BECKHOFF New Automation Technology

Original-Handbuch | DE

CX7050

Embedded-PC für CANopen-Commander (Master)

08.05.2024 | Version: 1.0

Inhaltsverzeichnis

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

1.1 Symbolerklärung

In der Dokumentation werden folgende Warnhinweise verwendet. Lesen und befolgen Sie die Warnhinweise.

Warnhinweise, die vor Personenschäden warnen:

⚠ 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 leichte Verletzung zur Folge haben kann.

Warnhinweise, die vor Sach- oder Umweltschäden warnen:

HINWEIS
Es besteht eine mögliche Gefährdung für Umwelt und Geräte.

Hinweise, die weitere Informationen oder Tipps anzeigen:

Dieser Hinweis gibt wichtige Informationen, die beim Umgang mit dem Produkt oder der Software helfen. Es besteht keine unmittelbare Gefahr für Produkt, Mensch und Umwelt.

1.2 Ausgabestände der Dokumentation

Version	Kommentar
1.0	Erste Version.

2 Zu Ihrer Sicherheit

Lesen Sie das Sicherheitskapitel und halten Sie die Hinweise ein, um sich vor Personenschäden und Sachschäden zu schützen.

Haftungsbeschränkungen

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Eigenmächtige Umbauten und Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind verboten und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.
Darüber hinaus werden folgende Punkte aus der Haftung der Beckhoff Automation GmbH & Co. KG ausgeschlossen:

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

2.1 Bestimmungsgemäße Verwendung

Der Embedded-PC ist ein Steuerungssystem für den Einsatz im Maschinen- und Anlagenbau zur Automatisierung, Visualisierung und Kommunikation. Der Embedded-PC ist für den Einbau in einen Schaltschrank oder Klemmenkasten vorgesehen und wird zusammen mit Bus- oder EtherCAT-Klemmen dazu verwendet, um digitale und analoge Signale von Sensoren aufzunehmen und an Aktoren auszugeben oder an übergeordnete Steuerungen weiterzuleiten.

Der Embedded-PC ist für ein Arbeitsumfeld entwickelt, welches der Schutzklasse IP20 genügt. Es besteht Fingerschutz und Schutz gegen feste Fremdkörper bis 12,5 mm, jedoch kein Schutz gegen Wasser. Der Betrieb der Geräte in nasser und staubiger Umgebung ist nicht gestattet, sofern nicht anders angegeben. Die angegebenen Grenzwerte für elektrische- und technische Daten müssen eingehalten werden.

Nicht bestimmungsgemäße Verwendung

Der Embedded-PC ist nicht für den Betrieb in folgenden Bereichen geeignet:

- In explosionsgefährdeten Bereichen.
- In Bereichen mit einer aggressiven Umgebung, die z.B. mit aggressiven Gasen oder Chemikalien angereichert ist.
- Im Wohnbereich. Im Wohnbereich müssen die entsprechenden Normen und Richtlinien für Störaussendungen eingehalten und die Geräte in Gehäuse oder Schaltkästen mit entsprechender Schirmdämpfung eingebaut werden.

2.2 Personalqualifikation

Alle Arbeitsschritte an der Beckhoff Soft- und Hardware dürfen nur vom Fachpersonal mit Kenntnissen in der Steuerungs- und Automatisierungstechnik durchgeführt werden. Das Fachpersonal muss über Kenntnisse in der Administration des eingesetzten Industrie-PCs und des jeweils eingesetzten Netzwerks verfügen.

Alle Eingriffe müssen mit Kenntnissen in der Steuerungs-Programmierung durchgeführt werden und das Fachpersonal muss die aktuellen Normen und Richtlinien für das Automatisierungsumfeld kennen.

2.3 Sicherheitshinweise

Folgende Sicherheitshinweise müssen während der Montage, der Arbeit mit Netzwerken und der Arbeit mit Software beachtet werden.

Montage

- Arbeiten Sie nicht an Geräten unter Spannung. Schalten Sie immer die Spannungsversorgung für das Gerät ab bevor Sie es montieren, Störungen beheben oder Wartungsarbeiten durchführen. Sichern Sie das Gerät gegen ein unbeabsichtigtes Einschalten ab.
- Beachten Sie die Unfallverhütungsvorschriften, die für Ihre Maschine zutreffend sind (z.B. die BGV A 3, Elektrische Anlagen und Betriebsmittel).
- Achten Sie auf einen normgerechten Anschluss und vermeiden Sie Gefahren für das Personal. Verlegen Sie die Daten- und Versorgungsleitungen normgerecht und achten Sie auf die korrekte Anschlussbelegung.
- Beachten Sie die für Ihre Anwendung zutreffenden EMV-Richtlinien.
- Vermeiden Sie die Verpolung der Daten- und Versorgungsleitungen, da dies zu Schäden an den Geräten führen kann.
- In den Geräten sind elektronische Bauteile integriert, die Sie durch elektrostatische Entladung bei Berührung zerstören können. Beachten Sie die Sicherheitsmaßnahmen gegen elektrostatische Entladung entsprechend DIN EN 61340-5-1/-3.

Arbeiten mit Netzwerken

- Beschränken Sie den Zugriff zu sämtlichen Geräten auf einen autorisierten Personenkreis.
- Ändern Sie die standardmäßig eingestellten Passwörter und verringern so das Risiko, dass Unbefugte Zugriff erhalten.
- Schützen Sie die Geräte mit einer Firewall.
- Wenden Sie die Vorgaben zur IT-Sicherheit nach der IEC 62443 an, um den Zugriff und die Kontrolle auf Geräte und Netzwerke einzuschränken.

2.4 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.

3 Transport und Lagerung

Transport

HINWEIS

Kurzschluss durch Feuchtigkeit

Feuchtigkeit kann sich bei Transporten in kalter Witterung oder bei extremen Temperaturunterschieden bilden.

Achten Sie darauf, dass sich keine Feuchtigkeit im Embedded-PC niederschlägt (Betauung) und gleichen Sie ihn langsam der Raumtemperatur an. Schalten Sie den Embedded-PC bei Betauung erst nach einer Wartezeit von mindestens 12 Stunden ein.

Trotz des robusten Aufbaus sind die eingebauten Komponenten empfindlich gegen starke Erschütterungen und Stöße. Schützen Sie den Embedded-PC bei Transporten vor:

- großer mechanischer Belastung und
- benutzen Sie für den Versand die Originalverpackung.

Tab. 1: Abmessungen und Gewicht.

	CX7050
Abmessungen (B x H x T)	49 mm x 100 mm x 73 mm
Gewicht	ca. 142 g

Lagerung

- Lagern Sie den Embedded-PC in der Originalverpackung.

4 Produktübersicht

Der Embedded-PC CX7050 verfügt über einen ARM-Cortex™-M7-Singlecore-Prozessor mit 480 MHz und hat die folgende Grundausstattung:

- einen MicroSD-Kartenslot mit integrierter 512 MB MicroSD-Karte,
- eine Ethernet-Schnittstelle (10/100 MBit/s, RJ45),
- eine USB-Schnittstelle (max. 12 Mbit/s, max. 100 mA),
- sowie integrierte Multifunktions-I/Os.

Der CX7050 wird mit TwinCAT 3 über die Ethernet-Schnittstelle programmiert. Zusätzlich steht der Beckhoff Device Manger als Webinterface für die Konfiguration des CX7050 zur Verfügung.

Die CANopen-Commander (Master)-Schnittstelle des CX7050 kann auch als CAN- oder CANopen-Responder (Slave) verwendet werden.

Multifunktions-I/Os

Als Besonderheit verfügt die CX7000-Serie über acht integrierte Multifunktionseingänge und vier integrierte Multifunktionsausgänge.

- 8 digitale Eingänge, 24 V DC, Filter 3 ms, Typ 3, 1-Leitertechnik
- 4 digitale Ausgänge, 24 V DC, 0,5 A, 1-Leitertechnik

Die integrierten Multifunktions-I/Os des CX7050 lassen sich über TwinCAT 3 für andere Betriebsarten (Modi) konfigurieren, sodass auch schnelles Zählen oder die Analogwertverarbeitung ermöglicht wird:

- Zähler-Modus: 1 x Zähler-Digitaleingang 100 kHz, 1 x Digitaleingang als Auf-/Abwärts-Zähler 20 kHz, 2 x Zähler-Digitalausgang
- Inkremental-Encoder-Modus: 2 x Digitaleingang für 250-kHz-Encoder-Signal (A-/B-Eingang), 2 x Encoder-Digitalausgang
- Analog-Signal-Modus: 2 x Digitaleingang konfiguriert als Analogeingang 0 bis 10 V, 12 Bit Auflösung bei 16 Bit Darstellung
- PWM-Signal-Modus: 2 x Digitalausgang konfiguriert für PWM-Signal, 15 Hz…100 kHz

Netzteilklemme

Auf der rechten Seite können wahlweise EtherCAT-Klemmen (E-Bus) oder Busklemmen (K-Bus) angereiht werden; der CX7050 erkennt in der Hochlaufphase automatisch, welches System angeschlossen ist. Sollen weitere elektrische Signale verarbeitet werden, kann somit der CX7050 ergänzend zu den integrierten I/Os bedarfsgerecht und äußerst flexibel durch EtherCAT-Klemmen oder Busklemmen erweitert werden.

Firmware

Als Betriebssystem bzw. Firmware wird das Echtzeitbetriebssystem TC/RTOS eingesetzt, welches auf FreeRTOS basiert. Beachten Sie, dass TC/RTOS ein geschlossenes System ist und eigene Softwareinstallationen nicht möglich sind. Damit ist eine gewisse Sicherheit gegeben, da Fremdsoftware wie Viren oder ähnliches nicht installiert werden können und der CX7050 an ein Netzwerk angeschlossen werden kann. Der CX7050 kann ab TwinCAT 3.1 Build 4024.12 verwendet werden. Folgende TC 3 Funktionen sind enthalten und lizensiert:

- TC1000 TC3 ADS
- TC1100 TC3 IO
- TC1200 TC3 PLC
- TF4100 TC3 Controller Toolbox
- TF4110 TC3 Temperature-Controller
- TF6255 TC3 Modbus-RTU
- TF6340 TC3 Serial-Communication
- TF6701 | TwinCAT 3 IoT Communication (MQTT)[*)]
- TF6730 | TwinCAT 3 IoT Communicator[*)]

*) Image-Version 114606 und TwinCAT 3 XAE 4024.47 oder höher erforderlich.

Die Open-Source-Lizenzen sind als ZIP-Datei auf der MicroSD-Karte einsehbar.

4.1 Aufbau

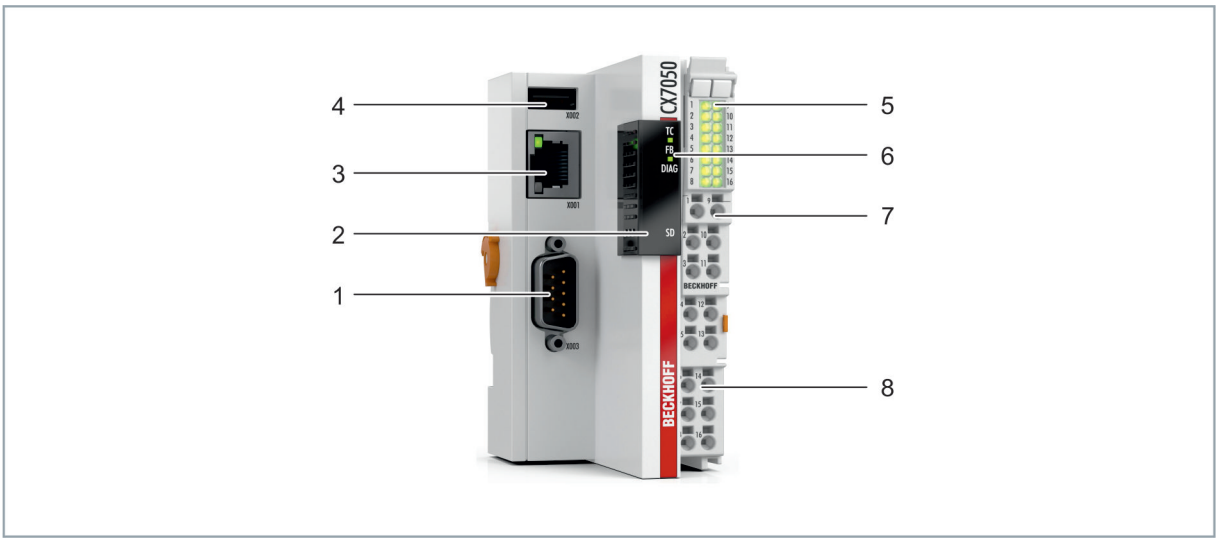


Abb. 1: Beispielaufbau eines Embedded-PCs CX7050.

Tab. 2: Legende zum Aufbau des CPU-Grundmoduls

Nr.	Komponente	Beschreibung
1	D-Sub-Stecker (X003).	CANopen-Schnittstelle des CX705x.
2	MicroSD-Kartenslot (unter der Abdeckung).	Steckplatz für industrietaugliche MicroSD-Karten. Speicherplatz für Firmware und TwinCAT 3 Projekte.
3	Ethernet-Schnittstelle (X001)	Für den Anschluss an lokale Netzwerke. Dient als Programmierschnittstelle.
4	USB-Schnittstelle (X002)	Schnittstelle für zusätzlichen USB-Datenspeicher.
5	I/O-Status-LEDs	Diagnose der Spannungsversorgung für Embedded-PC und Klemmbus. Status der E-Bus bzw. K-Bus Kommunikation und Multifunktions-I/Os.
6	Diagnose-LEDs	1 x TwinCAT-Status, 1 x Flash-Zugriff, 1 x Error-LED.
7	Federkraftklemmen, +24 V und 0V	Spannungsversorgung (Us) für Embedded-PC.
8	Federkraftklemmen, +24 V und 0V	Spannungsversorgung (Up) für integrierte Multifunktions-I/Os und Busklemmen über die Powerkontakte.

4.2 Typenschild

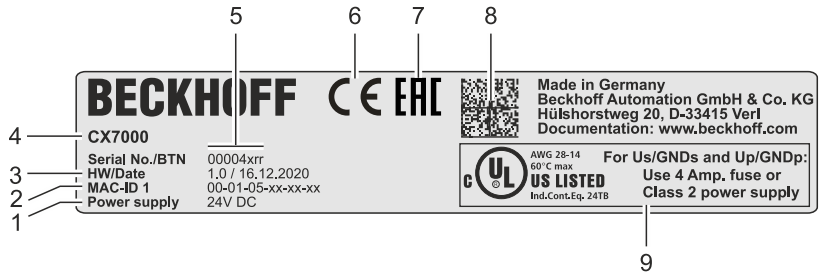


Abb. 2: Typenschild Beispielansicht.

Tab. 3: Informationen auf dem Typenschild.

Nr.	Beschreibung
1	Spannungsversorgung 24 V DC.
2	MAC-Adressen der eingebauten Ethernet-Schnittstelle.
3	Hardwarestand und Herstelldatum.
4	Produktbezeichnung zur Identifikation des Embedded-PCs.
5	Seriennummer/ Beckhoff Traceability Number (BTN) zur eindeutigen Identifizierung des Produkts. Der Hostname wird aus BTN- und der Seriennummer/ Beckhoff Traceability Number (BTN) gebildet. Beispiel: Aus der BTN 00004xrr ergibt sich der Hostname **BTN-00004xrr**.
6	CE-Kennzeichnung
7	EAC-Kennzeichnung
8	Maschinenlesbare Information in Form eines Data-Matrix-Codes (DMC, Code-Schema ECC200) der zur besseren Identifikation und Verwaltung genutzt werden kann.
9	UL-Kennzeichnung mit vorgeschriebenen Angaben zu Spannungsversorgung, Sicherung, Temperatur und Kabelquerschnitten.

4.3 Ethernet-Schnittstelle (X001)

Sie können den Embedded-PC CX7050 über die Ethernet-Schnittstelle X001 programmieren und in Betrieb nehmen. Die Ethernet-Schnittstelle erreicht Geschwindigkeiten von 10 / 100 Mbit/s.

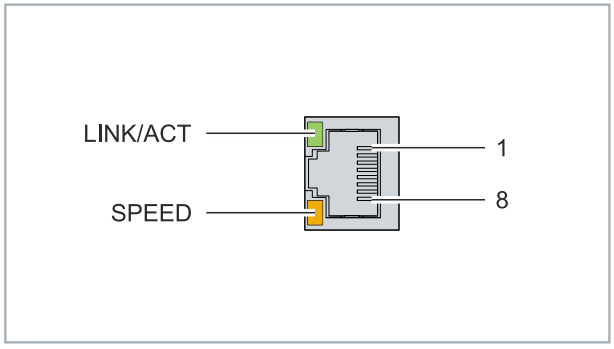


Abb. 3: Ethernet-Schnittstelle X001.

Die LEDs an der linken Seite der Schnittstelle zeigen den Status der Verbindung an. Die obere LED (LINK/ ACT) zeigt an, ob die Schnittstelle mit einem Netzwerk verbunden ist. Ist dies der Fall, leuchtet die LED grün auf. Wenn Daten auf der Schnittstelle übertragen werden, blinkt die LED.

Die untere LED (SPEED) zeigt die Geschwindigkeit der Verbindung an. Ist die Geschwindigkeit 10 Mbit/s, leuchtet die LED nicht. Bei 100 Mbit/s leuchtet die LED orange.

Tab. 4: Ethernet-Schnittstelle X001, PIN-Belegung.

PIN	Signal	Beschreibung
1	TD +	Transmit +
2	TD -	Transmit -
3	RD +	Receive +
4	connected	reserviert
5		
6	RD -	Receive -
7	connected	reserviert
8		

Übertragungsstandards

10Base5

Das Übertragungsmedium für 10Base5 ist ein dickes Koaxialkabel (Yellow Cable) mit einer max. Übertragungsgeschwindigkeit von 10 MBaud und einer Linien-Topologie mit Abzweigen (Drops), an die jeweils ein Teilnehmer angeschlossen wird. Da hier alle Teilnehmer an einem gemeinsamen Übertragungsmedium angeschlossen sind, kommt es bei 10Base5 zwangsläufig häufig zu Kollisionen.

10Base2

10Base2 (Cheaper net) ist eine Weiterentwicklung von 10Base5 und hat den Vorteil, dass dieses Koaxialkabel billiger und durch eine höhere Flexibilität einfacher zu verlegen ist. Es können mehrere Geräte an eine 10Base2-Leitung angeschlossen werden. Häufig werden die Abzweige eines 10Base5-Backbones als 10Base2 ausgeführt.

10BaseT

Beschreibt ein Twisted-Pair-Kabel für 10 MBaud. Hierbei wird das Netz sternförmig aufgebaut, so dass nun nicht mehr jeder Teilnehmer am gleichem Medium hängt. Dadurch führt ein Kabelbruch nicht mehr zum Ausfall des gesamten Netzes. Durch den Einsatz von Switches als Sternkoppler können Kollisionen vermindert oder bei Voll-Duplex Verbindungen auch vollständig vermieden werden.

100BaseT

Twisted-Pair-Kabel für 100 MBaud. Für die höhere Datengeschwindigkeit ist eine bessere Kabelqualität und die Verwendung entsprechender Hubs oder Switches erforderlich.

10BaseF

Der Standard 10BaseF beschreibt mehrere Lichtwellenleiter-Varianten.

Kurzbezeichnung der Kabeltypen für 10BaseT und 100BaseT

Twisted-Pair Kupferkabel für sternförmige Topologie, wobei der Abstand zwischen zwei Geräten 100 Meter nicht überschreiten darf.

UTP

Unshielded Twisted-Pair (nicht abgeschirmte, verdrillte Leitung)
Dieser Kabeltyp gehört zur Kategorie 3 und sind für industrielle Umgebungen nicht empfehlenswert.

S/UTP

Screened/Unshielded Twisted-Pair (mit Kupfergeflecht abgeschirmte, verdrillte Leitung)
Besitzen einen Gesamtschirm aus einem Kupfergeflecht zur Reduktion der äußeren Störeinflüsse. Dieses Kabel wird zum Einsatz mit den Buskopplern empfohlen.

FTP

Foilesshielded Twisted-Pair (mit Alufolie abgeschirmte, verdrillte Leitung)
Dieses Kabel hat eine alukaschierten Kunststoff-Folie-Gesamtschirm.

S/FTP

Screened/Foilesshielded Twisted-Pair (mit Kupfergeflecht und Alufolie abgeschirmte, verdrillte Leitung)
Besitzt einen alukaschierten Gesamtschirm mit einem darüber liegenden Kupfergeflecht. Solche Kabel können eine Störleistungsunterdrückung bis zu 70dB erreichen.

STP

Shielded Twisted-Pair (abgeschirmte, verdrillte Leitung)
Beschreibt ein Kabel mit Gesamtschirm ohne weitere Angabe der Art der Schirmung.

S/STP

Screened/Shielded Twisted-Pair (einzeln abgeschirmte, verdrillte Leitung)
Ein solche Bezeichnung kennzeichnet ein Kabel mit einer Abschirmung für jedes Leitungspaar sowie einen Gesamtschirm.

ITP

Industrial Twisted-Pair
Ist von Aufbau dem S/STP ähnlich, besitzt allerdings im Gegensatz zum S/STP nur 2 Leitungspaare.

4.4 USB-Schnittstelle (X002)

An die USB-Schnittstelle kann ein USB-Speicherstick angeschlossen und als zusätzlicher Speicher verwendet werden. Die USB-Schnittstelle unterstützt Übertragungsgeschwindigkeiten bis 12 Mbit/s und es können nicht mehr als 100 mA abgegeben werden. Der Dateizugriff erfolgt aus TwinCAT bzw. dem SPS-Programm mit Hilfe der dazugehörigen Funktionsbausteine. Es können keine anderen Geräte an die USB-Schnittstelle angeschlossen und verwendet werden.

Die gleiche Funktionsweise lässt sich für Dateizugriffe auf die MicroSD-Karte nutzen. Benutzen Sie für Zugriffe auf die MicroSD-Karte *C:*\ und für Zugriffe auf den USB-Speicherstick *D:*\ als Laufwerksbuchstaben.

Funktionsbausteine für Dateizugriffe

Mit den nachfolgend aufgeführten Funktionsbausteinen können lokal auf dem PC Dateien aus der SPS heraus bearbeitet werden. Durch die AMS-Netzwerkadresse wird das TwinCAT-Zielsystem identifiziert. Durch diesen Mechanismus ist es unter anderem möglich, Dateien auf anderen TwinCAT-Systemen des Verbundes anzulegen bzw. zu bearbeiten. Der Zugriff auf Dateien besteht aus drei aufeinanderfolgenden Phasen:

1. Öffnen der Datei
2. Lesender oder schreibender Zugriff auf die geöffnete Datei
3. Schließen der Datei

Das Öffnen der Datei dient dazu, eine temporäre Verbindung zwischen der externen Datei, von der zunächst nur der Name bekannt ist, und dem laufenden Programm herzustellen. Das Schließen der Datei dient dazu, das Ende der Bearbeitung anzuzeigen und sie in einen definierten Ausgangszustand für die Bearbeitung durch andere Programme zu versetzen.

Name	Beschreibung
FB_EOF	Testen auf Dateiende
FB_FileOpen	Öffnen einer Datei
FB_FileClose	Schließen einer Datei
FB_FileGets	String aus einer Text-Datei lesen
FB_FilePuts	Nullterminierten-String in eine Text-Datei schreiben
FB_FileRead	Lesen aus einer Datei
FB_FileWrite	Schreiben in eine Datei
FB_FileSeek	Verstellen des Dateizeigers
FB_FileTell	Ermitteln der aktuellen Position des Dateizeigers
FB_FileDelete	Löschen einer Datei
FB_FileRename	Umbenennen einer Datei
FB_CreateDir	Erstellen eines neuen Verzeichnisses
FB_RemoveDir	Löschen eines Verzeichnisses

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_System (System)

4.5 D-Sub-Stecker (X003)

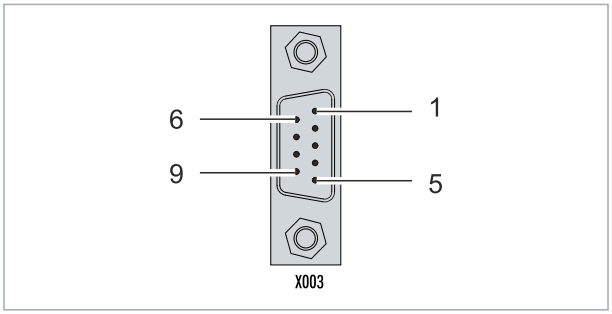


Abb. 4: CANopen-Schnittstelle X003.

Die CAN-Busleitung wird über eine 9polige D-Sub-Stecker mit folgender Belegung angeschlossen:

Pin	Belegung
1	nicht benutzt
2	CAN low (CAN-)
3	CAN Ground (intern verbunden mit Pin 6)
4	nicht benutzt
5	Schirm
6	CAN Ground (intern verbunden mit Pin 3)
7	CAN high (CAN+)
8	nicht benutzt
9	nicht benutzt

Die Hutschienenkontaktfeder und der Steckerschirm sind durchverbunden. An Pin 9 darf eine Hilfsspannung bis 30 V_{DC} angeschlossen werden, die von manchen CAN-Geräten zur Versorgung der Transceiver genutzt wird.

4.6 MicroSD-Karte

In der Grundausstattung enthält der CX7050 eine 512 MB MicroSD-Karte. Sie können den Embedded-PC optional mit einer größeren MicroSD-Karte (1 GB, 2 GB, 4 GB, 8 GB oder 16 GB) bestellen.

Die verwendeten Karten sind SLC-Speicher mit erweiterten Temperaturbereich für industrielle Anwendungen. Verwenden Sie ausschließlich von Beckhoff freigegebene MicroSD-Karten.

Bestellbezeichnung	Kapazität	Beschreibung
CX1900-0123	1 GB	MicroSD-Karte (SLC-Speicher) mit erweitertem Temperaturbereich für industrielle Anwendungen anstelle der 512 MB Karte (Bestelloption)
CX1900-0125	2 GB	
CX1900-0127	4 GB	
CX1900-0129	8 GB	
CX1900-0131	16 GB	

Bestellbezeichnung	Kapazität	Beschreibung
CX1900-0122	512 MB	MicroSD-Karte (SLC-Speicher) mit erweitertem Temperaturbereich für industrielle Anwendungen als Ersatzteil.
CX1900-0124	1 GB	
CX1900-0126	2 GB	
CX1900-0128	4 GB	
CX1900-0130	8 GB	
CX1900-0132	16 GB	

4.7 CANopen-Systemübersicht

CANopen ist eine weit verbreitete CAN-Anwendungsschicht, die im Verband CAN-in-Automation (CiA, http://www.can-cia.org) entwickelt und zur internationalen Normung angenommen wurde.

Gerätemodell

CANopen besteht aus der Protokolldefinition (Kommunikationsprofil) sowie den Geräteprofilen, die den Dateninhalt für die jeweilige Geräteklasse normieren. Zur schnellen Kommunikation der Ein- und Ausgangsdaten dienen die Prozessdatenobjekte (PDO) [▶ 27]. Die CANopen-Geräteparameter und Prozessdaten sind in einem Objektverzeichnis strukturiert. Der Zugriff auf beliebige Daten dieses Objektverzeichnisses erfolgt über die Servicedatenobjekte (SDO). Weiter gibt es einige Spezialobjekte (bzw. Telegrammarten) für Netzwerkmanagement (NMT), Synchronisation, Fehlermeldungen etc.

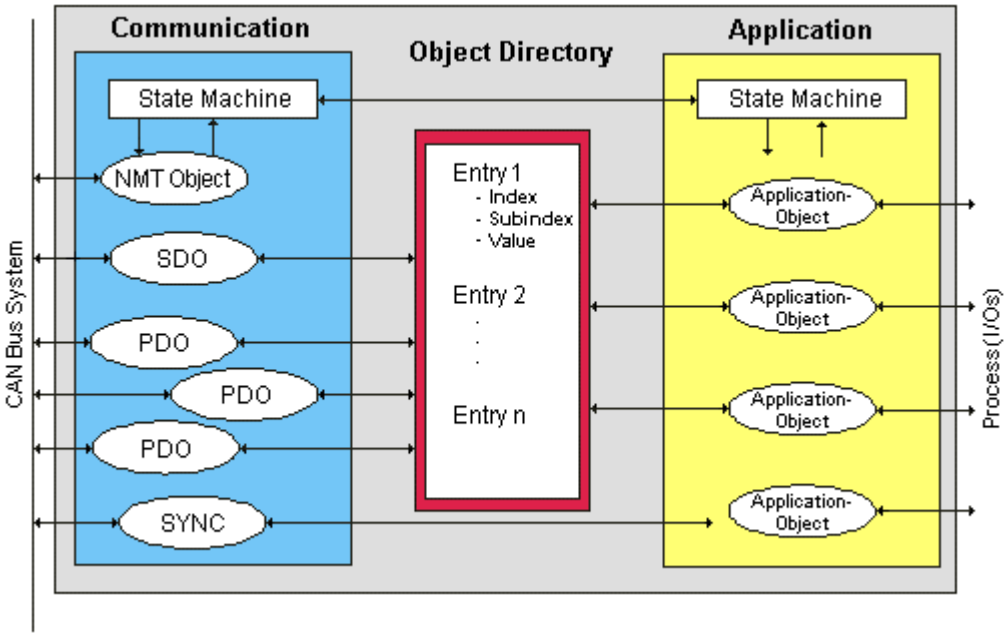


Abb. 5: CANopen Gerätemodell

Kommunikationsarten

CANopen definiert mehrere Kommunikationsarten für die Ein- und Ausgangsdaten (Prozessdatenobjekte):

- Ereignisgesteuert [▶ 29]: Telegramme werden versendet, sobald sich der Inhalt geändert hat. Hier wird nicht ständig das Prozessabbild, sondern nur die Änderung desselben übertragen.
- Zyklisch synchron [▶ 29]: Über ein SYNC Telegramm werden die Baugruppen veranlasst, die vorher empfangenen Ausgangsdaten zu übernehmen und neue Eingangsdaten zu senden.
- Angefordert (gepollt) [▶ 27]: Über ein CAN Datenanforderungstelegramm werden die Baugruppen veranlasst ihre Eingangsdaten zu senden.

Die gewünschte Kommunikationsart wird über den Parameter Transmission Type [▶ 27] eingestellt.

Geräteprofil

Die BECKHOFF CANopen-Geräte unterstützen alle E/A- Kommunikationsarten und entsprechen dem Geräteprofil für digitale und analoge Ein-/Ausgabebaugruppen (DS401 Version 1). Aus Gründen der Abwärtskompatibilität wurde das Default Mapping nicht der Profilversion DS401 V2 angepasst.

Übertragungsraten

Neun Übertragungsraten von 10 kBit/s bis 1 MBit/s stehen für unterschiedliche Buslängen zur Verfügung. Durch die effektive Nutzung der Busbandbreite erreicht CANopen kurze Systemreaktionszeiten bei vergleichsweise niedrigen Datenraten.

Topologie

CAN basiert auf einer linienförmigen Topologie. Die Anzahl der Teilnehmer pro Netz ist dabei von CANopen logisch auf 128 begrenzt, physikalisch erlaubt die aktuelle Treiber-Generation bis zu 64 Knoten in einem Netzsegment. Die bei einer bestimmten Datenrate maximal mögliche Netzausdehnung ist durch die auf dem Busmedium erforderliche Signallaufzeit begrenzt. Bei 1 MBit/s ist z. B. eine Netzausdehnung von 25 m, bei 50 kBit/s eine Netzausdehnung von 1000 m möglich. Bei niedrigen Datenraten kann die Netzausdehnung durch den Einsatz von Repeatern erhöht werden, diese ermöglichen auch den Aufbau von Baumstrukturen.

Buszugriffsverfahren

CAN arbeitet nach dem Verfahren Carrier Sense Multiple Access (CSMA), d.h. jeder Teilnehmer ist bezüglich des Buszugriffs gleichberechtigt und kann auf den Bus zugreifen, sobald dieser frei ist (Multi-Master-Buszugriff). Der Nachrichtenaustausch ist dabei nicht Teilnehmerbezogen sondern Nachrichtenbezogen. Das bedeutet, dass jede Nachricht mit einem priorisierten Identifier eindeutig gekennzeichnet ist. Damit beim Verschicken der Nachrichten verschiedener Teilnehmer keine Kollisionen auf dem Bus entstehen, wird beim Start der Datenübertragung eine bitweise Busarbitrierung durchgeführt. Die Busarbitrierung vergibt die Busbandbreite an die Nachrichten in der Reihenfolge ihrer Priorität, am Ende der Arbitrierungsphase belegt jeweils nur ein Busteilnehmer den Bus, Kollisionen werden vermieden und die Bandbreite wird optimal genutzt.

Konfiguration und Parametrierung

Mit dem TwinCAT System Manager können alle CANopen Parameter komfortabel eingestellt werden. Für die Parametrierung der Beckhoff CANopen-Geräte mit Konfigurationstools dritter Hersteller steht Ihnen auf der Beckhoff Website (http://www.beckhoff.de) ein eds-File (electronic data sheet) zur Verfügung.

Zertifizierung

Die Beckhoff CANopen-Geräte verfügen über eine leistungsfähige Protokollimplementierung und sind vom Verband CAN-in-Automation (http://www.can-cia.org) zertifiziert.

4.7.1 Netzwerkmanagement

Einfacher Boot-Up

CANopen erlaubt einen sehr einfachen Boot-Up des verteilten Netzwerkes. Die Module befinden sich nach der Initialisierung automatisch im Zustand *Pre-Operational*. In diesem Zustand kann bereits über Service-Datenobjekte (SDOs) mit Default-Identifiern auf das Objektverzeichnis zugegriffen werden, die Module können also konfiguriert werden. Da für alle Einträge im Objektverzeichnis Default-Einstellungen vorhanden sind, kann in den meisten Fällen auf eine Konfiguration verzichtet werden.

Zum Starten der Module ist dann nur eine einzige CAN-Nachricht erforderlich: Start_Remote_Node: Identifier *0*, zwei Datenbytes: 0x01, 0x00. Sie überführt die Knoten in den Zustand *Operational*.

Netzwerkstatus

Die Zustände im CANopen Boot-Up und die Zustandsübergänge sind aus dem Zustandsdiagramm ersichtlich:

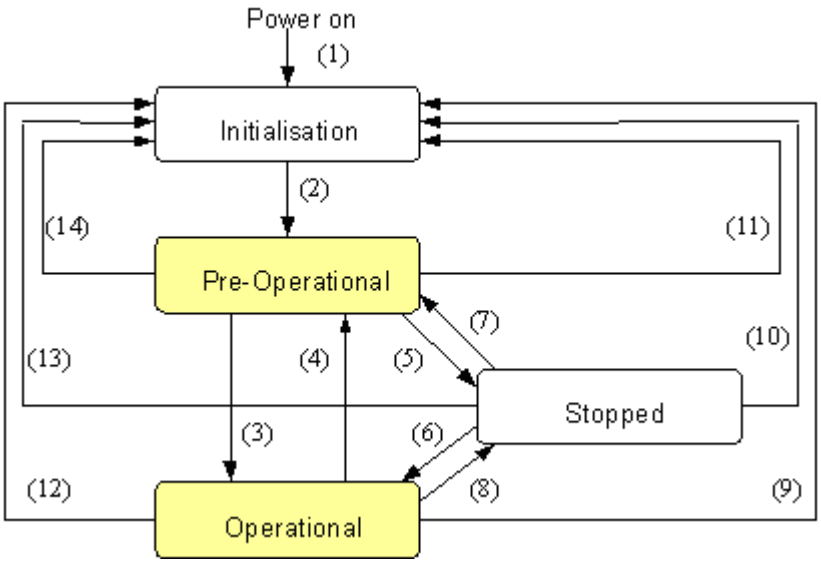


Abb. 6: Zustandsdiagramm CANopen Boot-up

Pre-Operational

Nach der Initialisierung geht der Buskoppler automatisch, d.h. ohne Befehl von außen, in den Zustand *Pre-Operational* über. In diesem Zustand kann er konfiguriert werden, denn die Servicedatenobjekte (SDOs) sind bereits aktiv. Die Prozessdatenobjekte sind hingegen noch gesperrt.

Operational

Im Zustand *Operational* sind auch die Prozessdatenobjekte aktiv.

Wenn der Buskoppler aufgrund äußerer Einflüsse (z. B. CAN-Störung, keine Ausgangs-Spannung) oder innerer Einflüsse (z. B. K-Bus-Fehler) nicht mehr in der Lage ist, Ausgänge zu setzen oder Eingänge zu lesen bzw. zu kommunizieren, so versucht er eine entsprechende Emergency-Nachricht zu senden, geht in den Fehlerzustand und fällt dabei in den Zustand *Pre-Operational* zurück. Damit kann auch die NMT-Statusmaschine des Netzwerkmasters fatale Fehler sofort erkennen.

Stopped

Im Zustand *Stopped* (früher *Prepared*) ist keine Datenkommunikation mit dem Koppler möglich - lediglich NMT-Nachrichten werden empfangen. Die Ausgänge gehen in den Fehlerzustand.

Statusübergänge

Die Netzwerkmanagement-Nachrichten haben einen sehr einfachen Aufbau: CAN-Identifier *0* mit zwei Byte Dateninhalt. Das erste Datenbyte enthält den sogenannten Command-Specifier (cs), das zweite Datenbyte die Knotenadresse, wobei die Knotenadresse *0* alle Knoten anspricht (Broadcast).

11-bit Identifier	2 Byte Nutzdaten							
0x00	cs	Node-ID						

Die folgende Tabelle gibt einen Überblick über alle CANopen Statusübergänge und die dazugehörigen Kommandos (Command Specifier im NMT Master-Telegramm):

Statusübergang	Command Specifier cs	Erläuterung
(1)	-	Der Initialisierungs-Status wird beim Einschalten selbsttätig erreicht
(2)	-	Nach der Initialisierung wird der Status Pre-Operational automatisch erreicht - dabei wird die Boot-Up-Nachricht abgeschickt.

Statusübergang	Command Specifier cs	Erläuterung
(3), (6)	cs = 1 = 0x01	Start_Remote_Node. Startet Modul, gibt Ausgänge frei, Startet Übertragung von PDOs.
(4), (7)	cs = 128 = 0x80	Enter_Pre-Operational. Stoppt PDO-Übertragung, SDO weiter aktiv.
(5), (8)	cs = 2 = 0x02	Stop_Remote_Node. Ausgänge gehen in den Fehlerzustand, SDO und PDO abgeschaltet.
(9), (10), (11)	cs = 129 = 0x81	Reset_Node. Führt Reset durch. Alle Objekte werden auf Power-On Defaults zurückgesetzt.
(12), (13), (14)	cs = 130 = 0x82	Reset_Communication. Führt Reset der Kommunikationsfunktionen durch. Objekte 0x1000 - 0x1FFF werden auf Power-On Defaults zurückgesetzt

Beispiel 1

Mit folgendem Telegramm werden netzwerkweit alle Baugruppen in den Fehlerzustand (Ausgänge sicherer Zustand) überführt:

11-bit Identifier	2 Byte Nutzdaten							
0x00	0x02	0x00						

Beispiel 2

Mit folgendem Telegramm wird Knoten 17 zurückgesetzt (resetted):

11-bit Identifier	2 Byte Nutzdaten							
0x00	0x81	0x11						

Boot-Up-Nachricht

Nach der Initialisierungsphase und dem Selbsttest sendet der Buskoppler die Boot-Up-Nachricht, eine CAN-Nachricht mit einem Datenbyte (0) auf dem Identifier der Guarding- bzw. Heartbeat-Nachricht: CAN-ID = 0x700 + Node-ID. Damit kann ein temporärer Ausfall einer Baugruppe während des Betriebs (z. B. durch einen Spannungseinbruch) oder eine nachträglich eingeschaltete Baugruppe zuverlässig auch ohne Node Guarding festgestellt werden. Der Sender kann über den Identifier der Nachricht (siehe Default-Identifier-Verteilung) bestimmt werden.

Außerdem ist es mit Hilfe der Boot-Up-Nachricht möglich, die beim Aufstarten am Netz befindlichen Knoten mit einem einfachen CAN-Monitor zu erkennen, ohne dass ein Schreibzugriff (z. B. Scannen des Netzwerks durch Auslesen von Parameter 0x1000) auf den Bus erforderlich ist.

Schließlich wird durch die Boot-Up-Nachricht das Ende der Initialisierungsphase kommuniziert; der Buskoppler signalisiert, dass er nun konfiguriert bzw. gestartet werden kann.

Firmwarestand BA

Bis Firmwarestand BA wurde für die Boot-Up-Nachricht der Emergency Identifier genutzt.

Format Boot-Up Nachricht

11-bit Identifier	1 Byte Nutzdaten							
0x700 (=1792) + Node-ID	0x00							

Knotenüberwachung

Für die Ausfallüberwachung des CANopen Netzwerkes stehen Heartbeat und Guarding-Mechanismen zur Verfügung. Diese sind bei CANopen besonders wichtig, da sich die Baugruppen in der ereignisgesteuerten Betriebsart nicht regelmäßig melden. Beim Guarding werden die Teilnehmer per Datenanforderungstelegramm (Remote Frame) zyklisch nach ihrem Status gefragt, beim Heartbeat senden die Knoten ihren Status von selbst.

Guarding: Node Guarding und Life Guarding

Über Node Guarding werden die dezentralen Peripherie-Baugruppen überwacht, die ihrerseits über Life Guarding den Ausfall des Guarding-Masters erkennen können. Beim Guarding setzt der Master Remote Frames (remote transmit request, Nachrichten-Anforderungstelegramme) auf die Guarding Identifier der zu überwachenden Slaves ab. Diese antworten mit der Guarding-Nachricht. Diese enthält den Status-Code des Slaves sowie ein Toggle-Bit, das nach jeder Nachricht wechseln muss. Falls Status- oder Toggle-Bit nicht mit den vom NMT-Master erwarteten übereinstimmen oder falls keine Antwort erfolgt geht der Master von einem Slave-Fehler aus.

Guarding-Verfahren

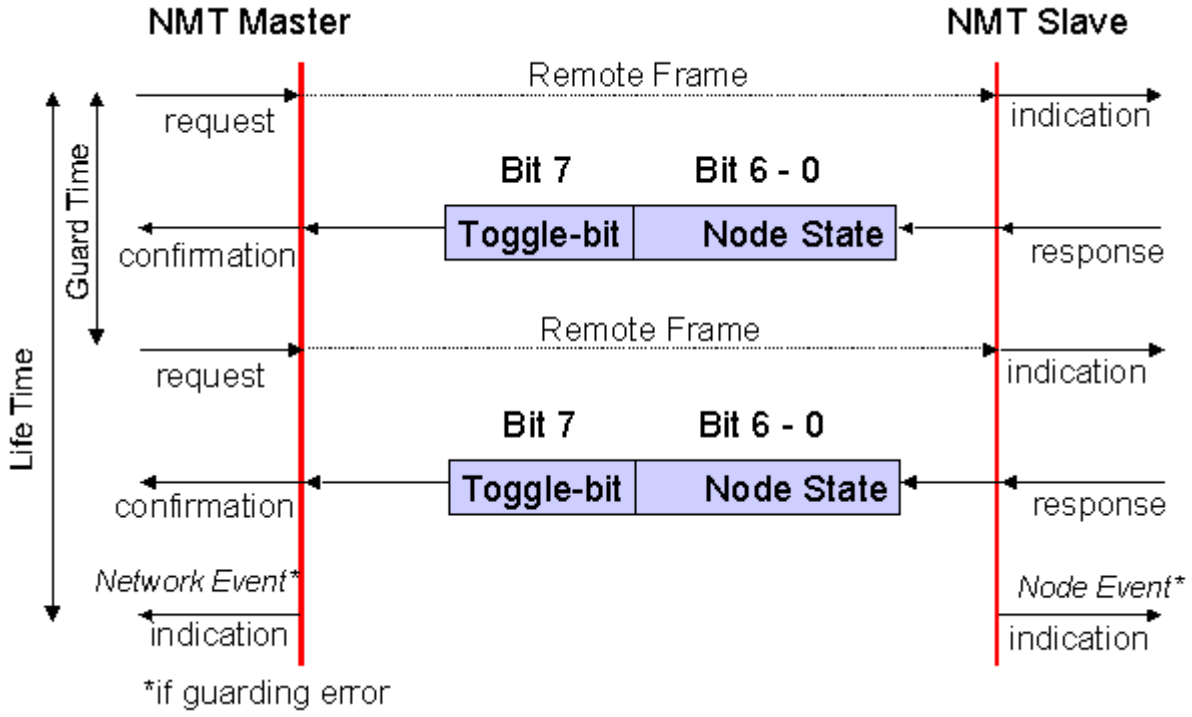


Abb. 7: Schematische Darstellung „Guarding-Verfahren“

Protokoll

Das im ersten Guarding-Telegramm übertragene Toggle-Bit (t) hat den Wert *0*. Anschließend wechselt (toggelt) das Bit in jedem Guarding-Telegramm und signalisiert so, ob ein Telegramm verloren ging. In den restlichen sieben Bit gibt der Knoten seinen Netzwerk Status (s) an:

s	Status
4 = 0x04	Stopped (früher: Prepared)
5 = 0x05	Operational
127 = 0x7F	Pre-Operational

Beispiel

Die Garding Nachricht des Knotens 27 (0x1B) muss mit einem Remote Frame mit Identifier 0x71B (1819_{dez}) angefragt werden. Wenn der Knoten *Operational* ist, wechselt das erste Datenbyte der Antwort-Nachricht zwischen 0x05 und 0x85, im Zustand *Pre-Operational* wechselt es zwischen 0x7F und 0xFF.

Guard Time und Life Time Factor

Wenn der Master die Guard-Nachrichten streng zyklisch anfordert, kann der Slave den Ausfall des Masters erkennen. Falls der Slave in diesem Fall innerhalb der eingestellten *Node Life Time* keine Nachrichtenanforderung vom Master erhält (Guarding-Fehler), geht er von einem Masterausfall aus (Watchdog-Funktion). Dann setzt er seine Ausgänge in den Fehlerzustand, sendet ein Emergency-Telegramm und fällt in den Zustand Pre-Operational zurück. Nach einem Guarding Time-Out kann das Verfahren durch Übertragen eines erneuten Guarding-Telegramms wieder angeregt werden.

Die Node Life-Time berechnet sich aus den Parametern Guard-Time (Objekt 0x100C) und Life-Time-Factor (Objekt 0x100D):

Life-Time = Guard-Time x Life-Time-Factor

Falls einer der beiden Parameter "0" ist (Default-Einstellung), erfolgt keine Überwachung des Masters (kein Life Guarding).

Heartbeat: Knotenüberwachung ohne Remote Frame

Beim Heartbeat-Verfahren senden die Knoten ihre jeweilige Statusmeldung zyklisch selbsttätig. Es kann daher auf Remote Frames verzichtet werden und es wird weniger Buslast erzeugt als beim Guarding-Verfahren.

Der Master sendet sein Heartbeat-Telegramm ebenfalls zyklisch, die Slaves können somit den Ausfall des Masters ebenfalls erkennen.

Heartbeat-Verfahren

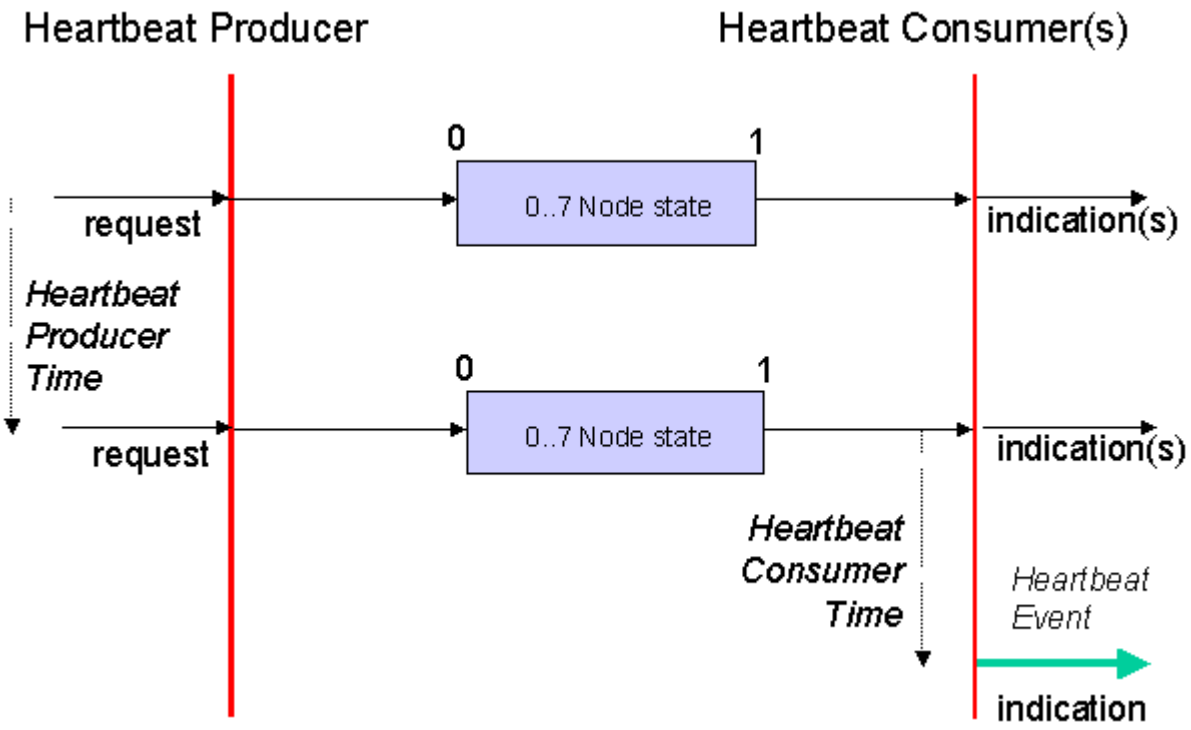


Abb. 8: Schematische Darstellung „Heartbeat-Verfahren“

Protokoll

Beim Heartbeat-Verfahren wird auf das Toggle-Bit verzichtet, die Knoten senden zyklisch Ihren Status (s). Siehe Guarding [▶ 25].

4.7.2 Prozessdatenobjekte (PDO)

Einführung

Bei vielen Feldbus-Systemen wird ständig das gesamte Prozessabbild übertragen - meist mehr oder weniger zyklisch. CANopen ist nicht auf dieses Kommunikationsprinzip beschränkt, da CAN durch die Multi-Master Buszugriffsregelung auch andere Möglichkeiten bietet: die Prozessdaten werden bei CANopen nicht im Master/Slave-Verfahren übertragen, sondern folgen dem Produzenten/Konsumenten-Modell (Producer/ Consumer). Hierbei sendet ein Busknoten seine Daten von sich aus (Producer), beispielsweise durch den Eintritt eines Ereignisses getriggert; alle anderen Knoten hören mit und entscheiden anhand des Identifiers, ob sie sich für dieses Telegramm interessieren und verarbeiten es entsprechend (Consumer).

Bei CANopen werden die Prozessdaten in Segmente zu maximal 8 Byte aufgeteilt. Diese Segmente heißen Prozessdatenobjekte (PDOs). Die PDOs entsprechen jeweils einem CAN-Telegramm und werden über dessen spezifischen CAN-Identifier zugeordnet und in ihrer Priorität bestimmt. Man unterscheidet Empfangs-PDOs (Receive-PDOs , RxPDOs) und Sende-PDOs (Transmit-PDOs , TxPDOs), wobei die Bezeichnung jeweils aus Gerätesicht erfolgt: eine Ein-/Ausgabebaugruppe sendet ihre Eingangsdaten mit TxPDOs, und empfängt die Ausgangsdaten in den RxPDOs. **Diese Bezeichnung wird im TwinCAT-System-Manager beibehalten.**

Kommunikationsparameter

Die PDOs können je nach Applikationsanforderung mit unterschiedlichen Kommunikationsparametern versehen werden. Wie alle CANopen-Parameter stehen auch diese im Objektverzeichnis des Gerätes, auf sie kann über die Servicedatenobjekte zugegriffen werden. Die Parameter für die Empfangs-PDOs stehen bei Index 0x1400 (RxPDO1) und folgende, bis zu 512 RxPDOs können vorhanden sein (Bereich bis Index 0x15FF). Entsprechend finden sich die Einträge für die Sende-PDOs bei Index 0x1800 (TxPDO1) bis 0x19FF (TxPDO512).

Für den Prozessdatenaustausch stehen auf den Beckhoff Buskopplern bzw. Feldbus Koppler Box Baugruppen jeweils 16 RxPDO und TxPDOs zur Verfügung (bei den Economy- und LowCost-Kopplern BK5110 und LC5100 sowie den Feldbus Boxen sind es jeweils 5 PDOs, da diese Geräte über weniger Prozessdaten verfügen). Die FC510x CANopen Master Karte unterstützt - beschränkt durch die DPRAM-Größe - je Kanal bis zu 192 Sende- und 192 Empfangs-PDOs. Die CANopen Klemme EL6751 organisiert das Prozessabbild dynamisch, d.h. die Prozessdaten werden hintereinander geschrieben, was eine höhere Datenübertragungsrate ermöglicht. Im Slave Mode können bis zu 32 TxPDOs und 32 RxPDOs verarbeitet werden.

Für jedes vorhandene Prozessdatenobjekt ist ein zugehöriges Kommunikationsparameter-Objekt vorhanden. Der TwinCAT-System-Manager ordnet die eingestellten Parameter automatisch den jeweiligen Objektverzeichniseinträgen zu. Im Folgenden werden diese Einträge samt ihrer Bedeutung für das Kommunikationsverhalten der Prozessdaten erläutert.

PDO-Identifier

Der wichtigste Kommunikationsparameter eines PDOs ist der CAN-Identifier (auch Communication Object Identifier, COB-ID genannt). Er dient zur Identifizierung der Daten und bestimmt deren Priorität beim Buszugriff. Für jedes CAN-Datentelegramm darf es nur einen Sendeknoten (Producer) geben; da CAN jedoch alle Nachrichten im Broadcast-Verfahren sendet kann ein Telegramm wie beschrieben von beliebig vielen Knoten empfangen werden (Consumer). Ein Knoten kann also seine Eingangsinformation mehreren Busteilnehmern gleichzeitig zur Verfügung stellen - auch ohne Weiterleitung durch einen logischen Busmaster. Der Identifier steht in Subindex 1 des Kommunikationsparametersatzes. Er ist als 32-Bit Wert kodiert, wobei die niederwertigsten 11 Bits (Bit 0...10) den eigentlichen Identifier enthalten. Die Datenbreite des Objektes von 32 Bit erlaubt auch den Eintrag von 29 Bit Identifiern nach CAN 2.0B, allerdings beziehen sich die Default-Identifier stets auf die üblichere 11 Bit-Variante. Allgemein geht CANopen sparsam mit den zur Verfügung stehenden Identifiern um, sodass der Einsatz der 29 Bit-Variante auf Sonderanwendungen beschränkt bleibt - und daher auch von den Beckhoff CANopen Geräten nicht unterstützt wird. Über das höchstwertige Bit (Bit 31) lässt sich das Prozessdatenobjekt aktivieren bzw. abschalten.

Im Anhang finden Sie eine komplette Identifier-Liste [▶ 204].

PDO Linking

Im System der Default-Identifier kommunizieren alle Knoten (hier: Slaves) mit einer Zentrale (Master), da kein Slave-Knoten per Default auf die Sende-Identifier eines anderen Slave-Knotens hört).

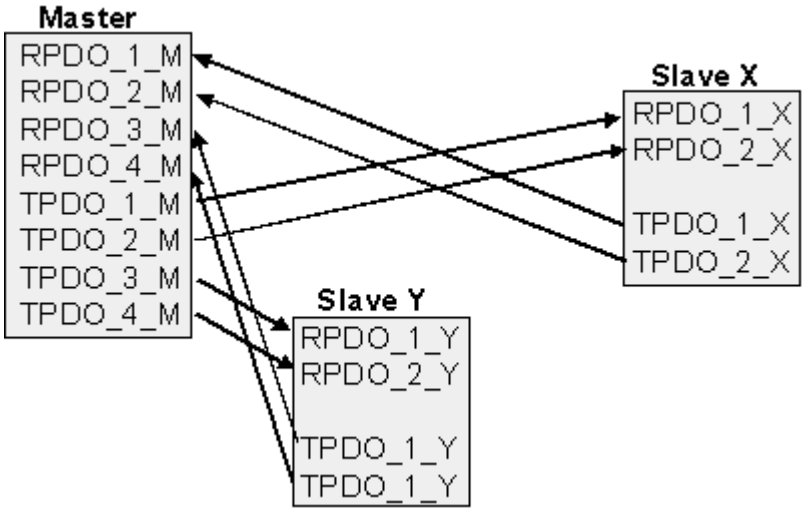


Abb. 9: Default Identifier-Verteilung: Master/Slave

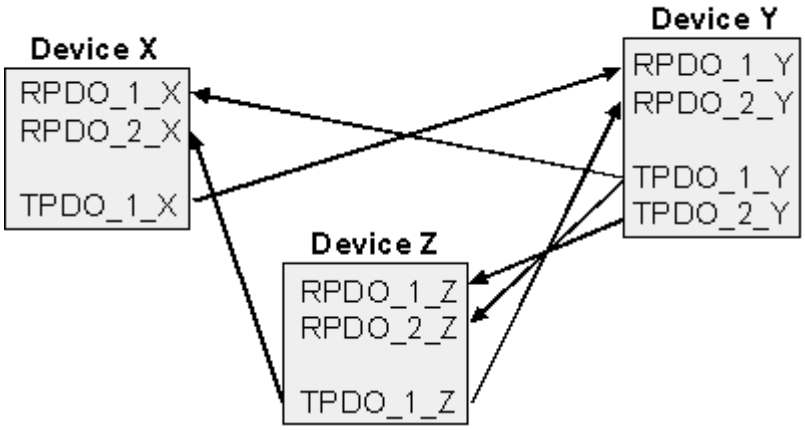


Abb. 10: PDO Linking: Peer to Peer

Wenn das Consumer-Producer-Modell der CANopen PDOs zum direkten Datenaustausch zwischen Knoten (ohne Master) genutzt werden soll, so muss die Identifier-Verteilung entsprechend angepasst werden, damit der TxPDO-Identifier des Producers mit dem RxPDO-Identifier des Consumers übereinstimmt. Dieses Verfahren nennt man PDO Linking. Es ermöglicht beispielsweise den einfachen Aufbau von elektronischen Getrieben, bei denen mehrere Slave-Achsen gleichzeitig auf den Ist-Wert im TxPDO der Master-Achse hören.

PDO-Kommunikationsarten: Überblick

CANopen bietet vielfältige Möglichkeiten, die Prozessdaten zu übertragen (siehe auch: Hinweise zur PDO Parametrierung [▶ 33])

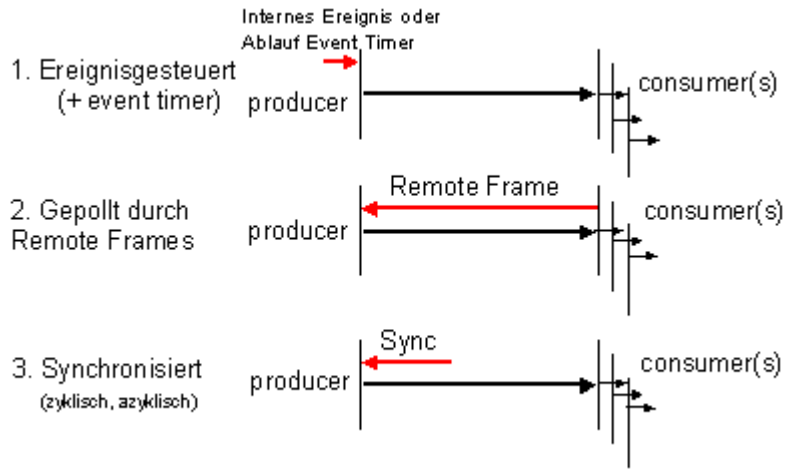


Abb. 11: Darstellung Übertragung CAN-Prozessdaten

Ereignisgesteuert

Das "Ereignis" ist die Änderung eines Eingangswertes, die Daten werden sofort nach dieser Änderung verschickt. Durch die Ereignissteuerung wird die Busbandbreite optimal ausgenutzt, da nicht ständig das Prozessabbild, sondern nur die Änderung desselben übertragen wird. Gleichzeitig wird eine kurze Reaktionszeit erreicht, da bei Änderung eines Eingangswertes nicht erst auf die nächste Abfrage durch einen Master gewartet werden muss.

Ab CANopen Version 4 kann die ereignisgesteuerte Kommunikationsart mit einem zyklischen Update kombiniert werden. Auch wenn gerade kein Ereignis aufgetreten ist, werden ereignisgesteuerte TxPDO nach Ablauf des Event Timers verschickt. Beim Auftreten eines Ereignisses wird der Event Timer zurückgesetzt. Bei RxPDOs wird der Event Timer als Watchdog benutzt um das Eintreffen von ereignisgesteuerten PDOs zu überwachen. Sollte innerhalb der eingestellten Zeit kein PDO eingetroffen sein, so geht der Busknoten in den Fehlerzustand.

Gepollt

Die PDOs können auch durch Datenanforderungstelegramme (Remote Frames) gepollt werden. Auf diese Art kann etwa das Eingangsprozessabbild bei ereignisgesteuerten Eingängen auch ohne deren Änderung auf den Bus gebracht werden, beispielsweise bei einem zur Laufzeit ins Netz aufgenommenen Monitor- oder Diagnosegerät. Das zeitliche Verhalten von Remote Frame und Antworttelegramm hängt von den verwendeten CAN-Controllern ab. Bausteine mit integrierter kompletter Nachrichtenfilterung ("FullCAN") beantworten ein Datenanforderungstelegramm in der Regel direkt und versenden sofort die im entsprechenden Sendebuffer stehenden Daten - dort muss die Applikation dafür Sorge tragen, dass die Daten ständig aktualisiert werden. CAN-Controller mit einfacher Nachrichtenfilterung (BasicCAN) reichen die Anforderung dagegen an die Applikation weiter, die nun das Telegramm mit den aktuellen Daten zusammenstellen kann. Das dauert länger, dafür sind die Daten aktuell. Beckhoff verwendet CAN Controller nach dem Basic CAN Prinzip.

Da dieses Geräteverhalten für den Anwender meist nicht transparent ist und zudem noch CAN-Controller in Verwendung sind, die Remote Frames überhaupt nicht unterstützen, kann die gepollte Kommunikationsart nur bedingt für den laufenden Betrieb empfohlen werden.

Synchronisiert

Nicht nur bei Antriebsanwendungen ist es sinnvoll, das Ermitteln der Eingangsinformation sowie das Setzen der Ausgänge zu synchronisieren. CANopen stellt hierzu das SYNC-Objekt zur Verfügung, ein CAN-Telegramm hoher Priorität ohne Nutzdaten, dessen Empfang von den synchronisierten Knoten als Trigger für das Lesen der Eingänge bzw. für das Setzen der Ausgänge verwendet wird.

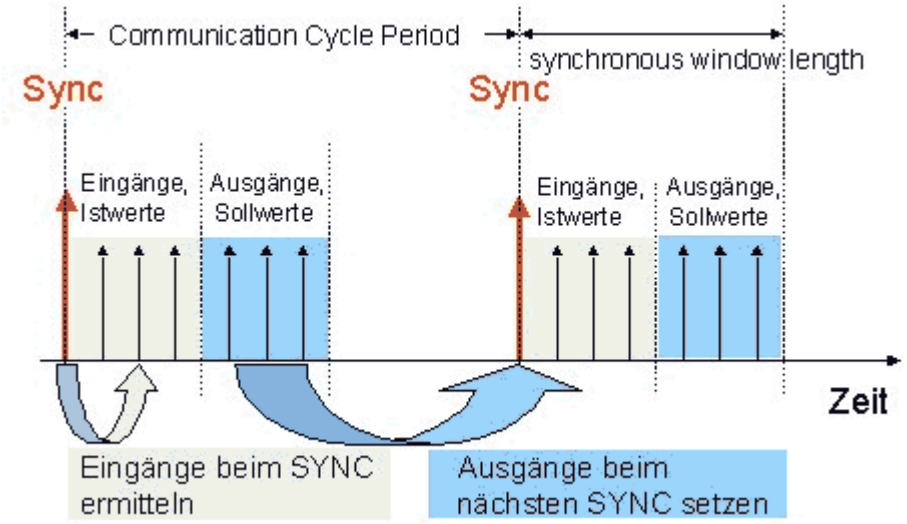


Abb. 12: Darstellung CAN Telegramm „SYNC“

PDO-Übertragungsart: Parametrierung

Der Parameter PDO-Übertragungsart (Transmission Type) legt fest, wie das Versenden des PDOs ausgelöst wird bzw. wie empfangene PDOs behandelt werden:

Übertragungsart	Zyklisch	Azyklisch	Synchron	Asynchron	Nur RTR
0		X	X		
1-240	X		X		
241-251	- reserviert -				
252			X		X
253				X	X
254, 255				X	

Die Übertragungsart wird für RxPDOs in den Objekten 0x1400ff, Subindex 2, und für TxPDOs in den Objekten 0x1800ff, Subindex 2 parametriert.

Azyklisch Synchron

PDOs der Übertragungsart 0 arbeiten synchron, aber nicht zyklisch. Ein RxPDO wird erst nach Empfang des nächsten SYNC-Telegramms ausgewertet. Damit lassen sich beispielsweise Achsgruppen nacheinander mit neuen Zielpositionen versehen, die alle beim nächsten SYNC gültig werden - ohne dass ständig Stützstellen ausgegeben werden müssen. Ein Gerät, dessen TxPDO auf Übertragungsart 0 konfiguriert ist, ermittelt seine Eingangsdaten beim Empfang des SYNC (synchrones Prozessabbild) und sendet sie anschließend, falls die Daten einem Ereignis entsprechen (beispielsweise eine Eingangsänderung) eingetreten ist. Die Übertragungsart 0 kombiniert also den Sendegrund "ereignisgesteuert" mit dem Sende- (und möglichst Sample-) bzw. Verarbeitungs-Zeitpunkt "SYNC-Empfang".

Zyklisch Synchron

Bei Übertragungsart 1-240 wird das PDO zyklisch gesendet: nach jedem "n-ten" SYNC (n=1...240). Da die Übertragungsart nicht nur im Netz, sondern auch auf einem Gerät kombiniert werden dürfen, kann so z. B. ein schneller Zyklus für digitale Eingänge vereinbart werden (n=1), während die Daten der Analogeingänge in einem langsameren Zyklus übertragen werden (z. B. n=10). RxPDOs unterscheiden in der Regel nicht zwischen den Übertragungsarten 0...240: ein empfangenes PDO wird beim nächsten SYNC-Empfang gültig gesetzt. Die Zykluszeit (SYNC-Rate) kann überwacht werden (Objekt 0x1006), das Gerät reagiert bei SYNC-Ausfall dann entsprechend der Definition des Geräteprofils und schaltet z. B. seine Ausgänge in den Fehlerzustand.

Die FC510x Karte / EL6751Klemme unterstützen die synchrone Kommunikationsart vollständig: das Versenden des SYNC Telegramms ist mit der verknüpften Task gekoppelt, sodass zu jedem Taskbeginn neue Eingangsdaten zur Verfügung stehen. Das Ausbleiben eines synchronen PDOs wird erkannt und an die Applikation gemeldet.

Nur RTR

Die Übertragungsarten 252 und 253 gelten für Prozessdatenobjekte, die ausschließlich auf Anforderung durch ein Remote Frame übertragen werden. 252 ist synchron: beim Empfang des SYNCs werden die Prozessdaten ermittelt, gesendet werden sie nur auf Anforderung. 253 ist asynchron, hier werden die Daten ständig ermittelt und auf Anforderung verschickt. Diese Übertragungsart ist generell nicht zu empfehlen, da das Abholen der Eingangsdaten von einigen CAN Controllern nur unvollständig unterstützt wird. Da die CAN Controller zudem teilweise selbsttätig auf Remote Frames antworten (ohne vorher aktuelle Eingangs-Daten anzufordern), ist die Aktualität der gepollten Daten unter Umständen fragwürdig. Die Übertragungsart 252 und 253 wird aus diesen Gründen von den Beckhoff PC-Karten / Klemmen nicht unterstützt.

Asynchron

Die Übertragungsarten 254 + 255 sind asynchron oder auch ereignisgesteuert. Bei Übertragungsart 254 ist das Ereignis herstellerspezifisch, bei 255 im Geräteprofil definiert. Im einfachsten Fall ist das Ereignis die Veränderung eines Eingangswertes - es wird also jede Werteänderung übertragen. Die Asynchrone Übertragungsart kann mit dem Event Timer gekoppelt werden und liefert so auch dann Eingangsdaten, wenn aktuell kein Ereignis aufgetreten ist.

Inhibit Zeit

Über den Parameter "Inhibit-Zeit" kann ein "Sende-Filter" aktiviert werden, der die Reaktionszeit bei der relativ ersten Eingangsänderung nicht verlängert, aber bei unmittelbar darauffolgenden Änderungen aktiv ist. Die Inhibit-Zeit (Sendeverzögerungszeit) beschreibt die Zeitspanne, die zwischen dem Versenden zweier gleicher Telegramme mindestens abgewartet werden muss. Wenn die Inhibit-Zeit genutzt wird, so kann die maximale Busbelastung und damit die Latenzzeit im "worst case"-Fall ermittelt werden.

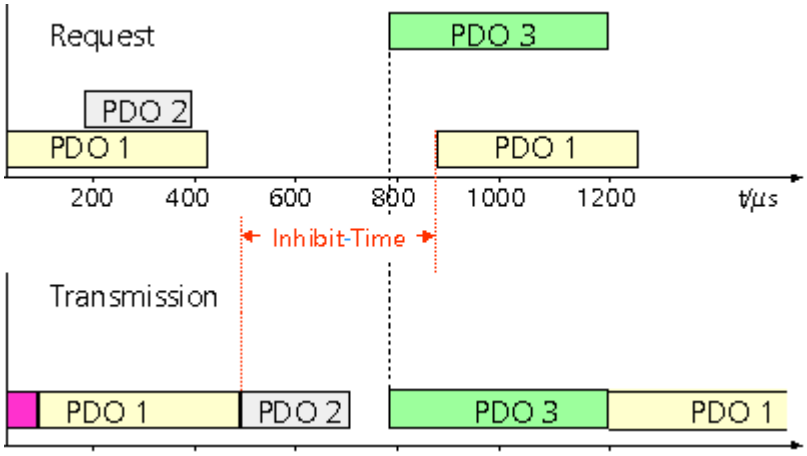


Abb. 13: Zeitl. Diagramm „Inhibit-Time“

Die Beckhoff PC-Karten FC510x / EL6751 Klemme können zwar die Inhibit-Zeit auf Slave-Geräten parametrieren, unterstützen sie jedoch selbst nicht. Eine Spreizung der gesendeten PDOs (Sendeverzögerung) ergibt sich automatisch aus der gewählten Zyklus-Zeit der SPS - und es macht wenig Sinn, die SPS schneller laufen zu lassen als es die Busbandbreite zulässt. Zudem kann die Busbelastung wirkungsvoll über die synchrone Kommunikation beeinflusst werden.

Event Timer

Über Subindex 5 der Kommunikationsparameter lässt sich ein Ereignis-Timer (Event Timer) für Sende-PDOs festlegen. Der Ablauf dieses Timers wird als zusätzlich eingetretenes Ereignis für das entsprechende PDO gewertet, das PDO wird also dann gesendet. Wenn das Applikationsereignis während einer Timer-Periode auftritt, so wird ebenfalls gesendet und der Timer wird zurückgesetzt.

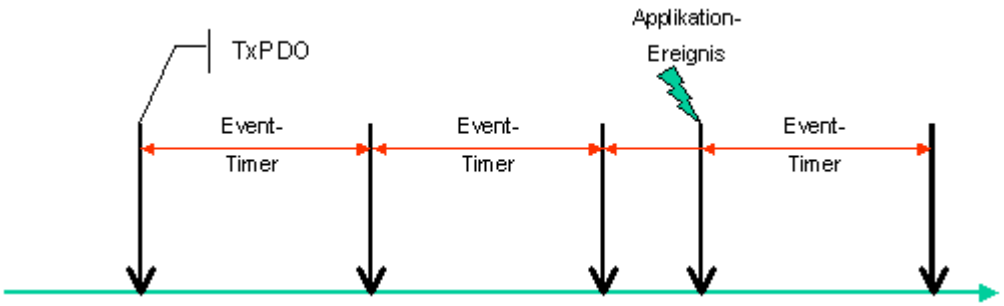


Abb. 14: Zeitliche Darstellung des Event-Timers

Bei Empfangs-PDOs wird der Timer-Parameter dazu verwendet, die Überwachungszeit für dieses PDO anzugeben: Die Applikation wird benachrichtigt, wenn kein entsprechendes PDO innerhalb der eingestellten Zeit empfangen wurde. Auf diese Art kann die FC510x / EL6751 jedes einzelne PDO individuell überwachen.

Hinweise zur PDO Parametrierung [▶ 33]

PDO Mapping

Unter PDO-Mapping versteht man die Abbildung der Applikationsobjekte (Echtzeitdaten) aus dem Objektverzeichnis in die Prozessdatenobjekte. Die CANopen-Geräteprofile sehen für jeden Gerätetyp ein Default Mapping vor, das für die meisten Anwendungen passend ist. So bildet das Default Mapping für digitale E/A einfach die Ein- bzw. Ausgänge ihrer physikalischen Reihenfolge gemäß in die Sende- bzw. Empfangs-Prozessdatenobjekte ab.

Die Default-PDOs für Antriebe enthalten jeweils 2 Byte Steuer- bzw. Statuswort und Soll- bzw. Istwert für die betreffende Achse.

Das aktuelle Mapping kann über entsprechende Einträge im Objektverzeichnis, die sogenannten Mapping-Tabellen, gelesen werden. An erster Stelle der Mapping Tabelle (Subindex 0) steht die Anzahl der gemappten Objekte, die im Anschluss aufgelistet sind. Die Tabellen befinden sich im Objektverzeichnis bei Index 0x1600 ff. für die RxPDOs bzw. 0x1A00ff für die TxPDOs.

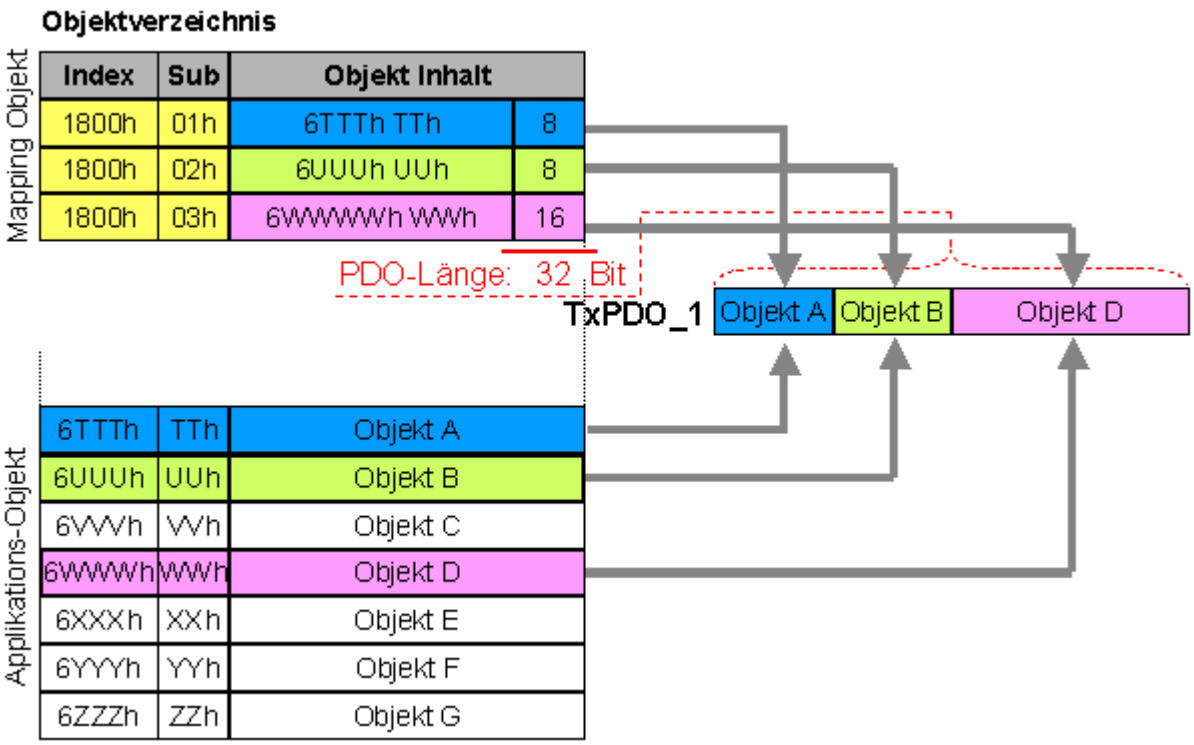


Abb. 15: Darstellung Mapping

Digitale und analoge Ein-/Ausgabebaugruppen: E/A-Anzahl auslesen

Die aktuelle Anzahl der digitalen und analogen Ein-/Ausgänge lässt sich durch Auslesen der entsprechenden Applikationsobjekte im Objektverzeichnis ermitteln bzw. verifizieren:

Parameter	Adresse Objektverzeichnis
Anzahl digitale Eingangsbytes	Index 0x6000, Subindex 0
Anzahl digitale Ausgangsbytes	Index 0x6200, Subindex 0
Anzahl analoge Eingänge	Index 0x6401, Subindex 0
Anzahl analoge Ausgänge	Index 0x6411, Subindex 0

Variables Mapping

In der Regel genügt die Default-Belegung der Prozessdatenobjekte (Default Mapping) bereits den Anforderungen. Für spezielle Anwendungsfälle kann die Belegung jedoch verändert werden: So unterstützen beispielsweise die Beckhoff CANopen Buskoppler das variable Mapping, bei dem die Applikationsobjekte (Ein- und Ausgangsdaten) frei den PDOs zugeordnet werden können. Hierzu müssen die Mapping-Tabellen konfiguriert werden: Ab CANopen Version 4 ist nur noch die folgende Vorgehensweise zulässig, die genau eingehalten werden muss:

1. Zunächst PDO löschen (0x1400ff, bzw. 0x1800ff, Subindex 1, Bit 31 auf "1" setzen)
2. Subindex 0 im Mapping Parameter (0x1600ff bzw. 0x1A00ff) auf "0" setzen
3. Mapping Einträge (0x1600ff bzw. 0x1A00ff, SI 1..8) verändern
4. Subindex 0 im Mapping Parameter auf gültigen Wert setzen. Das Gerät überprüft dann die Einträge auf Konsistenz.
5. PDO anlegen durch Eintragen d. Identifiers (0x1400ff bzw. 0x1800ff Subindex 1).

Dummy-Mapping

Ein weiteres Feature von CANopen ist das Mappen von Platzhaltern (Dummy-Einträgen). Als Platzhalter dienen die im Objektverzeichnis hinterlegten Datentyp-Einträge, die ja selbst nicht mit Daten versehen sind. Sind solche Einträge in der Mapping-Tabelle enthalten, so werden die entsprechenden Daten vom Gerät nicht ausgewertet. Auf diese Art können beispielsweise mehrere Antriebe über ein einziges CAN-Telegramm mit neuen Sollwerten versorgt werden oder Ausgänge auf mehreren Knoten auch im ereignisgesteuerten Modus gleichzeitig gesetzt werden.

4.7.3 PDO-Parametrierung

Auch wenn die meisten CANopen-Netze in der Default-Einstellung und damit mit minimalem Konfigurationsaufwand zufrieden stellend arbeiten, so sollte zumindest überprüft werden, ob die vorhandene Buslast vertretbar ist. 80% Busauslastung mag für ein rein zyklisch synchron arbeitendes Netzwerk akzeptabel sein, für ein rein ereignisgesteuertes Netz ist dieser Wert in der Regel zu hoch, da kaum Bandbreite für zusätzliche Ereignisse zur Verfügung steht.

Applikationsanforderungen berücksichtigen

Die Prozessdatenkommunikation sollte hinsichtlich einiger sich teilweise widersprechender Applikationsanforderungen optimiert werden. Hierzu gehören

- Geringer Parametrieraufwand - optimal sind brauchbare Default-Werte
- Garantierte Reaktionszeit auf bestimmte Ereignisse
- Zykluszeit bei Regelvorgängen über den Bus
- Sicherheitsreserven für Busstörungen (genügend Bandbreite für Nachrichtenwiederholung)
- Maximale Baud-Rate - hängt von der maximalen Buslänge ab
- Gewünschte Kommunikationspfade - wer spricht mit wem

Der bestimmende Faktor ist meist die zur Verfügung stehende Busbandbreite (Buslast).

Baud-Rate

Allgemein wird man beginnen, die Baud-Rate so groß zu wählen, wie es die Buslänge erlaubt. Hierbei sollte man berücksichtigen, dass serielle Bussysteme grundsätzlich um so empfindlicher auf Störeinflüsse reagieren, je höher die Baud-Rate ist. Es gilt also die Regel: so schnell wie nötig. 1000 kBit/s sind meist nicht erforderlich und uneingeschränkt nur bei Netzwerken innerhalb eines Schaltschrankes ohne galvanische Trennung der Busknoten empfehlenswert. Die Erfahrung zeigt auch, dass das Abschätzen der verlegten Buskabellänge häufig zu optimistisch erfolgt - die tatsächliche Kabellänge also größer ist.

Kommunikationsart bestimmen

Ist die Baud-Rate gewählt, so gilt es nun die PDO-Kommunikationsart(en) zu bestimmen. Diese haben unterschiedliche Vor- und Nachteile:

- Die zyklisch synchrone Kommunikation ergibt eine genau vorhersagbare Busbelastung und damit ein definiertes Zeitverhalten - man könnte auch sagen, der worst case ist Standard. Sie ist einfach zu konfigurieren: mit dem Parameter SYNC-Rate kann die Buslast global eingestellt werden. Die Prozessabbilder werden synchronisiert: Eingänge werden gleichzeitig gelesen, Ausgangsdaten gleichzeitig gültig gesetzt - die Qualität dieser Synchronisierung ist allerdings implementierungsabhängig. Die BECKHOFF PC-Karten FC510x / CANopen-Klemme EL6751 sind in der Lage, das CANopen Bussystems mit den Zyklen der Anwendungsprogramme (SPS bzw. NC) zu synchronisieren.

  Die garantierte Reaktionszeit ist bei der zyklisch synchronen Kommunikation immer mindestens so groß wie die Zykluszeit, und die Busbandbreite wird nicht optimal genutzt, da auch alte, sich nicht ändernde Daten ständig übertragen werden. Es ist aber möglich, das Netz durch die Wahl unterschiedlicher SYNC-Vielfacher (Transmission Types 1...240) zu optimieren und sich langsam ändernde Daten seltener zu übertragen als z. B. zeitkritische Eingänge. Berücksichtigt werden sollte jedoch, dass Eingangszustände, die kürzer anstehen als die Zykluszeit, nicht unbedingt kommuniziert werden. Ist dies gefordert, so sollten die entsprechenden PDOs für asynchrone Kommunikation vorgesehen werden.
- Die ereignisgesteuerte, asynchrone Kommunikation ist optimal hinsichtlich Reaktionszeit und Verwendung der Busbandbreite - man könnte sie als "CAN pur" bezeichnen. Bei ihrer Wahl muss allerdings berücksichtigt werden, dass unter Umständen viele Ereignisse gleichzeitig auftreten und sich dann entsprechende Verzögerungszeiten einstellen können, bis ein relativ niederpriores PDO verschickt werden kann - eine seriöse Netzwerkplanung erfordert demnach eine worst-case Betrachtung. Auch muss, z. B. durch Verwendung der Inhibit Zeit [▶ 27], verhindert werden, dass ein sich ständig ändernder Eingang mit hoher PDO-Priorität den Bus blockiert (Fachbegriff: "babbling idiot"). Aus diesem Grund ist beispielsweise die Ereignissteuerung bei Analogeingängen im Geräteprofil per Default abgeschaltet und muss gezielt aktiviert werden. Über den Ablauf-Timer lassen sich Zeitfenster für die Sende-PDOs einstellen: Das Telegramm wird frühestens nach Ablauf der Inhibit-Zeit [▶ 27] und spätestens nach Verstreichen des Ablauf-Timers erneut gesendet.
- Parametriert wird die Kommunikationsart über den Transmission Type [▶ 27].

Es ist auch möglich, beide PDO-Kommunikationsprinzipien zu kombinieren. So kann es beispielsweise sinnvoll sein, die Soll- und Istwerte einer Achsregelung zyklisch synchron auszutauschen, während Endschalter oder die mit Grenzwerten versehene Motortemperatur mit ereignisgesteuerten PDOs überwacht werden. So kombiniert man die Vorteile beider Prinzipien: Synchronität der Achskommunikation und kurze Reaktionszeit für Endschalter. Durch die dezentrale Grenzwertüberwachung wird trotz Ereignissteuerung vermieden, dass der Temperatur-Analogwert ständig zur Buslast beiträgt.

Im genannten Beispiel kann es auch sinnvoll sein, die Identifier-Verteilung gezielt zu beeinflussen, um den Buszugriff durch die Prioritätsverteilung zu optimieren: die höchste Priorität bekommt das PDO mit den Endschalterdaten, die niedrigste das mit den Temperaturwerten.

In aller Regel ist es aber nicht erforderlich, die Identifier-Verteilung anzupassen, um die Latenzzeit beim Buszugriff zu optimieren. Dagegen müssen die Identifier verändert werden, um eine masterlose Kommunikation zu ermöglichen (PDO Linking [▶ 27]). Im genannten Beispiel könnte je ein RxPDO der Achsen denselben Identifier wie das TxPDO des Endschalters zugewiesen bekommen und dadurch eine Veränderung des Eingangswertes verzögerungsfrei empfangen.

Buslast bestimmen

In jedem Fall ist es sinnvoll, die Buslast zu bestimmen. Doch welche Buslastwerte sind zulässig bzw. sinnvoll? Unterscheiden sollte man zunächst den kurzfristigen Burst von Telegrammen, bei dem eine Anzahl CAN-Nachrichten direkt aufeinander folgt - kurzzeitig 100% Buslast. Das ist nur dann problematisch, wenn die dadurch ausgelöste Folge von Empfangsinterrupts auf den CAN-Knoten nicht mehr abgearbeitet werden kann, es also zu einem Datenüberlauf (CAN-Queue-Overrun) kommt. Das kann bei sehr hohen Baud-Raten (> 500 kBit/s) bei Knoten mit Software-Telegrammfilterung und relativ langsamen oder stark ausgelasteten Mikro-Controllern vorkommen, wenn z. B. eine direkte Folge von Remote Frames (diese enthalten keine Datenbytes und haben daher minimale Länge) auf dem Bus ist (bei 1 Mbit/s kann so alle 40 µs ein Interrupt erzeugt werden; Beispiel: ein NMT-Master sendet alle Guarding-Anforderungen direkt hintereinander). Durch geschickte Implementierung läßt sich das vermeiden, der Anwender sollte davon ausgehen können, dass von den Geräteanbietern hierfür Sorge getragen wurde. Ein Burst-Zustand ist z. B. direkt nach dem SYNC Telegramm völlig normal: vom SYNC getriggert versuchen alle synchron arbeitenden Knoten quasi gleichzeitig Ihre Daten zu senden, es finden viele Arbitrierungsvorgänge statt, die Telegramme sortieren sich nacheinander in der Reihenfolge ihrer Priorität auf den Bus. Das ist in der Regel unkritisch, da es sich hier um Telegramme mit einigen Datenbytes handelt und die Telegrammfolge damit zwar eine schnelle, aber überschaubare Folge von Empfangsinterrupts auf den CAN-Knoten auslöst.

