgeschwindigkeit des Prozesses	
			REAL64	z. B. mm	±∞	Ausgleichende Wegdifferenz	
			REAL64	z. B. mm	> 0.0	Weglänge für Kompensation	
			}				

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsreich	Beschreibung	Anmerkung
0x00000030	ReadWrite	SERVO liefert die wirklich umgesetzten Größen als Rückgabewerte zurück	{ <b>READ+WRITE:</b>			Start Streckenkompensation (SERVO) Anmerkung: nur in 'TcMc2.lib' bzw. 'Tc2_MC2.library' enthalten	Geändert ab TwinCAT 2 211R3  TwinCAT 3
			UINT32	ENUM	s. Anhang	Kompensationstyp [►_162] (s. Anhang)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm/s <sup>2</sup>	≥ 0.0	=> Max. Beschleunigungserhöhung  <= Liefert umgesetzte Beschleunigungserhöhung zurück (neu in 'TcMc2.lib' bzw. 'Tc2_MC2.library')	
			REAL64	z. B. mm/s <sup>2</sup>	≥ 0.0	=> Max. Verzögerungserhöhung  <= Liefert umgesetzte Verzögerungserhöhung zurück (neu in 'TcMc2.lib' bzw. 'Tc2_MC2.library')	
			REAL64	z. B. mm/s	> 0.0	=> Angeforderte max. Erhöhungsgeschwindigkeit  <= Liefert umgesetzte Erhöhungsgeschwindigkeit zurück	
			REAL64	z. B. mm/s	> 0.0	Grundgeschwindigkeit des Prozesses	
			REAL64	z. B. mm	±∞	=> Angeforderte auszugleichende Wegdifferenz  <= Liefert umgesetzte Wegdifferenz zurück	
			REAL64	z. B. mm	> 0.0	=> Angeforderte max. Weglänge für Kompensation  <= Liefert umgesetzte Weglänge zurück	
			UINT32	1	≥ 0	<= Liefert Warnungs-ID (z. B. 0x4243) zurück	
			UINT32			Reserve (TC3)	
0x00000031	Write	SERVO	VOID			Stopp Streckenkompensation (SERVO)	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000032	Write	SERVO	{			Start Reversierbetrieb mit Geschwindigkeitssprüngen (SERVO): (kann zur Ermittlung der Geschwindigkeitssprungantwort verwendet werden)	Geändert ab TC3
			UINT32	ENUM	1	Starttyp [ <a href="#">▶ 160</a> ] (default: 1)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm/s	$\pm\infty$	Geforderte Geschwindigkeit 1 (auch negative Werte erlaubt)	
			REAL64	z. B. mm/s	$\pm\infty$	Geforderte Geschwindigkeit 2 (auch negative Werte erlaubt)	
			REAL64	s	> 0.0	Fahrzeit für Geschwindigkeit 1 und 2	
			REAL64	s	$\geq 0.0$	Pausenzeit (Idle time)	
			UINT32	1	0, 1,2,3...	Optional: Anzahl der Wiederholungen, Default "0": zeitlich unbegrenzt	
			UINT32			Reserve (TC3)	
0x00000033	Write	SERVO	{			Sinus Oscillation Sequence  - used as single sinus oscillation (sinus generator)  - used as sinus oscillation sequence (e.g. for bode plot)	Geändert ab TC3
			UINT32	ENUM	1	Starttyp [ <a href="#">▶ 160</a> ] (fixed to start type 1 yet)	
			UINT32			Reserve (TC3)	
			REAL64	e.g. mm/s	> 0.0	Base amplitude (e.g. 2.5 mm/s)	
			REAL64	Hz	[0.0 .... 10.0]	Base frequency (e.g. 1.953125 Hz)	
			REAL64	e.g. mm/s	$\geq 0.0$	Start amplitude at begin (e.g. 0.0 mm/s)	
			REAL64	e.g. mm/REV	> 0.0	Feed constant motor (per motor turn) (e.g. 10.0 mm/REV)	
			REAL64	Hz	$\geq 1.0$	Frequency range: start frequency (e.g. 20.0 Hz)	
			REAL64	Hz	$\leq 1/(2*dT)$	Frequency range: stop frequency (e.g. 500.0 Hz)	
			REAL64	s	> 0.0	Step duration (e.g. 2.048s)	
			UINT32	1	[1 ... 200]	Number of measurements (step cycles) (e.g. 20)	
			UINT32	1		Number of parallel measurements (e.g. 1) not used yet!	
			}				

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000034	Write	SERVO	{			Phasing - Start Phasing - Stop Phasing	
			UINT32	ENUM	1	PhasingType: 1: ABSOLUTE 2: RELATIVE 4096: STOP	
			UINT32	1	≥ 0	Control Mask Bit 0: Continuous Update	
			UINT32	1	≥ 0	Master axis ID (for multi master)	
			UINT32			Reserve	
			REAL64	e.g. mm	±∞	Phase shift	
			REAL64	e.g. mm/s	> 0.0	Velocity	
			REAL64	e.g. mm/ s <sup>2</sup>	≥ 0.0	Acceleration	
			REAL64	e.g. mm/ s <sup>2</sup>	≥ 0.0	Deceleration	
			REAL64	e.g. mm/ s <sup>3</sup>	≥ 0.0	Jerk	
			REAL64[4]			Reserve	
			UINT32			Reserve	
			UINT32	1	ENUM	Buffer mode (NOT IMPLEMENTED)	
			REAL64	e.g. mm	±∞	Blending position (NOT IMPLEMENTED)	
			}				

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000040 (0x00n00040)	Write	Master/Slave- Kopplung: (SERVO)	{			Master/Slave- Kopplung (SERVO):	Erweiterung für "Fliegende Säge!"  Winkel >0.0 und £ 90.0 Grad (Parallelsäge: 90.0 Grad)
			UINT32	ENUM	s. Anhang	Slavetyp [▶ 162]/ Kopplungstyp (s. Anhang)	
			UINT32	1	[1...255]	Achs-ID der Masterachse/Gruppe	
			UINT32	1	[0...8]	Subindex n der Masterachse (Default- Wert: 0)	
			UINT32	1	[0...8]	Subindex n der Slaveachse (Default- Wert: 0)	
			REAL64	1	[±1000000.0]	Parameter 1:Linear: Getriebefaktor  FlySawVelo: Reserve  FlySaw: Abs. Synchronposition Master [mm]	
			REAL64	1	[±1000000.0]	Parameter 2:Linear: Reserve  FlySawVelo: Reserve  FlySawPos: Abs. Synchronposition Slave [mm]	
			REAL64	1	[±1000000.0]	Parameter 3:Linear: Reserve  FlySawVelo: Neigungswinkel in [GRAD]  FlySawPos: Neigungswinkel in [GRAD]	
			REAL64	1	[±1000000.0]	Parameter 4:Linear: Reserve  FlySawVelo: Getriebefaktor  FlySawPos: Getriebefaktor	
}							

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung			
0x00000040 (0x00n00040)	Write	Master/Slave- Kopplung: (SERVO)	{			Master/Slave- Kopplung (SERVO):	Multi Master Kopplung (MC_GearInMul- tiMaster)  Version V1 und V2 Geändert ab Tc3			
			UINT32	ENUM	s. Anhang	Slavetyp  ▶ 162 / Kopplungstyp (s. Anhang)				
			UINT32	1	[1...255]	Achs-ID der Masterachse/Gruppe				
			UINT32	1	[1...8]	Subindex n der Masterachse (Default- Wert: 0)				
			UINT32	1	[1...8]	Subindex n der Slaveachse (Default- Wert: 0)				
			UINT32	1	[0...255]	Achs-ID Master 2				
			UINT32	1	[0...255]	Achs-ID Master 3				
			UINT32	1	[0...255]	Achs-ID Master 4				
			UINT32	1	[0...255]	Reserve (Achs-ID Master 5)				
			UINT32	1	[0...255]	Reserve (Achs-ID Master 6)				
			UINT32	1	[0...255]	Reserve (Achs-ID Master 7)				
			UINT32	1	[0...255]	Reserve (Achs-ID Master 8)				
			UINT32			Reserve (TC3)				
			REAL64	z. B. mm/ s^2		Maximale Beschleunigung/ Verzögerung der Slaveachse				
			UINT32	1	≥ 0	Steuermaske, bisher nicht verwendet (check and operation mode for profile)				
			UINT32			Reserve (TC3)				
			Erweiterung V2 (Optional):							
			REAL64	z. B. mm/ s^2	≥ 0.0	Maximale Verzögerung der Slaveachse				
			REAL64	z. B. mm/ s^3	≥ 0.0	Maximaler Ruck der Slaveachse				
			REAL64	z. B. mm/s	≥ 0.0	Maximale Geschwindigkeit der Slaveachse				
REAL64			Reserve							
REAL64			Reserve							
} 64 bzw. 104 Byte										
0x00000041	Write	Master/Slave- Entkopplung (SERVO)	VOID			Master/Slave Entkopplung (SERVO)				

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000041	Write	Master/Slave- Entkopplung mit konfigurierbarer Folgefunktion (SERVO)	{			Master/Slave- Entkopplung mit konfigurierbarer Folgefunktion (z. B. neue Endposition, neue Geschwindigkeit, Stop, E-Stop) (SERVO)	Noch nicht freigegeben!  Geändert ab TC3
			UINT32	ENUM	s. Anhang	Entkopplungstyp [►_162] (s. Anhang)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	$\pm\infty$	Optional: Neue Endposition	
			REAL64	z. B. mm/s	$> 0.0$	Optional: Neue geforderte Geschwindigkeit	
			REAL64	z. B. mm/ s <sup>2</sup>	$\geq 0.0$ (0: Default)	Optional: Beschleunigung für neue Endposition, neue Geschwindigkeit und Emergency Stop (E-Stop)	
			REAL64	z. B. mm/ s <sup>2</sup>	$\geq 0.0$ (0: Default)	Optional: Verzögerung für neue Endposition, neue Geschwindigkeit und Emergency Stop (E-Stop)	
			REAL64	z. B. mm/ s <sup>3</sup>	$\geq 0.0$ (0: Default)	Optional: Ruck für neue Endposition, neue Geschwindigkeit und Emergency Stop (E-Stop)	
		}					
0x00000042	Write	Master/Slave- Kopplung Typ: LINEAR (&SPECIAL)	{			Änderung der Kopplungsparameter (SERVO):	
			REAL64	1	$[\pm 1000000.0]$	Parameter 1: Linear: Getriebefaktor	
			REAL64	1	$[\pm 1000000.0]$	Parameter 2: Linear: Reserve	
			REAL64	1	$[\pm 1000000.0]$	Parameter 3: Linear: Reserve	
			REAL64	1	$[\pm 1000000.0]$	Parameter 4: Linear: Reserve	
					}		
0x00000043	Write	Master/Slave- Tabellenkopplung Typ: TABULAR	{			Änderung der Tabellen- Kopplungsparameter (SERVO):	
			REAL64	mm	$\pm\infty$	Slave-Positionsoffset	
			REAL64	mm	$\pm\infty$	Master-Positionsoffset	
					}		
0x00000043	Write	Master/Slave- Tabellenkopplung Typ: TABULAR und "Motion Function"	{			Änderung der Tabellen- Kopplungsparameter (SERVO):	Auch für "Motion Function"
			REAL64	mm	$\pm\infty$	Slave-Positionsoffset	
			REAL64	mm	$\pm\infty$	Master-Positionsoffset	
			REAL64	1	$\pm\infty (<> 0.0)$	Slave- Positionsskalierung	
			REAL64	1	$\pm\infty (<> 0.0)$	Master- Positionsskalierung	
					}		



Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000043	Write	Master/Slave- Tabellen- Kopplung Typ: TABULAR	{			Änderung der Tabellen- Kopplungsparameter (SERVO):	
			REAL64	mm	$\pm\infty$	Slave-Positionsoffset	
			REAL64	mm	$\pm\infty$	Master-Positionsoffset	
			REAL64	1	$\pm\infty$ (<> 0.0)	Slave- Positionsskalierung	
			REAL64	1	$\pm\infty$ (<> 0.0)	Master- Positionsskalierung	
			REAL64	z. B. mm	$\pm\infty$	Absolute Master- Aktivierungsposition	
		}					
0x00000044	Write	Slave-Stop (SERVO)	VOID			Stopp der "Fliegende Säge" (SERVO)	Nur für "Fliegende Säge"
0x00000045 (0x00n00045)	Write	Master/Slave- Tabellenkopplung (SERVO)	{			Master/Slave- Tabellenkopplung (SERVO):	
			UINT32	ENUM	s. Anhang	Slavetyp/ Kopplungstyp [► 162] (s. Anhang)	
			UINT32	1	[1...255]	Achs-ID der Masterachse	
			UINT32	1	[0...8]	Subindex n der Masterachse (Default- Wert: 0)	
			UINT32	1	[0...8]	Subindex n der Slaveachse (Default- Wert: 0)	
						<b>SOLO-TABELLEN- ABSCHNITT</b>	
			REAL64	mm	$\pm\infty$	Slave-Positionsoffset (Typ: TABULAR)	
			REAL64	mm	$\pm\infty$	Master-Positionsoffset (Typ: TABULAR)	
			UINT32	1	[0, 1]	Slavepositionen absolut (Typ: TABULAR)	
			UINT32	1	[0, 1]	Masterpositionen absolut (Typ: TABULAR)	
			UINT32	1	[1...255]	Tabellen-ID der Koppeltabelle (Typ: TABULAR)	
						<b>MULTI-TABELLEN- ABSCHNITT</b>	
			UINT16	1	[0...8]	Anzahl der Tabellen (Typ: MULTITAB)  Anmerkung: Missbraucht als Interpolations-Typ für Solo-Tabellen	
			UNIT16	1	[0...8]	Anzahl der Profil- Tabellen (Typ: MULTITAB)	
UNIT32[8]	1	[1...255]	Tabellen-IDs der Koppeltabellen (Typ: MULTITAB)				
		}					
0x00000046	Write	Master/Slave- Multitabellen	UINT32	1	[1...255]	Aktivierung Korrekturtabelle Korrektur-Tabellen-ID	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000046	Write	Master/Slave- Multitabellen	{			Aktivierung Korrekturtabelle	Geändert ab TC3
			UINT32	1	[1...255]	Korrektur-Tabellen-ID	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	$\pm\infty$	Absolute Master Aktivierungsposition	
			}				
0x00000047	Write	Master/Slave- Multitabellen	UINT32	1	[1..255]	Deaktivierung Profiltabelle am Zyklusende Tabellen- ID der aktuellen monozyklischen Profiltabelle	
0x00000048	ReadWrite	Master/Slave- Multitabellen	Write: UINT32	1	[1..255]	Lesen des letzten Korrekturoffsets: Tabellen-ID der Korrekturtabelle	
			Read: REAL32	z. B. mm	$\pm\infty$	Offset durch Abfahren der Korrekturtabelle mit der entsprechenden Tabellen-ID	
0x00000049	Write	Master/Slave- Tabellenkopplu- ng Typ: TABULAR	REAL64	1	$\pm\infty$	Ändern der Slave- Tabellenskalierung Skalierungsfaktor der Slave-Tabellenspalte (Default-Wert: 1.0)	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000004A(0x00n0004A)	Write	Master/Slave-Universelle-Tabellenkopplung (SERVO)	{			Master/Slave-Solo-Tabellenkopplung (SERVO):	Geändert ab TC3
			UINT32	ENUM	s. Anhang	Slavetyp/ Kopplungstyp  ▶ 1621 (s. Anhang)	
			UINT32	1	[1...255]	Achs-ID der Masterachse	
			UINT32	1	[0...8]	Subindex n der Masterachse (Default:-Wert:: 0)	
			UINT32	1	[0...8]	Subindex n der Slaveachse (Default:-Wert: 0)	
			UINT32	1	1...255]	Tabellen-ID der Koppeltabelle (Typ: TABULAR)	
			UINT32	1		Tabellen-Interpolationstyp	
			REAL64	mm	$\pm\infty$	Slave-Positionsoffset (Typ: TABULAR)	
			REAL64	mm	$\pm\infty$	Master-Positionsoffset (Typ: TABULAR)	
			REAL64	mm	$\pm\infty$	Slave-Positionsskalierung (Typ: TABULAR)	
			REAL64	mm	$\pm\infty$	Master-Positionsskalierung (Typ: TABULAR)	
			UINT32	1	[0,1]	Slavepositionen absolut ? (Typ: TABULAR)	
			UINT32	1	[0,1]	Masterpositionen absolut ? (Typ: TABULAR)	
			UINT32	ENUM	s. Anhang	Aktivierungstyp der Änderung: 0: 'instantaneous' (default) 1: 'at master cam position' 2: 'at master axis position' 3: 'next cycle'	
			UINT32			Reserve (TC3)	
			REAL64	mm	$\pm\infty$	Aktivierungsposition	
			UINT32	ENUM	s. Anhang	Master-Skalierungstyp: 0: user defined (default) 1: scaling with auto offset 2: off	
UINT32	ENUM	s. Anhang	Slave-Skalierungstyp: 0: user defined (default) 1: scaling with auto offset 2: off				
		}					

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung				
0x0000004A(0x00n0004A)	Write	Master/Slave-Universelle-Tabellenkopplung (SERVO)	{			Master/Slave-Solo-Tabellenkopplung (SERVO):	Geändert ab TC3				
			UINT32	ENUM	s. Anhang	Slavetyp/ Kopplungstyp  ▶ 162  (s. Anhang)					
			UINT32	1	[1...255]	Achs-ID der Masterachse					
			UINT32	1	[0...8]	Subindex n der Masterachse (Default:-Wert: 0)					
			UINT32	1	[0...8]	Subindex n der Slaveachse (Default:-Wert: 0)					
			UINT32	1	1...255]	Tabellen-ID der Koppeltabelle (Typ: TABULAR)					
			UINT32	1		Tabellen-Interpolationstyp					
			REAL64	mm	$\pm\infty$	Slave-Positionsoffset (Typ: TABULAR)					
			REAL64	mm	$\pm\infty$	Master-Positionsoffset (Typ: TABULAR)					
			REAL64	mm	$\pm\infty$	Slave-Positionsskalierung (Typ: TABULAR)					
			REAL64	mm	$\pm\infty$	Master-Positionsskalierung (Typ: TABULAR)					
			UINT32	1	[0,1]	Slavepositionen absolut ? (Typ: TABULAR)					
			UINT32	1	[0,1]	Masterpositionen absolut ? (Typ: TABULAR)					
			UINT32	ENUM	s. Anhang	Aktivierungstyp der Änderung: 0: 'instantaneous' (default) 1: 'at master cam position' 2: 'at master axis position' 3: 'next cycle'					
			UINT32			Reserve (TC3)					
			REAL64	mm	$\pm\infty$	Aktivierungsposition					
			UINT32	ENUM	s. Anhang	Master-Skalierungstyp: 0: user defined (default) 1: scaling with auto offset 2: off					
			UINT32	ENUM	s. Anhang	Slave-Skalierungstyp: 0: user defined (default) 1: scaling with auto offset 2: off					
			Erweiterung für MultiCam:								
			UINT32	ENUM	s. Anhang	Cam Operation Mode					
UINT32	1	[1...255]	Referenz-Tabellen-ID (Bezugstabelle)								
BYTE[104]			Reserve (TC3)								
}											

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000004B(0x00n0004B)	Write	Master/Slave Universelle Fliegende Säge (SERVO)	{			Master/Slave- Synchronisierungskopplung (SERVO):	Geändert ab TC3
			UINT32	ENUM	s. Anhang	Slavetyp/Kopplungstyp (s. Anhang)	
			UINT32	1	[1...255]	Achs-ID der Masterachse	
			UINT32	1	[0..8]	Subindex n der Masterachse (Default-Wert: 0)	
			UINT32	1	[0..8]	Subindex n der Slaveachse (Default-Wert: 0)	
			REAL64	1	$\pm\infty$ (<> 0.0)	Getriebefaktor	
			REAL64	mm	$\pm\infty$	Master-Synchronposition	
			REAL64	mm	$\pm\infty$	Slave-Synchronposition	
			REAL64	mm/s	$\geq 0.0$	Slavegeschwindigkeit (optional)	
			REAL64	mm/s <sup>2</sup>	$\geq 0.0$	Slavebeschleunigung (optional)	
			REAL64	mm/s <sup>2</sup>	$\geq 0.0$	Slaveverzögerung (optional)	
			REAL64	mm/s <sup>3</sup>	$\geq 0.0$	Slaveruck(optional)	
			UINT32	1	$\geq 0$	Bitmaske(Default-Wert: 0)	
			UINT32			Reserve (TC3)	
			}				

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung			
0x0000004D(0x00n0004D)	Write	Master/Slave- Tabellenkopplung  Typ: TABULAR und MF	{			Änderung der Tabellenskalierung (SERVO):	Geändert ab TC3			
			UINT32	ENUM	s. Anhang	Aktivierungstyp der Änderung  0: 'instantaneous' (default)  1: 'at master cam position'  2: 'at master axis position'  3: 'next cycle'				
			UINT32			Reserve (TC3)				
			REAL64	z. B. mm	$\pm\infty$	Aktivierungsposition				
			UINT32	ENUM	s. Anhang	Master-Skalierungstyp  0: user defined (default)  1: scaling with auto offset  2: off				
			UINT32	ENUM	s. Anhang	Slave-Skalierungstyp  0: user defined (default)  1: scaling with auto offset  2: off				
			REAL64	z. B. mm	$\pm\infty$	Master-Positionsoffset				
			REAL64	z. B. mm	$\pm\infty$	Slave-Positionsoffset				
			REAL64	1	$\pm\infty (<> 0.0)$	Master- Positionsskalierung				
			REAL64	1	$\pm\infty$	Slave- Positionsskalierung				
			Optionale Erweiterung für MultiCam:							
			UINT32	1	$\geq 0$	Cam Table ID				
			UINT32			Reserve (TC3)				
			}							
0x00000050	Write	every	VOID			Deaktiviere komplette Achse (Disable)				
0x00000051	Write	every	VOID			Aktiviere komplette Achse (Enable)				

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000052	Write	SERVO	{			Änderung des aktiven Achsregelkreises (Trippel aus Encoder, Regler und Achsinterfaces) mit/ ohne externe Sollwertvorgabe:	Geändert ab TC3
			UINT32	1	≥ 0	Nummer/Index des Achsregelkreises (Default -Wert: 0)	
			UINT32	ENUM	s.Anhang (>0)	Umschalttyp für Auf- <u>synchronisierungsver-</u> <u>halten</u> [▶ 172] 1: 'Standard'	
			REAL64	1	±∞	Synchronisierungswert für Umschaltung (optional)	
			UINT32	0/ 1	0/1	Externe Sollwertvorgabe mittels Achsinterface ? Anmerkung: Wird bisher nicht verwendet!	
			UINT32			Reserve (TC3)	
			}				
0x00000060	Write	every	VOID			Deaktiviere Drive-Output (Disable)	
0x00000061	Write	every	VOID			Aktiviere Drive-Output (Enable)	
0x00000062	Write	Eil/Schleich	UINT16	1	0/1	Feststellbremse lösen? 0: automatische Ansteuerung (Default) 1: zwingend immer gelöst Anmerkung: Wird beim Achsreset auf '0' zurückgesetzt!	
0x00000063	Write	only for SERCOS/SoE and CANopen/ CoE	{			Aktiviere „Drive Operation Mode“ (z. B. Position Velo, Torque, ...)	NEU ab TC 3.1 B4022 (NC 4443) Immer an SAF-Port 501!
			INT32	ENUM [▶ 169] (s. Anhang)	[0; 1, 2, 3, ...] Special cases: ≥ 100: SoE < 0: CoE	Neuer „Drive Operation Mode“ (generische Modi)	
			UINT32	1	0	Reserve	
			UINT32	1	0	Reserve	
			UINT32	1	0	Reserve	
			}				
0x00000070	Write	every	VOID			Rückführung der Achse aus z. B. einer 3D-Gruppe in ihre persönliche PTP-Gruppe	

\* Der folgende Gefahrenhinweis betrifft Index-Offset 0x0000002D:

 **GEFAHR**

**Lebensgefahr oder Gefahr von schweren Verletzungen oder Sachschäden durch unbeabsichtigte Bewegungen der Achse**

Bei Verwendung des Funktionsbausteins wird die Achse in den CST-Mode geschaltet. Nach Verwendung des Funktionsbausteins (insbesondere nach Fehlersituationen), kann es dazu kommen, dass sich die Achse weiterhin im CST-Mode befindet. Dieses kann beim Freigeben der Achse zu plötzlichen und ungeplanten Bewegungen (insbesondere bei Hubachsen) führen.

- Stellen Sie sicher, dass keine Gefahr im Sinne der Risikobewertung entsteht.
- Prüfen Sie die aktuelle Betriebsart über den Funktionsbaustein MC\_ReadDriveOperationMode.
- Wenn sich die Achse nicht in einer positionsbezogenen Betriebsart (CSV/CSP) befindet, überführen Sie diese vor einer Freigabe:
  - *direkt* mit MC\_WriteDriveOperationMode in die gewünschte positionsbezogene Betriebsart (CSV/CSP) oder
  - *indirekt* mit MC\_Halt / MC\_Stop in die gewünschte positionsbezogene Betriebsart (CSV/CSP) (ab TwinCAT 3.1.4024.40)

Weitere Funktionsbausteine, die die Achse indirekt in eine positionsbezogene Betriebsart schalten, können dies nur eingeschränkt und sind daher nicht für einen bewussten Betriebsartwechsel zu verwenden.

- ⇒ Anschließend ist ein erneutes Prüfen nötig, ob sich die Achse auch wirklich in einer positionsbezogenen Betriebsart (CSV/CSP) befindet, falls nicht, ist ein Abbruch mit Fehlerbehandlung erforderlich.



**3.1.1.5.4.4.4**     ***"Index-Offset" Spezifikation für zyklische Achsprozessdaten  
(Index-Group 0x4300 + ID)***

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitions bereich	Beschreibung	Anmerkung
0x00n00000	Read/Write	every (PLC→NC)	{ 128 Byte}		STRUCT s. Achs- interface	ACHS-STRUKTUR (PLC→NC) n = 0: Standardinterface der Achse n > 0: n-tes Interface der Achse (optional)	Write-Befehl nur Optional! Sicherheitsaspekte beachten! <i>PLCTONC_AX/ S_REF</i>
0x00n00001	Read/Write	every (PLC→NC)	UINT32	1	>0	Steuer-Doppelwort	Write-Befehl nur Optional! <i>Symbolischer Zugriff möglich!</i> „ControlDWord“
0x00n00002	Read/Write	every (PLC→NC)	UINT16	1	0/1	Reglerfreigabe	Nicht oszilloskopierbar!
0x00n00003	Read/Write	every (PLC→NC)	UINT16	1	0/1	Vorschubfreigabe Plus	Nicht oszilloskopierbar!
0x00n00004	Read/Write	every (PLC→NC)	UINT16	1	0/1	Vorschubfreigabe Minus	Nicht oszilloskopierbar!
0x00n00007	Read/Write	every (PLC→NC)	UINT16	1	0/1	Referenziernocke	Nicht oszilloskopierbar!
0x00n00021	Read/Write	every (PLC→NC)	UINT32	%	0...1000000	Geschwindigkeitsoverride (1000000 == 100%)	Write-Befehl nur Optional! <i>Symbolischer Zugriff möglich!</i> „OverrideV“
0x00n00022	Read/Write	every (PLC→NC)	UINT32	1	ENUM	Betriebsart Achse	Write-Befehl nur Optional!
0x00n00025	Read/Write	every (PLC→NC)	REAL64	z. B. mm		Istpositionskorrekturwert (Meßsystemfehlerkorrektur)	Write-Befehl nur Optional!
0x00n00026	Read/Write	every (PLC→NC)	REAL64	z. B. mm/s		Externer Regleranteil (Lageregleranteil)	Write-Befehl nur Optional!
0x00n00027	Read/Write	every (PLC→NC)	{ REAL64 REAL64 REAL64 INT32 UINT32 REAL64 }	z. B. mm z. B. mm/s z. B. mm/s <sup>2</sup> 1	±∞ ±∞ ±∞ +1, 0, -1	Externe Sollwertgenerierung Externe Sollposition Externe Sollgeschwindigkeit Externe Sollbeschleunigung Externe Sollfahrrichtung Reserve (TC3) Reserve (TC3)	Write-Befehl nur Optional!  Geändert ab TC3
0x00n00080	Read	every (PLC→NC)	{ 256 Byte}		STRUCT s. Achs- interface	ACHS-STRUKTUR (NC→PLC) Anm.: Größe und Alignment geändert n = 0: Standardinterface der Achse n > 0: n-tes Interface der Achse (optional)	Geändert ab TC3. <i>NCTOPLC_AXIS_REF</i>
0x00n00071	Read	every (PLC→NC)	UINT8	1	>0	Status-Doppelwort: Byte 1	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitions bereich	Beschreibung	Anmerkung
0x00n00072	Read	every (PLC→NC)	UINT8	1	>0	Status-Doppelwort: Byte 2	
0x00n00073	Read	every (PLC→NC)	UINT8	1	>0	Status-Doppelwort: Byte 3	
0x00n00074	Read	every (PLC→NC)	UINT8	1	>0	Status-Doppelwort: Byte 4	
0x00n00081	Read	every (PLC→NC)	UINT32	1	>0	Status-Doppelwort (komplett)	<i>Symbolischer Zugriff möglich!</i> „StateDWord“
0x00n00082	Read	every (PLC→NC)	UINT16	1	0/1	Achse ist betriebsbereit	Nicht oszilloskopierbar!
0x00n00083	Read	every (PLC→NC)	UINT16	1	0/1	Achse ist referenziert	Nicht oszilloskopierbar!
0x00n00084	Read	every (PLC→NC)	UINT16	1	0/1	Achse in geschützter Betriebsart (z. B. Slaveachse)	Nicht oszilloskopierbar!
0x00n00085	Read	every (PLC→NC)	UINT16	1	0/1	Achse in Eilgangsbetriebsart	Nicht oszilloskopierbar!
0x00n00088	Read	every (PLC→NC)	UINT16	1	0/1	Achse hat ungültige IO Daten	Nicht oszilloskopierbar!
0x00n00089	Read	every (PLC→NC)	UINT16	1	0/1	Achse ist im Fehlerzustand	Nicht oszilloskopierbar!
0x00n0008A	Read	every (PLC→NC)	UINT16	1	0/1	Achse fährt größer	Nicht oszilloskopierbar!
0x00n0008B	Read	every (PLC→NC)	UINT16	1	0/1	Achse fährt kleiner	Nicht oszilloskopierbar!
0x00n0008C	Read	every (PLC→NC)	UINT16	1	0/1	Achse ist im logischen Stillstand (es werden nur Sollwerte betrachtet) (Lageregler?)	Nicht oszilloskopierbar!
0x00n0008D	Read	every (PLC→NC)	UINT16	1	0/1	Achse ist am Referenzieren	Nicht oszilloskopierbar!
0x00n0008E	Read	every (PLC→NC)	UINT16	1	0/1	Achse ist im Positionsbereichsfenster	Nicht oszilloskopierbar!
0x00n0008F	Read	every (PLC→NC)	UINT16	1	0/1	Achse ist in Zielposition (Zielposition erreicht)	Nicht oszilloskopierbar!
0x00n00090	Read	every (PLC→NC)	UINT16	1	0/1	Achse hat V-Konst oder Drehzahl	Nicht oszilloskopierbar!
0x00n0009A	Read	every (PLC→NC)	UINT16	1	0/1	Betriebsart nicht ausgeführt (Busy)	Nicht oszilloskopierbar!
0x00n0009B	Read	every (PLC→NC)	UINT16	1	0/1	Achse hat Auftrag / Führt Auftrag aus	Nicht oszilloskopierbar!
0x00n000B1	Read	every (PLC→NC)	UINT32	1	≥0	Fehlercode Achse	
0x00n000B2	Read	every (PLC→NC)	UINT32	1	ENUM	Bewegungszustand der Achse (Masterzustand [►_169] / Slavezustand [►_169])	<i>Symbolischer Zugriff möglich!</i> „AxisState“
0x00n000B3	Read	every (PLC→NC)	UINT32	1	ENUM	Betriebsart der Achse (Rück. NC)	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitions bereich	Beschreibung	Anmerkung
0x00n000B4	Read	every (PLC→NC)	UINT32	1	ENUM	Referenzierstatus der Achse	<i>Symbolischer Zugriff möglich!</i> „HomingState“
0x00n000B5	Read	every (PLC→NC)	UINT32	1	ENUM	Koppelstatus der Achse	<i>Symbolischer Zugriff möglich!</i> „CoupleState“
0x00n000B6	Read	every (PLC→NC)	UINT32	1	≥0	SVB-Einträge/Aufträge der Achse (PRE-Tabelle)	
0x00n000B7	Read	every (PLC→NC)	UINT32	1	≥0	SAF-Einträge/Aufträge der Achse (EXE-Tabelle)	
0x00n000B8	Read	every (PLC→NC)	UINT32	1	≥0	Achs-ID	
0x00n000B9	Read	every (PLC→NC)	UINT32	1	≥0	Betriebsarten Status-Doppelwort: Bit 0: Positionsbereichsüberwachung aktiv? Bit 1: Zielpositionsfensterüberwachung aktiv? Bit 2: Schleifenweg aktiv? Bit 3: Physikalische Bewegungsüberwachung aktiv? Bit 4: PEH-Zeitüberwachung aktiv? Bit 5: Losekompensation aktiv? Bit 6: Verzögerte Fehlerreaktion aktiv? Bit 7: Modulo Betriebsart aktiv (Modulo-Achse)? Bit 16: Schleppabstandüberwachung Pos. aktiv? Bit 17: Schleppabstandüberwachung Gesch. aktiv? Bit 18: Endlagenüberwachung Min. aktiv? Bit 19: Endlagenüberwachung Max. aktiv? Bit 20: Istpositionskorrektur aktiv?	
0x00n000BA	Read	every (PLC→NC)	REAL64	z. B. mm		Istposition (verrechneter Absolutwert)	
0x00n000BB	Read	every (PLC→NC)	REAL64	z. B. mm		Modulo-Istposition	
0x00n000BC	Read	every (PLC→NC)	INT32	1		Modulo-Umdrehungen	
0x00n000BD	Read	every (PLC→NC)	REAL64	z. B. mm/s		Istgeschwindigkeit (optional)	
0x00n000BE	Read	every (PLC→NC)	REAL64	z. B. mm		Schleppabstand Position	
0x00n000BF	Read	every (PLC→NC)	REAL64	z. B. mm		Sollposition	

Index-Offset (Hex)	Zugriff	Achstyp	Datentyp	Phys. Einheit	Definitions bereich	Beschreibung	Anmerkung
0x00n000C0	Read	every (PLC→NC)	REAL64	z. B. mm/s		Sollgeschwindigkeit	
0x00n000C1	Read	every (PLC→NC)	REAL64	z. B. mm/s <sup>2</sup>		Sollbeschleunigung	
0x00n10000	Read/Write	Encoder: every (NC→IO)	{ 40 Byte }		STRUCT s. Encoder-IO-interface	ENCODER-OUTPUT-STRUKTUR (NC→IO, 40 Byte) <i>NCENCODERSTRUCT_OUT2</i>	Write-Befehl nur Optional! Sicherheitsaspekte beachten!
0x00n10080	Read	Encoder: every (IO→NC)	{ 40 Byte }		STRUCT s. Encoder-IO-interface	ENCODER-INPUT-STRUKTUR (IO→NC, 40 Byte) <i>NCENCODERSTRUCT_IN2</i>	
0x00n30000	Read/Write	Drive: every (NC→IO)	{ 40 Byte }		STRUCT s. Drive-IO-interface	DRIVE-OUTPUT-STRUKTUR (NC→IO, 40 Byte) <i>NCDRIVESTRUCT_OUT2</i>	Write-Befehl nur Optional! Sicherheitsaspekte beachten!
0x00n30080	Read	Drive: every (IO→NC)	{ 40 Byte }		STRUCT s. Drive-IO-interface	DRIVE-INPUT-STRUKTUR (NC→IO, 40 Byte) <i>NCDRIVESTRUCT_IN2</i>	

**3.1.1.5.4.5**      **Spezifikation Encoder****3.1.1.5.4.5.1**      ***"Index-Offset" Spezifikation für Encoderparameter (Index-Group 0x5000 + ID)***

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	UINT32	1	[1 ... 255]	Encoder-ID	
0x00000002	Read	every	UINT8[30+1]	1	30 Zeichen	Encodername	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Encodertyp [► 164]	
0x00000004	Read/Write	every	UINT32	1	Byteoffset	Input-Adress-Offset (IO-Input-Image)	Änderung der IO-Adresse
0x00000005	Read/Write	every	UINT32	1	Byteoffset	Output-Adress-Offset (IO-Output-Image)	Änderung der IO-Adresse
0x00000006	Read/Write	every	REAL64	z. B. mm/INC	[1.0E-12 ... 1.0E+30]	Resultierender Skalierungsfaktor (Zähler/Nenner)  Anm.: ab TC3 besteht der Skalierungsfaktor aus zwei Komponenten, Zähler und Nenner (Default: 1.0).	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00000007	Read/Write	every	REAL64	z. B. mm	[±1.0E+9]	Positionsoffset	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00000008	Read/Write	every	UINT16	1	[0,1]	Geberzählrichtung	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00000009	Read/Write	every	REAL64	z. B. mm	[0.001 ... 1.0E+9]	Modulo-Faktor	
0x0000000A	Read/Write	every	UINT32	1	s. ENUM (>0) im Anhang	Encodermodus [► 165]	
0x0000000B	Read/Write	every	UINT16	1	0/1	Softend-Min-Überwachung?	
0x0000000C	Read/Write	every	UINT16	1	0/1	Softend-Max-Überwachung?	
0x0000000D	Read/Write	every	REAL64	mm		Softendlage Min	
0x0000000E	Read/Write	every	REAL64	mm		Softendlage Max	
0x0000000F	Read/Write	every	UINT32	1	s. ENUM (≥0) im Anhang	Encoder-Auswerterichtung [► 165] (Freigabe log. Zählrichtung)	
0x00000010	Read/Write	every	REAL64	s	[0.0...60.0]	Filterzeit für Positionswert in Sekunden (P-T1)	
0x00000011	Read/Write	every	REAL64	s	[0.0...60.0]	Filterzeit für Geschwindigkeitswert in Sekunden (P-T1)	
0x00000012	Read/Write	every	REAL64	s	[0.0...60.0]	Filterzeit für Beschleunigungswert in Sekunden (P-T1)	
0x00000013	Read/Write	every	UINT8[10+1]	1		Physikalische Einheit	Nicht implementiert!
0x00000014	Read/Write	every	UINT32	1		Interpretation der Einheiten (Position, Geschwindigkeit, Zeit)  Bit 0: Geschwindigkeit in x/min statt x/s  Bit 1: Position in tausendstel der Basiseinheit	Nicht implementiert! Bitarray

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbereich	Beschreibung	Anmerkung
0x00000015	Read/Write	every	UINT32	INC	[0x0...0xFFFFFFFF]	Geber-Maske (Maximalwert des Geber-Istwertes in Inkrementen) Anm.: Die Geber-Maske darf ein beliebiger Zahlenwert sein (z. B. 3600000) und muss nicht mehr wie in der Vergangenheit einer durchgehenden Folge von binären Einsen entsprechen ( $2^n-1$ ).	Für den Schreibzugriff darf die Achse nicht freigegeben sein. s.a. Param. "Geber-Sub-Maske"
0x00000016	Read/Write	every	UINT16	1	0/1	Istpositionskorrektur (Meßsystemfehlerkorrektur)?	
0x00000017	Read/Write	every	REAL64	s	[0.0...60.0]	Filterzeit für Istpositionskorrektur in Sekunden (P-T1)	
0x00000018	Read/Write	every	UINT32	1	[0x0...0xFFFFFFFF]	Filtermaske für rohen Inkrementalwert(0x0: voller Durchlass )	
0x00000019	Read/Write	every	UINT32	1	s. ENUM (≥0) im Anhang	<u>Encoder-Bezugsmaßsystem</u> [►_165]	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x0000001A	Read/Write	every	UINT32	1	s. ENUM (≥0)	Encoder-Positionsinitialisierung	Nicht implementiert!
0x0000001B	Read/Write	every	REAL64	z. B. mm	[≥0, Modulo-Faktor/2]	Toleranzfenster für Modulo-Start	
0x0000001C	Read	every	UINT32	1	s. ENUM (≥0)	<u>Encoder-Vorzeichen-Interpretation</u> [►_165] (Datentyp)	
0x0000001D	Read	every	UINT16	1	0/1	Inkremental- oder Absolutencoder ? 0: Inkrementaler Encodertyp 1: Absoluter Encodertyp	
0x00000020	Read/Write	every	UINT32	1	s. ENUM (≥0)	Encoder-Totzeitkompensationsmodus 0: Aus (Default) 1: Ein (mittels Geschwindigkeit) 2: Ein (mittels Geschwindigkeit und Beschleunigung)	
0x00000021	Read/Write	every	UINT32	1		Steuerdoppelwort (32 Bits) für die Encoder-Totzeitkompensation: Bit 0 = 0: relative IO Zeiten (Default) Bit 0 = 1: absolute IO Zeiten	
0x00000022	Read/Write	every	INT32	ns	[±1.0E+9]	Summe der parametrisierten zeitlichen Verschiebung für die Encoder-Totzeitkompensation (typischerweise positive Zahlenwerte)	



Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000023	Read/Write	every	REAL64	z. B. mm/INC	[1.0E-12 ... 1.0E+30]	Komponente des Skalierungsfaktors: Zähler  (=> Skalierungsfaktor Zähler / Skalierungsfaktor Nenner)	NEU ab TC3 Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00000024	Read/Write	every	REAL64	1	[1.0E-12 ... 1.0E+30]	Komponente des Skalierungsfaktors: Nenner  (=> Skalierungsfaktor Zähler / Skalierungsfaktor Nenner)  Default: 1.0	NEU ab TC3 Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x00000025	Read/Write	every	{ REAL64 REAL64 } 16 bytes	z. B. mm/INC 1	[1.0E-12 ... 1.0E+30] [1.0E-12 ... 1.0E+30]	Komponente des Skalierungsfaktors: Zähler  Komponente des Skalierungsfaktors: Nenner  (=> Skalierungsfaktor Zähler / Skalierungsfaktor Nenner)	NEU ab TC3
0x00000030	Read/Write	every	UINT32	1		Internes Encoder Control Doppelwort zur Festlegung der Betriebsarten und Eigenschaften	NEU ab TC3
0x00000101	Read/Write	INC	UINT16	1	[0,1]	Suchrichtung für Ref.nocken invers?	
0x00000102	Read/Write	INC		1	[0,1]	Suchrichtung für Syncimpuls invers?	
0x00000103	Read/Write	INC	REAL64	z. B. mm	[±1.0E+9]	Referenzposition	
0x00000104	Read/Write	INC	UINT16	1	[0,1]	Abstandsüberwachung zwischen Ref.nocken und Syncimpuls aktiv?	Nicht implementiert!
0x00000105	Read/Write	INC	UINT32	INC	[0 ...65536]	Mindestabstand Ref.nocken zum Syncimpuls in Inkrementen	Nicht implementiert!
0x00000106	Read/Write	INC	UINT16	1	[0,1]	Externer Syncimpuls?	
0x00000107	Read/Write	INC	UINT32	1	s. ENUM (>0)	<u>Referenziermodus</u> <u>(Sync Condition)</u> <a href="#"> &gt;_166</a>	

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000108	Read/Write	INC	UINT32	1	[0x0000000F... 0xFFFFFFFF]Bi nämaske: (2 <sup>n</sup> - 1)	Geber-Sub-Maske (Maximalwert des Absolutbereichs des Geber-Istwertes in Inkrementen)  Wird z. B. verwendet als Referenzmarke für den Referenzier-Mode "Software Sync" und für die NC Retain Daten("ABSOLUTE (MODULO)", "INCREMENTAL (SINGLETURN ABSOLUTE)" ).  Anm.1: Die Geber- Sub-Maske muss kleiner gleich der Geber-Maske sein.  Anm.2: Die Geber- Maske muss ein ganzzahliges Vielfaches der Geber- Sub-Maske sein.  Anm.3: Die Geber- Sub-Maske muss einer durchgehenden Folge von binären Einsen entsprechen (2 <sup>n</sup> -1), z. B. 0x000FFFFF.	NEU s.a. Param. "Geber-Maske"
0x00000109	Read/Write	INC	UINT32	1	s. ENUM (≥0)	Homing Sensor Source [►_166] Legt die Quelle des Digitaleingangs der Referenziernocke fest.	
0x00000110	Read/Write	INC (Encodersimula tion)	REAL64	1	[0.0 ... 1000000.0]	Skalierung/ Gewichtung des Rauschanteils für Simulationsencoder	

**3.1.1.5.4.5.2**     *"Index-Offset" Spezifikation für Encoderzustand (Index-Group  
0x5100 + ID)*

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsreich	Beschreibung	Anmerkung
0x00000001	Read	every	INT32			Fehlerstatus Encoder	
0x00000002	Read	every	REAL64			Istposition (verrechnet mit Istpositionskorrekturwert)	Symbolischer Zugriff möglich! 'ActPos'
0x00000003	Read	every	REAL64			Modulo-Istposition	Symbolischer Zugriff möglich! 'ActPosModulo'
0x00000004	Read	every	INT32			Modulo-Istumdrehung	
0x00000005	Read	every	REAL64			Optional: Istgeschwindigkeit	Base Unit / s Symbolischer Zugriff möglich! 'ActVelo'
0x00000006	Read	every	REAL64			Optional: Istbeschleunigung	Base Unit / s <sup>2</sup> Symbolischer Zugriff möglich! 'ActAcc'
0x00000007	Read	every	INT32			Geber-Istinkremente	
0x00000008	Read	every	INT64			Software-Istinkrementalzähler	
0x00000009	Read/Write	every	UINT16			Referenzierflag ("Eichflag")	
0x0000000A	Read	every	REAL64			Istpositionskorrekturwert (Messsystemfehlerkorrektur)	
0x0000000B	Read	every	REAL64			Istposition ohne Istpositionskorrekturwert	
0x0000000C	Read	every	REAL64	z. B. mm		Istpositionskorrekturwert aufgrund der Totzeitkompensation	
0x0000000D	Read	every	REAL64	s		Summe der zeitlichen Verschiebung für Encoder-Totzeitkompensation (parametrierte und variable Totzeit)Anm.: Eine Totzeit wird im System als positiver Wert angegeben.	
0x0000000E	Read	every	REAL64	z. B. mm		Interner Positionsoffset als Korrekturwert für eine Wertereduktion auf die Grundperiode (Modulobereich)	
0x00000010	Read	every	REAL64	z. B. mm/s		Istgeschwindigkeit ohne Istpositionskorrekturwert	
0x00000012	Read	every	REAL64	z. B. mm		Ungefilterte Istposition (verrechnet mit Istpositionskorrekturwert)	
0x00000013	Read	every	REAL64	z. B. mm		Gefilterte Istposition (verrechnet mit Istpositionskorrekturwert, ohne Totzeitkompensation)	
0x00000014	Read	Type: SoE, CoE, MDP 742	REAL64	z. B. mm/s		Optional: Antriebs-Istgeschwindigkeit (direkt vom SoE, CoE oder MDP 742 Drive übertragen)	Base Unit / s NEU ab TC3.1 B4020.30
0x00000015	Read	every	REAL64	z. B. mm/s		Optional: Ungefilterte Istgeschwindigkeit	Base Unit / s

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000016	Read	every	<b>READ</b> (16 byte * N)			Lesen des Istpositions-Puffer	
			{				
			UINT32	ns	≥0	Zeitstempel (DcTimeStamp mit 32 Bit)	
			UINT32			Reserve	
			REAL64	z. B. mm	±∞	Istposition zum zugehörigen Zeitstempel	
			} [N]				
0x00000017	Read		REAL64	z. B. mm		Auslesen des MC_SetPosition-Offsets	
0x00000101	Read	INC	REAL64	z. B. mm		Rücklesen der Positionsdifferenz zwischen Aktivieren und Gültig werden des Hardwarelatches	Nicht oszilloskopierbar!
0x00000200	Read Write	<b>Function group "TouchProbeV2": - SERCOS/ SoE</b> - EtherCAT/ CoE (CANopen DS402) - SoftDrive (TCom), - MDP 511 (EL5101, EL5151, EL5021, EL7041, EL7342)	<b>WRITE</b> (24 byte)			Read "Touch Probe" state (state of external latch)	Only for SAF-port 501
			{				
			UINT32	1	[1,2,3,4]	Probe unit (probe 1, 2, 3, 4)	
			UINT32[5]			Reserved	
			}				
			<b>READ</b> (64 byte)				
			{				
			UINT32	1	[0/1]	Touch probe rising edge active?	
			UINT32	1	[0/1]	Touch probe rising edge became valid?	
			REAL64	e.g. mm		Touch probe rising edge position value	
			UINT32	1	≥0	Touch probe rising edge counter (continuous mode)	
			UINT32			Reserved	
			UINT32	1	[0/1]	Touch probe falling edge active?	
			UINT32	1	[0/1]	Touch probe falling edge became valid?	
			REAL64	e.g. mm		Touch probe falling edge position value	
UINT32	1	≥0	Touch probe falling edge counter (continuous mode)				
UINT32[5]			Reserved				
}							
0x00000201	Read	KL5101, SERCOS, AX2xxx, ProviDrive	UINT16	1	[0, 1]	"Externe Latchfunktion" aktiv? bzw. "Messtasterfunktion" aktiv? ( <i>flankenunabhängig</i> )	Nicht oszilloskopierbar!
0x00000201	Read	CANopen	UINT32[4]	1	[0, 1]	"Externe Latchfunktionen 1 bis 4" aktiv? bzw. "Messtasterfunktionen 1 bis 4" aktiv?	Nicht oszilloskopierbar!

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000202	Read	KL5101, SERCOS, AX2xxx, ProviDrive	UINT16	1	[0,1]	Externer Latchwert gültig geworden?  bzw. Messtaster gelatcht? ( <i>flankenunabhängig</i> )	s. a. Achsisinterface NcToPlc (Statusdoppelw ort)
0x00000202	Read	CANopen	UINT32[4]	1	[0,1]	Externe Latchwerte 1 bis 4 gültig geworden?  bzw. Messtaster 1 bis 4 gelatcht?	s. a. Achsisinterface NcToPlc (Statusdoppelw ort)
0x00000203	Read	KL5101, SERCOS, AX2xxx, ProviDrive	UINT32	INC		Externer / Messtaster Hardware- Inkremental-Latchwert	
0x00000204	Read	KL5101, SERCOS, AX2xxx, ProviDrive	UINT64	INC		Externer / Messtaster Software-Inkremental- Latchwert	
0x00000205	Read	KL5101, SERCOS, AX2xxx, ProviDrive	REAL64	z. B. mm		Externer / Messtaster Positions latchwert	Base Unit
0x00000205	Read	CANopen	REAL64[4]	z. B. mm		Externe Messtasterwerte / Positions latchwerte	Base Unit
0x00000206	Read	KL5101, SERCOS, AX2xxx, ProviDrive	UINT32	INC		Differenz Hardware- Inkremental- Latchwerte (NewLatch - LastLatch)	Nicht oszilloskopierba r!
0x00000207	Read	KL5101, SERCOS, AX2xxx, ProviDrive	UINT64	INC		Differenz Software- Inkremental- Latchwerte (NewLatch - LastLatch)	Nicht oszilloskopierba r!
0x00000208	Read	KL5101, SERCOS, AX2xxx, ProviDrive	REAL64	z. B. mm		Differenz Positions latchwerte (NewLatch - LastLatch)	Nicht oszilloskopierba r! Base Unit
0x00000210	Read	KL5101, AX2xxx, ProviDrive	UINT16	1	[0,1]	"Externe Latchfunktion" für <i>steigende Flanke</i> aktiv?  bzw. "Messtasterfunktion" für <i>steigende Flanke</i> aktiv?	Nicht oszilloskopierba r!
0x00000210	Read	CANopen	UINT16[4]	1	[0,1]	"Externe Latchfunktion" für <i>steigende Flanke</i> aktiv?  bzw. "Messtasterfunktion" für <i>steigende Flanke</i> aktiv?	Nicht oszilloskopierba r!
0x00000211	Read	KL5101, AX2xxx, ProviDrive	UINT16	1	[0,1]	"Externe Latchfunktion" für <i>fallende Flanke</i> aktiv?  bzw. "Messtasterfunktion" für <i>fallende Flanke</i> aktiv?	Nicht oszilloskopierba r!

