

Handbuch | DE

# CXxxxx-M510/B510

CANopen-Optionsschnittstelle für CX9020, CX5xx0 und CX20xx



# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b> .....	<b>5</b>
1.1	Hinweise zur Dokumentation .....	5
1.2	Sicherheitshinweise .....	6
1.3	Ausgabestände der Dokumentation .....	7
<b>2</b>	<b>Systemübersicht CANopen</b> .....	<b>8</b>
2.1	Netzwerkmanagement.....	10
2.1.1	Boot-Up-Nachricht .....	12
2.1.2	Knotenüberwachung.....	12
2.1.3	Prozessdatenobjekte (PDO) .....	14
2.1.4	Servicedatenobjekte (SDO) .....	19
2.2	Technische Daten - CANopen.....	23
<b>3</b>	<b>Anschluss und Verkabelung</b> .....	<b>24</b>
3.1	CANopen-Anschluss .....	24
3.2	Verkabelung .....	25
3.3	Topologie.....	30
<b>4</b>	<b>TwinCAT Registerkarten</b> .....	<b>31</b>
4.1	Strukturansicht.....	31
4.2	CANopen-Master .....	33
4.2.1	General .....	33
4.2.2	FC 51xx .....	34
4.2.3	ADS .....	35
4.3	CANopen-Slave .....	36
4.3.1	CAN Node.....	36
4.3.2	SDOs .....	39
4.3.3	PDO .....	40
<b>5</b>	<b>Parametrieren und in Betrieb nehmen</b> .....	<b>41</b>
5.1	PDO-Parametrierung.....	41
5.2	Mit TwinCAT 2 parametrieren.....	44
5.2.1	Zielsysteme suchen .....	44
5.2.2	CANopen-Slave anfügen .....	46
5.2.3	Virtuellen Slave anlegen .....	47
5.2.4	Adresse einstellen .....	48
5.2.5	Weitere PDOs anlegen .....	49
5.2.6	Variablen anlegen.....	50
5.2.7	Übertragungsart festlegen .....	51
5.2.8	PLC-Projekt erstellen.....	52
5.2.9	Variablen verknüpfen.....	54
5.2.10	Konfiguration auf CX laden.....	55
5.2.11	CANopen-Master anfügen .....	57
5.3	Mit TwinCAT 3 parametrieren.....	59
5.3.1	Zielsysteme suchen .....	59
5.3.2	CANopen-Slave anfügen .....	61
5.3.3	Virtuellen Slave anlegen.....	62

5.3.4	Adresse einstellen .....	63
5.3.5	Weitere PDOs anlegen .....	64
5.3.6	Variablen anlegen .....	65
5.3.7	Übertragungsart festlegen .....	66
5.3.8	PLC-Projekt erstellen .....	67
5.3.9	Variablen verknüpfen .....	69
5.3.10	Konfiguration auf CX laden .....	70
5.3.11	CANopen-Master anfügen .....	72
<b>6</b>	<b>Fehlerbehandlung und Diagnose .....</b>	<b>74</b>
6.1	Diagnose-LEDs .....	74
6.2	Statusmeldungen .....	75
6.3	Kommunikation .....	76
6.4	PDOs .....	77
<b>7</b>	<b>Anhang .....</b>	<b>78</b>
7.1	Zubehör .....	78
7.2	Zertifizierungen .....	79
7.3	Support und Service .....	80

# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

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

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

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

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

### Disclaimer

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

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

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

### Marken

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

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

### Patente

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

EP1590927, EP1789857, DE102004044764, DE102007017835

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

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

EP0851348, US6167425 mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

## 1.2 Sicherheitshinweise

### Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!  
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

### Haftungsausschluss

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

### Qualifikation des Personals

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

### Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

#### **GEFAHR**

##### **Akute Verletzungsgefahr!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

#### **WARNUNG**

##### **Verletzungsgefahr!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

#### **VORSICHT**

##### **Schädigung von Personen!**

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

#### **HINWEIS**

##### **Schädigung von Umwelt oder Geräten**

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

#### **Tipp oder Fingerzeig**

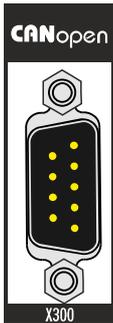


Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

## 1.3 Ausgabestände der Dokumentation

Version	Änderungen
1.0	Erste Version

## 2 Systemübersicht CANopen



Die Beckhoff Embedded-PCs können ab Werk mit einer Optionsschnittstelle z.B. PROFIBUS, CANopen oder RS232 bestellt werden. Einige der Optionsschnittstellen können als Master oder Slave ausgeliefert werden.

Folgende Embedded-PCs können mit einer Optionsschnittstelle bestellt werden:

- CX9020
- CX50x0
- CX51x0
- CX20x0

### CANopen-Master (M510)

Die Optionsschnittstelle M510 ist ein CANopen-Master und ermöglicht einen segmentartigen Aufbau von Steuerungsstrukturen in weitläufigen Anlagen und Maschinen. Weitere Beckhoff Feldbuskomponenten wie Buskoppler, Busklemmen Controller, Antriebstechnik etc. können mit einem Embedded-PC für den Aufbau von Steuerungsstrukturen benutzt werden.

Mit Feldbusmastern werden Prozessdaten und Signale dezentral auch bei weitläufigen Maschinen und Anlagen eingesammelt. Die Anzahl der möglichen Slaves, die an den Master angeschlossen werden können, ist nur durch das jeweilige Bussystem begrenzt. Durch den Einsatz von Master- bzw. Slaveanschlüssen können mehrere Embedded-PC untereinander über die Feldebene vernetzt werden.

In TwinCAT werden die Optionsschnittstellen erkannt, parametrieren, konfiguriert und die angeschlossenen I/O-Komponenten angefügt. Auch die Diagnose erfolgt in TwinCAT.

### CANopen-Slave (B510)

Die Optionsschnittstelle B510 ist ein CANopen-Slave und ermöglicht es einen Embedded-PC als unterlagerte dezentrale Steuerung für den Aufbau komplexer oder modularer Anlagen einzusetzen.

Der CANopen-Slave empfängt externe Prozessdaten vom Master, verarbeitet diese Daten weiter oder schickt Daten aus der eigenen Prozessperipherie aufbereitet an den Master zurück.

Die CANopen-Slave Optionsschnittstelle wird wie der CANopen-Master in TwinCAT parametrieren und konfiguriert.

### Funktionsweise

CANopen ist ein weit verbreitetes Feldbussystem, das im Verband CAN in Automation (CiA) entwickelt und inzwischen zur internationalen Normung angenommen wurde.

#### Weitere Informationen

CAN in Automation (CiA) Homepage:  
[www.can-cia.org](http://www.can-cia.org)

CANopen besteht aus der Protokolldefinition (Kommunikationsobjekte) und den Geräteprofilen.

Die Protokolldefinition beinhaltet folgende Kommunikationsobjekte:

- Netzwerkmanagement (NMT)
- Prozessdatenobjekte (PDO)
- Servicedatenobjekte (SDO)
- Und Protokolle mit speziellen Funktionen

Die Geräteprofile normieren den Dateninhalt für die jeweilige Geräteklasse. Unter den Begriff Geräteklasse fallen Geräte wie z.B. elektrische Antriebe, I/O-Module, Sensoren und Regler. Die Geräteprofile definieren die Funktionalität und den Aufbau des Objektverzeichnisses.

Die Geräteparameter und Prozessdaten sind in einem Objektverzeichnis strukturiert. Der Zugriff auf beliebige Daten dieses Objektverzeichnisses erfolgt über die Servicedatenobjekte (SDO).

Zur schnellen Kommunikation der Ein- und Ausgangsdaten dienen die Prozessdatenobjekte (PDO). Weiter gibt es einige Spezialobjekte (bzw. Telegrammarten) für Netzwerkmanagement (NMT), Synchronisation, Fehlermeldungen etc.

### **Kommunikationsarten**

CANopen definiert mehrere Kommunikationsarten für Prozessdatenobjekte:

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

Die gewünschte Kommunikationsart wird über den Parameter Transmission Type eingestellt (siehe: Registerkarte [PDO](#) [▶ 40]).

### **Buszugriffsverfahren**

CAN arbeitet nach dem Verfahren Carrier Sense Multiple Access (CSMA), d.h. dass jeder Teilnehmer gleichberechtigt ist und auf den Bus zugreift, 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 ([www.beckhoff.de](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 ([www.can-cia.org](http://www.can-cia.org)) zertifiziert.

## 2.1 Netzwerkmanagement

Das Netzwerkmanagement (NMT) definiert das Kommunikationsverhalten eines CANopen Teilnehmers und besteht aus den Zuständen Initialisation, Pre-Operational, Operational und Stopped.

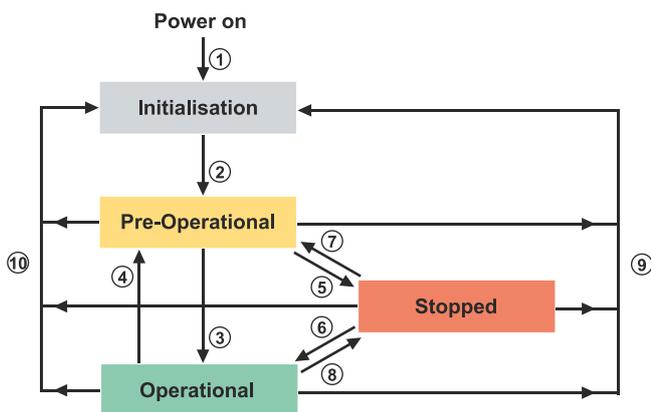
Wird ein Teilnehmer eingeschaltet oder neu gestartet, wechselt der Teilnehmer automatisch in den Zustand Initialisation. Sobald der Zustand Initialisation abgeschlossen ist, wechselt der Teilnehmer automatisch in den Zustand Pre-Operational.

Von diesem Zustand aus kann jeder andere Zustand eingenommen werden. Damit z.B. die Teilnehmer starten, ist nur eine einzige CAN-Nachricht erforderlich:

Start\_Remote\_Node: Identifier 0, zwei Datenbytes: 0x01, 0x00.

Diese Nachricht überführt die Teilnehmer in den Zustand Operational.

Das folgende Diagramm zeigt welche Zustände ein CANopen Teilnehmer einnehmen kann:



- Initialisation**  
 Der Teilnehmer geht automatisch in den Zustand Initialisation. Sobald der Zustand Initialisation abgeschlossen ist, geht der Teilnehmer automatisch in den Zustand Pre-Operational.
- Pre-Operational**  
 Nach der Initialisierung wechselt der Teilnehmer automatisch, d.h. ohne Befehl von außen, in den Zustand Pre-Operational. In diesem Zustand sind die Servicedatenobjekte (SDO) bereits aktiv und der Teilnehmer kann konfiguriert werden. Die Prozessdatenobjekte (PDO) sind noch gesperrt.
- Operational**  
 Im Zustand Operational sind die Prozessdatenobjekte (PDO) aktiv.  
 Wenn der Teilnehmer wegen äußerer Einflüsse (z.B. CAN-Störung, keine Ausgangsspannung) oder innerer Einflüsse (z.B. K-Bus-Fehler) nicht mehr in der Lage ist, Ausgänge zu setzen oder Eingänge zu lesen bzw. zu kommunizieren, dann versucht er eine entsprechende Emergency-Nachricht zu senden. Danach geht der Teilnehmer in den Fehlerzustand und wechselt in den Zustand Pre-Operational zurück. Damit kann auch die NMT-Statusmaschine des Masters fatale Fehler sofort erkennen.
- Stopped**  
 Im Zustand Stopped (früher *Prepared*) ist keine Kommunikation mit dem Teilnehmer möglich. Nur Netzwerkmanagement (NMT) Nachrichten werden empfangen. Die Ausgänge gehen in den Fehlerzustand.

**Zustandsübergänge**

Die Zustandsübergänge werden mit einer CAN-Nachricht ausgeführt. Die CAN-Nachrichten haben einen sehr einfachen Aufbau:  
CAN-Identifizier 0 mit zwei Byte Dateninhalt.

- Das erste Datenbyte enthält den sogenannten Command-Specifier (cs),
- das zweite Datenbyte die Knotenadresse (Node-ID), wobei die Knotenadresse 0 alle Knoten anspricht (Broadcast).

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

Die folgende Tabelle gibt einen Überblick über alle möglichen Zustandsübergänge und die dazugehörigen Command Specifier (cs). Nehmen Sie auch das zuvor gezeigte Diagramm, dass die Zustände veranschaulicht, als Übersicht dazu:

Zustandsübergang	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.
(3), (6)	cs = 1 = 0x01	Start_Remote_Node. Startet den Teilnehmer, gibt Ausgänge frei und startet die PDO-Übertragung.
(4), (7)	cs = 128 = 0x80	Enter_Pre-Operational. Stoppt die PDO-Übertragung, SDO weiter aktiv.
(5), (8)	cs = 2 = 0x02	Stop_Remote_Node. Ausgänge gehen in den Fehlerzustand, SDO und PDO abgeschaltet.
(9)	cs = 129 = 0x81	Reset_Node. Führt Reset durch. Alle Objekte werden auf Power-On Defaults zurückgesetzt.
(10)	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 Teilnehmer in den Fehlerzustand (Ausgänge sicherer Zustand) überführt:

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

**Beispiel 2**

Mit folgendem Telegramm wird Knoten 17 zurückgesetzt (neu gestartet):

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

## 2.1.1 Boot-Up-Nachricht

Nach der Initialisierungsphase und dem Selbsttest sendet der Teilnehmer 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 Teilnehmer signalisiert, dass er nun konfiguriert bzw. gestartet werden kann.

### Format Boot-Up Nachricht

11-bit Identifier	2 Byte Nutzdaten								
0x700 (=1792) + Node-ID	0x00								

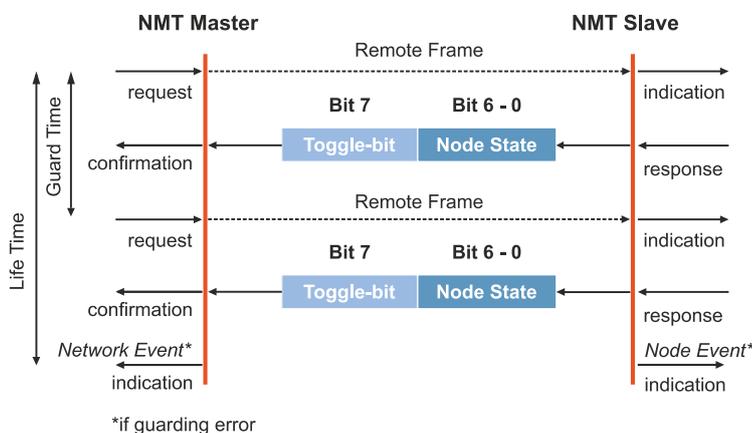
## 2.1.2 Knotenüberwachung

Damit Ausfälle der CANopen-Teilnehmer überwacht werden, stehen die Mechanismen Heartbeat und Guarding zur Verfügung. Diese Mechanismen sind bei CANopen besonders wichtig, da sich die Teilnehmer in der ereignisgesteuerten Betriebsart nicht regelmäßig melden. Beim Guarding werden die Teilnehmer mit einem 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 Slaves überwacht, die ihrerseits über Life-Guarding einen ausgefallenen Guarding-Master erkennen.

Beim Node-Guarding setzt der Master Anforderungstelegramme, sogenannte Remote Frames (remote transmit requests) auf die Guarding-Identifier der Slaves ab, die überwacht werden sollen. Die Slaves antworten mit einer Guarding-Nachricht. Die Nachricht enthält den Node-State des Slaves und ein Toggle-bit, welches nach jeder Nachricht wechseln muss. Falls der Node-State oder das Toggle-bit nicht mit den Werten übereinstimmen, die vom Master erwartet werden oder generell keine Antwort erfolgt, geht der Master von einem Fehler aus.



Bei der ersten Guarding-Nachricht des Slaves hat das Toggle-bit den Wert 0. Anschließend wechselt das Toggle-bit nach jeder Guarding-Nachricht und ermöglicht es festzustellen ob eine Nachricht verloren ging. Die restlichen sieben Bit enthalten den Node-State und übertragen damit den Status des Slaves an den Master.

Node-State	Status
4 = 0x04	Stopped
5 = 0x05	Operational
127 = 0x7F	Pre-Operational

**Beispiel**

Die Guarding Nachricht des Knotens 27 (0x1B) muss mit einem Remote Frame mit Identifier 0x71B (1819<sub>dez</sub>) 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 des eingestellten Node Life Time keine Nachrichten-anforderung vom Master erhält (Guarding-Fehler), geht der Slave von einem Masterausfall aus (Watchdog-Funktion).

In diesem Fall setzt der Slave seine Ausgänge in den Fehlerzustand, sendet ein Emergency-Telegramm und wechselt in den Zustand Pre-Operational zurück. Das Verfahren kann nach einem Guarding Time-Out wieder durch eine neue Guarding-Nachricht angeregt werden.

Die Node Life Time berechnet sich aus den Parametern:

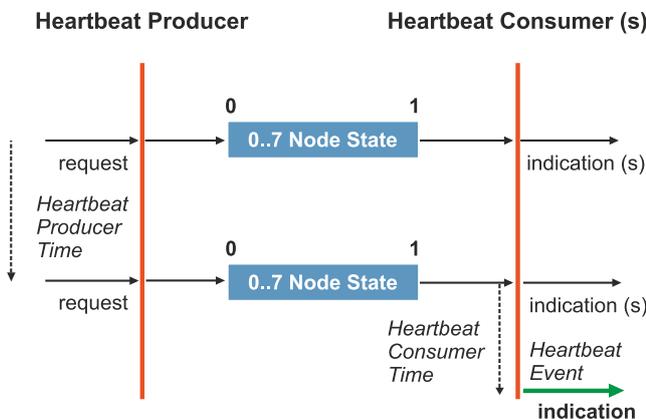
- Guard-Time (Objekt 0x100C) und
- Life-Time-Factor (Objekt 0x100D)

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

Falls einer der beiden Parameter 0 ist (Standardeinstellung), erfolgt keine Überwachung des Masters (kein Life-Guarding) (siehe: Registerkarte [CAN Node \[► 36\]](#)).

**Heartbeat: Knotenüberwachung ohne Remote Frame**

Beim Heartbeat-Verfahren senden die Teilnehmer selbsttätig zyklisch ihre jeweilige Statusmeldung. Anders als beim Guarding-Verfahren werden beim Heartbeat-Verfahren keine Remote Frames verschickt und es wird weniger Buslast erzeugt. Auch der Master sendet sein Heartbeat-Telegramm zyklisch und die Slaves können dadurch den Ausfall des Masters erkennen (siehe: Registerkarte [CAN Node \[► 36\]](#)).



Beim Heartbeat-Verfahren wird auf das Toggle-bit verzichtet und die Teilnehmer senden zyklisch Ihren Status.

### 2.1.3 Prozessdatenobjekte (PDO)

Bei vielen Feldbussystemen wird ständig das gesamte Prozessabbild übertragen - meist mehr oder weniger zyklisch. CANopen ist nicht auf dieses Kommunikationsprinzip beschränkt, da bei CANopen die Prozessdaten nach dem Produzenten/Konsumenten-Modell übertragen werden. Dabei sendet ein Teilnehmer seine Prozessdaten von sich aus (Produzent), alle anderen Teilnehmer hören mit und entscheiden anhand eines CAN-Identifizier, ob sie sich für dieses Telegramm interessieren und verarbeiten es entsprechend (Konsument).

Bei CANopen werden die Prozessdaten in Segmente zu maximal 8 Byte aufgeteilt. Diese Segmente heißen Prozessdatenobjekte (PDOs). Die Prozessdatenobjekte (PDOs) entsprechen jeweils einem CAN-Telegramm und werden durch einen spezifischen CAN-Identifizier zugeordnet und in ihrer Priorität bestimmt.

Die Prozessdatenobjekte (PDOs) werden in Empfangs-PDOs (RxPDOs) und Sende-PDOs (TxPDOs) unterteilt, wobei die Bezeichnung jeweils aus Sicht der Teilnehmer erfolgt. Ein Teilnehmer sendet seine Eingangsdaten mit TxPDOs, und empfängt die Ausgangsdaten in den RxPDOs. Diese Bezeichnung wird in TwinCAT beibehalten.