Unter Buslast versteht man meist den gemittelten Wert über mehrere Primärzyklen, also z. B. das Mittel über 100-500 ms. CAN, und damit CANopen, ist zwar in der Lage, nahe 100% Buslast auf Dauer zu bewältigen, aber dann steht keine Bandbreite für eventuelle Wiederholungen bei Störeinflüssen, asynchrone Fehlermeldungen, Parametrierung etc. zur Verfügung. Selbstverständlich hat die vorherrschende Art der Kommunikation einen großen Einfluss auf die sinnvolle Buslast: ein komplett zyklisch synchron arbeitendes Netz befindet sich ja bereits nahe am worst case Zustand und kann daher mit Werten von 70-80% betrieben werden. Für ein rein ereignisgesteuertes Netz ist diese Zahl nur schwer anzugeben: es muss hier abgeschätzt werden, wie viele zusätzliche Ereignisse im Vergleich zum derzeitigen Anlagenzustand auftreten können und für wie lange das zu einem Burst führt - also wie lange die relativ niederpriorste Nachricht dann verzögert würde. Ist dieser Wert von der Applikation her zulässig, so ist die aktuelle Buslast akzeptabel. Als Näherungswert kann meist angenommen werden, dass ein ereignisgesteuertes Netz mit 30-40% Grundlast genügend Reserven für worst-case-Szenarien hat - diese Annahme macht aber eine sorgfältige Analyse nicht überflüssig, wenn Verzögerungen zu kritischen Anlagenzuständen führen können.

Die BECKHOFF CANopen-Master-Karten FC510x / CANopen-Masterklemme EL6751 zeigen die Buslast über den System Manager ein. Diese Variable kann auch in der SPS verarbeitet oder in der Visualisierung zur Anzeige gebracht werden.

Neben den Kommunikationsparametern ist natürlich die Datenbelegung der Prozessdatenobjekte entscheidend: das PDO Mapping. [▶ 32]

4.7.4 Servicedatenobjekte (SDO)

Die im Objektverzeichnis aufgeführten Parameter werden über Servicedatenobjekte gelesen und beschrieben. Diese SDOs sind *Multiplexed Domains*, also Datenstrukturen beliebiger Größe, die mit einem Multiplexor (Adresse) versehen sind. Der Multiplexor besteht aus 16-Bit-Index und 8-Bit-Subindex, die die entsprechenden Einträge im Objektverzeichnis adressieren.

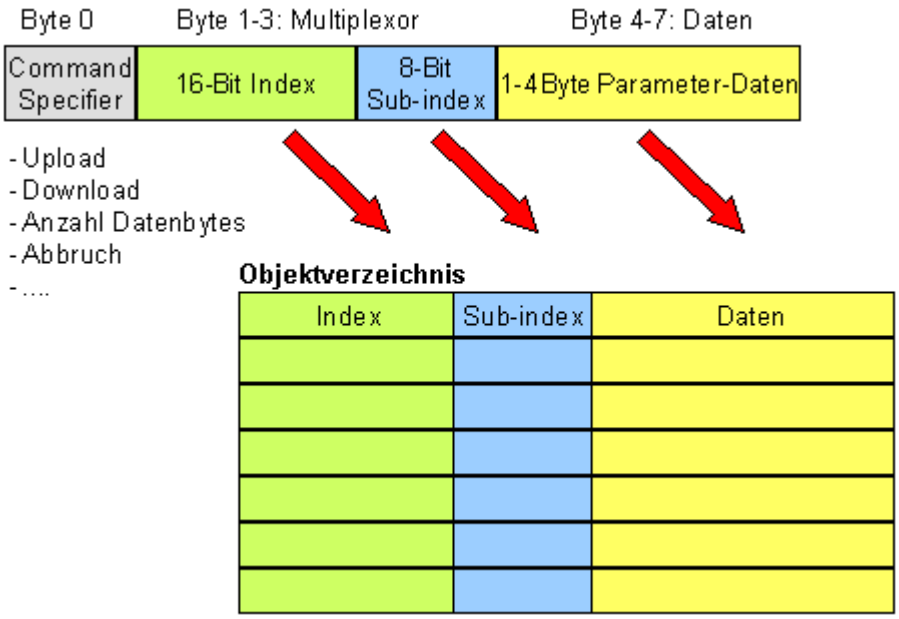


Abb. 16: SDO-Protokoll: Zugriff auf Objektverzeichnis

Die CANopen Buskoppler sind Server für das SDO, d.h. sie stellen auf Anforderung des Clients (z. B. des IPCs oder der SPS) Daten zur Verfügung (Upload) oder sie empfangen Daten vom Client (Download). Dabei findet ein Handshake zwischen Client und Server statt.

Wenn der zu übertragende Parameter bis zu 4 Bytes umfasst, genügt ein einziger Handshake (ein Telegrammpaar): Beim Download sendet der Client die Daten zusammen mit Index, Subindex und der Server bestätigt den Erhalt. Beim Upload fordert der Client die Daten an, indem er Index und Subindex des gewünschten Parameters überträgt, und der Server sendet den Parameter (incl. Index und Subindex) in seinem Antworttelegramm.

Für Upload und Download wird das gleiche Identifier-Paar verwendet. In den stets 8 Byte großen Telegrammen sind im ersten Datenbyte die unterschiedlichen Dienste codiert. Bis auf die Objekte 1008h, 1009h und 100Ah (Gerätename, Hardware- bzw. Softwareversion) sind alle Parameter der Buskoppler nur bis zu 4 Byte groß, daher beschränkt sich diese Beschreibung auf die Übertragung dieser Daten im beschleunigten Transfer (Expedited Transfer).

Protokoll

Im Folgenden wird der Aufbau der SDO-Telegramme beschrieben.

Client -> Server, Upload Request

11-bit Identifier	8 Byte Nutzdaten							
0x600 (=1536_{dez}) + Node-ID	0x40	Index0	Index1	SubIdx	0x00	0x00	0x00	0x00

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)

Client -> Server, Upload Response

11-bit Identifier	8 Byte Nutzdaten							
0x580 (=1408_{dez}) + Node-ID	0x4x	Index0	Index1	SubIdx	Data0	Data1	Data2	Data3

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)

Parameter	Erläuterung
SubIdx	Subindex (Unsigned8)
Data0	Daten Low-Low-Byte (LLSB)
Data3	Daten High-High-Byte (MMSB)

Parameter des Datentyps Unsigned8 werden im Byte D0 übertragen, Parameter des Typs Unsigned16 in D0 und D1.

Die Anzahl der gültigen Datenbytes ist im ersten CAN-Datenbyte (0x4x) wie folgt codiert:

Anzahl Parameter-Bytes	1	2	3	4
Erstes CAN-Datenbyte	0x4F	0x4B	0x47	0x43

Client -> Server, Download Request

11-bit Identifier	8 Byte Nutzdaten							
0x600 (=1536_{dez}) + Node-ID	0x22	Index0	Index1	SubIdx	Data0	Data1	Data2	Data3

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)
Data0	Daten Low-Low-Byte (LLSB)
Data3	Daten High-High-Byte (MMSB)

Optional ist es möglich, im ersten CAN-Datenbyte die Anzahl der gültigen Parameter-Datenbytes anzugeben

Anzahl Parameter-Bytes	1	2	3	4
Erstes CAN-Datenbyte	0x2F	0x2B	0x27	0x23

In der Regel ist das jedoch nicht erforderlich, da jeweils nur die niederwertigen Datenbytes bis zur Länge des zu beschreibenden Objektverzeichniseintrags ausgewertet werden. Ein Download von Daten bis zu 4 Byte Länge kann daher bei BECKHOFF Busknoten immer mit 22 h im ersten CAN-Datenbyte erfolgen.

Client -> Server, Download Response

11-bit Identifier	8 Byte Nutzdaten							
0x580 (=1408_{dez}) + Node-ID	0x60	Index0	Index1	SubIdx	0x00	0x00	0x00	0x00

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)

Abbruch Parameterkommunikation

Im Falle einer fehlerhaften Parameterkommunikation wird diese abgebrochen. Client bzw. Server senden dazu ein SDO-Telegramm folgender Struktur:

11-bit Identifier	8 Byte Nutzdaten							
0x580 (Client) oder 0x600 (Server) + Node-ID	0x80	Index0	Index1	SubIdx	Error0	Error1	Error2	Error3

Parameter	Erläuterung
Index0	Index Low-Byte (Unsigned16, LSB)
Index1	Index High-Byte (Unsigned16, MSB)
SubIdx	Subindex (Unsigned8)

Parameter	Erläuterung
Error0	SDO Fehler-Code Low-Low-Byte (LLSB)
Error3	SDO Fehler-Code High-High-Byte (MMSB)

Liste der SDO-Fehler-Codes (Abbruch-Grund des SDO-Transfers):

SDO-Fehler-Code	Erläuterung
0x05 03 00 00	Toggle Bit nicht geändert
0x05 04 00 01	SDO Command Specifier ungültig oder unbekannt
0x06 01 00 00	Zugriff auf dieses Objekt wird nicht unterstützt
0x06 01 00 02	Versuch, auf einen Read_Only Parameter zu schreiben
0x06 02 00 00	Objekt nicht im Objektverzeichnis vorhanden
0x06 04 00 41	Objekt kann nicht ins PDO gemappt werden
0x06 04 00 42	Anzahl und/oder Länge der gemappten Objekte würde PDO Länge überschreiten
0x06 04 00 43	Allgemeine Parameter Inkompatibilität
0x06 04 00 47	Allgemeiner interner Fehler im Gerät
0x06 06 00 00	Zugriff wegen Hardware-Fehler abgebrochen
0x06 07 00 10	Datentyp oder Parameterlänge stimmen nicht überein oder sind unbekannt
0x06 07 00 12	Datentyp stimmt nicht überein, Parameterlänge zu groß
0x06 07 00 13	Datentyp stimmt nicht überein, Parameterlänge zu klein
0x06 09 00 11	Subindex nicht vorhanden
0x06 09 00 30	allgemeiner Wertebereich-Fehler
0x06 09 00 31	Wertebereich-Fehler: Parameter wert zu groß
0x06 09 00 32	Wertebereich-Fehler: Parameter wert zu klein
0x06 0A 00 23	Resource nicht verfügbar
0x08 00 00 00	Allgemeiner Fehler
0x08 00 00 21	Zugriff wegen lokaler Applikation nicht möglich
0x08 00 00 22	Zugriff wegen aktuellem Gerätestatus nicht möglich

Für die Register-Kommunikation (Index 0x4500, 0x4501) wurden weitere, herstellerspezifische Fehler-Codes eingeführt:

SDO-Fehler-Code	Erläuterung
0x06 02 00 11	ungültige Tabelle: Tabelle oder Kanal nicht vorhanden
0x06 02 00 10	ungültiges Register: Tabelle nicht vorhanden
0x06 01 00 22	Schreibschutz noch gesetzt
0x06 07 00 43	fehlerhafte Anzahl Funktionsargumente
0x06 01 00 21	Funktion noch aktiv, später erneut versuchen
0x05 04 00 40	Allgemeiner Routing Fehler
0x06 06 00 21	Fehler Zugriff BC Tabelle
0x06 09 00 10	Allgemeiner Fehler bei Kommunikation mit Klemme
0x05 04 00 47	Time-out bei Kommunikation mit Klemme

4.7.5 Objektverzeichnis

4.7.5.1 Objektverzeichnis - Struktur

Im CANopen-Objektverzeichnis werden alle für den Buskoppler relevanten CANopen-Objekte eingetragen. Das Objektverzeichnis ist in drei verschiedene Bereiche aufgeteilt:

1. Kommunikationsspezifischer Profilbereich (Index 0x1000 - 0x1FFF).
   Enthält die Beschreibung aller spezifischen Parameter für die Kommunikation.

2. Herstellerspezifischer Profilbereich (Index 0x2000 - 0x5FFF).
   Enthält die Beschreibung herstellerspezifischen Einträge.
3. Standardisierter Geräteprofilbereich (0x6000 - 0x9FFF).
   Enthält die Objekte für das Geräteprofil nach DS-401.

Jeder Eintrag im Objektverzeichnis ist durch einen 16-Bit-Index gekennzeichnet. Falls ein Objekt aus mehreren Komponenten besteht (z.B. Objekttyp Array oder Record), sind die Komponenten über einen 8-Bit-Subindex gekennzeichnet. Der Objektname beschreibt die Funktion eines Objekts, das Datentyp-Attribut spezifiziert den Datentyp des Eintrags. Über das Zugriffsattribut ist spezifiziert, ob ein Eintrag nur gelesen werden kann, nur geschrieben werden oder gelesen und geschrieben werden darf.

Kommunikationsspezifischer Bereich

In diesem Bereich des Objektverzeichnisses stehen alle für die Kommunikation des CANopen-Buskopplers notwendigen Parameter und Objekte. Im Bereich 0x1000 - 0x1018 stehen verschiedene, allgemeine kommunikationsspezifische Parameter (z.B. der Gerätename).

Die Kommunikationsparameter (z.B. Identifier) der Receive-PDOs stehen im Bereich 0x1400 - 0x140F (plus Subindex). Die Mapping-Parameter der Receive-PDOs stehen im Bereich von 0x1600 - 0x160F (plus Subindex). Die Mappingparameter enthalten die Querverweise auf die Applikationsobjekte, die in die PDOs gemappt sind und die Datenbreite des entsprechenden Objektes (siehe auch Abschnitt PDO-Mapping).

Die Kommunikations- und Mapping-Parameter der Transmit-PDOs stehen in den Bereichen 0x1800 - 0x180F bzw. 0x1A00 - 0x1A0F.

Herstellerspezifischer Bereich

In diesem Bereich finden sich Einträge, die BECKHOFF spezifisch sind, z.B.:

- Datenobjekte für Sonderklemmen
- Objekte für die Register-Kommunikation, über die auf alle internen Register der Buskoppler und Busklemmen zugegriffen werden kann.
- Objekte für die vereinfachte Konfiguration der PDOs

Standardisierter Geräteprofilbereich

Im Standardisierten Geräteprofilbereich wird das CANopen-Geräteprofil DS-401 Version 1 unterstützt. Für Analogeingänge stehen dabei Funktionen zur Verfügung, um die Kommunikation in der ereignisgesteuerten Betriebsart an die Applikationsanforderungen anzupassen und die Buslast zu minimieren:

- Grenzwertüberwachung
- Deltafunktion
- Ereignissteuerung aktivieren / deaktivieren

4.7.5.2 Objektliste

Die Objekte aus dem Objektverzeichnis sind per SDO-Zugriff, jedoch nicht generell über die Konfigurations-Software KS2000removed link: KS2000 erreichbar. Dagegen sind alle Register, die per KS2000 konfiguriert werden können, auch per SDO-Zugriff auf das Objektverzeichnis (Objekte 0x4500 und 0x4501) erreichbar - wenn auch nicht mit dem gleichen Bedienungskomfort wie mit der Konfigurations-Software KS2000.

Parameter	Index	BK5120/ BK515x	BK5110	LC5100	BX5100/ BC5150	CX705x/ CX8051/B510
Gerätetyp [▶ 42]	0x1000	x	x	x		x
Fehlerregister [▶ 42]	0x1001	x	x	x	x	x *
Fehlerspeicher [▶ 42]	0x1003	x	x	x		
Sync Identifier [▶ 42]	0x1005	x	x	x	x	x
Sync Intervall [▶ 42]	0x1006	x	x	x	x	x

Parameter	Index	BK5120/ BK515x	BK5110	LC5100	BX5100/ BC5150	CX705x/ CX8051/B510
Gerätename [▶ 42]	0x1008	x	x	x	x	x *
Hardware-Version [▶ 42]	0x1009	x	x	x		
Software-Version [▶ 42]	0x100A	x	x	x	x	x
Knotennummer [▶ 42]	0x100B	x	x	x		
Guard Time [▶ 42]	0x100C	x	x	x	x	x
Life Time Factor [▶ 42]	0x100D	x	x	x	x	x
Guarding Identifier [▶ 42]	0x100E	x	x	x		
Parameter speichern [▶ 42]	0x1010	x	x	x		
Default-Werte laden [▶ 42]	0x1011	x	x	x		
Emergency Identifier [▶ 42]	0x1014	x	x	x		
Consumer Heartbeat Time [▶ 42]	0x1016	x	x	x	x	x
Producer Heartbeat Time [▶ 42]	0x1017	x	x	x	x	x
Gerätekennung (Identity Object) [▶ 42]	0x1018	x	x	x	x	x *
Server SDO Parameter [▶ 42]	0x1200	x	x	x		
Kommunikationsparameter 1.-5. RxPDO [▶ 42]	0x1400 - 0x1404	x	x	x	x	x
Kommunikationsparameter 6.-16. RxPDO [▶ 42]	0x1405 - 0x140F	x			x	x
Kommunikationsparameter 17.-32. RxPDO [▶ 42]	0x1410 - 0x141F				x nur BX5100	x
Mapping 1.-5. RxPDO [▶ 42]	0x1600 - 0x1604	x	x	x	x	x
Mapping 6. -16. RxPDO [▶ 42]	0x1605 - 0x160F	x			x	x
Mapping 17. -32. RxPDO [▶ 42]	0x1610 - 0x161F				x nur BX5100	x
Kommunikationsparameter 1.-5. TxPDO [▶ 42]	0x1800 - 0x1804	x	x	x	x	x
Kommunikationsparameter 6.-16. TxPDO [▶ 42]	0x1805 - 0x180F	x			x	x
Kommunikationsparameter 17.-32. TxPDO [▶ 42]	0x1810 - 0x181F				x nur BX5100	x
Mapping 1.-5. TxPDO [▶ 42]	0x1A00 - 0x1A04	x	x	x	x	x
Mapping 6. -16. TxPDO [▶ 42]	0x1A05 - 0x1A0F	x			x	x
Mapping 17. -32. TxPDO [▶ 42]	0x1A10 - 0x1A1F				x nur BX5100	x
Merkerbereich %MB0-511	0x2F00				x	
Merkerbereich %MB511-1023	0x2F01				x	
Merkerbereich %MB1024-1535	0x2F02				x	
Merkerbereich %MB1536-2047	0x2F03				x	
Merkerbereich %MB2048-2559	0x2F04				x	
Merkerbereich %MB2560-3071	0x2F05				x	
Merkerbereich %MB3072-3584	0x2F06				x	
Merkerbereich %MB3585-4095	0x2F07				x	
3-Byte Sonderklemmen, Eingangsdaten [▶ 42]	0x2600	x				

Parameter	Index	BK5120/ BK515x	BK5110	LC5100	BX5100/ BC5150	CX705x/ CX8051/B510
3-Byte Sonderklemmen, Ausgangsdaten [▶ 42]	0x2700	x				
4-Byte Sonderklemmen, Eingangsdaten [▶ 42]	0x2800	x				
4-Byte Sonderklemmen, Ausgangsdaten [▶ 42]	0x2900	x				
5-Byte Sonderklemmen, Eingangsdaten [▶ 42]	0x2A00	x				
5-Byte Sonderklemmen, Ausgangsdaten [▶ 42]	0x2B00	x				
6-Byte Sonderklemmen, Eingangsdaten [▶ 42]	0x2C00	x				
6-Byte Sonderklemmen, Ausgangsdaten [▶ 42]	0x2D00	x				
8-Byte Sonderklemmen, Eingangsdaten [▶ 42]	0x3000	x				
8-Byte Sonderklemmen, Ausgangsdaten [▶ 42]	0x3100	x				
Register-Kommunikation, Busknoten [▶ 42]	0x4500	x	x	x		
Register-Kommunikation, Busklemme/Erweiterungsbox [▶ 42]	0x4501	x	x	x		
PDOs aktivieren [▶ 42]	0x5500	x	x	x		
NetId	0x5FFE				x	
Digitale Eingänge [▶ 42]	0x6000	x	x	x		
Interrupt-Maske [▶ 42]	0x6126	x	x	x		
Digitale Ausgänge [▶ 42]	0x6200	x	x	x		
Analoge Eingänge [▶ 42]	0x6401	x				
Analoge Ausgänge [▶ 42]	0x6411	x				
Ereignissteuerung, analoge Eingänge [▶ 42]	0x6423	x				
Oberer Grenzwert, analoge Eingänge [▶ 42]	0x6424	x				
Unterer Grenzwert, analoge Eingänge [▶ 42]	0x6425	x				
Deltafunktion, analoge Eingänge [▶ 42]	0x6426	x				

* Wird ein ADS Server angemeldet, werden diese Objekte per ADS Notification an die SPS weitergeleitet und müssen dort beantwortet werden.

4.7.5.3 Objekte und Daten

Gerätetyp

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1000	0	Device Type	Unsigned32	ro	N	0x00000000	Angabe des Gerätetyps

Der 32Bit-Wert ist in zwei 16Bit-Felder unterteilt:

MSB	LSB
Additional Information	Geräteprofil-Nummer
0000 0000 0000 wxyz	0x191 (401_{dez})

Die *Additional Information* enthält Angaben über die Signalarten des E/A-Gerätes:
z=1 bedeutet digitale Eingänge,
y=1 bedeutet digitale Ausgänge,
x=1 bedeutet analoge Eingänge,
w=1 bedeutet analoge Ausgänge.
Ein BK5120 mit digitalen und analogen Eingängen, aber ohne Ausgänge, liefert also 0x00 05 01 91 zurück.

Sonderklemmen (z.B. serielle Schnittstellen, PWM-Ausgänge, Inkrementalencoder-Eingänge) werden nicht berücksichtigt. Ein Koppler, der z.B. nur serielle Schnittstellenklemmen KL6001 bestückt hat, liefert also 0x00 00 01 91 zurück.

Der Gerätetyp liefert nur eine grobe Klassifizierung des Gerätes. Für die detaillierte Identifizierung des Buskopplers und der angesteckten Klemmen kann das Klemmenbezeichnungs-Register des Buskopplers gelesen werden (Details siehe Register-Kommunikation Index 0x4500).

Fehlerregister

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1001	0	Error Register	Unsigned8	ro	N	0x00	Fehlerregister

Der 8Bit-Wert ist wie folgt kodiert:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ManSpec.	reserviert	reserviert	Comm.	reserviert	reserviert	reserviert	Generic

ManSpec. Herstellerspezifischer Fehler, wird in Objekt 1003 genauer spezifiziert.

Comm. Kommunikationsfehler (Overrun CAN)

Generic Ein nicht näher spezifizierter Fehler ist aufgetreten (Flag ist bei jeder Fehlermeldung gesetzt)

Fehlerspeicher

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1003	0x00	Predefined error field (Fehlerspeicher)	Unsigned8	rw	N	0x00	Objekt 1003h enthält eine Beschreibung der im Gerät aufgetretenen Fehler - Subindex 0 die Anzahl der gespeicherten Fehlerzustände.
	1	Actual error	Unsigned32	ro	N	Keiner	Letzter aufgetretener Fehlerzustand
	...	...	...	--	...	...	...
	10	Standard error field	Unsigned32	ro	N	Keiner	Es werden maximal 10 Fehlerzustände gespeichert.

Der 32Bit-Wert im Fehlerspeicher ist in zwei 16Bit-Felder unterteilt:

MSB	LSB
Additional Code	Error Code

Der Additional Code enthält den Error Trigger (siehe Emergency-Objekt) und damit eine detaillierte Fehlerbeschreibung.

Neue Fehler werden jeweils an Subindex 1 gespeichert, alle anderen Sub-indices werden entsprechend inkrementiert. Durch Schreiben einer 0 auf Subindex 0 wird der gesamte Fehlerspeicher gelöscht.

Wenn kein Fehler seit dem Power-On aufgetreten ist, dann besteht Objekt 0x1003 nur aus Subindex 0 mit eingetragener 0. Durch einen Reset oder Power Cycle wird der Fehlerspeicher gelöscht.

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

Sync Identifier

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1005	0	COB-ID Sync Message	Unsigned32	rw	N	0x80000080	Identifier der SYNC-Nachricht

Die unteren 11 Bit des 32-Bit Wertes enthalten den Identifier (0x80=128dez). Bit 30 gibt Auskunft, ob das Gerät das SYNC-Telegramm sendet (1) oder nicht (0). Die CANopen E/A Geräte empfangen das SYNC Telegramm, dementsprechend ist Bit 30=0. Bit 31 ist aus Gründen der Abwärtskompatibilität ohne Bedeutung.

Sync Intervall

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1006	0	Communication cycle period	Unsigned32	rw	N	0x00000000	Länge des SYNC-Intervalls in µs.

Wenn hier ein Wert ungleich Null eingetragen wird, so geht der Busknoten in den Fehlerzustand, wenn beim synchronen PDO-Betrieb innerhalb der Watchdog-Zeit kein SYNC-Telegramm empfangen wurde. Die Watchdog- Zeit entspricht hierbei dem 1,5-fachen der eingestellten communication cycle period - es kann also der vorgesehene SYNC-Abstand eingetragen werden.

Das E/A Update wird bei den Beckhoff CANopen Busknoten direkt nach Empfang des SYNC Telegramms durchgeführt, wenn folgende Voraussetzungen gegeben sind:
- Firmwarestand ab C0 (ab CANopen Version 4.01).
- alle PDOs, die über Daten verfügen, auf die synchrone Kommunikationsart eingestellt (0..240).
- Sync Intervall in Objekt 0x1006 eingetragen und (Sync Intervall x kleinste PDO Übertragungsart) kleiner als 90ms.

Die Baugruppen sind dann durchsynchronisiert.

Gerätename

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1008	0	Manufacturer Device Name	Visible String	ro	N	BK51x0, LC5100, IPxxxx-B510 od. ILxxxx-B510	Gerätename des Busknotens

Da der zurück gelieferte Wert größer als 4 Bytes ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

Hardware-Version

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1009	0	Manufacturer Hardware-Version	Visible String	ro	N	-	Hardwareversionsnummer des Busknotens

Da der zurück gelieferte Wert größer als 4 Bytes ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

Software-Version

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100A	0	Manufacturer Software-Version	Visible String	ro	N	-	Softwareversionsnummer des Busknotens

Da der zurück gelieferte Wert größer als 4 Bytes ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

Knotennummer

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100B	0	Node-ID	Unsigned32	ro	N	keiner	eingestellte Knotennummer

Die Knotennummer wird aus Kompatibilitätsgründen unterstützt.

Guard Time

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100C	0	Guard Time [ms]	Unsigned16	rw	N	0	Abstand zwischen zwei Guard Telegrammen. wird durch NMT-Master oder Konfigurationstool eingestellt.

Life Time Factor

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100D	0	Life Time Factor	Unsigned8	rw	N	0	Life Time Factor x Guard Time = Life Time (Watchdog für Life Guarding)

Wenn innerhalb der Life Time kein Guarding-Telegramm empfangen wurde, geht der Knoten in den Fehlerzustand. Wenn Life Time Factor und/oder Guard Time = 0 sind, so führt der Knoten kein Lifeguarding durch, kann aber dennoch vom Master überwacht werden (Node Guarding).

Guarding Identifier

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x100 E	0	COB-ID guarding protocol	Unsigned32	ro	N	0x000007xy, xy = NodeID	Identifier des Guarding Protokolls

Der Guarding Identifier wird aus Kompatibilitätsgründen unterstützt. Seit CANopen Version 4 darf der Guarding Identifier nicht mehr verändert werden.

Parameter speichern

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1010	0	Store Parameter	Unsigned8	ro	N	1	Anzahl der Speicheroptionen
	1	store all parameters	Unsigned32	rw	N	1	Speichert alle (speicherbaren) Parameter

Durch Schreiben der Signatur *save* im ASCII-Code (hexadezimal 0x65766173) auf Subindex 1 werden die aktuellen Parameter nichtflüchtig gespeichert. (Bytefolge auf dem Bus incl. SDO Protokoll: 0x23 0x10 0x10 0x01 0x73 0x61 0x76 0x65).

Der Speichervorgang dauert ca. 3 Sec., bei Erfolg wird anschließend durch das entsprechende TxSDO (0x60 im ersten Byte) bestätigt. Da der Buskoppler während des Speichervorgangs keine CAN-Telegramme senden und empfangen kann, kann nur gespeichert werden, wenn der Knoten im Zustand Pre-Operational ist. Es wird empfohlen, vor dem Abspeichern das gesamte Netz in den Zustand Pre-Operational zu versetzen. Dadurch wird ein Puffer-Überlauf vermieden.

Gespeichert werden:

- Die aktuelle Klemmenbestückung (Anzahl jeder Klemmenkategorie)
- Alle PDO Parameter (Identifier, Transmission Type, Inhibit Zeit, Mapping).

[Gefahrinformation hier einfügen!]

HinweisAnschließend gelten die gespeicherten Identifier, nicht mehr die aus der Knotenadresse abgeleiteten Default-Identifier. Änderungen der DIP-Schalter-Stellung beeinflussen die PDOs dann nicht mehr!

- Alle SYNC Parameter
- Alle Guarding Parameter
- Grenzwerte, Deltawerte und Interrupt Enable für Analogeingänge

Die in den Klemmen über Register-Kommunikation direkt gespeicherten Parameter werden dort sofort nichtflüchtig gespeichert.

Default-Werte laden

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1011	0	Restore Parameter	Unsigned8	ro	N	4	Anzahl der Rücksetze-Optionen
	1	Restore all parameters	Unsigned32	rw	N	1	Setzt alle Parameter auf Default-Werte zurück
	4	Set manufacturer Defaults	Unsigned32	rw	N	1	Setzt alle Koppler-Parameter auf Hersteller-Einstellungen zurück (auch Register)

Durch Schreiben der Signatur *load* im ASCII-Code (hexadezimal 0x64616F6C) auf Subindex 1 werden alle Parameter **beim nächsten Booten (Reset)** auf Default-Werte (Auslieferungszustand) zurückgesetzt.

(Bytefolge auf dem Bus incl. SDO Protokoll: 0x23 0x11 0x10 0x01 0x6C 0x6F 0x61 0x64).

Hierdurch werden die Default-Identifier für die PDOs wieder aktiv.

Emergency Identifier

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1014	0	COB-ID Emergency	Unsigned32	rw	N	0x00000080, + NodeID	Identifier des Emergency-Telegramms

Die unteren 11 Bit des 32-Bit Wertes enthalten den Identifier (0x80=128dez). Über das MSBit lässt sich einstellen ob das Gerät das Emergency-Telegramm sendet (1) oder nicht (0).

Alternativ lässt sich die Diagnose-Funktion der Busknoten auch durch das Bit *Gerätediagnose* in der K-Buskonfiguration (siehe Objekt 0x4500) abschalten.

Consumer Heartbeat Time

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1016	0	Anzahl Elemente	Unsigned8	ro	N	2	Die Consumer Heartbeat Time beschreibt die erwartete Heartbeat-Zykluszeit sowie die Node-ID des überwachten Knotens
	1	Consumer Heartbeat Time	Unsigned32	rw	N	0	Watchdog Zeit in ms und Node-ID des überwachten Knotens

Der 32Bit-Wert wird wie folgt verwendet:

MSB		LSB
Bit 31...24	Bit 23...16	Bit 15...0
reserviert (0)	Node-ID (Unsigned8)	heartbeat time in ms (Unsigned16)

Aus der Node-ID ergibt sich der überwachte Identifier durch die Default-Identifier-Verteilung: Guard-ID = 0x700 + Node-ID.

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

Producer Heartbeat Time

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1017	0	Producer Heartbeat Time	Unsigned16	rw	N	0	Zeitspanne in ms zwischen zwei gesendeten Heartbeat-Telegrammen

Gerätekennung (Identity Object)

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1018	0	Identity Object: Anzahl Elemente	Unsigned8	ro	N	4	Das Identity Objekt enthält allgemeine Angaben zu Art und Ausgabestand des Gerätes.
	1	Vendor ID	Unsigned32	ro	N	0x00000002	Herstellerkennung. Beckhoff hat die Vendor-ID 2
	2	Product Code	Unsigned32	ro	N	abhängig vom Produkt	Gerätekennung
	3	Revision Number	Unsigned32	ro	N	-	Versionsnummer
	4	Serial Number	Unsigned32	ro	N	-	Produktionsdatum Low-Wort, High-Byte: Kalenderwoche (dez), Low-Wort, Low-Byte: Kalenderjahr

Produkt	Product Code
BK5120	0x11400
BK5110	0x113F6
LC5100	0x113EC
IPwxyz-B510	0x2wxyz
IL2301-B510	0x2008FD

Server SDO Parameter

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1200	0	Anzahl Elemente	Unsigned8	ro	N	2	Kommunikationsparameter des Server SDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID Client ->Server	Unsigned32	ro	N	0x000006xy, xy=Node-ID	COB-ID RxSDO (Client -> Server)
	2	COB-ID Server ->Client	Unsigned32	ro	N	0x00000580 + Node-ID	COB-ID TxSDO (Client -> Server)

Aus Gründen der Abwärtskompatibilität im Objektverzeichnis enthalten.

Kommunikationsparameter 1. RxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1400	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des ersten Receive-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x000002xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

Subindex 1 (COB-ID): Die unteren 11 Bit des 32-Bit Wertes (Bits 0-10) enthalten den CAN-Identifier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktuell existiert (0) oder nicht (1), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1). Es ist nicht erlaubt, den Identifier (Bit 0-10) zu ändern, während das Objekt existiert (Bit 31=0). Der Subindex 2 enthält die Übertragungsart (siehe Einführung PDOs).

Kommunikationsparameter 2. RxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1401	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des zweiten Receive-PDOs.
	1	COB-ID	Unsigned32	rw	N	0x000003xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO2
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

Kommunikationsparameter 3. RxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1402	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des dritten Receive-PDOs.
	1	COB-ID	Unsigned32	rw	N	0x000004xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO3
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

Kommunikationsparameter 4. RxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1403	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des vierten Receive-PDOs.
	1	COB-ID	Unsigned32	rw	N	0x000005xy, xy=Node-ID	COB-ID (Communication Object Identifier) RxPDO4
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

Kommunikationsparameter 5.-16. RxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1404 - 0x140F (je nach Geräte Typ)	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des 5. bis 16. Receive-PDOs.
	1	COB-ID	Unsigned32	rw	N	0x8000000	COB-ID (Communication Object Identifier) RxPDO5...16
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Aus Gründen der Abwärtskompatibilität vorhanden, im RxPDO nicht genutzt.
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer. Definiert Watchdog Zeit für Empfangsüberwachung des PDOs.

Die Anzahl der RxPDOs je Busknoten-Typ kann den technischen Daten entnommen werden.

Mapping-Parameter 1. RxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1600	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des ersten Receive-PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x62000108	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x62000208	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x62000808	8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das erste Empfangs-PDO (RxPDO1) ist per Default für digitale Ausgangsdaten vorgesehen. Je nach Anzahl der bestückten Ausgänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die digitalen Ausgänge byteweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

Mapping-Änderungen

Um das Mapping zu verändern muss folgende Reihenfolge eingehalten werden (ab CANopen Version 4 vorgeschrieben):

1. PDO löschen (Bit 31 im Identifier-Eintrag (Subindex1) des Kommunikations-Parameters auf 1 setzen)
2. Mapping deaktivieren (Subindex 0 des Mapping Eintrages auf 0 setzen)
3. Mapping Einträge ändern (Subindices 1...8)
4. Mapping aktivieren (Subindex 0 des Mapping Eintrages auf die korrekte Anzahl der gemappten Objekte setzen)
5. PDO anlegen (Bit 31 im Identifier-Eintrag (Subindex 1) des Kommunikations-Parameters auf 0 setzen)

Mapping-Parameter 2. RxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1601	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des zweiten Receive-PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x64110110	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x64110210	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x00000000	8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das zweite Empfangs-PDO (RxPDO2) ist per Default für analoge Ausgänge vorgesehen. Je nach Anzahl der bestückten Ausgänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die analogen Ausgänge wortweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

Mapping-Parameter 3.-16. RxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1602-0x160F (je nach Geräte Typ)	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des 3.-16. Receive-PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	1. gemapptes Applikation sobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	2. gemapptes Applikation sobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x00000000(Siehe Text)	8. gemapptes Applikation sobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das 3. bis 16. Empfangs-PDO (RxPDO3ff) wird vom Busknoten je nach Klemmen-Bestückung (bzw. je nach Erweiterungs-Modulen) automatisch mit einem Default Mapping versehen. Die Vorgehensweise ist im Kapitel PDO-Mapping beschrieben.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

[Gefahrinformation hier einfügen!]

HinweisDS401 V2 schreibt für die PDOs 3+4 als Default Mapping analoge Ein- bzw. Ausgangsdaten vor. Das entspricht dem Beckhoff Default Mapping dann, wenn weniger als 65 digitale Ein- bzw. Ausgänge vorhanden sind. Um die Abwärtskompatibilität zu gewährleisten wird das Beckhoff Default Mapping beibehalten - die Geräte entsprechen damit in ihrem Mapping-Verhalten DS401 V1, in allen anderen Belangen DS401 V2.

Kommunikationsparameter 1. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1800	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des ersten Sende-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x00000180 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

Subindex 1 (COB-ID): Die unteren 11 Bit des 32-Bit Wertes (Bits 0-10) enthalten den CAN-Identifier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktuell existiert (0) oder nicht (1), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1). Es ist nicht erlaubt, den Identifier (Bit 0-10) zu ändern, während das Objekt existiert (Bit 31=0). Der Subindex 2 enthält die Übertragungsart, Subindex 3 die Wiederholungsverzögerung zwischen zwei gleichen PDOs, Subindex 5 enthält den Event Timer. Subindex 4 ist aus Kompatibilitätsgründen vorhanden, wird aber nicht genutzt. (siehe auch Einführung PDOs).

Kommunikationsparameter 2. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1801	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des zweiten Sende-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x00000280 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

Das zweite Sende-PDO ist per Default für analoge Eingänge vorgesehen und für ereignisgesteuerte Übertragung konfiguriert (Transmission Type 255). Die Ereignissteuerung muss zunächst aktiviert werden (siehe Objekt 0x6423), ansonsten können die Eingänge nur per Remote Transmission Request (RTR) abgefragt (gepollt) werden.

Kommunikationsparameter 3. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1802	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des dritten Sende-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x00000380 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

Das dritte Sende-PDO wird in der Regel analoge Eingangsdaten enthalten (siehe Mapping). Es ist für ereignisgesteuerte Übertragung konfiguriert (Transmission Type 255). Die Ereignissteuerung muss zunächst aktiviert werden (siehe Objekt 0x6423), ansonsten können die Eingänge nur per Remote Transmission Request (RTR) abgefragt (gepollt) werden.

Kommunikationsparameter 4. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1803	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des vierten Sende-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x00000480 + Node-ID	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

Das vierte Sende-PDO wird in der Regel analoge Eingangsdaten enthalten (siehe Mapping). Es ist für ereignisgesteuerte Übertragung konfiguriert (Transmission Type 255). Die Ereignissteuerung muss zunächst aktiviert werden (siehe Objekt 0x6423), ansonsten können die Eingänge nur per Remote Transmission Request (RTR) abgefragt (gepollt) werden.

Kommunikationsparameter 5.-16. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1804-0x180F (je nach Geräte-typ)	0	Anzahl Elemente	Unsigned8	ro	N	5	Kommunikationsparameter des 5.-16. Sende-PDOs. Subindex 0: Anzahl der folgenden Parameter
	1	COB-ID	Unsigned32	rw	N	0x0000000	COB-ID (Communication Object Identifier) TxPDO1
	2	Transmission Type	Unsigned8	rw	N	255	Übertragungsart des PDOs
	3	Inhibit Time	Unsigned16	rw	N	0	Wiederholungsverzögerung [Wert x 100 µs]
	4	CMS Priority Group	Unsigned8	rw	N	-	Aus Gründen der Abwärtskompatibilität vorhanden, nicht genutzt.
	5	Event Timer	Unsigned16	rw	N	0	Event-Timer

Mapping 1. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1A00	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des ersten Transmit PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x60000108	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x60000208	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x60000808	8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das erste Sende-PDO (TxPDO1) ist per Default für digitale Eingangsdaten vorgesehen. Je nach Anzahl der bestückten Eingänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die digitalen Eingänge byteweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

Mapping 2. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1A01	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des zweiten Transmit PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x64010110	1. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x64010210	2. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N		8. gemapptes Applikationsobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das zweite Sende-PDO (TxPDO2) ist per Default für analoge Eingangsdaten vorgesehen. Je nach Anzahl der bestückten Eingänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die analogen Eingänge wortweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

Mapping 3.-16. TxPDO

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x1A02-0x1A0F (je nach Geräte Typ)	0	Anzahl Elemente	Unsigned8	rw	N	abhängig von Typ und Bestückung	Mapping-Parameter des 3.-16. Transmit PDOs; Subindex 0: Anzahl der gemappten Objekte.
	1	1. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	1. gemapptes Applikation sobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	2	2. gemapptes Objekt	Unsigned32	rw	N	0x00000000 (Siehe Text)	2. gemapptes Applikation sobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)
	...	...	...	...	...	...	...
	8	8. gemapptes Objekt	Unsigned32	rw	N	0x00000000(Siehe Text)	8. gemapptes Applikation sobjekt (2 Byte Index, 1 Byte Subindex, 1 Byte Bitbreite)

Das 3. bis 16. Sende-PDO (TxPDO3ff) wird vom Busknoten je nach Klemmen-Bestückung (bzw. je nach Erweiterungs-Modulen) automatisch mit einem Default Mapping versehen. Die Vorgehensweise ist im Kapitel PDO-Mapping beschrieben.

Um das Mapping zu verändern muss eine bestimmte Reihenfolge eingehalten werden (siehe Objekt Index 0x1600).

[Gefahrinformation hier einfügen!]

HinweisDS401 V2 schreibt für die PDOs 3+4 als Default Mapping analoge Ein- bzw. Ausgangsdaten vor. Das entspricht dem Beckhoff Default Mapping dann, wenn weniger als 65 digitale Ein- bzw. Ausgänge vorhanden sind. Um die Abwärtskompatibilität zu gewährleisten wird das Beckhoff Default Mapping beibehalten - die Geräte entsprechen damit in ihrem Mapping-Verhalten DS401 V1, in allen anderen Belangen DS401 V2.

Im Objektverzeichnis (und damit auch im eds File) sind der Vollständigkeit halber zusätzlich folgende Objekteinträge vorhanden:

Index	Bedeutung
0x2000	Digitale Eingänge (Funktion identisch mit Objekt 0x6000)
0x2100	Digitale Ausgänge (Funktion identisch mit Objekt 0x6200)
0x2200	1-Byte Sonderklemmen, Eingänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2300	1-Byte Sonderklemmen, Ausgänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2400	2-Byte Sonderklemmen, Eingänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2500	2-Byte Sonderklemmen, Ausgänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2E00	7-Byte Sonderklemmen, Eingänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)
0x2F00	7-Byte Sonderklemmen, Ausgänge (derzeit keine entsprechenden Klemmen im Produktprogramm vorhanden)

3-Byte Sonderklemmen, Eingangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2600	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 3-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned24	ro	Y	0x000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0X80	128. input block	Unsigned24	ro	Y	0x000000	128. Eingangskanal

Beispiel für Sonderklemmen mit 3-Byte Eingangsdaten (in Default-Einstellung): KL2502 (PWM Ausgänge, 2 x 3 Bytes)

3-Byte Sonderklemmen, Ausgangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2700	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 3-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned24	rww	Y	0x000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0X80	128. output block	Unsigned24	rww	Y	0x000000	128. Ausgangskanal

Beispiel für Sonderklemmen mit 3-Byte Ausgangsdaten (in der Default-Einstellung): KL2502 (PWM Ausgänge, 2 x 3 Bytes)

4-Byte Sonderklemmen, Eingangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2800	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 4-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned32	ro	Y	0x00000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0X80	128. input block	Unsigned32	ro	Y	0x00000000	128. Eingangskanal

Beispiele für Sonderklemmen mit 4-Byte Eingangsdaten (in der Default-Einstellung): KL5001, KL6001, KL6021, KL6051

4-Byte Sonderklemmen, Ausgangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2900	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 4-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned32	rww	Y	0x00000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0X80	128. output block	Unsigned32	rww	Y	0x00000000	128. Ausgangskanal

Beispiele für Sonderklemmen mit 4-Byte Ausgangsdaten (in der Default-Einstellung): KL5001, KL6001, KL6021, KL6051

5-Byte Sonderklemmen, Eingangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2A00	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 5-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned40	ro	Y	0x0000000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0X40	64. input block	Unsigned40	ro	Y	0x0000000000	64. Eingangskanal

Beispiel für Sonderklemmen mit 5-Byte Eingangsdaten (in der Default-Einstellung): KL1501

5-Byte Sonderklemmen, Ausgangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2B00	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 5-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned40	rww	Y	0x0000000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0X40	64. output block	Unsigned40	rww	Y	0x0000000000	64. Ausgangskanal

Beispiel für Sonderklemmen mit 5-Byte Ausgangsdaten (in der Default-Einstellung): KL1501

6-Byte Sonderklemmen, Eingangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2C00	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 6-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned48	ro	Y	0x0000000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0X40	64. input block	Unsigned48	ro	Y	0x0000000000	64. Eingangskanal

Beispiel für Sonderklemmen mit 6-Byte Eingangsdaten (in der Default-Einstellung): KL5051, KL5101, KL5111

6-Byte Sonderklemmen, Ausgangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x2D00	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 6-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned48	rww	Y	0x0000000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0X40	64. output block	Unsigned48	rww	Y	0x0000000000	64. Ausgangskanal

Beispiel für Sonderklemmen mit 6-Byte Ausgangsdaten (in der Default-Einstellung): KL5051, KL5101, KL5111

8-Byte Sonderklemmen, Eingangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x3000	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 6-Byte Sonderkanäle, Eingänge
	1	1st input block	Unsigned64	ro	Y	0x0000000000	1. Eingangskanal
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned64	ro	Y	0x0000000000	64. Eingangskanal

Beispiel für Sonderklemmen mit 8-Byte Eingangsdaten: KL5101 (mit Word-Alignment, nicht in der Default-Einstellung)

8-Byte Sonderklemmen, Ausgangsdaten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x3100	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer 6-Byte Sonderkanäle, Ausgänge
	1	1st output block	Unsigned64	rww	Y	0x0000000000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0X40	64. output block	Unsigned64	rww	Y	0x0000000000	64. Ausgangskanal

Beispiel für Sonderklemmen mit 8-Byte Ausgangsdaten: KL5101 (mit Word-Alignment, nicht in der Default-Einstellung)

Register-Kommunikation Busknoten

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x4500	0	Register Access	Unsigned32	rw	N	keiner	Zugriff interne Register Busknoten

Der 32Bit-Wert ist wie folgt aufgebaut:

MSB			LSB
Zugriff (Bit7) + Tabellennummer (Bit 6...0)	Registernummer	High-Byte Registerwert	Low-Byte Registerwert
[0..1] + [0...0x7F]	[0...0xFF]	[0...0xFF]	[0...0xFF]

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

Durch Zugriff auf Index 0x4500 können beliebige Register der Busstation beschrieben oder gelesen werden. Die Kanalnummer und Register werden hierbei im 32Bit-Datenwert adressiert.

Registerwert lesen

Zunächst muss dem Koppler mitgeteilt werden, welches Register gelesen werden soll. Hierzu muss ein SDO-Schreibzugriff auf die entsprechende Index/Subindex-Kombination erfolgen mit:
- Tabellennummer (Zugriffs-Bit=0) in Byte 3
- Registeradresse in Byte 2 des 32 Bit Datenwertes.

Bytes 1 und 0 werden nicht ausgewertet, wenn das Zugriffs-Bit (MSB in Byte 3) = 0 ist. Anschließend kann der Registerwert auf derselben Index/Subindex-Kombination gelesen werden.

Der Koppler setzt das Zugriffs-Bit nach dem Schreiben der auszulesenden Registeradresse so lange auf 1, bis der korrekte Wert zur Verfügung steht. Beim SDO-Lesezugriff ist also zu überprüfen, dass die Tabellennummer im Wertebereich 0...0x7F liegt.

Ein Zugriffsfehler bei der Register-Kommunikation wird durch entsprechende Rückgabewerte des SDO-Protokolls angezeigt (siehe Kapitel SDO, Abbruch Parameterkommunikation).

Beispiel Registerwert lesen

Es soll festgestellt werden, welcher Baud-Ratenindex der Schalterstellung 1,1 (DIP 7,8) zugeordnet ist (siehe Kapitel *Netzwerkadresse und Baud-Raten*). Hierzu muss der Wert in Tabelle 100, Register 3 gelesen werden. Es müssen also folgende SDO Telegramme gesendet werden:

Schreibzugriff (Download Request) auf Index 4500, Subindex 0 mit 32 Bit Datenwert 0x64 03 00 00.

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 00 00 03 64

Anschließend Lesezugriff (Upload Request) auf den gleichen Index/Subindex, hierbei ist der Datenwert beliebig (hier 00).

Id=0x600+Node-ID DLC=8; Data=40 00 45 00 00 00 00 00

Der Koppler antwortet mit dem Upload Response Telegramm:

Id=0x580+Node-ID DLC=8; Data=43 00 45 00 04 00 03 64

Es steht hier also der Wert 4 in diesem Register, dieser Baud-Ratenindex entspricht 125 kBit/s (Default-Wert).

Registerwert Schreiben

SDO-Schreibzugriff auf die entsprechende Index/Subindex-Kombination mit:
- Tabellennummer + 0x80 (Zugriffs-Bit=1) in Byte 3
- Registeradresse in Byte 2
- High-Byte Registerwert in Byte 1
- Low-Byte Registerwert in Byte 0 des 32 Bit Datenwertes

Koppler-Schreibschutz aufheben

Bevor die Register des Buskopplers beschrieben werden können muss zunächst der Schreibschutz aufgehoben werden. Hierzu müssen die folgenden Werte in der angegebenen Reihenfolge auf die entsprechenden Register geschrieben werden:

Arbeitsschritt	Tabelle	Register	Wert	entsprechender SDO Download-Wert (0x4500/0)
1.	99	2	45054 (0xAFFE)	0xE3 02 AF FE (0xE3=0x63(=99)+ 0x80)
2.	99	1	1 (0x0001)	0xE3 01 00 01
3.	99	0	257 (0x0101)	0xE3 00 01 01

Koppler-Schreibschutz aufheben (CAN Darstellung)

Um den Koppler-Schreibschutz aufzuheben müssen also folgende SDO-Telegramme (Download Requests) an den Koppler geschickt werden:

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 FE AF 02 E3

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 01 00 01 E3

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 01 01 00 E3

Beispiel Registerwert Schreiben

Nachdem der Schreibschutz aufgehoben wurde, soll nun der Baud-Ratenindex für die DIP-Schalterstellung 1,1 auf den Wert 7 gesetzt werden. Damit wird dieser Schalterstellung die Baud-Rate 20 kBaud zugeordnet.

Hierzu muss Tabelle 100, Register 3 mit dem Wert 7 beschrieben werden, das erfolgt durch SDO-Schreibzugriff (Download Request) auf Index 0x4500, Subindex 0 mit dem 32 Bit-Wert E4 03 00 07 (0xE4 = 0x64+0x80):

Id=0x600+Node-ID DLC=8; Data=23 00 45 00 07 00 03 E4

Klemmen identifizieren

Über die Tabelle 9 des Buskopplers kann die Kennung des Kopplers (bzw. der Busstation) und der angesteckten Busklemmen gelesen werden. Dabei enthält Register 0 die Kennung des Buskopplers selbst, Register 1 die Kennung der ersten Klemme und Register n die Kennung der n-ten Klemme:

Tabellennummer	Registernummer	Beschreibung	Wertebereich
9	0	Busstation-Kennung	0 - 65535
9	1-255	Kennung Erweiterungsmodul/ Busklemme	0 - 65535

Die Buskopplerbeschreibung in Registernummer 0 enthält 5120 = 0x1400 beim BK5120, 5110 = 0x13F6 beim BK5110 und 5100 = 0x13EC beim LC5100. Bei den Feldbus Box Baugruppen steht in Register 0 die Kennung 510dez =0x1FE bzw. 518dez = 0x206.

Die Kennung der Erweiterungsmodule bzw. Klemmenbeschreibung enthält bei analogen und Sonderklemmen die Klemmenbezeichnung (Dez);
Beispiel: ist als dritte Klemme eine KL3042 gesteckt, so enthält Register 3 den Wert 3042_{dez} (0x0BE2).

Bei digitalen Klemmen wird folgende Bit-Kennung verwendet:

MSB								LSB							
1	s6	s5	s4	s3	s2	s1	s0	0	0	0	0	0	0	a	e

s6...s1: Datenbreite in Bit; a=1: Ausgangsklemme; e=1: Eingangsklemme

Diese Kennung führt zu den unten aufgeführten Klemmenbeschreibungen bei den Klemmen:

Kennung Klemmen	Bedeutung
0x8201	2 Bit digitale Eingangsklemme,z.B. KL1002, KL1052, Kl9110, KL9260
0x8202	2 Bit digitale Ausgangsklemme, z.B. KL2034, KL2612, KL2702
0x8401	4 Bit digitale Eingangsklemme, z.B. KL1104, KL1124, KL1194
0x8402	4 Bit digitale Ausgangsklemme,z.B. KL2124, KL2134, KL2184
0x8403	4 Bit digitale Ein/Ausgangsklemme,z.B. KL2212

und folgende Kennung bei den Erweiterungs Box Module:

Kennung Erweiterungs Box Module	Bedeutung
0x000A	4 Bit Eingangs- und 4 Bit Ausgangsmodul
0x0011	8 Bit Eingangs- und 8 Bit Ausgangsmodul
0x0014	8 Bit digitales Eingangsmodul
0x0015	8 Bit digitales Ausgangsmodul

Allgemeine Koppler-Konfiguration (Tabelle 0)

Die Tabelle 0 des Buskopplers enthält die Daten für die allgemeine Kopplerkonfiguration. In der Regel muss diese nicht verändert werden; für besondere Anwendungsfälle können die Einstellungen jedoch über die KS2000 Konfigurations-Software oder den direkten Zugriff über die Register-Kommunikation verändert werden. Hierzu muss zunächst der Schreibschutz aufgehoben werden (siehe oben).

Im Folgenden werden die relevanten Registereinträge beschrieben:

K-Buskonfiguration

Tabelle 0, Register 2 enthält die K-Buskonfiguration und ist wie folgt codiert (Default-Wert: 0x0006):

MSB								LSB							
0	0	0	0	0	0	0	0	0	0	0	0	0	D	G	A

A: Autoreset

Bei K-Bus-Fehler wird zyklisch versucht, den K-Bus durch Reset wieder zu aufzustarten. Wenn Emergencies und Guarding nicht ausgewertet werden, so kann es bei aktiviertem Autoreset vorkommen, dass Aus- und Eingangsinformation unerkannt verloren geht.

0: kein Autoreset (Default)

1: Autoreset aktiv

G: Gerätediagnose

Meldung (über Emergency), z.B. dass
- Drahtbruch bei Stromeingängen (mit Diagnose)
- 10 V überschritten bei 1-10V Eingangsklemme

0: Gerätediagnose abgeschaltet

1: Gerätediagnose aktiv (Default)

D: Diagnosedaten

digitaler Klemmen ins Prozessabbild einblenden (z.B. KL2212). Diese Flag wird nur ausgewertet, wenn die Gerätediagnose aktiv ist (siehe oben).

0: Nicht einblenden

1: Einblenden (Default)

Prozessabbildbeschreibung

Tabelle 0, Register 3 enthält die Prozessabbildbeschreibung und ist wie folgt codiert (Default-Wert: 0x0903):

MSB								LSB							
0	0	0	0	k1	k0	f1	f0	0	0	a	0	d	k	1	1

k0...k1: Reaktion auf K-Bus-Fehler

0,2: Eingänge bleiben unverändert (Default=2);

1: Eingänge auf 0 setzen (TxPDO mit Nullen wird verschickt)

f0...f1: Reaktion auf Feldbusfehler

0: Stoppen der K-Bus Zyklen, Watchdog auf Klemmen spricht an, Fehlerausgangswerte werden aktiv. Beim Neustart werden zunächst die alten Ausgangswerte gesetzt.

1: Ausgänge auf 0 setzen, Stoppen der K-Bus Zyklen (Default). 2: Ausgänge bleiben unverändert.

a: Word-Alignment von Analog- und Sonderklemmen

0: kein Alignment (Default)

1: Daten auf Wortgrenzen mappen (Prozessdatum beginnt stets auf gerader Adresse im PDO)

d: Datenformat komplexe Klemmen (Analog- und Sonderklemmen)

0: Intel-Format (Default)

1: Motorola-Format

k: Auswertung komplexe Klemmen (Analog- und Sonderklemmen)

0: nur Nutzdaten (Default)

1: komplette Auswertung (Achtung: Analogkanäle benötigen dann statt z.B. 2 Eingangsbytes je 3 Eingangs- und 3 Ausgangsbytes; statt 4 Kanäle je PDO werden für 2 Kanäle je ein Rx- und ein TxPDO benötigt)

Register-Kommunikation Busklemme/Erweiterungsbox

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x4501	0	Access Terminal Register	Unsigned8	ro	N	keiner	Index 0x4501 ermöglicht den Zugriff auf alle Register der Busklemmen bzw. Erweiterungsmodule. Subindex 0 enthält die Anzahl der gesteckten Busklemmen.
	1	Access Reg. Terminal 1	Unsigned32	rw	N	keiner	Zugriff Register Busklemme bzw. E-Modul 1
	...	...	...	...	...	...	...
	0XFE	Access Reg. Terminal 254	Unsigned32	rw	N	keiner	Zugriff Register Busklemme bzw. E-Modul 254

Der 32Bit-Wert ist wie folgt aufgebaut:

MSB			LSB
Zugriff (Bit7) + Kanalnummer (Bit 6...0)	Registernummer	High-Byte Registerwert	Low-Byte Registerwert
[0..1] + [0...0x7F]	[0...0xFF]	[0...0xFF]	[0...0xFF]

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

Durch Zugriff auf Index 0x4501 können die Anwenderregister der Busklemmen bzw. Erweiterungsmodule beschrieben oder gelesen werden. Die Baugruppen verfügen über einen Registersatz je Ein- bzw. Ausgangskanal. Die Adressierung der Baugruppen erfolgt über den Subindex, die Kanalnummer und Register werden im 32Bit-Datenwert adressiert. Hierbei entspricht die Kanalnummer 0 dem ersten Kanal, 1 dem zweiten Kanal etc.

Registerwert lesen

Zunächst muss dem Koppler mitgeteilt werden, welches Register gelesen werden soll. Hierzu muss ein SDO-Schreibzugriff auf die entsprechende Index/Subindex-Kombination erfolgen mit:
- Kanalnummer (Zugriffs-Bit=0) in Byte 3
- Registeradresse in Byte 2 des 32 Bit Datenwertes.

Bytes 1 und 0 werden nicht ausgewertet, wenn das Zugriffs-Bit (MSB in Byte 3) = 0 ist. Anschließend kann der Registerwert auf derselben Index/Subindex-Kombination gelesen werden.

Der Koppler setzt das Zugriffs-Bit nach dem Schreiben der auszulesenden Registeradresse so lange auf 1, bis der korrekte Wert zur Verfügung steht. Beim SDO-Lesezugriff ist also zu überprüfen, dass die Tabellennummer im Wertebereich 0...0x7F liegt.

Ein Zugriffsfehler bei der Register-Kommunikation wird durch entsprechende Rückgabewerte des SDO-Protokolls angezeigt (siehe Kapitel SDO, Abbruch Parameterkommunikation).

Beispiel Registerwert lesen

Bei einer Thermoelement-Eingangsklemme KL3202 soll festgestellt werden, auf welchen Thermoelement-Typ der zweite Eingangs-Kanal eingestellt ist. Hierzu muss das Feature-Register 32 gelesen werden. Die Klemme befindet sich am fünften Steckplatz neben dem Buskoppler. Es müssen also folgende SDO-Telegramme gesendet werden:

Schreibzugriff (Download Request) auf Index 4501, Subindex 5 mit 32 Bit Datenwert 01 20 00 00 (0x01 = 2. Kanal, 0x20 = Register 32)
Id=0x600+Node-ID DLC=8; Data=23 01 45 05 00 00 20 01

Anschließend Lesezugriff (Upload Request) auf den gleichen Index/Subindex, hierbei ist der Datenwert beliebig (hier: 0x00).
Id=0x600+Node-ID DLC=8; Data=40 01 45 05 00 00 00 00

Der Koppler antwortet mit dem Upload Response Telegramm:
Id=0x580+Node-ID DLC=8; Data=43 01 45 05 06 31 20 01

Es steht hier also der Wert 31 06 im Feature-Register. Die obersten 4 Bit kennzeichnen den Thermoelement-Typ. Sie sind hier 3, demnach ist der eingestellte Typ für diesen Kanal PT500 (siehe Dokumentation KL3202).

Registerwert Schreiben

SDO-Schreibzugriff auf die entsprechende Index/Subindex-Kombination mit:
- Kanalnummer + 0x80 (Zugriffs-Bit=1) in Byte 3
- Registeradresse in Byte 2
- High-Byte Registerwert in Byte 1
- Low-Byte Registerwert in Byte 0 des 32 Bit Datenwertes

HINWEIS

[Gefahrinformation hier einfügen!]

AchtungWenn der Schreibschutz nicht aufgehoben wurde (z.B. fehlerhaftes Codewort), so wird ein Schreibzugriff auf die Klemmenregister zwar bestätigt (SDO Download Response), der Wert jedoch nicht in das Register übernommen. Es wird deshalb empfohlen, den geschriebenen Wert anschließend auszulesen und zu vergleichen.

Klemmen-Schreibschutz aufheben

Bevor die Anwender-Register der Busklemmen (Register 32-xx, je nach Klemmentyp bzw. Erweiterungsmodul) beschrieben werden können muss zunächst der Schreibschutz aufgehoben werden. Hierzu wird das folgende Codewort in das Register 31 des entsprechenden Kanals geschrieben:

Schreibschutz	Kanal	Register	Wert	entsprechender SDO Download-Wert (0x4500/0)
	1,2, 3 oder 4	31 (0x1F)	4661 (0x1235)	8y 1F 12 35 (y=Kanalnummer)

Klemmen-Schreibschutz aufheben (CAN Darstellung)

Um den Klemmen-Schreibschutz aufzuheben muss also das folgende SDO-Telegramm an den Koppler geschickt werden:

Id=600 + Node-ID DLC=8; Data=23 01 45 xx 35 12 1F 8y

wobei xx den Steckplatz der Klemme und y den Kanal kennzeichnen.

Beispiel Schreibschutz aufheben

Steckt also beispielsweise an einem BK5120 mit der Knotenadresse 3 eine Thermoelement-Eingangsklemme KL3202 an Steckplatz 5, so ist der Schreibschutz für den ersten Kanal wie folgt aufzuheben:

Id=0x603 DLC=8; Data=23 01 45 05 35 12 1F 80

Für den zweiten Kanal ist folgendes Telegramm zu senden:

Id=0x603 DLC=8; Data=23 01 45 05 35 12 1F 81

Beispiel Registerwert Schreiben

Der Thermoelement-Typ des zweiten Kanals der KL3202 Klemme an Steckplatz 5 soll nun auf PT1000 umgestellt werden. Hierzu müssen die obersten 4 Bits (oberstes Nibble) im Feature-Register mit dem Wert 2 beschrieben werden. Es wird davon ausgegangen, dass für alle anderen Bits des Feature-Registers die Default-Werte übernommen werden sollen. Nachdem der Schreibschutz aufgehoben wurde, ist per SDO Schreibzugriff (Download Request) der folgende 32Bit-Wert auf Index 0x4501, Subindex 05 zu schreiben: 81 20 21 06 (0x81=01+0x80; 0x20=32;0x2106 = Registerwert).

Das entsprechende Telegramm sieht auf dem Bus wie folgt aus:

Id=0x600+Node-ID DLC=8; Data=23 01 45 05 06 21 20 81

PDOs aktivieren

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x5500	0	Activate PDO Defaults	Unsigned32	rw	N	0x00000000	setzt PDO Communication Parameter für PDO 2...11

CANopen definiert Default-Identifier für jeweils 4 Sende (Tx) und Empfangs (Rx) PDOs, alle anderen PDOs sind nach dem Aufstarten der Knoten zunächst deaktiviert. Über den Index 0x5500 lassen sich alle PDOs aktivieren, die gemäß Klemmenbestückung mit Prozessdaten vorbelegt sind (herstellerspezifisches Default Mapping). Dabei wird für PDO5...11 eine herstellerspezifische Default-Identifier-Verteilung vorgenommen sowie für PDO 2...11der Transmission Type und eine einheitliche Inhibit Zeit eingestellt. Nicht mit Prozessdaten versehene (also in der aktuellen Konfiguration überzählige) PDOs werden nicht aktiviert.

[Gefahrinformation hier einfügen!]

HinweisDieses Objekt kann nur im Pre-Operational Zustand beschrieben werden!

Der 32Bit-Wert wird wie folgt verwendet:

MSB			LSB
Transmission Type RxPDOs	Transmission Type TxPDOs	High-Byte Inhibit Zeit	Low-Byte Inhibit Zeit

Wie bei CANopen üblich wird das LSB zuerst und das MSB zuletzt übertragen.

Beispiel:

PDOs aktivieren für Busknoten Nummer 1, Inhibit Zeit auf 10ms (=100 x 100µs) setzen, Transmission Type TxPDOs auf 255 setzen, Transmission Type RxPDOs auf 1 setzen. Folgendes Telegramm ist zu senden:
Id=0x601 DLC=8; Data=23 00 55 00 64 00 FF 01

Der Knoten antwortet mit folgendem Telegramm:
Id=0x601 DLC=8; Data=60 00 55 00 00 00 00 00

Verwendete Identifier

Die Default-Identifier-Verteilung für die zusätzlichen PDOs läßt die vordefinierten Bereiche für Guarding, SDOs etc. frei, geht ab PDO6 von maximal 64 Knoten im Netz aus und erfolgt nach folgendem Schema:

Objekt	Function Code	resultierende COB-ID (hex)	resultierende COB-ID (dez)
TxPDO5	1101	0x681 - 0x6BF	1665 - 1727
RxPDO5	1111	0x781 - 0x7BF	1921 - 1983
TxPDO6	00111	0x1C1 - 0x1FF	449 - 511
RxPDO6	01001	0x241 - 0x27F	577 - 639
TxDPO7	01011	0x2C1 - 0x2FF	705 - 767
RxPDO7	01101	0x341 - 0x37F	833 - 895
TxPDO8	01111	0x3C1 - 0x3FF	961 - 1023
RxPDO8	10001	0x441 - 0x47F	1089 - 1151
TxPDO9	10011	0x4C1 - 0x4FF	1217 - 1279
RxPDO9	10101	0x541 - 0x57F	1345 - 1407
TxDPO10	10111	0x5C1 - 0x5FF	1473 - 1535
RxPDO10	11001	0x641 - 0x67F	1601 - 1663
TxPDO11	11011	0x6C1 - 0x6FF	1729 - 1791
RxPDO11	11101	0x741 - 0x77F	1857 - 1919

HINWEIS

[Gefahrinformation hier einfügen!]

AchtungEs ist darauf zu achten, dass der Index 0x5500 nicht genutzt wird, wenn Buskoppler mit mehr als 5 PDOs in Netzen mit Knoten-Adressen >64 vorhanden sind, da es sonst zu Identifier-Überschneidungen kommen kann. In diesem Fall müssen die PDO Identifier individuell eingestellt werden.

Der Übersichtlichkeit halber sind die nach CANopen definierten Default-Identifier hier ebenfalls aufgeführt:

Objekt	Function Code	resultierende COB-ID (hex)	resultierende COB-ID (dez)
Emergency	0001	0x81 - 0xBF [0xFF]	129 - 191 [255]
TxPDO1	0011	0x181 - 0x1BF [0x1FF]	385 - 447 [511]
RxPDO1	0100	0x201 - 0x23F [0x27F]	513 - 575 [639]
TxPDO2	0101	0x281 - 0x2BF [0x2FF]	641 - 676 [767]
RxPDO2	0110	0x301 - 0x33F [0x37F]	769 - 831 [895]
TxDPO3	0111	0x381 - 0x3BF [0x3FF]	897 - 959 [1023]
RxPDO3	1000	0x401 - 0x43F [0x47F]	1025 - 1087 [1151]
TxPDO4	1001	0x481 - 0x4BF [0x4FF]	1153 - 1215 [1279]
RxPDO4	1010	0x501 - 0x53F [0x57F]	1281- 1343 [1407]
SDO (Tx)	1011	0x581 - 0x5BF [0x5FF]	1409 - 1471 [1535]
SDO (Rx)	1100	0x601 - 0x63F [0x67F]	1537 - 1599 [1663]
Guarding / Heartbeat/ Bootup	1110	0x701 - 0x73F [0x77F]	1793 - 1855 [1919]

Angegeben sind die Identifier, die sich aus den DIP-Schalter-Einstellungen am Koppler ergeben, sowie in eckigen Klammern der Identifier-Bereich für die Knotenadressen 64...127 (am Buskoppler BK5110, BK5120 und LC5100 nicht einstellbar). Bei den Feldbus Box-Modulen und dem Buskoppler BK515x lassen sich die Adressen 1...99 einstellen.

Eine tabellarische Übersicht über alle Identifier findet sich im Anhang.

Digitale Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6000	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer digitaler 8-Bit Eingangsdatenblöcke
	1	1st input block	Unsigned8	ro	Y	0x00	1. Eingangskanal
	...	...	...	...	...	...	...
	0XFE	254. input block	Unsigned8	ro	Y	0x00	254. Eingangskanal

Interrupt Maske

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6126	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ	Anzahl der 32-Bit Interrupt Masken = 2 x Anzahl TxDPOs
	1	IR-Mask0 TxPDO1	Unsigned32	rw	N	0xFFFFFFFF	IR-Maske Bytes 0...3 TxPDO1
	2	IR-Mask1 TxPDO1	Unsigned32	rw	N	0xFFFFFFFF	IR-Maske Bytes 4...7 TxPDO1
	3	IR-Mask0 TxPDO2	Unsigned32	rw	N	0xFFFFFFFF	IR-Maske Bytes 0...3 TxPDO2
	...	...	...	...	...	...	...
	0x20	IR-Mask1 TxPDO16	Unsigned32	rw	N	0xFFFFFFFF	IR-Maske Bytes 4...7 TxPDO16

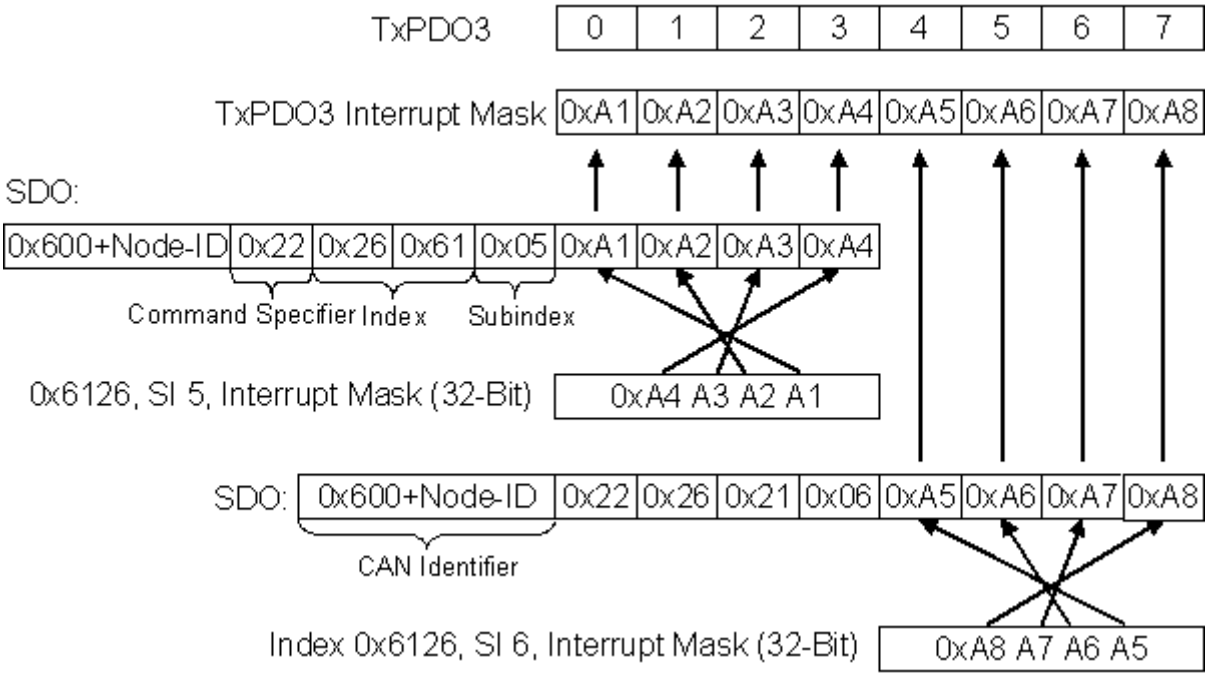
Per Default führt jede Änderung eines Wertes im ereignisgesteuerten PDO zum Versenden des Telegramms. Mit der Interrupt Maske kann bestimmt werden, welche Daten-Änderungen hierfür ausgewertet werden. Durch Nullen der entsprechenden Bereiche innerhalb der PDOs werden diese bei der Ereignissteuerung ("Interrupt-Steuerung") ausmaskiert. Die Interrupt Maske umfasst nicht nur die PDOs mit digitalen Eingängen, sondern alle vorhandenen TxPDOs. Falls die TxPDOs kürzer als 8 Bytes sind, wird der überzählige Teil der IR-Maske nicht ausgewertet.

Die Interrupt Maske beeinflusst nur TxPDOs mit Transmission Type 254 und 255. Sie wird nicht auf dem Gerät gespeichert (auch nicht durch das Objekt 0x1010). Änderungen der Maske zur Laufzeit (im Operational Status) sind möglich und werden bei der nächsten Eingangsdaten-Änderung ausgewertet.

Die Interrupt Maske wird für TxPDOs mit analogen Eingangsdaten nicht ausgewertet, wenn für die Eingänge Grenzwerte (0x6424, 0x6425) oder die Delta Funktion (0x6426) aktiviert wurden.

Dieser Eintrag ist ab Firmware Stand C3 implementiert.

Beispiel zur Zuordnung der Daten



Anwendungsbeispiel

Der Zählerwert eines schnellen Zählereingangs soll nur übertragen werden, sobald sich Bits im Statuswort (z.B. der Latch-Eingang) geändert haben. Hierzu muss der 32-Bit Zählerwert in der Interrupt Maske ausmaskiert (=genullt) werden. Der Status befindet sich im Byte 0, der Zählerwert liegt per Default in den Bytes 1..4 des entsprechenden PDOs (im Beispiel TxPDO3, da <65 digitale und <5 analoge Eingänge vorhanden sind).
Also muss in Index 0x6126, Subindex5 der Wert 0x0000 00FF und in Subindex6 der Wert 0xFFFF FF00 eingetragen werden.

Die entsprechenden SDOs sehen demnach wie folgt aus:

11-bit Identifier	8 Byte Nutzdaten							
0x600+ Node-ID	0x22	0x26	0x61	0x05	0xFF	0x00	0x00	0x00

11-bit Identifier	8 Byte Nutzdaten							
0x600+ Node-ID	0x22	0x26	0x61	0x06	0x00	0xFF	0xFF	0xFF

Digitale Ausgänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6200	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer digitaler 8-Bit Ausgangsdatenblöcke
	1	1st input block	Unsigned8	rw	Y	0x00	1. Ausgangskanal
	...	...	...	...	...	...	...
	0XFE	254. input block	Unsigned8	rw	Y	0x00	254. Ausgangskanal

Analoge Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6401	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Eingangs-Kanäle
	1	1st input	Unsigned16	ro	Y	0x0000	1. Eingangskanal
	...	...	...	...	...	...	...
	0XFE	254. input	Unsigned16	ro	Y	0x0000	254. Eingangskanal

Die analogen Signale werden linksbündig dargestellt. Damit wird die Darstellung im Prozessabbild unabhängig von der tatsächlichen Auflösung. Details zum Datenformat finden sich beim jeweiligen Signaltyp.

Analoge Ausgänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6411	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Ausgangs-Kanäle
	1	1st input block	Unsigned16	rw	Y	0x0000	1. Ausgangskanal
	...	...	...	...	...	...	...
	0XFE	254. input block	Unsigned16	rw	Y	0x0000	254. Ausgangskanal

Die analogen Signale werden linksbündig dargestellt. Damit wird die Darstellung im Prozessabbild unabhängig von der tatsächlichen Auflösung. Details zum Datenformat finden sich beim jeweiligen Signaltyp.

Ereignissteuerung Analoge Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6423	0	Global Interrupt Enable	Boolean	rw	N	FALSE (0)	Aktiviert das ereignisgesteuerte Senden von PDOs mit Analogeingängen.

Nach CANopen sind die Analogeingänge in TxPDO2..4 zwar per Default auf den Transmission Type 255 (ereignisgesteuert) eingestellt, jedoch ist das Ereignis (die Änderung eines Eingangswertes) über die Ereignissteuerung im Objekt 0x6423 deaktiviert, um ein Überfluten des Busses mit Analogsignalen zu verhindern. Es empfiehlt sich, das Datenaufkommen der Analog-PDOs entweder durch synchrone Kommunikation oder durch Verwendung des Event Timers zu kontrollieren. Im ereignisgesteuerten Betrieb kann das Sendeverhalten der Analog-PDOs vor dem Aktivieren durch Einstellen von Inhibit-Zeit (Objekt 0x1800ff, Subindex 3) und/oder Grenzwertüberwachung (Objekt 0x6424 + 0x6425) und/oder Deltafunktion (Objekt 0x6426) parametriert werden.

Oberer Grenzwert Analoge Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6424	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Eingangs-Kanäle
	1	upper limit 1st input	Unsigned16	rw	Y	0x0000	Oberer Grenzwert 1. Eingangskanal
	...	...	...	...	...	...	...
	0XFE	upper limit 254. input	Unsigned16	rw	Y	0x0000	Oberer Grenzwert 254. Eingangskanal

Werte ungleich 0 aktivieren den oberen Grenzwert für diesen Kanal. Ein PDO wird dann abgesetzt wenn dieser Grenzwert überschritten wird. Zusätzlich muss die Ereignissteuerung aktiviert sein (Objekt 0x6423). Das Datenformat entspricht dem der Analogeingänge.

Unterer Grenzwert Analoge Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6425	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Eingangs-Kanäle
	1	lower limit 1st input	Unsigned16	rw	Y	0x0000	Unterer Grenzwert 1. Eingangskanal
	...	...	...	...	...	...	...
	0XFE	lower limit 254. input	Unsigned16	rw	Y	0x0000	Unterer Grenzwert 254. Eingangskanal

Werte ungleich 0 aktivieren den unteren Grenzwert für diesen Kanal. Ein PDO wird dann abgesetzt wenn dieser Grenzwert unterschritten wird. Zusätzlich muss die Ereignissteuerung aktiviert sein (Objekt 0x6423). Das Datenformat entspricht dem der Analogeingänge.

Deltafunktion Analoge Eingänge

Index	Subindex	Name	Typ	Attrb.	Map.	Default-Wert	Bedeutung
0x6426	0	Anzahl Elemente	Unsigned8	ro	N	abhängig von Typ und Bestückung	Anzahl verfügbarer analoger Eingangs-Kanäle
	1	delta value 1st input	Unsigned16	rw	Y	0x0000	Deltawert 1. Eingangskanal
	...	...	...	...	...	...	...
	0XFE	delta value 254. input	Unsigned16	rw	Y	0x0000	Deltawert 254. Eingangskanal

Werte ungleich 0 aktivieren die Deltafunktion für diesen Kanal. Ein PDO wird dann abgesetzt wenn sich der Wert seit dem letzten Senden um mehr als den Deltawert verändert hat. Zusätzlich muss die Ereignissteuerung aktiviert sein (Objekt 0x6423). Das Datenformat entspricht dem der Analogeingänge (Deltawert: nur positive Werte).

5 Inbetriebnahme

5.1 Montage

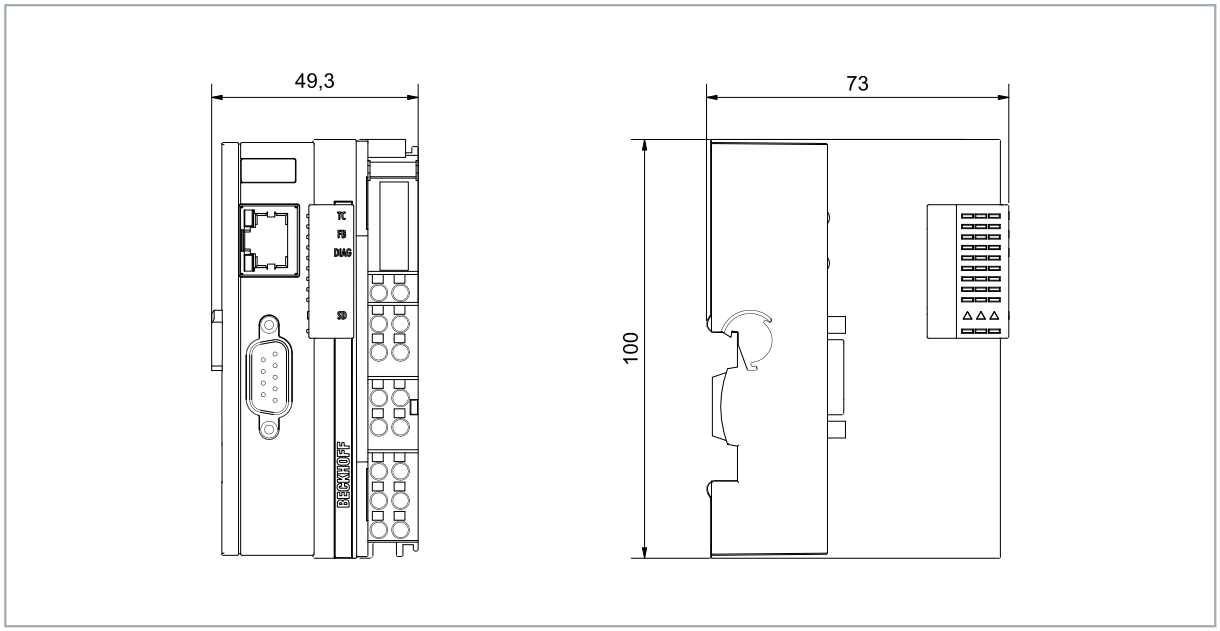


Abb. 17: Embedded-PC CX70xx, Abmessungen.

5.1.1 Zulässige Einbaulagen beachten

HINWEIS

Überhitzung

Bei einer falsch gewählten Einbaulage und nicht eingehaltenen Mindestabständen kann der Embedded-PC überhitzen. Halten Sie die maximale Umgebungstemperatur von 60°C und die Montagevorschriften ein.

Montieren Sie den Embedded-PC waagerecht im Schaltschrank auf einer Hutschiene, damit die Wärme optimal abgeführt wird.

Beachten Sie folgende Vorgaben für den Schaltschrank:

- Betreiben Sie den Embedded-PC nur bei Umgebungstemperaturen von -25 °C bis 60 °C. Messen Sie dazu die Temperatur unter dem Embedded-PC in einem Abstand von 30 mm zu den Kühlrippen, um die Umgebungstemperatur korrekt zu ermitteln.
- Halten Sie die Mindestabstände von 30 mm ober- und unterhalb des Embedded-PCs ein.
- Weitere elektrische Geräte beeinflussen die Wärmeentwicklung im Schaltschrank. Wählen Sie eine passende Schaltschrankgröße abhängig vom Anwendungsfall oder sorgen Sie dafür, dass überschüssige Wärme aus dem Schaltschrank abtransportiert wird.

Der Embedded-PC muss waagerecht auf die Hutschiene montiert werden. Die Lüftungsöffnungen befinden sich auf der Gehäuseunter- und Gehäuseoberseite. Auf diese Weise kommt ein optimaler Luftstrom zustande, der den Embedded-PC in vertikaler Richtung durchströmt. Zusätzlich ist ein Freiraum von mindestens 30 mm oberhalb und unterhalb des Embedded-PCs erforderlich, um eine ausreichende Belüftung zu gewährleisten.

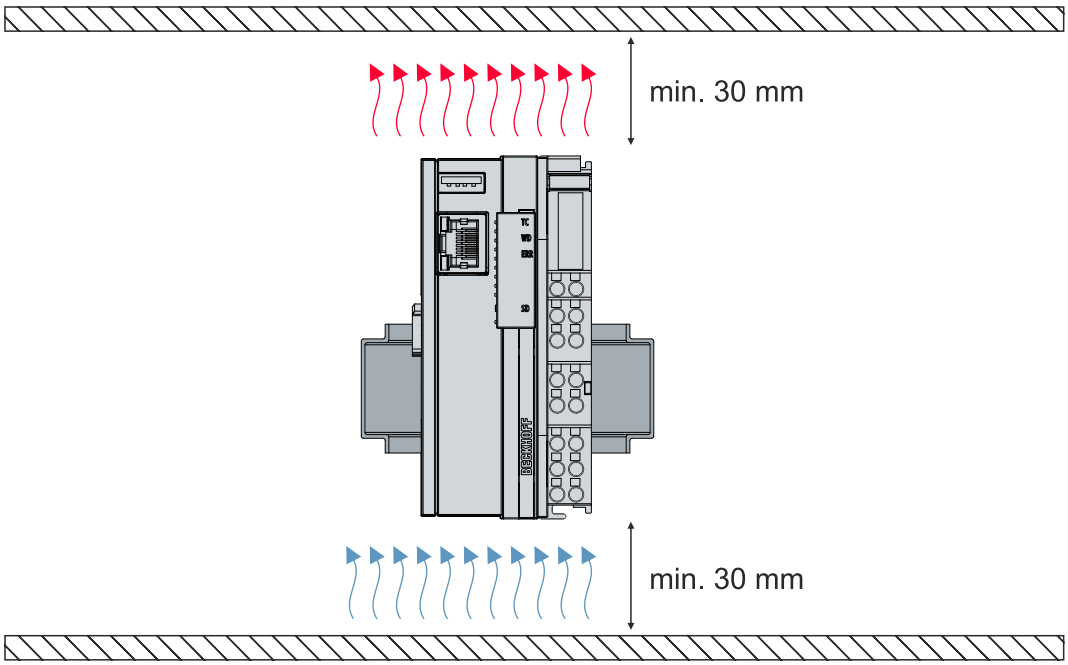


Abb. 18: Embedded-PC CX70xx, zulässige Einbaulage.

Wenn Vibrationen und Stöße in der gleichen Richtung verlaufen wie die Hutschiene, muss der Embedded-PC zusätzlich mit einer Halterung fixiert werden, damit er nicht verrutscht.

Einbaulagen mit eingeschränktem Temperaturbereich bis 45 °C

Sie können den Embedded-PC auch senkrecht oder liegend auf der Tragschiene montieren. Beachten Sie dabei, dass Sie den Embedded-PC dann nur bis zu einer Umgebungstemperatur von 45 °C betreiben können.

