

**BECKHOFF** New Automation Technology

Dokumentation | DE

# EL600x, EL602x

Serielle Schnittstellenklemmen





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b> .....	<b>5</b>
1.1	Übersicht Serielle Schnittstellenklemmen .....	5
1.2	Hinweise zur Dokumentation .....	6
1.3	Sicherheitshinweise .....	7
1.4	Ausgabestände der Dokumentation .....	8
1.5	Versionsidentifikation von EtherCAT-Geräten .....	10
1.5.1	Beckhoff Identification Code (BIC).....	14
<b>2</b>	<b>Produktübersicht</b> .....	<b>16</b>
2.1	EL6001, EL6021 .....	20
2.1.1	Einführung .....	20
2.1.2	Technische Daten .....	21
2.2	EL6002, EL6022 .....	23
2.2.1	Einführung .....	23
2.2.2	Technische Daten .....	24
2.3	Start Up .....	24
<b>3</b>	<b>Grundlagen der Kommunikation</b> .....	<b>25</b>
3.1	EtherCAT-Grundlagen .....	25
3.2	EtherCAT-Verkabelung - Drahtgebunden .....	25
3.3	Allgemeine Hinweise zur Watchdog-Einstellung .....	26
3.4	EtherCAT State Machine .....	28
3.5	CoE-Interface .....	30
3.6	Distributed Clock .....	35
<b>4</b>	<b>Montage und Verdrahtung</b> .....	<b>36</b>
4.1	Hinweise zum ESD-Schutz .....	36
4.2	EL6001, EL6021 .....	36
4.2.1	Tragschienenmontage .....	36
4.2.2	Anschluss .....	39
4.2.3	Positionierung von passiven Klemmen .....	44
4.2.4	LEDs und Anschlussbelegung .....	45
4.3	EL6002, EL6022 .....	47
4.3.1	Montage und Demontage - Frontentriegelung oben .....	47
4.3.2	Empfohlene Tragschienen .....	49
4.3.3	LEDs und Anschlussbelegung .....	50
4.4	Positionierung von passiven Klemmen .....	53
4.5	Montagevorschriften für erhöhte mechanische Belastbarkeit .....	54
4.6	Einbaulagen .....	55
4.7	UL-Hinweise .....	57
4.8	ATEX - Besondere Bedingungen (erweiterter Temperaturbereich) .....	58
4.9	Weiterführende Dokumentation zum Explosionsschutz .....	59
<b>5</b>	<b>Inbetriebnahme</b> .....	<b>60</b>
5.1	TwinCAT Quickstart .....	60
5.1.1	TwinCAT 2 .....	63
5.1.2	TwinCAT 3 .....	73

5.2	TwinCAT Entwicklungsumgebung .....	86
5.2.1	Installation TwinCAT Realtime Treiber .....	87
5.2.2	Hinweise ESI-Gerätebeschreibung.....	92
5.2.3	TwinCAT ESI Updater .....	96
5.2.4	Unterscheidung Online/Offline .....	96
5.2.5	OFFLINE Konfigurationserstellung .....	97
5.2.6	ONLINE Konfigurationserstellung.....	102
5.2.7	EtherCAT Teilnehmerkonfiguration .....	110
5.3	Allgemeine Inbetriebnahmehinweise des EtherCAT Slaves .....	120
5.4	Betriebsarten und Prozessdaten .....	128
5.5	Hinweise TcVirtualComDriver.....	135
5.6	Kommunikationseigenschaften.....	137
5.7	LIN Master Feature EL6001 .....	139
5.8	Beispielprogramme.....	142
5.8.1	Beispielprogramm 1 .....	142
5.8.2	Beispielprogramm 2.....	145
5.8.3	Beispielprogramm 3 (LIN).....	147
<b>6</b>	<b>Übersicht CoE Objekte EL6001, EL6021 .....</b>	<b>151</b>
6.1	Objektbeschreibung und Parametrierung .....	151
6.1.1	Objekte für die Inbetriebnahme .....	151
6.1.2	Standardobjekte (0x1000-0x1FFF).....	153
6.1.3	Profilspezifische Objekte (0x6000-0xFFFF) [ab Hardwarestand 03] .....	171
6.2	Control- und Status-Wort.....	173
<b>7</b>	<b>Übersicht CoE Objekte EL6002, EL6022 .....</b>	<b>177</b>
7.1	Objektbeschreibung und Parametrierung .....	177
7.1.1	Objekte für die Inbetriebnahme .....	177
7.1.2	Standardobjekte (0x1000-0x1FFF).....	178
7.1.3	Profilspezifische Objekte (0x6000-0xFFFF) [ab Hardwarestand 03] .....	190
7.2	Control- und Status-Daten .....	193
<b>8</b>	<b>Anhang .....</b>	<b>195</b>
8.1	EtherCAT AL Status Codes .....	195
8.2	Firmware Kompatibilität .....	195
8.3	Firmware Update EL/ES/ELM/EM/EPxxxx .....	196
8.3.1	Gerätebeschreibung ESI-File/XML .....	197
8.3.2	Erläuterungen zur Firmware .....	200
8.3.3	Update Controller-Firmware *.efw .....	201
8.3.4	FPGA-Firmware *.rbf .....	203
8.3.5	Gleichzeitiges Update mehrerer EtherCAT-Geräte .....	207
8.4	Wiederherstellen des Auslieferungszustandes.....	208
8.5	Support und Service .....	209

# 1 Vorwort

## 1.1 Übersicht Serielle Schnittstellenklemmen

[EL6001 \[► 20\]](#) (1 kanalige Serielle Schnittstellenklemme, RS232C)

[EL6021 \[► 20\]](#) (1 kanalige Serielle Schnittstellenklemme, RS422/RS485)

[EL6002 \[► 23\]](#) (2 kanalige Serielle Schnittstellenklemme, RS232C)

[EL6022 \[► 23\]](#) (2 kanalige Serielle Schnittstellenklemme, RS422/RS485)

## 1.2 Hinweise zur Dokumentation

### Zielgruppe

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 G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH. Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

### Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

## 1.3 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 Hinweise

In der vorliegenden Dokumentation werden die folgenden Hinweise verwendet.  
Diese Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

#### **GEFAHR**

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

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

#### **WARNUNG**

##### **Verletzungsgefahr!**

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

#### **VORSICHT**

##### **Schädigung von Personen!**

Wenn dieser Sicherheitshinweis nicht beachtet wird, können Personen geschädigt werden!

#### **HINWEIS**

##### **Schädigung von Umwelt/Geräten oder Datenverlust**

Wenn dieser Hinweis nicht beachtet wird, können Umweltschäden, Gerätebeschädigungen oder Datenverlust entstehen.



##### **Tipp oder Fingerzeig**

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

## 1.4 Ausgabestände der Dokumentation

Version	Kommentar
4.8	<ul style="list-style-type: none"> <li>• Update Kapitel "Grundlagen"</li> <li>• Update Struktur</li> <li>• Update Revisionsstand</li> </ul>
4.7	<ul style="list-style-type: none"> <li>• Update Kapitel "Objektbeschreibung und Parametrierung"</li> <li>• Update Struktur</li> <li>• Update Revisionsstand</li> </ul>
4.6	<ul style="list-style-type: none"> <li>• Korrektur RS232 Pegel</li> <li>• Update Kapitel "Technische Daten"</li> <li>• Update Struktur</li> <li>• Update Revisionsstand</li> </ul>
4.5	<ul style="list-style-type: none"> <li>• Update Kapitel "Inbetriebnahme"</li> <li>• Update Revisionsstand</li> </ul>
4.4	<ul style="list-style-type: none"> <li>• Update Kapitel "Technische Daten"</li> <li>• Kapitel "Betriebsarten und Prozessdaten" aktualisiert</li> <li>• Kapitel "Kommunikationseigenschaften" aktualisiert</li> <li>• Strukturupdate</li> <li>• Update Revisionsstand</li> </ul>
4.3	<ul style="list-style-type: none"> <li>• Update Kapitel "Technische Daten"</li> <li>• Hinweis zum ESD-Schutz eingefügt</li> <li>• Kapitel "TwinCAT Quickstart" hinzugefügt</li> <li>• Kapitel "ATEX - Besondere Bedingungen" ausgetauscht gegen Kapitel "ATEX - Besondere Bedingungen (erweiterter Temperaturbereich)"</li> <li>• Update Revisionsstand</li> </ul>
4.2	<ul style="list-style-type: none"> <li>• Update Kapitel „LEDs und Anschlussbelegung“</li> </ul>
4.1	<ul style="list-style-type: none"> <li>• Ergänzung in Kapitel „LEDs und Anschlussbelegung“</li> </ul>
4.0	<ul style="list-style-type: none"> <li>• Migration und Überarbeitung</li> <li>• Kapitel "Montage und Demontage" im Kapitel "EL6002, EL6022" in "Montage und Verdrahtung" mit "Frontentriegelung oben" ergänzt</li> <li>• Kapitel "Montagevorschriften für Klemmen mit erhöhter mechanischer Belastbarkeit" vom Unterkapitel "EL6001, EL6021" verschoben nach Kapitel "Montage und Verdrahtung"</li> <li>• Kapitel "Einbaulagen" aus Unterkapitel "EL6001, EL6021" entfernt (da bereits im übergeordn. Kapitel vorhanden)</li> <li>• Kapitel "Konfiguration mit dem TwinCAT System Manager" in das Unterkapitel "TwinCAT 2.1x" vom Kapitel "Inbetriebnahme" verschoben</li> <li>• Kapitel "LIN Feature EL6001" zum Kapitel "Inbetriebnahme" hinzugefügt</li> <li>• Kapitel "Beispielprogramm 1" und "Beispielprogramm 2" zusammengefasst in ein neues Kapitel "Beispielprogramme"; neues Kapitel "Beispielprogramme" untergebracht im Kapitel "Inbetriebnahme"</li> <li>• Kapitel "Beispielprogramme" mit Kapitel "Beispielprogramm 3 (LIN)" ergänzt</li> </ul>
3.8	<ul style="list-style-type: none"> <li>• Update Kapitel "Technische Daten"</li> <li>• Kapitel "Montagehinweise bei erhöhter mechanischer Belastbarkeit" ergänzt</li> <li>• Strukturupdate</li> <li>• Update Revisionsstand</li> </ul>

Version	Kommentar
3.7	<ul style="list-style-type: none"> <li>• LED Beschreibung aktualisiert</li> <li>• Revisionsstand aktualisiert</li> </ul>
3.6	<ul style="list-style-type: none"> <li>• Revisionsstand aktualisiert</li> <li>• Strukturupdate</li> </ul>
3.5	<ul style="list-style-type: none"> <li>• Update Kapitel "Technische Daten"</li> <li>• Update Kapitel "Objektbeschreibung und Parametrierung"</li> <li>• Update Kapitel "Kommunikationseigenschaften"</li> <li>• Update Kapitel "Technologie"</li> <li>• Update Kapitel "Prozessdaten"</li> <li>• Strukturupdate</li> </ul>
3.4	<ul style="list-style-type: none"> <li>• Update Kapitel "Technologie"</li> </ul>
3.3	<ul style="list-style-type: none"> <li>• Technische Daten aktualisiert</li> </ul>
3.2	<ul style="list-style-type: none"> <li>• Technische Daten aktualisiert</li> </ul>
3.1	<ul style="list-style-type: none"> <li>• Hinweise und Beschreibung zum Kommando-Mode hinzugefügt</li> </ul>
3.0	<ul style="list-style-type: none"> <li>• Kapitel "Objektbeschreibung" aktualisiert</li> </ul>
2.9	<ul style="list-style-type: none"> <li>• Kapitel "Kommunikationseigenschaften und TcVirtualComDriver" hinzugefügt</li> </ul>
2.8	<ul style="list-style-type: none"> <li>• Aktualisierung Technische Daten</li> </ul>
2.7	<ul style="list-style-type: none"> <li>• Aktualisierung Technische Daten</li> </ul>
2.6	<ul style="list-style-type: none"> <li>• Aktualisierung Kapitel "Technologie" und "Prozessdaten"</li> </ul>
2.5	<ul style="list-style-type: none"> <li>• Aktualisierung Kapitel "Technologie"</li> </ul>
2.4	<ul style="list-style-type: none"> <li>• Objektbeschreibung und Technische Hinweise ergänzt</li> </ul>
2.3	<ul style="list-style-type: none"> <li>• Ergänzung zur Firmware-Kompatibilität, Technische Hinweise ergänzt</li> </ul>
2.2	<ul style="list-style-type: none"> <li>• Ergänzungen</li> </ul>
2.1	<ul style="list-style-type: none"> <li>• Ergänzungen</li> </ul>
2.0	<ul style="list-style-type: none"> <li>• Erste Veröffentlichung</li> </ul>
0.3	<ul style="list-style-type: none"> <li>• Ergänzungen</li> </ul>
0.2	<ul style="list-style-type: none"> <li>• Korrekturen &amp; Ergänzungen</li> </ul>
0.1	<ul style="list-style-type: none"> <li>• Vorläufige Dokumentation für EL60xx</li> </ul>

## 1.5 Versionsidentifikation von EtherCAT-Geräten

### Bezeichnung

Ein Beckhoff EtherCAT-Gerät hat eine 14stellige technische Bezeichnung, die sich zusammensetzt aus

- Familienschlüssel
- Typ
- Version
- Revision

Beispiel	Familie	Typ	Version	Revision
EL3314-0000-0016	EL-Klemme (12 mm, nicht steckbare Anschlussebene)	3314 (4 kanalige Thermoelementklemme)	0000 (Grundtyp)	0016
ES3602-0010-0017	ES-Klemme (12 mm, steckbare Anschlussebene)	3602 (2 kanalige Spannungsmessung)	0010 (Hochpräzise Version)	0017
CU2008-0000-0000	CU-Gerät	2008 (8 Port FastEthernet Switch)	0000 (Grundtyp)	0000

### Hinweise

- die oben genannten Elemente ergeben die **technische Bezeichnung**, im Folgenden wird das Beispiel EL3314-0000-0016 verwendet.
- Davon ist EL3314-0000 die Bestellbezeichnung, umgangssprachlich bei „-0000“ dann oft nur EL3314 genannt. „-0016“ ist die EtherCAT-Revision.
- Die **Bestellbezeichnung** setzt sich zusammen aus
  - Familienschlüssel (EL, EP, CU, ES, KL, CX, ...)
  - Typ (3314)
  - Version (-0000)
- Die **Revision** -0016 gibt den technischen Fortschritt wie z. B. Feature-Erweiterung in Bezug auf die EtherCAT Kommunikation wieder und wird von Beckhoff verwaltet.  
Prinzipiell kann ein Gerät mit höherer Revision ein Gerät mit niedrigerer Revision ersetzen, wenn nicht anders z. B. in der Dokumentation angegeben.  
Jeder Revision zugehörig und gleichbedeutend ist üblicherweise eine Beschreibung (ESI, EtherCAT Slave Information) in Form einer XML-Datei, die zum Download auf der Beckhoff Webseite bereitsteht. Die Revision wird seit 2014/01 außen auf den IP20-Klemmen aufgebracht, siehe Abb. „EL5021 EL-Klemme, Standard IP20-IO-Gerät mit Chargennummer und Revisionskennzeichnung (seit 2014/01)“.
- Typ, Version und Revision werden als dezimale Zahlen gelesen, auch wenn sie technisch hexadezimal gespeichert werden.

### Identifizierungsnummer

Beckhoff EtherCAT Geräte der verschiedenen Linien verfügen über verschiedene Arten von Identifizierungsnummern:

#### Produktionslos/Chargennummer/Batch-Nummer/Seriennummer/Date Code/D-Nummer

Als Seriennummer bezeichnet Beckhoff im IO-Bereich im Allgemeinen die 8-stellige Nummer, die auf dem Gerät aufgedruckt oder auf einem Aufkleber angebracht ist. Diese Seriennummer gibt den Bauzustand im Auslieferungszustand an und kennzeichnet somit eine ganze Produktions-Charge, unterscheidet aber nicht die Module einer Charge.

Aufbau der Seriennummer: **KK YY FF HH**

KK - Produktionswoche (Kalenderwoche)  
 YY - Produktionsjahr  
 FF - Firmware-Stand  
 HH - Hardware-Stand

Beispiel mit

Ser. Nr.: 12063A02: 12 - Produktionswoche 12 06 - Produktionsjahr 2006 3A - Firmware-Stand 3A 02 - Hardware-Stand 02

Ausnahmen können im **IP67-Bereich** auftreten, dort kann folgende Syntax verwendet werden (siehe jeweilige Gerätedokumentation):

Syntax: D ww yy x y z u

D - Vorsatzbezeichnung

ww - Kalenderwoche

yy - Jahr

x - Firmware-Stand der Busplatine

y - Hardware-Stand der Busplatine

z - Firmware-Stand der E/A-Platine

u - Hardware-Stand der E/A-Platine

Beispiel: D.22081501 Kalenderwoche 22 des Jahres 2008 Firmware-Stand Busplatine: 1 Hardware Stand Busplatine: 5 Firmware-Stand E/A-Platine: 0 (keine Firmware für diese Platine notwendig) Hardware-Stand E/A-Platine: 1

### Eindeutige Seriennummer/ID, ID-Nummer

Darüber hinaus verfügt in einigen Serien jedes einzelne Modul über eine eindeutige Seriennummer.

Siehe dazu auch weiterführende Dokumentation im Bereich

- IP67: [EtherCAT Box](#)
- Safety: [TwinSafe](#)
- Klemmen mit Werkskalibrierzertifikat und andere Messtechnische Klemmen

### Beispiele für Kennzeichnungen



Abb. 1: EL5021 EL-Klemme, Standard IP20-IO-Gerät mit Seriennummer/ Chargennummer und Revisionskennzeichnung (seit 2014/01)



Abb. 2: EK1100 EtherCAT Koppler, Standard IP20-IO-Gerät mit Seriennummer/ Chargennummer

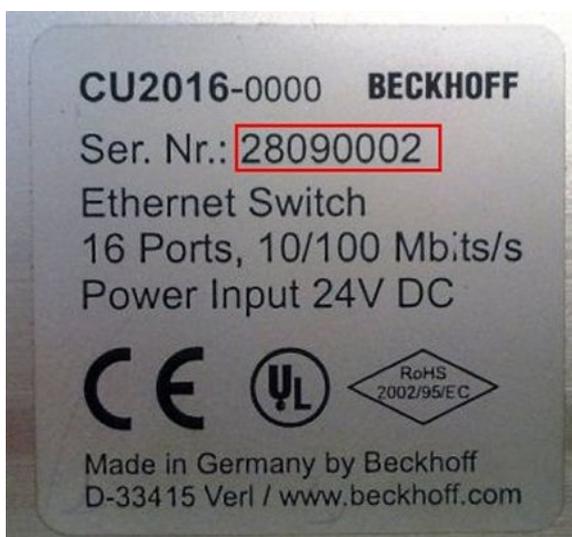


Abb. 3: CU2016 Switch mit Seriennummer/ Chargennummer

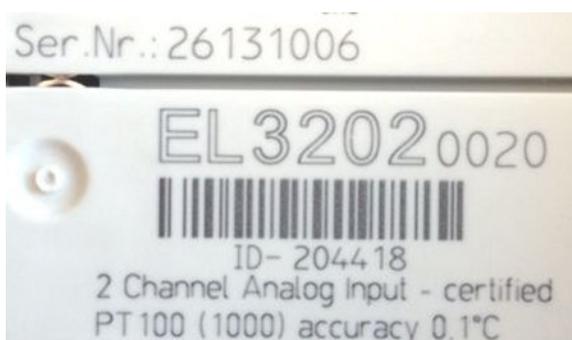


Abb. 4: EL3202-0020 mit Seriennummer/ Chargennummer 26131006 und eindeutiger ID-Nummer 204418



Abb. 5: EP1258-00001 IP67 EtherCAT Box mit Chargennummer/ DateCode 22090101 und eindeutiger Seriennummer 158102



Abb. 6: EP1908-0002 IP67 EtherCAT Safety Box mit Chargennummer/ DateCode 071201FF und eindeutiger Seriennummer 00346070



Abb. 7: EL2904 IP20 Safety Klemme mit Chargennummer/ DateCode 50110302 und eindeutiger Seriennummer 00331701



Abb. 8: ELM3604-0002 Klemme mit eindeutiger ID-Nummer (QR Code) 100001051 und Seriennummer/ Chargennummer 44160201

## 1.5.1 Beckhoff Identification Code (BIC)

Der Beckhoff Identification Code (BIC) wird vermehrt auf Beckhoff-Produkten zur eindeutigen Identitätsbestimmung des Produkts aufgebracht. Der BIC ist als Data Matrix Code (DMC, Code-Schema ECC200) dargestellt, der Inhalt orientiert sich am ANSI-Standard MH10.8.2-2016.

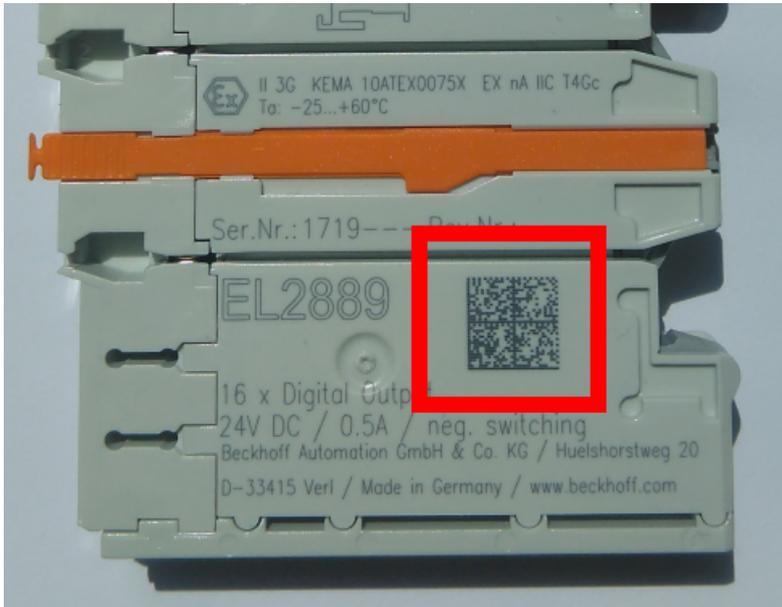


Abb. 9: BIC als Data Matrix Code (DMC, Code-Schema ECC200)

Die Einführung des BIC erfolgt schrittweise über alle Produktgruppen hinweg. Er ist je nach Produkt an folgenden Stellen zu finden:

- auf der Verpackungseinheit
- direkt auf dem Produkt (bei ausreichendem Platz)
- auf Verpackungseinheit und Produkt

Der BIC ist maschinenlesbar und enthält Informationen, die auch kundenseitig für Handling und Produktverwaltung genutzt werden können.

Jede Information ist anhand des so genannten Datenidentifikators (ANSI MH10.8.2-2016) eindeutig identifizierbar. Dem Datenidentifikator folgt eine Zeichenkette. Beide zusammen haben eine maximale Länge gemäß nachstehender Tabelle. Sind die Informationen kürzer, werden sie um Leerzeichen ergänzt. Die Daten unter den Positionen 1 bis 4 sind immer vorhanden.

Folgende Informationen sind enthalten:

Pos-Nr.	Art der Information	Erklärung	Datenidentifikator	Anzahl Stellen inkl. Datenidentifikator	Beispiel
1	Beckhoff-Artikelnummer	<b>Beckhoff - Artikelnummer</b>	1P	8	<b>1P</b> 072222
2	Beckhoff Traceability Number (BTN)	<b>Eindeutige Seriennummer, Hinweis s. u.</b>	S	12	<b>S</b> BTNk4p562d7
3	Artikelbezeichnung	<b>Beckhoff Artikelbezeichnung, z. B. EL1008</b>	1K	32	<b>1K</b> EL1809
4	Menge	<b>Menge in Verpackungseinheit, z. B. 1, 10...</b>	Q	6	<b>Q</b> 1
5	Chargennummer	Optional: Produktionsjahr und -woche	2P	14	<b>2P</b> 401503180016
6	ID-/Seriennummer	Optional: vorheriges Seriennummer-System, z. B. bei Safety-Produkten oder kalibrierten Klemmen	51S	12	<b>51S</b> 678294104
7	Variante	Optional: Produktvarianten-Nummer auf Basis von Standardprodukten	30P	32	<b>30P</b> F971, 2*K183
...					

Weitere Informationsarten und Datenidentifikatoren werden von Beckhoff verwendet und dienen internen Prozessen.

**Aufbau des BIC**

Beispiel einer zusammengesetzten Information aus den Positionen 1 bis 4 und 6. Die Datenidentifikatoren sind zur besseren Darstellung jeweils rot markiert:

**BTN**

Ein wichtiger Bestandteil des BICs ist die Beckhoff Traceability Number (BTN, Pos.-Nr. 2). Die BTN ist eine eindeutige, aus acht Zeichen bestehende Seriennummer, die langfristig alle anderen Seriennummern-Systeme bei Beckhoff ersetzen wird (z. B. Chargenbezeichnungen auf IO-Komponenten, bisheriger Seriennummernkreis für Safety-Produkte, etc.). Die BTN wird ebenfalls schrittweise eingeführt, somit kann es vorkommen, dass die BTN noch nicht im BIC codiert ist.

<b>HINWEIS</b>
Diese Information wurde sorgfältig erstellt. Das beschriebene Verfahren wird jedoch ständig weiterentwickelt. Wir behalten uns das Recht vor, Verfahren und Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern. Aus den Angaben, Abbildungen und Beschreibungen in dieser Information können keine Ansprüche auf Änderung geltend gemacht werden.

## 2 Produktübersicht

### Technologie

Die seriellen Schnittstellenklemmen EL600x, EL602x ermöglichen den Anschluss von Geräten mit einer RS232 (bzw. RS485 / RS422) -Schnittstelle. Bei der EL600x werden die Daten im Vollduplexbetrieb mit der Steuerung ausgetauscht, bei der EL602x ist zusätzlich noch der Halbduplexbetrieb möglich. Die Klemme verfügt über einen Empfangs/Sendepuffer je Kanal, siehe technische Daten. Der Datentransfer zwischen Klemme und Steuerung wird über einen Handshake abgewickelt.

Die Werkseinstellung der Klemmen ist

- 9600 Baud
- 8N1: 8 Datenbits, 1 Stopp-Bit, no Parity
- bei der EL600x ist die RTS/CTS-Kontrolle aktiv
- die EL602x arbeitet im Vollduplex-Modus mit deaktivierter Punkt-zu-Punkt Verbindung.

### Grundlagen

Beim Übertragen von mehreren Bytes an Daten werden die Daten (insgesamt x Bytes respektive 8\*x Bits) nach der Codierungsvorschrift (beispielsweise 7E2 oder 8N1) in einzelnen Telegrammen zu je 7 resp. 8 Bits verschickt. Dabei besteht ein Telegramm aus

- StartBit
- Datenbits (7 oder 8, begonnen wird mit dem LSB [least signifikant bit])
- Optional: Parity-Bit
  - "E" EVEN: Paritätsbit wird vom Sender so gesetzt, dass gerade Parität hergestellt wird
  - "O" ODD: Paritätsbit wird vom Sender so gesetzt, dass ungerade Parität hergestellt wird
  - "N" NOT: kein Paritätsbit
  - "M" MARK: Paritätsbit wird vom Sender = 1 gesetzt
  - "S" SPACE: Paritätsbit wird vom Sender = 0 gesetzt
- StopBit (1 oder 2)

Entsprechend bedeutet die Codierungsvorschrift 8N1: 8 Datenbits, kein Paritätsbit, 1 Stopbit.

Wenn eine 7-Bit-Codierung ausgewählt wird, werden von jedem Datenbyte, das von der PLC an die Klemme über die zyklischen Prozessdaten übermittelt wird, nur die unteren 7 Bit versendet. D.h. werden 10 Byte Daten (Umfang 8 Bit) an die EL60xx geschickt, versendet diese 10 Telegramme zu je 7 Bit.

Wenn eine 9-Bit-Codierung ausgewählt wird, ist das 16-Bit-Prozessdateninterface zu wählen. Die Klemme erwartet dann die 9 Nutzbits in den unteren 9 Bits des 16-Bit-words.

### Frequenz

Die Frequenz der Datenübertragung muß in Sender und Empfänger bekannt sein und bis auf einige % übereinstimmen. Nur dadurch kann der Empfänger die Pegelwechsel auf der Leitung korrekt erkennen.

### Handshake

Ein zusätzlicher Handshake zwischen Sender und Empfänger kann eingesetzt werden, damit der Empfänger dem Sender mitteilen kann ob er empfangsbereit ist. Die EL60xx unterstützen 2 Arten von Handshake:

- über besondere Datenleitungen RTS/CTS
  - Die Aktivierung dieses Features ist im CoE erforderlich.
  - Nur bei EL6001/EL6002 möglich.
- über besondere Datentelegramme
  - Die Aktivierung dieses Features ist im CoE erforderlich.

**RS-Standards**

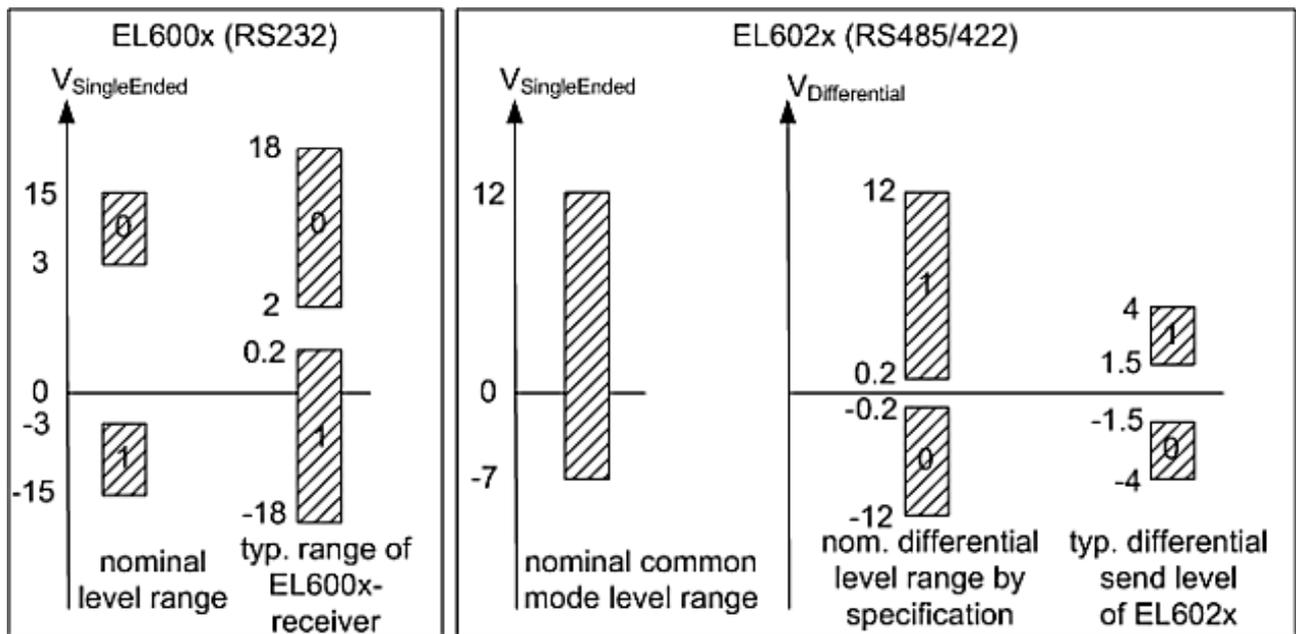
Der Begriff RS232 (RS-232) wird in diesem Dokument kurz für den Standard ANSI/EIA/TIA-232-F verwendet.