#### Kommunikationsparameter

Die Prozessdatenobjekte (PDOs) können je nach Anforderung mit unterschiedlichen Kommunikationsparametern versehen werden. Wie alle CANopen-Parameter stehen auch diese im Objektverzeichnis des Gerätes. Auf die Kommunikationsparameter kann über Servicedatenobjekte (SDOs) zugegriffen werden.

Die Parameter für die Empfangs-PDOs stehen bei Index 0x1400 (RxPDO1) bis Index 0x15FF (RxPDO512), damit können bis zu 512 RxPDOs vorhanden sein. Entsprechend finden sich die Einträge für die Sende-PDOs bei Index 0x1800 (TxPDO1) bis Index 0x19FF (TxPDO512).

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

#### CAN-Identifizier

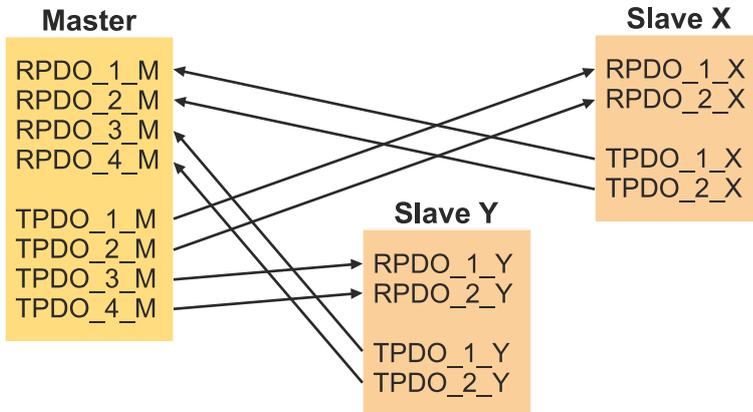
Der wichtigste Kommunikationsparameter für Prozessdatenobjekte (PDOs) ist der CAN-Identifizier (auch Communication Object Identifier, COB-ID genannt). Der CAN-Identifizier dient zur Identifizierung der CAN-Telegramme und bestimmt deren Priorität beim Buszugriff. Für jedes CAN-Telegramm darf es nur einen Sender (Produzenten) geben, da CAN jedoch alle Nachrichten im Broadcast-Verfahren senden kann ein Telegramm wie beschrieben von beliebig vielen Teilnehmern empfangen werden (Konsumenten). Ein Teilnehmer kann also seine Eingangsinformation mehreren Teilnehmern gleichzeitig zur Verfügung stellen - auch ohne Weiterleitung durch einen logischen Master.

Der CAN-Identifizier steht in Subindex 1 des Kommunikationsparametersatzes. Er ist als 32-Bit Wert kodiert, wobei die niederwertigsten 11 Bits (Bit 0...10) den eigentlichen CAN-Identifizier enthalten. Die Datenbreite des Objektes von 32 Bit erlaubt auch den Eintrag von 29 Bit CAN-Identifizieren. Der Einsatz der 29Bit-Variante bleibt auf Sonderanwendungen beschränkt - und wird daher auch von den Beckhoff CANopen-Geräten nicht unterstützt. Über das höchstwertige Bit (Bit 31) lässt sich das Prozessdatenobjekt aktivieren bzw. abschalten.

**PDO-Linking**

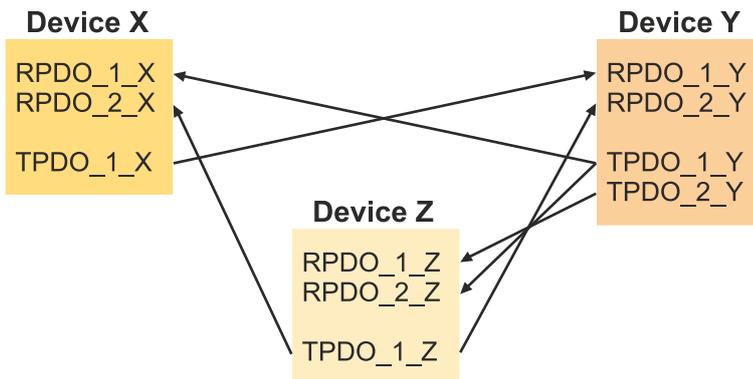
Standardmäßig kommunizieren alle Teilnehmer (Slaves) mit einer Zentrale (Master). Kein Slave hört wegen seinen Voreinstellungen auf die Prozessdatenobjekte (PDOs) und damit auf die CAN-Identifizier eines anderen Slaves.

Standardkommunikation zwischen mehreren Slaves und einem Master:



Sollen die Prozessdatenobjekte (SDOs) direkt zwischen den Teilnehmern ohne Master ausgetauscht werden, so müssen die CAN-Identifizier entsprechend angepasst werden. Die TxPDO-Identifizier des Produzenten müssen mit dem RxPDO-Identifizier des Konsumenten übereinstimmen. 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.

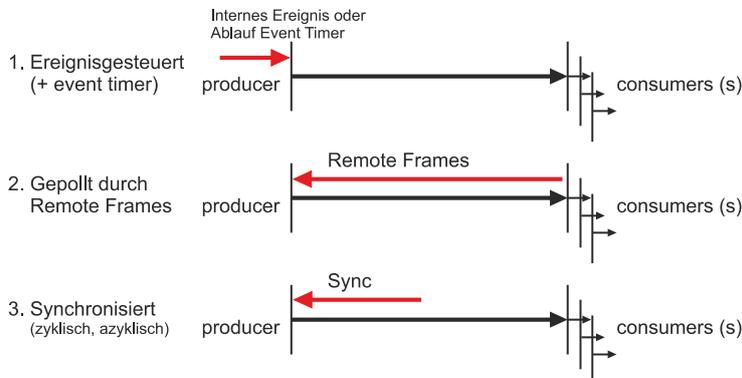
Kommunikation ohne Master mit PDO-Linking:



## PDO-Übertragungsarten

CANopen bietet folgende Möglichkeiten die Prozessdatenobjekte (PDOs) zu übertragen:

- Ereignisgesteuert
- Durch Polling
- Synchronisiert



- **Ereignisgesteuert:**

Ändert sich ein Eingangswert, überträgt ein Teilnehmer sofort seine Prozessdatenobjekte (PDOs). Dadurch wird die Übertragungsbandbreite optimal ausgenutzt, da nur die Änderung des Prozessabbildes übertragen wird. Gleichzeitig ist die Reaktionszeit kurz, die Teilnehmer warten bei einer Änderung der Eingangswerte nicht auf die Abfrage durch einen Master.

Ab CANopen Version 4 kann die ereignisgesteuerte Kommunikationsart mit einem zyklischen Update kombiniert werden. Dabei werden die Prozessdatenobjekte (in diesem Fall sind es die TxPDOs) dann übertragen, wenn eine zuvor eingestellte Zeit (Event Timer) abgelaufen ist. Ändert sich ein Eingangswert innerhalb der eingestellten Zeit, wird die Zeit (Event Timer) zurückgesetzt.

Bei RxPDOs wird die eingestellte Zeit (Event Timer) dazu benutzt das Eintreffen der ereignisgesteuerten Prozessdatenobjekte (PDOs) zu überwachen. Ein Teilnehmer geht in den Fehlerzustand, wenn innerhalb der eingestellten Zeit keine Prozessdatenobjekte (PDOs) eintreffen.

- **Gepollt:**

Die Prozessdatenobjekte (PDOs) werden durch Anforderungstelegramme (Remote Frames) gepollt. Auf diese Weise werden die Prozessdatenobjekte (PDOs) auch ohne deren Änderung übertragen, beispielsweise bei einem zur Laufzeit ins Netz aufgenommenen Monitor oder Diagnosegerät.

Bausteine mit integrierter kompletter Nachrichtenfilterung (FullCAN) beantworten ein Anforderungstelegramm 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, wird die gepollte Kommunikationsart nur bedingt für den laufenden Betrieb empfohlen.

- **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 Teilnehmern als Trigger für das Lesen der Eingänge bzw. für das Setzen der Ausgänge verwendet wird.

**PDO-Übertragungsarten: Parametrierung**

Die PDO-Übertragungsart (Transmission Type) legt fest, wie die Übertragung der Prozessdatenobjekte (PDOs) ausgelöst wird bzw. wie empfangene Prozessdatenobjekte (PDOs) behandelt werden:

Übertragungsart	Zyklisch	Azyklisch	Synchron	Asynchron
0		X	X	
1-240	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. In TwinCAT wird die PDO-Übertragungsart auf der Registerkarte PDO unter Transmission Type eingestellt (siehe: Registerkarte [PDO \[▶ 40\]](#)).

- **Azyklisch Synchron:**

PDOs der Übertragungsart 0 arbeiten synchron, aber nicht zyklisch. Ein RxPDO wird erst nach dem nächsten SYNC-Telegramm 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.

Im Vergleich dazu ermittelt ein TxPDO seine Eingangsdaten nach einem SYNC-Telegramm (synchrones Prozessabbild) und sendet seine Eingangsdaten dann weiter, wenn sich die Eingangsdaten geändert haben.

Die Übertragungsart 0 kombiniert also den Sendegrund „Synchronisiert“ mit dem Sendegrund „Ereignisgesteuert“.

- **Zyklisch Synchron:**

Bei Übertragungsart 1-240 wird das PDO zyklisch nach jedem "n-ten" SYNC gesendet (n=1...240). Auf diese Weise kann beispielsweise ein schneller Zyklus für digitale Eingänge parametriert 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-Telegramm 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.

Das SYNC-Telegramm ist mit der verknüpften Task gekoppelt, sodass zu jedem Taskbeginn neue Eingangsdaten zur Verfügung stehen. Bleibt ein synchrones PDO aus, wird es erkannt und an die Applikation gemeldet.

- **Asynchron:**

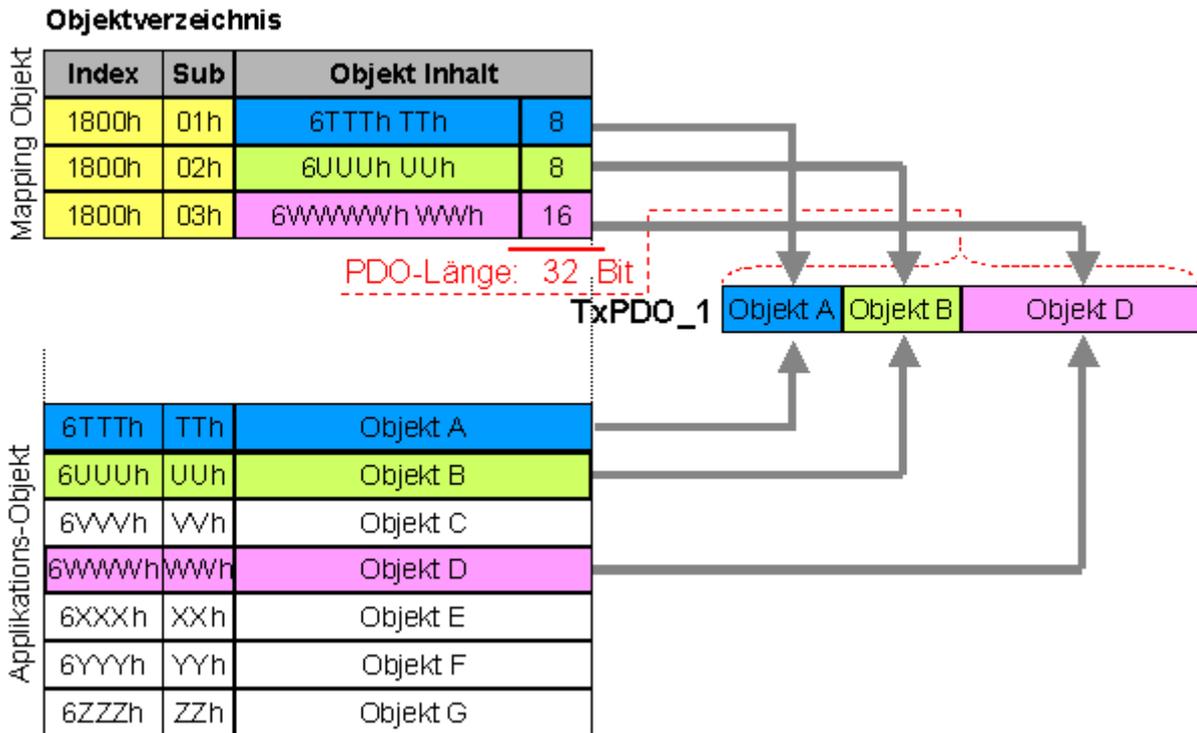
Die Übertragungsarten 254 + 255 sind asynchron oder ereignisgesteuert. Bei Übertragungsart 254 ist das Ereignis herstellerepezifisch, bei 255 im Geräteprofil definiert. Im einfachsten Fall ist das Ereignis die Veränderung eines Eingangswertes - es wird also jede Werteänderung übertragen. Die Asynchrone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

**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.



**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**

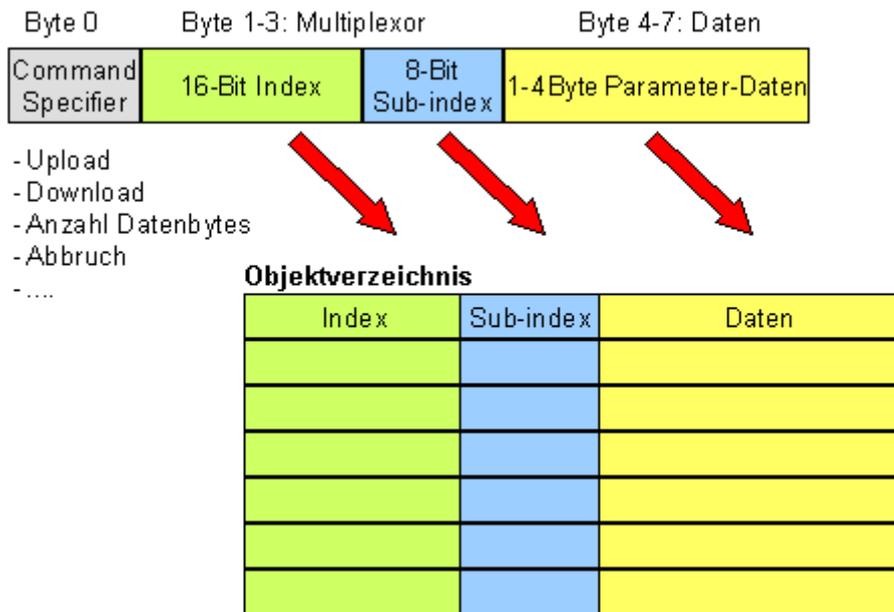
Ein weiteres Feature von CANopen ist das Mappen von Platzhaltern (Dummy-Einträgen). Als Platzhalter dienen die im Objektverzeichnis hinterlegten Datentyp-Einträge, die ja selbst nicht mit Daten versehen sind. Sind solche Einträge in der Mapping-Tabelle enthalten, so werden die entsprechenden Daten vom 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.

**Sehen Sie dazu auch**

📄 PDO-Parametrierung [▶ 41]

### 2.1.4 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.



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 (=1536dez) + 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 (=1408dez) + Node-ID	0x4x	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)

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 (=1536dez) + 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 22h im ersten CAN-Datenbyte erfolgen.

**Client -> Server, Download Response**

11-bit Identifier	8 Byte Nutzdaten							
0x580 (=1408dez) + 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)
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 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

## 2.2 Technische Daten - CANopen

### Optionsschnittstelle-M510

Technische Daten	M510
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
Eigenschaften	CANopen – unterstützte PD-Kommunikationsarten: ereignisgesteuert, zeitgesteuert, synchron, polling; Emergency-Message-Handling, Guarding und Heartbeat, Boot-Up nach DS302; Online-Bus-Load-Monitor und Bus-Trace, Error-Management für jeden Teilnehmer frei konfigurierbar

### Optionsschnittstelle-B510

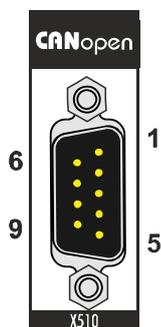
Technische Daten	B510
Feldbus	CANopen
Übertragungsrate	10, 20, 50, 100, 125, 250, 500, 800, 1.000 kBaud
Businterface	1 x D-Sub-Buchse, 9-polig
Erweiterbares Prozessabbild	bis zu 15 virtuelle Slaves zusätzlich
max. Prozessabbild	16 Slaves x (16 Tx PDOs / 16 Rx PDOs (8 Byte pro PDO))

### 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.

## 3 Anschluss und Verkabelung

### 3.1 CANopen-Anschluss

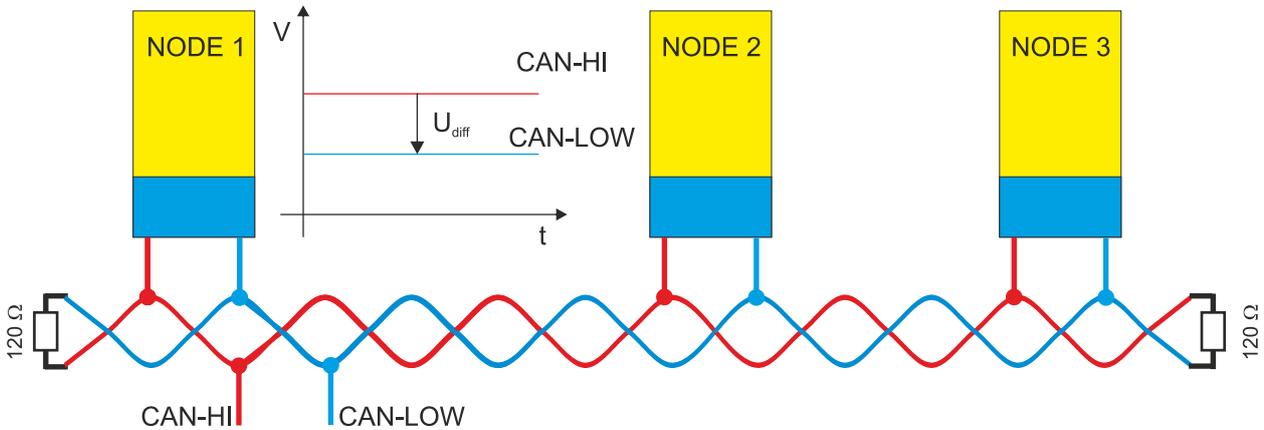


Die CAN Busleitung wird über eine 9polige D-sub-Buchse mit folgender Belegung angeschlossen:

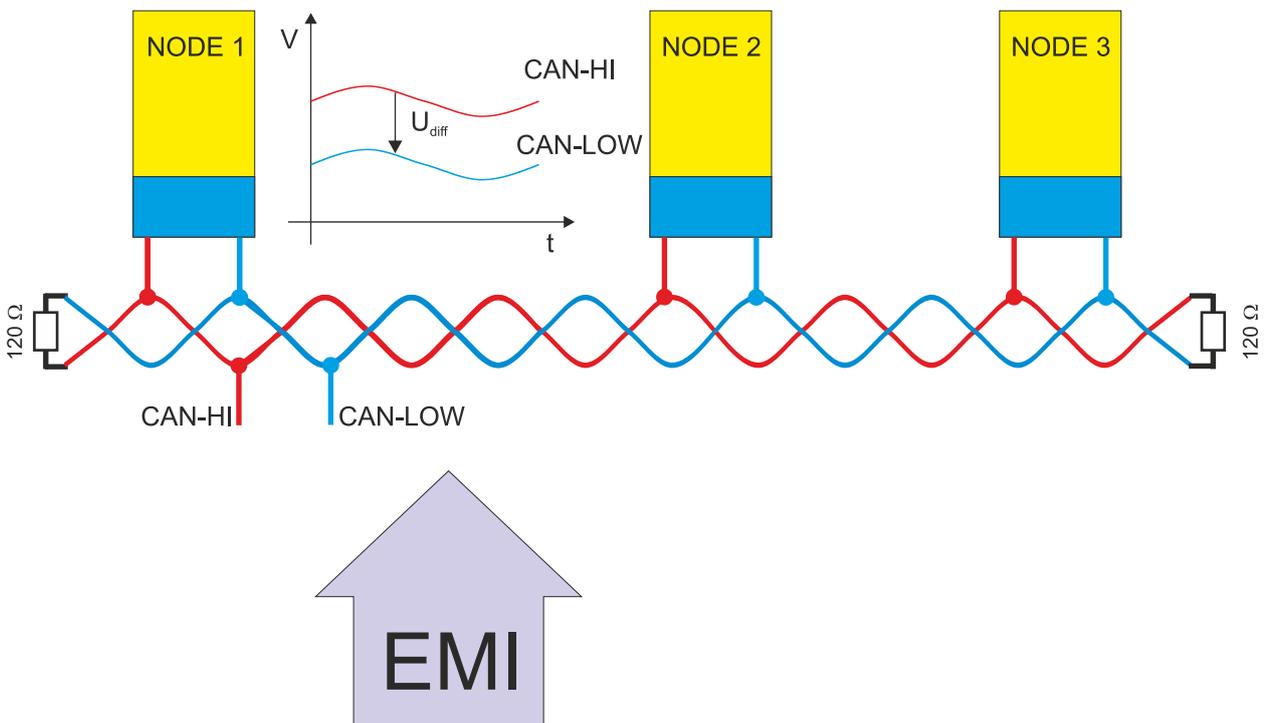
Pin	Belegung
1	nicht benutzt
2	CAN low (CAN-)
3	CAN Ground (intern verbunden mit Pin 6)
4	nicht benutzt
5	Schirm
6	CAN Ground (intern verbunden mit Pin 3)
7	CAN high (CAN+)
8	nicht benutzt
9	nicht benutzt