Achten Sie darauf, dass Busklemmen, die an den Embedded-PCs angeschlossenen werden, für den senkrechten oder liegenden Betrieb ausgelegt sind.

Einschränkungen bei E-Bus/K-Bus-Strom

Der maximale E-Bus/K-Bus-Strom variiert abhängig von der gewählten Einbaulage und Umgebungstemperatur.

Tab. 5: Maximaler E-Bus/K-Bus-Strom abhängig von Einbaulage und Umgebungstemperatur.

E-Bus/K-Bus-Strom	Einbaulage	Umgebungstemperatur
max. 1,5 A	beliebig	-25…45 °C
max. 1,3 A	horizontal	-25…55 °C
max. 1 A	beliebig	-25…55 °C
max. 1 A	horizontal	-25…60 °C

5.1.2 Auf Hutschiene befestigen

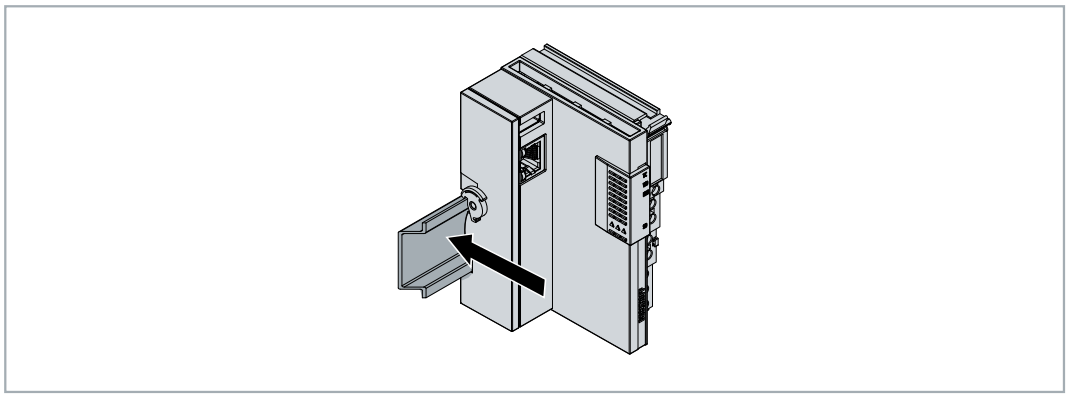
Das Gehäuse ist so konstruiert, dass der Embedded-PC an die Hutschiene gehalten und auf diese eingerastet werden kann.

Voraussetzungen:

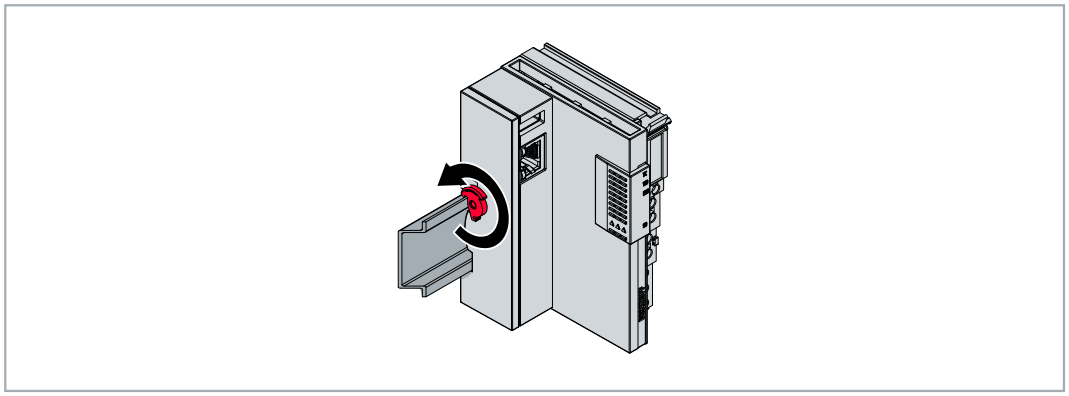
- Hutschiene von Typ TS35/7.5 oder TS35/15 nach DIN EN 60715.

Befestigen Sie den Embedded-PC wie folgt auf der Hutschiene:

1. Setzen Sie den Embedded-PC auf die Hutschiene. Drücken Sie den Embedded-PC leicht an die Hutschiene, bis es leise klickt und der Embedded-PC eingerastet ist.



2. Verriegeln Sie anschließend die Arretierung auf der linken Seite des Embedded-PCs.
3. Drehen Sie die Arretierung gegen den Uhrzeigersinn, bis die Arretierung leise klickt und einrastet.



⇨ Sie haben den Embedded-PC erfolgreich montiert. Überprüfen Sie noch mal die korrekte Montage und ob der Embedded-PC auf der Hutschiene eingerastet ist.

5.1.3 MicroSD-Karte wechseln

Datenverlust

MicroSD-Karten werden im Betrieb stark beansprucht und müssen viele Schreibzyklen und extreme Umweltbedingungen aushalten. MicroSD-Karten anderer Hersteller können ausfallen, was zu Datenverlust führt.

Verwendet Sie ausschließlich industrietaugliche MicroSD-Karten die von Beckhoff geliefert werden.

Der MicroSD-Kartenslot ist für eine industrietaugliche MicroSD-Karte vorgesehen. Auf der MicroSD-Karte wird die Firmware des Embedded-PCs gespeichert. Die MicroSD-Karte kann bei Bedarf aus TwinCAT 3 heraus beschrieben werden und dadurch benutzerdefinierte Daten gespeichert werden.

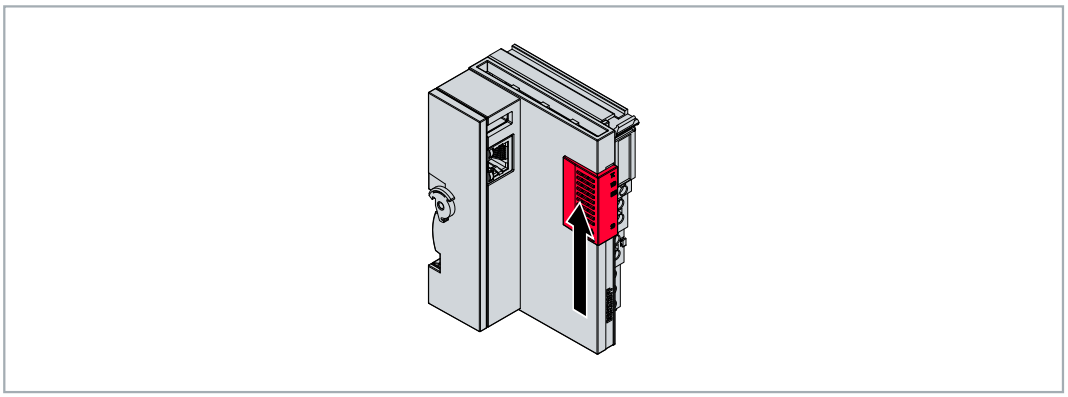
Die Auswurfmechanik wird nach dem Push-Push-Prinzip betätigt. Im Folgenden wird gezeigt, wie die MicroSD-Karte gewechselt wird.

Voraussetzungen:

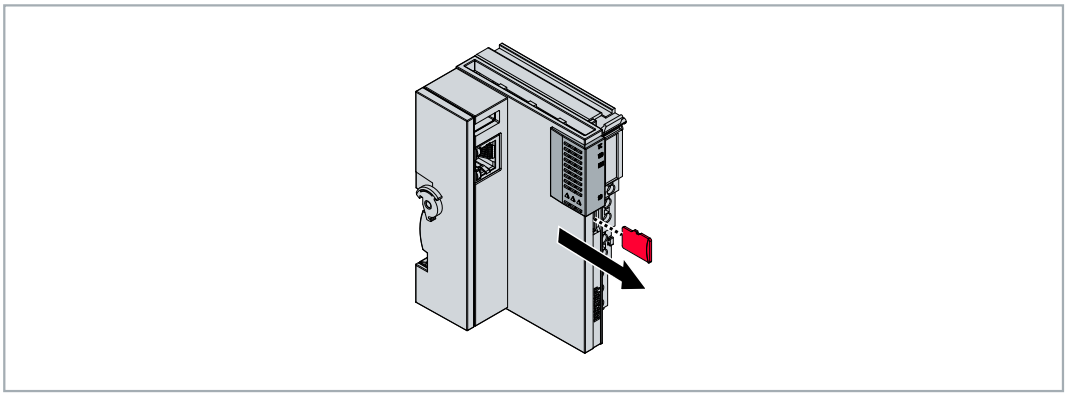
- Der Embedded-PC muss ausgeschaltet sein. Die MicroSD-Karte darf nur im ausgeschalteten Zustand ein- oder ausgebaut werden.

MicroSD-Karte wechseln

1. Schieben Sie die schwarze Abdeckung nach oben.



2. Drücken Sie leicht auf die MicroSD-Karte.
3. Die Karte wird mit einem leisen Klicken entriegelt und um ca. 2-3 mm aus dem Gehäuse gehoben.



4. Schieben Sie die neue MicroSD-Karte mit den Kontakten voran in den Kartenslot. Die Kontakte zeigen nach rechts.
5. Die MicroSD-Karte rastet mit einem leisen Klicken ein.

⇨ Die Karte sitzt richtig, wenn sie sich ca.1 mm tiefer als die Frontseite des Gehäuses befindet.

5.1.4 Passive EtherCAT-Klemmen montieren

Falsch montierte passive EtherCAT-Klemmen

Das E-Bus Signal zwischen einem Embedded-PC und den EtherCAT-Klemmen kann durch falsch montierte passive EtherCAT-Klemmen geschwächt werden.

Montieren Sie passive EtherCAT-Klemmen nicht direkt an das Netzteil.

EtherCAT-Klemmen, die nicht aktiv am Datenaustausch teilnehmen, werden als passive Klemmen bezeichnet. Dadurch haben passive EtherCAT-Klemmen kein Prozessabbild und benötigen keinen Strom aus dem Klemmbus (E-Bus).

Passive EtherCAT-Klemmen (z.B. eine EL9195) können Sie in TwinCAT erkennen. Die EtherCAT-Klemme wird im Strukturbaum ohne Prozessabbild angezeigt und der Wert in der Spalte „E-Bus (mA)“ verändert sich im Vergleich zu der vorangehenden EtherCAT-Klemme nicht.

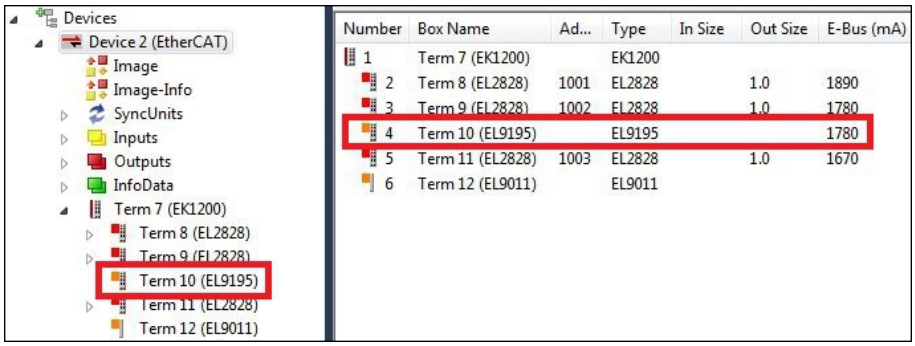


Abb. 19: Passive EtherCAT-Klemme in TwinCAT identifizieren.

In den technischen Daten einer EtherCAT-Klemme können Sie unter dem Eintrag „Stromaufnahme aus dem E-Bus“ nachlesen, ob eine bestimmte EtherCAT-Klemme Strom aus dem Klemmbus (E-Bus) benötigt.

Die folgende Abbildung zeigt die zulässige Montage einer passiven EtherCAT-Klemme. Die passive EtherCAT-Klemme wurde nicht direkt an das Netzteil angereiht.

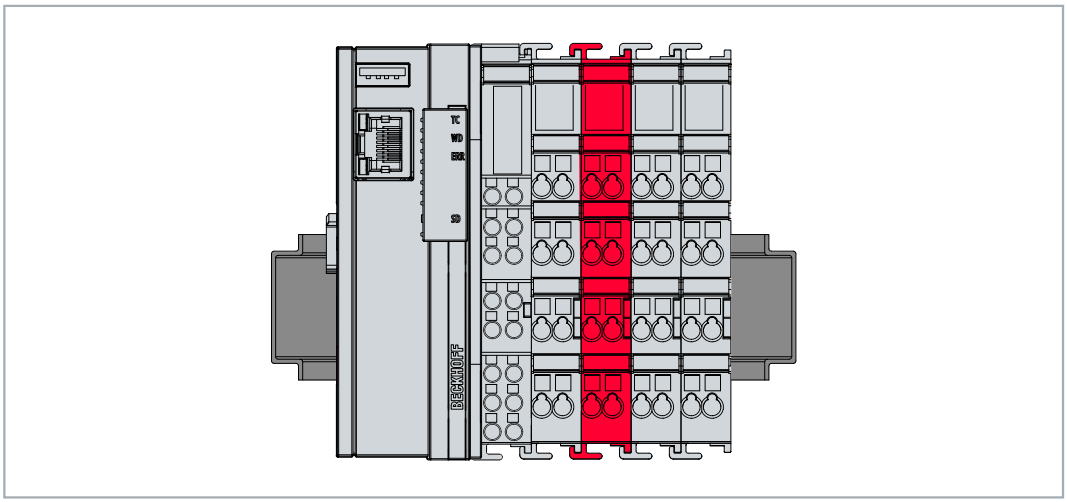


Abb. 20: Passive EtherCAT-Klemmen, zulässige Montage.

5.2 Spannungsversorgung

HINWEIS

Schäden an den Embedded-PCs

Die Embedded-PCs können während der Verdrahtung beschädigt werden. Schließen Sie die Leitungen für die Spannungsversorgung nur im spannungsfreien Zustand an.

Für die Spannungsversorgung der Netzteilklemme ist eine externe Spannungsquelle erforderlich, die eine 24 V Gleichspannung (-15 % / +20 %) bereitstellt.

Verkabeln Sie den Embedded-PC im Schaltschrank entsprechend der Norm EN 60204-1:2006 Schutzkleinspannungen (PELV = Protective Extra Low Voltage):

- Die Leiter “PE“ und “0 V“ der Spannungsquelle für ein CPU-Grundmodul müssen auf dem gleichen Potential liegen (im Schaltschrank verbunden).
- Die Norm EN 60204-1:2006 Abschnitt 6.4.1:b: schreibt vor, dass eine Seite des Stromkreises oder ein Punkt der Energiequelle dieses Stromkreises an das Schutzleitersystem angeschlossen werden muss.

Anschlüsse

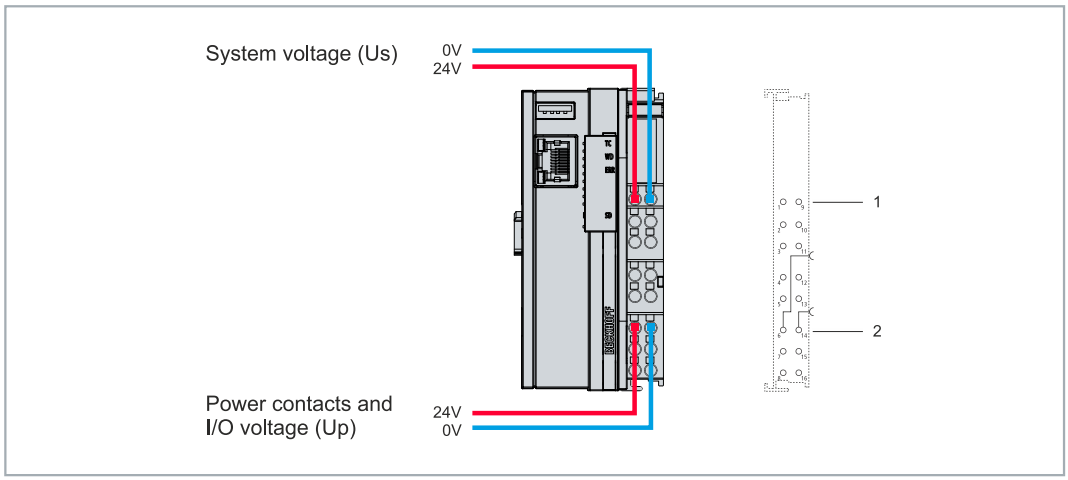


Abb. 21: Anschlüsse für Systemspannung (Us) und Powerkontakte (Up).

Tab. 6: Legende zum Anschlussbeispiel.

Nr.	Beschreibung
1	Die oberen Federkraftklemmen mit der Bezeichnung “+24 V Us“ und “0 V Us“ versorgen das CPU-Grundmodul und den Klemmenbus (Datenübertragung über K- oder E-Bus) mit Spannung.
2	Die Federkraftklemmen mit der Bezeichnung “+24 V Up“ und “0 V Up“ versorgen die Multifunktions-I/Os, die Busklemmen und EtherCAT-Klemmen über die Powerkontakte mit Spannung.

Sicherung

- Beachten Sie bei der Dimensionierung der Sicherung für die Systemspannung (Us) die max. Leistungsaufnahme des Embedded-PCs (siehe: Technische Daten [▶ 201])
- Sichern Sie Powerkontakte (Up) mit einer Sicherung von max. 10 A (träge) ab.

Spannungsversorgung unterbrechen / abschalten

Um den Embedded-PC abzuschalten, darf nicht die Masse (0 V) getrennt werden, da sonst je nach Gerät der Strom über den Schirm weiterfließt und der Embedded-PC oder die Peripherie beschädigt wird.

Trennen Sie immer die 24 V Leitung. An dem Embedded-PC angeschlossene Geräte mit eigener Stromversorgung (z.B. ein Panel) müssen für “PE“ und „0 V“ das gleiche Potential wie der Embedded-PC haben (keine Potentialdifferenz).

5.2.1 Embedded-PC anschließen

Die Leitungen einer externen Spannungsquelle werden mit Federkraftklemmen an der Netzteilklemme verbunden. Beachten Sie die erforderlichen Leiterquerschnitte und Abisolierlängen.

Tab. 7: Erforderliche Leiterquerschnitte und Abisolierlängen.

Leiterquerschnitt	e*: 0,08 ... 1,5 mm^2 f*: 0,25 ... 1,5 mm^2 a*: 0,14 ... 0,75 mm^2	e*: AWG 28 ... 16 f*: AWG 22 ... 16 a*: AWG 26 ... 19
Abisolierlänge	8 ... 9 mm	0.33 inch

*e: eindrähtig, Draht massiv; f: feindrähtig, Litze; a: mit Aderendhülse

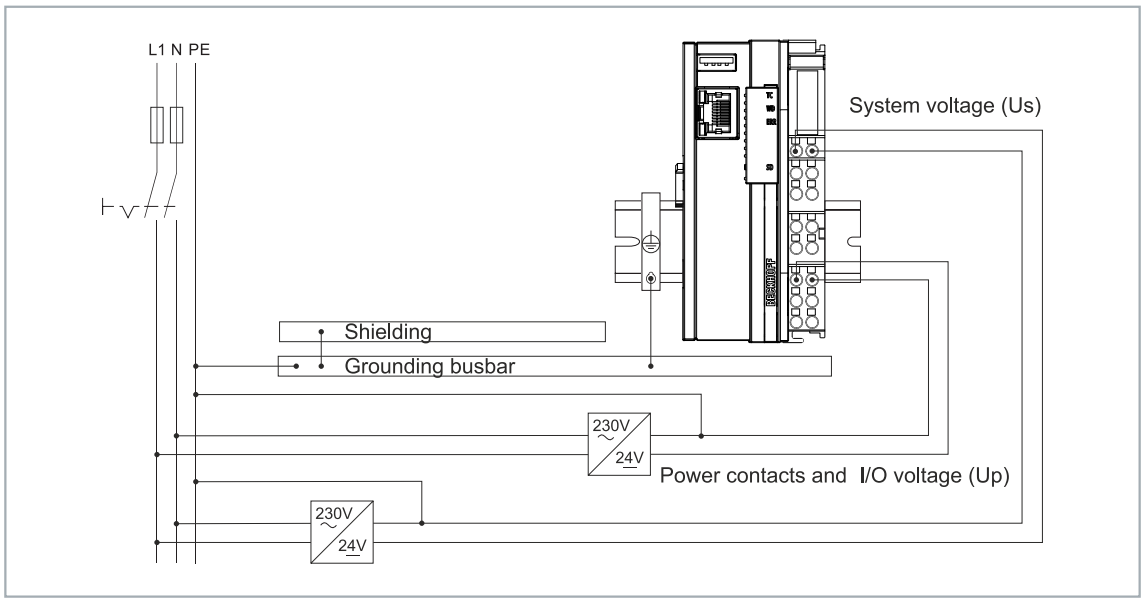
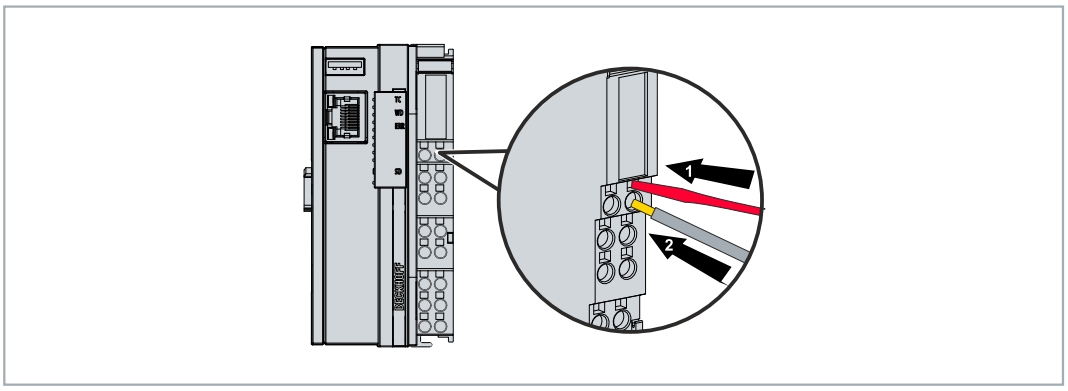


Abb. 22: Anschlussbeispiel mit einem CX7000.

Schließen Sie den Embedded-PC wie folgt an:

1. Öffnen Sie eine Federkraftklemme, indem Sie mit einem Schraubendreher oder einem Dorn leicht in die viereckige Öffnung über der Klemme drücken.



2. Der Draht kann nun ohne Widerstand in die runde Klemmenöffnung eingeführt werden.
3. Durch Rücknahme des Druckes schließt sich die Klemme automatisch und hält den Draht sicher und dauerhaft fest.

⇨ Sie haben die Spannungsquelle erfolgreich an die Netzteilklemme angeschlossenen, wenn die beiden oberen LEDs der Netzteilklemme grün aufleuchten.

Die linke LED (Us 24V) zeigt die Versorgung des CPU-Grundmoduls und des Klemmenbusses an. Die rechte LED (Up 24V) zeigt die Versorgung der Busklemmen über die Powerkontakte an.

5.2.2 UL-Anforderungen

Die Embedded-PCs CX7050 sind UL-zertifiziert. Das entsprechende UL-Label befindet sich auf dem Typenschild.

Die Embedded-PCs CX7050 können damit in Bereichen eingesetzt werden, in denen spezielle UL-Anforderungen eingehalten werden müssen. Diese Anforderungen gelten für die Systemspannung (Us) und für die Powerkontakte (Up). Einsatzbereiche ohne spezielle UL-Anforderungen sind von den UL-Vorschriften nicht betroffen.

UL-Anforderungen:

- Die Embedded-PCs dürfen nicht mit unbegrenzten Spannungsquellen verbunden werden.
- Embedded-PCs dürfen nur mit einer Spannungsquelle von 24 V Gleichspannung versorgt werden. Die Spannungsquelle muss isoliert sein und mit einer Sicherung (entsprechend UL248) von maximal 4 A geschützt werden.
- Oder die Spannungsversorgung muss von einer Spannungsquelle stammen, die NEC class 2 entspricht. Eine Spannungsquelle entsprechend NEC class 2 darf dabei nicht seriell oder parallel mit einer anderen NEC class 2 Spannungsquelle verbunden werden.

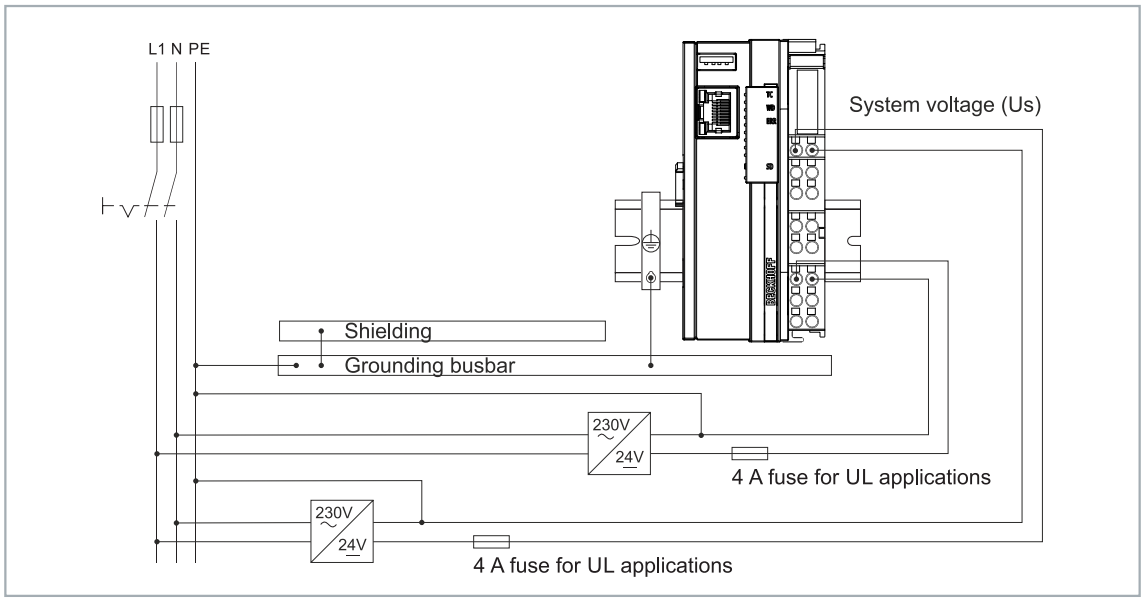


Abb. 23: Anschlussbeispiel für Bereiche mit speziellen UL-Anforderungen.

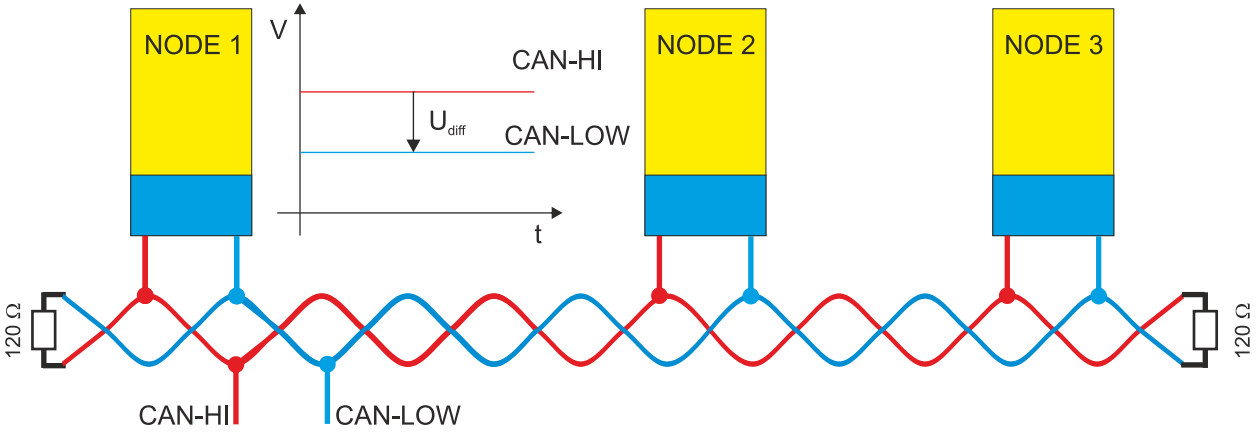
5.3 CANopen: Anschluss und Verdrahtung

HINWEIS

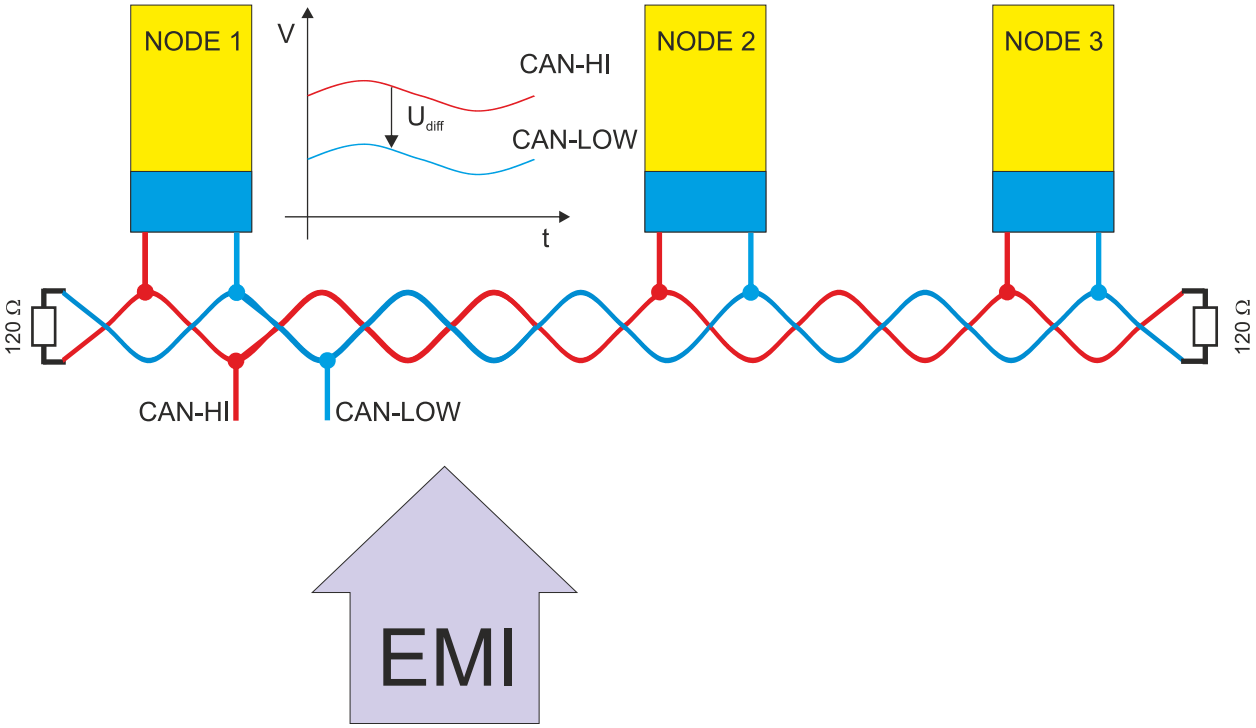
Unsachgemäße Verkabelung

Durch die nicht vorhandene galvanische Trennung kann durch unsachgemäße Verkabelung der CAN-Treiber zerstört oder geschädigt werden. Verkabeln sie immer im ausgeschalteten Zustand. Verkabeln Sie erst die Spannungsversorgung und legen erst dann den CAN auf.

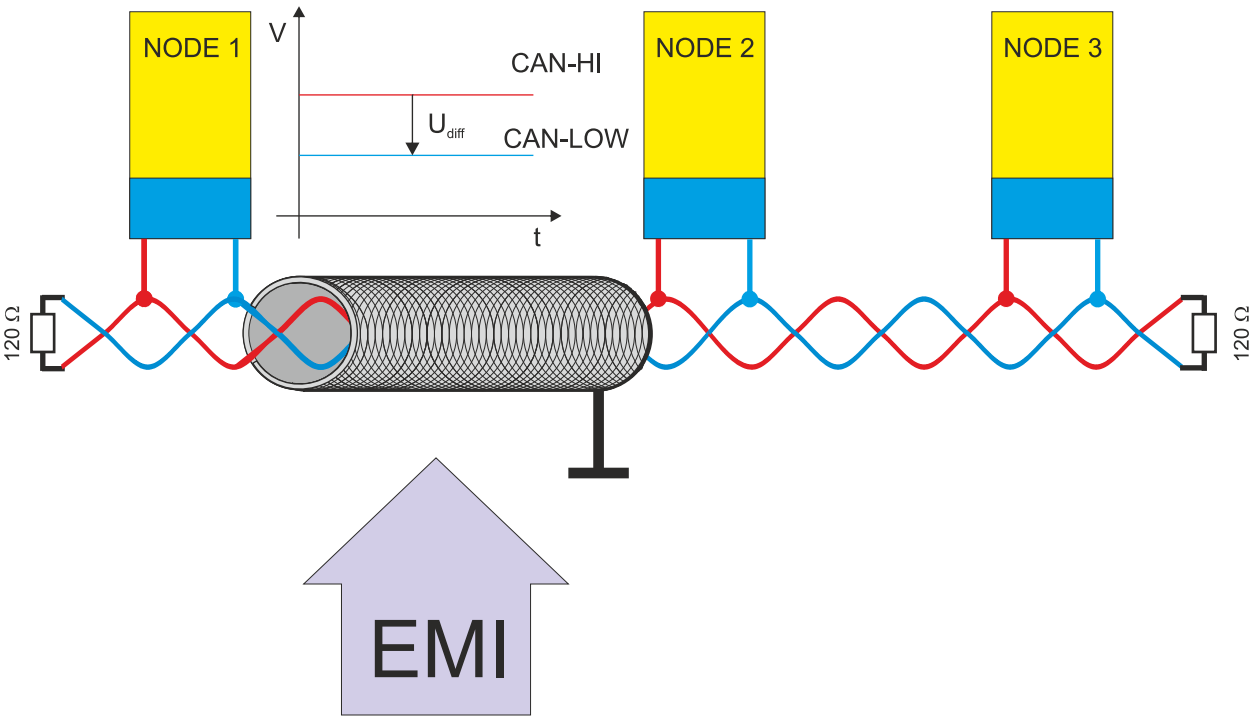
CAN ist ein 2-Draht-Bussystem, an dem alle Teilnehmer parallel (d.h. mit kurzen Stichleitungen) angeschlossen werden. Der Bus muss an jedem Ende mit einem Abschlusswiderstand von 120 (bzw. 121) Ohm abgeschlossen werden, um Reflexionen zu vermeiden. Dies ist auch bei sehr kurzen Leitungslängen erforderlich!



Da die CAN-Signale als Differenzpegel auf dem Bus dargestellt werden, ist die CAN-Leitung vergleichsweise unempfindlich gegen eingeprägte Störungen (EMI). Es sind jeweils beide Leitungen betroffen, somit verändert die Störung den Differenzpegel kaum.



Bei einer zusätzlichen Abschirmung der verdrillten Leitungsadern können störende Einflüsse durch EMI weiter eliminiert werden.



Buslänge

Die maximale Buslänge wird bei CAN vorwiegend durch die Signallaufzeit beschränkt. Das Multi-Master-Buszugriffsverfahren (Arbitrierung) erfordert, dass die Signale quasi gleichzeitig (vor der Abtastung innerhalb einer Bitzeit) an allen Knoten anliegen. Da die Signallaufzeit in den CAN-Anschaltungen (Transceiver, Optokoppler, CAN-Controller) nahezu konstant sind, muss die Leitungslänge an die Baud-Rate angepasst werden.

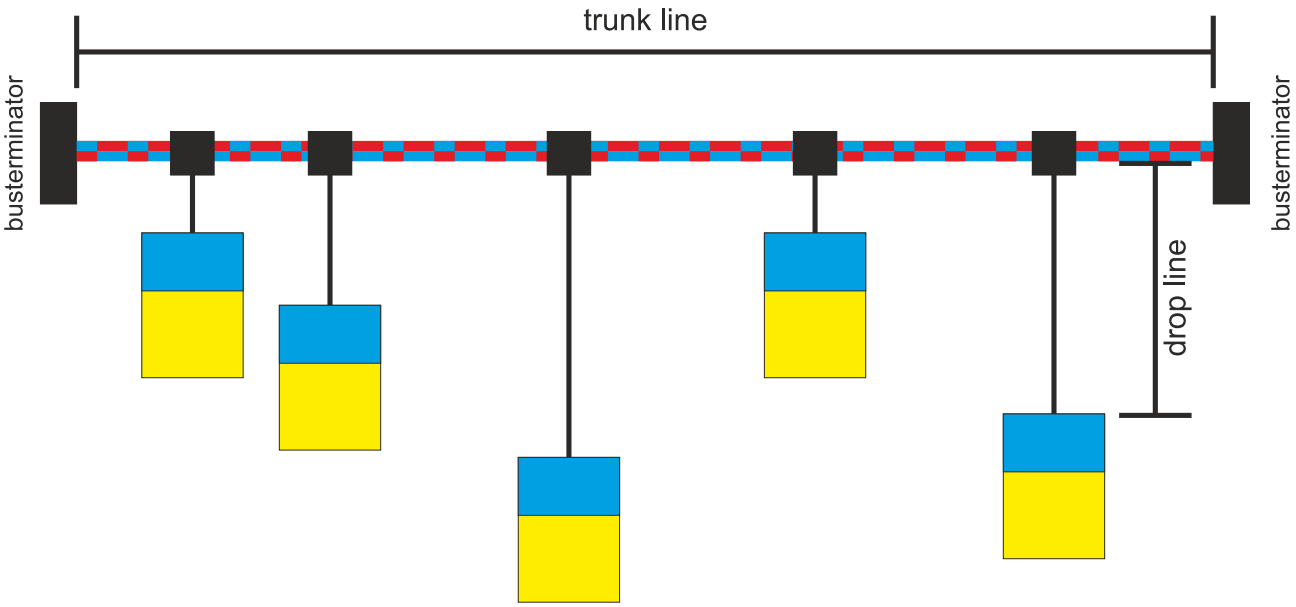
Baud-Rate	Buslänge
1 MBit/s	< 20 m*
500 kBit/s	< 100 m
250 kBit/s	< 250 m
125 kBit/s	< 500 m
50 kBit/s	< 1000 m
20 kBit/s	< 2500 m
10 kBit/s	< 5000 m

*) Häufig findet man in der Literatur für CAN die Angabe 40m bei 1 MBit/s. Dies gilt jedoch nicht für Netze mit optoentkoppelten CAN-Controllern. Die worst case Berechnung mit Optokopplern ergibt bei 1 MBit/s eine maximale Buslänge von 5m - erfahrungsgemäß sind jedoch 20m problemlos erreichbar.

Bei Buslängen über 1000m kann der Einsatz von Repeatern notwendig werden.

Stichleitungen

Stichleitungen ("drop lines") sind nach Möglichkeit zu vermeiden, da sie grundsätzlich zu Signalreflexionen führen. Die durch Stichleitungen hervorgerufenen Reflexionen sind jedoch in der Regel unkritisch, wenn sie vor dem Abtastzeitpunkt vollständig abgeklungen sind.



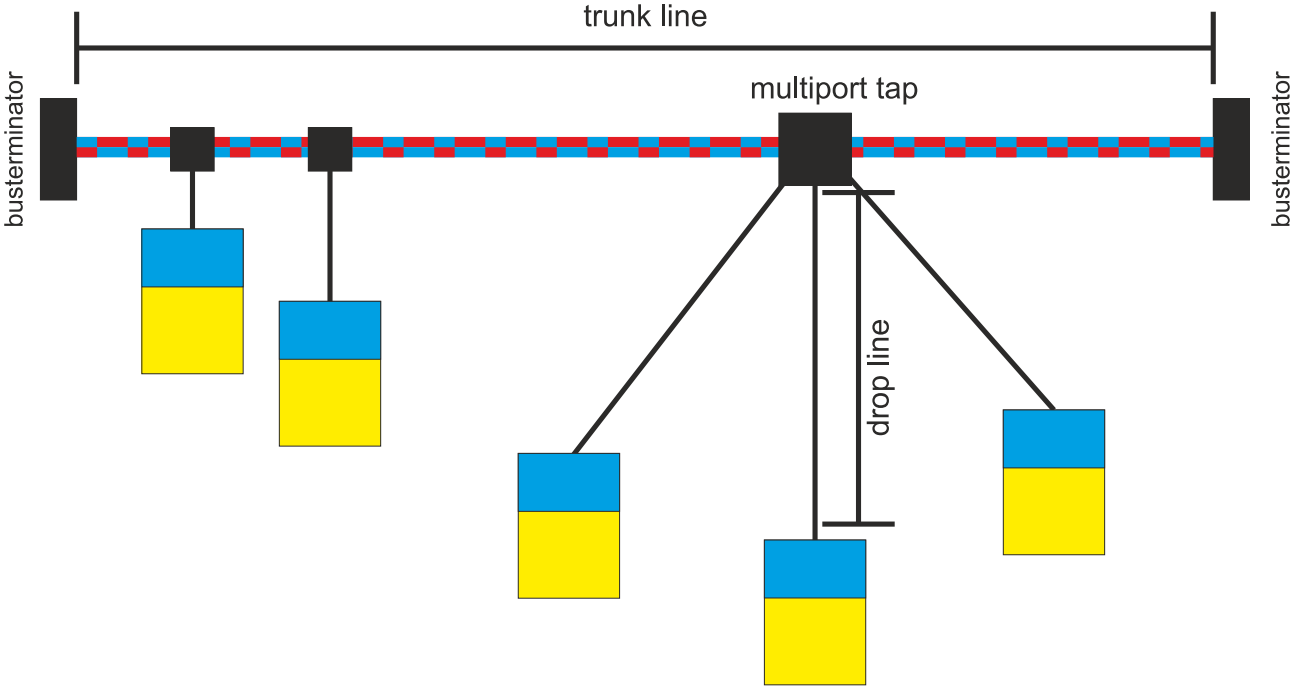
Bei den in den Buskopplern gewählten Bit-Timing-Einstellungen kann dies angenommen werden, wenn folgende Stichleitungslängen nicht überschritten werden:

Baud-Rate	Länge Stichleitung	gesamte Länge aller Stichleitungen
1 MBit/s	< 1m	< 5 m
500 kBit/s	< 5 m	< 25 m
250 kBit/s	< 10m	< 50 m
125 kBit/s	< 20m	< 100 m
50 kBit/s	< 50m	< 250 m

Stichleitungen dürfen nicht mit Abschlusswiderständen versehen werden.

Sternverteiler (Multiport Tap)

Beim Einsatz von passiven Verteilern ("Multiport Taps"), z.B. der BECKHOFF Verteilerbox ZS5052-4500 sind kürzere Stichleitungslängen einzuhalten.



Die folgende Tabelle gibt die maximalen Stichleitungslängen und die maximale Länge der Trunk Line (ohne Stichleitungen) an:

Baud-Rate	Länge Stichleitung bei Multiport Topologie	Länge Trunk Line (ohne Stichleitungen)
1 MBit/s	< 0,3 m	< 25 m
500 kBit/s	< 1,2 m	< 66 m
250 kBit/s	< 2,4 m	< 120 m
125 kBit/s	< 4,8 m	< 310 m

5.3.1 D-Sub-Stecker (X003)

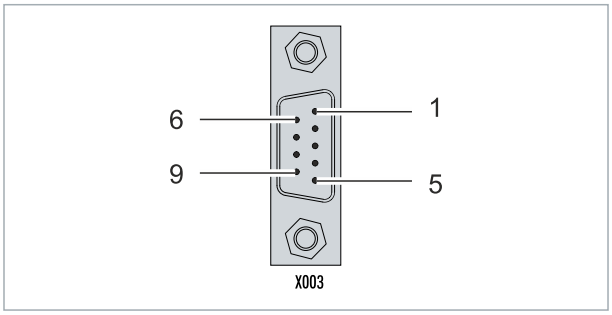


Abb. 24: CANopen-Schnittstelle X003.

Die CAN-Busleitung wird über eine 9polige D-Sub-Stecker mit folgender Belegung angeschlossen:

Pin	Belegung
1	nicht benutzt
2	CAN low (CAN-)
3	CAN Ground (intern verbunden mit Pin 6)
4	nicht benutzt
5	Schirm
6	CAN Ground (intern verbunden mit Pin 3)
7	CAN high (CAN+)
8	nicht benutzt
9	nicht benutzt

Die Hutschienenkontaktfeder und der Steckerschirm sind durchverbunden. An Pin 9 darf eine Hilfsspannung bis 30 V_{DC} angeschlossen werden, die von manchen CAN-Geräten zur Versorgung der Transceiver genutzt wird.

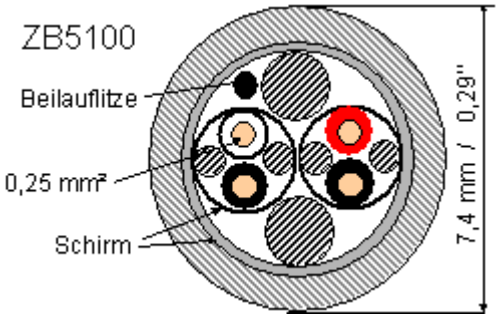
5.3.2 Kabel und Schirmung

Für die CAN-Verdrahtung wird die Verwendung von paarig verdrillten, geschirmten Kabeln (2x2) mit einem Wellenwiderstand von 108...132 Ohm empfohlen. Wenn das Bezugspotential der CAN-Transceiver (CAN-Ground) nicht verbunden werden soll, so kann auf das zweite Adernpaar verzichtet werden (nur bei kleinen Netzausdehnungen mit gemeinsamer Speisung aller Teilnehmer empfehlenswert).

ZB5100 CAN-Kabel

BECKHOFF hat ein hochwertiges CAN-Kabel mit folgenden Eigenschaften im Programm:

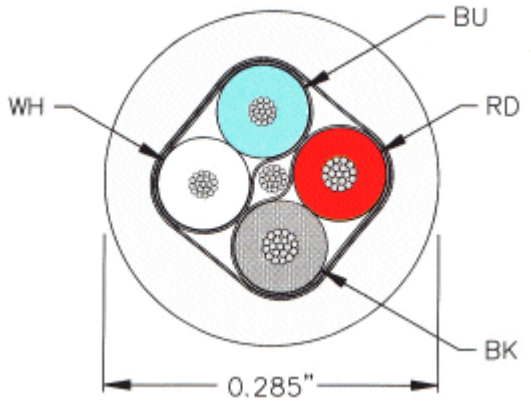
- 2 x 2 x 0,25 mm² (AWG 24) paarig verseilt, Kabelfarben: rot/schwarz + weiß/schwarz
- doppelt geschirmt
- Schirmgeflecht mit Beilauflitze (kann direkt auf Pin3 der 5-pol Anschlussklemme aufgelegt werden),
- flexibel (Mindestbiegeradius 35mm bei einmaligem Biegen, 70mm bei mehrmaligem Biegen)
- Wellenwiderstand (60kHz): 120 Ohm
- Leiterwiderstand < 80 Ohm/km
- Mantel: PVC grau, Außendurchmesser 7,3 +/- 0,4 mm
- Gewicht: 64 kg/km.
- Bedruckt mit "BECKHOFF ZB5100 CAN-BUS 2x2x0.25" und Metermarkierung (Längenangaben, alle 20cm)



ZB5200 CAN/DeviceNet-Kabel

Das Kabelmaterial ZB5200 entspricht der DeviceNet Spezifikation und eignet sich ebenso für CANopen Systeme. Aus diesem Kabelmaterial sind auch die vorkonfektionierten Busleitungen ZK1052-xxxx-xxxx für die Feldbus Box Baugruppen gefertigt. Es hat folgende Spezifikation:

- 2 x 2 x 0,34 mm² (AWG 22) paarig verseilt
- doppelt geschirmt · Schirmgeflecht mit Beilauflitze
- Wellenwiderstand (1 MHz): 126 Ohm
- Leiterwiderstand 54 Ohm/km
- Mantel: PVC grau, Außendurchmesser 7,3 mm
- Bedruckt mit "InterlinkBT DeviceNet Type 572" sowie UL und CSA Ratings
- Litzenfarben entsprechen DeviceNet Spezifikation
- UL anerkanntes AWM Type 2476 Rating
- CSA AWM I/II A/B 80°C 300V FT1
- Entspricht DeviceNet "Thin Cable" Spezifikation



Schirmung

Der Schirm ist über das gesamte Buskabel zu verbinden und nur an einer Stelle galvanisch zu erden um Masseschleifen zu vermeiden.
Das Schirmungskonzept mit HF-Ableitung von Störungen über R/C-Glieder auf die Tragschiene geht davon aus, dass die Tragschiene entsprechend geerdet und störungsfrei ist. Sollte dies nicht der Fall sein, so kann es vorkommen, dass HF-Störpegel über die Tragschiene auf den Schirm des Buskabels übertragen werden. In diesem Fall sollte der Schirm an den Kopplern nicht aufgelegt werden - aber dennoch komplett durchverbunden sein.

Kabelfarben

Vorschlag für die Verwendung der Beckhoff CAN-Kabel:

Funktion	Kabelfarbe ZB5100	Kabelfarbe ZB5200
CAN Ground	**schwarz** /(rot)	**schwarz**
CAN Low	**schwarz**	**blau**
Schirm	Beilauflitze	Beilauflitze
CAN high	**weiß**	**weiß**
nicht benutzt	**(rot)**	**(rot)**

6 Multifunktions-I/Os

Für die Konfiguration der Betriebsarten stehen insgesamt vier einstellbare Slots zur Verfügung. Ein Slot ist eine bestimmte Anzahl von Ein- und Ausgängen. Für jeden Slot kann maximal ein Modul (DI, DIO, ENC, CNT oder PWM) zugewiesen werden, welches wiederum die Betriebsart für den jeweiligen Slot festlegt. Ein Modul ist also eine Funktion, die diese Ein-und Ausgänge übernehmen können. Die aktuelle Modulkonfiguration wird in TwinCAT unter der CX7028-Schnittstelle aufgelistet. Beachten Sie, dass die CX7028-Schnittstelle für die Steuerung der Multifunktions-I/Os eine eigene CPU hat und die CX7028-Schnittstelle unter TwinCAT nicht angezeigt wird bzw. nicht funktioniert, wenn die Spannungsversorgung (Up) nicht angeschlossen ist.

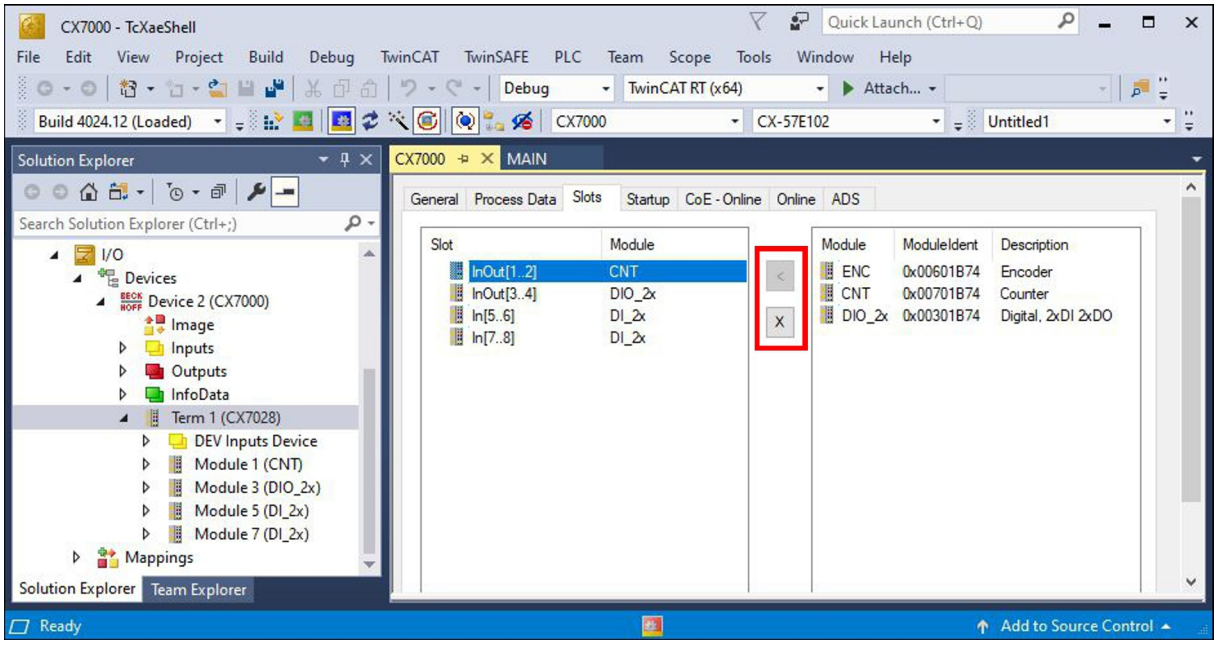


Abb. 25: CX7028-Schnittstelle, Slot- und Modul-Konfiguration unter TwinCAT.

Mit der Schaltfläche **<** können Module einem bestimmten Slot zugewiesen oder mit **x** wieder entfernt werden. Abhängig vom verwendeten Slot stehen unterschiedliche Module zur Auswahl. Welche Module von welchem Slot unterstützt werden, wird im Folgenden aufgelistet.

Zykluszeit für Multifunktions-I/Os

Die Kommunikation zu den Multifunktions-I/Os wird mit einem festen Watchdog von 100 ms überwacht. Das bedeutet, dass die Zykluszeit für die Multifunktions-I/O schneller als 100 ms sein muss.

Slot 1:

Bei der Verwendung von Slot 1 werden die Eingänge 1, 2 und (*3) sowie die Ausgänge 1 und 2 konfiguriert.

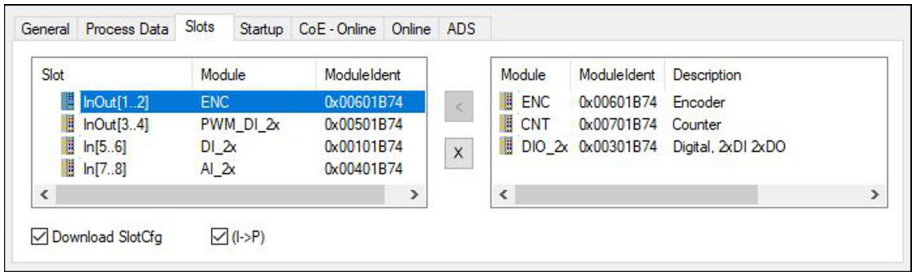


Abb. 26: Unterstützte Module bei der Verwendung von Slot 1.

- ENC (Inkremental-Encoder-Modus). 2 x Digitaleingang für 250-kHz-Encoder-Signal, 2 x Encoder-Digitalausgang.
- CNT (Zähler-Modus). 1 x Zähler-Digitaleingang 100 kHz, 1 x Digitaleingang als Auf-/Abwärts-Zähler 20 kHz, 2 x Zähler-Digitalausgang.

- DIO_2x (Digitale Ein- und Ausgänge). 2 x Digitaleingang, 24 V DC, Filter 3 ms, Typ 3, 2 x Digitalausgang, 24 V DC, 0,5 A, 1-Leitertechnik.

*) Der Eingang 3 ist nur im Inkremental-Encoder-Modus verfügbar. Bei einem High-Pegel kann der Wert des Inkremental-Enconders gelatcht oder der Zähler zurückgesetzt werden.

Slot 2:

Bei der Verwendung von Slot 2 werden die Eingänge 3 und 4, sowie die Ausgänge 3 und 4 konfiguriert.

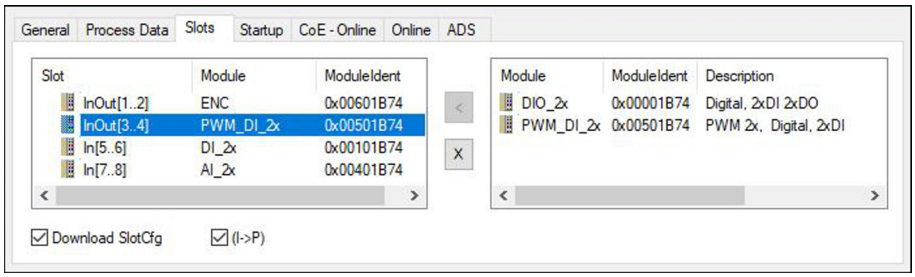


Abb. 27: Unterstützte Module bei der Verwendung von Slot 2.

- DIO_2x (Digitale Ein- und Ausgänge). 2 x Digitaleingang, 24 V DC, Filter 3 ms, Typ 3, 2 x Digitalausgang, 24 V DC, 0,5 A, 1-Leitertechnik.
- PWM_DI_2x (PWM-Signal-Modus). 2 x Digitaleingang, 24 V DC, Filter 3 ms, 2 x Digitalausgang konfiguriert für PWM-Signal.

Slot 3:

Bei der Verwendung von Slot 3 werden die Eingänge 5 und 6 konfiguriert.

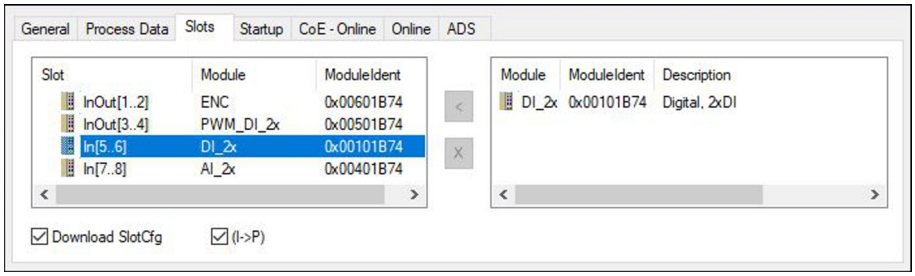


Abb. 28: Unterstützte Module bei der Verwendung von Slot 3.

Slot 3 beinhaltet nur ein Modul und kann daher nicht anders konfiguriert werden. Das Modul unterstützt 2 x Digitaleingang, 24 V DC, Filter 3 ms, Typ 3.

Slot 4:

Bei der Verwendung von Slot 4 werden die Eingänge 7 und 8 konfiguriert.

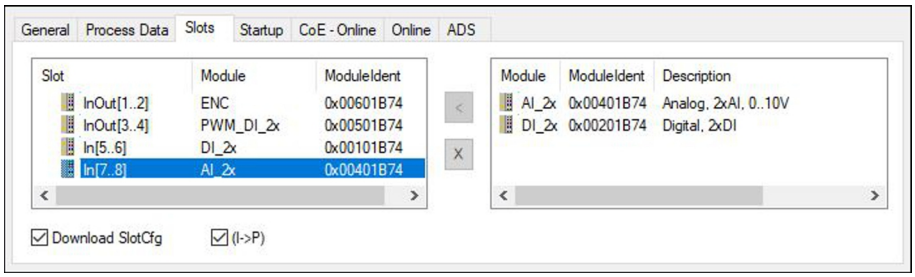


Abb. 29: Unterstützte Module bei der Verwendung von Slot 4.

- AI_2x (Analog-Signal-Modus). 2 x Digitaleingang konfiguriert als Analogeingang 0 bis 10 V, 12 Bit
- DI_2x (Digitaler Eingang). 2 x Digitaleingang, 24 V DC, Filter 3 ms, Typ 3

6.1 Digitale Eingänge

Die digitalen Eingänge erfassen binäre Steuersignale aus der Prozessebene. Typischerweise sind dies mechanische Kontakte wie Öffner- oder Schließer-Kontakte, elektronische Sensoren, wie induktive Näherungsschalter, optische Sensoren oder andere Methoden, um ein Low-/High-Signal im Sinne der Steuerungstechnik zu erzeugen. Der CX70xx verfügt dank integrierter Multifunktions-I/Os über insgesamt 8 digitale Eingänge, 24 V DC, Filter 3 ms, Typ 3.

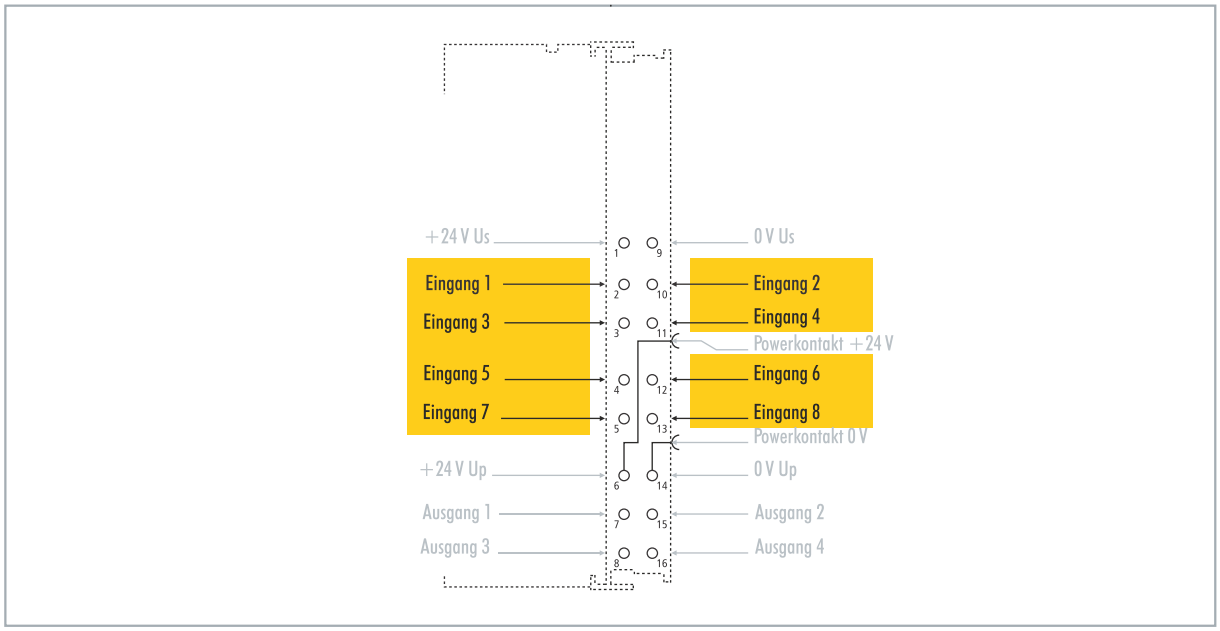


Abb. 30: Konfigurierbare digitale Eingänge.

Die digitalen Eingänge verfügen über einen 3 ms-Eingangsfilter. Der Signalzustand der einzelnen Eingänge wird durch jeweils eine Leuchtdiode angezeigt. Für die digitalen Eingänge 3, 4, 5 und 6 können zusätzliche Filtereinstellungen in den passenden CoE-Objekten vorgenommen und beispielsweise die Auflösung und Filterzeit eingestellt werden.

Tab. 8: Technische Daten, Multifunktions-I/Os als digitale Eingänge.

Technische Daten	CX7050
Anschlusstechnik	1-Leiter
Anzahl Eingänge	8
Nennspannung	24 V DC (-15 %/+20 %)
Spezifikation	EN 61131-2, Typ 3
Signalspannung „0“	-3…+5 V
Signalspannung „1“	11…30 V
Eingangsfilter	Konfigurierbar, Standard: 3 ms, min.: 10 µs
Anschlussquerschnitt	e*: 0,08…1,5 mm², f*: 0,25…1,5 mm², a*: 0,14…0,75 mm²
Anschlussquerschnitt AWG	e*: AWG 28…16, f*: AWG 22…16, a*: AWG 26…19
Abisolierlänge	8…9 mm

*e: eindrähtig, Draht massiv; f: feindrähtig, Litze; a: mit Aderendhülse

6.2 Digitale Ausgänge

HINWEIS

Rückspeisung bei den 24-V-Ausgängen

Eine Spannung von 24 V an den Ausgängen kann das Gerät zerstören, wenn die Spannungsversorgung (Up) nicht angeschlossen ist (Rückspeisung). Schließen Sie die Spannungsversorgung (Up) an, damit 24 V an den Ausgängen angelegt werden dürfen.

Die digitalen Ausgänge schalten binäre 24-V-DC-Steuersignale galvanisch getrennt zur Prozessebene an Aktoren weiter. Dabei entspricht der High-Pegel bei der positiv schaltenden Logik der Versorgungsspannung.

Die Ausgänge 3 und 4 verfügen über eine PWM-Endstufe. Wenn die beiden digitalen Ausgänge als normale digitale Ausgänge verwendet werden, kommt es durch die interne Beschaltung zu einem Leckstrom von kleiner 100 µA, der eine Spannung von ca. 5 V verursacht. Will man beim Low-Pegel des Ausgangs annähernd 0 V erreichen, muss ein 47 kΩ Widerstand gegen Masse geschaltet werden.

Eine weitere Möglichkeit besteht darin, die beiden Ausgänge im PWM-Modus zu betreiben und die Variable `PWM output` des PWM-Signals für FALSE mit 0x0000 und für TRUE mit 0xFFFF zu beschreiben. Damit ist die PWM-Endstufe aktiv, die keinen Leckstrom erzeugt.

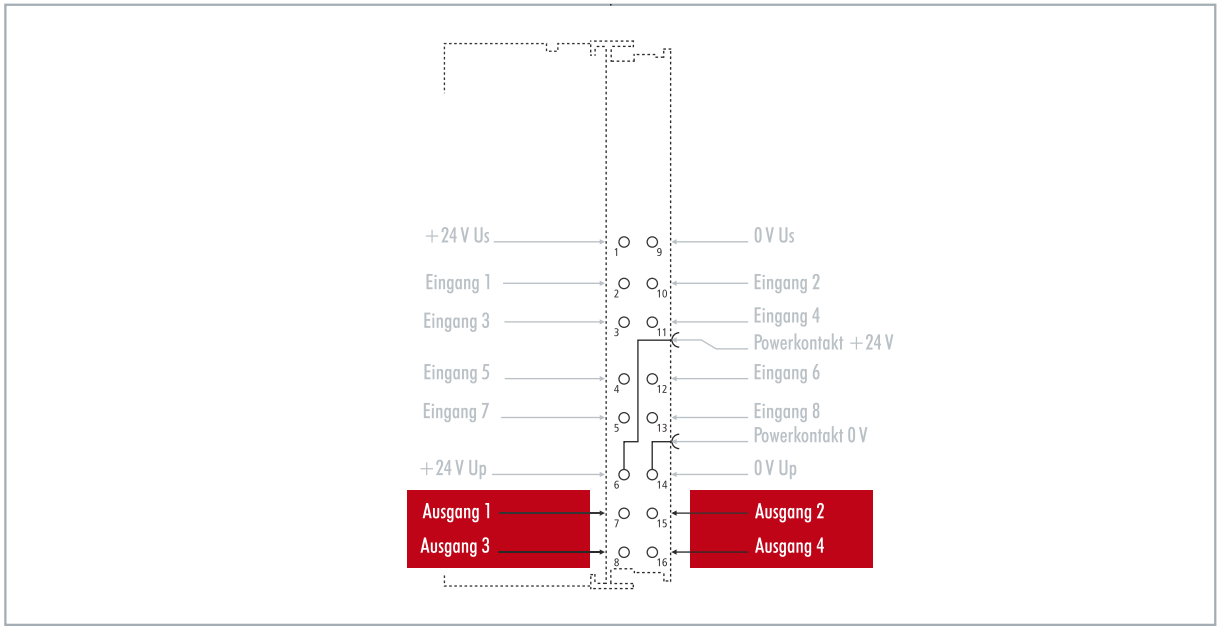


Abb. 31: Konfigurierbare digitale Ausgänge.

Der CX7050 enthält insgesamt vier Ausgänge, die ihren Signalzustand durch Leuchtdioden anzeigen. Mit den Ausgängen lassen sich Standardaktoren wie beispielsweise Schütze und Ventile schalten.

Tab. 9: Technische Daten, Multifunktions-I/Os als digitale Ausgänge.

Technische Daten	CX7050
Anschlusstechnik	1-Leiter
Anzahl Ausgänge	4
Nennspannung	24 V DC (-15 %/+20 %)
Lastart	ohmsch, induktiv, Lampenlast
Ausgangsstrom max.	24 V/0,5 A (kurzschlussfest)
Schaltzeiten	T_{ON}: 20 µs typ., T_{OFF}: 10 µs typ.
Kurzschlussstrom	< 2 A typ.
Abschaltenergie (ind.) max.	< 150 mJ/Kanal

Technische Daten	CX7050
Anschlussquerschnitt	e*: 0,08…1,5 mm², f*: 0,25…1,5 mm², a*: 0,14…0,75 mm²
Anschlussquerschnitt AWG	e*: AWG 28…16, f*: AWG 22…16, a*: AWG 26…19
Abisolierlänge	8…9 mm

*e: eindrähtig, Draht massiv; f: feindrähtig, Litze; a: mit Aderendhülse

6.3 Zähler-Modus

Der Embedded-PC CX7050 kann als ein Vor-/Rückwärtszähler konfiguriert werden, der das Zählen eines Pulses ermöglicht. Der Embedded-PC ist für schnelle Zählaufgaben mit einer Grenzfrequenz bis 100 kHz geeignet, wobei der CX7050 im 1-Zähler-Modus betrieben werden kann.

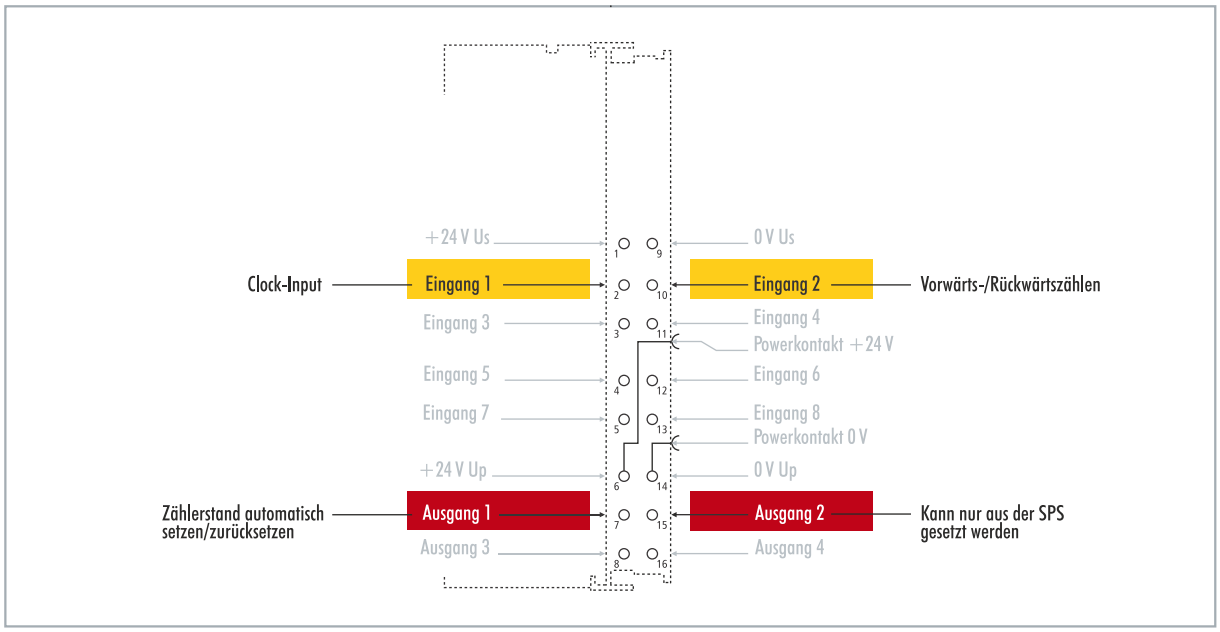


Abb. 32: Konfigurierbare Ein- und Ausgänge im Zähler-Modus.

Der CX7050 unterstützt im Zähler-Modus drei Betriebsarten:

- Vor-/Rückwärtszähler
- Vorwärtszähler
- Rückwärtszähler

Zusätzlich lässt sich Ausgang 1 abhängig vom Zählerstand schalten. Ausgang 2 kann aus der SPS geschaltet werden. Damit können schnelle Steuersignale für Feldgeräte genutzt und geschaltet werden.

Die Betriebsarten werden in TwinCAT über CoE-Objekte eingestellt.

Vor-/Rückwärtszähler

Bei der Betriebsart Vor-/Rückwärtszähler wird der Puls, der gezählt werden soll, über den digitalen Eingang 1 detektiert. Über den digitalen Eingang 2 wird die Zählrichtung vorgegeben.

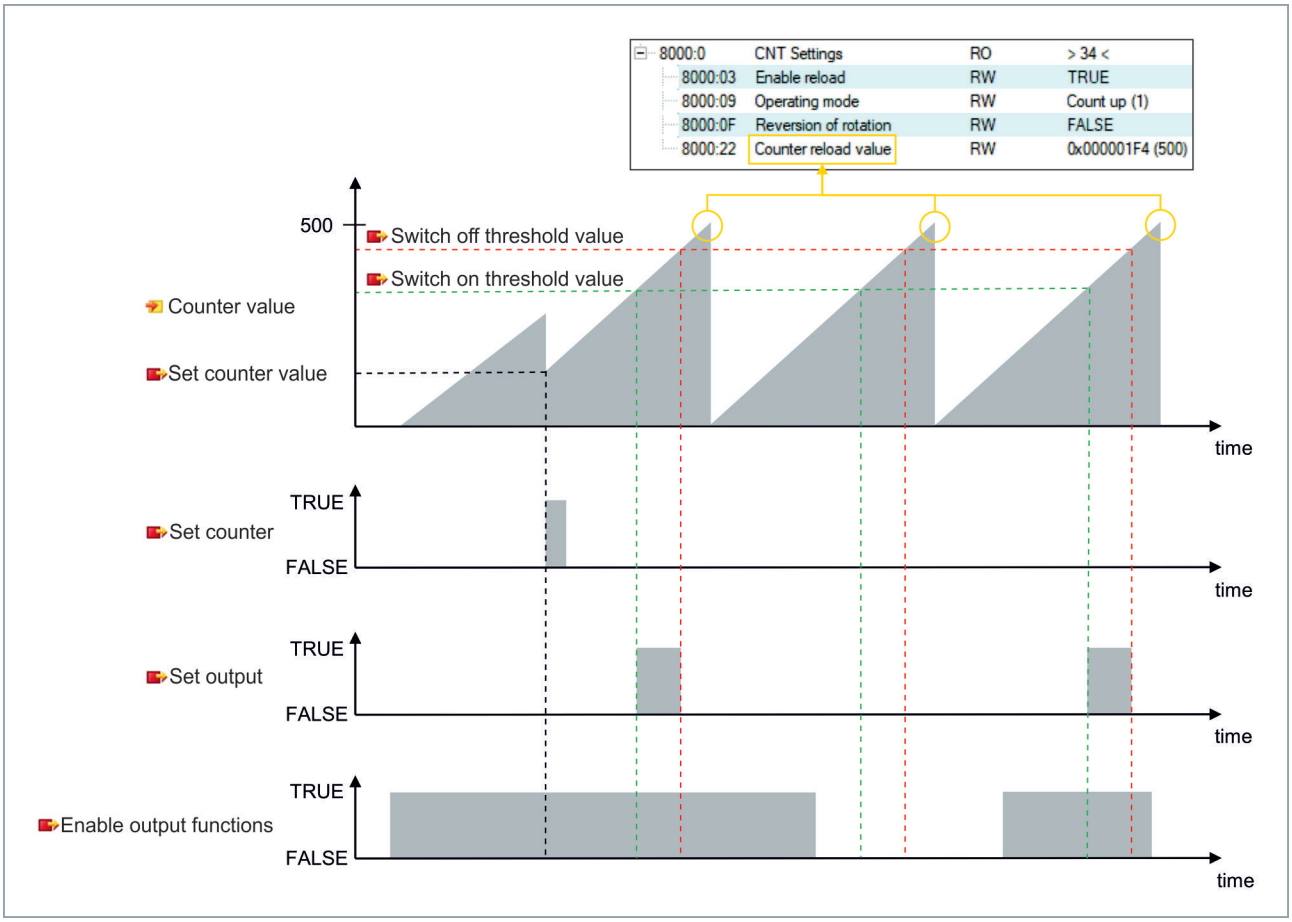
Liegt am Eingang 1 und gleichzeitig am Eingang 2 ein High-Pegel an, wird vorwärts gezählt. Liegt am Eingang 1 ein High-Pegel und am Eingang 2 ein Low-Pegel an, wird rückwärts gezählt.

Vorwärtszähler

Bei dieser Betriebsart wird das Signal am digitalen Eingang 1 detektiert.

Rückwärtszähler

Bei dieser Betriebsart wird das Signal am digitalen Eingang 1 detektiert.



Tab. 10: Technische Daten, Multifunktions-I/Os im Zähler-Modus.

Technische Daten	CX7050
Anzahl der Zähler	1 x Vor-/Rückwärtszähler, 1 x Vor- oder Rückwärtszähler
Nennspannung	24 V DC (-15 %/+20 %)
Spezifikation	EN 61131-2, Typ 3
Signalspannung „0“	-3…+5 V
Signalspannung „1“	11…30 V
Grenzfrequenz	Vor-/Rückwärtszähler: 20 kHz[1)], Zählen nur in eine Richtung: 100 kHz
Zählertiefe	32 Bit
Ausgangsstrom max.	24 V/0,5 A (kurzschlussfest)
Besondere Eigenschaften	Zähler setzen, Ausgänge schalten, Zähler zurücksetzen
Anschlussquerschnitt	e*: 0,08…1,5 mm², f*: 0,25…1,5 mm², a*: 0,14…0,75 mm²
Anschlussquerschnitt AWG	e*: AWG 28…16, f*: AWG 22…16, a*: AWG 26…19
Abisolierlänge	8…9 mm

[1)] Der Vor-/Rückwärtszähler kann auch bis 100 kHz zählen, nur bei einer Richtungsumkehr muss die Zählfrequenz <= 20 kHz sein, sonst gehen Impulse verloren.

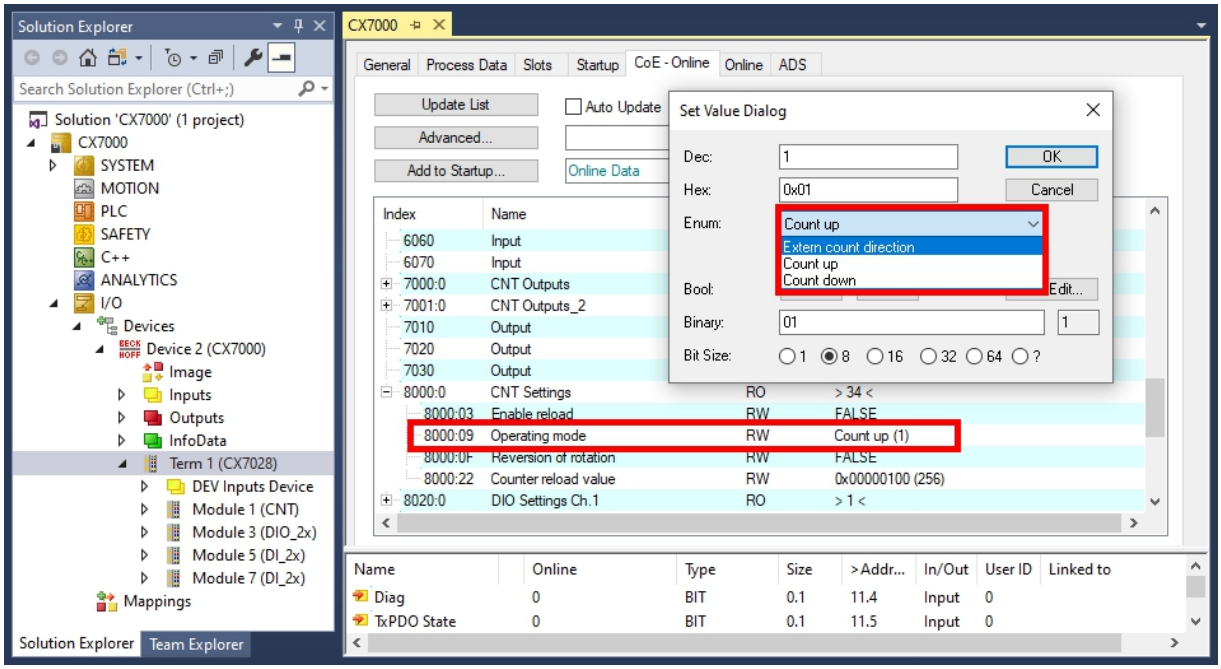
*e: eindrähtig, Draht massiv; f: feindrähtig, Litze; a: mit Aderendhülse

6.3.1 Betriebsart wählen

Der CX7050 unterstützt im Zähler-Modus drei Betriebsarten. Die Betriebsart wird in TwinCAT über CoE-Objekte eingestellt. Sie können zwischen den drei Betriebsarten Vor-/Rückwärtszähler, Vorwärtszähler und Rückwärtszähler wählen.

Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum auf das **CX7028-Device**.
2. Klicken Sie auf die Registerkarte **CoE-Online**.



3. Klicken Sie doppelt auf das CoE-Objekt **8000:09 Operating mode**.
4. Wählen Sie unter der Option **Enum** die erforderliche Betriebsart.

⇨ Die Betriebsart wird übernommen. Beachten Sie, dass Sie beim CX7050 immer nur eine Betriebsart gleichzeitig verwenden können und eine Mischung aus den Betriebsarten nicht möglich ist.

6.3.2 Ausgänge schalten

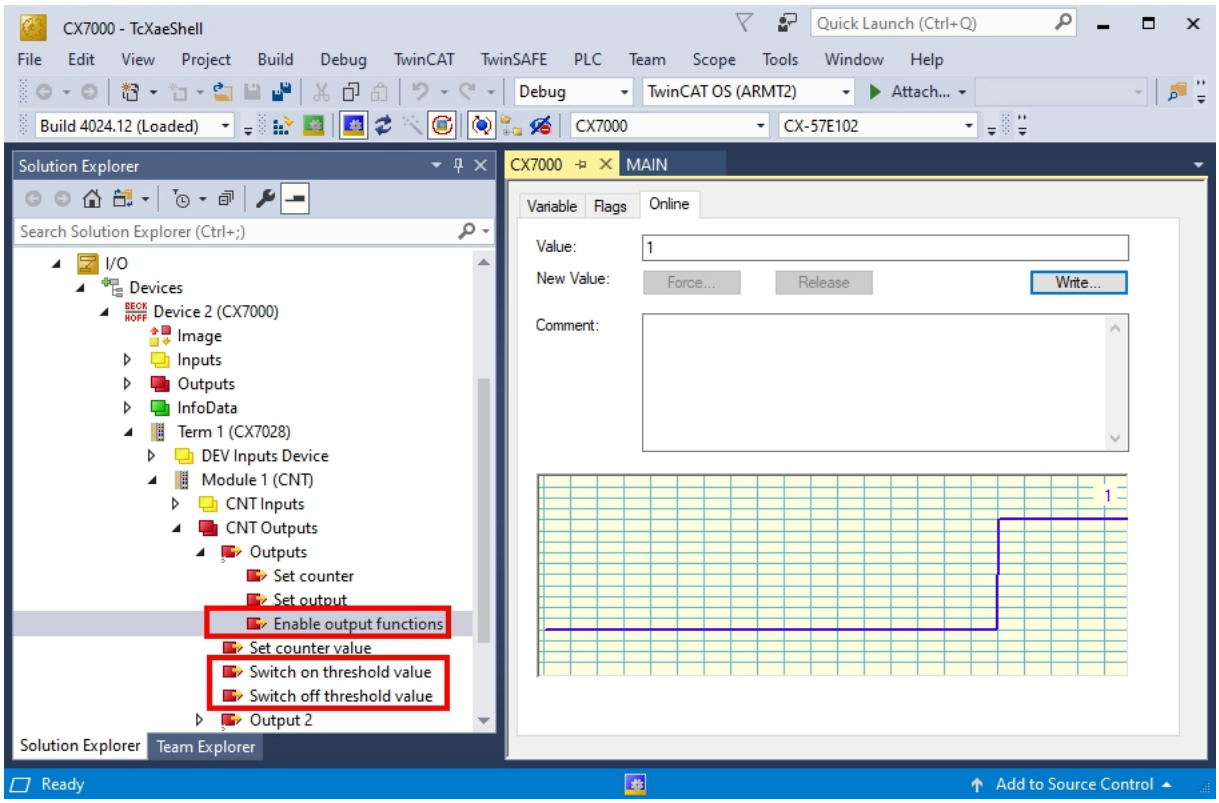
Beim CX7050 ist es möglich den Ausgang 1 selbsttätig zu schalten, sobald ein bestimmter Zählerstand erreicht wird. Das ermöglicht eine schnelle und ohne die SPS laufende Bearbeitung. Ein zweiter Ausgang, der Ausgang 2, kann unabhängig vom Zählerstand über die SPS geschaltet werden.

Über die Variablen **Switch on threshold value** und **Switch off threshold value** wird der Ausgang 1 geschaltet bzw. abgeschaltet:

- Wird der eingestellte Wert unter **Switch on threshold value** erreicht, wird der Ausgang geschaltet.
- Wird der eingestellte Wert unter **Switch off threshold value** erreicht, wird der Ausgang ausgeschaltet.

Beim Rückwärtszählen wird die entsprechende Schaltanweisung umgekehrt ausgeführt. Wird der eingestellte Wert unter **Switch on threshold value** unterschritten, wird der Ausgang 1 abgeschaltet.

Gehen Sie wie folgt vor:



1. Geben Sie über die Variable **Switch on threshold value** einen Zählerstand vor, bei dem der Ausgang geschaltet werden soll.
2. Geben Sie über die Variable **Switch off threshold value** einen Zählerstand vor, bei dem der Ausgang abgeschaltet werden soll.
3. Setzen Sie anschließend die Variable **Enable output functions** auf **True**, damit die Einstellungen übernommen werden.

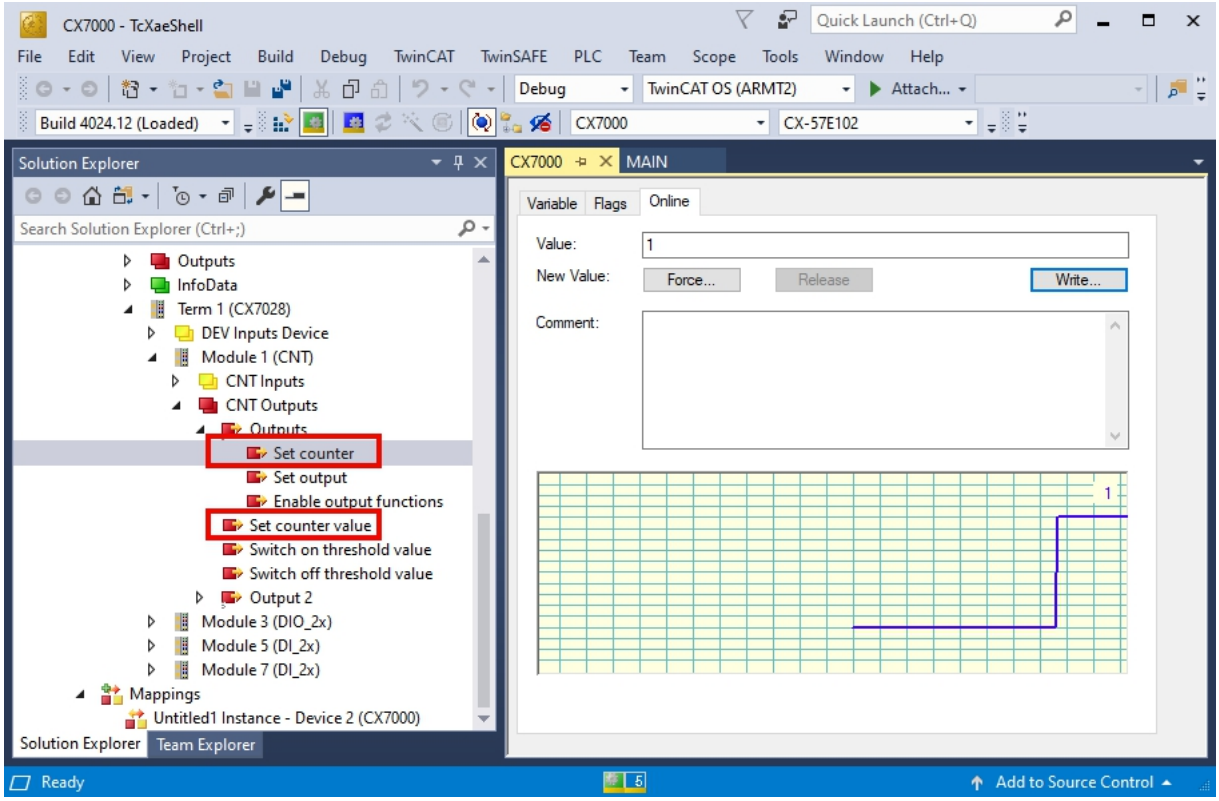
⇨ Erst wenn die Variable **Enable output functions** auf **True** gesetzt wird, wird die Funktion eingeschaltet und der Ausgang geschaltet.