Der Begriff RS422 (RS-422) wird in diesem Dokument kurz für den Standard ITU-T V.11 nach ANSI/TIA/EIA-422-B-1994 verwendet.

Der Begriff RS485 (RS-485) wird in diesem Dokument kurz für den Standard ANSI/TIA/EIA-485-A-98 verwendet.

**Pegel Schnittstellen**

Die EL6001/6002 arbeiten auf RS232-Pegel gegen GND-Bezug, die EL6021/6022 mit differentiellen RS485/422-Pegel.



voltages on wire depends on load and cabling

Abb. 10: Pegel Schnittstellen RS232, RS485/422

**Terminierung und Topologie**

Die seriellen Kommunikationstechnologien RS422 und RS485 arbeiten mit Spannungspegeln auf einer 2-Draht-Leitung. Durch Reflexionen an hochohmigen Leitungsenden können Signale verfälscht werden. Deshalb werden Abschlusswiderstände als Terminierung am Receiver benötigt. Bei RS422/485 sind dies 120 Ω-Widerstände, die zusammen mit dem Leitungswiderstand zum Spannungsabfall über die Übertragungsstrecke führen.

**Zulässige Leitungslänge**

**I** Der Leitungswiderstand führt mit dem Terminierungswiderstand zum Gesamtspannungsabfall über die Übertragungsstrecke. Eine unzulässig hohe Anzahl von Terminierungswiderständen dämpft das Signal zu stark!

Bei der Streckenauslegung ist darauf zu achten, dass am Receiver noch mindestens die erforderlichen 200 mV (s. Abb. ) anliegen können.

Im RS422-Betrieb muss jede der Leitungen am Receiver mit 120 Ω terminiert werden.

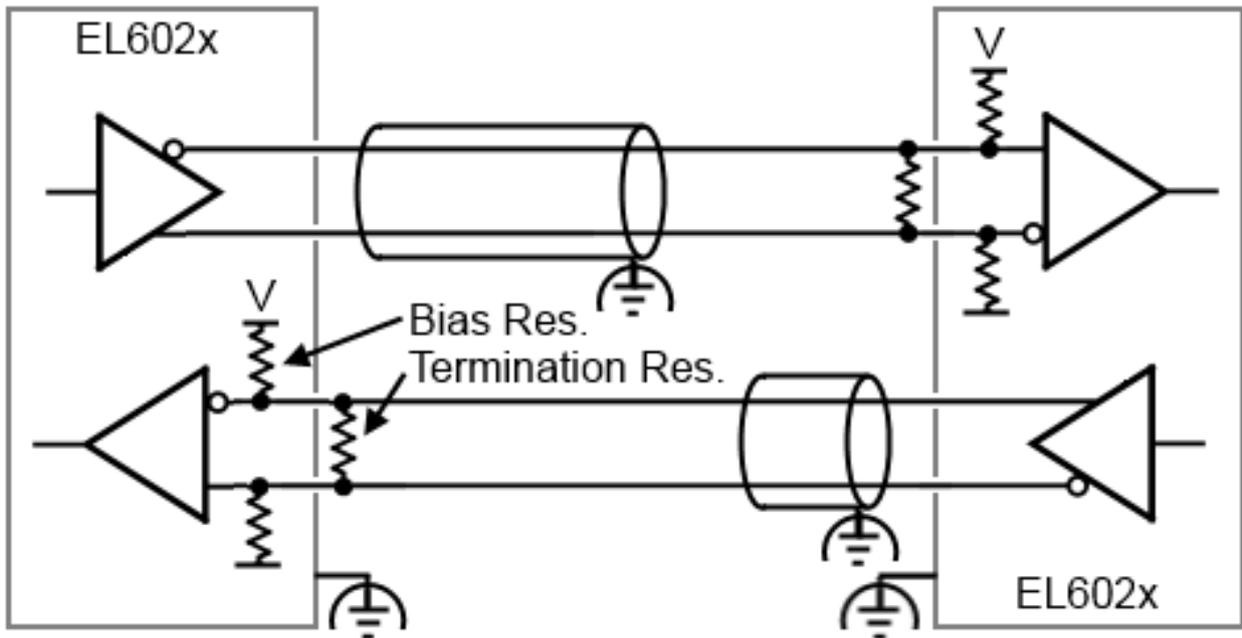


Abb. 11: Terminierung RS422

Im RS485-Betrieb mit mehreren Teilnehmern werden nur an den beiden äußeren Teilnehmern Terminierungswiderstände eingesetzt.

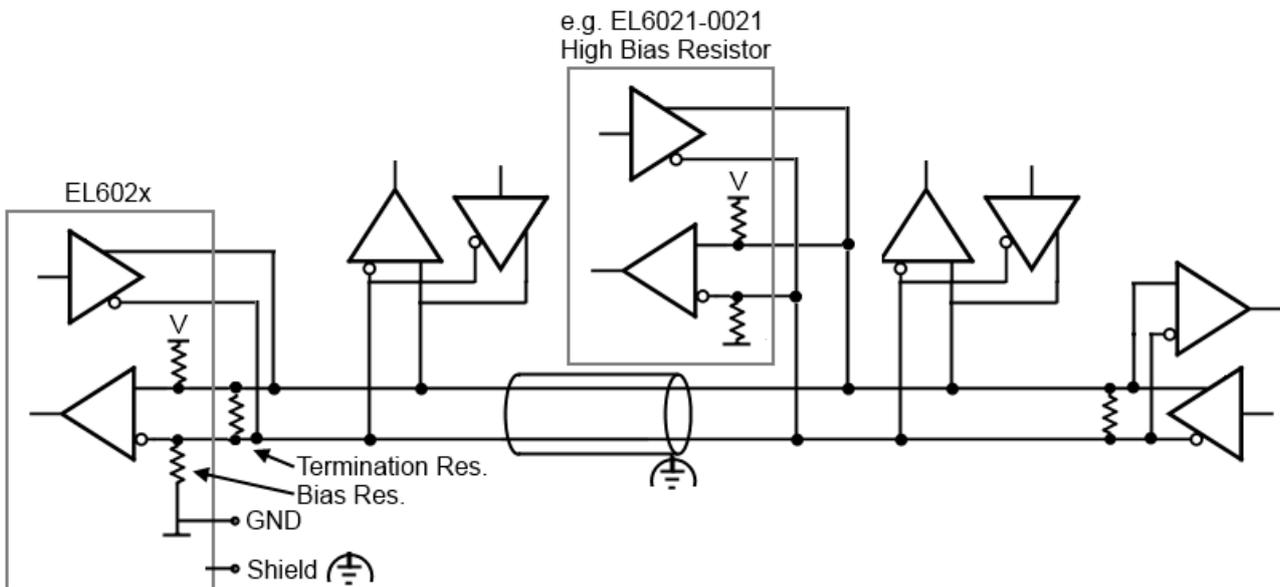


Abb. 12: Terminierung RS485

Hintergrund ist die unterschiedliche Konzeption von RS422/EIA-422 bzw. RS485/EIA-485:

- RS422: 1 Rx → n Tx (maximal 10 Empfänger)
- RS485: n Rx → m Tx (maximal 32/128 Teilnehmer, abhängig von resultierenden Busbelastung)

Komponenten für RS485 verfügen in der Regel u.a. über eine höhere Eingangsimpedanz, belasten also den Bus weniger.

**i Terminierung mit EL602x und Bias-Widerständen**

Die EL602x verfügen über keine integrierten Terminierungswiderstände, damit sie auch im Busbetrieb eingesetzt werden können. Ggf. erforderliche Terminierung muss außerhalb der Klemme angeschlossen werden.

Die EL602x verfügen über integrierte Bias-Widerstände < 1 kOhm, die die Busleitungen auch bei getrennter Leitung auf definierte Pegel ziehen.

Werden mehrere EL602x in einem Bus miteinander verbunden, können die parallelen Bias-Widerstände die Datenübertragung erschweren. Dann ist als Zwischengerät die EL6021-0021 zu verwenden - diese verfügt über deutlich größere Bias-Widerstände.

**Topologie**

Die Terminierung und Bias-Widerstände belasten den Bus. Jedoch sind sie für eindeutige Buspegel unerlässlich. Deshalb sind sie mit Sorgfalt zu platzieren. Idealerweise ist der RS422/485-Bus als DaisyChain oder Kette aufgebaut, vgl. [Abb. \[▶ 19\]](#) Problematisch können sein:

- Sterntopologien: jeder Endpunkt sollte eigentlich terminiert werden, dies kann jedoch zu hoher Busbelastung und uneindeutigen Pegeln führen. Außerdem ist mit Reflexionen und Laufzeitschwankungen zu rechnen.
- vermaschte Topologien: keine eindeutigen Endpunkte, Reflexionen und Kreisströme möglich

**Schirmung/Shield**

**HINWEIS**

**Funktionserde nicht zum Ableiten von Fehlerströmen oder Potentialdifferenzen vorsehen!**

Die EL60xx bieten einen Schirmanschluss zum Ableiten von EMV-Einflüssen über den Kabelschirm (FE, Funktionserde). Der Schirm darf nicht zum Ableiten von Fehlerströmen oder Potentialdifferenzen missbraucht werden.

Die EL60xx bieten einen Schirmanschluss zum Ableiten von EMV-Einflüssen über den Kabelschirm (FE, Funktionserde). Der Schirm darf nicht zum Ableiten von Fehlerströmen oder Potentialdifferenzen missbraucht werden.

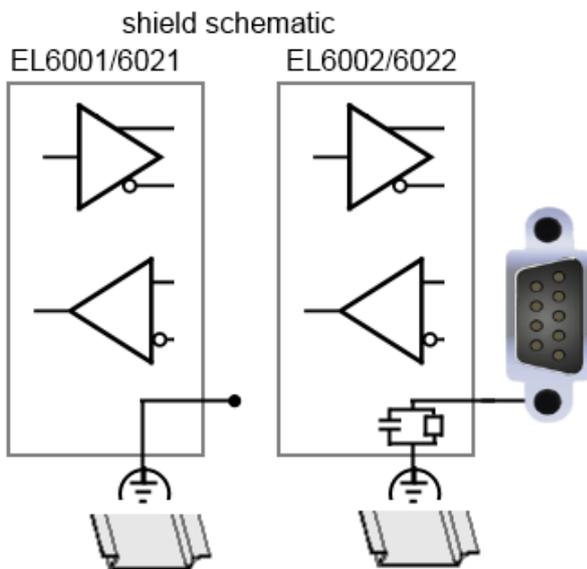
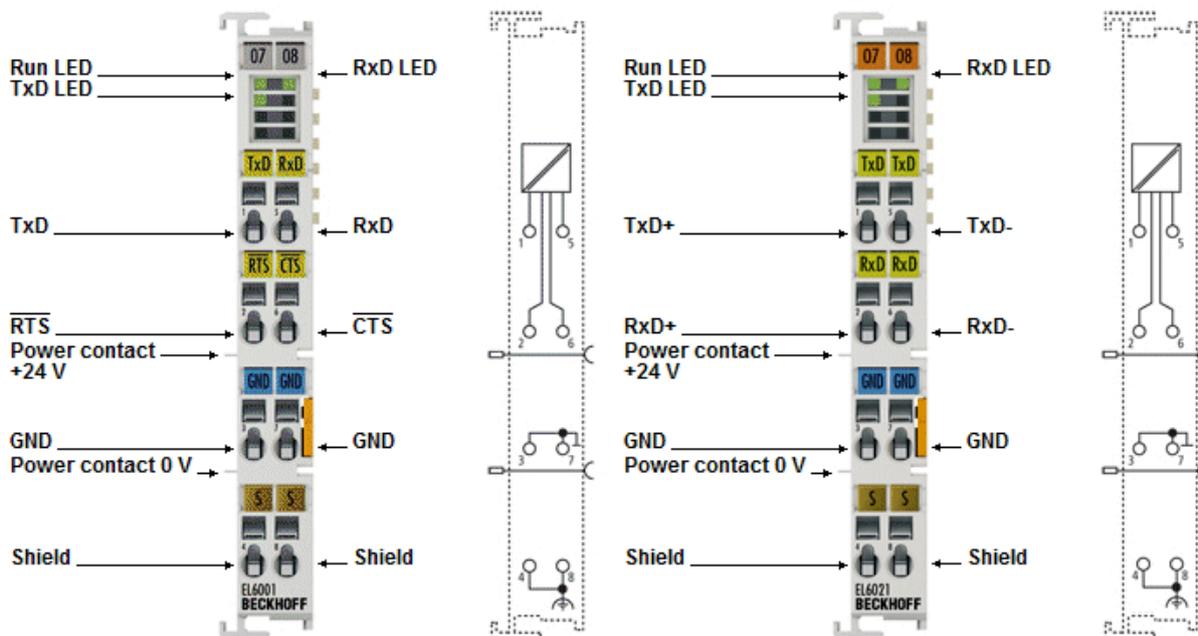


Abb. 13: Schirmanschluss in EL60xx

In den 2-kanaligen Varianten ist der D-Sub9-Schirm über eine hochohmige RC-Kombination mit der Montageschiene verbunden.

## 2.1 EL6001, EL6021

### 2.1.1 Einführung



#### Serielle Schnittstellen-Klemmen (RS232C/RS422/RS485), 1 Kanal

Die seriellen Schnittstellen EL6001 und EL6021 ermöglichen den Anschluss von Geräten mit einer RS232-Schnittstelle bzw. RS422/RS485 Schnittstelle. Die EL6001 arbeitet normkonform nach CCITT V.28/DIN 66 259-1.

Das an die EtherCAT-Klemmen EL6001/ EL6021 angeschlossene Gerät kommuniziert über den Koppler mit dem Automatisierungsgerät. Der aktive Kommunikationskanal arbeitet unabhängig vom überlagerten Bussystem im Vollduplexbetrieb bzw. auch wahlweise Halbduplexbetrieb (EL6021) mit bis zu 115,2 kBaud. Die RS232-Schnittstelle garantiert hohe Störsicherheit durch galvanisch getrennte Signale, was bei der EL6021 zusätzlich durch die Differenzsignalübertragung nach RS422 garantiert wird.

In Verbindung mit dem TwinCAT Virtual Serial COM Driver (siehe TwinCAT Supplements – Communication) kann die EL6001/ EL6021 als normale Windows-COM-Schnittstelle genutzt werden.

#### Quick Links

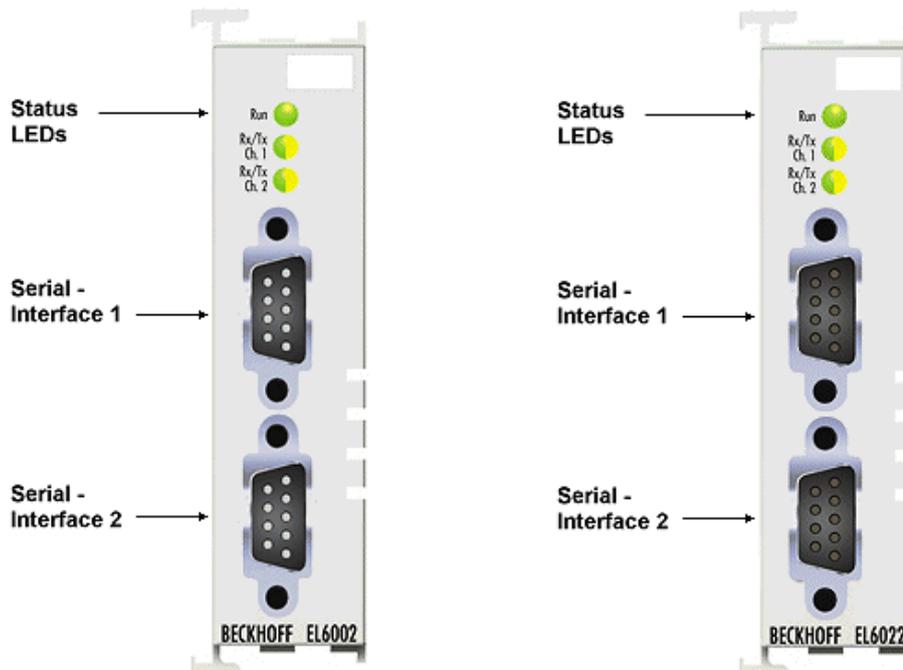
- [EtherCAT Funktionsgrundlagen \[► 25\]](#)
- [Technologie Serielle Schnittstellen Klemmen \[► 16\]](#)
- [Inbetriebnahme \[► 60\]](#)
- [Prozessdaten, Allgemeine Hinweise \[► 128\]](#)
- [CoE-Objektbeschreibung und Parametrierung EL60x1 \[► 151\]](#)
- [Control- und Status Daten EL60x1 \[► 173\]](#)

**2.1.2 Technische Daten**

Technische Daten	EL6001	EL6021
Übertragungskanäle	TxD und RxD, voll duplex	TxD und RxD, voll-/halbduplex
Übertragungsrate	2400...115200 Baud, default: 9600 Baud, 8 Datenbits, keine Parity, 1 Stopbit ab <a href="#">Firmware 07</a> [ <a href="#">► 195</a> ]: auch 12000 Baud und 14400 Baud  ab <a href="#">Firmware 11</a> [ <a href="#">► 195</a> ]: jede ganzzahlige Baudrate 1000 ... 115200	2400...115200 Baud, default: 9600 Baud, 8 Datenbits, keine Parity, 1 Stopbit
Datenpuffer	864 Byte Empfangspuffer, 128 Byte Sendepuffer EL6001 ab FW08: 250 Byte Sendepuffer	
Bitübertragung	-	mit Differenzsignal
Pegel Schnittstelle	RS232	RS485/422
Bitverzerrung	< 3 %	-
Leitungslänge	max. 15 m	max. 1000 m (Twisted Pair)
Leitungsimpedanz	-	120 Ω
Bereitstellung externe Versorgung	-	-
Diagnose	Status-LEDs	
Spannungsversorgung	über den E-Bus	
Stromaufnahme aus dem E-Bus	typ. 120 mA	typ. 170 mA (im Kurzschlussfall: typ. 250 mA)
Potenzialtrennung	500 V (E-Bus/RS232C)	500 V (E-Bus/RS422, E-Bus/RS485)
Bitbreite im Prozessabbild	1 x 8 Bit Control/Status, Inputs/Outputs: 3 x 8 Bit Nutzdaten oder 1 x 8 Bit Control/Status, Inputs/Outputs: 5 x 8 Bit Nutzdaten oder 1 x 16 Bit Control/Status, Inputs/Outputs: 22 x 8 Bit Nutzdaten (konfigurierbar)	
Konfiguration	keine Adresseinstellung erforderlich Konfiguration über TwinCAT System Manager	
Gewicht	ca. 55 g	
zulässiger Umgebungstemperaturbereich im Betrieb	-25°C ... +60°C (erweiterter Temperaturbereich)	
zulässiger Umgebungstemperaturbereich bei Lagerung	-40°C ... +85°C	
zulässige relative Luftfeuchtigkeit	95%, keine Betauung	
Abmessungen (B x H x T)	ca. 15 mm x 100 mm x 70 mm (Breite angereicht: 12 mm)	
Montage	auf 35 mm Tragschiene nach EN 60715	
Vibrations- / Schockfestigkeit	gemäß EN 60068-2-6 / EN 60068-2-27, siehe auch <a href="#">Montagevorschriften</a> [ <a href="#">► 54</a> ] für Klemmen mit erhöhter mechanischer Belastbarkeit	
EMV-Festigkeit / Aussendung	gemäß EN 61000-6-2 / EN 61000-6-4	
Schutzart	IP20	
Einbaulage	beliebig	
Zulassung	CE <a href="#">ATEX</a> [ <a href="#">► 58</a> ] <a href="#">cULus</a> [ <a href="#">► 57</a> ] EAC	CE <a href="#">ATEX</a> [ <a href="#">► 58</a> ] <a href="#">cULus</a> [ <a href="#">► 57</a> ] IECEX EAC

## 2.2 EL6002, EL6022

### 2.2.1 Einführung



#### Serielle Schnittstellen-Klemme (RS232/RS422/RS485), 2 Kanäle

Die seriellen Schnittstellen EL6002 und EL6022 ermöglichen den Anschluss von Geräten mit zwei RS232- bzw. zwei RS422/RS485-Schnittstellen an jeweils einem D-Sub Anschluss (9 polig). Die Schnittstellen sind zum EtherCAT und untereinander galvanisch getrennt.

Die an die EtherCAT-Klemmen EL6002/EL6022 angeschlossenen Geräte kommunizieren über den Koppler mit dem Automatisierungsgerät. Der aktive Kommunikationskanal arbeitet unabhängig vom überlagerten EtherCAT-System im Vollduplexbetrieb mit 300 Baud bis zu 115,2 kBaud.

Die RS232/RS422/RS485-Schnittstellen garantieren hohe Störsicherheit durch galvanisch getrennte Signale. Die EL6022 kann 2 x 5 V/20 mA aus der E-bus-Versorgung (galvanisch getrennt, kurzschlussfest) zur Speisung externer Geräte bereitstellen.

In Verbindung mit dem TwinCAT Virtual Serial COM Driver kann die EL60xx als normale Windows-COM-Schnittstelle genutzt werden.

#### Quick Links

- [EtherCAT Funktionsgrundlagen \[► 25\]](#)
- [Technologie Serielle Schnittstellen Klemmen \[► 16\]](#)
- [Inbetriebnahme \[► 60\]](#)
- [Prozessdaten, Allgemeine Hinweise \[► 128\]](#)
- [CoE-Objektbeschreibung und Parametrierung EL60x2 \[► 177\]](#)
- [Control- und Status Daten EL60x2 \[► 193\]](#)

## 2.2.2 Technische Daten

Technische Daten	EL6002	EL6022
Übertragungskanäle	2, TxD und RxD, voll duplex	2, TxD und RxD, voll-/halb duplex
Anschluss	2 x D-Sub Stecker (DE9), 9 polig	2 x D-Sub Buchse (DE9), 9 polig
Übertragungsrate	300...115200 Baud default: 9600 Baud, 8 Datenbits, keine Parity, 1 Stoppbit	
Datenpuffer	864 Byte Empfangspuffer, 128 Byte Sendepuffer je Kanal	
Pegel Schnittstelle	RS232	RS485/422
Leitungslänge	max. 15 m	max. 1000 m (Twisted Pair)
Bereitstellung externe Versorgung	-	2x typ. 5 V ( $\pm 20\%$ ), aus E-Bus-Versorgung (galv. getrennt), max. 20 mA, kurzschlussfest
Diagnose	Status-LEDs	
Spannungsversorgung	über den E-Bus	
Stromaufnahme aus dem E-Bus	typ. 170 mA	typ. 250 mA (im Kurzschlussfall: typ. 250 mA)
Potenzialtrennung	500 V (E-Bus/RS232C)	500 V (E-Bus/RS422, E-Bus/RS485)
Bitbreite im Prozessabbild	1 x 16 Bit Control/Status, Inputs/Outputs: 22 x 8 Bit Nutzdaten	
Konfiguration	keine Adresseinstellung erforderlich Konfiguration über TwinCAT System Manager	
zulässiger Umgebungstemperaturbereich im Betrieb	-25°C ... +60°C (erweiterter Temperaturbereich)	
zulässiger Umgebungstemperaturbereich bei Lagerung	-40°C ... +85°C	
zulässige relative Luftfeuchtigkeit	95%, keine Betauung	
Gewicht	ca. 70 g	
Abmessungen (B x H x T)	ca. 26 mm x 100 mm x 52 mm (Breite angereiht: 23 mm)	
Montage <a href="#">[► 47]</a>	auf 35 mm Tragschiene nach EN 60715	
Vibrations- / Schockfestigkeit	gemäß EN 60068-2-6 / EN 60068-2-27	
EMV-Festigkeit / Aussendung	gemäß EN 61000-6-2 / EN 61000-6-4	
Schutzart	IP20	
Einbaulage	beliebig	
Zulassung	CE ATEX <a href="#">[► 58]</a> cULus <a href="#">[► 57]</a> EAC	

## 2.3 Start Up

### Start

Zur Inbetriebsetzung:

- montieren Sie den EL600x / EL602x wie im Kapitel [Montage und Verdrahtung \[► 36\]](#) beschrieben
- konfigurieren Sie den EL600x / EL602x in TwinCAT wie im Kapitel [Inbetriebnahme \[► 60\]](#) beschrieben.

### 3 Grundlagen der Kommunikation

#### 3.1 EtherCAT-Grundlagen

Grundlagen zum Feldbus EtherCAT entnehmen Sie bitte der [EtherCAT System-Dokumentation](#).

#### 3.2 EtherCAT-Verkabelung - Drahtgebunden

Die zulässige Leitungslänge zwischen zwei EtherCAT-Geräten darf maximal 100 Meter betragen. Dies resultiert aus der FastEthernet-Technologie, die vor allem aus Gründen der Signaldämpfung über die Leitungslänge eine maximale Linklänge von 5 + 90 + 5 m erlaubt, wenn Leitungen mit entsprechenden Eigenschaften verwendet werden. Siehe dazu auch die [Auslegungsempfehlungen zur Infrastruktur für EtherCAT/Ethernet](#).

##### Kabel und Steckverbinder

Verwenden Sie zur Verbindung von EtherCAT-Geräten nur Ethernet-Verbindungen (Kabel + Stecker), die mindestens der Kategorie 5 (CAT5) nach EN 50173 bzw. ISO/IEC 11801 entsprechen. EtherCAT nutzt 4 Adern des Kabels für die Signalübertragung.

EtherCAT verwendet beispielsweise RJ45-Steckverbinder. Die Kontaktbelegung ist zum Ethernet-Standard (ISO/IEC 8802-3) kompatibel.

Pin	Aderfarbe	Signal	Beschreibung
1	gelb	TD+	Transmission Data +
2	orange	TD-	Transmission Data -
3	weiß	RD+	Receiver Data +
6	blau	RD-	Receiver Data -

Aufgrund der automatischen Kabelerkennung (Auto-Crossing) können Sie zwischen EtherCAT-Geräten von Beckhoff sowohl symmetrisch (1:1) belegte als auch Cross-Over-Kabel verwenden.

##### ● Empfohlene Kabel

- i** Es wird empfohlen die entsprechenden Beckhoff Komponenten zu verwenden, z. B.
- Kabelsätze ZK1090-9191-xxxx bzw.
  - feldkonfektionierbare RJ45 Stecker ZS1090-0005
  - feldkonfektionierbare Ethernet Leitung ZB9010, ZB9020

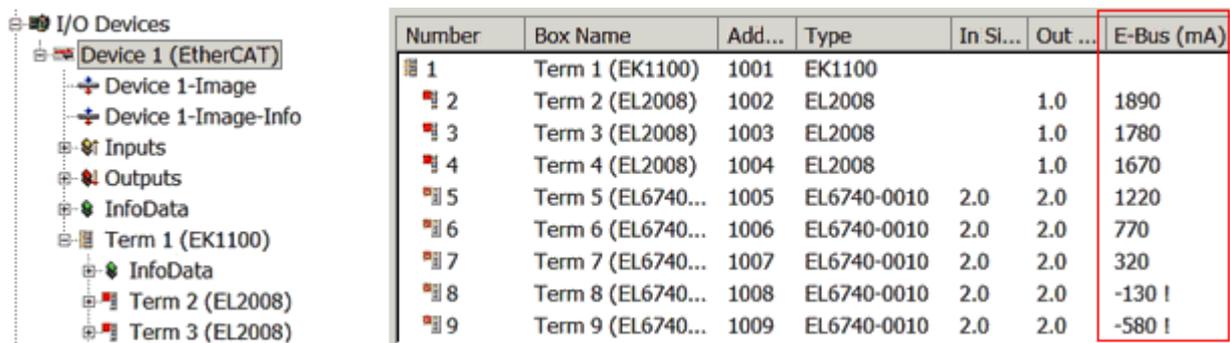
Geeignete Kabel zur Verbindung von EtherCAT-Geräten finden Sie auf der [Beckhoff Website!](#)

##### E-Bus-Versorgung

Ein Buskoppler kann die an ihm angefügten EL-Klemmen mit der E-Bus-Systemspannung von 5 V versorgen, in der Regel ist ein Koppler dabei bis zu 2 A belastbar (siehe Dokumentation des jeweiligen Gerätes).

Zu jeder EL-Klemme ist die Information, wie viel Strom sie aus der E-Bus-Versorgung benötigt, online und im Katalog verfügbar. Benötigen die angefügten Klemmen mehr Strom als der Koppler liefern kann, sind an entsprechender Position im Klemmenstrang Einspeiseklemmen (z.B. [EL9410](#)) zu setzen.

Im TwinCAT System Manager wird der vorberechnete theoretische maximale E-Bus-Strom angezeigt. Eine Unterschreitung wird durch negativen Summenbetrag und Ausrufezeichen markiert, vor einer solchen Stelle ist eine Einspeiseklemme zu setzen.



Number	Box Name	Add...	Type	In Si...	Out ...	E-Bus (mA)
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL2008)	1002	EL2008		1.0	1890
3	Term 3 (EL2008)	1003	EL2008		1.0	1780
4	Term 4 (EL2008)	1004	EL2008		1.0	1670
5	Term 5 (EL6740...)	1005	EL6740-0010	2.0	2.0	1220
6	Term 6 (EL6740...)	1006	EL6740-0010	2.0	2.0	770
7	Term 7 (EL6740...)	1007	EL6740-0010	2.0	2.0	320
8	Term 8 (EL6740...)	1008	EL6740-0010	2.0	2.0	-130 I
9	Term 9 (EL6740...)	1009	EL6740-0010	2.0	2.0	-580 I

Abb. 14: System Manager Stromberechnung

### HINWEIS

#### Fehlfunktion möglich!

Die E-Bus-Versorgung aller EtherCAT-Klemmen eines Klemmenblocks muss aus demselben Massepotential erfolgen!

## 3.3 Allgemeine Hinweise zur Watchdog-Einstellung

Die ELxxxx Klemmen sind mit einer Sicherungseinrichtung (Watchdog) ausgestattet, die z. B. bei unterbrochenem Prozessdatenverkehr nach einer voreinstellbaren Zeit die Ausgänge in einen sicheren Zustand schaltet, in Abhängigkeit vom Gerät und Einstellung z. B. auf AUS.

Der EtherCAT Slave Controller (ESC) verfügt dazu über zwei Watchdogs:

- SM-Watchdog (default: 100 ms)
- PDI-Watchdog (default: 100 ms)

#### SM-Watchdog (SyncManagerWatchdog)

Der SyncManager-Watchdog wird bei jeder erfolgreichen EtherCAT-Prozessdaten-Kommunikation mit der Klemme zurückgesetzt. Findet z. B. durch eine Leitungsunterbrechung länger als die eingestellte und aktivierte SM-Watchdog-Zeit keine EtherCAT-Prozessdaten-Kommunikation mit der Klemme statt, löst der Watchdog aus und setzt die Ausgänge auf FALSE. Der OP-Status der Klemme bleibt davon unberührt. Der Watchdog wird erst wieder durch einen erfolgreichen EtherCAT-Prozessdatenzugriff zurückgesetzt. Die Überwachungszeit ist nach unten genanntem Verfahren einzustellen.