### 3.2 Verkabelung

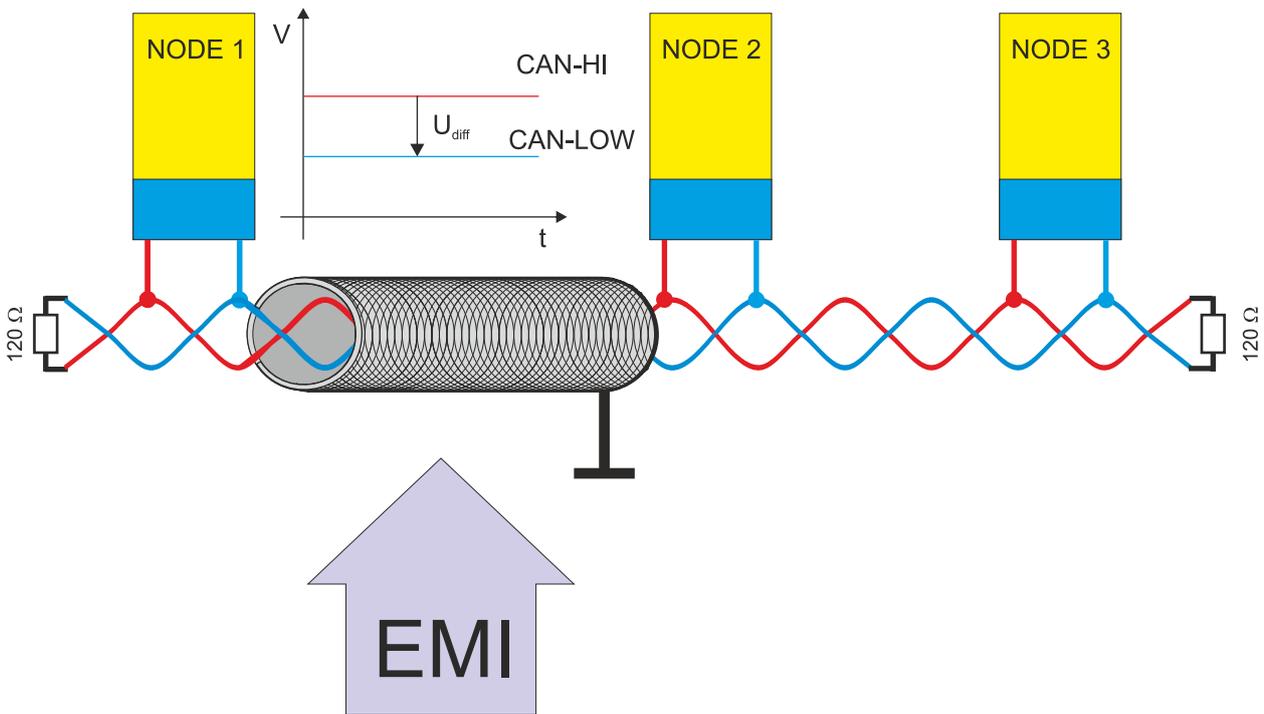
CAN ist ein 2-Draht-Bussystem, an dem alle Teilnehmer parallel (d.h. mit kurzen Stichleitungen) angeschlossen werden. Der Bus muss an jedem Ende mit einem Abschlusswiderstand von 120 (bzw. 121) Ohm abgeschlossen werden, um Reflexionen zu vermeiden. Dies ist auch bei sehr kurzen Leitungslängen erforderlich!



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 verdrehten Leitungsadern können störende Einflüsse durch EMI weiter eliminiert werden.



### Buslänge

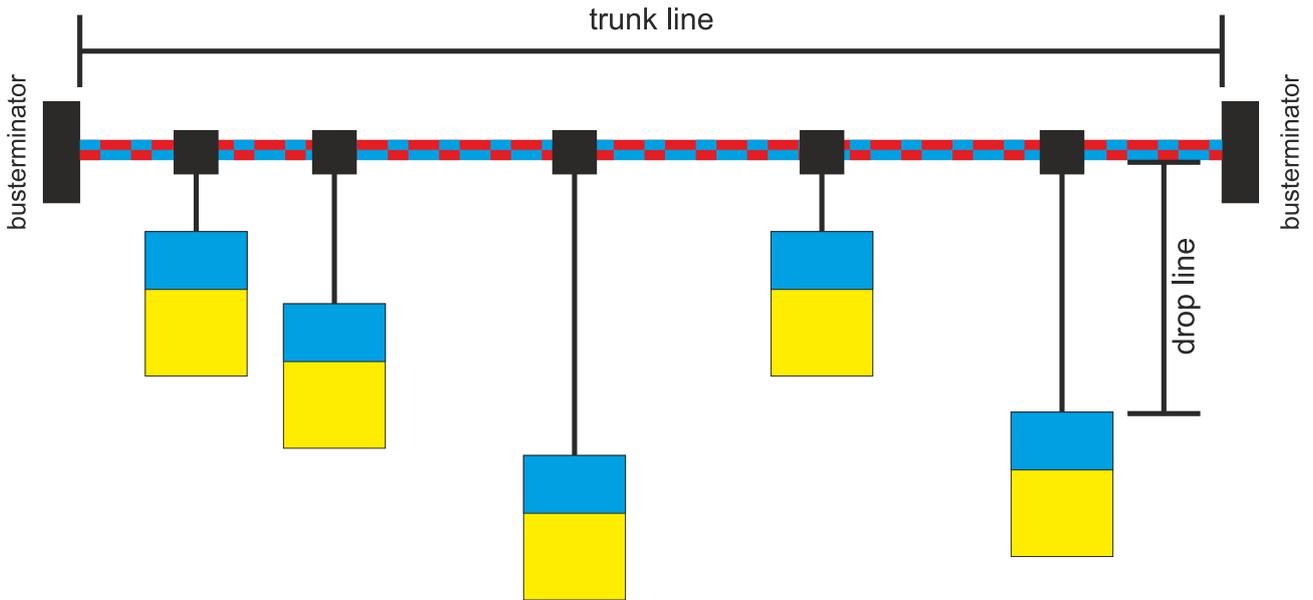
Die maximale Buslänge wird bei CAN vorwiegend durch die Signallaufzeit beschränkt. Das Multi-Master-Buszugriffsverfahren (Arbitrierung) erfordert, dass die Signale quasi gleichzeitig (vor der Abtastung innerhalb einer Bitzeit) an allen Knoten anliegen. Da die Signallaufzeit in den CAN-Anschaltungen (Transceiver, 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**

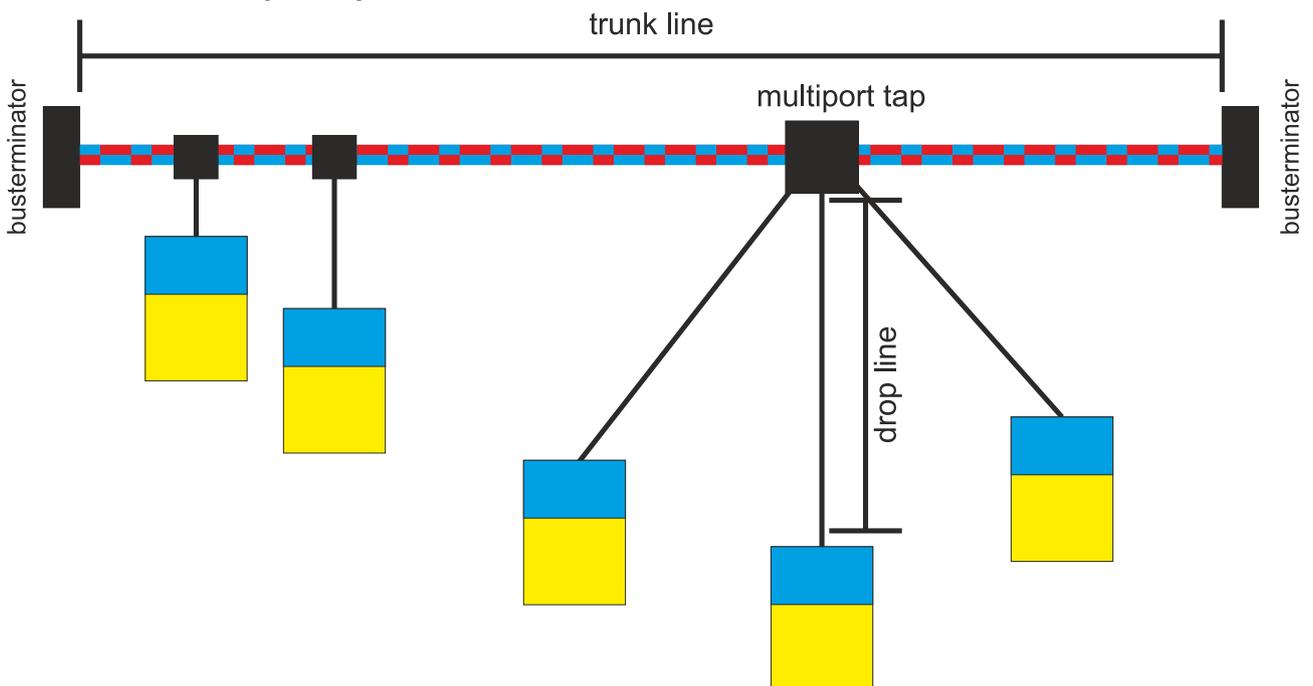


Stichleitungen ("drop lines") sind nach Möglichkeit zu vermeiden, da sie grundsätzlich zu Signalreflexionen führen. Die durch Stichleitungen hervorgerufenen Reflexionen sind jedoch in der Regel unkritisch, wenn sie vor dem Abtastzeitpunkt vollständig abgeklungen sind. Bei den in den Buskopplern gewählten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

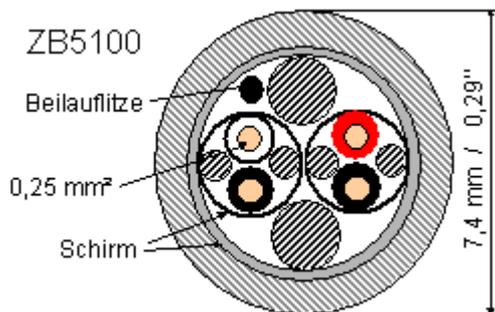
### CAN-Kabel

Für die CAN-Verdrahtung wird die Verwendung von paarig verdrehten, 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 mm<sup>2</sup> (AWG 24) paarig verseilt, Kabelfarben: rot/schwarz + weiß/schwarz
- doppelt geschirmt
- Schirmgeflecht mit Beilaufitze (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)

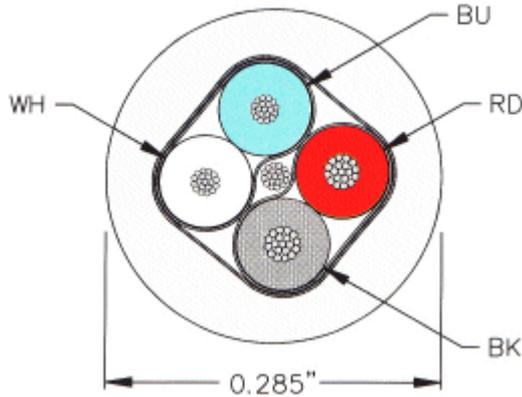


### ZB5200 CAN/DeviceNet-Kabel

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 mm<sup>2</sup> (AWG 22) paarig verseilt
- doppelt geschirmt · Schirmgeflecht mit Beilaufitze
- 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**

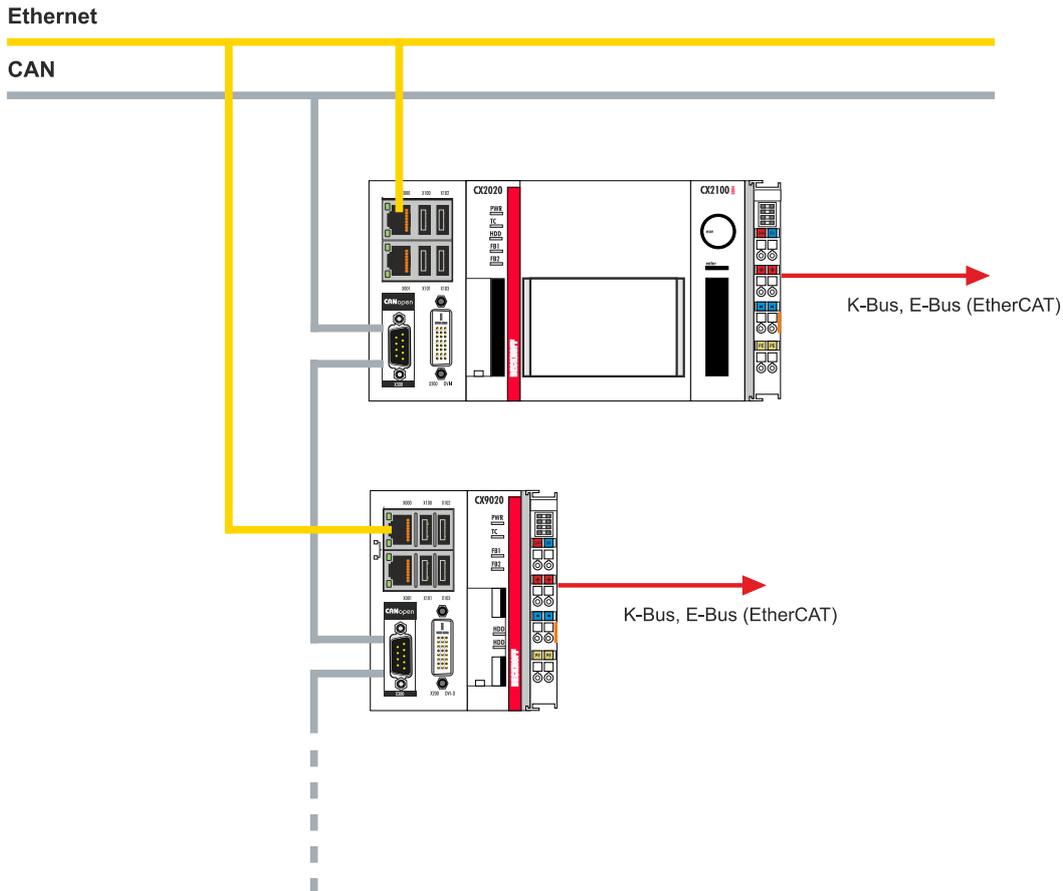
Vorschlag für die Verwendung der Beckhoff CAN-Kabel:

Funktion	Kabelfarbe ZB5100	Kabelfarbe ZB5200
CAN Ground	<b>schwarz</b> /(rot)	<b>schwarz</b>
CAN Low	<b>schwarz</b>	<b>blau</b>
Schirm	Beilaufnitze	Beilaufnitze
CAN high	<b>weiß</b>	<b>weiß</b>
nicht benutzt	<b>(rot)</b>	<b>(rot)</b>

### 3.3 Topologie

Alle Teilnehmer werden parallel angeschlossen. Der Bus muss an jedem Ende mit einem Abschlusswiderstand von 120 Ohm abgeschlossen werden. Die Anzahl der Teilnehmer pro Netz ist dabei von CANopen auf 64 Teilnehmer begrenzt.

Die maximal mögliche Netzausdehnung wird durch die Datenrate begrenzt. Beispielsweise ist bei 1 MBit/s eine Netzausdehnung von 20 m und bei 50 kBit/s eine Netzausdehnung von 1000 m möglich. Bei niedrigen Datenraten kann die Netzausdehnung mit Repeatern erhöht werden. Die Repeater ermöglichen auch den Aufbau von Baumstrukturen.



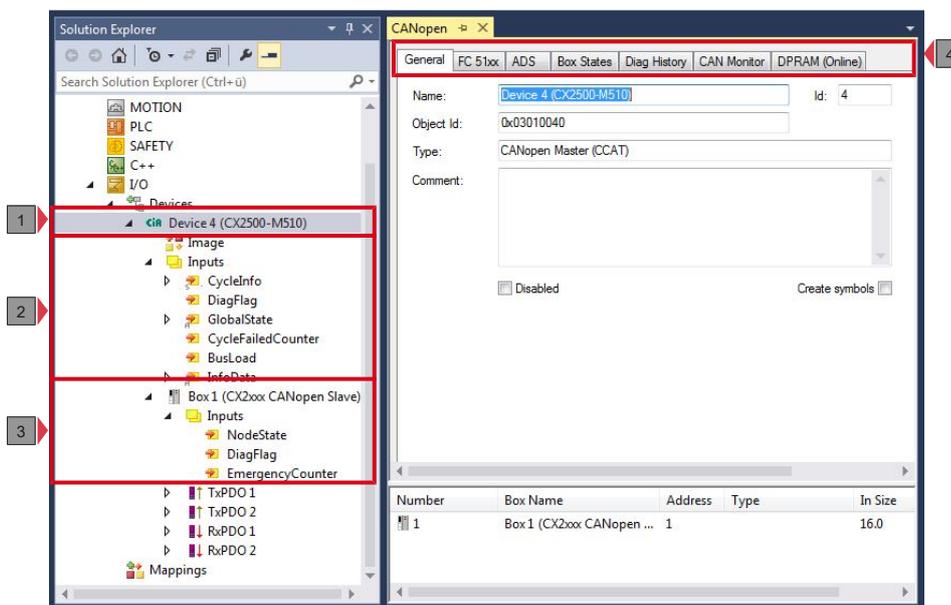
## 4 TwinCAT Registerkarten

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.

Die Strukturansicht und die Registerkarten für eine CANopen-Schnittstelle sind unter TwinCAT2 und TwinCAT3 identisch.

### 4.1 Strukturansicht

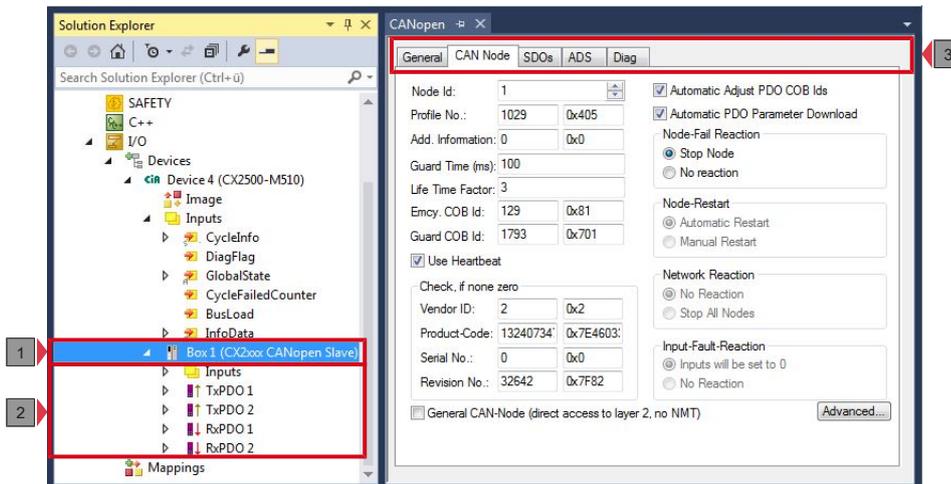
Ein CANopen-Master und ein CANopen-Slave werden wie folgt in der Strukturansicht angezeigt:



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 angefügt.