Ist der parametrierte Zählerstand aus **Switch on/off threshold** erreicht bzw. überschritten, aber die Variable **Enable output functions** nicht gesetzt, wird der Schaltauftrag nicht ausgeführt. Sobald dann **Enable output functions** gesetzt wird, wird umgehend der Ausgang geschaltet. Genauso wirkt sich ein nachträglich aktivierter Zählerstand **Switch on/off threshold** bei erfüllter Schaltbedingung umgehend auf den Ausgang aus.

6.3.3 Zählerstand setzen

In diesem Schritt wird gezeigt, wie Sie den Zählerstand auf einen bestimmten Wert setzen können. Über die Variable **Set counter value** wird ein Wert vorgegeben und über die Variable **Set counter** wird der Zählerstand gesetzt. Beide Variablen können aus der SPS gesteuert werden.

Gehen Sie wie folg vor:



1. Geben Sie über die Variable **Set counter value** einen Wert vor, der als Zählerstand gesetzt werden soll.
2. Setzen Sie anschließend die Variable **Set counter** auf **True**, damit die Einstellungen übernommen werden.

⇨ Erst wenn die Variable **Set counter** auf **True** gesetzt wird, wird der eingestellte Wert unter **Set counter value** für den Zählerstand übernommen.

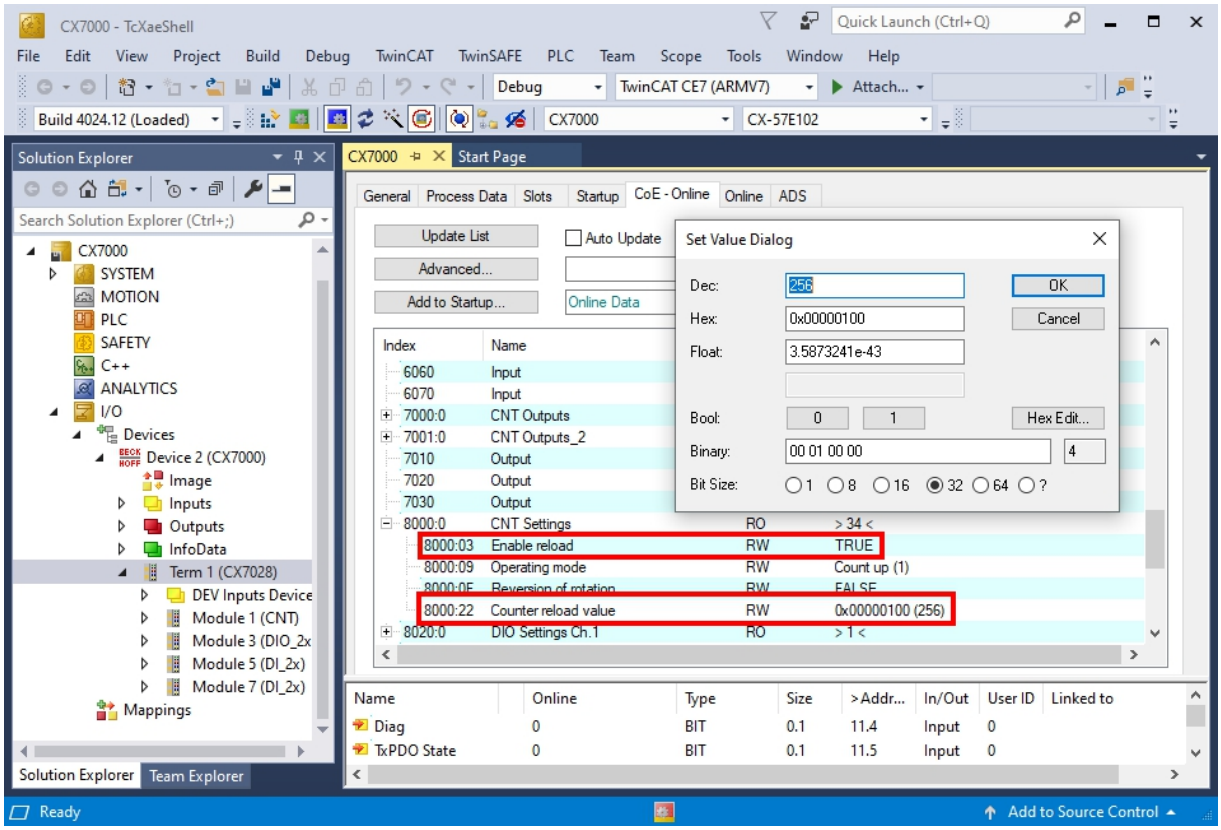
6.3.4 Grenzwert für Zähler festlegen

In diesem Schritt wird gezeigt, wie Sie in TwinCAT einen Grenzwert festlegen, ab dem der Zählerstand automatisch wieder auf null zurückgesetzt wird. Beim Vorwärtszählen wird der Zählerstand beim Erreichen des Grenzwertes auf null zurückgesetzt. Beim Rückwärtszählen wird der Zählerstand beim Erreichen von Null auf den eingestellten Grenzwert zurückgesetzt.

Der Zählerstand ist eine UDINT-Variable. Der Zähler zählt nur im positiven Bereich von 0-0xFFFF_FFFF (4294967295). Beim Unterschreiten von Null, wird der Zähler auf den maximalen positiven Wert gesetzt. Beim Überschreiten von 4294967295 wird der Zähler auf null gesetzt. Die beiden Variablen **Counter underflow** bzw. **Counter overflow**, zeigen den Überlauf an und werden entweder beim erreichten von 0x4000 in positiver Richtung oder 0xFFFFC000 in negativer Richtung zurückgesetzt oder wenn der entsprechend andere Überlauf erreicht wurde.

Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum auf das **CX7028-Device**.
2. Klicken Sie auf die Registerkarte **CoE-Online**.



3. Klicken Sie doppelt auf das CoE-Objekt **8000:22 Counter reload value** und legen Sie den Grenzwert fest.
4. Klicken Sie anschließend doppelt auf das CoE-Objekt **8000:03 Enable reload** und setzen Sie den Wert auf **True**.

⇨ Erst wenn d das CoE-Objekt **8000:03 Enable reload** auf **True** gesetzt ist, ist die Funktion und der definierte Grenzwert aktiv.

6.4 Inkremental-Encoder-Modus

Im Inkremental-Encoder-Modus kann der CX7050 als ein Interface zum direkten Anschluss von 24-V-Inkremental-Encodern konfiguriert werden. Dabei wird eine Vierfachauswertung verwendet und sowohl High-Pegel als auch Low-Pegel an Eingang 1 und Eingang 2 detektiert.

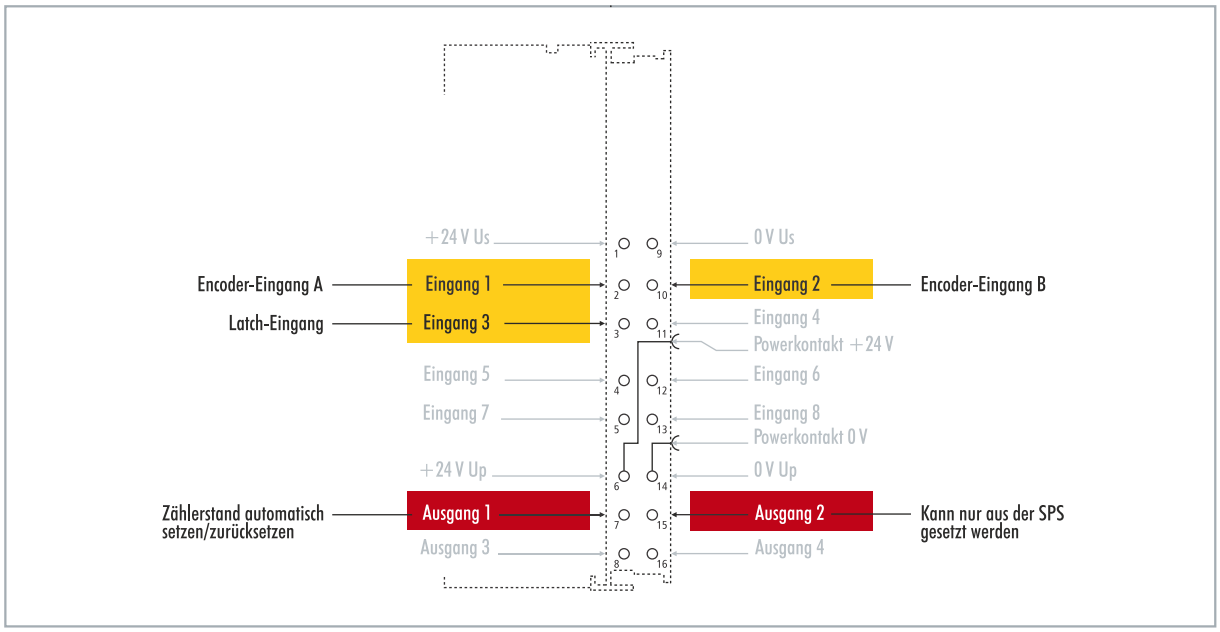
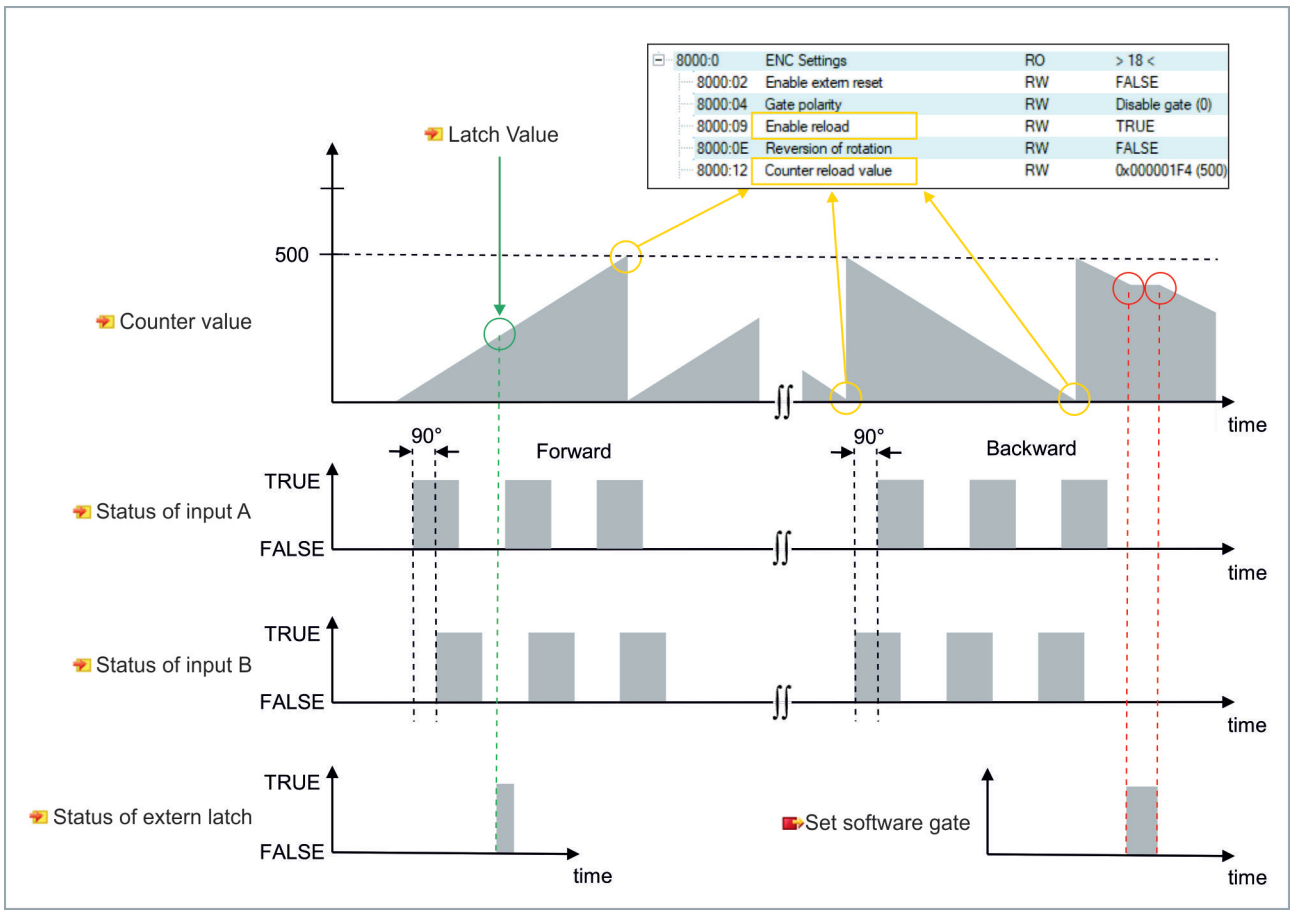


Abb. 33: Konfigurierbare Ein- und Ausgänge im Inkremental-Encoder-Modus.

Der Funktionsumfang im Encoders-Modus entspricht dem Funktionsumfang im Zähler-Modus. Wobei zusätzlich der Zählerstand an Eingang 3 gelatcht werden kann, d. h., dass der Wert bei einem High-Pegel an Eingang 3 in die Prozessdaten eingetragen wird. Alternativ kann der Zähler bei einem High-Pegel an Eingang 3 zurückgesetzt werden.

Zusätzlich lässt sich Ausgang 1 abhängig vom Zählerstand schalten. Ausgang 2 kann aus der SPS geschaltet werden. Damit können schnelle Steuersignale für Feldgeräte genutzt und geschaltet werden.



Tab. 11: Technische Daten, Multifunktions-I/Os im Encoder-Modus.

Technische Daten	CX7050
Technik	Inkremental-Encoder-Interface
Nennspannung	24 V DC (-15 %/+20 %)
Spezifikation	EN 61131-2, Typ 3
Geberanschluss	1 x A, B: 24 V, single-ended
Zusätzliche Eingänge	Latch-Eingang, 24 V DC
Grenzfrequenz	250.000 Inkremente/s (bei 4-fach-Auswertung), entspr. 62,5 kHz
Zählertiefe	32 Bit
Quadraturdecoder	4-fach-Auswertung
Ausgangsstrom max.	24 V/0,5 A (kurzschlussfest)
Besondere Eigenschaften	Latch-Funktion, Software-Gate, Zähler setzen, Ausgänge schalten, Zähler zurücksetzen
Anschlussquerschnitt	e*: 0,08…1,5 mm², f*: 0,25…1,5 mm², a*: 0,14…0,75 mm²
Anschlussquerschnitt AWG	e*: AWG 28…16, f*: AWG 22…16, a*: AWG 26…19
Abisolierlänge	8…9 mm

*e: eindrähtig, Draht massiv; f: feindrähtig, Litze; a: mit Aderendhülse

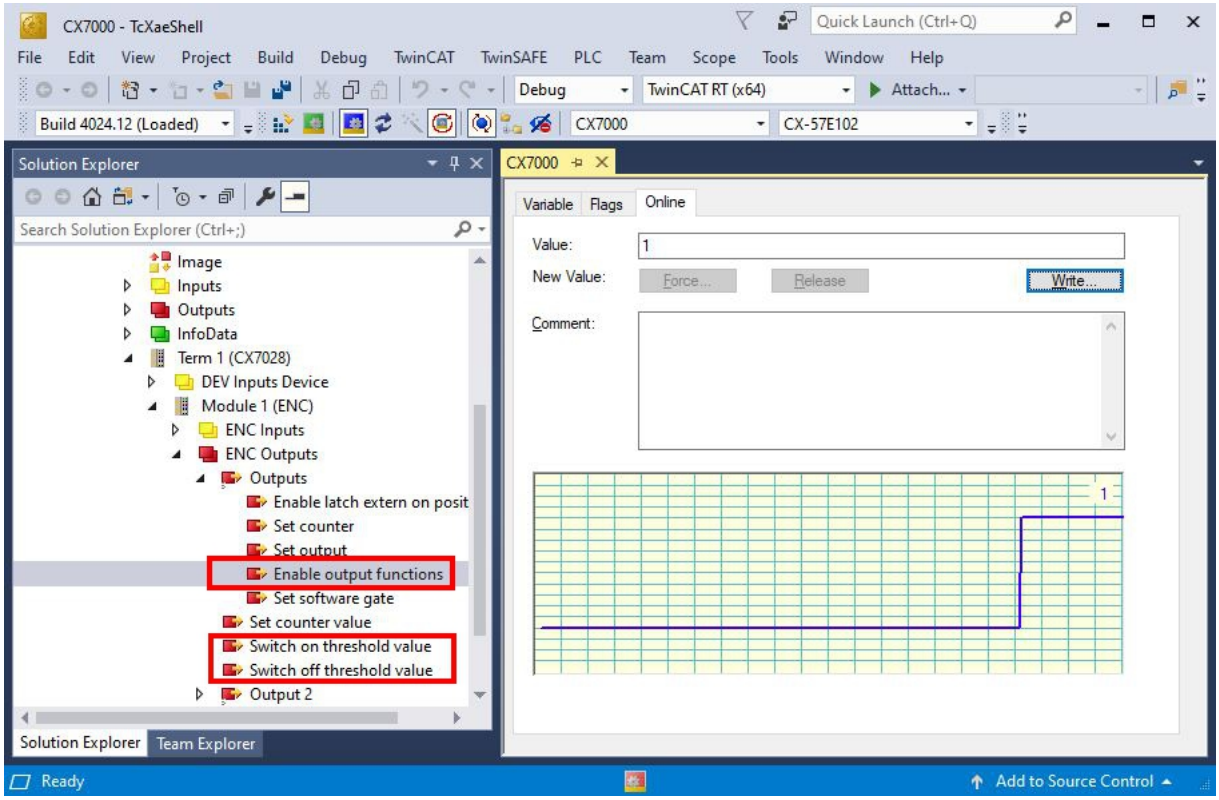
6.4.1 Ausgänge schalten

Beim CX7050 ist es möglich, den Ausgang 1 selbsttätig zu schalten, sobald ein bestimmter Zählerstand erreicht wird. Das ermöglicht eine schnelle und ohne die SPS laufende Bearbeitung. Ein zweiter Ausgang, der Ausgang 2, kann unabhängig vom Zählerstand über die SPS geschaltet werden.

Über die Variablen **Switch on threshold value** und **Switch off threshold value** wird der Ausgang 1 geschaltet bzw. abgeschaltet:

- Wird der eingestellte Wert unter **Switch on threshold value** erreicht, wird der Ausgang geschaltet.
- Wird der eingestellte Wert unter **Switch off threshold value** erreicht, wird der Ausgang ausgeschaltet.

Gehen Sie wie folgt vor:



1. Geben Sie über die Variable **Switch on threshold value** einen Zählerstand vor, bei dem der Ausgang geschaltet werden soll.
2. Geben Sie über die Variable **Switch off threshold value** einen Zählerstand vor, bei dem der Ausgang abgeschaltet werden soll.
3. Setzen Sie anschließend die Variable **Enable output functions**, damit die Einstellungen übernommen werden.

⇨ Erst wenn die Variable **Enable output functions** auf **True** gesetzt wird, wird die Funktion eingeschaltet und die Einstellungen übernommen.

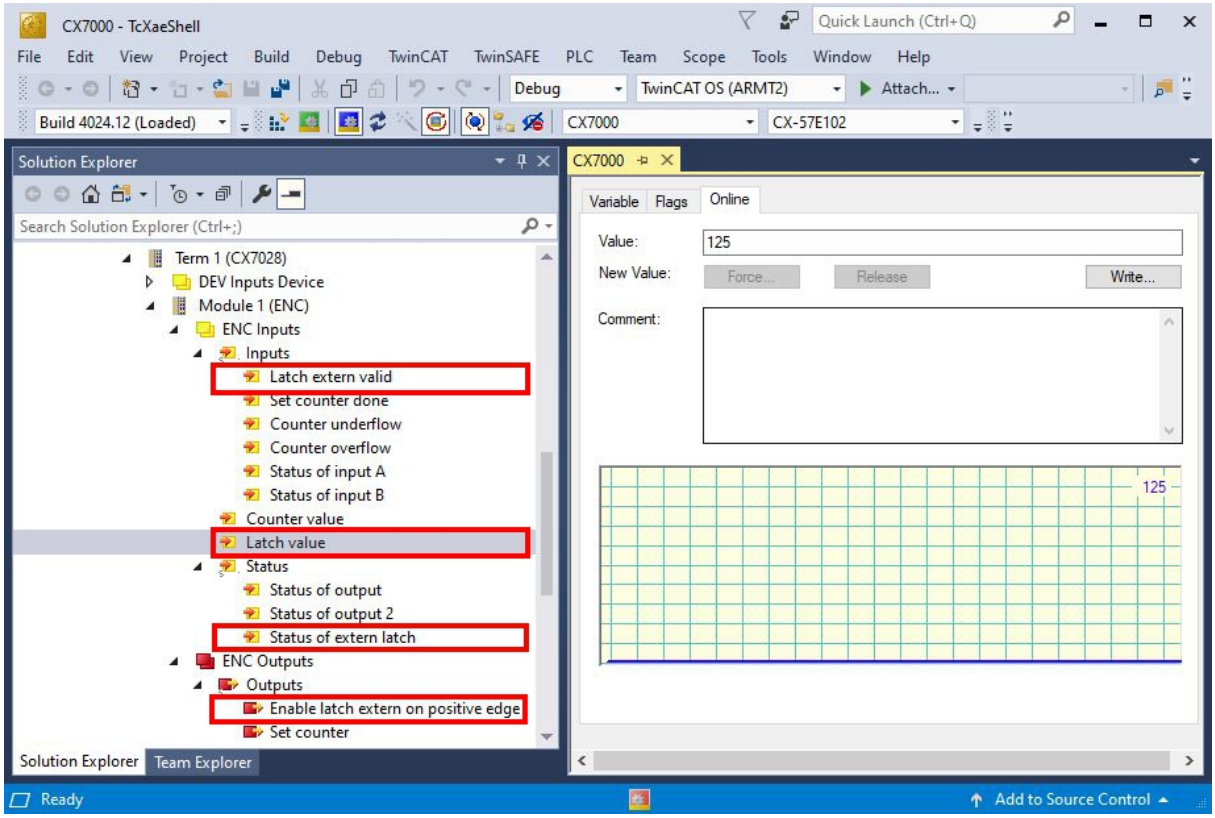
Ist der parametrierte Zählerstand aus **Switch on/off threshold** erreicht bzw. überschritten, aber die Variable **Enable output functions** nicht gesetzt, wird der Schaltauftrag nicht ausgeführt. Sobald dann **Enable output functions** gesetzt wird, wird umgehend der Ausgang geschaltet. Genauso wirkt sich ein nachträglich aktivierter Zählerstand **Switch on/off threshold** bei erfüllter Schaltbedingung umgehend auf den Ausgang aus.

6.4.2 Zählerstand latchen

Im Incremental-Encoder-Modus kann der Zählerstand gelatcht und dadurch der aktuelle Wert in die Prozessdaten eingetragen werden. Als Latch-Eingang wird Eingang 3 verwendet.

Um die Funktion zu aktivieren, muss die Variable **Enable latch extern on positive edge** auf **True** gesetzt sein. Bei einem High-Pegel an Eingang 3 wird der aktuelle Zählerstand in die Variable **Latch Value** eingetragen. Sie können die Gültigkeit der Variable überwachen. Sobald der Latch-Wert eingetragen wird, wird auch die Variable **Latch extern valid** auf **True** gesetzt.

Gehen Sie wie folgt vor:



1. Setzen Sie die Variable **Enable latch extern on positive edge** auf **True**, um die Latch-Funktion zu aktivieren.
2. Überwachen Sie den Status des Latch-Eingangs mit der Variable **Status of extern latch**.
3. Bei einem High-Pegel an Eingang 3 wird der aktuelle Zählerstand in die Variable **Latch Value** eingetragen.
4. Überwachen Sie die Gültigkeit des Latch-Wertes über die Variable **Latch extern valid**. Sobald der Latch-Wert geschrieben wird, wird auch die Variable auf **True** gesetzt.

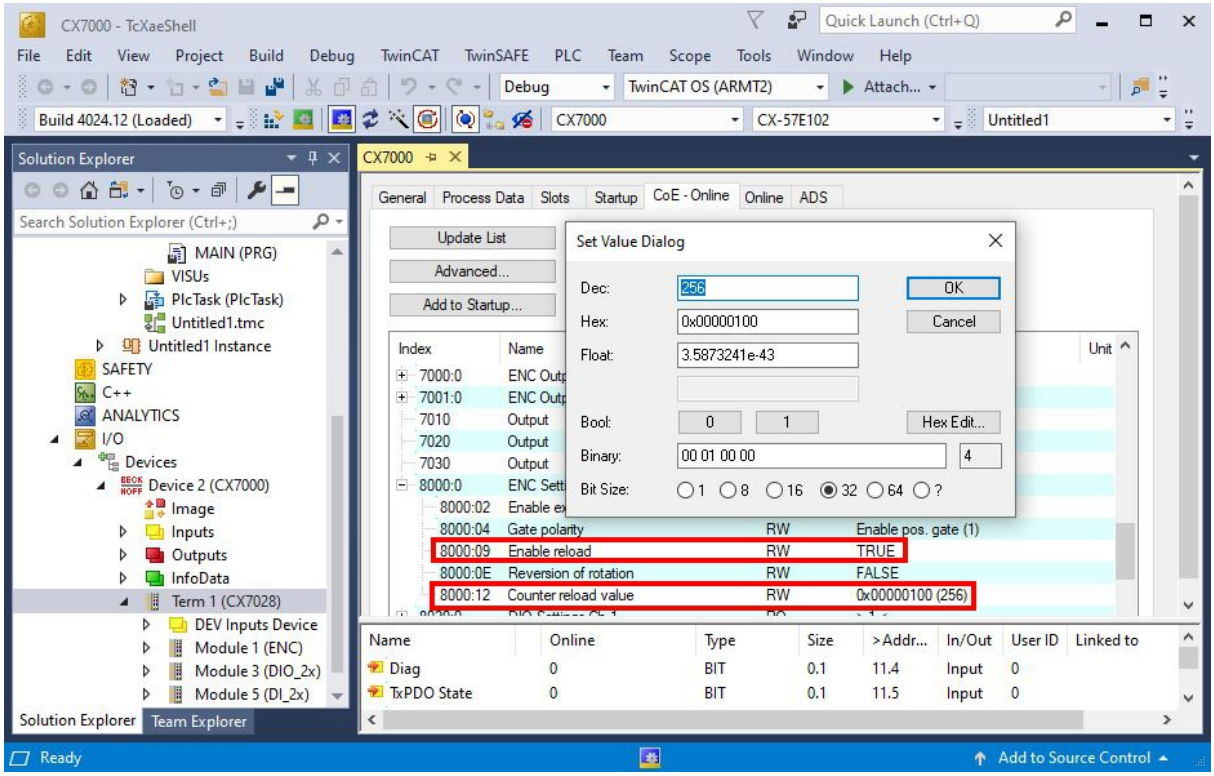
⇨ Um erneut einen Latch auszuführen, muss die Variable **Enable latch extern on positive edge** wieder einen High-Pegel bekommen.

6.4.3 Grenzwert für Zähler festlegen

In diesem Schritt wird gezeigt, wie Sie in TwinCAT einen Grenzwert festlegen können, ab dem der Zählerstand automatisch wieder auf null zurückgesetzt wird. Beim Vorwärtszählen wird der Zählerstand beim Erreichen des Grenzwertes auf null zurückgesetzt. Beim Rückwärtszählen wird der Zählerstand beim Erreichen von Null auf den eingestellten Grenzwert zurückgesetzt.

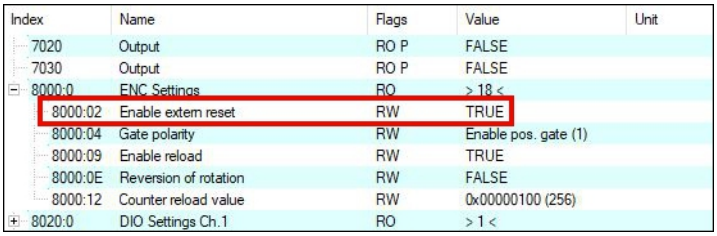
Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum auf das **CX7028-Device**.
2. Klicken Sie auf die Registerkarte **CoE-Online**.



3. Klicken Sie doppelt auf das CoE-Objekt **8000:12 Counter reload value** und legen Sie den Grenzwert fest.
4. Klicken Sie anschließend doppelt auf das CoE-Objekt **8000:09 Enable reload** und setzen Sie den Wert auf **True**.

⇨ Erst wenn **Enable reload** gesetzt ist, ist die Funktion aktiv. Alternativ kann der Latch-Eingang verwendet und damit der Zählerstand extern zurückgesetzt werden. Dafür muss die Latch-Funktion deaktiviert und das CoE-Objekt **Enable extern reset** auf **True** gesetzt werden. Mit dieser Einstellung wird bei einem High-Pegel an Eingang 3 der aktuelle Zählerstand auf null gesetzt.



Index	Name	Flags	Value	Unit
7020	Output	RO P	FALSE	
7030	Output	RO P	FALSE	
8000:0	ENC Settings	RO	> 18 <	
8000:02	Enable extern reset	RW	TRUE	
8000:04	Gate polarity	RW	Enable pos. gate (1)	
8000:09	Enable reload	RW	TRUE	
8000:0E	Reversion of rotation	RW	FALSE	
8000:12	Counter reload value	RW	0x00000100 (256)	
8020:0	DIO Settings Ch.1	RO	> 1 <	

6.5 Analog-Signal-Modus

Die Single-ended-Eingänge 7 und 8 erfassen Signale im Bereich von 0 bis 10 V.

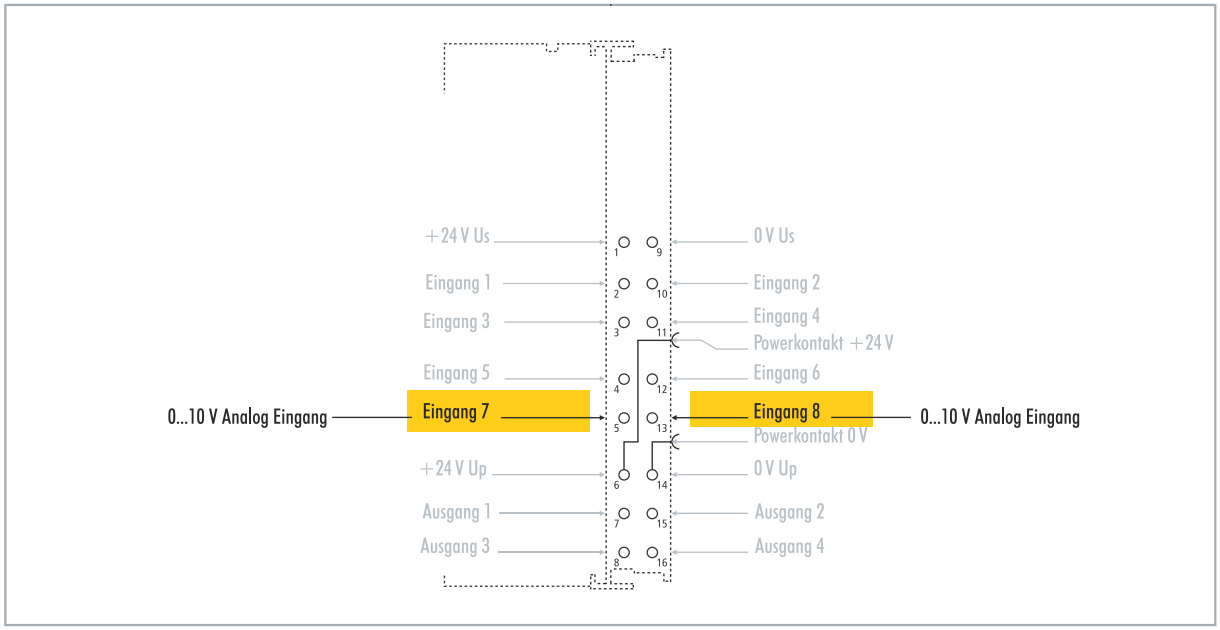


Abb. 34: Konfigurierbare analoge Eingänge.

Die Spannung wird mit einer Auflösung von 12 Bit digitalisiert. Der Signalzustand wird durch Leuchtdioden angezeigt.

Tab. 12: Technische Daten, Multifunktions-I/Os im Analog-Modus.

Technische Daten	CX7050
Technik	single-ended
Anzahl Eingänge	2
Signalspannung	0…10 V
Innenwiderstand	500 kΩ
Grenzfrequenz Eingangsfilter	2 kHz
Auflösung	12 Bit (16-Bit-Darstellung)
Messfehler	< ±0,3 % (bezogen auf den Messbereichsendwert)
Anschlussquerschnitt	e*: 0,08…1,5 mm², f*: 0,25…1,5 mm², a*: 0,14…0,75 mm²
Anschlussquerschnitt AWG	e*: AWG 28…16, f*: AWG 22…16, a*: AWG 26…19
Abisolierlänge	8…9 mm

*e: eindrähtig, Draht massiv; f: feindrähtig, Litze; a: mit Aderendhülse

6.6 PWM-Signal-Modus

HINWEIS

Rückspeisung bei den 24-V-Ausgängen

Eine Spannung von 24 V an den Ausgängen 3 und 4 kann das Gerät zerstören (Rückspeisung). Im PWM-Modus darf keine Spannung an den Ausgängen angelegt werden.

Der PWM-Signal-Modus ermöglicht auf Ausgang 3 und Ausgang 4 ein binäres Signal pulsweitenmoduliert auszugeben.

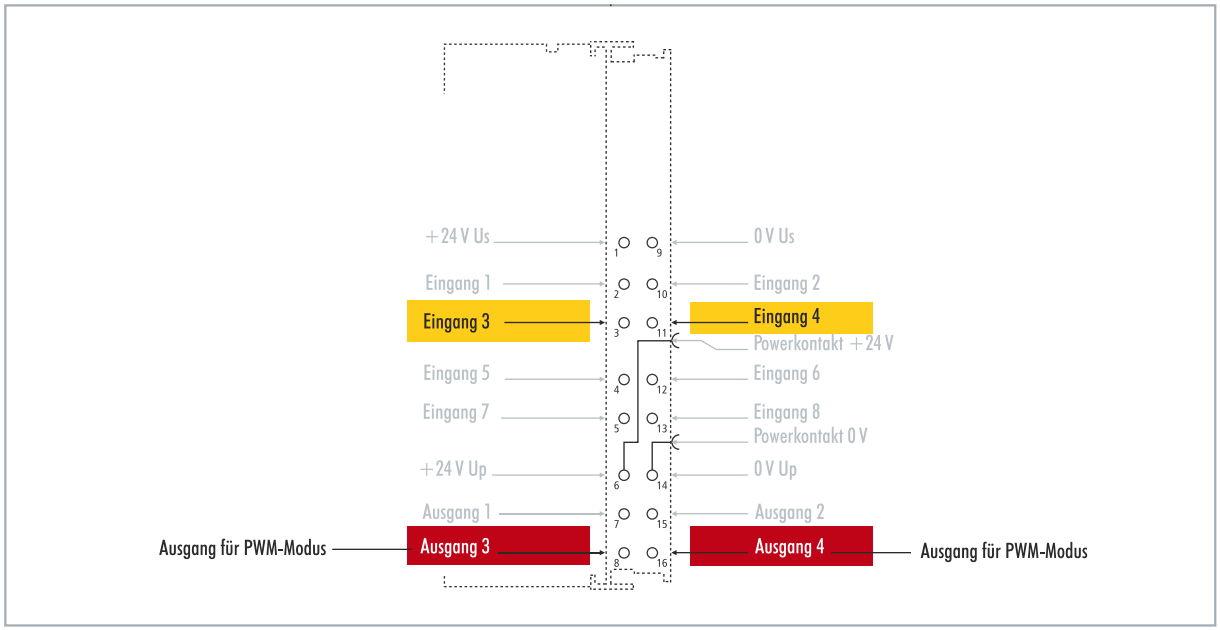
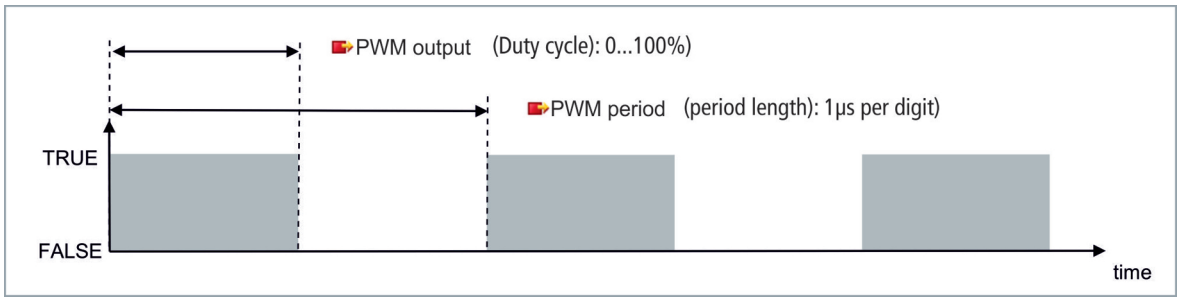


Abb. 35: Konfigurierbare Ein- und Ausgänge im PWM-Signal-Modus

Dieses Signal ist getrennt in Tastverhältnis (0…100%) und PWM-Taktfrequenz (15 Hz…100 kHz). Die LEDs sind mit den Ausgängen getaktet und zeigen durch ihre Helligkeit das Tastverhältnis an. Die Signalwerte werden in 16-Bit-Werten übergeben.



Tab. 13: Technische Daten, Multifunktions-I/Os im PWM-Modus.

Technische Daten	Digitale Eingänge
Anschlusstechnik	PWM-Ausgang
Anzahl Ausgänge	2
Nennspannung	24 V DC (-15 %/+20 %)
Lastart	ohmsch, induktiv, Lampenlast
Ausgangsstrom max.	24 V/0,5 A (kurzschlussfest)
PWM-Taktfrequenz	15 Hz…100 kHz
Tastverhältnis	0…100 % (T_{ON} > 20 ns, T_{OFF} > 200 ns)
Kurzschlussstrom	< 2 A typ.
Besondere Eigenschaften	separate Frequenz für jeden Kanal einstellbar

Technische Daten	Digitale Eingänge
Anschlussquerschnitt	e*: 0,08…1,5 mm², f*: 0,25…1,5 mm², a*: 0,14…0,75 mm²
Anschlussquerschnitt AWG	e*: AWG 28…16, f*: AWG 22…16, a*: AWG 26…19
Abisolierlänge	8…9 mm

*e: eindrähtig, Draht massiv; f: feindrähtig, Litze; a: mit Aderendhülse

6.6.1 PWM-Taktfrequenz und Tastverhältnis festlegen

Die Signale auf Ausgang 3 und Ausgang 4 werden pulsweitenmoduliert ausgeben, wobei die Signale in Tastverhältnis und PWM-Taktfrequenz getrennt sind. Für beide Ausgänge lassen sich eigene Werte für Tastverhältnis und PWM-Taktfrequenz definieren.

Tab. 14: PWM output (Tastverhältnis), Darstellung des PWM-Signals im Auslieferungszustand.

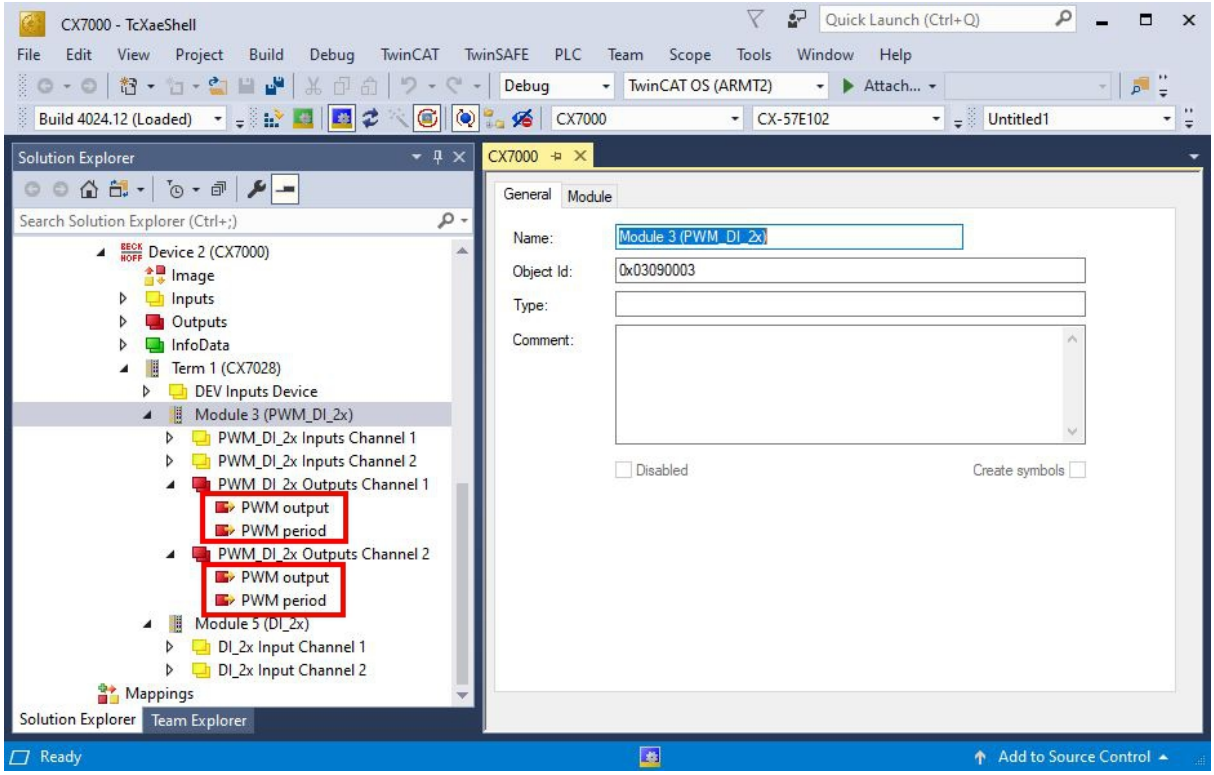
Wert	Dezimal	Hexadezimal
0 %	0	0x0000
25 %	16383	0x3FFF
50 %	32767	0x7FFF
100 %	65.535	0xFFFF

Tab. 15: PWM period (PWM-Taktfrequenz), Darstellung des PWM-Signals im Auslieferungszustand.

Wert	Dezimal	Hexadezimal	Frequenz
0,010 ms	0..10	0x0000-0x000A	100 kHz
0,011 ms	11	0x000B	90,909 kHz
0,100 ms	100	0x0064	10 kHz
1,000 ms	1000	0x03E8	1 kHz
16,38 ms	16383	0x3FFF	61,04 Hz
65,53 ms	65535	0xFFFF	15,26 Hz

Dabei entsprechen die Variablen **PWM output** dem Tastverhältnis und **PWM period** der PWM-Taktfrequenz, mit der das Signals ausgegeben wird.

Gehen Sie wie folgt vor:



1. Wählen Sie links im Strukturbaum einen Ausgang, für den Sie das Tastverhältnis und PWM-Taktfrequenz festlegen möchten.
2. Verknüpfen Sie die Variablen **PWM output** und **PWM period** mit den passenden Variablen aus Ihrem SPS-Projekt.
3. Legen Sie in den Variablen die Werte für Tastverhältnis und PWM-Taktfrequenz entsprechend der oben genannten Tabellen fest.

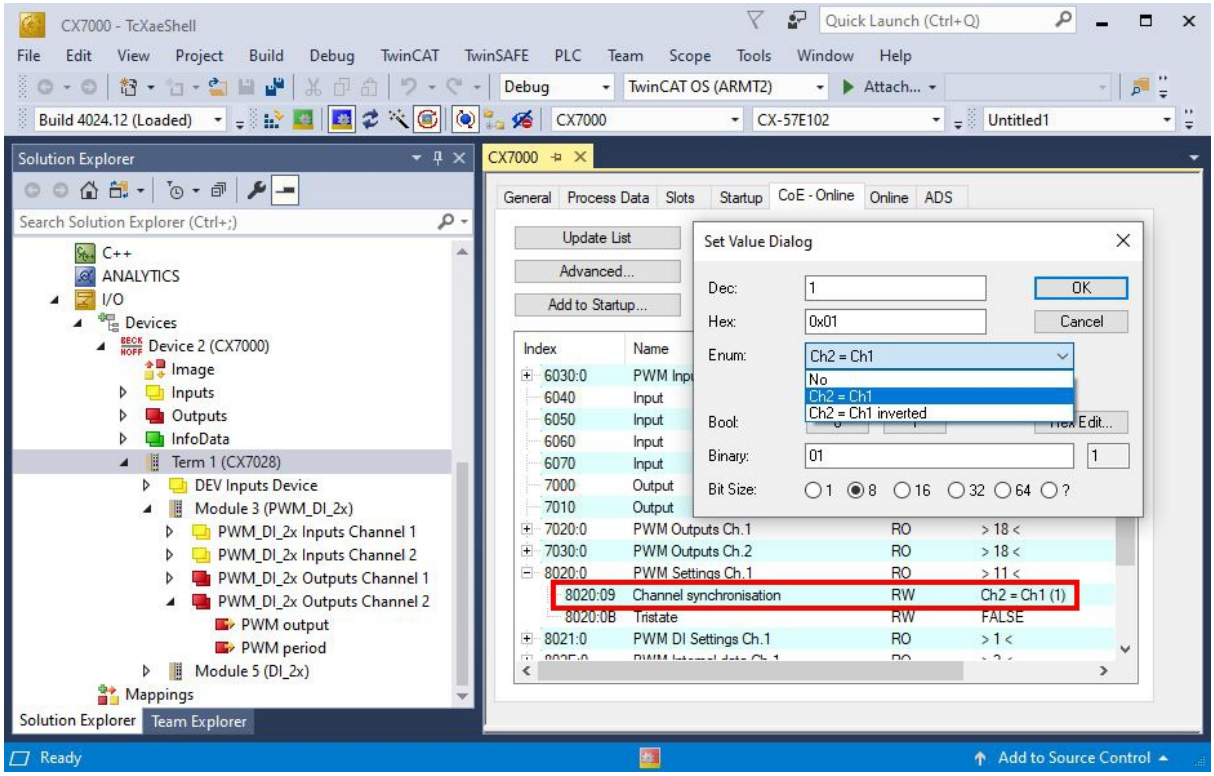
6.6.2 Kanalsynchronisation einstellen

Die Option Kanalsynchronisierung macht die Ausgabe von Ausgang 2 abhängig von Ausgang 1. In den CoE-Objekten stehen folgende Werte zur Verfügung:

- No: keine Abhängigkeit
- Ch2 = Ch1: Tastverhältnis und PWM-Taktfrequenz von Ausgang 1 werden auch auf Ausgang 2 angewendet. Die Phasenlage beträgt 0, d. h. dass steigende und fallende Flanken von Ausgang 1 und Ausgang 2 synchronisiert sind.
- Ch2 = Ch1 inverted: Tastverhältnis und PWM-Taktfrequenz von Ausgang 1 werden auch auf Ausgang 2 angewendet. Die PWM-Taktfrequenz wird allerdings invertiert. Die Phasenlage beträgt 0, d.h. dass eine steigende Flanke bei Ausgang 1 zugleich eine fallende Flanke an Ausgang 2 auslöst.

Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum auf das **CX7028-Device**.
2. Klicken Sie auf die Registerkarte **CoE-Online**.



3. Klicken Sie doppelt auf das CoE-Objekt **8020:09 Channel synchronisation**.
4. Wählen Sie unter der Option **Enum** die erforderliche Synchronisationsart.

7 Konfiguration

7.1 Beckhoff Device Manager starten

Mit dem Beckhoff Device Manager kann ein Industrie-PC per Fernzugriff mit Hilfe eines Webbrowsers konfiguriert werden. Der Zugriff erfolgt über das HTTP-Protokoll und Port 80 (TCP).

Voraussetzungen:

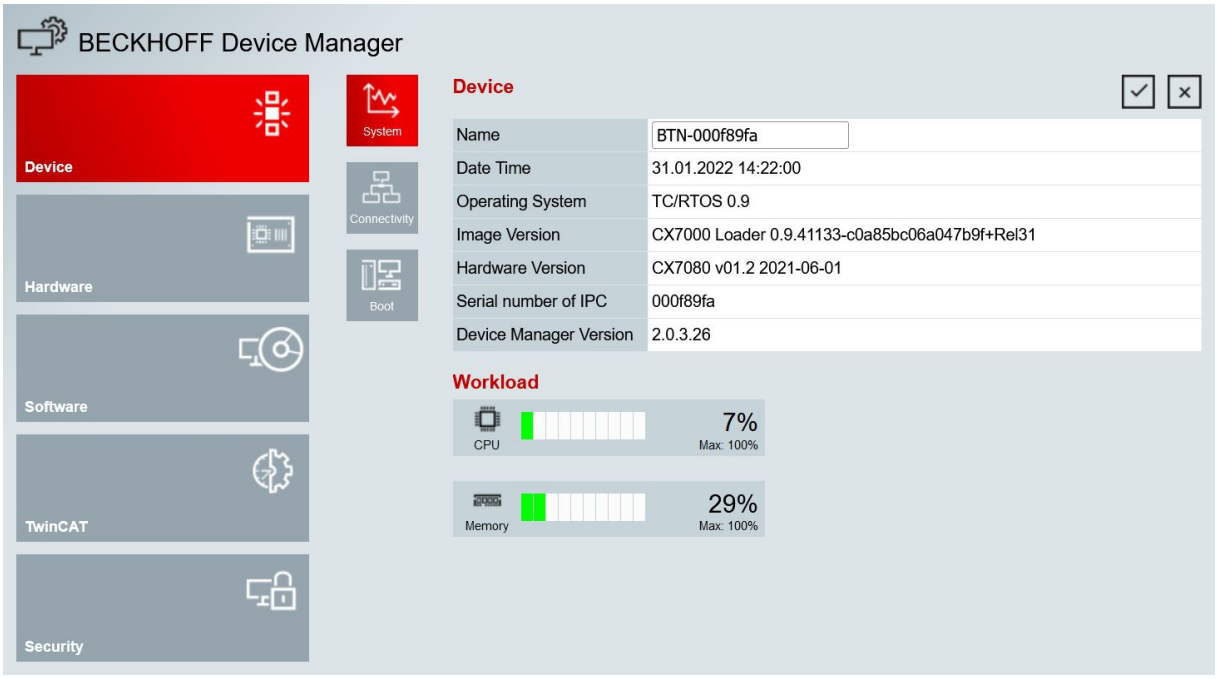
- Host-PC und Embedded-PC müssen sich im gleichen Netzwerk befinden. Die Netzwerkfirewall muss den Zugriff über Port 80 (HTTP) zulassen.
- IP-Adresse oder Hostname des Embedded-PCs.

Tab. 16: Zugangsdaten zum Beckhoff Device Manager bei Auslieferung.

Benutzername	Passwort
Administrator	1

Starten Sie den Beckhoff Device Manager wie folgt:

1. Öffnen Sie einen Webbrowser auf dem Host-PC.
2. Geben Sie die IP-Adresse oder den Hostnamen des Industrie-PCs im Webbrowser ein, um den Beckhoff Device Manager zu starten.
   - Beispiel mit IP-Adresse: http://169.254.136.237/config
   - Beispiel mit Hostnamen: http://BTN-000f89fa/config
3. Geben Sie den Benutzernamen und das Passwort ein. Die Startseite erscheint:



⇨ Navigieren Sie weiter im Menü und konfigurieren Sie den Industrie-PC. Beachten Sie, dass Änderungen erst nach einer Bestätigung wirksam werden. Gegebenenfalls muss der Industrie-PC neu gestartet werden.

7.2 Persistente Daten

HINWEIS

Applikationsbeispiel

In dem folgenden Beispiel kann das Verändern der Verbraucher, der Spannungsversorgung oder nur das Altern der Bauteile dazu führen, dass die Applikation nicht mehr die gewünschte Funktion erfüllt. Beckhoff übernimmt keine Verantwortung für die Umsetzung des Beispiels in einer Applikation.

Normalerweise werden persistente Daten nur beim TwinCAT-Stopp oder durch einen Baustein gespeichert. In diesem Kapitel wird gezeigt, wie Sie persistente Daten ohne USV auf einem CX7050 speichern können.

Bei einem Embedded-PC mit USV ist der Baustein in der Regel mit der USV verknüpft. Der Baustein wird aktiv, sobald ein Spannungsausfall erkannt wird, schreibt die persistenten Daten und fährt dann den Embedded-PC herunter. Bei einer 1-Sekunden-USV entfällt das Herunterfahren des Embedded-PCs, da dafür zu wenig Zeit bleibt.

Bei einer Kleinsteuerung wie dem CX7050, der ohne 1-Sekunden-USV ausgeliefert wird, kann man diese Funktionalität trotzdem nutzen. Es muss lediglich ein Netzteil verwenden werden, welches genug Restenergie hat, um den CX7050 eine gewisse Zeit mit dieser Restenergie mit Spannung zu versorgen. Ein kleiner Test kann ihnen zeigen, ob das mit ihrem Netzteil möglich ist:

Netzteil testen

Wenn der CX7050 läuft, schalten Sie die AC-Spannung ihres Netzteils ab und messen Sie, wie lange der CX7050 noch weiterläuft. Wenn es mehr als drei Sekunden sind, können Sie das Netzteil unter Umständen als Ersatz für eine 1-Sekunden-USV verwenden. Beachten Sie, dass auch Netzteile altern und Kapazität verlieren. Sie sollten daher einen Sicherheitsfaktor einrechnen, beispielsweise einen Faktor von drei, sodass sie genug Reserve haben, um über eine längere Zeit das Netzteil als Ersatz für eine 1-Sekunden-USV betreiben zu können.

Bestimmen Sie nun, wie lange das Netzteil die Spannungsversorgung aufrechterhält. Dafür brauchen Sie eine EL1722, die Sie an die AC-Seite des Netzteils anschließen. Dann schreiben Sie ein kleines Programm:

```
VAR
    bPower230V AT %I* : BOOL; (*link to the EL1722*)
END_VAR

VAR RETAIN
    Counter : INT;
END_VAR

Program:
IF NOT bPower230V THEN (*bPower230V is linked to the EL1722*)
    Counter:=counter+1; (*the counter is a retain value*)
END_IF
```

Erzeugen Sie ein Bootprojekt und schalten Sie die AC-Spannung des Netzteils ab. Sobald die EL1722 keinen Wert mehr anzeigt, wird der Counter hochgezählt und die Daten in das interne NOVRAM kopiert. Schalten Sie die AC-Spannung wieder ein und loggen Sie sich ein. Den Wert, den der Counter hat, müssen Sie nun mit der Taskzeit multiplizieren. Wiederholen Sie das ein paar Mal, damit Sie sicher sind, dass sich das Netzteil immer gleich verhält. Als Nächstes müssen Sie den Baustein `FB_WritePersistentData` einfügen. Dieser ist in der Tc2_Utilities Library enthalten (unter dem Ordner „TwinCAT PLC“).

Bestimmen Sie anschließend, wie lange das Speichern der persistenten Daten dauert. Wiederholen Sie auch diesen Vorgang ein paar Mal, damit Sie einen konstanten Wert erhalten und bei Schwankungen einen Maximalwert bestimmen können. Die benötigte Zeit können Sie über das Busy-Flag bestimmen. Der Baustein wird bearbeitet, solange das Busy-Flag gesetzt ist. Den ermittelten Wert multiplizieren Sie noch mit zwei, um einen weiteren Sicherheitsfaktor einzubauen.

Beispiel:

Sie messen, dass das Netzteil drei Sekunden die Spannungsversorgung aufrechterhält und die persistenten Daten in ca. 400 ms geschrieben werden. Mit den empfohlenen Sicherheitsfaktoren hält das Netzteil eine Sekunde und die persistenten Daten werden in ca. 800 ms geschrieben.

Die Spannungsversorgung wird also für einen längeren Zeitraum aufrechterhalten, als für das Speichern der persistenten Daten benötigt wird. Damit können Sie das Beispiel-Netzteil als Ersatz für die 1-Skunden-USV nutzen.

7.3 NOVRAM

Das NOVRAM kann dazu verwendet werden, um wichtige Variablenwerte, wie z.B. Betriebsdaten oder Zählerstände bei einem Spannungsausfall sicher zu speichern. Die Speichergröße des NOVRAMs ist beschränkt und eignet sich für kleinere Datenmengen bis zu einer Größe von maximal 4 kB.

In diesem Kapitel wird gezeigt, wie das NOVRAM unter in TwinCAT 3 verwendet wird.

Funktionsweise

Das NOVRAM (Non-Volatile Random Access Memory) ist ein spezieller Speicherbaustein der dazu verwendet wird, um wichtige Daten sicher zu speichern. Das NOVRAM besteht aus zwei Bereichen. Einem volatilen Speicher und einem non-volatilen Speicher.

TwinCAT schreibt nur in den volatilen Speicher des NOVRAMs. Bei einem Spannungsausfall werden die Daten automatisch aus dem volatilen Speicher in den non-volatilen Speicher kopiert. Die dafür notwendige Energie liefert ein Kondensator. Sobald die Spannungsversorgung wieder vorhanden ist, werden die Daten automatisch in den volatilen Speicher zurückkopiert und können in TwinCAT weiterverwendet werden.

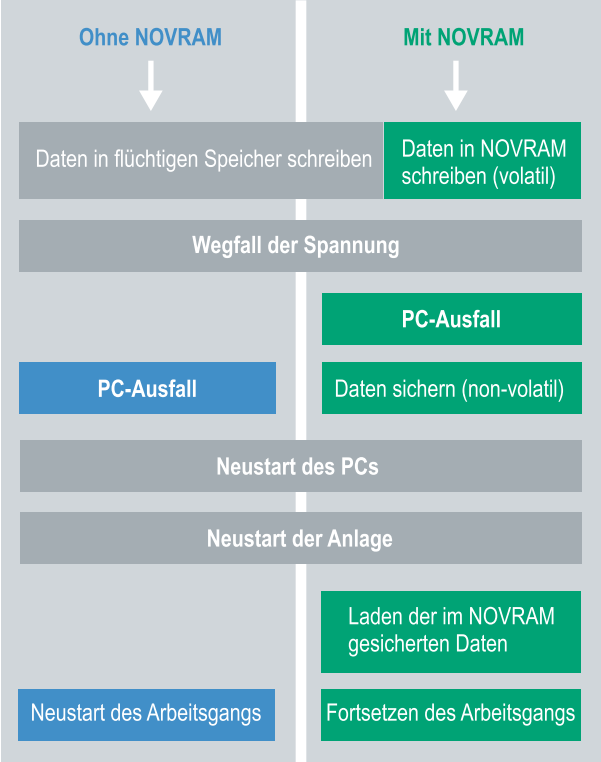


Abb. 36: Verhalten der Steuerung ohne und mit NOVRAM.

Speichergröße

Das NOVRAM hat eine Speichergröße von 4 kB. Die Daten werden zyklisch und wechselweise nach dem Doppelpufferprinzip gespeichert, um damit das Risiko von Dateninkonsistenz zu vermeiden.

Voraussetzungen

Entwicklungsumgebung	Zielplattformen	Hardware	Einzubindende SPS-Bibliotheken
TwinCAT 3.1 Build: 4020	PC oder CX (x86, x64, ARM)	CX70xx, CX9020, CX20x0, CX20x2, CX20x3	Tc2_IoFunctions

7.3.1 Retain-Handler anlegen

Unter TwinCAT 3 ab Build 4020 wird ein Delta-Algorithmus benutzt, um Daten im NOVRAM zu speichern. Der Algorithmus speichert nicht alle Variablen auf einmal, sondern sucht nach Änderungen (Deltas) im Vergleich zum letzten Zyklus und speichert nur veränderte Variablen im NOVRAM.

Um den Delta-Algorithmus zu nutzen, muss in TwinCAT 3 ein Retain-Handler angelegt werden und die relevanten Variablen in der SPS mit dem Schlüsselwort VAR_RETAIN deklariert werden.

Neu an dieser Methode ist, dass keine Funktionsbausteine benutzt werden müssen. Der Retain Handler speichert Daten in das NOVRAM und stellt sie nach einem Spannungsausfall wieder bereit.

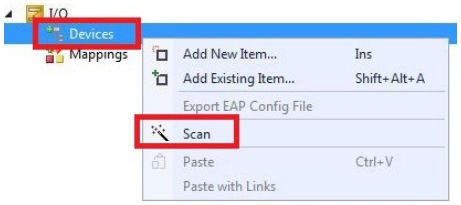
In diesem Kapitel wird beschrieben, wie Sie den Retain-Handler in TwinCAT 3 anlegen. Der Retain Handler speichert Daten in das NOVRAM und stellt sie wieder bereit. Dadurch bleiben wichtige Variablenwerte, wie z.B. Betriebsdaten oder Zählerstände, auch nach einem Neustart oder Spannungsausfall erhalten.

Voraussetzungen für diesen Arbeitsschritt:

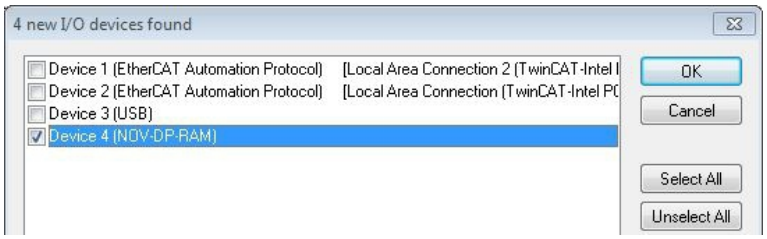
- TwinCAT 3.1 Build: 4020.
- Ein in TwinCAT ausgewähltes Zielgerät.

Legen Sie den Retain-Handler wie folgt an:

1. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf **Devices**.
2. Klicken Sie im Kontextmenü auf **Scan**.

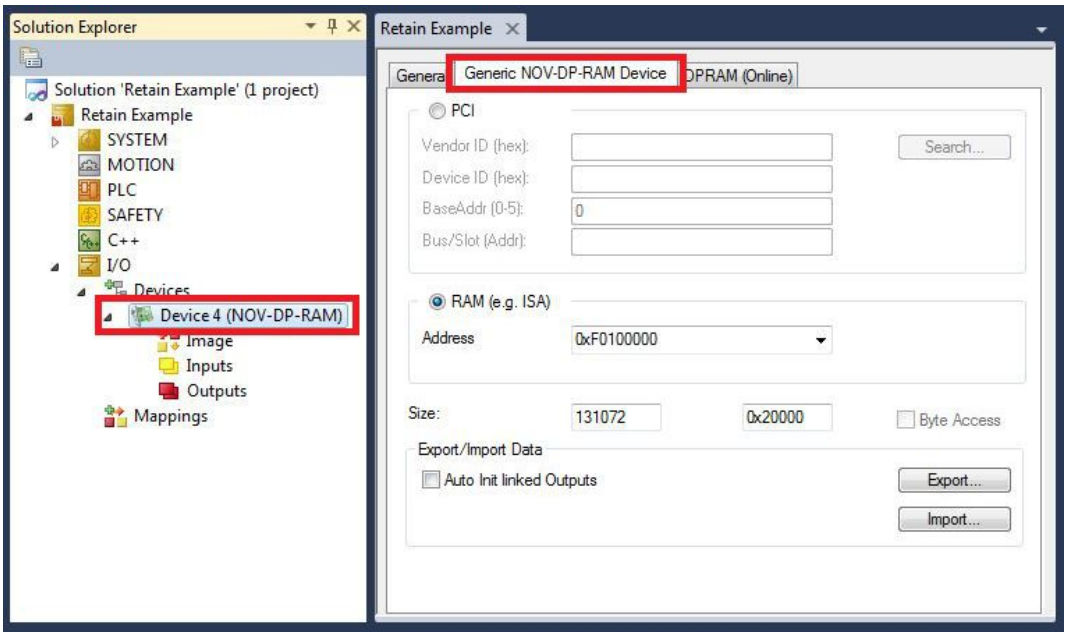


3. Wählen Sie **Device (NOV-DP-RAM)** und bestätigen Sie die Auswahl mit **OK**.

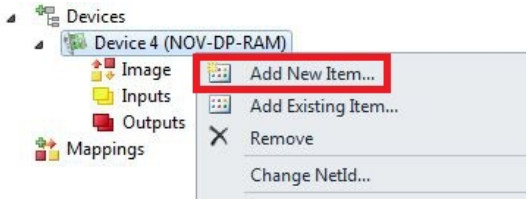


4. Klicken Sie auf **Ja**, um nach Boxen zu suchen.

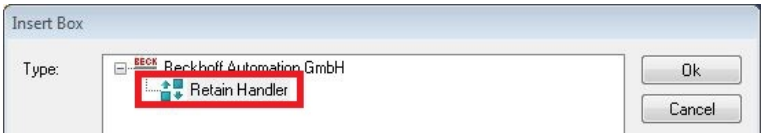
5. Klicken Sie links in der Strukturansicht auf **Device (NOV-DP-RAM)** und anschließend auf die Registerkarte **Generic NOV-DP-RAM Device**.



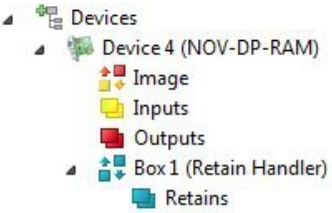
6. Klicken Sie auf die Option **RAM**.
7. Klicken Sie in der Strukturansicht mit rechter Maustaste auf **Device (NOV-DP-RAM)** und danach auf **Add New Item**.



8. Wählen Sie den **Retain Handler** und klicken Sie auf **OK**.



⇨ Sie haben erfolgreich einen Retain-Handler in TwinCAT angelegt.



Im nächsten Schritt können Sie Retain-Variablen in der SPS anlegen und mit dem Retain-Handler verknüpfen.

7.3.2 Variablen anlegen und verknüpfen

Nachdem Sie einen Retain-Handler in TwinCAT angelegt haben, können Sie in der SPS Variablen deklarieren und mit einem Retain-Handler verknüpfen. Die Variablen müssen in der SPS mit dem Schlüsselwort VAR_RETAIN gekennzeichnet werden.

Voraussetzung für diesen Arbeitsschritt:

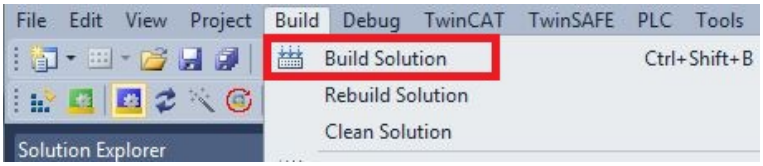
- Ein SPS-Projekt angelegt in TwinCAT.

Legen Sie Variablen wie folgt an:

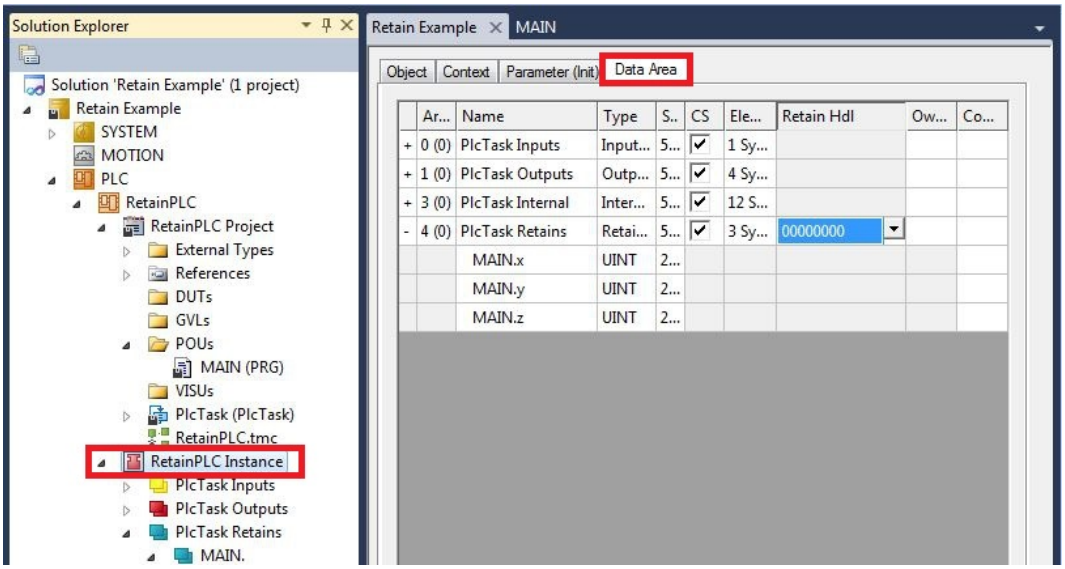
1. Legen Sie die Variablen in Ihrem SPS-Projekt in einem VAR RETAIN-Bereich an.

```
1  PROGRAM MAIN
2
3  VAR RETAIN
4      x       :UINT;
5      y       :UINT;
6      z       :UINT;
7  END_VAR
8
9  VAR
10
11     datain  AT%I*: REAL;
12     dataout AT%Q*: BYTE;
13
14 END_VAR
```

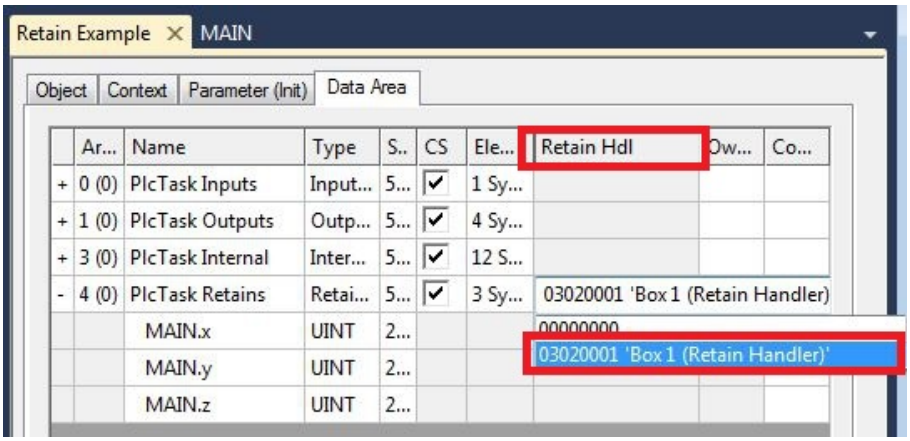
2. Klicken Sie oben auf der Symbolleiste auf **Build** und dann auf **Build Solution**.



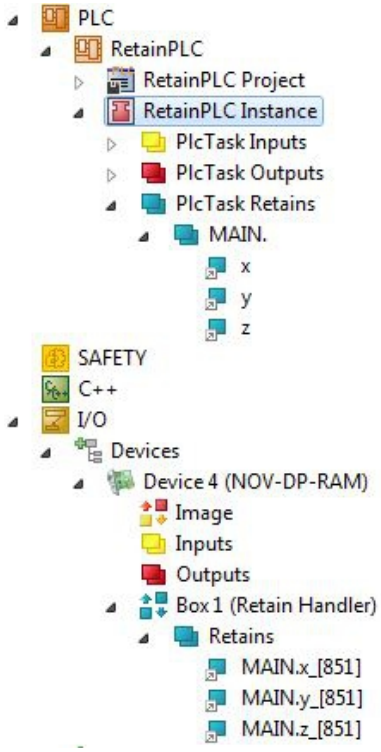
3. Klicken Sie links in der Strukturansicht auf Ihre **PLC Instance** und anschließend auf die Registerkarte **Data Area**.



4. Wählen Sie unter **Retain Hdl** den Retain Handler, den Sie angelegt haben.



⇨ Nachdem Sie einen Retain-Handler als Ziel ausgewählt haben, werden die Symbole in der Strukturansicht verknüpft und ein Mapping erzeugt.
In der Strukturansicht werden die Variablen aus der SPS unter dem Retain-Handler angelegt und sind mit den Variablen aus der SPS Instanz verknüpft.



Eine bestehende Verknüpfung wird mit einem Pfeilsymbol angezeigt.

7.3.3 Variablen unter dem Retain-Handler löschen

Wenn Variablen in der SPS gelöscht werden, dann wird die Verknüpfung mit dem Retain-Handler aufgehoben. Die Variablen werden aber unter dem Retain-Handler weiter angezeigt und werden nicht gelöscht.

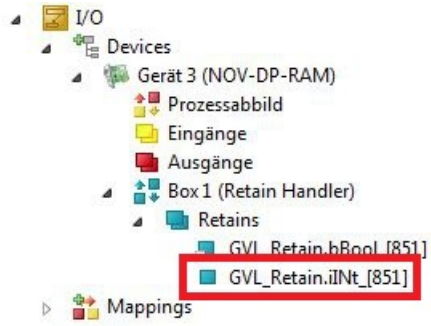
Diese Variablen müssen unter TwinCAT 3 manuell gelöscht werden.

Voraussetzungen für diesen Arbeitsschritt:

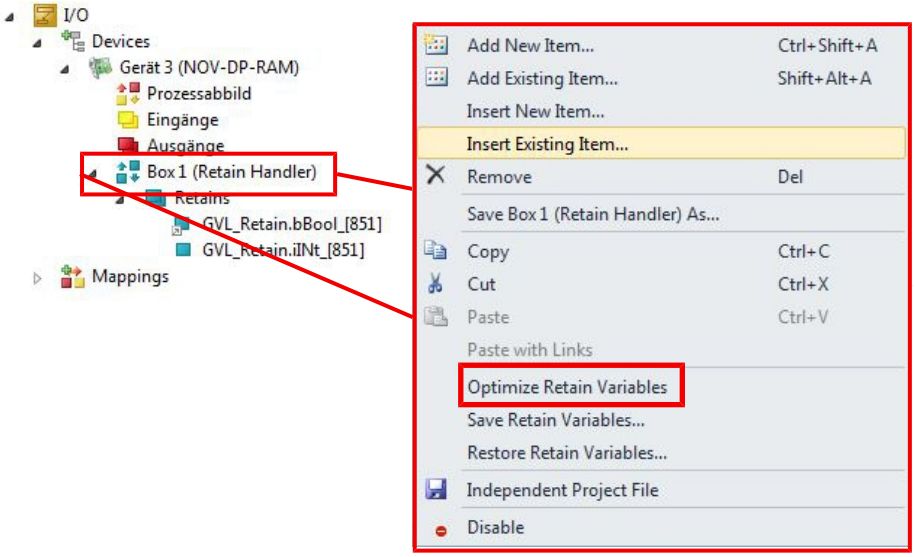
- Mit VAR_RATAIN deklarierte Variablen wurden in der SPS gelöscht.

Löschen Sie Variablen unter dem Retain-Handler wie folgt:

1. Die Variable GVL_Retain.iNt unter dem Retain-Handler soll gelöscht werden.



2. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf den **Retain-Handler**.
3. Klicken Sie im Kontextmenü auf **Optimize Retain Variables**.



⇨ Die Variable unter dem Retain-Handler wird gelöscht.

7.4 Softwarekonfiguration

7.4.1 Benutzername und Passwort

Im Auslieferungszustand verfügt der CX7050 über einen voreingestellten Benutzernamen mit Passwort, welcher für die Anmeldung in TwinCAT oder dem Beckhoff Device Manager notwendig ist.

- Benutzername: `Administrator`
- Passwort: `1`

Der Benutzername ist fix und kann nicht geändert werden. Es kann auch kein weiterer Benutzername hinzugefügt werden. Das voreingestellte Passwort kann über den Beckhoff Device Manager geändert werden (siehe: Beckhoff Device Manager starten). Das Passwort kann maximal 32 Zeichen enthalten. Erlaubt sind Zahlen, Buchstaben und Sonderzeichen, wobei auch zwischen Groß- und Kleinschreibung unterschieden wird.

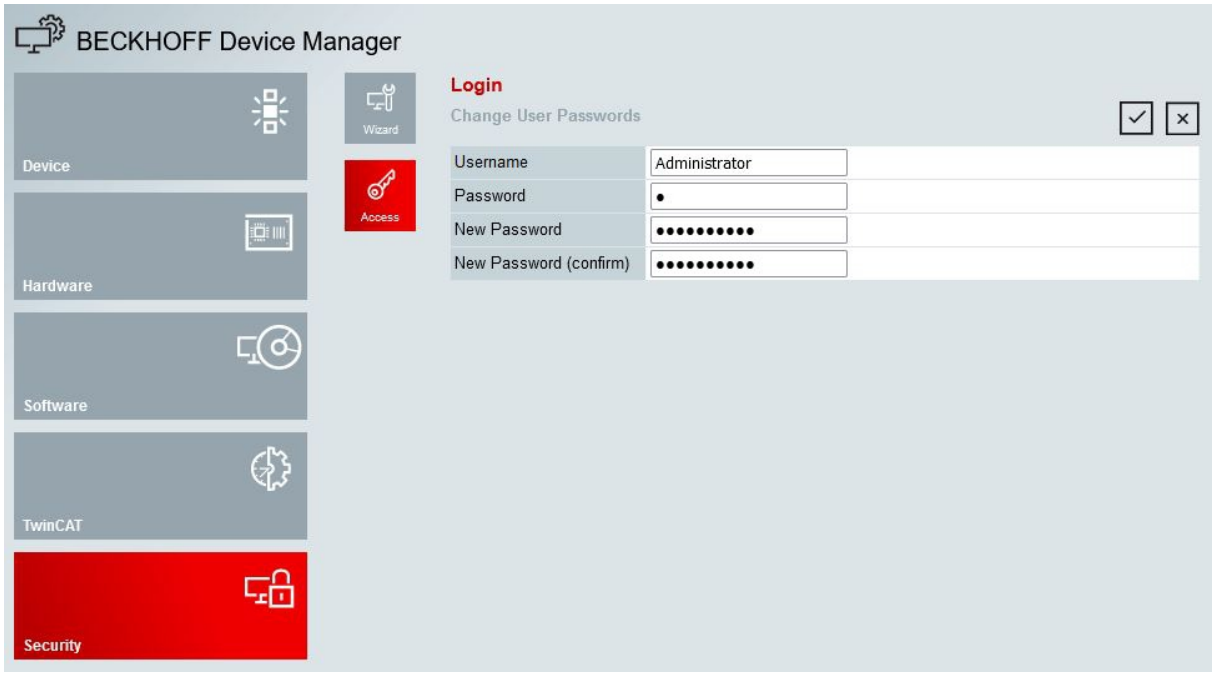


Abb. 37: Änderung des Passworts im Beckhoff Device Manager.

Sie können den Auslieferungszustand und das voreingestellte Passwort wiederherstellen, wenn Sie die MicroSD-Karte ausbauen, auf die MicroSD-Karte mit einem Kartenleser zugreifen und die Datei `device.conf` im Ordner `/etc` löschen. Ohne physischen Zugriff auf den CX7050 und damit auf die MicroSD-Karte, kann das Passwort nicht zurückgesetzt werden.

7.4.2 IP-Adresse einstellen

Beim CX7050 ist standardmäßig DHCP aktiv. Ohne DHCP-Server verwendet der CX7050 eine lokale IP-Adresse im Adressbereich 169.254.x.x

Beim Embedded-PC CX7050 gibt es mehrere Möglichkeiten die IP-Adresse einzustellen. Eine Möglichkeit besteht darin, den Beckhoff Device Manager aufzurufen und die IP-Adresse für den CX7050 im Browser einzustellen (siehe: Beckhoff Device Manager starten).

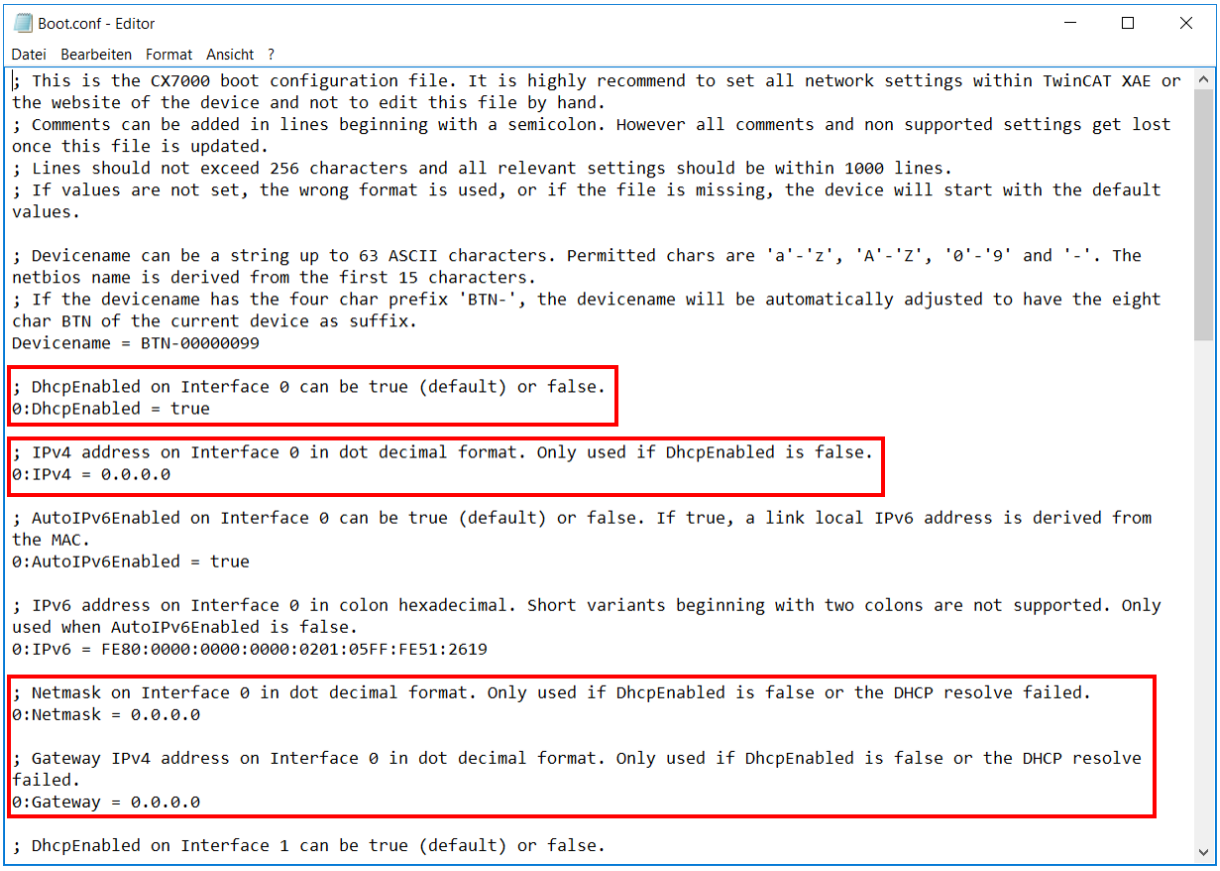
Eine andere Möglichkeit die IP-Adresse einzustellen, bietet die boot.conf-Datei, die nach dem ersten Start auf der MicroSD-Karte angelegt wird. In diesem Schritt wird gezeigt, wie Sie die IP-Adresse in der boot.conf-Datei einstellen können.

Voraussetzungen:

- MicroSD-Kartenleser

Gehen Sie wie folgt vor:

1. Schalten Sie den Embedded-PC aus und entfernen Sie die MicroSD-Karte aus dem Embedded-PC.
2. Öffnen Sie die Datei Boot.conf unter \etc



3. Stellen Sie den Eintrag **DhcpEnabled** auf **false**.
4. Vergeben Sie eine IP-Adresse unter **IPv4**.
5. Legen Sie die Einstellungen für Subnetzmaske, Gateway und DNS-Server fest.

⇨ Speichern Sie die Änderungen und bauen Sie die MicroSD-Karte wieder in den Embedded-PC ein. Die Einstellungen sind nach dem Start wirksam.

7.4.3 Image aktualisieren

HINWEIS

Ausfall der Spannungsversorgung

Der Bootloader kann beschädigt werden, wenn die Aktualisierung unterbrochen wird. Der CX70x0 wird damit unbrauchbar und muss eingeschickt werden. Sorgen Sie für eine stabile Spannungsversorgung während des ersten Starts und unterbrechen Sie nicht die Aktualisierung.

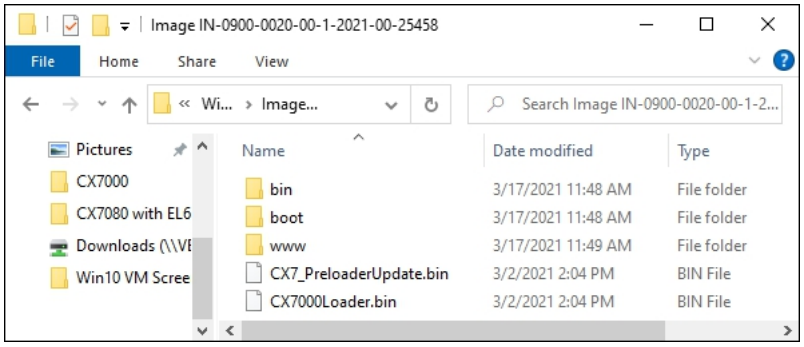
Das neue Image wird direkt auf die MicroSD-Karte kopiert, um das Image des Embedded-PCs zu aktualisieren. Das neue Image wird vom Beckhoff Service zur Verfügung gestellt. Führen Sie das Update nur nach Rücksprache mit dem Beckhoff Service durch.

Voraussetzungen:

- Kartenleser für MicroSD-Karten.