Der SyncManager-Watchdog ist also eine Überwachung auf korrekte und rechtzeitige Prozessdatenkommunikation mit dem ESC von der EtherCAT-Seite aus betrachtet.

#### PDI-Watchdog (Process Data Watchdog)

Findet länger als die eingestellte und aktivierte PDI-Watchdog-Zeit keine PDI-Kommunikation mit dem EtherCAT Slave Controller (ESC) statt, löst dieser Watchdog aus.

PDI (Process Data Interface) ist die interne Schnittstelle des ESC, z. B. zu lokalen Prozessoren im EtherCAT Slave. Mit dem PDI-Watchdog kann diese Kommunikation auf Ausfall überwacht werden.

Der PDI-Watchdog ist also eine Überwachung auf korrekte und rechtzeitige Prozessdatenkommunikation mit dem ESC, aber von der Applikations-Seite aus betrachtet.

Die Einstellungen für SM- und PDI-Watchdog sind im TwinCAT System Manager für jeden Slave gesondert vorzunehmen:

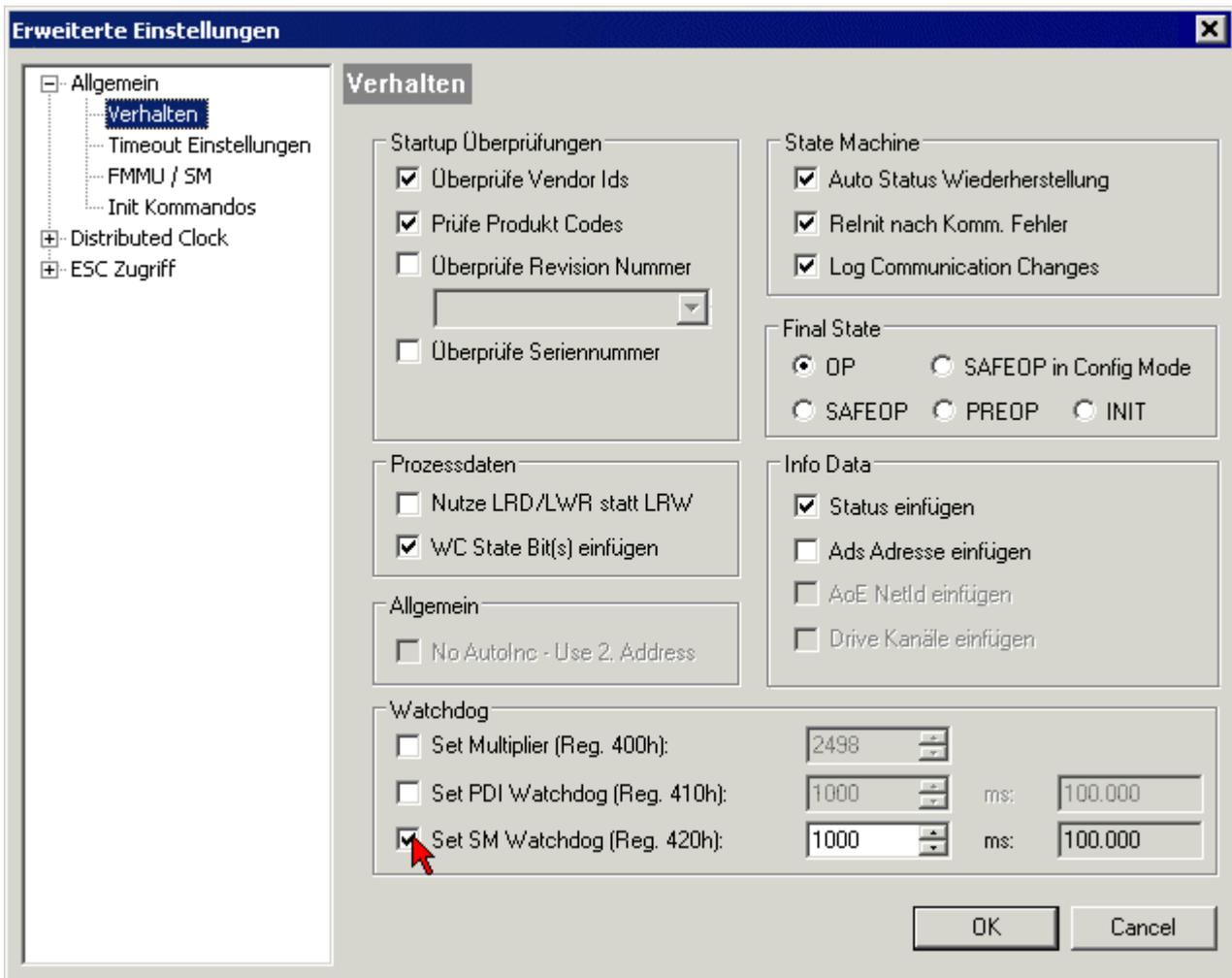


Abb. 15: Karteireiter EtherCAT -> Erweiterte Einstellungen -> Verhalten --> Watchdog

Anmerkungen:

- der Multiplier ist für beide Watchdogs gültig.
- jeder Watchdog hat dann noch eine eigene Timer-Einstellung, die zusammen mit dem Multiplier eine resultierende Zeit ergibt.
- Wichtig: die Multiplier/Timer-Einstellung wird nur beim Start in den Slave geladen, wenn die Checkbox davor aktiviert ist.  
Ist diese nicht aktiviert, wird nichts herunter geladen und die im ESC befindliche Einstellung bleibt unverändert.

**Multiplier**

Beide Watchdogs erhalten ihre Impulse aus dem lokalen Klemmentakt, geteilt durch den Watchdog-Multiplier:

$$1/25 \text{ MHz} * (\text{Watchdog-Multiplier} + 2) = 100 \mu\text{s} \text{ (bei Standard-Einstellung 2498 für den Multiplier)}$$

Die Standard Einstellung 1000 für den SM-Watchdog entspricht einer Auslösezeit von 100 ms.

Der Wert in Multiplier + 2 entspricht der Anzahl 40ns-Basisticks, die einen Watchdog-Tick darstellen. Der Multiplier kann verändert werden, um die Watchdog-Zeit in einem größeren Bereich zu verstellen.

**Beispiel „Set SM-Watchdog“**

Die Checkbox erlaubt eine manuelle Einstellung der Watchdog-Zeiten. Sind die Ausgänge gesetzt und tritt eine EtherCAT-Kommunikationsunterbrechung auf, löst der SM-Watchdog nach der eingestellten Zeit ein Löschen der Ausgänge aus. Diese Einstellung kann dazu verwendet werden, um eine Klemme an langsame

EtherCAT-Master oder sehr lange Zykluszeiten anzupassen. Der Standardwert des SM-Watchdog ist auf 100 ms eingestellt. Der Einstellbereich umfasst 0...65535. Zusammen mit einem Multiplier in einem Bereich von 1...65535 deckt dies einen Watchdog-Zeitraum von 0...~170 Sekunden ab.

### Berechnung

Multiplier = 2498 → Watchdog-Basiszeit =  $1 / 25 \text{ MHz} * (2498 + 2) = 0,0001 \text{ Sekunden} = 100 \mu\text{s}$   
SM Watchdog = 10000 →  $10000 * 100 \mu\text{s} = 1 \text{ Sekunde Watchdog-Überwachungszeit}$

#### **⚠ VORSICHT**

##### **Ungewolltes Verhalten des Systems möglich!**

Die Abschaltung des SM-Watchdog durch SM Watchdog = 0 funktioniert erst in Klemmen ab Version -0016. In vorherigen Versionen wird vom Einsatz dieser Betriebsart abgeraten.

#### **⚠ VORSICHT**

##### **Beschädigung von Geräten und ungewolltes Verhalten des Systems möglich!**

Bei aktiviertem SM-Watchdog und eingetragenen Wert 0 schaltet der Watchdog vollständig ab! Dies ist die Deaktivierung des Watchdogs! Gesetzte Ausgänge werden dann bei einer Kommunikationsunterbrechung NICHT in den sicheren Zustand gesetzt!

## 3.4 EtherCAT State Machine

Über die EtherCAT State Machine (ESM) wird der Zustand des EtherCAT-Slaves gesteuert. Je nach Zustand sind unterschiedliche Funktionen im EtherCAT-Slave zugänglich bzw. ausführbar. Insbesondere während des Hochlaufs des Slaves müssen in jedem State spezifische Kommandos vom EtherCAT Master zum Gerät gesendet werden.

Es werden folgende Zustände unterschieden:

- Init
- Pre-Operational
- Safe-Operational und
- Operational
- Boot

Regulärer Zustand eines jeden EtherCAT Slaves nach dem Hochlauf ist der Status OP.

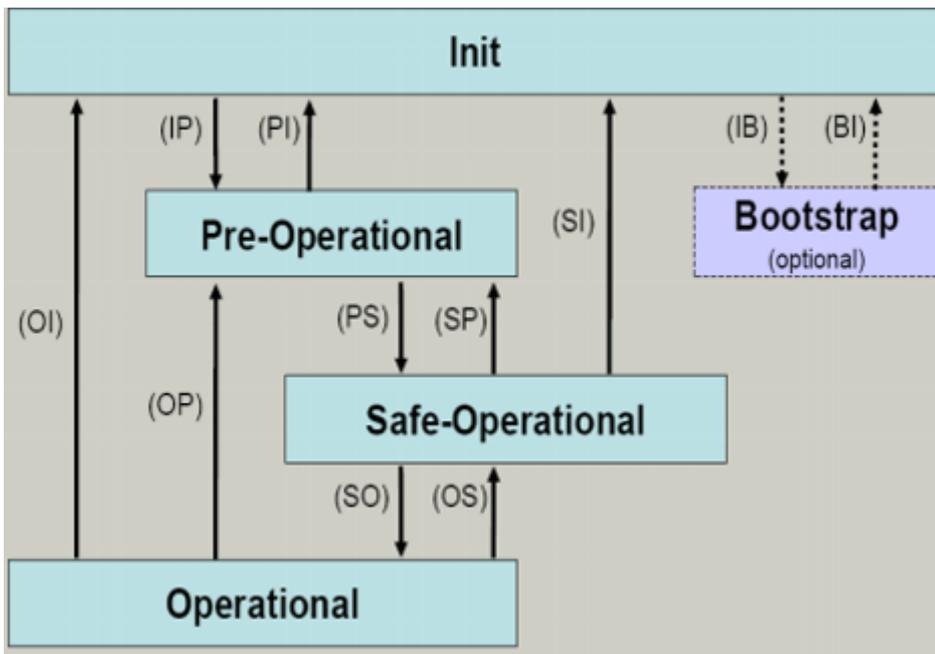


Abb. 16: Zustände der EtherCAT State Machine

### Init

Nach dem Einschalten befindet sich der EtherCAT-Slave im Zustand Init. Dort ist weder Mailbox- noch Prozessdatenkommunikation möglich. Der EtherCAT-Master initialisiert die Sync-Manager-Kanäle 0 und 1 für die Mailbox-Kommunikation.

### Pre-Operational (Pre-Op)

Beim Übergang von *Init* nach *Pre-Op* prüft der EtherCAT-Slave, ob die Mailbox korrekt initialisiert wurde.

Im Zustand *Pre-Op* ist Mailbox-Kommunikation aber keine Prozessdaten-Kommunikation möglich. Der EtherCAT-Master initialisiert die Sync-Manager-Kanäle für Prozessdaten (ab Sync-Manager-Kanal 2), die FMMU-Kanäle und falls der Slave ein konfigurierbares Mapping unterstützt das PDO-Mapping oder das Sync-Manager-PDO-Assignment. Weiterhin werden in diesem Zustand die Einstellungen für die Prozessdatenübertragung sowie ggf. noch klemmenspezifische Parameter übertragen, die von den Defaulteinstellungen abweichen.

### Safe-Operational (Safe-Op)

Beim Übergang von *Pre-Op* nach *Safe-Op* prüft der EtherCAT-Slave, ob die Sync-Manager-Kanäle für die Prozessdatenkommunikation sowie ggf. ob die Einstellungen für die Distributed-Clocks korrekt sind. Bevor er den Zustandswechsel quittiert, kopiert der EtherCAT-Slave aktuelle Inputdaten in die entsprechenden DP-RAM-Bereiche des EtherCAT-Slave-Controllers (ECSC).

Im Zustand *Safe-Op* ist Mailbox- und Prozessdaten-Kommunikation möglich, allerdings hält der Slave seine Ausgänge im sicheren Zustand und gibt sie noch nicht aus. Die Inputdaten werden aber bereits zyklisch aktualisiert.

### ● Ausgänge im SAFEOP

**i** Die standardmäßig aktivierte Watchdogüberwachung [► 26] bringt die Ausgänge im Modul in Abhängigkeit von den Einstellungen im SAFEOP und OP in einen sicheren Zustand - je nach Gerät und Einstellung z. B. auf AUS. Wird dies durch Deaktivieren der Watchdogüberwachung im Modul unterbunden, können auch im Geräte-Zustand SAFEOP Ausgänge geschaltet werden bzw. gesetzt bleiben.

**Operational (Op)**

Bevor der EtherCAT-Master den EtherCAT-Slave von *Safe-Op* nach *Op* schaltet, muss er bereits gültige Outputdaten übertragen.

Im Zustand *Op* kopiert der Slave die Ausgangsdaten des Masters auf seine Ausgänge. Es ist Prozessdaten- und Mailbox-Kommunikation möglich.

**Boot**

Im Zustand *Boot* kann ein Update der Slave-Firmware vorgenommen werden. Der Zustand *Boot* ist nur über den Zustand *Init* zu erreichen.

Im Zustand *Boot* ist Mailbox-Kommunikation über das Protokoll *File-Access over EtherCAT (FoE)* möglich, aber keine andere Mailbox-Kommunikation und keine Prozessdaten-Kommunikation.

## 3.5 CoE-Interface

**Allgemeine Beschreibung**

Das CoE-Interface (CAN application protocol over EtherCAT) ist die Parameterverwaltung für EtherCAT-Geräte. EtherCAT-Slaves oder auch der EtherCAT-Master verwalten darin feste (ReadOnly) oder veränderliche Parameter, die sie zum Betrieb, Diagnose oder Inbetriebnahme benötigen.

CoE-Parameter sind in einer Tabellen-Hierarchie angeordnet und prinzipiell dem Anwender über den Feldbus lesbar zugänglich. Der EtherCAT-Master (TwinCAT System Manager) kann über EtherCAT auf die lokalen CoE-Verzeichnisse der Slaves zugreifen und je nach Eigenschaften lesend oder schreibend einwirken.

Es sind verschiedene Typen für CoE-Parameter möglich wie String (Text), Integer-Zahlen, Bool'sche Werte oder größere Byte-Felder. Damit lassen sich ganz verschiedene Eigenschaften beschreiben. Beispiele für solche Parameter sind Herstellerkennung, Seriennummer, Prozessdateneinstellungen, Gerätename, Abgleichwerte für analoge Messung oder Passwörter.

Die Ordnung erfolgt in zwei Ebenen über hexadezimale Nummerierung: zuerst wird der (Haupt)Index genannt, dann der Subindex. Die Wertebereiche sind

- Index: 0x0000...0xFFFF (0...65535<sub>dez</sub>)
- SubIndex: 0x00...0xFF (0...255<sub>dez</sub>)

Üblicherweise wird ein so lokalisierter Parameter geschrieben als 0x8010:07 mit voranstehendem „0x“ als Kennzeichen des hexadezimalen Zahlenraumes und Doppelpunkt zwischen Index und Subindex.

Die für den EtherCAT-Feldbusanwender wichtigen Bereiche sind

- 0x1000: hier sind feste Identitäts-Informationen zum Gerät hinterlegt wie Name, Hersteller, Seriennummer etc. Außerdem liegen hier Angaben über die aktuellen und verfügbaren Prozessdatenkonstellationen.
- 0x8000: hier sind die für den Betrieb erforderlichen funktionsrelevanten Parameter für alle Kanäle zugänglich wie Filtereinstellung oder Ausgabefrequenz.

Weitere wichtige Bereiche sind:

- 0x4000: hier befinden sich bei manchen EtherCAT-Geräten die Kanalparameter. Historisch war dies der erste Parameterbereich, bevor der 0x8000 Bereich eingeführt wurde. EtherCAT Geräte, die früher mit Parametern in 0x4000 ausgerüstet wurden und auf 0x8000 umgestellt wurden, unterstützen aus Kompatibilitätsgründen beide Bereiche und spiegeln intern.
- 0x6000: hier liegen die Eingangs-PDO („Eingang“ aus Sicht des EtherCAT-Masters)
- 0x7000: hier liegen die Ausgangs-PDO („Ausgang“ aus Sicht des EtherCAT-Masters)

**Verfügbarkeit**

**i** Nicht jedes EtherCAT Gerät muss über ein CoE-Verzeichnis verfügen. Einfache I/O-Module ohne eigenen Prozessor verfügen in der Regel über keine veränderlichen Parameter und haben deshalb auch kein CoE-Verzeichnis.

Wenn ein Gerät über ein CoE-Verzeichnis verfügt, stellt sich dies im TwinCAT System Manager als ein eigener Karteireiter mit der Auflistung der Elemente dar:

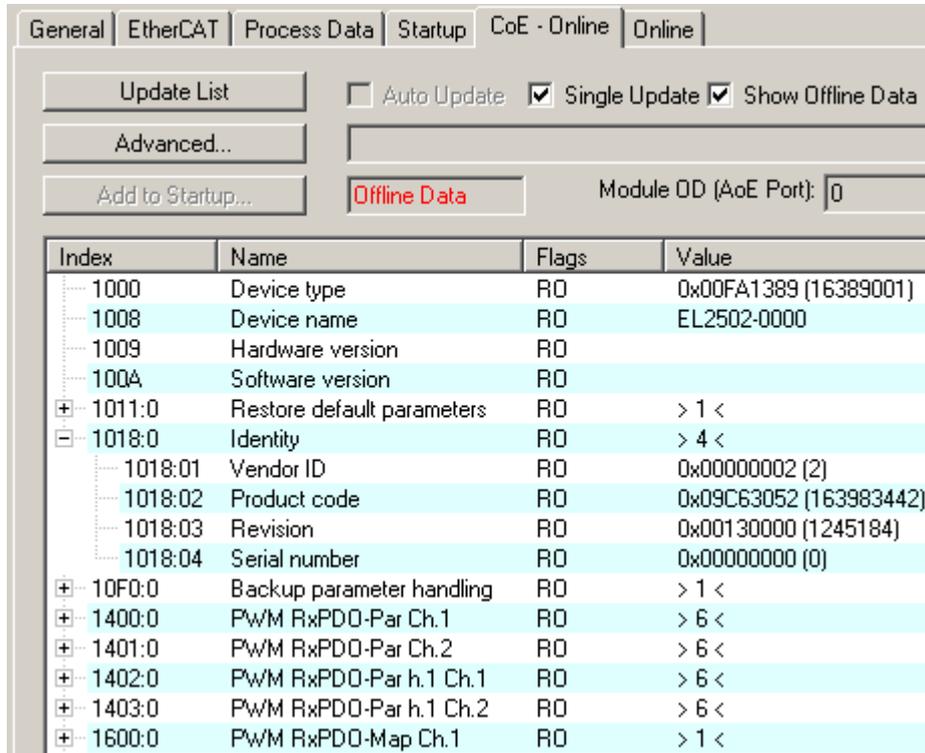


Abb. 17: Karteireiter „CoE-Online“

In der oberen Abbildung sind die im Gerät „EL2502“ verfügbaren CoE-Objekte von 0x1000 bis 0x1600 zusehen, die Subindizes von 0x1018 sind aufgeklappt.

**Datenerhaltung und Funktion „NoCoeStorage“**

Einige, insbesondere die vorgesehenen Einstellungsparameter des Slaves sind veränderlich und beschreibbar. Dies kann schreibend/lesend geschehen

- über den System Manager (Abb. Karteireiter „CoE-Online“) durch Anklicken  
Dies bietet sich bei der Inbetriebnahme der Anlage/Slaves an. Klicken Sie auf die entsprechende Zeile des zu parametrierenden Indizes und geben sie einen entsprechenden Wert im „SetValue“-Dialog ein.
- aus der Steuerung/PLC über ADS z. B. durch die Bausteine aus der TcEtherCAT.lib Bibliothek  
Dies wird für Änderungen während der Anlangenlaufzeit empfohlen oder wenn kein System Manager bzw. Bedienpersonal zur Verfügung steht.

**i Datenerhaltung**

Werden online auf dem Slave CoE-Parameter geändert, wird dies in Beckhoff-Geräten üblicherweise ausfallsicher im Gerät (EEPROM) gespeichert. D. h. nach einem Neustart (Repower) sind die veränderten CoE-Parameter immer noch erhalten. Andere Hersteller können dies anders handhaben.

Ein EEPROM unterliegt in Bezug auf Schreibvorgänge einer begrenzten Lebensdauer. Ab typischerweise 100.000 Schreibvorgängen kann eventuell nicht mehr sichergestellt werden, dass neue (veränderte) Daten sicher gespeichert werden oder noch auslesbar sind. Dies ist für die normale Inbetriebnahme ohne Belang. Werden allerdings zur Maschinenlaufzeit fortlaufend CoE-Parameter über ADS verändert, kann die Lebensdauergerenze des EEPROM durchaus erreicht werden.

Es ist von der FW-Version abhängig, ob die Funktion NoCoeStorage unterstützt wird, die das Abspeichern veränderter CoE-Werte unterdrückt. Ob das auf das jeweilige Gerät zutrifft, ist den technischen Daten dieser Dokumentation zu entnehmen.

- wird unterstützt: die Funktion ist per einmaligem Eintrag des Codeworts 0x12345678 in CoE 0xF008 zu aktivieren und solange aktiv, wie das Codewort nicht verändert wird. Nach dem Einschalten des Gerätes ist sie nicht aktiv. Veränderte CoE-Werte werden dann nicht im EEPROM abgespeichert, sie können somit beliebig oft verändert werden.
- wird nicht unterstützt: eine fortlaufende Änderung von CoE-Werten ist angesichts der o.a. Lebensdauergerenze nicht zulässig.

**i Startup List**

Veränderungen im lokalen CoE-Verzeichnis der Klemme gehen im Austauschfall mit der alten Klemme verloren. Wird im Austauschfall eine neue Klemme mit Werkseinstellungen ab Lager Beckhoff eingesetzt, bringt diese die Standardeinstellungen mit. Es ist deshalb empfehlenswert, alle Veränderungen im CoE-Verzeichnis eines EtherCAT Slave in der Startup List des Slaves zu verankern, die bei jedem Start des EtherCAT Feldbus abgearbeitet wird. So wird auch ein im Austauschfall ein neuer EtherCAT Slave automatisch mit den Vorgaben des Anwenders parametrieret.

Wenn EtherCAT Slaves verwendet werden, die lokal CoE-Wert nicht dauerhaft speichern können, ist zwingend die StartUp-Liste zu verwenden.

**Empfohlenes Vorgehen bei manueller Veränderung von CoE-Parametern**

- gewünschte Änderung im System Manager vornehmen  
Werte werden lokal im EtherCAT Slave gespeichert
- wenn der Wert dauerhaft Anwendung finden soll, einen entsprechenden Eintrag in der StartUp-Liste vornehmen.  
Die Reihenfolge der StartUp-Einträge ist dabei i.d.R. nicht relevant.

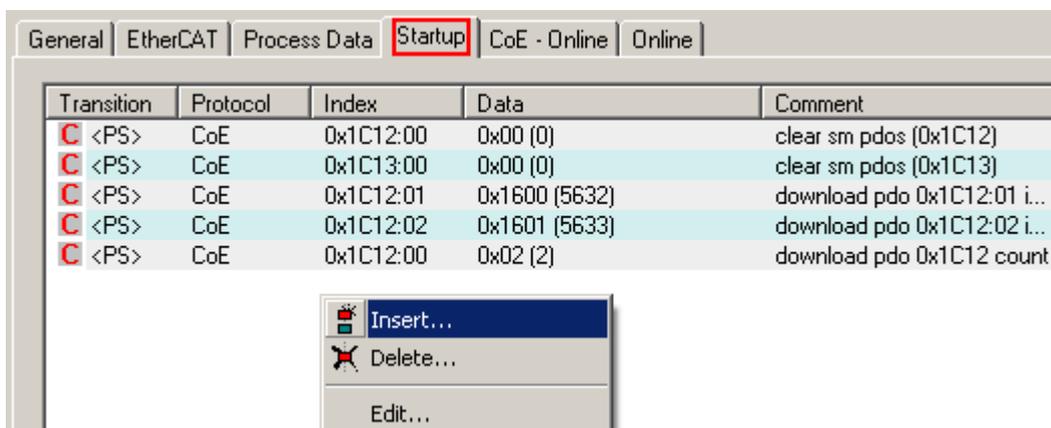


Abb. 18: StartUp-Liste im TwinCAT System Manager

In der StartUp-Liste können bereits Werte enthalten sein, die vom System Manager nach den Angaben der ESI dort angelegt werden. Zusätzliche anwendungsspezifische Einträge können angelegt werden.

**Online/Offline Verzeichnis**

Während der Arbeit mit dem TwinCAT System Manager ist zu unterscheiden ob das EtherCAT-Gerät gerade „verfügbar“, also angeschaltet und über EtherCAT verbunden und damit **online** ist oder ob ohne angeschlossene Slaves eine Konfiguration **offline** erstellt wird.

In beiden Fällen ist ein CoE-Verzeichnis nach Abb. „Karteireiter ‚CoE-Online‘“ zu sehen, die Konnektivität wird allerdings als offline/online angezeigt.

- wenn der Slave offline ist:
  - wird das Offline-Verzeichnis aus der ESI-Datei angezeigt. Änderungen sind hier nicht sinnvoll bzw. möglich.
  - wird in der Identität der konfigurierte Stand angezeigt
  - wird kein Firmware- oder Hardware-Stand angezeigt, da dies Eigenschaften des realen Gerätes sind.
  - ist ein rotes **Offline** zu sehen

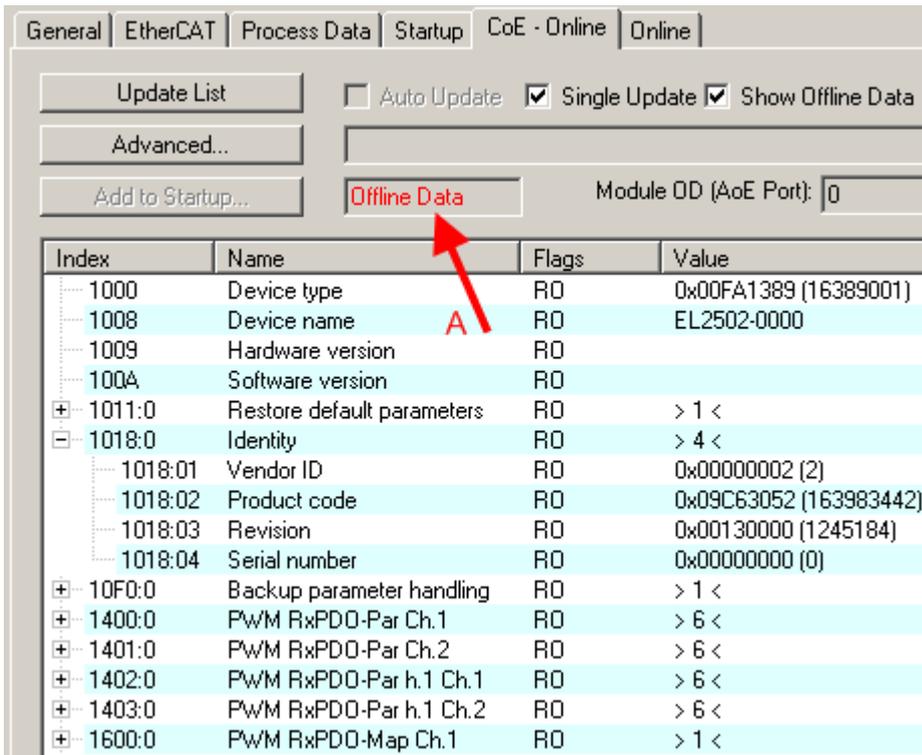


Abb. 19: Offline-Verzeichnis

- wenn der Slave online ist
  - wird das reale aktuelle Verzeichnis des Slaves ausgelesen. Dies kann je nach Größe und Zykluszeit einige Sekunden dauern.
  - wird die tatsächliche Identität angezeigt
  - wird der Firmware- und Hardware-Stand des Gerätes laut elektronischer Auskunft angezeigt
  - ist ein grünes **Online** zu sehen

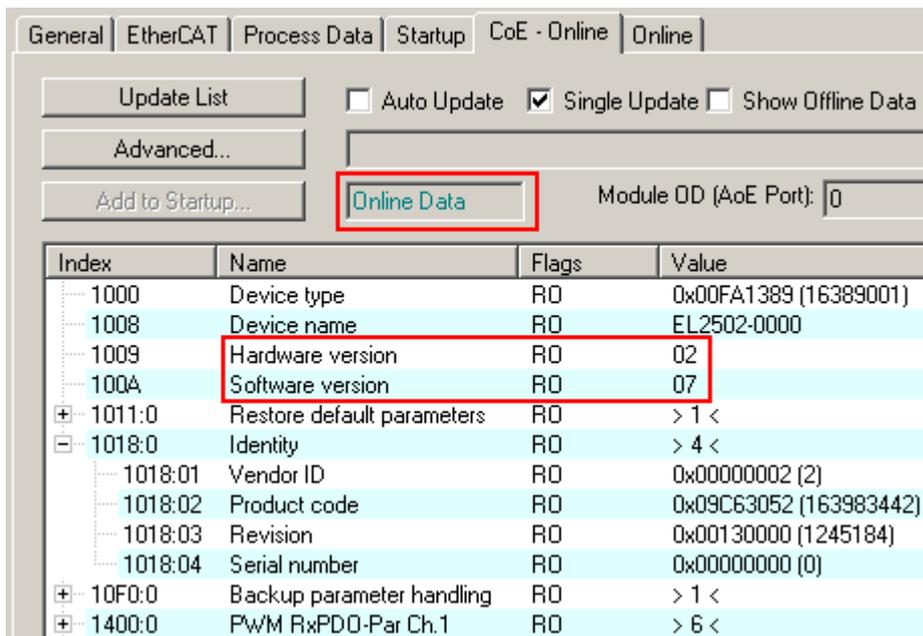


Abb. 20: Online-Verzeichnis

### Kanalweise Ordnung

Das CoE-Verzeichnis ist in EtherCAT Geräten angesiedelt, die meist mehrere funktional gleichwertige Kanäle umfassen. z. B. hat eine 4 kanalige Analogeingangsklemme 0...10 V auch vier logische Kanäle und damit vier gleiche Sätze an Parameterdaten für die Kanäle. Um in den Dokumentationen nicht jeden Kanal auflisten zu müssen, wird gerne der Platzhalter „n“ für die einzelnen Kanalnummern verwendet.