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000211	Read	<b>CANopen</b>	UINT16[4]	1	[0,1]	"Externe Latch Funktion" für <i>fallende Flanke</i> aktiv?  bzw. "Messtasterfunktion" für <i>fallende Flanke</i> aktiv?	Nicht oszilloskopierbar!
0x00000212	Read	<b>CANopen</b>	UINT16	1	[0,1]	Status vom "Touch Probe 1" Eingangssignal	Nicht oszilloskopierbar! Ab TC3.1 B4024.11
0x00000213	Read	<b>CANopen</b>	UINT16	1	[0,1]	Status vom "Touch Probe 2" Eingangssignal	Nicht oszilloskopierbar! Ab TC3.1 B4024.11

**3.1.1.5.4.5.3**     *"Index-Offset" Spezifikation für Encoderfunktionen (Index-Group 0x5200 + ID)*



Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000001A	Write	every	{			Setze Istposition Encoder/Achse	Base Unit
			UINT32	ENUM	s. Anhang	Istpositionstyp [►_161] (s. Anhang)	
			REAL64	mm	±∞	Istposition für Encoder/Achse <b>Vorsicht bei Benutzung!</b>	
			}				
0x0000001B	Write	every	VOID			Reinitialisierung der Encoder-Istposition Anm.: Wirkung bei Referenz System „ABSOLUTE MULTITURN RANGE (with single overflow)“ und „ABSOLUTE SINGLETURN RANGE (with single overflow)“.	NEU ab TC3
0x00000200	Write	Function group "TouchProbeV2": - SERCOS/ SoE, - EtherCAT/ CoE (CANopen DS402) - SoftDrive (TCom), - MDP 511 (EL5101, EL5151, EL5021, EL7041, EL7342)	{			Activate "Touch Probe" (external latch)	Only for SAF-port 501
			UINT32	1	[1,2,3,4]	Probe unit (probe 1, 2, 3, 4)	
			UINT32	1	[0,1]	Signal edge (0=rising edge, 1=falling edge)	
			UINT32	1	[1,2]	Probe mode (1=single, 2=continuous, ...)	
			UINT32	1	[1,2,3,4; 128,129]	Signal source (1=input 1, 2=input 2, ...)	
			UINT32			Reserved	
			UINT32			Reserved	
}			} 24 bytes				
0x00000201	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	VOID			Aktiviere "Externes Latch"  bzw. Aktiviere "Messtasterfunktion"(t ypischerweise steigende Flanke)	
0x00000201	Write	CANopen	UINT32[4]			Aktiviere "Externes Latch" 1 bis 4  bzw. Aktiviere "Messtasterfunktion" 1 bis 4(typischerweise steigende Flanke)	
0x00000202	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	VOID			Aktiviere "Externes Latch"  bzw. Aktiviere "Messtasterfunktion"(f allende Flanke)	
0x00000202	Write	CANopen	UINT32[4]			Aktiviere "Externes Latch" 1 bis 4  bzw. Aktiviere "Messtasterfunktion" 1 bis 4(fallende Flanke)	

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000205	Write	Function group "TouchProbeV2 ": - SERCOS/ SoE, - EtherCAT/ CoE (CANopen DS402) - SoftDrive (TCom), - MDP 511 (EL5101, EL5151, EL5021, EL7041, EL7342)	{			Deactivate "Touch Probe" (external latch)	Only for SAF- port 501
			UINT32	1	[1,2,3,4]	Probe unit (probe 1, 2, 3, 4)	
			UINT32	1	[0,1]	Signal edge (0=rising edge, 1=falling edge)	
			UINT32			Reserved	
			UINT32			Reserved	
			UINT32			Reserved	
			UINT32			Reserved	
		} 24 byte					
0x00000205	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	VOID			Deaktiviere "Externes Latch" bzw. Deaktiviere "Messtasterfunktion"	
0x00000205	Write	CANopen	UINT32[4]			Deaktiviere "Externes Latch" bzw. Deaktiviere "Messtasterfunktion"	
0x00000210	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	REAL64	z. B. mm	$\pm\infty$	Setze "Externes Latchereignis" und "Externe Latchposition"	Nur für Simulation!

**3.1.1.5.4.5.4**     ***"Index-Offset" Spezifikation für zyklische Encoderprozessdaten  
(Index-Group 0x5300 + ID)***

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung			
0x00000000	Read/Write	every (NC→IO)	{		STRUCT s. Encoder- Interface	ENCODER-OUTPUT- STRUKTUR (NC→IO, 40 Byte) <i>NCENCODERSTRUC T_OUT2</i>	Write-Befehl nur optional! Sicherheitsaspe- kte beachten!			
			INT32	INC	≥ 0	nDataOut1				
			INT32	INC	≥ 0	nDataOut2				
			UINT8	1	≥ 0	nCtrl1				
			UINT8	1	≥ 0	nCtrl2				
			UINT8	1	≥ 0	nCtrl3				
			UINT8	1	≥ 0	nCtrl4				
			INT32	INC	≥ 0	nDataOut3				
			INT32	INC	≥ 0	nDataOut4				
			INT32	INC	≥ 0	nDataOut5				
			INT32	INC	≥ 0	nDataOut6				
			UINT8	1	≥ 0	nCtrl5				
			UINT8	1	≥ 0	nCtrl6				
			UINT8	1	≥ 0	nCtrl7				
			UINT8	1	≥ 0	nCtrl8				
			INT32		≥ 0	Reserviert				
			INT32		≥ 0	Reserviert				
						} 40 Byte				
0x00000000	Read/Write	every (NC→IO), Optionales 64 Bit Encoder- Interface  (z. B. MDP513 mit 64Bit)	{		STRUCT s. Encoder- Interface	Optionale ENCODER- OUTPUT-STRUKTUR (NC→IO, 80 Byte) <i>NCENCODERSTRUC T_OUT3</i>	Write-Befehl nur optional! Sicherheitsaspe- kte beachten! NEU ab TC3			
			UINT64	INC	≥ 0	nDataOut1				
			UINT64	INC	≥ 0	nDataOut2				
			UINT64	INC	≥ 0	nDataOut3				
			UINT64	INC	≥ 0	nDataOut4				
			UINT64	INC	≥ 0	nDataOut5				
			UINT64	INC	≥ 0	nDataOut6				
			UINT64	INC	≥ 0	nDataOut7				
			UINT64	INC	≥ 0	nDataOut8				
			UINT16	1	≥ 0	nCtrl1				
			UINT16	1	≥ 0	nCtrl2				
			UINT16	1	≥ 0	nCtrl3				
			UINT16	1	≥ 0	nCtrl4				
			UINT16	1	≥ 0	nCtrl5				
			UINT16	1	≥ 0	nComCtrl				
			INT32	1	≥ 0	reserviert				
						} 80 Byte				

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung	
0x00000001	Write	every (NC→IO)	{		STRUCT s. Encoder- Interface	Bitweiser Zugriff auf ENCODER-OUTPUT- STRUKTUR (NC→IO, 40 Byte) <i>NCENCODERSTRUC T_OUT2</i>	Write-Befehl nur optional! Sicherheitsaspe- kte beachten!	
			UINT32	1	[0 ... 39]	ByteOffset  Relative address offset [0..39] in output structure.  E.G.: To write "nControl1" the ByteOffset must be 8.		
			UINT32	1	[0x00000000... 0xFFFFFFFF]	BitSelectMask (BSM)  The mask defines write enabled bits in a DWORD. Zero bits are protected and remain unaffected.		
			UINT32	1	[0x00000000... 0xFFFFFFFF]	Value  Only those bits in value are overwritten where BSM equals 1.		
			}					
0x00000080	Read	every (IO→NC)	{		STRUCT s. Encoder- Interface	ENCODER-INPUT- STRUKTUR (IO→NC, 40 Byte) <i>NCENCODERSTRUC T_IN2</i>		
			INT32	INC	≥ 0	nDataIn1		
			INT32	INC	≥ 0	nDataIn2		
			UINT8	1	≥ 0	nState1		
			UINT8	1	≥ 0	nState2		
			UINT8	1	≥ 0	nState3		
			UINT8	1	≥ 0	nState4 (Bit0: <i>WcState</i> , Bit1: <i>InputToggle</i> )		
			INT32	INC	≥ 0	nDataIn3		
			INT32	INC	≥ 0	nDataIn4		
			INT32	INC	≥ 0	nDataIn5		
			INT32	INC	≥ 0	nDataIn6		
			UINT8	1	≥ 0	nState5		
			UINT8	1	≥ 0	nState6		
			UINT8	1	≥ 0	nState7		
			UINT8	1	≥ 0	nState8		
			INT32	[ns]	≥ 0	nDcInputTime (absoluter/relativer <i>DcInputShift</i> für Totzeitkompensation)		
			INT32		≥ 0	Reserviert		
			}			40 Byte		

Index-Offset (Hex)	Zugriff	Gruppentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000080	Read	every (NC→IO), Optionales 64 Bit Encoder- Interface  (z. B. MDP513 mit 64Bit)	{		STRUCT s. Encoder- Interface	Optionale ENCODER- INPUT-STRUKTUR (IO→NC, 80 Byte) NCENCODERSTRUC T_IN3	NEU ab TC3
			UINT64	INC	≥ 0	nDataIn1	
			UINT64	INC	≥ 0	nDataIn2	
			UINT64	INC	≥ 0	nDataIn3	
			UINT64	INC	≥ 0	nDataIn4	
			UINT64	INC	≥ 0	nDataIn5	
			UINT64	INC	≥ 0	nDataIn6	
			UINT64	INC	≥ 0	nDataIn7	
			UINT64	INC	≥ 0	nDataIn8	
			UINT16	1	≥ 0	nState1	
			UINT16	1	≥ 0	nState2	
			UINT16	1	≥ 0	nState3	
			UINT16	1	≥ 0	nState4	
			UINT16	1	≥ 0	nState5	
			UINT16	1	≥ 0	nComState (Bit0: <i>WcState</i> , Bit1: <i>InputToggle</i> )	
			INT32	[ns]	≥ 0	nDcInputTime (absoluter/relativer <i>DcInputShift</i> für Totzeitkompensation)	
			} 80 Byte				

**3.1.1.5.4.6**      **Spezifikation Regler****3.1.1.5.4.6.1**      ***"Index-Offset" Spezifikation für Reglerparameter (Index-Group  
0x6000 + ID)***

Index-Offset (Hex)	Zugriff	Reglertyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	UINT32	1	[1 ... 255]	Regler-ID	
0x00000002	Read	every	UINT8[30+1]	1	30 Zeichen	Reglername	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Reglertyp <a href="#">▶_163</a>	
0x0000000A	Read/Write	every	UINT32	1	s. ENUM (>0)	Reglermodus	Default: 1=STANDARD
0x0000000B	Read/Write	every	REAL64	%	[0.0 ... 1.0]	Gewichtung der Geschwindigkeitsvorsteuerung (Standardwert: 1.0 == 100%)	
0x00000010	Read/Write	every	UINT16	1	0/1	Schleppabstandüberw. . Pos.?	
0x00000011	Read/Write	every	UINT16	1	0/1	Schleppabstandüberw. . Geschw.?	
0x00000012	Read/Write	every	REAL64	mm	[0.0...1.0E.6]	Max. Schleppabstand Position	
0x00000013	Read/Write	every	REAL64	s	[0.0...600]	Max. Schleppfilterzeit Position	
0x00000014	Read/Write	every	REAL64	mm/s	[0.0...1.0E.6]	Max. Schleppabstand Geschw.	
0x00000015	Read/Write	every	REAL64	s	[0.0...600]	Max. Schleppfilterzeit Geschw.	
0x00000021	Read/Write	every	REAL64	1	[0.0...1000000.0]	Skalierungsfaktor (Multiplikator) für Differenz der Schleppabstände zwischen Master und Slaveachse  (Umrechnung in dasselbe Koordinatensystem des Masters)	Reservierte Funktion, kein Standard!
0x00000100	Read/Write	P/PID (Pos., (Geschw.))	REAL64	1	[0.0...1.0]	Maximale Ausgabebegrenzung ( $\pm$ ) für Regler-Gesamtausgabe	(Standardwert: 0.5 == 50%)
0x00000102	Read/Write	P/PID (Pos.)	REAL64	(mm/s) / mm	[0.0...1000.0]	Proportionalverstärkung $k_p$ bzw. $k_v$	Base Unit / s / Base Unit Positionsregelung
0x00000103	Read/Write	PID (Pos.)	REAL64	s	[0.0 ... 60.0]	Nachstellzeit $T_n$	Positionsregelung
0x00000104	Read/Write	PID (Pos.)	REAL64	s	[0.0 ... 60.0]	Vorhaltzeit $T_v$	Positionsregelung
0x00000105	Read/Write	PID (Pos.)	REAL64	s	[0.0 ... 60.0]	Verzögerungszeit $T_d$	Positionsregelung
0x00000106	Read/Write	PP (Pos.)	REAL64	(mm/s) / mm	[0.0...1000.0]	Zusätzlicher Proportionalverstärkung $k_p$ bzw. $k_v$ , die oberhalb einer Grenzggeschwindigkeit in Prozent gilt.	Base Unit / s / Base Unit Positionsregelung
0x00000107	Read/Write	PP (Pos.)	REAL64	%	[0.0...1.0]	Schwellgeschwindigkeit in Prozent, oberhalb derer die zusätzliche Proportionalverstärkung $k_p$ bzw. $k_v$ gilt	(Standardwert: 0.01 == 1%)
0x00000108	Read/Write	P/PID (Acc.)	REAL64	s	[0.0 ... 100.0]	Proportionalverstärkung $k_a$	Beschleunigung s- vorsteuerung
0x0000010A	Read/Write	every	UINT32	1	ENUM	Filter für Maximalsteigung der Sollgeschwindigkeit (beschleunigungsbegrenzt): 0: Off, 1: Velo, 2: Pos+Velo	Reservierte Funktion, kein Standard!



Index-Offset (Hex)	Zugriff	Reglertyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000010B	Read/Write	every	REAL64	mm/s <sup>2</sup>		Filterwert für die Maximalsteigung der Sollgeschwindigkeit (Maximalbeschleunigung)	Reservierte Funktion, kein Standard!
0x0000010D	Read/Write	P/PID	REAL64	mm	[0.0 ... 10000.0]	Totzone ("dead band") für Positionsfehler (Regelabweichung) (für P/PID-Regler mit Geschwindigkeits- oder Momenten-Interface)	Reservierte Funktion
0x0000010F	Read/Write	P/PP/PID (Pos.) Slave-Regelung	REAL64	(mm/s) / mm	[0.0...1000.0]	Slave-Koppeldifferenzregelung: Proportionalverstärkung $k_{cp}$	Slave-Koppeldifferenzregelung
0x00000110	Read/Write	P (Pos.)	UINT16	1	0/1	Automatischer Offsetabgleich: aktiv/passiv	
0x00000111	Read/Write	P (Pos.)	UINT16	1	0/1	Automatischer Offsetabgleich: Halte-Modus	
0x00000112	Read/Write	P (Pos.)	UINT16	1	0/1	Automatischer Offsetabgleich: Fading-Modus	
0x00000114	Read/Write	P (Pos.)	REAL64	%	[0.0 ... 1.0]	Automatischer Offsetabgleich: Vorsteuer-Grenze	(Standardwert: 0.05 == 5%)
0x00000115	Read/Write	P (Pos.)	REAL64	s	[0.1 ... 60.0]	Automatischer Offsetabgleich: Zeitkonstante	
0x00000116	Read/Write	PID (Pos.)	REAL64	%	[0.0...1.0]	Maximale Ausgabebeschränkung ( $\pm$ ) für I-Anteil in Prozent (Default-Einstellung: 0.1 == 10%)	
0x00000117	Read/Write	PID (Pos.)	REAL64	%	[0.0...1.0]	Maximale Ausgabebeschränkung ( $\pm$ ) für D-Anteil in Prozent (Default-Einstellung: 0.1 == 10%)	
0x00000118	Read/Write	PID (Pos.)	UINT16	1	0/1	Abschalten des I-Anteils während eines aktiven Positioniervorganges (sofern I-Anteil aktiv)? (Default-Einstellung: 0 = FALSE)	
0x00000120	Read/Write	PID (Pos.)	REAL64	s	$\geq 0$	PT-1-Filterwert für Positionsfehler (Pos.-Regeldifferenz)	Reservierte Funktion, kein Standard!
0x00000202	Read/Write	P/PID (Geschw.)	REAL64	1	[0.0...1000.0]	Proportionalverstärkung $k_p$ bzw. $k_v$	Geschwindigkeitsregelung
0x00000203	Read/Write	PID (Geschw.)	REAL64	s	[0.0 ... 60.0]	Nachstellzeit $T_n$	Geschwindigkeitsregelung
0x00000204	Read/Write	PID (Geschw.)	REAL64	s	[0.0 ... 60.0]	Vorhaltzeit $T_v$	Geschwindigkeitsregelung
0x00000205	Read/Write	PID (Geschw.)	REAL64	s	[0.0 ... 60.0]	Verzögerungszeit $T_d$	Geschwindigkeitsregelung

Index-Offset (Hex)	Zugriff	Reglertyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000206	Read/Write	PID (Geschw.)	REAL64	%	[0.0...1.0]	Maximale Ausgabebeschränkung ( $\pm$ ) für I-Anteil in Prozent (Default-Einstellung: 0.1 == 10%)	Geschwindigkeits- regelung
0x00000207	Read/Write	PID (Geschw.)	REAL64	%	[0.0...1.0]	Maximale Ausgabebeschränkung ( $\pm$ ) für D-Anteil in Prozent (Default-Einstellung: 0.1 == 10%)	Geschwindigkeits- regelung
0x0000020D	Read/Write	P/PID (Geschw.)	REAL64	mm/s	[0.0 ... 10000.0]	Totzone ("dead band") für Geschwindigkeitsfehler (Regelabweichung) für P/PID-Regler mit Geschwindigkeits- oder Momenten-Interface)	Reservierte Funktion
0x00000220	Read/Write	P/PID (Geschw.)	REAL64	s	$\geq 0$	PT-2-Filterwert für Geschwindigkeitsfehler (Geschw.-Regeldifferenz)	Geschwindigkeits- regelung, kein Standard!
0x00000221	Read/Write	P/PID (Geschw.)	REAL64	s	$\geq 0$	PT-1-Filterwert für Geschwindigkeitsfehler (Geschw.-Regeldifferenz)	Reservierte Funktion, kein Standard!
0x00000250	Read/Write	P/PI (Beobachter)	UINT32	1	s. ENUM ( $\geq 0$ )	Beobachtermodus [►_163] für Regelung im Momenten-Interface 0: OFF (default) 1: LUENBERGER	
0x00000251	Read/Write	P/PI (Beobachter)	REAL64	Nm / A	$> 0.0$	Motor: Drehmomentkonstante $K_T$	
0x00000252	Read/Write	P/PI (Beobachter)	REAL64	kg m <sup>2</sup>	$> 0.0$	Motor: Trägheitsmoment $J_M$	
0x00000253	Read/Write	P/PI (Beobachter)	REAL64	Hz	[100.0 ... 2000.0] Default: 500	Bandbreite $f_0$	
0x00000254	Read/Write	P/PI (Beobachter)	REAL64	1	[0.0 ... 2.0] Default: 1.0	Korrekturfaktor $k_c$	
0x00000255	Read/Write	P/PI (Beobachter)	REAL64	s	[0.0 ... 0.01] Default: 0.001	Geschwindigkeitsfilter (1. Ordnung): Zeitkonstante T	
0x00000A03	Read/Write	PID (MW)	REAL64	cm <sup>2</sup>	[0.0 ...1000000]	Zylinderfläche $A_A$ der A-Seite in cm <sup>2</sup>	
0x00000A04	Read/Write	PID (MW)	REAL64	cm <sup>2</sup>	[0.0 ...1000000]	Zylinderfläche $A_B$ der B-Seite in cm <sup>2</sup>	
0x00000A05	Read/Write	PID (MW)	REAL64	cm <sup>3</sup> /s	[0.0 ...1000000]	Nennvolumenstrom $Q_{\text{nenn}}$ in cm <sup>3</sup> /s	
0x00000A06	Read/Write	PID (MW)	REAL64	bar	[0.0 ...1000000]	Nenndruck bzw. Ventildruckabfall $P_{\text{nenn}}$ in bar	
0x00000A07	Read/Write	PID (MW)	UINT32	1	[1 ... 255]	Achs-ID für den Systemdruck $P_o$	

**3.1.1.5.4.6.2**     ***"Index-Offset" Spezifikation für Reglerzustand (Index-Group 0x6100 + ID)***

Index-Offset (Hex)	Zugriff	Reglertyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	INT32			Fehlerstatus Regler	
0x00000002	Read	every	REAL64	z. B. mm/s		Reglerausgabe in absoluten Einheiten	Base Unit / s <i>Symbolischer Zugriff möglich!</i> <i>„CtrlOutput“</i>
0x00000003	Read	every	REAL64	%		Reglerausgabe in Prozent	Nicht oszilloskopierbar!
0x00000004	Read	every	REAL64	V		Reglerausgabe in Volt	Nicht oszilloskopierbar!
0x0000000D	Read	every	REAL64	mm		Schleppabstand Position(ohne Totzeitkompensation)	Base Unit
0x0000000E	Read	every	REAL64	mm		Schleppabstand Position (ohne Sollpositionskorrektur)	Base Unit
0x0000000F	Read	every	REAL64	mm		Schleppabstand Position (mit Sollpositionskorrektur und mit Totzeitkompensation)	Base Unit <i>Symbolischer Zugriff möglich!</i> <i>„PosDiff“</i>
0x00000010	Read	every	REAL64	mm		PeakHold-Wert für maximalen negativen Schleppabstand der Position	Base Unit
0x00000011	Read	every	REAL64	mm		PeakHold-Wert für minimalen positiven Schleppabstand der Position	Base Unit
0x00000012	Read	every	REAL64	mm/s		Schleppabstand Geschwindigkeit	Base Unit / s
0x00000021	Read	every	REAL64	mm		Differenz (Abweichung) der Positions-Schleppabstände zwischen Master- und Slaveachse (Masterminus Slaveschleppabstand)	Base Unit <i>Symbolischer Zugriff über Achse möglich!</i> <i>„PosDiffCouple“</i>
0x00000022	Read	every	REAL64	mm		PeakHold-Wert für maximale negative Differenz der Schleppabstände (Position) zwischen Master- und Slaveachse	Base Unit
0x00000023	Read	every	REAL64	mm		PeakHold-Wert für maximale positive Differenz der Schleppabstände (Position) zwischen Master- und Slaveachse	Base Unit
0x00000101	Read	P/PID (Pos.)	REAL64	z. B. mm/s		P-Anteil des Reglers in absoluten Einheiten	
0x00000102	Read	PID (Pos.)	REAL64	z. B. mm/s		I-Anteil des Reglers in absoluten Einheiten	
0x00000103	Read	PID (Pos.)	REAL64	z. B. mm/s		D-Anteil des Reglers in absoluten Einheiten	
0x00000104	Read	PID (Pos.)	UINT16	1	0/1	Begrenzung des I-Anteils aktiv?	
0x00000105	Read	PID (Pos.)	UINT16	1	0/1	Begrenzung des D-Anteils aktiv?	
0x00000106	Read	PID (Pos.)	UINT16	1	0/1	ARW-Massnahmen des I-Anteils aktiv?	ARW: Anti Reset Windup

Index-Offset (Hex)	Zugriff	Reglertyp	Datentyp	Phys. Einheit	Definitionsbereich	Beschreibung	Anmerkung
0x0000010F	Read	P/PP/PID (Geschw.)	REAL64	z. B. mm/s		Anteil des automatischen Offsetabgleichs in absoluten Einheiten	NEU
0x00000110	Read	PID (Pos.)	REAL64	z. B. mm/s		Beschleunigungsvorsteuerung $Y_{acc}$ des Reglers in absoluten Einheiten Anmerkung: Funktion abhaengig vom Reglertyp!	Beschleunigungsvorsteuerung
0x00000111	Read	PP (Pos.)	REAL64	mm/s/mm	$\geq 0$	Interne interpolierte Proportionalverstärkung $k_p$ bzw. $k_v$	PP-Regler
0x0000011A 0x0000011B 0x0000011C 0x0000011D 0x0000011E 0x0000011F 0x00000120 0x00000121 0x00000122 0x00000123 0x00000124	Read	P (Pos.)	UINT32 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64	1 mm mm/s mm/s mm/s <sup>2</sup> mm mm mm/s mm/s <sup>2</sup> mm/s mm/s <sup>2</sup>		Sollgeschwindigkeitsfilter: InternalPhase InternalPosSollError! TestVeloSoll InternalLimitedVeloSoll InternalAccSollRel InternalPosSollRel PosSollCorrected! VeloSollCorrected! AccSollCorrected! TestVeloSollCorrected TestAccSollCorrected	Auflistung! Reservierte Funktion, kein Standard!
0x00000201	Read	P,PID (Geschw.)	REAL64	z. B. mm/s		Geschwindigkeitsanteil des Reglers	Base Unit / s
0x00000202	Read	P,PID (Geschw.)	REAL64	%		Geschwindigkeitsanteil des Reglers in Prozent	Nicht oszilloskopierbar!
0x00000203	Read	P,PID (Geschw.)	REAL64	V		Geschwindigkeitsanteil des Reglers in Volt	Nicht oszilloskopierbar!
0x00000201	Read	P/PID (Geschw.)	REAL64	z. B. mm/s		P-Anteil des Reglers in absoluten Einheiten	
0x00000202	Read	P/PID (Geschw.)	REAL64	z. B. mm/s		I-Anteil des Reglers in absoluten Einheiten	
0x00000203	Read	P/PID (Geschw.)	REAL64	z. B. mm/s		D-Anteil des Reglers in absoluten Einheiten	
0x00000204	Read	P/PID (Geschw.)	UINT16	1	0/1	Begrenzung des I-Anteils aktiv?	
0x00000205	Read	P/PID (Geschw.)	UINT16	1	0/1	Begrenzung des D-Anteils aktiv?	
0x00000206	Read	P/PID (Geschw.)	UINT16	1	0/1	ARW-Massnahmen des I-Anteils aktiv?	ARW: Anti Reset Windup
0x0000020A	Read	P/PID (Geschw.)	REAL64	z. B. mm/s		Gesamteingangsgröße des Geschwindigkeitsreglers	
0x00000250	Read	P/PI (Beobachter)	REAL64	z. B. mm		Beobachter: Positionsdifferenz (Istposition - Beobachter-Position)	
0x00000251	Read	P/PI (Beobachter)	REAL64	z. B. mm		Beobachter: Position	
0x00000252	Read	P/PI (Beobachter)	REAL64	z. B. mm/s		Beobachter: Geschwindigkeit 2 (für P-Anteil)	
0x00000253	Read	P/PI (Beobachter)	REAL64	z. B. mm/s		Beobachter: Geschwindigkeit 1 (für I-Anteil)	

Index-Offset (Hex)	Zugriff	Reglertyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000254	Read	P/PI (Beobachter)	REAL64	z. B. mm/s^2		Beobachter: Beschleunigung	
0x00000255	Read	P/PI (Beobachter)	REAL64	A		Beobachter: Motor-Ist-Strom	
0x00000256	Read	P/PI (Beobachter)	UINT16	1	0/1	Beobachter: Begrenzung des I-Anteils aktiv?	
0x00000A00	Read	PID (MW)	REAL64	%	[-1.0...1.0]	Verrechnung der Sollgeschwindigkeit (Vorsteuerung) in Prozent	
0x00000A01	Read	PID (MW)	REAL64	z. B. mm/s		P-Anteil des Reglers in absoluten Einheiten oder Prozent (je nach Ausgabegewichtung)	
0x00000A02	Read	PID (MW)	REAL64	z. B. mm/s		I-Anteil des Reglers in absoluten Einheiten oder Prozent (je nach Ausgabegewichtung)	
0x00000A03	Read	PID (MW)	REAL64	z. B. mm/s		D-Anteil des Reglers in absoluten Einheiten oder Prozent (je nach Ausgabegewichtung)	
0x00000A04	Read	PID (MW)	UINT16	1	0/1	Begrenzung des I-Anteils aktiv?	
0x00000A05	Read	PID (MW)	UINT16	1	0/1	Begrenzung des D-Anteils aktiv?	
0x00000A10	Read	PID (Pos.)	REAL64	z. B. mm/s		Beschleunigungsvorsteuerung $Y_{acc}$ des Reglers in absoluten Einheiten	Beschleunigung s- vorsteuerung

**3.1.1.5.4.6.3 "Index-Offset" Spezifikation für Reglerfunktionen (Index-Group 0x6200+ ID)**

Index-Offset (Hex)	Zugriff	Reglertyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung

**3.1.1.5.4.7**      **Spezifikation Drive****3.1.1.5.4.7.1**      ***"Index-Offset" Spezifikation für Drive-Parameter (Index-Group 0x7000 + ID)***

Index-Offset (Hex)	Zugriff	Drive-Typ	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	UINT32	1	[1 ... 255]	Drive-ID	
0x00000002	Read	every	UINT8[30+1]	1	30 Zeichen	Drive-Name	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Drive-Typ <a href="#">▶ 168</a>	
0x00000004	Read/Write	every	UINT32	1	Byteoffset	Input-Adress-Offset (IO-Input-Image)	Änderung der IO-Adresse
0x00000005	Read/Write	every	UINT32	1	Byteoffset	Output-Adress-Offset (IO-Output-Image)	Änderung der IO-Adresse
0x00000006	Read/Write	every	UINT16	1	[0,1]	Motorpolarität	Schreiben ist bei erteilter Reglerfreigabe nicht erlaubt.
0x0000000A	Read/Write	every	UINT32	1	s. ENUM (>0)	Drive-Modus	Default: 1=STANDARD
0x0000000B	Read/Write	every	REAL64	%	[-1.0 ... 1.0]	Minimale Ausgabebeschränke (Ausgabelimitierung) (Default-Einstellung: -1.0 == -100%)	
0x0000000C	Read/Write	every	REAL64	%	[-1.0 ... 1.0]	Maximale Ausgabebeschränke (Ausgabelimitierung) (Default-Einstellung: 1.0 == 100%)	
0x0000000D	Read	every	UINT32	INC		Maximale Anzahl von Ausgabeinkrementen (Ausgabemaske)	
0x00000010	Read/Write	every	UINT32	1		Internes Drive-Control-Doppelwort zur Festlegung der Antriebs-Betriebsarten	Reserviert!
0x00000011	Read/Write	every	UINT32	1	≥ 5	Interner Drive-Reset-Zähler (Zeit in NC-Zyklen für Enable und Reset)	Reserviert!
0x00000020	Read/Write	every	UINT32	1	s. ENUM (≥0) s. Anhang	Drive-Totzeitkompensations-Modus  0: Aus (Default) 1: Ein (mittels Geschwindigkeit) 2: Ein (mittels Geschwindigkeit und Beschleunigung)	
0x00000021	Read/Write	every	UINT32	1		Steuerdoppelwort (32 Bits) für die Drive-Totzeitkompensation: Bit 0 = 0: relative IO Zeiten (Default) Bit 0 = 1: absolute IO Zeiten	
0x00000022	Read/Write	every	INT32	ns	[±1.0E+9]	Summe der parametrisierten zeitlichen Verschiebung für die Drive-Totzeitkompensation (typischerweise positive Zahlenwerte)	
0x00000031	Read/Write	every	REAL64	z. B. %/ INC	[-1.0E+30 ... 1.0E+30]	Skalierungsfaktor für Drehmoment-Istwert vom Antrieb (bzw. Kraft- oder Strom-Istwert)  z. B. AX5xxx: 0.1 => ±100%	NEU ab TC3.1



Index-Offset (Hex)	Zugriff	Drive-Typ	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000032	Read/Write	every	REAL64	s	[0.0 ... 60.0]	P-T1-Filterzeit für Drehmoment-Istwert (bzw. Kraft- oder Strom-Istwert)	NEU ab TC3.1
0x00000033	Read/Write	every	REAL64	s	[0.0 ... 60.0]	P-T1-Filterzeit für zeitliches Ableitung des Drehmoment-Istwerts (bzw. Kraft- oder Strom-Istwert)	NEU ab TC3.1
0x00000101	Read/Write	Servo	REAL64	z. B. mm/s	>0.0	Bezugsgeschwindigkeit bei Bezugs- bzw. Referenzoutput (Geschwindigkeitsvors- steuerung)	Base Unit / s
0x00000102	Read/Write	Servo	REAL64	%	[0.0 ... 5.0]	Bezugs- bzw. Referenzoutput in Prozent	
0x00000103	Read	Servo	REAL64	z. B. mm/s	>0.0	Resultierende Geschwindigkeit bei 100% Output	Base Unit / s
0x00000104	Read/Write	Servo	REAL64	z. B. mm/s	$\pm\infty$	Geschwindigkeitsoffset (DAC-Offset) für Driftabgleich (Offsetabgleich) der Achse	Base Unit / s
0x00000105	Read/Write	Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	Geschwindigkeitsskali- erung (Skalierungsfaktor um auf Wichtung im Antrieb zu reagieren)	Für Sercos, Profi Drive, AX200x, CANopen
0x00000106	Read/Write	Profi Drive DSC	UINT32	0.001 * 1/s	$\geq 0$	Profibus/Profi Drive DSC: Lageregelverstärkung K <sub>pc</sub>	Nur für Profi Drive DSC
0x00000107	Read/Write	Profi Drive DSC	REAL64	1	$\geq 0.0$	Profibus/Profi Drive DSC: Skalierung für Berechnung von 'XERR' (Default: 1.0)	Nur für Profi Drive DSC
0x00000109	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	Positionsskalierung (Skalierungsfaktor, um auf Wichtung im Antrieb zu reagieren)	Für Sercos, CANopen
0x0000010A	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	Beschleunigungsskali- erung (Skalierungsfaktor, um auf Wichtung im Antrieb zu reagieren)	Für Sercos, Profi Drive, AX200x, CANopen
0x0000010B	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	Drehmomentskalierung (rot. Motor) bzw. Kraftskalierung (Linearmotor) (Skalierungsfaktor, um auf Wichtung im Antrieb zu reagieren) für „TorqueOffset“ (additives Moment als Vorsteuerung)	Für Sercos, Profi Drive, AX200x, CANopen
0x0000010C	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	Drehmomentskalierung (rot. Motor) bzw. Kraftskalierung (Linearmotor) (Skalierungsfaktor, um auf Wichtung im Antrieb zu reagieren) für „SetTorque“ (z.B. MC_TorqueControl mit Drive OpMode CST)	Für Sercos, Profi Drive, AX200x, CANopen Ab TC 3.1 B4024.2

Index-Offset (Hex)	Zugriff	Drive-Typ	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000010D	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Verzögerungszeit für Drive-Geschwindigkeitsausgabe	Für Sercos, CANopen
0x0000010E	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Verzögerungszeit für Drive-Beschleunigungsausgabe	Für Sercos, CANopen
0x0000010F	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Verzögerungszeit für Drive-Drehmomentausgabe bzw. Kraftausgabe	Für Sercos, CANopen
0x00000120	Read/Write	Servo/ Hydraulik/	UINT32	1	≥ 0	Tabellen-ID (0: keine Tabelle)	Nur für KL4xxx, M2400, Universal
0x00000121	Read/Write	Servo/ Hydraulik	UINT32	1	≥ 0	Interpolation-Type 0: Linear 2: Spline	Nur für KL4xxx, M2400, Universal
0x00000122	Read/Write	Servo/ Hydraulik	REAL64	%	[-1.0 ... 1.0]	Ausgabeoffset in Prozent  Anmerkung: Wirkt nach der Kennlinienauswertung!	Nur für KL4xxx, M2400, Universal
0x00000151	Read/Write	Servo / Nichtlinear	REAL64	1	[0.0 ... 100.0]	Quadrantenausgleichs faktor (Verhältnis zwischen I und III Quadr.)	
0x00000152	Read/Write	Servo / Nichtlinear	REAL64	1	[0.01 ... 1.0]	Geschwindigkeit-Stützstelle in Prozent (1.0 == 100%)	
0x00000153	Read/Write	Servo / Nichtlinear	REAL64	1	[0.01 ... 1.0]	Ausgabe-Stützstelle in Prozent(1.0 == 100%)	
0x00000301	Read/Write	Schrittmotor	UINT8			Bit-Maske: Zyklus 1	
0x00000302	Read/Write	Schrittmotor	UINT8			Bit-Maske: Zyklus 2	
0x00000303	Read/Write	Schrittmotor	UINT8			Bit-Maske: Zyklus 3	
0x00000304	Read/Write	Schrittmotor	UINT8			Bit-Maske: Zyklus 4	
0x00000305	Read/Write	Schrittmotor	UINT8			Bit-Maske: Zyklus 5	
0x00000306	Read/Write	Schrittmotor	UINT8			Bit-Maske: Zyklus 6	
0x00000307	Read/Write	Schrittmotor	UINT8			Bit-Maske: Zyklus 7	
0x00000308	Read/Write	Schrittmotor	UINT8			Bit-Maske: Zyklus 8	
0x00000310	Read/Write	Schrittmotor	UINT8			Bit-Maske: Haltestrom	

**3.1.1.5.4.7.2**     *"Index-Offset" Spezifikation für Drive-Zustand (Index-Group 0x7100 + ID)*

Index-Offset (Hex)	Zugriff	Drive-Typ	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	INT32			Fehlerstatus Drive	
0x00000002	Read	every	REAL64	z. B. mm/s		Gesamtausgabe in absoluten Einheiten	Base Unit / s <i>Symbolischer Zugriff möglich!</i> „DriveOutput“
0x00000003	Read	every	REAL64	%		Gesamtausgabe in Prozent	
0x00000004	Read	every	REAL64	V		Gesamtausgabe in Volt	Nicht oszilloskopierbar!
0x00000005	Read	every	REAL64	z. B. mm/s		PeakHold-Wert für maximale negative Gesamtausgabe	Base Unit / s
0x00000006	Read	every	REAL64	z. B. mm/s		PeakHold-Wert für maximale positive Gesamtausgabe	Base Unit / s
0x00000007	Read	every	REAL64	z. B. 100%=1000, z. B. Nm bzw. N		Istmoment bzw. Istkraft (typisch 100%=1000)	ab TC3.1 B4022 <i>Symbolischer Zugriff möglich!</i> „ActTorque“
0x00000008	Read	every	REAL64	z. B. Nm/s bzw. N/s	$\pm\infty$	Istmomentänderung bzw. Istkraftänderung  (zeitliche Ableitung des Istmomentes bzw. der Istkraft)	ab TC3.1 B4024
0x0000000C	Read	every	REAL64	z. B. mm		Sollpositions-Korrekturwert für Drive Ausgabe aufgrund der Totzeitkompensation	
0x0000000D	Read	every	REAL64	s		Summe der zeitlichen Verschiebung für Drive-Totzeitkompensation (parametrierte und variable Totzeit)  Anm.: Eine Totzeit wird im System als positiver Wert angegeben.	
0x00000013	Read	every	REAL64	%		Gesamtausgabe in Prozent (nach nichtlinearer Kennlinie!)	
0x00000014	Read	every	REAL64	V		Gesamtausgabe in Volt (nach nichtlinearer Kennlinie!)	Nicht oszilloskopierbar!
0x0000011A	Read	Servo (Sercos, CANopen)	REAL64	z. B. mm		Optionale Ausgabefilterung: Gefilterte Sollposition	NEU Für Sercos, CANopen
0x0000011E	Read	Servo (Sercos, CANopen)	REAL64	z. B. mm/s		Optionale Ausgabefilterung: Gefilterte Sollgeschwindigkeit	NEU Für Sercos, CANopen
0x0000011F	Read	Servo (Sercos, CANopen)	REAL64	z. B. mm/s <sup>2</sup>		Optionale Ausgabefilterung: Gefilterte Sollbeschleunigung / Sollverzögerung	NEU Für Sercos, CANopen

Index-Offset (Hex)	Zugriff	Drive-Typ	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000200	ReadWrite		<b>READ:</b>			Lesen des Zustandes der digitalen Eingänge 1 bis 8	ab TC3.1 B4024.12 Nur für SAF-Port 501!
			UINT32	1	0/1	Zustand des ausgewählten Eingangs	
			<b>WRITE:</b>				
			UINT32	1	[1...8]	Auswahl des Eingangs 1 bis 8	

**3.1.1.5.4.7.3 "Index-Offset" Spezifikation für Drive-Funktionen (Index-Group 0x7200 + ID)**

Index-Offset (Hex)	Zugriff	Drive-Typ	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000102	Write	SERVO	{			Austragen und Löschen der Kennlinien-Tabelle im Drive	Nur für SAF-Port 501!
			ULONG	1	>0	Tabellen-ID s. Achsfunktion mit Index-Offset 0x00000012	
			}				

**3.1.1.5.4.7.4**     ***"Index-Offset" Spezifikation für zyklische Drive-Prozessdaten  
(Index-Group 0x7300 + ID)***

Index-Offset (Hex)	Zugriff	Drive-Typ	Daten- typ	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000000	Read/Write	every (NC→IO)	{		STRUCT s. Drive-Interface	DRIVE-OUTPUT- STRUKTUR (NC→IO, 40 Byte) <i>NCDRIVESTRUCT_0 UT2</i>	Write-Befehl nur optional! Sicherheitsaspe kte beachten!
			INT32	INC	≥ 0	nOutData1	
			INT32	INC	±2^31	nOutData2	
			UINT8	1	≥ 0	nControl1	
			UINT8	1	≥ 0	nControl2	
			UINT8	1	≥ 0	nControl3	
			UINT8	1	≥ 0	nControl4	
			INT32	INC	≥ 0	nOutData3	
			INT32	INC	≥ 0	nOutData4	
			INT32	INC	≥ 0	nOutData5	
			INT32	INC	≥ 0	nOutData6	
			UINT8	1	≥ 0	nControl5	
			UINT8	1	≥ 0	nControl6	
			UINT8	1	≥ 0	nControl7	
			UINT8	1	≥ 0	nControl8	
			INT32	1	≥ 0	reserviert	
			INT32	1	≥ 0	reserviert	
			}				
0x00000001	Write	every (NC→IO)	{		STRUCT s. Drive-Interface	Bitweiser Zugriff auf DRIVE-OUTPUT- STRUKTUR (NC→IO, 40 Byte) <i>NCDRIVESTRUCT_0 UT2</i>	Write-Befehl nur optional! Sicherheitsaspe kte beachten!
			UINT32	1	[0 ... 39]	ByteOffset Relative address offset [0..39] in output structure. E.G.: To write "nControl1" the ByteOffset must be 8.	
			UINT32	1	[0x00000000... 0xFFFFFFFF]	BitSelectMask (BSM) The mask defines write enabled bits in a DWORD. Zero bits are protected and remain unaffected.	
			UINT32	1	[0x00000000... 0xFFFFFFFF]	Value Only those bits in value are overwritten where BSM equals 1.	
			}				

Index-Offset (Hex)	Zugriff	Drive-Typ	Daten- typ	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000080	Read	every (IO→NC)	{		STRUCT s. Drive-Interface	DRIVE-INPUT- STRUKTUR (IO→NC, 40 Byte) NCDRIVESTRUCT_IN 2	
			INT32	INC	≥ 0	nInData1	
			INT32	INC	±2^31	nInData2	
			UINT8	1	≥ 0	nStatus1	
			UINT8	1	≥ 0	nStatus2	
			UINT8	1	≥ 0	nStatus3	
			UINT8	1	≥ 0	nStatus4	
			INT32	INC	≥ 0	nInData3	
			INT32	INC	≥ 0	nInData4	
			INT32	INC	≥ 0	nInData5	
			INT32	INC	≥ 0	nInData6	
			UINT8	1	≥ 0	nStatus5	
			UINT8	1	≥ 0	nStatus6	
			UINT8	1	≥ 0	nStatus7	
			UINT8	1	≥ 0	nStatus8	
			INT32	1	≥ 0	Reserviert	
			INT32	1	≥ 0	Reserviert	
			}				



**3.1.1.5.4.8 Spezifikation Tabellen****3.1.1.5.4.8.1 *"Index-Offset" Spezifikation für Tabellenparameter (Index-Group 0xA000 + ID)***

Index-Offset (Hex)	Zugriff	Tabellentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000001	Read	every	UINT32	1	[1 ... 255]	Tabellen-ID	
0x00000002	Read	every	UINT8[30+1]	1	30 Zeichen	Tabellenname	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	<a href="#">Tabellen-Untertypen</a> [►_170]	
0x00000004	Read	every	UINT32	1	s. ENUM (>0)	<a href="#">Tabellen-Haupttypen</a> [►_170]	
0x00000010	Read	every	UINT32	1	[0... 16777216]	Anzahl von Zeilen (n)	
0x00000011	Read	every	UINT32	1	[0... 16777216]	Anzahl von Spalten (m)	
0x00000012	Read	every	UINT32	1	≥0	Anzahl von Gesamtelementen (n*m)	
0x00000013	Read	äquidistante Tab.	REAL64	z. B. mm	≥0.0	Schrittweite (Positionsdelta) (äquidistante Tabellen)	Base Unit
0x00000014	Read	zyklische Tab.	REAL64	z. B. Grad	≥0.0	Masterperiode (zyklische Tabellen)	Base Unit
0x00000015	Read	zyklische Tab.	REAL64	z. B. Grad	≥0.0	Slavedifferenz pro Masterperiode (zyklische Tabellen)	Base Unit
0x0000001A	Read /Write	"Motion Function" (Bewegungsgesetze)	{ UINT32 REAL64 UINT32 UINT32 }	ENUM z. B. mm ENUM ENUM	s. Anhang $\pm \infty$ s. Anhang s. Anhang	Aktivierungstyp für Online-Änderungen von Tabellendaten (nur MF) Aktivierungstyp 0: 'instantaneous' (default) 1: 'master cam pos.' 2: 'master' axis pos.' 3: 'next cycle' 4: 'next cycle once' 5: 'as soon as possible' 6: 'off' 7: 'delete queued data' Reserve (TC3) Aktivierungsposition Master-Skalierungstyp 0: user defined (default) 1: scaling with auto offset 2: off Slave-Skalierungstyp 0: user defined (default) 1: scaling with auto offset 2: off	Geändert ab TC3
0x00000020	Read /Write	every	{ UINT32 UINT32 REAL64 }	1 1 z. B. mm	$\pm \infty$	Schreiben Einzelwert [n,m]: n-te Zeile m-te Spalte Einzelwert	

Index-Offset (Hex)	Zugriff	Tabellentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000021	ReadWrite	every	*REAL64	z. B. mm	$\pm \infty$	Slaveposition zu r vorgegebenen Masterposition lesen (bezieht sich auf "Rohwerte" der Tabelle)	
0x00000022	ReadWrite	"Motion Function" (Bewegungsgesetze)	<b>Write</b>			Lesen der "Motion Function" als "Punktewolke"	Nur zeilenweise möglich! (ganzzahliges vielfaches)  Geändert in TC3
			{				
			UINT 16	1	0 / 1	Konsistente Datenübernahme veranlassen?	
			UINT16	1	Bitmask ( $\geq 0$ )	Auswahl-Bitmaske (Spaltenanzahl m ist Masterposition plus Anzahl Bits): Bit 0: Pos (Slave) Bit 1: Velo (Slave) Bit 2: Acc (Slave) Bit 3: Jerk (Slave)	
			UINT32			Reserve (TC3)	
			REAL64	z. B. mm	$\pm \infty$	Startposition (Master)	
			REAL64	z. B. mm	$> 0.0$	Schrittweite	
			}				
			<b>Read</b>				
			{				
REAL64[x*m]	z. B. mm	$\pm \infty$	Lesen von x Zeilen ab der Master-Startposition : (x*m)-Werte (eine oder mehrere Zeilen)				
}							
0x00000023	ReadWrite	every	<b>Write</b>			Slavewerte zur vorgegebenen Masterposition lesen (bezieht sich auf "Rohwerte" der Tabelle)	
			REAL64	z. B. mm	$\pm \infty$	Masterposition	
			<b>Read</b>				
			{				
			REAL64	z. B. mm	$\pm \infty$	Slaveposition	
			REAL64	mm/s	$\pm \infty$	Slavegeschwindigkeit	
REAL64	mm/s <sup>2</sup>	$\pm \infty$	Slavebeschleunigung				
}							
0x00000050	Read /Write	every	REAL64 [64]	1	$\pm \infty$	<u>Charakteristische Kennwerte der Tabelle</u> [►_172]	

Index-Offset (Hex)	Zugriff	Tabellentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00000050	ReadWrite	every	<b>Write</b>			Lesen der Charakteristischen Kennwerte einer Tabelle in Abhängigkeit der nominalen Mastergeschwindigkeit	Geändert ab TC3
			REAL64 [64]	...	$\pm \infty$	Optionale nominale Masterbezugsgeschwindigkeit "fMasterVeloNom" (normiert => 1.0 mm/s), die restlichen Element werden nicht ausgewertet	
			<b>Read</b>				
			REAL64 [64]	...	$\pm \infty$	Lesen der <u>Charakteristische Kennwerte einer Tabelle</u> [172]	
0x00000115	Write	monoton linear, monoton zykl.,	{			Setzen/Ändern der Tabellenskalierung:	
			REAL64	1	[ ± 1000000.0]	Originalgewichtung der Tabelle	
			REAL64	z. B. mm	[ ± 1000000.0]	Positionsoffset der Masterspalte	
			REAL64	1	[ ± 1000000.0]	Skalierung der Masterspalte	
			REAL64	z. B. mm	[ ± 1000000.0]	Positionsoffset der Slavespalte	
			REAL64	1	[ ± 1000000.0]	Skalierung der Slavespalte	
			REAL64	z. B. mm	[ ± 1000000.0]	Untere Bereichsgrenze (Anfangsposition)	
			REAL64	z. B. mm	[ ± 1000000.0]	Obere Bereichsgrenze (Endposition)	
			}				
0x01000000 +n-te Startzeile	Read/ Write[≤16777216]	every	{ REAL64[x*m] }	z. B. mm	$\pm \infty$	Lesen/Schreiben von x Zeilen ab der n-ten Zeile: (x*m)-Werte (eine oder mehrere Zeilen) Wertebereich n: [0 ... 16777216]	Nur zeilenweise möglich! (ganzzahliges vielfaches)
0x02000000 +m-te Startspalte	Read/ Write[≤16777216]	every	{ REAL64[x*n] }	z. B. mm	$\pm \infty$	Lesen/Schreiben x Spalten ab der m-ten Spalte: (x*n)-Werte (eine oder mehrere Spalten)Wertebereich m: [0 ... 16777216]	Nur spaltenweise möglich! (ganzzahliges vielfaches)

Index-Offset (Hex)	Zugriff	Tabellentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x05000000 +n-te Startzeile	Read/ Write[<=167772 16]	"Motion Function" (Bewegungs- gesetze) Daten:STRUCT [x*m]	{			Lesen/Schreiben von x Zeilen ab der n-ten Zeile: (x*m)- Strukturen (eine oder mehrere Zeilen) Wertebereich n: [0 ... 16777216]	Nur zeilenweise möglich! (ganzzahliges vielfaches)  Geändert ab TC3
			UINT32	1		Abs. Punktindex (nicht ausgewertet)	
			UINT16	ENUM		Funktionsstyp 1: Polynom 1 15: Polynom 5	
			UINT16	ENUM		Punkttyp 0: default 1: ignore	
			INT32	1		Rel. Adressindex auf Zielpunkt (Default: 1)	
			UINT32			Reserve (TC3)	
			REAL64	mm		Masterposition	
			REAL64	mm		Slaveposition	
			REAL64	mm /s		Slavegeschwind.	
			REAL64	mm/s^2		Slavebeschleun.	
			REAL64	mm/s^3		Slaveruck	
			}				
0x06000000 +m-te Startspalte	Read/ Write[<=167772 16]	"Motion Function" (Bewegungs- gesetze) Daten:STRUCT [x* n ]	{			Lesen/Schreiben x Spalten ab der m-ten Spalte: (x*n)- Strukturen (eine oder mehrere Spalten) Wertebereich m: [0 ... 16777216]	Nur spaltenweise möglich! (ganzzahliges vielfaches)  Geändert ab TC3
			UINT32	1		Abs. Punktindex (nicht ausgewertet)	
			UINT16	ENUM		Funktionsstyp 1: Polynom 1 15: Polynom 5	
			UINT16	ENUM		Punkttyp 0: default 1: ignore	
			INT32	1		Rel. Adressindex auf Zielpunkt (Default: 1)	
			UINT32			Reserve (TC3)	
			REAL64	mm		Masterposition	
			REAL64	mm		Slaveposition	
			REAL64	mm /s		Slavegeschwind.	
			REAL64	mm/s^2		Slavebeschleun.	
			REAL64	mm/s^3		Slaveruck	
			}				

**3.1.1.5.4.8.2 "Index-Offset" Spezifikation für Tabellenzustand (Index-Group 0xA100 + ID)**

Index-Offset (Hex)	Zugriff	Tabellentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x0000000A	Read	every	INT32	1	≥ 0	'User Counter' (Anzahl der Nutzer dieser Tabelle)	Nicht oszilloskopierba r!

**3.1.1.5.4.8.3 "Index-Offset" Spezifikation für Tabellenfunktionen (Index-Group 0xA200 + ID)**

Index-Offset (Hex)	Zugriff	Tabellentyp	Datentyp	Phys. Einheit	Definitionsbe- reich	Beschreibung	Anmerkung
0x00010000	Write	Kurvenscheibe	{			Erzeugt Kurvenscheiben-Tabelle mit Dimension (n*m):	Tabellentypen: 1,2,3,4 Dimension: mind. 2x1
			UINT32	1	s. ENUM (>0)	Tabellentyp [► 170] (s. Anhang)	
			UINT32	1	[2...16777216]	Anzahl der Zeilen	
			UINT32	1	[1...16777216]	Anzahl der Spalten	
			}				
0x00010001	Write	Kennlinie	{			Erzeugt Kennlinien-Tabelle mit Dimension (n*m):	Tabellentypen: 1,3 Dimension: mind. 2x1
			UINT32	1	s. ENUM (>0)	Tabellentyp [► 170] (s. Anhang)	
			UINT32	1	[2...16777216]	Anzahl der Zeilen	
			UINT32	1	[1...16777216]	Anzahl der Spalten	
			}				
0x00010010	Write	"Motion Function" (Bewegungsgesetze)	{			Erzeugt "Motion Function"-Tabelle mit Dimension (n*m):	Tabellentypen: 3,4 Dimension: mind. 2x1
			UINT32	1	s. ENUM (>0)	Tabellentyp [► 170] (s. Anhang)	
			UINT32	1	[2...16777216]	Anzahl der Zeilen	
			UINT32	1	[1...16777216]	Anzahl der Spalten	
			}				
0x00020000	Write	every	VOID			Löscht Tabelle mit Dimension (n*m)	Tabellentypen: 1,2,3,4
0x00030000	Write	every	VOID			Initialisiert Tabelle Initialisierung ist nicht mehr nötig, da dies jetzt automatisch in folgenden Fällen passiert: a) beim Ankoppeln mittels Tabelle b) beim Auslesen der Slaveposition (s. Tabellenpara.)	