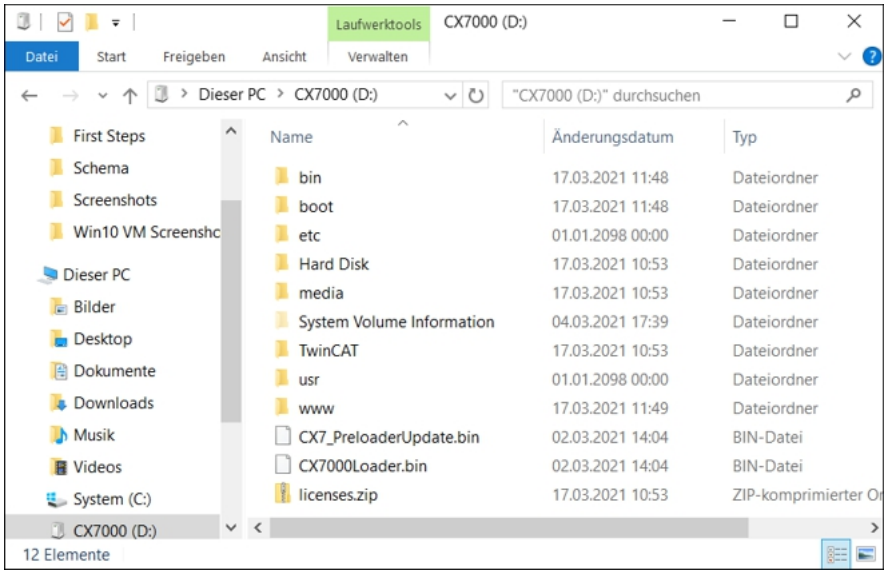
Aktualisieren Sie das Image wie folgt:

1. Schalten Sie den Embedded-PC aus und entfernen Sie die MicroSD-Karte aus dem Embedded-PC.
2. Stecken Sie die MicroSD-Karte in einen externen Kartenleser und öffnen Sie die Ordnerstruktur der MicroSD-Karte.
3. Löschen Sie alle Dateien und Ordner auf der MicroSD-Karte.
4. Kopieren Sie alle Dateien und Ordner des neuen Images auf die leere MicroSD-Karte.



5. Bauen Sie die MicroSD-Karte wieder in den Embedded-PC ein und starten Sie den Embedded-PC.

⇨ Der Embedded-PC wird gestartet und speichert die aktuelle Hardwarekonfiguration. Neue Ordner, wie beispielsweise Hard Disk oder TwinCAT, werden erzeugt. Damit wurde das Image erfolgreich aktualisiert.



7.4.4 Firmware für Multifunktions-I/Os aktualisieren

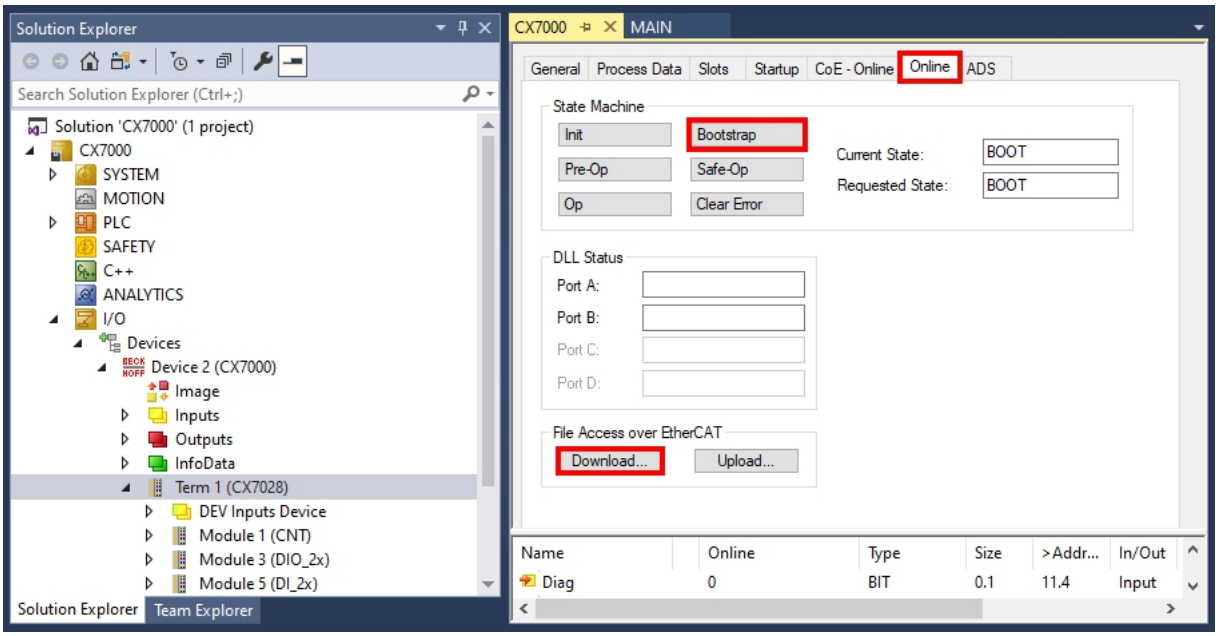
In diesem Schritt wird gezeigt, wie Sie die Firmware der Multifunktions-I/Os aktualisieren können. Die Firmware wird vom Beckhoff-Service zur Verfügung gestellt und die Aktualisierung in TwinCAT ausgeführt.

Voraussetzungen:

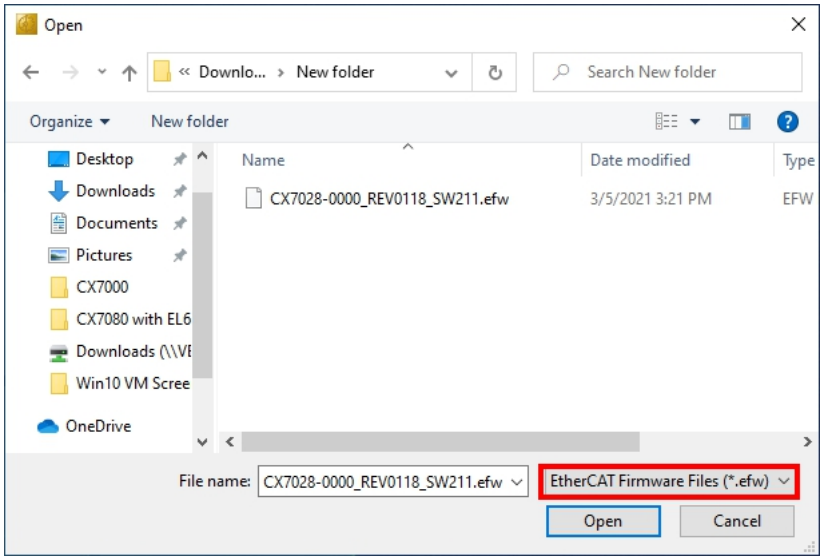
- EtherCAT-Firmware-File (*.efw)

Gehen Sie wie folg vor:

1. Starten Sie TwinCAT im Konfigurationsmodus (Config-Modus).
2. Klicken Sie links im Strukturbaum auf das Gerät CX7028 und anschließend auf die Registerkarte **Online**.



3. Klicken Sie auf die Schaltfläche **Bootstrap**, um die Multifunktions-I/Os in den Zustand Bootstrap zu schalten.
4. Klicken Sie auf die Schaltfläche **Download** und wählen Sie eine aktuelle efw-Datei aus.



⇨ Die Aktualisierung benötigt ca. 3 bis 4 Minuten. Ein Fortschrittsbalken zeigt den Fortschritt der Aktualisierung an. Schalten Sie den CX7050 während dieser Zeit nicht aus.

Wechseln Sie nach Abschluss der Aktualisierung wieder in den Zustand Operational (Op), indem Sie auf die Schaltfläche **Op** Klicken.

7.4.5 ESI-Gerätebeschreibung aktualisieren

Der TwinCAT System Manager und der TwinCAT EtherCAT-Master benötigt zur Konfiguration im Online- und Offline-Modus die Gerätebeschreibungsdateien aller EtherCAT-Geräte. Diese Gerätebeschreibungen sind die sogenannten ESI-Dateien (EtherCAT Slave Information) im XML-Format. Diese Dateien können vom jeweiligen Hersteller angefordert werden bzw. werden zum Download bereitgestellt. Eine *.xml-Datei kann dabei mehrere Gerätebeschreibungen enthalten.

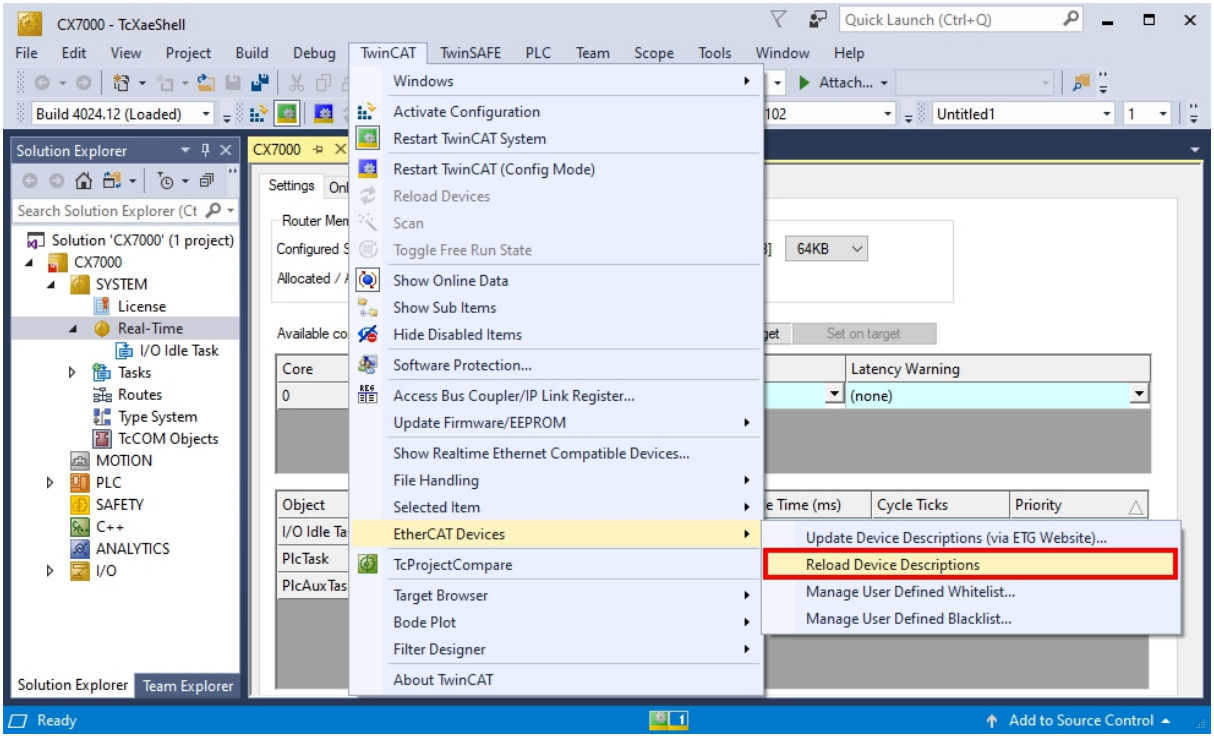
ESI-Dateien für Beckhoff EtherCAT-Geräte werden unter https://www.beckhoff.com bereitgestellt.

Voraussetzungen:

- ESI-Datei für den CX7050 im XML-Format.
- Gegebenenfalls die dazugehörige *.xsd-Datei, die den Aufbau der XML-Datei beschreibt.

Gehen Sie wie folgt vor:

1. Kopieren Sie die ESI-Datei in das TwinCAT-Installationsverzeichnis: *\TwinCAT\3.1\Config\Io\OnboardIo*.
2. Wenn der Ordner nicht existiert, erstellen Sie den Ordner manuell.
3. Öffnen Sie TwinCAT und klicken Sie im Menü unter **TwinCAT > EtherCAT Devices** auf **Reload Device Description**.



⇨ Die ESI-Datei wird in TwinCAT neu eingelesen. Sollte eine fehlerhafte ESI-Datei vorliegen, wird ein Fehler ausgegeben. Überprüfen Sie, ob Aufbau der *.xml der dazugehörigen *.xsd-Datei entspricht oder ob die Dateien zum CX7050 passen.

8 TwinCAT

8.1 Erste Schritte

8.1.1 Mit CX70x0 verbinden

Bevor Sie den CX7050 in TwinCAT konfigurieren können, müssen Sie eine Verbindung zwischen ihrem Engineering-Rechner und dem CX7050 (Zielsystem) herstellen. Der Engineering-Rechner und der Embedded-PC müssen sich im gleichen Netzwerk und Subnetz befinden oder alternativ direkt mit einem Ethernet-Kabel (Peer-to-Peer) verbunden werden.

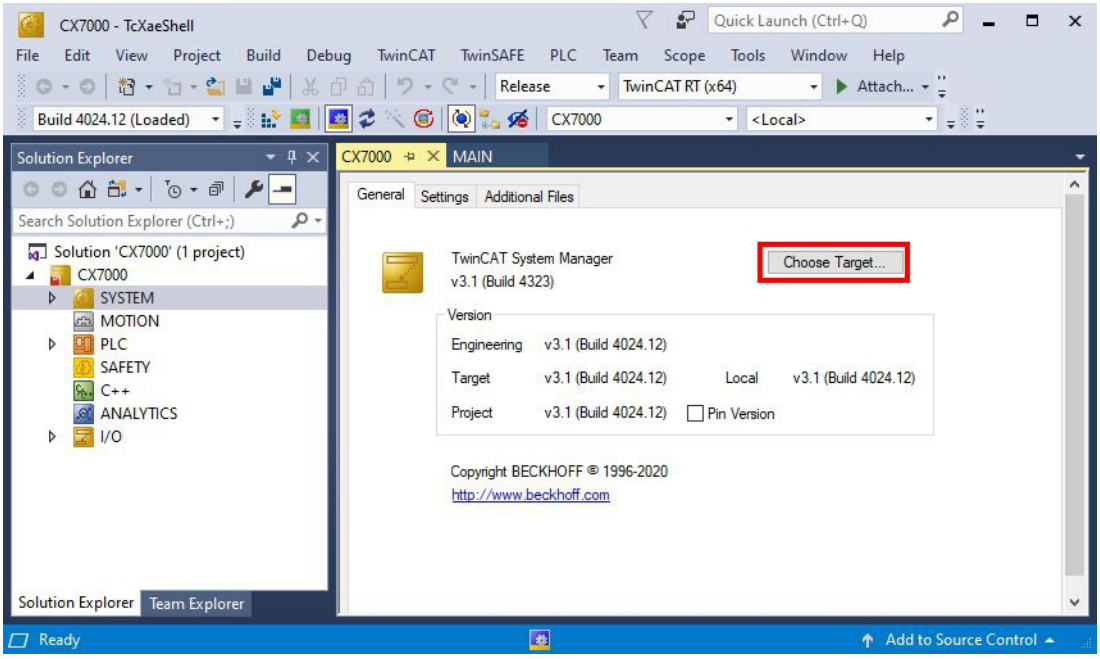
Für die Verbindung ist die IP-Adresse oder der Hostname des CX7050 notwendig.

Voraussetzungen:

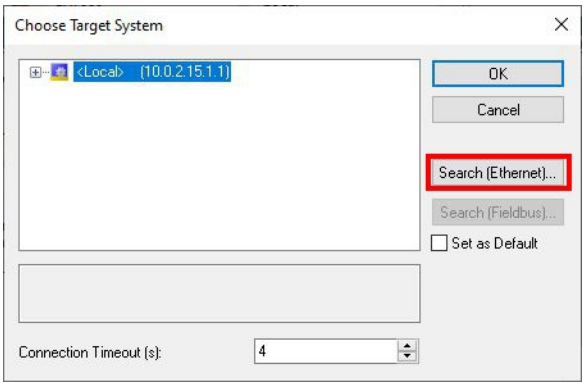
- TwinCAT muss sich im Config-Modus befinden.
- IP-Adresse oder Hostname des Embedded-PCs.

Stellen Sie eine Verbindung wie folgt her:

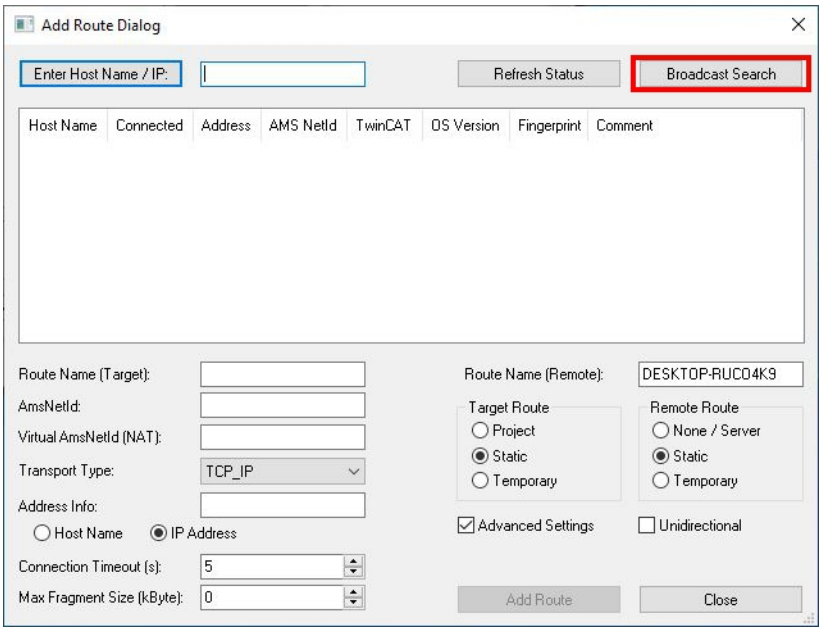
1. Klicken Sie oben im Menü auf **File > New > Project** und erstellen Sie ein neues TwinCAT XAE Projekt.
2. Klicken Sie links in der Strukturansicht auf **SYSTEM** und dann auf **Choose Target**.



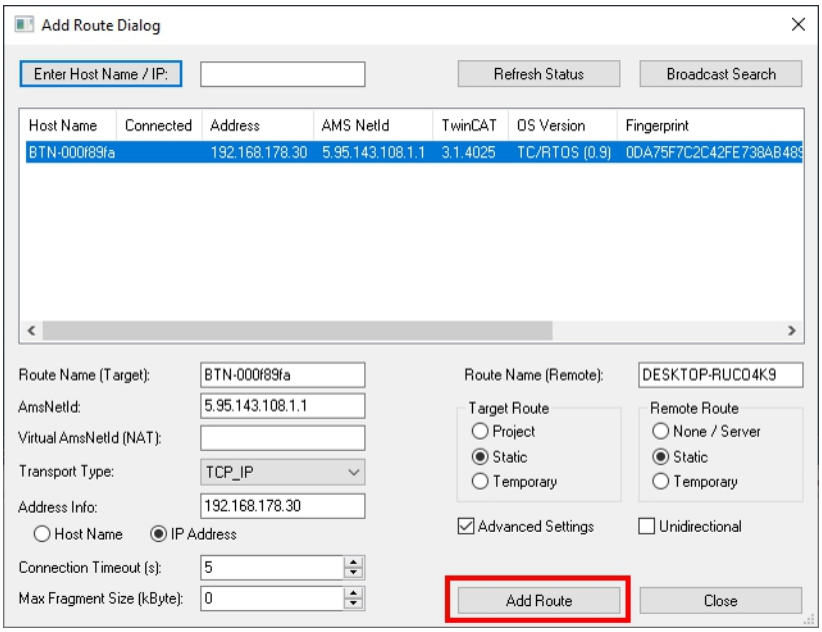
3. Klicken Sie auf **Search (Ethernet)**.



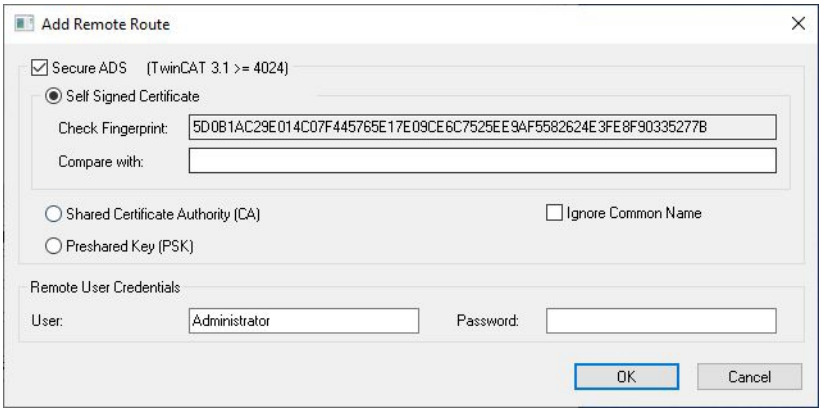
4. Klicken Sie auf **Broadcast Search** und suchen Sie nach verfügbaren Geräten im Netzwerk.



5. Markieren Sie den passenden CX7050 und klicken Sie auf **Add Route**. Der Hostname und die IP-Adresse erleichtern dabei die Identifikation.

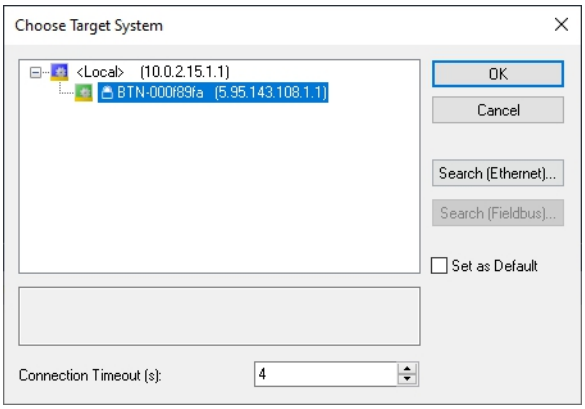


6. Geben Sie im Feld **User** und im Feld **Password** den Benutzernahmen und das Passwort ein und klicken Sie auf **OK**. Benutzername: `Administrator` Passwort: `1`

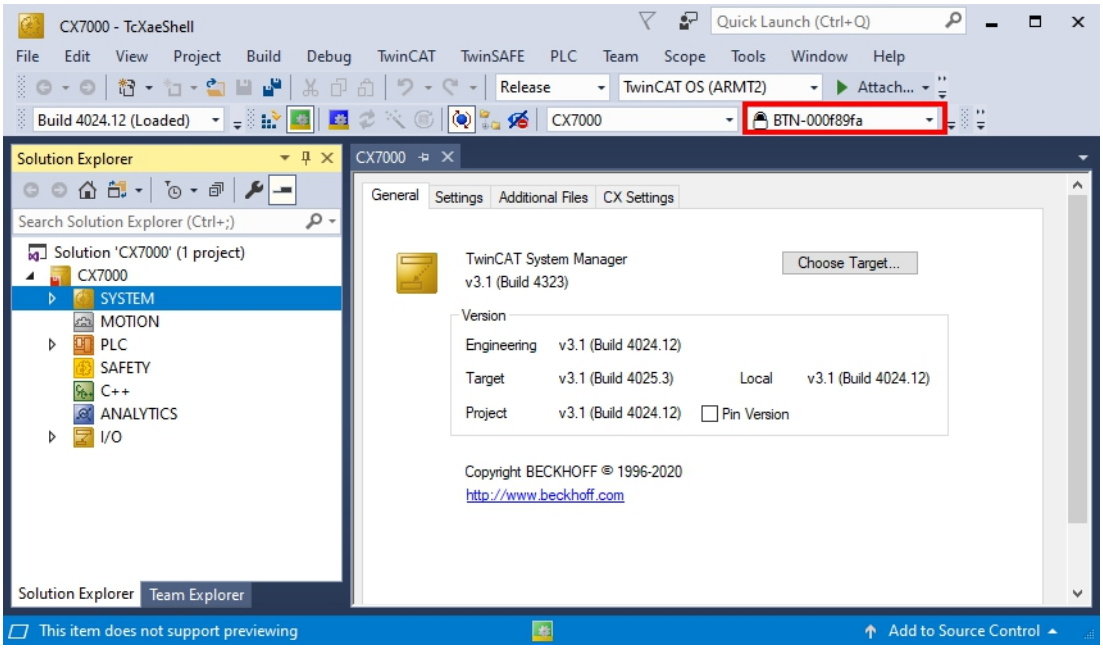


7. Das neue Gerät wird im Fenster **Choose Target System** angezeigt.

8. Markieren Sie das Gerät welches Sie als Zielsystem festlegen wollen und klicken Sie auf **OK**.



⇨ Sie haben erfolgreich in TwinCAT eine Verbindung zwischen ihrem Engineering-Rechner und dem CX7050 (Zielsystem) hergestellt. In der Menüleiste wird das neue Zielsystem mit dem Hostnamen angezeigt.



Mit dieser Vorgehensweise können Sie nach allen verfügbaren Geräten suchen und auch jederzeit zwischen den Zielsystemen wechseln.

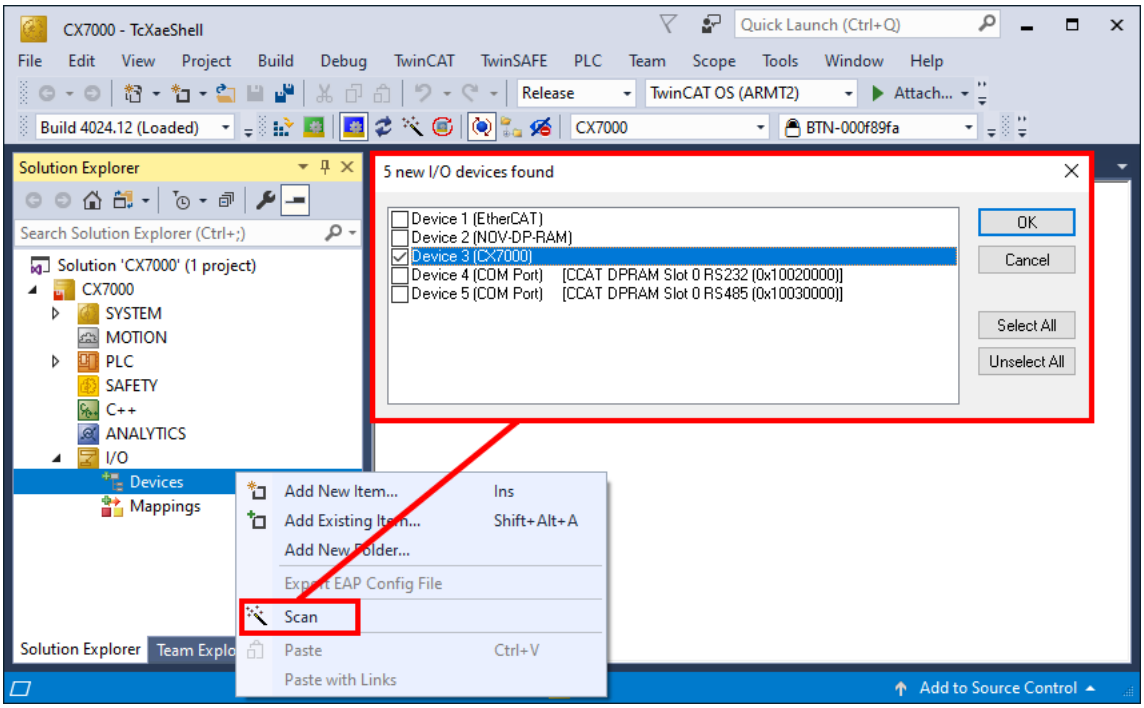
8.1.2 Multifunktions-I/Os scannen

Als Besonderheit verfügt die CX7000-Serie über acht integrierte Multifunktionseingänge und vier integrierte Multifunktionsausgänge. In diesem Kapitel wird gezeigt, wie Sie die Multifunktions-I/Os in TwinCAT scannen und anlegen können.

Beachten Sie, dass die CX7028-Schnittstelle für die Steuerung der Multifunktions-I/Os eine eigene CPU hat und die CX7028-Schnittstelle unter TwinCAT nicht angezeigt wird bzw. nicht funktioniert, wenn die Spannungsversorgung(Up) nicht angeschlossen ist.

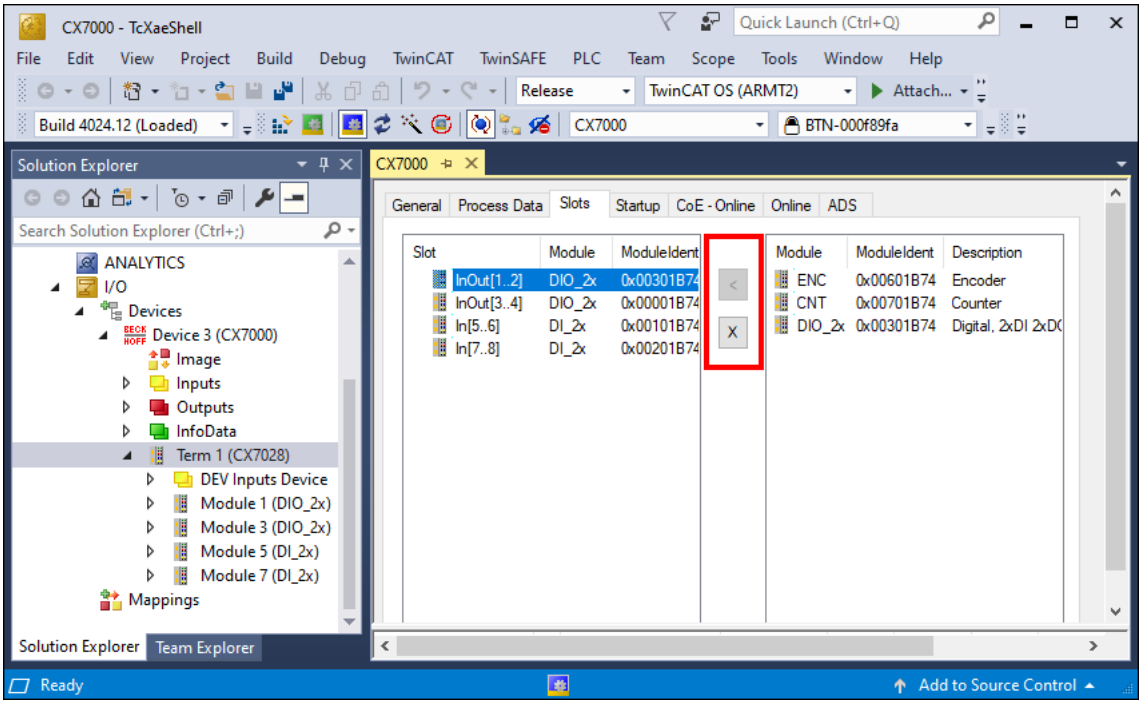
Gehen Sie wie folgt vor:

1. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf **Devices** und anschließend auf **Scan**.



Alle verfügbaren I/O-Geräte werden angezeigt.

2. Wählen Sie die passenden I/O-Geräte aus. Für dieses Beispiel muss mindestens die CX7028-Schnittstelle, also das CX7000-Gerät ausgewählt werden. Wenn Sie noch Bus- oder EtherCAT-Klemmen am CX7000 betreiben wollen, dann müssen Sie zusätzlich EtherCAT als Gerät auswählen.
3. Es werden insgesamt vier Slots angelegt. Für jeden Slot kann maximal ein Modul (DI, DIO, ENC, CNT oder PWM) zugewiesen werden, welches wiederum die Betriebsart für den jeweiligen Slot festlegt.



4. Mit der Schaltfläche **<** können Module einem bestimmten Slot zugewiesen oder mit **x** wieder entfernt werden.

⇨ Legen Sie die benötigten Module ihren Anforderungen entsprechend fest. Abhängig vom verwendeten Slot stehen unterschiedliche Module zur Auswahl. Welche Module von welchem Slot unterstützt werden, wird im Kapitel Multifunktions-I/Os [▶ 97] aufgelistet.

8.1.3 ADS-Kommunikation herstellen

In diesem Kapitel wird gezeigt, wie Sie einen CX7050 mit einem anderen CX70x0 oder einer beliebigen TwinCAT-Steuerung verbinden können. Das ADS-Protokoll bietet die einfachste Möglichkeit, zwei TwinCAT-Systeme miteinander zu verbinden. Mit dem ADS-Protokoll können Daten sowohl gelesen als auch geschrieben werden. Für gewöhnlich werden ADS-Bausteine für die Kommunikation verwendet, die in der Tc2_System-Bibiliothek enthalten sind. In dem folgenden Beispiel sollen Daten von einem Speicherbereich gelesen und geschrieben werden.

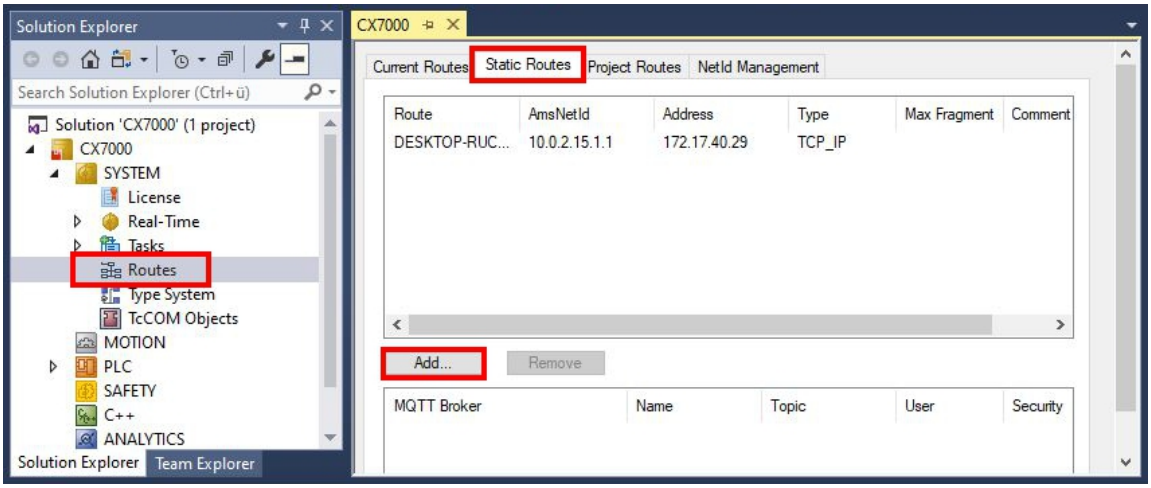
Um eine ADS-Verbindung einzurichten, wird zunächst eine ADS-Route erstellt. Die Kommunikation erfolgt dann über Ethernet und der Datenaustausch über das TCP/IP-Protokoll. Die ADS-Route ist dann die Schnittstelle zwischen der ADS- und TCP/IP-Verbindung. Durch die ADS-Route ist bekannt, welche AmsNetId welcher TCP/IP-Adresse zugewiesen ist. Dadurch wird in den ADS-Bausteinen nicht mehr die TCP/IP-Adresse, sondern nur noch die AmsNetId verwendet.

Voraussetzungen:

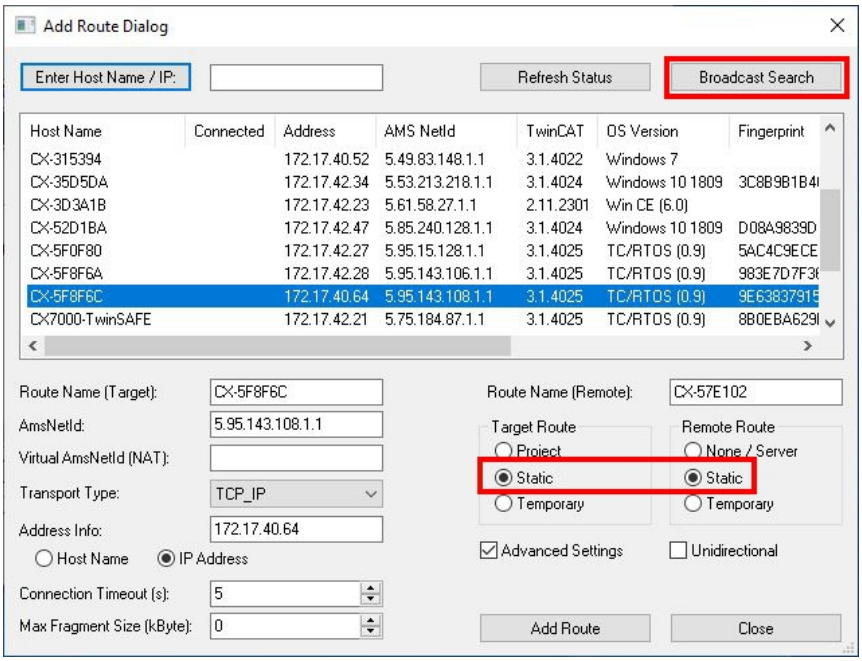
- Zwei CX70x0 Embedded-PCs.
- Beide CX70x0 sind im gleichen Netzwerk und über ADS erreichbar.

Gehen Sie wie folgt vor:

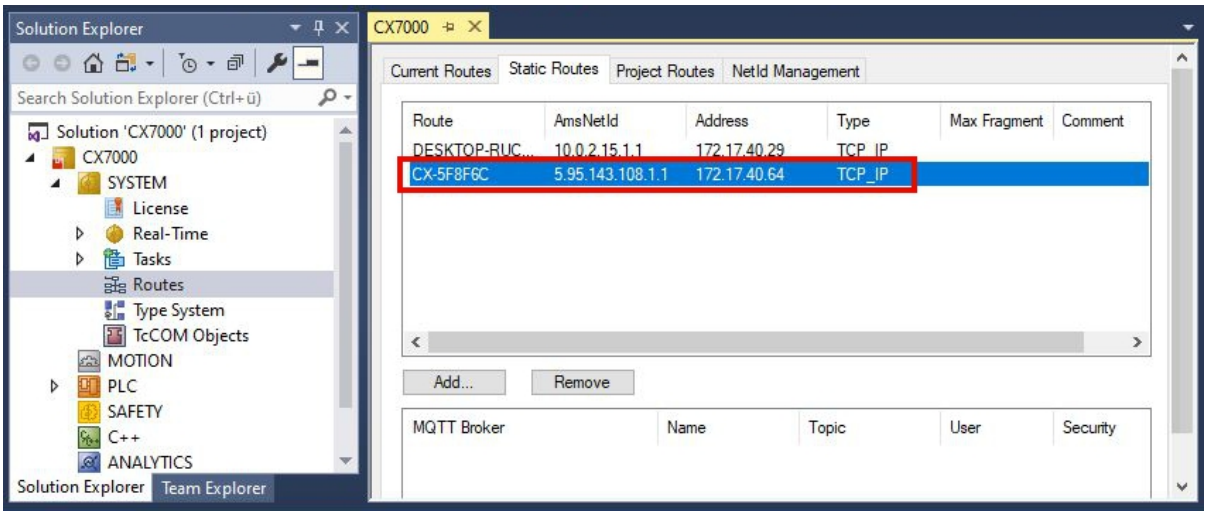
1. Starten Sie TwinCAT und verbinden Sie sich mit dem ersten CX70x0 (siehe: Mit CX70x0 verbinden [▶ 131]).
2. Klicken Sie links in der Strukturansicht auf **Routes**, wählen Sie die Registerkarte **Static Routes** und klicken Sie auf die Schaltfläche **Add**.



3. Wählen Sie unter **Remote Route** die Option **Static**, damit die ADS-Route im Projekt bestehen bleibt und klicken Sie anschließend auf die Schaltfläche **Broadcast Search**.



4. Wählen Sie den zweiten CX70x0 als Ziel der ADS-Route aus. Die ADS-Route wird bei beiden Embedded-PCs eingetragen. Die AmsNetId des zweiten CX70x0 wird angezeigt und kann im Programm für ADS-Bausteine verwendet werden.



5. Verbinden Sie sich jetzt mit dem zweiten CX70x0, welcher als Ziel der ADS-Route festgelegt wurde und schreiben Sie ein kleines Programm. Definieren Sie ein Array und zählen Sie einen Wert des Arrays hoch.

```
VAR
    MarksTest AT %MB0     : ARRAY[0..9] of INT;
END_VAR

Program:
MarksTest[0]:=MarksTest[0]+1;
```

6. Aktivieren Sie die Konfiguration und schalten Sie den CX70x0 in den Run-Modus.
7. Schreiben Sie für den ersten CX70x0 ein Programm, welches den hochgezählten Wert des Arrays ausliest.

```
VAR
    ADSREAD : ADSREAD;
    NetID : STRING:='5.81.38.23.1.1'; (* AMSNetId of the target*)
    Value : INT; (* value of target MarksTest[0]*)
    Error : INT;
    NoError : INT;
END_VAR
```

```
Program:
    ADSREAD(
        NETID:=NetID ,
        PORT:=851 , (* plc port of the target*)
        IDXGRP:=16#4020 , (* Marks %MB*)
        IDXOFFS:=0 , (* Marks offset in byte*)
        LEN:=2 , (* length of data in byte*)
        DESTADDR:=ADR(Value) , (* pointer to the data in which the value is to be stored *)
        READ:=TRUE ,
        TMOUT:= ,
        BUSY=> ,
        ERR=> ,
        ERRID=> );
    IF NOT ADSREAD.BUSY THEN
        IF NOT ADSREAD.ERR THEN
            NoError:=NoError+1;
        ELSE
            Error:=Error+1;
        END_IF
    ADSREAD(Read:=FALSE);
    END_IF
```

8. Der hochgezählte Wert wird ausgelesen und an den ersten CX70x0 übermittelt.

⇨ Sie sollten beim ersten CX70x0 sehen, wie der Wert der Variable `Value` hochgezählt wird. Analog dazu funktioniert das Schreiben der Daten. Daten können mit dem Baustein `ADSWRITE` geschrieben werden. Achten Sie darauf, dass Sie in diesem Beispielaufbau den Offset (`IDXOFFSET`) auf 10 setzen, damit das Array [4…9] beschrieben wird. Beschränken Sie die Länge auf 10 Byte, da ein Array von 0…9 vom Typ INT angelegt wurde und damit der Speicher %MB0…MB19 (10 * 2 Byte) verwendet (Die Elemente 0...4 zum Lesen und die Elemente 5...9 zum Schreiben des Arrays).

Verwenden Sie einen ADS-Befehl nach dem anderen. Warten Sie, bis der ADS-Dienst fertig ist, also der Ausgang `BUSY` des Bausteins auf `FALSE` geschaltet wird und erst danach den nächsten ADS-Baustein verwenden. Um den Zugriff zeitlich zu optimieren, können Sie auch einen `ADSREADWRITE` Baustein verwenden, der die Daten liest und gleichzeitig schreibt.

8.1.4 SPS-Projekt erstellen

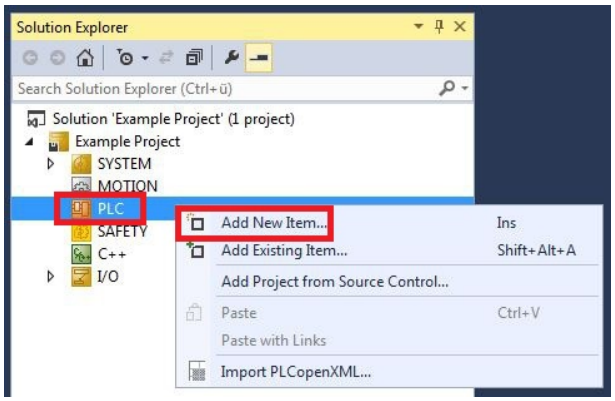
In den nächsten Schritten wird beschrieben, wie Sie ein PLC-Projekt in TwinCAT erstellen und in der Strukturansicht einfügen.

Voraussetzungen für diesen Arbeitsschritt:

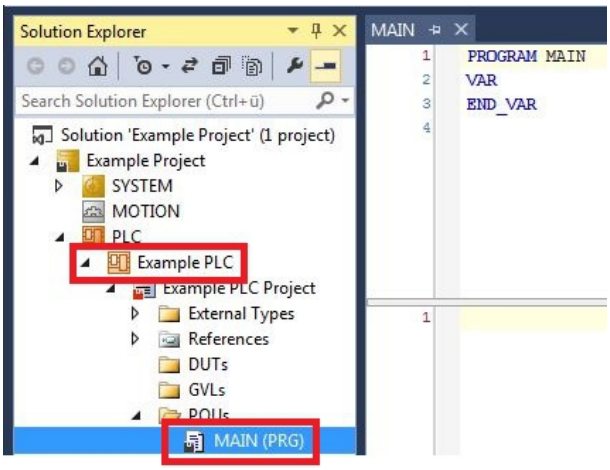
- Ein neu angelegtes TwinCAT XAE Projekt.

Erstellen Sie ein PLC-Projekt wie folgt:

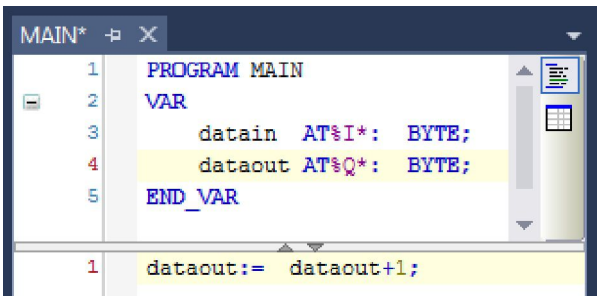
1. Klicken Sie in der Strukturansicht mit der rechten Maustaste auf **PLC**.
2. Klicken Sie im Kontextmenü auf **Ad New Item** und wählen Sie das **Standard PLC Project**.



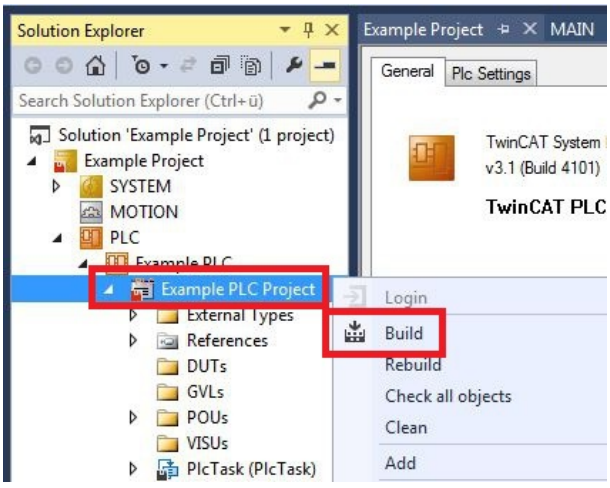
3. Klicken Sie in der Strukturansicht auf das neu erstellte PLC-Projekt und dann unter **POUs** doppelt auf **MAIN (PRG)**.



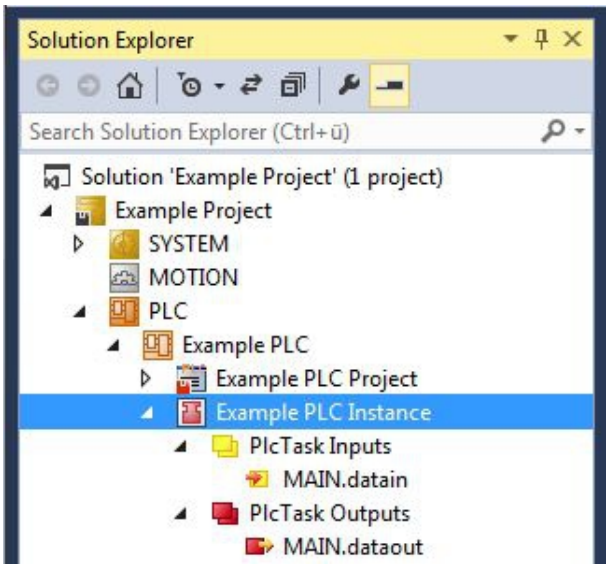
4. Schreiben Sie ein kleines Programm wie im folgenden Bild.



5. Klicken Sie in der Strukturansicht mit der rechten Maustaste auf das PLC-Projekt und dann im Kontextmenü auf **Build**.



⇨ Sie haben erfolgreich ein PLC-Projekt erstellt und das Projekt in TwinCAT angefügt. Es wird eine PLC-Instanz mit den Variablen für die Eingänge und Ausgänge aus dem PLC-Projekt erstellt.



Im nächsten Schritt können Sie die Variablen mit der Hardware verknüpfen.

8.1.5 Variablen verknüpfen

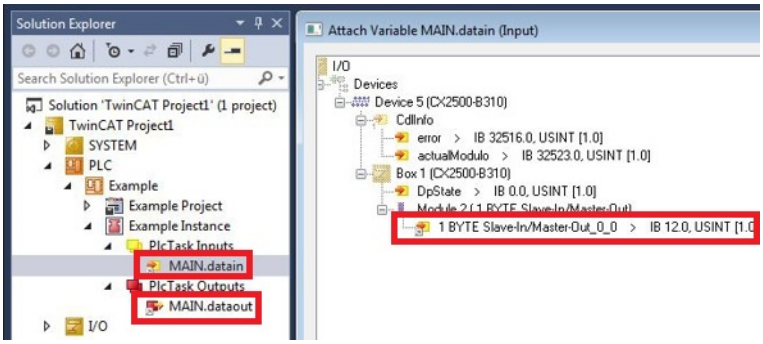
Wurde das PLC-Projekt erfolgreich im System Manager angefügt, können die Sie die neu angelegten Ein- und Ausgangsvariablen aus dem PLC-Projekt mit den Ein- und Ausgängen Ihrer Hardware verknüpfen.

Voraussetzungen für diesen Arbeitsschritt:

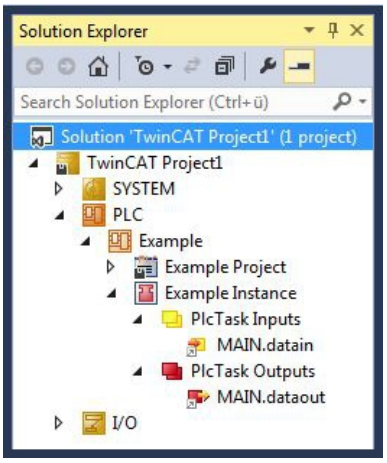
- Ein angefügtes PLC-Programm in TwinCAT.

Verknüpfen Sie die Variablen wie folgt:

1. Klicken Sie doppelt auf die Ein- bzw. Ausgangsvariablen in der Strukturansicht unter **PLC**.
   Das Fenster **Attach Variable** erscheint und zeigt an, welche Eingänge bzw. Ausgänge mit den Variablen aus dem PLC-Projekt verknüpft werden können.

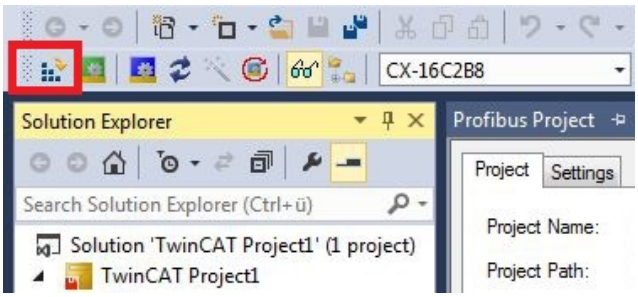


2. Klicken Sie doppelt im Fenster **Attach Variable** auf die Ein bzw. Ausgänge der Hardware.
   Verknüpfen Sie die Eingangsvariablen mit den Eingängen und die Ausgangsvariablen mit den Ausgängen der Hardware.



   Bereits verknüpfte Variablen werden in TwinCAT mit einem kleinen Pfeilsymbol markiert.

3. Klicken Sie in der Symbolleiste auf **Activate Configuration**.



4. Bestätigen Sie die Anfrage, ob TwinCAT im Free Run Modus gestartet werden soll, mit **Ja**.

⇨ Sie haben erfolgreich Variablen mit der Hardware verknüpft. Mit Activate Configuration wird die aktuelle Konfiguration gesichert und aktiviert.

Als nächstes kann die Konfiguration auf den CX geladen werden, um TwinCAT automatisch im Run Modus und dann das PLC-Projekt zu starten.

8.1.6 Konfiguration auf CX laden

Wenn Variablen verknüpft sind, kann die Konfiguration gespeichert und auf den CX geladen werden. Das hat den Vorteil, dass das PLC-Projekt automatisch geladen und gestartet werden kann, wenn der CX eingeschaltet wird. Der Start des zuvor erstellten PLC-Projekts kann damit automatisiert werden.

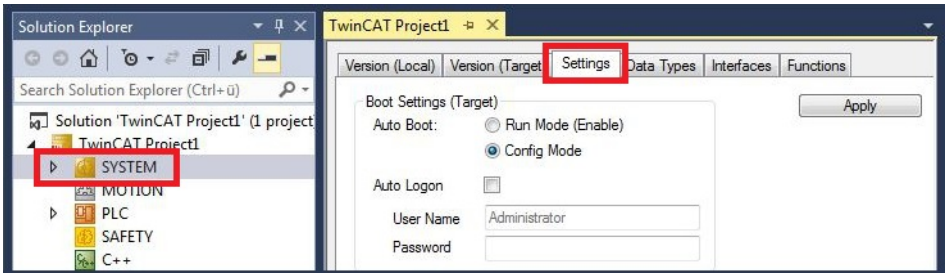
Voraussetzungen für diesen Arbeitsschritt:

- Ein fertiges und im System Manager angefügtes PLC-Projekt.
- Variablen aus dem PLC-Projekt verknüpft mit der Hardware im System Manager.
- Ein CX ausgewählt als Zielsystem.

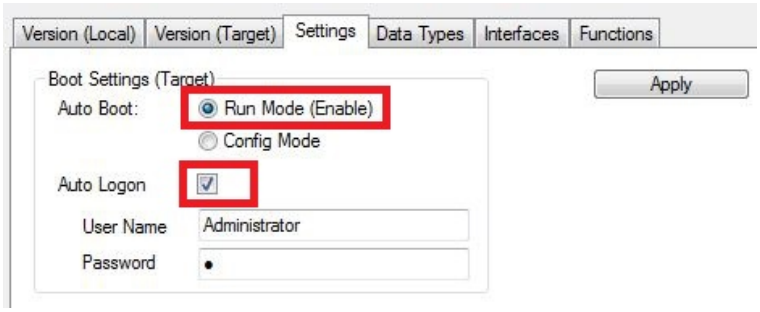
Laden Sie die Konfiguration aus dem System Manager wie folgt auf den CX:

1. Klicken Sie links in der Strukturansicht auf **SYSTEM**.

2. Klicken Sie auf die Registerkarte **Settings**.



3. Wählen Sie unter Boot Settings die Option **Run Mode (Enable)** und Aktivieren Sie das Kontrollkästchen **Auto Logon**.

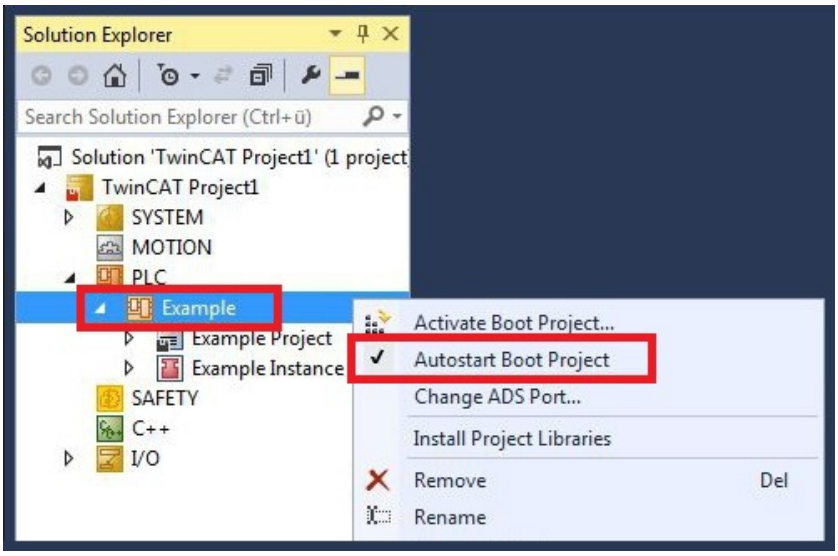


4. Geben Sie im Feld **User Name** und im Feld **Password** den Benutzernahmen und das Passwort für den CX ein.

5. Klicken Sie auf **Apply**.

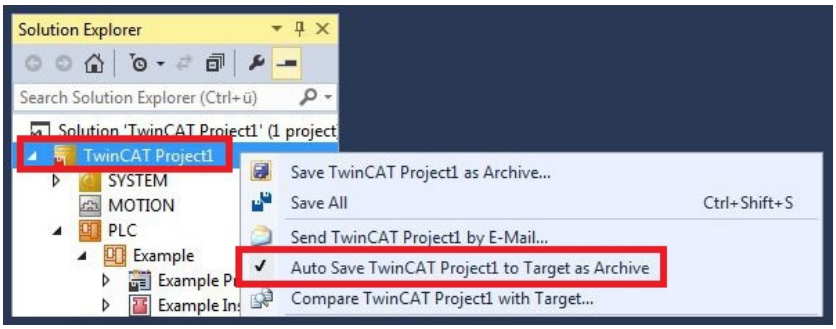
6. Klicken Sie links in der Strukturansicht unter **PLC** mit der rechten Maustaste auf das PLC-Projekt.

7. Klicken Sie im Kontextmenü auf **Autostart Boot Project**.
   Die Eistellung wird markiert



8. Klicken Sie in der Strukturansicht mit der rechten Maustaste auf den Projektordner.

9. Klicken Sie im Kontextmenü auf **Auto Save to Target as Archive**.
   Die Einstellung wird markiert.



⇨ Sie haben erfolgreich die Konfiguration auf den CX geladen. Ab jetzt wird bei jedem Start TwinCAT im Run Mode und das PLC-Projekt gestartet.

Als nächstes kann der Master in einem neuen Projekt im System Manager angefügt und über den Master nach den fertig eingerichteten Slaves gesucht werden.

8.2 TwinCAT-Registerkarten

In TwinCAT werden unter den Registerkarten Informationen und Einstellungen für die CANopen-Schnittstelle einsortiert. In diesem Kapitel werden die wichtigsten TwinCAT-Registerkarten beschrieben. Zusätzlich dazu wird gezeigt, wie die CANopen-Schnittstelle unter TwinCAT in der Strukturansicht angezeigt wird.

8.2.1 Strukturansicht

Ein CANopen-Master und ein daran angeschlossener CANopen-Slave, werden wie folgt in der Strukturansicht angezeigt:

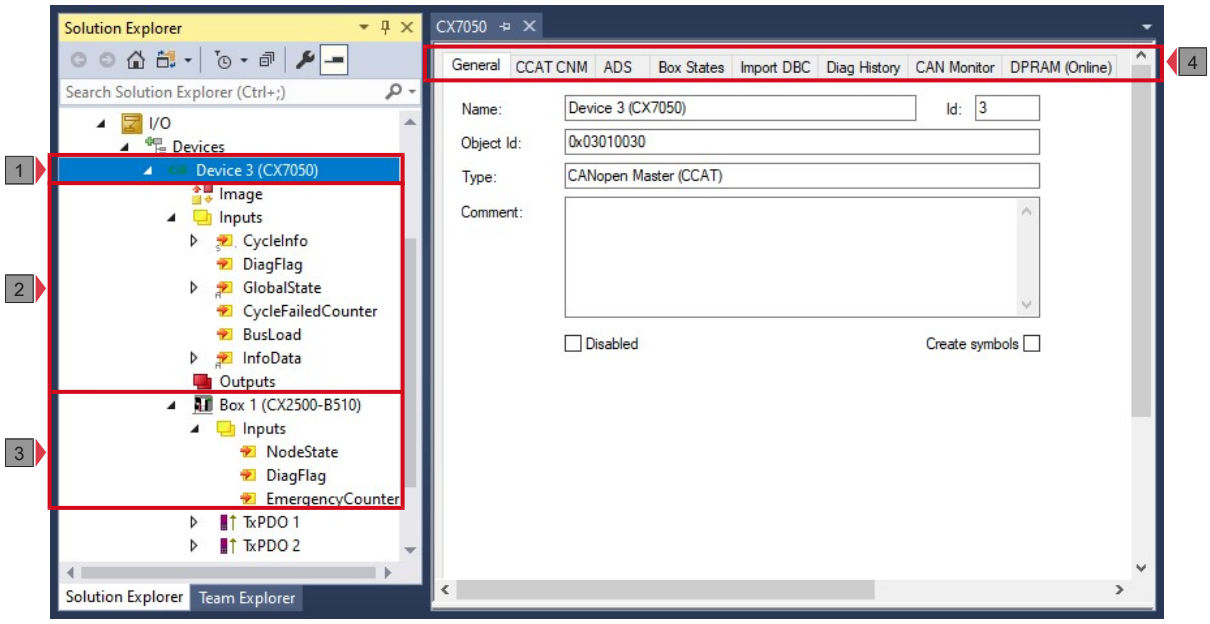


Abb. 38: CANopen-Master und CANopen-Slave in der TwinCAT-Strukturansicht mit Registerkarten.

In diesem Beispiel wurde der Slave mit dem Master verbunden. Anschließend wurde in TwinCAT nach dem Master gescannt und der Master zusammen mit dem Slave in TwinCAT angelegt.

Nr.	Beschreibung
1	Der Gerätename des Masters wird in Klammern angezeigt. Alle CANopen-Slaves werden unter dem Master einsortiert.
2	Unter dem CANopen-Master werden Statusmeldungen als Eingangsvariablen aufgelistet. Die Variablen können mit der SPS verknüpft und für Diagnosezwecke verwendet werden (z.B. Fehlercodes, Zähler usw.).
3	CANopen-Slaves werden unter dem Master einsortiert, als Box bezeichnet und fortlaufend nummeriert. Der Gerätename erscheint in Klammern dahinter. Jeder CANopen-Slave hat eigene Eingangsvariablen für Diagnosezwecke, die den Zustand der Kommunikation anzeigen.
4	Unter den Registerkarten lassen sich weitere Einstellungen für den CANopen-Master oder Slave vornehmen. Abhängig davon, ob der Master oder Slave in der Strukturansicht ausgewählt wird, werden andere Registerkarten angezeigt.

Ein CANopen-Slave und die dazugehörigen Registerkarten werden wie folgt in der Strukturansicht angezeigt:

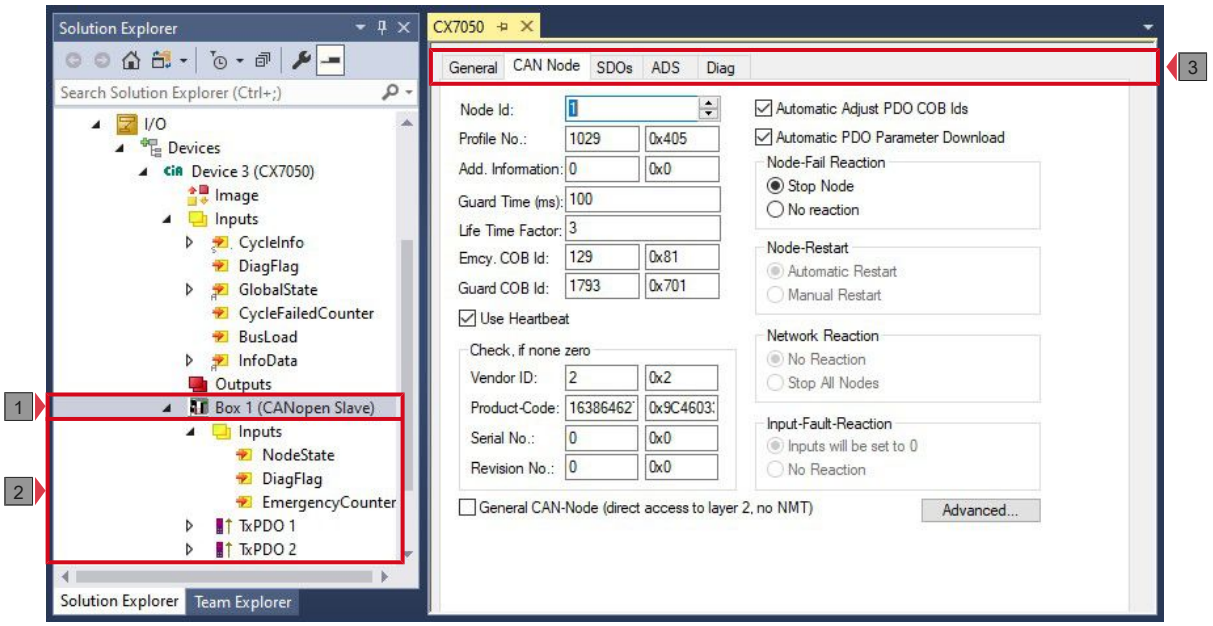


Abb. 39: CANopen-Slave in der TwinCAT-Strukturansicht mit dazugehörigen Registerkarten.

Nr.	Beschreibung
1	Unter dem CANopen-Slave werden Statusmeldungen als Eingangsvariablen aufgelistet. Die Variablen können mit der SPS verknüpft und für Diagnosezwecke verwendet werden.
2	Unter dem CANopen-Slave werden die Prozessdatenobjekte (PDO) angezeigt. An dieser Stelle werden auch die Variablen für den Datentransfer angelegt. Die Variablen können mit der SPS verknüpft werden. Die Datentransferrichtung ist aus Sicht des Slaves beschrieben: • RxPDOs werden vom Teilnehmer empfangen. • TxPDOs werden vom Teilnehmer gesendet.
3	Unter den Registerkarten lassen sich weitere Einstellungen für den CANopen-Slave vornehmen. Abhängig davon, ob der Slave oder andere Einträge in der Strukturansicht ausgewählt werden, werden andere Registerkarten angezeigt.

Wird das SPS-Prozessabbild eingelesen, können die Variablen für Statusmeldungen und die Variablen unter den Prozessdatenobjekten mit den Variablen aus dem SPS-Programm verknüpft werden. Mit einem Doppelklick auf den Variablennamen in der Strukturansicht wird der Verknüpfungsdialog geöffnet. Die verknüpften Variablen werden mit einem kleinen Pfeilsymbol markiert.

Weitere Informationen zu TwinCAT finden Sie in der TwinCAT Dokumentation auf der Beckhoff Homepage: www.beckhoff.de

8.2.2 CANopen-Master

8.2.2.1 General

Die Registerkarte General liefert allgemeine Informationen zu einem CANopen-Gerät, wie Name, Typ und die Id.

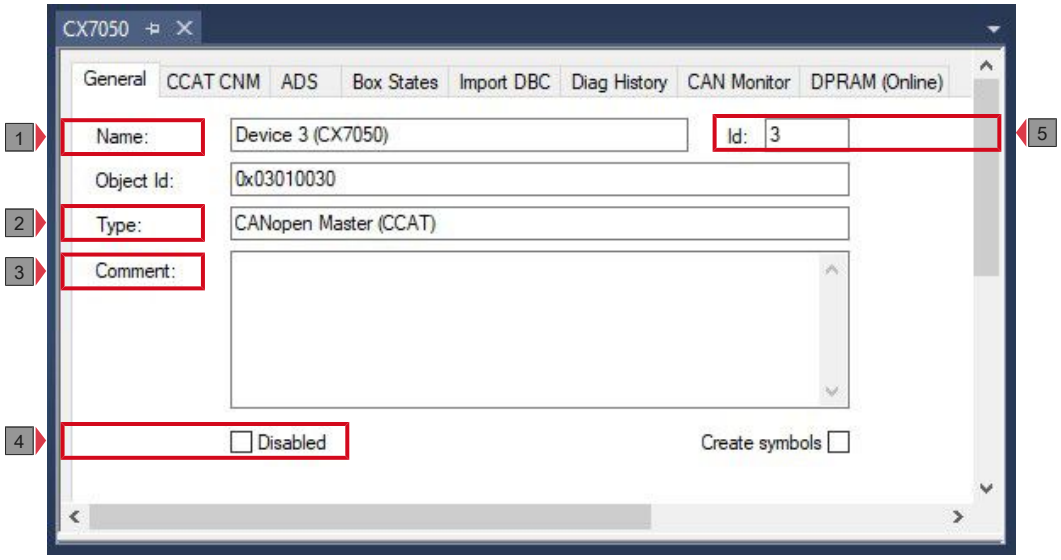


Abb. 40: General-Registerkarte eines CANopen-Masters in TwinCAT.

Nr.	Beschreibung
1	Name des CANopen-Geräts
2	Typ des CANopen-Geräts
3	Hier können Sie einen Kommentar (z.B. Anmerkungen zum Anlagenteil) hinzufügen
4	Hier können Sie das CANopen-Gerät deaktivieren
5	Laufende Nr.

Über diese Registerkarte kann das CANopen-Gerät ausgeschaltet werden. Ein Kommentarfeld bietet die Möglichkeit einer Beschriftung, um auf diese Weise zusätzliche Informationen zu dem Gerät bereitzustellen.

8.2.2.2 CCAT CNM

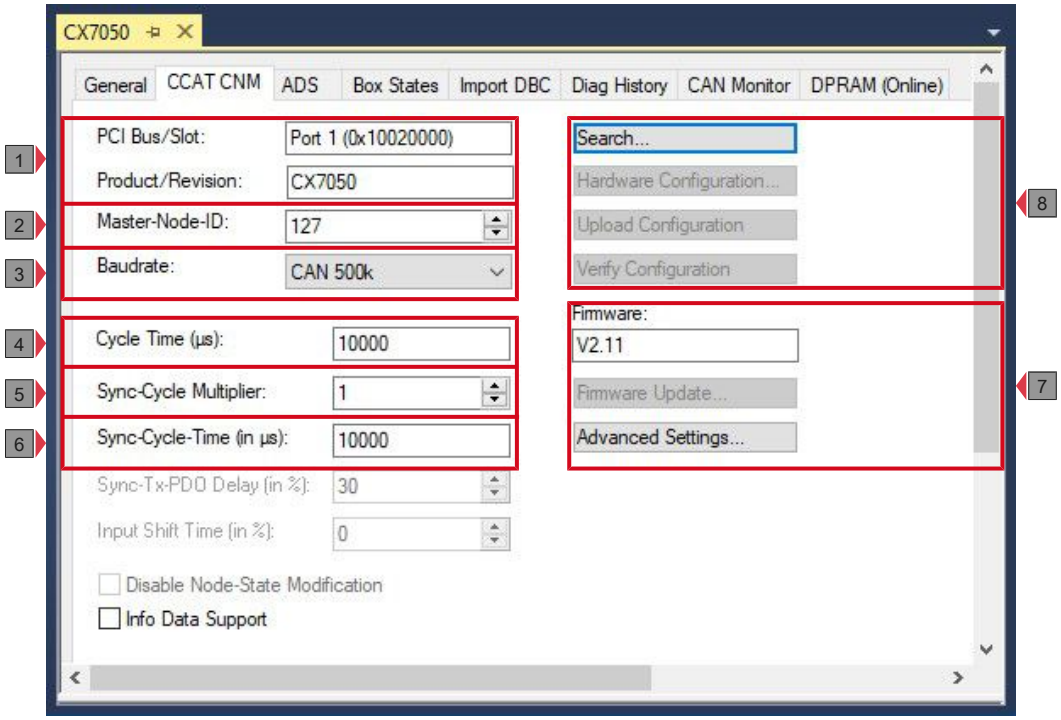


Abb. 41: CCAT-CNM-Registerkarte eines CANopen-Masters in TwinCAT.

Nr.	Beschreibung
1	Bezeichnung der physikalischen Schnittstelle. Name und Typ des CANopen-Gerätes.
2	Adresse des CANopen-Masters. Wertebereich von 1 bis 127. Bestimmt den Identifier des Master-Heartbeat-Telegramms und darf nicht mit einer Slave-Knotenadresse übereinstimmen.
3	Hier wird die Baudrate eingestellt. Es wird automatisch überprüft ob die angeschlossenen Slaves diese Baudrate unterstützen.
4	Hier wird die Zykluszeit der zugehörigen höchstprioren Task angezeigt.
5	Bei CANopen werden häufig eine ereignisgesteuerte Kommunikation und eine zyklisch synchrone Kommunikation kombiniert. Um schnell auf ein Ereignis reagieren zu können, muss die Task Zykluszeit kleiner sein als die Zykluszeit des Sync-Telegramms. Wird der Sync-Cycle Multiplier auf Werte > 1 eingestellt, so wird die TwinCAT Task entsprechend mehrfach aufgerufen, bevor das Sync-Telegramm erneut gesendet wird.
6	Hier wird die Zykluszeit des Sync-Telegramms angezeigt. Sie ergibt sich aus der Zykluszeit der höchstprioren Task, deren Prozessdaten und aus dem Sync-Cycle Multiplier. Cync-Cycle-Time = Cycle Time **x** Sync-Cycle Multiplier
7	An dieser Stelle wird die aktuelle Firmware angezeigt.
8	Über die Schaltfläche **Search** wird nach der physikalischen Schnittstelle gesucht und die gewünschte ausgewählt, wenn nicht bereits automatisch geschehen.

8.2.2.3 ADS

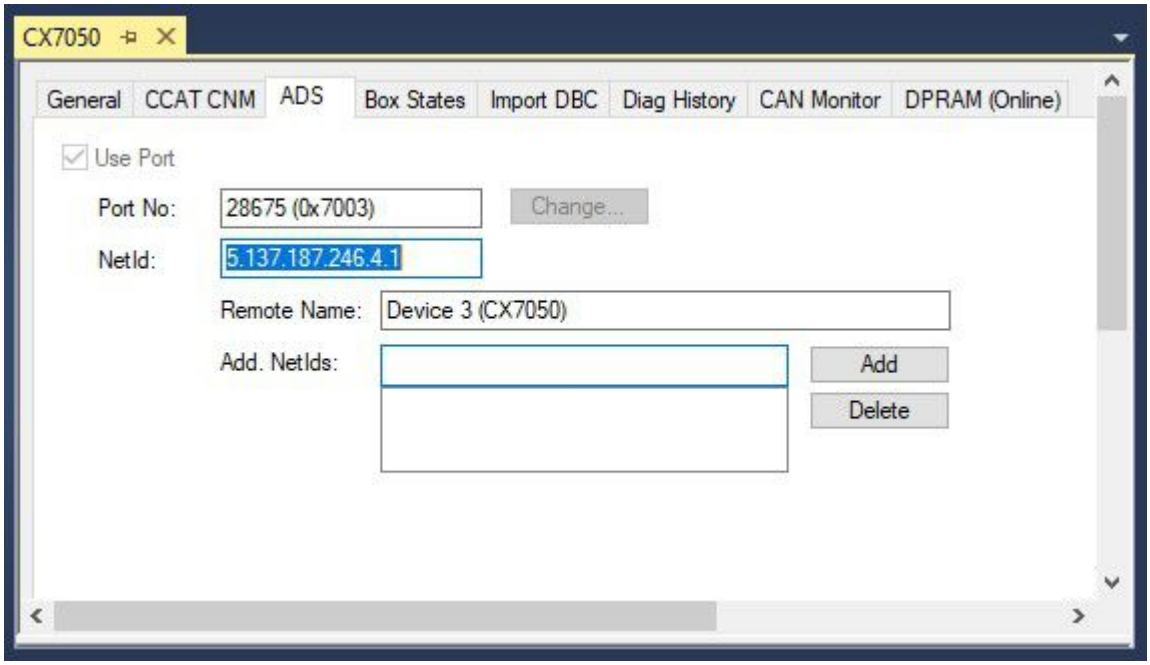


Abb. 42: ADS-Registerkarte eines CANopen-Masters in TwinCAT.

Der CANopen-Master ist ein ADS-Device mit einer eigenen Net-Id, die hier verändert werden kann. Alle ADS-Dienste (Diagnose, azyklische Kommunikation), die an den CANopen-Master gehen, müssen diese Net-Id und Port-Nummer adressieren.

8.2.3 CANopen-Slave

8.2.3.1 CAN Node

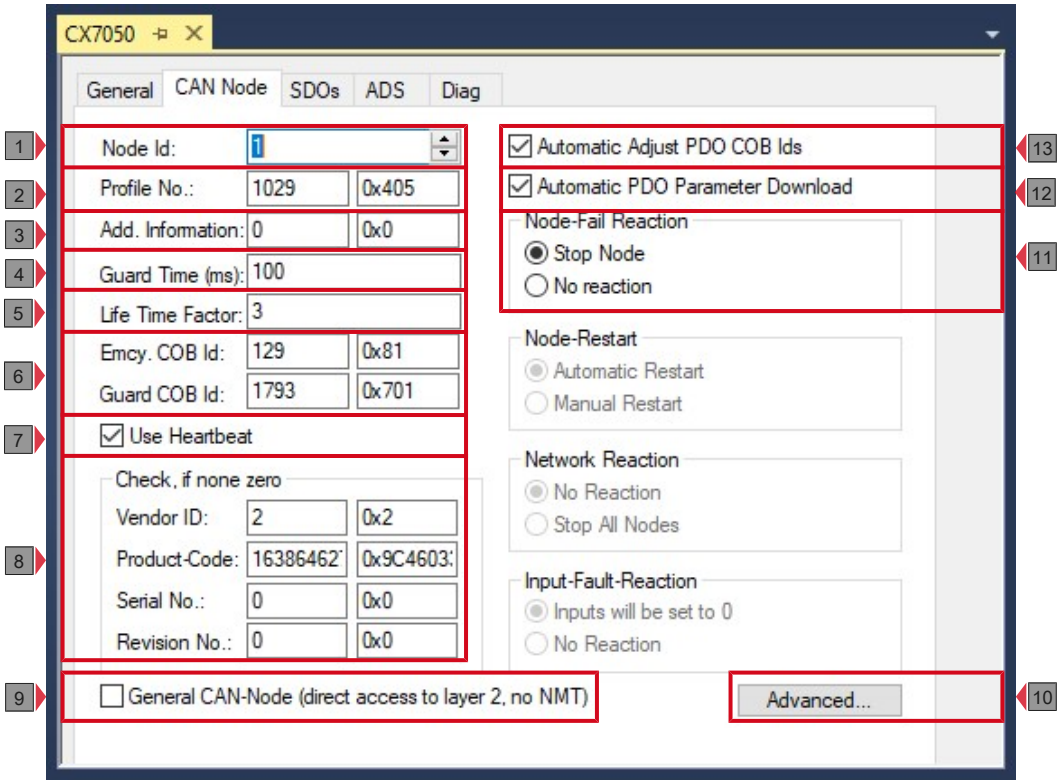


Abb. 43: CAN-Node-Registerkarte eines CANopen-Slaves in TwinCAT.

Nr.	Beschreibung
1	Hier wird die Adresse eingestellt.
2	Nach CANopen enthält der Parameter 0x1000 "Device Type" in den beiden niederwertigsten Bytes die Nummer des vom Gerät unterstützten Geräteprofils. Diese Nummer wird hier eingetragen und beim Systemstart mit dem im Gerät vorhandenen Parameter verglichen. Falls kein Geräteprofil unterstützt wird, so enthält der Parameter den Wert 0.
3	Add. Information:‘ Die Add. Information steht in den beiden höchstwertigen Bytes des Objektverzeichniseintrages 0x1000 (Device Type). Der Vergleich Soll-/ Ist-Konfiguration erfolgt nur, wenn Profile No. oder Add. Information (also Objektverzeichniseintrag 0x1000) auf Wert ungleich null konfiguriert sind. Falls die erwarteten Werte beim Systemstart nicht mit den vorhandenen übereinstimmen, wird der Start des Knotens abgebrochen und eine entsprechende Fehlermeldung auf der Registerkarte Diag angezeigt.
4	Guard Time: Die Guard Time bestimmt das Intervall, in dem der Knoten überwacht wird (Node Guarding). Der eingetragene Wert wird auf das nächste Vielfache von 10ms aufgerundet. 0 bedeutet keine Überwachung.
5	Life Time Factor: Guard Time x Life Time Factor bestimmt die Watchdog-Länge für die gegenseitige Überwachung von Master und Slave. Der Eintrag 0 bedeutet, dass der Slave den Master nicht überwacht. Bei 0 nimmt der Master die Guard Time direkt als Watchdog-Länge. Es wird auch das Heartbeat-Protokoll unterstützt und es wird versucht zunächst diese Form der Knotenüberwachung auf dem CANopen-Knoten zu starten. Falls dieser Versuch fehlschlägt, wird Guarding aktiviert.

Nr.	Beschreibung
6	Die Emcy COB Id / Guard COB ID sind Identifier für Emergency Nachrichten bzw. das Guarding Protocol. Diese ergeben sich aus der Knotenadresse.
7	Zur Überwachung des Knoten wird Heartbeat verwendet. Ist Heartbeat deaktiviert, wird das Guarding zur Überwachung verwendet. Eingetragen werden die Guard Time als Producer Heartbeat Time und (Guard Time x Life Time Factor) als Consumer Heartbeat Time. In diesem Fall wird ein Heartbeat Telegramm mit der kleinsten konfigurierten Guard Time gesendet. Die Guard Time kann für jeden Knoten individuell eingestellt werden.
8	Falls hier Werte ungleich null eingetragen sind, so werden diese Einträge des Identity Objektes (0x1018 im Objektverzeichnis) beim Systemstart ausgelesen und mit den konfigurierten Werten verglichen. Nur wenn die Werte übereinstimmen, wird der entsprechende Knoten gestartet. Es ist auch möglich, nur einen Teil der Werte (z.B. die Vendor ID und den Product Code) zu vergleichen – nicht gewünschte Parameter müssen dann auf null gesetzt werden.
9	Wenn diese Option angewählt ist, ist das gesamte CANopen Netzwerkmanagement für diesen Teilnehmer deaktiviert. Er wird nicht gestartet, überwacht usw. Die PDO-Einträge werden als reine CAN-Telegramme (Schicht 2) aufgefasst und ereignisgesteuert der Steuerung zur Verfügung gestellt.
10	Öffnet ein Fenster mit weiteren Einstellungen, die aktiviert werden können: • Upload Objekt 0x1000 ausschalten. • Download Objekt 0x1006 ausschalten. • Automatisches Senden von Start Node ausschalten (muss dann manuell gesendet werden.) • Start SDOs weitersenden, im Falle eines Abbruchs.
11	Mit der Option StopNode wird der Knoten nach einem Fehler in den "Stopped" Zustand versetzt. Damit können Knoten in einen sicheren Zustand versetzt, aber nicht mehr über SDO angesprochen werden.
12	Wenn die Option angewählt ist, werden in TwinCAT automatisch Einträge angelegt, die beim Systemstart über SDO übertragen werden (siehe: Registerkarte SDOs [▶ 150]).
13	Wenn Option angewählt ist, werden die Default-Identifier der Prozessdatenobjekte bei Änderung der Node-ID entsprechend nachgeführt (siehe: Nr. 6).

8.2.3.2 SDOs

Auf der SDO-Registerkarte werden Einträge angezeigt und verwaltet, die beim Startup zum Knoten geschickt werden.

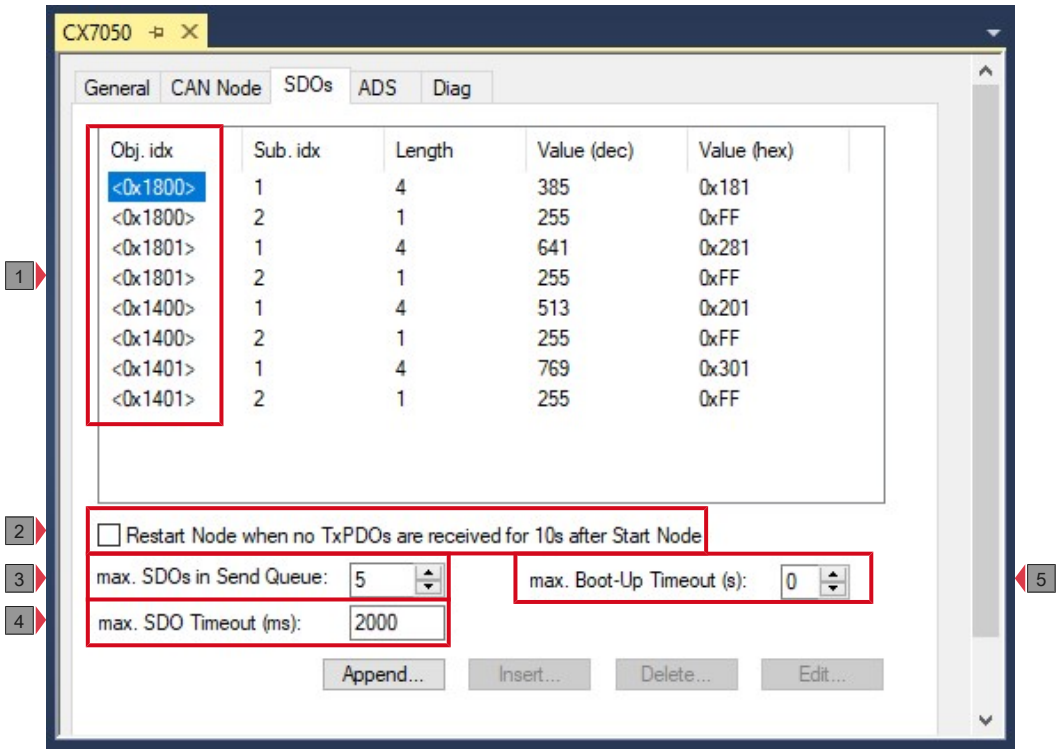


Abb. 44: SDO-Registerkarte eines CANopen-Slaves in TwinCAT.

Nr.	Beschreibung
1	Objektindexeinträge, die in spitzen Klammern stehen, sind automatisch aufgrund der aktuellen Konfiguration erzeugt worden. Weitere Einträge können über "Append", "Insert", Delete" und "Edit" erzeugt und verwaltet werden.
2	Wird diese Option angewählt, wird der Slave nach dem Start neu gestartet, wenn kein TxPDO nach 10 Sekunden empfangen wurde.
3	Über diese Option lässt sich die maximale Anzahl von SDOs in der Sende-Warteschlange einstellen.
4	Hier wird das maximale Timeout (ms) für die SDO eingestellt.
5	An dieser Stelle wird das Boot-Up Timeout (s) eingestellt.

8.2.3.3 PDO

Diese Registerkarte erscheint, wenn Sie in der Strukturansicht auf ein Prozessdatenobjekt (PDO) klicken.

Prozessdatenobjekte (PDOs) sind CAN-Telegramme die Prozessdaten transportieren.

- RxPDOs werden vom Teilnehmer empfangen.
- TxPDOs werden vom Teilnehmer gesendet.

Ein Teilnehmer sendet seine Eingangsdaten mit TxPDOs, und empfängt die Ausgangsdaten in den RxPDOs. Diese Bezeichnung wird in TwinCAT beibehalten.

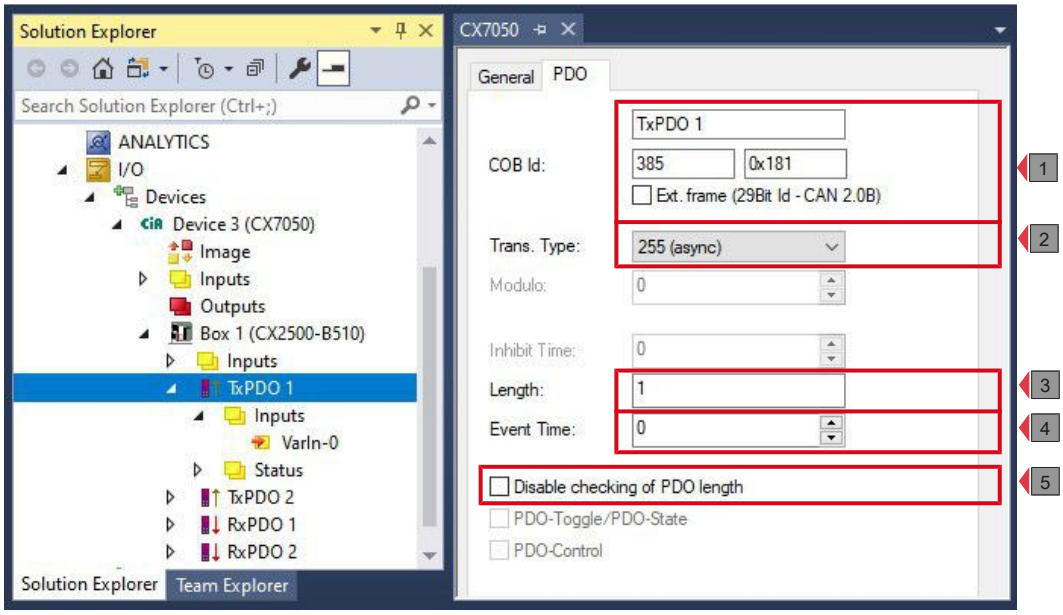


Abb. 45: PDO-Registerkarte eines CANopen-Slaves in TwinCAT.