Im CoE-System sind für die Menge aller Parameter eines Kanals eigentlich immer 16 Indizes mit jeweils 255 Subindizes ausreichend. Deshalb ist die kanalweise Ordnung in  $16_{\text{dez}}/10_{\text{hex}}$ -Schritten eingerichtet. Am Beispiel des Parameterbereichs 0x8000 sieht man dies deutlich:

- Kanal 0: Parameterbereich 0x8000:00 ... 0x800F:255
- Kanal 1: Parameterbereich 0x8010:00 ... 0x801F:255
- Kanal 2: Parameterbereich 0x8020:00 ... 0x802F:255
- ...

Allgemein wird dies geschrieben als 0x80n0.

Ausführliche Hinweise zum CoE-Interface finden Sie in der [EtherCAT-Systemdokumentation](#) auf der Beckhoff Website.

## 3.6 Distributed Clock

Die Distributed Clock stellt eine lokale Uhr im EtherCAT Slave Controller (ESC) dar mit den Eigenschaften:

- Einheit *1 ns*
- Nullpunkt *1.1.2000 00:00*
- Umfang *64 Bit* (ausreichend für die nächsten 584 Jahre); manche EtherCAT-Slaves unterstützen jedoch nur einen Umfang von 32 Bit, d.h. nach ca. 4,2 Sekunden läuft die Variable über
- Diese lokale Uhr wird vom EtherCAT Master automatisch mit der Master Clock im EtherCAT Bus mit einer Genauigkeit  $< 100$  ns synchronisiert.

Detaillierte Informationen entnehmen Sie bitte der vollständigen [EtherCAT-Systembeschreibung](#).

## 4 Montage und Verdrahtung

### 4.1 Hinweise zum ESD-Schutz

#### HINWEIS

##### Zerstörung der Geräte durch elektrostatische Aufladung möglich!

Die Geräte enthalten elektrostatisch gefährdete Bauelemente, die durch unsachgemäße Behandlung beschädigt werden können.

- Sie müssen beim Umgang mit den Komponenten elektrostatisch entladen sein; vermeiden Sie außerdem die Federkontakte (s. Abb.) direkt zu berühren.
- Vermeiden Sie den Kontakt mit hoch isolierenden Stoffen (Kunstfaser, Kunststofffolien etc.)
- Beim Umgang mit den Komponenten ist auf gute Erdung der Umgebung zu achten (Arbeitsplatz, Verpackung und Personen)
- Jede Busstation muss auf der rechten Seite mit der Endkappe [EL9011](#) oder [EL9012](#) abgeschlossen werden, um Schutzart und ESD-Schutz sicher zu stellen.

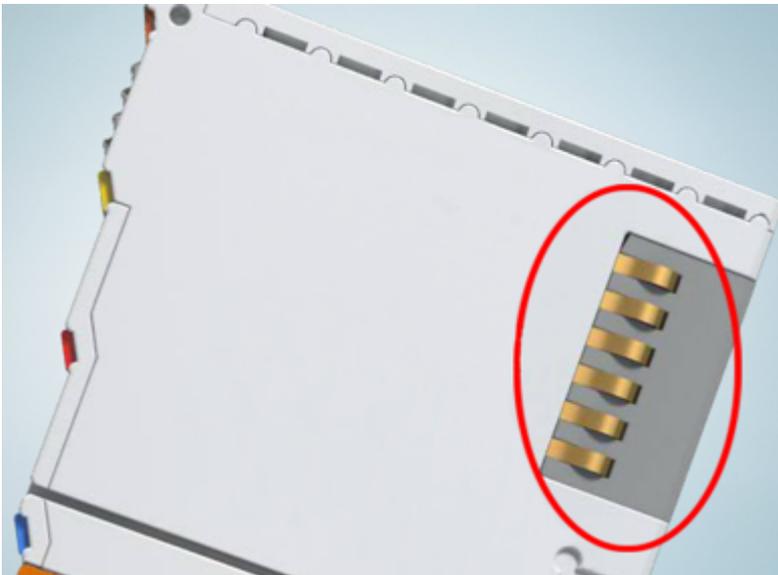


Abb. 21: Federkontakte der Beckhoff I/O-Komponenten

### 4.2 EL6001, EL6021

#### 4.2.1 Tragschienenmontage

#### ⚠️ WARNUNG

##### Verletzungsgefahr durch Stromschlag und Beschädigung des Gerätes möglich!

Setzen Sie das Busklemmen-System in einen sicheren, spannungslosen Zustand, bevor Sie mit der Montage, Demontage oder Verdrahtung der Busklemmen beginnen!

**Montage**

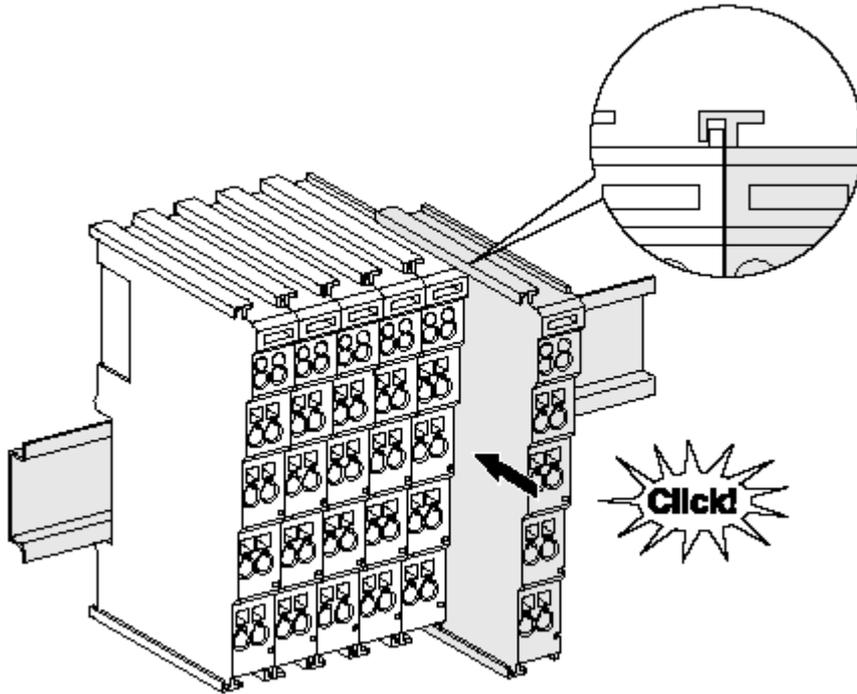


Abb. 22: Montage auf Tragschiene

Die Buskoppler und Busklemmen werden durch leichten Druck auf handelsübliche 35 mm Tragschienen (Hutschienen nach EN 60715) aufgerastet:

1. Stecken Sie zuerst den Feldbuskoppler auf die Tragschiene.
2. Auf der rechten Seite des Feldbuskopplers werden nun die Busklemmen angereiht. Stecken Sie dazu die Komponenten mit Nut und Feder zusammen und schieben Sie die Klemmen gegen die Tragschiene, bis die Verriegelung hörbar auf der Tragschiene einrastet.

Wenn Sie die Klemmen erst auf die Tragschiene schnappen und dann nebeneinander schieben ohne das Nut und Feder ineinander greifen, wird keine funktionsfähige Verbindung hergestellt! Bei richtiger Montage darf kein nennenswerter Spalt zwischen den Gehäusen zu sehen sein.

**i Tragschienenbefestigung**

Der Verriegelungsmechanismus der Klemmen und Koppler reicht in das Profil der Tragschiene hinein. Achten Sie bei der Montage der Komponenten darauf, dass der Verriegelungsmechanismus nicht in Konflikt mit den Befestigungsschrauben der Tragschiene gerät. Verwenden Sie zur Befestigung von Tragschienen mit einer Höhe von 7,5 mm unter den Klemmen und Kopplern flache Montageverbindungen wie Senkkopfschrauben oder Blindnieten.

## Demontage

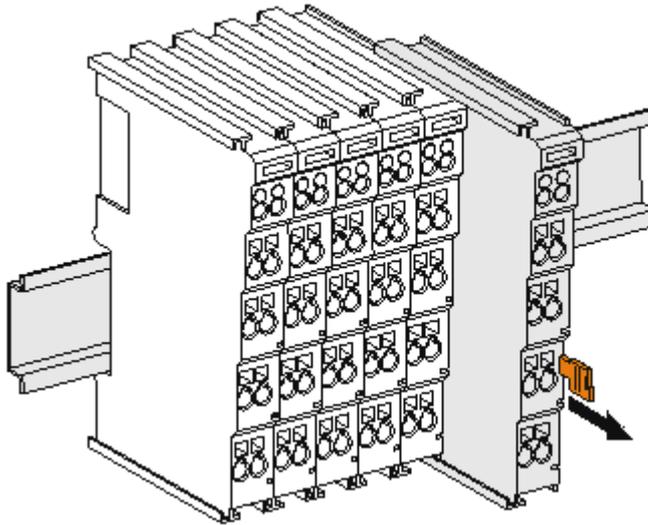


Abb. 23: Demontage von Tragschiene

Jede Klemme wird durch eine Verriegelung auf der Tragschiene gesichert, die zur Demontage gelöst werden muss:

1. Ziehen Sie die Klemme an ihren orangefarbenen Laschen ca. 1 cm von der Tragschiene herunter. Dabei wird die Tragschienerriegelung dieser Klemme automatisch gelöst und Sie können die Klemme nun ohne großen Kraftaufwand aus dem Busklemmenblock herausziehen.
2. Greifen Sie dazu mit Daumen und Zeigefinger die entriegelte Klemme gleichzeitig oben und unten an den Gehäuseflächen und ziehen sie aus dem Busklemmenblock heraus.

## Verbindungen innerhalb eines Busklemmenblocks

Die elektrischen Verbindungen zwischen Buskoppler und Busklemmen werden durch das Zusammenstecken der Komponenten automatisch realisiert:

- Die sechs Federkontakte des K-Bus/E-Bus übernehmen die Übertragung der Daten und die Versorgung der Busklemmenelektronik.
- Die Powerkontakte übertragen die Versorgung für die Feldelektronik und stellen so innerhalb des Busklemmenblocks eine Versorgungsschiene dar. Die Versorgung der Powerkontakte erfolgt über Klemmen auf dem Buskoppler (bis 24 V) oder für höhere Spannungen über Einspeiseklemmen.

### **i** Powerkontakte

Beachten Sie bei der Projektierung eines Busklemmenblocks die Kontaktbelegungen der einzelnen Busklemmen, da einige Typen (z.B. analoge Busklemmen oder digitale 4-Kanal-Busklemmen) die Powerkontakte nicht oder nicht vollständig durchschleifen. Einspeiseklemmen (KL91xx, KL92xx bzw. EL91xx, EL92xx) unterbrechen die Powerkontakte und stellen so den Anfang einer neuen Versorgungsschiene dar.

## PE-Powerkontakt

Der Powerkontakt mit der Bezeichnung PE kann als Schutzterde eingesetzt werden. Der Kontakt ist aus Sicherheitsgründen beim Zusammenstecken voreilend und kann Kurzschlussströme bis 125 A ableiten.

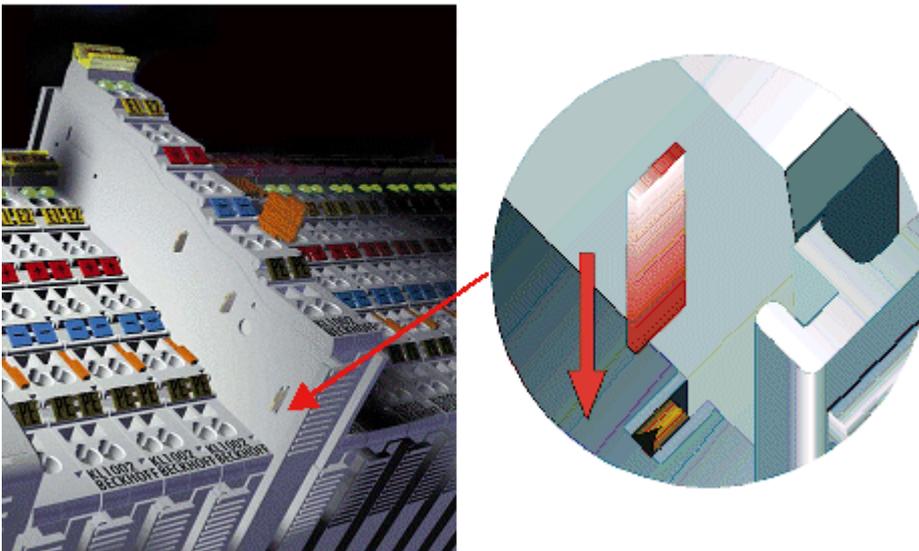


Abb. 24: Linksseitiger Powerkontakt

**HINWEIS**

**Beschädigung des Gerätes möglich**

Beachten Sie, dass aus EMV-Gründen die PE-Kontakte kapazitiv mit der Tragschiene verbunden sind. Das kann bei der Isolationsprüfung zu falschen Ergebnissen und auch zur Beschädigung der Klemme führen (z. B. Durchschlag zur PE-Leitung bei der Isolationsprüfung eines Verbrauchers mit 230 V Nennspannung). Klemmen Sie zur Isolationsprüfung die PE- Zuleitung am Buskoppler bzw. der Einspeiseklemme ab! Um weitere Einspeisestellen für die Prüfung zu entkoppeln, können Sie diese Einspeiseklemmen entriegeln und mindestens 10 mm aus dem Verbund der übrigen Klemmen herausziehen.

**⚠️ WARNUNG**

**Verletzungsgefahr durch Stromschlag!**

Der PE-Powerkontakt darf nicht für andere Potentiale verwendet werden!

**4.2.2 Anschluss**

**4.2.2.1 Anschlusstechnik**

**⚠️ WARNUNG**

**Verletzungsgefahr durch Stromschlag und Beschädigung des Gerätes möglich!**

Setzen Sie das Busklemmen-System in einen sicheren, spannungslosen Zustand, bevor Sie mit der Montage, Demontage oder Verdrahtung der Busklemmen beginnen!

**Übersicht**

Mit verschiedenen Anschlussoptionen bietet das Busklemmensystem eine optimale Anpassung an die Anwendung:

- Die Klemmen der Serien ELxxxx und KLxxxx mit Standardverdrahtung enthalten Elektronik und Anschlussebene in einem Gehäuse.
- Die Klemmen der Serien ESxxxx und KSxxxx haben eine steckbare Anschlussebene und ermöglichen somit beim Austausch die stehende Verdrahtung.
- Die High-Density-Klemmen (HD-Klemmen) enthalten Elektronik und Anschlussebene in einem Gehäuse und haben eine erhöhte Packungsdichte.

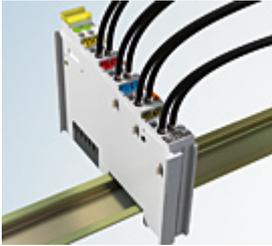
**Standardverdrahtung (ELxxxx / KLxxxx)**

Abb. 25: Standardverdrahtung

Die Klemmen der Serien ELxxxx und KLxxxx sind seit Jahren bewährt und integrieren die schraublose Federkrafttechnik zur schnellen und einfachen Montage.

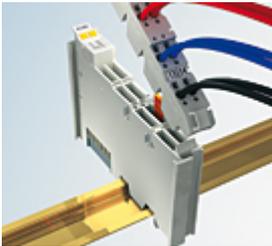
**Steckbare Verdrahtung (ESxxxx / KSxxxx)**

Abb. 26: Steckbare Verdrahtung

Die Klemmen der Serien ESxxxx und KSxxxx enthalten eine steckbare Anschlussebene. Montage und Verdrahtung werden wie bei den Serien ELxxxx und KLxxxx durchgeführt. Im Servicefall erlaubt die steckbare Anschlussebene, die gesamte Verdrahtung als einen Stecker von der Gehäuseoberseite abzuziehen. Das Unterteil kann, über das Betätigen der Entriegelungslasche, aus dem Klemmenblock herausgezogen werden. Die auszutauschende Komponente wird hineingeschoben und der Stecker mit der stehenden Verdrahtung wieder aufgesteckt. Dadurch verringert sich die Montagezeit und ein Verwechseln der Anschlussdrähte ist ausgeschlossen.

Die gewohnten Maße der Klemme ändern sich durch den Stecker nur geringfügig. Der Stecker trägt ungefähr 3 mm auf; dabei bleibt die maximale Höhe der Klemme unverändert.

Eine Lasche für die Zugentlastung des Kabels stellt in vielen Anwendungen eine deutliche Vereinfachung der Montage dar und verhindert ein Verheddern der einzelnen Anschlussdrähte bei gezogenem Stecker.

Leiterquerschnitte von 0,08 mm<sup>2</sup> bis 2,5 mm<sup>2</sup> können weiter in der bewährten Federkrafttechnik verwendet werden.

Übersicht und Systematik in den Produktbezeichnungen der Serien ESxxxx und KSxxxx werden wie von den Serien ELxxxx und KLxxxx bekannt weitergeführt.

**High-Density-Klemmen (HD-Klemmen)**

Abb. 27: High-Density-Klemmen

Die Klemmen dieser Baureihe mit 16 Klemmstellen zeichnen sich durch eine besonders kompakte Bauform aus, da die Packungsdichte auf 12 mm doppelt so hoch ist wie die der Standard-Busklemmen. Massive und mit einer Aderendhülse versehene Leiter können ohne Werkzeug direkt in die Federklemmstelle gesteckt werden.

---

● **Verdrahtung HD-Klemmen**

**i** Die High-Density-Klemmen der Serien ELx8xx und KLx8xx unterstützen keine steckbare Verdrahtung.

---

**Ultraschall-litzenverdichtete Leiter**

---

● **Ultraschall-litzenverdichtete Leiter**

**i** An die Standard- und High-Density-Klemmen können auch ultraschall-litzenverdichtete (ultraschallverschweißte) Leiter angeschlossen werden. Beachten Sie die Tabellen zum Leitungsquerschnitt!

---

### 4.2.2.2 Verdrahtung

#### ⚠️ WARNUNG

**Verletzungsgefahr durch Stromschlag und Beschädigung des Gerätes möglich!**

Setzen Sie das Busklemmen-System in einen sicheren, spannungslosen Zustand, bevor Sie mit der Montage, Demontage oder Verdrahtung der Busklemmen beginnen!

**Klemmen für Standardverdrahtung ELxxxx/KLxxxx und für steckbare Verdrahtung ESxxxx/KSxxxx**

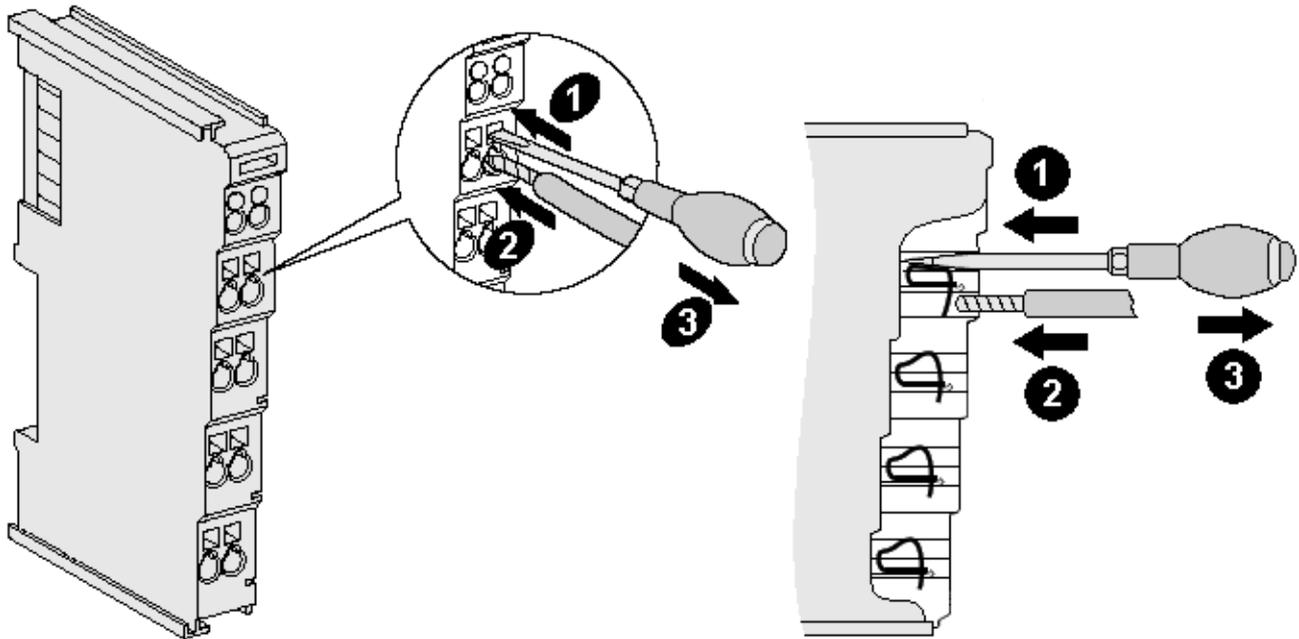


Abb. 28: Anschluss einer Leitung an eine Klemmstelle

Bis zu acht Klemmstellen ermöglichen den Anschluss von massiven oder feindrähtigen Leitungen an die Busklemme. Die Klemmstellen sind in Federkrafttechnik ausgeführt. Schließen Sie die Leitungen folgendermaßen an:

1. Öffnen Sie eine Klemmstelle, indem Sie einen Schraubendreher gerade bis zum Anschlag in die viereckige Öffnung über der Klemmstelle drücken. Den Schraubendreher dabei nicht drehen oder hin und her bewegen (nicht hebeln).
2. Der Draht kann nun ohne Widerstand in die runde Klemmenöffnung eingeführt werden.
3. Durch Rücknahme des Druckes schließt sich die Klemmstelle automatisch und hält den Draht sicher und dauerhaft fest.

Den zulässigen Leiterquerschnitt entnehmen Sie der nachfolgenden Tabelle.

Klemmgehäuse	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Leitungsquerschnitt (massiv)	0,08 ... 2,5 mm <sup>2</sup>	0,08 ... 2,5 mm <sup>2</sup>
Leitungsquerschnitt (feindrähtig)	0,08 ... 2,5 mm <sup>2</sup>	0,08 ... 2,5 mm <sup>2</sup>
Leitungsquerschnitt (Aderleitung mit Aderendhülse)	0,14 ... 1,5 mm <sup>2</sup>	0,14 ... 1,5 mm <sup>2</sup>
Abisolierlänge	8 ... 9 mm	9 ... 10 mm

#### High-Density-Klemmen (HD-Klemmen [► 40]) mit 16 Klemmstellen

Bei den HD-Klemmen erfolgt der Leiteranschluss bei massiven Leitern werkzeuglos, in Direktstecktechnik, das heißt der Leiter wird nach dem Abisolieren einfach in die Klemmstelle gesteckt. Das Lösen der Leitungen erfolgt, wie bei den Standardklemmen, über die Kontakt-Entriegelung mit Hilfe eines Schraubendrehers. Den zulässigen Leiterquerschnitt entnehmen Sie der nachfolgenden Tabelle.

Klemmgehäuse	HD-Gehäuse
Leitungsquerschnitt (massiv)	0,08 ... 1,5 mm <sup>2</sup>
Leitungsquerschnitt (feindrätig)	0,25 ... 1,5 mm <sup>2</sup>
Leitungsquerschnitt (Aderleitung mit Aderendhülse)	0,14 ... 0,75 mm <sup>2</sup>
Leitungsquerschnitt (ultraschall-litzenverdichtet)	nur 1,5 mm <sup>2</sup>
Abisolierlänge	8 ... 9 mm

#### 4.2.2.3 Schirmung



##### Schirmung

Encoder, analoge Sensoren und Aktoren sollten immer mit geschirmten, paarig verdrehten Leitungen angeschlossen werden.

### 4.2.3 Positionierung von passiven Klemmen



#### Hinweis zur Positionierung von passiven Klemmen im Busklemmenblock

EtherCAT-Klemmen (ELxxxx / ESxxxx), die nicht aktiv am Datenaustausch innerhalb des Busklemmenblocks teilnehmen, werden als passive Klemmen bezeichnet. Zu erkennen sind diese Klemmen an der nicht vorhandenen Stromaufnahme aus dem E-Bus. Um einen optimalen Datenaustausch zu gewährleisten, dürfen nicht mehr als zwei passive Klemmen direkt aneinander gereiht werden!

#### Beispiele für die Positionierung von passiven Klemmen (hell eingefärbt)

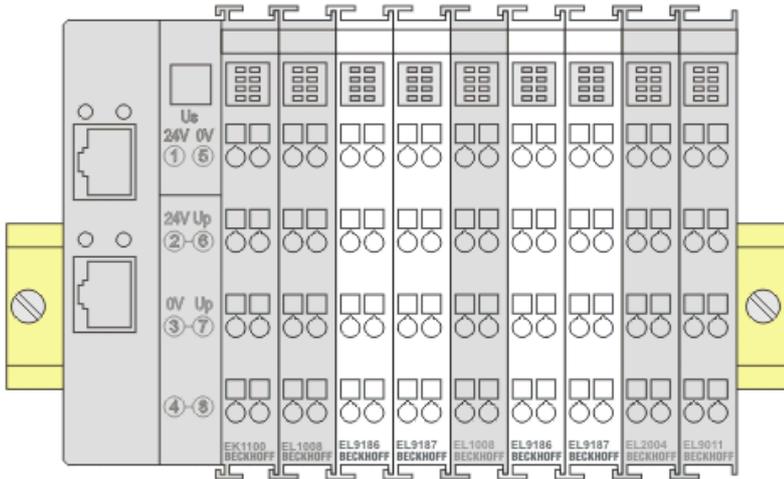


Abb. 29: Korrekte Positionierung

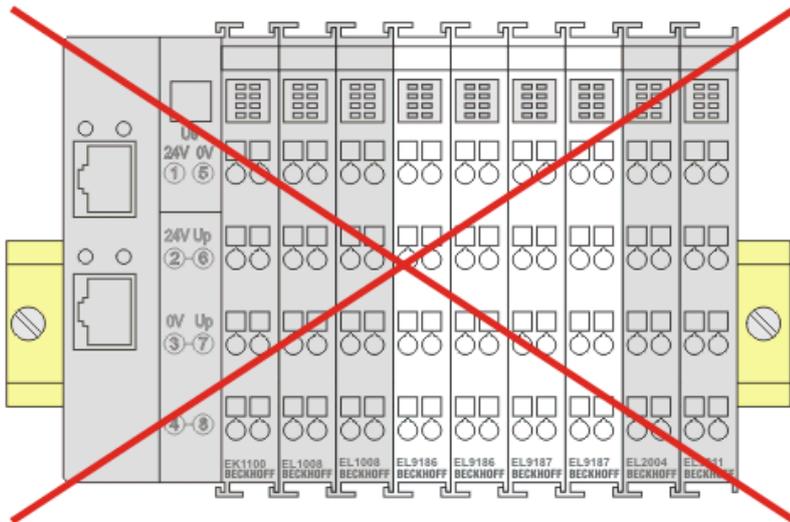


Abb. 30: Inkorrekte Positionierung

### 4.2.4 LEDs und Anschlussbelegung

#### LEDs

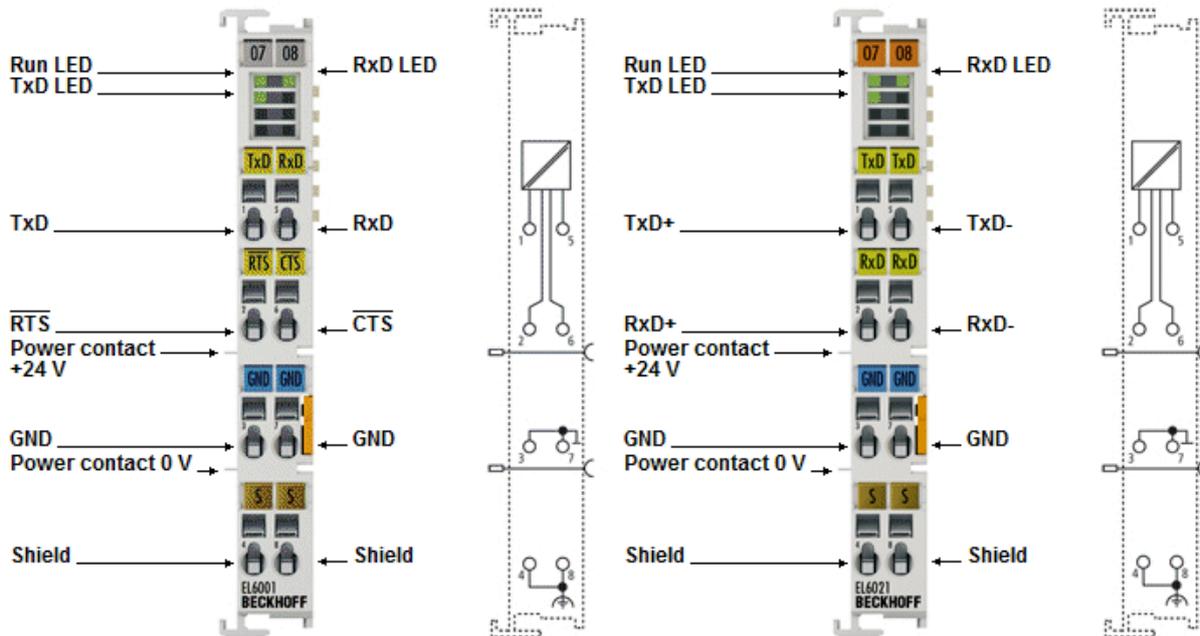


Abb. 31: EL6001, EL6021 - LEDs und Anschlussbelegung

LED	Farbe	Bedeutung	
RUN	Grün	Diese LED gibt den Betriebszustand der Klemme wieder:	
		Aus	Zustand der EtherCAT State Machine [► 28]: INIT = Initialisierung der Klemme oder BOOTSTRAP = Funktion für Firmware Updates [► 196] der Klemme
		Blinkend	Zustand der EtherCAT State Machine: PREOP = Funktion für Mailbox-Kommunikation und abweichende Standard-Einstellungen gesetzt
		Einzelblitz	Zustand der EtherCAT State Machine: SAFEOP = Überprüfung der Kanäle des Sync-Managers [► 112] und der Distributed Clocks. Ausgänge bleiben im sicheren Zustand
	An	Zustand der EtherCAT State Machine: OP = normaler Betriebszustand; Mailbox- und Prozessdatenkommunikation ist möglich	
TxD	Grün	Zustand der Sende-Signalleitung (an: HI-Pegel auf Sendeleitung)	
RxD	Grün	Zustand der Empfangs-Signalleitung (an: HI-Pegel auf Empfangsleitung)	

**Anschlussbelegung**

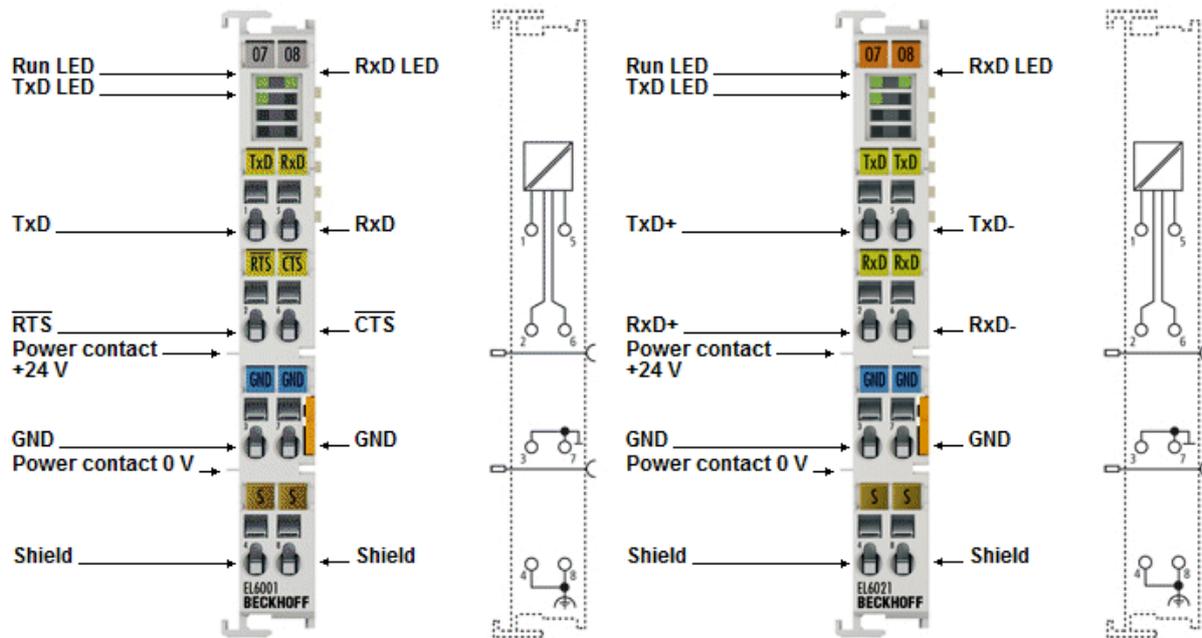


Abb. 32: EL6001, EL6021 - LEDs und Anschlussbelegung

**Anschlussbelegung EL6001**

Klemmstelle	Name	Signal
1	TxD	Signalleitung (Transmit Data)
5	RxD	Signalleitung (Receive Data)
2	RTS	Steuerleitung (Request To Send)
6	CTS	Steuerleitung (Clear To Send)
3	GND	Masseanschluss (intern gebrückt mit Klemme 7)
7	GND	Masseanschluss (intern gebrückt mit Klemme 3)
4	Shield	Schirmanschluss (intern gebrückt mit Klemme 8)
8	Shield	Schirmanschluss (intern gebrückt mit Klemme 4)

**Anschlussbelegung EL6021**

Klemmstelle	Name	Signal
1	TxD+	Signalleitung + (Transmit Data)
5	TxD-	Signalleitung - (Transmit Data)
2	RxD+	Signalleitung + (Receive Data)
6	RxD-	Signalleitung - (Receive Data)
3	GND	Masseanschluss (intern gebrückt mit Klemme 7)
7	GND	Masseanschluss (intern gebrückt mit Klemme 3)
4	Shield	Schirmanschluss (intern gebrückt mit Klemme 8)
8	Shield	Schirmanschluss (intern gebrückt mit Klemme 4)

**Anschluss bei RS422 Übertragung**

Bei der RS422 Betriebsart können die Daten voll duplex übertragen werden. Es können nur Punkt zu Punkt Verbindungen hergestellt werden.