**3.1.1.5.4.9 Anhang**

**Enum Kanaltypen**

Define	Kanaltypen
1	Standard
2	Interpreter
3	FIFO
4	Kinematische Transformation

**Enum Interpretertypen**

Define	Interpretertypen
0	NICHT DEFINIERT
1	NC Interpreter DIN 66025 (GST)
2	NC Interpreter DIN 66025 (Klassischer Dialekt)

**Enum Interpreter-Betriebsarten (Operation-Mode)**

Define	Interpreter-Betriebsart
0x0	Default-Belegung (Abwahl der übrigen Betriebsarten)
0x1	Einzelsatzbetrieb im NC-Kern (Satzausführungstask/SAF)
0x1000	reserviert
0x2000	reserviert
0x4000	Einzelsatzbetrieb im Interpreter

**Enum Interpolations-Lade-Logmodus**

Define	Lade-Logmodus
0	Loaderlog aus
1	nur Source
2	Source & Compiled

**Enum Interpolations-Trace-Modus**

Define	Trace-Modus
0	Trace aus
1	Trace Zeilennummern
2	Trace Source

**Enum Interpreterstatus**

verschoben nach: System Manager Interface für den Interpreter - Interpreterelement

**Enum Gruppentypen**

Define	Gruppentypen
0	NICHT DEFINIERT
1	PTP-Gruppe + x Slave
2	1D-Gruppe + x Slave
3	2D-Gruppe + x Slave
4	3D-Gruppe + x Slave
5	Eil/Schleich + x Slave
6	Low Cost Schrittmotor (dig. IO) + x Slave
7	Tabellen-Gruppe + x Slave
9	Encodergruppe + x Slave
11	FIFO-Gruppe + x Slave
12	Kinematik-Transformations-Gruppe + x Slave

**Enum Kurvengeschwindigkeitsreduktionsmethode**

verschoben nach: System Manager Interface für den Interpreter - Gruppenelement

**Enum Achstypen**

Define	Achstypen
0	NICHT DEFINIERT
1	Kontinuierliche Achse (Servo)
2	Diskrete Achse (Eil/Schleich)
3	Kontinuierliche Achse (Schrittmotor)
5	Encoder Achse
6	Kontinuierliche Achse (mit Betriebsartumschaltung für Positions-/ Druck-Regelung)
7	Time Base Generator
100	

**Enum Schrittmotorbetriebsart**

Define	Schrittmotorbetriebsart
0	NICHT DEFINIERT
1	2-Phasige Erregung (4 Zyklen)
2	1-2-Phasige Erregung (6 Zyklen)
3	Leistungsteil

**Enum Override-Typen für PTP-Achsen (Geschwindigkeitsoverride)**

Define	Override-Typen
1	Reduziert Alte Variante, abgelöst durch "(3) Reduziert (iteriert)"
2	Original Alte Variante, abgelöst durch "(4) Original (iteriert)"
3	Reduziert (iteriert) Default-Wert: Der Overridewert wird auf die im Sonderfall intern reduzierte Geschwindigkeit bezogen. Somit ergibt sich für den gesamten Overridebereich von 0...100% eine direkt proportionale Geschwindigkeit (=> linearer Zusammenhang).
4	Original (iteriert) Der Overridewert wird immer auf die durch den Anwender programmierte Geschwindigkeit bezogen. Wenn allerdings diese Geschwindigkeit nicht gefahren werden kann, dann ergibt sich ein maximaler Overridewert, ab dem keine höhere Geschwindigkeit erreicht wird (=> Limitierung).

**Enum Gruppen/Achs-Starttypen**

Define	Gruppen/Achs-Starttypen
0	NICHT DEFINIERT
1	Absolutstart
2	Relativstart
3	Endlosstart positiv
4	Endlosstart negativ
5	Modulostart (ALT)
261	Modulostart auf kürzestem Weg
517	Modulostart in positiver Fahrtrichtung (mit Modulo-Toleranzfenster)
773	Modulostart in negativer Fahrtrichtung (mit Modulo-Toleranzfenster)
4096	Stopp und Sperre (Achse wird für Bewegungskommandos gesperrt)
8192	Halt (ohne Bewegungs-Sperre)



**Enum Kommandospeichertypen (buffer mode) für universellen Achsstart (UAS)**

Define	Kommandospeichertypen (buffer mode)
0	ABORTING (Default) (instantan, löst eine aktuelle Bewegung ab und löscht gepufferte Kommandos)
1	BUFFERED (Auftrag wird im Kommando Zwischenspeicher gepuffert um ihn im Anschluss an eine aktive Bewegung auszuführen)
18	BLENDING LOW (gepuffert, kein Stopp, Zwischenziel wird mit der niedrigsten Geschwindigkeit zweier Kommandos durchlaufen)
19	BLENDING PREVIOUS (gepuffert, kein Stopp, Zwischenziel wird mit der Geschwindigkeit des aktiven Kommandos durchlaufen)
20	BLENDING NEXT (gepuffert, kein Stopp, Zwischenziel wird mit der Geschwindigkeit des gepufferten Kommandos durchlaufen)
21	BLENDING HIGH (gepuffert, kein Stopp, Zwischenziel wird mit der höheren Geschwindigkeit zweier Kommandos durchlaufen)

**Enum Endpositionstypen (Neue Endposition)**

Define	Endpositionstypen
0	NICHT DEFINIERT
1	Absolutposition
2	Relativposition
3	Endlosposition positiv
4	Endlosposition negativ
5	Moduloposition

**Enum Kommandotypen für neue Endposition mit neuer Geschwindigkeit (Neue Endposition und/oder neue Geschwindigkeit)**

Define	Kommandotypen für neue Endposition mit neuer Geschwindigkeit
0	NICHT DEFINIERT
1	Position (instantan)
2	Geschwindigkeit (instantan)
3	Position und Geschwindigkeit (instantan)
9	Position (Umschaltposition)
10	Geschwindigkeit (Umschaltposition)
11	Position und Geschwindigkeit (Umschaltposition)

**Enum Istpositionstypen (Setze Istposition)**

Define	Istpositionstypen
0	NICHT DEFINIERT
1	Absolutposition
2	Relativposition
5	Moduloposition

**Enum Kompensationstypen (Streckenkompensation bzw. Superimposed)**

Define	Kompensationstypen
0	NICHT DEFINIERT
1	VELOREDUCTION_ADDITIVEMOTION Die max. Geschwindigkeit VelocityDiff wird reduziert. Die Strecke, auf der die Ausgleichsfahrt wirkt, setzt sich additiv aus Length+Distance zusammen.
2	VELOREDUCTION_LIMITEDMOTION Die max. Geschwindigkeit VelocityDiff wird reduziert. Die Strecke, auf der die Ausgleichsfahrt wirkt, ist durch den Parameter Length festgelegt.
3	LENGTHREDUCTION_ADDITIVEMOTION Die max. zur Verfügung stehende Strecke wird reduziert und setzt sich maximal aus Length+Distance zusammen. Es wird versucht, die max. Geschwind. VelocityDiff zu nutzen.
4	LENGTHREDUCTION_LIMITEDMOTION Die max. zur Verfügung stehende Strecke wird reduziert und ist durch den Parameter Length begrenzt. Es wird versucht, die max. Geschwind. VelocityDiff zu nutzen.

**Enum Slavetypen**

Define	Slavetypen
0	NICHT DEFINIERT
1	Linear
2	Fliegende Säge (Geschwindigkeit, ruckbegrenzttes Profil)
3	Fliegende Säge (Position und Geschwindigkeit, ruckbegrenzttes Profil)
5	Synchronisierungsgenerator (Geschwindigkeit, ruckbegrenzttes Profil)
6	Synchronisierungsgenerator (Position und Geschwindigkeit, ruckbegrenzttes Profil)
10	Tabular
11	Multitabular
13	'Motion Function' (MF)
15	Linear mit zyklischer Getriebefaktoränderung (Rampenfilter zur Beschleunigungsbegrenzung)
100	Specific

**Enum Slave-Entkopplungstypen (für nachfolgendes Achskommando)**

Define	Slave Entkopplungstypen (für nachfolgendes Achskommando)
0	Stopp, E-Stopp oder P-Stopp (Default) (STOP)
1	Orientierter Stopp (O-Stopp) (ORIENTEDSTOP)
2	Beschleunigungsfrei fahren (force-free) und weiterfahren auf endlose Zielposition (ENDLESS)
3	Weiterfahren auf endlose Zielposition mit neuer geforderter Geschwindigkeit (ENDLESS_NEWVELO)
4	neue Endposition (NEWPOS)
5	neue Endposition und neue geforderte Geschwindigkeit (NEWPOSANDVELO)
6	logisch Abkoppeln und Achse sofort ohne Geschwindigkeitsrampe stillsetzen (INSTANTANEOUSSTOP)

**Enum Reglertypen**

Define	Reglertypen
0	NICHT DEFINIERT
1	P-Regler (Standard) (Position)
2	PP-Regler (mit ka) (Position)
3	PID-Regler (mit ka) (Position)
5	P-Regler (Geschwindigkeit)
6	PI-Regler (Geschwindigkeit)
7	Eil/Schleich-Regler (Position)
8	Schrittmotor-Regler (Position)
9	SERCOS-Regler (Position im Antrieb)
10	RESERVIERT
11	RESERVIERT
12	RESERVIERT
13	RESERVIERT
14	TCom Controller (Soft Drive) (Position im Antrieb)

**Enum Regler-Beobachtermodus**

Define	Regler-Beobachtermodus
0	Kein Beobachter aktiv (Default)
1	"Luenberger"-Beobachter (klassischer Beobachter-Entwurf)

## Enum Encodertypen

Define	Encodertypen
0	NICHT DEFINIERT
1	Simulation Encoder (Inkremental)
2	M3000 Encoder (Multi/Single-Turn) (Absolut)
3	M31x0 / M2000 Encoder (Inkremental)
4	MDP 511 Encoder: EL7041, EL7342, EL5101, EL5151, EL2521, EL5021, IP5101 (Inkremental)
5	MDP 500/501 Enc.: EL5001, IP5009, KL5001 (SSI) (Absolut)
6	MDP 510 Encoder: KL5051, KL2502-30K Encoder (BiSSI) (Inkremental)
7	KL30xx Encoder (Analog) (Absolut)
8	SERCOS und EtherCAT SoE (Position) (Inkremental)
9	SERCOS und EtherCAT SoE (Position und Geschwindigkeit) (Inkremental)
10	Binärer Encoder (0/1) (Inkremental)
11	M2510 Encoder (Absolut)
12	FOX50 Encoder (Absolut)
14	AX2000 (Lightbus) (Inkremental)
15	Provi-Drive MC (Simodrive 611U) (Inkremental)
16	Universal Encoder (variable Bitmaske) (Inkremental)
17	NC Rückwand (Inkremental)
18	spezieller CANopen Typ (z. B. Lenze Drive 9300) (Inkremental)
19	MDP 513 (DS402): CANopen und EtherCAT CoE (AX2xx-B1x0/ B510, EL7201) (Inkremental)
20	AX2xx-B900 (Ethernet) (Inkremental)
21	KL5151 Encoder (Inkremental)
24	IP5209 Encoder (Inkremental)
25	KL2531/KL2541 Encoder (Stepper Motor) (Inkremental)
26	KL2532/KL2542 Encoder (DC Motor), KL2535/KL2545 (PWM Stromklemme) (Inkremental)
27	Time Base Encoder (Zeitgeber) (Inkremental)
28	TCom Encoder (Soft Drive) (Inkremental)

**Enum Encodermodus**

Define	Encodermodus
0	NICHT DEFINIERT
1	Ermittlung Position
2	Ermittlung Position und Geschwindigkeit
3	Ermittlung Position, Geschwindigkeit und Beschleunigung

**Enum Encoder-Auswerterichtung (log. Zählrichtung)**

Define	Encoder-Auswerterichtung (log. Zählrichtung)
0	Auswertung in positive und negative Zählrichtung (Defaultbelegung, d.h. kompatibel zum bisherigen Zustand)
1	Auswertung nur in positive Zählrichtung
2	Auswertung nur in negative Zählrichtung
3	Auswertung weder in positive noch in negativer Zählrichtung (Auswertung gesperrt)



Nicht für alle Encodertypen, sondern nur für KL5101, KL5151, KL2531, KL2541, IP5209, Universal-Encoder, ...

Enc.-Auswerterichtung (Log. Zählrichtung)	Encodertypen		
	KL5101, ...	Universal-Encoder	übrige Typen
0: positiv und negativ	√	√	—
1: nur positiv	√	√	—
2: nur negativ	√	√	—
3: gesperrt	√	√	—

**Enum Vorzeicheninterpretation (Datentyp) des Encoders**

Define	Vorzeicheninterpretation (Datentyp) der Encoder-Istinkremente
0	NICHT DEFINIERT (Defaultbelegung, d.h. kompatibel zum bisherigen Stand)
1	UNSIGNED: Vorzeichenlose Interpretation der Encoder-Istinkremente
2	SIGNED: Vorzeichenbehafete Interpretation der Encoder-Istinkremente



Vorerst nur für KL30xx/KL31xx

**Enum Encoder-Bezugsmaßsystem**

Define	Encoder-Bezugsmaßsystem
0	INC: Inkrementelles Bezugsmaßsystem mit Unter- und Überlaufverrechnung (Default, d.h. kompatibel zum bisherigen Stand)
1	ABS: Absolutes Bezugsmaßsystem ohne Unter- und Überlaufverrechnung (keine Unter- oder Überläufe des Gebers erlaubt)
2	ABS MODULO: Bedingt absolutes Bezugsmaßsystem, da Unter- und Überlaufverrechnung (Absolutwert, der sich modulo (endlos) fortsetzt)



Nicht für alle Encodertypen, sondern nur für Profi Drive MC, M3000, KL5001/EL5001, IP5009, SERCOS, UNIVERSAL, ...

## Enum Referenziermodus für Inkremental-Encoder

Define	Parametertext	Referenziermodus für Inkremental-Encoder
0	Default	NICHT DEFINIERT (Default-Belegung, d. h. kompatibel zum bisherigen Stand)
1	Homing Sensor Only (PLC cam or digital input)	Latchereignis: Herunterfahren von der SPS Nocke (negative Flanke)
2	Hardware Sync (feedback reference pulse)	Latchereignis: Hardware Syncimpuls (Nullspur)
3	Hardware Latch 1 (pos. Edge)	Latchereignis: Externes Hardware Latch mit positiver Flanke (Messtaster bzw. Fliegendes Messen mit pos. Flanke)
4	Hardware Latch 1 (neg. Edge)	Latchereignis: Externes Hardware Latch mit negativer Flanke (Messtaster bzw. Fliegendes Messen mit neg. Flanke)
5	Software Sync	Latchereignis: Synthetisch nachgebildeter Software Syncimpuls (Software Nullspur); VORAUSSETZUNG: absolut pro Motorumdreh., z. B. Resolver!
6	Hardware Latch 1 (pos. Edge), Drive defined	Latchereignis: im Antrieb definiertes Hardware Latch Ereignis mit positiver Flanke (z. B. für SoftDrive)
7	Hardware Latch 1 (neg. edge), Drive defined	Latchereignis: im Antrieb definiertes Hardware Latch Ereignis mit negativer Flanke (z. B. für SoftDrive)
20	Application (PLC code)	Anwenderspezifische Implementierung der Referenzierung (SPS Code): Anwender Anforderung wird der SPS mittels des ApplicationRequest-Bits signalisiert

Encodertypen	: Latchereignis					
	0: nicht definiert	1: SPS Nocke (neg. Flanke)	2: Hardware Syncimpuls (Null-/C-Spur)	3: Externes Hardware Latch mit pos. Flanke	4: Externes Hardware Latch mit neg. Flanke	5: Software Syncimpuls (Software Nullspur)
AX2xxx-B200 (Lightbus)	—	√	√	√	√	√ (nur Resolver)
AX2xxx-B510 (CANopen)	—	√	—	—	—	√ (nur Resolver) (s. Param. "Referenz Maske")
AX2xxx-B1x0 (EtherCAT)	—	√	√	√	√	√ (nur Resolver) (fest 20 Bit)
AX2xxx-B900 (Ethernet)	—	√	√	√	√	√ (nur Resolver)
Sercos	—	√	√	√ (AX5xxx spezifisch implementiert)	√	√ (s. Param. "Referenz Maske")
Profi Drive	—	√	√	√	√	√
KL5101 IP5109	—	√	√	√	√	√
KL5111	—	√	√	—	—	√
KL5151	—	√	√	√	√	√ (nicht sinnvoll)
IP5209	—	√	√	—	—	√ (nicht sinnvoll)
CANopen (z. B. Lenze)	—	√	—	√ (Eingang E1)	√ (Eingang E2)	√ (nur Resolver) (fest 16 Bit)
übrige Typen	—	—	—	—	—	—

## Enum Homing Sensor Source

Der Parameter legt die Quelle des Digitaleingangs der Referenziernocke (Homing Sensor) fest. Gleichzeitig wird festgelegt, ob das Signal High- oder Low-aktiv ist.

Define	Parametertext	Homing Sensor Source
0	Default: PLC cam (MC_Home)	Referenziernocke wird von der SPS bereitgestellt. Eingang bCalibrationCam des Funktionsbausteins MC_Home.
1	Digital Input 1 (Active High), device dependent mapping	Drive->Inputs->nState8.bit0 or E1 of MDP703/733 device e.g. 7031,7041,7201,7411
2	Digital Input 2 (Active High), device dependent mapping	Drive->Inputs->nState8.bit1 or E2 of MDP703/733 device e.g. L7031,7041,7201,7411
3	Digital Input 3 (Active High)	Drive->Inputs->nState8.bit2
4	Digital Input 4 (Active High)	Drive->Inputs->nState8.bit3
5	Digital Input 5 (Active High)	Drive->Inputs->nState8.bit4
6	Digital Input 6 (Active High)	Drive->Inputs->nState8.bit5
7	Digital Input 7 (Active High)	Drive->Inputs->nState8.bit6
8	Digital Input 8 (Active High)	Drive->Inputs->nState8.bit7
9	Digital Input 1 (Active Low), device dependent mapping	Drive->Inputs->nState8.bit2
10	Digital Input 2 (Active Low), device dependent mapping	Drive->Inputs->nState8.bit0 or E1 of MDP703/733 device e.g. L7031,7041,7201,7411
11	Digital Input 3 (Active Low)	Drive->Inputs->nState8.bit1 or E2 of MDP703/733 device e.g. L7031,7041,7201,7411
12	Digital Input 4 (Active Low)	Drive->Inputs->nState8.bit2
13	Digital Input 5 (Active Low)	Drive->Inputs->nState8.bit3
14	Digital Input 6 (Active Low)	Drive->Inputs->nState8.bit4
15	Digital Input 7 (Active Low)	Drive->Inputs->nState8.bit5
16	Digital Input 8 (Active Low)	Drive->Inputs->nState8.bit6

**Digital Input [1-8]**

Es wird ein mit dem NC-Prozess verlinkter digitaler Eingang verwendet. Dazu ist ein allgemeines Drive Status Byte mit 8 digitalen Eingängen im Prozessabbild definiert (Drive->Inputs->nState8), das als Signalquelle für den Homing-Sensor dienen kann. Ein zu verwendender digitaler Eingang muss also von Hand an die gewünschte Position in diesem Byte gemappt werden.



Die digitalen Eingänge 1 und 2 können je nach verwendeter Hardware hiervon abweichen. Für die Hardware MDP703/733 (z.B. EL7031, EL7041, EL7201, EL7411) werden stattdessen die direkten digitalen Eingänge E1 und E2 der Klemme verwendet, die sich im Byte Drive.nState2 der Klemme an Bitposition 3 (E1) und 4 (E2) befinden. Die unteren beiden Bits des Drive.nState8 sind in diesem Fall nicht belegt.

## Enum Drive-Typen

Define	Drive-Typen
0	NICHT DEFINIERT
1	Analog Servo Drive: M2400 DAC 1 (Analog)
2	Analog Servo Drive: M2400 DAC 2 (Analog)
3	Analog Servo Drive: M2400 DAC 3 (Analog)
4	Analog Servo Drive: M2400 DAC 4 (Analog)
5	MDP 252 Drive: Analog Servo Drive: KL4xxx, KL2502-30K (Analog)
6	MDP 252 Drive: Analog Servo Drive (Nichtlinear): KL4xxx, KL2502-30K (Analog)
7	Eil/Schleich-Drive (Digital)
8	Schrittmotor-Drive (Digital)
9	SERCOS-Drive (Digital)
10	MDP 510 Drive: KL5051 (BiSSI-Interface) (Digital)
11	AX2000 (Lightbus) (Digital)
12	Provi-Drive MC (Simodrive 611U) (Digital)
13	Universal Drive (Analog)
14	NC Rückwand (Analog)
15	spezieller CANopen Typ (z. B. Lenze Drive 9300) (Digital)
16	MDP 742 (DS402): CANopen und EtherCAT CoE (AX2xx-B1x0/B510) (Digital)
17	AX2xx-B900 Drive (Ethernet) (Digital)
20	KL2531/KL2541 Encoder (Stepper Motor) (Digital)
21	KL2532/KL2542 Encoder (DC Motor), KL2535/KL2545 Encoder (PWM Stromklemme) (Digital)
22	TCom Drive (Soft Drive) (Digital)
23	MDP 733 Drive: Profile MDP 733 (EL7332, EL7342, EP7342) (Digital)
24	MDP 703 Drive: Profile MDP 703 (EL7031, EL7041, EP7041) (Digital)

## Enum Drive-Output-Starttypen

Define	Drive-Output-Starttypen
0	NICHT DEFINIERT
1	Ausgabe als Prozentwert
2	Ausgabe als Geschwindigkeit z. B. m/min



**Enum Drive Operation Mode**

Define	Drive Operation Mode (generische antriebsunabhängige Betriebsarten)
0	DEFAULT Mode (reactivates the NC default operation mode if mode is known)
1 (standard type)	torque control
2 (standard type)	velocity control with feedback 1
3 (standard type)	velocity control with feedback 2
4 (standard type)	position control with feedback 1 (lag less)
5 (standard type)	position control with feedback 2 (lag less)
6 (CANopen/CoE specific)	torque control with commutation angle
17 (oversampling type)	torque control using dynamic container
18 (oversampling type)	velocity control with feedback 1 using dynamic container
19 (oversampling type)	velocity control with feedback 2 using dynamic container
20 (oversampling type)	position control with feedback 1 (lag less) using dynamic container
21 (oversampling type)	position control with feedback 2 (lag less) using dynamic container
38 (CANopen/CoE specific)	IO drive controlled homing mode (for third party devices)
100 (Sercos/SoE specific)	Sercos/SoE primary operation mode 0 (s. S-0-0032)
101 (Sercos/SoE specific)	Sercos/SoE secondary operation mode 1 (s. S-0-0033)
102 (Sercos/SoE specific)	Sercos/SoE secondary operation mode 2 (s. S-0-0034)
103 (Sercos/SoE specific)	Sercos/SoE secondary operation mode 3 (s. S-0-0035)
104 (Sercos/SoE specific)	Sercos/SoE secondary operation mode 4 (s. S-0-0284)
105 (Sercos/SoE specific)	Sercos/SoE secondary operation mode 5 (s. S-0-0285)
106 (Sercos/SoE specific)	Sercos/SoE secondary operation mode 6 (s. S-0-0286)
107 (Sercos/SoE specific)	Sercos/SoE secondary operation mode 7 (s. S-0-0287)

**Enum Fahrphasen/Bewegungszustand für Masterachsen**

Define	Fahrphasen/Bewegungszustand (unterschieden nach interner und externer Sollwertgenerierung)
Interne Sollwertgenerierung:	
0	Sollwertgenerator nicht aktiv (INACTIVE)
1	Sollwertgenerator aktiv (RUNNING)
2	Geschwindigkeitsoverride ist Null (OVERRIDE_ZERO)
3	Konstante Geschwindigkeit (PHASE_VELOCONST)
4	Beschleunigungsphase (PHASE_ACCPOS)
5	Verzögerungsphase (PHASE_ACCNEG)
Externe Sollwertgenerierung:	
41	Externe Sollwertgenerierung aktiv (EXTSETGEN_MODE1)
42	Interne und externe Sollwertgenerierung aktiv (EXTSETGEN_MODE2)

**Enum Fahrphasen/Bewegungszustand für Slaveachsen**

Define	Fahrphasen / Bewegungszustand
0	Slavegenerator nicht aktiv (INACTIVE)
11	Slave befindet sich in einer Bewegungs-Vorphase (PREPHASE)
12	Slave ist am Aufsynchronisieren (SYNCHRONIZING)
13	Slave ist aufsynchronisiert und fährt synchron (SYNCHRON)



Vorerst nur für Slaves vom Typ Synchronisierungsgenerator

**Enum Tabellen-Haupttypen**

Define	Tabellen-Haupttypen
1	(n*m) Kurvenscheiben Tabellen (Camming)
10	n*m Kennlinien Tabellen (Characteristics) (z. B. Hydraulik Ventilkennlinien) Es werden nur nichtzyklische Tabellen-Untertypen (1, 3) unterstützt!
16	n*m 'Motion Function' Tabellen (MF) es werden nur nichtäquidistante Tabellen-Untertypen (3, 4) unterstützt!

**Enum Tabellen-Untertypen**

Define	Tabellen-Untertypen
1	n*m Tabelle mit äquidistanten Masterpositionen und keiner zyklischen Fortsetzung des Masterprofils (äquidistant linear)
2	(n*m) Tabelle mit äquidistanten Masterpositionen und einer zyklischen Fortsetzung des Masterprofils (äquidistant zyklisch)
3	n*m) Tabelle mit nicht äquidistanten aber streng monoton steigenden Masterpositionen und einer nicht zyklischen Fortsetzung des Masterprofils (monoton linear)
4	(n*m) Tabelle mit nicht äquidistanten aber streng monoton steigenden Masterpositionen und einer zyklischen Fortsetzung des Masterprofils (monoton zyklisch)

**Enum Tabellen-Interpolationstypen**

Define	Tabellen-Interpolationstyp zwischen den Stützstellen
0	Linear-Interpolation (NC_INTERPOLATIONTYPE_LINEAR) (Standard)
1	4-Punkt-Interpolation (NC_INTERPOLATIONTYPE_4POINT) (nur für äquidistante Tabellentypen)
2	kubische Spline-Interpolation über alle Stützstellen ("globaler Spline") (NC_INTERPOLATIONTYPE_SPLINE)
3	gleitende kubische Spline-Interpolation über n Stützstellen ("lokaler Spline") (NC_INTERPOLATIONTYPE_SLIDINGSPLINE)

**Enum Tabellen-Betriebsart (Operation mode)**

Define	Tabellen-Betriebsart zum Hinzufügen, Austausch und Entfernen von Tabellen
0	(Default/Standard)
1	Additive – Hinzufügen einer weiteren Tabelle
2	Exchange – Austausch einer vorhandenen Tabelle gegen eine neue Tabelle
3	Remove – Entfernen einer vorhandenen Tabelle

**Struktur der Tabellen-(CAM)-Kopplungsinformationen**

Tabellen		(CAM) Kopplungsinformationen
nTableID;	1.	cam table ID
nTableMainType;	2.	e.g. CAMMING, CHARACTERISTIC, MOTIONFUNCTION
nTableSubType;	3.	e.g. EQUIDIST_LINEAR, EQUIDIST_CYCLE, NONEQUIDIST_LINEAR, NONEQUIDIST_CYCLE
nInterpolationType;	4.	e.g. LINEAR, 4POINT, SPLINE
nNumberOfRows;	5.	number of rows/elements
nNumberOfColumns;	6.	number of columns
fMasterCamStartPos	7.	master camming start position (first point in tabular)
fSlaveCamStartPos	8.	slave camming start position (first point in tabular)
fRawMasterPeriod;	9.	master period/cycle (raw value, not scaled)
fRawSlaveStroke;	10.	slave difference per master period/cycle (raw value, not scaled)
fMasterAxisCouplingPos	11.	total absolute master offset of cam origin when slave has been coupled
fSlaveAxisCouplingPos	12.	total absolute slave offset of cam origin when slave has been coupled
nMasterAbsolute	13.	master absolute position (0/1)
nSlaveAbsolute	14.	slave absolute position (0/1)
fMasterOffset;	15.	total master offset
fSlaveOffset;	16.	total slave offset
fMasterScaling;	17.	total master scaling
fSlaveScaling;	18.	total slave scaling
fSumOfSlaveStrokes	19.	sum of the slave strokes up to "fActualMasterAxisPos"
fSumOfSuperpositionDistance	20.	sum of superposition distance (position compensation offset)
fActualMasterAxisPos;	21.	actual master axis setpos (absolute)
fActualSlaveAxisPos;	22.	actual slave axis setpos (absolute)
fActualMasterCamPos;	23.	actual master cam setpos
fActualSlaveCamPos;	24.	actual master cam setpos
nSlaveStateDWord	25.	slave state DWORD (s. AxisRef)
...	...	...

## Struktur der charakteristischen Kennwerte

Charakteristische Kennwerte		
fMasterVeloNom;	1.	master nominal velocity (normed: => 1.0)
fMasterPosStart;	2.	master start position
fSlavePosStart;	3.	slave start position
fSlaveVeloStart;	4.	slave start velocity
fSlaveAccStart;	5.	slave start acceleration
fSlaveJerkStart;	6.	slave start jerk
fMasterPosEnd;	7.	master end position
fSlavePosEnd;	8.	slave end position
fSlaveVeloEnd;	9.	slave end velocity
fSlaveAccEnd;	10.	slave end acceleration
fSlaveJerkEnd;	11.	slave end jerk
fMPosAtSPosMin;	12.	master pos. at slave min. position
fSlavePosMin;	13.	slave minimum position
fMPosAtSVeloMin;	14.	master pos. at slave min. velocity
fSlaveVeloMin;	15.	slave minimum velocity
fMPosAtSAccMin;	16.	master pos. at slave min. acceleration
fSlaveAccMin;	17.	slave minimum acceleration
fSVeloAtSAccMin;	18.	slave velocity at slave min. acceleration
fSlaveJerkMin;	19.	slave minimum jerk
fSlaveDynMomMin;	20.	slave minimum dynamic momentum (NOT SUPPORTED YET!)
fMPosAtSPosMax;	21.	master pos. at slave max. position
fSlavePosMax;	22.	slave maximum position
fMPosAtSVeloMax;	23.	master pos. at slave max. velocity
fSlaveVeloMax;	24.	slave maximum velocity
fMPosAtSAccMax;	25.	master pos. at slave max. acceleration
fSlaveAccMax;	26.	slave maximum acceleration
fSVeloAtSAccMax;	27.	slave velocity at slave max. acceleration
fSlaveJerkMax;	28.	slave maximum jerk
fSlaveDynMomMax;	29.	slave maximum dynamic momentum (NOT SUPPORTED YET!)
fSlaveVeloMean;	30.	slave mean absolute velocity
fSlaveAccEff;	31.	slave effective acceleration
nCamTableID;	32.	Cam table ID
nNumberOfRows;	33.	Number of rows/entries e.g. number of points
nNumberOfColumns;	34.	Number of columns (typically 1 or 2)
nCamTableType;	35.	cam table type (10=EQUIDIST, 11=NONEQUIDIST, 22=MOTIONFUNC, 23=CHARACTERISTIC)
nPeriodic;	36.	linear or cyclic/periodic
nReserved	37.	reserved

## Enum Achsregelkreis-Umschalttypen

Define	Achsregelkreis-Umschalttypen
0	NICHT DEFINIERT
1	Einfaches Umschalten (ähnlich einem Achsreset) (STANDARD)
2	Umschalten/Aufsynchronisieren mittels I/D-Anteil des Reglers auf einen internen Initialwert (ruckfrei/stoßfrei)
3	Umschalten/Aufsynchronisieren mittels I/D-Anteil des Reglers auf einen parametrierbaren Initialwert

### 3.1.2 AmsNAT

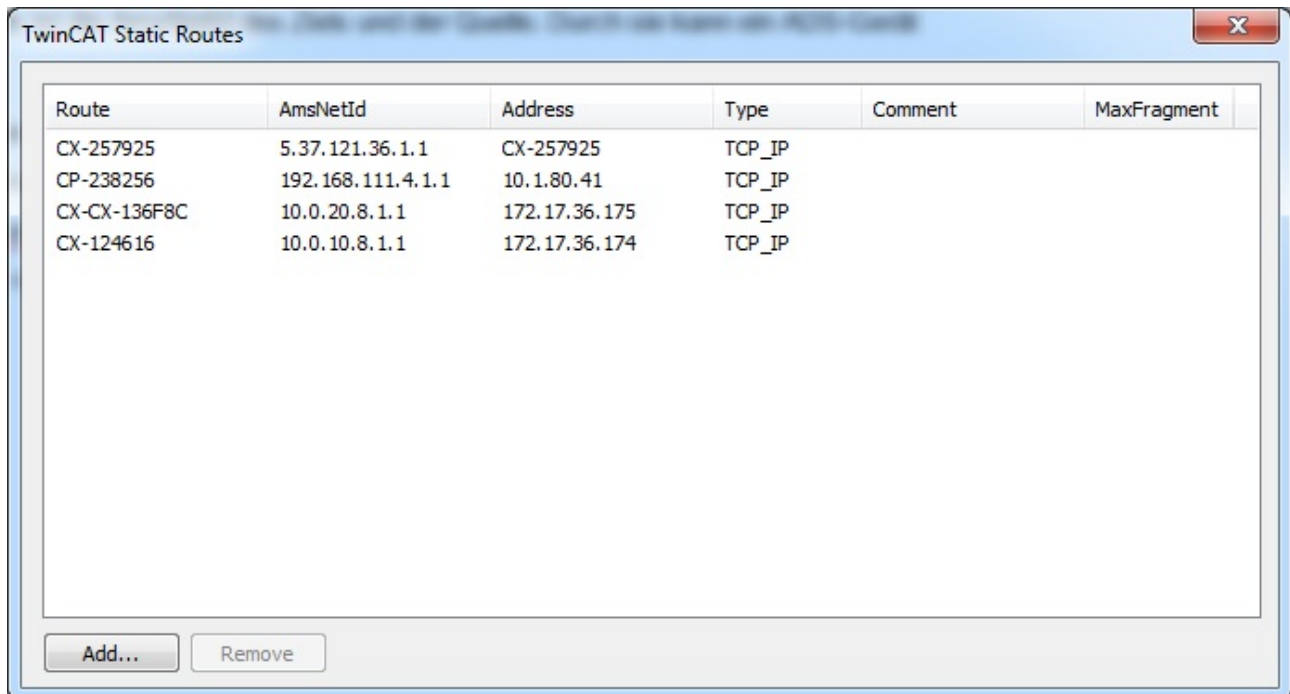
#### 3.1.2.1 Einführung

Um die AmsNAT-Funktion besser zu verstehen, ist es wichtig, den Unterschied zwischen ADS und AMS zu kennen und zu wissen, was eine ADS-Route ist.

**ADS** (Automation Device Specification) ist das Kommunikationsprotokoll von TwinCAT, das die Interaktion zwischen zwei ADS-Teilnehmern spezifiziert. Es legt zum Beispiel fest, welche Operationen auf einem anderen ADS-Gerät ausgeführt werden können, welche Parameter dazu erforderlich sind und welcher Rückgabewert nach Ausführung gesendet wird.

**AMS** (Automation Message Specification) spezifiziert den Austausch der ADS-Daten. Wesentlicher Bestandteil des Kommunikationsprotokolls ist die AmsNetId. Diese wird in einem AMS/ADS-Paket für das Quell- und Zielgerät angegeben. Anhand der AmsNetId kann ein ADS-Gerät eindeutig adressiert werden.

Damit zwei Teilnehmer kommunizieren können, muss in TwinCAT eine **Route** zwischen diesen eingerichtet sein. Diese wird auf beiden Seiten konfiguriert und enthält typischerweise den Routennamen, die AmsNetId und die Adresse des Kommunikationspartners sowie den Verbindungstyp. Die folgende Abbildung zeigt die Konfiguration neuer Routen und eine Übersicht bestehender Routen eines TwinCAT-Systems.



Falls auf dem Ziel ein Scan der Hardware vorgenommen werden soll, so müssen relative NetIds verwendet werden:

Current Routes Static Routes Project Routes NetId Management

Local NetId:  Change

Target NetId:

Project NetIds:

NetId	Owner	Type
-------	-------	------

Use Relative NetIds Change Project NetId

### 3.1.2.2 Allgemeine Beschreibung

Die AmsNAT-Funktion ermöglicht XAE-Systemen das Herstellen von Routen zu zwei oder mehreren Steuerungen, welche die gleiche AmsNetId besitzen (Abbildung 2). Darüber hinaus bietet AmsNAT eine Lösung, mit der verschiedene ADS-Geräte mit gleicher AmsNetId miteinander per ADS kommunizieren können. Bei AmsNAT werden virtuelle AmsNetIds verwendet. Eine virtuelle AmsNetId ist eine eindeutige Adresse für ein verbundenes ADS-Gerät, die bei der Kommunikation durch die reale AmsNetId des Zielsystems ersetzt wird. Das heißt, bei jeder über ADS geführten Kommunikation sorgt die AmsNAT-Funktion dafür, dass die AmsNetId des Zielsystems ausgetauscht wird.

XAE

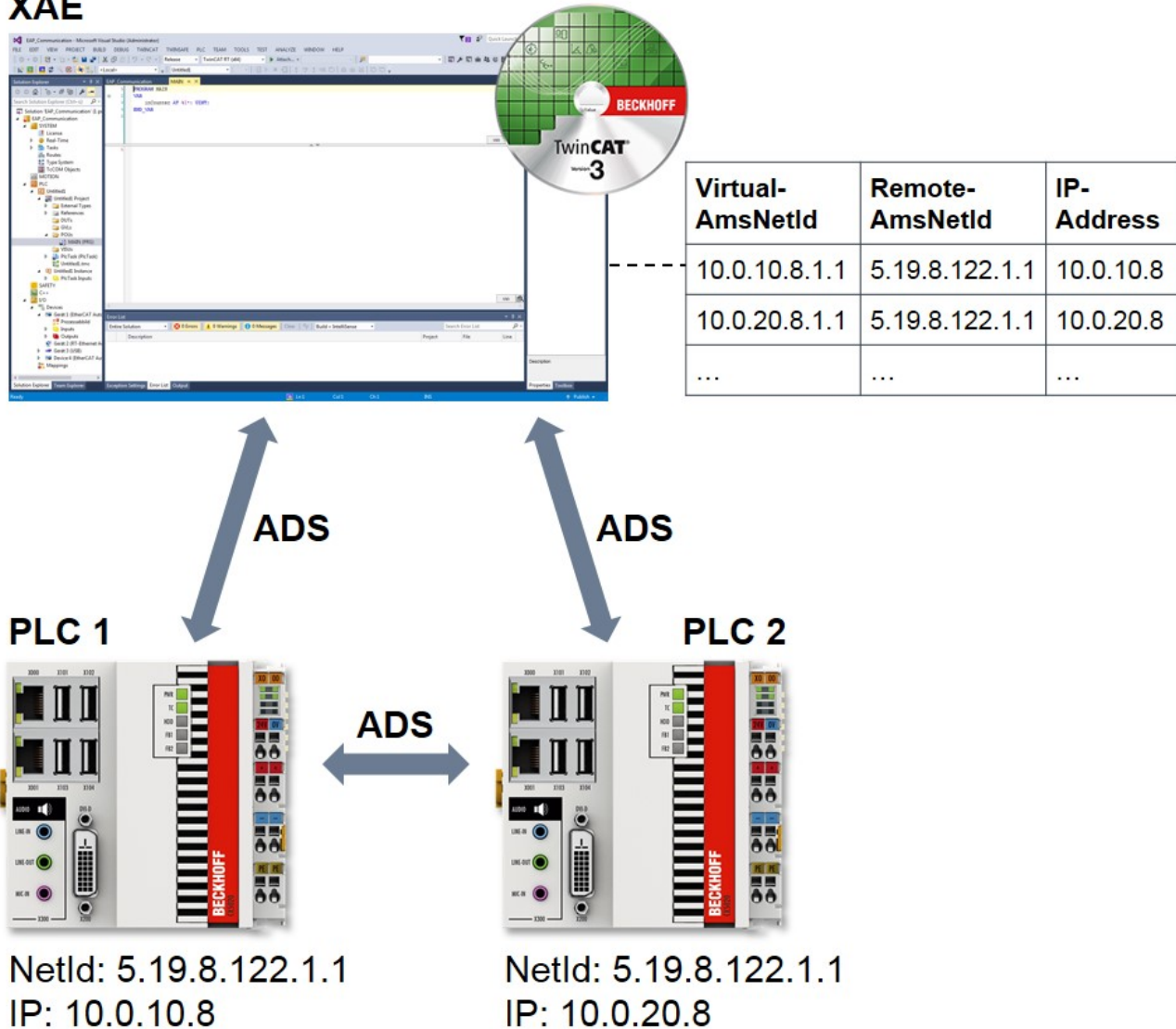


Abb. 1: Kommunikation mit/zwischen TwinCAT-Systemen mit gleicher NetId

3.1.2.3 Motivation

Ein häufig auftretender Anwendungsfall im Serien-Maschinenbau ist das Klonen (also ein vollständiges 1:1 Abbild) einer Steuerung. Bei dem Einsatz von TwinCAT resultiert daraus, dass alle geklonten Instanzen die gleiche AmsNetId besitzen. Dies ist erst einmal nicht problematisch. Wenn die geklonten Instanzen jedoch mit demselben Engineering-System parallel verbunden sein sollen oder untereinander per ADS kommunizieren sollen, ist dies zunächst einmal nicht möglich, da die AmsNetId nicht eindeutig ist. Mit der AmsNAT-Funktion wird genau diese Einschränkung aufgehoben, indem die Systeme mit virtuellen AmsNetIds arbeiten. Diese können mit sehr geringem Aufwand konfiguriert werden.

Die AmsNAT-Funktion kann für jede beliebige Route zu einem ADS-Gerät eingesetzt werden. Damit wird ein hoher Grad an Flexibilität bereitgestellt und es müssen nicht länger die AmsNetIds auf den Maschinenrechnern angepasst werden, was zu einer deutlichen Reduzierung von Konfigurationsaufwänden führt.

3.1.2.4 Funktionsweise

Die Funktionsweise von AmsNAT soll anhand eines typischen Anwendungsfalls erläutert werden. In dem Anwendungsfall existieren ein TwinCAT-Engineering-System und zwei TwinCAT-Runtimes mit gleicher AmsNetId und IP-Adresse. Die Konfiguration ist in Abbildung 3 dargestellt. Von dem Engineering-System soll ein AdsRead-Befehl an die PLC 1 gesendet werden, von der eine entsprechende Antwort erwartet wird.



Da beide Runtimes eine identische IP-Adresse besitzen, werden zusätzlich zwei IP NATs verwendet. Ihre Aufgabe ist es, eine eindeutige Adressierung zu realisieren. Dazu werden je nach Kommunikationsrichtung die ersten drei Stellen der lokalen/globalen IP-Adresse durch die ersten drei Stellen der globalen/globalen IP-Adresse ersetzt.

Im ersten Schritt des Anwendungsbeispiels sendet das Engineering-System einen AdsRead-Befehl an die PLC 1. Da es sich bei dieser AmsNetId um eine virtuelle handelt, ersetzt der TwinCAT-System-Service mithilfe seiner Routing-Tabelle diese durch die Remote-AmsNetId 5.19.8.122.1.1. Dies ist die real auf dem System vorhandene AmsNetId. Sie wird in das Feld "AmsNetId Target" des AMS-Paketes eingetragen.

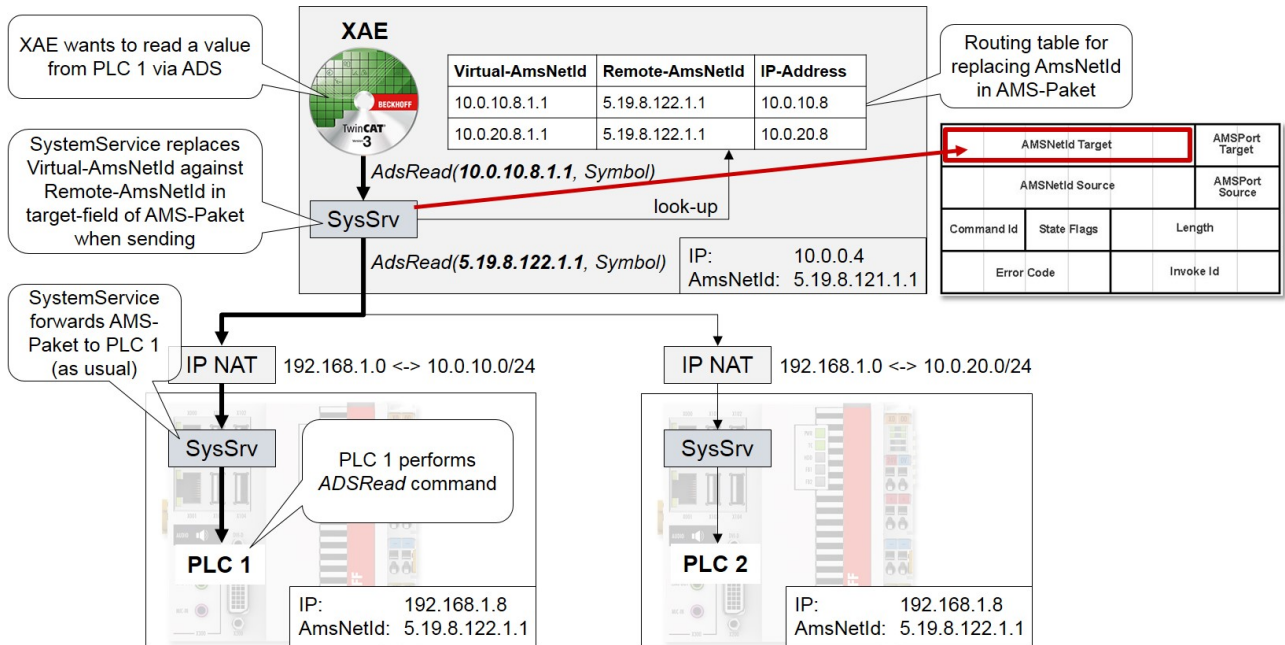


Abb. 2: Ablauf für das Senden eines AdsRead-Befehls unter Verwendung von AmsNAT

Der TwinCAT-System-Service von PLC 1 leitet das AMS-Paket unverändert weiter. Die PLC 1 führt den AdsRead-Befehl aus und sendet anschließend die entsprechende Rückantwort an das Engineering-System. Den Ablauf der Kommunikation für die Rückantwort zeigt Abbildung 4.

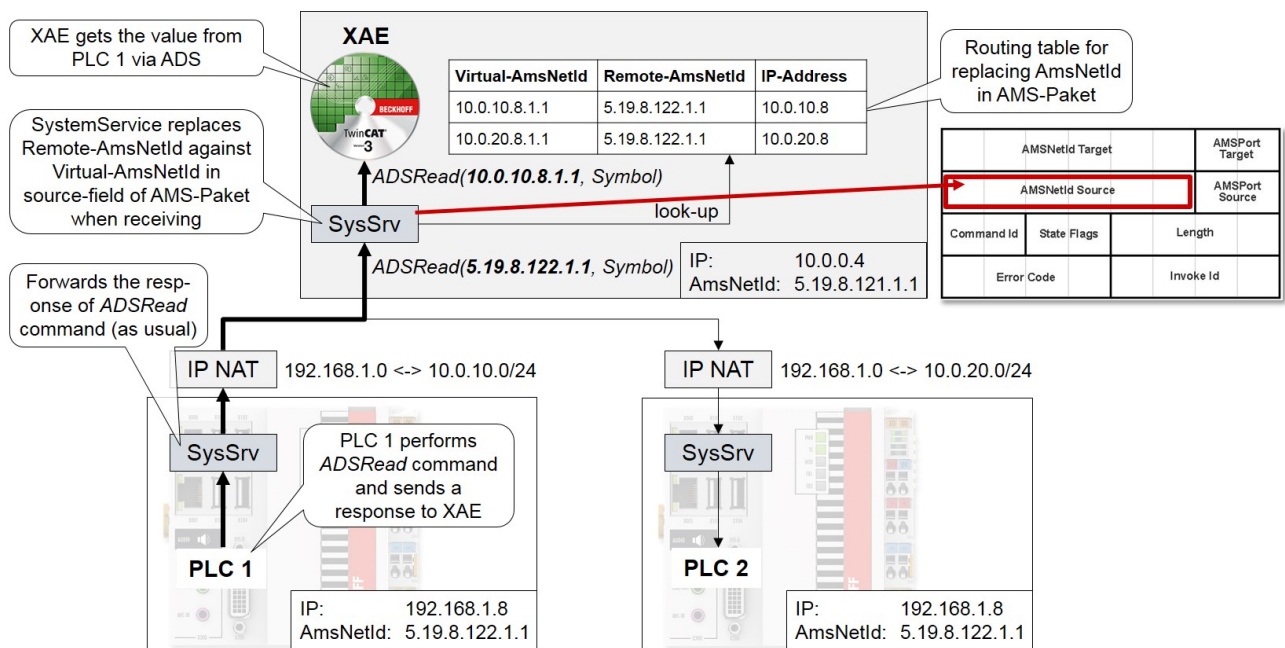


Abb. 3: Ablauf für das Senden der Rückantwort zum AdsRead-Befehl



Bei der Rückantwort leitet zunächst der TwinCAT-System-Service von PLC 1 das AMS-Paket unverändert weiter. Anschließend erreicht es den TwinCAT-System-Service vom Engineering System. Da in dem Feld "AmsNetId Source" des AMS-Pakets die reale AmsNetId von PLC 1 eingetragen ist, muss diese anhand der Routing-Tabelle durch die virtuelle AmsNetId ersetzt werden. Danach kann das Engineering-System die Rückantwort eindeutig zuordnen und verarbeiten.

Bei Verwendung der AmsNAT-Funktion werden die gesendeten Daten nicht verändert, lediglich der AMS-Header. Daher ist darauf zu achten, dass wenn Konfigurationsdaten die AmsNetId enthalten, dies dazu führen kann, dass die virtuelle AmsNetId verwendet wird. Eine Möglichkeit für das Engineering von I/O Devices ist die Verwendung von relativen AmsNetIds. Hier werden ausschließlich die letzten beiden Stellen der AmsNetId berücksichtigt und die ersten vier Stellen werden nicht betrachtet.

### 3.1.2.5 Konfiguration

Zur Konfiguration von AmsNAT öffnen Sie die Datei *StaticRoutes.xml*, die sich im Installationsverzeichnis von TwinCAT unter dem Pfad *TwinCAT\3.1\Target* befindet. Definieren Sie in der Datei für jede Route das Attribut "RemoteNetId", wie nachfolgend zu sehen ist:

```
<?xml version="1.0" encoding="UTF-8"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
  <RemoteConnections>
    <Route>
      <Name>CX-111111</Name>
      <Address>10.0.10.8</Address>
      <NetId RemoteNetId="5.19.8.122.1.1">10.0.10.8.1.1</NetId>
      <Type>TCP_IP</Type>
    </Route>
    <Route>
      <Name>CX-222222</Name>
      <Address>10.0.20.8</Address>
      <NetId RemoteNetId="5.19.8.122.1.1">10.0.20.8.1.1</NetId>
      <Type>TCP_IP</Type>
    </Route>
  </RemoteConnections>
</TcConfig>
```

Mit dem Attribut "RemoteNetId" wird die tatsächlich auf dem entfernten ADS-Gerät vergebene AmsNetId angegeben. Diese muss nicht eindeutig sein. In dem TwinCAT-System mit konfigurierter AmsNAT-Funktion ist allein die im Feld <NetId> definierte AmsNetId des Zielsystems bekannt.

Um die vorgenommene Konfiguration der AmsNAT-Funktion zu aktivieren, starten Sie den TwinCAT-System-Service neu. Schalten Sie dazu das TwinCAT-System vom Run-Modus in den Konfig-Modus. Befindet sich TwinCAT bereits im Konfig-Modus, öffnen Sie diesen erneut, um die vorgenommenen Einstellungen zu laden.

## 3.1.3 ADS-over-MQTT

### 3.1.3.1 Allgemeine Beschreibung

„ADS over MQTT“ ist aus Sicht des ADS Protokolls ein neuer Transportkanal. Es werden also exakt die gleichen ADS Kommandos über MQTT übertragen, wie auch über andere Kommunikationsprotokolle.

Hierzu wird vom TwinCAT-Router eine Verbindung zum Broker aufgebaut, um Kommandos des ADS Protokolls sowohl abzusenden, wie auch zu empfangen.

Am lokalen Gerät wird damit also der Endpunkt des Brokers konfiguriert. Hieraus ergibt sich, dass sich die 1:1 Beziehung einer ADS-Route erst im Zusammenspiel mit der passenden Broker Konfiguration ergibt.