Nr.	Beschreibung
1	CAN-Identifier des PDOs. Für zwei Sende- und Empfangs-PDOs je Knoten stellt CANopen Default-Identifier zur Verfügung. Diese können dann geändert werden.
2	Der Transmission Type bestimmt das Sendeverhalten des PDOs. 255 entspricht dem ereignisgesteuerten Senden (siehe: Übertragungsart festlegen).
3	Die Länge des PDOs hängt von den angelegten Variablen ab und kann hier daher nicht editiert werden.
4	Hier wird der Wert für den Event Timer in ms eingetragen. Bei Sende-PDOs (RxPDOs) werden nach einem abgelaufenen Timer erneut PDOs gesendet. Bei Empfangs-PDOs (TxPDOs) werden die eingetroffenen PDOs überwacht und ggf. der Box-State des Knotens verändert. TwinCAT erzeugt aus den hier eingegebenen Parametern entsprechende Einträge im Objektverzeichnis des Knotens, die beim Systemstart über SDO übertragen werden. Die Einträge können auf der Registerkarte SDO eingesehen werden (siehe: SDOs [▶ 150]). Diese Funktion kann über das Kontrollkästchen **Automatic PDO Parameter Download** auf der Registerkarte **CAN Node** deaktiviert werden (siehe: CAN Node [▶ 148]).
5	Hier kann die Längenprüfung der PDOs deaktiviert werden.

8.3 CX7050 als Master anlegen

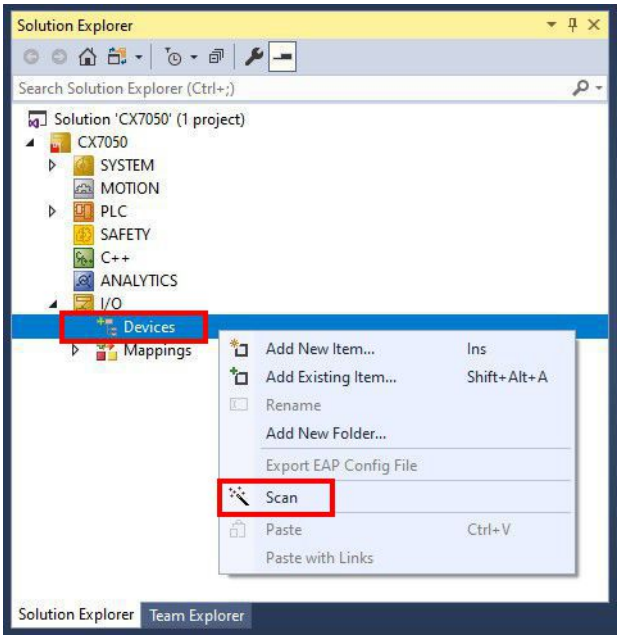
Wenn der CX7050 als CANopen-Master angelegt werden soll, kann in TwinCAT nach dem Gerät gescannt und zusätzlich alle darin angeschlossenen Slaves automatisch angelegt werden. Im Folgenden wird gezeigt, wie Sie einen CANopen-Master in TwinCAT anlegen.

Voraussetzungen für diesen Arbeitsschritt:

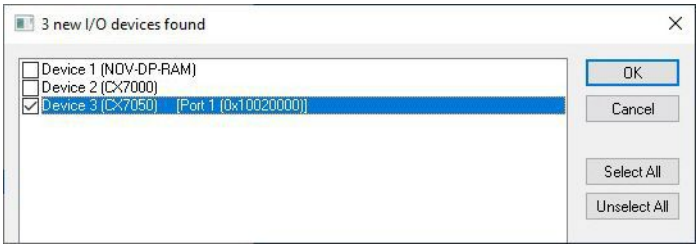
- TwinCAT muss sich im Config-Mode befinden.
- Ein ausgewähltes Zielsystem.

Legen Sie ein CANopen-Gerät wie folgt an:

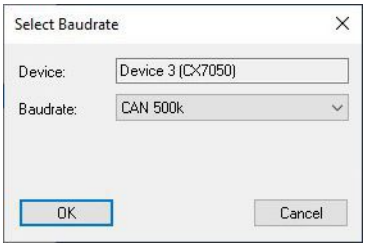
1. Klicken Sie links in der Strukturansicht mit rechter Maustaste auf **Devices**.
2. Klicken Sie im Kontextmenü auf **Scan**.



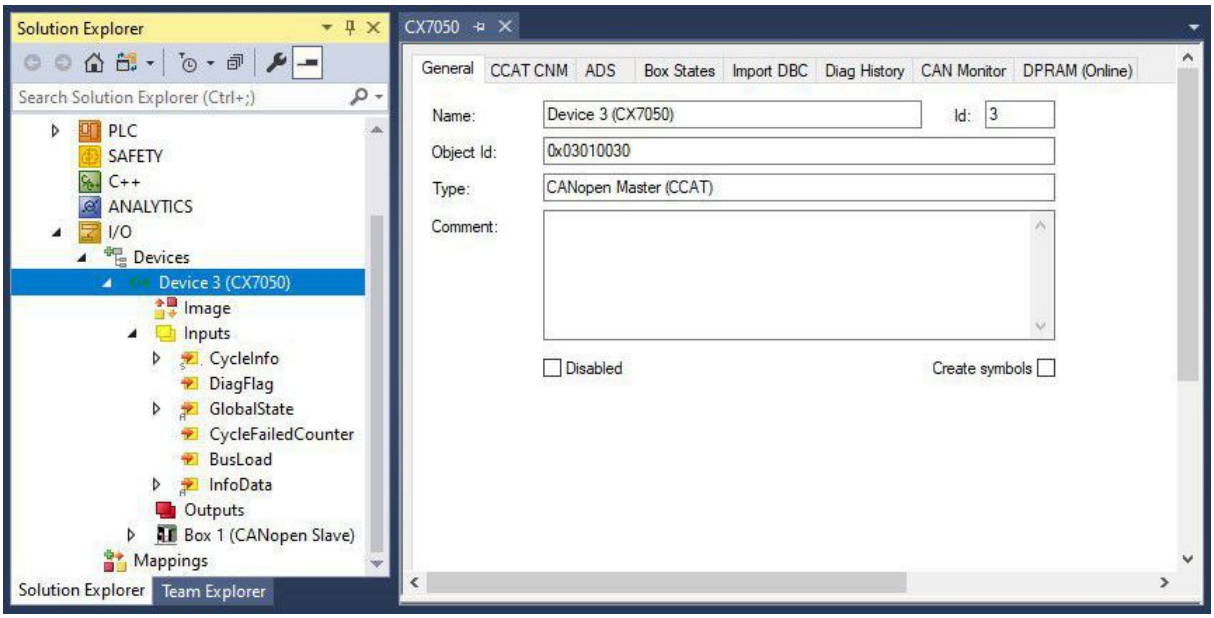
3. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.



4. Bestätigen Sie die Anfrage mit Ja, um nach Boxen zu suchen.
   Das Fenster **Select Baudrate** erscheint.
5. Wählen Sie unter Baudrate die passende Baudrate für den CANopen-Master.



⇨ Alle gefundenen Geräte und Slave Boxen werden links in der Strukturansicht angezeigt. Auch die Busklemmen, die an den Geräten oder Slave Boxen angeschlossen sind, werden angezeigt.



Wiederholen Sie die Arbeitsschritte, wenn nicht alle Geräte angezeigt werden. Sollten Sie trotz Wiederholung nicht alle Geräte und Slave Boxen finden, müssen Sie die Verkabelung der Geräte und Slave Boxen überprüfen.

8.3.1 SDO-Kommunikation aus der SPS

Für die SDO-Kommunikation aus der SPS heraus verwendet man die ADS-Bausteine. Mit diesen Bausteinen ist es möglich, SDO-Telegramme zu versenden und die Antwort des Slaves zu empfangen (ADSWRITE/ADSREAD).

Eingangsparameter	Beschreibung
NETID	ADS NetId des CAN Interface
Port Nummer	$0x1000_{hex}$ + NodeId (Slave Nummer)
IDXGRP	SDO-Index
IDXOFFS	SDO Subindex
LEN	Länge der SDO-Daten (1...4)

Manuelles Netzwerkmanagement

Der CANopen Status (STOPPED, PRE-OPERATIONAL, OPERATIONAL) eines CANopen-Slaves kann per ADS Write Control verändert werden. Dabei ist die AMS-Adresse wie bei der SDO-Kommunikation einzustellen, die anderen Parameter ergeben sich anhand der folgenden Tabelle:

ADS State	Device State	CANopen state transition
ADSSTATE_RUN (5)	0	OP->PREOP
ADSSTATE_RUN (5)	1	PREOP->OP
ADSSTATE_STOP (6)	0	OP->STOP
ADSSTATE_RUN (5)	1	STOP->OP (mit Reset Communication)
ADSSTATE_RUN (5)	3	STOP->OP (ohne Reset Communication)
ADSSTATE_STOP (6)	0	PREOP->STOP
ADSSTATE_RUN (5)	2	STOP->PREOP (ohne Reset Communication)

CAN-Interface neu Starten

Mit dem ADSWRTCTL Baustein kann das CAN-Interface gestoppt und neu gestartet werden. Führen Sie als erstes ein Stopp aus und als nächstes einen Start aus.

Eingangsparameter	Beschreibung
NETID	ADS NetId des CAN Interface
Port Nummer	200_{dez}
ADSSTATE	ADSSTATE_STOP, ADSSTATE_RUN
DEVSTATE	0
LEN	0
SRCADDR	0

8.3.2 CAN-Interface

Fast alle CANopen-Master von Beckhoff bieten das sogenannte CAN-Interface als Schnittstelle an. Das CAN-Interface ist eine Layer-2-Implementierung der CAN-Schnittstelle. Das ermöglicht beliebige CAN-Telegramme zu empfangen wie auch zu senden. Hierbei spielt das verwendete überlagerte Protokoll keine Rolle, d.h. es können damit alle CAN-basierenden Protokolle verwendet werden, wobei der Protokoll-Teil dann in der SPS realisiert werden muss.

Eine detaillierte Beschreibung zum CAN-Interface finde Sie unter:
https://download.beckhoff.com/download/document/io/infrastructure-components/can-interface_en.pdf

Das CAN-Interface erlaubt beliebige CAN-Daten zu verschicken. Es kann zwischen 11 Bit Identifier (CAN 2.0A) und dem 29 Bit Identifier (CAN 2.0B) ausgewählt werden.

Message-Struktur mit 29-Bit Unterstützung

- Length (0..8)

- CobId
  - Bit 0-28: 11 Bit-Identifier / 29 Bit-Identifier
  - Bit 30: RTR
  - Bit 31: 0: normal Message (11 Bit Identifier), 1: extended Message (29 Bit-Identifier)
- Data[8]

Daten senden: Tragen Sie in "NoOfTxMessabes die Anzahl der Daten ein, die Sie aus dem Tx Buffer senden wollen. Ist der Buffer 10 Einträge groß können Sie maximal 10 Telegramme hintereinander verschicken. Mit "Length" wir die Anzahl der Datenbytes des PDOs definiert (maximal 8 Byte). Füllen Sie die Daten und tragen die den Identifire des CAN-Message in "codId" ein. Nun inkrementieren Sie den Wert TxCounter.

Beispielcode: Senden von Messages von der SPS

```
if Outputs.TxCounter = Inputs.TxCounter then
     for i=0 to NumberOfMessagesToSend do
          Outputs.TxMessage[i] = MessageToSend[i];
     End_for
     Outputs.NoOfTxMessages = NumberOfMessagesToSend;
     Outputs.TxCounter := Outputs.TxCounter + 1;
end_if
```

Beispielcode: Empfangen von Messages von der SPS

```
if Outputs.RxCounter <> Inputs.RxCounter then
     for i := 0 to (Inputs.NoOfRxMessages-1) do
          MessageReceived[i] := Inputs.RxMessage [i];
     End_for
     Outputs.RxCounter := Outputs.RxCounter+1;
end_if
```

8.4 CX705x als Slave anlegen

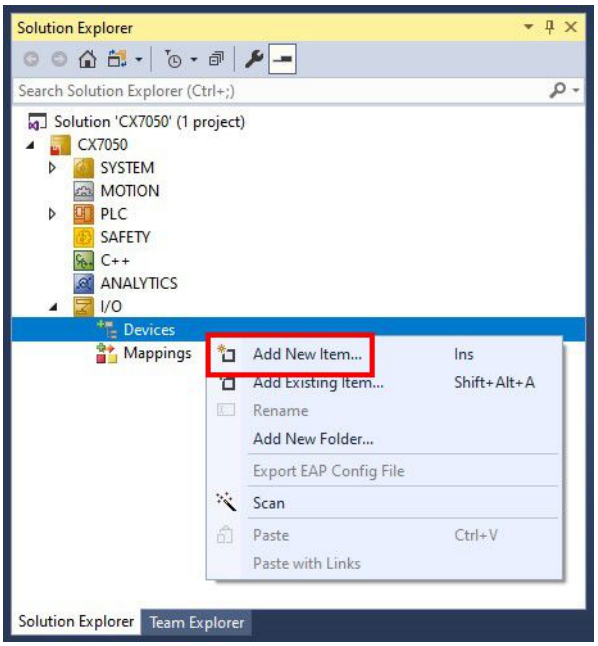
In diesem Abschnitt wird gezeigt, wie Sie einen CX7050 als CANopen-Slave anlegen können. Damit der CANopen-Slave später von einem CANopen-Master mit allen Ein- und Ausgängen erkannt wird, muss der CANopen-Slave zuerst in TwinCAT angelegt und mit allen dazugehörigen PDOs und Variablen konfiguriert werden.

Voraussetzungen für diesen Arbeitsschritt:

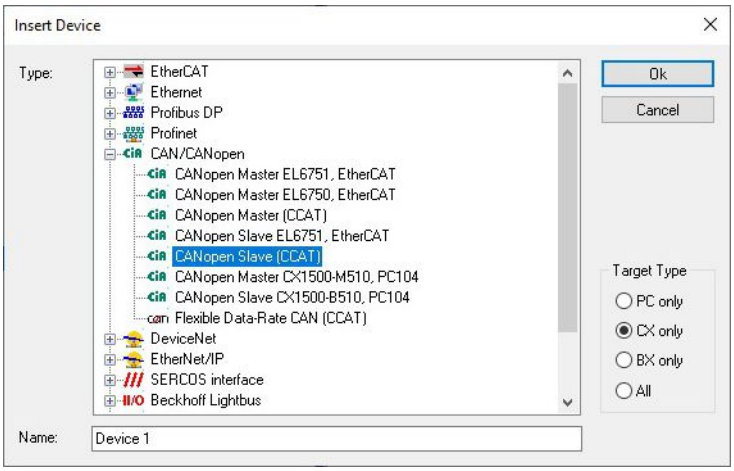
- CX7050 als Zielgerät ausgewählt.

Legen Sie den CANopen-Slave wie folgt an:

1. Klicken Sie links in der Strukturansicht mit rechter Maustaste auf **Devices**.
2. Klicken Sie im Kontextmenü auf **Add New Item**.

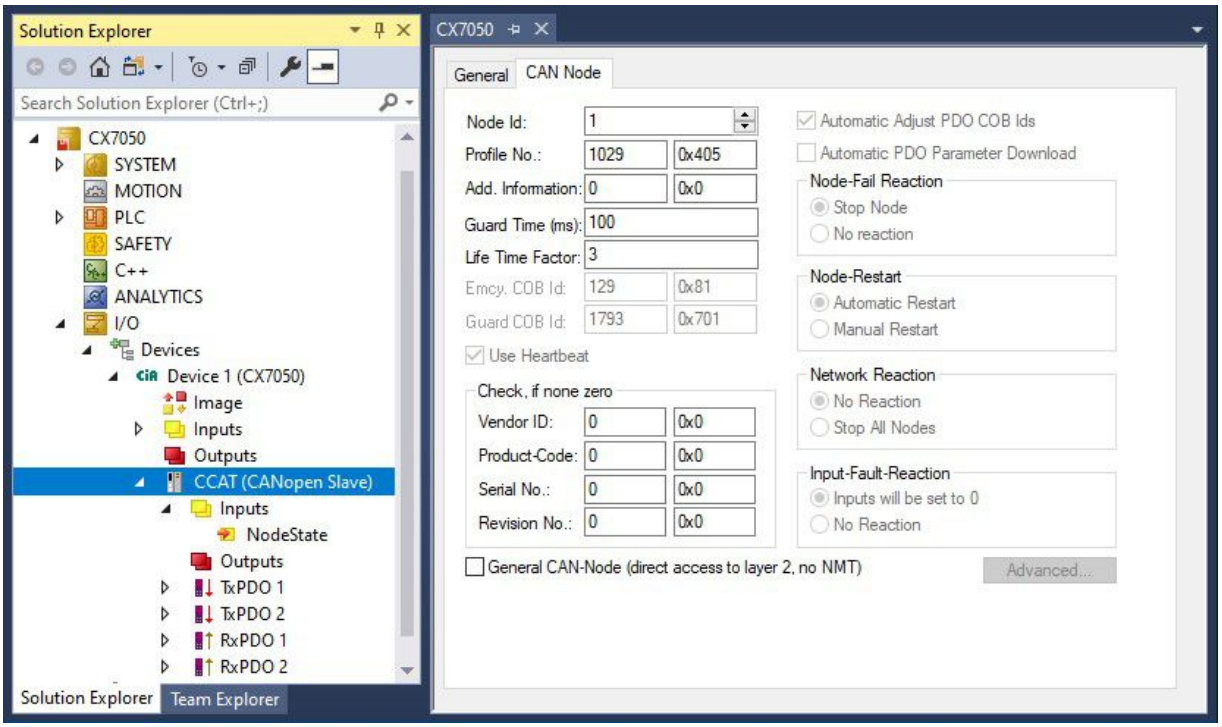


3. Wählen Sie als Gerät **CANopen Slave (CCAT)** und bestätigen die Auswahl mit **OK**.



4. Bestätigen Sie die Anfrage mit Ja, um nach Boxen zu suchen.

⇨ Der CANopen-Slave wurde erfolgreich in TwinCAT angefügt und wird in der Strukturansicht mit den Ein- und Ausgängen angezeigt.



Im nächsten Schritt können Sie das Prozessabbild erweitern, indem Sie zusätzliche virtuelle Slaves anlegen. Oder Sie können die Adresse einstellen, wenn Sie den Slave fertig konfiguriert haben.

8.4.1 Virtuelle Slaves anlegen

Es können zusätzliche virtuelle Slaves auf der gleichen Hardwareschnittstelle angelegt werden. Dadurch können mehr Daten mit einem CANopen-Master ausgetauscht oder eine Verbindung mit einem zweiten CANopen-Master angelegt werden. Es können bis zu drei virtuelle Slaves auf der gleichen Hardwareschnittstelle eines Slaves angelegt werden.

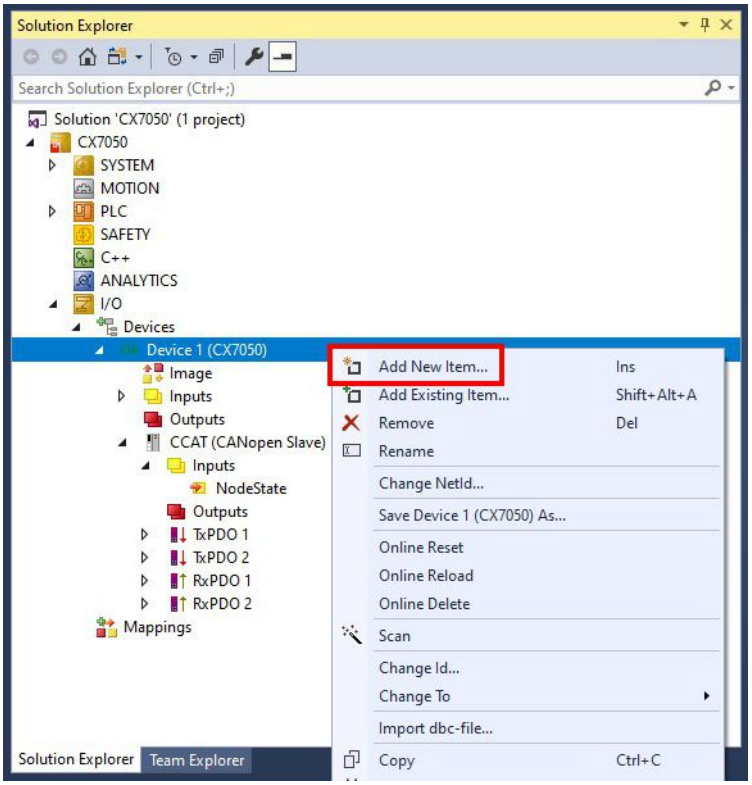
Weil für jeden Slave maximal 16 PDOs konfiguriert werden können, erhöht sich mit den zusätzlichen drei virtuellen Slaves die maximal mögliche Anzahl der PDOs auf 4 x 16 PDOs in jede Senderichtung. Jeder virtuelle Slave bekommt über TwinCAT eine eigene Adresse und wird für den CANopen-Master wie ein eigenständiges Gerät konfiguriert.

Voraussetzungen für diesen Arbeitsschritt:

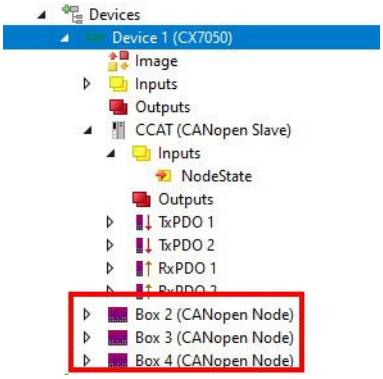
- Ein CANopen-Slave angelegt in TwinCAT.

Legen Sie einen virtuellen Slave wie folgt an:

1. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf einen CANopen-Slave.
2. Klicken Sie im Kontextmenü auf **Add New Item**.



⇨ Eine weitere Box (virtueller Slave) wird angelegt.



Für den virtuellen Slave können jetzt eigene Variablen angelegt werden. Im nächsten Schritt können Sie die Adresse für den Slave einstellen.

8.4.2 Adresse einstellen

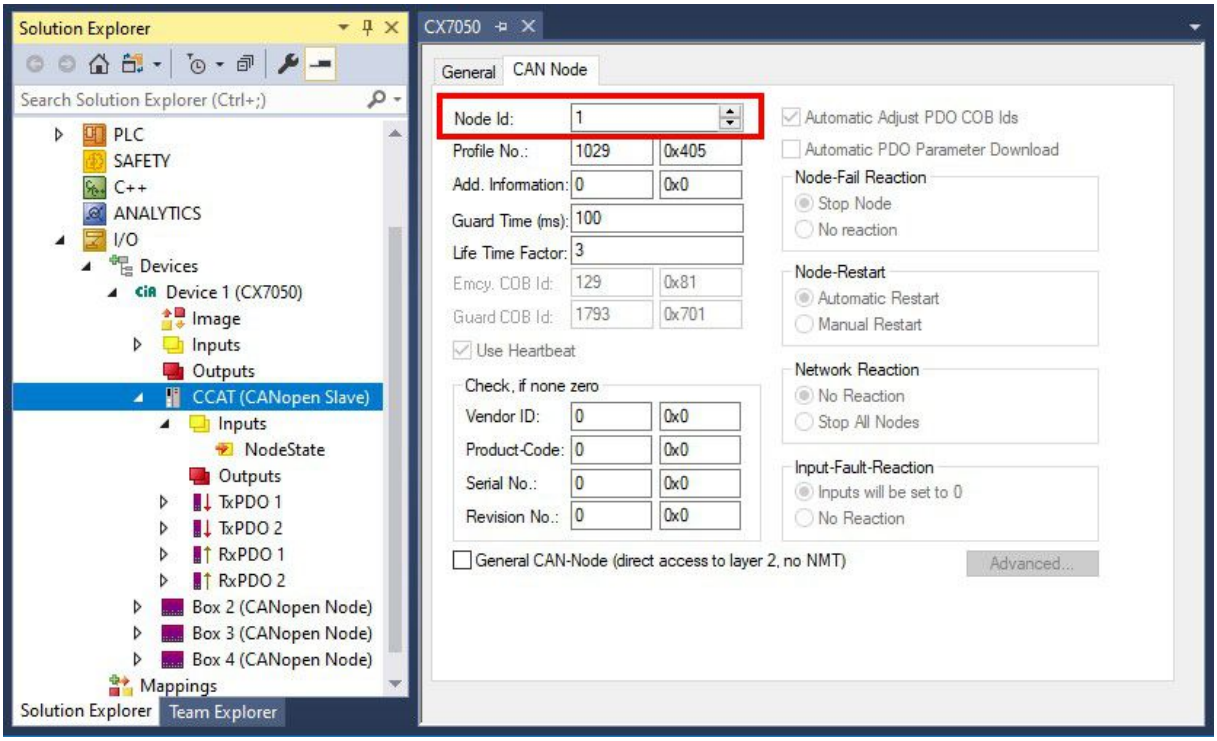
Nachdem der CANopen-Slave erfolgreich in TwinCAT angefügt wurde, kann die Adresse des CANopen-Slaves eingestellt werden. In diesem Arbeitsschritt wird gezeigt, wie die Adresse in TwinCAT eingestellt wird, damit der CANopen-Slave über diese Adresse für den CANopen-Master erreichbar ist.

Voraussetzungen für diesen Arbeitsschritt:

- Ein eingefügter CANopen-Slave in TwinCAT.

Parametrieren Sie den CANopen-Slave wie folgt:

1. Klicken Sie auf eine Slave Box.
2. Klicken Sie auf die Registerkarte **CAN Node**.
3. Tippen Sie im Feld **Node Id** einen Wert für die CANopen-Adresse ein, z.B. „1“.



⇨ Sie haben erfolgreich die Adresse eingestellt. Mit der eingestellten Adresse ist der CANopen-Slave für den CANopen-Master erreichbar. Als nächstes können Sie weitere PDOs anlegen.

8.4.3 Weitere PDOs anlegen

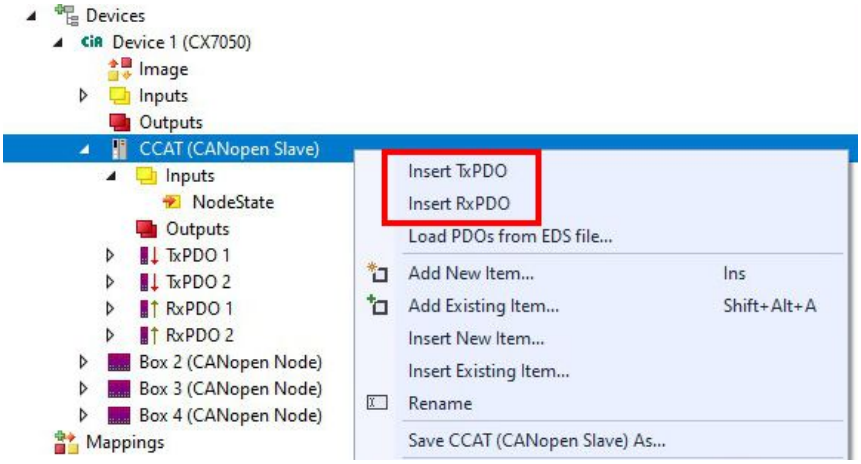
Der CANopen-Slave kann bis zu 16 PDOs mit jeweils 8 Byte Prozessdaten in Eingangs- und Ausgangsrichtung mit dem CANopen-Master austauschen. Standartmäßig werden 2 PDOs in Tx- und Rx-Richtung angelegt. An dieser Stelle wird gezeigt, wie bei einem CANopen-Slave weitere PDOs angelegt werden.

Voraussetzungen für diesen Arbeitsschritt:

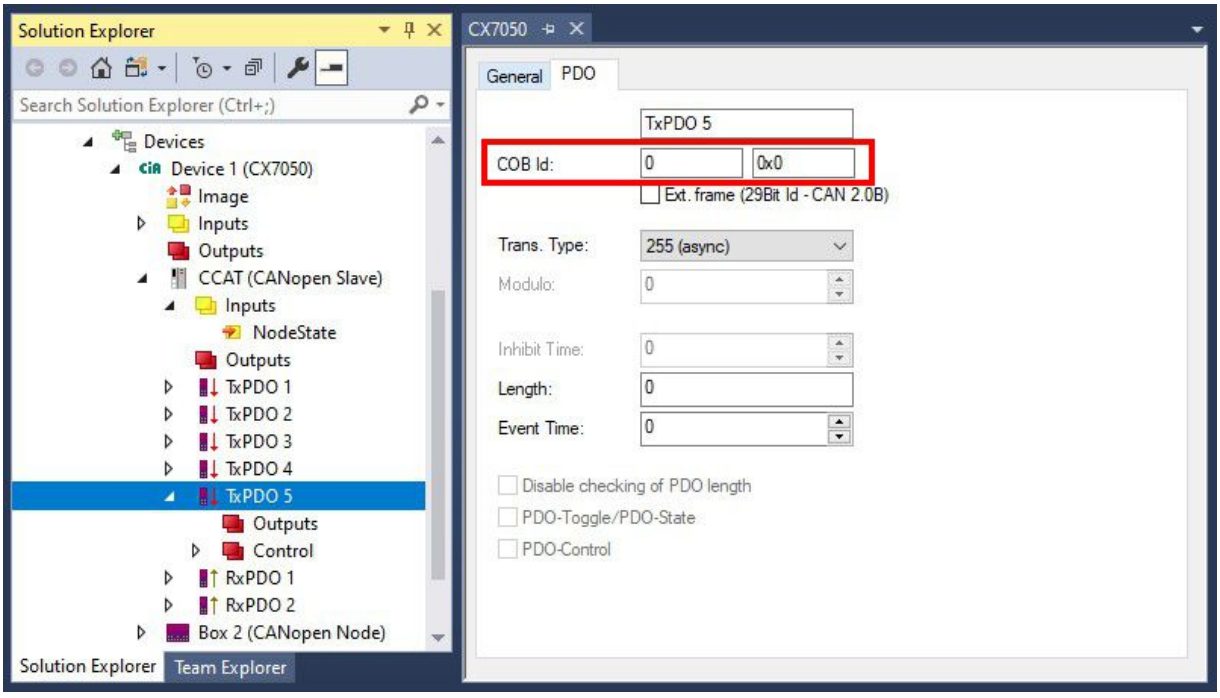
- Ein CANopen-Slave angefügt in der Strukturansicht.

Legen Sie die PDOs wie folgt an:

1. Klicken Sie in der Strukturansicht mit der rechten Maustaste auf einen CANopen-Slave.
2. Klicken Sie im Kontextmenü auf **Insert TxPDO** oder **Insert RxPDO**, um PDOs in Tx- oder Rx-Richtung anzulegen.



Die neuen TxPDOs oder RxPDOs werden unter den bereits angelegten PDOs eingefügt und in der Strukturansicht fortlaufend nummeriert. ***Hinweis*** Ab dem fünften PDO in Tx- oder Rx-Richtung wird die COB Id nicht mehr automatisch eingetragen (siehe folgendes Bild).



3. Klicken Sie ab dem fünften PDO in Tx- oder Rx-Richtung auf die Registerkarte **PDO**.
4. Tippen Sie im Feld **COB Id** den gewünschten Wert ein.

⇨ Sie haben erfolgreich weitere PDOs angelegt und können im nächsten Schritt Variablen unter den PDOs für den Datenaustausch anlegen.

8.4.4 Variablen anlegen

Die PDOs werden in TwinCAT mit Variablen gefüllt, die später mit dem SPS-Programm verknüpft werden können. Unter den entsprechenden PDOs können maximal 8 Byte Daten anlegen werden. Es ist auch erlaubt verschiedene Variablentypen zu verwenden, einzig die Grenze von 8 Byte pro PDO muss eingehalten werden.

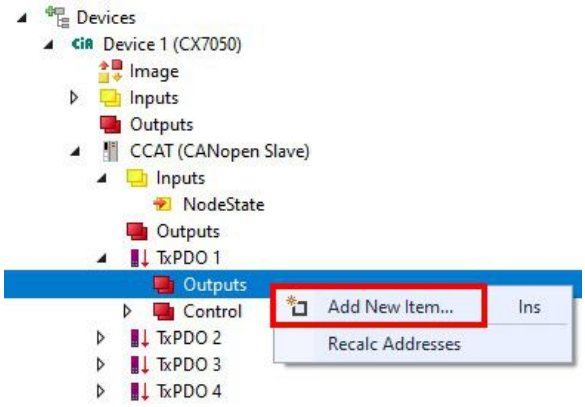
Wenn es im Master nicht anders konfiguriert ist, werden die Daten automatisch bei jeder Änderung verschickt. Achten Sie also schon bei der Planung darauf, dass sich die Daten in einem PDO bedingt ändern und nicht im ms Takt. Wird dies nicht beachtet, kann der CAN überlastet werden, was gerade bei kleinen Baudraten schnell passieren kann.

Voraussetzungen für diesen Arbeitsschritt:

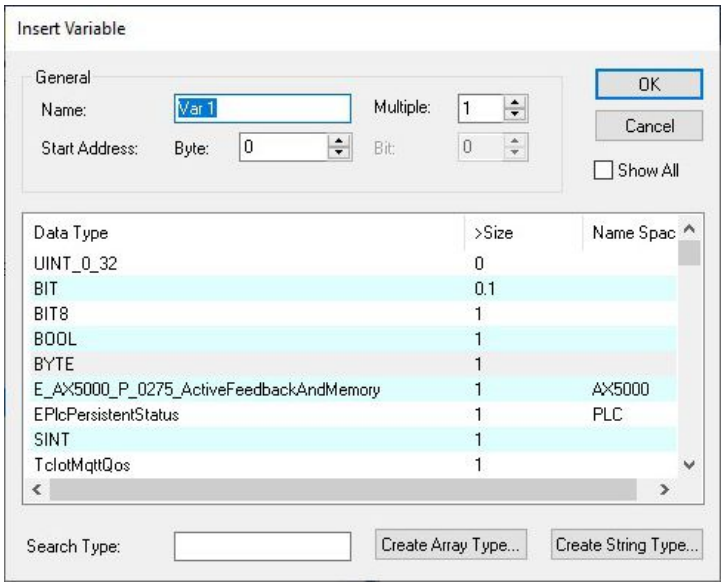
- Fertig angelegte PDOs, die mit Variablen gefüllt werden sollen.

Legen Sie die Variablen wie folgt an:

1. Klicken Sie in der Strukturansicht auf ein TxPDO oder RxPDO, um mehr Informationen einzublenden.
2. Klicken Sie mit der rechten Maustaste auf Outputs oder Inputs, je nachdem ob Sie ein TxPDO oder RxPDO ausgewählt haben.



3. Klicken Sie im Kontextmenü auf **Add New Item**.
   Das Fenster **Insert Variable** erscheint.
4. Klicken Sie auf die passende Variable und klicken Sie auf **OK**.



⇨ Sie haben erfolgreich Variablen angelegt. Die neue Variable wird links in der Strukturansicht angezeigt. Auf diese Weise können Sie weitere Variablen für den CANopen-Slave anfügen. Im nächsten Schritt können Sie die Übertragungsart bestimmen und damit festlegen, wie die Prozessdatenobjekte übertragen werden.

8.4.5 Übertragungsart festlegen

Die Übertragungsart legt fest, wie die Prozessdatenobjekte übertragen werden. Die Übertragungsart wird für die RxPDOs und TxPDOs auf der Registerkarte PDO eingestellt.

Die Übertragungsarten azyklisch synchron, zyklisch synchron und asynchron stehen dabei zur Verfügung.

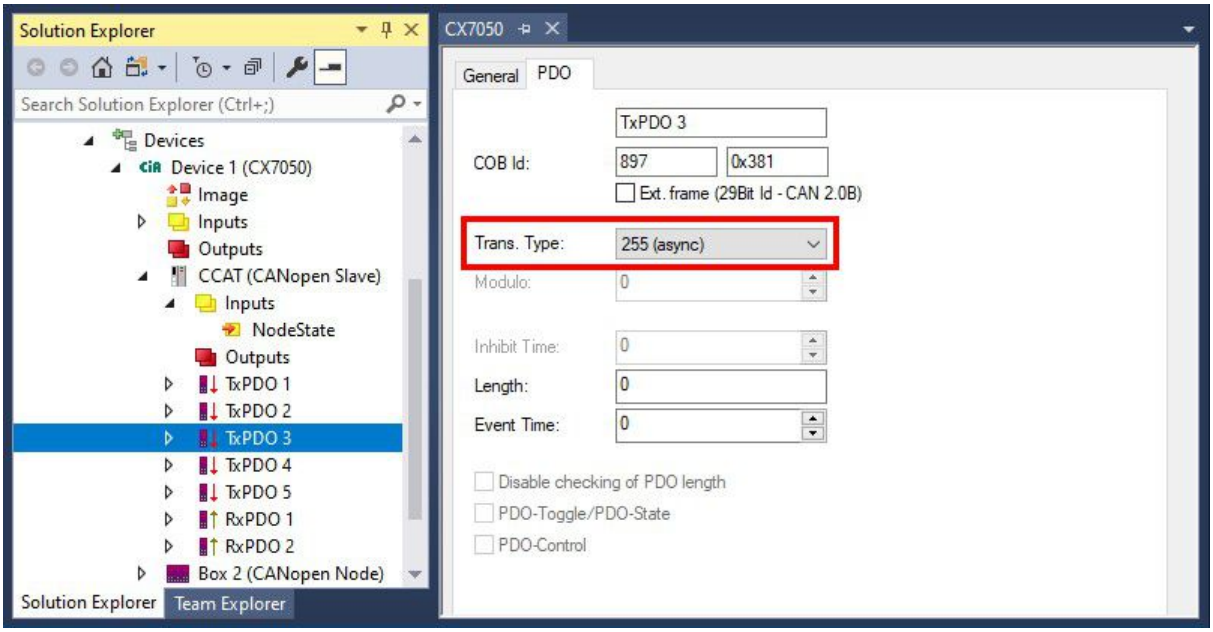
Übertragungsart:	Azyklisch synchron	Zyklisch synchron	Asynchron
Bezeichnung in TwinCAT:	(acyc, sync)	(cyc, sync)	(async)

Voraussetzungen für diesen Arbeitsschritt:

- CANopen-Slave mit Protzessdatenobjekten (PDO) angefügt in TwinCAT

Legen Sie die Übertragungsart wie folgt fest:

1. Klicken Sie links in der Strukturansicht auf ein Protzessdatenobjekt (PDO).
2. Klicken Sie auf die Registerkarte **PDO**.
3. Wählen Sie unter **Trans. Type** die passende Übertragungsart.



⇨ Sie haben erfolgreich eine Übertragungsart für ein Prozessdatenobjekt festgelegt. Auf die gleiche Weise werden die Übertragungsarten für die restlichen Prozessdatenobjekte festgelegt. Als nächstes können Sie ein PLC-Projekt für den CANopen-Slave erstellen.

8.4.6 SDO-Daten in der SPS empfangen

SDO-Daten, die der CANopen Teil der Software nicht kennt und nicht selbstständig bearbeitet, werden in die SPS weitergeleitet und können hier per ADS-Notification ausgewertet und beantwortet werden.

Hierfür muss im System Manager untern dem CAN-Gerät der ADS-Port freigeschaltet werden.

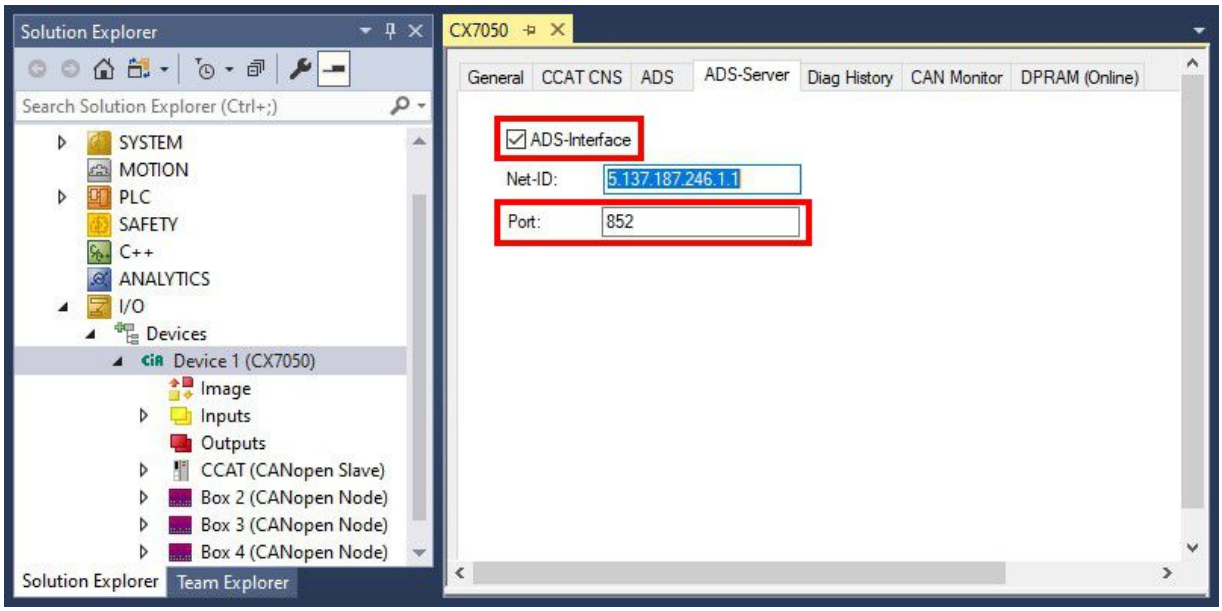


Abb. 46: Freischaltung eines ADS-Ports für einen CANopen-Slave.

SDO Read request

Daten, die gelesen werden sollen, müssen mit einem ADSREADIND empfangen werden und mit ADSREADRES beantwortet werden.

Eingangsparameter ADSREADIND	Beschreibung
NETID	NetID der CAN-Schnittstelle
Port Nummer	$0x1000_{hex}$ + Node Nummer
IDXGRP	16#8000_0000 + SDO Index (IDXGRP.31 = ADS-Notification)
IDXOFFS	SDO Subindex
LEN	wird nicht beim Lesen benötigt

Nun müssen Sie auf das ADS-Indication antworten mit einem ADS Read Response.

Eingangsparameter ADSREADRES	Beschreibung
NETID	NetID der CAN-Schnittstelle
Port Nummer	$0x1000_{hex}$ + Node Nummer
INVOKEID	INVOKEID des ADSREADIND Bausteins
RESULT	Fehler <> 0, fehlerfrei = 0
LEN	Länge der Daten

SDO Write request

Daten, die geschrieben werden sollen, müssen mit einem ADSWRITEIND empfangen werden und mit ADSWRITERES beantwortet werden.

Ausgangsparameter ADSWRITEIND	Beschreibung
NETID	NetID der CAN-Schnittstelle
Port Nummer	$0x1000_{hex}$ + Node Nummer

Ausgangsparameter ADSWRITEIND	Beschreibung
IDXGRP	16#8000_0000 + SDO Index (IDXGRP.31 = ADS Notification)
IDXOFFS	SDO Subindex
LEN	Anzahl des empfangenden Daten in BYTE

Nun müssen Sie auf das ADS-Indication antworten mit einem ADS Write Response.

Eingangsparameter ADSWRITERES	Beschreibung
NETID	NetID der CAN-Schnittstelle
Port Nummer	$0x1000_{hex}$ + Node Nummer
INVOKEID	INVOKEID des ADSWRITEIND Bausteins
RESULT	Fehler <> 0, fehlerfrei = 0

8.4.7 Slave-Node aus der SPS in PreOp schalten

Mit dem Baustein ADSWRTCTL können Sie einzelne CANopen Knoten in den Pre-Operational oder Operational Zustand versetzen. Vorraussetzung ist eine fest eingestellte Baudrate.

Eingangsparameter	Beschreibung
NETID	NetId der CAN Schnittstelle
Port Nummer	$0x1000_{hex}$ + NodeId (Slave Nummer)
ADSSTATE	ADSSTATE_RUN
DEVSTATE	0 - Pre / 1 - Operational
LEN	0
SRCADDR	0

8.5 CAN-Baudrate auslesen

Über die Variable InfoData[1] kann die Baudrate angezeigt und ausgewertet werden. Dies kann beim Slave mit AutoBaud helfen, wenn beispielsweise die Kommunikation nicht läuft. Damit lässt sich überprüfen, ob die richtige Baudrate mit AutoBaud eingestellt wurde.

Wert NodeState	Beschreibung
0x01040400	1 MBaud
0x01040600	800 kBaud
0x01040C00	500 kBaud
0x010A0C00	250 kBaud
0x01160C00	125 kBaud
0x011C0C00	100 kBaud
0x013A0C00	50 kBaud
0x01940C00	20 kBaud
0x01941A10	10 kBaud

8.6 Beliebige CAN-Telegramme verschicken

Mit dem Befehl ADSWRITE ist es möglich, eine beliebige CAN-Nachricht zu versenden.

Eingangsparameter	Beschreibung
NETID	NetId der CAN-Schnittstelle
Port Nummer	200
IDXGRP	16#0000F921
IDXOFFS	0
LEN	11 Bytes
SRCADDR	Pointer auf ein ARRAY von 11 Byte

Tab. 17: Aufbau der 11 Byte CAN-Daten

Byte	Beschreibung	Beispiel Node 7 SDO 0x607 Len 8 Download Request 0x2100 (Index) Sub Index 1 - Value "1"
1	COB-ID LowByte	0x06 (SDO Low Byte)
2	COB-ID HighByte	0x07 (SDO High Byte)
3	LEN (Länge)	0x08 (Len, kann hier auch 5 sein)
4	Daten[1]	0x22 (Download Request)
5	Daten[2]	0x00 (Index Low Byte)
6	Daten[3]	0x21 (Index High Byte)
7	Daten[4]	0x01 (Sub Index)
8	Daten[5]	0x01 (Value "1")
9	Daten[6]	0x00
10	Daten[7]	0x00
11	Daten[8]	0x00

8.7 IP- und MAC-Adresse auslesen

In diesem Beispiel wird gezeigt, wie Sie die IP- und die MAC-Adresse auslesen können. Mit dem Funktionsbaustein FB_MDP_NIC_Read können Informationen des Netzwerkadapters abgerufen werden.

Beispiel

```
Var
    FB_MDP_NIC_Read    : FB_MDP_NIC_Read;
END_VAR

PROGRAM:
FB_MDP_NIC_Read(
    bExecute:=TRUE ,
    tTimeout:= ,
    iModIdx:= ,
    sAmsNetId:= ,
    bBusy=> ,
    bError=> ,
    nErrID=> ,
    iErrPos=> ,
    stMDP_ModuleHeader=> ,
    stMDP_ModuleContent=> );
```

Der Ausgang stMDP_ModuleHeader zeigt die Header-Informationen an. Der Ausgang stMDP_ModuleContent zeigt unter anderem die Informationen zur IP- und MAC-Adresse an.

stMDP_ModuleHeader	ST_MDP_ModuleHea...	
iLen	UINT	4
nAddr	DWORD	131072
sType	T_MaxString	'Nic'
sName	T_MaxString	'st'
nDevType	DWORD	141072
stMDP_ModuleContent	ST_MDP_NIC_Prope...	
iLen	UINT	8
sMACAddress	T_MaxString	'00:01:05:5f:0f:7a'
sIPAddress	T_MaxString	'169.254.123.15'
sSubnetMask	T_MaxString	'255.255.0.0'
bDHCP	BOOL	TRUE
iReserved	BYTE	0

Abb. 47: Inhalt des MDP-Modules mit IP-und MAC-Adresse.

8.8 Virtuelle Ethernet-Schnittstelle

Die virtuelle Ethernet-Schnittstelle bindet Netzwerkadapter in das TwinCAT-System ein. Damit ist es möglich, eine virtuelle Ethernet-Kommunikation über ADS, TCP oder UDP zu einem BK9xx0 aufzubauen. Verwenden Sie dabei nicht mehr als zwei BK9xx0 und eine Zykluszeit > 50 ms.

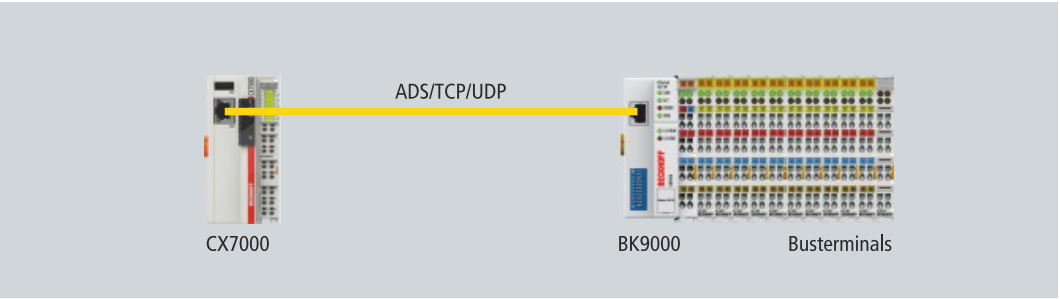
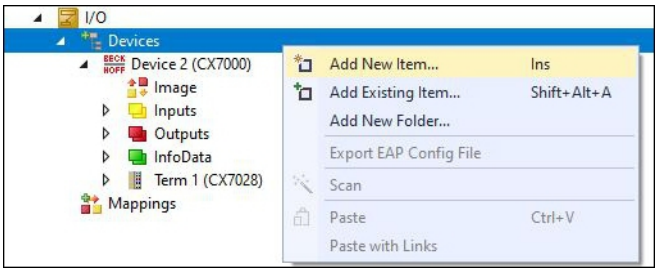


Abb. 48: Virtuelle Ethernet-Kommunikation über ADS, TCP oder UDP.

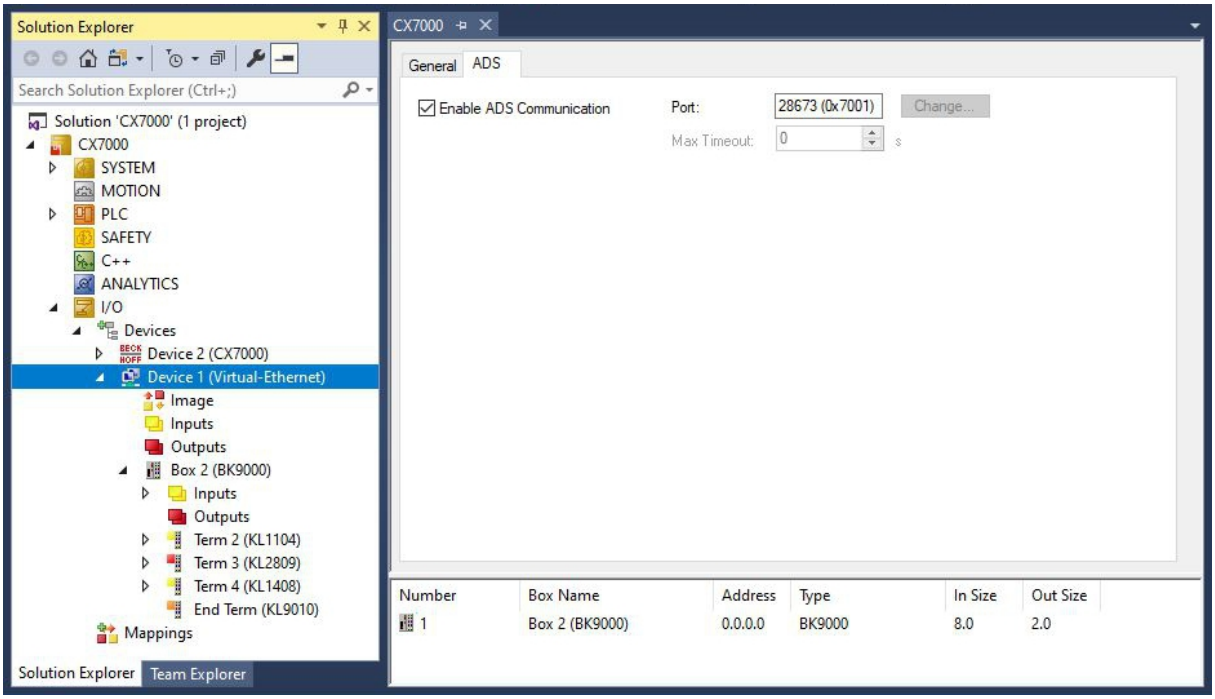
Gehen Sie wie folgt vor:

1. Klicken Sie links im Strukturbaum mit der rechten Maustaste auf **Devices**.



2. Klicken Sie auf **Add New Item** und Wählen Sie das **Virtual Ethernet Interface**.

⇨ Das Virtual Ethernet Interface wird links im Strukturbaum angelegt. Unter der Registerkarte **ADS** kann die ADS-Portnummer ausgelesen werden. Die Option **Enable ADS Communication** muss aktiv sein, damit eine ADS-Kommunikation zum BK9xx0 möglich ist.



8.9 CoE-Zugriff auf Multifunktions-I/Os

Mit dem Funktionsbaustein `FB_EcCoeSdoReadEx` können per SDO-Daten (Service Daten Objekt) Daten aus dem Objektverzeichnis eines EtherCAT-Slaves ausgelesen werden. Mit Hilfe der Parameter nSubIndex und nIndex wird ausgewählt, welches Objekt ausgelesen werden soll. Über `bCompleteAccess := TRUE` kann der Parameter mit Unterelementen eingelesen werden.

Beispiel: Firmware-Version der Multifunktions-I/Os auslesen.

```
VAR
AMSNetID AT %I*:T_AmsNetIdArr;
Port AT %I*:T_AmsPort;
FB_EcCoESdoReadEx: FB_EcCoESdoReadEx;
FirmwareVersion: STRING;
END_VAR
```

Für die Kommunikation mit der CX7028-Schnittstelle wird die AmsNetId und die Port-Nummer benötigt. Die Eingänge des Funktionsbausteins `FB_EcCoeSdoReadEx` können mit den Eingangsvariablen `netId` und `port` unter TwinCAT verlinkt werden, damit der Funktionsbaustein dauerhaft mit der CX7028-Schnittstelle verbunden ist.

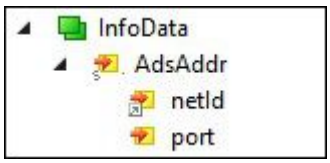


Abb. 49: CoE-Zugriff auf Multifunktions-I/Os, Eingangsvariablen "netId" und "port" unter TwinCAT.

Der Eingang `sNetId` des Funktionsbausteins entspricht dem Eingang `netId` unter TwinCAT. Der Funktionsbaustein verlangt einen String und die Verknüpfung liefert ein Byte-Array. Sie können das Byte-Array mit der Funktion `F_CreateAmsNetId` in einen String umwandeln. Der Eingang `nSlaveAddr` entspricht dem Eingang `port` unter TwinCAT.

```
FB_EcCoESdoReadEx(
sNetId:=F_CreateAmsNetId(nIds:=AMSNetID ) , (* AmsNetId of the CX7028 Interface *)
nSlaveAddr:=Port , (* Port Number(nSlaveAddr): 0x1000 *)
nSubIndex:= ,
nIndex:=16#100A , (* Index Number *)
pDstBuf:=ADR(FirmwareVersion) ,
cbBufLen:=SIZEOF(FirmwareVersion) ,
bExecute:=TRUE ,
tTimeout:= ,
bCompleteAccess:= ,
bBusy=> ,
bError=> ,
nErrId=> );
```

Die Index-Nummer für das CoE-Objekt **Software version** befindet sich unter der Registerkarte CoE-Online.

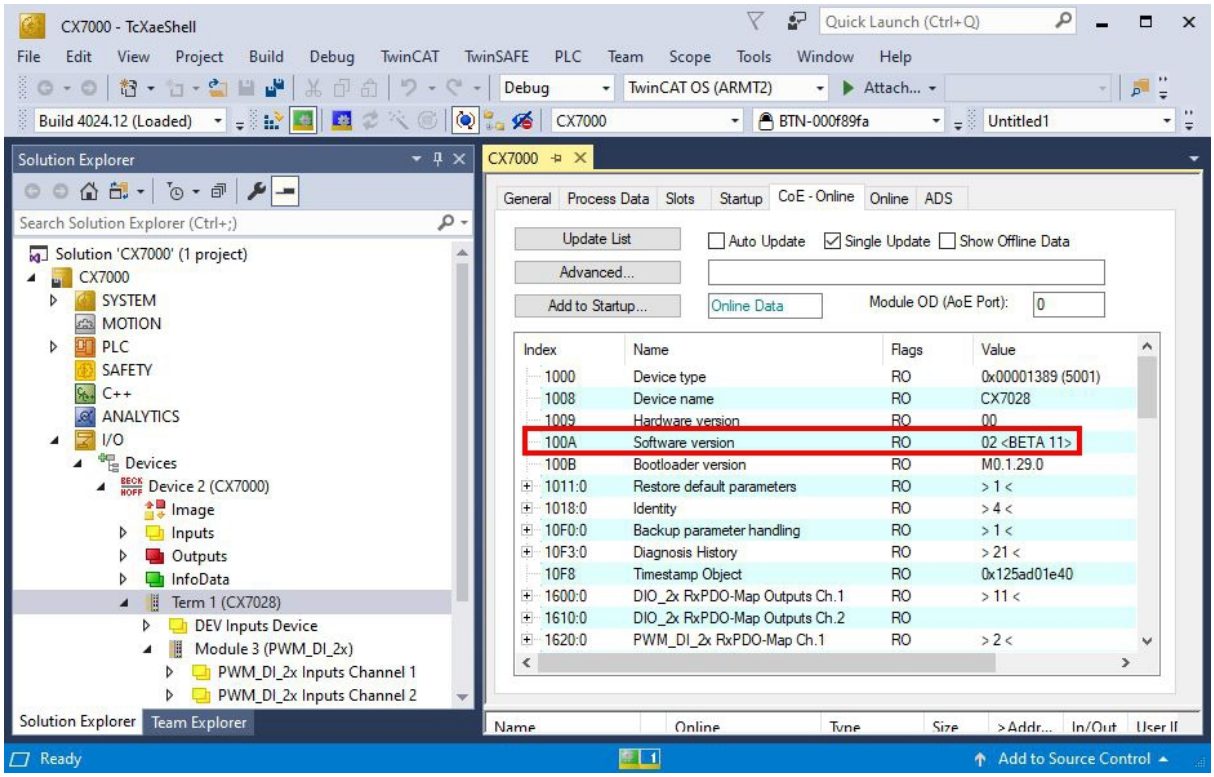


Abb. 50: CoE-Kommunikation, Auflistung der CoE-Objekte mit passender Index-Nummer.

Mit dem Funktionsbaustein `FB_EcCoeSdoWriteEx` kann per SDO-Download ein Objekt aus dem Objektverzeichnis eines EtherCAT Slaves beschrieben werden. Achten Sie darauf, ob auf das Objekt lesen zugriffen werden kann, was in der Spalte Flags angezeigt wird. Mit Hilfe der Parameter nSubIndex und nIndex wird ausgewählt, welches Objekt beschrieben werden soll. Über `bCompleteAccess := TRUE` kann der Parameter mit Unterelementen geschrieben werden.

8.10 Netzteilklemme

Auf der rechten Seite können wahlweise EtherCAT-Klemmen (E-Bus) oder Busklemmen (K-Bus) angereiht werden; der CX7050 erkennt in der Hochlaufphase automatisch, welches System angeschlossen ist.

K-Bus-Interface

Der CX7050 liest beim Scannen die Klemmentypen aus und legt sie im System Manager unter einem Buskoppler an.

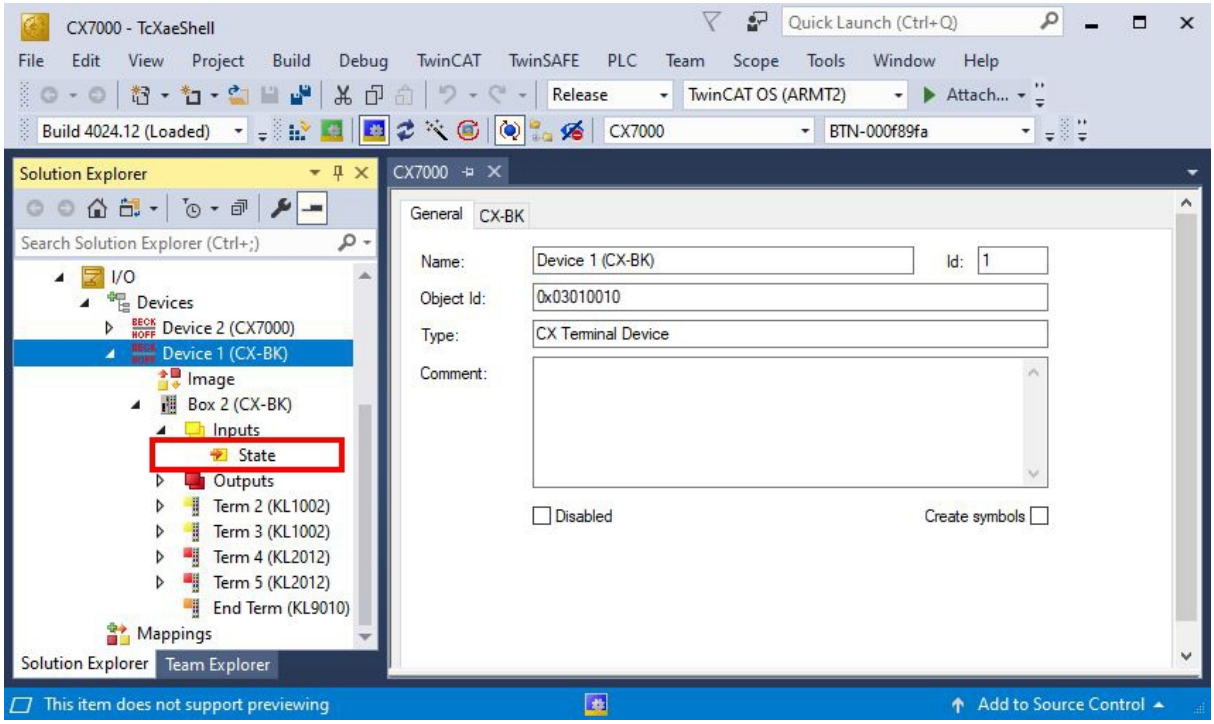


Abb. 51: K-Bus-Interface eines CX7050 im TwinCAT System Manager.

Für die K-Bus-Diagnose gibt es in TwinCAT unter dem Buskoppler eine Status-Variable, die für Diagnosezwecke genutzt werden kann und den Status der K-Bus-Kommunikation anzeigt. Weitere Informationen finden Sie im Kapitel „Fehlerbehandlung und Diagnose“ unter K-Bus [▶ 185].

E-Bus-Interface

Distributed-Clocks

Die Embedded-PCs der Serie CX7000 eignet sich nicht für den Einsatz von EtherCAT-Slaves, die Distributed-Clocks verwenden oder zwingend voraussetzen.

Auch der Betrieb von EtherCAT-Klemmen und EtherCAT-Geräten ist am CX7050 möglich. Der CX7050 erkennt auch diese Klemmen beim Scannen automatisch, liest die Klemmentypen aus und legt sie im System Manager unter einem EtherCAT-Koppler an.

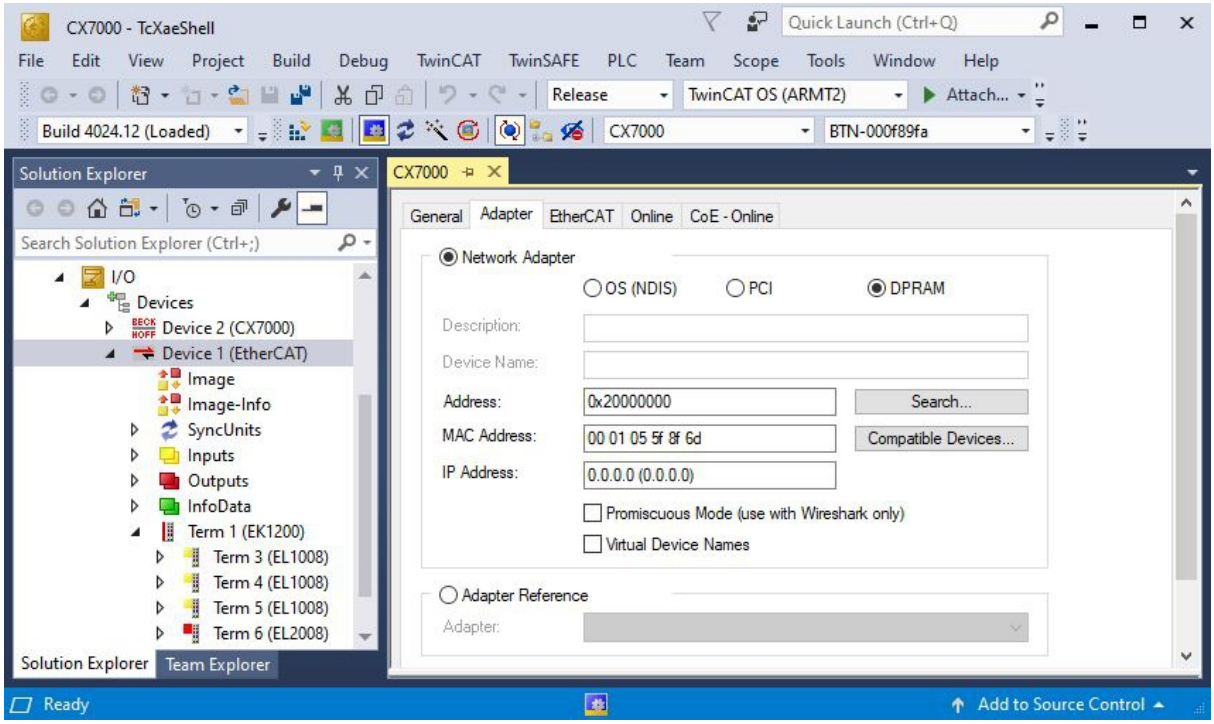


Abb. 52: E-Bus-Interface eines CX7050 im TwinCAT System Manager.

Weitere Informationen zur Diagnose finden Sie im Kapitel „Fehlerbehandlung und Diagnose“ unter E-Bus [▶ 188].

8.11 Zyklus- und Bearbeitungszeiten

8.11.1 Bearbeitungszeit im SPS-Programm messen

In diesem Beispiel wird gezeigt, wie Sie die Bearbeitungszeit eines Programmcodes mithilfe eines kleinen SPS-Programms bestimmen können. Damit können Sie beispielsweise messen, wie lange die SPS für eine mathematische Funktion eine Schleife oder einen bestimmten Programmteil benötigt. Die Auflösung beträgt 1 ns pro Digit.

Beispiel

```
VAR
    MeasureStart    : T_DCTIME64;
    MeasureResult   : T_DCTIME64;
END_VAR

PROGRAM:
MeasureStart:=F_GetActualDcTime64(); (*Insert your program code to measure the processing time*)
MeasureResult:=F_GetActualDcTime64()-MeasureStart;
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_EtherCAT

8.11.2 Real-Time-Clock (RTC)

Der CX7050 hat eine interne, kondensatorgepufferte Realtime-Clock (RTC) für Zeit und Datum, die im ausgeschalteten Zustand weiterläuft. Die Kapazität des Kondensators reicht für mindestens 30 Tage und ist anders als eine batteriegestützte Lösung wartungsfrei. Ist der CX7050 länger als 30 Tage ausgeschaltet, geht die Uhrzeit verloren und muss neu eingestellt werden

Folgende Einstellungen sind in der boot.conf-Datei möglich:

- SNTP-Server
- Update-Time (Standardeinstellung = 1 Stunde)
- Change UTC Offset
- DHCP-Server

Beispiel

In dem unteren Beispiel wird gezeigt, wie Sie die Uhrzeit auslesen können. In dem Beispiel wird die Uhrzeit als UTC-Zeit ausgegeben und eine Stunde addiert, um die MEZ-Zeit zu erhalten.

```
VAR
    FB_LocalSystemTime    : FB_LocalSystemTime;
    DATEANDTIME           : DATE_AND_TIME;
    DATEANDTIME_Add1h     : DATE_AND_TIME;
END_VAR

PROGRAM:
FB_LocalSystemTime(
        sNetID:= ,
        bEnable:=TRUE ,
        dwCycle:= ,
        dwOpt:= ,
        tTimeout:= ,
        bValid=> ,
        systemTime=> ,
        tzID=> );

DATEANDTIME:=SYSTEMTIME_TO_DT(TIMESTR:=FB_LocalSystemTime.systemTime );    (*UTC Time*)
DATEANDTIME_Add1h:=DATEANDTIME+T#1H;    (*UTC Time + 1h*)
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Utilities (System)

8.11.3 Zykluszeit von 250 µs

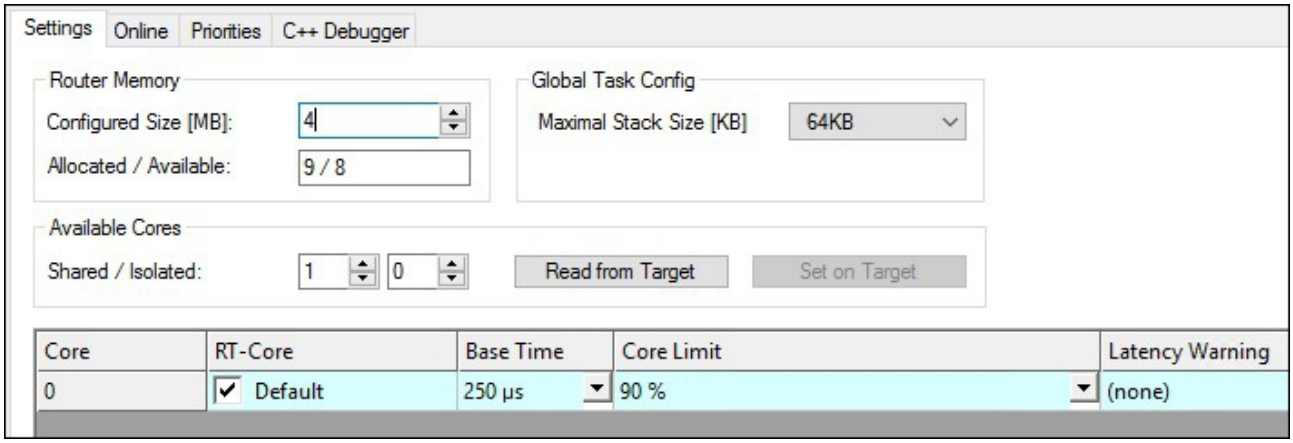
Beachten Sie, dass eine Zykluszeit von 250 µs auf einem CX7050 ein extremes Optimum darstellt und alle Rahmenbedingungen passen müssen. Des Weiteren ist eine Zykluszeit von 250 µs nur dann sinnvoll, wenn die Ein- und Ausgänge entsprechend schnell sind.

Der CX7050 hat unterschiedliche Schnittstellen, darunter beispielsweise den K-Bus. Der K-Bus schafft unter optimalen Bedingungen vielleicht 1 ms und ist daher nicht für Zykluszeiten von 250 µs geeignet. Der E-Bus (EtherCAT) ist viel schneller, nur ist der Aufbau eines EtherCAT-Frames und das Zusammenfügen der Daten in einen EtherCAT-Frame wesentlich aufwendiger, sodass auch hier nur 1 ms möglich sind.

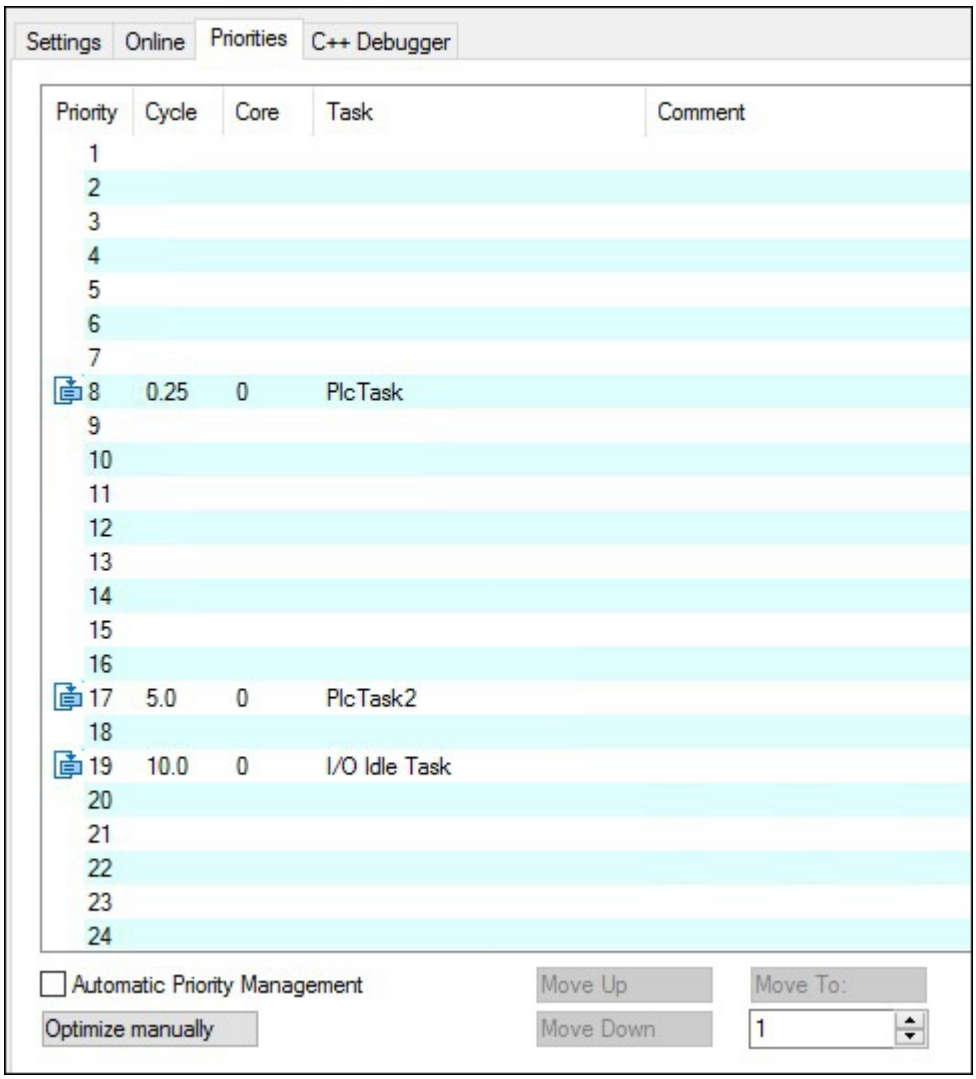
Natürlich kann EtherCAT mit anderen Industrie-PCs unter 100 µs betrieben werden. Diese sind in der Regel aber mit leistungsfähigeren CPUs ausgestattet und nutzen eventuell einen DMA-Controller für die EtherCAT-Bearbeitung. Das ist beim CX7050 allerdings nicht der Fall, sodass die CPU-Leistung und die Schnittstellen zum EtherCAT die begrenzenden Faktoren sind. Natürlich ist der CX7050 als Kleinsteuerung nicht für Highspeed-Anwendungen entwickelt worden und sollte aufgrund seiner Kosteneffizienz nicht mit leistungsfähigeren Industrie-PCs verglichen werden.

Zykluszeit von 250 µs einstellen

Eine Zykluszeit von 250 µs ist auf einem CX7050 möglich, wenn die Rahmenbedingungen stimmen. Beim CX7050 helfen die Multifunktions-I/Os, die über eine schnelle IO-Verbindung mit der CPU verbunden sind. Die Verbindung ist sehr schlank gehalten und hat einen entsprechend guten Datendurchsatz. Mithilfe der Multifunktions-I/Os ist es möglich, die 250 µs zu erreichen. Das SPS-Programm darf natürlich nur sehr wenig Code enthalten und das Core-Limit auf 90 % eingestellt werden, was wiederum die beschriebenen Nachteile mit sich bringt (siehe: Echtzeit und CPU-Auslastung [▶ 199]).



Zusätzlich sollten Sie die Priorität der Task so einstellen, dass die 250 µs Task die höchste Priorität im System hat.



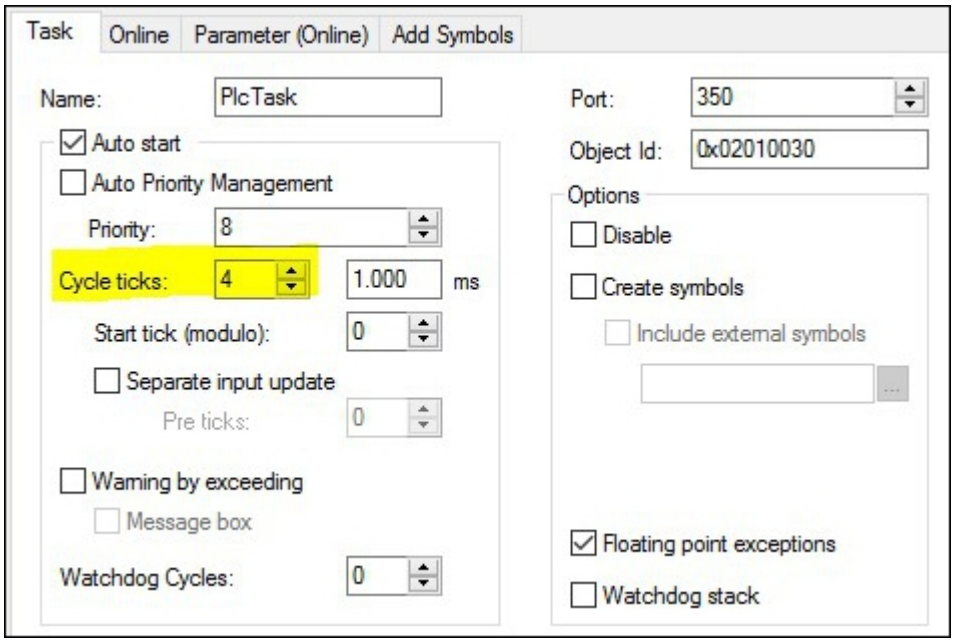
Wenn Sie nun einen digitalen Ausgang des CX7028-Schnittstelle toggeln lassen, in der 250 µs Task beispielsweise mit `Out_01:=not Out_01`, so wird diese mit einer Frequenz von 2 kHz ausgegeben. Damit der Ausgang optimal schnell ist, sollte dieser Ausgang eine Last haben. Verkabeln Sie den Ausgang nur mit einem digitalen Eingang, dadurch ist die Last sehr klein und das Ausschaltverhalten des Treibers relativ langsam. Langsam bezieht sich hier auf die 250 µs Taskzeit. Es macht schon einen Unterschied aus, ob der Ausgang 50 µs oder 100 µs zum Ausschalten benötigt. Wenn sie nun die Reaktionszeit messen wollen, also die Zeit wie lange der CX7050 braucht, um auf einen Eingang zu reagieren, ist folgender Hintergrund wichtig:

Ab einer Zykluszeit von 1 ms oder größer wird ein optimaler Zyklus gefahren, d.h. die Eingänge der CX7028-Schnittstelle werden ca. 20 % vor dem neuen Taskzyklus vom Prozessor der CX7028-Schnittstelle eingelesen. Ist die Taskzeit schneller als 1 ms, so reicht die Zeit für die optimierte Reaktionszeit nicht aus. Hier werden dann die Eingänge mit dem Taskzyklus eingelesen. Das hat zur Folge, dass eine Taskzeit von 500 µs die gleiche Reaktionszeit erreicht wie Taskzeit von 1 ms. Unter 1 ms Taskzeit braucht das Update für einen Zyklus 4 Taskzyklen. In 1 ms oder langsamer zwei Taskzyklen. Dies soll ihnen bewusst machen, dass nicht immer die Verkürzung der Zykluszeit die Reaktionszeit verkürzt, sondern auch der interne Ablauf, der eine entscheidende Rolle beim Einlesen der Daten spielt.

Ein Beispiel, damit Sie dieses Verhalten selber reproduzieren können, die Unterschiede sehen und messen können:

1. Schließen Sie die Spannungsversorgung +24 V Up und“0 V Up zur Versorgung der Multifunktions-I/Os an.
2. Verbinden Sie den Ausgang 1 mit dem Eingang 1, um den Ausgang wie beschrieben zu toggeln.
3. Verbinden Sie den Ausgang 2 mit dem Eingang 2.
4. Stellen Sie das Core-Limit auf 90 % ein, die Base-Zeit auf 250 µs, die Priorität der schnellen Task auf die höchste Priorität und die Idle-Task auf 10 ms.

Die Eingänge haben nur eine minimale Filterzeit und sind daher gut für die Messung geeignet. Eine Last am Ausgang ist in dem Fall nicht notwendig. Die Base-Zeit belassen wir für die folgenden Beispiele immer auf 250 µs und erhöhen nur die Anzahl der Cycle-Ticks, um die entsprechende Taskzeit einzustellen.



Beispielprogramm

```
PROGRAM MAIN
VAR
    bOut_1 AT %Q*:BOOL; (*toggle Output link to digital Output pin 7*)
    bOut_2 AT %Q*:BOOL; (*reaction time link to digital Output pin 14*)

    bIn_1 AT %I*: BOOL; (*toggle Output link to digital Input pin 2*)
    bIn_2 AT %I*: BOOL; (*reaction time link to digital Input pin 10*)

    fbTimer : TON;
    fbflanke1 : R_TRIG;
    fbflanke2 : R_TRIG;

    cnt1: INT; (*toggle Output*)
    cnt1_M: INT; (*toggle Output*)

    cnt2: INT; (*reaction time*)
    cnt2_M: INT; (*reaction time*)
END_VAR


PROGRAM MAIN
bOut_1:= NOT bOut_1; (*toggle Output*)
bOut_2:= NOT bIn_2; (*reaction time*)

fbflanke1(CLK:=bIn_1);
IF fbflanke1.Q THEN
    cnt1:=cnt1+1; (*toggle Output*)
END_IF

fbflanke2(CLK:=bIn_2);
IF fbflanke2.Q THEN
    cnt2:=cnt2+1; (*reaction time*)
END_IF

fbTimer(PT:=T#1S,in:=NOT fbTimer.Q);

IF fbTimer.Q THEN
    cnt2_M:=cnt2; (*reaction time*)
    cnt1_M:=cnt1; (*toggle Output*)
    cnt1:=0;
    cnt2:=0;
END_IF
```

Das Toggeln des Ausgangs bewirkt eine Frequenz von 2 kHz, 250 µs On 250 µs Off, also eine Periodendauer von 500 µs. Bei einer Messung der positiven Flanke sind dies 2000 Flankenwechsel in einer Sekunde.

bOut_1	BOOL	TRUE
bOut_2	BOOL	TRUE
bIn_1	BOOL	FALSE
bIn_2	BOOL	FALSE
fbTimer	TON	
fbflanke1	R_TRIG	
fbflanke2	R_TRIG	
cnt1	INT	1014
cnt1_M	INT	2000
cnt2	INT	253
cnt2_M	INT	500

Abb. 53: Messung bei einer Taskzeit von 250 µs.

Bei der Reaktionszeit sind es 500 Wechsel in einer Sekunde, da hier der optimierte Zugriff auf die Eingänge nicht greift.

bOut_1	BOOL	TRUE
bOut_2	BOOL	TRUE
bIn_1	BOOL	TRUE
bIn_2	BOOL	FALSE
fbTimer	TON	
fbflanke1	R_TRIG	
fbflanke2	R_TRIG	
cnt1	INT	68
cnt1_M	INT	1001
cnt2	INT	17
cnt2_M	INT	250

Abb. 54: Messung bei einer Taskzeit von 500 µs.

Wie zu erwarten sind bei einer doppelt so großen Taskzeit die Werte nur halb so groß.

bOut_1	BOOL	FALSE
bOut_2	BOOL	TRUE
bIn_1	BOOL	FALSE
bIn_2	BOOL	FALSE
fbTimer	TON	
fbflanke1	R_TRIG	
fbflanke2	R_TRIG	
cnt1	INT	169
cnt1_M	INT	501
cnt2	INT	84
cnt2_M	INT	251

Abb. 55: Messung bei einer Taskzeit von 1 ms.

Bei einer Taskzeit von 1 ms kann man deutlich sehen, dass der optimierte Modus tatsächlich hilft, die Reaktionszeit zu verringern. Während der Toggle-Wechsel sich wieder halbiert hat, also bei einer Taskzeit von 1 ms jetzt noch bei 500 Hz liegt, ist der Wert bei der Reaktionszeit gleichgeblieben.

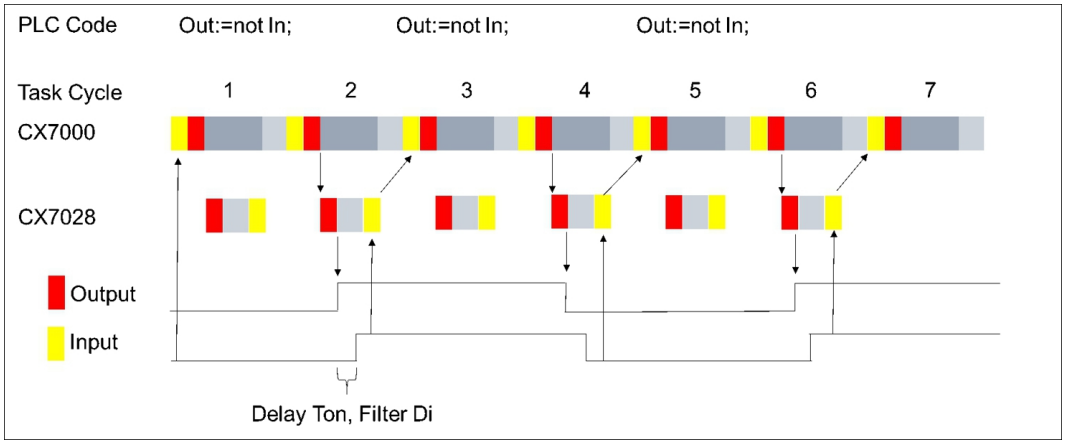
8.11.3.1 Zykluszeit ≥1 ms

Abb. 56: CX7050 CPU und SPS.

Gelb und rot: Mapping und Update der IOs.
Hellgrau: Restzeit bis zum erneuten Task-Begin (OS)
Dunkelgrau:. PLC Zyklus.

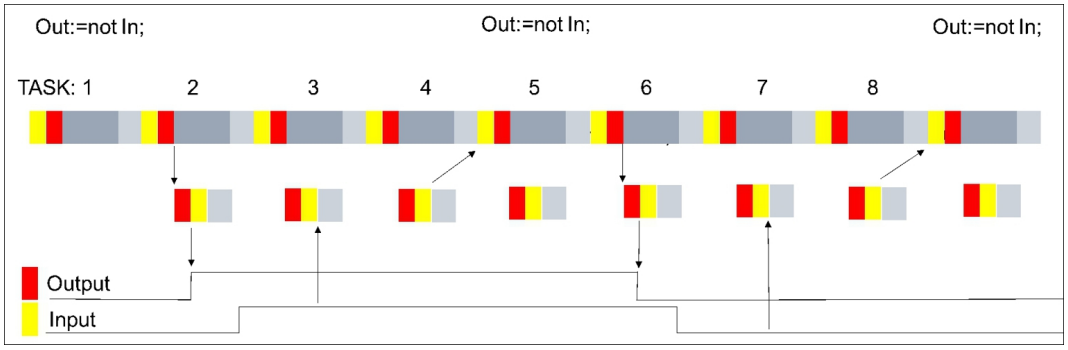
Abb. 57: CPU der CX7028-Schnittstelle.