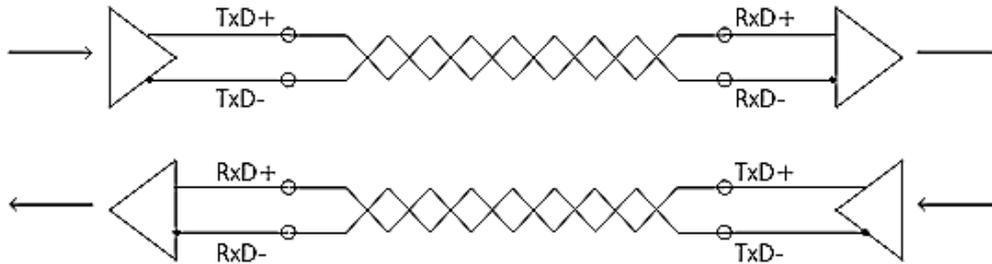


Abb. 33: Anschluss bei RS422 Übertragung

**Anschluss bei RS485 Übertragung**

Bei der RS485 Betriebsart werden die Daten in halbduplex Übertragung ausgetauscht. In dieser Betriebsart kann eine Busstruktur aufgebaut werden.



Abb. 34: Anshchluss bei RS485 Übertragung

Im RS485 Betriebsmodus werden die Sende- und Empfangsleitungen miteinander verbunden. Dadurch empfängt die Klemme nicht nur die Daten anderer Teilnehmer, sondern auch ihre eigenen Sendedaten. Mit dem Index 0x8000:06 "Enable half duplex" im Settings-Objekt kann dies unterdrückt werden.

Das Empfangen neuer Daten wird in der Betriebsart RS485 erst möglich, wenn das Senden abgeschlossen ist.

"Enable half duplex"	"Enable point to point connection (RS422)"	Modus
0	0	RS485: Die Klemme empfängt ihre eigenen und die Daten anderer Teilnehmer
0	1	RS422: Normaler Betriebsmodus; Die Klemme arbeitet im Vollduplex-Modus.
1	0	RS485: Die Klemme empfängt nur Daten anderer Teilnehmer
1	1	RS422: Der Empfänger wird erst freigeschaltet, nachdem die letzten Daten gesendet wurden

**4.3 EL6002, EL6022**

**4.3.1 Montage und Demontage - Frontriegelung oben**

Die Klemmenmodule werden mit Hilfe einer 35 mm Tragschiene (z.B. Hutschiene TH 35-15) auf der Montagefläche befestigt.

### **i** Tragschienenbefestigung

Der Verriegelungsmechanismus der Klemmen reicht in das Profil der Tragschiene hinein. Achten Sie bei der Montage der Komponenten darauf, dass der Verriegelungsmechanismus nicht in Konflikt mit den Befestigungsschrauben der Tragschiene gerät. Verwenden Sie zur Befestigung der empfohlenen Tragschienen unter den Klemmen flache Montageverbindungen wie Senkkopfschrauben oder Blindnieten.

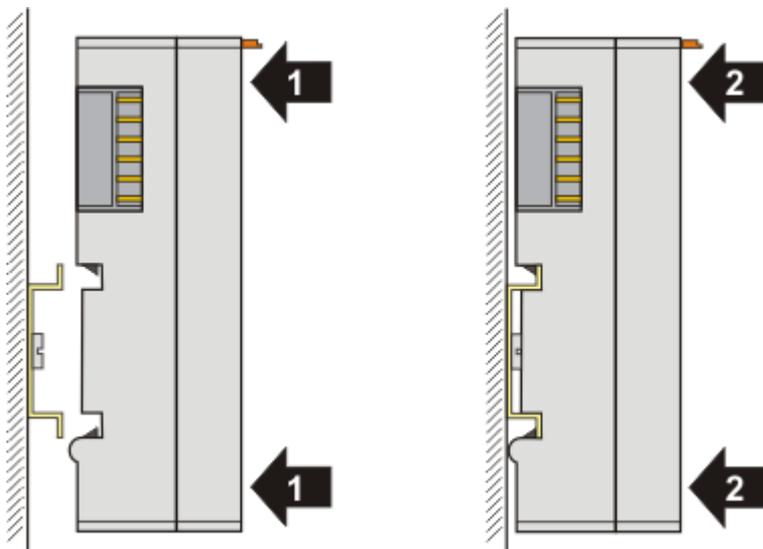
### **⚠** WARNUNG

#### **Verletzungsgefahr durch Stromschlag und Beschädigung des Gerätes möglich!**

Setzen Sie das Busklemmen-System in einen sicheren, spannungslosen Zustand, bevor Sie mit der Montage, Demontage oder Verdrahtung der Busklemmen beginnen!

### Montage

- Montieren Sie die Tragschiene an der vorgesehenen Montagestelle

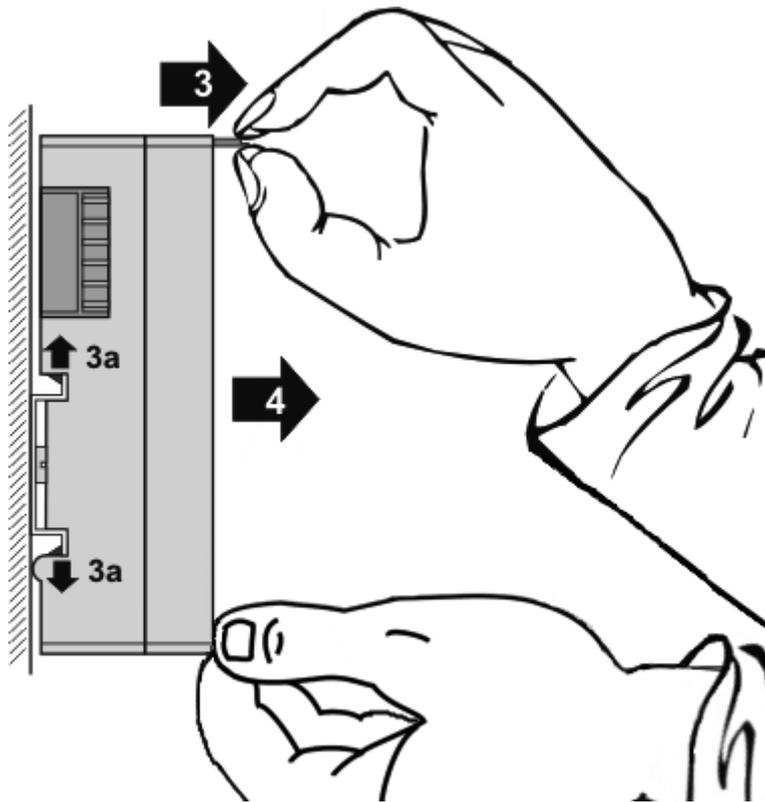


und drücken Sie (1) das Klemmenmodul gegen die Tragschiene, bis es auf der Tragschiene einrastet (2).

- Schließen Sie die Leitungen an.

### Demontage

- Entfernen Sie alle Leitungen.
- Ziehen Sie mit Daumen und Zeigefinger die orange Entriegelungslasche (3) zurück. Dabei ziehen sich über einen internen Mechanismus die beiden Rastnasen (3a) an der Hutschiene ins Klemmenmodul zurück.



- Ziehen Sie (4) das Klemmenmodul von der Montagefläche weg. Vermeiden Sie ein Verkanten; stabilisieren Sie das Modul ggf. mit der freien Hand

### 4.3.2 Empfohlene Tragschienen

Klemmenmodule und EtherCAT-Module der Serien KMxxxx, EMxxxx, sowie Klemmen der Serien EL66xx und EL67xx können Sie auf folgende Tragschienen aufrasten:

- Hutschiene TH 35-7.5 mit 1 mm Materialstärke (nach EN 60715)
- Hutschiene TH 35-15 mit 1,5 mm Materialstärke

#### ● Materialstärke der Hutschiene beachten

**i** Klemmenmodule und EtherCAT-Module der Serien KMxxxx, EMxxxx, sowie Klemmen der Serien EL66xx und EL67xx **passen nicht auf** die Hutschiene TH 35-15 mit **2,2 bis 2,5 mm Materialstärke** (nach EN 60715)!

### 4.3.3 LEDs und Anschlussbelegung

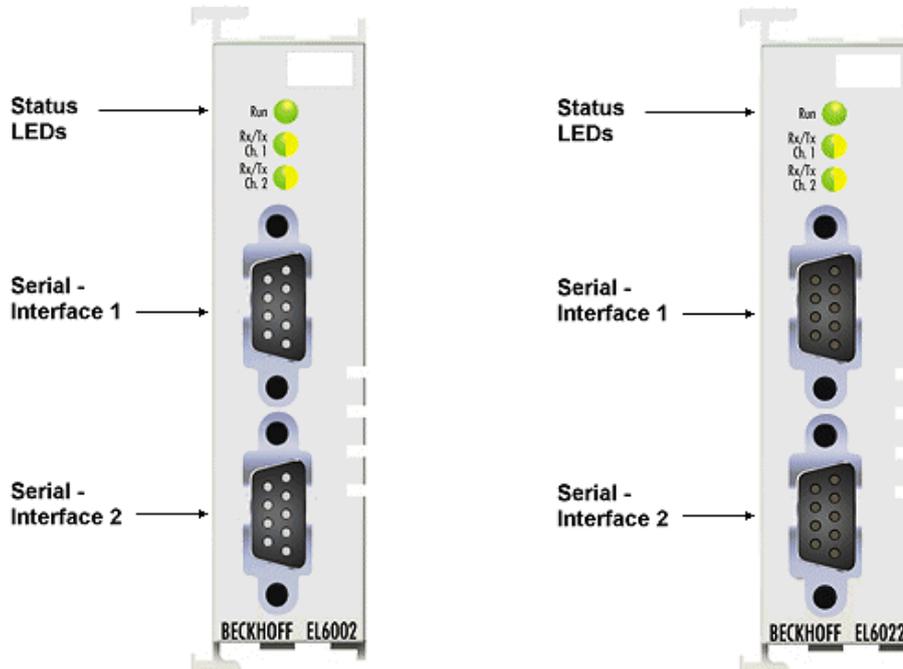


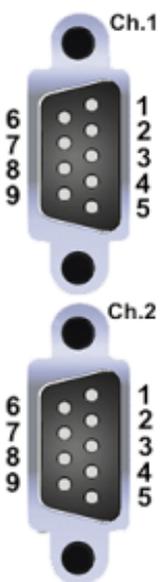
Abb. 35: EL6002, EL6022 - LEDs

#### LEDs

LED	Farbe	Bedeutung	
RUN	Grün	Diese LED gibt den Betriebszustand der Klemme wieder:	
		Aus	Zustand der EtherCAT State Machine [► 28]: INIT = Initialisierung der Klemme oder BOOTSTRAP = Funktion für Firmware Updates [► 196] der Klemme
		Blinkend	Zustand der EtherCAT State Machine: PREOP = Funktion für Mailbox-Kommunikation und abweichende Standard-Einstellungen gesetzt
		Einzelblitz	Zustand der EtherCAT State Machine: SAFEOP = Überprüfung der Kanäle des Sync-Managers [► 112] und der Distributed Clocks. Ausgänge bleiben im sicheren Zustand
An	Zustand der EtherCAT State Machine: OP = normaler Betriebszustand; Mailbox- und Prozessdatenkommunikation ist möglich		
TxCh. 1	Orange	Serieller Port an diesem Anschluss sendet Daten (Kanal 1)	
RxCh. 1	Grün	Serieller Port an diesem Anschluss empfängt Daten (Kanal 1)	
TxCh. 2	Orange	Serieller Port an diesem Anschluss sendet Daten (Kanal 2)	
RxCh. 2	Grün	Serieller Port an diesem Anschluss empfängt Daten (Kanal 2)	

**Anschlussbelegung EL6002**

2 x D-Sub Anschlussstecker; 9-polig

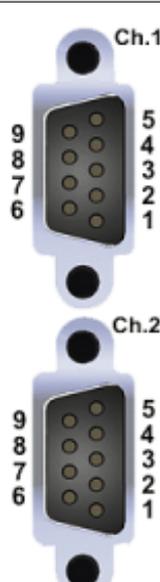
D-Sub Anschlussstecker, male (Draufsicht)	Pin- Belegung Kanal 1		Pin- Belegung Kanal 2	
	Pin	RS232	Pin	RS232
	1	DCD (intern gebrückt mit PIN 4 und 6)	1	DCD (intern gebrückt mit PIN 4 und 6)
	2	RxCH1	2	RxCH2
	3	TxCH1	3	TxCH2
	4	DTR (intern gebrückt mit PIN 1 und 6)	4	DTR (intern gebrückt mit PIN 1 und 6)
	5	GND	5	GND
	6	DSR (intern gebrückt mit PIN 1 und 4)	6	DSR (intern gebrückt mit PIN 1 und 4)
	7	RTS CH1	7	RTS CH2
	8	CTS CH1	8	CTS CH2
	9	-	9	-

**i GND Anschlüsse**

GND beider Kanäle ist intern über eine hochohmige RC-Kombination verbunden

**Anschlussbelegung EL6022**

2 x D-Sub Anschlussbuchse, 9-polig

D-Sub Anschlussstecker, female (Draufsicht)	Pin- Belegung Kanal 1		Pin- Belegung Kanal 2	
	Pin	RS485/RS422	Pin	RS485/RS422
	1	-	1	-
	2	Tx+ CH1	2	Tx+ CH2
	3	Rx+ CH1	3	Rx+ CH2
	4	-	4	-
	5	GND	5	GND
	6	+5 V	6	+5 V
	7	Tx- CH1	7	Tx- CH2
	8	Rx- CH1	8	Rx- CH2
	9	-	9	-

**i GND Anschlüsse**

GND beider Kanäle ist intern über eine hochohmige RC-Kombination verbunden

**Anschluss bei RS422 Übertragung**

Bei der RS422 Betriebsart können die Daten voll duplex übertragen werden. Es können nur Punkt zu Punkt Verbindungen hergestellt werden.

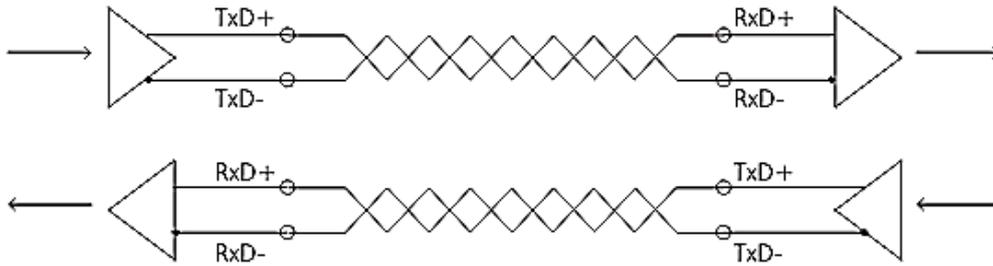


Abb. 36: Anschluss bei RS422 Übertragung

**Anschluss bei RS485 Übertragung**

Bei der RS485 Betriebsart werden die Daten in halbduplex Übertragung ausgetauscht. In dieser Betriebsart kann eine Busstruktur aufgebaut werden.

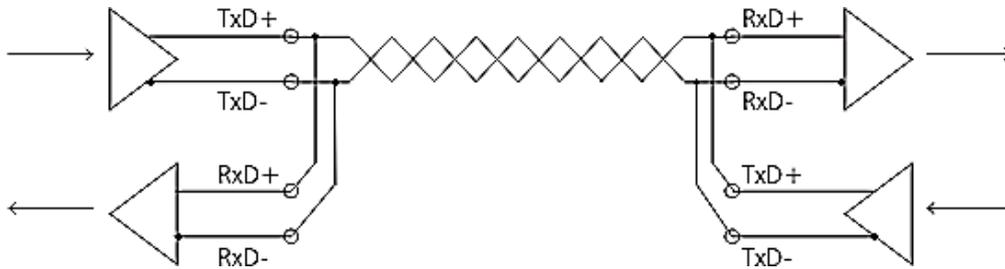


Abb. 37: Anshchluss bei RS485 Übertragung

Im RS485 Betriebsmodus werden die Sendeleitungen miteinander verbunden. Dadurch empfängt die Klemme nicht nur die Daten anderer Teilnehmer, sondern auch ihre eigenen Sendedaten. Mit dem Index 0x8000:06 "Enable half duplex" im Settings-Objekt kann dies unterdrückt werden.

Das Empfangen neuer Daten wird in der Betriebsart RS485 erst möglich, wenn das Senden abgeschlossen ist.

"Enable half duplex"	"Enable point to point connection (RS422)"	Modus
0	0	RS485: Die Klemme empfängt ihre eigenen und die Daten anderer Teilnehmer
0	1	RS422: Normaler Betriebsmodus; Die Klemme arbeitet im Vollduplex-Modus.
1	0	RS485: Die Klemme empfängt nur Daten anderer Teilnehmer
1	1	RS422: Der Empfänger wird erst freigeschaltet, nachdem die letzten Daten gesendet wurden

## 4.4 Positionierung von passiven Klemmen

### **i** Hinweis zur Positionierung von passiven Klemmen im Busklemmenblock

EtherCAT-Klemmen (ELxxxx / ESxxxx), die nicht aktiv am Datenaustausch innerhalb des Busklemmenblocks teilnehmen, werden als passive Klemmen bezeichnet. Zu erkennen sind diese Klemmen an der nicht vorhandenen Stromaufnahme aus dem E-Bus. Um einen optimalen Datenaustausch zu gewährleisten, dürfen nicht mehr als zwei passive Klemmen direkt aneinander gereiht werden!

#### Beispiele für die Positionierung von passiven Klemmen (hell eingefärbt)

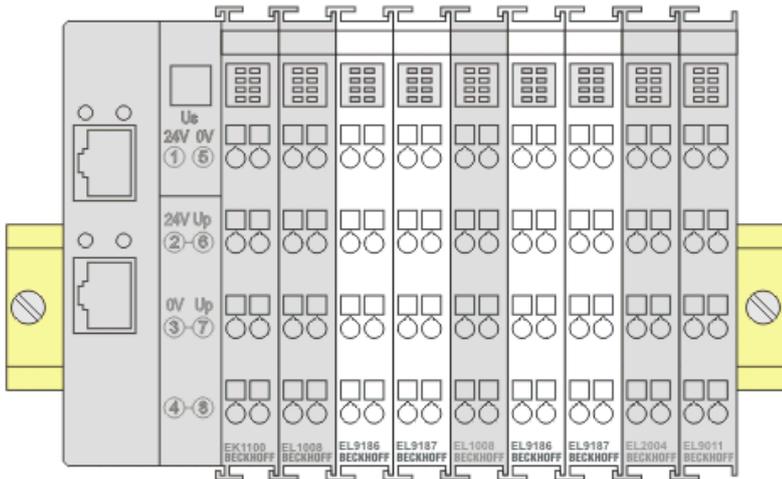


Abb. 38: Korrekte Positionierung

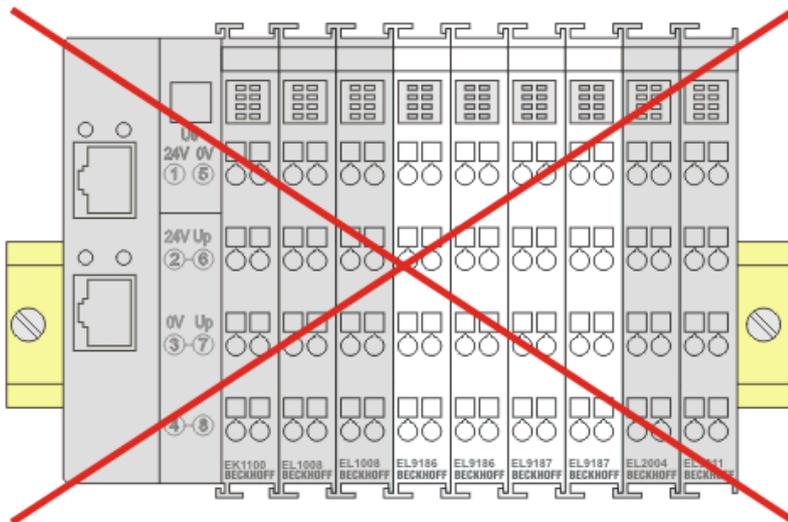


Abb. 39: Inkorrekte Positionierung

## 4.5 Montagevorschriften für erhöhte mechanische Belastbarkeit

### ⚠️ WARNUNG

#### Verletzungsgefahr durch Stromschlag und Beschädigung des Gerätes möglich!

Setzen Sie das Busklemmen-System in einen sicheren, spannungslosen Zustand, bevor Sie mit der Montage, Demontage oder Verdrahtung der Busklemmen beginnen!

#### Zusätzliche Prüfungen

Die Klemmen sind folgenden zusätzlichen Prüfungen unterzogen worden:

Prüfung	Erläuterung
Vibration	10 Frequenzdurchläufe, in 3-Achsen
	6 Hz < f < 60 Hz Auslenkung 0,35 mm, konstante Amplitude
	60,1 Hz < f < 500 Hz Beschleunigung 5 g, konstante Amplitude
Schocken	1000 Schocks je Richtung, in 3-Achsen
	25 g, 6 ms

#### Zusätzliche Montagevorschriften

Für die Klemmen mit erhöhter mechanischer Belastbarkeit gelten folgende zusätzliche Montagevorschriften:

- Die erhöhte mechanische Belastbarkeit gilt für alle zulässigen Einbaulagen
- Es ist eine Tragschiene nach EN 60715 TH35-15 zu verwenden
- Der Klemmenstrang ist auf beiden Seiten der Tragschiene durch eine mechanische Befestigung, z.B. mittels einer Erdungsklemme oder verstärkten Endklammer zu fixieren
- Die maximale Gesamtausdehnung des Klemmenstrangs (ohne Koppler) beträgt:  
64 Klemmen mit 12 mm oder 32 Klemmen mit 24 mm Einbaubreite
- Bei der Abkantung und Befestigung der Tragschiene ist darauf zu achten, dass keine Verformung und Verdrehung der Tragschiene auftritt, weiterhin ist kein Quetschen und Verbiegen der Tragschiene zulässig
- Die Befestigungspunkte der Tragschiene sind in einem Abstand vom 5 cm zu setzen
- Zur Befestigung der Tragschiene sind Senkkopfschrauben zu verwenden
- Die freie Leiterlänge zwischen Zugentlastung und Leiteranschluss ist möglichst kurz zu halten; der Abstand zum Kabelkanal ist mit ca. 10 cm zu einhalten

## 4.6 Einbaulagen

### HINWEIS

#### Einschränkung von Einbaulage und Betriebstemperaturbereich

Entnehmen Sie den technischen Daten zu einer Klemme, ob sie Einschränkungen bei Einbaulage und/oder Betriebstemperaturbereich unterliegt. Sorgen Sie bei der Montage von Klemmen mit erhöhter thermischer Verlustleistung dafür, dass im Betrieb oberhalb und unterhalb der Klemmen ausreichend Abstand zu anderen Komponenten eingehalten wird, so dass die Klemmen ausreichend belüftet werden!

#### Optimale Einbaulage (Standard)

Für die optimale Einbaulage wird die Tragschiene waagrecht montiert und die Anschlussflächen der EL/KL-Klemmen weisen nach vorne (siehe Abb. *Empfohlene Abstände bei Standard-Einbaulage*). Die Klemmen werden dabei von unten nach oben durchlüftet, was eine optimale Kühlung der Elektronik durch Konvektionslüftung ermöglicht. Bezugsrichtung „unten“ ist hier die Erdbeschleunigung.

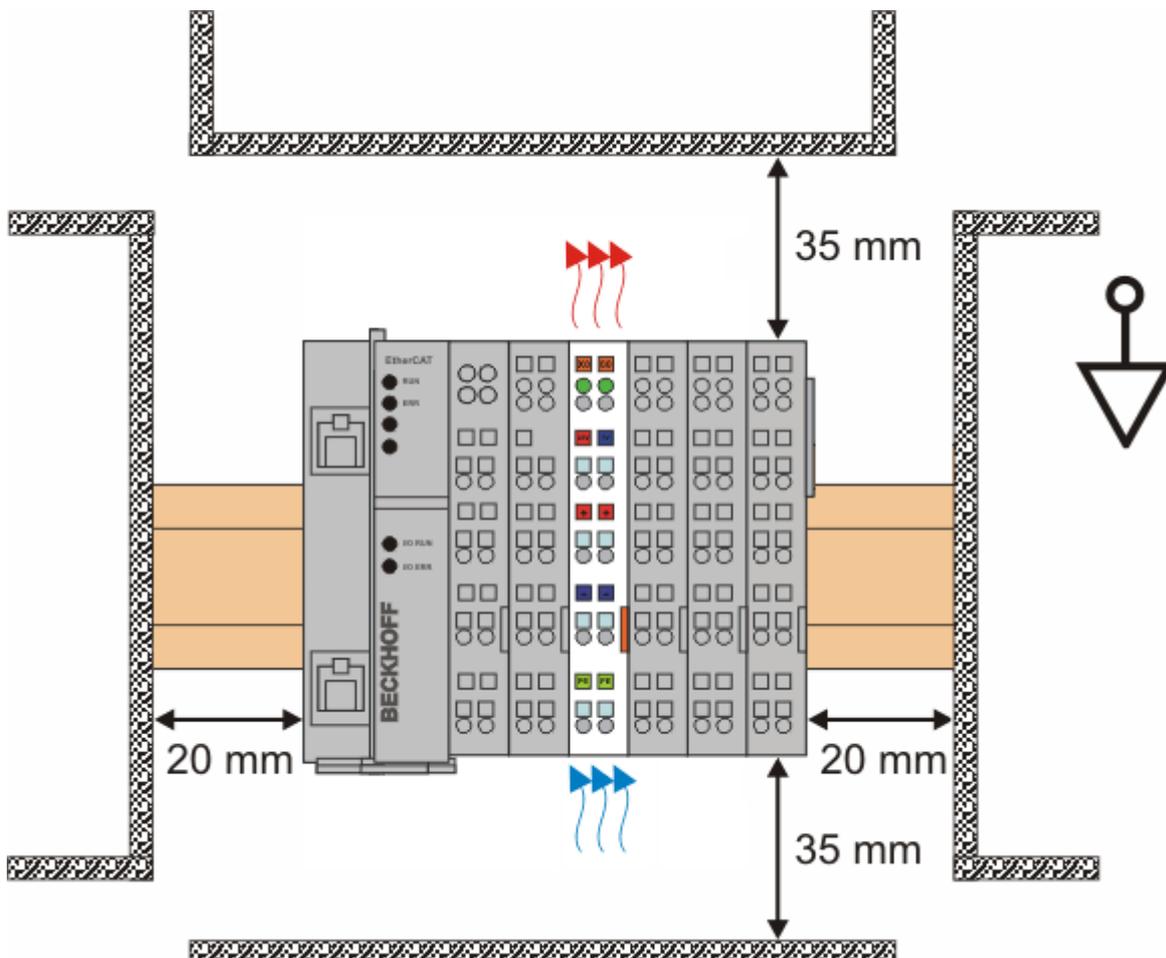


Abb. 40: Empfohlene Abstände bei Standard-Einbaulage

Die Einhaltung der Abstände nach Abb. *Empfohlene Abstände bei Standard-Einbaulage* wird empfohlen.

#### Weitere Einbaulagen

Alle anderen Einbaulagen zeichnen sich durch davon abweichende räumliche Lage der Tragschiene aus, siehe Abb. *Weitere Einbaulagen*.

Auch in diesen Einbaulagen empfiehlt sich die Anwendung der oben angegebenen Mindestabstände zur Umgebung.