Diese Dokumentation gibt sowohl einen Überblick über die Möglichkeiten des Einsatzes, wie auch die technische Beschreibung wie ein „virtuelles ADS-Netzwerk“ über einen MQTT Message Broker konfiguriert werden kann.

### Vorteile eines MQTT-basierten ADS-Netzwerks

- **Subnetze, NAT-basierte Netzwerke und Firewalls:**  
In einem klassischen ADS-Aufbau werden eingehende TCP/IP-Verbindungen in beide Richtungen verwendet. Dieses bedingt, dass die Geräte im Normalfall im gleichen Netzwerk stehen müssen. In verteilten Anlagen mit unterschiedlichen Subnetzen ergeben sich aufwändige Konfigurationen, um die entsprechenden ADS-Routen nutzbar zu machen.  
Bei MQTT-basierten ADS-Netzwerken wird nur eine ausgehende TCP/IP-Verbindung von den Geräten verwendet. Hierdurch kann der Broker im überlagerten Netz zwischen allen Teilnehmern vermitteln. Durch die ausgehenden Verbindungen kann eine typische Firewall verwendet werden und es müssen keine eingehenden Ports hinterlegt werden.
- **Zugriffskontrolle:**  
In einem klassischen ADS-Aufbau kann, nach Anlegen der entsprechenden Routen, eine bidirektionale Kommunikation durchgeführt werden.  
Ein Zugriff von Teilnehmer A der auf B zugreift, erlaubt also auch, dass Teilnehmer B auf A zugreifen kann.  
Im MQTT-basierten ADS-Netzwerk kann konfiguriert werden, dass ein Teilnehmer A auf B zugreifen kann, jedoch nicht andersrum.
- **Security / Verschlüsselung:**  
Die Kommunikation von TwinCAT zu dem Broker kann durch TLS (mit Zertifikaten oder PreSharedKey (PSK)) verschlüsselt werden.

Nachteilig ist der erhöhte administrative Aufwand anzusehen. Dieser wird jedoch pro Teilnehmer bei einem größeren Netzwerk vertretbar gering ausfallen.

### HINWEIS

#### ADS Zugriff bedeutet Vollzugriff

Wie im [Security Advisory 2017-01](#) beschrieben, bietet ADS Vollzugriff auf ein Gerät. Secure ADS bietet eine Autorisierung sowie Verschlüsselung für die Kommunikation, es stellt also eine Transport-Verschlüsselung dar. Wenn eine ADS Route besteht, existiert also Vollzugriff. Dezierten, Rollen-bezogenen Zugriff auf einzelne Daten bieten Lösungen, wie z. B. OPC-UA.

### 3.1.3.2 Voraussetzungen

#### ● TwinCAT 3.1 Build 4022.0 erforderlich

**i** Das ADS-over-MQTT ist eine Erweiterung von Build 4022 und damit erst ab diesem Release verfügbar.

- ADS-over-MQTT ist Bestandteil von TC1000 und kann ohne Lizenzkosten genutzt werden.
- Die verwendeten Geräte benötigen eine ausgehende Netzwerkkommunikation zum Broker.

- Es muss ein MQTT Broker bereitgestellt werden, über den die Kommunikation erfolgen kann.
- Die bereitgestellte Erweiterung ist für den Eclipse Mosquitto Broker verfügbar.
- Zur TLS-Verschlüsselung müssen ggf. entsprechende Zertifikate generiert und signiert werden.

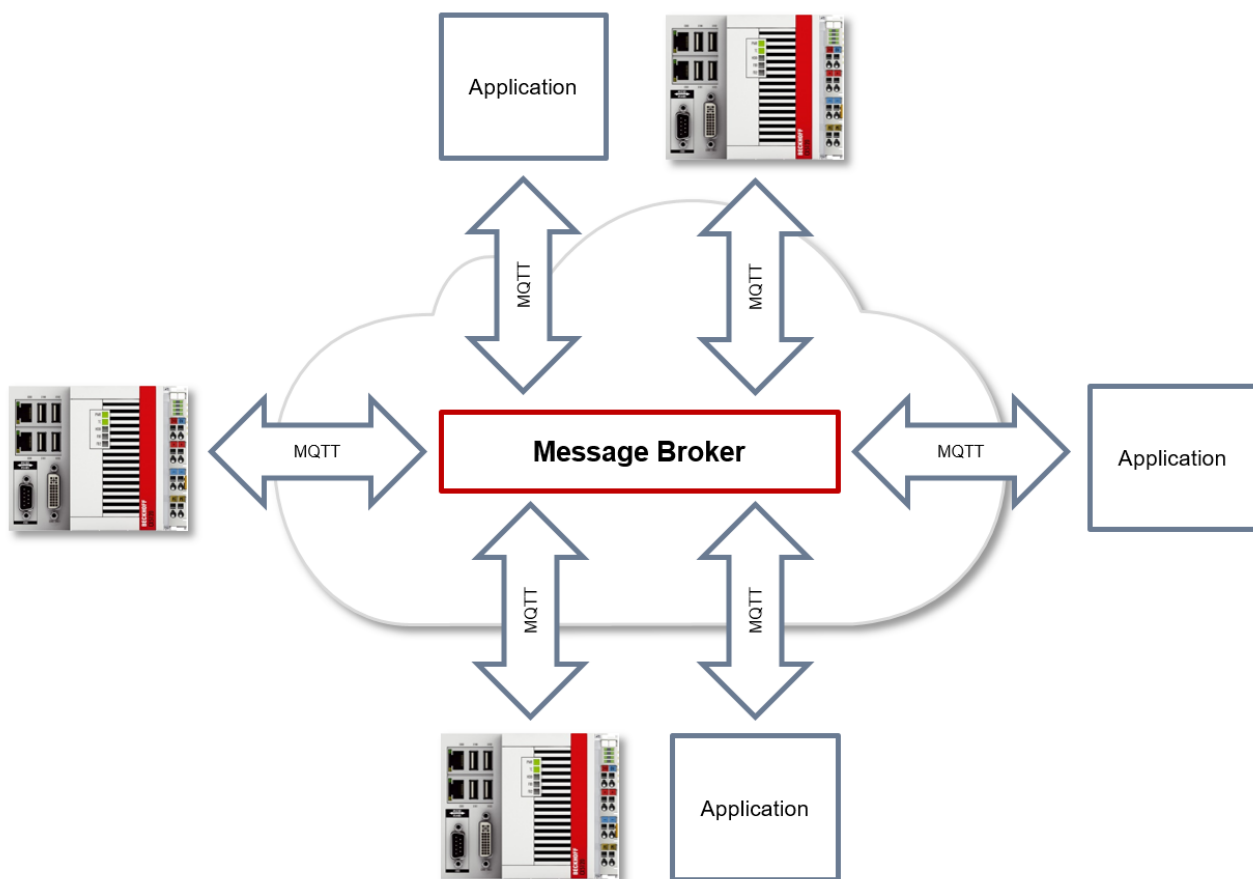
### 3.1.3.3 Technische Einführung

In diesem Abschnitt wird ein Überblick über die verwendeten Technologien sowie die grundsätzliche Architektur eines „virtuellen ADS-Netzwerks“ gegeben.

ADS-over-MQTT führt dafür einen zusätzlichen Kommunikationskanal ein, wodurch sich ADS-Routen über MQTT ergeben. Diesen können die Programme, die als ADS Devices auf den Geräten gestartet werden, nutzen, ohne dass sie modifiziert werden.

#### 3.1.3.3.1 MQTT Grundlagen

MQTT (Message Queueing Telemetry Transport) ist ein Publisher/Subscriber-basiertes Kommunikationsprotokoll, welches eine Nachrichten-basierte Übertragung zwischen Applikationen ermöglicht. Eine zentrale Komponente bei dieser Art der Übertragung ist der sogenannte Message Broker. Dieser hat die Aufgabe, Nachrichten zwischen den einzelnen Applikationen, bzw. dem Sender und Empfänger einer Nachricht, zu verteilen. Der Message Broker entkoppelt dabei Sender und Empfänger voneinander, sodass diese keine gegenseitigen Addressinformationen kennen und austauschen müssen. Alle Kommunikationsteilnehmer wenden sich beim Senden und Empfangen an den Message Broker und dieser übernimmt die Verteilung der Nachrichten.



#### ClientID

Beim Herstellen einer Verbindung mit dem Message Broker übermittelt der Client eine sogenannte ClientID, welche zur eindeutigen Identifizierung des Clients auf dem Message Broker dient. Der MQTT-Kommunikationstreiber aus TwinCAT3 erzeugt automatisch eine eigene ClientID, welche sich an dem folgenden Namensschema orientiert:

*PlcProjectName-TcMqttClient%n*

%n ist hierbei ein inkrementeller Zähler für die Nummer der jeweiligen MQTT Client Instanz. Jede Instanz des Funktionsbausteins FB\_IotMqttClient erhöht hierbei diesen Zähler. In den meisten Fällen ist das Verwenden dieses ClientID Formats ausreichend. In speziellen Fällen, z.B. abhängig vom Message Broker oder auch durch die eigene MQTT Applikation bedingt, muss eine anwendungsspezifische ClientID vergeben werden. Dies kann über einen entsprechenden Eingang an den Funktionsbausteinen FB\_IotMqttClient und FB\_IotMqtt5Client erfolgen.

Soll eine eindeutige ClientID automatisch beim Start des SPS Projekts generiert werden, so bietet sich die Verwendung einer GUID an, welche über den Funktionsbaustein FB\_CreateGuid aus der Bibliothek Tc2\_System erzeugt werden kann. Der folgende Beispielcode verdeutlicht die Verwendung dieses Funktionsbausteins.

```
PROGRAM MAIN
VAR
  fbGuid : FB_CreateGUID;
  objGuid : GUID;
  sGuid : STRING;
  nState : UINT;
  bStart : BOOL; // set to TRUE to start this sample
END_VAR

CASE nState OF
  0 :
    IF bStart THEN
      bStart := FALSE;
      nState := nState + 1;
    END_IF

  1 : // create GUID using FB_CreateGuid from Tc2_System library
    fbGuid(bExecute := TRUE, pGuidBuffer := ADR(objGuid), nGuidBufferSize := SIZEOF(objGuid));
    IF NOT fbGuid.bBusy THEN
      fbGuid(bExecute := FALSE);
      IF NOT fbGuid.bError THEN
        nState := nState + 1;
      ELSE
        nState := 255; // go to error state
      END_IF
    END_IF

  2: // GUID has been created, now convert to STRING
    sGuid := GUID_TO_STRING(objGuid);
    nState := nState + 1;

  3: // done

  255: // error state

END_CASE
```

Nach Ausführung dieser State Machine enthält die Variable sGuid die generierte GUID als STRING. Diese kann dann an den Funktionsbausteinen FB\_IotMqttClient und FB\_IotMqtt5Client als ClientID verwendet werden.

## Payload

Der Nachrichteninhalte einer MQTT-Nachricht wird als sogenannter Payload bezeichnet. Es können beliebige Daten übertragen werden, beispielsweise ein Text, ein einzelner Zahlenwert oder eine gesamte Informationsstruktur.

### Message-Payload-Formatierung

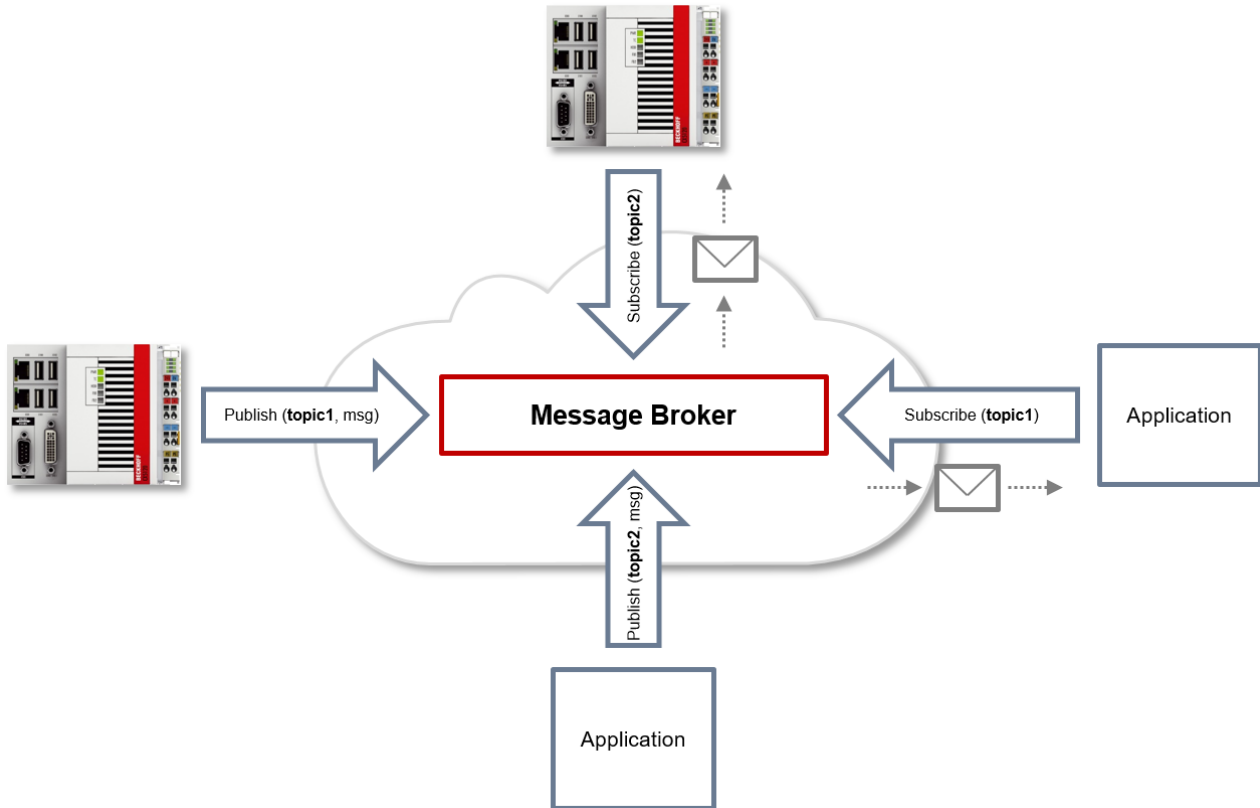
**i** Beachten Sie, dass der Datentyp und die Formatierung des Inhalts der Sender- und Empfängerseite bekannt sein müssen, insbesondere beim Versand von Binärinformationen (Alignment) oder Strings (mit/ohne Nullterminierung).

## Topics

Bei Verwendung eines Message Brokers, welcher auf dem Protokoll MQTT basiert, wird das Senden (Publish) und Abonnieren (Subscribe) von Nachrichten mithilfe sogenannter Topics organisiert. Der Message Broker filtert eingehende Nachrichten anhand dieser Topics für jeden verbundenen Client. Ein Topic kann hierbei auch aus mehreren Ebenen bestehen, wobei die Ebenen durch ein „/“ voneinander getrennt sind.

Beispiel: Campus / Building1 / Floor2 / Room3 / Temperature

Der Publisher einer Nachricht gibt beim Versand immer an, für welches Topic eine Nachricht gedacht ist. Ein Subscriber hingegen gibt an, für welches Topic er sich interessiert. Der Message Broker leitet dann die Nachricht entsprechend weiter.



Beispielkommunikation 1 in der Grafik oben:

- Eine Applikation subscribed sich auf das Topic mit dem Namen „topic1“.
- Ein Controller published eine Nachricht an das Topic mit dem Namen „topic1“.
- Der Message Broker leitet die Nachricht entsprechend an die Applikation weiter.

Beispielkommunikation 2 in der Grafik oben:

- Ein Controller subscribed sich auf das Topic mit dem Namen „topic2“.
- Eine Applikation published eine Nachricht an das Topic mit dem Namen „topic2“.
- Der Message Broker leitet die Nachricht entsprechend an den Controller weiter.

**Wildcards**

Bei der Verwendung von Topics können auch sogenannte „Wildcards“ benutzt werden. Eine Wildcard ersetzt einen Teil des Topics. Ein Subscriber erhält dann ggf. Nachrichten aus mehreren Topics. Es werden zwei Arten von Wildcards unterschieden:

- Single Level Wildcards
- Multi Level Wildcards

Beispiel „Single Level Wildcard“:

Das +-Symbol beschreibt eine Single Level Wildcard. Wird es z.B. vom Subscriber wie folgt verwendet, so werden entsprechende Nachrichten an die Topics entweder vom Subscriber empfangen oder nicht empfangen.

- Der Empfänger subscribed sich auf Campus/Building1/Floor2+/Temperature
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Temperature - OK
- Der Publisher sendet an Campus/Building1/Floor2/Room2/Temperature - OK

- Der Publisher sendet an Campus/Building42/Floor1/Room1/Temperature - NOK
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Fridge/Temperature - NOK

Beispiel „Multi Level Wildcard“:

Das #-Symbol beschreibt eine Multi Level Wildcard. Wird es z.B. vom Subscriber wie folgt verwendet, so werden entsprechende Nachrichten an die Topics entweder vom Subscriber empfangen oder nicht empfangen. Das #-Symbol muss hierbei immer als letztes Symbol im Topic-String verwendet werden.

- Der Empfänger subscribed sich auf Campus/Building1/Floor2/#
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Temperature - OK
- Der Publisher sendet an Campus/Building1/Floor2/Room2/Temperature - OK
- Der Publisher sendet an Campus/Building42/Floor1/Room1/Temperature - NOK
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Fridge/Temperature – OK
- Der Publisher sendet an Campus/Building1/Floor2/Room1/Humidity - OK

### QoS (Quality of Service)

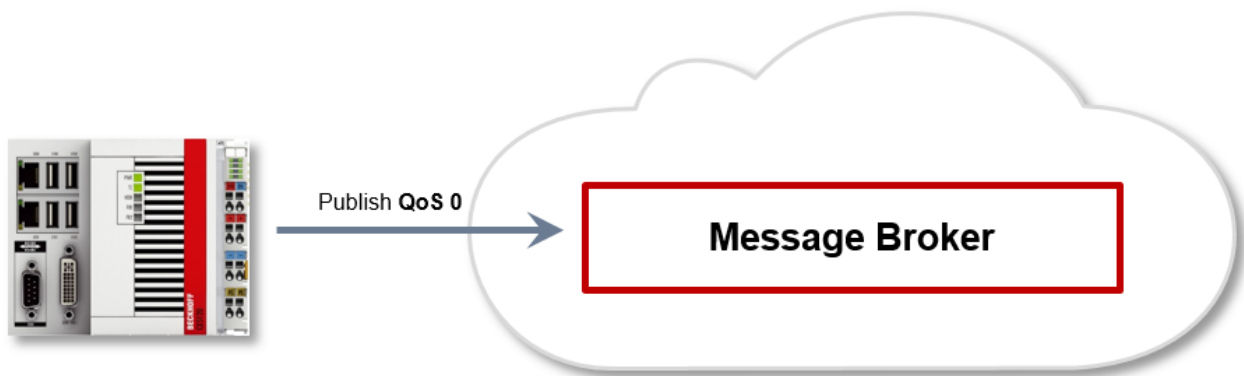
QoS ist eine Vereinbarung zwischen dem Sender und Empfänger einer Nachricht in Bezug auf das Garantieren der Nachrichtenübermittlung. Es existieren drei verschiedene Level in MQTT:

- 0 – höchstens einmal
- 1 – mindestens einmal
- 2 – genau einmal

Beide Kommunikationsarten (Publish/Subscribe) zum Message Broker müssen berücksichtigt und getrennt voneinander betrachtet werden. Das QoS-Level, welches ein Client beim Publishen einer Nachricht verwendet, wird vom jeweiligen Client gesetzt. Wenn der Broker nun die Nachricht an einen Client weiterleitet, der sich entsprechend auf das Topic subscribed hat, wird das QoS-Level vom Subscriber verwendet, welches beim Herstellen der Subscription angegeben wurde. Dies bedeutet, dass ein QoS-Level, welches vom Publisher vielleicht mit 2 angegeben wurde, vom Subscriber mit 0 „überschrieben“ werden kann.

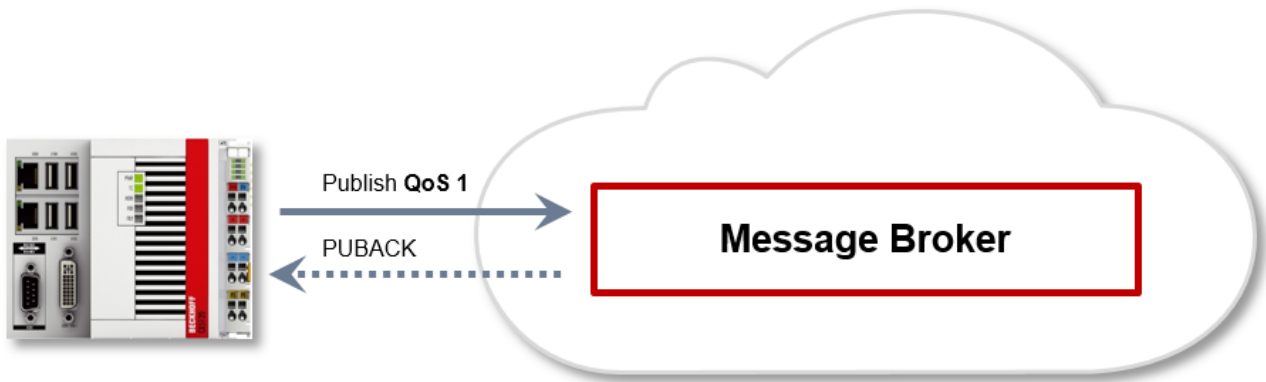
#### QoS-Level 0

Bei diesem QoS-Level erfolgt keine Bestätigung des Empfängers, ob die Nachrichten empfangen wurden oder nicht. In der Folge wird die Nachricht auch kein zweites Mal gesendet.



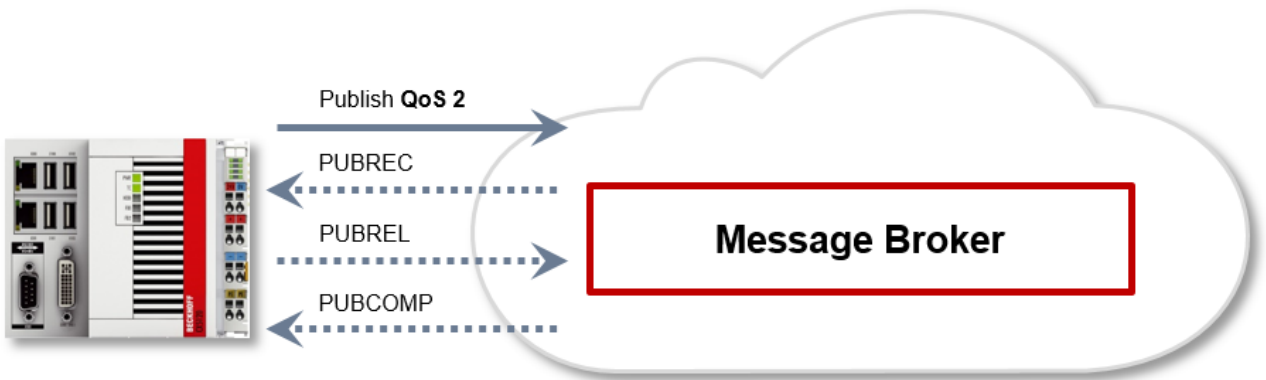
#### QoS-Level 1

Bei diesem QoS-Level wird garantiert, dass die Nachricht zumindest einmal beim Empfänger ankommt. Aber die Nachricht kann unter Umständen auch mehrfach beim Empfänger eintreffen. Der Sender speichert die Nachricht intern bis er eine Bestätigung in Form einer PUBACK-Nachricht vom Empfänger erhält. Wenn die PUBACK-Nachricht für eine bestimmte Zeit ausbleibt, wird die Nachricht erneut gesendet.



**QoS-Level 2**

Bei diesem QoS-Level wird garantiert, dass die Nachricht maximal einmal beim Empfänger ankommt. Dies wird MQTT-seitig durch einen Handshake-Mechanismus realisiert. QoS-Level 2 ist der sicherste (aus Sicht der Nachrichtenübermittlung), aber auch langsamste QoS-Level. Wenn ein Empfänger eine Nachricht mit QoS-Level 2 erhält, bestätigt er die Nachricht mit einem PUBREC. Der Absender der Nachricht merkt sich diese intern bis er ein PUBCOMP empfangen hat. Dieser zusätzliche Handshake (verglichen mit QoS 1) ist wichtig, damit die Nachricht nicht doppelt übertragen wird. Wenn der Absender der Nachricht ein PUBREC erhält, kann er den initialen Publish verwerfen, da er weiß, dass die Nachricht einmal vom Empfänger empfangen wurde. Er merkt sich somit intern den PUBREC und sendet seinerseits ein PUBREL. Nachdem der Empfänger ein PUBREL empfangen hat, kann er die sich zuvor gemerkten Zustände verwerfen und mit einem PUBCOMP antworten. Umgekehrt genauso. Immer dann, wenn ein Paket verloren geht, ist der jeweilige Kommunikationsteilnehmer dafür verantwortlich, die zuletzt gesendete Nachricht nach einer bestimmten Zeit noch einmal zu senden.



Der LastWill ist eine Nachricht, die im Falle eines irregulären Verbindungsabbruches vom Broker an alle Clients gesendet wird, die das passende Topic abonniert haben. Verliert der MQTT-Client in der SPS die Verbindung zum Broker und es wurde beim Verbindungsaufbau ein LastWill hinterlegt, so wird dieser LastWill vom Broker kommuniziert, ohne dass der Client sich darum kümmern muss.

Bei einem geplanten Disconnect wird der LastWill laut Spezifikation nicht zwingend übertragen. Aus Sicht des SPS-Programmierers kann dieser entscheiden, ob er vor Aufruf des Disconnects den LastWill publishen will. Dazu wird auf dem LastWill-Topic die LastWill-Nachricht noch einmal gepublished. Das ist notwendig, da der Broker aufgrund der regulären Verbindungsabbruches die Last Will-Nachricht nicht veröffentlichen würde.

Bei einem TwinCAT-Kontextwechsel und einem daraus folgenden Neustart der MQTT-Kommunikation sendet der IoT-Treiber den vorher spezifizierten LastWill an den Broker, weil in diesem Moment aus der SPS keine Möglichkeit mehr dazu besteht. Wenn bei Verbindungsherstellung kein LastWill definiert wurde, wird auch keine Nachricht vor dem Disconnect übertragen.

**Sicherheit**

Bei der Herstellung einer Verbindung zum Message Broker können hierbei auch Sicherheitsmechanismen, wie TLS eingesetzt werden, um die Kommunikationsverbindung zu verschlüsseln oder eine Authentifizierung zwischen Client und Message Broker zu realisieren.



## Quellen

Für weitere und detailliertere Informationen zu MQTT empfehlen wir die folgenden Webseiten:

HiveMq Blog: <http://www.hivemq.com/blog/mqtt-essentials/> (Hauptgrundlage dieses Artikels)

### 3.1.3.3.2 Architektur

Der ADS Router in jedem Gerät übernimmt die Übermittlung der ADS-Kommandos zwischen den lokalen und auch entfernten „ADS Devices“.

Dieser Router kann so konfiguriert werden, dass ADS-Kommunikation auch über einen Broker erfolgen kann.

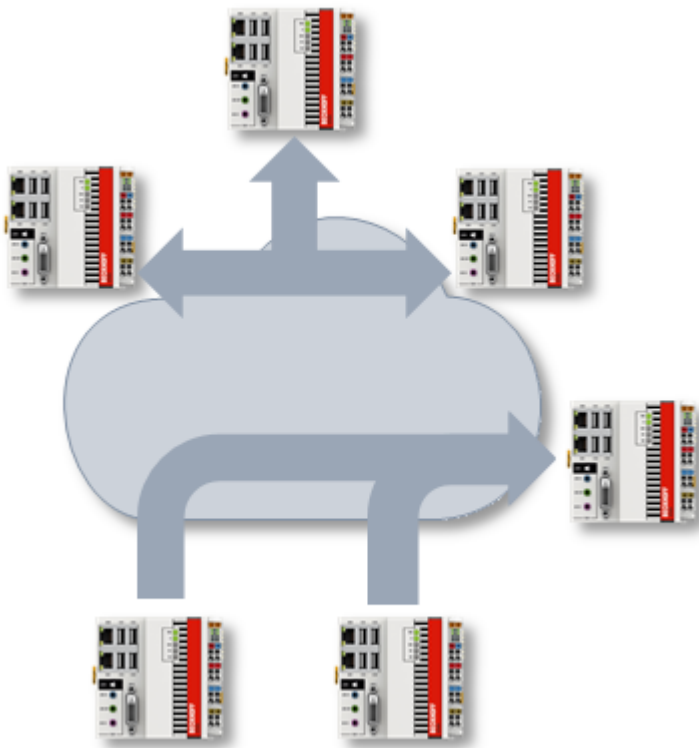
Der Broker übernimmt dabei eine Vermittlung der eintreffenden ADS-Kommandos anhand der hinterlegten Konfiguration.

#### Virtuelles AMS Netzwerk

Im Broker können unterschiedliche „Virtuelle AMS-Netzwerke“ mit unterschiedlichen Teilnehmern definiert werden. Jeder TwinCAT Router öffnet hierzu eine in seiner Konfiguration eingestellte MQTT-Verbindung zum Broker.

Im Broker wird konfiguriert, welche Geräte auf welche anderen Geräte zugreifen dürfen.

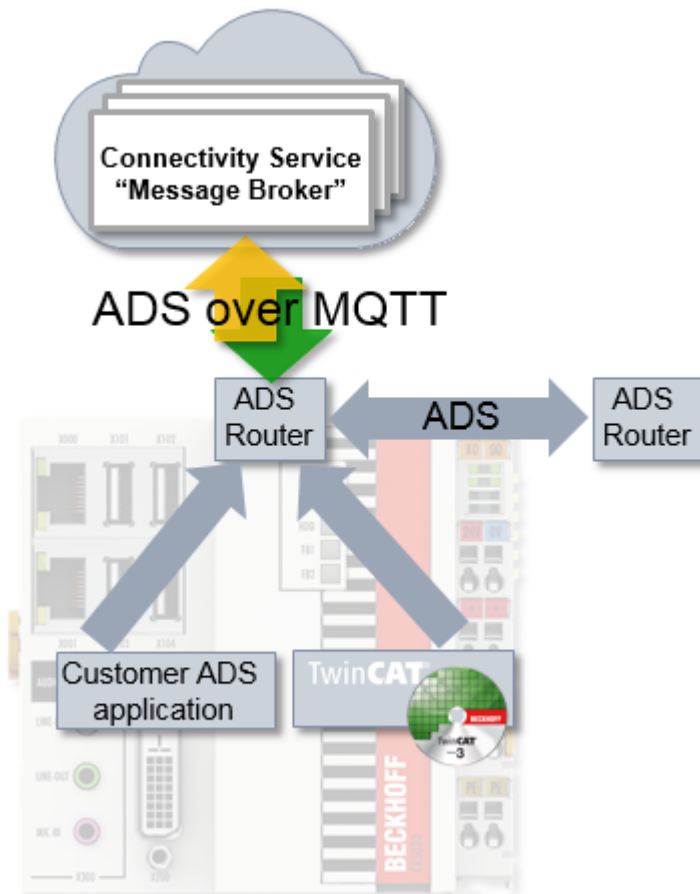
Insgesamt können virtuelle AMS-Netzwerke über einen Broker abgebildet werden.



#### Lokale Realisierung

Die Realisierung der ADS-over-MQTT-Anbindung erfolgt über den TwinCAT Router als zusätzlichen Transportkanal. Hierdurch ist die Erweiterung bezüglich der ADS-Clients wie auch die ADS-Server auf den jeweiligen Geräten transparent.





### Technische Realisierung

Auf MQTT Protokoll-Ebene wird jeder ADS-Router als „User“ abgebildet, wobei dieses keine exklusive Beziehung darstellen muss.

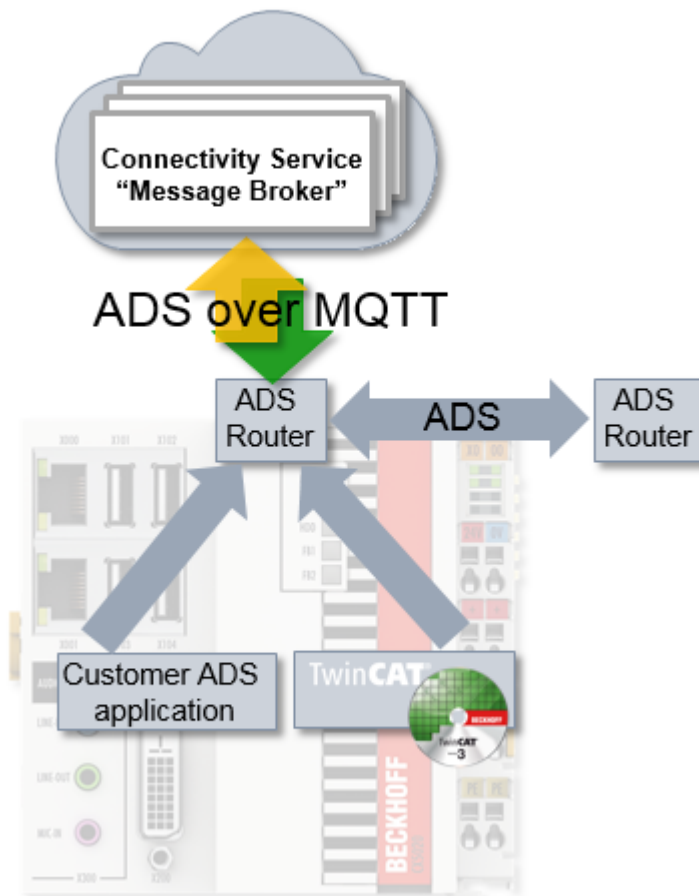
Es werden zwei unterschiedliche Topics-Kategorien von einem jeweiligen Kommunikationsteilnehmer verwendet:

- Discovery: **<NetworkName>/<AmsNetId>/info**  
Ein sich verbindender Router sendet eine **RETAIN** Nachricht an dieses Topic, gleichzeitig subscribed er sich auf **<NetworkName>/+/info (QoS2)**, sodass er über weitere verbundene Router informiert wird.
- Kommunikation: **<NetworkName>/<AmsNetId>/ams/#**  
Ein Router subscribed sich auf die **<NetworkName>/<AmsNetId>/ams/# (QoS2)**.  
Die ADS-Kommandos werden zu diesem Router an **<NetworkName>/<AmsNetId>/ams** gesendet und die Antworten über **<NetworkName>/<AmsNetId>/ams/res**.

Somit ergibt sich, dass der Broker **RETAIN**-Topics sowie **QoS** implementieren muss, wie in der Einleitung beschrieben. Beispielhaft sei hier der Eclipse Mosquitto-Broker genannt.

#### 3.1.3.3 Transparente Nachrüstung

Durch die Realisierung von ADS-over-MQTT innerhalb des TwinCAT Routers wird ein nachträgliches „Retrofitting“ von Anwendungen ermöglicht. Alle ADS Anwendungen (Client und Server), dazu zählen auch Anwendungen die von Kunden geschrieben wurden, müssen nicht neu übersetzt werden.



Die ADS Anwendungen nutzen ADS Routen, um den Kommunikationspartner zu identifizieren. Diese ADS Route ist unabhängig vom Transportkanal und wird im TwinCAT Router beschrieben.

Wenn die genutzte Route auf eine ADS-over-MQTT Verbindung umgestellt wird, wird der ADS-Verkehr über den Broker (und damit ggf. abgesichert) transportiert.

### 3.1.3.4 Konfiguration

Die Konfiguration erfolgt über XML-Dateien sowohl auf Seiten des TwinCAT Systems, wie auch für den MQTT Broker.

#### 3.1.3.4.1 TwinCAT

Der TwinCAT Router wird durch eine XML konfiguriert, um eine Verbindung mit einem oder mehreren Routern aufzunehmen.

Hierfür können die hier beschriebenen XML-Dateien mit beliebigen Namen unter dem Ordner `C:\TwinCAT\3.x\Target\Routes` (Windows CE: `\Hard Disk\TwinCAT\3.x\Target\Routes\`) (x = TwinCAT Versionsnummer) abgelegt werden. Gespeicherte Änderungen werden übernommen, wenn der TwinCAT Router initialisiert wird, was beispielsweise bei den Übergängen von RUN->CONFIG oder auch CONFIG->CONFIG erfolgt.

Die XML Datei ist wie folgt aufgebaut:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
  <RemoteConnections>
    <Mqtt>
      <Address Port="1883">BROKER-ADDRESS</Address>
      <Topic>VirtualAmsNetwork1</Topic>
      <User>CX-123456</User>
    </Mqtt>
  </RemoteConnections>
</TcConfig>
```

```

</Mqtt>
</RemoteConnections>
</TcConfig>

```

Hierbei wird eine Verbindung aufgebaut und der TwinCAT-Router meldet sich mit dem gegebenen Namen (hier: CX-123456) bei dem Broker, welcher unter BROKER-ADDRESS erreichbar ist, und dem Port 1883 an. Die BROKER-ADDRESS ist dabei IP oder Name des Rechners auf dem der Broker läuft.

Dabei ist er Teilnehmer am Netzwerk „VirtualAmsNetwork1“ im Broker, welches sich in den verwendeten Topics wiederfindet, wie es unter [Architektur \[► 185\]](#) beschrieben ist.

Das Element <User> gibt dabei den User auf MQTT-Ebene an und kann im Broker z.B. in der [Mosquitto-Erweiterung \[► 187\]](#) genutzt werden, um Zugriffe zu konfigurieren.

Optional kann das <Mqtt>-Element ein Attribut ClientId tragen, um die MQTT-ClientID anzugeben. Diese wird ansonsten aus dem <User> und einem beliebigen String gebildet.

Diese Konfiguration baut eine unverschlüsselte Verbindung auf; Möglichkeiten der Verschlüsselung sind unter [Security \[► 189\]](#) dokumentiert.

### 3.1.3.4.2 Broker

Der MQTT Broker wird genutzt, um die ADS-Kommandos zwischen den Routern zu vermitteln. Die verwendete Topic-Struktur ist unter [Architektur \[► 185\]](#) beschrieben.

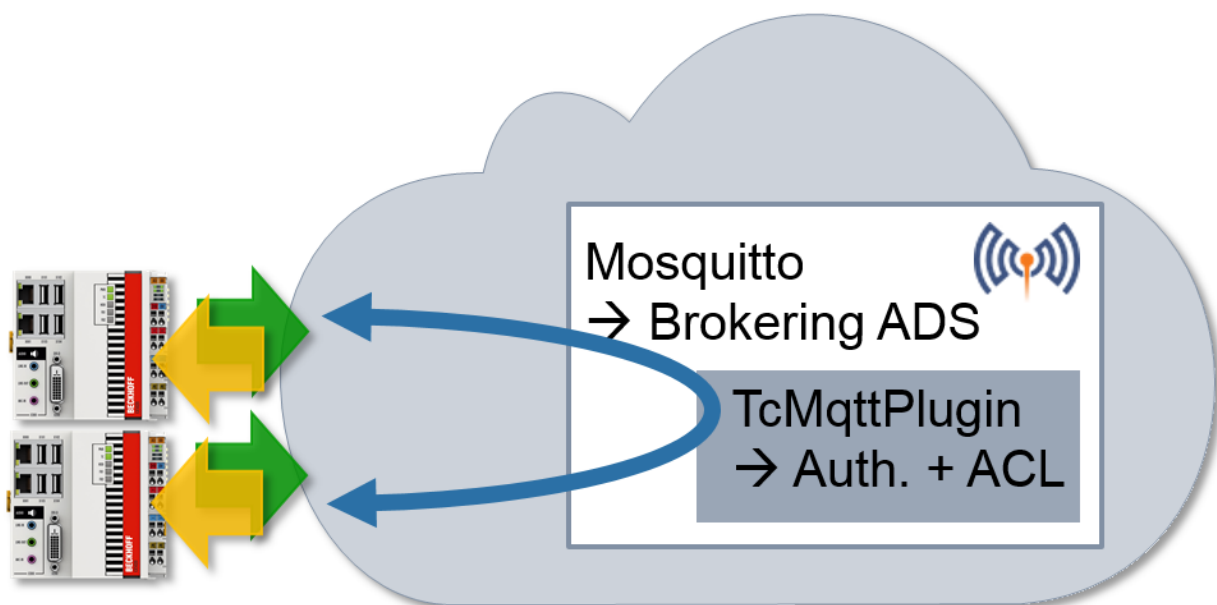
#### Allgemeines

Für ADS-over-MQTT kann ein beliebiger MQTT-Broker mit passender Unterstützung von beispielsweise RETAIN und QoS genutzt werden.

Ist dieser Broker im Sinne von Security schützenswert, da die ADS-Nachrichten schützenswert sind, müssen entsprechende Maßnahmen ergriffen werden. Die Security-Konfiguration auf Seiten von TwinCAT und beispielhaft auch für den Eclipse Mosquitto-Broker ist unter [Security \[► 189\]](#) beschrieben.

#### Tc-Plugin TcMqttPlugin.dll für den Eclipse Mosquitto Broker

Um in dem MQTT-Broker ein virtuelles Netzwerk von ADS-Geräten zu definieren, existiert eine Erweiterung für den [Eclipse Mosquitto Broker](#). Mittels dieser Erweiterung können sowohl Zugriffsrechte per PreSharedKey auf den Broker, wie auch Zugriffe zwischen den TwinCAT Routern mittels einer ACL (AccessControlList) eingestellt werden.



Das Plugin wird mit der TwinCAT-Installation ausgeliefert und befindet sich im Ordner C:\TwinCAT\AdsApi\TcMqttPlugin bzw. C:\TwinCAT\AdsApi\x64\TcMqttPlugin, falls ein 64-bit Mosquitto Broker eingesetzt wird.

In der Mosquitto-Konfiguration wird das Plugin wie folgt eingebunden:

```
auth_plugin <Path>TcMqttPlugin.dll
auth_opt_xml_file <Path>ACL.xml
```

Die Mosquitto-Konfigurationsdatei wird beim Starten des Mosquitto Brokers mittels des Parameters „-c“ angegeben, womit das Plugin inkl. der Konfiguration dann geladen wird.

Die Datei ACL.xml wird dabei in den folgenden Abschnitten beschrieben und stellt auf der einen Seite die Zugriffskonfiguration per PreSharedKey auf den Broker selbst bereit, wie aber auch die Konfiguration der Kommunikation zwischen den verbundenen TwinCAT Routern.

### Konfiguration „Virtuelles AmsNetzwerk“

Das Plugin TcMqttPlugin bietet die Möglichkeit virtuelle AmsNetzwerke zu konfigurieren. Hierzu wird zu jedem Zielgerät angegeben, welches Gerät Zugriff auf welches andere Gerät hat.

Im Gegensatz zu den klassischen ADS-Routen sind diese Verbindungen gerichtet: Ein Ziel hat also nicht gleichzeitig das Recht auf die Quelle zuzugreifen.

```
<TcMqttAclConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\TwinCAT\3.1\Config\Modules\TcMqttAclConfig.xsd"
AnonymousLogin="true">
<!-- PSK Elements, if used -->
<Ams>
<Topic>VirtualAmsNetwork1</Topic>
<User>
<Name>EngineeringStation</Name>
</User>
<User>
<Name>CX-123456</Name>
<Access>EngineeringStation</Access>
</User>
<User>
<Name>CX-567890</Name>
<Access>EngineeringStation</Access>
</User>
</Ams>
</TcMqttAclConfig>
```

Innerhalb eines <Ams>-Knotens wird der Name des AmsNetzwerks definiert. Er wird in den genutzten MQTT Topics zur Identifizierung der Netzwerke verwendet.

Einzelne <User> Elemente beschreiben die Teilnehmer. Diese Elemente besitzen ein <Name>-Element, welches die MQTT-Identity beschreibt, mit der die Verbindung aufgebaut wurde – im Normalfall der Name des Teilnehmers. Zusätzlich werden zugriffsberechtigte Geräte über das <Access>-Element definiert.

In dem Beispiel hat „EngineeringStation“ also Zugriff auf zwei CX-Geräte, die CX-Geräte jedoch nicht auf die „EngineeringStation“ und auch nicht untereinander.

Die Datei wird zyklisch neu geladen, sodass ein Neustart des Brokers nicht nötig ist.

Da in dieser Erklärung keine Verschlüsselung vorgesehen ist, wird AnonymousLogin="true" verwendet.

### Einschränkungen bzgl. der anzumeldenden AmsNetId

Bei dieser Konfiguration kann jedes gültig verbundene Gerät eine beliebige AmsNetID und damit Identität aus Sicht von ADS annehmen. Dieses kann eingeschränkt werden:

```
<User>
<Name>CX-567890</Name>
<Access>EngineeringStation</Access>
<NetId>192.168.56.1.1.1</NetId>
</User>
```

Sobald mindestens eine NetId angegeben ist, kann nur eine NetId aus dieser Liste angemeldet werden.

## Mosquitto Einstellungen

Im Zusammenhang mit der Konfiguration mittels TcMqttPlugin sind einige Einstellungen auf Seiten des Mosquitto Brokers wichtig zu beachten. Hierzu gehören:

- `psk_hint` Bezeichnet den `psk_hint` für den Verbindungsaufbau. Wird aktuell auf TwinCAT-Seite nicht überprüft.
- `port <1883|8883>` Der Port bezeichnet den Netzwerk-Port, den der Broker bereitstellt. Typischerweise ist 1883 unverschlüsselt und 8883 verschlüsselt.
- `require_certificate <true|false>` Bezeichnet die Notwendigkeit von Zertifikaten.
- `use_identity_as_username true` Bezeichnet, ob die Identity von Zertifikaten als Username auf Ebene von MQTT benutzt wird. Zur Nutzung des TcMqttPlugin wird dieses benutzt, sodass es als `true` anzulegen ist.

Minimal-Konfigurationsbeispiele sind in Abhängigkeiten von der genutzten TLS Verbindung in den entsprechenden Abschnitten beschrieben.

### 3.1.3.4.3 Security

Es stehen Möglichkeiten zur Absicherung der Kommunikation bereit. Hierfür kann eine TLS Verbindung auf Basis von X.509 Zertifikaten oder ein PreSharedKey (PSK) genutzt werden.

Insbesondere, wenn über nicht-vertrauenswürdige Netze (beispielsweise Internet) kommuniziert wird, wird empfohlen die Kommunikation mit TLS abzusichern. Der Broker selbst muss in einer vertrauenswürdigen Umgebung betrieben werden, da dort alle Nachrichten ungesichert sind.

#### ● Kompromittierung des virtuellen ADS Netzwerks

**i** Auch wenn die Kommunikation zwischen den Teilnehmern und dem Broker verschlüsselt über TLS erfolgt, existiert keine Absicherung der Teilnehmer untereinander. Die ADS-Kommandos liegen auf dem Broker unverschlüsselt vor.

Wenn ein Teilnehmer kompromittiert wurde, kann der Angreifer über die erlangten Rechte alle ADS-Kommandos ausführen. Zu diesen Kommandos gehören auch Datei-Lese-Operationen oder Operationen zum Starten von Prozessen.

#### 3.1.3.4.3.1 TLS / PreSharedKey (PSK)

PreSharedKeys (PSK) sind Passwörter, die auf beiden Seiten einer Verbindung vorab durch einen Konfigurationsvorgang aufgebracht werden. Genutzt wird zur Kommunikation eine TLS 1.2 Verbindung.

#### TwinCAT Konfiguration mit PSK

Für einen TwinCAT Router kann in der Konfigurationsdatei für die Route mit einem PSK versehen werden, wobei der Key als HexString angegeben wird.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
<RemoteConnections>
  <Mqtt>
    <Address Port="8883">BROKER-ADDRESS</Address>
    <Topic>VirtualAmsNetwork1</Topic>
    <Psk>
      <Identity>EngineeringStation</Identity>
      <Key>4D65696E5061737377C3B67274[...]</Key>
    </Psk>
  </Mqtt>
</RemoteConnections>
</TcConfig>
```

#### ● Sicheres PSK

**i** Ein sinnvoller PreSharedKey wird aus einem HexString von 64 Zeichen gebildet.

Alternativ kann der Key auch von TwinCAT bestimmt werden, damit eine einfachere Eingabe vorgenommen werden kann. Hierfür wird ein Passwort als normaler String im <Pwd> Element angegeben. TwinCAT berechnet aus diesem und der Identity mittels Sha256('Identity'+ 'Pwd') den zu nutzenden PSK. Ist das Attribut „IdentityCaseSensitive“ auf „false“ (oder nicht) gesetzt, wird die Identity als UpperCase-String für die Key-Berechnung verwendet.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
<RemoteConnections>
  <Mqtt>
    <Address Port="8883">BROKER</Address>
    <Topic>VirtualAmsNetwork1</Topic>
    <Psk>
      <Identity>EngineeringStation</Identity>
      <Pwd IdentityCaseSensitive="false">!ABCDEFGHIjklmn123545</Pwd>
    </Psk>
  </Mqtt>
</RemoteConnections>
</TcConfig>
```

### Minimale Mosquitto-Konfiguration

Als einfachste Mosquitto-Konfiguration können für PSKs folgende Einträge genutzt werden:

```
port 8883
psk_hint AHint
use_identity_as_username true
auth_plugin C:\TwinCAT\AdsApi\TcMqttPlugin\TcMqttPlugin.dll
auth_opt_xml_file ACL.xml
```

### Broker-Konfiguration mit PSK

Das Plugin TcMqttPlugin bietet die Möglichkeit einen PSK im Broker zu verwenden, um auf einen Broker zuzugreifen. Die Konfiguration ist in der Konfigurationsdatei des Plugins abgelegt, wobei der PSK als HexString angegeben wird.

Das IdentityCaseSensitive bietet die Möglichkeit, die Identitäten unabhängig von ihrer Groß-/Klein-Schreibweise zu betrachten.

```
<TcMqttAclConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\TwinCAT\3.1\Config\Modules\TcMqttAclConfig.xsd"
IdentityCaseSensitive="false">
  <Psk>
    <Identity>EngineeringStation</Identity>
    <Key>4D65696E5061737377C3B67274</Key>
  </Psk>
<!-- <Ams> Elements following -->
</TcMqttAclConfig>
```

Alternativ kann der Key auch von dem TcMqttPlugin bestimmt werden, damit eine einfachere Eingabe vorgenommen werden kann. Hierfür wird ein Passwort als normaler String im <Pwd> Element angegeben. TwinCAT berechnet aus diesem und der Identity mittels Sha256('Identity'+ 'Pwd') den zu nutzenden PSK. Ist das Attribut auf Ebene von <TcMqttAclConfig> „IdentityCaseSensitive“ auf „false“ (oder nicht) gesetzt, wird die Identity als UpperCase-String für die Key-Berechnung verwendet.