Nr.	Beschreibung
1	Der Gerätenamen 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 Gerätenamen 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:



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: <ul style="list-style-type: none"> <li>RxPDOs werden vom Teilnehmer empfangen.</li> <li>TxPDOs werden vom Teilnehmer gesendet.</li> </ul>
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.

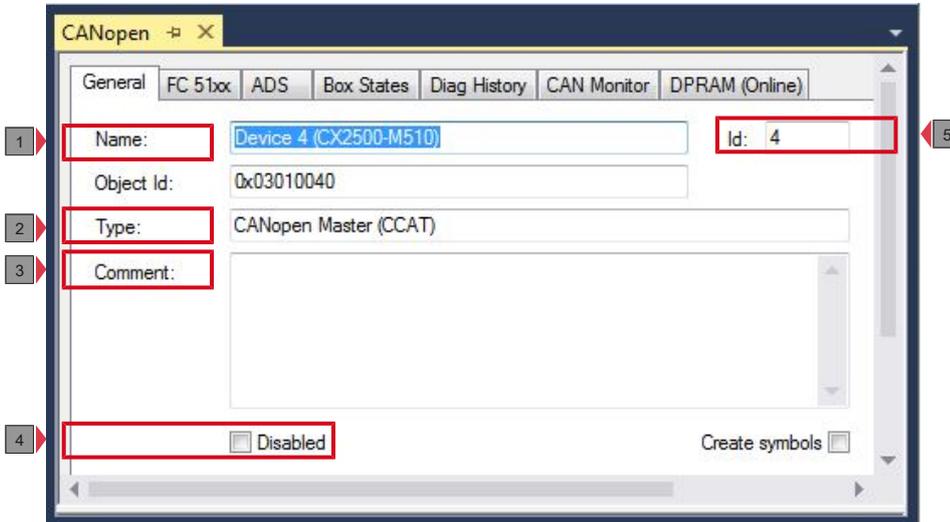
Wird das SPS-Prozessabbild eingelesen, können die Variablen für Statusmeldungen und die Variablen unter den Prozessdatenobjekten mit den Variablen aus dem SPS-Programm verknüpft werden. Mit einem Doppelklick auf den Variablennamen in der Strukturansicht wird der Verknüpfungsdialo geöffnet. Die verknüpften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

Weitere Informationen zu TwinCAT finden Sie in der TwinCAT Dokumentation auf der Beckhoff Homepage: [www.beckhoff.de](http://www.beckhoff.de)

## 4.2 CANopen-Master

### 4.2.1 General

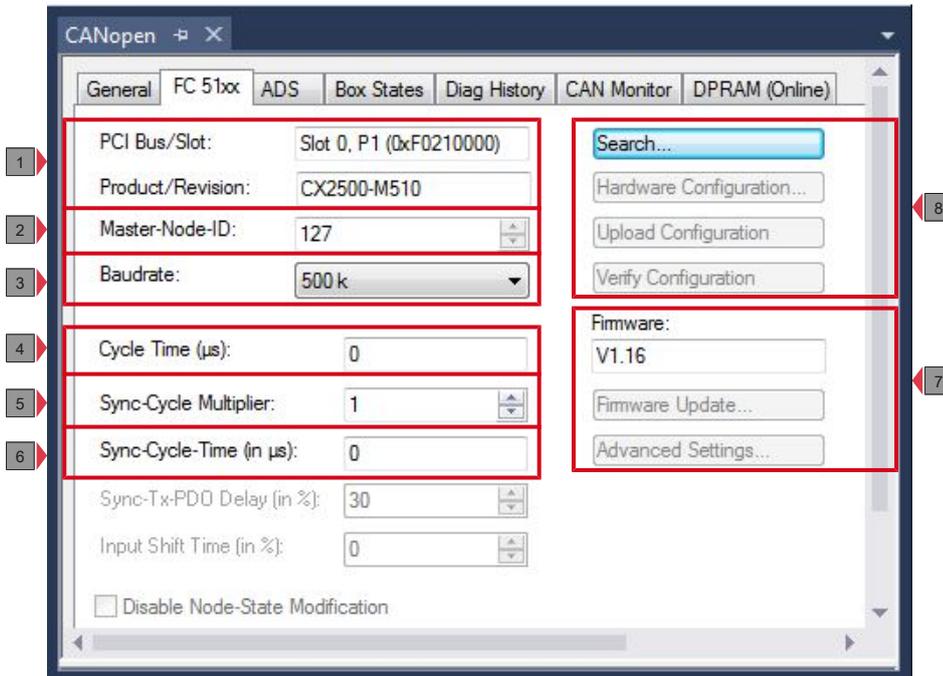
Die Registerkarte General liefert allgemeine Informationen zu einem CANopen-Gerät, wie Name, Typ und die Id.



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.

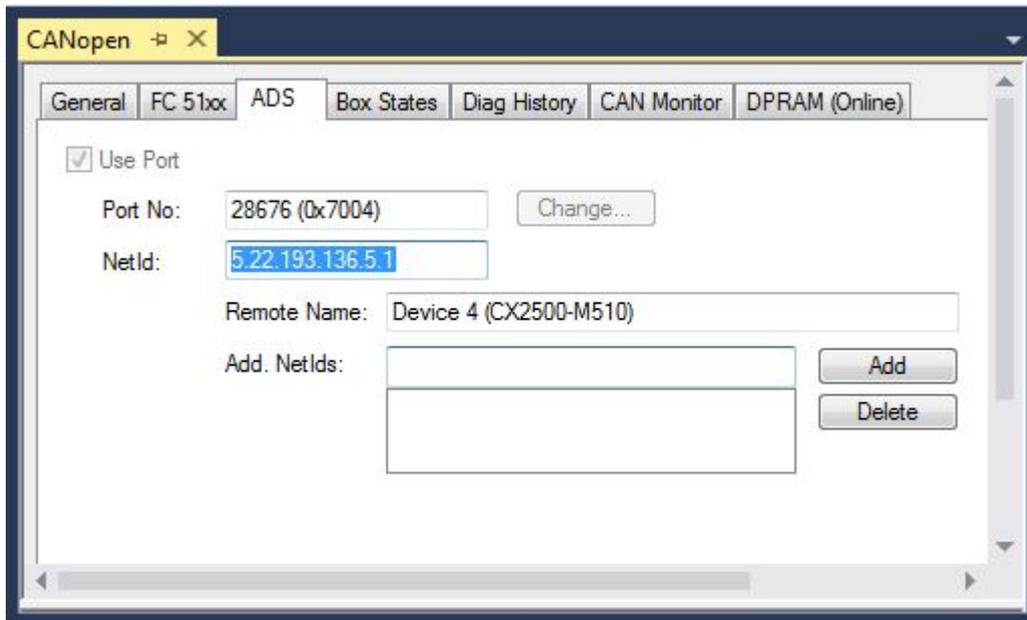
Über diese Registerkarte kann das CANopen-Gerät ausgeschaltet werden. Ein Kommentarfeld bietet die Möglichkeit einer Beschriftung, um auf diese Weise zusätzliche Informationen zu dem Gerät bereitzustellen.

## 4.2.2 FC 51xx



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. 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 höchstpriorigen 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 höchstpriorigen Task, deren Prozessdaten und aus dem Sync-Cycle Multiplier. Sync-Cycle-Time = Cycle Time x Sync-Cycle Multiplier
7	An dieser Stelle wird die aktuelle Firmware angezeigt.
8	Über die Schaltfläche <b>Search</b> wird nach der physikalischen Schnittstelle gesucht und die gewünschte ausgewählt, wenn nicht bereits automatisch geschehen.

### 4.2.3 ADS



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-Nr. adressieren.

## 4.3 CANopen-Slave

### 4.3.1 CAN Node

The screenshot shows the 'CANopen' configuration dialog box with the 'CAN Node' tab selected. The dialog is divided into several sections, with red boxes highlighting specific areas and numbered callouts (1-13) pointing to various elements.

**General Section:**

- 1 Node Id: 55
- 2 Profile No.: 1029 0x405
- 3 Add. Information: 0 0x0
- 4 Guard Time (ms): 100
- 5 Life Time Factor: 3
- 6 Emcy. COB Id: 183 0xB7
- 6 Guard COB Id: 1847 0x737
- 7  Use Heartbeat
- 8  Check, if none zero
- Vendor ID: 2 0x2
- Product-Code: 13240734 0x7E4603
- Serial No.: 0 0x0
- Revision No.: 32642 0x7F82
- 9  General CAN-Node (direct access to layer 2, no NMT)

**Advanced Section:**

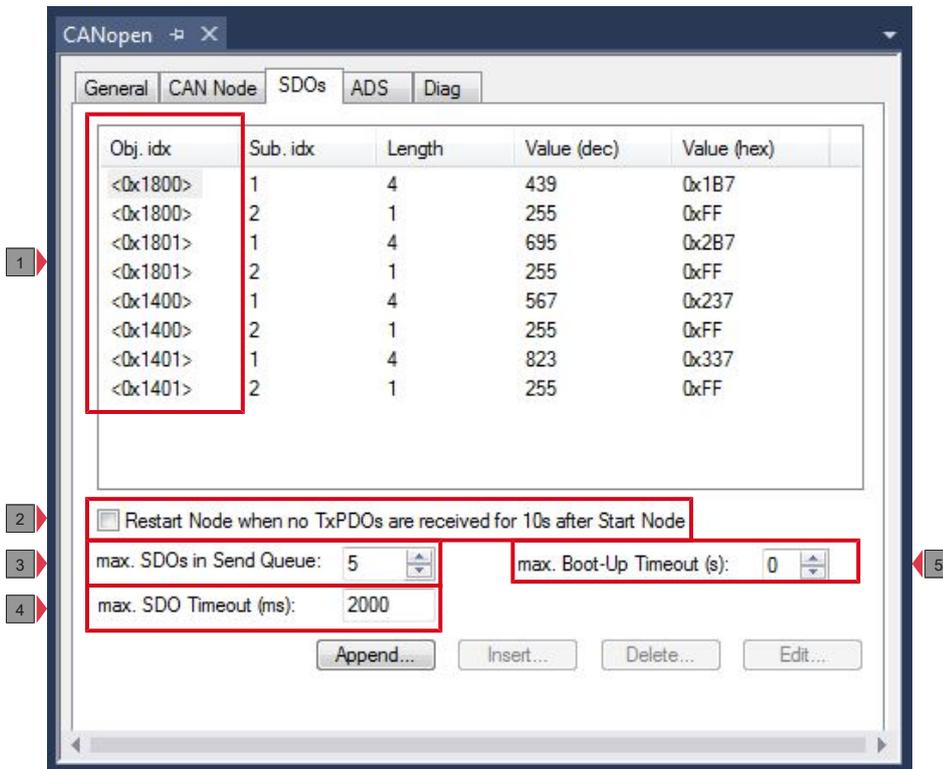
- 13  Automatic Adjust PDO COB Ids
- 12  Automatic PDO Parameter Download
- 11 **Node-Fail Reaction:**
  - Stop Node
  - No reaction
- Node-Restart:**
  - Automatic Restart
  - Manual Restart
- Network Reaction:**
  - No Reaction
  - Stop All Nodes
- Input-Fault-Reaction:**
  - Inputs will be set to 0
  - No Reaction
- 10

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.
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: <ul style="list-style-type: none"> <li>• Upload Objekt 0x1000 ausschalten.</li> <li>• Download Objekt 0x1006 ausschalten.</li> <li>• Automatisches Senden von Start Node ausschalten (muss dann manuell gesendet werden.)</li> <li>• Start SDOs weiter senden, im Falle eines Abbruchs.</li> </ul>

Nr.	Beschreibung
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 <u>SDOs</u> [ <a href="#">▶ 39</a> ]).
13	Wenn Option angewählt ist, werden die Default-Identifizier der Prozessdatenobjekte bei Änderung der Node-ID entsprechend nachgeführt (siehe: Nr. 6).

### 4.3.2 SDOs

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



Nr.	Beschreibung
1	Objekt Index Einträge 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.

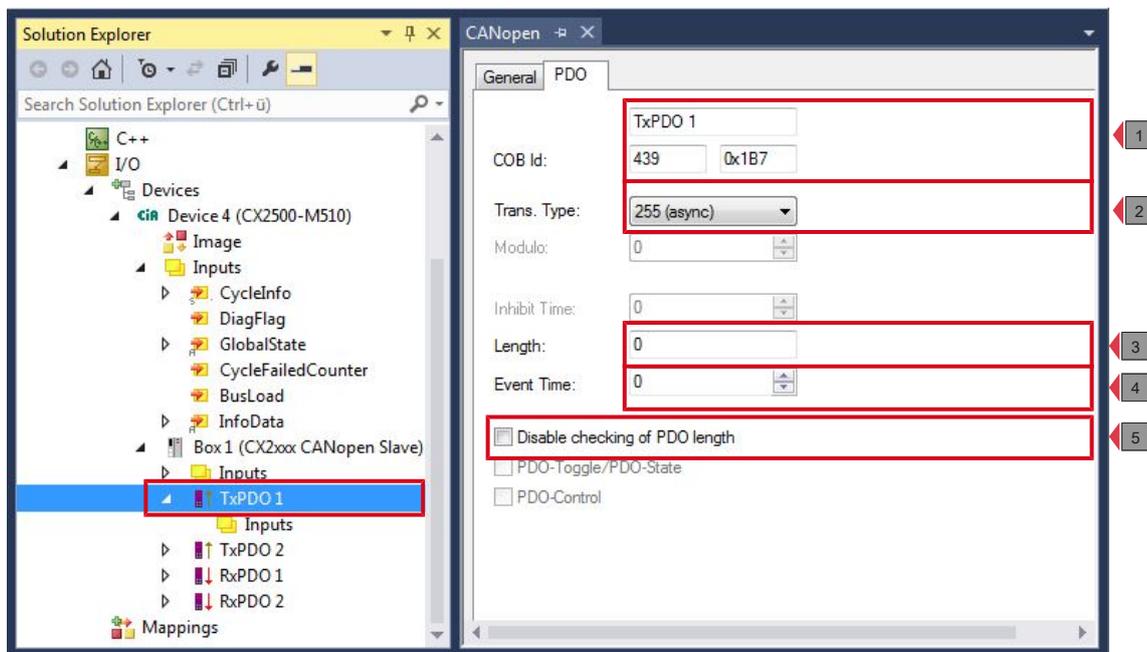
### 4.3.3 PDO

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.



Nr.	Beschreibung
1	CAN-Identifizier des PDOs. Für zwei Sende- und Empfangs-PDOs je Knoten stellt CANopen Default-Identifizier zur Verfügung. Diese können dann geändert werden.
2	Der Transmission Type bestimmt das Sendeverhalten des PDOs. 255 entspricht dem ereignisgesteuerten Senden (siehe: <a href="#">Übertragungsart festlegen [► 66]</a> ).
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: <a href="#">SDOs [► 39]</a> ).  Diese Funktion kann über das Kontrollkästchen <b>Automatic PDO Parameter Download</b> auf der Registerkarte <b>CAN Node</b> deaktiviert werden (siehe: <a href="#">CAN Node [► 36]</a> ).
5	Hier kann die Längenprüfung der PDOs deaktiviert werden.

## 5 Parametrieren und in Betrieb nehmen

In dieser Dokumentation werden exemplarisch CANopen-Geräte für die Inbetriebnahme verwendet. Die im Kapitel gezeigten Konfigurationsmöglichkeiten können damit für alle Embedded-PCs mit CANopen-Schnittstelle verwendet werden.

Folgende Geräte werden in dieser Dokumentation verwendet:

- CX2020-M510 (Embedded-PC mit Optionsschnittstelle CANopen-Master, D-Sub-Buchse, 9-polig)
- CX2500-B510 (Embedded-PC mit Feldbusmodul CX2500-B510 CANopen-Slave, D-Sub-Buchse, 9-polig)
- BK5100 (CANopen-Slave, D-Sub-Buchse, 9-polig)

Als Software für die Konfiguration der Geräte wird TwinCat2 und TwinCat3 verwendet.

Für weitere Informationen stehen die Dokumentationen für TwinCat2 und TwinCat3 auf der Beckhoff Homepage zur Verfügung:

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

### 5.1 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 bestimmen

Allgemein wird man beginnen, die Baud-Rate so groß zu wählen, wie es die Buslänge erlaubt. Hierbei sollte man berücksichtigen, dass serielle Bussysteme grundsätzlich umso 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 Schaltschranks ohne galvanische Trennung der Busknoten empfehlenswert. Die Erfahrung zeigt auch, dass das Abschätzen der verlegten Buskabelänge häufig zu optimistisch erfolgt - die tatsächliche Kabelänge 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 niederprioritäres PDO verschickt werden kann - eine seriöse Netzwerkplanung erfordert demnach eine worst-case Betrachtung. Auch muss, z.B. durch Verwendung der Inhibit Zeit, 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 und spätestens nach Verstreichen des Ablauf-Timers erneut gesendet.
- Parametriert wird die Kommunikationsart über den Transmission Type.

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). 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.

## 5.2 Mit TwinCAT 2 parametrieren

In diesem Kapitel wird beispielhaft gezeigt, wie Sie CANopen-Geräte mit Hilfe von TwinCAT 2 parametrieren können. Für das Beispiel werden insgesamt drei Geräte verwendet, darunter ein CANopen-Master an dem zwei CANopen-Slaves angeschlossen sind.

Zuerst wird gezeigt, wie Sie ein Zielsystem in TwinCAT suchen und auswählen können. Als nächstes wird ein CANopen-Slave in TwinCAT eingefügt, parametriert und die CANopen-Adresse des Slaves eingestellt. Danach wird ein kleines PLC-Projekt erstellt und in TwinCAT angefügt. Anschließend werden die Variablen aus dem PLC-Projekt mit der Hardware verknüpft und die fertige Konfiguration auf den CANopen-Slave geladen.

Als letzten Schritt wird der CANopen-Master in TwinCAT angefügt und nach den beiden CANopen-Slaves über den Master gesucht.

### 5.2.1 Zielsysteme suchen

Bevor Sie mit den Geräten arbeiten können, müssen Sie Ihren lokalen Rechner mit dem Zielgerät verbinden. Danach können Sie mit Hilfe der IP-Adresse oder dem Host Namen nach den Geräten suchen.

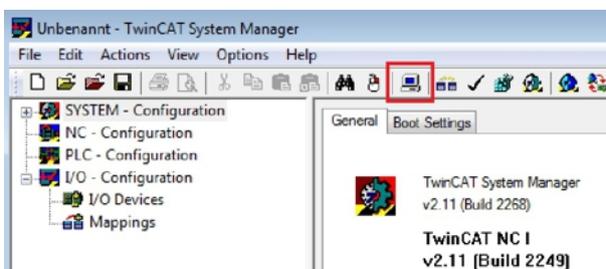
Der lokale PC und die Zielgeräte müssen mit dem gleichen Netzwerk oder direkt über ein Ethernet Kabel miteinander verbunden werden. In TwinCAT kann auf diese Weise nach allen Geräten gesucht und anschließend projektiert werden.

Voraussetzungen für diesen Arbeitsschritt:

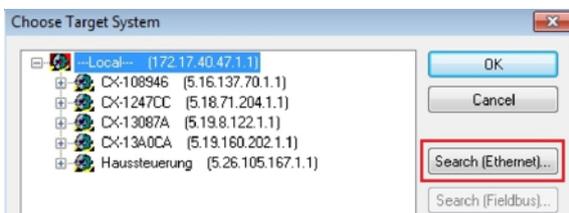
- TwinCAT 2 muss sich im Config Mode befinden.
- IP-Adresse oder Host Name des Gerätes. Der Hostname setzt sich aus CX- und den letzten 3 Byte der Mac Adresse zusammen. Die MAC Adresse befindet sich auf der Seite des Geräts.