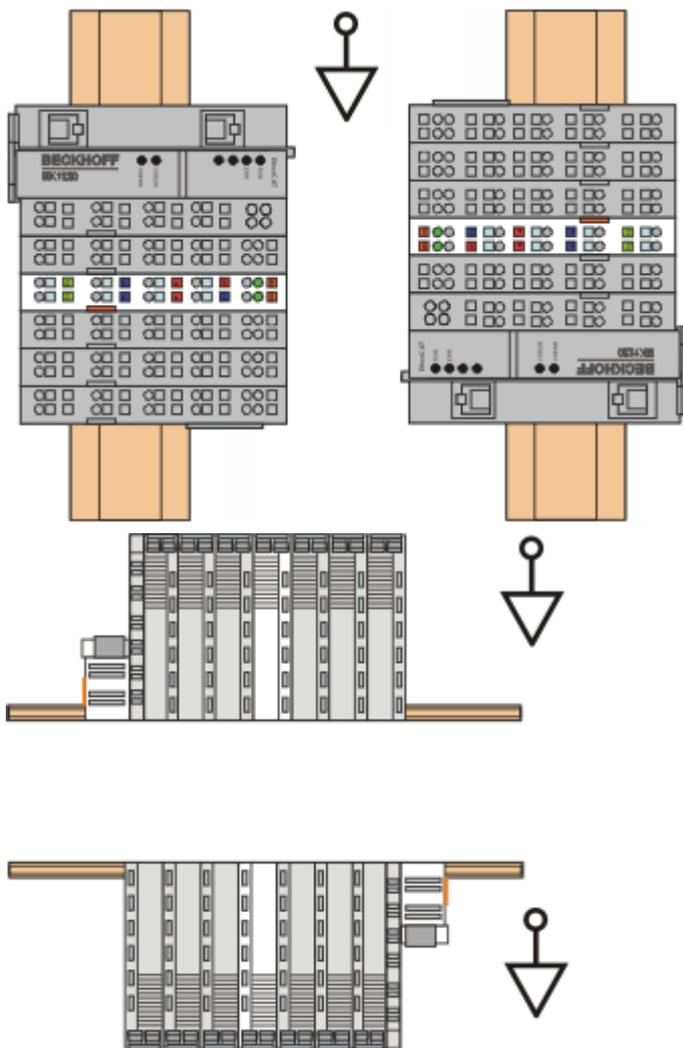


Abb. 41: Weitere Einbaulagen

## 4.7 UL-Hinweise

	<p><b>Application</b> The modules are intended for use with Beckhoff's UL Listed EtherCAT System only.</p>
	<p><b>Examination</b> For cULus examination, the Beckhoff I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).</p>
	<p><b>For devices with Ethernet connectors</b> Not for connection to telecommunication circuits.</p>

### Grundlagen

UL-Zertifizierung nach UL508. Solcherart zertifizierte Geräte sind gekennzeichnet durch das Zeichen:



## 4.8 ATEX - Besondere Bedingungen (erweiterter Temperaturbereich)

### ⚠️ WARNUNG

**Beachten Sie die besonderen Bedingungen für die bestimmungsgemäße Verwendung von Beckhoff-Feldbuskomponenten mit erweitertem Temperaturbereich (ET) in explosionsgefährdeten Bereichen (Richtlinie 2014/34/EU)!**

- Die zertifizierten Komponenten sind in ein geeignetes Gehäuse zu errichten, das eine Schutzart von mindestens IP54 gemäß EN 60079-15 gewährleistet! Dabei sind die Umgebungsbedingungen bei der Verwendung zu berücksichtigen!
- Für Staub (nur die Feldbuskomponenten der Zertifikatsnummer KEMA 10ATEX0075 X Issue 9): Das Gerät ist in ein geeignetes Gehäuse einzubauen, das eine Schutzart von IP54 gemäß EN 60079-0 für Gruppe IIIA oder IIIB und IP6X für Gruppe IIIC bietet, wobei die Umgebungsbedingungen, unter denen das Gerät verwendet wird, zu berücksichtigen sind.
- Beachten Sie für Beckhoff-Feldbuskomponenten mit erweitertem Temperaturbereich (ET) beim Einsatz in explosionsgefährdeten Bereichen den zulässigen Umgebungstemperaturbereich von -25 bis 60°C!
- Es müssen Maßnahmen zum Schutz gegen Überschreitung der Nennbetriebsspannung durch kurzzeitige Störspannungen um mehr als 40% getroffen werden!
- Die einzelnen Klemmen dürfen nur aus dem Busklemmensystem gezogen oder entfernt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Anschlüsse der zertifizierten Komponenten dürfen nur verbunden oder unterbrochen werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Sicherung der Einspeiseklemmen KL92xx/EL92xx dürfen nur gewechselt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Adresswahlschalter und ID-Switche dürfen nur eingestellt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!

### Normen

Die grundlegenden Sicherheits- und Gesundheitsanforderungen werden durch Übereinstimmung mit den folgenden Normen erfüllt:

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010
- EN 60079-31:2013 (nur für Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

## Kennzeichnung

Die gemäß ATEX-Richtlinie für den explosionsgefährdeten Bereich zertifizierten Beckhoff-Feldbuskomponenten mit erweitertem Temperaturbereich (ET) tragen die folgende Kennzeichnung:



**II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: -25 ... +60°C**

II 3D KEMA 10ATEX0075 X Ex tc IIC T135°C Dc Ta: -25 ... +60°C

(nur für Feldbuskomponenten mit Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

oder



**II 3G KEMA 10ATEX0075 X Ex nC IIC T4 Gc Ta: -25 ... +60°C**

II 3D KEMA 10ATEX0075 X Ex tc IIC T135°C Dc Ta: -25 ... +60°C

(nur für Feldbuskomponenten mit Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

## 4.9 Weiterführende Dokumentation zum Explosionsschutz

### ● Explosionsschutz für Klemmensysteme

**i** Beachten Sie auch die weiterführende Dokumentation

Hinweise zum Einsatz der Beckhoff Klemmensysteme in explosionsgefährdeten Bereichen gemäß ATEX und IECEx

die Ihnen auf der Beckhoff-Homepage <https://www.beckhoff.de> im Bereich Download zur Verfügung steht!

## 5 Inbetriebnahme

### 5.1 TwinCAT Quickstart

TwinCAT stellt eine Entwicklungsumgebung für Echtzeitsteuerung mit Multi-SPS-System, NC Achsregelung, Programmierung und Bedienung dar. Das gesamte System wird hierbei durch diese Umgebung abgebildet und ermöglicht Zugriff auf eine Programmierumgebung (inkl. Kompilierung) für die Steuerung. Einzelne digitale oder analoge Eingänge bzw. Ausgänge können auch direkt ausgelesen bzw. beschrieben werden, um diese z.B. hinsichtlich ihrer Funktionsweise zu überprüfen.

Weitere Informationen hierzu erhalten Sie unter <http://infosys.beckhoff.de>:

- **EtherCAT Systemhandbuch:**  
Feldbuskomponenten → EtherCAT-Klemmen → EtherCAT System Dokumentation → Einrichtung im TwinCAT System Manager
- **TwinCAT 2** → TwinCAT System Manager → E/A- Konfiguration
- Insbesondere zur TwinCAT – Treiberinstallation:  
**Feldbuskomponenten** → Feldbuskarten und Switche → FC900x – PCI-Karten für Ethernet → Installation

Geräte, d. h. „devices“ beinhalten jeweils die Klemmen der tatsächlich aufgebauten Konfiguration. Dabei gibt es grundlegend die Möglichkeit sämtliche Informationen des Aufbaus über die „Scan“ - Funktion einzubringen („online“) oder über Editorfunktionen direkt einzufügen („offline“):

- **„offline“:** der vorgesehene Aufbau wird durch Hinzufügen und entsprechendes Platzieren einzelner Komponenten erstellt. Diese können aus einem Verzeichnis ausgewählt und Konfiguriert werden.
  - Die Vorgehensweise für den „offline“ – Betrieb ist unter <http://infosys.beckhoff.de> einsehbar:  
**TwinCAT 2** → TwinCAT System Manager → EA - Konfiguration → Anfügen eines E/A-Gerätes
- **„online“:** die bereits physikalisch aufgebaute Konfiguration wird eingelesen
  - Sehen Sie hierzu auch unter <http://infosys.beckhoff.de>:  
**Feldbuskomponenten** → Feldbuskarten und Switche → FC900x – PCI-Karten für Ethernet → Installation → Geräte suchen

Vom Anwender –PC bis zu den einzelnen Steuerungselementen ist folgender Zusammenhang vorgesehen:

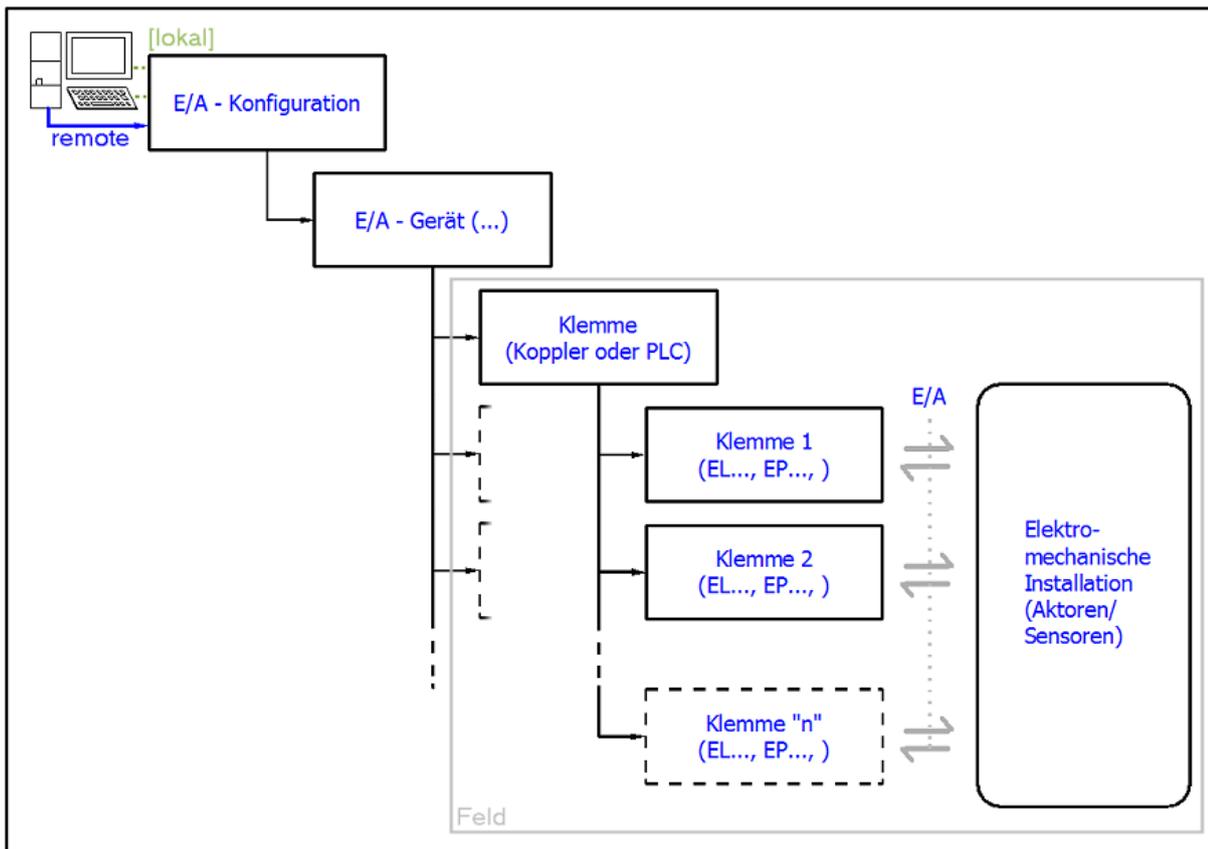


Abb. 42: Bezug von der Anwender Seite (Inbetriebnahme) zur Installation

Das anwenderseitige Einfügen bestimmter Komponenten (E/A – Gerät, Klemme, Box,...) erfolgt bei TwinCAT 2 und TwinCAT 3 auf die gleiche Weise. In den nachfolgenden Beschreibungen wird ausschließlich der „online“ Vorgang angewandt.

### Beispielkonfiguration (realer Aufbau)

Ausgehend von der folgenden Beispielkonfiguration wird in den anschließenden Unterkapiteln das Vorgehen für TwinCAT 2 und TwinCAT 3 behandelt:

- Steuerungssystem (PLC) **CX2040** inkl. Netzteil **CX2100-0004**
- Rechtsseitig angebunden am CX2040 (E-Bus):  
**EL1004** (4-Kanal-Digital-Eingangsklemme 24 V<sub>DC</sub>)
- Über den X001 Anschluss (RJ-45) angeschlossen: **EK1100** EtherCAT-Koppler
- Rechtsseitig angebunden am EK1100 EtherCAT-Koppler (E-Bus):  
**EL2008** (8-Kanal-Digital-Ausgangsklemme 24 V<sub>DC</sub>; 0,5 A)
- (Optional über X000: ein Link zu einen externen PC für die Benutzeroberfläche)

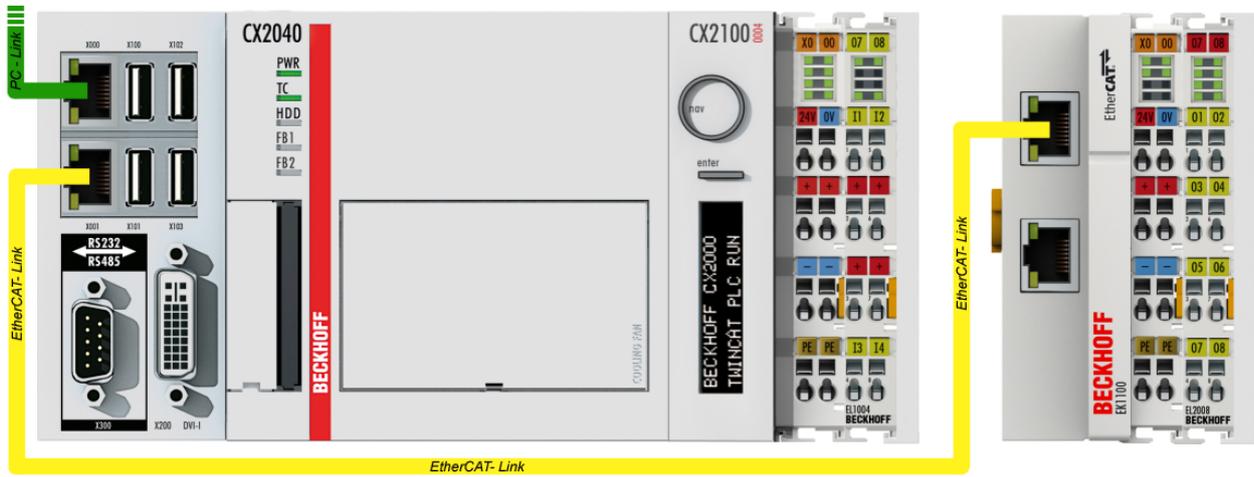


Abb. 43: Aufbau der Steuerung mit Embedded-PC, Eingabe (EL1004) und Ausgabe (EL2008)

Anzumerken ist, dass sämtliche Kombinationen einer Konfiguration möglich sind; beispielsweise könnte die Klemme EL1004 ebenso auch nach dem Koppler angesteckt werden oder die Klemme EL2008 könnte zusätzlich rechts an dem CX2040 angesteckt sein – dann wäre der Koppler EK1100 überflüssig.

## 5.1.1 TwinCAT 2

### Startup

TwinCAT 2 verwendet grundlegend zwei Benutzeroberflächen: den „TwinCAT System Manager“ zur Kommunikation mit den elektromechanischen Komponenten und „TwinCAT PLC Control“ für die Erstellung und Kompilierung einer Steuerung. Begonnen wird zunächst mit der Anwendung des „TwinCAT System Manager“.

Nach erfolgreicher Installation des TwinCAT-Systems auf den Anwender PC der zur Entwicklung verwendet werden soll, zeigt der TwinCAT 2 (System Manager) folgende Benutzeroberfläche nach dem Start:

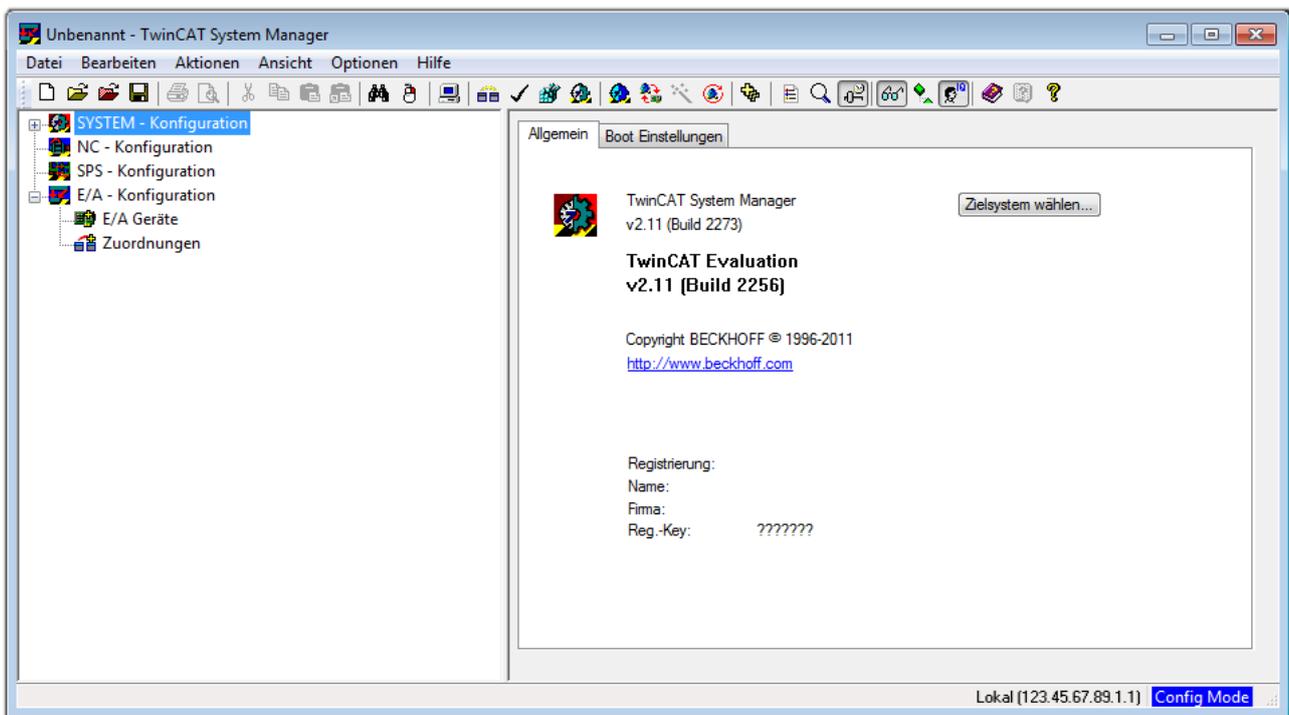


Abb. 44: Initiale Benutzeroberfläche TwinCAT 2

Es besteht generell die Möglichkeit das TwinCAT „lokal“ oder per „remote“ zu verwenden. Ist das TwinCAT System inkl. Benutzeroberfläche (Standard) auf dem betreffenden PLC installiert, kann TwinCAT „lokal“ eingesetzt werden und mit Schritt „Geräte einfügen [▶ 65]“ fortgesetzt werden.

Ist es vorgesehen, die auf einem PLC installierte TwinCAT Laufzeitumgebung von einem anderen System als Entwicklungsumgebung per „remote“ anzusprechen, ist das Zielsystem zuvor bekannt zu machen. Im

Menü unter „Aktionen“ → „Auswahl des Zielsystems...“, über das Symbol „“ oder durch Taste „F8“ wird folgendes Fenster hierzu geöffnet:

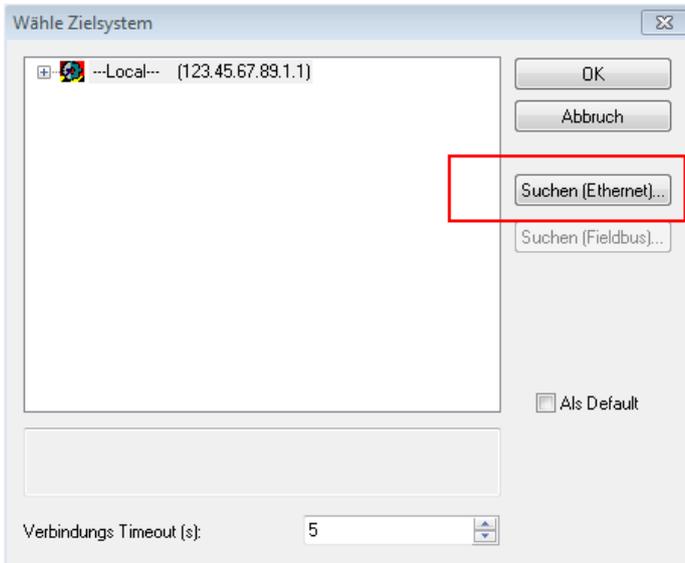


Abb. 45: Wähle Zielsystem

Mittels „Suchen (Ethernet)...“ wird das Zielsystem eingetragen. Dadurch wird ein weiterer Dialog geöffnet um hier entweder:

- den bekannten Rechnernamen hinter „Enter Host Name / IP:“ einzutragen (wie rot gekennzeichnet)
- einen „Broadcast Search“ durchzuführen (falls der Rechnernamen nicht genau bekannt)
- die bekannte Rechner - IP oder AmsNetId einzutragen

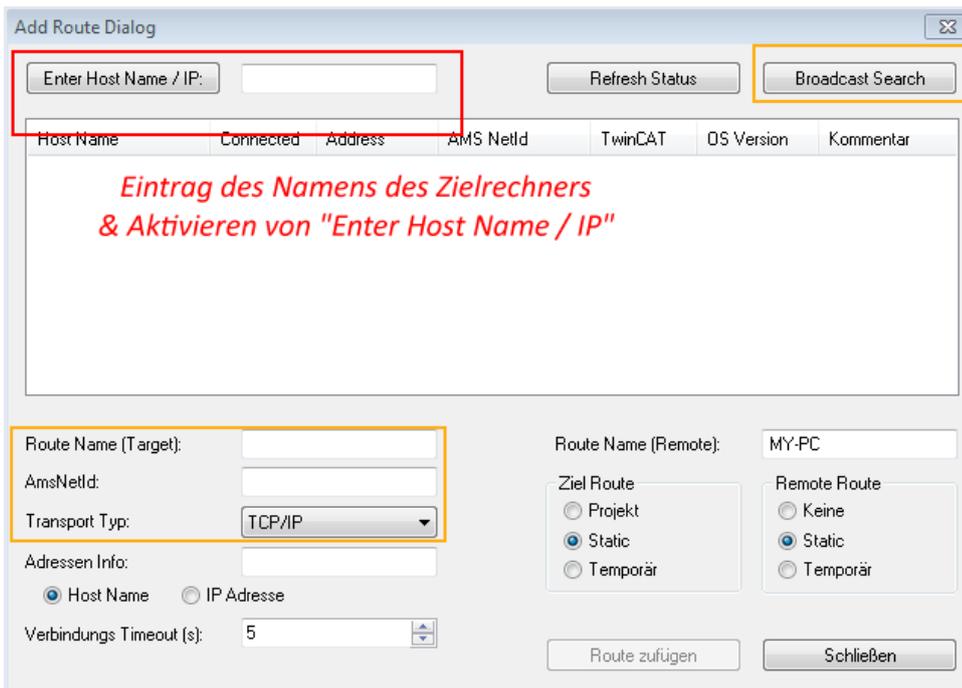
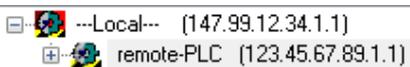


Abb. 46: PLC für den Zugriff des TwinCAT System Managers festlegen: Auswahl des Zielsystems

Ist das Zielsystem eingetragen steht dieses wie folgt zur Auswahl (ggf. muss zuvor das korrekte Passwort eingetragen werden):



Nach der Auswahl mit „OK“ ist das Zielsystem über den System Manager ansprechbar.

**Geräte einfügen**

In dem linksseitigen Konfigurationsbaum der TwinCAT 2 – Benutzeroberfläche des System Managers wird „E/A Geräte“ selektiert und sodann entweder über Rechtsklick ein Kontextmenü geöffnet und „Geräte

Suchen...“ ausgewählt oder in der Menüleiste mit  die Aktion gestartet. Ggf. ist zuvor der TwinCAT

System Manager in den „Konfig Modus“ mittels  oder über das Menü „Aktionen“ → „Startet/ Restarten von TwinCAT in Konfig-Modus“(Shift + F4) zu versetzen.

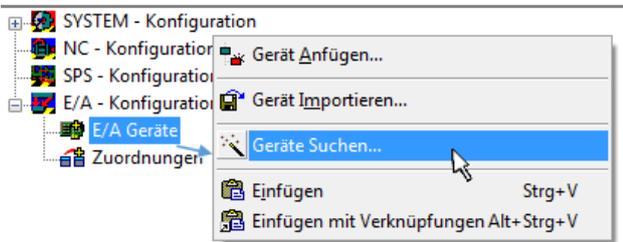


Abb. 47: Auswahl „Gerät Suchen..“

Die darauf folgende Hinweismeldung ist zu bestätigen und in dem Dialog die Geräte „EtherCAT“ zu wählen:

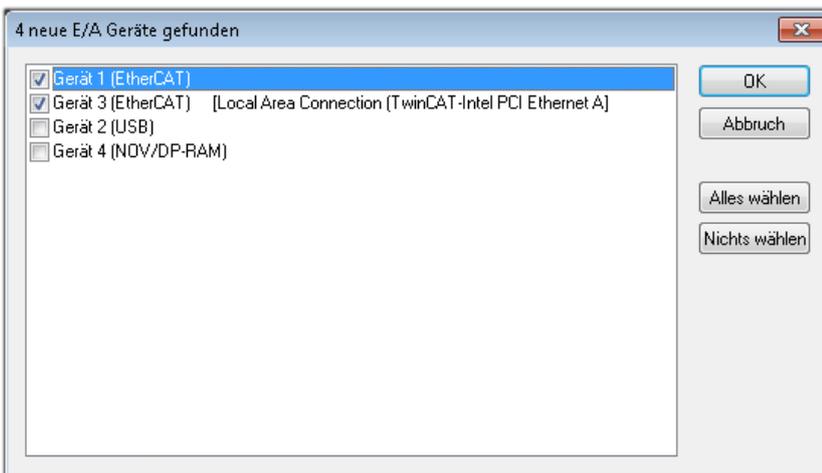


Abb. 48: Automatische Erkennung von E/A Geräten: Auswahl der einzubindenden Geräte

Ebenfalls ist anschließend die Meldung „nach neuen Boxen suchen“ zu bestätigen, um die an den Geräten angebotenen Klemmen zu ermitteln. „Free Run“ erlaubt das Manipulieren von Ein- und Ausgangswerten innerhalb des „Config Modus“ und sollte ebenfalls bestätigt werden.

Ausgehend von der am Anfang dieses Kapitels beschriebenen Beispielkonfiguration [► 61] sieht das Ergebnis wie folgt aus:

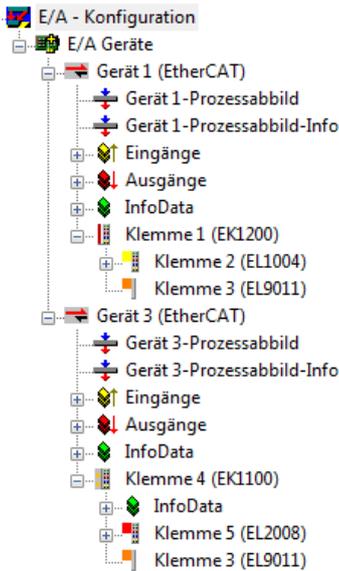


Abb. 49: Abbildung der Konfiguration im TwinCAT 2 System Manager

Der gesamte Vorgang setzt sich aus zwei Stufen zusammen, die auch separat ausgeführt werden können (erst das Ermitteln der Geräte, dann das Ermitteln der daran befindlichen Elemente wie Boxen, Klemmen o. ä.). So kann auch durch Markierung von „Gerät ..“ aus dem Kontextmenü eine „Suche“ Funktion (Scan) ausgeführt werden, die hierbei dann lediglich die darunter liegenden (im Aufbau vorliegenden) Elemente einliest:

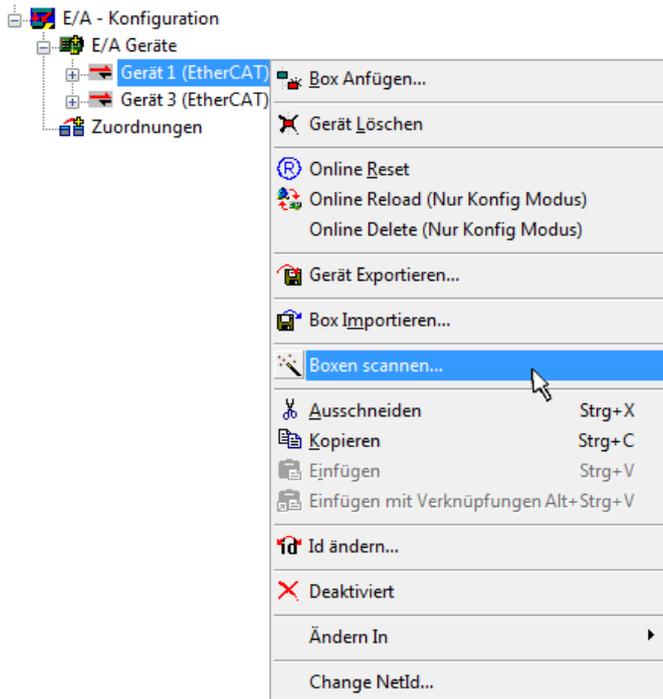


Abb. 50: Einlesen von einzelnen an einem Gerät befindlichen Klemmen

Diese Funktionalität ist nützlich, falls die Konfiguration (d. h. der „reale Aufbau“) kurzfristig geändert wird.

**PLC programmieren und integrieren**

TwinCAT PLC Control ist die Entwicklungsumgebung zur Erstellung der Steuerung in unterschiedlichen Programmumgebungen: Das TwinCAT PLC Control unterstützt alle in der IEC 61131-3 beschriebenen Sprachen. Es gibt zwei textuelle Sprachen und drei grafische Sprachen.

- **Textuelle Sprachen**
  - Anweisungsliste (AWL, IL)

- Strukturierter Text (ST)
- **Grafische Sprachen**
  - Funktionsplan (FUP, FBD)
  - Kontaktplan (KOP, LD)
  - Freigrafischer Funktionsplaneditor (CFC)
  - Ablaufsprache (AS, SFC)

Für die folgenden Betrachtungen wird lediglich vom strukturierten Text (ST) Gebrauch gemacht.