```
<TcMqttAclConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\TwinCAT\3.1\Config\Modules\TcMqttAclConfig.xsd"
IdentityCaseSensitive="false">
  <Psk>
    <Identity>EngineeringStation</Identity>
    <Pwd>!ABCDEFGHIjklmn123545</Pwd>
  </Psk>
<!-- <Ams> Elements following -->
</TcMqttAclConfig>
```

### 3.1.3.4.3.2 TLS / Zertifikate

Zertifikate nach X.509 Standard können verwendet werden, um die entsprechende MQTT-Verbindung zum Broker abzusichern.

## TwinCAT Konfiguration mit Zertifikaten

Für einen TwinCAT-Router können in den MQTT-Routen die Pfade zu den X.509 Zertifikaten konfiguriert werden:

```
<?xml version="1.0" encoding="UTF-8"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/TcConfig">
<RemoteConnections>
  <Mqtt>
    <Address Port="8883">BROKER-ADDRESS</Address>
    <Topic>VirtualAmsNetwork1</Topic>
    <Tls>
      <Ca>C:\TwinCAT\3.1\Target\Certificates\CA.crt</Ca>
      <Cert>C:\TwinCAT\3.1\Target\Certificates\Device.crt</Cert>
      <Key>C:\TwinCAT\3.1\Target\Certificates\Device.key</Key>
    </Tls>
  </Mqtt>
</RemoteConnections>
</TcConfig>
```

Hier werden im Element <Tls> die entsprechenden Pfade zu den Dateien angegeben. <Ca> ist dabei das X.509 Zertifikat der CertificateAuthority, also die ausstellende Stelle, von welcher Zertifikate akzeptiert werden sollen.

Die Elemente <Cert> sowie <Key> enthalten Pfade zu dem öffentlichen bzw. privaten Schlüssel des zu verwendenden Zertifikates.

- Der Hostname des Brokers („BROKER-ADDRESS“) muss zu dem Common Name des verwendeten Zertifikats passen. Dieses wird von den Clients überprüft.
- Der Common Name des Client Zertifikats wird als Identity in MQTT (und im TcMqttPlugin.dll) verwendet.

### Minimale Mosquitto-Konfiguration

Als einfachste Mosquitto-Konfiguration können für die Nutzung von Zertifikaten folgende Einträge genutzt werden:

```
port 8883
cafile cert/CA.crt
certfile cert/Broker.crt
keyfile cert/Broker.key
require_certificate true
use_identity_as_username true
auth_plugin C:\TwinCAT\AdsApi\TcMqttPlugin\TcMqttPlugin.dll
auth_opt_xml_file ACLCerts.xml
```

### Broker-Konfiguration mit Zertifikaten

Die Identity, die in den <Ams> Elementen verwendet wird um das AmsNetzwerk zu beschreiben, wird über den CN des Zertifikates festgelegt.

Durch die CertificateAuthority ist festgelegt, welche Zertifikate Zugriff erhalten.

Eine zusätzliche Konfiguration auf Brokerseite entfällt damit.

### 3.1.3.5 Anwendungsszenarien

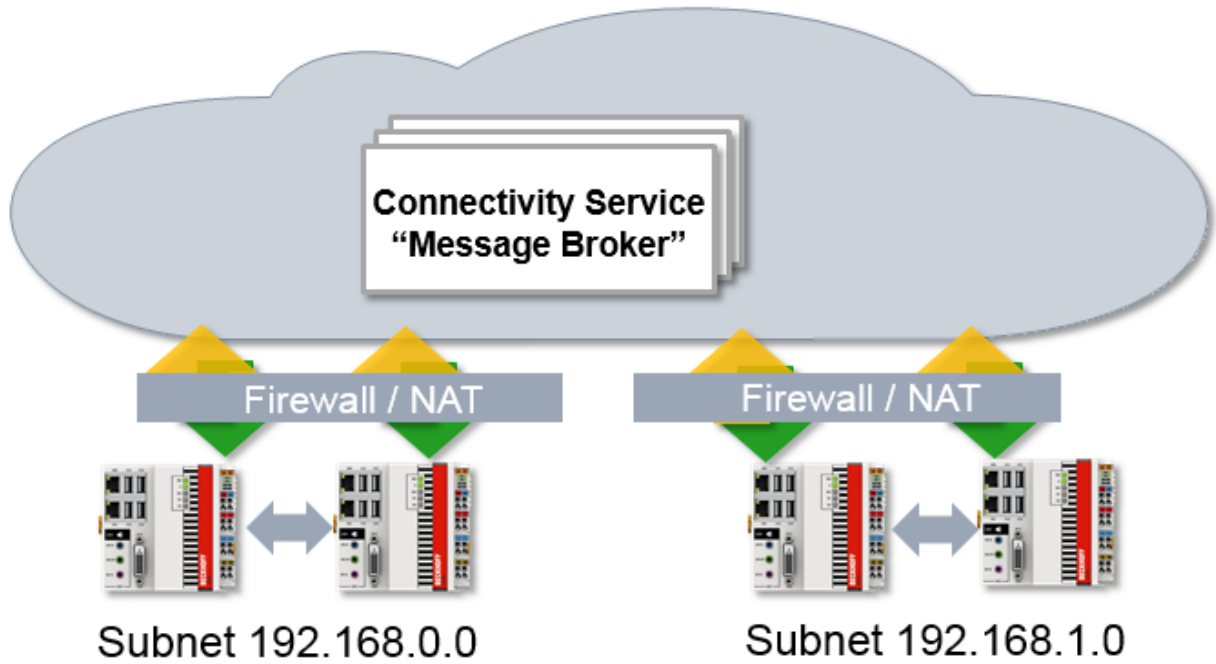
An dieser Stelle sollen einige Anwendungsszenarien beschrieben werden, um den Mehrwert von ADS-over-MQTT darzustellen.

#### 3.1.3.5.1 NAT-basierte Netzwerke

Durch die ausgehenden MQTT-Verbindungen von TwinCAT zu dem Broker wird eine einfache Kommunikation zwischen Subnetzen ermöglicht: Alle angeschlossenen Geräte müssen eine ausgehende Verbindung zu dem Broker aufbauen können – diese eine Verbindung wird für die gesamte ADS-Kommunikation verwendet. Der Broker ist die einzige Komponente mit eingehenden Verbindungen. Dies ist insbesondere bei Produktionsprozessen in großen, ggf. verteilten Anlagen von Vorteil. Hier werden häufig Subnetze mit NAT, Firewalls etc. eingesetzt. Trotzdem soll gelegentlich eine ADS-Kommunikation



über die Netzwerkgrenzen hinweg ermöglicht werden.  
 In vielen Fällen ist eine solche Kommunikationsmöglichkeit wertvoll. Subnetze können aber auch aus Security-Gründen eingerichtet werden, sodass eine Kommunikation nicht gewünscht ist (Stichwort: „Zoning“).

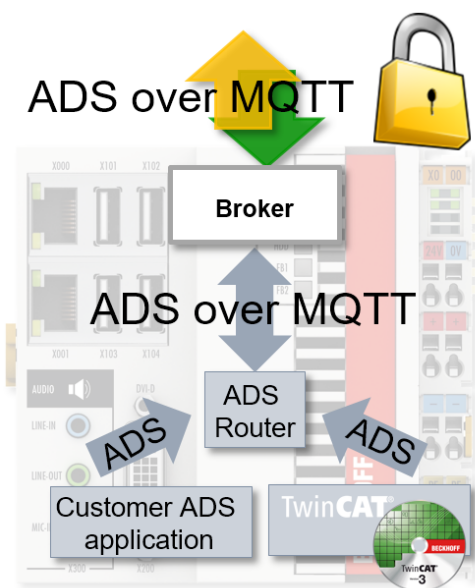


**3.1.3.5.2 ADS Verschlüsselung**

Durch die Fähigkeit von TLS, welches für eine Verschlüsselung von MQTT auf Transportebene genutzt wird, kann MQTT auch verwendet werden, um eine verschlüsselte ADS-Kommunikation zu ermöglichen. Dafür kann der Broker lokal auf den PC-basierten Steuerungen installiert werden. Der Broker wird dabei so konfiguriert, dass er lediglich die lokale Steuerung als Zugriffspunkt in einem virtuellen AmsNetzwerk anbietet.

Wird dann eine Verschlüsselung aktiviert und genutzt, entsteht eine auf TLS basierende, abgesicherte Verbindung per ADS zu einem TwinCAT-System.

Wenn diese genutzt wird, können ADS-Verbindungen durch die Firewall geblockt werden.



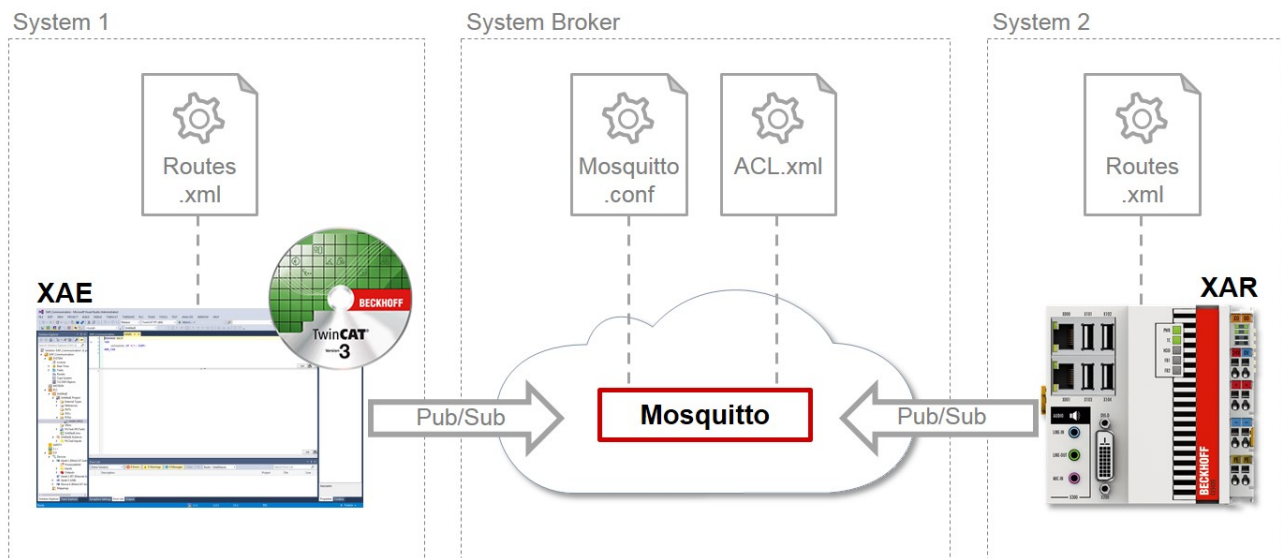


### 3.1.3.6 Beispiel



Dieses Beispiel stellt lediglich einen Workflow dar, um eine Testumgebung aufzusetzen. Sämtliche Parameter wie Zertifikatlaufzeiten, Schlüssellängen etc. sind entsprechend der realen Umgebung und Anwendung vorzunehmen.

Die Funktionsweise und Konfiguration von ADS-over-MQTT soll im Folgenden anhand eines Beispiels genauer erläutert werden. In dem Beispiel wird als Message Broker der Eclipse Mosquitto Broker verwendet sowie OpenSSL zur Erstellung der Zertifikate. Der Mosquitto Message Broker soll den Datenaustausch zwischen einem TwinCAT XAE und einer TwinCAT XAR realisieren. Um die Kommunikation abzusichern, wird das Verschlüsselungsprotokoll TLS in Kombination mit X.509 Zertifikaten oder TLS in Kombination mit PSK genutzt. Im nachfolgenden Bild ist der Aufbau des Anwendungsbeispiels schematisch dargestellt.



Zur Nutzung von ADS-over-MQTT sind die im Bild aufgeführten Konfigurationsdateien zu erstellen und entsprechend anzupassen. Im Folgenden werden zwei Beispiele vorgestellt.

In dem ersten wird ADS-over-MQTT mit TLS und X.509 Zertifikaten [► 193] konfiguriert und in dem zweiten mit TLS und PreSharedKeys (PSK) [► 200].

#### 3.1.3.6.1 ADS-over-MQTT mit TLS und X.509 Zertifikaten

In diesem Abschnitt wird ein Beispiel zur Konfiguration von ADS-over-MQTT mit PSK- und X.509-Zertifikaten eingeführt. Die einzelnen Schritte zur Realisierung der Kommunikationsschnittstelle sind:

- ✓ TwinCAT 3.1 Build 4022.0 oder höher ist auf System 1 als XAE-Version installiert und auf System 2 als XAR-Version.
- 1. Erzeugen Sie die Zertifikate zur sicheren Kommunikation per TLS. Hierfür verwenden Sie das Programm OpenSSL, das Sie unter <https://www.openssl.org/source/> laden und anschließend installieren.

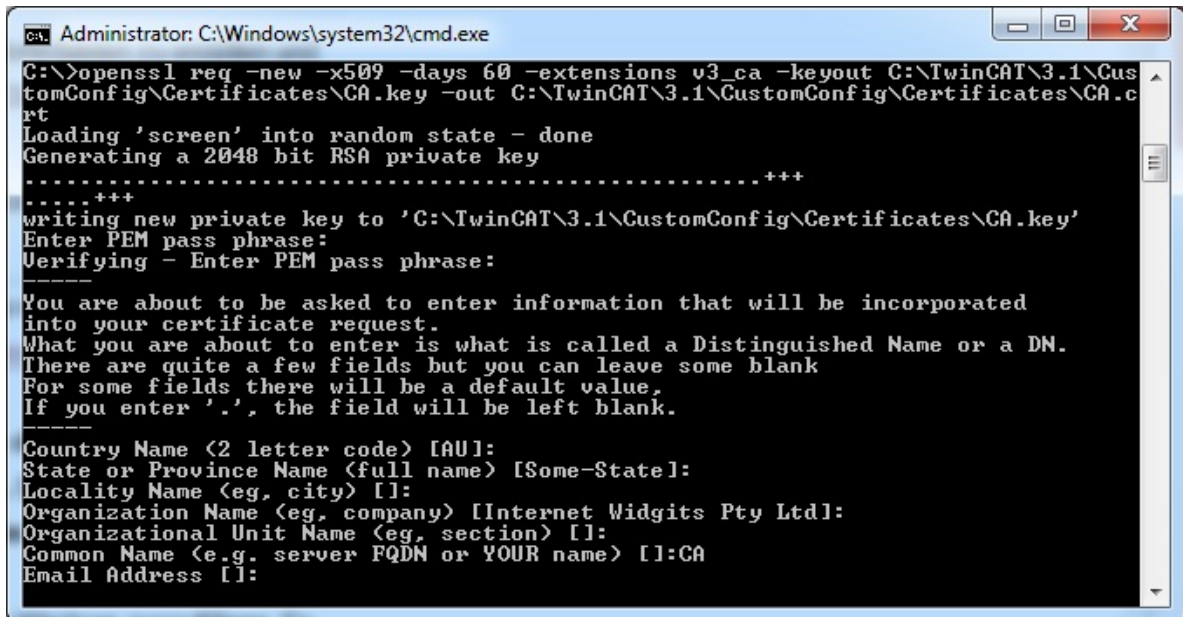
**Hinweis** Bei einem Windows-Betriebssystem muss der Pfad, in dem sich die Konfigurationsdatei von OpenSSL befindet, als Umgebungsvariable gesetzt sein. Führen Sie dies im Kommandozeilenprogramm eines x64-Systems mit folgendem Befehl durch: set

```
OPENSSL_CONF=C:\OpenSSL-Win64\bin\openssl.cfg
```

Mit Abschluss der Installation führen Sie das Kommandozeilenprogramm von Windows aus. Es wird mit der Generierung des CA-Zertifikats (Certificate Authority) begonnen. Dabei wird die Eingabe eine Passphrase verlangt. Geben Sie diese ein, merken Sie sich und geben Sie weitere Informationen zum CA an. Der entsprechende Befehl zur Generierung des CA Zertifikats lautet wie folgt:

```
openssl req -new -x509 -days 60 -extensions v3_ca -keyout C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key -out C:\TwinCAT\3.1\CustomConfig\Certificates\CA.crt
```

⇒ Im Kommandozeilenprogramm sollte das Ergebnis wie folgt aussehen:



```

Administrator: C:\Windows\system32\cmd.exe
C:\>openssl req -new -x509 -days 60 -extensions v3_ca -keyout C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key -out C:\TwinCAT\3.1\CustomConfig\Certificates\CA.crt
Loading 'screen' into random state - done
Generating a 2048 bit RSA private key
.....+++
++++
writing new private key to 'C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:CA
Email Address []:

```

- Erstellen Sie das Broker Zertifikat. Wichtig ist hierbei als CN (Common Name) den Hostnamen oder die IP-Adresse des Systems zu verwenden, auf dem der Mosquitto Message Broker betrieben werden soll. Zudem ist zu beachten, dass das System über die IP-Adresse bzw. den Hostnamen von den Clients erreichbar ist. Zur Generierung der Broker Zertifikate sind folgende Befehle im Kommandozeilenprogramm auszuführen:

Erstellen des Zertifikates:

```
openssl genrsa -out C:\TwinCAT\3.1\CustomConfig\Certificates\broker.key 2048
```

Erstellen des Certificate Signing Requests:

```
openssl req -out C:\TwinCAT\3.1\CustomConfig\Certificates\broker.csr -key C:\TwinCAT\3.1\CustomConfig\Certificates\broker.key -new
```

Signieren des CSR durch die zuvor erstellte CA, wofür das Passwort benötigt wird, welches bei der Erstellung der CA angegeben wurde:

```
openssl x509 -req -in C:\TwinCAT\3.1\CustomConfig\Certificates\broker.csr -CA C:\TwinCAT\3.1\CustomConfig\Certificates\CA.crt -CAkey C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key -CAcreateserial -out C:\TwinCAT\3.1\CustomConfig\Certificates\broker.crt -days 60
```

⇒ Im Kommandozeilenprogramm sollte das Ergebnis wie folgt aussehen:

```

Administrator: C:\Windows\system32\cmd.exe
C:\>openssl genrsa -out C:\TwinCAT\3.1\CustomConfig\Certificates\broker.key 2048
Loading 'screen' into random state - done
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)

C:\>openssl req -out C:\TwinCAT\3.1\CustomConfig\Certificates\broker.csr -key C:\TwinCAT\3.1\CustomConfig\Certificates\broker.key -new
Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:192.168.1.8
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

C:\>openssl x509 -req -in C:\TwinCAT\3.1\CustomConfig\Certificates\broker.csr -CA
C:\TwinCAT\3.1\CustomConfig\Certificates\CA.crt -CAkey C:\TwinCAT\3.1\CustomCo
nfig\Certificates\CA.key -CAcreateserial -out C:\TwinCAT\3.1\CustomConfig\Certif
icates\broker.crt -days 60
Loading 'screen' into random state - done
Signature ok
subject=C=AU/ST=Some-State/O=Internet Widgits Pty Ltd/CN=192.168.1.8
Getting CA Private Key
Enter pass phrase for C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key:

```

3. Generieren Sie die beiden Client Zertifikate für das TwinCAT XAE und TwinCAT XAR. Die OpenSSL-Befehle dazu sind im Folgenden angegeben.

Generierung des XAE Zertifikats:

```
openssl genrsa -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.key 2048
```

Erstellen des CSR:

```
openssl req -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.csr -key C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.key -new
```

Signieren des CSR durch die zuvor erstellte CA, wofür das Passwort benötigt wird, welches bei der Erstellung der CA angegeben wurde:

```
openssl x509 -req -in C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.csr -CA C:\TwinCAT\3.1\CustomConfig\Certificates\CA.crt -CAkey C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key -CAcreateserial -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.crt -days 60
```

Generierung des XAR Zertifikats:

```
openssl genrsa -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.key 2048
```

Erstellen des CSR:

```
openssl req -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.csr -key C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.key -new
```

Signieren des CSR durch die zuvor erstellte CA, wofür das Passwort benötigt wird, welches bei der Erstellung der CA angegeben wurde:

```
openssl x509 -req -in C:
```

```
\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.csr -CA C:  
\TwinCAT\3.1\CustomConfig\Certificates\CA.crt -CAkey C:  
\TwinCAT\3.1\CustomConfig\Certificates\CA.key -CAcreateserial -out C:  
\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.crt -days 60
```

⇒ Im Kommandozeilenprogramm sollte das Ergebnis wie folgt aussehen:  
TwinCAT XAE:

```

Administrator: C:\Windows\system32\cmd.exe
C:\>openssl genrsa -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.key
2048
Loading 'screen' into random state - done
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)

C:\>openssl req -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.csr -key C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.key -new
Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:TwinCAT_XAE
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

C:\>openssl x509 -req -in C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.csr -CA C:\TwinCAT\3.1\CustomConfig\Certificates\CA.crt -CAkey C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key -CAcreateserial -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.crt -days 60
Loading 'screen' into random state - done
Signature ok
subject=/C=AU/ST=Some-State/O=Internet Widgits Pty Ltd/CN=TwinCAT_XAE
Getting CA Private Key
Enter pass phrase for C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key:
    
```

TwinCAT XAR:

```

Administrator: C:\Windows\system32\cmd.exe
C:\>openssl genrsa -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.key
2048
Loading 'screen' into random state - done
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)

C:\>openssl req -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.csr -key C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.key -new
Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:TwinCAT_XAR
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

C:\>openssl x509 -req -in C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.csr -CA C:\TwinCAT\3.1\CustomConfig\Certificates\CA.crt -CAkey C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key -CAcreateserial -out C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAR.crt -days 60
Loading 'screen' into random state - done
Signature ok
subject=/C=AU/ST=Some-State/O=Internet Widgits Pty Ltd/CN=TwinCAT_XAR
Getting CA Private Key
Enter pass phrase for C:\TwinCAT\3.1\CustomConfig\Certificates\CA.key:
    
```



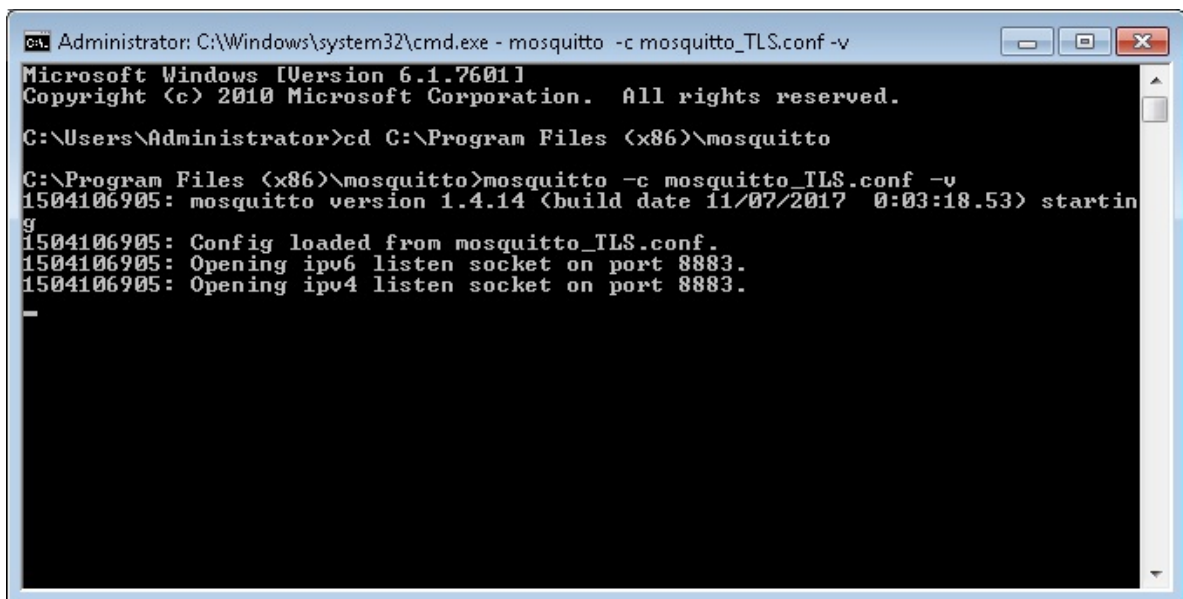
- Im Anschluss an die Generierung der Zertifikate installieren Sie den Mosquitto Broker. Laden Sie ihn von <https://mosquitto.org/download/> und installieren Sie ihn auf dem entsprechenden System.
- Nach der Installation des Mosquitto Brokers erstellen Sie für diesen die Konfigurationsdatei `mosquitto_TLS.conf` zur Verwendung von TLS mit Zertifikaten. Als Speicherort wählen Sie das Installationsverzeichnis von Mosquitto (Standardmäßig: `C:\Program Files (x86)\mosquitto`). Die Konfigurationsdatei soll folgende Einträge enthalten:

```
port 8883
allow_anonymous false
require_certificate true
use_identity_as_username true
cafile C:\TwinCAT\3.1\CustomConfig\Certificates\CA.crt
certfile C:\TwinCAT\3.1\CustomConfig\Certificates\broker.crt
keyfile C:\TwinCAT\3.1\CustomConfig\Certificates\broker.key
auth_plugin C:\TwinCAT\AdsApi\TcMqttPlugin\TcMqttPlugin.dll
auth_opt_xml_file C:\TwinCAT\AdsApi\TcMqttPlugin\ACL.xml
```

- Nun starten Sie den Mosquitto Message Broker über das Kommandozeilenprogramm von Windows. Dazu wechseln Sie in das Installationsverzeichnis des Mosquittos und führen Sie den nachfolgend aufgeführten Befehl aus. Bei dem Befehl sorgt `-v` für eine Ausgabe der Nachrichten, die durch den Broker versendet oder auch abgelehnt werden. Diese Option ist gerade während Tests sinnvoll zu nutzen.

```
mosquitto -c mosquitto_TLS.conf -v
```

⇒ Anschließend sollte das Ergebnis wie folgt aussehen:



```
Administrator: C:\Windows\system32\cmd.exe - mosquitto -c mosquitto_TLS.conf -v
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2010 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\Program Files (x86)\mosquitto

C:\Program Files (x86)\mosquitto>mosquitto -c mosquitto_TLS.conf -v
1504106905: mosquitto version 1.4.14 (build date 11/07/2017 0:03:18.53) starting
1504106905: Config loaded from mosquitto_TLS.conf.
1504106905: Opening ipv6 listen socket on port 8883.
1504106905: Opening ipv4 listen socket on port 8883.
```

- Erstellen Sie für den Mosquitto als nächstes die `ACL.xml`, in der die Zugriffsrechte der Clients festgelegt werden. Legen Sie sie im Verzeichnis `C:\TwinCAT\AdsApi\TcMqttPlugin\` ab. Nehmen Sie in der `ACL.xml` die folgenden Einträge vor:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcMqttAclConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:
\TwinCAT\3.1\Config\Modules\TcMqttAclConfig.xsd">
<Ams>
  <Topic>VirtualAmsNetwork1</Topic>
  <User>
    <Name>TwinCAT_XAE</Name>
  </User>
  <User>
    <Name>TwinCAT_XAR</Name>
    <Access>TwinCAT_XAE</Access>
```

```
</User>  
</Ams>  
</TcMqttAclConfig>
```

8. Konfigurieren Sie nun das TwinCAT XAE und TwinCAT XAR für ADS-over-MQTT. Dazu erstellen Sie auf beiden Systemen im Verzeichnis C:\TwinCAT\3.x\Target\ einen Ordner „Routes“ und erzeugen Sie in diesem eine Datei mit dem Namen „MyRoute.xml“ (der Dateiname ist dabei beliebig). Folgend ist der Inhalt der Datei vom TwinCAT XAE dargestellt. Passen Sie für das TwinCAT XAR im <Cert>- und <Key>-Feld die Pfade entsprechend an. Wichtig ist immer, dass im <Address>-Feld der gleiche Eintrag angegeben wird, wie für den CN des Mosquitto Broker Zertifikats.

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/  
TcConfig">  
<RemoteConnections>  
  <Mqtt>  
    <Address Port="8883">192.168.1.8</Address>  
    <Topic>VirtualAmsNetwork1</Topic>  
    <Tls>  
      <Ca>C:\TwinCAT\3.1\CustomConfig\Certificates\CA.crt</Ca>  
      <Cert>C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.crt</  
Cert>  
      <Key>C:\TwinCAT\3.1\CustomConfig\Certificates\TwinCAT_XAE.key</  
Key>  
    </Tls>  
  </Mqtt>  
</RemoteConnections>  
</TcConfig>
```

9. Damit die gespeicherte Konfiguration von ADS-over-MQTT für die TwinCAT-Systeme wirksam wird, initialisieren Sie jeweils den TwinCAT Router neu. Dies geschieht, indem Sie vom RUN-Mode in den CONFIG-Mode schalten oder vom CONFIG-Mode erneut in den CONFIG-Mode.

- ⇒ Zuletzt prüfen Sie, ob vom XAE eine Verbindung zum XAR aufgebaut werden kann. Ist dies der Fall, sollten die Ausgaben des Mosquitto Message Brokers wie folgt aussehen:

```

C:\Windows\system32\cmd.exe - mosquitto -p 8883 -v -c mosquitto_TLS.conf
172.17.214.125.1.1/ams/res', ... (56 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (56 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (56 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (64 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (64 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (36 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (36 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (304 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (304 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (64 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/5.19.111.141.1.1/ams', ... (64 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (92 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (92 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (36 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/172.17.214.125.1.1/ams/res', ... (36 bytes))

```

- ⇒ Damit ist ADS-over-MQTT mit TLS auf Basis von Zertifikaten für TwinCAT XAE und XAR erfolgreich eingerichtet.

### 3.1.3.6.2 ADS-over-MQTT mit TLS und PSK

Neben der Verwendung von TLS mit Zertifikaten kann MQTT-over-ADS auch auf Basis von PSK (Pre Shared Key) konfiguriert werden. Für diesen Anwendungsfall soll ebenfalls ein kurzes Beispiel eingeführt werden, dass bei der Umsetzung unterstützen soll. Es sind folgende Schritte durchzuführen:

1. Zunächst erstellen Sie die Konfigurationsdatei des Mosquittos (mosquitto\_PSK.conf) in dessen Installationsverzeichnis (Standardmäßig: C:\Program Files (x86)\mosquitto). Danach nehmen Sie in der Datei die folgenden Einträge vor:

```

auth_plugin C:\TwinCAT\AdsApi\TcMqttPlugin\TcMqttPlugin.dll
auth_opt_xml_file C:\TwinCAT\AdsApi\TcMqttPlugin\ACL.xml
port 8883
psk_hint something
use_identity_as_username true

```

2. Im nächsten Schritt führen Sie den Mosquitto Message Broker aus. Der entsprechende Befehl dazu lautet:

```

mosquitto -c mosquitto_PSK.conf -v

```



3. In die ACL.xml tragen Sie die Schlüssel für das TwinCAT XAR und XAE ein:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcMqttAclConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:
\TwinCAT\3.1\Config\Modules\TcMqttAclConfig.xsd">
<Psk>
  <Identity>TwinCAT_XAE</Identity>
  <Pwd>abcdef1234!</Pwd>
</Psk>
<Psk>
  <Identity>TwinCAT_XAR</Identity>
  <Pwd>ghijkl5678?</Pwd>
</Psk>
<Ams>
  <Topic>VirtualAmsNetwork1</Topic>
  <User>
    <Name>TwinCAT_XAE</Name>
  </User>
  <User>
    <Name>TwinCAT_XAR</Name>
    <Access>TwinCAT_XAE</Access>
  </User>
</Ams>
</TcMqttAclConfig>
```

4. Machen Sie die in der ACL.xml definierten Schlüssel ebenfalls dem TwinCAT XAR und XAE bekannt. Dazu passen Sie auf beiden Systemen die Routes.xml im Verzeichnis C:\TwinCAT\3.x\Target\Routes an bzw. erstellen Sie sie. Für das TwinCAT XAE sind die Einträge nachfolgend aufgeführt:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.beckhoff.com/schemas/2015/12/
TcConfig">
<RemoteConnections>
  <Mqtt>
    <Address Port="8883">192.168.1.8</Address>
    <Topic>VirtualAmsNetwork1</Topic>
    <Psk>
      <Identity>TwinCAT_XAE</Identity>
      <Pwd>abcdef1234!</Pwd>
    </Psk>
  </Mqtt>
</RemoteConnections>
</TcConfig>
```

5. Die Einträge für das TwinCAT XAR sind fast identisch. Passen Sie lediglich die Werte der Felder <Identity> und <Pwd> entsprechend der Angaben in der ACL.xml an.

6. Nachdem auf beiden Systemen die Routes.xml fertig konfiguriert ist, initialisieren Sie jeweils den TwinCAT Router neu. Dazu schalten Sie vom RUN-Mode in den CONFIG-Mode oder vom CONFIG-Mode erneut in den CONFIG-Mode.

7. Abschließend prüfen Sie anhand der Ausgaben des Mosquitto Message Brokers, ob sich beide Systeme mit dem Broker verbinden können:

```

ca. Select Administrator: C:\Windows\system32\cmd.exe - mosquitto -p 8883 -v -c mosquitto_TLS.conf
172.17.214.125.1.1/ams/res', ... (56 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/
5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/172.17.214.125.1.1/ams/res', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/
172.17.214.125.1.1/ams/res', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/
5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/172.17.214.125.1.1/ams/res', ... (56 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/
172.17.214.125.1.1/ams/res', ... (56 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/
5.19.111.141.1.1/ams', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/172.17.214.125.1.1/ams/res', ... (44 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/
172.17.214.125.1.1/ams/res', ... (44 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/5.19.111.141.1.1/ams', ... (64 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/
5.19.111.141.1.1/ams', ... (64 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/172.17.214.125.1.1/ams/res', ... (36 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/
172.17.214.125.1.1/ams/res', ... (36 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/5.19.111.141.1.1/ams', ... (304 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/
5.19.111.141.1.1/ams', ... (304 bytes))
1506330088: Received PUBLISH from TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/5.19.111.141.1.1/ams', ... (64 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork1/
5.19.111.141.1.1/ams', ... (64 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/172.17.214.125.1.1/ams/res', ... (92 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/
172.17.214.125.1.1/ams/res', ... (92 bytes))
1506330088: Received PUBLISH from TwinCAT_XAR (d0, q0, r0, m0, 'VirtualAmsNetwork
k1/172.17.214.125.1.1/ams/res', ... (36 bytes))
1506330088: Sending PUBLISH to TwinCAT_XAE (d0, q0, r0, m0, 'VirtualAmsNetwork1/
172.17.214.125.1.1/ams/res', ... (36 bytes))

```

⇒ Damit ist ADS-over-MQTT mit TLS auf Basis von PSK für TwinCAT XAE und XAR erfolgreich eingerichtet.

### 3.1.4 Secure ADS

#### 3.1.4.1 Allgemeine Beschreibung

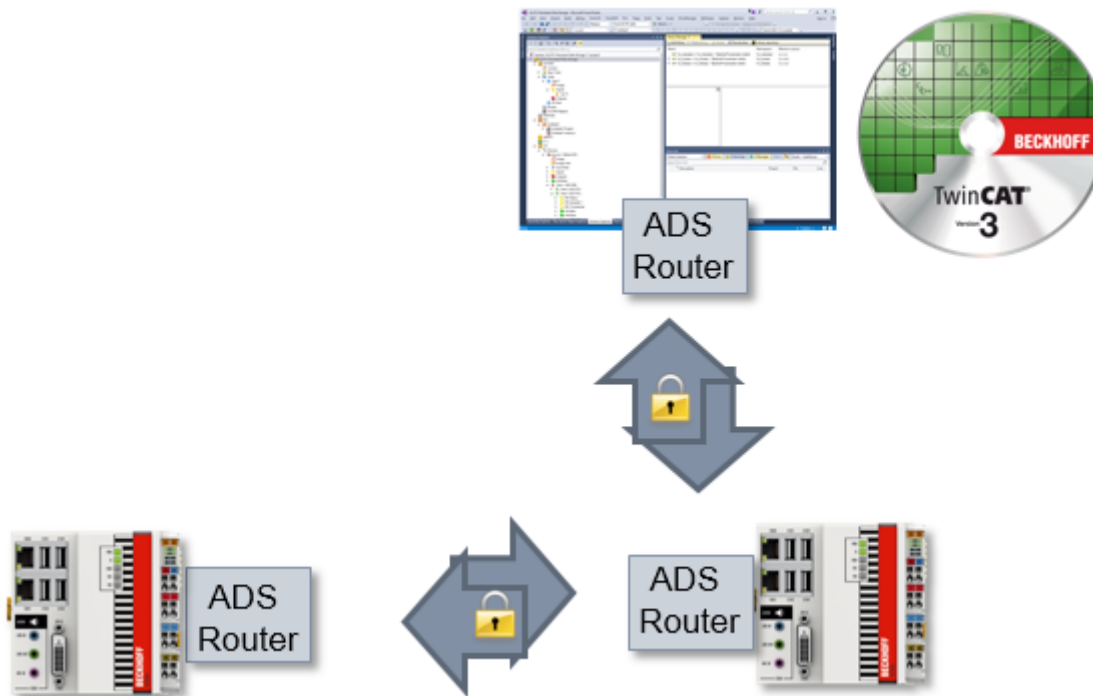
##### **i** Ab TwinCAT 3.1. Build 4024.0

Die hier beschriebene Funktionalität ist ab TwinCAT 3.1. 4024.0 verfügbar.

Secure ADS ist aus Sicht des ADS Protokolls ein weiterer Transportkanal. Es werden exakt die gleichen ADS Kommandos über eine sichere Verbindung übertragen, wie auch über andere Kommunikationsprotokolle.

Hierfür wird eine mittels TLSv1.2 verschlüsselte Verbindung von einem TwinCAT Router zu einem anderen TwinCAT Router aufgebaut.

Durch die Implementierung innerhalb des TwinCAT Routers müssen existierende Anwendungen nicht modifiziert werden. Sie können durch einfache Parametrierung der verwendeten Route dazu gebracht werden, die verschlüsselte Verbindung zu verwenden.




Diese Dokumentation stellt die unterschiedlichen Möglichkeiten von Secure ADS insbesondere im Blick auf die Bereitstellung der Schlüssel dar.


**Erkennen einer SecureADS Route**

TwinCAT stellt eine SecureADS Route durch ein Schloss dar.

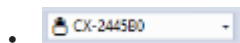
Dieses wird an den entsprechenden Stellen angezeigt:

- Routen-Übersicht eines Systems

 TwinCAT Static Routes

Route	Connected	AmsNetId	Address	Type
CX-2445B0		5.36.69.176.1.1	CX-2445B0	TCP_IP

- Bei der Auswahl des Zielsystems bei der Engineering-Umgebung XAE:



**3.1.4.2 Limitierungen**

**● Ab TwinCAT 3.1. Build 4024.0**

**i** Die hier beschriebene Funktionalität ist ab TwinCAT 3.1. 4024.0 verfügbar.

- Secure ADS ist nur zwischen ADS Routern verfügbar.
- Für Secure ADS Verbindungen gilt, wie für alle anderen ADS Verbindungen, dass sie einen Vollzugriff für die verbundenen Systeme darstellen, wie es auch im [Security Advisory 2017-01](#) beschrieben ist. Durch [Unidirektionale \[▶ 205\]](#) ADS Routen ist dieser Zugriff pro System konfigurierbar.

### 3.1.4.3 Voraussetzungen

#### **i** Ab TwinCAT 3.1. Build 4024.0

Die hier beschriebene Funktionalität ist ab TwinCAT 3.1. 4024.0 verfügbar.

- Secure ADS ist Bestandteil von TC1000 und kann ohne Lizenzkosten genutzt werden.
- Die verwendeten Geräte benötigen eine Netzwerkkommunikation. Secure ADS wird über den TCP Port 8016 eingehend kommuniziert.
- Zur TLS-Verschlüsselung müssen ggf. entsprechende Zertifikate generiert und signiert werden.

### 3.1.4.4 Technische Einführung

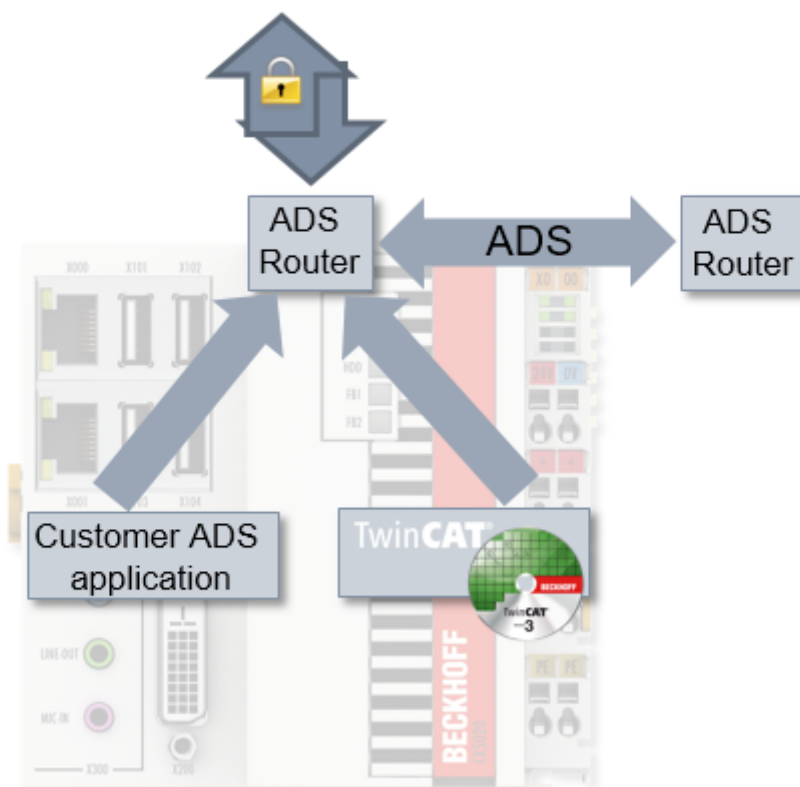
In diesem Abschnitt wird die grundsätzliche Funktionsweise unabhängig von der konkreten Konfiguration beschrieben.

Secure ADS führt einen zusätzlichen Kommunikationskanal für das bekannte ADS Protokoll ein. Diesen können Programme nutzen, ohne dass sie für den neuen Kommunikationskanal angepasst werden müssen.

Aus Security-Sicht handelt es sich also um eine Transportverschlüsselung aber keine Ende-zu-Ende-Verschlüsselung zwischen den Komponenten, denn alle lokal auf einem Gerät laufenden Anwendungen können diese verschlüsselte Verbindung gemeinsam nutzen – genau, wie es auch für ADS-Routen ist.

#### Lokale Realisierung

Secure ADS ist Bestandteil des ADS-Routers und wird auch hier konfiguriert. Der ADS Router baut eine verschlüsselte Verbindung zu einem anderen TwinCAT Router auf und stellt diese den Anwendungen bereit. Es ist also darauf zu achten, dass die ADS Geräte Anwendungen nicht selber verschlüsselt kommunizieren, sondern dies zwischen den Routern erfolgt.



## Transparente Nachrüstung

Durch die Realisierung von Secure ADS innerhalb des TwinCAT Routers wird ein nachträgliches „Retrofitting“ von Anwendungen ermöglicht. Alle ADS Anwendungen (Client und Server), dazu zählen auch Anwendungen die von Kunden geschrieben wurden, müssen nicht neu übersetzt werden.

Die ADS Anwendungen nutzen ADS Routen, um den Kommunikationspartner zu identifizieren. Diese ADS Route ist unabhängig vom Transportkanal und wird im TwinCAT Router beschrieben.

Wenn die genutzte Route auf eine Secure ADS Verbindung umgestellt wird, wird der ADS-Verkehr verschlüsselt transportiert.

### 3.1.4.4.1 Gerichtete ADS Kommunikation

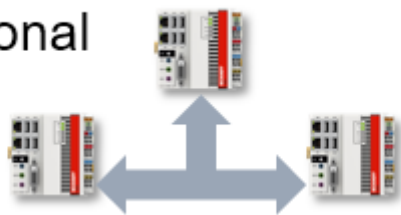
Eine Eigenschaft von ADS Routen ist, dass diese gerichtet sein können. Diese Eigenschaft wurde im Rahmen von Secure ADS ergänzt, ist jedoch allgemein für Routen verfügbar.

ADS Routen werden, nachdem sie auf Netzwerkebene geöffnet wurden, beidseitig von den jeweiligen ADS Anwendungen zur Kommunikation genutzt. Dieses Verhalten ist sehr effizient, aber jedoch ggf. unerwünscht. Beispielsweise soll ein Engineering-Rechner (XAE) im Normalfall zwar per ADS Zugriff auf ein Runtime (XAR) System haben, jedoch ist es nicht notwendig, dass ein XAR-System per ADS auf das XAE System zugreift.

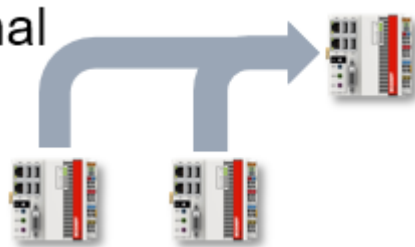
Diese Richtung kann deswegen eingeschränkt werden, welches dadurch realisiert wird, dass ein entsprechendes System (im Beispiel das XAE) keine ADS-Request Befehle über die Route akzeptiert.

Das Kapitel [Konfiguration](#) [► 207] beschreibt das Vorgehen, um die Eigenschaften einzuschränken.

#### Bidirectional



#### Unidirectional



### 3.1.4.4.2 Server

Eine normale ADS Route wird von beiden Teilnehmern aufgebaut, sobald diese benötigt wird.

Ist eine Route einmal aufgebaut, wird diese in beiden Richtungen genutzt.

Als Erweiterung für Secure ADS wird eine Server-Konfiguration angeboten. Eine solche Konfiguration stellt die Basis dar um konkrete Routen einzurichten.

```
<TcConfig>
  <RemoteConnections>
    <Server>
      ...
    </Server>
  </RemoteConnections>
</TcConfig>
```

Für [PSK](#) [► 209] sowie [Kunden-bereitgestellte Zertifikate](#) [► 210] wird dieses genutzt, um die initiale Konfiguration auf einer Seite abzulegen.

Beim Einrichten der konkreten Route werden dann die Server-Einträge überprüft, ob eine Berechtigung vorliegt. Wenn dieses der Fall ist, wird eine normale Route eingerichtet.

### 3.1.4.4.3 Schlüsselaustausch

Es werden drei Möglichkeiten durch Secure ADS angeboten um die zur Verschlüsselung nötigen Schlüssel bereitzustellen, welche hier mit ihren Vor- und Nachteilen beschrieben werden sollen.

Allen gemein ist, dass das jeweilige Gerät in Bezug auf den Zugriff auf die Geheimnisse (PreSharedKeys, Zertifikate) abgeschottet werden muss. Werden diese Geheimnisse kompromittiert, ist das System neu aufzusetzen um die Integrität des Gesamtsystems wieder herzustellen.

#### SelfSigned Certificates (SSC)

TwinCAT erzeugt beim ersten Starten (z.B. nach der Installation) ein selbst-signiertes Zertifikat.

Das Nutzen solcher Zertifikate hat den Vorteil, dass sie lokal erzeugt werden und bereitstehen. Um ein Vertrauensverhältnis aufzubauen muss jedoch zwischen allen Kommunikationsteilnehmern jeweils eine Überprüfung der Zertifikate durchgeführt werden.

Diese Zertifikate eignen sich also für die initiale Inbetriebnahme oder auch statische Maschinen, die ohne Dynamik in der Anlagenstruktur oder auch den Zugangsberechtigten auskommen.

Ab TwinCAT 4024.0 werden diese Zertifikate als Standard bei der Nutzung vorgesehen. Im Kapitel [Konfiguration](#) [▶ 208] wird beschrieben, wie sie zum Aufbau einer ADS-Route eingesetzt werden.

#### Laufzeiten der Zertifikate

Die erzeugten Zertifikate haben eine feste Laufzeit von 1.1.2000 bis 1.1.2061. Aus Security-Sicht ist dieses zu lang, sodass organisatorische Maßnahmen getroffen werden müssen um den Security-Ansprüchen zu genügen. Beckhoff stellt durch diese über-lange Laufzeit sicher, dass eine Kommunikation nicht fehlschlägt, auch wenn beispielsweise falsche Zeiten im lokalen System eingestellt sind.

Sollte dieses Verhalten unerwünscht sein, können eigene Zertifikate bereitgestellt und genutzt werden (vgl. Kunden-bereitgestellte Zertifikate).

#### PreSharedKeys (PSK)

In einem TwinCAT System können Pre-Shared-Keys abgelegt werden. Diese werden zur Autorisierung der eingehenden ADS-Routen beim Aufbau der Verbindung genutzt.

Da die Pre-Shared-Keys konfiguriert werden müssen, eignen sie sich insbesondere um z.B. Wartungspersonal Zugriff zu gewähren. Dabei können die Pre-Shared-Keys Personen-gebunden verwendet werden.

Pre-Shared-Keys haben keine Laufzeit, wie es für Zertifikate vorgesehen ist. Auch werden sie direkt in Dateien abgelegt, sodass sie nicht (wie normalerweise Passwörter) als Hashwert abgelegt werden. Sie sind damit gegen direkte Einsichtnahme nicht geschützt.

Im Kapitel [Konfiguration](#) [▶ 209] wird beschrieben, wie Pre-Shared-Keys auf beiden Seiten der Kommunikation verwendet werden.

#### Kunden-bereitgestellte Zertifikate (CA mit Zertifikaten)

Secure ADS stellt auch die Möglichkeit bereit, dass Kunden eigene Zertifikate erzeugen und verwalten.

Dadurch sind insbesondere dynamische Konstellationen gut abbildbar, da es eine gemeinsame Certificate Authority (CA) geben kann. Alle Teilnehmer, die dieser CA vertrauen, können ohne weitere Konfiguration verschlüsselt untereinander kommunizieren, selbst wenn sie sich zuvor nicht begegnet sind.

Im Kapitel [Konfiguration](#) [▶ 210] wird beschrieben, wie diese Zertifikate in TwinCAT eingebunden werden können.

### HINWEIS

#### Ablauf der Zertifikate

Zertifikate haben ein Ablaufdatum. Es müssen organisatorische Maßnahmen getroffen werden, um Zertifikate vor ihrem Ablauf zu ersetzen.



### 3.1.4.5 Konfiguration

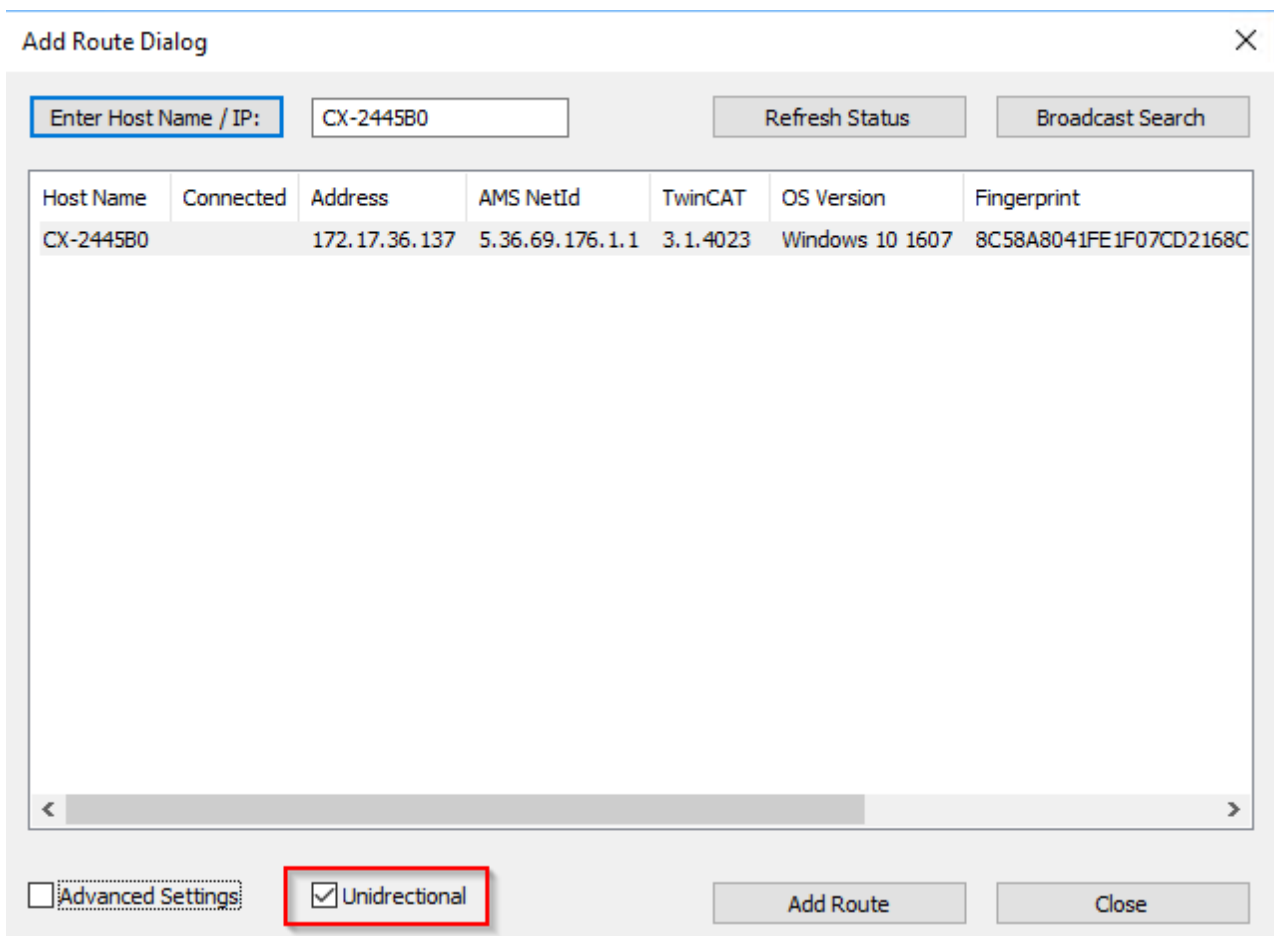
Es werden drei Möglichkeiten durch Secure ADS angeboten, um die zur Verschlüsselung notwendigen Schlüssel bereitzustellen. An dieser Stelle wird die Konfiguration getrennt voneinander beschrieben.

Während die Konfiguration Server vs. Route innerhalb der drei Möglichkeiten beschrieben wird, werden gerichtete ADS Verbindungen [► 207] unabhängig dargestellt.

#### 3.1.4.5.1 Gerichtete ADS Kommunikation

Die Konfiguration einer gerichteten ADS Kommunikation erfolgt über die Checkbox **Unidirectional** beim Anlegen der Route.

Ist diese Checkbox gesetzt, wird TwinCAT über die zugehörige Route keine ADS Befehl-Aufrufe vom gegenüberliegenden Zielsystem entgegennehmen. Selbst werden ADS Befehl-Aufrufe (Requests) gesendet und auch Antworten (Response) empfangen.



In der XML-Konfiguration erfolgt diese Einstellung über das Attribut `Unidirectional="true"` erfolgen:

```
<RemoteConnections>
<Route Unidirectional="true">
<Name>CX-123456</Name>
<Address>CX-123456</Address>
<NetId>5.36.69.176.1.1</NetId>
<Type>TCP_IP</Type>
<Flags>128</Flags>
<Tls>
...
</Tls>
</Route>
</RemoteConnections>
```

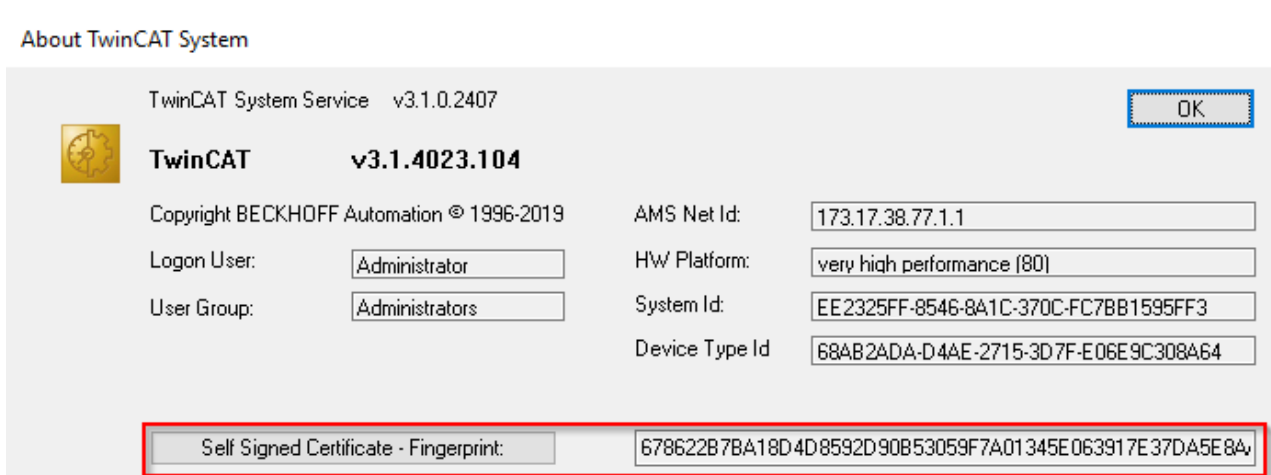
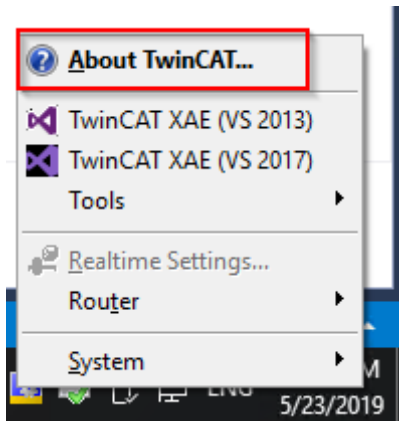
### 3.1.4.5.2 SelfSigned Zertifikate (SSC)

Selbst-Signierte Zertifikate benötigen beim Einrichten der Verbindung die Überprüfung des Kommunikationsteilnehmers, da automatisiert kein Vertrauensverhältnis existiert.

Diese Überprüfung wird in TwinCAT durch den Fingerprint des gegenüberliegenden Systems ermöglicht.

#### Anzeige des SSC-Fingerprints auf einem System

Der Fingerprint des eigenen Systems wird im **About TwinCAT** Dialog angezeigt:

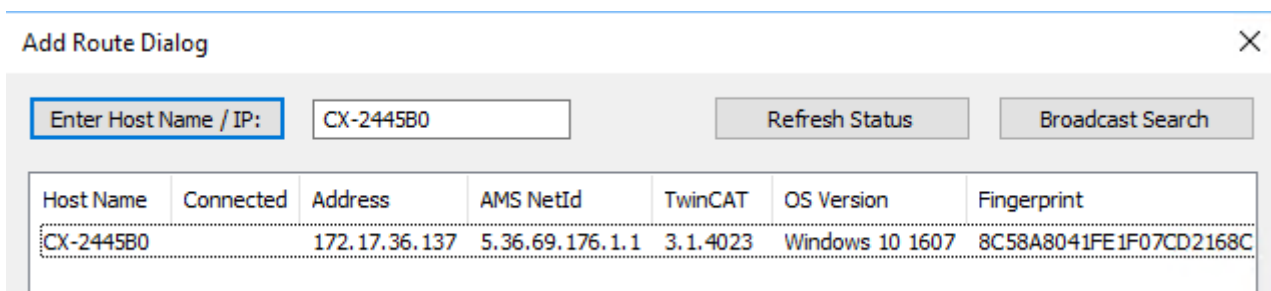


Der Button **Self Signed Certificate – Fingerprint:** kopiert den rechts aufgeführten Fingerprint in die Zwischenablage.

Für CE Systeme existiert dieser Dialog nicht. Hier kann in der Datei `\Hard Disk\TwinCAT\3.1\Target\TcSelfSigned.xml` der Fingerprint angezeigt werden.

#### Aufbau der Verbindung

Der Fingerprint wird rein informativ und kryptographisch nicht gesichert nach dem Discovery angezeigt:





Die endgültige Überprüfung des Fingerprints findet beim Einrichten der Route statt:

Das **Compare with** Feld kann dabei z. B. beim Copy&Paste zur Überprüfung verwendet werden: Wird dort der gleiche Fingerprint eingetragen, erscheint das Feld grün, sonst rot.

Damit kann beispielsweise eine RDP-Verbindung genutzt werden, um den Fingerprint eines Systems über den **Self Signed Certificate – Fingerprint**-Button in die Zwischenablage zu bekommen und hier einzufügen.

Damit das Zielsystem den Routenaufbau akzeptiert, wird ein dort gültiger System-Login mit entsprechenden Administrator-Rechten genutzt.

Diese Login-Daten werden bereits verschlüsselt übertragen.

Bei CE-Systemen wird mit TwinCAT 3.1 4024.5 immer der Hostname eingetragen, auch wenn beim Anlegen der Route **IP-Adresse** ausgewählt wurde. Sollte also ein Netzwerk ohne funktionierenden Hostname-Lookup genutzt werden, muss in der Datei `\Hard Disk\TwinCAT\3.1\Target\StaticRoutes.xml` manuell der Hostname durch die IP-Adresse geändert werden.

### 3.1.4.5.3 Preshared Keys (PSK)

Pre-Shared-Keys werden auf einer Seite als Server eingerichtet und auf einer Seite zur Authentisierung und Autorisierung genutzt.

#### Einrichten von Preshared Keys als Server

Preshared Keys werden im Normalfall mit Server-Verbindungen eingesetzt werden. Die Konfiguration erfolgt über einen Eintrag in der Routen-Konfiguration.

Hierfür können in der `C:\TwinCAT\3.x\Target\StaticRoutes.xml` Datei folgende Einträge vorgenommen werden:

```
<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<RemoteConnections>
<Server>
<Tls>
<Psk>
<Identity>MY_IDENTITY</Identity>
<Pwd>MySecret</Pwd>
</Psk>
<Psk>
<Identity>MY_IDENTITY2</Identity>
<Pwd>MyOtherSecret</Pwd>
</Psk>
```

```

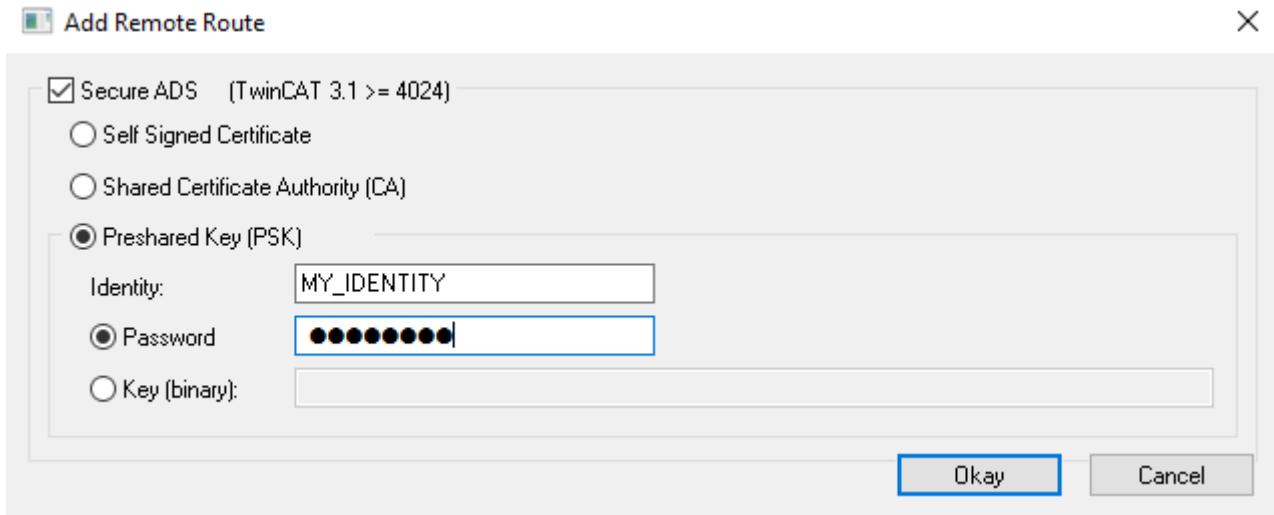
</Tls>
</Server>
</RemoteConnections>
</TcConfig>

```

Gespeicherte Änderungen werden übernommen, wenn der TwinCAT Router initialisiert wird, was beispielsweise bei den Übergängen von RUN->CONFIG oder auch CONFIG->CONFIG erfolgt.

### Nutzen einer Preshared Keys Servers

Beim Hinzufügen einer Route wird der Eintrag **Preshared Key (PSK)** ausgewählt und die entsprechenden Credentials eingetragen.



Wenn dieses erfolgreich ist, wird eine konkrete Route im Zielsystem hinterlegt, welche für zukünftige Verbindungsaufbauten genutzt wird.

### 3.1.4.5.4 Kunden-bereitgestellte Zertifikate (CA mit Zertifikaten)

Die Konfiguration von Kunden-bereitgestellten Zertifikaten erfolgt über einen Eintrag in der Routen-Konfiguration.

Hierfür können in der `C:\TwinCAT\3.x\Target\StaticRoutes.xml` Datei folgende Einträge vorgenommen werden:

```

<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<RemoteConnections>
<Server>
<Tls IgnoreCn="true"> <!--see below-->
  <Ca>C:\TwinCAT\3.1\Target\CACerts\rootCA.pem</Ca>
  <Cert>C:\TwinCAT\3.1\Target\CACerts\ipc.crt</Cert>
  <Key>C:\TwinCAT\3.1\Target\CACerts\ipc.key</Key>
</Tls>
</Server>
</RemoteConnections>
</TcConfig>

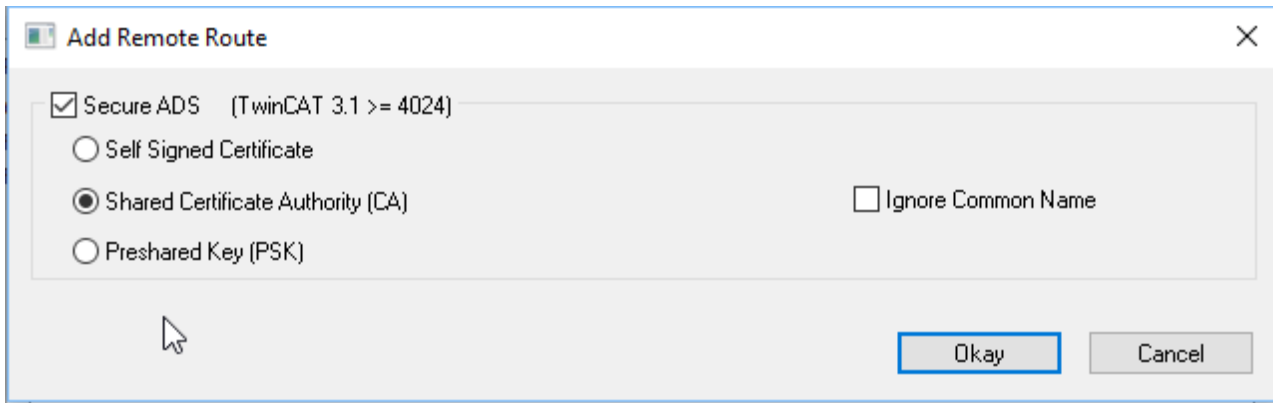
```

Gespeicherte Änderungen werden übernommen, wenn der TwinCAT Router initialisiert wird, was beispielsweise bei den Übergängen von RUN->CONFIG oder auch CONFIG->CONFIG erfolgt.

Die Zertifikate sind dabei X.509 Zertifikate, wie sie beispielsweise mit OpenSSL generiert werden können. Sollte der Schlüssel (XML-Element `<Key>`) durch ein Passwort geschützt sein, kann dieser über das XML-Element `<KeyPwd>` angegeben werden. Es wird das .der und .pem Format unterstützt.

Der „CommonName“ des Zertifikates muss dabei dem beim Verbindungsaufbau genutzten Namen (XML-Element `<Name>`) entsprechen. Dieses Verhalten kann durch die Option `IgnoreCn="true"` abgeschaltet werden.

Wenn beide Seiten passende Zertifikate einer gemeinsamen CA haben, kann die Route ohne weitere Informationen mittels diesen Dialogs Angelegt werden:



Wie unter [Server \[► 205\]](#) beschrieben, wird hierdurch auf beiden Seiten eine konkrete Route angelegt.

### 3.1.4.5.5 Abschalten von ADS

- Das unverschlüsselte ADS wird über den TCP Port 48898 (0xBF02) übertragen
- Das Discovery („Broadcast Search“) wird über den UDP Port 48899 (0xBF03) übertragen

Beide Ports können in der Firewall blockiert werden.

Das Zielsystem kann in Bezug auf die zu nutzenden Ports konfiguriert werden.

Unterhalb von `KEY_LOCAL_MACHINE\SOFTWARE\[WOW6432Node]Beckhoff\TwinCAT3\System` sind die folgenden Schlüssel verfügbar:

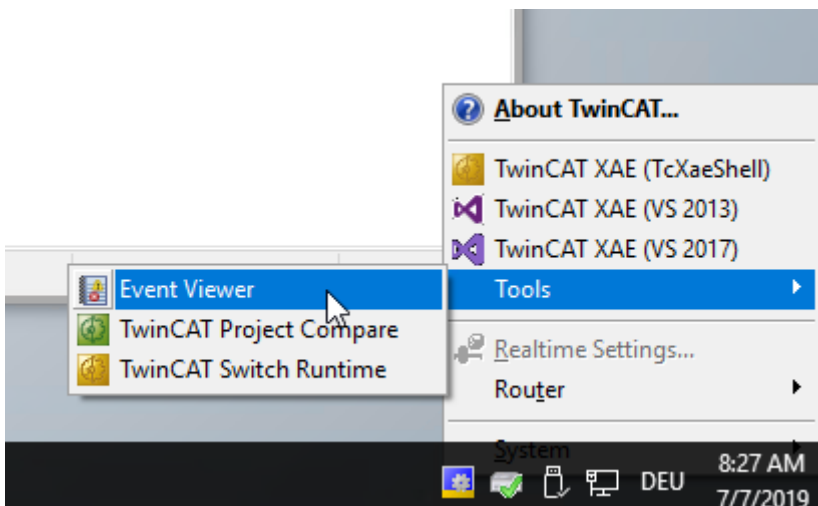
ADS Ports		
DisableAdsTcpListening	REG_DWORD	1 = Verhindert das Öffnen des TCP Ports 0xBF02 für unverschlüsseltes ADS.
DisableAdsTlsListening	REG_DWORD	1 = Verhindert das Öffnen des TCP Ports 8016 für Secure ADS
DisableAdsDiscovery	REG_DWORD	1 = Verhindert das Öffnen des UDP Ports 0xBF03 für das ADS Discovery („Broadcast Search“)

Über die Datei `StaticRoutes.xml` kann zusätzlich das Attribut `SecureOnly="True"` verwendet werden. Der ADS Port 0xBF02 wird dabei weiterhin geöffnet, jedoch wird über den Port keine ADS Kommunikation mehr erlaubt.

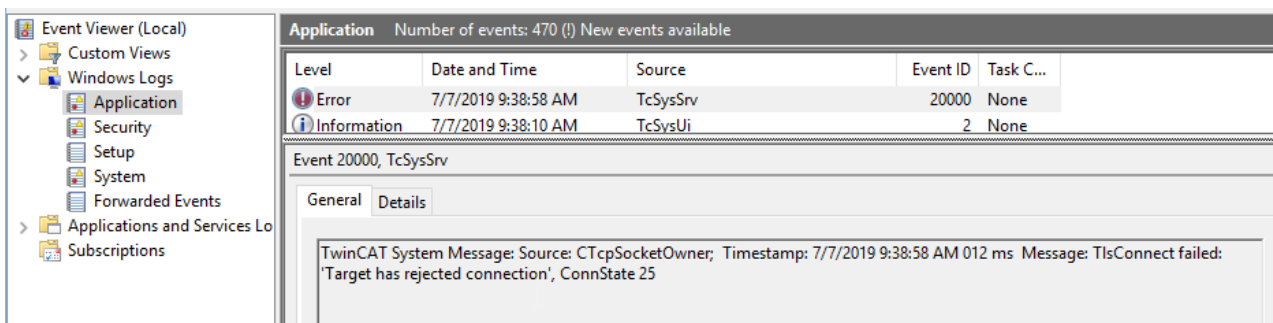
```
<RemoteConnections SecureOnly="True">
```

### 3.1.4.5.6 Logging

Secure ADS schreibt Informationen über fehlgeschlagene Verbindungsaufbauten in das Windows Event Log, welches über das TwinCAT System Tray Icon verfügbar ist.



Die Meldungen befinden sich unter der Kategorie **Windows Logs > Application**:



### 3.1.4.6 Beispiel

#### 3.1.4.6.1 Kunden-bereitgestellte Zertifikate (CA mit Zertifikaten)

An dieser Stelle wird mittels OpenSSL Zertifikate erzeugt, die für die Secure ADS Verbindung genutzt werden können.

Diese Anleitung stellt keine umfassende Beratung zur Erstellung und Umgang mit Zertifikaten dar. Insbesondere die Laufzeiten müssen beachtet werden, welches organisatorische Maßnahmen erfordert um vor Ablauf der Gültigkeiten (hier: 3600 Tage für die CA und 360 Tage für die jeweiligen Zertifikate) für einen Austausch zu sorgen.

In diesem Beispiel wird eine Certificate Authority (CA) erzeugt, die für beide Seiten (hier IPC und CX genannt) der Kommunikation ein Zertifikat unterschreibt.

Die Bedeutung der Aufrufparameter kann im Detail durch „openssl help“ nachgesehen werden.

✓ OpenSSL ist installiert und als verfügbar von der Kommandozeile.

1. Erzeugen eines Schlüssels für die Certificate Authority, der später vertraut wird.

```
openssl genrsa -out rootCA.key 2048
```

2. Erzeugen des Zertifikats mit einer Laufzeit von 3600 Tagen. Über den Parameter „-subj“ werden Inhaberinformationen beigesteuert.

```
openssl req -x509 -new -nodes -key rootCA.key -sha256 -subj "/C=DE/ST=NRW/L=Verl/O=Bk/OU=TCPM/CN=RootCA" -days 3600 -out rootCA.pem
```

3. Erzeugen eines Schlüssels für den IPC

```
openssl genrsa -out ipc.key 2048
```

4. Erzeugen eines Certificate Signing Requests (CSR) für diesen Schlüssel:  
Bitte beachten: Die als CN angegebene Adresse (hier IP Adresse) muss als Namen beim Verbindungsaufbau verwendet werden. Alternativ muss die Route mit IgnoreCN parametrieren werden.  

```
openssl req -out ipc.csr -key ipc.key -subj "/C=DE/ST=NRW/L=Verl/O=Bk/OU=TCPM/CN=192.168.2.1" -new
```
5. Signieren des CSR des IPC mit der CA mit Gültigkeit von 360 Tagen  

```
openssl x509 -req -in ipc.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out ipc.crt -days 360 -sha256
```

⇒ Mittels dieser Dateien kann die Route nun auf dem IPC eingerichtet werden: rootCA.pem, ipc.key und ipc.pem
6. Erzeugen eines Schlüssels für den CX  

```
openssl genrsa -out cx.key 2048
```
7. Erzeugen eines Certificate Signing Requests (CSR) für diesen Schlüssel:  
Bitte beachten: Die als CN angegebene Adresse (hier IP Adresse) muss als Namen beim Verbindungsaufbau verwendet werden. Alternativ muss die Route mit IgnoreCN parametrieren werden.  

```
openssl req -out cx.csr -key cx.key -subj "/C=DE/ST=NRW/L=Verl/O=Bk/OU=TCPM/CN=192.168.2.2" -new
```
8. Signieren des CSR des IPC mit der CA mit Gültigkeit von 360 Tagen  

```
openssl x509 -req -in cx.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out cx.crt -days 360 -sha256
```

⇒ Mittels dieser Dateien kann die Route nun auf dem CX eingerichtet werden: rootCA.pem, cx.key und cx.pem

⇒ Die Route kann genutzt werden.

## 3.2 Ordner und Dateitypen

### 3.2.1 Dateien TwinCAT-SPS-Projekt

#### 3.2.1.1 Port\_xxx.app

Binärdatei des SPS-Projekts

**Ablageort**

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Bootprojekt aktivieren</li> <li>• SPS einloggen mit Update Bootprojekt</li> </ul>
<b>Voraussetzung</b>	-	-

#### 3.2.1.2 Port\_xxx.autostart

Leere Datei, die die Option Autostart aktiviert

**Ablageort**

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Option Autostart aktivieren (projektunabhängige Systemeinstellung)</li> </ul>
<b>Voraussetzung</b>	-	

### 3.2.1.3 Port\_xxx.cid

Datei, die die Compileinfo\_IDs enthält

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Bootprojekt aktivieren</li> <li>• SPS einloggen mit Update Bootprojekt</li> </ul>
<b>Voraussetzung</b>	-	-

### 3.2.1.4 Port\_xxx.crc

Datei, die die Prüfsumme des SPS-Projekts enthält

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Bootprojekt aktivieren</li> <li>• SPS einloggen mit Update Bootprojekt</li> </ul>
<b>Voraussetzung</b>	-	-

### 3.2.1.5 Port\_xxx.occ

Symbolik des SPS-Projekts

- Die Datei enthält die Änderungen der Symbolik des SPS-Projekts für einen Online-Change.
- Wenn die Option **Symbolic Mapping** nicht aktiviert ist, enthält diese Datei auch die Änderungen der Mapping-Konfiguration für ein Aktivieren/Update Bootprojekt.
- Beim Aktivieren der Konfiguration wird die occ-Datei in beiden Verzeichnissen zurückgesetzt.

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Bootprojekt aktivieren</li> <li>• SPS einloggen mit Update Bootprojekt</li> </ul>
<b>Voraussetzung</b>	-	-

### 3.2.1.6 Port\_xxx.oce

Die Datei enthält die Änderungen der Eventklassen zum Zeitpunkt eines OnlineChanges, welche in einem SPS-Projekt verwendet werden.

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	-	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	-	Bei Änderung der genutzten Eventklassen und OnlineChange
<b>Voraussetzung</b>	-	-

### 3.2.1.7 Port\_xxx.ocm

Beschreibungsdatei der Mapping-Konfiguration

- Wenn die Option **Symbolic Mapping** aktiviert ist, enthält die Datei die Änderungen der Mapping-Konfiguration des SPS-Projekts für ein Aktivieren/Update Bootprojekt.
- Beim Aktivieren der Konfiguration wird die ocm-Datei in beiden Verzeichnissen zurückgesetzt.

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• TwinCAT-Projekt erstellen</li> <li>• TwinCAT-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Bootprojekt aktivieren</li> <li>• SPS einloggen mit Update Bootprojekt</li> </ul>
<b>Voraussetzung</b>	-	-

### 3.2.1.8 Port\_xxx\_boot.tizip

Archivordner, der die COMPILEINFO-Datei des Bootprojekts enthält

Die COMPILEINFO-Datei enthält die Kompilier-Informationen und die Login-Information des SPS-Projekts.

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Bootprojekt aktivieren</li> <li>• SPS einloggen mit Update Bootprojekt</li> </ul>
<b>Voraussetzung</b>	-	-

### 3.2.1.9 Port\_xxx\_act.tizip

Archivordner, der die COMPILEINFO-Datei des aktuell laufenden SPS-Projekts enthält

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	-	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	-	<ul style="list-style-type: none"> <li>• SPS einloggen mit Änderung</li> </ul>
<b>Voraussetzung</b>	-	-

### 3.2.1.10 Port\_xxx.bootdata

Bootdatei, die die persistenten Daten speichert

Nachdem das TwinCAT-System gestartet und die SPS geladen worden ist, wird die Dateiergung .bootdata in .bootdata-old umbenannt.

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	-	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	-	<ul style="list-style-type: none"> <li>• TwinCAT-System stoppen</li> <li>• Verwendung von FB_WritePersistentData</li> </ul>
<b>Voraussetzung</b>	-	-

### 3.2.1.11 Port\_xxx.bootdata-old

Backup-Datei für die persistenten Daten

Die Datei wird gelöscht, nachdem die neue Bootdatei vollständig geschrieben worden ist.

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	-	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	-	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• TwinCAT-System neustarten</li> </ul>
<b>Voraussetzung</b>	-	-

### 3.2.1.12 PLC\_Name.tpzip

Archivordner des SPS-Projekts

Der Umfang des Inhalts kann in den Projekteigenschaften konfiguriert werden.

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\CurrentConfig\	C:\TwinCAT\3.1\Boot\CurrentConfig\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Bootprojekt aktivieren</li> <li>• SPS einloggen mit Update Bootprojekt</li> </ul>
<b>Voraussetzung</b>	-	-



**3.2.1.13 PLC\_Name.tmc**

TC3-Modul-Beschreibungsdatei

**Ablageort**

	<b>Projektverzeichnis</b>	<b>TwinCAT-Bootverzeichnis</b>
<b>Pfad</b>	A)..\<<Solution name>\<Project name>\<PLC name>\ B)..\<<Solution name>\<Projektname>\_Boot\<Platform>\Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Bootprojekt aktivieren</li> <li>• SPS einloggen mit Update Bootprojekt</li> </ul>
<b>Voraussetzung</b>	A) - B) TMC als Target File aktiviert	<ul style="list-style-type: none"> <li>• TMC als Target File aktiviert</li> </ul>

**3.2.1.14 PLC\_Name.tpy**

TC2-SPS-Beschreibungsdatei

**Ablageort**

	<b>Projektverzeichnis</b>	<b>TwinCAT-Bootverzeichnis</b>
<b>Pfad</b>	A)..\<<Solution name>\<Project name>\<PLC name>\ B)..\<<Solution name>\<Project name>\_Boot\<Platform>\Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Bootprojekt aktivieren</li> <li>• SPS einloggen mit Update Bootprojekt</li> </ul>
<b>Voraussetzung</b>	A) - B) TPY als Target File aktiviert	<ul style="list-style-type: none"> <li>• TPY als Target File aktiviert</li> </ul>

### **3.2.2 Dateien TwinCAT-C++-Projekt**

Datei	Beschreibung	Weitere Informationen
<b>Engineering / XAE</b>		
*.sln	Visual Studio Solution-Datei, beherbergt TwinCAT- und Nicht-TwinCAT-Projekte	
*.tsproj	TwinCAT Projekt, Sammlung aller verschachtelten TwinCAT-Projekte, wie TwinCAT C++ oder TwinCAT SPS-Projekt	
_Config/	Ordner enthält weitere Konfigurationsdateien (*.xti), die zum TwinCAT-Projekt gehören.	Siehe Menü Tools  Options  TwinCAT  XAE-Environment  File Settings
_Deployment/	Ordner für kompilierte TwinCAT C++ Treiber	
*.tmc	TwinCAT Module Class Datei (XML-basiert)	Siehe TwinCAT Module Class Editor (TMC)
*.rc	Ressourcendatei	Siehe Siehe Version/ Herstellerinformation
*.vcxproj.*	Visual Studio C++ Projektdateien	
*ClassFactory.cpp/.h	Class Factory für diesen TwinCAT Treiber	
*Ctrl.cpp/.h	Treiber hochladen und entfernen für TwinCAT UM Plattform	
*Driver.cpp/.h	Treiber hochladen und entfernen für TwinCAT RT Plattform	
*Interfaces.cpp/.h	Deklaration der TwinCAT COM Schnittstellenklassen	
*W32.cpp/.def/.idl		
*.cpp/.h	Eine C++/Header-Datei pro TwinCAT Modul im Treiber. Benutzercode hier einfügen.	
Resource.h	Wird von *.rc Datei benötigt	
TcPch.cpp/.h	Wird für die Erstellung von vorkompiliertem Header verwendet	
%TC_INSTALLPATH%\Repository\<Vendor>\<PrjName>\<Version>\<Platform>\*.tmx	Kompilierter Treiber, der über das TcLoader geladen wird. <i>C:\TwinCAT\3.x\Repository\C++ Module Vendor\Untitled1\0.0.0.1\TwinCAT RT *Untitled1.tmx</i>	Siehe Versionierte C++ Projekte
%TC_INSTALLPATH%\CustomConfig\Modules\*	Veröffentlichtes TwinCAT Treiberpaket normalerweise C: <i>\TwinCAT\3.x\CustomConfig\Modules\*</i>	Siehe Module exportieren
<b>Laufzeit / XAR</b>		
%TC_BOOTPRJPATH%\CurrentConfig\*	Derzeitiges Konfigurationssetup <i>Windows: C:\TwinCAT\3.x\Boot TwinCAT/BSD: /usr/local/etc/TwinCAT/3.x/Boot</i>	
%TC_DRIVERAUTOINSTALLPATH%\*.sys/pdb	Kompilierter, plattformspezifischer Treiber, der über das Betriebssystem geladen wird. <i>Windows: C:\TwinCAT\3.x\Driver\AutoInstall (System geladen) TwinCAT/BSD: &lt;nicht verfügbar&gt;</i>	
%TC_INSTALLPATH%\Boot\Repository\<Vendor>\<PrjName>\<Version>\*.tmx	Kompilierter Plattform-spezifischer Treiber, der über den TcLoader geladen wird. <i>Windows: C:\TwinCAT\3.x\Boot\Repository\C++ Module Vendor\Untitled1\0.0.0.1\Untitled1.tmx TwinCAT/BSD: /usr/local/etc/TwinCAT/3.x/Boot\Repository\C++ Module Vendor\Untitled1\0.0.0.1\Untitled1.tmx</i>	

Datei	Beschreibung	Weitere Informationen
%TC_BOOTPRJPATH%\TM\OBJECTID.tmi	TwinCAT Module Instance Datei Beschreibt Variablen des Treibers Dateiname lautet <i>ObjectID.tmi</i> Windows: C:\TwinCAT\3.x\Boot\TM\OTCID.tmi TwinCAT/BSD: /usr/local/etc/TwinCAT/3.x/Boot/TMI/OTCID.tmi	
<b>Temporäre Dateien</b>		
*.sdf	IntelliSense Datenbank	
*.suo / *.v12.suo	Benutzerspezifische und Visual Studio-spezifische Dateien	
*.tsproj.bak	Automatisch generierte Sicherungsdatei von <i>tsproj</i>	
ipch/	Für vorkompilierten Header erstelltes Zwischen-Verzeichnis	

### 3.2.3 Dateien TwinCAT-Projekt

#### 3.2.3.1 CurrentConfig.xml

Beschreibungsdatei der aktuellen Konfiguration.

##### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\	C:\TwinCAT\3.1\Boot\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>TwinCAT-Projekt erstellen</li> <li>TwinCAT-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>Konfiguration aktivieren</li> </ul>
<b>Voraussetzung</b>	-	-

#### 3.2.3.2 CurrentConfig.tzip

Archivordner, der die tsproj-Datei sowie alle referenzierten xti-Dateien enthält.

##### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\	C:\TwinCAT\3.1\Boot\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>TwinCAT-Projekt erstellen</li> <li>TwinCAT-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>Konfiguration aktivieren</li> </ul>
<b>Voraussetzung</b>	<ul style="list-style-type: none"> <li>Auto Save &lt;TwinCAT-Projektname&gt; to Target as Archive ist aktiv</li> </ul>	

### 3.2.4 Dateien PLC HMI

#### 3.2.4.1 Port\_xxx.textlistname.txt

Für jede im Projekt vorhandene Textliste wird eine Datei angelegt, die alle Einträge dieser Textliste enthält.

##### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\Port_xxx\Visu	C:\TwinCAT\3.1\Boot\Plc\Port_xxx\Visu
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Online-Change / Download</li> </ul>
<b>Voraussetzung</b>	<ul style="list-style-type: none"> <li>• Target- und/oder WebVisualisierungs-Objekt hinzugefügt</li> </ul>	

### 3.2.4.2 Port\_xxx Folder

In diesem Ordner wird automatisch ein weiterer Ordner „Visu“ angelegt, in dem wiederum die Dateien sowie die Bilder der PLC HMI gespeichert werden.

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\ Plc\	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> </ul>
<b>Voraussetzung</b>	<ul style="list-style-type: none"> <li>• Target- und/oder WebVisualisierungs-Objekt hinzugefügt</li> </ul>	

## 3.2.5 Dateien PLC HMI (Target Visualisierung)

### 3.2.5.1 tc3plchmi.ini

Konfigurationsdatei, die die Einstellungen des TargetVisualisierungs-Client enthält

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\ Plc	C:\TwinCAT\3.1\Boot\Plc\
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Online-Change / Download</li> </ul>
<b>Voraussetzung</b>	<ul style="list-style-type: none"> <li>• TargetVisualisierungs-Objekt hinzugefügt</li> </ul>	

## 3.2.6 Dateien PLC HMI Web

### 3.2.6.1 port\_xxx.imagepoolcollection.csv

Datei, die eine Auflistung der Einträge aller im SPS-Projekt verfügbaren Bildersammlungen (ImagePools) enthält

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\Port_xxx\Visu	C:\TwinCAT\3.1\Boot\Plc\Port_xxx\Visu
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Online-Change / Download</li> </ul>
<b>Voraussetzung</b>	<ul style="list-style-type: none"> <li>• WebVisualisierungs-Objekt hinzugefügt</li> </ul>	

### 3.2.6.2 webvisu.cfg.json

Konfigurationsdatei, die die Einstellungen des WebVisualisierungs-Objekts enthält

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\Port_xxx\Visu	C:\TwinCAT\3.1\Boot\Plc\Port_xxx\Visu
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Online-Change / Download</li> </ul>
<b>Voraussetzung</b>	• WebVisualisierungs-Objekt hinzugefügt	

### 3.2.6.3 webvisu.htm

HTML-Seite, die zum Anzeigen der Visualisierung im Internet-Browser verwendet wird

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\Port_xxx\Visu	C:\TwinCAT\3.1\Boot\Plc\Port_xxx\Visu
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Online-Change / Download</li> </ul>
<b>Voraussetzung</b>	• WebVisualisierungs-Objekt hinzugefügt	

### 3.2.6.4 webvisu.js

Datei, die die JavaScript-Logik enthält, die in der Visualisierung verwendet wird

#### Ablageort

	Projektverzeichnis	TwinCAT-Bootverzeichnis
<b>Pfad</b>	..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\Port_xxx\Visu	C:\TwinCAT\3.1\Boot\Plc\Port_xxx\Visu
<b>Entstehungszeitpunkt</b>	<ul style="list-style-type: none"> <li>• SPS-Projekt erstellen</li> <li>• SPS-Projekt neu erstellen</li> </ul>	<ul style="list-style-type: none"> <li>• Konfiguration aktivieren</li> <li>• Online-Change / Download</li> </ul>
<b>Voraussetzung</b>	• WebVisualisierungs-Objekt hinzugefügt	

## 3.3 Maschinen-Update auf Dateiebene

### 3.3.1 Übersicht

Wenn keine TwinCAT-3-Entwicklungsumgebung (XAE) zur Verfügung steht, können Sie die Bootdaten eines TwinCAT-SPS-Systems oder eines gesamten TwinCAT-Systems mittels einer Dateikopie aktualisieren.

- [SPS-Update durchführen \[► 223\]](#)
- [C++-Update durchführen \[► 223\]](#)
- [Gesamtes Maschinen-Update durchführen \[► 223\]](#)
- [Clonen einer Maschine \[► 224\]](#)

Eine Beschreibung der verschiedenen Dateien sowie Informationen zu deren Ablageort innerhalb des zugehörigen Projekts (Projektverzeichnis) und auf der Maschine (TwinCAT-Bootverzeichnis) finden Sie im Abschnitt [Ordner- und Dateitypen \[► 213\]](#).

### 3.3.2 SPS-Update durchführen

- ✓ TwinCAT-Version TC3.1.4022.0 oder höher
  - ✓ Für die Plattform der Maschine sind mittels (Neu-) Erstellen des SPS-Projekts Bootdaten erzeugt worden. Beim (Neu-) Erstellen des Projekts muss keine Verbindung zum Zielsystem bestehen.
  - ✓ Das Prozessabbild und die Hardware-Konfiguration haben sich seit der letzten Aktualisierung nicht geändert.
  - ✓ In den SPS-Projekteinstellungen ist die Option **Symbolic Mapping** aktiviert.
1. Kopieren Sie die Bootdaten des SPS-Projekts, d. h. alle Dateien und Ordner, aus dem Ordner `..\<Solution name>\<Project name>\_Boot\<Platform>\Plc\`.
  2. Ersetzen Sie die Bootdaten im TwinCAT-SPS-Bootverzeichnis `C:\TwinCAT\3.1\Boot\Plc\` der Maschine durch die kopierten Bootdaten.
  3. Starten Sie das TwinCAT-System der Maschine neu.
- ⇒ Die Bootdaten des TwinCAT-SPS-Systems und somit die SPS-Laufzeit selbst sind aktualisiert.

#### ● Update des Quellcodes

**I** Falls Sie zusätzlich zu den Boot-Daten auch den Quellcode des SPS-Projektes auf das Laufzeitsystem spielen, dann können Sie bei einem Update auf Dateiebene auch den Archivordner aus dem Ordner `..\<Solution name>\<Project name>\_Boot\<Platform>\CurrentConfig\` in den Ordner `C:\TwinCAT\3.1\Boot\CurrentConfig\` auf dem Laufzeitsystem kopieren.

### 3.3.3 C++-Update durchführen

Die Laufzeitdateien können per Dateikopie von einer zur anderen Maschine übertragen werden, wenn sie von derselben Plattform stammen und mit äquivalenter Hardware-Ausrüstung verbunden sind.

Die folgenden Schritte beschreiben ein einfaches Verfahren zum Übertragen einer Binärkonfiguration von einer Maschine „Quelle“ zu einer anderen Maschine „Ziel“.

- ✓ Leeren Sie den Ordner `C:\TwinCAT\3.x\Boot` auf der Quell-Maschine.
1. Erstellen (oder aktivieren) Sie das Modul auf der Quellmaschine.
  2. Übertragen Sie den Ordner `C:\TwinCAT\3.x\Boot` von der Quelle zum Ziel.  
Dieser Ordner enthält auch das Repository, welches die nötigen TMX Dateien enthält.
  3. Bei TwinCAT-Treiber-Projekten (.sys): Übertragen Sie den Treiber selber von `C:\TwinCAT\3.x\Driver\AutoInstall\MYDRIVER.sys` und ggf auch PDB-Datei.
  4. Bei TwinCAT-Treiber-Projekten (.sys) und falls Treiber neu auf einer Maschine sind:  
TwinCAT muss einmalig eine Registrierung ausführen. Schalten Sie TwinCAT dafür per SysTray (Rechts-Klick->**System**->**Start/Restart**) in den RUN-Modus.  
Alternativ kann dieser Aufruf verwendet werden („%1“ als Treibernamen ersetzen):  