Suchen Sie nach den Geräten wie folgt:

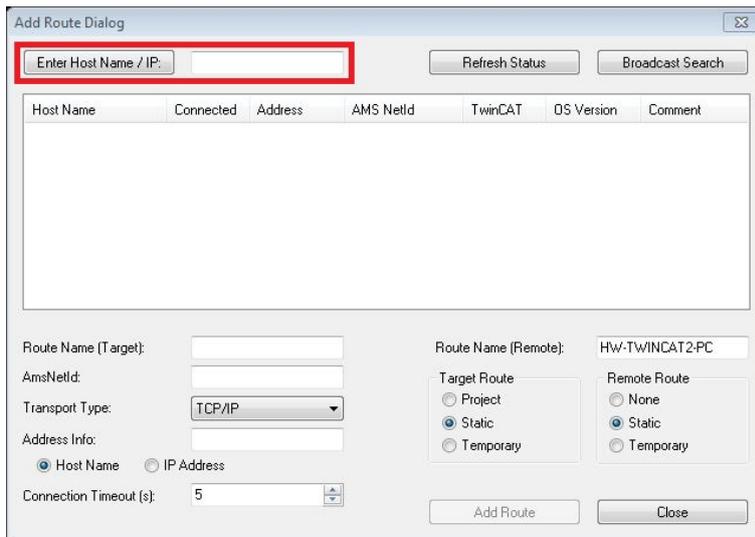
1. Klicken Sie oben im Menü auf **File > New**.
2. Klicken Sie oben in der Symbolleiste auf **Choose Target System**.



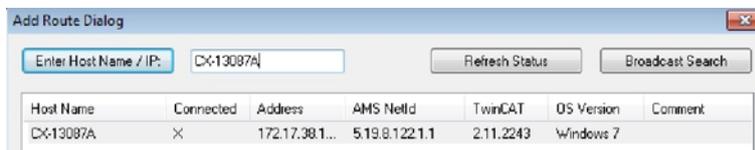
3. Klicken Sie auf **Search (Ethernet)**.



- Tippen Sie im Feld **Enter Host Name / IP** den Host Namen oder die IP-Adresse des Gerätes ein und drücken Sie **[Enter]**.



- Markieren Sie das gefundene Gerät und klicken Sie auf **Add Route**.



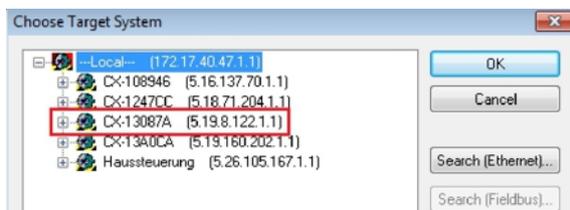
Das Fenster Logon Information erscheint.

- Geben Sie im Feld **User Name** und im Feld **Password** den Benutzernamen und das Passwort für den CX ein und klicken Sie auf **OK**.



Als Standard ist bei den CXen folgende Information eingestellt:  
**User name:** Administrator **Password:** 1

- Klicken Sie auf **Close**, wenn Sie keine weiteren Geräte suchen wollen und schließen damit das Add Route Fenster.  
 Das neue Gerät wird im Fenster Choose Target System angezeigt.
- Markieren Sie das das Gerät welches Sie als Zielsystem festlegen wollen und klicken Sie auf **OK**.



- ⇒ Sie haben erfolgreich in TwinCAT nach einem Gerät gesucht und das Gerät als Zielsystem eingefügt. In der Ecke unten rechts wird das neue Zielsystem mit Host Name und IP-Adresse (AMS Net ID) angezeigt.

**CX-13087A (5.19.8.122.1) Config Mode**

Mit dieser Vorgehensweise können Sie nach allen verfügbaren Geräten suchen und auch jederzeit zwischen den Zielsystemen wechseln. Als nächstes können Sie das Gerät in TwinCAT in die Strukturansicht anfügen.

## 5.2.2 CANopen-Slave anfügen

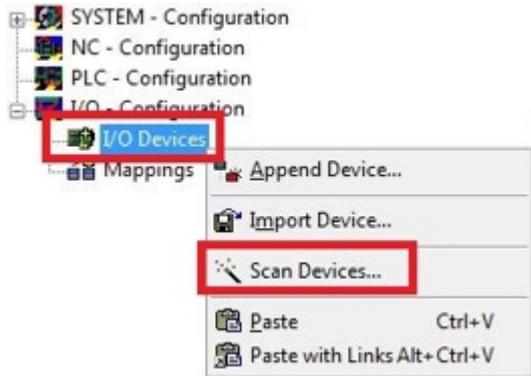
In dem exemplarischen Aufbau ist ein CANopen-Slave CX2020 mit Feldbusmodul CX2500-B510 am CANopen-Master angeschlossen. Damit der CANopen-Slave konfiguriert und später vom CANopen-Master mit allen Ein- und Ausgängen erkannt wird, muss der CANopen-Slave zuerst in TwinCAT angefügt werden.

Voraussetzungen für diesen Arbeitsschritt:

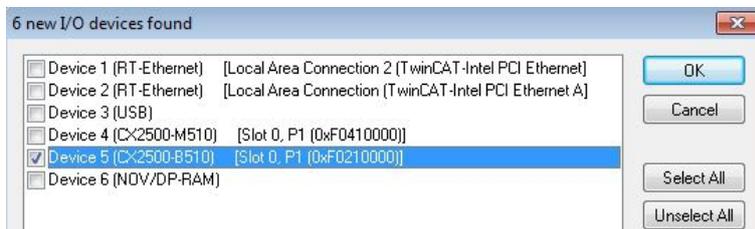
- Ein gescanntes und ausgewähltes Zielgerät mit CANopen-Slave. In diesem Beispiel ist es der CX2020 mit Feldbusmodul CX2500-B510.

**Fügen Sie den CANopen-Slave wie folgt ein:**

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

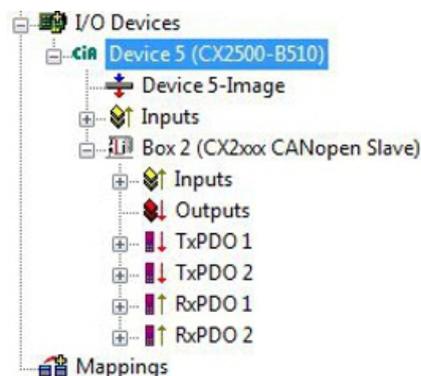


3. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.



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

⇒ 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.

### 5.2.3 Virtuellen Slave anlegen

Es können zusätzliche virtuelle Slaves auf der gleichen Hardware Schnittstelle angelegt werden. Dadurch können mehr Daten mit einem CANopen-Master ausgetauscht oder eine Verbindung mit einem zweiten CANopen-Master angelegt werden.

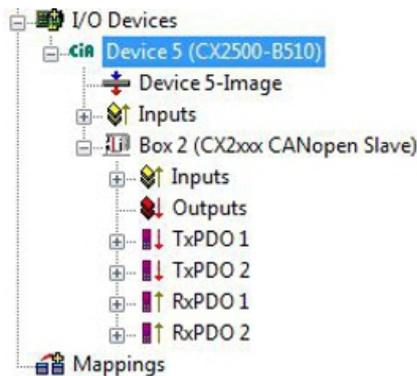
Jeder virtuelle Slave bekommt über TwinCAT eine eigene Adresse und wird für den CANopen-Master wie ein eigenständiges Gerät konfiguriert.

Voraussetzungen für diesen Arbeitsschritt:

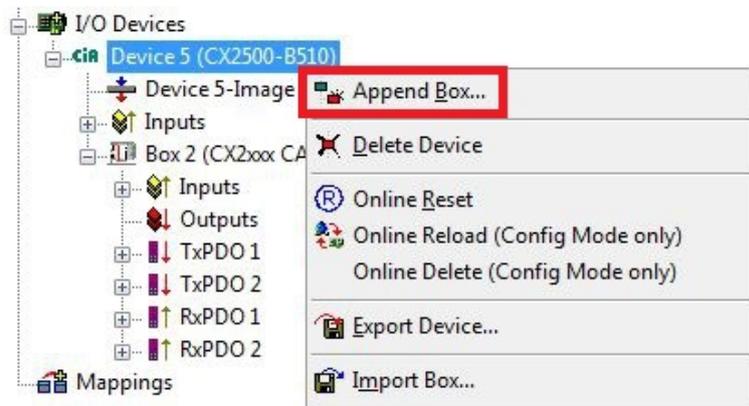
- Ein CANopen-Slave angelegt in TwinCAT.

**Legen Sie einen virtuellen Slave wie folgt ein:**

1. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf einen CANopen-Slave.



2. Klicken Sie im Kontextmenü auf **Append Box**.



⇒ 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.

## 5.2.4 Adresse einstellen

Nachdem der CANopen-Slave erfolgreich in TwinCAT angefügt wurde, kann die Adresse des CANopen-Slaves eingestellt werden. Geräte mit einem Dip-Schalter haben eine voreingestellte Adresse. Die Adresse auf dem Dip-Schalter muss mit der in TwinCAT eingestellten Adresse übereinstimmen.

Bei Geräten ohne Dip-Schalter wird die Adresse nur in TwinCAT eingestellt.

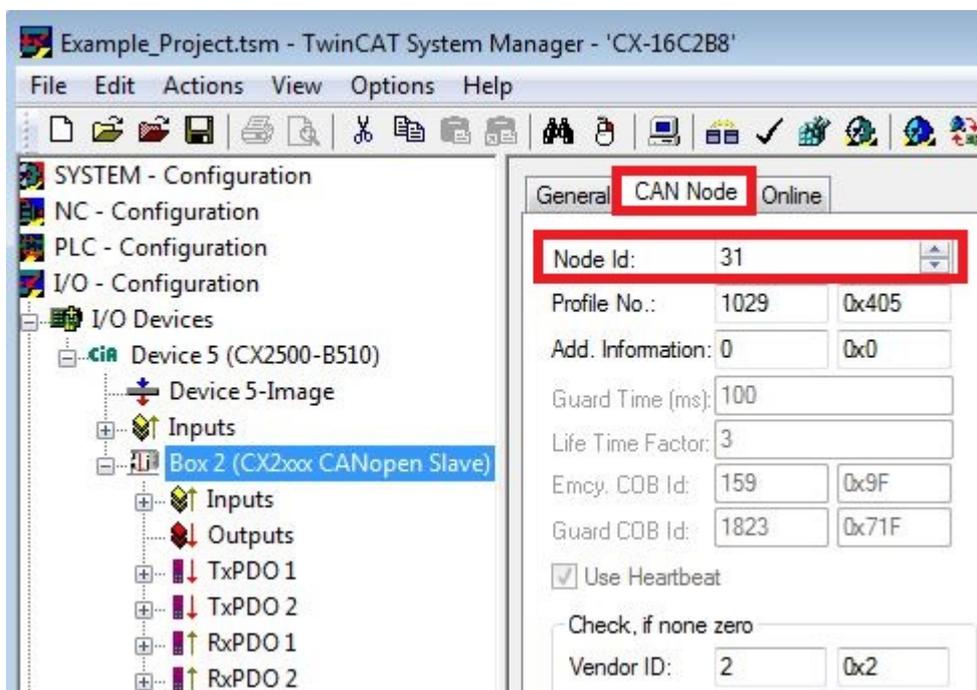
In diesem Arbeitsschritt wird die Adresse in TwinCAT eingestellt, damit der CANopen-Slave über diese Adresse für den CANopen-Master erreichbar ist.

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. „31“.



- ⇒ 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.

### 5.2.5 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. Standardmäßig werden 2 PDOs in Tx- und Rx-Richtung angelegt.

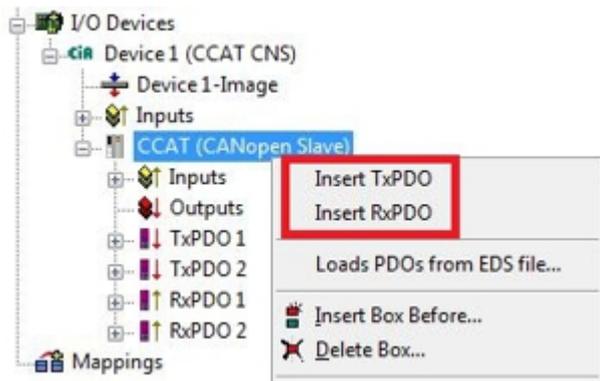
An dieser Stelle wird gezeigt, wie bei einem CANopen-Slave weitere PDOs angelegt werden.

Voraussetzungen für diesen Arbeitsschritt:

- Ein CANopen-Slave angefügt in der Strukturansicht.

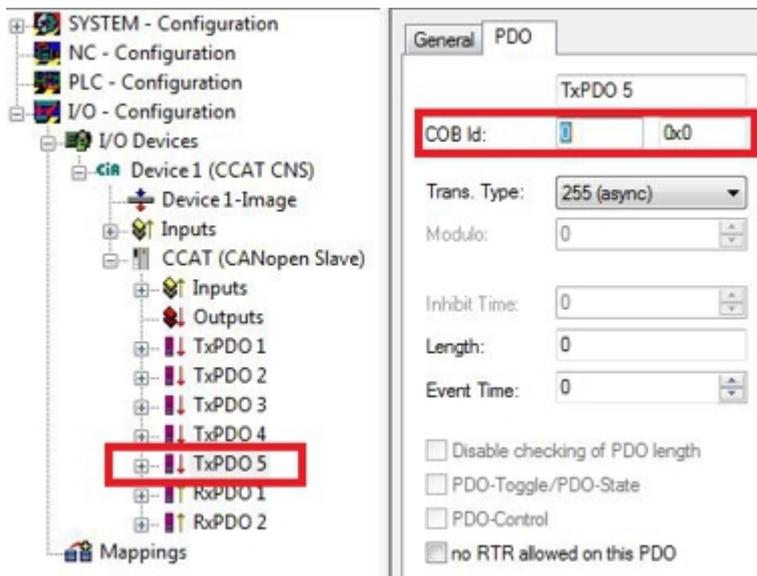
Legen Sie die PDOs wie folgt an:

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.

## 5.2.6 Variablen anlegen

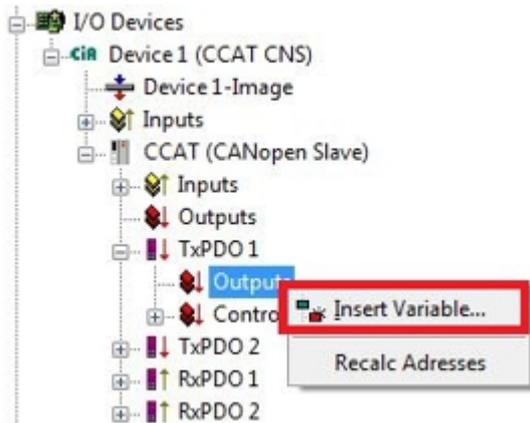
Die PDOs werden in TwinCAT mit Variablen gefüllt, die später mit dem SPS-Programm verknüpft werden können. In diesem Kapitel wird beschrieben, wie Sie Variablen anlegen.

Voraussetzungen für diesen Arbeitsschritt:

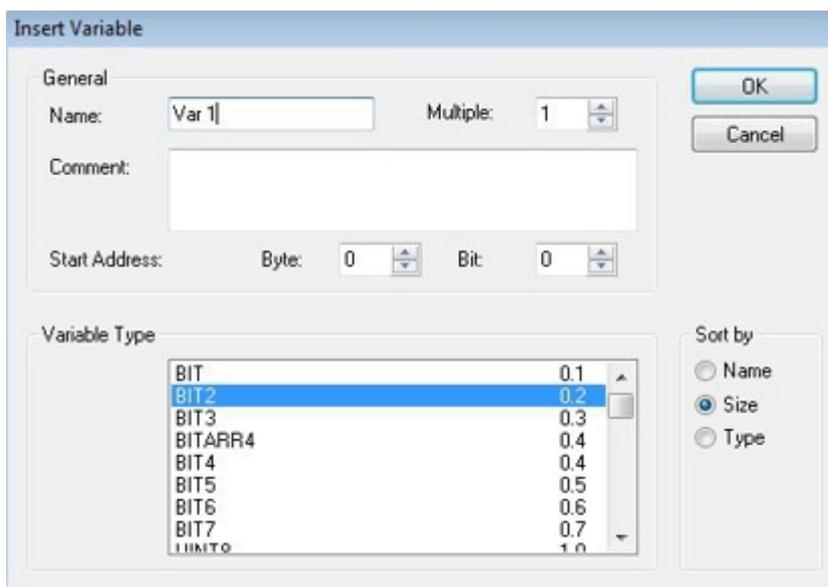
- Fertig angelegte PDOs, die mit Variablen gefüllt werden sollen.

**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 **Insert Variable**. 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.

### 5.2.7 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.

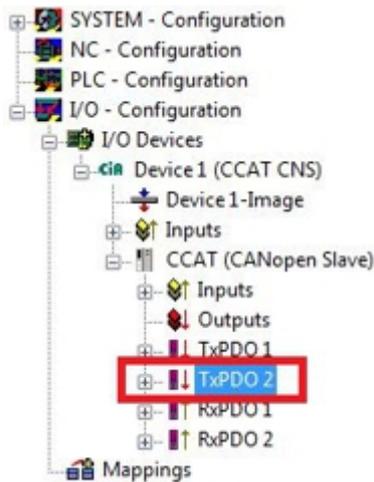
Übertragungsart:	Azyklisch Synchron	Zyklisch Synchron	Asynchron
Bezeichnung in TwinCAT:	(acyc, sync)	(cyc, sync)	(async)

Voraussetzungen für diesen Arbeitsschritt:

- Ein CANopen-Slave mit Prozessdatenobjekten (PDO) angefügt in TwinCAT

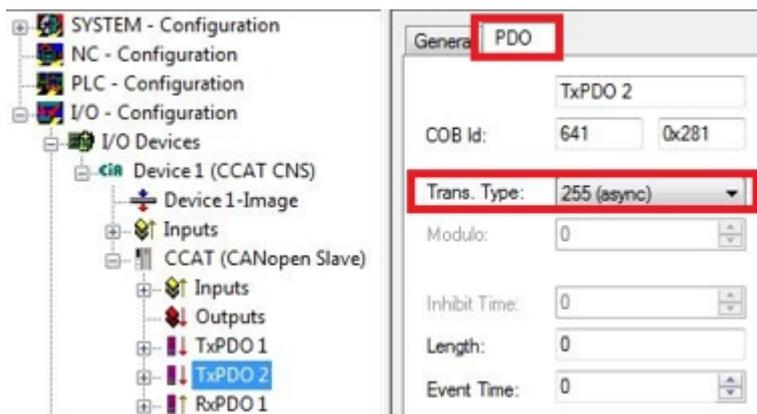
**Legen Sie die Übertragungsart wie folgt fest:**

1. Klicken Sie links in der Strukturansicht auf ein Prozessdatenobjekt (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.

## 5.2.8 PLC-Projekt erstellen

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

Voraussetzungen für diesen Arbeitsschritt:

- Ein Embedded-PC, eingefügt in TwinCAT.

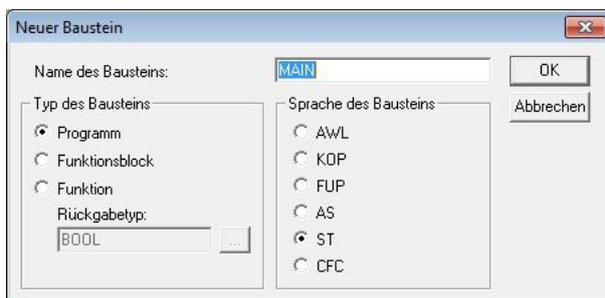
**Erstellen Sie ein PLC-Projekt wie folgt:**

1. Klicken Sie im Startmenü mit rechter Maustaste auf das TwinCAT Symbol.
2. Klicken Sie im Kontextmenü auf **PLC Control**.