Nach dem Start von TwinCAT PLC Control wird folgende Benutzeroberfläche für ein initiales Projekt dargestellt:

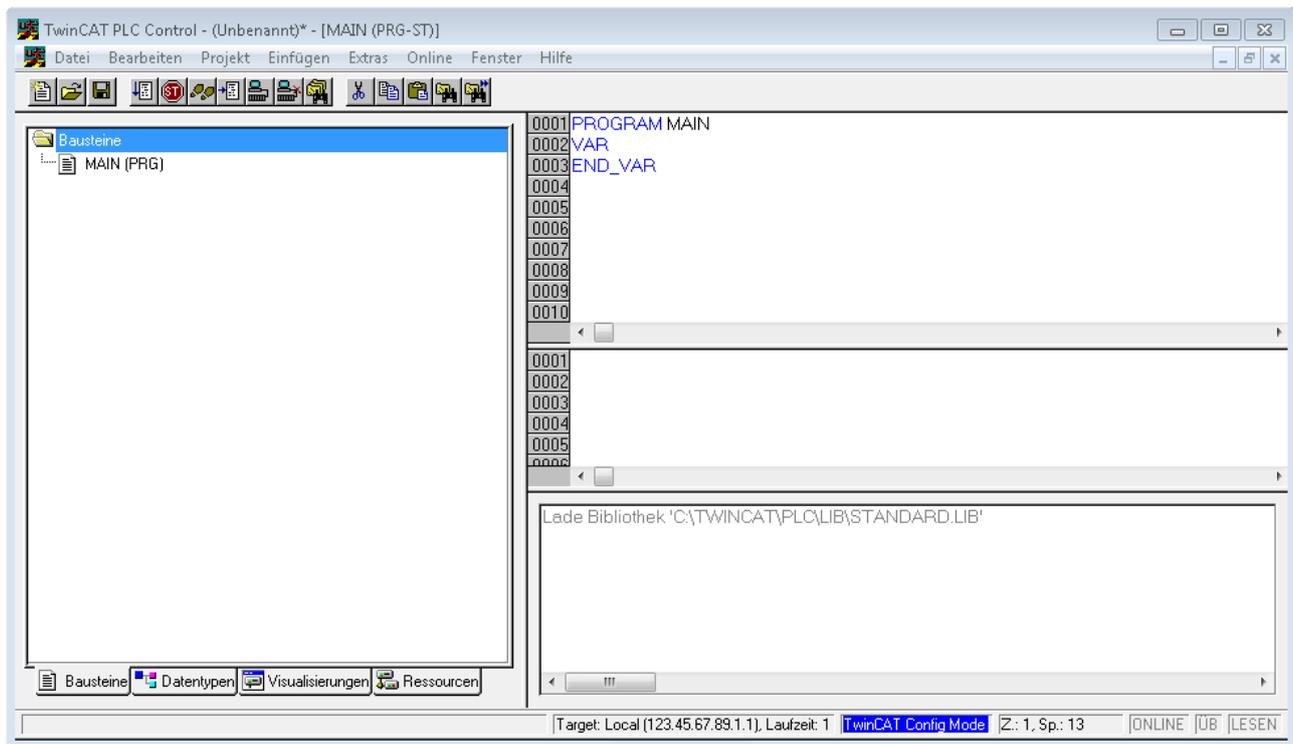


Abb. 51: TwinCAT PLC Control nach dem Start

Nun sind für den weiteren Ablauf Beispielvariablen sowie ein Beispielprogramm erstellt und unter dem Namen „PLC\_example.pro“ gespeichert worden:

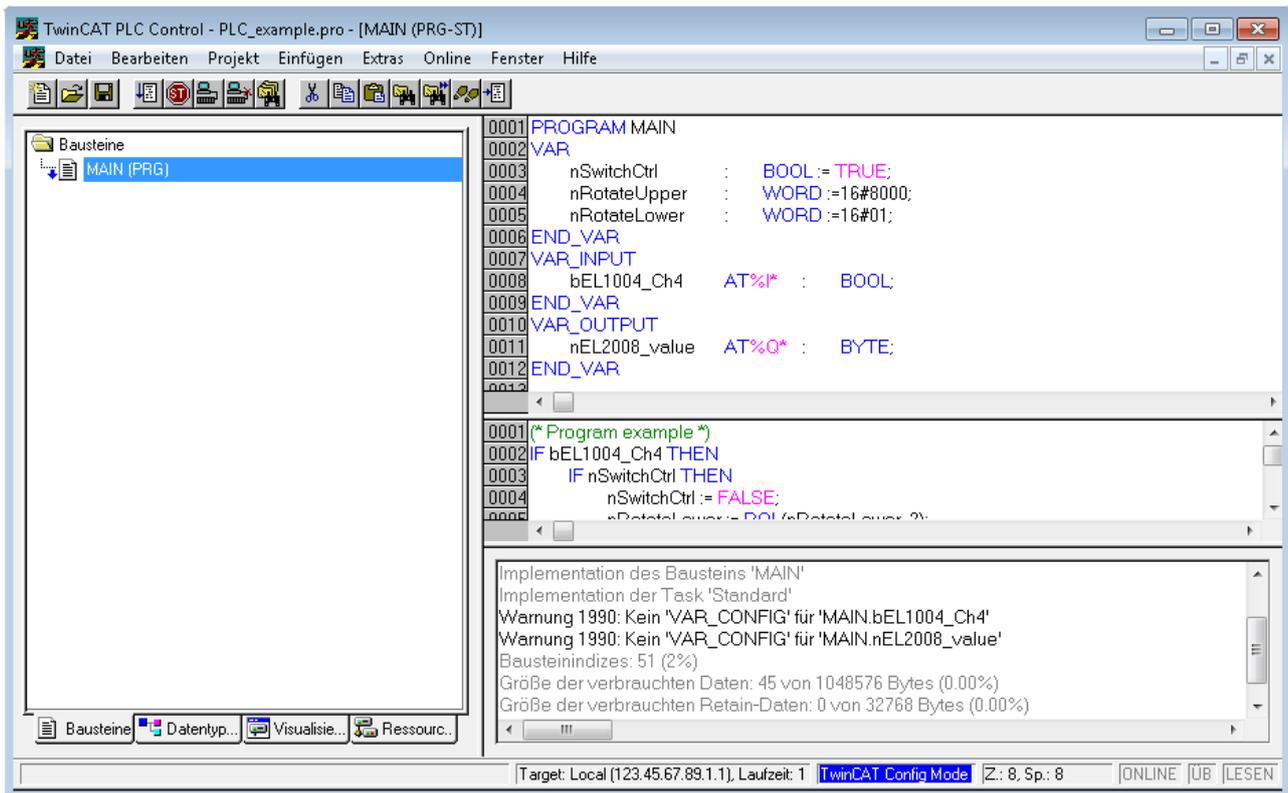


Abb. 52: Beispielprogramm mit Variablen nach einem Kompilervorgang (ohne Variablenanbindung)

Die Warnung 1990 (fehlende „VAR\_CONFIG“) nach einem Kompilervorgang zeigt auf, dass die als extern definierten Variablen (mit der Kennzeichnung „AT%I\*“ bzw. „AT%Q\*“) nicht zugeordnet sind. Das TwinCAT PLC Control erzeugt nach erfolgreichem Kompilervorgang eine „\*.tpy“ Datei in dem Verzeichnis in dem das Projekt gespeichert wurde. Diese Datei („\*.tpy“) enthält u.a. Variablenzuordnungen und ist dem System Manager nicht bekannt, was zu dieser Warnung führt. Nach dessen Bekanntgabe kommt es nicht mehr zu dieser Warnung.

**Im System Manager** ist das Projekt des TwinCAT PLC Control zunächst einzubinden. Dies geschieht über das Kontext Menü der „SPS- Konfiguration“ (rechts-Klick) und der Auswahl „SPS Projekt Anfügen...“:

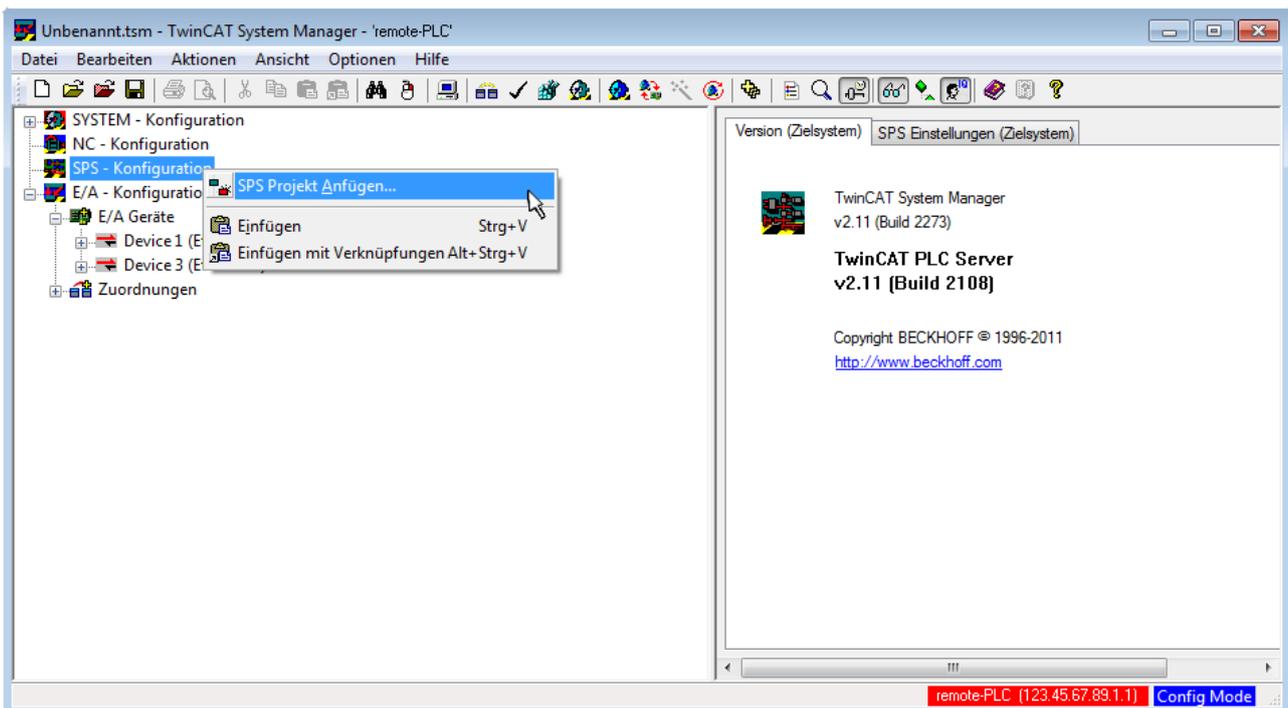


Abb. 53: Hinzufügen des Projektes des TwinCAT PLC Control

Über ein dadurch geöffnetes Browserfenster wird die PLC- Konfiguration „PLC\_example.tpy“ ausgewählt. Dann ist in dem Konfigurationsbaum des System Manager das Projekt inklusive der beiden „AT“ – gekennzeichneten Variablen eingebunden:

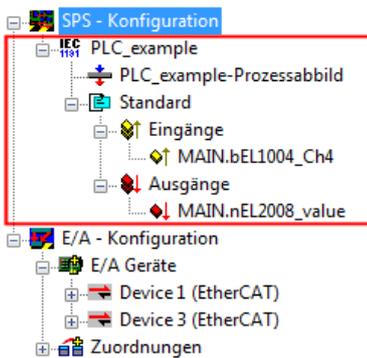


Abb. 54: Eingebundenes PLC Projekt in der SPS- Konfiguration des System Managers

Die beiden Variablen „bEL1004\_Ch4“ sowie „nEL2008\_value“ können nun bestimmten Prozessobjekten der E/A - Konfiguration zugeordnet werden.

### Variablen Zuordnen

Über das Kontextmenü einer Variable des eingebundenen Projekts „PLC\_example“ unter „Standard“ wird mittels „Verknüpfung Ändern...“ ein Fenster zur Auswahl eines passenden Prozessobjektes (PDOs) geöffnet:

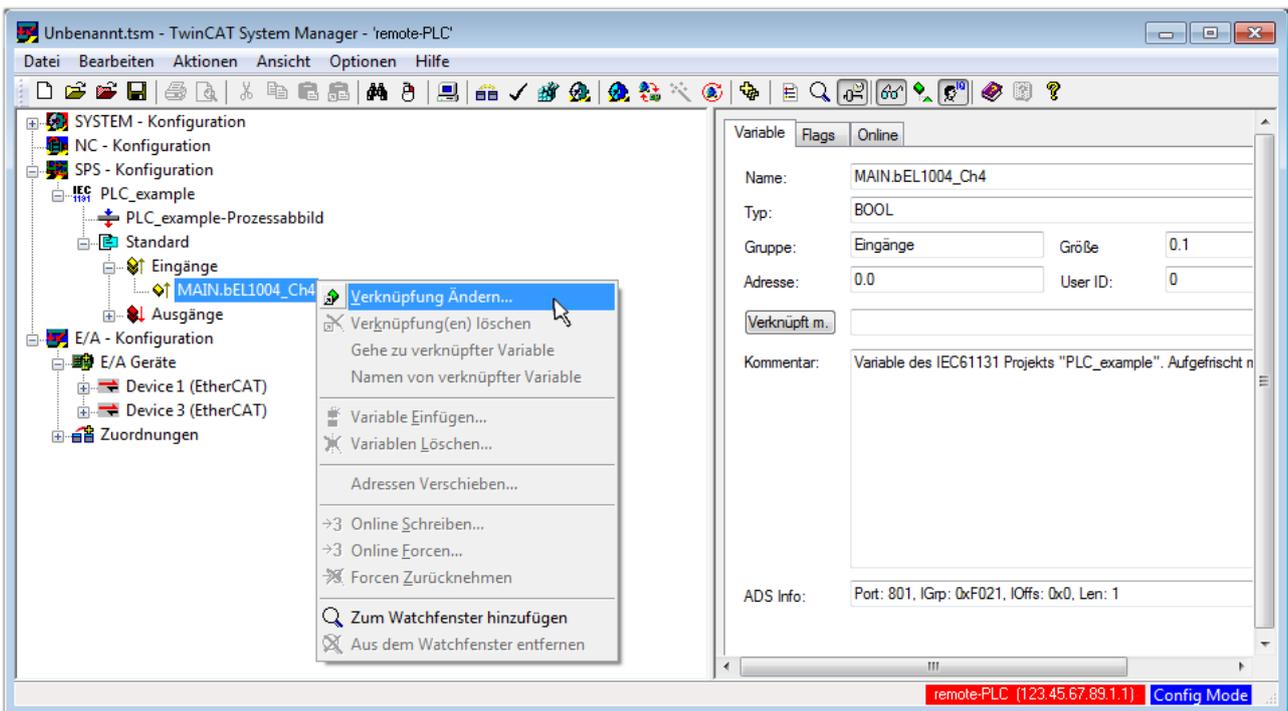


Abb. 55: Erstellen der Verknüpfungen PLC-Variablen zu Prozessobjekten

In dem dadurch geöffneten Fenster kann aus dem SPS-Konfigurationsbaum das Prozessobjekt für die Variable „bEL1004\_Ch4“ vom Typ BOOL selektiert werden:

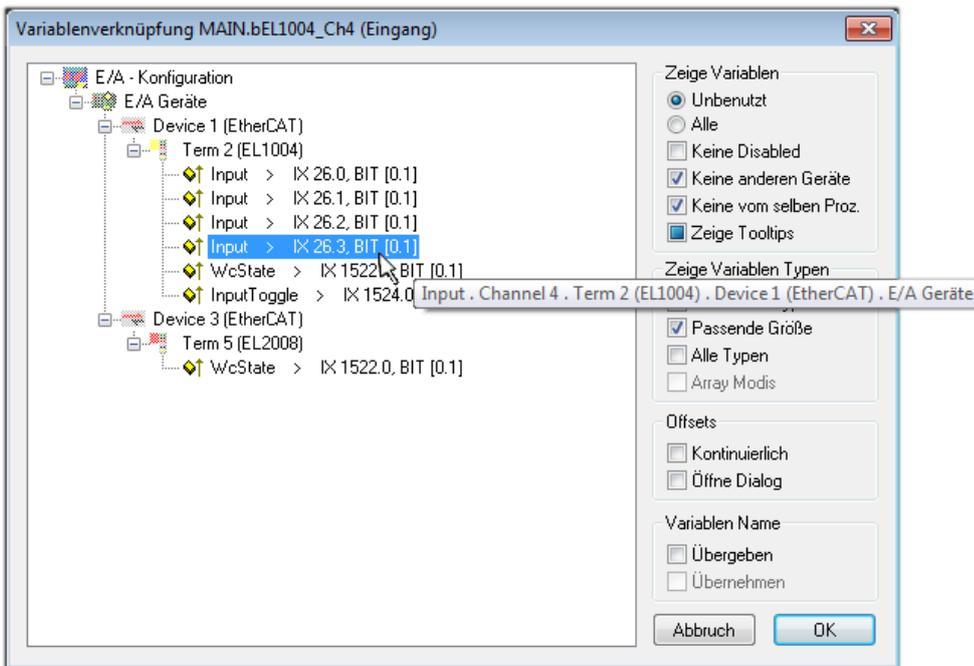


Abb. 56: Auswahl des PDO vom Typ BOOL

Entsprechend der Standarteinstellungen stehen nur bestimmte PDO Objekte zur Auswahl zur Verfügung. In diesem Beispiel wird von der Klemme EL1004 der Eingang von Kanal 4 zur Verknüpfung ausgewählt. Im Gegensatz hierzu muss für das Erstellen der Verknüpfung der Ausgangsvariablen die Checkbox „Alle Typen“ aktiviert werden, um in diesem Fall eine Byte-Variable einen Satz von acht separaten Ausgangsbits zuzuordnen. Die folgende Abbildung zeigt den gesamten Vorgang:

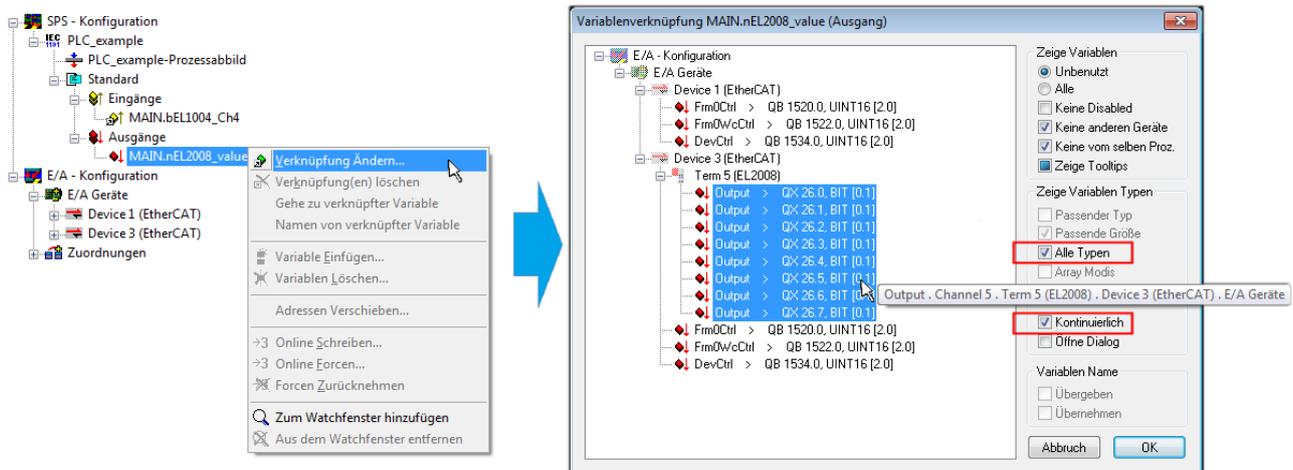


Abb. 57: Auswahl von mehreren PDO gleichzeitig: Aktivierung von „Kontinuierlich“ und „Alle Typen“

Zu sehen ist, dass überdies die Checkbox „Kontinuierlich“ aktiviert wurde. Dies ist dafür vorgesehen, dass die in dem Byte der Variablen „nEL2008\_value“ enthaltenen Bits allen acht ausgewählten Ausgangsbits der Klemme EL2008 der Reihenfolge nach zugeordnet werden sollen. Damit ist es möglich, alle acht Ausgänge der Klemme mit einem Byte entsprechend Bit 0 für Kanal 1 bis Bit 7 für Kanal 8 von der PLC im Programm später anzusprechen. Ein spezielles Symbol (  ) an dem gelben bzw. roten Objekt der Variablen zeigt an, dass hierfür eine Verknüpfung existiert. Die Verknüpfungen können z. B. auch überprüft werden, indem „Goto Link Variable“ aus dem Kontextmenü einer Variable ausgewählt wird. Dann wird automatisch das gegenüberliegende verknüpfte Objekt, in diesem Fall das PDO selektiert:

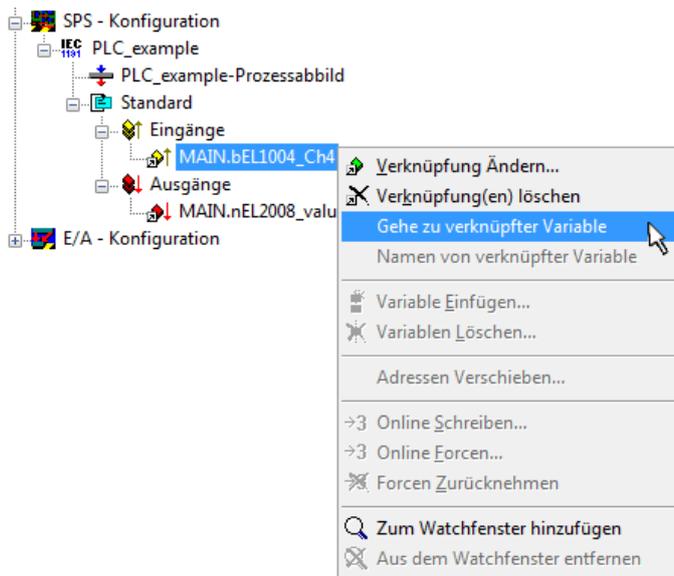


Abb. 58: Anwendung von „Goto Link Variable“ am Beispiel von „MAIN.bEL1004\_Ch4“

Anschließend wird mittels Menüauswahl „Aktionen“ → „Zuordnung erzeugen...“ oder über  der Vorgang des Zuordnens von Variablen zu PDO abgeschlossen.

Dies lässt sich entsprechend in der Konfiguration einsehen:



Der Vorgang zur Erstellung von Verknüpfungen kann auch in umgekehrter Richtung, d. h. von einzelnen PDO ausgehend zu einer Variablen erfolgen. In diesem Beispiel wäre dann allerdings eine komplette Auswahl aller Ausgangsbits der EL2008 nicht möglich, da die Klemme nur einzelne digitale Ausgänge zur Verfügung stellt. Hat eine Klemme einen Byte, Word, Integer oder ein ähnliches PDO, so ist es möglich dies wiederum einen Satz von bit-typisierten Variablen (Typ „BOOL“) zuzuordnen. Auch hier kann ebenso in die andere Richtung ein „Goto Link Variable“ ausgeführt werden, um dann die betreffende Instanz der PLC zu selektieren.

### Aktivieren der Konfiguration

Die Zuordnung von PDO zu PLC Variablen hat nun die Verbindung von der Steuerung zu den Ein- und

Ausgängen der Klemmen hergestellt. Nun kann die Konfiguration aktiviert werden. Zuvor kann mittels  (oder über „Aktionen“ → „Konfiguration überprüfen...“) die Konfiguration überprüft werden. Falls kein Fehler

vorliegt, kann mit  (oder über „Aktionen“ → „Aktiviert Konfiguration...“) die Konfiguration aktiviert werden, um dadurch Einstellungen im System Manager auf das Laufzeitsystem zu übertragen. Die darauf folgenden Meldungen „Alte Konfigurationen werden überschrieben!“ sowie „Neustart TwinCAT System in Run Modus“ werden jeweils mit „OK“ bestätigt.

Einige Sekunden später wird der Realtime Status **Echtzeit 0%** unten rechts im System Manager angezeigt. Das PLC System kann daraufhin wie im Folgenden beschrieben gestartet werden.

### Starten der Steuerung

Ausgehend von einem remote System muss nun als erstes auch die PLC Steuerung über „Online“ → „Choose Run-Time System...“ mit dem embedded PC über Ethernet verbunden werden:

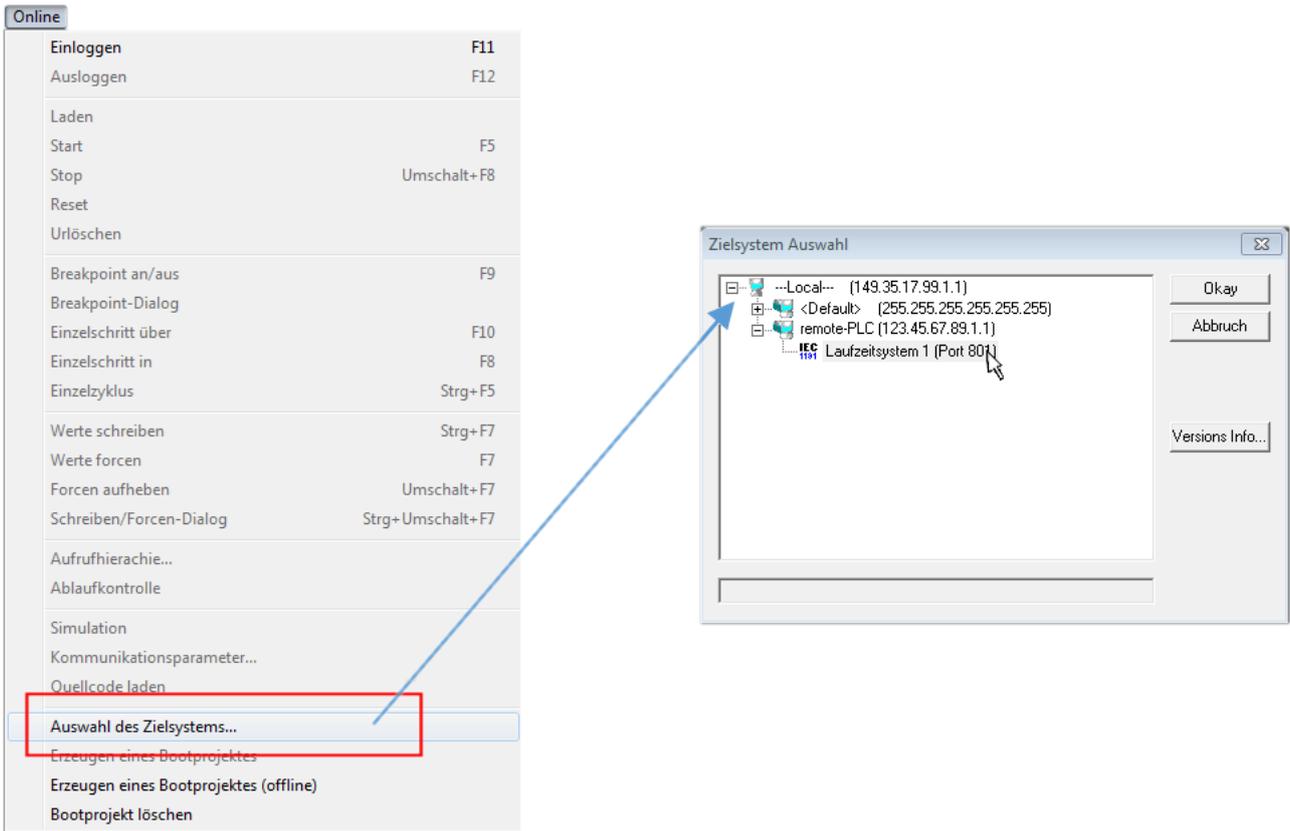


Abb. 59: Auswahl des Zielsystems (remote)

In diesem Beispiel wird das „Laufzeitsystem 1 (Port 801)“ ausgewählt und bestätigt. Mittels Menüauswahl

„Online“ → „Login“, Taste F11 oder per Klick auf  wird auch die PLC mit dem Echtzeitsystem verbunden und nachfolgend das Steuerprogramm geladen, um es ausführen lassen zu können. Dies wird entsprechend mit der Meldung „Kein Programm auf der Steuerung! Soll das neue Programm geladen werden?“ bekannt gemacht und ist mit „Ja“ zu beantworten. Die Laufzeitumgebung ist bereit zum Programmstart:

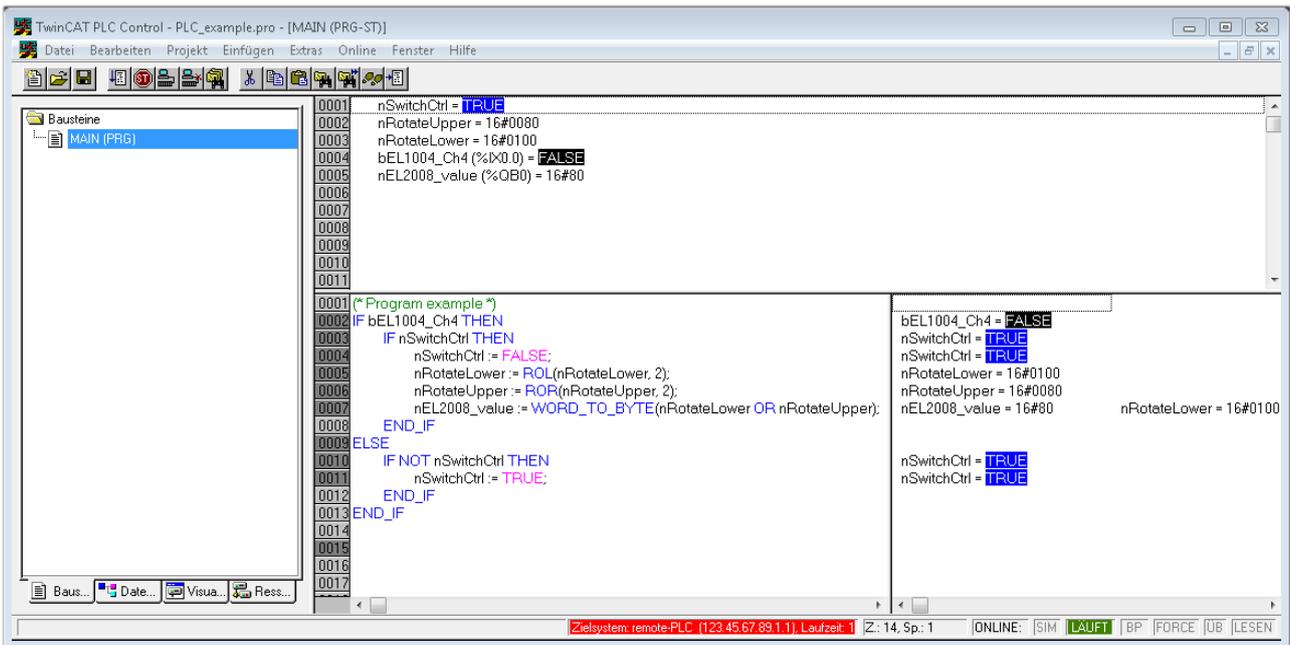


Abb. 60: PLC Control Logged-in, bereit zum Programmstart

Über „Online“ → „Run“, Taste F5 oder  kann nun die PLC gestartet werden.

## 5.1.2 TwinCAT 3

### Startup

TwinCAT 3 stellt die Bereiche der Entwicklungsumgebung durch das Microsoft Visual-Studio gemeinsam zur Verfügung: in den allgemeinen Fensterbereich erscheint nach dem Start linksseitig der Projektmappen-Explorer (vgl. „TwinCAT System Manager“ von TwinCAT 2) zur Kommunikation mit den elektromechanischen Komponenten.