Rot: Output Update.
Grau: CPU Bearbeitung der Multifunktions-IOs.
Gelb: Input Update (ab 1 ms Zykluszeit wird mit dem Update der Eingangssignale bis ca. 80 % der Zykluszeit gewartet, so das möglichst spät, also vor dem nächsten Zyklus die Eingänge eingelesen werden).



8.11.3.2 Zykluszeit < 1 ms

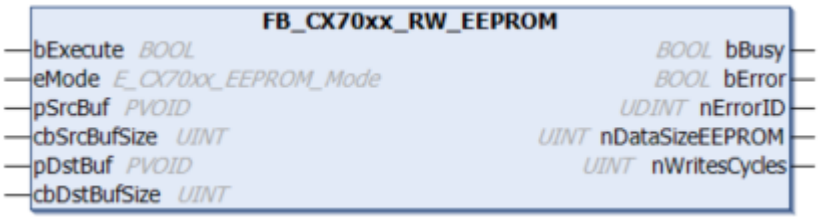
Ab einer Zykluszeit von < 1 ms wird der Update der Eingangssignale sofort durchgeführt und steht damit erst mit dem nächsten Zyklus zur Verfügung. Die Eingangssignale sind somit immer ein Zyklus alt.



Mit diesem Hintergrundwissen sollten Sie in der Lage sein, für ihre Anwendung die richtigen Einstellungen am CX7050 vorzunehmen.

8.12 Funktionsbausteine

8.12.1 FB_CX70xx_RW_EEPROM



Der Baustein erlaubt das Schreiben von maximal 120 Byte in das EEPROM (Hardware) des CX70xx. Das EEPROM darf maximal 200-mal beschrieben werden. Der Speicher ist für das einmalige Schreiben gedacht.

Dieser Funktionsbaustein kann dafür verwendet werden, den CX70xx zu personalisieren. Das heißt, im einfachsten Fall schreiben Sie Ihre Firmenkennung in das EEPROM. Beim Start des CX70xx Programms lesen Sie den Inhalt des Speichers aus. Ist dieser zum Beispiel leer, können Sie das Programm nicht weiter ausführen, da es sich nicht mehr um Ihren originalen CX70xx handelt, den Sie programmiert haben.

Sollten Sie einen CX70xx gegen ein neues Gerät austauschen wollen, muss das EEPROM erneut von Ihnen geschrieben werden.

Eingänge

```
VAR_INPUT
    bExecute       : BOOL;         // rising edge triggers process with selected mode
    eMode          : E_CX70xx_EEPROM_Mode;    // select RW mode
    pSrcBuf        : PVOID;        // pointer to WRITE EEPROM data buffer
    cbSrcBufSize   : UINT;         // size of WRITE EEPROM data buffer (max.120 Bytes)
    pDstBuf        : PVOID;        // pointer to READ EEPROM data buffer
    cbDstBufSize   : UINT;         // max.size of READ EEPROM data buffer (max.120 Bytes)
END_VAR
```

Name	Typ	Beschreibung
bExecute	BOOL	Positive Flanke startet den Baustein.
eMode	E_CX70xx_EEPROM_Mode	ReadOnly: EEPROM Lesen WriteOnly: EEPROM Schreiben WriteAndRead: EEPROM Schreiben und im Anschluss Lesen
pSrcBuf	PVOID	Pointer auf den Datenpuffer, der geschrieben werden soll.
cbSrcBufLen	UINT	Länge der zu schreibenden Daten (maximal 120 Bytes)
pDstBuf	PVOID	Pointer auf den Datenpuffer, in den der Inhalt des EEPROM kopiert werden soll.
cbDstBufLen	UINT	Länge der zu lesenden Daten. (maximal 120 Bytes) Beim Lesen muss die Längeninformation größer gleich der im EEPROM enthalten Daten sein.

Ausgänge

```
VAR_OUTPUT
    bBusy            : BOOL;         // FB is working
    bError           : BOOL;         // FB has an Error
    nErrorID         : UDINT;        (* Error Code
    If nErrorID=DEVICE_INVALIDACCESS the EEPROM write cycles reached max. value.
    If nErrorID=DEVICE_INVALIDPARM the given pointer parameter is invalid/null.
    If nErrorID=DEVICE_INVALIDSIZE the given buffer size is too small or too big.
    If nErrorID=DEVICE_SRVNOTSUPP probably the image version need to be updated to support this feat
ure. *)
    nDataSizeEEPROM  : UINT;         // current size of (read) EEPROM data in bytes (max.120 Bytes)
    nWritesCycles    : UINT;         // already performed EEPROM write cycles (maximum possible = 20
0)
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv und arbeitet.
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiele: DEVICE_INVALIDACCESS: Die EEPROM Schreibzyklen haben den maximalen Wert erreicht. Das EEPROM kann nicht erneut beschrieben werden. DEVICE_INVALIDPARM: Die zugewiesenen Pointer sind ungültig/ NULL. DEVICE_INVALIDSIZE: Die zugewiesene Puffergröße ist zu klein oder zu groß. DEVICE_SRVNOTSUPP: Die Image-Version des CX70xx unterstützt dieses Feature nicht. Ein Update (>=35695) ist notwendig.
nDataSizeEEPROM	UINT	Aktuelle Größe in Bytes der gelesenen EEPROM Daten
nWritesCycles	UINT	Anzahl der noch zur Verfügung stehenden Schreibvorgänge

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX70xx	Tc2_SystemCX (System) >= 3.4.8.0

8.12.2 FB_CX70xx_ResetOnBoardIO

Der Baustein erlaubt es, einen Reset vom OnBoard-IO des Embedded-PC CX70xx auszuführen.

Typischer Anwendungsfall ist nach einem Fehler in der Kommunikation zu den OnBoard-IOs (CX7028). Zu einem solchen Fehler kommt es, wenn die Spannungsversorgung (Up) der OnBoard-IOs unterbrochen wird.

HINWEIS

Zustand der IOs

Ausgänge, die im Prozessabbild noch gesetzt sind, werden nach einem Reset sofort wieder eingeschaltet.

Weitere Details zum OnBoard-IO finden Sie in der Dokumentation des Embedded-PC CX70xx.

Eingänge

```
VAR_INPUT
    bExecute      : BOOL;         // rising edge triggers process
    sNetId        : T_AmsNetID;   // AMS Net ID of the OnBoard IOs
    tTimeout      : TIME := DEFAULT_ADS_TIMEOUT; // maximum time allowed for execution of this ADS c
ommand
END_VAR
```

Name	Typ	Beschreibung
bExecute	BOOL	Positive Flanke startet den Baustein.
sNetId	T_AmsNetID	AMS Net ID der OnBoard-IOs
tTimeout	TIME	Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

Ausgänge

```
VAR_OUTPUT
    bBusy       : BOOL;       // FB is working
    bError      : BOOL;       // FB has an Error
    nErrorID    : UDINT;      (* Error Code. If nErrorID=DEVICE_SRVNOTSUPP probably the image versio
n need to be updated to support this feature. *)
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv und arbeitet.
bError	BOOL	Der Baustein hat einen Fehler.
nErrorID	UDINT	ADS Fehler Code Beispiele: DEVICE_SRVNOTSUPP: Die Image-Version des CX70xx unterstützt dieses Feature nicht. Ein Update (>=47912) ist notwendig.

Beispiel:

```
FUNCTION_BLOCK FB_Test_ResetOnboardIO
VAR
    AMSNetID    : T_AmsNetIdArr;    // link to the AMS Net ID of the OnBoard IOs
    State       : WORD;             // link to the State of the OnBoard IOs
    bReset      : BOOL;             // if Ready to Reset you can reset the OnBoard IOs
    fbReset     : FB_CX70xx_ResetOnBoardIO;
END_VAR
```

```
IF State<>8 AND NOT State.8 AND State.4 THEN // if OnBoard IO device signals an error and is not OP
but present
    bReset := TRUE;
ELSE
    bReset := FALSE;
END_IF

IF NOT fbReset.bBusy AND bReset THEN
    fbReset(bExecute:=TRUE, sNetId:=F_CreateAmsNetId(AMSNetID));
ELSE
    fbReset(bExecute:=FALSE);
END_IF
```

Entwicklungsumgebung	Zielplattform	Einzubindende SPS- Bibliotheken
TwinCAT v3.1.4024.26	CX70xx	Tc2_SystemCX (System) >= 3.4.8.0

8.13 Wichtige Attribut-Pragmas

Attribut-Pragmas dienen dazu, die Kompilierung und die Vorkompilierung zu beeinflussen. TwinCAT unterstützt eine Reihe von vordefinierten Attribut-Pragmas. Attribute werden im Deklarationsteil definiert.

8.13.1 Attribut 'Tc2GvlVarNames'

Das Pragma bewirkt, dass Symbole, welche in einer GVL deklariert sind, über ADS genauso angesprochen werden wie in TwinCAT 2, ohne die Verwendung des GVL-Namens als Namespace.

Syntax: {attribute ‘Tc2GvlVarNames‘}

Beispiel:

```
{attribute 'Tc2GvlVarNames'}
VAR_GLOBAL
    Test : INT;
END_VAR

GVL.Test:=GVL.Test+1;     (*without attribute*)
Test:=Test+1;             (*with attribute*)
```

8.13.2 Attribut 'pack_mode'

Dieses Attribut-Pragma legt fest, wie eine Datenstruktur während der Allozierung gepackt wird. Das Attribut muss oberhalb der Datenstruktur eingefügt werden und wirkt sich auf das Packen der gesamten Struktur aus.

Syntax: {attribute 'pack_mode' := '<Value>'}

Beispiel

```
{attribute 'pack_mode' := '0'}
TYPE str_Test :
STRUCT
      byTest1    : BYTE;
      iTest      : DINT;
      byTest2    : BYTE;
      nValue     : INT;
END_STRUCT
END_TYPE
```

In diesem Beispiel wurde der Pack-Modus auf 0 gestellt. Wenn Sie die Größe der Struktur im Beispiel mit SIZEOF bestimmen, dann erhalten Sie den Wert 8.

1 Byte + 4 Byte (DINT) + 1 Byte + 2 Byte (INT) = 8 Byte

Wenn Sie den Pack-Modus auf 2 stellen (WordAlignment), erhalten Sie den Wert 10, weil nach jedem Byte noch ein Füllbyte eingefügt wird. Wenn Sie den Pack-Modus auf 4 stellen (DWordAlignment), dann erhalten Sie den Wert 12, weil diesmal nach jedem Byte drei Füllbytes eingefügt werden. Beim einem Pack-Modus von 8 (LWordAlignment) verändert sich nichts, da in dem Beispiel keine Variablen verwendet werden, die 8 Byte benötigen.

Der CX7050 arbeitet mit dem DWordAlignment (Pack-Modus 4), wenn Sie das Attribut nicht verwenden.

Weitere Informationen zum Attribut 'pack_mode' finden Sie unter: Attribut 'pack_mode'

8.13.3 Attribut 'TcCallAfterOutputUpdate'

Das Attribut-Pragma `TcCallAfterOutputUpdate` bewirkt, dass das IO-Update vor dem SPS-Zyklus stattfindet und nicht wie es standardmäßig eingestellt ist, nach dem SPS-Programm.

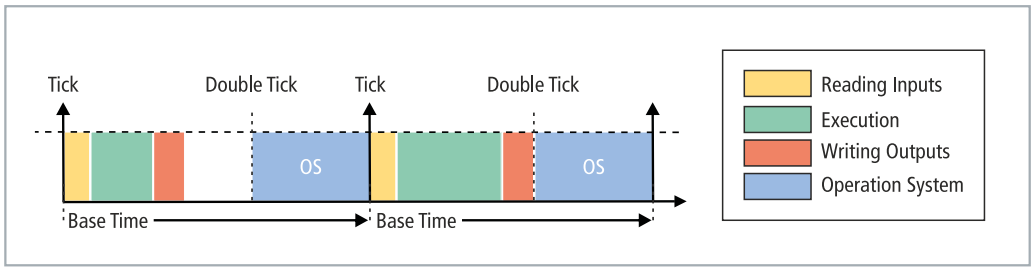


Abb. 58: Standard-Aufruf einer SPS-Task.

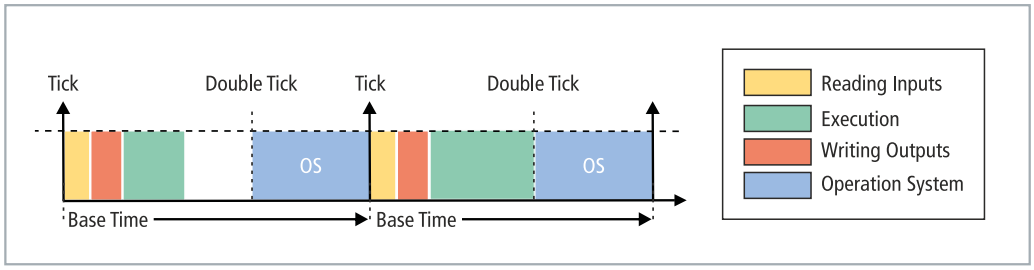


Abb. 59: Aufruf einer SPS-Task mit Attribut TcCallAfterOutputUpdate.

Diese Funktionalität kann für Projekte mit stark schwankenden Zykluszeiten eingesetzt werden. Bei Projekten mit stark schwankenden Zykluszeiten werden die Ausgänge, da sie nach dem SPS-Zyklus geschrieben werden, mal früher (kurze SPS-Zykluszeit) und mal später (lange SPS-Zykluszeit) geschrieben. Diese Schwankungen verursachen einen Jitter in den Ausgängen. Der Nachteil besteht darin, dass durch

das Attribut nicht ganz so schnell reagiert werden kann und immer ein Zyklus verloren geht. Sie müssen sich entscheiden, ob Sie schnell auf einen Eingang reagieren wollen (Standardeinstellung) oder ob sie lieber ein deterministisches Verhalten der Ausgänge haben wollen (setzen des Attributes).

Syntax: {attribute 'TcCallAfterOutputUpdate'}

Einfügeort: Dieses Attribut muss zu allen Programm POUs hinzugefügt werden, die nach dem Output Update aufgerufen werden sollen.

Beispiel:

Um das Verhalten zu verdeutlichen, brauchen Sie eine digitale Ausgangsklemme, beispielsweise eine EL2008 und ein Oszilloskop.

Schreiben Sie ein kleines SPS-Programm und verlinken Sie die Variable `bOut` mit einem digitalen Ausgang:

```
bOut:=not bOut;
```

Das SPS-Programm ist sehr einfach und verursacht keine Schwankungen. Der Puls wird auf dem Oszilloskop wie folgt dargestellt:

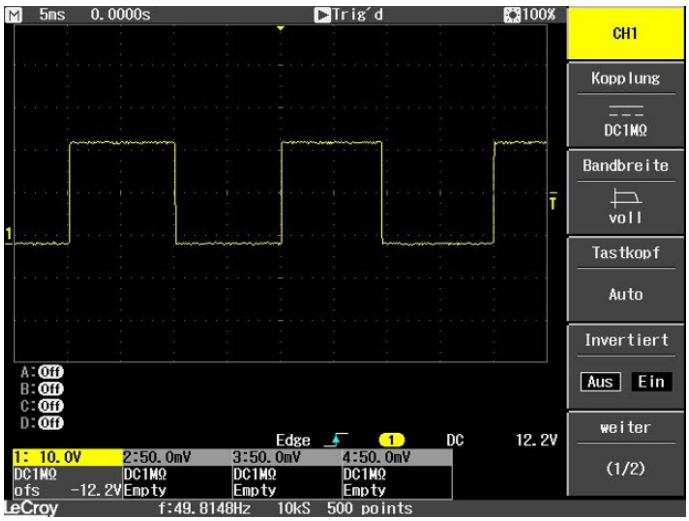


Abb. 60: Puls eines digitalen Ausgangs ohne Last.

Erweitern Sie nun das SPS-Programm um eine For-Schleife, um eine Programmlast zu erzeugen. Die verwendete mathematische Funktion spielt keine Rolle und soll nur eine Last erzeugen:

```
bOut:=not bOut;

IF bOut THEN
    For loop:=1 to 2000 do
        lrTest:=SIN(INT_TO_LREAL(loop)*3.14);
    END_FOR
END_IF
```

Wann immer der Ausgang auf TRUE gesetzt wird, wird die Schleife durchlaufen und eine Last erzeugt. Dadurch wird mehr Zeit für die Ausführung der SPS benötigt und der Ausgang später als sonst geschrieben. Beim nächsten Zyklus wird der Ausgang wieder auf FALSE gesetzt, die Schleife wird nicht durchlaufen und der Ausgang wird schneller auf FALSE gesetzt, da das SPS-Programm ohne For-Schleife wieder schneller fertig ist. Das Ergebnis ist, dass der Puls sehr viel kürzer ist.

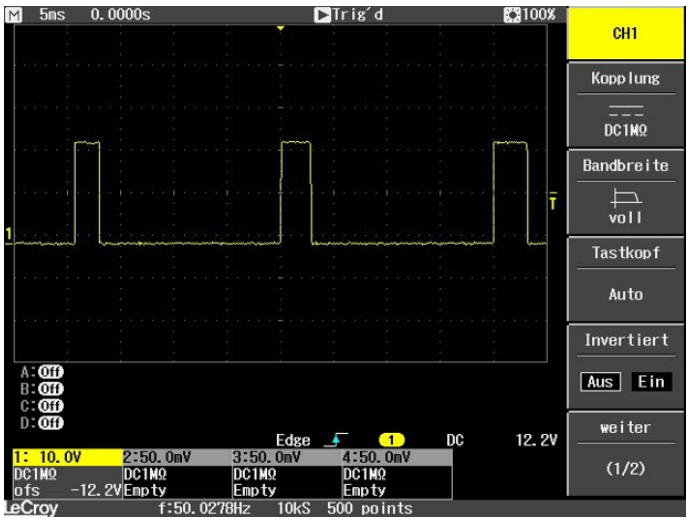


Abb. 61: Verkürzter Puls eines digitalen Ausgangs mit Last.

Wenn die For-Schleife statt dem TRUE beim FALSE aufgerufen wird, wird das Ergebnis invertiert.

```
bOut:=not bOut;

IF not bOut THEN
    For loop:=1 to 2000 do
        lrTest:=SIN(INT_TO_LREAL(loop)*3.14);
    END_FOR
END_IF
```

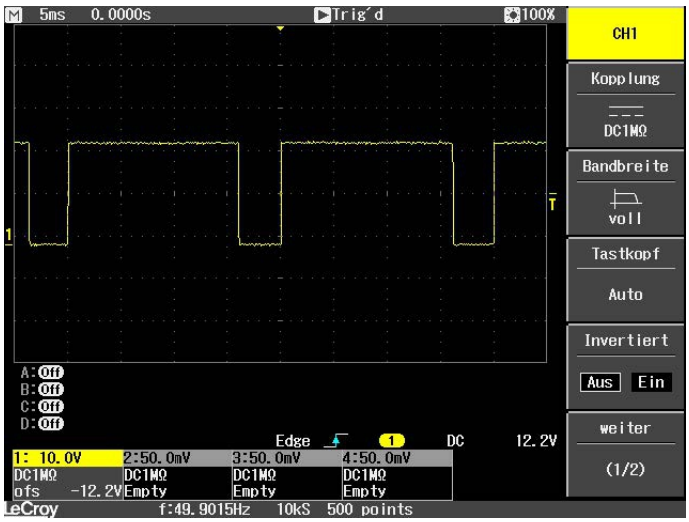


Abb. 62: Invertierte Darstellung eines digitalen Ausgangs.

Mit dem Attribut-Pragma `TcCallAfterOutputUpdate` ist der Puls konstant und ist unabhängig davon, wie lange die For-Schleife benötigt oder ob sie aufgerufen wird. Das Ganze funktioniert nur dann, wenn die SPS-Task nicht überschritten wird. Achten Sie also beim Reproduzieren des Beispiels auf die Überschreitungszähler der Task.

SPS-Programm mit unterschiedlichen Laufzeiten erkennen

Um SPS-Programme mit unterschiedlichen Laufzeiten zu erkennen, muss das SPS-Programm ergänzt werden. Im Online-View sind unterschiedliche Laufzeiten nicht erkennbar, da immer ein Mittelwert über mehrere Zyklen gebildet wird. Daher sind Ausreißer nur zu erkennen, wenn diese über der Taskzeit liegen. Liegen die Ausreißer noch innerhalb der Taskzeit, sind diese nicht ohne weiteres zu sehen.

Hierfür verwenden wir dann die Systemvariablen: PlcTaskSystemInfo

```
VAR
    bOut : BOOL;
    PlcTaskSystemInfo : PlcTaskSystemInfo;
    udiValue : ARRAY[0..19] of UDINT;
    Cnt : INT;
```

```
END_VAR

Program:
bOut:=not bOut;

IF bOut THEN
    For loop:=1 to 2000 do
        lrTest:=SIN(INT_TO_LREAL(loop)*3.14);
    END_FOR
END_IF

PlcTaskSystemInfo:=_TaskInfo[1];

udiValue[Cnt]:= PlcTaskSystemInfo.LastExecTime;
cnt:=cnt+1;
IF Cnt >19 THEN
        Cnt:=0;
END_IF
```

Mit dieser Programmerweiterung sieht man, dass das SPS-Programm mit For-Schleife 7,7 ms und ohne For-Schleife 1,1 ms benötigt. Die Angabe ist 100 ns pro Digit.

udiValue	ARRAY [0..19] OF U...	
udiValue[0]	UDINT	77728
udiValue[1]	UDINT	10713
udiValue[2]	UDINT	71049
udiValue[3]	UDINT	11065
udiValue[4]	UDINT	69882
udiValue[5]	UDINT	11027
udiValue[6]	UDINT	77084
udiValue[7]	UDINT	11939
udiValue[8]	UDINT	77494
udiValue[9]	UDINT	18527
udiValue[10]	UDINT	76724
udiValue[11]	UDINT	11043
udiValue[12]	UDINT	71519
udiValue[13]	UDINT	11406
udiValue[14]	UDINT	79004
udiValue[15]	UDINT	11118
udiValue[16]	UDINT	70745
udiValue[17]	UDINT	12007

Abb. 63: Ermittlung unterschiedlicher Lautzeiten beim SPS-Programm.

Die Messung deckt sich mit den Anzeigen auf dem Oszilloskop, auf denen erkennbar ist, dass ein Puls mal 6,5 ms länger bzw. 6,5 ms kürzer ist. Sie können die Bearbeitungszeit der For-Schleife messen (Bearbeitungszeit im SPS-Programm messen [▶ 171]). Das Ergebnis dieser Messung wird sich mit den beobachtet Werten durch die Programmerweiterung decken, mit einer gewissen Ungenauigkeit und Jitter.

9 Fehlerbehandlung und Diagnose

9.1 Diagnose-LEDs

Anzeige	LED	Farbe	Beschreibung
	TC	Grün	TwinCAT ist im Run-Modus.
		Rot	TwinCAT ist im Stop-Modus. Zeigt zusätzlich Fehler beim Systemstart durch Fehlercode und Fehlerargument an (siehe Tabelle: TC-LED, Fehlerbeschreibung und Abhilfe). Die LED blinkt rot mit zwei unterschiedlichen Frequenzen.
		Blau	TwinCAT ist im Konfig-Modus.
		Gelb	Fehler oder Absturz der SPS.
	FB	Grün	CAN ist in Ordnung.
		Rot	CAN im Bus-off.
		Grün und Rot blinkend, 200 ms	CAN-Warnung.
		Aus	CAN nicht konfiguriert.
	DIAG	Grün	Alle Nodes haben den NodeState = 0
		Grün und Rot blinkend, 200 ms	Alle Boxen im OP-Modus, die Task ist aber noch nicht gestartet.
		Rot 200 ms	Nicht alle Nodes im OP-Modus.
		Aus	Keine Boxen konfiguriert.
		Rot	Wenn beim Starten des CX70xx nur die DIAG-LED angeht, dann ist der Bootloader beschädigt und das Gerät muss eingeschickt werden.

Die TC-LED blinkt in einer festgelegten Frequenz und Reihenfolge und zeigt damit den Fehlercode und das Fehlerargument an.

Tab. 18: TC-LED, Reihenfolge und Bedeutung.

Reihenfolge	Bedeutung
Schnelles Blinken	Start der Sequenz
Erste langsame Sequenz	**Fehlercode**
Keine Anzeige	Pause, die LED ist aus
Zweite langsame Sequenz	**Fehlerargument**

Zählen Sie, wie oft die rote TC-LED blinkt, um den Fehlercode und das Fehlerargument zu ermitteln.

Tab. 19: TC-LED, Fehlerbeschreibung und Abhilfe.

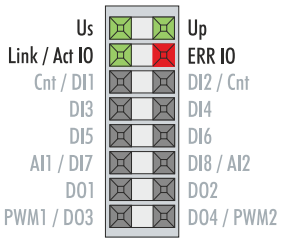
Fehlercode	Fehlerargument	Beschreibung	Abhilfe
1	1	MicroSD-Karte nicht erkannt	MicroSD-Karte prüfen.
	2	Card init failed - preloader	Image nicht in Ordnung. Installieren Sie ein neues Image auf der MicroSD-Karte.
	3	No partition found - preloader	
	4	Filesystem mount failed - preloader	
	5	Card init failed - loader	
	6	No partition found - loader	
	7	Filesystem mount failed - loader	
2	1	Loader not found	

Fehlercode	Fehlerargument	Beschreibung	Abhilfe
	2	Loader file invalid (checksum, size, read error)	
	3	TC dll not found	
	4	TC dll checksum error	
	5	EEPROM file missing or invalid	
	6	TcOsSys.dll version not compatible with loader	
3	1	Rbf not found	
	2	CCAT 1 init failed	
	3	CCAT 2 init failed	
	4	CCAT EEPROM writing failed	
	5	CCAT 1 EEPROM reloaded failed	
	6	CCAT 2 EEPROM reloaded failed	
4	1	Peripheral not working	Hardware defekt, tauschen sie den CX
	2	Voltage Vo not reached	
	3	Low speed external oscillator not running	
	4	High speed external oscillator not running	
	5	Flash failed	
	6	Device overclocked (old Hardware)	
5	5	RAM error detected	

9.1.1 K-Bus

Die angeschlossenen Busklemmen werden vom Netzteil auf Fehler überprüft. Die rote LED „K-BUS ERR“ ist aus, wenn keine Fehler vorhanden sind. Die rote LED „K-BUS ERR“ blinkt, wenn Fehler im Bereich der Busklemmen vorhanden sind.

Tab. 20: Diagnose-LEDs im K-Bus-Modus.

Anzeige	LED	Bedeutung
	Us 24V	Spannungsversorgung für CPU-Grundmodul. Die LED leuchtet grün bei korrekter Spannungsversorgung.
	Up 24V	Spannungsversorgung für Klemmenbus. Die LED leuchtet grün bei korrekter Spannungsversorgung.
	K-BUS RUN	Diagnose K-Bus. Die grüne LED leuchtet, um den fehlerfreien Betrieb anzuzeigen. Fehlerfrei bedeutet, dass auch die Kommunikation mit dem Feldbussystem fehlerfrei läuft.
	K-BUS ERR	Diagnose K-Bus. Die rote LED blinkt zur Fehleranzeige. Die rote LED blinkt mit zwei unterschiedlichen Frequenzen.

Durch die Frequenz und Anzahl des Blinkens kann der Fehlercode und das Fehlerargument ermittelt werden. Ein Fehler wird durch die LED „K-BUS ERR“ in einer festen Reihenfolge angezeigt.

Tab. 21: K-BUS ERR LED, Reihenfolge der Fehleranzeige durch die LED.

Reihenfolge	Bedeutung
Schnelles Blinken	Start der Sequenz
Erste langsame Sequenz	**Fehlercode**
Keine Anzeige	Pause, die LED ist aus

Reihenfolge	Bedeutung
Zweite langsame Sequenz	**Fehlerargument**

Zählen Sie, wie oft die rote LED K-BUS ERR blinkt, um den Fehlercode und das Fehlerargument zu ermitteln. Bei dem Fehlerargument zeigt die Anzahl der Impulse die Position der letzten Busklemme vor dem Fehler an. Passive Busklemmen, wie zum Beispiel eine Einspeiseklemme, werden nicht mitgezählt.

Tab. 22: K-BUS ERR LED, Fehlerbeschreibung und Abhilfe.

Fehlercode	Fehlerargument	Beschreibung	Abhilfe
Ständiges, konstantes Blinken		EMV Probleme.	• Spannungsversorgung auf Unter- oder Überspannungsspitzen kontrollieren. • EMV-Maßnahmen ergreifen. • Liegt ein K-Bus-Fehler vor, kann durch erneutes Starten (Aus- und Wiedereinschalten des Netzteils) der Fehler lokalisiert werden.
3 Impulse	0	K-Bus-Kommandofehler.	• Keine Busklemme gesteckt. • Eine der Busklemmen ist defekt, angehängte Busklemmen halbieren und prüfen ob der Fehler bei den übrigen Busklemmen noch vorhanden ist. Dieses Vorgehen wiederholen, bis die defekte Busklemme lokalisiert ist.
4 Impulse	0	K-Bus-Datenfehler, Bruchstelle hinter dem Netzteil.	Kontrollieren, ob die Busendklemme 9010 gesteckt ist.
	n	Bruchstelle hinter Busklemme n.	Prüfen, ob die Busklemme n+1 hinter dem Netzteil richtig gesteckt ist, gegebenenfalls tauschen.
5 Impulse	n	K-Bus-Fehler bei Register-Kommunikation mit Busklemme n.	Busklemme an Stelle n tauschen.
6 Impulse	0	Fehler bei der Initialisierung.	Embedded-PC tauschen.
	1	Interner Datenfehler.	Hardware-Reset des Embedded-PCs (aus - und wieder einschalten).
	8	Interner Datenfehler.	Hardware-Reset des Embedded-PCs (aus- und wieder einschalten).
7 Impulse	0	Prozessdatenlängen der Soll- und Ist-Konfiguration stimmen nicht überein.	Konfiguration und Busklemmen auf Konsistenz prüfen.

Bei manchen Fehlern geht die LED „K-BUS ERR“ nicht aus, obwohl der Fehler beseitigt wurde. Schalten Sie die Spannungsversorgung für das Netzteil aus und wieder ein, damit die LED nach der Fehlerbeseitigung ausgeschaltet wird.

State-Variable

In TwinCAT gibt es unter dem Buskoppler die Variable State, für die K-Bus-Diagnose.

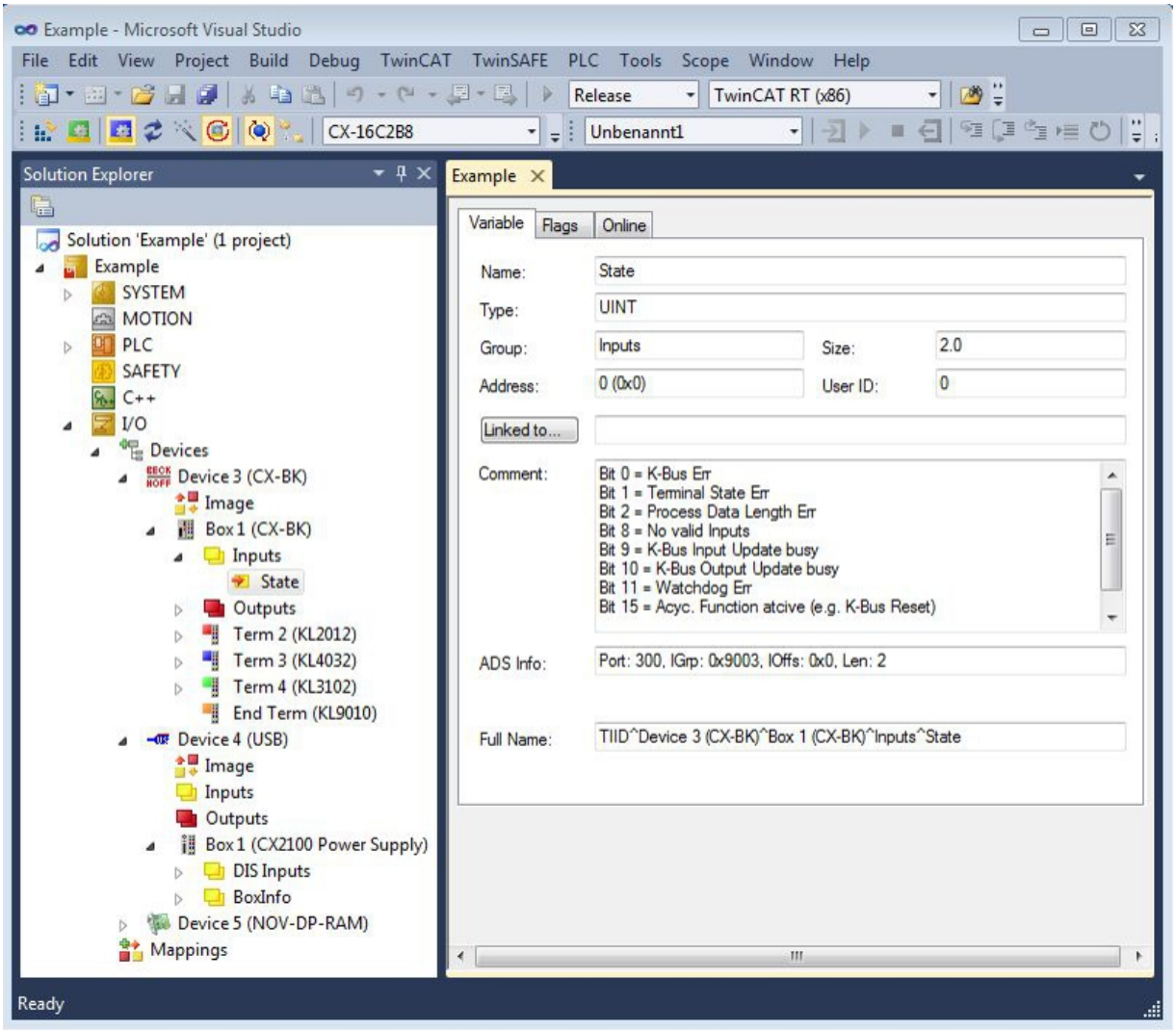


Abb. 64: Status-Variable für Fehlerbehandlung und Diagnose unter TwinCAT.

Ist der Wert „0“ so arbeitet der K-Bus synchron und ohne Fehler. Sollte der Wert <> „0“ sein, kann ein Fehler vorliegen. Es kann aber auch nur ein Hinweis sein, das zum Beispiel der K-Bus-Zyklus länger dauert, als die verwendete Task. Damit ist er dann nicht mehr synchron zu der Task. Die Taskzeit sollte schneller als 100 ms sein. Wir empfehlen eine Taskzeit kleiner 50 ms. Typischerweise liegt die K-Bus-Update-Zeit zwischen einer und fünf ms.

Tab. 23: Beschreibung der Werte bei der State-Variable.

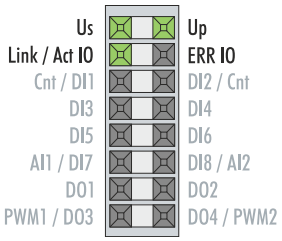
Bit	Beschreibung
Bit 0	K-Bus-Fehler.
Bit 1	Klemmenkonfiguration hat sich seit dem Start geändert.
Bit 2	Prozessabbildlängen stimmen nicht überein.
Bit 8	(noch) keine gültigen Eingänge.
Bit 9	K-Bus ist im Inputupdate noch nicht fertig.
Bit 10	K-Bus ist im Output-Update noch nicht fertig.
Bit 11	Watchdog.
Bit 15	azyklische K-Bus-Funktion aktiv (z.B. K-Bus-Reset).

Liegt ein K-Bus-Fehler vor, kann dieser über den Funktionsbaustein IOF_DeviceReset (in der TcIoFunctions.lib) zurückgesetzt werden.

9.1.2 E-Bus

Die angeschlossenen EtherCAT-Klemmen werden vom Netzteil überprüft. Im E-Bus-Modus leuchtet die LED „Link/Act IO“. Wenn Daten übertragen werden, blinkt die LED „Link/Act IO“.

Tab. 24: Diagnose-LEDs im K-Bus-Modus.

Anzeige	LED		Bedeutung
	Us		Spannungsversorgung für CPU-Grundmodul. Die LED leuchtet grün bei korrekter Spannungsversorgung.
	Up		Spannungsversorgung für Klemmenbus. Die LED leuchtet grün bei korrekter Spannungsversorgung.
	Link/Act IO	aus	E-Bus nicht angeschlossen.
		an	E-Bus angeschlossen / Kein Datenverkehr.
		blinkt	E-Bus angeschlossen / Datenverkehr auf dem E-Bus.

9.2 CANopen-Diagnose

9.2.1 Statusmeldungen

Die CANopen-Statusmeldungen liefern zusätzliche Informationen und können für Diagnosezwecke eingesetzt werden.

Die folgende Tabelle zeigt welche Werte die Variablen annehmen können:

Eingänge	Bedeutung
CycleInfo	Cycle Counter: Dieser Zähler wird nach jedem Zyklus um eins erhöht. Error: Zeigt die Anzahl der Boxen mit einem BoxState ungleich Null an. ActualCycle Time: Für zukünftige Verwendung reserviert
DiagFlag	Diese Variable liefert Informationen darüber, ob sich die Diagnosedaten verändert haben. • 0: Daten nicht verändert. • 1: Daten verändert. Benutze ADS-Read, um die Daten auszulesen.
GlobalState	Diese Variable liefert Informationen über den Status des Masters. GlobalState[0]: 0: Gerät ist im Status RUN. 1: Gerät ist im Status RESET. 2: Gerät ist im Status OFFLINE. 3: Gerät ist im Status STOP. GlobalState[1] (FW V02.14 und höher): Bit 0-7: RxError-Counter des CAN-Controllers. Bit 8-15: TxError-Counter des CAN-Controllers. GlobalState[2]: Bit 0: CAN-Controller ist im BUS-OFF. Bit 1: CAN-Controller Warnbegrenzung erreicht. Bit 2: Rx-Queue überschritten. Bit 3: Hi-Prio Tx-Queue überschritten. Bit 4: Lo-Prio Tx-Queue überschritten. Bit 5: CAN-Send Error (FW V02.14 und höher). Bit 6-14: für zukünftige Verwendung reserviert. Bit 15: schaltet bei jeder gesendeten SYNC-Nachricht. GlobalState[3]: Bus Auslastung in %.
CycleFailedCounter	Dieser Zähler wird jedes Mal um eins erhöht, wenn am Anfang eines TwinCAT-Zyklus der letzte Bus-Zyklus nicht abgeschlossen wurde.
BusLoad	Bus Auslastung in %.
InfoData	

9.2.2 Kommunikation

In der Strukturansicht werden unter dem Menüpunkt **Inputs** Eingangsvariablen aufgelistet, die Informationen über einen CANopen-Gerät zur Verfügung stellen.

Über die Variable NodeState können Sie sich den Zustand der CANopen-Kommunikation anzeigen lassen und wissen damit, ob sich der Slave im Datenaustausch befindet oder einen Fehler vorliegt.

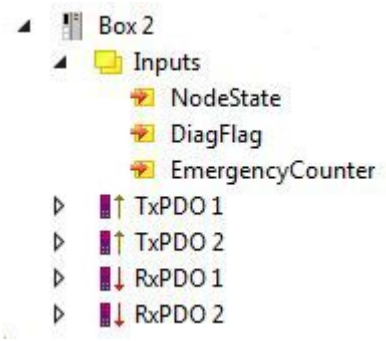


Abb. 65: Diagnose der CANopen-Kommunikation mit den Variablen NodeState, DiagFlag und EmergencyCounter.

NodeState

Die folgende Tabelle zeigt, welche Werte die Variable NodeState annehmen kann:

Wert	Bedeutung
0	No error
1	Node deactivated
2	Node not found
4	SDO syntax error at StartUp
5	SDO data mismatch at StartUp
8	Node StartUp in progress
11	FC510x Bus-OFF
12	Pre-Operational
13	Severe bus fault
14	Guarding: toggle error
20	TxPDO too short
22	Expected TxPDO is missing
23	Node is Operational but not all TxPDOs were received
31	only for EtherCAT gateways: WC-State of cyclic EtherCAT frame is 1
128	Node is Operational but not all RxPDOs were received
129	Node is Pre-Operational
130	Node is Stopped

DiagFlag

Die folgende Tabelle zeigt, welche Werte die Variable DiagFlag annehmen kann. Diese Variable liefert Informationen darüber, ob sich die Diagnosedaten verändert haben.

Wert	Bedeutung
0	Daten nicht verändert.
1	Daten verändert. Benutze ADS-Read, um die Daten auszulesen.

EmergencyCounter

Bei der Variable EmergencyCounter wird der Zähler um eins erhöht, wenn ein Emergency Telegramm erhalten wurde.

Tab. 25: Auslesen der Emergency-Telegramme mit dem Funktionsbaustein ADSREAD.

Eingangsparameter	Beschreibung
NETID	NetId der CAN-Schnittstelle
Port Nummer	200
IDXGRP	16#xxxxF180 (xxxx) Node-Id, das Diag Flag wird nur beim Auslesen von mindesten 106 Byte zurückgesetzt 16#xxxxF181 (xxxx) Node-Id, das Diag Flag wird sofort zurückgesetzt
IDXOFFS	Byte Offset

Tab. 26: Beschreibung des Arrays

Offset	Bit	Wert / Beschreibung
0 - 1	Bit 0	reserviert
	Bit 1	Boot-Up-Message nicht empfangen oder fehlerhaft
	Bit 2	Emergency-Overflow
	Bit 3 - 15	reserviert
2 - 3	Bit 0 - 14	TX-PDO (i+1) empfangen
	Bit 15	alle TX-PDOs 16-n empfangen
4 - 5	Bit 0 - 4	1: falsche TX-PDO-Länge
		2: synchrone TX-PDO fehlt
		3: Node meldet PRE-OPERATIONAL
		4: Event-Timer bei einer TX-PDO abgelaufen
		5: keine Antwort beim Guarden
		6: mehrmals kein Toggeln beim Guarden
	Bit 5 - 15	zugehörige COB-ID
6	Bit 0 - 7	1: falscher Wert bei einem SDO-Upload
		2: falsche Länge bei einem SDO-Upload
		3: Abort bei einem SDO-Up-/Download
		4: falsches Datum bei einer Boot-Up-Message
		5: Timeout beim Warten auf Boot-Up-Message
7	Bit 0 - 7	2: falscher SDO-Command specifier
		3: SDO-Toggle-Bit hat sich nicht geändert
		4: SDO-Länge zu groß
		5: SDO-Abort
		6: SDO-Timeout
8 - 9	Bit 0 - 7	Index des SDO-Up/Downloads
10	Bit 0 - 7	Subindex des SDO-Up/Downloads
11	Bit 0 - 7	reserviert
12	Bit 0 - 7	errorClass des Aborts
13	Bit 0 - 7	errorCode des Aborts
14 - 15	Bit 0 - 15	additionalCode des Aborts
16 - 19		gelesener Wert (falls Offset 6 = 1)
20 - 23		erwarteter Wert (falls Offset 6 = 1)
24 - 25		Anzahl der folgenden Emergencies
26 - n		Emergencies (jeweils 8 Byte)

9.2.3 PDOs

SendCounter

Bei den TxPDOs gibt es zusätzlich unter dem Menüpunkt **Control** die Variable SendCounter.

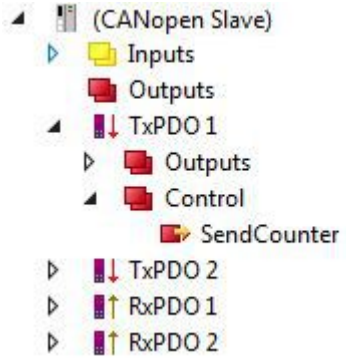


Abb. 66: Diagnosevariable SendCounter eines CANopen-Slaves.

Standardmäßig werden PDOs automatisch bei einer Änderung gesendet. Diese Variable kann verwendet werden, wenn die Daten nicht nur bei einer Änderung gesendet werden sollen, sondern auch dann, wenn sich an den Daten im PDO nichts geändert hat.

Die Variable muss also um eins hochgezählt werden, wenn die Daten im PDO auch ohne Änderung gesendet werden sollen. Sollte im gleichen Zyklus die Variable hochgezählt werden und parallel dazu eine Änderung der Daten erfolgen, wird nur ein Telegramm gesendet.

Abgesehen von diesem Szenario kann mit dieser Variable überwacht werden, ob das entsprechende PDO bei einer Änderung der Daten gesendet wurde.

ReceiveCounter

Bei den RxPDO gibt es zusätzlich unter dem Menüpunkt **Status** die Variable ReceiveCounter.

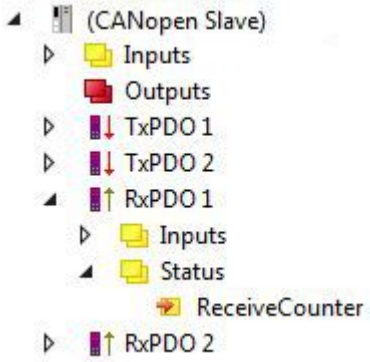


Abb. 67: Diagnosevariable ReceiveCounter eines CANopen-Slaves.

Die Eingangsvariable wird immer um eins hochgezählt, wenn ein PDO eintrifft. Auf diese Weise werden immer neu eingetroffene PDO erfasst, auch wenn sich z.B. die Daten im PDO nicht ändern. Zusätzlich gibt die Variable Auskunft darüber, ob ein Teilnehmer noch regelmäßig Daten sendet. Es ist nützlich diese Variable mit der SPS zu verknüpfen und zu überwachen.

9.2.4 Fehlersuche

Error Frames

Fehler in der CAN-Verkabelung, der Adressvergabe und der Baud-Rateneinstellung zeigen sich u.a. durch eine erhöhte Anzahl an Error Frames: die Diagnose LEDs zeigen dann *Warning Limit wird überschritten* oder *Bus-Off-Zustand erreicht*.

Error Frames

Überschrittenes Warning Limit, Error Passive oder Bus-Off Zustand werden zunächst bei demjenigen Knoten angezeigt, der die meisten Fehler entdeckt hat. Dieser Knoten muss nicht unbedingt die Ursache für das Auftreten dieser Error Frames sein!
Wenn z. B. ein Knoten überdurchschnittlich stark zum Busverkehr beiträgt (z. B. weil er als einziger über analoge Eingänge verfügt, deren Daten in kurzen Abständen ereignisgesteuerte PDOs auslösen), so werden auch seine Telegramme mit großer Wahrscheinlichkeit zunächst gestört - entsprechend erreicht sein Fehlerzähler als erster kritische Zustände.

Node-ID / Baud Rate Einstellung

Es muss sorgfältig darauf geachtet werden, dass keine Knotenadresse doppelt vergeben ist: für jedes CAN-Datentelegramm darf es nur einen Sender geben.

Test 1

Knotenadressen überprüfen. Falls die CAN-Kommunikation wenigstens zeitweise funktioniert und alle Geräte die Boot-Up-Nachricht unterstützen, so kann die Adressvergabe auch durch Aufzeichnen der Boot-Up-Nachrichten nach dem Einschalten der Geräte überprüft werden - hierdurch wird aber kein Vertauschen von Knotenadressen erkannt.

Test 2

Überprüfen, ob überall die gleiche Baud-Rate eingestellt ist. Bei Sondergeräten: Wenn Bittiming Parameter zugänglich, stimmen diese mit den CANopen-Definitionen überein (Abtastzeitpunkt, SJW, Oszillator).

Test der CAN-Verkabelung

Diese Tests nicht ausführen, wenn das Netzwerk aktiv ist: Während der Tests sollte keine Kommunikation stattfinden. Die folgenden Tests sollten in der angegebenen Reihenfolge ausgeführt werden, da manche Tests davon ausgehen, dass der vorhergehende Test erfolgreich war. In der Regel sind nicht alle Tests notwendig.

Netzwerkabschluss und Signalleitungen

Für diesen Test sollten die Knoten ausgeschaltet oder die CAN-Leitung abgesteckt sein, da die Messergebnisse sonst durch die aktiven CAN-Transceiver verfälscht werden können.

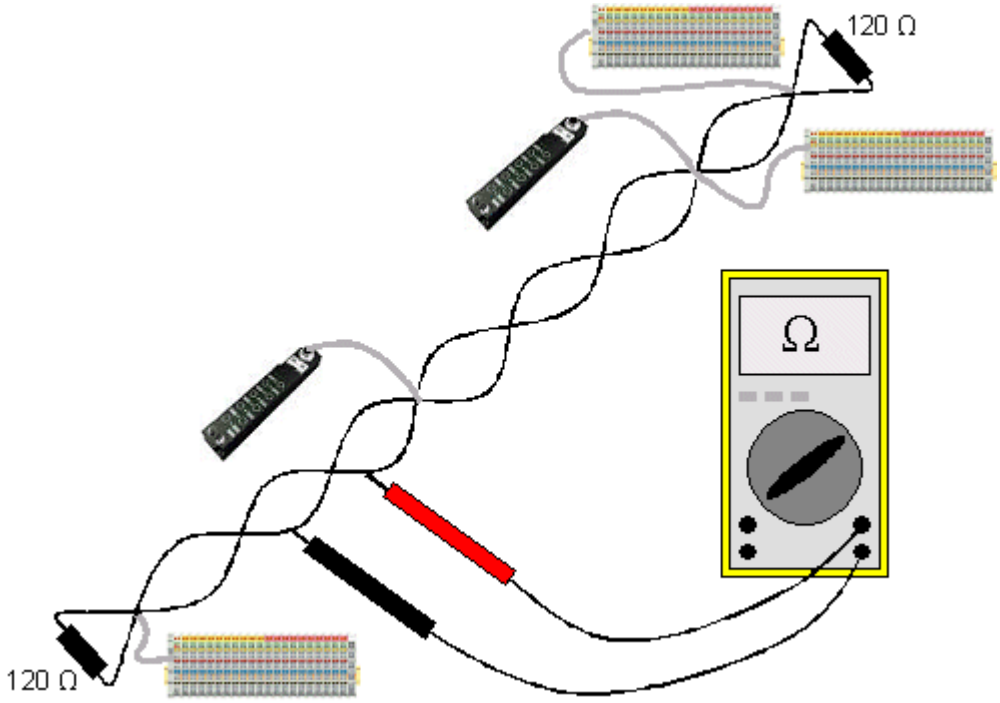


Abb. 68: Verdrahtungsplan für Testaufbau

Test 3

Widerstand zwischen CAN-high und CAN-low ermitteln - ggf. bei jedem Gerät.

Wenn der Messwert über 65 Ohm liegt, deutet dies auf fehlende Abschlusswiderstände oder den Bruch einer Signalleitung hin. Wenn der Messwert kleiner 50 Ohm ist, nach Kurzschluss zwischen CAN-Leitung, überzähligen Abschlusswiderständen oder fehlerhaften Transceivern suchen.

Test 4

Auf Kurzschluss zwischen CAN-Ground und den Signalleitungen sowie zwischen Schirm und Signalleitungen prüfen.

Test 5

Erdung von CAN-Ground und Schirm auftrennen. Auf Kurzschluss zwischen CAN-Ground und Schirm prüfen.

Topologie

Die Leitungslänge bei CAN-Netzwerken hängt stark von der gewählten Baud-Rate ab. CAN toleriert dabei kurze Stichleitungen - ebenfalls in Abhängigkeit von der Baud-Rate. Die erlaubte Stichleitungslänge sollte nicht überschritten werden. Häufig wird die verlegte Leitungslänge unterschätzt - die Schätzung liegt teilweise Faktor 10 unter der tatsächlichen Länge. Deshalb empfiehlt sich folgender Test:

Test 6

Die Stichleitungslängen sowie die Busgesamtlänge nachmessen, nicht nur grob schätzen und mit den Topologie-Regeln (Baud-Ratenabhängig) vergleichen.

Schirmung und Erdung

Stromversorgung und Schirm sollten sorgfältig, einmalig und niederohmig beim Netzteil geerdet werden. Alle Verbindungsstellen, Abzweige etc. im CAN-Kabel müssen neben den Signalleitungen (und evtl. CAN-GND) auch den Schirm durchverbinden. In den Beckhoff IP20 Buskopplern wird der Schirm über ein R/C-Glied hochfrequenzmäßig geerdet.

Test 7

Mit DC-Strommessgerät (16 Amp max.) Strom zwischen Spannungsversorgungs-Masse und Schirm am vom Netzteil entfernten Ende des Netzes messen. Es sollte ein Ausgleichsstrom vorhanden sein. Wenn kein Strom vorhanden ist, so ist der Schirm nicht durchgängig verbunden oder das Netzteil ist nicht richtig geerdet. Wenn das Netzteil in der Mitte des Netzwerkes angeordnet ist, so sollte an beiden Enden gemessen werden. Dieser Test kann u.U. auch an den Stichleitungsenden durchgeführt werden.

Test 8

Den Schirm an mehreren Stellen auftrennen und den Verbindungsstrom messen. Wenn ein Stromfluss vorhanden ist, so ist der Schirm an mehreren Stellen geerdet (Erdschleife)

Potentialunterschiede

Der Schirm muss für diesen Test durchgängig sein und darf keinen Strom führen (vorher getestet).

Test 9

Spannung zwischen Schirm und Spannungsversorgungs-Erde an jedem Knoten ermitteln und notieren. Der maximale Potentialunterschied zwischen zwei beliebigen Geräten sollte kleiner als 5 Volt sein.

Fehler erkennen und lokalisieren

Am besten funktioniert in der Regel der "Low-tech-Ansatz": Teile des Netzes abhängen und beobachten, wann der Fehler verschwindet.

Aber: Dies funktioniert nicht gut bei Problemen wie zu großen Potentialunterschieden, Masseschleifen, EMV und Signalverfälschung da die Verkleinerung des Netzes häufig das Problem löst, ohne dass der „fehlende“ Teil ursächlich war. Auch die Buslast ändert sich beim Verkleinern des Netzes - damit können externe Störungen seltener CAN-Telegramme "treffen".

Die Diagnose mittels Oszilloskop führt meist nicht zum Erfolg: CAN-Signale sehen auch im ungestörten Zustand teilweise recht wirr aus. Unter Umständen kann mit einem Speicheroszilloskop auf Error Frames getriggert werden - diese Art der Diagnose ist aber Messtechnik-Experten vorbehalten.

Protokollprobleme

In seltenen Fällen sind auch Protokollprobleme (z. B. fehlerhafte oder unvollständige CANopen-Implementierung, unglückliches Timing im Boot-Up etc.) Ursache von Störungen. Hier ist dann ein Mitschrieb (Trace) des Busverkehrs mit anschließender Auswertung durch CANopen Experten erforderlich - das Beckhoff Support Team kann hier helfen.
Für solch einen Trace eignet sich ein freier Kanal einer Beckhoff FC5102 CANopen PCI-Karte - die erforderliche Trace-Software stellt Beckhoff im Internet zur Verfügung. Alternativ kann selbstverständlich auch ein handelsübliches CAN-Analysetool eingesetzt werden.

Protokollprobleme lassen sich vermeiden, indem auf den Einsatz von Geräten verzichtet wird, die nicht Conformance getestet sind. Der offizielle CANopen-Conformance-Test und das entsprechende Zertifikat sind beim CAN in Automation Verband (https://www.can-cia.org) erhältlich.

9.3 Diagnose der Multifunktions-I/Os

Dieses Kapitel beschreibt die Diagnosemöglichkeiten der Multifunktions-I/O-Kommunikation. Das ist beispielsweise dann wichtig, wenn die 24-V-Spannungsversorgung für die Multifunktions-I/Os ausfällt oder die Sicherung ausgelöst wird.

Statusvariable

Die Statusvariable `state` kann für Diagnosezwecke eingesetzt werden. Im Normalzustand nimmt die Statusvariable den Wert `0x___8` (OP, Operational) an und zeigt damit an, dass alles Fehlerfrei ist.

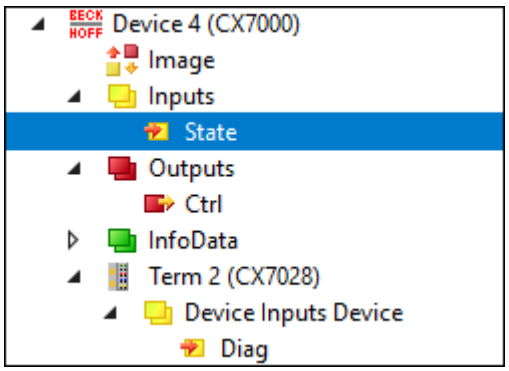


Abb. 69: Multifunktions-I/O Statusvariable.

Die folgende Tabelle zeigt, welche Werte die Variable annehmen kann:

Wert	Bedeutung
0x___1	Slave in 'INIT' state
0x___2	Slave in 'PREOP' state
0x___3	Slave in 'BOOT' state
0x___4	Slave in 'SAFEOP' state
0x___8	Slave in 'OP' state
0x001_	Slave signals error
0x002_	Invalid vendorId, productCode... read
0x004_	Initialization error occurred
0x010_	Slave not present

Sollte es zu einem Ausfall der Spannungsversorgung kommen, gehen die Multifunktions-I/Os nicht automatisch wieder in den Datenaustausch. Dafür müssen die Multifunktions-I/Os zurückgesetzt werden. Ein Funktionsbaustein, mit dem die Multifunktions-I/Os zurückgesetzt werden können, ist der Funktionsbaustein FB_CX70xx_ResetOnBoardIO [▶ 178].

Hinweis : Sind in der SPS noch Ausgänge gesetzt, dann werden die Ausgänge der Multifunktions-I/Os sofort wieder aktiv, sobald die Multifunktions-I/Os mit dem Funktionsbaustein zurückgesetzt werden.

Weitere Diagnosevariablen

Die Diagnosevariablen `Diag` und `TxPDO State` sind derzeit nicht in Gebrauch und für zukünftige Verwendung reserviert. Die Variable `Input cycle counter` hingegen erhöht sich mit jedem Zyklus und zeigt die Anzahl der I/O-Zyklen an, die mit den Multifunktions-I/Os ausgetauscht werden. Sobald die Variable nicht mehr inkrementiert wird, werden keine I/O-Zyklen mehr mit den Multifunktions-I/Os ausgetauscht.

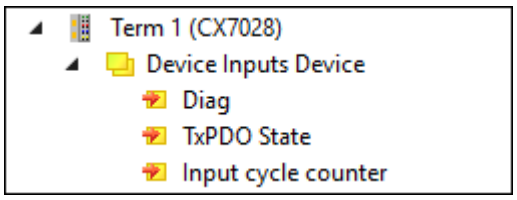


Abb. 70: Weitere Diagnosevariablen für Multifunktions-I/Os

Variable	Bedeutung
Diag	Reserviert, wird aktuell nicht verwendet.
TxPDO State	Reserviert, wird aktuell nicht verwendet.
Input cycle counter	Wird mit jedem Zyklus um 1 inkrementiert. Wenn dieser Zähler stehen bleibt, dann werden keine I/O-Zyklen mehr mit den Multifunktions-I/Os ausgetauscht.

9.4 Speicherauslastung

Der CX7050 hat 32 MB Arbeitsspeicher, der von der Firmware (TC/RTOS) und von TwinCAT (TwinCAT-Speicher) verwendet wird. Der TwinCAT-Speicher unterteilt sich weiter in den Router-Speicher und den SPS-Speicher. Der Router-Speicher wird für die ADS-Kommunikation und der SPS-Speicher für das eigentliche SPS-Programm inklusive TcConfiguration, Mapping und Daten verwendet.

Dem CX7050, stehen 19,1 MB TwinCAT-Speicher zur Verfügung. Weil die Größe des Speichers begrenzt ist, ist es wichtig die Speicherauslastung zu kontrollieren und ihr SPS-Projekt bei einer Überschreitung anzupassen.

Router-Speicher

Sie können zum einen die Größe des Router-Speichers in TwinCAT anpassen und abhängig von der tatsächlich verwendeten ADS-Kommunikation, einen kleineren Router-Speicher einstellen.

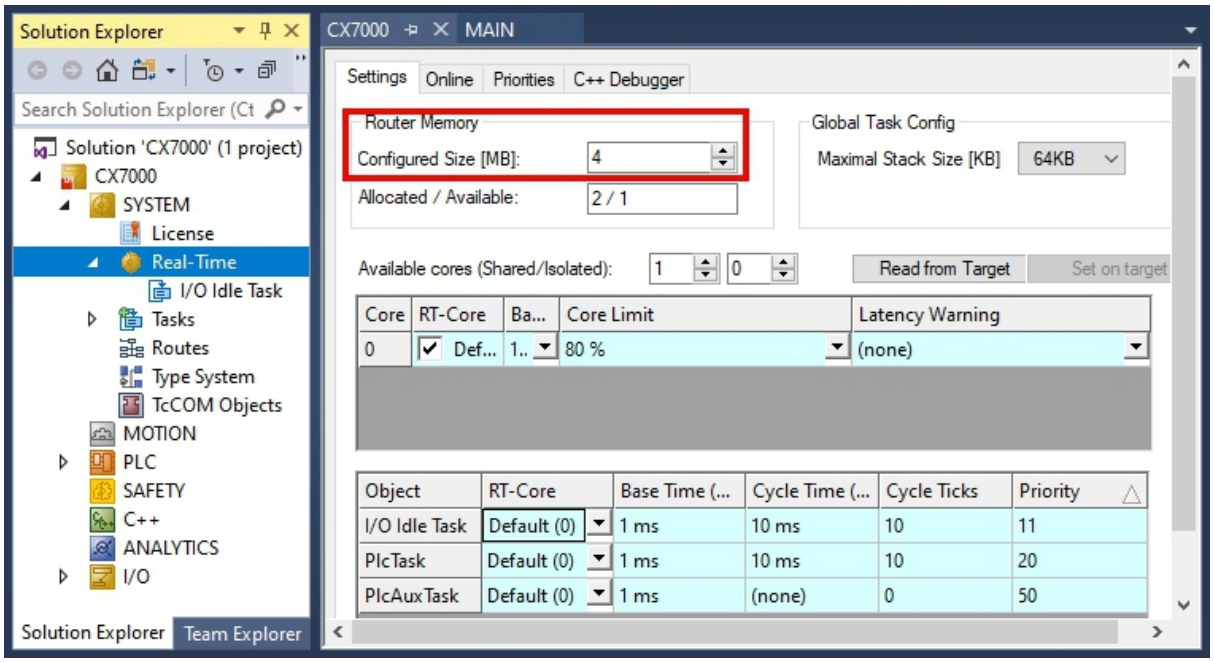


Abb. 71: Einstellungen für Router-Speicher im TwinCAT System Manager.

Standardmäßig wird in TwinCAT ein Wert von 32 MB eingetragen, der beim CX7050 wiederum auf 9 MB begrenzt wird, wegen des kleinen Arbeitsspeicher beim CX7050. Ein Router-Speicher von 9 MB ist in der Regel viel zu groß für eine Kleinsteuerung. Beim CX7050 ist ein Router-Speicher von 4 MB empfehlenswert und kann unter Umständen sogar noch kleiner gewählt werden, wenn wenig bis gar keine ADS-Kommunikation verwendet wird. Ein Router-Speicher von mindestens 1 MB sollte jedoch eingehalten und nicht unterschritten werden. Wieviel Router-Speicher verwendet wird, lässt sich mit dem Funktionsbaustein FB_GetRouterStatusInfo oder alternativ mit dem Beckhoff Device Manager kann ermitteln.

Beachten Sie, dass der Router-Speicher erst mit einem Power Off/On des CX7050 neu angelegt wird. Ein TwinCAT-Neustart reicht nicht aus. Als Faustregel gilt: Je kleiner der Router-Speicher für die ADS-Kommunikation gewählt wird, desto größer kann die Applikation sein, also das SPS-Programm, TcConfiguration, Mapping und Daten.

Speicherauslastung bestimmen

Mit dem Funktionsbaustein FB_GetRouterStatusInfo oder alternativ mit dem Beckhoff Device Manager kann ermittelt werden, wie groß der Speicherbedarf des Router-Speichers ist.

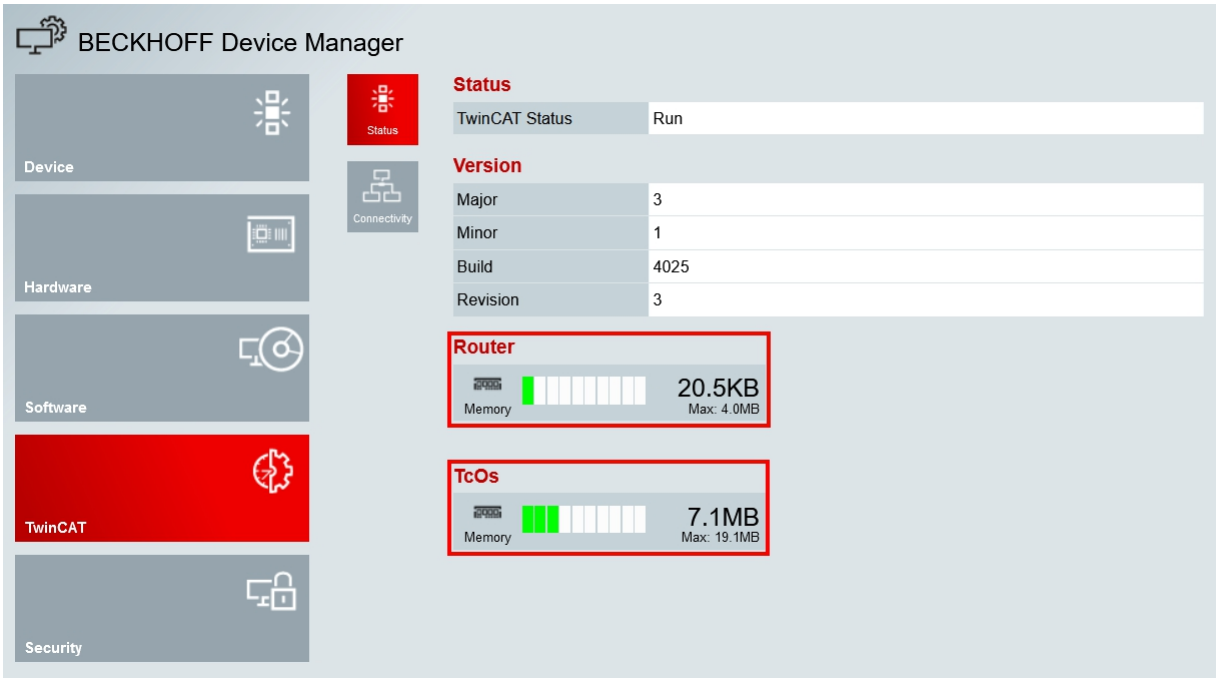


Abb. 72: Auslastung des Router-und TwinCAT-Speichers.

Über die Anzeige **Router** kann der Speicherbedarf des Router-Speichers bestimmt werden. In diesem Beispiel werden 20,5 kB von maximal 4 MB belegt. Die Anzeige **TcOs** zeigt den gesamten Speicherverbrauch des TwinCAT-Speichers inklusive Router-Speicher und SPS-Programm an. In diesem Beispiel werden insgesamt 7,1 MB belegt.

Mit Hilfe dieser Anzeige, kann auch die Größe des SPS-Programms berechnet werden, da der Router-Speicher mit 4 MB fest definiert und Teil des TwinCAT-Speichers ist. Zieht man die 4 MB von 7,1 MB ab, so belegt das SPS-Programm 3,1 MB.

Speicherreserve

Da in diesem Beispiel beim TwinCAT-Speicher 7,1 MB von 19,1 MB belegt werden, bleibt eine Reserve von 12 MB für das SPS-Programm. Beachten Sie, dass für ein Online-Change in TwinCAT kurzzeitig mehr Speicher gebraucht wird. Wenn Sie die Funktion Online-Change verwenden wollen, ist es ratsam immer eine gewisse Reserve vorzuhalten. Im extremsten Fall kann das Doppelte des aktuell verbrauchten SPS-Programms erforderlich sein, um ein Online-Change auszuführen. In TwinCAT wird eine Fehlermeldung angezeigt, sobald nicht genügend Speicher für den Online-Change zur Verfügung steht.

9.5 Echtzeit und CPU-Auslastung

Für die einwandfreie Funktionsweise des CX7050 ist es wichtig, die CPU-Auslastung und die Einhaltung der Echtzeit im Blick zu behalten. Andernfalls arbeitet der CX7050 bei einer Überlastung nicht mehr zuverlässig. Beachten Sie, dass bei einer Überlastung auch die Auslastungsanzeige betroffen ist und keine aktuellen Werte mehr liefert. So kann fälschlicherweise eine Auslastung von 40 %angezeigt werden, die SPS aber bereits nicht mehr in Echtzeit arbeiten und das System überlastet sein. Sie sollten sich also bei einer Kleinsteuerung schrittweise an die Belastungsgrenze herantasten.

Was versteht man in diesem Zusammenhang unter Echtzeit? Die SPS arbeitet standardmäßig zyklussynchron, das bedeutet, dass immer eine Taskzeit definiert und zu einem festen Zeitpunkt aufgerufen wird. Die SPS arbeitet zyklussynchron, wenn die Taskzeit nicht überschritten wird. Wenn Sie beispielsweise eine Taskzeit von 10 ms definieren und die SPS nur 2 ms für die Bearbeitung braucht, ist die gewählte Taskzeit in Ordnung und die SPS arbeiten zyklussynchron.

Auch wenn Sie die Echtzeit nicht brauchen, ist es empfehlenswert die Echtzeit einzuhalten, weil es sonst zu negativen Effekten kommen kann. Das können Verbindungprobleme oder Probleme mit Subsystemen wie K-Bus oder EtherCAT sein. Folgende Schritte können Sie durchführen, um zu überprüfen, ob der CX7050 optimal eingestellt oder eher überlastet ist:

- Exceed-Counter beobachten.
- CPU-Auslastung kontrollieren.

Exceed-Counter beobachten

Sobald die SPS nicht mehr zyklussynchron arbeitet und die definierte Taskzeit überschritten wird, wird der Exceed-Counter hochgezählt. Im Idealfall sollte der Zählerstand Null sein.

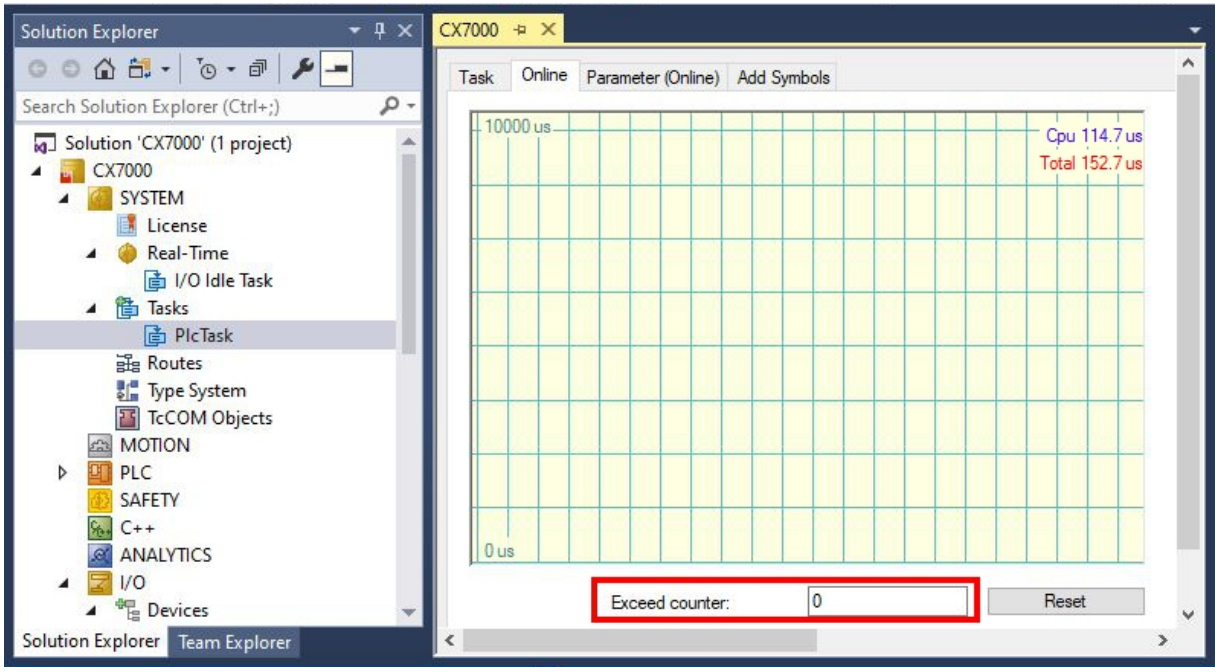


Abb. 73: Anzeige des Exceed-Counters in TwinCAT.

Es kann vorkommen, dass der Exceed-Counter beim Start der SPS hochgezählt wird, weil die SPS beispielsweise zum ersten Mal aufgerufen wird oder bestimmte Bestandteile initialisiert werden. Beobachten Sie den Exceed-Counter über einen Zeitraum von mehreren Stunden. Erst wenn der Exceed-Counter über einen längeren Zeitraum nicht mehr hochgezählt wird, kann von einem stabilen Zustand gesprochen werden.

CPU-Auslastung kontrollieren

In TwinCAT wird unter Realtime und der Registerkarte Online, die CPU-Auslastung angezeigt. Kontrollieren Sie den Wert, um zu ermitteln, ob Sie zusätzlichen Programmcode ausführen oder die Taskzeit verkürzen können.

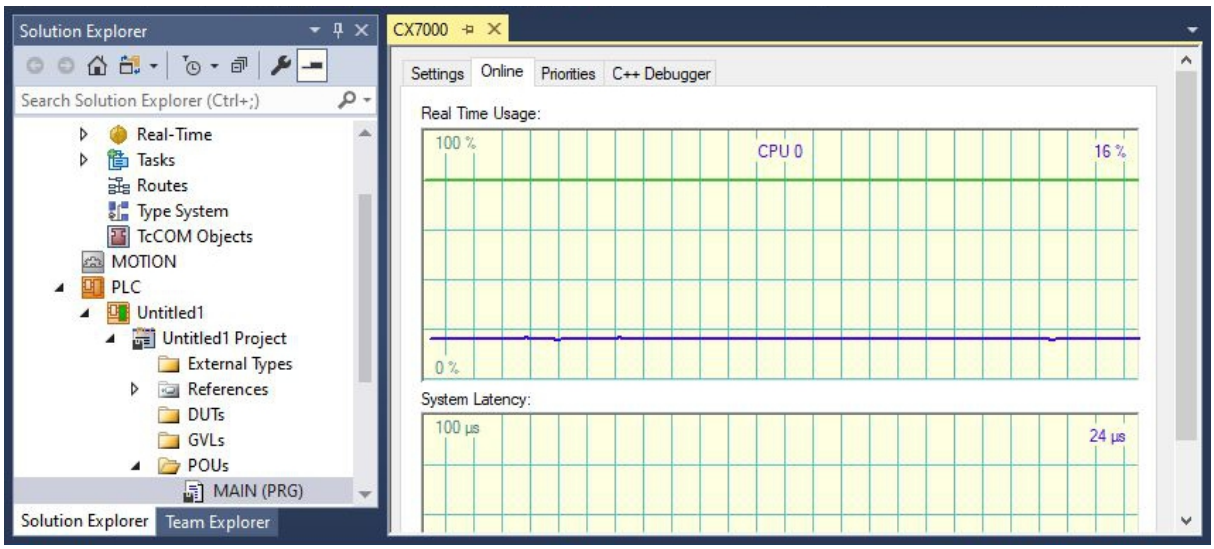


Abb. 74: Anzeige der CPU-Auslastung in TwinCAT

Die hellgrüne Linie zeigt das voreingestellte CPU-Limit an. Wenn die Auslastung ≥ 65 % beträgt, ist der CX7050 bereits gut ausgelastet und es sollte kein weiterer Programmcode ausgeführt oder die Taskszeit verkürzt werden. Sie sollten nicht bis an die Grenzen gehen und den CX7050 voll auslasten.

Maßnahmen bei einer Überlastung

Wird eine Überlastung mit Hilfe der gezeigten Schritte festgestellt, kann die Auslastung durch eine verbesserte Programmierung oder eine Erhöhung der Taskzeit reduzieren werden. Um Stellen im Programmcode mit langen Bearbeitungszeiten zu finden, kann das Beispiel unter: Bearbeitungszeit im SPS-Programm messen [▶ 171] genutzt werden.

Einen Einfluss auf die Echtzeit, hat auch das gewählte Klemmensystem. Abhängig von der Anzahl der Klemmen, kann beispielsweise der K-Bus zusätzlich mehrere Millisekunden beanspruchen und muss bei der Wahl der Taskzeit berücksichtig werden. Es kann durchaus sein, dass bei einer eingestellten Taskzeit von 10 ms das SPS-Programm nur 5 ms benötigt, der Exceed-Counter aber trotzdem hochzählt. Das liegt daran, dass der K-Bus mehr als 5 ms für die Bearbeitung benötigt und die Taskzeit von 10 ms inklusive SPS-Programm und K-Bus überschritten wird. Dieses Problem kann gelöst werden, wenn die Anzahl der Klemmen reduziert wird oder die Taskzeit erhöht wird.

Standardmäßig ist die Echtzeit auf 80 % eingestellt. Das ist bereits der maximale Wert und eine Erhöhung auf 90 % ist mit einer Erhöhung auf 100 % gleichzusetzen.

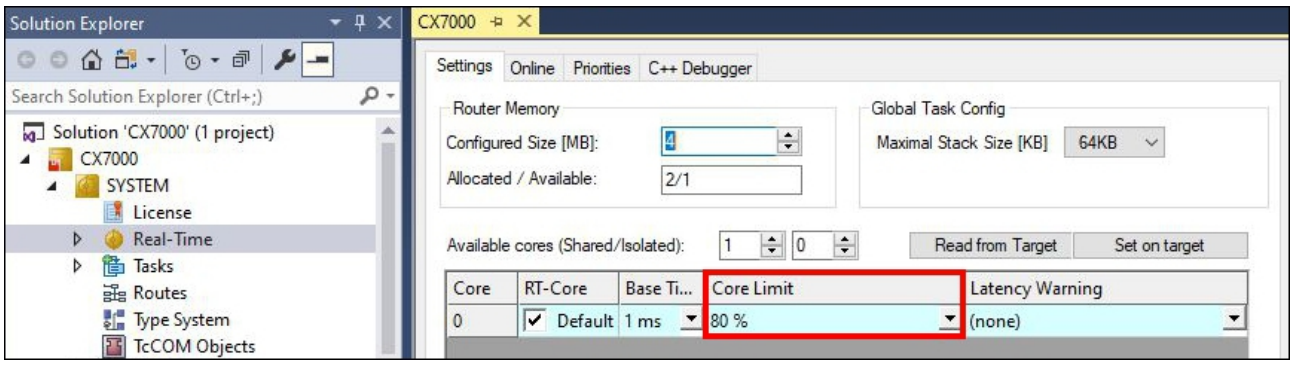


Abb. 75: Einstellung der Echtzeitauslastung in TwinCAT.

TwinCAT würde dann die gesamte CPU-Leistung beanspruchen und Dienste, die das Betriebssystem bedient, nicht mehr oder unzureichend funktionieren. Wenn Sie die Echtzeitauslastung auf 90 % erhöhen, sollten Sie sich der möglichen Folgen für das Betriebssystem bewusst sein.

10 Technische Daten

Tab. 27: Technische Daten, Abmessungen und Gewicht.

	CX7050
Abmessungen (B x H x T)	49 mm x 100 mm x 73 mm
Gewicht	142 g

Tab. 28: Technische Daten, allgemeine Daten.

Technische Daten	CX7050
Prozessor	ARM Cortex™-M7, 480 MHz
Anzahl Kerne	1
Flash-Speicher	512 MB MicroSD (optional 1 GB, 2 GB, 4 GB oder 8 GB)
Arbeitsspeicher	32 MB SDR (intern, nicht erweiterbar)
Anzahl Eingänge	8 Multifunktionseingänge (24 V DC)
Anzahl Ausgänge	4 Multifunktionsausgänge (24 V DC, 0,5 A, 1-Leitertechnik)
NOVRAM	4 kB
Schnittstellen	1 x RJ45 10/100 MBit/s, 1x USB (max 12 Mbit/s, max 100 mA)
Businterface	D-Sub-Stecker, 9-polig, 1 x CANopen-Commander (Master), CAN 2.0A/2.0B
Übertragungsrate	10, 20, 50, 100, 125, 250, 500, 800, 1000 kBaud
Diagnose-LED	1 x TC-Status, 1 x WD-LED, 1 x ERR-LED
Uhr	interne, kondensatorgepufferte Realtime-Clock für Zeit und Datum (Speicher > 21 Tage)
Betriebssystem	TC/RTOS
Steuerungssoftware	TwinCAT 3 Runtime (XAR)
Spannungsversorgung	24 V_{DC} (-15 %/+20 %)
max. Leistungsaufnahme	< 2 W (max. 12 W mit E-Bus/K-bus)
Enthaltene TwinCAT 3 Functions	TC1000 TwinCAT 3 ADS, TC1100 TwinCAT 3 I/O, TC1200 TwinCAT 3 PLC, TF4100 TwinCAT 3 Controller Toolbox, TF4110 TwinCAT 3 Temperature Controller, TF6255 TwinCAT 3 Modbus RTU, TF6340 TwinCAT 3 Serial Communication, TF6701 TwinCAT 3 IoT Communication (MQTT), TF6730 TwinCAT 3 IoT Communicator
Zulassungen	CE, UL

Tab. 29: Technische Daten, I/O-Klemmen.

Technische Daten	CX7050
I/O-Anschluss	via Netzteilklemme (E-Bus oder K-Bus, automatische Erkennung)
Stromversorgung für I/O-Klemmen	max. 1,5 A (Einbaulage beliebig, Temp. -25…45 °C) max. 1,3 A (Einbaulage horizontal, Temp. -25…55 °C) max. 1 A (Einbaulage beliebig, Temp. -25…55 °C) max. 1 A (Einbaulage horizontal, Temp. -25…60 °C)
Strombelastung Powerkontakte	max. 10 A
Prozessdaten K-Bus	max. 512 Byte In und 512 Byte Out
max. Anzahl der Klemmen (K-Bus)	64 (255 mit K-Bus-Verlängerung)
Prozessdaten E-Bus	max. 4 kByte In und 4 kByte Out
max. Anzahl der Klemmen (E-Bus)	bis zu 65534 Klemmen.

Tab. 30: Technische Daten, Umgebungsbedingungen.

Technische Daten	CX7050
Umgebungstemperatur im Betrieb	-25° C ... +60° C

Technische Daten	CX7050
Umgebungstemperatur bei Lagerung	-40° C ... +85° C siehe Hinweise unter: Transport und Lagerung [▶ 12]
Relative Feuchte	95 % ohne Betauung
Schwingungsfestigkeit	gemäß EN 60068-2-6
Schockfestigkeit	gemäß EN 60068-2-27
EMV-Festigkeit	gemäß EN 61000-6-2
EMV-Aussendung	gemäß EN 61000-6-4
Schutzart	IP 20

Tab. 31: Technische Daten, Ethernet-Schnittstelle X001.

Technische Daten	CX7050
Übertragungsmedium	4 x 2 Twisted-Pair-Kupferkabel Kategorie 5 (100 MBit/s)
Leitungslänge	100 m vom Switch bis zum CX7050
Übertragungsrate	10/100 MBit/s
Topologie	sternförmige Verkabelung
Protokolle	alle nicht Echtzeitfähigen Protokolle die auf TCP oder UDP basieren und keine Echtzeiterweiterung benötigen

Tab. 32: Technische Daten, CANopen-Schnittstelle X003.

Technische Daten	CX7050
Feldbus	CANopen
Übertragungsrate	10, 20, 50, 100, 125, 250, 500, 800, 1.000 kBaud
Businterface	1 x D-Sub-Buchse, 9-polig
Busteilnehmer	max. 64
max. Prozessabbild	512 Tx PDOs / 512 Rx PDOs
Autobaud	-
Galvanische Trennung	Ja
Protokoll	
CANopen Slave	Ja
CAN (virtueller Slave)	Ja
ADS-Interface	ja (nur über Ethernet)
Dienste	
CAN-Layer 2	Ja
CAN 2.0A	Ja
CAN 2.0B	Ja, nur über das CAN-Interface nutzbar

Tab. 33: Technische Daten, CANopen-Schnittstelle X003 parametriert als Slave.

Technische Daten	CX7050 parametriert als Slave
Feldbus	CANopen
Übertragungsrate	10, 20, 50, 100, 125, 250, 500, 800, 1.000 kBaud
Businterface	1 x D-Sub-Buchse, 9-polig
Erweiterbares Prozessabbild	bis zu 3 virtuelle Slaves zusätzlich
max. Prozessabbild	4 Slaves x (16 Tx PDOs / 16 Rx PDOs (8 Byte pro PDO))
Autobaud	Ja
Galvanische Trennung	Ja
Protokoll	
CANopen Slave	Ja
CAN (virtueller Slave)	4 (3 virtuelle CANopen Nodes)
ADS-Interface	ja (nur über Ethernet)

Technische Daten	CX7050 parametriert als Slave
Dienste	
CAN-Layer 2	Nein
CAN 2.0A	Nach CANopen
CAN 2.0B	Nein

11 Anhang

11.1 CAN-Identifier-Liste

Die hier aufgeführte Liste soll bei der Identifizierung und Zuordnung von CANopen Nachrichten helfen. Aufgeführt sind alle von der CANopen Default Identifier Verteilung zugeordneten Identifier, sowie die von BECKHOFF via Objekt 0x5500 vergebenen herstellerspezifischen Default Identifier (nur in Netzen mit Knotenadressen <64 zu verwenden).