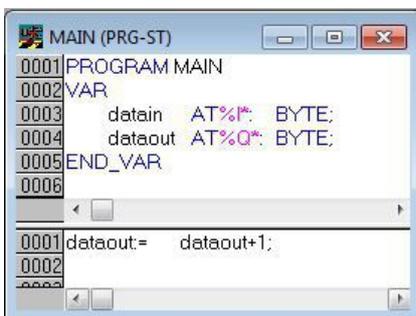


Das Fenster TwinCAT PLC Control erscheint.

3. Klicken Sie im Menü auf **File > New** und wählen Sie die Option **PC oder CX (x86)**.
4. Wählen Sie unter **Typ des Bausteins** die Option **Programm** und unter Sprache des Bausteins die Option **ST (strukturierter Text)**.



5. Schreiben Sie ein kleines Programm.



6. Speichern Sie das PLC-Projekt und klicken Sie im Menü auf **Projekt > Übersetzen**.

⇒ Nachdem das Projekt übersetzt wurde, wird eine Datei mit der Endung .tpy am gleichen Speicherort wie die Projektdatei erzeugt. Der Dateiname der neu erzeugten Datei ist identisch mit dem Dateinamen des PLC-Projekts.

Im nächsten Schritt können Sie das übersetzte PLC-Projekt im TwinCAT System Manager anfügen.

**PLC-Projekt anfügen**

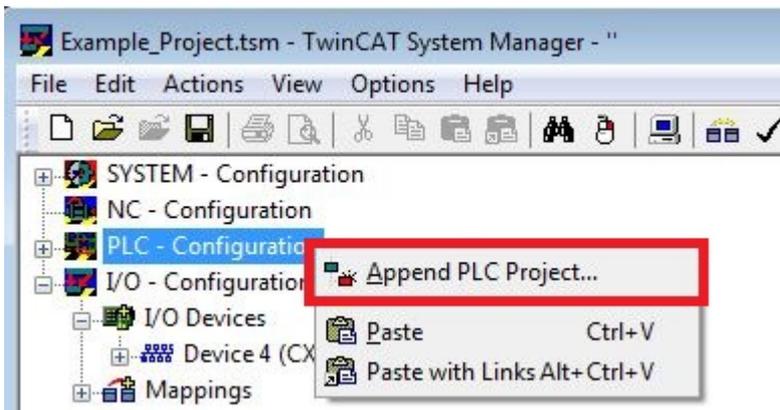
Das PLC-Projekt kann im System Manager angefügt werden. Die neu angelegten Variablen aus einem PLC-Projekt werden im System Manager eingebunden und können mit den Ein- und Ausgängen der Hardware verknüpft werden.

Voraussetzungen für diesen Arbeitsschritt:

- Ein Embedded-PC, eingefügt in TwinCAT.
- Ein fehlerfrei übersetztes PLC-Projekt und eine .tpy-Datei.

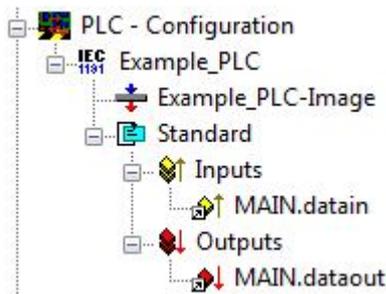
**Gehen Sie wie folgt vor:**

1. Wechseln Sie wieder zum Fenster System Manager.
2. Klicken Sie links in der Strukturansicht mit der rechten Maustaste auf **PLC – Configuration**.
3. Klicken Sie im Kontextmenü auf **Append PLC Project**.



4. Wählen Sie eine Datei mit der Dateiendung .tpy in Ihrem Systemverzeichnis und bestätigen Sie mit **OK**.

Das PLC-Projekt wird in der Strukturansicht unter PLC – Configuration eingefügt. Die im Projekt definierten Variablen werden unter den Ein- und Ausgängen angezeigt.



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

## 5.2.9 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 Geräte verknüpfen.

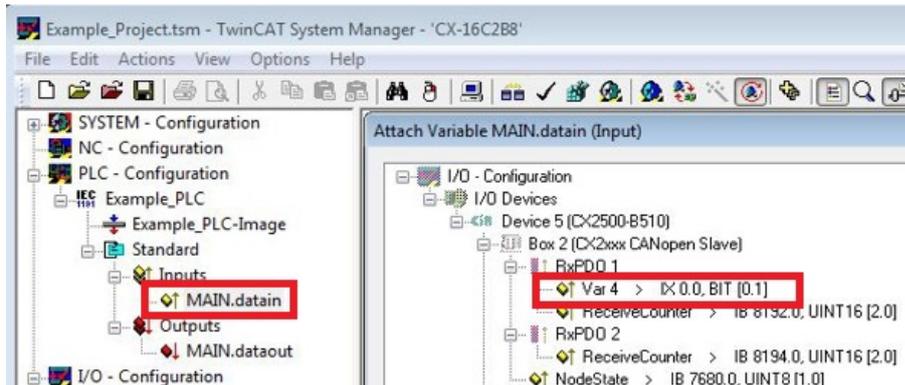
Voraussetzungen für diesen Arbeitsschritt:

- Ein angefügtes PLC-Projekt im System Manager.

**Verknüpfen Sie die Variablen wie folgt:**

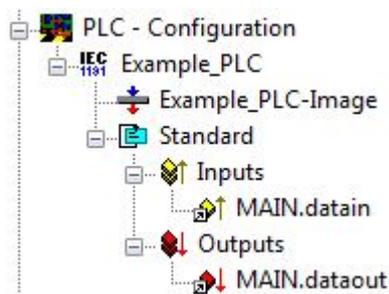
1. Klicken Sie doppelt auf die Ein- bzw. Ausgangsvariablen in der Strukturansicht unter **PLC - Configuration**.

Das Fenster **Attach Variable** erscheint und zeigt an, welche Eingänge bzw. Ausgänge mit Variablen verknüpft werden können.



2. Klicken Sie doppelt im Fenster Attach Variable auf die Ein bzw. Ausgänge.

Die Eingangsvariablen werden damit mit den Eingängen und die Ausgangsvariablen mit den Ausgängen Ihrer Hardware verknüpft.



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.

## 5.2.10 Konfiguration auf CX laden

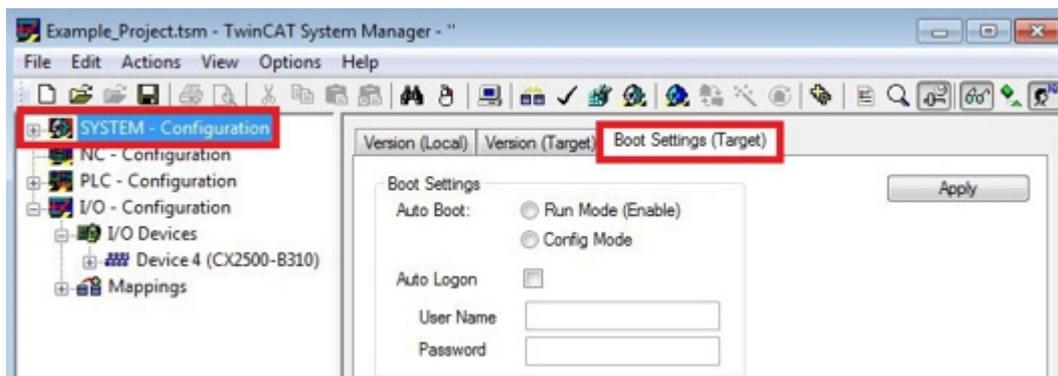
Wenn alle Variablen verknüpft sind, kann die Konfiguration gespeichert und auf den CX geladen werden. Das hat den Vorteil, dass das PLC-Projekt automatisch geladen und gestartet werden kann, wenn der CX eingeschaltet wird. Der Start des zuvor erstellten PLC-Projekts kann damit automatisiert werden.

Voraussetzungen für diesen Arbeitsschritt:

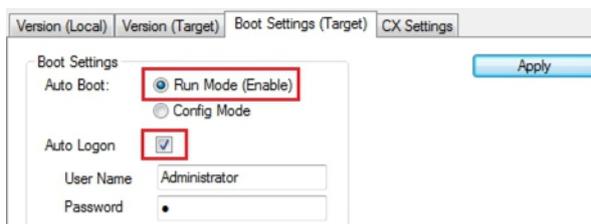
- Ein fertiges und im System Manager angefügtes PLC-Projekt.
- Variablen aus dem PLC-Projekt verknüpft mit der Hardware im System Manager.
- Ein CX ausgewählt als Zielsystem.

Laden Sie die Konfiguration wie folgt auf den CX:

1. Klicken Sie links in der Strukturansicht auf **SYSTEM – Configuration**.
2. Klicken Sie auf die Registerkarte **Boot Settings (Target)**.



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 Benutzernamen und das Passwort für den CX ein.
5. Klicken Sie auf **Apply**.  
Das Fenster Logon Information erscheint.



6. Geben Sie den Benutzernamen und das Passwort erneut ein und klicken Sie auf **OK**.

7. Klicken Sie links in der Strukturansicht auf **PLC – Configuration** und dann auf die Registerkarte **Pic Settings (Target)**.



8. Wählen Sie die Start PLC unter Boot Project und klicken Sie auf **Apply**.

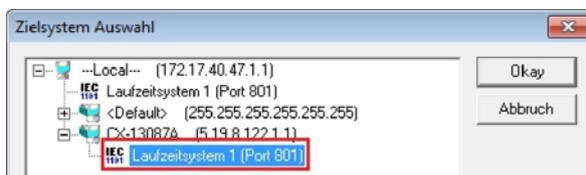


9. Starten Sie PLC-Control und öffnen Sie das PLC-Projekt.



10. Klicken Sie oben in der Menüleiste auf **Online** und dann auf **Choose Run- Time System**.

11. Wählen Sie das Laufzeitsystem von dem CX aus und klicken Sie auf **Okay**.



12. Klicken Sie oben in der Menüleiste auf **Online** und dann auf **Login**. Das PLC-Projekt wird eingeloggt.

13. Klicken Sie oben in der Menüleiste auf **Online** und dann auf **Create Bootproject**.

⇒ 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.

## 5.2.11 CANopen-Master anfügen

Der CANopen-Master wird wie die anderen Geräte mit dem TwinCAT System Manager eingefügt. Über den angefügten Master kann dann nach allen angeschlossenen Slaves gesucht werden. Im Folgenden wird gezeigt, wie Sie einen CANopen-Master in TwinCAT anfügen.

Voraussetzungen für diesen Arbeitsschritt:

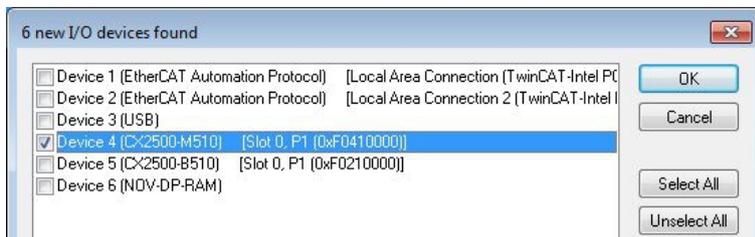
- TwinCAT muss sich im Config Mode befinden.
- Ein ausgewähltes Zielsystem (in diesem Beispiel ist es der Embedded-PC CX2020-M510)

### Fügen Sie ein CANopen-Gerät wie folgt an:

1. Starten Sie den System Manager.
2. Klicken Sie links in der Strukturansicht mit rechter Maustaste auf **I/O Devices**.
3. Klicken Sie im Kontextmenü auf **Scan Devices**.

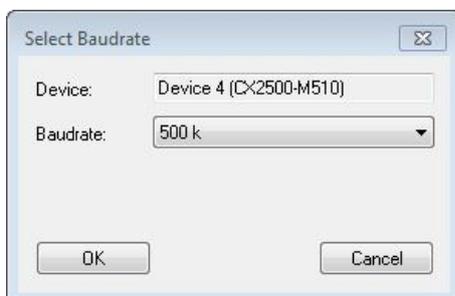


4. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.

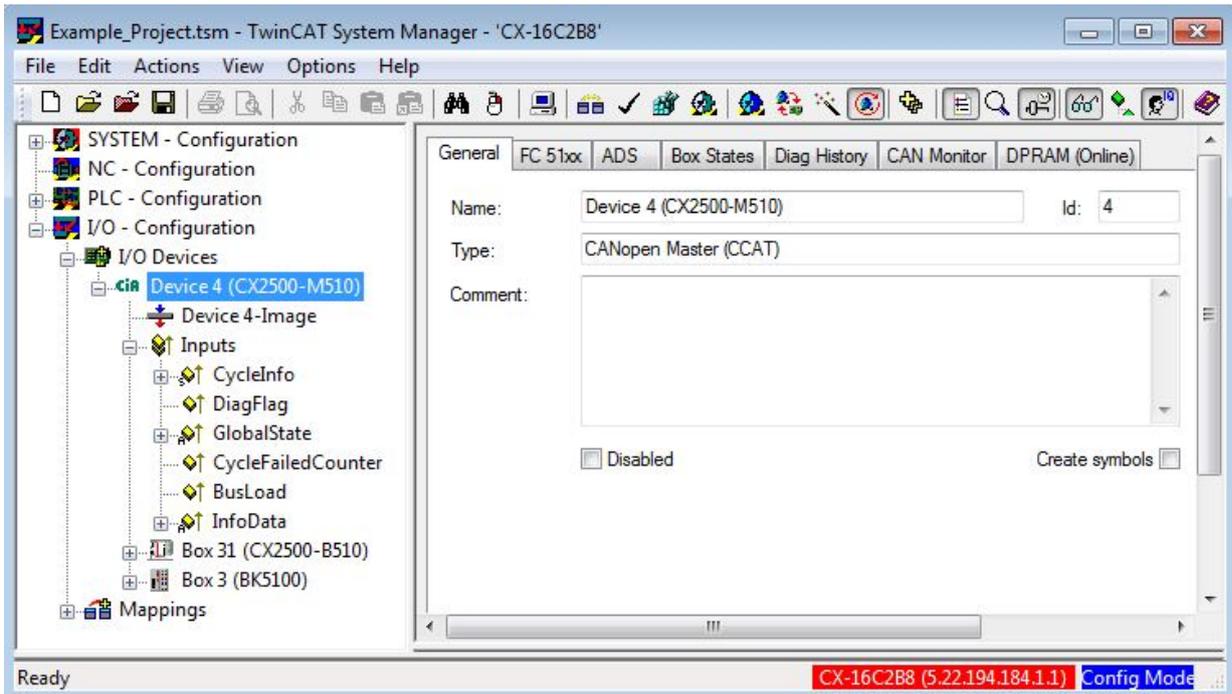


5. Bestätigen Sie die Anfrage mit Ja, um nach Boxen zu suchen. Das Fenster **Select Baudrate** erscheint.

6. 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.

## 5.3 Mit TwinCAT 3 parametrieren

In diesem Kapitel wird beispielhaft gezeigt, wie Sie CANopen-Geräte mit Hilfe von TwinCAT 3 parametrieren können. Für das Beispiel werden insgesamt drei Geräte verwendet, darunter ein CANopen-Master an dem zwei CANopen-Slaves angeschlossen sind.

Zuerst wird gezeigt, wie Sie ein Zielsystem in TwinCAT suchen und auswählen können. Als nächstes wird ein CANopen-Slave in TwinCAT eingefügt, parametriert und die CANopen-Adresse des Slaves eingestellt. Danach wird ein kleines PLC-Projekt erstellt und in TwinCAT angefügt. Anschließend werden die Variablen aus dem PLC-Projekt mit der Hardware verknüpft und die fertige Konfiguration auf den CANopen-Slave geladen.

Als letzten Schritt wird der CANopen-Master in TwinCAT angefügt und nach den beiden CANopen-Slaves über den Master gesucht.

### 5.3.1 Zielsysteme suchen

Bevor Sie mit den Geräten arbeiten können, müssen Sie Ihren lokalen Rechner mit dem Zielgerät verbinden. Danach können Sie mit Hilfe der IP-Adresse oder dem Host Namen nach Geräten suchen.

Der lokale PC und die Zielgeräte müssen mit dem gleichen Netzwerk oder direkt über ein Ethernet Kabel miteinander verbunden werden. In TwinCAT kann auf diese Weise nach allen Geräten gesucht und anschließend projektiert werden.

Voraussetzungen für diesen Arbeitsschritt:

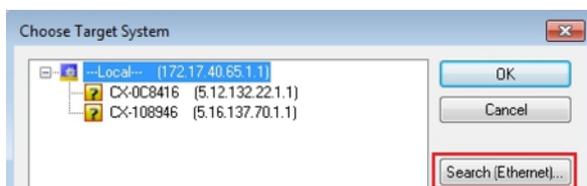
- TwinCAT 3 muss sich im Config Mode befinden.
- IP-Adresse oder Host Name des Gerätes.

**Suchen Sie nach den Geräten wie folgt:**

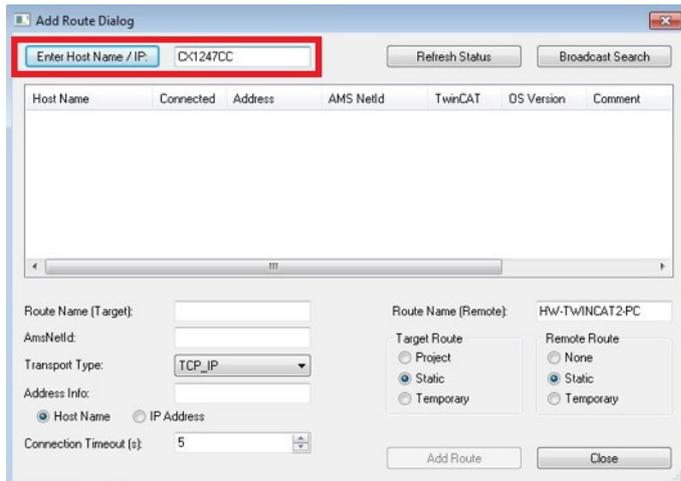
1. Klicken Sie oben im Menü auf **File > New > Project** und erstellen Sie ein neues TwinCAT XAE Projekt.
2. Klicken Sie links in der Strukturansicht auf **SYSTEM** und dann auf **Choose Target**.



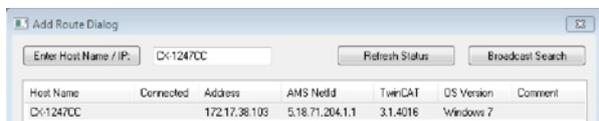
3. Klicken Sie auf **Search (Ethernet)**.



- Tippen Sie im Feld **Enter Host Name / IP** den Host Namen oder die IP-Adresse des Gerätes ein und drücken Sie **[Enter]**.



- Markieren Sie das gefundene Gerät und klicken Sie auf **Add Route**.



Das Fenster Logon Information erscheint.

Geben Sie im Feld **User Name** und im Feld **Password** den Benutzernamen und das Passwort für den CX ein und klicken Sie auf **OK**.

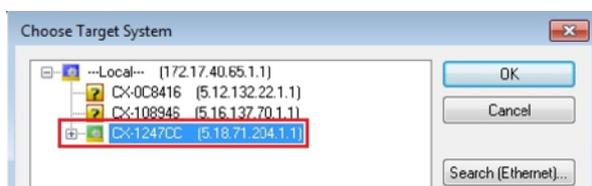


Als Standard ist bei den CXen folgende Information eingestellt:

**User name:** Administrator **Password:** 1

- Klicken Sie auf **Close**, wenn Sie keine weiteren Geräte suchen wollen und schließen damit das Add Route Fenster.  
Das neue Gerät wird im Fenster Choose Target System angezeigt.

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



- ⇒ Sie haben erfolgreich in TwinCAT nach einem Gerät gesucht und das Gerät als Zielsystem eingefügt. In der Menüleiste wird das neue Zielsystem mit dem Host Namen angezeigt.



Mit dieser Vorgehensweise können Sie nach allen verfügbaren Geräten suchen und auch jederzeit zwischen den Zielsystemen wechseln. Als nächstes können Sie das Gerät in TwinCAT in die Strukturansicht anfügen.