```
sc create %1 binPath= c:\TwinCAT\3.1\Driver\AutoInstall\%1.sys type= kernel
start= auto group= "file system" DisplayName= %1 error= normal
```
- ⇒ Sie können nun die Zielmaschine starten.

#### ● Umgang mit Lizenzen

**I** Beachten Sie, dass die Lizenzen nicht auf diese Weise übertragen werden können. Verwenden Sie bitte vorinstallierte Lizenzen, Volumenlizenzen oder andere Mechanismen, um Lizenzen bereitzustellen.

### 3.3.4 Gesamtes Maschinen-Update durchführen

- ✓ Für die Plattform der Maschine sind mittels (Neu-) Erstellen des TwinCAT-Projekts Bootdaten erzeugt worden. Beim (Neu-) Erstellen des Projekts muss keine Verbindung zum Zielsystem bestehen.

- ✓ Die reale Hardware-Konfiguration entspricht der Projektkonfiguration.
  - ✓ Wenn das Maschinen-Update auf mehreren Maschinen durchgeführt werden soll und nicht auf einer spezifischen Maschine, sind die folgenden Optionen aktiviert:  
**Use Relative NetIds** in den Routes-Einstellungen (System > Routes, Registerkarte NetIdManagement) und  
**Virtual Device Names** in den Adapter-Einstellungen aller Netzwerk- und USB-Geräte (z. B. I/O > Devices > EtherCAT-Master, Registerkarte Adapter)  
 Die Netzwerk-Adapternamen der Maschine müssen mit dem Adapternamen der Konfiguration übereinstimmen.
1. Kopieren Sie die Bootdaten des TwinCAT-Projekts, d. h. alle Dateien und Ordner, aus dem Ordner `..\<Solution name>\<Project name>\_Boot\<Platform>\`.
  2. Ersetzen Sie die Bootdaten im TwinCAT-Bootverzeichnis `C:\TwinCAT\3.1\Boot\` der Maschine durch die kopierten Bootdaten.
  3. Falls Sie C++-Module verwenden, bitte die C++-Treiber kopieren (beschreiben im Kapitel [\(C++-Update durchführen \[► 223\]\)](#)).
  4. Starten Sie das TwinCAT-System der Maschine neu.
- ⇒ Die Bootdaten des TwinCAT-Systems und somit das TwinCAT-System selbst sind aktualisiert.

### 3.3.5 Clonen einer Maschine

Um die Bootdaten eines TwinCAT- oder SPS-Projekts von einer Maschine auf eine andere Maschine zu übertragen, kopieren Sie die Bootdaten aus dem Bootverzeichnis der einen Maschinen und ersetzen Sie die Bootdaten im Bootverzeichnis der anderen Maschine.

Wenn sich das TwinCAT-System, dessen Bootdaten kopiert werden sollen, im Run-Modus befindet und auch persistente Daten ausgetauscht werden sollen, schalten Sie das TwinCAT-System zunächst vom Run-Modus in den Config-Modus, sodass die persistenten Daten in der Datei `.bootdata` gespeichert werden und im Bootverzeichnis zum Kopieren zur Verfügung stehen. (Siehe `Port_xxx.bootdata`)

## 3.4 Programm automatisch starten

TwinCAT 3 bietet die Möglichkeit, nach dem Start ausgewählte Programme automatisch zu starten. Dies ist vor allem für Programme nützlich, bei denen TwinCAT vor der Ausführung gestartet sein muss, z. B. Visualisierungssoftware.

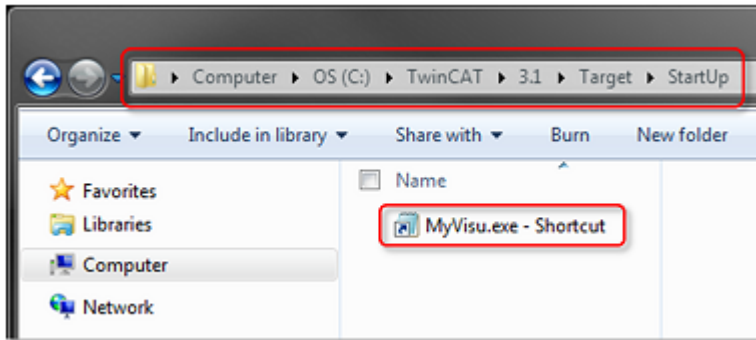
Um ein Programm nach dem Start von TwinCAT automatisch zu starten, muss in einem speziellen Startup-Ordner im TwinCAT-Verzeichnis eine Verknüpfung des Programms angelegt werden. Das Programm selbst muss lokal auf dem gleichen PC wie TwinCAT installiert sein. Nach der ersten Aktivierung des Run Mode nach dem Start des TwinCAT-Laufzeitsystems werden die Verknüpfungen im Startup-Ordner ausgeführt.

Der Pfad `<TwinCAT>\3.x\Target\StartUp` führt zum Startup-Ordner. Die Benennung ergibt sich wie folgt:

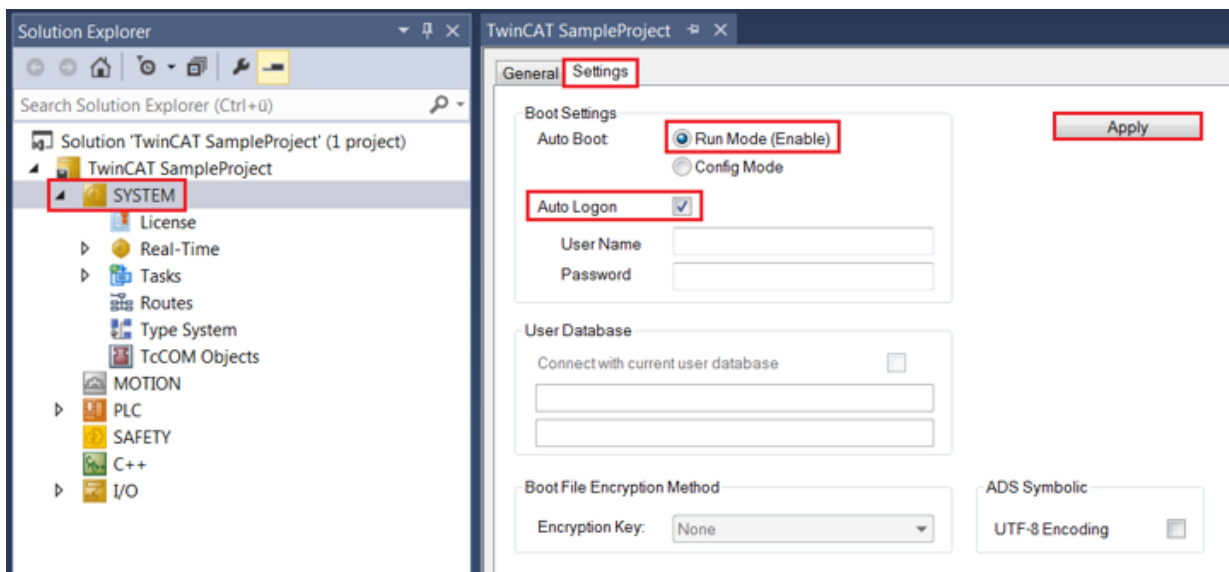
<code>&lt;TwinCAT&gt;</code>	Installationsordner von TwinCAT (Standard: <code>\TwinCAT</code> )
<code>3.x</code>	TwinCAT-Version (alle Versionen von TwinCAT werden in separaten Ordnern im Installationsordner von TwinCAT abgelegt).
<code>x</code>	Platzhalter für das Build von TwinCAT, z. B. "3.1".



1. Speichern Sie eine Verknüpfung zum Programm im Ordner <TwinCAT>\3.x\Target\StartUp.



2. Stellen Sie sicher, dass TwinCAT im Run Mode startet.
3. Klicken Sie im TwinCAT-Projektbaum doppelt auf **SYSTEM** und wählen Sie die Registerkarte **Settings**.



4. Aktivieren Sie die Optionen **Run Mode (Enable)** und **Auto Logon**.
5. Klicken Sie auf **Apply**.

## 3.5 Korrigierte Zeitstempel

### 3.5.1 Übersicht

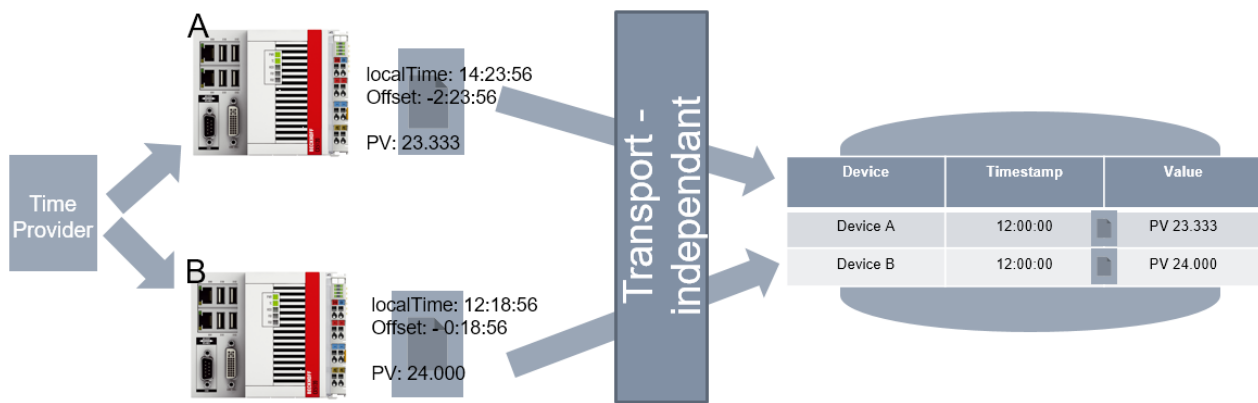
Steuerungen erzeugen Daten, welche in modernen, verteilten Anlagen gesammelt und in Verbindung gebracht werden sollen.

Da Steuerungen erstmal alleinstehende Geräte sind, besitzen diese auch unabhängige Zeitbasen. In einer gemeinsamen Datenbank würden Daten damit nicht in zeitliche Korrelation gebracht werden können.

Um diesem Problem zu begegnen ist es seit langer Zeit möglich Steuerungen untereinander zu synchronisieren, dazu dient beispielsweise das Netzwerkprotokoll IEEE1588, bzw. PTP.

In vielen Szenarien reicht es jedoch aus, die Daten mit einem einheitlichen Zeitstempel zu versehen, die Steuerungen jedoch unabhängig voneinander zu betreiben, sodass zum einen die mit den oben genannten Protokollen verbundenen Hardwarekosten reduziert werden, zum anderen aber auch keine technische Abhängigkeit zwischen den Steuerungen existiert.

In diesem Kapitel werden deswegen die TwinCAT Komponenten zur Adaption von Zeitstempeln für das Speichern von zeitsynchronen Daten beschrieben.



Die Abbildung verdeutlicht die grundlegende Idee: Unabhängige Steuerungen beziehen den lokalen Zeitstempel und passen diesen mittels eines Offset an, welcher dann für die Speicherung der gemeinsamen Daten verwendet wird.

Hierfür steht in der TwinCAT Echtzeit eine zentrale Komponente bereit, das „External Time Interface“. Diese Komponente

- erhält von einer konfigurierten Quelle („External Time Provider“) den Offset zur korrigierten Uhrzeit.
- stellt in Abhängigkeit von der aktuellen, lokalen Zeit eine korrigierte Uhrzeit dem „External Time Consumer“ bereit.

Diese korrigierte Uhrzeit kann nun von unterschiedlichen Komponenten innerhalb und außerhalb der Echtzeit verwendet werden.

Als Quelle dient typischerweise entweder ein NTP Server oder ein DC-Zeitsignal basierend auf EtherCAT und z. B. synchronisiert mittels EL6688 durch PTP (IEEE1588). Eine Quelle kann aber auch vom Kunden implementiert werden, sodass andere Zeitsignale als Quelle realisiert werden können.

Neben der oben beschriebenen zentralen Komponente in der TwinCAT Echtzeit umfasst das Konzept also zwei Arten von Komponenten:

1. External Time Provider: Stellen einen Offset für die Anpassung von Zeitstempeln der zentralen Komponente bereit.  
Ein Provider bezieht z. B. via NTP (Network Time Protocol, vgl. RFC 4330) einen Zeitstempel, woraus er im Vergleich zu der lokalen Systemzeit einen Offset berechnet und diesen bereitstellt.
2. External Time Consumer: Verwenden einen Offset, den sie aus der zentralen Komponente beziehen. Somit kann in den Komponenten ein Zeitstempel verwendet werden, der auf entfernten Geräten zu vergleichbaren Daten führt.  
Als Consumer kommen dabei alle TwinCAT Komponenten, welche Zeitstempel verwenden in Frage, aber auch Kundenanwendungen.

### 3.5.2 Systemvoraussetzungen

Technische Daten	Voraussetzung
Betriebssystem	Windows 7/10, Windows Embedded Standard 7
Zielplattform	PC-Architektur (x86, x64 )
Minimale TwinCAT-Version	TwinCAT 3.1 Build 4024.0 und höher
Erforderliches TwinCAT-Setup-Level	TwinCAT 3 XAE, XAR
Erforderliche TwinCAT-Lizenz	Beliebige Laufzeitlizenz (PLC, C++)

### 3.5.3 Limitierungen

Einige wichtige Einschränkungen sind zu berücksichtigen:

- Die TwinCAT System Zeit wird nicht verändert durch die hier beschriebene External Time Schnittstelle

- Die bereitgestellten External Time Offsets werden wie vom Provider bereitgestellt den Consumern bereitgestellt. Hieraus ergibt sich, dass
  - der Offset vom Provider korrekt berechnet werden muss.
  - keine Monotonie in den Zeitstempeln garantiert werden kann.
- Die External Time Offsets werden nicht gespeichert und nachträglich zum Abruf bereitgestellt. Es werden im TwinCAT System also nur die aktuellen Offsets verwaltet.

### 3.5.4 Technische Einführung

TwinCAT stellt sowohl auf Seite des External Time Provider wie auch des External Time Consumer unterschiedliche Schnittstellen bereit, um das Konzept der korrigierten Zeitstempel nutzbar zu machen.

Auf External Time Consumer-Seite sind unterschiedliche TwinCAT Komponenten in der Lage den externen Zeitstempel zu verwenden. Zusätzlich gibt es auch für Anwendungen unterschiedliche Zugriffsmöglichkeiten.

Auf External Time Provider-Seite werden Module bereitgestellt, die per NTP einen Offset berechnen und bereitstellen können. Zusätzlich gibt es ein Modul, welches per DC den Offset verwenden kann. Die entsprechende Schnittstelle zur Bereitstellung des Offsets wird für TwinCAT C++ ebenfalls angeboten, sodass eigene External Time Provider erstellt werden können.

#### Zeitstempel für unterschiedliche Anwendungsfälle

Wichtig ist hierbei, dass TwinCAT vier Typen von Zeitstempeln in diesem Konzept unterscheidet:

1. None: Lokale Systemzeit und keine Korrektur
2. Soft: Empfohlene Verwendung z. B. für NTP
3. Medium: Empfohlene Verwendung z. B. für IEEE1588
4. Hard: Empfohlene Verwendung z. B. für Hardware-Synchronisierung, bei der kein Drift entstehen sollte

Ein External Time Provider stellt einen der möglichen Offsets bereit, es lässt sich nur ein Provider pro Typ definieren.

Ein External Time Consumer kann dann beliebigen Offset verwenden, es lassen sich alle vier Typen von Offset individuell verwenden. So ist es in unterschiedlichen Kompletten oder Betriebsarten möglich, unterschiedliche Zeitstempel zu verwenden. Beispielsweise kann eine lokale Diagnose auch mit der lokalen Systemzeit erfolgen, während gleichzeitig etwa aggregierte Daten von unterschiedlichen Systemen mit dem Offset vom Typ „Soft“ korrigiert und in einer gemeinsamen Datenbank abgelegt werden.

Die Schnittstellen der korrigierten Zeitstempel verwenden Datentypen mit der Länge 8 Byte und werden vom 1.1.1601 in 100ns gezählt.

#### 3.5.4.1 Consumer

External Time Consumer sind Komponenten, die die lokal gültige Systemzeit durch einen Offset korrigieren können. Hierfür müssen die Komponenten einen Offset vom Typ Soft, Medium oder Hard auswählen bzw. konfigurieren und entsprechend abfragen.

##### 3.5.4.1.1 TwinCAT Komponenten als Offset Consumer

Die folgenden Komponenten von TwinCAT unterstützen den Ansatz der korrigierten Zeitstempel – die jeweilige Dokumentation beschreibt, wie diese Funktionalität aktiviert werden kann:

- TwinCAT 3 Eventlogger

Diese Liste wird erweitert.

##### 3.5.4.1.2 Anwendungs-Implementierung

Anwendungen können die External Time Offsets in unterschiedlichen Komponenten verwenden:

- Echtzeit - PLC: Die PLC kann einen Offset abfragen bzw. einen lokalen Zeitstempel entsprechend korrigieren lassen.
- Echtzeit - C++: C++ TcCOM Module sind in der Lage den Offset abzufragen und entsprechend zu handeln.
- User-Mode – ADS Device Notifications: Die mit den ADS Device Notifications versendeten Zeitstempel können korrigiert sein.
- User-Mode – ADS Read: Der korrigierte Zeitstempel kann durch ein ADS Read abgerufen werden. Dieses kann in ADS Summenkommandos verwendet werden um einen Zeitstempel zusammen mit Daten abzurufen.

Unter den entsprechenden API Kapiteln sind die Schnittstellen dokumentiert.

### 3.5.4.2 Provider

External Time Provider sind Komponenten, die einen External Time Offset in Bezug zur lokalen Systemzeit durch eine externe Informationsquelle bestimmen und in TwinCAT bereitstellen. Hierdurch können External Time Consumer unabhängig vom Provider eine korrigierte Uhrzeit erhalten.

TwinCAT liefert Provider mit:

- NTP Provider: Eine Implementierung, die per (S)NTP ein Zeitsignal von einem NTP-Server abfragt und bereitstellt.
- DC Provider: Eine Implementierung, die die DC Zeit aus dem EtherCAT Master (z.B. per IEEE1588 oder PTP) als Offset weiterreicht an TwinCAT
- Zusätzlich ist der Kunde in der Lage eigene Provider bereitzustellen.

#### 3.5.4.2.1 NTP Provider

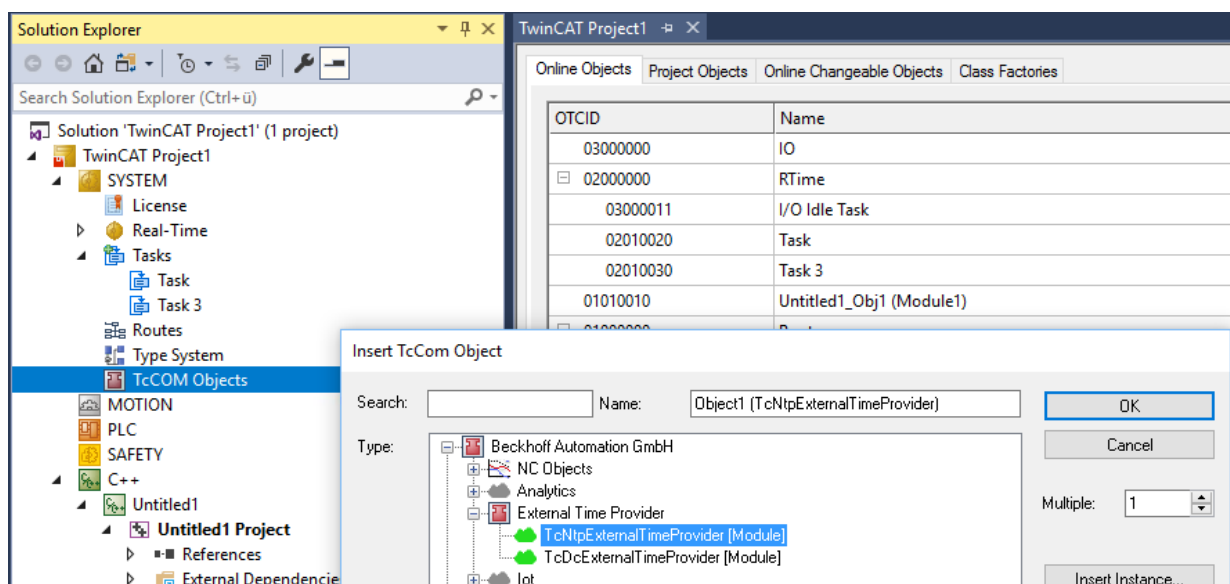
Der NTP Provider ist ein (S)NTP Client, der ein Zeitsignal von einem NTP Server zyklisch bezieht. Hierdurch kann er einen Offset der Systemzeit von dem Zeitsignal des NTP Servers berechnen und entsprechend bereitstellen.

#### Konfiguration

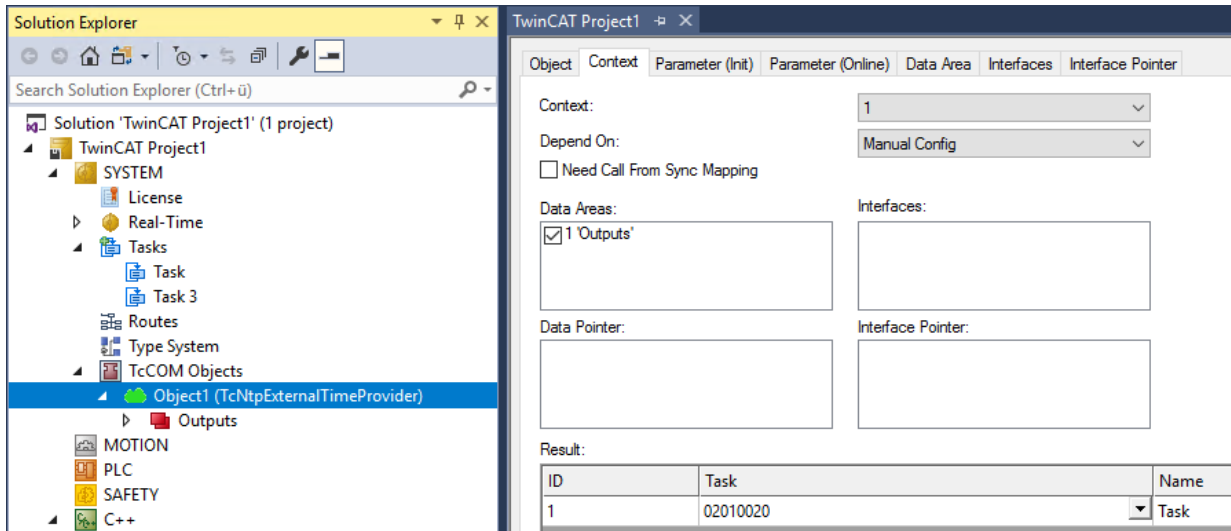
Der NTP Provider ist als TcCOM Modul „TcNtpExternalTimeProvider“ implementiert. Dieses Modul wird als TcCOM Modul folgendermaßen in Betrieb genommen:

✓ TwinCAT Projekt

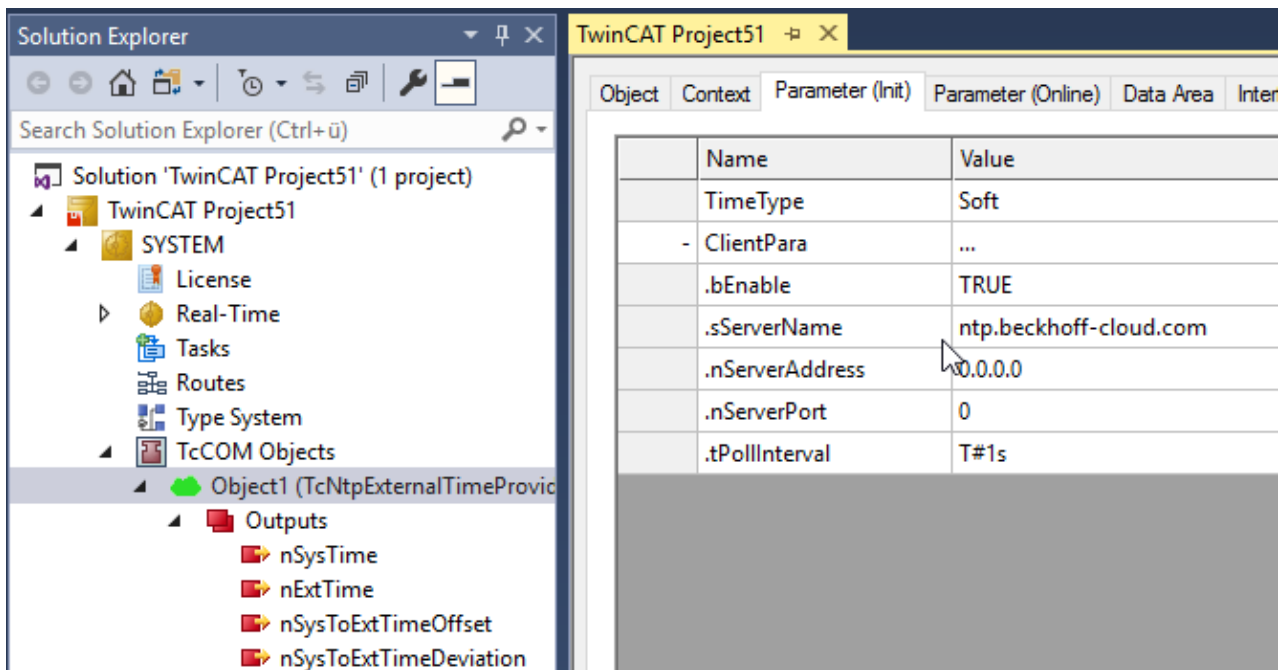
1. Einfügen eines TcCOM Moduls unter System->TcCOM Objects unter Auswahl des Typen TcNtpExternalTimeProvider in der Kategorie External Time Provider.



2. Das Modul benötigt eine Task, von dem es aufgerufen wird. Diese wird über den **Context**-Tab des Moduls parametrisiert:



⇒ Das TcCOM Modul kann parametrisiert werden:



Die Konfiguration erfolgt im Tab **Parameter (Init)**. Die Parameter bedeuten:

- **TimeTime**: Der Type des Offsets für den dieses Modul einen Offset ermitteln soll.

**Client Para:**

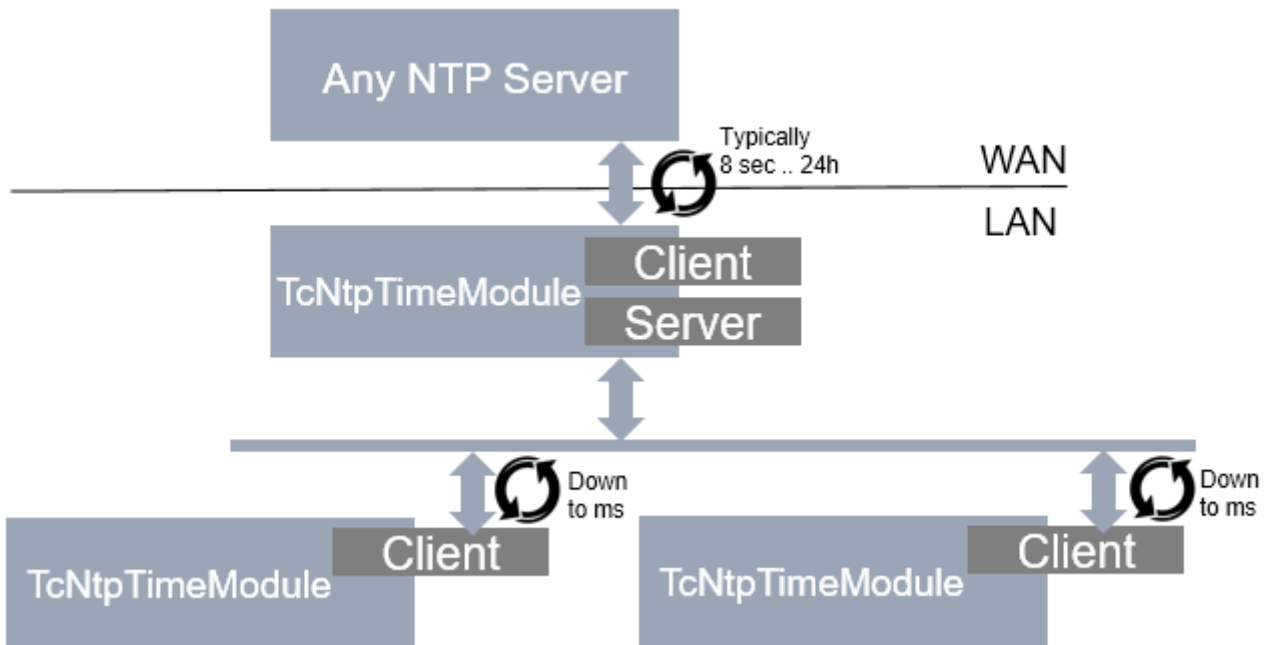
- **bEnable**: Zum Unterbinden der NTP Kommunikation kann das Modul deaktiviert werden.
- **sServerName**: Der Name des NTP Servers, der als Quelle genutzt werden soll.
- **nServerAddress**: Die IP Adresse eines NTP Servers (wird genutzt, falls sServerName leer ist).
- **nServerPort**: Der zu nutzende UDP Port des NTP Servers (default: 123).
- **tPollIntervall**: Das Intervall, in dem die NTP Abfragen gestartet werden sollen. Es wird dabei das vom Server angegebene Maximum berücksichtigt, sodass ggf. langsamere Anfragen entstehen.

Dieses Modul reicht einen ermittelten Offset zum einen über das [ITcSetExternalTime \[P\\_235\]](#) Interface zu TwinCAT. Zum anderen stehen auch Ausgänge für das Mapping bereit.

## NTP Provider als NTP Server

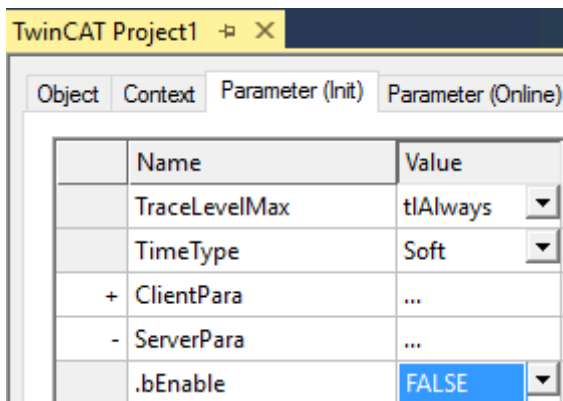
Optional kann das gleiche Modul auch als NTP Server fungieren. Somit kann ein Zeitsignal von einem externen NTP Server (als Client) bezogen werden und gleichzeitig an unterlagerte Systeme bereitgestellt werden.

Für den externen Server gilt nach NTP Protokoll typischerweise eine minimale Abfragezeit von 8 Sekunden oder mehr. Der NTP Provider als NTP Server lässt hingegen häufigere Abfrageintervalle zu.



## Server Funktion

Die Server-Funktionalität ist normalerweise ausgeblendet. Sie lässt sich über **Show Hidden Parameter** einblenden und konfigurieren:



- **bEnable:** Einschalten der NTP-Server Funktionalität für dieses Modul. Dafür muss der Port udp/123 in der Windows Firewall entsprechend geöffnet werden.
- **nPort:** Der UDP Port, der genutzt wird, um den Server anzubieten (default: 123).

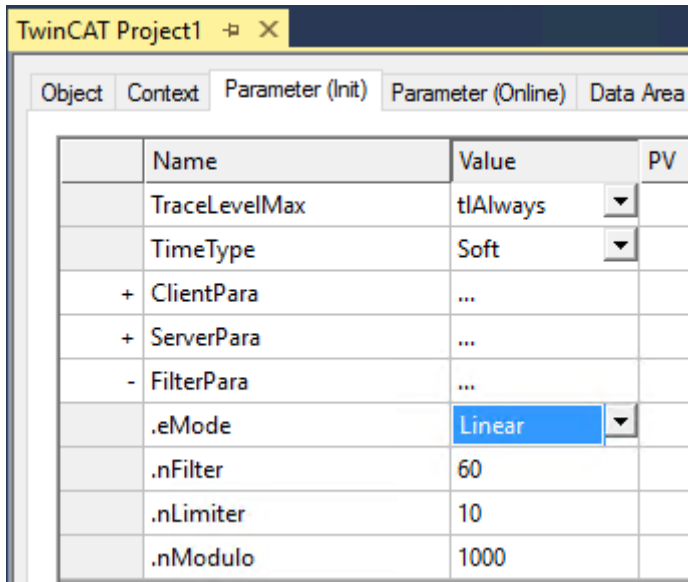
Die folgenden Parameter dienen dazu, die bereitgestellten NTP Informationen anzupassen. Per Default werden sie gesetzt wie im Protokoll vorgesehen, können hier jedoch überschrieben werden:

- **nLeap:** Manuelle Konfiguration des Leap Indicators.
- **nStratum:** Manuelle Konfiguration des Stratums.
- **nRoot:** Manuelle Konfiguration der Root-Server Information, wie sie in Abhängigkeit von Stratum definiert wird.

## Filter-Funktion

Werden Offsets durch die NTP Server-Abfrage ermittelt, kann das Modul selbstständig einen Übergang des alten Offsets zum neuen Offset durchführen.

Diese Funktionalität ist normalerweise ausgeblendet. Sie lässt sich über **Show Hidden Parameter** einblenden und konfigurieren:



- **eMode**: Eine Auswahl an Modes. Aktuell werden entweder keine oder eine lineare Anpassung (Default) vorgenommen.

Bei der Auswahl von „Linear“ als eMode gelten folgende Parameter:

- **nFilter**: Anzahl der Werte über die gemittelt wird, also Anzahl der NTP Responses. Bei einem PollInterval von 1s würde nFilter = 60 ein Filter über eine Minute bewirken. (Default: 60).
- **nLimiter**: Der Offset wird maximal um diesen Wert pro Zyklus verändert. Wenn die Differenz der lokalen zur externen Uhr 100ms und die Zykluszeit 1ms wäre, würde beim nLimiter = 10 es somit 100.000 Zyklen oder 1,6 Minuten dauern bis der Offset eingeregelt wurde. (Default 1µs).
- **nModulo**: Rundung des Offsets. Üblicherweise sollte dieses in Abhängigkeit von der Zykluszeit gewählt werden. Der Offset wird über diesen Modulo angepasst, damit keine „unrunden“ Zeiten entstehen. Die DC Time wird Modulo der Zykluszeit zurückgegeben, korrigiert mit dem Offset bleibt der Zeitstempel somit „rund“. Dadurch verändert sich der Offset/Zeitstempel aber auch mit kleinen Sprüngen, wenn eine Anpassung stattfindet. Wie unter nLimiter beschrieben und mit nModulo = 1000 würde sich der Offset und somit der relative Zeitstempel jeden 100ten Zyklus um 0,1ms inkrementieren.

## Diagnose

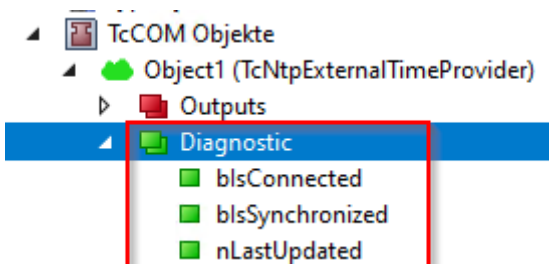
Unter dem Tab **Parameter (Online)** können entsprechende Diagnose-Informationen eingesehen werden.



Object	Context	Parameter (Init)	Parameter (Online)	Data Area	Interfaces	Interface Pointer
		Name	Online	CS	Unit	
-		ServerInfo	...	<input type="checkbox"/>		
		.nLeap	0			
		.nVersion	4			
		.nMode	4			
		.nStratum	2			
		.tPollIntv	T#1m4s			[ms]
		.fPollPrec	1e-07			[s]
		.fRootDelay	0.00047302968			[s]
		.fRootDisp	0.0029449912			[s]
		.sRefId	129.70.130.70			
		.nRefTime	2019-06-11T07:24:03.9653238			
		.nOrgTime	2019-06-11T07:24:05.1789999			
		.nRecTime	2019-06-11T07:24:05.4467752			
		.nTmtTime	2019-06-11T07:24:05.4468292			
		.nDstTime	2019-06-11T07:24:05.199			

Zu jeder Zeile befindet sich in der Spalte **Kommentar** eine entsprechende Beschreibung.

Zusätzlich sind zur programmatischen Auswertung entsprechende Symbole bereit:



### 3.5.4.2.2 DC Provider

Der DC Provider bezieht ein Offset durch das Mapping von einem EtherCAT Master. Es kann verwendet werden um aus dem I/O Bereich Zeitwerte als Offset zu verwenden, wie sie beispielsweise durch den EtherCAT Master (DC Zeit) oder eine EL6695 bereitgestellt werden.

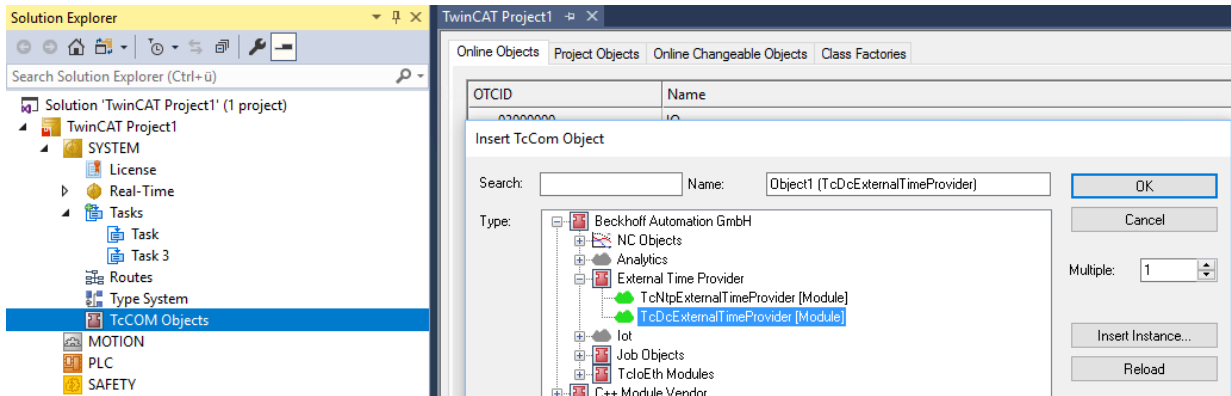
#### Konfiguration

Der DC Provider ist als TcCOM Modul „TcDcExternalTimeProvider“ implementiert. Dieses Modul wird als TcCOM Modul folgendermaßen in Betrieb genommen:

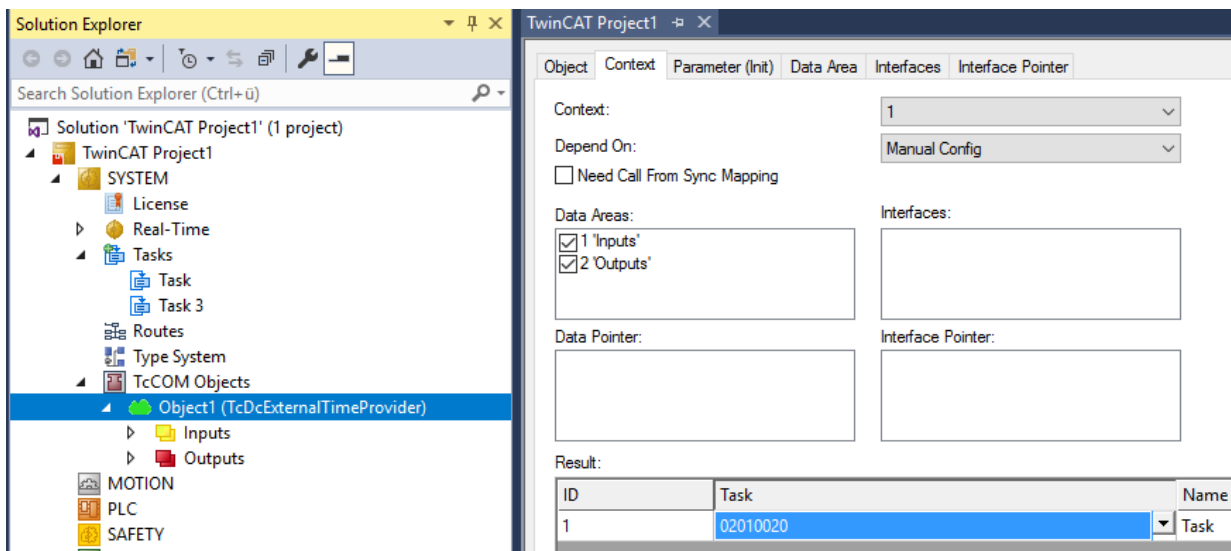
- ✓ TwinCAT Projekt



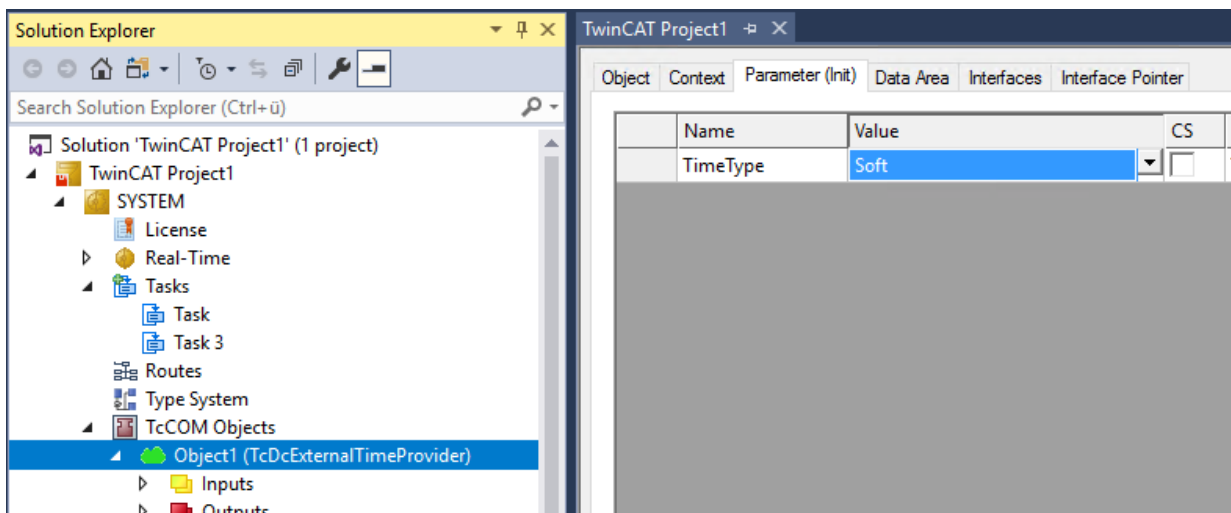
1. Einfügen eines TcCOM Moduls unter System->TcCOM Objects unter Auswahl des Typen TcDcExternalTimeProvider in der Kategorie External Time Provider.



2. Das Modul benötigt einen Task, von dem es aufgerufen wird. Dieser wird über den Context-Tab des Moduls parametrisiert:



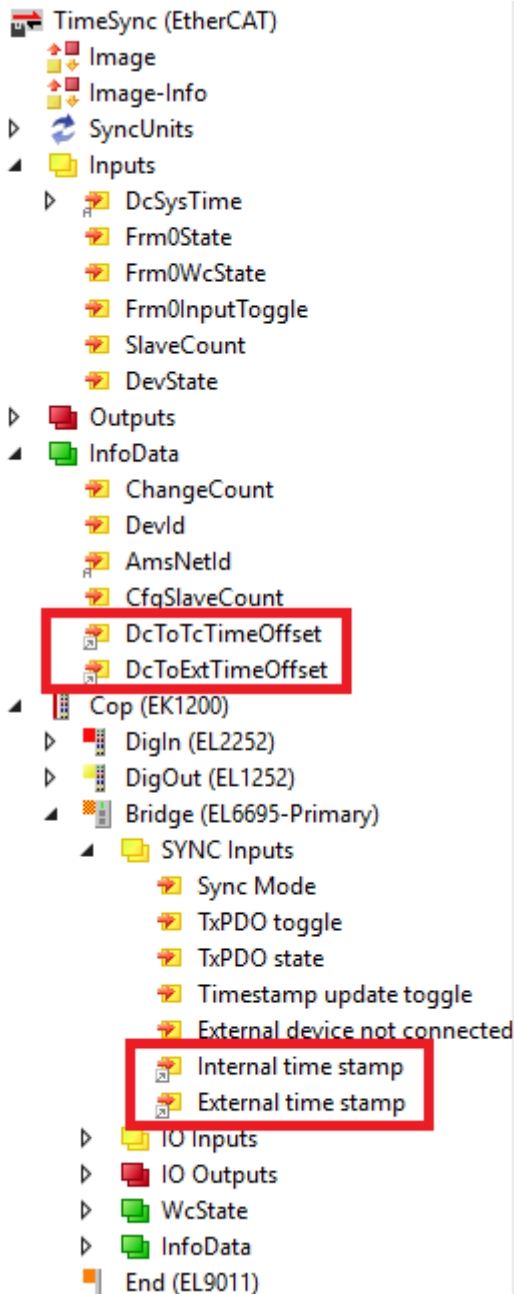
⇒ Das Modul kann parametrisiert werden:



Die Konfiguration erfolgt im Tab „Parameter (Init)“. Die Parameter bedeuten:

- TimeTime: Der Type des Offsets für den dieses Modul einen Offset ermitteln soll.

- Den Offset selber bezieht dieses Modul per Mapping:



Dieses Modul reicht einen ermittelten Offset zum einen über das `ITcSetExternalTime` [► 235] Interface zu TwinCAT. Zum anderen stehen auch Ausgänge für das Mapping bereit.

### 3.5.4.2.3 Anwendungs-Implementierung

Eine Anwendung kann eigene TimeOffset Provider bereitstellen, indem in TwinCAT C++ das Interface `ITcSetExternalTime` verwendet wird.

Dieses Modul stellt für die jeweiligen Offsets ggf. zyklisch einen Wert bereit.

#### Ablauf

Ein Modul realisiert den folgenden Ablauf

- ✓ Ein TcCOM Modul wurde instanziiert
- 1. Mittels `RegisterExternalTimeProvider` registriert sich das Modul als Provider eines bestimmten Typs von Offset (Soft/Medium/Hard)

- 2. Mittels SetExternalTimeOffset kann ggf. zyklisch ein Offset bereitgestellt werden
- 3. Mittels UnregisterExternalTimeProvider meldet sich das Modul entsprechend ab

Mittels der Registrierung wird sichergestellt, dass zu einem Zeitpunkt ein Offset von nur genau einem Modul genutzt werden kann.

Die genauere Beschreibung der Schnittstelle ITcSetExternalTime finden Sie im Kapitel [Schnittstelle ITcSetExternalTime](#) [▶ 235].

### 3.5.5 Realtime API

An dieser Stelle sind Schnittstellen und Strukturen dokumentiert um mit den korrigierten Zeitstempeln aus der Echtzeit umzugehen.

#### 3.5.5.1 Strukturen

##### 3.5.5.1.1 Enum TimeType

TwinCAT stellt vier unterschiedliche Zeitstempel bereit. Um diese zu unterscheiden wird das Enum TimeType verwendet.

##### Syntax

```
enum TimeType {
    SystemTime = 0,
    ExternalTimeHard = 1,
    ExternalTimeMedium = 2,
    ExternalTimeSoft = 3, // e.g. NTP
};
```

##### Werte

Die Verwendung der 3 externen Zeitstempel-Typen unterliegt der Anwendung; die unten beschriebene Verwendung ist lediglich ein Vorschlag.

Name	Beschreibung
ExternalTimeHard	Vorgeschlagene Verwendung bei harten Offsets, die keinen Drift haben
ExternalTimeMedium	Vorgeschlagene Verwendung bei genauen Offsets wie z.B. IEE1588
ExternalTimeSoft	Vorgeschlagene Verwendung bei allgemeinen Offsets, wie z.B. NTP

#### 3.5.5.2 Schnittstellen

An dieser Stelle werden die Schnittstellen beschrieben, welche für die korrigierten Zeitstempel verwendet werden.

Zu den unterschiedlichen Zeitformaten und Darstellungen befindet sich im C++ SDK eine entsprechende Liste.

Siehe: Infosys [C/C++](#)

##### 3.5.5.2.1 Schnittstelle ITcSetExternalTime

Die Schnittstelle ITcSetExternalTime wird vom TcCOM Object Server implementiert. Sie kann genutzt werden um einen extern ermittelten Offset bereitzustellen.

**Syntax**

```
TCOM_DECL_INTERFACE("00000067-0000-0000-e000-000000000064", ITcSetExternalTime)
struct __declspec(novtable) ITcSetExternalTime : public ITcExternalTime
```

**Methoden**

Name	Beschreibung
<a href="#">RegisterExternalTimeProvider</a> [▶ 236]	Einen Provider für einen Offset in Bezug auf TimeType anmelden
<a href="#">UnregisterExternalTimeProvider</a> [▶ 236]	Einen Provider für einen Offset in Bezug auf TimeType abmelden
<a href="#">SetExternalTimeOffset</a> [▶ 237]	Für den angemeldeten TimeType einen neuen Offset bereitstellen

**Anmerkungen**

Dieses Interface steht nicht für die SPS zur Verfügung.

**3.5.5.2.1.1 Methode RegisterExternalTimeProvider**

Einen Provider für einen Offset in Bezug auf TimeType anmelden

**Syntax**

```
HRESULT TCOMAPI RegisterExternalTimeProvider(OTCID oidProvider, TimeType type) = 0;
```

**Parameter**

**oidProvider:** (Typ: OTCID) Die ObjektID des Providers; normalerweise die des Aufrufenden

**type:** (Typ: [TimeType](#) [▶ 235]) Der zu registrierende TimeOffset Typ.

**Rückgabewert**

Typ: HRESULT

Informiert über den Erfolg der Registrierung

**Beschreibung****3.5.5.2.1.2 Methode UnregisterExternalTimeProvider**

Einen Provider für einen Offset in Bezug auf TimeType abmelden

**Syntax**

```
HRESULT TCOMAPI UnregisterExternalTimeProvider(OTCID oidProvider, TimeType type) = 0;
```

**Parameter**

**oidProvider:** (Typ: OTCID) Die ObjektID des Providers; normalerweise die des Aufrufenden

**type:** (Typ: [TimeType](#) [▶ 235]) Der abzumeldende TimeOffset Typ.

**Rückgabewert**

Typ: HRESULT

Informiert über den Erfolg der Abmeldung

**Beschreibung**

**3.5.5.2.1.3 Methode SetExternalTimeOffset**

Für den angemeldeten TimeType einen neuen Offset bereitstellen

**Syntax**

```
HRESULT TCOMAPI SetExternalTimeOffset(OTCID oidProvider, TimeType type, __int64 offset) = 0;
```

**Parameter**

**oidProvider:** (Typ: OTCID) Die ObjektID des Providers; normalerweise die des Aufrufenden

**type:** (Typ: TimeType [▶ 235]) Der TimeOffset Typ

**offset:** (Typ: \_\_int64) Der neu Offset-Wert.

**Rückgabewert**

Typ: HRESULT

Informiert über den Erfolg.

**Beschreibung**

Es gilt für den Offset ExternalTime = Internal Time + Offset. D.h. wenn die Zeit in TwinCAT in der Vergangenheit liegt, muss der Offset grösser 0 sein.

**3.5.5.2.2 Schnittstelle ITcExternalTime**

Die Schnittstelle ITcExternalTime wird vom TcCOM Object Server implementiert. Sie kann genutzt werden um einen extern ermittelten Offset abzurufen und zu verwenden.

**Syntax**

```
TCOM_DECL_INTERFACE("00000066-0000-0000-e000-000000000064", ITcExternalTime)
struct __declspec(novtable) ITcExternalTime : public ITcUnknown
```

 **Methoden**

Name	Beschreibung
<a href="#">SystemTimeToExternalTime [▶ 237]</a>	Berechnung eines korrigierten Zeitstempels in Bezug auf die Systemzeit
<a href="#">ExternalTimeToSystemTime [▶ 238]</a>	Berechnung der Systemzeit in Bezug auf einen korrigierten Zeitstempel
<a href="#">GetExternalTimeOffset [▶ 238]</a>	Abruf eines Offsets in Bezug auf den TimeType
<a href="#">GetExternalTimeProvider [▶ 238]</a>	Abfrage der ObjectID des aktuellen Providers

**3.5.5.2.2.1 Methode SystemTimeToExternalTime**

Berechnung eines korrigierten Zeitstempels in Bezug auf die Systemzeit

**Syntax**

```
HRESULT TCOMAPI SystemTimeToExternalTime(TimeType type, __int64& time) = 0;
```

**Parameter**

**type:** (Typ: TimeType [▶ 235]) Der zur Berechnung zu verwendende TimeOffset Typ

**time:** (Typ: \_\_int64&) Der um den Offset zu korrigierende Zeitstempel

**Rückgabewert**

Typ: HRESULT

Informiert über den Erfolg.

**Beschreibung****3.5.5.2.2 Methode ExternalTimeToSystemTime**

Berechnung der Systemzeit in Bezug auf einen korrigierten Zeitstempel

**Syntax**

```
HRESULT TCOMAPI ExternalTimeToSystemTime(TimeType type, __int64& time) = 0;
```

**Parameter**

**Typ:** (Typ: [TimeType](#) [[▶ 235](#)]) Der zur Berechnung zu verwendende TimeOffset Typ

**time:** (Typ: \_\_int64&) Der korrigierte Zeitstempel, welcher um den Offset bereinigt wird.

**Rückgabewert**

Typ: HRESULT

Informiert über den Erfolg.

**Beschreibung**

Es wird der zum Zeitpunkt des Aufrufes gültige Offset verwendet um die lokale Systemzeit zu ermitteln.

**3.5.5.2.3 Methode GetExternalTimeOffset**

Abruf eines Offsets in Bezug auf den TimeType

**Syntax**

```
HRESULT TCOMAPI GetExternalTimeOffset(TimeType type, __int64& offset) = 0;
```

**Parameter**

**type:** (Typ: [TimeType](#) [[▶ 235](#)]) Der abzurufende TimeOffset Typ

**offset:** (Typ: \_\_int64&) Der Wert, welcher auf den Offset gesetzt wird.

**Rückgabewert**

Typ: HRESULT

Informiert über den Erfolg.

**Beschreibung****3.5.5.2.4 Methode GetExternalTimeProvider**

Abfrage der ObjectID des aktuellen Providers

**Syntax**

```
HRESULT TCOMAPI GetExternalTimeProvider(TimeType type, OTCID& oidProvider) = 0;
```

**Parameter**

**type:** (Typ: [TimeType](#) [▶ 235]) Der TimeOffset Typ, dessen Provider abgefragt werden soll.

**oidProvider:** (Typ: OTCID&) Die ObjectID, welche auf die ObjectID des Provider gesetzt wird.

**Rückgabewert**

Typ: HRESULT

Informiert über den Erfolg.

**Beschreibung**

### 3.5.6 ADS API

Die TimeOffsets können auch über ADS abgefragt werden. Hierfür gibt es zwei Wege

1. ADS Notification: ADS Notifications beinhalten einen Zeitstempel, der den Änderungszeitpunkt der Daten beinhaltet.  
Ein ADS Client sendet dafür vor dem AddDeviceNotification ein ADS Kommando, wodurch das Zielsystem registriert, welcher Typ von korrigiertem Zeitstempel von diesem ADS Client gewünscht ist.
2. ADS Read: Per ADS Read kann ein korrigierter Zeitstempel ausgelesen werden. Dieses kann verwendet werden um in einem ADS Summen Kommando einen korrigierten Zeitstempel zu dem Zeitpunkt zu erhalten, an dem die ADS Kommandos ausgeführt wurden.

Index Group	Index Offset	Zugriff	Datentyp	Beschreibung	Anmerkung
<b>ADSIGRP_EXT ERNALTIME</b> 0xF088					
	ADSIOFFS_EX TERNALTIME_ SET  0x0000	R	LONG	Lesen des aktuell konfigurierten Offset-Typen für den jeweiligen ADS Client (AmsNetAddr inkl. Client-Port).	Rückgabewert ist Type 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ SET  0x00__	W		Setzen des Offset-Typen für die ADSDevice Notifications des jeweiligen ADS Clients (AmsNetAddr inkl. Client-Port).	__ ist Type 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ OFFSET  0x01__	R	LONGLONG	Lesen des aktuellen Offsets zu einem Typ.	__ ist Type: 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ OFFSET  0x01__	W	LONGLONG	Setzen des aktuellen Offsets zu einem Typ.	__ ist Type: 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ ABSOLUTE  0x02__	R	LONGLONG	Lesen des korrigierten Zeitstempels.	__ ist Type: 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ PROVIDER  0x03__	R	ULONG	Lesen der ObjektID von dem TimeOffset Provider.	__ ist Type: 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ SETALL  0x0400	R	LONG	Lesen des Typen, welcher verwendet wird, wenn kein anderer Typ gesetzt wurde.	Rückgabewert ist Type 0 = None, 1 = Hard, 2 = Medium, 3 = Soft
	ADSIOFFS_EX TERNALTIME_ SETALL  0x04__	W		Setzen des Typen, welcher verwendet wird, wenn kein anderer Typ gesetzt wurde.	__ ist Type 0 = None, 1 = Hard, 2 = Medium, 3 = Soft

Die Defines finden sich in der „Ads.h“ Datei.

Das [Beispiel ADS Consumer \[► 241\]](#) verdeutlicht die Verwendung.

### 3.5.7 Beispiele

Es werden verschiedene Beispiele zur Nutzung der Korrigierten Zeitstempel für den Anwender bereitgestellt:

- [PLC Consumer \[► 241\]](#): Ein PLC Programm greift auf korrigierte Zeitstempel zu.
- [C++ Consumer \[► 242\]](#): Ein C++ TcCOM Modul greift auf korrigierte Zeitstempel zu.
- [ADS Consumer \[► 241\]](#): Ein ADS Client im Usermode greift auf die korrigierten Zeitstempel zu.



- [C++ Provider \[► 242\]](#): Ein C++ TcCOM Modul ermittelt einen Offset und stellt diesen bereit.

Davon unabhängig werden die Korrigierten Zeitstempel von weiteren Komponenten des TwinCAT Systems genutzt. Eine nötige Konfiguration ist bei den jeweiligen Komponenten zu finden.

### 3.5.7.1 ADS Consumer

Das Beispiel ADS Consumer ruft korrigierte Zeitstempel ab, wie es in der [ADS API \[► 239\]](#) beschrieben ist.

#### Download

Hier erhalten Sie den [https://infosys.beckhoff.com/content/1031/tc3\\_Grundlagen/Resources/7705550603.zip](https://infosys.beckhoff.com/content/1031/tc3_Grundlagen/Resources/7705550603.zip) für dieses Beispiel.

- ✓ Starten Sie das TwinCAT Zielsystem, mit dem das ADS Consumer Beispiel kommunizieren soll. Es kann das [PLC Consumer \[► 241\]](#) Beispiel verwendet werden.
1. Entpacken Sie die heruntergeladene ZIP-Datei.
  2. Öffnen Sie die enthaltene vcxproj-Datei im Visual Studio.
  3. Passen Sie die AmsNetID des Ziels an. (TcExternalTimeAdsClient.cpp, Zeile 119)
- ⇒ Das Beispiel ist einsatzbereit.

#### Beschreibung

Der Code des Beispiels befindet sich in der CPP Datei TcExternalTimeAdsClient.cpp

In der Main() Methode werden unterschiedliche UseCases für den Empfang von korrigierten Zeitstempel gezeigt:


- Lesen des Providers, des Offsets sowie des korrigierten Zeitstempel vom System Service für die unterschiedlichen Offsets unkorrigiert(0), soft(1), medium(2), hard(3) aber auch einen ungültigen Wert (4) um das Fehlverhalten darzustellen.
- Lesen der korrigierten Zeitstempel von einem PLC Programm wiederum für die unterschiedlichen Offsets.
- Lesen des verwendeten Providers und lesen aller Provider.
- Subscriben auf eine Variable in dem PLC, wobei die per Notification bereitgestellte Zeit einen korrigierten Zeitstempel aufweist. Die Ausgabe hiervon erfolgt in der Methode AdsNotificationCallback().

### 3.5.7.2 PLC Consumer

Das Beispiel PLC Consumer ruft einen korrigierten Zeitstempel aus dem TwinCAT-System ab und verwendet diesen.

#### Download

Hier erhalten Sie den [https://infosys.beckhoff.com/content/1031/tc3\\_Grundlagen/Resources/7705583115.zip](https://infosys.beckhoff.com/content/1031/tc3_Grundlagen/Resources/7705583115.zip) für dieses Beispiel.

1. Öffnen Sie die enthaltene tzip-Datei in TwinCAT 3 mit einem Klick auf **Open Project ....**
  2. Wählen Sie Ihr Zielsystem aus.
  3. Bauen Sie das Beispiel auf Ihrer lokalen Maschine (z. B. **Build->Build Solution**).
  4. Aktivieren Sie die Konfiguration mit einem Klick auf  .
- ⇒ Das Beispiel ist einsatzbereit.

#### Beschreibung

Unterhalb von **System >TcCOM Objects** ist der TcNtpExternalTimeProvider konfiguriert. Hier kann unter **Parameter (Init)** ein eigener NTP-Server parametrisiert werden, wenn der voreingestellte pool.ntp.org nicht erreichbar ist.

Das PLC Programm besteht im Wesentlichen aus dem Funktionsbaustein FB\_TcExternalTime. Dieser stellt Funktionen bereit, um einen korrigierten Zeitstempel aus dem TwinCAT System auszulesen. Die Variable \_eTimeType stellt dabei den Typen (Soft, Medium, Hard) dar und kann parametrierbar werden.


In der MAIN wird dieser Funktionsbaustein für den eTimeType „Soft“ verwendet, um die per NTP gesetzte korrigierte Zeit zu verwenden.

### 3.5.7.3 C++ Consumer

Das Beispiel C++ Consumer ruft einen korrigierten Zeitstempel aus dem TwinCAT-System ab und verwendet diesen.

#### Download

Hier erhalten Sie den [https://infosys.beckhoff.com/content/1031/tc3\\_Grundlagen/Resources/7705552907.zip](https://infosys.beckhoff.com/content/1031/tc3_Grundlagen/Resources/7705552907.zip) für dieses Beispiel.

1. Öffnen Sie die enthaltene zip-Datei in TwinCAT 3 mit einem Klick auf **Open Project ....**
2. Wählen Sie Ihr Zielsystem aus.
3. Bauen Sie das Beispiel auf Ihrer lokalen Maschine (z. B. **Build->Build Solution**).
4. Aktivieren Sie die Konfiguration mit einem Klick auf  .  
⇒ Das Beispiel ist einsatzbereit.

#### Beschreibung

Unterhalb von **System >TcCOM Objects** ist der TcNtpExternalTimeProvider konfiguriert.

Hier kann unter **Parameter (Init)** ein eigener NTP-Server parametrierbar werden, wenn der voreingestellte pool.ntp.org nicht erreichbar ist.

Das C++ Modul ermittelt in der CycleUpdate() Methode zyklisch einen lokalen Zeitstempel und lässt diesen korrigieren, was mittels Debugger in den jeweiligen Schritten verfolgt werden kann. Der korrigierte Zeitstempel wird als Parameter (Online) bereitgestellt.


Der dafür notwendige Typ wird als Parameter „TimeType“ am TcCOM Objekt konfigurierbar.

### 3.5.7.4 C++ Provider

Das Beispiel C++ Provider ermittelt einen Offset und legt diesen im TwinCAT-System ab, sodass er von den Consumern verwendet werden kann.

#### Download

Hier erhalten Sie den [https://infosys.beckhoff.com/content/1031/tc3\\_Grundlagen/Resources/7705555211.zip](https://infosys.beckhoff.com/content/1031/tc3_Grundlagen/Resources/7705555211.zip) für dieses Beispiel.

1. Entpacken Sie die heruntergeladene .zip-Datei.
2. Öffnen Sie die enthaltene .zip-Datei in TwinCAT 3 mit einem Klick auf **Open Project ....**
3. Wählen Sie Ihr Zielsystem aus.
4. Bauen Sie das Beispiel auf Ihrer lokalen Maschine (z. B. **Build->Build Solution**).
5. Aktivieren Sie die Konfiguration mit einem Klick auf  .  
⇒ Das Beispiel ist einsatzbereit.

#### Beschreibung

Der Offset Provider bekommt den bereitzustellenden Offset als DataArea „ExternalTime.nOffset“. Dieser wird dem TwinCAT-System als TimeType Medium übergeben was unter **Parameter (Init)** auch zur Laufzeit konfiguriert werden kann.

In der CycleUpdate() Methode wird hierfür die Methode SetExternalTimeOffset verwendet, nachdem ein entsprechendes Register mittels RegisterExternalTimeProvider für einen TimeType erfolgt ist.

## 3.5.8 FAQ

### 3.5.8.1 Windows als NTP Client

Windows selber bietet einen NTP Client für die Systemzeit. Zusätzlich kann eine NTP Zeit auch mittels folgendem Skript abgeholt werden, welches für Debugging-Zwecke sinnvoll ist:

```
@echo off
set /p Server=Server:
w32tm /stripchart /computer:%Server% /packetinfo /samples:10
pause
```

### 3.5.8.2 Windows als NTP Server

Windows selber bietet einen NTP Server an um Zeitstempel bereitzustellen.

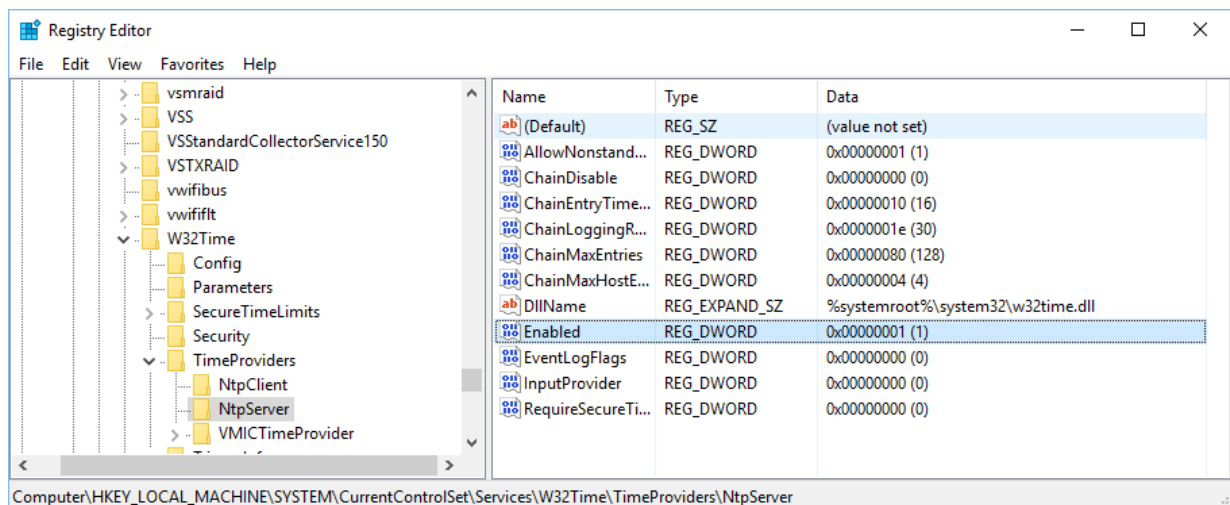
Bitte beachten Sie hierbei, dass nur eine Komponente den Port für NTP (udp/123) nutzen kann. Es kann also entweder die [TwinCAT NTP Server Funktionalität \[► 228\]](#) verwendet werden oder der Windows NTP Server.

Der Windows NTP Server ist standardmäßig ausgeschaltet und kann nachträglich aktiviert werden:

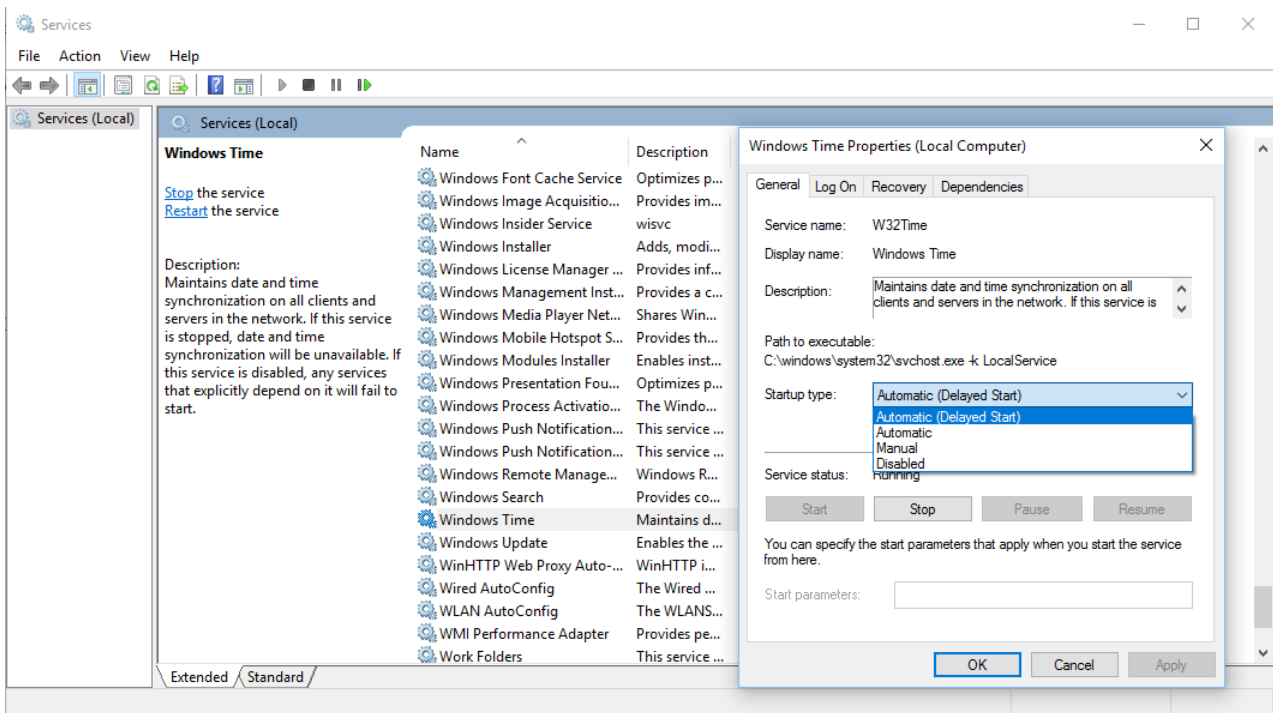
✓ Windows 7 / 10

1. Der Registry Key wird gesetzt:

```
HKLM\System\CurrentControlSet\Services\W32Time\TimeProviders\NtpServer
Enabled = 1
```



2. Und der System Dienst "Windows Time" wird gestartet, ggf auf Autostart gestellt.



### 3.6 TcRTelinstall

Das Werkzeug TcRTelinstall verwaltet Echtzeit-Ethernet kompatible Geräte des Steuerungssystems. Dabei wird ein echtzeitfähiger Treiber für den Standard-Ethernet-Anschluss eines Steuerungssystems installiert.

#### ● **TwinCAT-3-Installation notwendig**

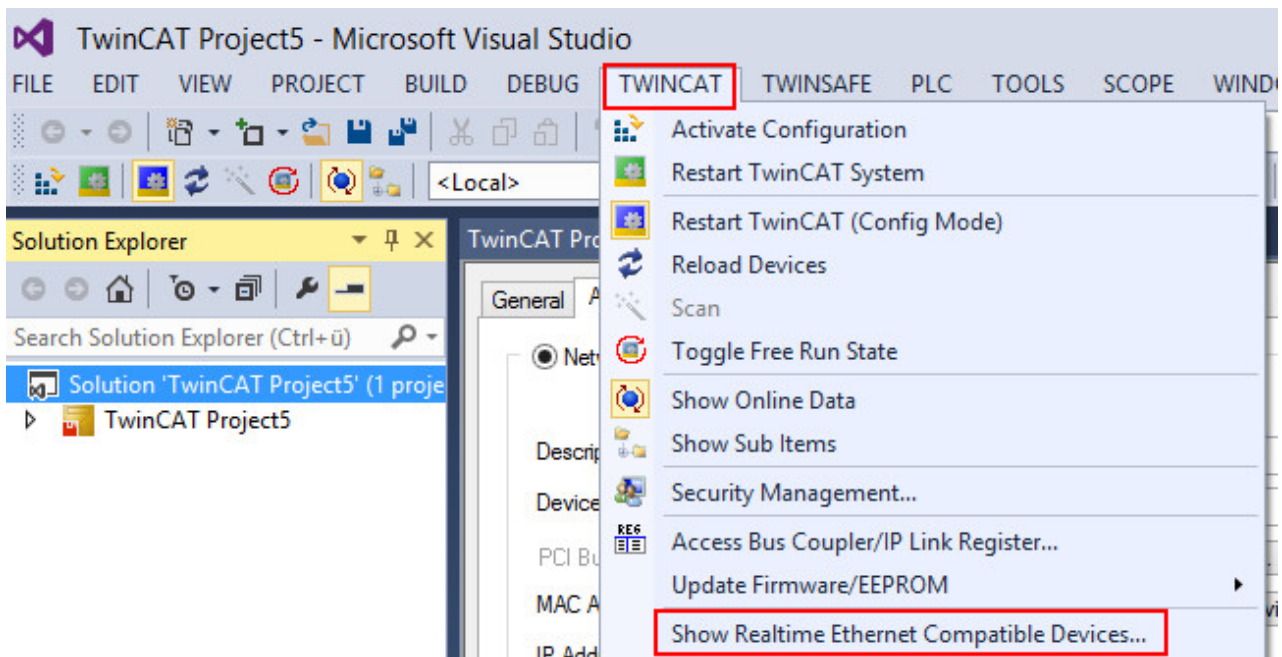
**i** Die Nutzung der TcRTelinstall-Anwendung ist nur in Kombination mit einer vollständigen Installation von TwinCAT 3 (XAE, Laufzeitumgebung, XAR) möglich.

#### ● **Administratorrechte erforderlich**

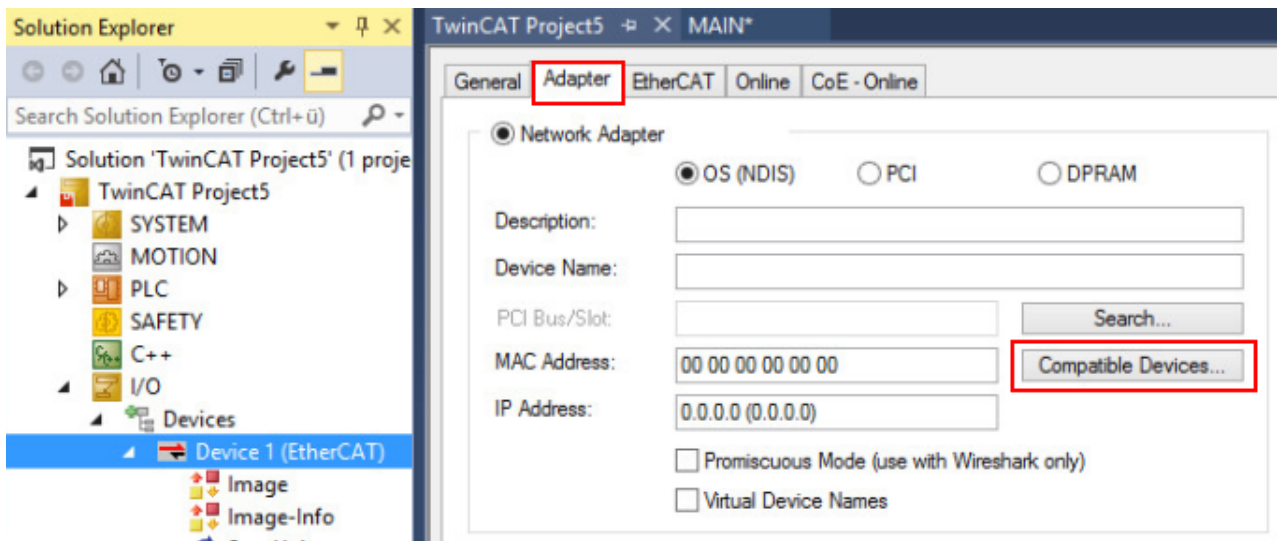
**i** Zur Ausführung der TcRTelinstall-Anwendung benötigen Sie Administratorrechte auf dem Steuerungssystem.

**Aufruf in TwinCAT 3 XAE**

Rufen Sie den Treiber über das Menü **TWINCAT** → **Show Realtime Ethernet Compatible Devices...** auf.



Alternativ können Sie den Treiber installieren, indem Sie der E/A-Konfiguration ein netzwerkfähiges Gerät hinzufügen (z.B. EtherCAT). Rufen Sie im Adapter-Dialog des netzwerk-fähigen Gerätes mit der Schaltfläche **Compatible Devices...** die TcRTEInstall-Anwendung auf:



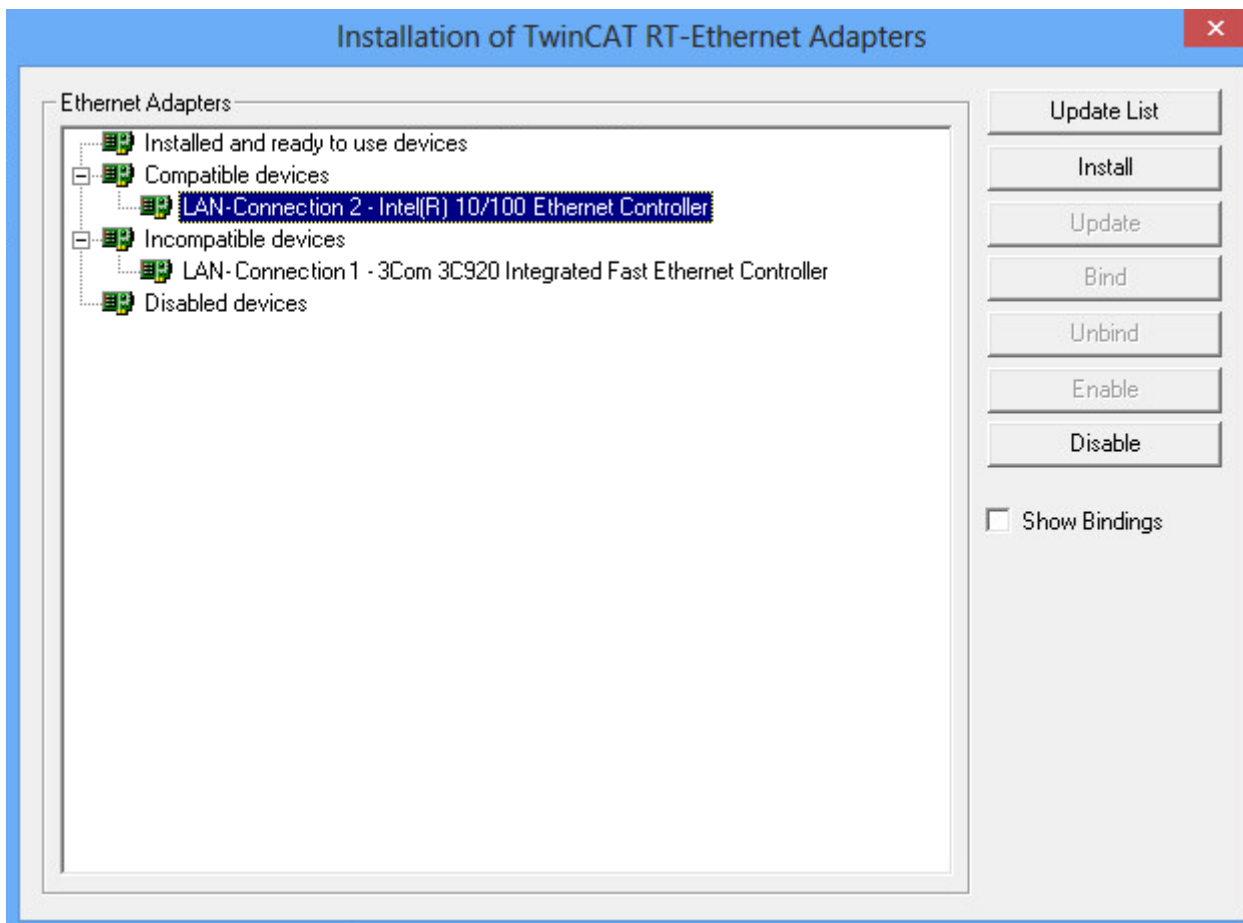
**Aufruf in TwinCAT Laufzeitumgebungen**

Sie können die Installationsanwendung für den TwinCAT RT-Ethernet-Adapter auf einem TwinCAT-3-Laufzeitsystem direkt aufrufen.

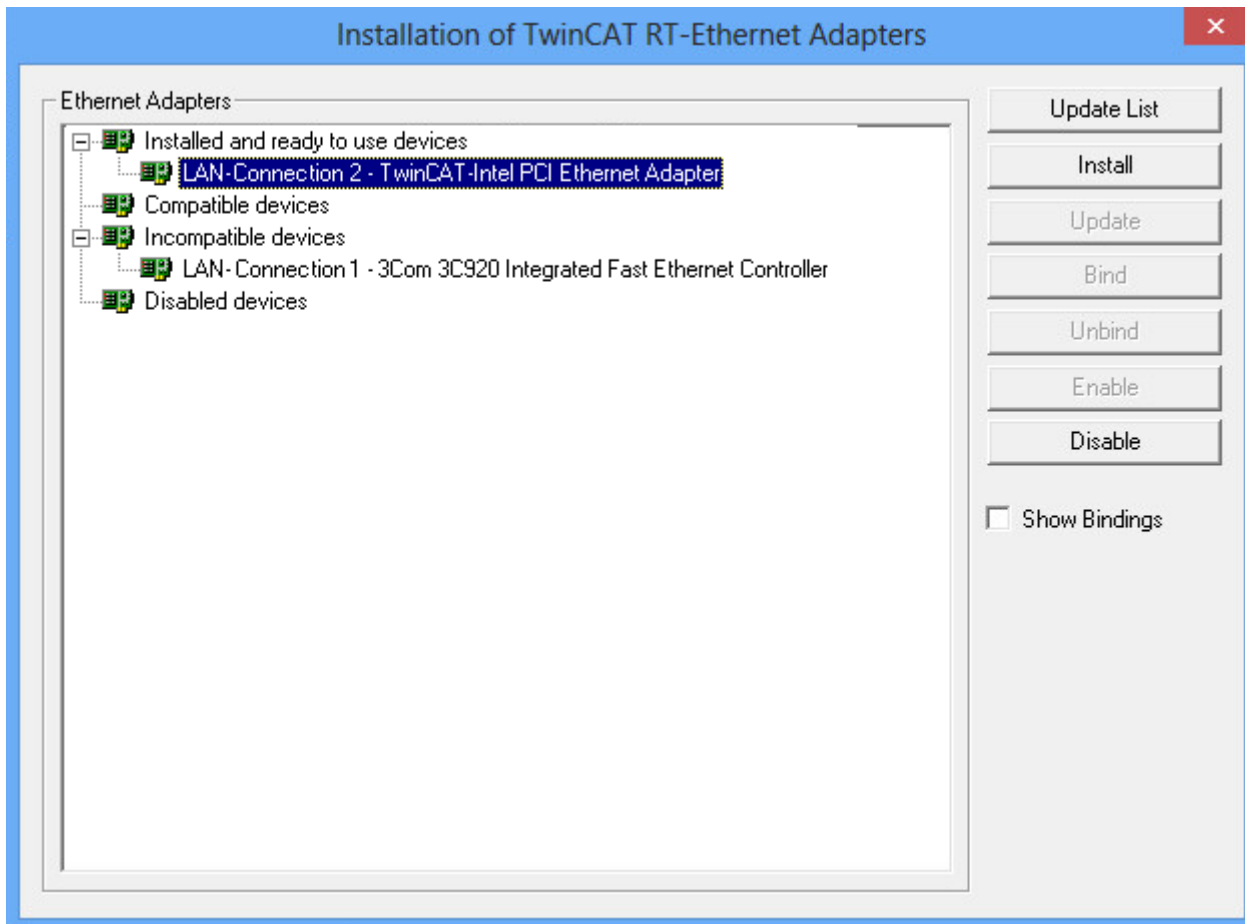
Speicherort: *c:\TwinCAT3.1\System\TcRteInstall.exe*

**Netzwerk-Anschlüsse verwalten**

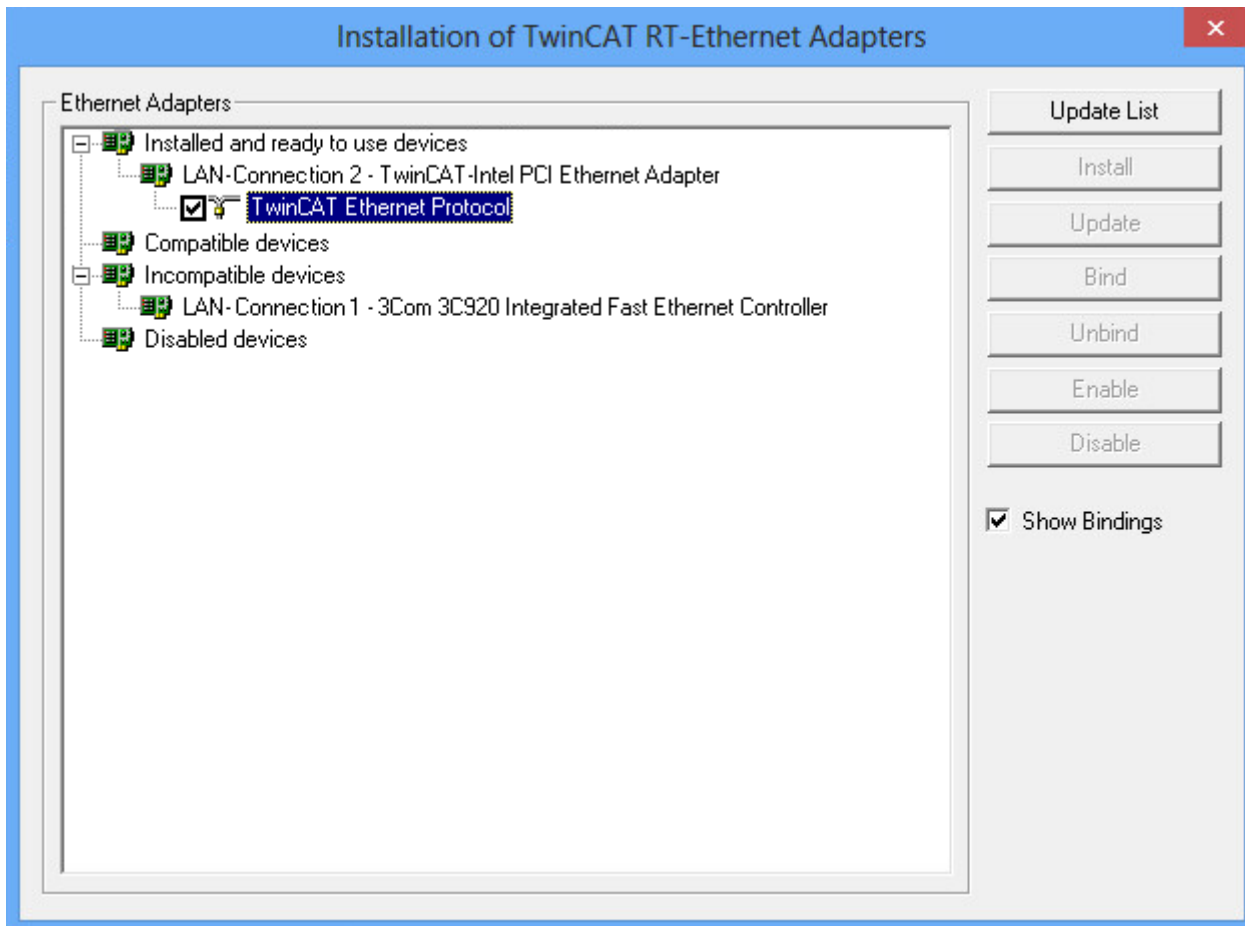
TcRteInstall zeigt die echtzeitfähigen (Compatible devices) und nicht echtzeitfähigen (Incompatible devices) Netzwerkadapter an.



1. Wählen Sie einen echtzeitfähigen Netzwerkadapter aus der Liste der „compatible devices“.
  2. Klicken Sie auf die Schaltfläche *Install*.
- ⇒ Für das ausgewählte Gerät wird der TwinCAT-Treiber für Echtzeit-Ethernet und das TwinCAT-Ethernet-Protokoll installiert.



Die Option **Show Bindings** zeigt das verbundene Protokoll des installierten RT-Ethernet-Gerätes an.





## 4 Typsystem

TwinCAT 3 stellt ein Typsystem für die Verwaltung von Datentypen bereit. Das Typsystem besteht aus System-Basistypen und kann durch das Kundenprojekt um eigene Datentypen erweitert werden.

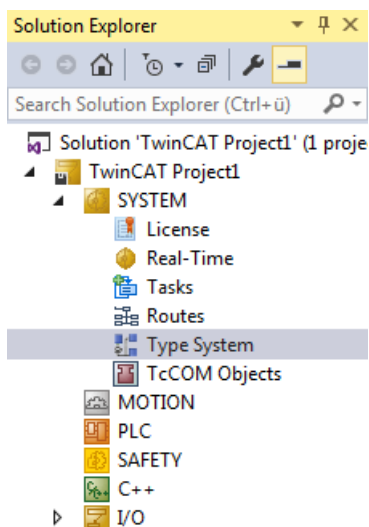
Diese Dokumentation beschreibt das TwinCAT 3 Typsystem und die Verwaltung von Datentypen. Der TMC-Editor, mit dem Datentypen erstellt und beschrieben werden, wird in der Dokumentation „C++“ im Abschnitt TwinCAT Module Class Editor (TMC) beschrieben.

### 4.1 Projektbasiertes Typsystem

Das TwinCAT 3 Typsystem ist projektspezifisch, d. h. es ist fester Bestandteil eines TwinCAT-3-Projekts in einer Visual Studio Solution.

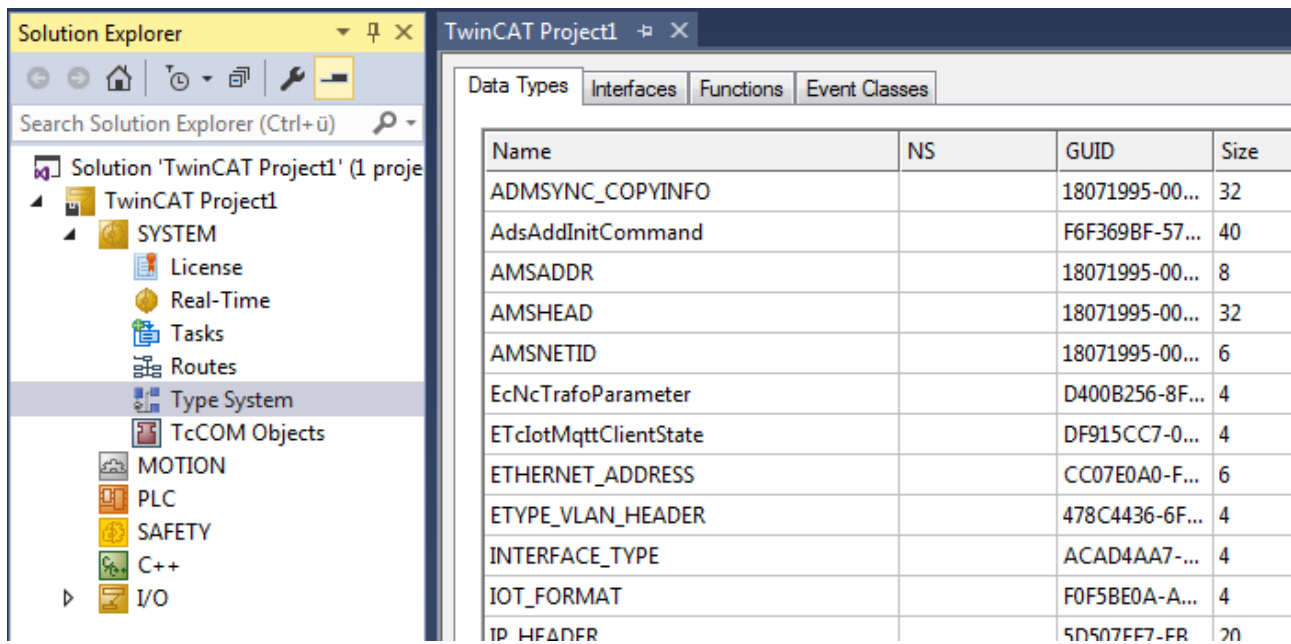
Datentypen können an unterschiedlichen Stellen definiert und bei Bedarf in das TwinCAT 3 Typsystem transferiert werden. So können auch lokale Datentypen existieren, die nicht im TwinCAT 3 Typsystem vorhanden sind.

Sie finden das Typsystem im TwinCAT-3-Projektbaum als Objekt im Teilbaum SYSTEM.



### 4.2 Arten von Datentypen

Das TwinCAT-3-Typsystem stellt die Datentypen in einem Editor in vier unterschiedlichen Registerkarten dar. Sie öffnen den Editor mit einem Doppelklick auf das Objekt „Type System“ im TwinCAT-3-Projektbaum.



In der Registerkarte **Data Types** werden die folgenden Arten von Datentypen (TMC-Editor: „Specifications“) dargestellt:

- **Alias:** Diese Datentypen sind einfach Synonyme für andere Datentypen. So kann beispielsweise ein Zeitbereich (Duration) als UINT projektspezifisch festgelegt werden.
- **Struct:** Diese Datentypen sind Strukturen von anderen Datentypen, die auch wieder Strukturen sein können.
- **Enum:** Diese Datentypen beschreiben Enumerations, also Aufzählungen.
- **Array:** Diese Datentypen sind Arrays mit definierter Dimensionen-Anzahl sowie jeweiliger Länge.

In der Registerkarte **Interfaces** werden die Interfaces dargestellt. Dieser Datentyp beschreibt eine Schnittstelle, die von unterschiedlichen Komponenten wie Funktionsbausteinen oder TcCOM-Modulen bereitgestellt oder genutzt werden kann. Ein Interface besteht aus Methoden, die eine jeweilige Signatur haben.

In der Registerkarte **Functions** werden SPS-Funktionen und SPS-Funktionsbausteine dargestellt, deren Definition aus in einer TMC-/TML-Datei gelesen wurde.

In der Registerkarte **Event Classes** werden Eventklassen definiert, welche für den TwinCAT 3 Eventlogger genutzt werden.

## 4.3 Handhabung von Datentypen

Um einen Datentyp über das TwinCAT 3 Typsystem anzulegen oder zu verändern, wählen Sie in der entsprechenden Registerkarte des Typsystem-Editors im Kontextmenü der ersten Tabellenspalte den Befehl **New** bzw. **Edit**. Beide Befehle öffnen den TMC-Editor, in dem Sie den Datentyp bearbeiten können.

### Datentypen aus SPS-Projekten

In einem SPS-Projekt können Datentypen (DUTs) angelegt und gespeichert werden. Diese Datentypen sind erst einmal lokal in dem SPS-Projekt vorhanden und aus Sicht des TwinCAT 3 Typsystems nicht nutzbar. Wenn die Datentypen im Ein-/Ausgangsspeicherabbild (%I\* / %Q\*) verwendet werden, werden sie im TwinCAT 3 Typsystem eingebracht, sodass sie auch durch das Mapping verknüpfbar sind.

Mit dem Befehl **Convert to Global Type** im Kontextmenü eines DUTs im SPS-Projektbaum können Sie den DUT in das Typsystem des übergeordneten TwinCAT-Projekts übertragen. Danach ist der Datentyp in der SPS über die externen Typen nutzbar und wird im TwinCAT 3 Typsystem verwaltet.

Um einen Datentyp vom TwinCAT 3 Typsystem in ein SPS-Projekt zu übertragen, können Sie den Quellcode in dem „Data Types“-Dialog verwenden.

### Datentypen aus C++-Projekten

In C++-Projekten werden die Datentypen im TMC-Editor parallel zu den Modulen definiert. Diese Datentypen sind analog zu den SPS-Projekt-internen DUTs lokal und damit im TwinCAT 3 Typsystem nicht sichtbar.

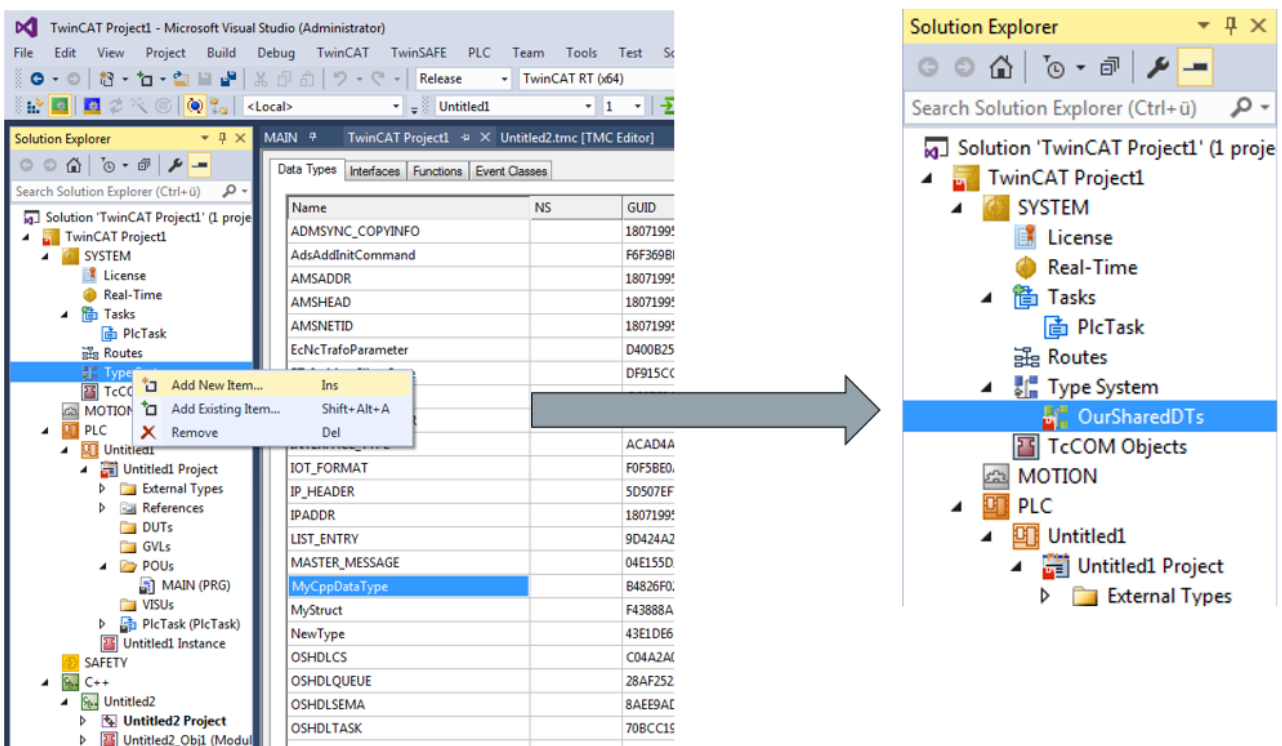
Durch Verwendung der Datentypen in einem C++/Matlab-Modul, welches auch instanziiert wurde, werden die Datentypen in das TwinCAT 3 Typsystem eingefügt.

Sie können einen Datentyp auch durch das Aktivieren des Auswahlkästchens **Persistent (even if unused)** in das TwinCAT 3 Typsystem einfügen, ohne dass der Datentyp in einem instanziierten C++-Modul verwendet wird.

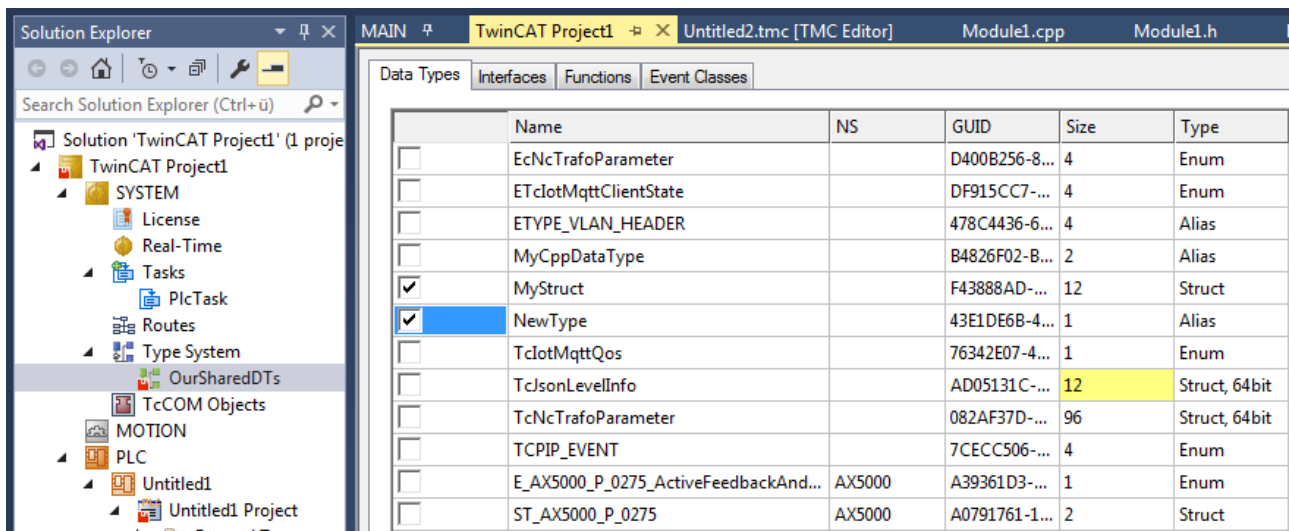
### Nutzung von Datentypen in mehreren Projekten

In einigen Fällen kann es sinnvoll sein, Datentypen in mehreren Projekten zu verwenden. Insbesondere für EAP-/Netzwerkvariablen kann es sinnvoll sein, auf Publisher- und Subscriber-Seite den gleichen Datentyp zu nutzen.

Unter dem Knoten „Type System“ können Sie hierfür einzelne TMC-Dateien anlegen.

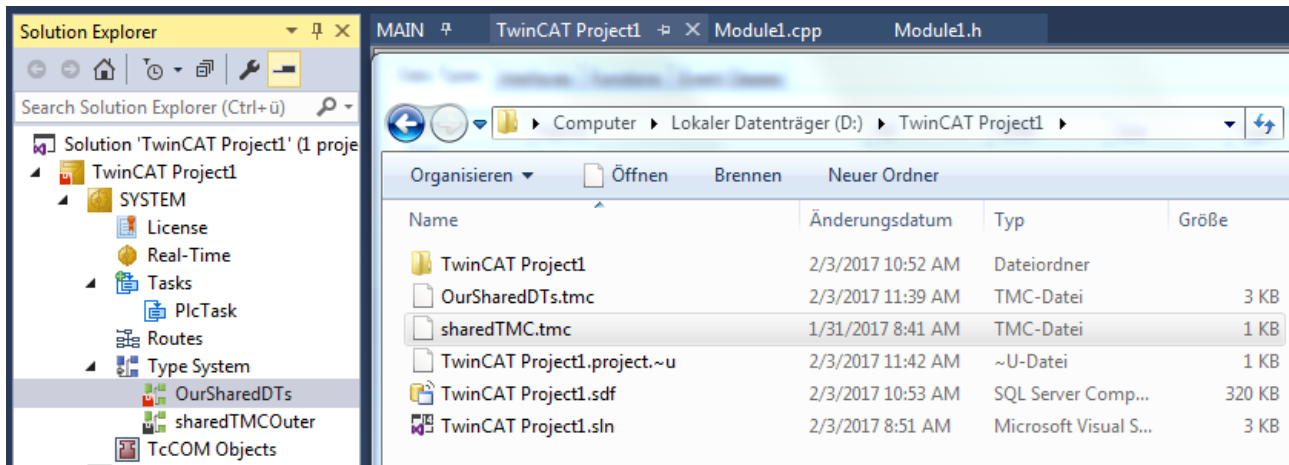


Im Editorfenster der TMC-Dateien erscheint vor jedem Datentyp ein Auswahlkästchen. Über das Auswahlkästchen können Sie angeben, welcher Datentyp in der jeweiligen TMC-Datei abgelegt werden soll.



Die Datentypen werden dabei zusätzlich in den TMC-Dateien abgelegt. So können diese Dateien z. B. per Dateiaustausch oder Versionskontrolle auf unterschiedlichen Rechnern und in unterschiedlichen Projekten verwendet werden.

Die Datei selber darf dabei jedoch nicht gleichzeitig von unterschiedlichen Projekten verwendet werden, sodass diese normalerweise in dem Projektverzeichnis abgelegt werden und dieses Projekt dann z. B. über die Versionskontrolle auf unterschiedlichen Rechnern als Kopie vorhanden ist.



Da die GUID zur Identifizierung von Datentypen genutzt wird, erkennt das Typsystem diese doppelte Ablage automatisch.

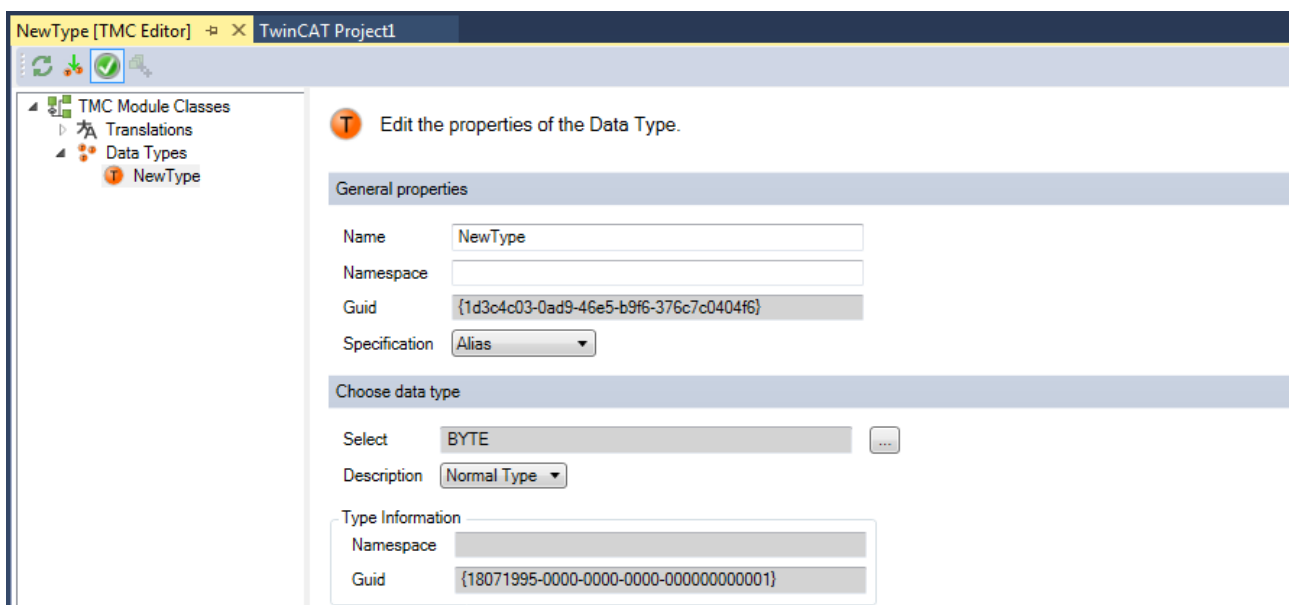
Beachten Sie bei der Verwendung von Datentypen, nachdem sie in mehreren Projekten eingebunden wurden, dass Änderungen an den Datentypen möglichst nur an einer Stelle vorgenommen werden. Ansonsten können die unterschiedlichen Varianten nicht mehr auf einen gemeinsamen Stand zusammengeführt werden können.

**Siehe auch:**

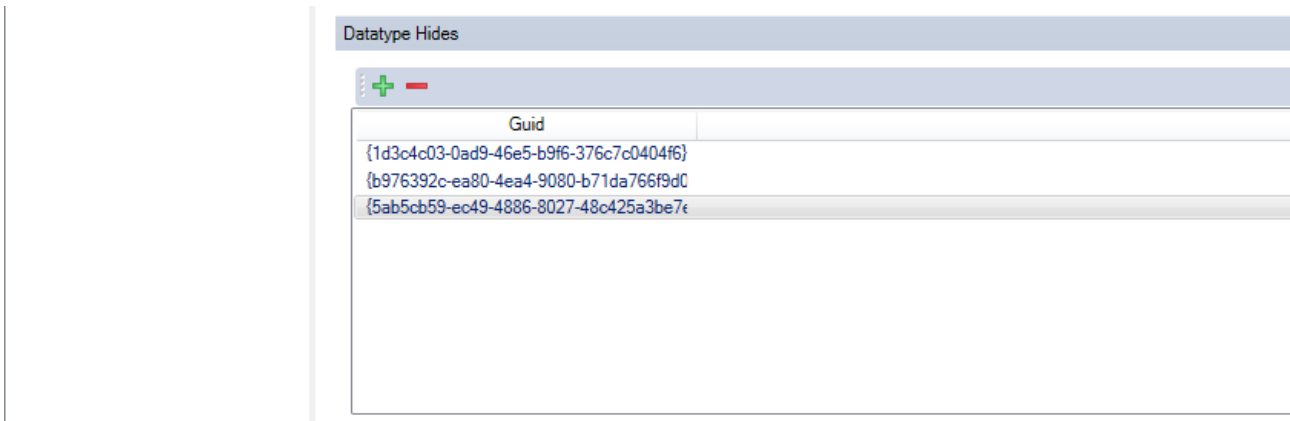
[Verwaltung und Identifizierung von Datentypen \[▶ 252\]](#)

## 4.4 Verwaltung und Identifizierung von Datentypen

Datentypen im TwinCAT 3 Typsystem werden grundsätzlich anhand ihrer GUID identifiziert. Somit können mehrere Datentypen mit gleichen Namen existieren. Dies gilt auch für unterschiedliche Versionen eines Datentyps. Jede Version eines Datentyps bekommt eine neue GUID zugewiesen.



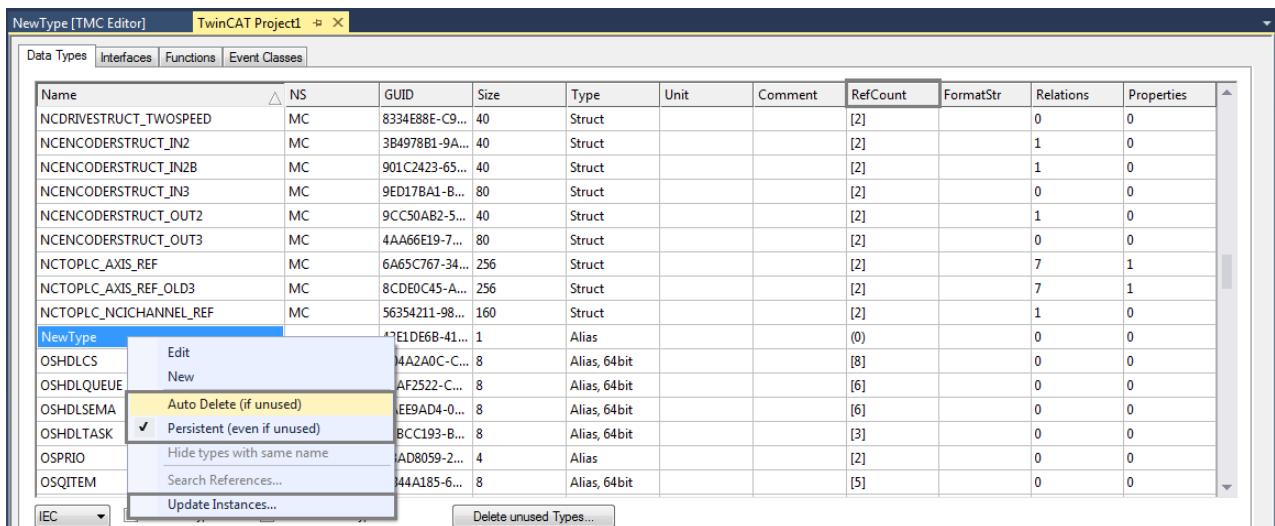
Gleichzeitig besitzt jeder Datentyp eine Liste von Datentypen, die er versteckt („Datatype Hides“).



Hieraus ergibt sich die Möglichkeit unterschiedliche Versionen eines Datentyps gleichzeitig im Projekt zu nutzen.

Der Befehl **Update Instances...** im Kontextmenü eines Datentyps im Editor des Typsystems (Registerkarte **Data Types**) setzt für ausgewählte Verwendungen eines Datentyps die jeweils neueste Version ein.

TwinCAT besitzt für jeden Datentyp einen sogenannten Reference Counter. Dieser Zähler ist im Editor des Typsystems in der Spalte **RefCount** zu sehen. Jede Verwendung des Datentyps in einem Projekt, aber auch in einem Editor usw. erhöht den Zähler. Wenn ein Zähler 0 ist, wird der Datentyp nicht mehr genutzt und verworfen.



Wenn die Einstellung **Persistent (even if unused)** im Kontextmenü eines Datentyps aktiviert ist, wird die Datentypbeschreibung in der TwinCAT-Projektdatei (\*.tsproj) gespeichert, auch wenn der Datentyp nicht im TwinCAT-Projekt verwendet wird. Bei Datentypen, die direkt über den Editor des Typsystems neu angelegt werden, ist die Einstellung standardmäßig aktiviert. So wird sichergestellt, dass die Datentypen nicht direkt gelöscht werden, wenn das TwinCAT-Projekt gespeichert wird, bevor die neuen Datentypen verwendet werden.

Wenn im TwinCAT-Projektbaum unterhalb des Objekts **Type System** ein SharedTMC verwendet wird, sollte die Einstellung für Datentypen in dieser Datei nicht aktiviert werden, da die Datentypen sonst sowohl in dem Projekt als auch in der SharedTMC abgelegt werden. Bei Datentypen, die über den Editor einer SharedTMC neu angelegt werden, ist die Einstellung standardmäßig nicht aktiviert.

Die Einstellung **Auto Delete (if unused)** sollte manuell nicht geändert werden, wird aber zur Vollständigkeit angezeigt. Datentypen, bei denen diese Einstellung aktiviert ist, werden für SPS-Projekte ausgeblendet und können dort nicht verwendet werden. Die Einstellung sollte nicht verwendet werden, um z. B. das Typsystem automatisch zu bereinigen. Unbenutzte Datentypen werden im TwinCAT-Projekt automatisch nicht gespeichert und sind nach einem Neuladen des TwinCAT-Projekts dann nicht mehr im Typsystem.

## 4.5 Alignment von Datentypen

Das Speicher-Layout eines Datentyps wird durch das Alignment bestimmt. Weitere Informationen zum Alignment finden Sie in der Dokumentation „PLC“ im Abschnitt „Alignment“.

Mit dem Default-Alignment von 8-Bytes kann gewährleistet werden, dass der Zugriff auf Datentypen auf unterschiedlichen Plattformen optimal im Sinne von Laufzeit und Zugriff funktioniert. Nur in Ausnahmefällen sollte hiervon abgewichen werden.

Das TwinCAT 3 Typsystem markiert Datentypen farbig.


- Gelb, wenn die Länge des Datentyps nicht ein Vielfaches des größten, internen Feldes (max. 8 Byte) ist. Dadurch entspricht bei einem Array eines solchen Datentyps das Alignment nicht mehr den Regeln.

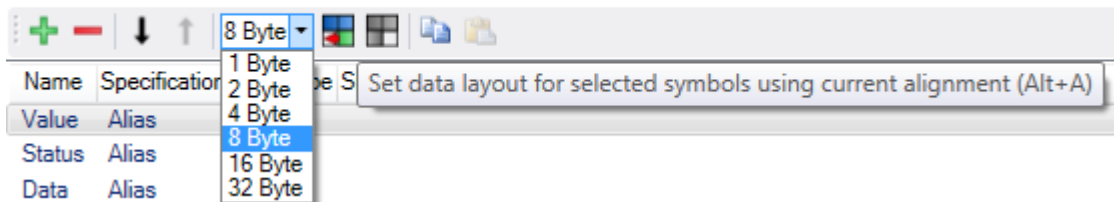
AlignmentMismatch	035500DD-FE07-4D6D-B195-...	10	Struct
-------------------	-----------------------------	----	--------

- Rot, wenn innerhalb des Datentyps das Alignment nicht den Regeln entspricht.

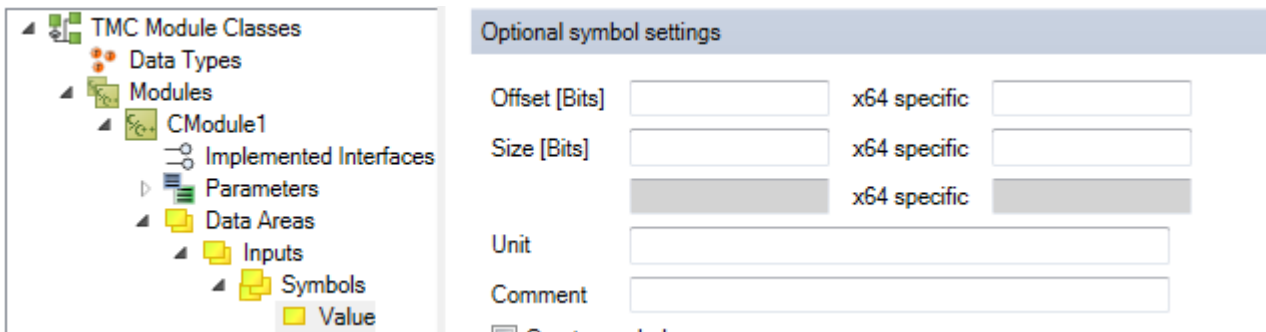
OuterType	566B0D7D-3403-4C85-8BEC-...	6	Struct
-----------	-----------------------------	---	--------

Der TMC-Editor bietet die Möglichkeit für ein ausgewähltes Alignment das Speicher-Layout eines Datentyps festzulegen.

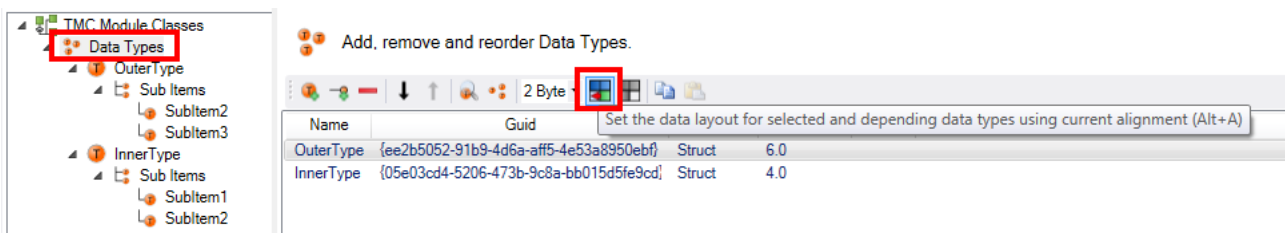
 Add, remove and reorder Symbols.



Alternativ kann das Layout über Offsets manuell festgelegt werden.



Wird die Größe eines Datentyps verändert, der in einem anderen Datentyp verwendet wird, muss auch dieser Datentyp angepasst werden. Hierfür bietet der TMC-Editor auf Ebene der Datentypen-Übersicht eine entsprechende rekursive Funktion.



## 4.6 Dateien im Zusammenhang mit dem Typsystem

Das TwinCAT 3 Typsystem ist vollständig in XML formuliert.

Je nach Anwendungsbereich gibt es unterschiedliche Dateien, die die Datentypen enthalten:

- .tsproj-Datei – TwinCAT Projekt  
Diese Datei umfasst das gesamte TwinCAT-Projekt einschließlich des gesamten TwinCAT 3 Typsystems.
- .tmc-Dateien – TwinCAT Module Class Dateien  
Diese Dateien werden verwendet, um die TcCOM-Module selbst zu beschreiben. Sie umfassen Modulklassenbeschreibungen und verwendete Datentypen. Gleichzeitig werden diese Dateien genutzt um, wie oben beschrieben, einen Austausch von Datentypen zwischen Projekten zu realisieren.
- .tmi-Dateien – TwinCAT Module Instance Dateien  
Diese Dateien beschreiben die Instanz einer Klasse. Sie werden vom TwinCAT 3 Engineering auf dem Ziel abgelegt, um eine Instanz einer Klasse zu beschreiben. Zusätzlich können per .tmi-Datei auch Instanz-Informationen von einem Projekt zum anderen übertragen werden.



## 5 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

### Downloadfinder

Unser [Downloadfinder](#) beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

### Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den [lokalen Support und Service](#) zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: [www.beckhoff.com](http://www.beckhoff.com)

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

### Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157

E-Mail: [support@beckhoff.com](mailto:support@beckhoff.com)

### Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460

E-Mail: [service@beckhoff.com](mailto:service@beckhoff.com)

### Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49 5246 963-0

E-Mail: [info@beckhoff.com](mailto:info@beckhoff.com)

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





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

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)