In der *chm-Ausgabe der Dokumentation dienen die folgenden Werte als Suchhilfe und "Einsprungpunkte" in die umfangreiche Identfier-Tabelle:

Dezimal: 400 [▶ 205], 500 [▶ 209], 600 [▶ 209], 700 [▶ 206], 800 [▶ 206], 900 [▶ 207], 1000 [▶ 211], 1100 [▶ 211], 1200 [▶ 207], 1300 [▶ 208], 1400 [▶ 212], 1500 [▶ 212], 1600 [▶ 213], 1700 [▶ 208], 1800 [▶ 215], 1900 [▶ 213]

Hexadezimal: 0x181 [▶ 205], 0x1C1 [▶ 209], 0x201 [▶ 205], 0x301 [▶ 206], 0x401 [▶ 207], 0x501 [▶ 208], 0x601 [▶ 214], 0x701 [▶ 215]

Die Identifier-Verteilung via Objekt 0x5500 folgt diesem Schema:

Objekt	resultierende COB-ID (dez)	resultierende COB-ID (hex)
Emergency [▶ 205]	129 bis 191 [255]	0x81 bis 0xBF [0xFF]
TxPDO1 [▶ 205]	385 bis 447 [511]	0x181 bis 0x1BF [0x1FF]
RxPDO1 [▶ 205]	513 bis 575 [639]	0x201 bis 0x23F [0x27F]
TxPDO2 [▶ 206]	641 bis 676 [767]	0x281 bis 0x2BF [0x2FF]
RxPDO2 [▶ 206]	769 bis 831 [895]	0x301 bis 0x33F [0x37F]
TxDPO3 [▶ 207]	897 bis 959 [1023]	0x381 bis 0x3BF [0x3FF]
RxPDO3 [▶ 207]	1025 bis 1087 [1151]	0x401 bis 0x43F [0x47F]
TxPDO4 [▶ 207]	1153 bis 1215 [1279]	0x481 bis 0x4BF [0x4FF]
RxPDO4 [▶ 208]	1281 bis 1343 [1407]	0x501 bis 0x53F [0x57F]
TxPDO5 [▶ 208]	1665 bis 1727	0x681 bis 0x6BF
RxPDO5 [▶ 209]	1921 bis 1983	0x781 bis 0x7BF
TxPDO6 [▶ 209]	449 bis 511	0x1C1 bis 0x1FF
RxPDO6 [▶ 209]	577 bis 639	0x241 bis 0x27F
TxDPO7 [▶ 210]	705 bis 767	0x2C1 bis 0x2FF
RxPDO7 [▶ 210]	833 bis 895	0x341 bis 0x37F
TxPDO8 [▶ 211]	961 bis 1023	0x3C1 bis 0x3FF
RxPDO8 [▶ 211]	1089 bis 1151	0x441 bis 0x47F
TxPDO9 [▶ 211]	1217 bis 1279	0x4C1 bis 0x4FF
RxPDO9 [▶ 212]	1345 bis 1407	0x541 bis 0x57F
TxDPO10 [▶ 212]	1473 bis 1535	0x5C1 bis 0x5FF
RxPDO10 [▶ 213]	1601 bis 1663	0x641 bis 0x67F
TxPDO11 [▶ 213]	1729 bis 1791	0x6C1 bis 0x6FF
RxPDO11 [▶ 213]	1857 bis 1919	0x741 bis 0x77F
SDO (Tx) [▶ 214]	1409 bis 1471 [1535]	0x581 bis 0x5BF [0x5FF]
SDO (Rx) [▶ 214]	1537 bis 1599 [1663]	0x601 bis 0x63F [0x67F]
Guarding / Heartbeat / Bootup [▶ 215]	1793 bis 1855 [1919]	0x701 bis 0x73F [0x77F]

Identifierliste

Mit * gekennzeichnete Identifier werden auf den Buskopplern nach Beschreiben von Index 0x5500 herstellerspezifisch vergeben.

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
0	0x00	NMT	149	0x95	EMCY Nd.21	171	0xAB	EMCY Nd.43
128	0x80	SYNC	150	0x96	EMCY Nd.22	172	0xAC	EMCY Nd.44
129	0x81	EMCY Nd.1	151	0x97	EMCY Nd.23	173	0xAD	EMCY Nd.45
130	0x82	EMCY Nd.2	152	0x98	EMCY Nd.24	174	0xAE	EMCY Nd.46
131	0x83	EMCY Nd.3	153	0x99	EMCY Nd.25	175	0xAF	EMCY Nd.47
132	0x84	EMCY Nd.4	154	0x9A	EMCY Nd.26	176	0xB0	EMCY Nd.48
133	0x85	EMCY Nd.5	155	0x9B	EMCY Nd.27	177	0xB1	EMCY Nd.49
134	0x86	EMCY Nd.6	156	0x9C	EMCY Nd.28	178	0xB2	EMCY Nd.50
135	0x87	EMCY Nd.7	157	0x9D	EMCY Nd.29	179	0xB3	EMCY Nd.51
136	0x88	EMCY Nd.8	158	0x9E	EMCY Nd.30	180	0xB4	EMCY Nd.52
137	0x89	EMCY Nd.9	159	0x9F	EMCY Nd.31	181	0xB5	EMCY Nd.53
138	0x8A	EMCY Nd.10	160	0xA0	EMCY Nd.32	182	0xB6	EMCY Nd.54
139	0x8B	EMCY Nd.11	161	0xA1	EMCY Nd.33	183	0xB7	EMCY Nd.55
140	0x8C	EMCY Nd.12	162	0xA2	EMCY Nd.34	184	0xB8	EMCY Nd.56
141	0x8D	EMCY Nd.13	163	0xA3	EMCY Nd.35	185	0xB9	EMCY Nd.57
142	0x8E	EMCY Nd.14	164	0xA4	EMCY Nd.36	186	0xBA	EMCY Nd.58
143	0x8F	EMCY Nd.15	165	0xA5	EMCY Nd.37	187	0xBB	EMCY Nd.59
144	0x90	EMCY Nd.16	166	0xA6	EMCY Nd.38	188	0xBC	EMCY Nd.60
145	0x91	EMCY Nd.17	167	0xA7	EMCY Nd.39	189	0xBD	EMCY Nd.61
146	0x92	EMCY Nd.18	168	0xA8	EMCY Nd.40	190	0xBE	EMCY Nd.62
147	0x93	EMCY Nd.19	169	0xA9	EMCY Nd.41	191	0xBF	EMCY Nd.63
148	0x94	EMCY Nd.20	170	0xAA	EMCY Nd.42			

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
385	0x181	TxPDO1, DI, Nd.1	406	0x196	TxPDO1, DI, Nd.22	427	0x1AB	TxPDO1, DI, Nd.43
386	0x182	TxPDO1, DI, Nd.2	407	0x197	TxPDO1, DI, Nd.23	428	0x1AC	TxPDO1, DI, Nd.44
387	0x183	TxPDO1, DI, Nd.3	408	0x198	TxPDO1, DI, Nd.24	429	0x1AD	TxPDO1, DI, Nd.45
388	0x184	TxPDO1, DI, Nd.4	409	0x199	TxPDO1, DI, Nd.25	430	0x1AE	TxPDO1, DI, Nd.46
389	0x185	TxPDO1, DI, Nd.5	410	0x19A	TxPDO1, DI, Nd.26	431	0x1AF	TxPDO1, DI, Nd.47
390	0x186	TxPDO1, DI, Nd.6	411	0x19B	TxPDO1, DI, Nd.27	432	0x1B0	TxPDO1, DI, Nd.48
391	0x187	TxPDO1, DI, Nd.7	412	0x19C	TxPDO1, DI, Nd.28	433	0x1B1	TxPDO1, DI, Nd.49
392	0x188	TxPDO1, DI, Nd.8	413	0x19D	TxPDO1, DI, Nd.29	434	0x1B2	TxPDO1, DI, Nd.50
393	0x189	TxPDO1, DI, Nd.9	414	0x19E	TxPDO1, DI, Nd.30	435	0x1B3	TxPDO1, DI, Nd.51
394	0x18A	TxPDO1, DI, Nd.10	415	0x19F	TxPDO1, DI, Nd.31	436	0x1B4	TxPDO1, DI, Nd.52
395	0x18B	TxPDO1, DI, Nd.11	416	0x1A0	TxPDO1, DI, Nd.32	437	0x1B5	TxPDO1, DI, Nd.53
396	0x18C	TxPDO1, DI, Nd.12	417	0x1A1	TxPDO1, DI, Nd.33	438	0x1B6	TxPDO1, DI, Nd.54
397	0x18D	TxPDO1, DI, Nd.13	418	0x1A2	TxPDO1, DI, Nd.34	439	0x1B7	TxPDO1, DI, Nd.55
398	0x18E	TxPDO1, DI, Nd.14	419	0x1A3	TxPDO1, DI, Nd.35	440	0x1B8	TxPDO1, DI, Nd.56
399	0x18F	TxPDO1, DI, Nd.15	420	0x1A4	TxPDO1, DI, Nd.36	441	0x1B9	TxPDO1, DI, Nd.57
400	0x190	TxPDO1, DI, Nd.16	421	0x1A5	TxPDO1, DI, Nd.37	442	0x1BA	TxPDO1, DI, Nd.58
401	0x191	TxPDO1, DI, Nd.17	422	0x1A6	TxPDO1, DI, Nd.38	443	0x1BB	TxPDO1, DI, Nd.59
402	0x192	TxPDO1, DI, Nd.18	423	0x1A7	TxPDO1, DI, Nd.39	444	0x1BC	TxPDO1, DI, Nd.60
403	0x193	TxPDO1, DI, Nd.19	424	0x1A8	TxPDO1, DI, Nd.40	445	0x1BD	TxPDO1, DI, Nd.61
404	0x194	TxPDO1, DI, Nd.20	425	0x1A9	TxPDO1, DI, Nd.41	446	0x1BE	TxPDO1, DI, Nd.62
405	0x195	TxPDO1, DI, Nd.21	426	0x1AA	TxPDO1, DI, Nd.42	447	0x1BF	TxPDO1, DI, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
513	0x201	RxPDO1, DO, Nd.1	534	0x216	RxPDO1, DO, Nd.22	555	0x22B	RxPDO1, DO, Nd.43
514	0x202	RxPDO1, DO, Nd.2	535	0x217	RxPDO1, DO, Nd.23	556	0x22C	RxPDO1, DO, Nd.44
515	0x203	RxPDO1, DO, Nd.3	536	0x218	RxPDO1, DO, Nd.24	557	0x22D	RxPDO1, DO, Nd.45
516	0x204	RxPDO1, DO, Nd.4	537	0x219	RxPDO1, DO, Nd.25	558	0x22E	RxPDO1, DO, Nd.46
517	0x205	RxPDO1, DO, Nd.5	538	0x21A	RxPDO1, DO, Nd.26	559	0x22F	RxPDO1, DO, Nd.47
518	0x206	RxPDO1, DO, Nd.6	539	0x21B	RxPDO1, DO, Nd.27	560	0x230	RxPDO1, DO, Nd.48
519	0x207	RxPDO1, DO, Nd.7	540	0x21C	RxPDO1, DO, Nd.28	561	0x231	RxPDO1, DO, Nd.49

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
520	0x208	RxPDO1, DO, Nd.8	541	0x21D	RxPDO1, DO, Nd.29	562	0x232	RxPDO1, DO, Nd.50
521	0x209	RxPDO1, DO, Nd.9	542	0x21E	RxPDO1, DO, Nd.30	563	0x233	RxPDO1, DO, Nd.51
522	0x20A	RxPDO1, DO, Nd.10	543	0x21F	RxPDO1, DO, Nd.31	564	0x234	RxPDO1, DO, Nd.52
523	0x20B	RxPDO1, DO, Nd.11	544	0x220	RxPDO1, DO, Nd.32	565	0x235	RxPDO1, DO, Nd.53
524	0x20C	RxPDO1, DO, Nd.12	545	0x221	RxPDO1, DO, Nd.33	566	0x236	RxPDO1, DO, Nd.54
525	0x20D	RxPDO1, DO, Nd.13	546	0x222	RxPDO1, DO, Nd.34	567	0x237	RxPDO1, DO, Nd.55
526	0x20E	RxPDO1, DO, Nd.14	547	0x223	RxPDO1, DO, Nd.35	568	0x238	RxPDO1, DO, Nd.56
527	0x20F	RxPDO1, DO, Nd.15	548	0x224	RxPDO1, DO, Nd.36	569	0x239	RxPDO1, DO, Nd.57
528	0x210	RxPDO1, DO, Nd.16	549	0x225	RxPDO1, DO, Nd.37	570	0x23A	RxPDO1, DO, Nd.58
529	0x211	RxPDO1, DO, Nd.17	550	0x226	RxPDO1, DO, Nd.38	571	0x23B	RxPDO1, DO, Nd.59
530	0x212	RxPDO1, DO, Nd.18	551	0x227	RxPDO1, DO, Nd.39	572	0x23C	RxPDO1, DO, Nd.60
531	0x213	RxPDO1, DO, Nd.19	552	0x228	RxPDO1, DO, Nd.40	573	0x23D	RxPDO1, DO, Nd.61
532	0x214	RxPDO1, DO, Nd.20	553	0x229	RxPDO1, DO, Nd.41	574	0x23E	RxPDO1, DO, Nd.62
533	0x215	RxPDO1, DO, Nd.21	554	0x22A	RxPDO1, DO, Nd.42	575	0x23F	RxPDO1, DO, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
641	0x281	TxPDO2, AI, Nd.1	662	0x296	TxPDO2, AI, Nd.22	683	0x2AB	TxPDO2, AI, Nd.43
642	0x282	TxPDO2, AI, Nd.2	663	0x297	TxPDO2, AI, Nd.23	684	0x2AC	TxPDO2, AI, Nd.44
643	0x283	TxPDO2, AI, Nd.3	664	0x298	TxPDO2, AI, Nd.24	685	0x2AD	TxPDO2, AI, Nd.45
644	0x284	TxPDO2, AI, Nd.4	665	0x299	TxPDO2, AI, Nd.25	686	0x2AE	TxPDO2, AI, Nd.46
645	0x285	TxPDO2, AI, Nd.5	666	0x29A	TxPDO2, AI, Nd.26	687	0x2AF	TxPDO2, AI, Nd.47
646	0x286	TxPDO2, AI, Nd.6	667	0x29B	TxPDO2, AI, Nd.27	688	0x2B0	TxPDO2, AI, Nd.48
647	0x287	TxPDO2, AI, Nd.7	668	0x29C	TxPDO2, AI, Nd.28	689	0x2B1	TxPDO2, AI, Nd.49
648	0x288	TxPDO2, AI, Nd.8	669	0x29D	TxPDO2, AI, Nd.29	690	0x2B2	TxPDO2, AI, Nd.50
649	0x289	TxPDO2, AI, Nd.9	670	0x29E	TxPDO2, AI, Nd.30	691	0x2B3	TxPDO2, AI, Nd.51
650	0x28A	TxPDO2, AI, Nd.10	671	0x29F	TxPDO2, AI, Nd.31	692	0x2B4	TxPDO2, AI, Nd.52
651	0x28B	TxPDO2, AI, Nd.11	672	0x2A0	TxPDO2, AI, Nd.32	693	0x2B5	TxPDO2, AI, Nd.53
652	0x28C	TxPDO2, AI, Nd.12	673	0x2A1	TxPDO2, AI, Nd.33	694	0x2B6	TxPDO2, AI, Nd.54
653	0x28D	TxPDO2, AI, Nd.13	674	0x2A2	TxPDO2, AI, Nd.34	695	0x2B7	TxPDO2, AI, Nd.55
654	0x28E	TxPDO2, AI, Nd.14	675	0x2A3	TxPDO2, AI, Nd.35	696	0x2B8	TxPDO2, AI, Nd.56
655	0x28F	TxPDO2, AI, Nd.15	676	0x2A4	TxPDO2, AI, Nd.36	697	0x2B9	TxPDO2, AI, Nd.57
656	0x290	TxPDO2, AI, Nd.16	677	0x2A5	TxPDO2, AI, Nd.37	698	0x2BA	TxPDO2, AI, Nd.58
657	0x291	TxPDO2, AI, Nd.17	678	0x2A6	TxPDO2, AI, Nd.38	699	0x2BB	TxPDO2, AI, Nd.59
658	0x292	TxPDO2, AI, Nd.18	679	0x2A7	TxPDO2, AI, Nd.39	700	0x2BC	TxPDO2, AI, Nd.60
659	0x293	TxPDO2, AI, Nd.19	680	0x2A8	TxPDO2, AI, Nd.40	701	0x2BD	TxPDO2, AI, Nd.61
660	0x294	TxPDO2, AI, Nd.20	681	0x2A9	TxPDO2, AI, Nd.41	702	0x2BE	TxPDO2, AI, Nd.62
661	0x295	TxPDO2, AI, Nd.21	682	0x2AA	TxPDO2, AI, Nd.42	703	0x2BF	TxPDO2, AI, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
769	0x301	RxPDO2, AO, Nd.1	790	0x316	RxPDO2, AO, Nd.22	811	0x32B	RxPDO2, AO, Nd.43
770	0x302	RxPDO2, AO, Nd.2	791	0x317	RxPDO2, AO, Nd.23	812	0x32C	RxPDO2, AO, Nd.44
771	0x303	RxPDO2, AO, Nd.3	792	0x318	RxPDO2, AO, Nd.24	813	0x32D	RxPDO2, AO, Nd.45
772	0x304	RxPDO2, AO, Nd.4	793	0x319	RxPDO2, AO, Nd.25	814	0x32E	RxPDO2, AO, Nd.46
773	0x305	RxPDO2, AO, Nd.5	794	0x31A	RxPDO2, AO, Nd.26	815	0x32F	RxPDO2, AO, Nd.47
774	0x306	RxPDO2, AO, Nd.6	795	0x31B	RxPDO2, AO, Nd.27	816	0x330	RxPDO2, AO, Nd.48
775	0x307	RxPDO2, AO, Nd.7	796	0x31C	RxPDO2, AO, Nd.28	817	0x331	RxPDO2, AO, Nd.49
776	0x308	RxPDO2, AO, Nd.8	797	0x31D	RxPDO2, AO, Nd.29	818	0x332	RxPDO2, AO, Nd.50
777	0x309	RxPDO2, AO, Nd.9	798	0x31E	RxPDO2, AO, Nd.30	819	0x333	RxPDO2, AO, Nd.51
778	0x30A	RxPDO2, AO, Nd.10	799	0x31F	RxPDO2, AO, Nd.31	820	0x334	RxPDO2, AO, Nd.52
779	0x30B	RxPDO2, AO, Nd.11	800	0x320	RxPDO2, AO, Nd.32	821	0x335	RxPDO2, AO, Nd.53
780	0x30C	RxPDO2, AO, Nd.12	801	0x321	RxPDO2, AO, Nd.33	822	0x336	RxPDO2, AO, Nd.54
781	0x30D	RxPDO2, AO, Nd.13	802	0x322	RxPDO2, AO, Nd.34	823	0x337	RxPDO2, AO, Nd.55
782	0x30E	RxPDO2, AO, Nd.14	803	0x323	RxPDO2, AO, Nd.35	824	0x338	RxPDO2, AO, Nd.56
783	0x30F	RxPDO2, AO, Nd.15	804	0x324	RxPDO2, AO, Nd.36	825	0x339	RxPDO2, AO, Nd.57
784	0x310	RxPDO2, AO, Nd.16	805	0x325	RxPDO2, AO, Nd.37	826	0x33A	RxPDO2, AO, Nd.58
785	0x311	RxPDO2, AO, Nd.17	806	0x326	RxPDO2, AO, Nd.38	827	0x33B	RxPDO2, AO, Nd.59
786	0x312	RxPDO2, AO, Nd.18	807	0x327	RxPDO2, AO, Nd.39	828	0x33C	RxPDO2, AO, Nd.60

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
787	0x313	RxPDO2, AO, Nd.19	808	0x328	RxPDO2, AO, Nd.40	829	0x33D	RxPDO2, AO, Nd.61
788	0x314	RxPDO2, AO, Nd.20	809	0x329	RxPDO2, AO, Nd.41	830	0x33E	RxPDO2, AO, Nd.62
789	0x315	RxPDO2, AO, Nd.21	810	0x32A	RxPDO2, AO, Nd.42	831	0x33F	RxPDO2, AO, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
897	0x381	TxPDO3*, Nd.1	918	0x396	TxPDO3*, Nd.22	939	0x3AB	TxPDO3*, Nd.43
898	0x382	TxPDO3*, Nd.2	919	0x397	TxPDO3*, Nd.23	940	0x3AC	TxPDO3*, Nd.44
899	0x383	TxPDO3*, Nd.3	920	0x398	TxPDO3*, Nd.24	941	0x3AD	TxPDO3*, Nd.45
900	0x384	TxPDO3*, Nd.4	921	0x399	TxPDO3*, Nd.25	942	0x3AE	TxPDO3*, Nd.46
901	0x385	TxPDO3*, Nd.5	922	0x39A	TxPDO3*, Nd.26	943	0x3AF	TxPDO3*, Nd.47
902	0x386	TxPDO3*, Nd.6	923	0x39B	TxPDO3*, Nd.27	944	0x3B0	TxPDO3*, Nd.48
903	0x387	TxPDO3*, Nd.7	924	0x39C	TxPDO3*, Nd.28	945	0x3B1	TxPDO3*, Nd.49
904	0x388	TxPDO3*, Nd.8	925	0x39D	TxPDO3*, Nd.29	946	0x3B2	TxPDO3*, Nd.50
905	0x389	TxPDO3*, Nd.9	926	0x39E	TxPDO3*, Nd.30	947	0x3B3	TxPDO3*, Nd.51
906	0x38A	TxPDO3*, Nd.10	927	0x39F	TxPDO3*, Nd.31	948	0x3B4	TxPDO3*, Nd.52
907	0x38B	TxPDO3*, Nd.11	928	0x3A0	TxPDO3*, Nd.32	949	0x3B5	TxPDO3*, Nd.53
908	0x38C	TxPDO3*, Nd.12	929	0x3A1	TxPDO3*, Nd.33	950	0x3B6	TxPDO3*, Nd.54
909	0x38D	TxPDO3*, Nd.13	930	0x3A2	TxPDO3*, Nd.34	951	0x3B7	TxPDO3*, Nd.55
910	0x38E	TxPDO3*, Nd.14	931	0x3A3	TxPDO3*, Nd.35	952	0x3B8	TxPDO3*, Nd.56
911	0x38F	TxPDO3*, Nd.15	932	0x3A4	TxPDO3*, Nd.36	953	0x3B9	TxPDO3*, Nd.57
912	0x390	TxPDO3*, Nd.16	933	0x3A5	TxPDO3*, Nd.37	954	0x3BA	TxPDO3*, Nd.58
913	0x391	TxPDO3*, Nd.17	934	0x3A6	TxPDO3*, Nd.38	955	0x3BB	TxPDO3*, Nd.59
914	0x392	TxPDO3*, Nd.18	935	0x3A7	TxPDO3*, Nd.39	956	0x3BC	TxPDO3*, Nd.60
915	0x393	TxPDO3*, Nd.19	936	0x3A8	TxPDO3*, Nd.40	957	0x3BD	TxPDO3*, Nd.61
916	0x394	TxPDO3*, Nd.20	937	0x3A9	TxPDO3*, Nd.41	958	0x3BE	TxPDO3*, Nd.62
917	0x395	TxPDO3*, Nd.21	938	0x3AA	TxPDO3*, Nd.42	959	0x3BF	TxPDO3*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1025	0x401	RxPDO3*, Nd.1	1046	0x416	RxPDO3*, Nd.22	1067	0x42B	RxPDO3*, Nd.43
1026	0x402	RxPDO3*, Nd.2	1047	0x417	RxPDO3*, Nd.23	1068	0x42C	RxPDO3*, Nd.44
1027	0x403	RxPDO3*, Nd.3	1048	0x418	RxPDO3*, Nd.24	1069	0x42D	RxPDO3*, Nd.45
1028	0x404	RxPDO3*, Nd.4	1049	0x419	RxPDO3*, Nd.25	1070	0x42E	RxPDO3*, Nd.46
1029	0x405	RxPDO3*, Nd.5	1050	0x41A	RxPDO3*, Nd.26	1071	0x42F	RxPDO3*, Nd.47
1030	0x406	RxPDO3*, Nd.6	1051	0x41B	RxPDO3*, Nd.27	1072	0x430	RxPDO3*, Nd.48
1031	0x407	RxPDO3*, Nd.7	1052	0x41C	RxPDO3*, Nd.28	1073	0x431	RxPDO3*, Nd.49
1032	0x408	RxPDO3*, Nd.8	1053	0x41D	RxPDO3*, Nd.29	1074	0x432	RxPDO3*, Nd.50
1033	0x409	RxPDO3*, Nd.9	1054	0x41E	RxPDO3*, Nd.30	1075	0x433	RxPDO3*, Nd.51
1034	0x40A	RxPDO3*, Nd.10	1055	0x41F	RxPDO3*, Nd.31	1076	0x434	RxPDO3*, Nd.52
1035	0x40B	RxPDO3*, Nd.11	1056	0x420	RxPDO3*, Nd.32	1077	0x435	RxPDO3*, Nd.53
1036	0x40C	RxPDO3*, Nd.12	1057	0x421	RxPDO3*, Nd.33	1078	0x436	RxPDO3*, Nd.54
1037	0x40D	RxPDO3*, Nd.13	1058	0x422	RxPDO3*, Nd.34	1079	0x437	RxPDO3*, Nd.55
1038	0x40E	RxPDO3*, Nd.14	1059	0x423	RxPDO3*, Nd.35	1080	0x438	RxPDO3*, Nd.56
1039	0x40F	RxPDO3*, Nd.15	1060	0x424	RxPDO3*, Nd.36	1081	0x439	RxPDO3*, Nd.57
1040	0x410	RxPDO3*, Nd.16	1061	0x425	RxPDO3*, Nd.37	1082	0x43A	RxPDO3*, Nd.58
1041	0x411	RxPDO3*, Nd.17	1062	0x426	RxPDO3*, Nd.38	1083	0x43B	RxPDO3*, Nd.59
1042	0x412	RxPDO3*, Nd.18	1063	0x427	RxPDO3*, Nd.39	1084	0x43C	RxPDO3*, Nd.60
1043	0x413	RxPDO3*, Nd.19	1064	0x428	RxPDO3*, Nd.40	1085	0x43D	RxPDO3*, Nd.61
1044	0x414	RxPDO3*, Nd.20	1065	0x429	RxPDO3*, Nd.41	1086	0x43E	RxPDO3*, Nd.62
1045	0x415	RxPDO3*, Nd.21	1066	0x42A	RxPDO3*, Nd.42	1087	0x43F	RxPDO3*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1153	0x481	TxPDO4*, Nd.1	1174	0x496	TxPDO4*, Nd.22	1195	0x4AB	TxPDO4*, Nd.43
1154	0x482	TxPDO4*, Nd.2	1175	0x497	TxPDO4*, Nd.23	1196	0x4AC	TxPDO4*, Nd.44
1155	0x483	TxPDO4*, Nd.3	1176	0x498	TxPDO4*, Nd.24	1197	0x4AD	TxPDO4*, Nd.45
1156	0x484	TxPDO4*, Nd.4	1177	0x499	TxPDO4*, Nd.25	1198	0x4AE	TxPDO4*, Nd.46
1157	0x485	TxPDO4*, Nd.5	1178	0x49A	TxPDO4*, Nd.26	1199	0x4AF	TxPDO4*, Nd.47
1158	0x486	TxPDO4*, Nd.6	1179	0x49B	TxPDO4*, Nd.27	1200	0x4B0	TxPDO4*, Nd.48

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1159	0x487	TxPDO4*, Nd.7	1180	0x49C	TxPDO4*, Nd.28	1201	0x4B1	TxPDO4*, Nd.49
1160	0x488	TxPDO4*, Nd.8	1181	0x49D	TxPDO4*, Nd.29	1202	0x4B2	TxPDO4*, Nd.50
1161	0x489	TxPDO4*, Nd.9	1182	0x49E	TxPDO4*, Nd.30	1203	0x4B3	TxPDO4*, Nd.51
1162	0x48A	TxPDO4*, Nd.10	1183	0x49F	TxPDO4*, Nd.31	1204	0x4B4	TxPDO4*, Nd.52
1163	0x48B	TxPDO4*, Nd.11	1184	0x4A0	TxPDO4*, Nd.32	1205	0x4B5	TxPDO4*, Nd.53
1164	0x48C	TxPDO4*, Nd.12	1185	0x4A1	TxPDO4*, Nd.33	1206	0x4B6	TxPDO4*, Nd.54
1165	0x48D	TxPDO4*, Nd.13	1186	0x4A2	TxPDO4*, Nd.34	1207	0x4B7	TxPDO4*, Nd.55
1166	0x48E	TxPDO4*, Nd.14	1187	0x4A3	TxPDO4*, Nd.35	1208	0x4B8	TxPDO4*, Nd.56
1167	0x48F	TxPDO4*, Nd.15	1188	0x4A4	TxPDO4*, Nd.36	1209	0x4B9	TxPDO4*, Nd.57
1168	0x490	TxPDO4*, Nd.16	1189	0x4A5	TxPDO4*, Nd.37	1210	0x4BA	TxPDO4*, Nd.58
1169	0x491	TxPDO4*, Nd.17	1190	0x4A6	TxPDO4*, Nd.48	1211	0x4BB	TxPDO4*, Nd.59
1170	0x492	TxPDO4*, Nd.18	1191	0x4A7	TxPDO4*, Nd.49	1212	0x4BC	TxPDO4*, Nd.60
1171	0x493	TxPDO4*, Nd.19	1192	0x4A8	TxPDO4*, Nd.40	1213	0x4BD	TxPDO4*, Nd.61
1172	0x494	TxPDO4*, Nd.20	1193	0x4A9	TxPDO4*, Nd.41	1214	0x4BE	TxPDO4*, Nd.62
1173	0x495	TxPDO4*, Nd.21	1194	0x4AA	TxPDO4*, Nd.42	1215	0x4BF	TxPDO4*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1281	0x501	RxPDO4*, Nd.1	1302	0x516	RxPDO4*, Nd.22	1323	0x52B	RxPDO4*, Nd.43
1282	0x502	RxPDO4*, Nd.2	1303	0x517	RxPDO4*, Nd.23	1324	0x52C	RxPDO4*, Nd.44
1283	0x503	RxPDO4*, Nd.3	1304	0x518	RxPDO4*, Nd.24	1325	0x52D	RxPDO4*, Nd.45
1284	0x504	RxPDO4*, Nd.4	1305	0x519	RxPDO4*, Nd.25	1326	0x52E	RxPDO4*, Nd.46
1285	0x505	RxPDO4*, Nd.5	1306	0x51A	RxPDO4*, Nd.26	1327	0x52F	RxPDO4*, Nd.47
1286	0x506	RxPDO4*, Nd.6	1307	0x51B	RxPDO4*, Nd.27	1328	0x530	RxPDO4*, Nd.48
1287	0x507	RxPDO4*, Nd.7	1308	0x51C	RxPDO4*, Nd.28	1329	0x531	RxPDO4*, Nd.49
1288	0x508	RxPDO4*, Nd.8	1309	0x51D	RxPDO4*, Nd.29	1330	0x532	RxPDO4*, Nd.50
1289	0x509	RxPDO4*, Nd.9	1310	0x51E	RxPDO4*, Nd.30	1331	0x533	RxPDO4*, Nd.51
1290	0x50A	RxPDO4*, Nd.10	1311	0x51F	RxPDO4*, Nd.31	1332	0x534	RxPDO4*, Nd.52
1291	0x50B	RxPDO4*, Nd.11	1312	0x520	RxPDO4*, Nd.32	1333	0x535	RxPDO4*, Nd.53
1292	0x50C	RxPDO4*, Nd.12	1313	0x521	RxPDO4*, Nd.33	1334	0x536	RxPDO4*, Nd.54
1293	0x50D	RxPDO4*, Nd.13	1314	0x522	RxPDO4*, Nd.34	1335	0x537	RxPDO4*, Nd.55
1294	0x50E	RxPDO4*, Nd.14	1315	0x523	RxPDO4*, Nd.35	1336	0x538	RxPDO4*, Nd.56
1295	0x50F	RxPDO4*, Nd.15	1316	0x524	RxPDO4*, Nd.36	1337	0x539	RxPDO4*, Nd.57
1296	0x510	RxPDO4*, Nd.16	1317	0x525	RxPDO4*, Nd.37	1338	0x53A	RxPDO4*, Nd.58
1297	0x511	RxPDO4*, Nd.17	1318	0x526	RxPDO4*, Nd.38	1339	0x53B	RxPDO4*, Nd.59
1298	0x512	RxPDO4*, Nd.18	1319	0x527	RxPDO4*, Nd.39	1340	0x53C	RxPDO4*, Nd.60
1299	0x513	RxPDO4*, Nd.19	1320	0x528	RxPDO4*, Nd.40	1341	0x53D	RxPDO4*, Nd.61
1300	0x514	RxPDO4*, Nd.20	1321	0x529	RxPDO4*, Nd.41	1342	0x53E	RxPDO4*, Nd.62
1301	0x515	RxPDO4*, Nd.21	1322	0x52A	RxPDO4*, Nd.42	1343	0x53F	RxPDO4*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1665	0x681	TxPDO5*, Nd.1	1686	0x696	TxPDO5*, Nd.22	1707	0x6AB	TxPDO5*, Nd.43
1666	0x682	TxPDO5*, Nd.2	1687	0x697	TxPDO5*, Nd.23	1708	0x6AC	TxPDO5*, Nd.44
1667	0x683	TxPDO5*, Nd.3	1688	0x698	TxPDO5*, Nd.24	1709	0x6AD	TxPDO5*, Nd.45
1668	0x684	TxPDO5*, Nd.4	1689	0x699	TxPDO5*, Nd.25	1710	0x6AE	TxPDO5*, Nd.46
1669	0x685	TxPDO5*, Nd.5	1690	0x69A	TxPDO5*, Nd.26	1711	0x6AF	TxPDO5*, Nd.47
1670	0x686	TxPDO5*, Nd.6	1691	0x69B	TxPDO5*, Nd.27	1712	0x6B0	TxPDO5*, Nd.48
1671	0x687	TxPDO5*, Nd.7	1692	0x69C	TxPDO5*, Nd.28	1713	0x6B1	TxPDO5*, Nd.49
1672	0x688	TxPDO5*, Nd.8	1693	0x69D	TxPDO5*, Nd.29	1714	0x6B2	TxPDO5*, Nd.50
1673	0x689	TxPDO5*, Nd.9	1694	0x69E	TxPDO5*, Nd.30	1715	0x6B3	TxPDO5*, Nd.51
1674	0x68A	TxPDO5*, Nd.10	1695	0x69F	TxPDO5*, Nd.31	1716	0x6B4	TxPDO5*, Nd.52
1675	0x68B	TxPDO5*, Nd.11	1696	0x6A0	TxPDO5*, Nd.32	1717	0x6B5	TxPDO5*, Nd.53
1676	0x68C	TxPDO5*, Nd.12	1697	0x6A1	TxPDO5*, Nd.33	1718	0x6B6	TxPDO5*, Nd.54
1677	0x68D	TxPDO5*, Nd.13	1698	0x6A2	TxPDO5*, Nd.34	1719	0x6B7	TxPDO5*, Nd.55
1678	0x68E	TxPDO5*, Nd.14	1699	0x6A3	TxPDO5*, Nd.35	1720	0x6B8	TxPDO5*, Nd.56
1679	0x68F	TxPDO5*, Nd.15	1700	0x6A4	TxPDO5*, Nd.36	1721	0x6B9	TxPDO5*, Nd.57
1680	0x690	TxPDO5*, Nd.16	1701	0x6A5	TxPDO5*, Nd.37	1722	0x6BA	TxPDO5*, Nd.58
1681	0x691	TxPDO5*, Nd.17	1702	0x6A6	TxPDO5*, Nd.38	1723	0x6BB	TxPDO5*, Nd.59

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1682	0x692	TxPDO5*, Nd.18	1703	0x6A7	TxPDO5*, Nd.39	1724	0x6BC	TxPDO5*, Nd.60
1683	0x693	TxPDO5*, Nd.19	1704	0x6A8	TxPDO5*, Nd.40	1725	0x6BD	TxPDO5*, Nd.61
1684	0x694	TxPDO5*, Nd.20	1705	0x6A9	TxPDO5*, Nd.41	1726	0x6BE	TxPDO5*, Nd.62
1685	0x695	TxPDO5*, Nd.21	1706	0x6AA	TxPDO5*, Nd.42	1727	0x6BF	TxPDO5*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1921	0x781	RxPDO5*, Nd.1	1942	0x796	RxPDO5*, Nd.22	1963	0x7AB	RxPDO5*, Nd.43
1922	0x782	RxPDO5*, Nd.2	1943	0x797	RxPDO5*, Nd.23	1964	0x7AC	RxPDO5*, Nd.44
1923	0x783	RxPDO5*, Nd.3	1944	0x798	RxPDO5*, Nd.24	1965	0x7AD	RxPDO5*, Nd.45
1924	0x784	RxPDO5*, Nd.4	1945	0x799	RxPDO5*, Nd.25	1966	0x7AE	RxPDO5*, Nd.46
1925	0x785	RxPDO5*, Nd.5	1946	0x79A	RxPDO5*, Nd.26	1967	0x7AF	RxPDO5*, Nd.47
1926	0x786	RxPDO5*, Nd.6	1947	0x79B	RxPDO5*, Nd.27	1968	0x7B0	RxPDO5*, Nd.48
1927	0x787	RxPDO5*, Nd.7	1948	0x79C	RxPDO5*, Nd.28	1969	0x7B1	RxPDO5*, Nd.49
1928	0x788	RxPDO5*, Nd.8	1949	0x79D	RxPDO5*, Nd.29	1970	0x7B2	RxPDO5*, Nd.50
1929	0x789	RxPDO5*, Nd.9	1950	0x79E	RxPDO5*, Nd.30	1971	0x7B3	RxPDO5*, Nd.51
1930	0x78A	RxPDO5*, Nd.10	1951	0x79F	RxPDO5*, Nd.31	1972	0x7B4	RxPDO5*, Nd.52
1931	0x78B	RxPDO5*, Nd.11	1952	0x7A0	RxPDO5*, Nd.32	1973	0x7B5	RxPDO5*, Nd.53
1932	0x78C	RxPDO5*, Nd.12	1953	0x7A1	RxPDO5*, Nd.33	1974	0x7B6	RxPDO5*, Nd.54
1933	0x78D	RxPDO5*, Nd.13	1954	0x7A2	RxPDO5*, Nd.34	1975	0x7B7	RxPDO5*, Nd.55
1934	0x78E	RxPDO5*, Nd.14	1955	0x7A3	RxPDO5*, Nd.35	1976	0x7B8	RxPDO5*, Nd.56
1935	0x78F	RxPDO5*, Nd.15	1956	0x7A4	RxPDO5*, Nd.36	1977	0x7B9	RxPDO5*, Nd.57
1936	0x790	RxPDO5*, Nd.16	1957	0x7A5	RxPDO5*, Nd.37	1978	0x7BA	RxPDO5*, Nd.58
1937	0x791	RxPDO5*, Nd.17	1958	0x7A6	RxPDO5*, Nd.38	1979	0x7BB	RxPDO5*, Nd.59
1938	0x792	RxPDO5*, Nd.18	1959	0x7A7	RxPDO5*, Nd.39	1980	0x7BC	RxPDO5*, Nd.60
1939	0x793	RxPDO5*, Nd.19	1960	0x7A8	RxPDO5*, Nd.40	1981	0x7BD	RxPDO5*, Nd.61
1940	0x794	RxPDO5*, Nd.20	1961	0x7A9	RxPDO5*, Nd.41	1982	0x7BE	RxPDO5*, Nd.62
1941	0x795	RxPDO5*, Nd.21	1962	0x7AA	RxPDO5*, Nd.42	1983	0x7BF	RxPDO5*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
449	0x1C1	TxPDO6*, Nd.1	470	0x1D6	TxPDO6*, Nd.22	491	0x1EB	TxPDO6*, Nd.43
450	0x1C2	TxPDO6*, Nd.2	471	0x1D7	TxPDO6*, Nd.23	492	0x1EC	TxPDO6*, Nd.44
451	0x1C3	TxPDO6*, Nd.3	472	0x1D8	TxPDO6*, Nd.24	493	0x1ED	TxPDO6*, Nd.45
452	0x1C4	TxPDO6*, Nd.4	473	0x1D9	TxPDO6*, Nd.25	494	0x1EE	TxPDO6*, Nd.46
453	0x1C5	TxPDO6*, Nd.5	474	0x1DA	TxPDO6*, Nd.26	495	0x1EF	TxPDO6*, Nd.47
454	0x1C6	TxPDO6*, Nd.6	475	0x1DB	TxPDO6*, Nd.27	496	0x1F0	TxPDO6*, Nd.48
455	0x1C7	TxPDO6*, Nd.7	476	0x1DC	TxPDO6*, Nd.28	497	0x1F1	TxPDO6*, Nd.49
456	0x1C8	TxPDO6*, Nd.8	477	0x1DD	TxPDO6*, Nd.29	498	0x1F2	TxPDO6*, Nd.50
457	0x1C9	TxPDO6*, Nd.9	478	0x1DE	TxPDO6*, Nd.30	499	0x1F3	TxPDO6*, Nd.51
458	0x1CA	TxPDO6*, Nd.10	479	0x1DF	TxPDO6*, Nd.31	500	0x1F4	TxPDO6*, Nd.52
459	0x1CB	TxPDO6*, Nd.11	480	0x1E0	TxPDO6*, Nd.32	501	0x1F5	TxPDO6*, Nd.53
460	0x1CC	TxPDO6*, Nd.12	481	0x1E1	TxPDO6*, Nd.33	502	0x1F6	TxPDO6*, Nd.54
461	0x1CD	TxPDO6*, Nd.13	482	0x1E2	TxPDO6*, Nd.34	503	0x1F7	TxPDO6*, Nd.55
462	0x1CE	TxPDO6*, Nd.14	483	0x1E3	TxPDO6*, Nd.35	504	0x1F8	TxPDO6*, Nd.56
463	0x1CF	TxPDO6*, Nd.15	484	0x1E4	TxPDO6*, Nd.36	505	0x1F9	TxPDO6*, Nd.57
464	0x1D0	TxPDO6*, Nd.16	485	0x1E5	TxPDO6*, Nd.37	506	0x1FA	TxPDO6*, Nd.58
465	0x1D1	TxPDO6*, Nd.17	486	0x1E6	TxPDO6*, Nd.38	507	0x1FB	TxPDO6*, Nd.59
466	0x1D2	TxPDO6*, Nd.18	487	0x1E7	TxPDO6*, Nd.39	508	0x1FC	TxPDO6*, Nd.60
467	0x1D3	TxPDO6*, Nd.19	488	0x1E8	TxPDO6*, Nd.40	509	0x1FD	TxPDO6*, Nd.61
468	0x1D4	TxPDO6*, Nd.20	489	0x1E9	TxPDO6*, Nd.41	510	0x1FE	TxPDO6*, Nd.62
469	0x1D5	TxPDO6*, Nd.21	490	0x1EA	TxPDO6*, Nd.42	511	0x1FF	TxPDO6*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
577	0x241	RxPDO6*, Nd.1	598	0x256	RxPDO6*, Nd.22	619	0x26B	RxPDO6* Nd.43
578	0x242	RxPDO6*, Nd.2	599	0x257	RxPDO6*, Nd.23	620	0x26C	RxPDO6, Nd.44
579	0x243	RxPDO6*, Nd.3	600	0x258	RxPDO6*, Nd.24	621	0x26D	RxPDO6*, Nd.45
580	0x244	RxPDO6*, Nd.4	601	0x259	RxPDO6*, Nd.25	622	0x26E	RxPDO6*, Nd.46
581	0x245	RxPDO6*, Nd.5	602	0x25A	RxPDO6*, Nd.26	623	0x26F	RxPDO6*, Nd.47

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
582	0x246	RxPDO6*, Nd.6	603	0x25B	RxPDO6*, Nd.27	624	0x270	RxPDO6*, Nd.48
583	0x247	RxPDO6*, Nd.7	604	0x25C	RxPDO6*, Nd.28	625	0x271	RxPDO6*, Nd.49
584	0x248	RxPDO6*, Nd.8	605	0x25D	RxPDO6*, Nd.29	626	0x272	RxPDO6*, Nd.50
585	0x249	RxPDO6*, Nd.9	606	0x25E	RxPDO6*, Nd.30	627	0x273	RxPDO6*, Nd.51
586	0x24A	RxPDO6*, Nd.10	607	0x25F	RxPDO6*, Nd.31	628	0x274	RxPDO6*, Nd.52
587	0x24B	RxPDO6*, Nd.11	608	0x260	RxPDO6*, Nd.32	629	0x275	RxPDO6*, Nd.53
588	0x24C	RxPDO6*, Nd.12	609	0x261	RxPDO6*, Nd.33	630	0x276	RxPDO6*, Nd.54
589	0x24D	RxPDO6*, Nd.13	610	0x262	RxPDO6*, Nd.34	631	0x277	RxPDO6*, Nd.55
590	0x24E	RxPDO6*, Nd.14	611	0x263	RxPDO6*, Nd.35	632	0x278	RxPDO6*, Nd.56
591	0x24F	RxPDO6*, Nd.15	612	0x264	RxPDO6*, Nd.36	633	0x279	RxPDO6*, Nd.57
592	0x250	RxPDO6*, Nd.16	613	0x265	RxPDO6*, Nd.3	634	0x27A	RxPDO6*, Nd.58
593	0x251	RxPDO6*, Nd.17	614	0x266	RxPDO6*, Nd.8	635	0x27B	RxPDO6*, Nd.59
594	0x252	RxPDO6*, Nd.18	615	0x267	RxPDO6*, Nd39	636	0x27C	RxPDO6*, Nd.60
595	0x253	RxPDO6*, Nd.19	616	0x268	RxPDO6*, N.40	637	0x27D	RxPDO6*, Nd.61
596	0x254	RxPDO6*, Nd.20	617	0x269	RxPDO6*, d.41	638	0x27E	RxPDO6*, Nd.62
597	0x255	RxPDO6*, Nd.21	618	0x26A	RxPDO6*,Nd.42	639	0x27F	RxPDO6*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
705	0x2C1	TxPDO7*, Nd.1	726	0x2D6	TxPDO7*, Nd.22	747	0x2EB	TxPDO7*, Nd.43
706	0x2C2	TxPDO7*, Nd.2	727	0x2D7	TxPDO7*, Nd.23	748	0x2EC	TxPDO7*, Nd.44
707	0x2C3	TxPDO7*, Nd.3	728	0x2D8	TxPDO7*, Nd.24	749	0x2ED	TxPDO7*, Nd.45
708	0x2C4	TxPDO7*, Nd.4	729	0x2D9	TxPDO7*, Nd.25	750	0x2EE	TxPDO7*, Nd.46
709	0x2C5	TxPDO7*, Nd.5	730	0x2DA	TxPDO7*, Nd.26	751	0x2EF	TxPDO7*, Nd.47
710	0x2C6	TxPDO7*, Nd.6	731	0x2DB	TxPDO7*, Nd.27	752	0x2F0	TxPDO7*, Nd.48
711	0x2C7	TxPDO7*, Nd.7	732	0x2DC	TxPDO7*, Nd.28	753	0x2F1	TxPDO7*, Nd.49
712	0x2C8	TxPDO7*, Nd.8	733	0x2DD	TxPDO7*, Nd.29	754	0x2F2	TxPDO7*, Nd.50
713	0x2C9	TxPDO7*, Nd.9	734	0x2DE	TxPDO7*, Nd.30	755	0x2F3	TxPDO7*, Nd.51
714	0x2CA	TxPDO7*, Nd.10	735	0x2DF	TxPDO7*, Nd.31	756	0x2F4	TxPDO7*, Nd.52
715	0x2CB	TxPDO7*, Nd.11	736	0x2E0	TxPDO7*, Nd.32	757	0x2F5	TxPDO7*, Nd.53
716	0x2CC	TxPDO7*, Nd.12	737	0x2E1	TxPDO7*, Nd.33	758	0x2F6	TxPDO7*, Nd.54
717	0x2CD	TxPDO7*, Nd.13	738	0x2E2	TxPDO7*, Nd.34	759	0x2F7	TxPDO7*, Nd.55
718	0x2CE	TxPDO7*, Nd.14	739	0x2E3	TxPDO7*, Nd.35	760	0x2F8	TxPDO7*, Nd.56
719	0x2CF	TxPDO7*, Nd.15	740	0x2E4	TxPDO7*, Nd.36	761	0x2F9	TxPDO7*, Nd.57
720	0x2D0	TxPDO7*, Nd.16	741	0x2E5	TxPDO7*, Nd.37	762	0x2FA	TxPDO7*, Nd.58
721	0x2D1	TxPDO7*, Nd.17	742	0x2E6	TxPDO7*, Nd.38	763	0x2FB	TxPDO7*, Nd.59
722	0x2D2	TxPDO7*, Nd.18	743	0x2E7	TxPDO7*, Nd.39	764	0x2FC	TxPDO7*, Nd.60
723	0x2D3	TxPDO7*, Nd.19	744	0x2E8	TxPDO7*, Nd.40	765	0x2FD	TxPDO7*, Nd.61
724	0x2D4	TxPDO7*, Nd.20	745	0x2E9	TxPDO7*, Nd.41	766	0x2FE	TxPDO7*, Nd.62
725	0x2D5	TxPDO7*, Nd.21	746	0x2EA	TxPDO7*, Nd.42	767	0x2FF	TxPDO7*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
833	0x341	RxPDO7*, Nd.1	854	0x356	RxPDO7*, Nd.22	875	0x36B	RxPDO7*, Nd.43
834	0x342	RxPDO7*, Nd.2	855	0x357	RxPDO7*, Nd.23	876	0x36C	RxPDO7*, Nd.44
835	0x343	RxPDO7*, Nd.3	856	0x358	RxPDO7*, Nd.24	877	0x36D	RxPDO7*, Nd.45
836	0x344	RxPDO7*, Nd.4	857	0x359	RxPDO7*, Nd.25	878	0x36E	RxPDO7*, Nd.46
837	0x345	RxPDO7*, Nd.5	858	0x35A	RxPDO7*, Nd.26	879	0x36F	RxPDO7*, Nd.47
838	0x346	RxPDO7*, Nd.6	859	0x35B	RxPDO7*, Nd.27	880	0x370	RxPDO7*, Nd.48
839	0x347	RxPDO7*, Nd.7	860	0x35C	RxPDO7*, Nd.28	881	0x371	RxPDO7*, Nd.49
840	0x348	RxPDO7*, Nd.8	861	0x35D	RxPDO7*, Nd.29	882	0x372	RxPDO7*, Nd.50
841	0x349	RxPDO7*, Nd.9	862	0x35E	RxPDO7*, Nd.30	883	0x373	RxPDO7*, Nd.51
842	0x34A	RxPDO7*, Nd.10	863	0x35F	RxPDO7*, Nd.31	884	0x374	RxPDO7*, Nd.52
843	0x34B	RxPDO7*, Nd.11	864	0x360	RxPDO7*, Nd.32	885	0x375	RxPDO7*, Nd.53
844	0x34C	RxPDO7*, Nd.12	865	0x361	RxPDO7*, Nd.33	886	0x376	RxPDO7*, Nd.54
845	0x34D	RxPDO7*, Nd.13	866	0x362	RxPDO7*, Nd.34	887	0x377	RxPDO7*, Nd.55
846	0x34E	RxPDO7*, Nd.14	867	0x363	RxPDO7*, Nd.35	888	0x378	RxPDO7*, Nd.56
847	0x34F	RxPDO7*, Nd.15	868	0x364	RxPDO7*, Nd.36	889	0x379	RxPDO7*, Nd.57
848	0x350	RxPDO7*, Nd.16	869	0x365	RxPDO7*, Nd.37	890	0x37A	RxPDO7*, Nd.58

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
849	0x351	RxPDO7*, Nd.17	870	0x366	RxPDO7*, Nd.38	891	0x37B	RxPDO7*, Nd.59
850	0x352	RxPDO7*, Nd.18	871	0x367	RxPDO7*, Nd.39	892	0x37C	RxPDO7*, Nd.60
851	0x353	RxPDO7*, Nd.19	872	0x368	RxPDO7*, Nd.40	893	0x37D	RxPDO7*, Nd.61
852	0x354	RxPDO7*, Nd.20	873	0x369	RxPDO7*, Nd.41	894	0x37E	RxPDO7*, Nd.62
853	0x355	RxPDO7*, Nd.21	874	0x36A	RxPDO7*, Nd.42	895	0x37F	RxPDO7*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
961	0x3C1	TxPDO8*, Nd.1	982	0x3D6	TxPDO8*, Nd.22	1003	0x3EB	TxPDO8*, Nd.43
962	0x3C2	TxPDO8*, Nd.2	983	0x3D7	TxPDO8*, Nd.23	1004	0x3EC	TxPDO8*, Nd.44
963	0x3C3	TxPDO8*, Nd.3	984	0x3D8	TxPDO8*, Nd.24	1005	0x3ED	TxPDO8*, Nd.45
964	0x3C4	TxPDO8*, Nd.4	985	0x3D9	TxPDO8*, Nd.25	1006	0x3EE	TxPDO8*, Nd.46
965	0x3C5	TxPDO8*, Nd.5	986	0x3DA	TxPDO8*, Nd.26	1007	0x3EF	TxPDO8*, Nd.47
966	0x3C6	TxPDO8*, Nd.6	987	0x3DB	TxPDO8*, Nd.27	1008	0x3F0	TxPDO8*, Nd.48
967	0x3C7	TxPDO8*, Nd.7	988	0x3DC	TxPDO8*, Nd.28	1009	0x3F1	TxPDO8*, Nd.49
968	0x3C8	TxPDO8*, Nd.8	989	0x3DD	TxPDO8*, Nd.29	1010	0x3F2	TxPDO8*, Nd.50
969	0x3C9	TxPDO8*, Nd.9	990	0x3DE	TxPDO8*, Nd.30	1011	0x3F3	TxPDO8*, Nd.51
970	0x3CA	TxPDO8*, Nd.10	991	0x3DF	TxPDO8*, Nd.31	1012	0x3F4	TxPDO8*, Nd.52
971	0x3CB	TxPDO8*, Nd.11	992	0x3E0	TxPDO8*, Nd.32	1013	0x3F5	TxPDO8*, Nd.53
972	0x3CC	TxPDO8*, Nd.12	993	0x3E1	TxPDO8*, Nd.33	1014	0x3F6	TxPDO8*, Nd.54
973	0x3CD	TxPDO8*, Nd.13	994	0x3E2	TxPDO8*, Nd.34	1015	0x3F7	TxPDO8*, Nd.55
974	0x3CE	TxPDO8*, Nd.14	995	0x3E3	TxPDO8*, Nd.35	1016	0x3F8	TxPDO8*, Nd.56
975	0x3CF	TxPDO8*, Nd.15	996	0x3E4	TxPDO8*, Nd.36	1017	0x3F9	TxPDO8*, Nd.57
976	0x3D0	TxPDO8*, Nd.16	997	0x3E5	TxPDO8*, Nd.37	1018	0x3FA	TxPDO8*, Nd.58
977	0x3D1	TxPDO8*, Nd.17	998	0x3E6	TxPDO8*, Nd.38	1019	0x3FB	TxPDO8*, Nd.59
978	0x3D2	TxPDO8*, Nd.18	999	0x3E7	TxPDO8*, Nd.39	1020	0x3FC	TxPDO8*, Nd.60
979	0x3D3	TxPDO8*, Nd.19	1000	0x3E8	TxPDO8*, Nd.40	1021	0x3FD	TxPDO8*, Nd.61
980	0x3D4	TxPDO8*, Nd.20	1001	0x3E9	TxPDO8*, Nd.41	1022	0x3FE	TxPDO8*, Nd.62
981	0x3D5	TxPDO8*, Nd.21	1002	0x3EA	TxPDO8*, Nd.42	1023	0x3FF	TxPDO8*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1089	0x441	RxPDO8*, Nd.1	1110	0x456	RxPDO8*, Nd.22	1131	0x46B	RxPDO8*, Nd.43
1090	0x442	RxPDO8*, Nd.2	1111	0x457	RxPDO8*, Nd.23	1132	0x46C	RxPDO8*, Nd.44
1091	0x443	RxPDO8*, Nd.3	1112	0x458	RxPDO8*, Nd.24	1133	0x46D	RxPDO8*, Nd.45
1092	0x444	RxPDO8*, Nd.4	1113	0x459	RxPDO8*, Nd.25	1134	0x46E	RxPDO8*, Nd.46
1093	0x445	RxPDO8*, Nd.5	1114	0x45A	RxPDO8*, Nd.26	1135	0x46F	RxPDO8*, Nd.47
1094	0x446	RxPDO8*, Nd.6	1115	0x45B	RxPDO8*, Nd.27	1136	0x470	RxPDO8*, Nd.48
1095	0x447	RxPDO8*, Nd.7	1116	0x45C	RxPDO8*, Nd.28	1137	0x471	RxPDO8*, Nd.49
1096	0x448	RxPDO8*, Nd.8	1117	0x45D	RxPDO8*, Nd.29	1138	0x472	RxPDO8*, Nd.50
1097	0x449	RxPDO8*, Nd.9	1118	0x45E	RxPDO8*, Nd.30	1139	0x473	RxPDO8*, Nd.51
1098	0x44A	RxPDO8*, Nd.10	1119	0x45F	RxPDO8*, Nd.31	1140	0x474	RxPDO8*, Nd.52
1099	0x44B	RxPDO8*, Nd.11	1120	0x460	RxPDO8*, Nd.32	1141	0x475	RxPDO8*, Nd.53
1100	0x44C	RxPDO8*, Nd.12	1121	0x461	RxPDO8*, Nd.33	1142	0x476	RxPDO8*, Nd.54
1101	0x44D	RxPDO8*, Nd.13	1122	0x462	RxPDO8*, Nd.34	1143	0x477	RxPDO8*, Nd.55
1102	0x44E	RxPDO8*, Nd.14	1123	0x463	RxPDO8*, Nd.35	1144	0x478	RxPDO8*, Nd.56
1103	0x44F	RxPDO8*, Nd.15	1124	0x464	RxPDO8*, Nd.36	1145	0x479	RxPDO8*, Nd.57
1104	0x450	RxPDO8*, Nd.16	1125	0x465	RxPDO8*, Nd.37	1146	0x47A	RxPDO8*, Nd.58
1105	0x451	RxPDO8*, Nd.17	1126	0x466	RxPDO8*, Nd.38	1147	0x47B	RxPDO8*, Nd.59
1106	0x452	RxPDO8*, Nd.18	1127	0x467	RxPDO8*, Nd.39	1148	0x47C	RxPDO8*, Nd.60
1107	0x453	RxPDO8*, Nd.19	1128	0x468	RxPDO8*, Nd.40	1149	0x47D	RxPDO8*, Nd.61
1108	0x454	RxPDO8*, Nd.20	1129	0x469	RxPDO8*, Nd.41	1150	0x47E	RxPDO8*, Nd.62
1109	0x455	RxPDO8*, Nd.21	1130	0x46A	RxPDO8*, Nd.42	1151	0x47F	RxPDO8*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1217	0x4C1	TxPDO9*, Nd.1	1238	0x4D6	TxPDO9*, Nd.22	1259	0x4EB	TxPDO9*, Nd.43
1218	0x4C2	TxPDO9*, Nd.2	1239	0x4D7	TxPDO9*, Nd.23	1260	0x4EC	TxPDO9*, Nd.44
1219	0x4C3	TxPDO9*, Nd.3	1240	0x4D8	TxPDO9*, Nd.24	1261	0x4ED	TxPDO9*, Nd.45
1220	0x4C4	TxPDO9*, Nd.4	1241	0x4D9	TxPDO9*, Nd.25	1262	0x4EE	TxPDO9*, Nd.46

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1221	0x4C5	TxPDO9*, Nd.5	1242	0x4DA	TxPDO9*, Nd.26	1263	0x4EF	TxPDO9*, Nd.47
1222	0x4C6	TxPDO9*, Nd.6	1243	0x4DB	TxPDO9*, Nd.27	1264	0x4F0	TxPDO9*, Nd.48
1223	0x4C7	TxPDO9*, Nd.7	1244	0x4DC	TxPDO9*, Nd.28	1265	0x4F1	TxPDO9*, Nd.49
1224	0x4C8	TxPDO9*, Nd.8	1245	0x4DD	TxPDO9*, Nd.29	1266	0x4F2	TxPDO9*, Nd.50
1225	0x4C9	TxPDO9*, Nd.9	1246	0x4DE	TxPDO9*, Nd.30	1267	0x4F3	TxPDO9*, Nd.51
1226	0x4CA	TxPDO9*, Nd.10	1247	0x4DF	TxPDO9*, Nd.31	1268	0x4F4	TxPDO9*, Nd.52
1227	0x4CB	TxPDO9*, Nd.11	1248	0x4E0	TxPDO9*, Nd.32	1269	0x4F5	TxPDO9*, Nd.53
1228	0x4CC	TxPDO9*, Nd.12	1249	0x4E1	TxPDO9*, Nd.33	1270	0x4F6	TxPDO9*, Nd.54
1229	0x4CD	TxPDO9*, Nd.13	1250	0x4E2	TxPDO9*, Nd.34	1271	0x4F7	TxPDO9*, Nd.55
1230	0x4CE	TxPDO9*, Nd.14	1251	0x4E3	TxPDO9*, Nd.35	1272	0x4F8	TxPDO9*, Nd.56
1231	0x4CF	TxPDO9*, Nd.15	1252	0x4E4	TxPDO9*, Nd.36	1273	0x4F9	TxPDO9*, Nd.57
1232	0x4D0	TxPDO9*, Nd.16	1253	0x4E5	TxPDO9*, Nd.37	1274	0x4FA	TxPDO9*, Nd.58
1233	0x4D1	TxPDO9*, Nd.17	1254	0x4E6	TxPDO9*, Nd.38	1275	0x4FB	TxPDO9*, Nd.59
1234	0x4D2	TxPDO9*, Nd.18	1255	0x4E7	TxPDO9*, Nd.39	1276	0x4FC	TxPDO9*, Nd.60
1235	0x4D3	TxPDO9*, Nd.19	1256	0x4E8	TxPDO9*, Nd.40	1277	0x4FD	TxPDO9*, Nd.61
1236	0x4D4	TxPDO9*, Nd.20	1257	0x4E9	TxPDO9*, Nd.41	1278	0x4FE	TxPDO9*, Nd.62
1237	0x4D5	TxPDO9*, Nd.21	1258	0x4EA	TxPDO9*, Nd.42	1279	0x4FF	TxPDO9*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1345	0x541	RxPDO9*, Nd.1	1366	0x556	RxPDO9*, Nd.22	1387	0x56B	RxPDO9*, Nd.43
1346	0x542	RxPDO9*, Nd.2	1367	0x557	RxPDO9*, Nd.23	1388	0x56C	RxPDO9*, Nd.44
1347	0x543	RxPDO9*, Nd.3	1368	0x558	RxPDO9*, Nd.24	1389	0x56D	RxPDO9*, Nd.45
1348	0x544	RxPDO9*, Nd.4	1369	0x559	RxPDO9*, Nd.25	1390	0x56E	RxPDO9*, Nd.46
1349	0x545	RxPDO9*, Nd.5	1370	0x55A	RxPDO9*, Nd.26	1391	0x56F	RxPDO9*, Nd.47
1350	0x546	RxPDO9*, Nd.6	1371	0x55B	RxPDO9*, Nd.27	1392	0x570	RxPDO9*, Nd.48
1351	0x547	RxPDO9*, Nd.7	1372	0x55C	RxPDO9*, Nd.28	1393	0x571	RxPDO9*, Nd.49
1352	0x548	RxPDO9*, Nd.8	1373	0x55D	RxPDO9*, Nd.29	1394	0x572	RxPDO9*, Nd.50
1353	0x549	RxPDO9*, Nd.9	1374	0x55E	RxPDO9*, Nd.30	1395	0x573	RxPDO9*, Nd.51
1354	0x54A	RxPDO9*, Nd.10	1375	0x55F	RxPDO9*, Nd.31	1396	0x574	RxPDO9*, Nd.52
1355	0x54B	RxPDO9*, Nd.11	1376	0x560	RxPDO9*, Nd.32	1397	0x575	RxPDO9*, Nd.53
1356	0x54C	RxPDO9*, Nd.12	1377	0x561	RxPDO9*, Nd.33	1398	0x576	RxPDO9*, Nd.54
1357	0x54D	RxPDO9*, Nd.13	1378	0x562	RxPDO9*, Nd.34	1399	0x577	RxPDO9*, Nd.55
1358	0x54E	RxPDO9*, Nd.14	1379	0x563	RxPDO9*, Nd.35	1400	0x578	RxPDO9*, Nd.56
1359	0x54F	RxPDO9*, Nd.15	1380	0x564	RxPDO9*, Nd.36	1401	0x579	RxPDO9*, Nd.57
1360	0x550	RxPDO9*, Nd.16	1381	0x565	RxPDO9*, Nd.37	1402	0x57A	RxPDO9*, Nd.58
1361	0x551	RxPDO9*, Nd.17	1382	0x566	RxPDO9*, Nd.38	1403	0x57B	RxPDO9*, Nd.59
1362	0x552	RxPDO9*, Nd.18	1383	0x567	RxPDO9*, Nd.39	1404	0x57C	RxPDO9*, Nd.60
1363	0x553	RxPDO9*, Nd.19	1384	0x568	RxPDO9*, Nd.40	1405	0x57D	RxPDO9*, Nd.61
1364	0x554	RxPDO9*, Nd.20	1385	0x569	RxPDO9*, Nd.41	1406	0x57E	RxPDO9*, Nd.62
1365	0x555	RxPDO9*, Nd.21	1386	0x56A	RxPDO9*, Nd.42	1407	0x57F	RxPDO9*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1473	0x5C1	TxPDO10*, Nd.1	1494	0x5D6	TxPDO10*, Nd.22	1515	0x5EB	TxPDO10*, Nd.43
1474	0x5C2	TxPDO10*, Nd.2	1495	0x5D7	TxPDO10*, Nd.23	1516	0x5EC	TxPDO10*, Nd.44
1475	0x5C3	TxPDO10*, Nd.3	1496	0x5D8	TxPDO10*, Nd.24	1517	0x5ED	TxPDO10*, Nd.45
1476	0x5C4	TxPDO10*, Nd.4	1497	0x5D9	TxPDO10*, Nd.25	1518	0x5EE	TxPDO10*, Nd.46
1477	0x5C5	TxPDO10*, Nd.5	1498	0x5DA	TxPDO10*, Nd.26	1519	0x5EF	TxPDO10*, Nd.47
1478	0x5C6	TxPDO10*, Nd.6	1499	0x5DB	TxPDO10*, Nd.27	1520	0x5F0	TxPDO10*, Nd.48
1479	0x5C7	TxPDO10*, Nd.7	1500	0x5DC	TxPDO10*, Nd.28	1521	0x5F1	TxPDO10*, Nd.49
1480	0x5C8	TxPDO10*, Nd.8	1501	0xDE	TxPDO10*, Nd.29	1522	0x5F2	TxPDO10*, Nd.50
1481	0x5C9	TxPDO10*, Nd.9	1502	0x5DE	TxPDO10*, Nd.30	1523	0x5F3	TxPDO10*, Nd.51
1482	0x5CA	TxPDO10*, Nd.10	1503	0x5DF	TxPDO10*, Nd.31	1524	0x5F4	TxPDO10*, Nd.52
1483	0x5CB	TxPDO10*, Nd.11	1504	0x5E0	TxPDO10*, Nd.32	1525	0x5F5	TxPDO10*, Nd.53
1484	0x5CC	TxPDO10*, Nd.12	1505	0x5E1	TxPDO10*, Nd.33	1526	0x5F6	TxPDO10*, Nd.54
1485	0x5CD	TxPDO10*, Nd.13	1506	0x5E2	TxPDO10*, Nd.34	1527	0x5F7	TxPDO10*, Nd.55
1486	0x5CE	TxPDO10*, Nd.14	1507	0x5E3	TxPDO10*, Nd.35	1528	0x5F8	TxPDO10*, Nd.56
1487	0x5CF	TxPDO10*, Nd.15	1508	0x5E4	TxPDO10*, Nd.36	1529	0x5F9	TxPDO10*, Nd.57

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1488	0x5D0	TxPDO10*, Nd.16	1509	0x5E5	TxPDO10*, Nd.37	1530	0x5FA	TxPDO10*, Nd.58
1489	0x5D1	TxPDO10*, Nd.17	1510	0x5E6	TxPDO10*, Nd.38	1531	0x5FB	TxPDO10*, Nd.59
1490	0x5D2	TxPDO10*, Nd.18	1511	0x5E7	TxPDO10*, Nd.39	1532	0x5FC	TxPDO10*, Nd.60
1491	0x5D3	TxPDO10*, Nd.19	1512	0x5E8	TxPDO10*, Nd.40	1533	0x5FD	TxPDO10*, Nd.61
1492	0x5D4	TxPDO10*, Nd.20	1513	0x5E9	TxPDO10*, Nd.41	1534	0x5FE	TxPDO10*, Nd.62
1493	0x5D5	TxPDO10*, Nd.21	1514	0x5EA	TxPDO10*, Nd.42	1535	0x5FF	TxPDO10*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1601	0x641	RxPDO10*, Nd.1	1622	0x656	RxPDO10*, Nd.22	1643	0x66B	RxPDO10*, Nd.43
1602	0x642	RxPDO10*, Nd.2	1623	0x657	RxPDO10*, Nd.23	1644	0x66C	RxPDO10*, Nd.44
1603	0x643	RxPDO10*, Nd.3	1624	0x658	RxPDO10*, Nd.24	1645	0x66D	RxPDO10*, Nd.45
1604	0x644	RxPDO10*, Nd.4	1625	0x659	RxPDO10*, Nd.25	1646	0x66E	RxPDO10*, Nd.46
1605	0x645	RxPDO10*, Nd.5	1626	0x65A	RxPDO10*, Nd.26	1647	0x66F	RxPDO10*, Nd.47
1606	0x646	RxPDO10*, Nd.6	1627	0x65B	RxPDO10*, Nd.27	1648	0x670	RxPDO10*, Nd.48
1607	0x647	RxPDO10*, Nd.7	1628	0x65C	RxPDO10*, Nd.28	1649	0x671	RxPDO10*, Nd.49
1608	0x648	RxPDO10*, Nd.8	1629	0x65D	RxPDO10*, Nd.29	1650	0x672	RxPDO10*, Nd.50
1609	0x649	RxPDO10*, Nd.9	1630	0x65E	RxPDO10*, Nd.30	1651	0x673	RxPDO10*, Nd.51
1610	0x64A	RxPDO10*, Nd.10	1631	0x65F	RxPDO10*, Nd.31	1652	0x674	RxPDO10*, Nd.52
1611	0x64B	RxPDO10*, Nd.11	1632	0x660	RxPDO10*, Nd.32	1653	0x675	RxPDO10*, Nd.53
1612	0x64C	RxPDO10*, Nd.12	1633	0x661	RxPDO10*, Nd.33	1654	0x676	RxPDO10*, Nd.54
1613	0x64D	RxPDO10*, Nd.13	1634	0x662	RxPDO10*, Nd.34	1655	0x677	RxPDO10*, Nd.55
1614	0x64E	RxPDO10*, Nd.14	1635	0x663	RxPDO10*, Nd.35	1656	0x678	RxPDO10*, Nd.56
1615	0x64F	RxPDO10*, Nd.15	1636	0x664	RxPDO10*, Nd.36	1657	0x679	RxPDO10*, Nd.57
1616	0x650	RxPDO10*, Nd.16	1637	0x665	RxPDO10*, Nd.37	1658	0x67A	RxPDO10*, Nd.58
1617	0x651	RxPDO10*, Nd.17	1638	0x666	RxPDO10*, Nd.38	1659	0x67B	RxPDO10*, Nd.59
1618	0x652	RxPDO10*, Nd.18	1639	0x667	RxPDO10*, Nd.39	1660	0x67C	RxPDO10*, Nd.60
1619	0x653	RxPDO10*, Nd.19	1640	0x668	RxPDO10*, Nd.40	1661	0x67D	RxPDO10*, Nd.61
1620	0x654	RxPDO10*, Nd.20	1641	0x669	RxPDO10*, Nd.41	1662	0x67E	RxPDO10*, Nd.62
1621	0x655	RxPDO10*, Nd.21	1642	0x66A	RxPDO10*, Nd.42	1663	0x67F	RxPDO10*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1729	0x6C1	TxPDO11*, Nd.1	1750	0x6D6	TxPDO11*, Nd.22	1771	0x6EB	TxPDO11*, Nd.43
1730	0x6C2	TxPDO11*, Nd.2	1751	0x6D7	TxPDO11*, Nd.23	1772	0x6EC	TxPDO11*, Nd.44
1731	0x6C3	TxPDO11*, Nd.3	1752	0x6D8	TxPDO11*, Nd.24	1773	0x6ED	TxPDO11*, Nd.45
1732	0x6C4	TxPDO11*, Nd.4	1753	0x6D9	TxPDO11*, Nd.25	1774	0x6EE	TxPDO11*, Nd.46
1733	0x6C5	TxPDO11*, Nd.5	1754	0x6DA	TxPDO11*, Nd.26	1775	0x6EF	TxPDO11*, Nd.47
1734	0x6C6	TxPDO11*, Nd.6	1755	0x6DB	TxPDO11*, Nd.27	1776	0x6F0	TxPDO11*, Nd.48
1735	0x6C7	TxPDO11*, Nd.7	1756	0x6DC	TxPDO11*, Nd.28	1777	0x6F1	TxPDO11*, Nd.49
1736	0x6C8	TxPDO11*, Nd.8	1757	0x6DD	TxPDO11*, Nd.29	1778	0x6F2	TxPDO11*, Nd.50
1737	0x6C9	TxPDO11*, Nd.9	1758	0x6DE	TxPDO11*, Nd.30	1779	0x6F3	TxPDO11*, Nd.51
1738	0x6CA	TxPDO11*, Nd.10	1759	0x6DF	TxPDO11*, Nd.31	1780	0x6F4	TxPDO11*, Nd.52
1739	0x6CB	TxPDO11*, Nd.11	1760	0x6E0	TxPDO11*, Nd.32	1781	0x6F5	TxPDO11*, Nd.53
1740	0x6CC	TxPDO11*, Nd.12	1761	0x6E1	TxPDO11*, Nd.33	1782	0x6F6	TxPDO11*, Nd.54
1741	0x6CD	TxPDO11*, Nd.13	1762	0x6E2	TxPDO11*, Nd.34	1783	0x6F7	TxPDO11*, Nd.55
1742	0x6CE	TxPDO11*, Nd.14	1763	0x6E3	TxPDO11*, Nd.35	1784	0x6F8	TxPDO11*, Nd.56
1743	0x6CF	TxPDO11*, Nd.15	1764	0x6E4	TxPDO11*, Nd.36	1785	0x6F9	TxPDO11*, Nd.57
1744	0x6D0	TxPDO11*, Nd.16	1765	0x6E5	TxPDO11*, Nd.37	1786	0x6FA	TxPDO11*, Nd.58
1745	0x6D1	TxPDO11*, Nd.17	1766	0x6E6	TxPDO11*, Nd.38	1787	0x6FB	TxPDO11*, Nd.59
1746	0x6D2	TxPDO11*, Nd.18	1767	0x6E7	TxPDO11*, Nd.39	1788	0x6FC	TxPDO11*, Nd.60
1747	0x6D3	TxPDO11*, Nd.19	1768	0x6E8	TxPDO11*, Nd.40	1789	0x6FD	TxPDO11*, Nd.61
1748	0x6D4	TxPDO11*, Nd.20	1769	0x6E9	TxPDO11*, Nd.41	1790	0x6FE	TxPDO11*, Nd.62
1749	0x6D5	TxPDO11*, Nd.21	1770	0x6EA	TxPDO11*, Nd.42	1791	0x6FF	TxPDO11*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1857	0x741	RxPDO11*, Nd.1	1878	0x756	RxPDO11*, Nd.22	1899	0x76B	RxPDO11*, Nd.43
1858	0x742	RxPDO11*, Nd.2	1879	0x757	RxPDO11*, Nd.23	1900	0x76C	RxPDO11*, Nd.44
1859	0x743	RxPDO11*, Nd.3	1880	0x758	RxPDO11*, Nd.24	1901	0x76D	RxPDO11*, Nd.45

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1860	0x744	RxPDO11*, Nd.4	1881	0x759	RxPDO11*, Nd.25	1902	0x76E	RxPDO11*, Nd.46
1861	0x745	RxPDO11*, Nd.5	1882	0x75A	RxPDO11*, Nd.26	1903	0x76F	RxPDO11*, Nd.47
1862	0x746	RxPDO11*, Nd.6	1883	0x75B	RxPDO11*, Nd.27	1904	0x770	RxPDO11*, Nd.48
1863	0x747	RxPDO11*, Nd.7	1884	0x75C	RxPDO11*, Nd.28	1905	0x771	RxPDO11*, Nd.49
1864	0x748	RxPDO11*, Nd.8	1885	0x75D	RxPDO11*, Nd.29	1906	0x772	RxPDO11*, Nd.50
1865	0x749	RxPDO11*, Nd.9	1886	0x75E	RxPDO11*, Nd.30	1907	0x773	RxPDO11*, Nd.51
1866	0x74A	RxPDO11*, Nd.10	1887	0x75F	RxPDO11*, Nd.31	1908	0x774	RxPDO11*, Nd.52
1867	0x74B	RxPDO11*, Nd.11	1888	0x760	RxPDO11*, Nd.32	1909	0x775	RxPDO11*, Nd.53
1868	0x74C	RxPDO11*, Nd.12	1889	0x761	RxPDO11*, Nd.33	1910	0x776	RxPDO11*, Nd.54
1869	0x74D	RxPDO11*, Nd.13	1890	0x762	RxPDO11*, Nd.34	1911	0x777	RxPDO11*, Nd.55
1870	0x74E	RxPDO11*, Nd.14	1891	0x763	RxPDO11*, Nd.35	1912	0x778	RxPDO11*, Nd.56
1871	0x74F	RxPDO11*, Nd.15	1892	0x764	RxPDO11*, Nd.36	1913	0x779	RxPDO11*, Nd.57
1872	0x750	RxPDO11*, Nd.16	1893	0x765	RxPDO11*, Nd.37	1914	0x77A	RxPDO11*, Nd.58
1873	0x751	RxPDO11*, Nd.17	1894	0x766	RxPDO11*, Nd.38	1915	0x77B	RxPDO11*, Nd.59
1874	0x752	RxPDO11*, Nd.18	1895	0x767	RxPDO11*, Nd.39	1916	0x77C	RxPDO11*, Nd.60
1875	0x753	RxPDO11*, Nd.19	1896	0x768	RxPDO11*, Nd.40	1917	0x77D	RxPDO11*, Nd.61
1876	0x754	RxPDO11*, Nd.20	1897	0x769	RxPDO11*, Nd.41	1918	0x77E	RxPDO11*, Nd.62
1877	0x755	RxPDO11*, Nd.21	1898	0x76A	RxPDO11*, Nd.42	1919	0x77F	RxPDO11*, Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1409	0x581	SDO Tx Nd.1	1430	0x596	SDO Tx Nd.22	1451	0x5AB	SDO Tx Nd.43
1410	0x582	SDO Tx Nd.2	1431	0x597	SDO Tx Nd.23	1452	0x5AC	SDO Tx Nd.44
1411	0x583	SDO Tx Nd.3	1432	0x598	SDO Tx Nd.24	1453	0x5AD	SDO Tx Nd.45
1412	0x584	SDO Tx Nd.4	1433	0x599	SDO Tx Nd.25	1454	0x5AE	SDO Tx Nd.46
1413	0x585	SDO Tx Nd.5	1434	0x59A	SDO Tx Nd.26	1455	0x5AF	SDO Tx Nd.47
1414	0x586	SDO Tx Nd.6	1435	0x59B	SDO Tx Nd.27	1456	0x5B0	SDO Tx Nd.48
1415	0x587	SDO Tx Nd.7	1436	0x59C	SDO Tx Nd.28	1457	0x5B1	SDO Tx Nd.49
1416	0x588	SDO Tx Nd.8	1437	0x59D	SDO Tx Nd.29	1458	0x5B2	SDO Tx Nd.50
1417	0x589	SDO Tx Nd.9	1438	0x59E	SDO Tx Nd.30	1459	0x5B3	SDO Tx Nd.51
1418	0x58A	SDO Tx Nd.10	1439	0x59F	SDO Tx Nd.31	1460	0x5B4	SDO Tx Nd.52
1419	0x58B	SDO Tx Nd.11	1440	0x5A0	SDO Tx Nd.32	1461	0x5B5	SDO Tx Nd.53
1420	0x58C	SDO Tx Nd.12	1441	0x5A1	SDO Tx Nd.33	1462	0x5B6	SDO Tx Nd.54
1421	0x58D	SDO Tx Nd.13	1442	0x5A2	SDO Tx Nd.34	1463	0x5B7	SDO Tx Nd.55
1422	0x58E	SDO Tx Nd.14	1443	0x5A3	SDO Tx Nd.35	1464	0x5B8	SDO Tx Nd.56
1423	0x58F	SDO Tx Nd.15	1444	0x5A4	SDO Tx Nd.36	1465	0x5B9	SDO Tx Nd.57
1424	0x590	SDO Tx Nd.16	1445	0x5A5	SDO Tx Nd.37	1466	0x5BA	SDO Tx Nd.58
1425	0x591	SDO Tx Nd.17	1446	0x5A6	SDO Tx Nd.38	1467	0x5BB	SDO Tx Nd.59
1426	0x592	SDO Tx Nd.18	1447	0x5A7	SDO Tx Nd.39	1468	0x5BC	SDO Tx Nd.60
1427	0x593	SDO Tx Nd.19	1448	0x5A8	SDO Tx Nd.40	1469	0x5BD	SDO Tx Nd.61
1428	0x594	SDO Tx Nd.20	1449	0x5A9	SDO Tx Nd.41	1470	0x5BE	SDO Tx Nd.62
1429	0x595	SDO Tx Nd.21	1450	0x5AA	SDO Tx Nd.42	1471	0x5BF	SDO Tx Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1537	0x601	SDO Rx Nd.1	1558	0x616	SDO Rx Nd.22	1579	0x62B	SDO Rx Nd.43
1538	0x602	SDO Rx Nd.2	1559	0x617	SDO Rx Nd.23	1580	0x62C	SDO Rx Nd.44
1539	0x603	SDO Rx Nd.3	1560	0x618	SDO Rx Nd.24	1581	0x62D	SDO Rx Nd.45
1540	0x604	SDO Rx Nd.4	1561	0x619	SDO Rx Nd.25	1582	0x62E	SDO Rx Nd.46
1541	0x605	SDO Rx Nd.5	1562	0x61A	SDO Rx Nd.26	1583	0x62F	SDO Rx Nd.47
1542	0x606	SDO Rx Nd.6	1563	0x61B	SDO Rx Nd.27	1584	0x630	SDO Rx Nd.48
1543	0x607	SDO Rx Nd.7	1564	0x61C	SDO Rx Nd.28	1585	0x631	SDO Rx Nd.49
1544	0x608	SDO Rx Nd.8	1565	0x61D	SDO Rx Nd.29	1586	0x632	SDO Rx Nd.50
1545	0x609	SDO Rx Nd.9	1566	0x61E	SDO Rx Nd.30	1587	0x633	SDO Rx Nd.51
1546	0x60A	SDO Rx Nd.10	1567	0x61F	SDO Rx Nd.31	1588	0x634	SDO Rx Nd.52
1547	0x60B	SDO Rx Nd.11	1568	0x620	SDO Rx Nd.32	1589	0x635	SDO Rx Nd.53
1548	0x60C	SDO Rx Nd.12	1569	0x621	SDO Rx Nd.33	1590	0x636	SDO Rx Nd.54
1549	0x60D	SDO Rx Nd.13	1570	0x622	SDO Rx Nd.34	1591	0x637	SDO Rx Nd.55
1550	0x60E	SDO Rx Nd.14	1571	0x623	SDO Rx Nd.35	1592	0x638	SDO Rx Nd.56

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1551	0x60F	SDO Rx Nd.15	1572	0x624	SDO Rx Nd.36	1593	0x639	SDO Rx Nd.57
1552	0x610	SDO Rx Nd.16	1573	0x625	SDO Rx Nd.37	1594	0x63A	SDO Rx Nd.58
1553	0x611	SDO Rx Nd.17	1574	0x626	SDO Rx Nd.38	1595	0x63B	SDO Rx Nd.59
1554	0x612	SDO Rx Nd.18	1575	0x627	SDO Rx Nd.39	1596	0x63C	SDO Rx Nd.60
1555	0x613	SDO Rx Nd.19	1576	0x628	SDO Rx Nd.40	1597	0x63D	SDO Rx Nd.61
1556	0x614	SDO Rx Nd.20	1577	0x629	SDO Rx Nd.41	1598	0x63E	SDO Rx Nd.62
1557	0x615	SDO Rx Nd.21	1578	0x62A	SDO Rx Nd.42	1599	0x63F	SDO Rx Nd.63

dec	hex	Telegram type	dec	hex	Telegram type	dec	hex	Telegram type
1793	0x701	Guarding Nd.1	1814	0x716	Guarding Nd.22	1835	0x72B	Guarding Nd.43
1794	0x702	Guarding Nd.2	1815	0x717	Guarding Nd.23	1836	0x72C	Guarding Nd.44
1795	0x703	Guarding Nd.3	1816	0x718	Guarding Nd.24	1837	0x72D	Guarding Nd.45
1796	0x704	Guarding Nd.4	1817	0x719	Guarding Nd.25	1838	0x72E	Guarding Nd.46
1797	0x705	Guarding Nd.5	1818	0x71A	Guarding Nd.26	1839	0x72F	Guarding Nd.47
1798	0x706	Guarding Nd.6	1819	0x71B	Guarding Nd.27	1840	0x730	Guarding Nd.48
1799	0x707	Guarding Nd.7	1820	0x71C	Guarding Nd.28	1841	0x731	Guarding Nd.49
1800	0x708	Guarding Nd.8	1821	0x71D	Guarding Nd.29	1842	0x732	Guarding Nd.50
1801	0x709	Guarding Nd.9	1822	0x71E	Guarding Nd.30	1843	0x733	Guarding Nd.51
1802	0x70A	Guarding Nd.10	1823	0x71F	Guarding Nd.31	1844	0x734	Guarding Nd.52
1803	0x70B	Guarding Nd.11	1824	0x720	Guarding Nd.32	1845	0x735	Guarding Nd.53
1804	0x70C	Guarding Nd.12	1825	0x721	Guarding Nd.33	1846	0x736	Guarding Nd.54
1805	0x70D	Guarding Nd.13	1826	0x722	Guarding Nd.34	1847	0x737	Guarding Nd.55
1806	0x70E	Guarding Nd.14	1827	0x723	Guarding Nd.35	1848	0x738	Guarding Nd.56
1807	0x70F	Guarding Nd.15	1828	0x724	Guarding Nd.36	1849	0x739	Guarding Nd.57
1808	0x710	Guarding Nd.16	1829	0x725	Guarding Nd.37	1850	0x73A	Guarding Nd.58
1809	0x711	Guarding Nd.17	1830	0x726	Guarding Nd.38	1851	0x73B	Guarding Nd.59
1810	0x712	Guarding Nd.18	1831	0x727	Guarding Nd.39	1852	0x73C	Guarding Nd.60
1811	0x713	Guarding Nd.19	1832	0x728	Guarding Nd.40	1853	0x73D	Guarding Nd.61
1812	0x714	Guarding Nd.20	1833	0x729	Guarding Nd.41	1854	0x73E	Guarding Nd.62
1813	0x715	Guarding Nd.21	1834	0x72A	Guarding Nd.42	1855	0x73F	Guarding Nd.63

11.2 Komponenten Dritter

Dieses Gerät enthält Software von Beckhoff und Dritten.
Bitte beachten Sie die auf dem Speichermedium enthaltene Lizenzdatei.

11.3 Zubehör

Tab. 34: MicroSD-Karten.

Bestellnummer	Beschreibung
CX1900-0122	512-MB-MicroSD-Karte
CX1900-0132	16-GB-MicroSD-Karte

Tab. 35: Weitere Ersatzteile.

Bestellnummer	Beschreibung
ZB8701	Schlitzschraubendreher 2,0 x 40 mm, HD-Klemmen

11.4 Zertifizierungen

FCC Approvals for the United States of America

FCC: Federal Communications Commission Radio Frequency Interference Statement

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

FCC Approval for Canada

FCC: Canadian Notice

This equipment does not exceed the Class A limits for radiated emissions as described in the Radio Interference Regulations of the Canadian Department of Communications.

Tabellenverzeichnis

Abbildungsverzeichnis

Mehr Informationen:
www.beckhoff.com/CX7050

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