### 5.3.2 CANopen-Slave anfügen

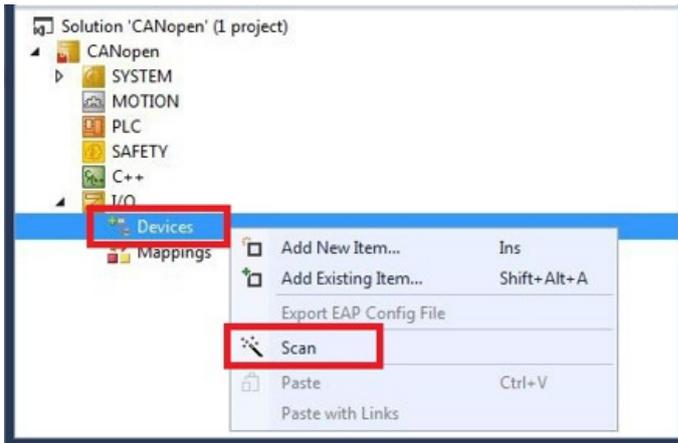
In dem exemplarischen Aufbau ist ein CANopen-Slave CX2020 mit Feldbusmodul CX2500-B510 am CANopen-Master angeschlossen. Damit der CANopen-Slave konfiguriert und später vom CANopen-Master mit allen Ein- und Ausgängen erkannt wird, muss der CANopen-Slave zuerst in TwinCAT angefügt werden.

Voraussetzungen für diesen Arbeitsschritt:

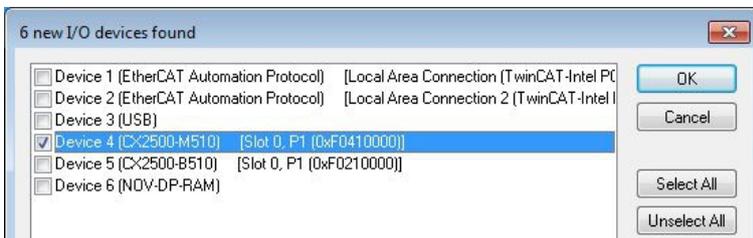
- Ein gescanntes und ausgewähltes Zielgerät mit CANopen-Slave. In diesem Beispiel ist es der CX2020 mit Feldbusmodul CX2500-B510.

**Fügen Sie den CANopen-Slave wie folgt ein:**

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

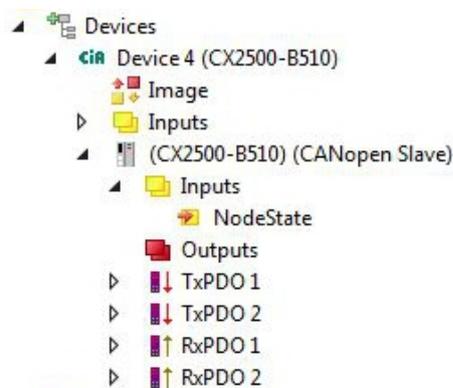


3. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.



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

⇒ 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.

### 5.3.3 Virtuellen Slave anlegen

Es können zusätzliche virtuelle Slaves auf der gleichen Hardware Schnittstelle angelegt werden. Dadurch können mehr Daten mit einem CANopen-Master ausgetauscht oder eine Verbindung mit einem zweiten CANopen-Master angelegt werden.

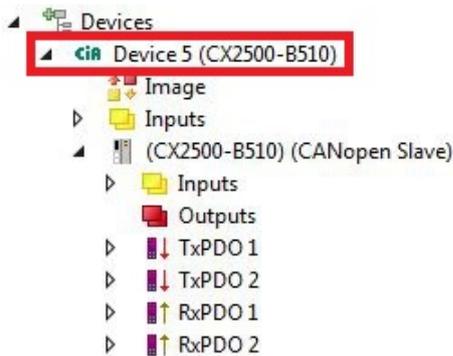
Jeder virtuelle Slave bekommt über TwinCAT eine eigene Adresse und wird für den CANopen-Master wie ein eigenständiges Gerät konfiguriert.

Voraussetzungen für diesen Arbeitsschritt:

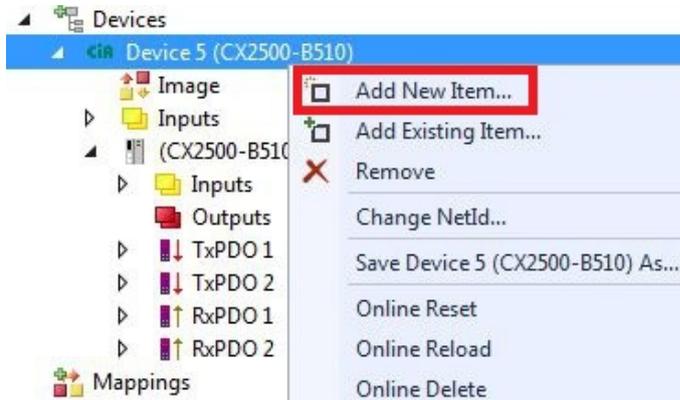
- Ein CANopen-Slave angelegt in TwinCAT.

**Legen Sie einen virtuellen Slave wie folgt ein:**

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.

### 5.3.4 Adresse einstellen

Nachdem der CANopen-Slave erfolgreich in TwinCAT angefügt wurde, kann die Adresse des CANopen-Slaves eingestellt werden. Geräte mit einem Dip-Schalter haben eine voreingestellte Adresse. Die Adresse auf dem Dip-Schalter muss mit der in TwinCAT eingestellten Adresse übereinstimmen.

Bei Geräten ohne Dip-Schalter wird die Adresse nur in TwinCAT eingestellt.

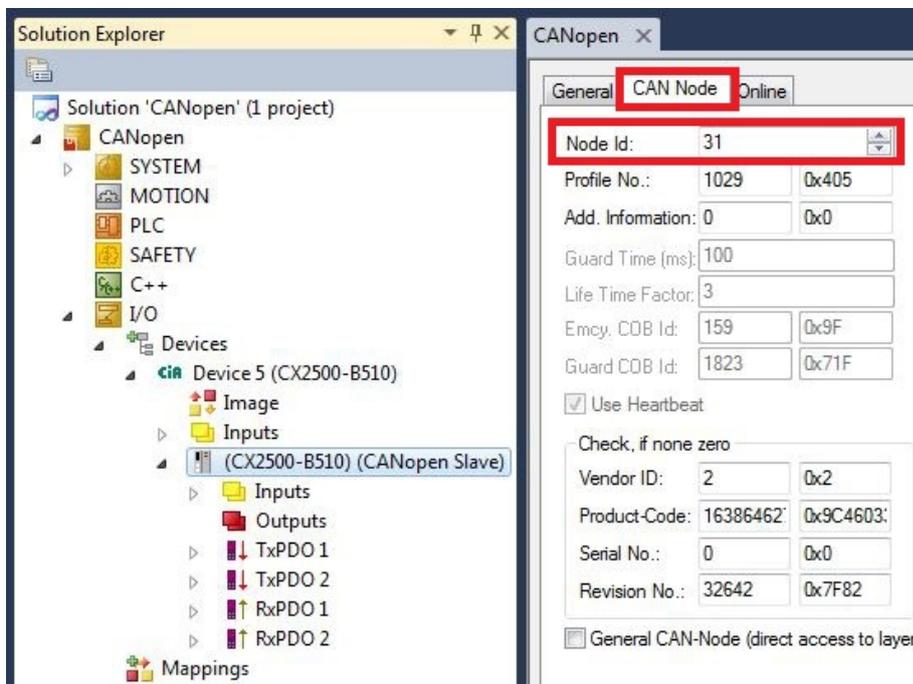
In diesem Arbeitsschritt wird die Adresse in TwinCAT eingestellt, damit der CANopen-Slave über diese Adresse für den CANopen-Master erreichbar ist.

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. „31“.



- ⇒ 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.

### 5.3.5 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. Standardmäßig werden 2 PDOs in Tx- und Rx-Richtung angelegt.

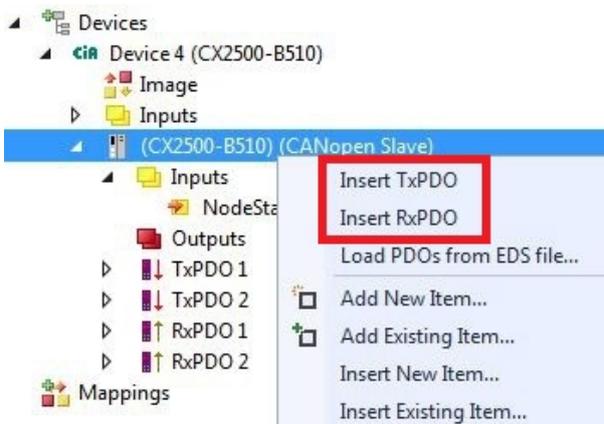
An dieser Stelle wird gezeigt, wie bei einem CANopen-Slave weitere PDOs angelegt werden.

Voraussetzungen für diesen Arbeitsschritt:

- Ein CANopen-Slave angefügt in der Strukturansicht.

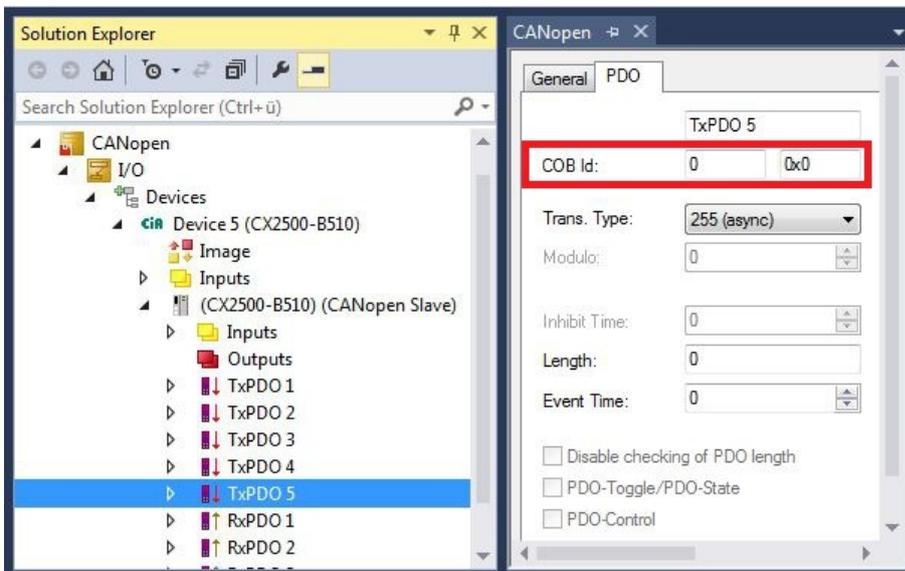
**Legen Sie die PDOs wie folgt an:**

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.

### 5.3.6 Variablen anlegen

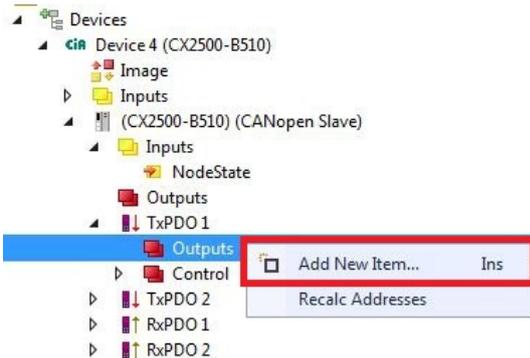
Die PDOs werden in TwinCAT mit Variablen gefüllt, die später mit dem SPS-Programm verknüpft werden können. In diesem Kapitel wird beschrieben, wie Sie Variablen anlegen.

Voraussetzungen für diesen Arbeitsschritt:

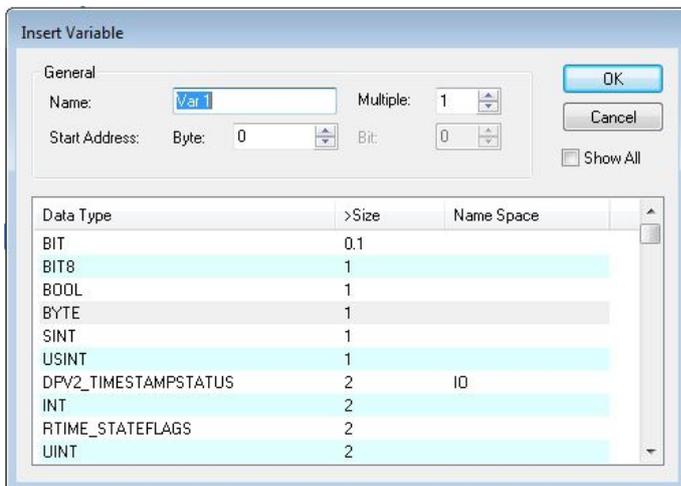
- Fertig angelegte PDOs, die mit Variablen gefüllt werden sollen.

**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.

### 5.3.7 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.

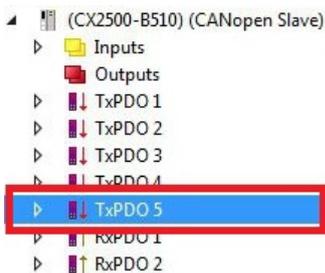
Übertragungsart:	Azyklisch Synchron	Zyklisch Synchron	Asynchron
Bezeichnung in TwinCAT:	(acyc, sync)	(cyc, sync)	(async)

Voraussetzungen für diesen Arbeitsschritt:

- CANopen-Slave mit Prozessdatenobjekten (PDO) angefügt in TwinCAT

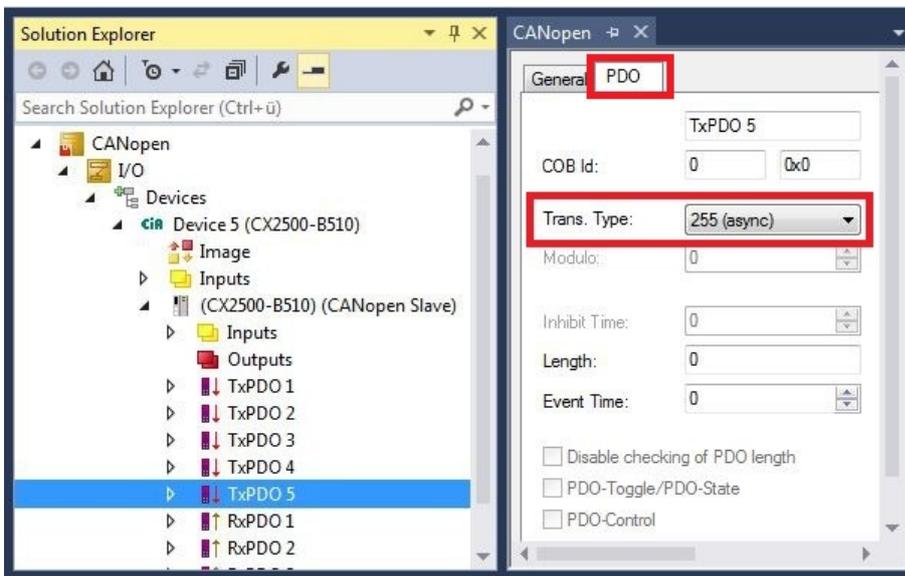
**Legen Sie die Übertragungsart wie folgt fest:**

1. Klicken Sie links in der Strukturansicht auf ein Prozessdatenobjekt (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.

### 5.3.8 PLC-Projekt erstellen

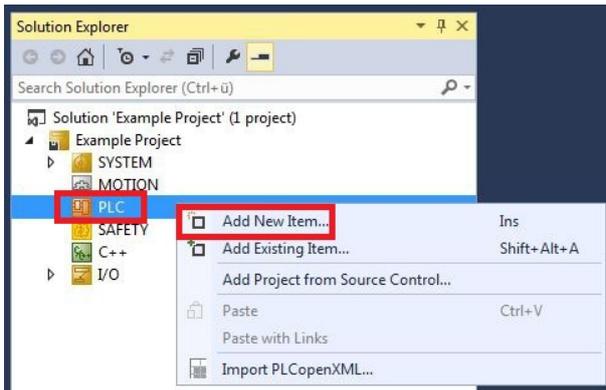
In den nächsten Schritten wird beschrieben, wie Sie ein PLC-Projekt in TwinCAT erstellen und in der Strukturansicht einfügen.

Voraussetzungen für diesen Arbeitsschritt:

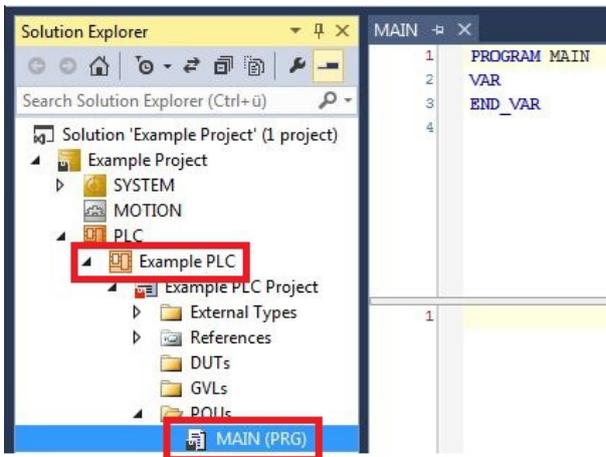
- Ein neu angelegtes TwinCAT XAE Projekt.

**Erstellen Sie ein PLC-Projekt wie folgt:**

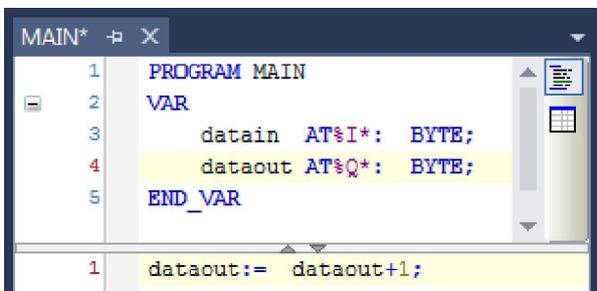
1. Klicken Sie in der Strukturansicht mit der rechten Maustaste auf **PLC**.
2. Klicken Sie im Kontextmenü auf **Add New Item** und wählen Sie das **Standard PLC Project**.



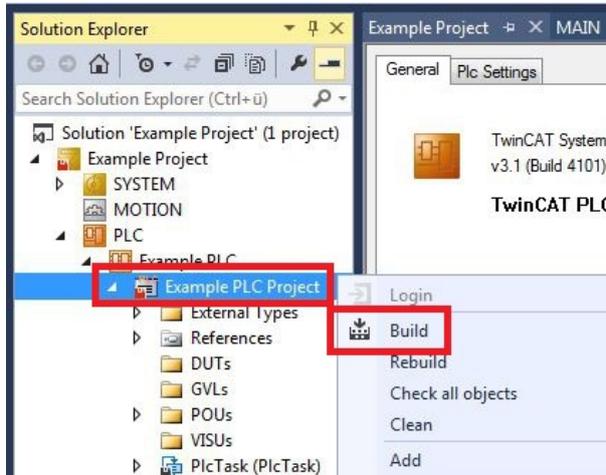
3. Klicken Sie in der Strukturansicht auf das neu erstellte PLC-Projekt und dann unter **POUs** doppelt auf **MAIN (PRG)**.



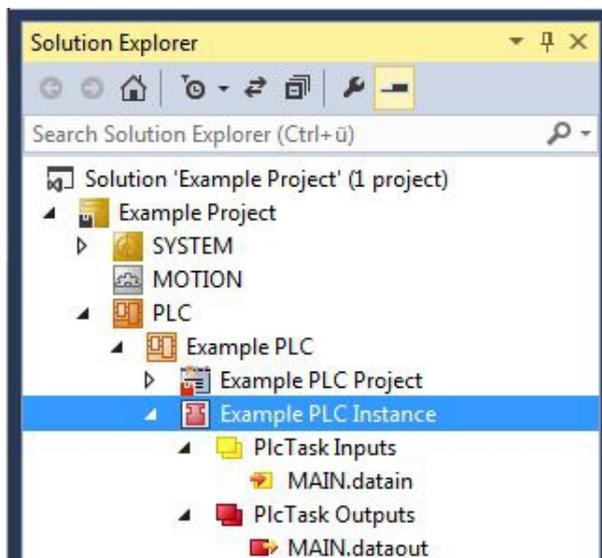
4. Schreiben Sie ein kleines Programm wie im folgenden Bild.



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.

### 5.3.9 Variablen verknüpfen

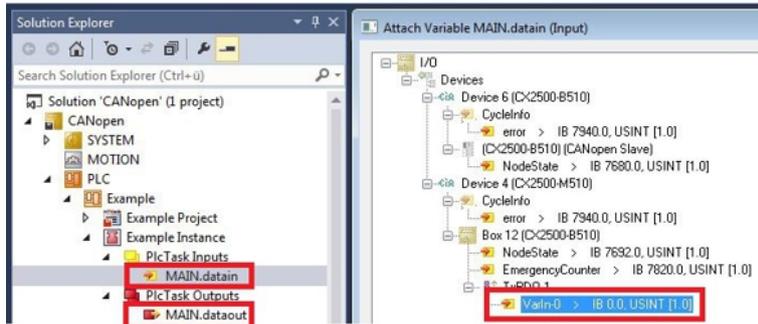
Wurde das PLC-Projekt erfolgreich im System Manager angefügt, können die Sie die neu angelegten Ein- und Ausgangsvariablen aus dem PLC-Projekt mit den Ein- und Ausgängen Ihrer Hardware verknüpfen.

Voraussetzungen für diesen Arbeitsschritt:

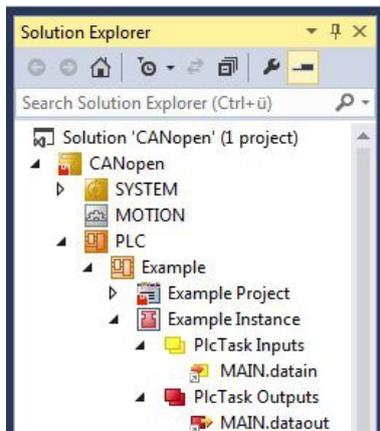
- Ein angefügtes PLC-Programm in TwinCAT.

**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.

### 5.3.10 Konfiguration auf CX laden

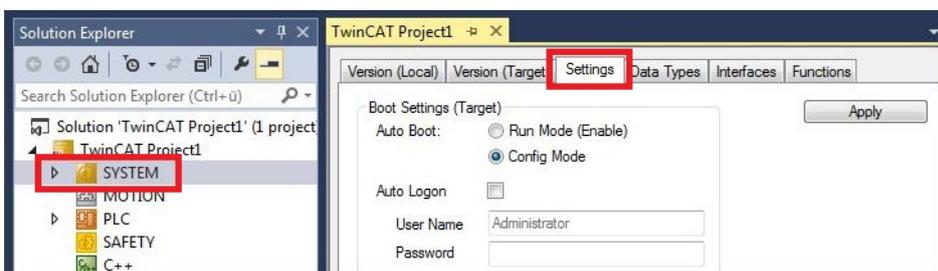
Wenn Variablen verknüpft sind, kann die Konfiguration gespeichert und auf den CX geladen werden. Das hat den Vorteil, dass das PLC-Projekt automatisch geladen und gestartet werden kann, wenn der CX eingeschaltet wird. Der Start des zuvor erstellten PLC-Projekts kann damit automatisiert werden.

Voraussetzungen für diesen Arbeitsschritt:

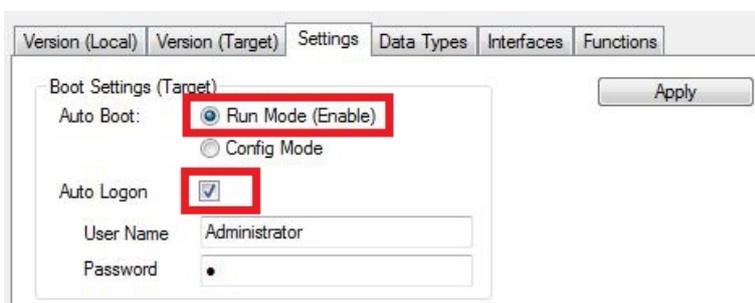
- Ein fertiges und im System Manager angefügtes PLC-Projekt.
- Variablen aus dem PLC-Projekt verknüpft mit der Hardware im System Manager.
- Ein CX ausgewählt als Zielsystem.

**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**.

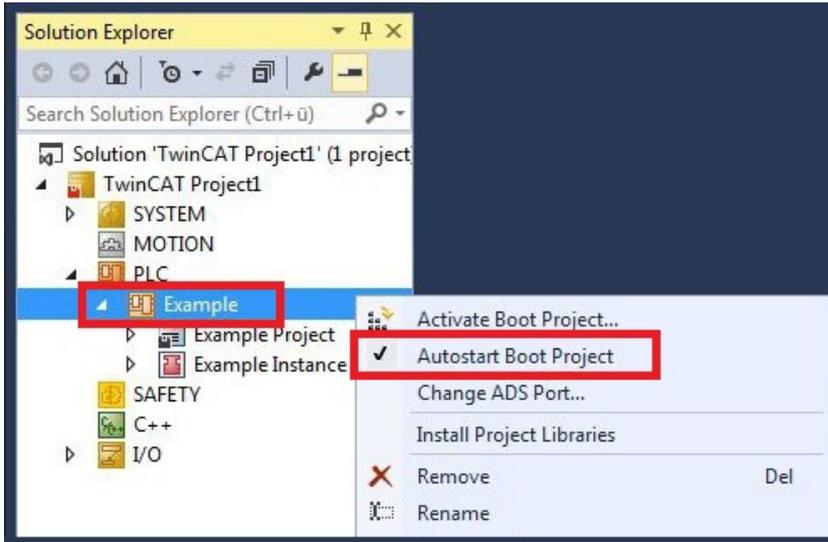


3. Wählen Sie unter Boot Settings die Option **Run Mode (Enable)** und Aktivieren Sie das Kontrollkästchen **Auto Logon**.

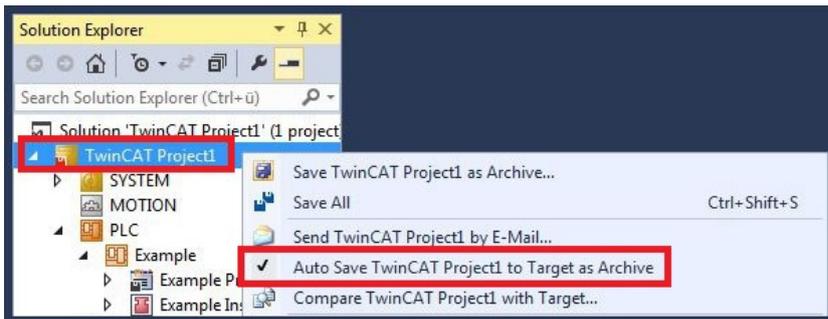


4. Geben Sie im Feld **User Name** und im Feld **Password** den Benutzernamen und das Passwort für den CX ein.
5. Klicken Sie auf **Apply**.
6. Klicken Sie links in der Strukturansicht unter **PLC** mit der rechten Maustaste auf das PLC-Projekt.

7. Klicken Sie im Kontextmenü auf **Autostart Boot Project**.  
Die Einstellung 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.

### 5.3.11 CANopen-Master anfügen

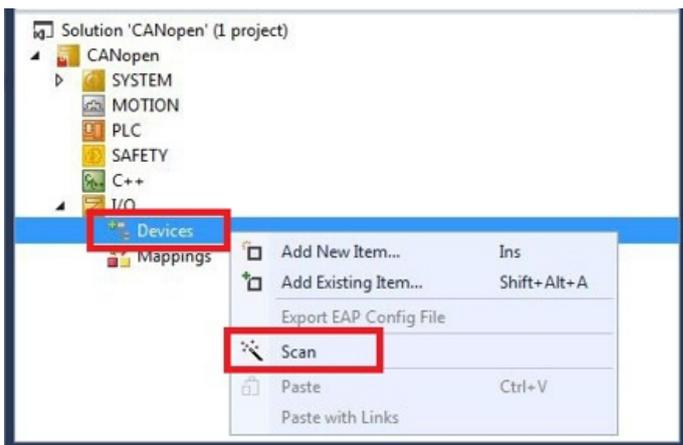
Der CANopen-Master wird wie die anderen Geräte mit dem TwinCAT System Manager eingefügt. Über den angefügten Master kann dann nach allen angeschlossenen Slaves gesucht werden. Im Folgenden wird gezeigt, wie Sie einen CANopen-Master in TwinCAT anfügen.

Voraussetzungen für diesen Arbeitsschritt:

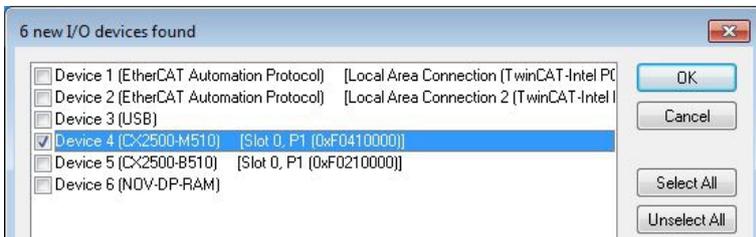
- TwinCAT muss sich im Config Mode befinden.
- Ein ausgewähltes Zielsystem (in diesem Beispiel ist es der Embedded-PC CX2020-M510)

**Fügen Sie ein CANopen-Gerät wie folgt an:**

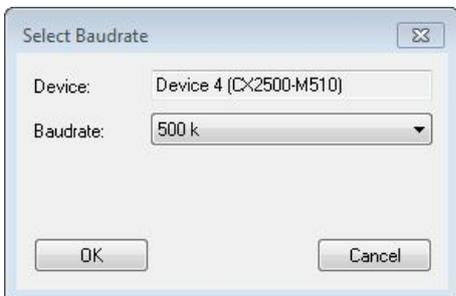
1. Klicken Sie links in der Strukturansicht mit rechter Maustaste auf **Devices**.
2. Klicken Sie im Kontextmenü auf **Scan**.



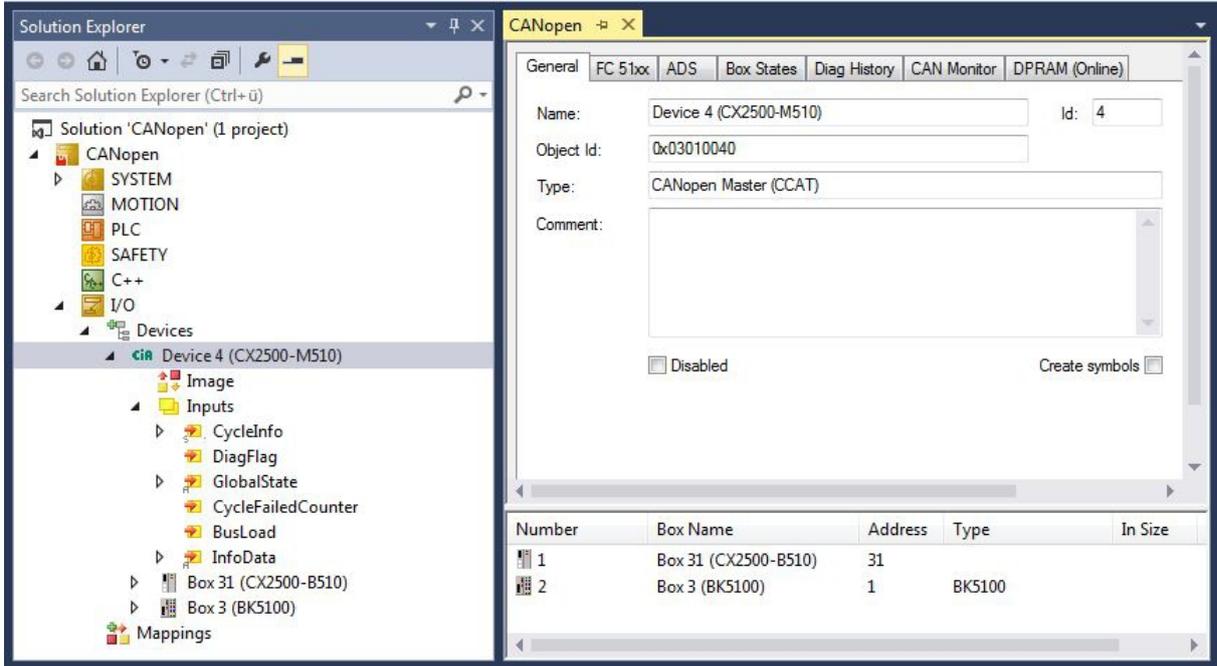
3. Wählen Sie die Geräte, die Sie verwenden wollen und bestätigen die Auswahl mit **OK**.



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.

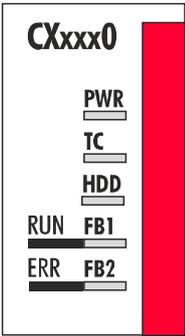
## 6 Fehlerbehandlung und Diagnose

### 6.1 Diagnose-LEDs

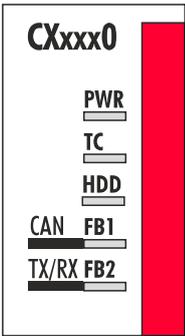
An dieser Stelle werden die Diagnose-LEDs eines CANopen-Masters und CANopen-Slaves beschrieben. Die Beschriftung der Diagnose-LEDs auf einem Feldbusmodul CX2500 und einem Embedded-PC mit Optionsschnittstelle ist identisch.

In der Beschreibung der LEDs wird deshalb nur zwischen CANopen-Master und CANopen-Slave unterschieden.

#### M510 (Master)

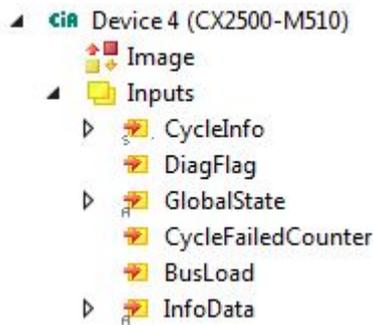
Anzeige	LED	Farbe	Bedeutung
	RUN	Grün	CAN ist in Ordnung
		Rot	CAN im BUS off
		Grün 200 ms / rot 200ms	CAN Warnung
		Sehr schnelles blinken von grün und rot	Baudraten Suche aktiv
		Grün aus / Rot aus	CAN nicht konfiguriert
	ERR	Grün an / Rot aus	Alle Nodes haben den NodeState = 0
		Grün 200 ms / Rot 200 ms	Alle Boxen im OP aber die Task noch nicht gestartet
		Grün aus / Rot 200 ms	Nicht alle Nodes im OP
		Grün aus / Rot aus	Keine Boxen Konfiguriert

#### B510 (Slave)

Anzeige	LED	Farbe	Bedeutung
	CAN Zeigt den Status des CAN an	Grün	CAN ist in Ordnung
		Rot	CAN im BUS off
		Grün 200 ms / rot 200ms	CAN Warnung
		Sehr schnelles blinken von grün und rot	Baudraten Suche aktiv
		Grün aus / Rot aus	CAN nicht konfiguriert
	Tx/Rx Zeigt Fehler des CAN an	Grün an	Alles in Ordnung
		Grün 200 ms / Rot 200 ms	Alle Boxen im OP aber die Task ist noch nicht gestartet
		Rot 200 ms	Nicht alle Boxen im OP
	Grün aus / Rot aus	Keine Boxen konfiguriert	

## 6.2 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	<p>Cycle Counter: Dieser Zähler wird nach jedem Zyklus um eins erhöht.</p> <p>Error: Zeigt die Anzahl der Boxen an mit einem BoxState ungleich Null.</p> <p>ActualCycle Time: Für zukünftige Verwendung reserviert</p>
DiagFlag	<p>Diese Variable liefert Informationen darüber, ob sich die Diagnosedaten verändert haben.</p> <ul style="list-style-type: none"> <li>• 0: Daten nicht verändert.</li> <li>• 1: Daten verändert. Benutze ADS-Read um die Daten auszulesen.</li> </ul>
GlobalState	<p>Diese Variable liefert Informationen über den Status des Masters.</p> <p>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.</p> <p>GlobalState[1] (FW V02.14 und höher): Bit 0-7: RxError-Counter des CAN-Controllers. Bit 8-15: TxError-Counter des CAN-Controllers.</p> <p>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.</p> <p>GlobalState[3]: Bus Auslastung in %.</p>
CycleFailedCounter	<p>Dieser Zähler wird jedes Mal um eins erhöht, wenn am Anfang eines TwinCAT-Zyklus der letzte Bus-Zyklus nicht abgeschlossen wurde.</p>
BusLoad	<p>Bus Auslastung in %.</p>
InfoData	

## 6.3 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.



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

Die folgende Tabelle zeigt, welche Werte die Variable **DiagFlag** annehmen kann. Diese Variable liefert Informationen darüber, ob sich die Diagnosedaten verändert haben.

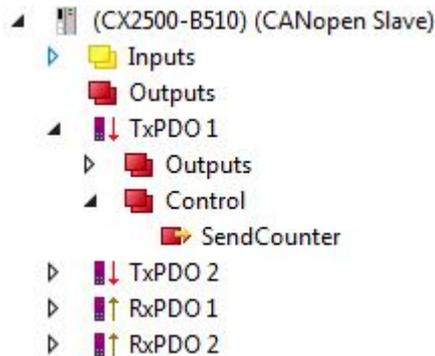
Wert	Bedeutung
0	Daten nicht verändert.
1	Daten verändert. Benutze ADS-Read um die Daten auszulesen.

Bei der Variable **EmergencyCounter** wird der Zähler um eins erhöht, wenn ein Emergency Telegramm erhalten wurde.

## 6.4 PDOs

### SendCounter

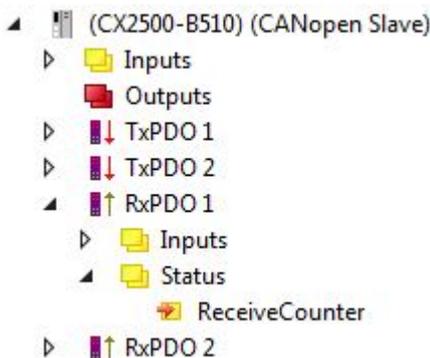
Bei den TxPDOs gibt es zusätzlich unter dem Menüpunkt **Control** die Variable SendCounter.



Die Ausgangsvariable wird immer um eins hochgezählt wenn ein PDO gesendet wird.

### ReceiveCounter

Bei den RxPDO gibt es zusätzlich unter dem Menüpunkt **Status** die Variable ReceiveCounter.



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. Dennoch 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.

## 7 Anhang

### 7.1 Zubehör

Kabel und Stecker für die Verbindung der CAN-Komponenten.

#### Kabel

Artikelnummer	Beschreibung
ZB5100	CAN-Kabel, 4-adrig, feste Verlegung 2 x 2 x 0,25 mm <sup>2</sup> , Preis pro Meter

#### Stecker

Artikelnummer	Beschreibung
ZS1051-3000	Businterface-Connector, D-Sub für CANopen, im Gehäuse, mit zuschaltbarem Abschlusswiderstand
ZS1052-3000	Businterface-Connector, 5-polig für BK5xxx, im Gehäuse, mit zuschaltbarem Abschlusswiderstand
ZS1052-5150	CAN-Diagnose-Interface

## 7.2 Zertifizierungen

Prinzipiell sind alle Produkte der Embedded-PC-Familie CE, UL und EAC zertifiziert. Da sich aber die Produktfamilie ständig weiterentwickelt, kann hier keine Auflistung angegeben werden. Die aktuelle Auflistung der zertifizierten Produkte kann auf der Internetseite [www.beckhoff.de](http://www.beckhoff.de) unter Embedded-PC nachgelesen werden.

### **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.

## 7.3 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

### Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49(0)5246/963-157  
Fax: +49(0)5246/963-9157  
E-Mail: support@beckhoff.com

### Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49(0)5246/963-460  
Fax: +49(0)5246/963-479  
E-Mail: service@beckhoff.com

Weitere Support- und Serviceadressen finden Sie auf unseren Internetseiten unter <http://www.beckhoff.de>.

### Beckhoff Firmenzentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49(0)5246/963-0  
Fax: +49(0)5246/963-198  
E-Mail: info@beckhoff.com

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unseren Internetseiten:

<http://www.beckhoff.de>

Dort finden Sie auch weitere [Dokumentationen](#) zu Beckhoff Komponenten.



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