Nach erfolgreicher Installation des TwinCAT-Systems auf den Anwender PC der zur Entwicklung verwendet werden soll, zeigt der TwinCAT 3 (Shell) folgende Benutzeroberfläche nach dem Start:

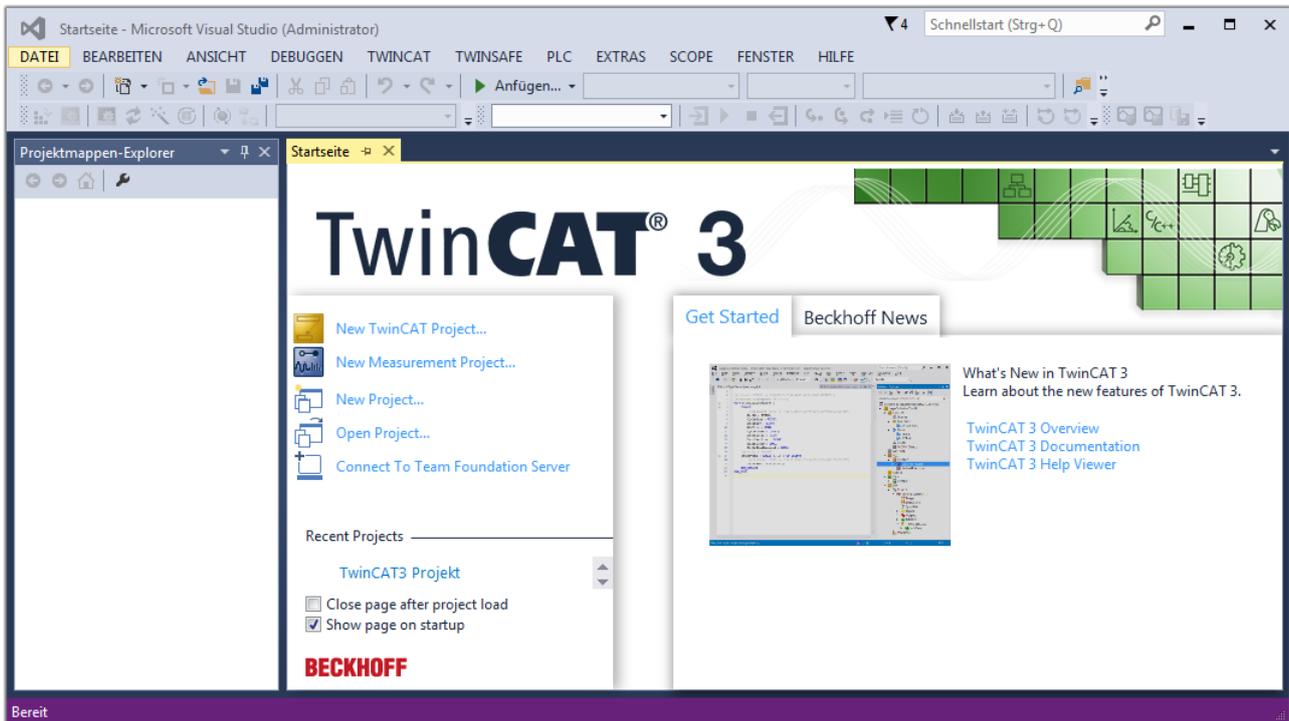


Abb. 61: Initiale Benutzeroberfläche TwinCAT 3

Zunächst ist die Erstellung eines neues Projekt mittels  **New TwinCAT Project...** (oder unter „Datei“→“Neu“→“Projekt...“) vorzunehmen. In dem darauf folgenden Dialog werden die entsprechenden Einträge vorgenommen (wie in der Abbildung gezeigt):

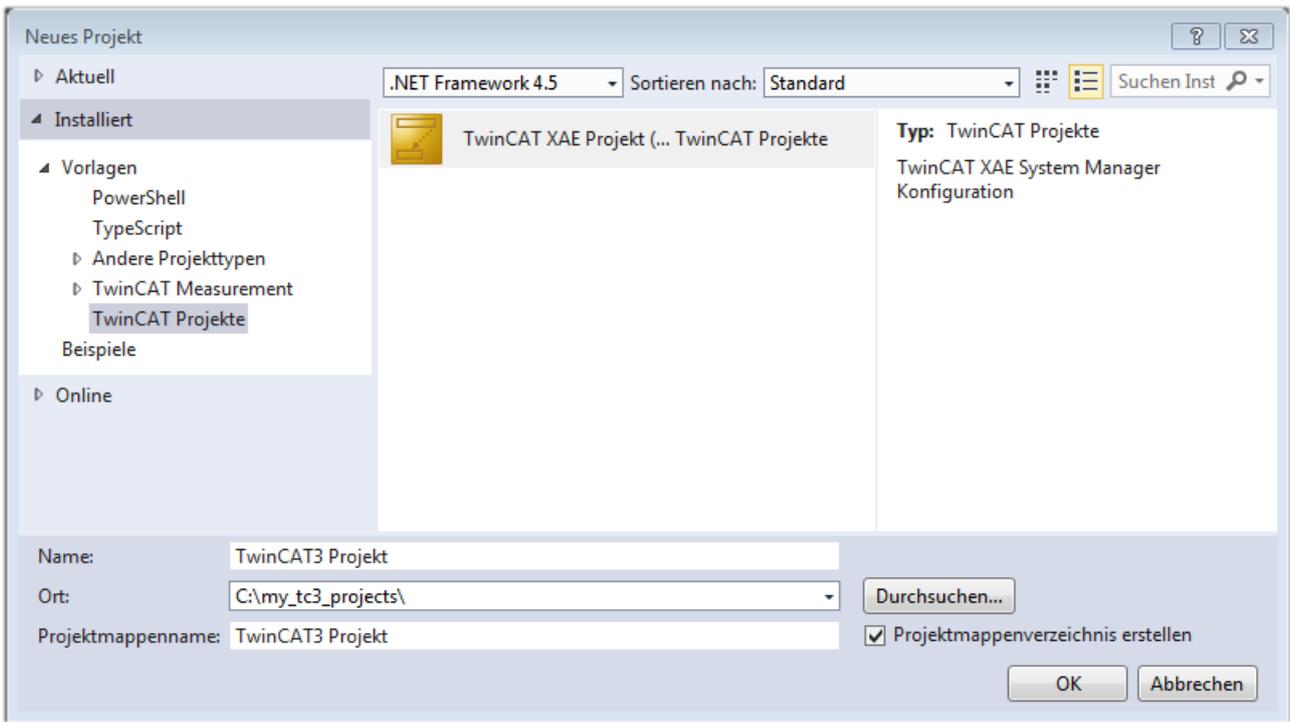


Abb. 62: Neues TwinCAT 3 Projekt erstellen

Im Projektmappen-Explorer liegt sodann das neue Projekt vor:

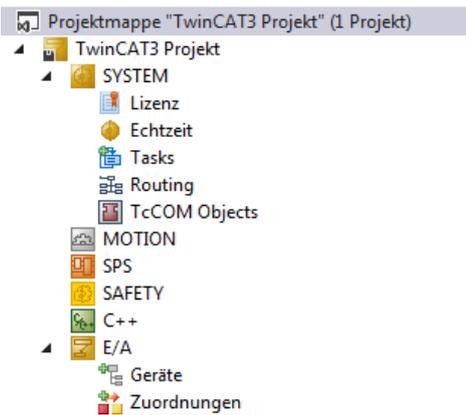
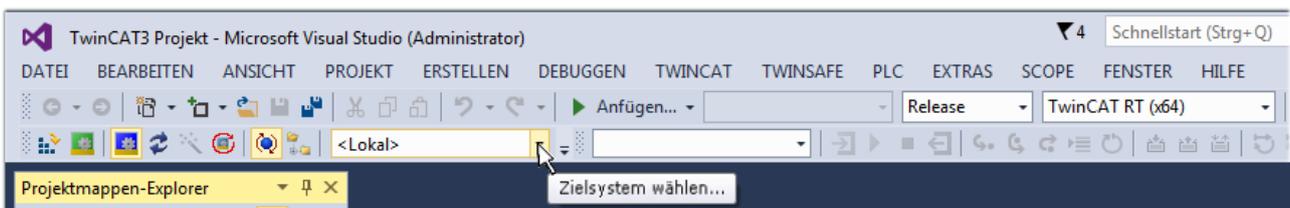


Abb. 63: Neues TwinCAT 3 Projekt im Projektmappen-Explorer

Es besteht generell die Möglichkeit das TwinCAT „lokal“ oder per „remote“ zu verwenden. Ist das TwinCAT System inkl. Benutzeroberfläche (Standard) auf dem betreffenden PLC (lokal) installiert, kann TwinCAT „lokal“ eingesetzt werden und mit Schritt „Geräte einfügen |> 76|“ fortgesetzt werden.

Ist es vorgesehen, die auf einem PLC installierte TwinCAT Laufzeitumgebung von einem anderen System als Entwicklungsumgebung per „remote“ anzusprechen, ist das Zielsystem zuvor bekannt zu machen. Über das Symbol in der Menüleiste:



wird das pull-down Menü aufgeklappt:



und folgendes Fenster hierzu geöffnet:

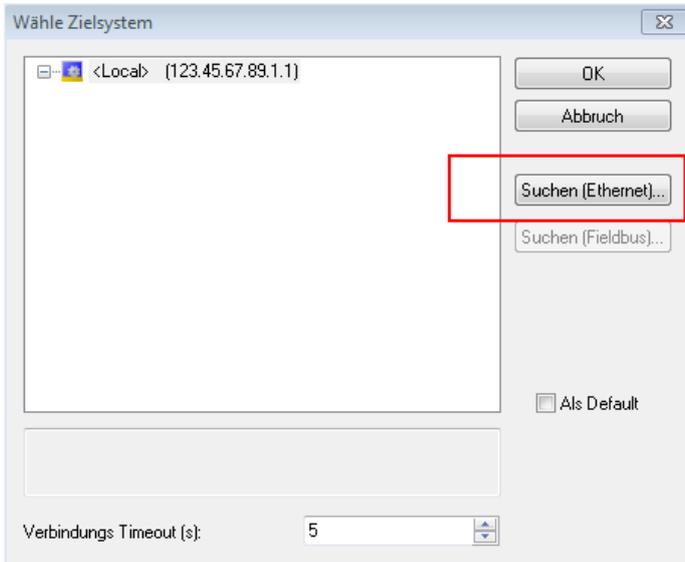


Abb. 64: Auswahldialog: Wähle Zielsystem

Mittels „Suchen (Ethernet)...“ wird das Zielsystem eingetragen. Dadurch wird ein weiterer Dialog geöffnet um hier entweder:

- den bekannten Rechnernamen hinter „Enter Host Name / IP:“ einzutragen (wie rot gekennzeichnet)
- einen „Broadcast Search“ durchzuführen (falls der Rechnernamen nicht genau bekannt)
- die bekannte Rechner - IP oder AmsNetId einzutragen

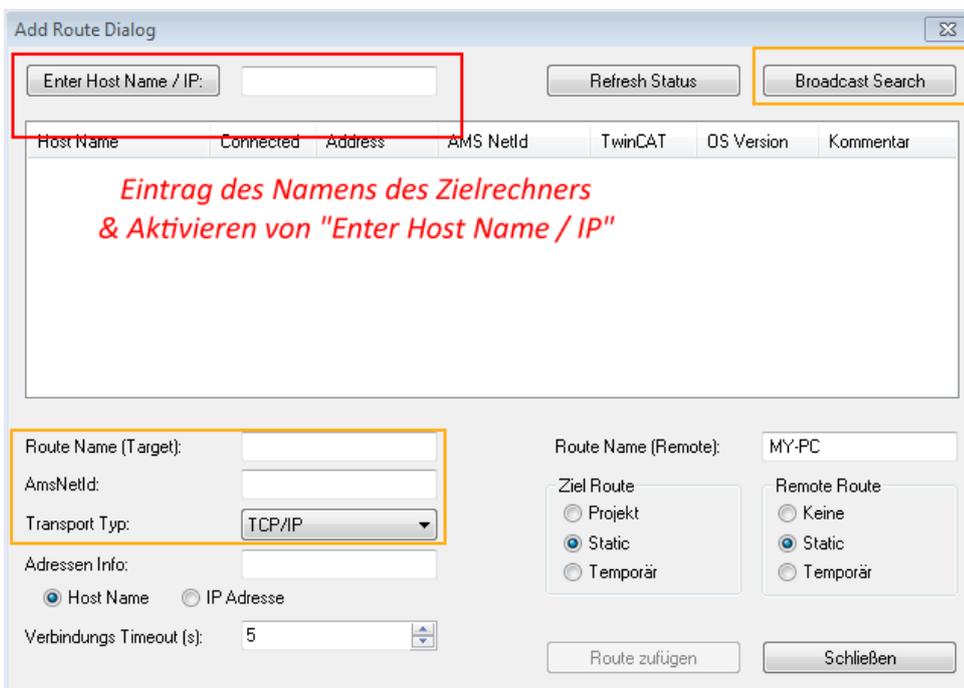
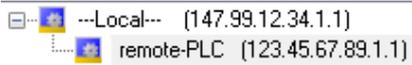


Abb. 65: PLC für den Zugriff des TwinCAT System Managers festlegen: Auswahl des Zielsystems

Ist das Zielsystem eingetragen, steht dieses wie folgt zur Auswahl (ggf. muss zuvor das korrekte Passwort eingetragen werden):



Nach der Auswahl mit „OK“ ist das Zielsystem über das Visual Studio Shell ansprechbar.

## Geräte einfügen

In dem linksseitigen Projektmappen-Explorer der Benutzeroberfläche des Visual Studio Shell wird innerhalb des Elementes „E/A“ befindliche „Geräte“ selektiert und sodann entweder über Rechtsklick ein Kontextmenü

geöffnet und „Scan“ ausgewählt oder in der Menüleiste mit  die Aktion gestartet. Ggf. ist zuvor der

TwinCAT System Manager in den „Konfig Modus“ mittels  oder über das Menü „TWINCAT“ → „Restart TwinCAT (Config Mode)“ zu versetzen.

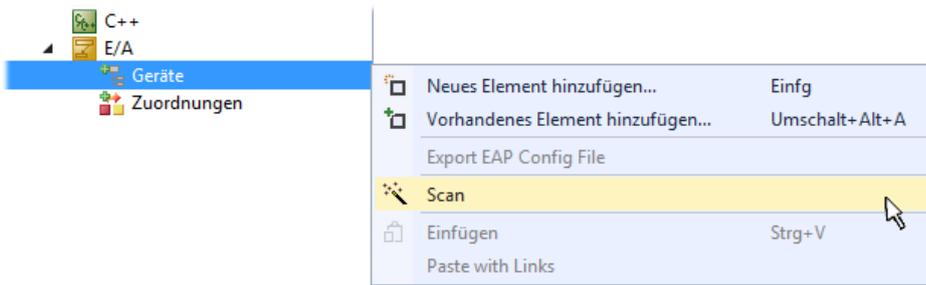


Abb. 66: Auswahl „Scan“

Die darauf folgende Hinweismeldung ist zu bestätigen und in dem Dialog die Geräte „EtherCAT“ zu wählen:

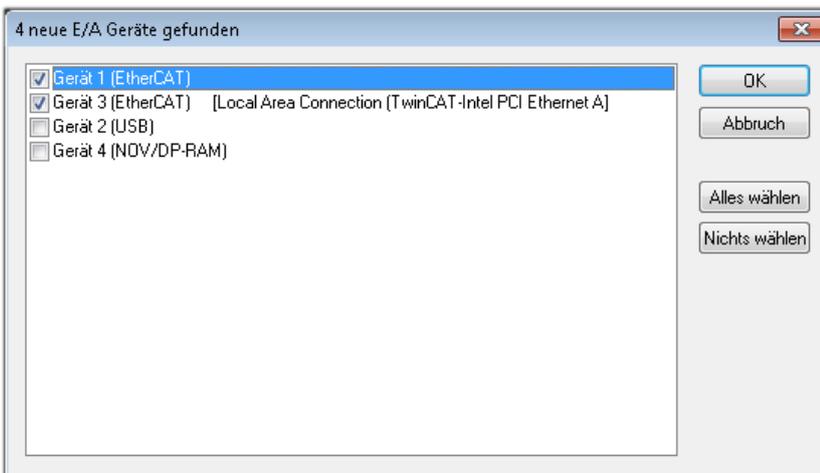


Abb. 67: Automatische Erkennung von E/A Geräten: Auswahl der einzubindenden Geräte

Ebenfalls ist anschließend die Meldung „nach neuen Boxen suchen“ zu bestätigen, um die an den Geräten angebotenen Klemmen zu ermitteln. „Free Run“ erlaubt das Manipulieren von Ein- und Ausgangswerten innerhalb des „Config Modus“ und sollte ebenfalls bestätigt werden.

Ausgehend von der am Anfang dieses Kapitels beschriebenen [Beispielkonfiguration](#) [► 61] sieht das Ergebnis wie folgt aus:

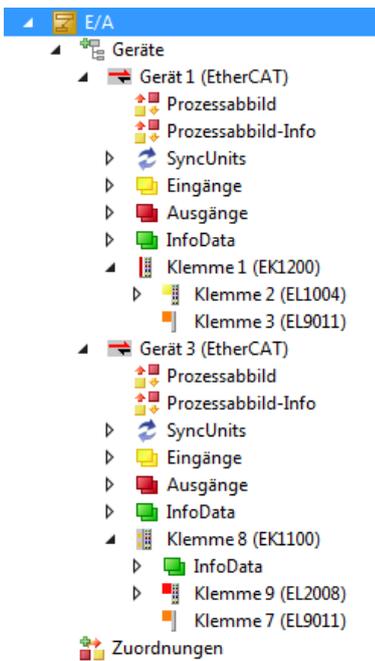


Abb. 68: Abbildung der Konfiguration in VS Shell der TwinCAT 3 Umgebung

Der gesamte Vorgang setzt sich aus zwei Stufen zusammen, die auch separat ausgeführt werden können (erst das Ermitteln der Geräte, dann das Ermitteln der daran befindlichen Elemente wie Boxen, Klemmen o. ä.). So kann auch durch Markierung von „Gerät ..“ aus dem Kontextmenü eine „Suche“ Funktion (Scan) ausgeführt werden, die hierbei dann lediglich die darunter liegenden (im Aufbau vorliegenden) Elemente einliest:

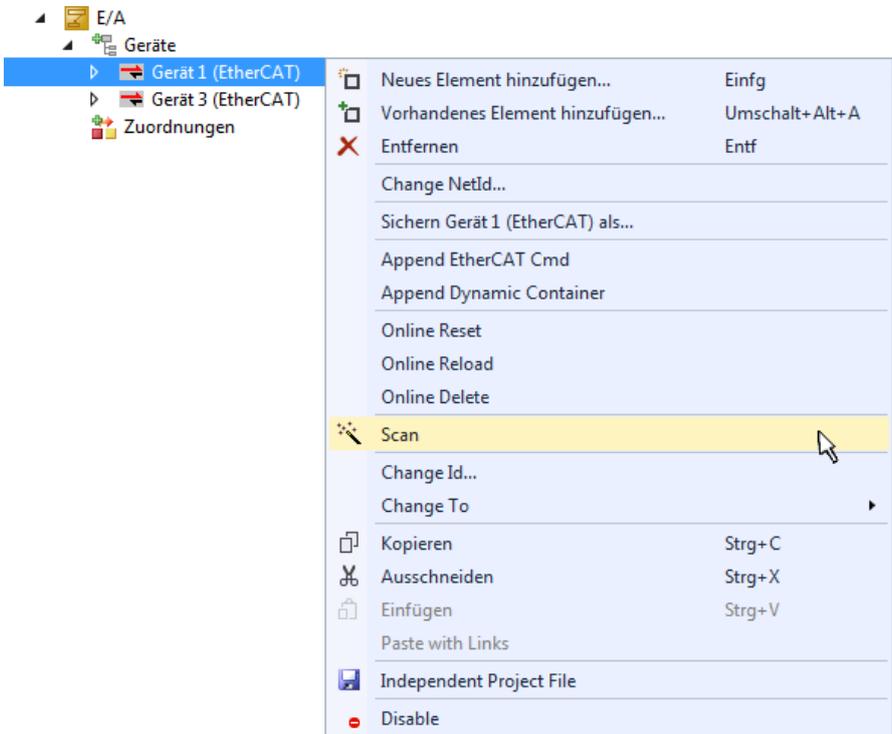


Abb. 69: Einlesen von einzelnen an einem Gerät befindlichen Klemmen

Diese Funktionalität ist nützlich, falls die Konfiguration (d. h. der „reale Aufbau“) kurzfristig geändert wird.

## PLC programmieren

TwinCAT PLC Control ist die Entwicklungsumgebung zur Erstellung der Steuerung in unterschiedlichen Programmumgebungen: Das TwinCAT PLC Control unterstützt alle in der IEC 61131-3 beschriebenen Sprachen. Es gibt zwei textuelle Sprachen und drei grafische Sprachen.

- **Textuelle Sprachen**
  - Anweisungsliste (AWL, IL)
  - Strukturierter Text (ST)
- **Grafische Sprachen**
  - Funktionsplan (FUP, FBD)
  - Kontaktplan (KOP, LD)
  - Freigrafischer Funktionsplaneditor (CFC)
  - Ablaufsprache (AS, SFC)

Für die folgenden Betrachtungen wird lediglich vom strukturierten Text (ST) Gebrauch gemacht.

Um eine Programmierumgebung zu schaffen, wird dem Beispielprojekt über das Kontextmenü von „SPS“ im Projektmappen-Explorer durch Auswahl von „Neues Element hinzufügen...“ ein PLC Unterprojekt hinzugefügt:

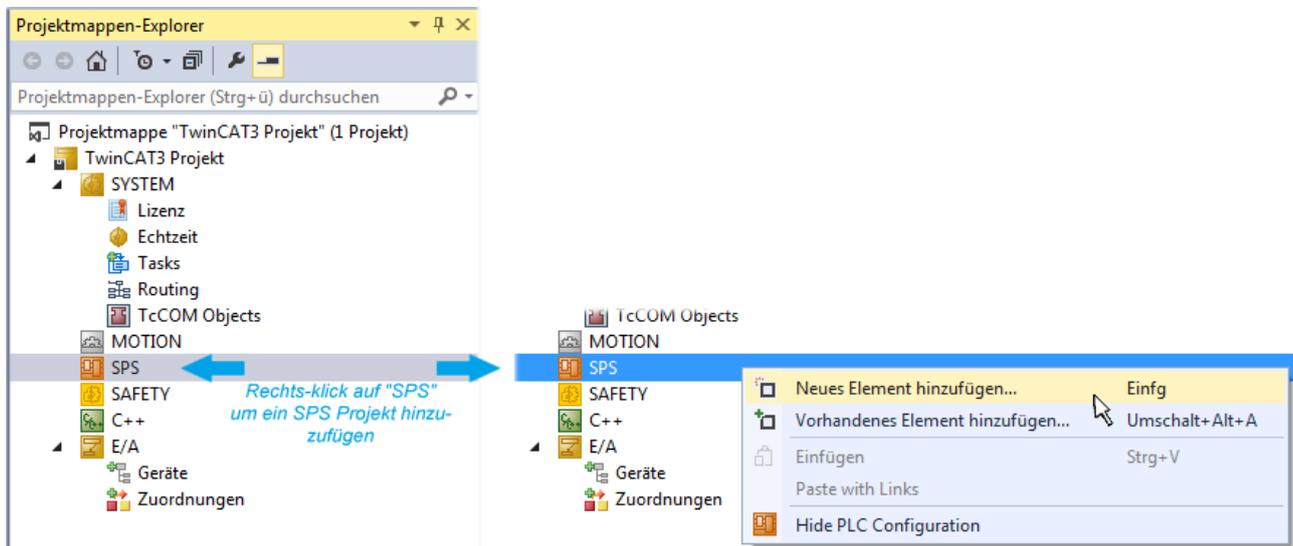


Abb. 70: Einfügen der Programmierumgebung in „SPS“

In dem darauf folgenden geöffneten Dialog wird ein „Standard PLC Projekt“ ausgewählt und beispielsweise als Projektname „PLC\_example“ vergeben und ein entsprechendes Verzeichnis ausgewählt:

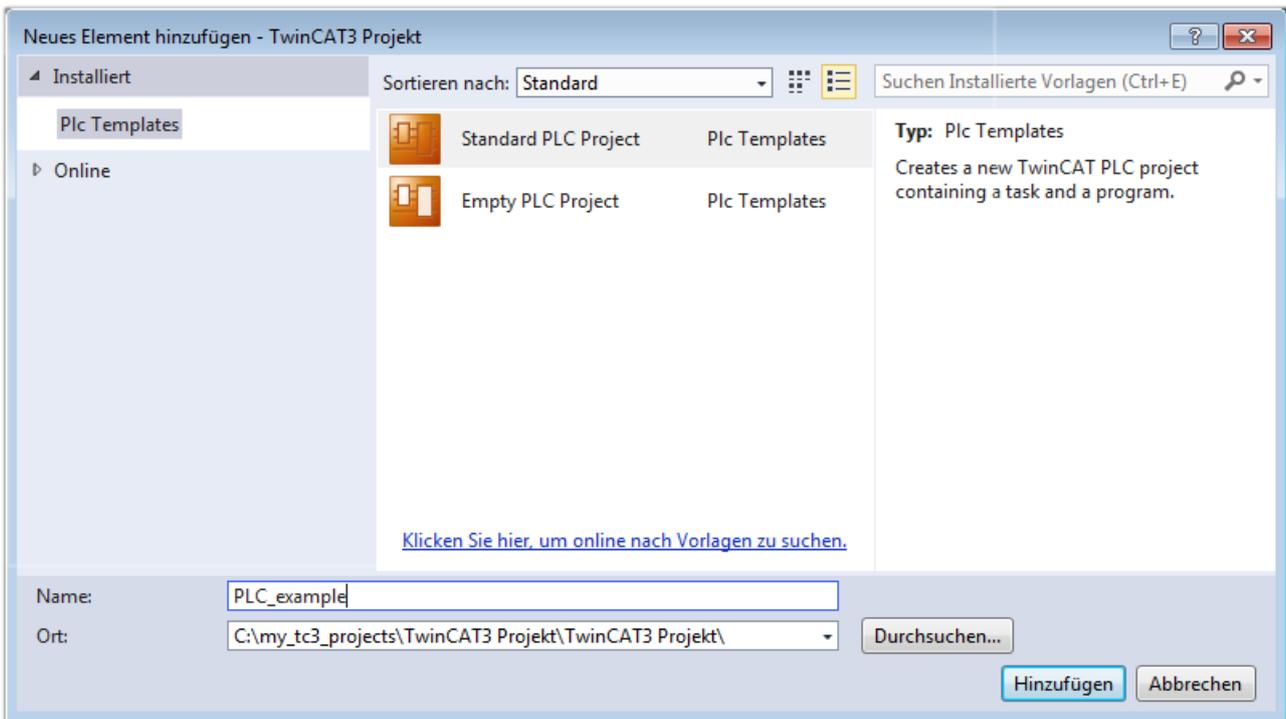


Abb. 71: Festlegen des Namens bzw. Verzeichnisses für die PLC Programmierungsumgebung

Das durch Auswahl von „Standard PLC Projekt“ bereits existierende Programm „Main“ kann über das „PLC\_example\_Project“ in „POUs“ durch Doppelklick geöffnet werden. Es wird folgende Benutzeroberfläche für ein initiales Projekt dargestellt:

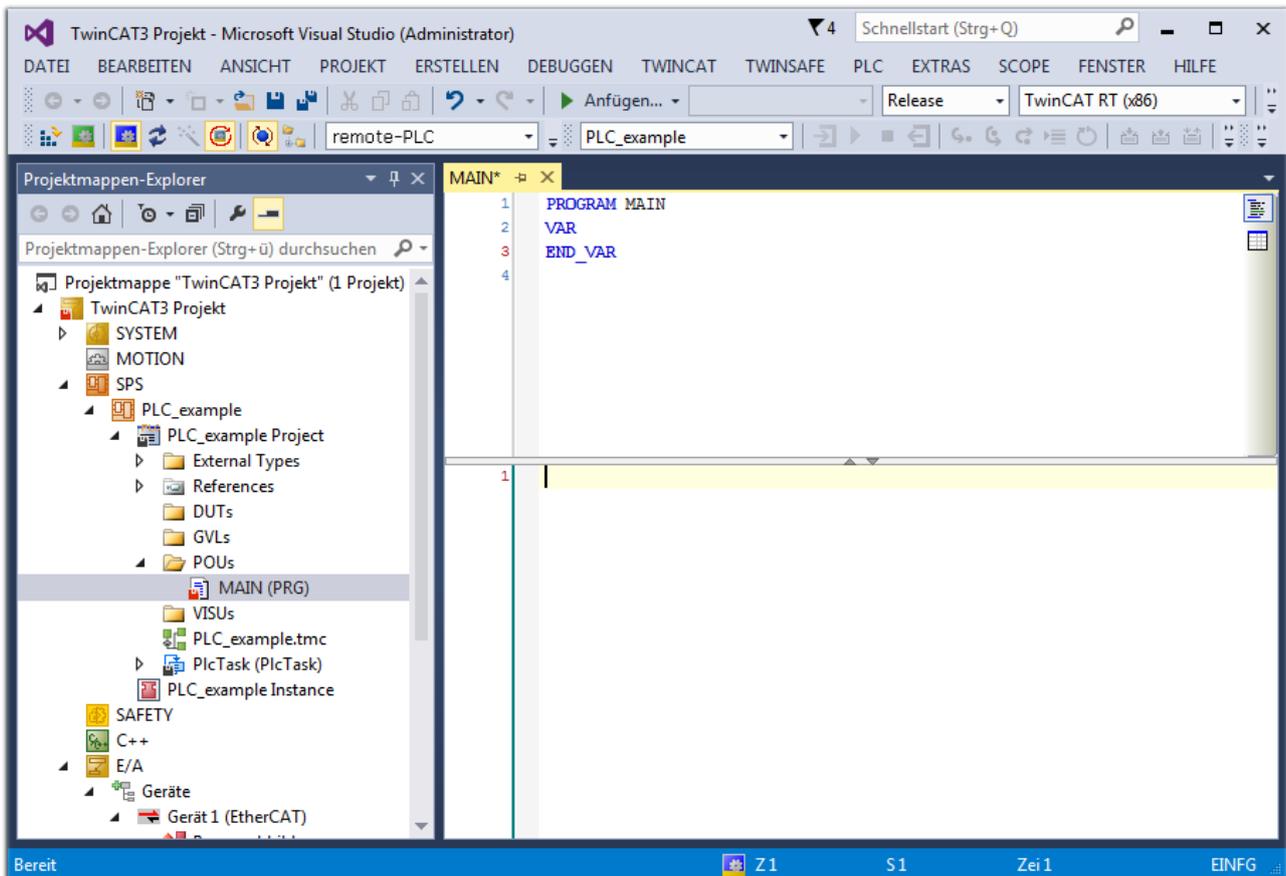


Abb. 72: Initiales Programm „Main“ des Standard PLC Projektes

Nun sind für den weiteren Ablauf Beispielvariablen sowie ein Beispielprogramm erstellt worden:

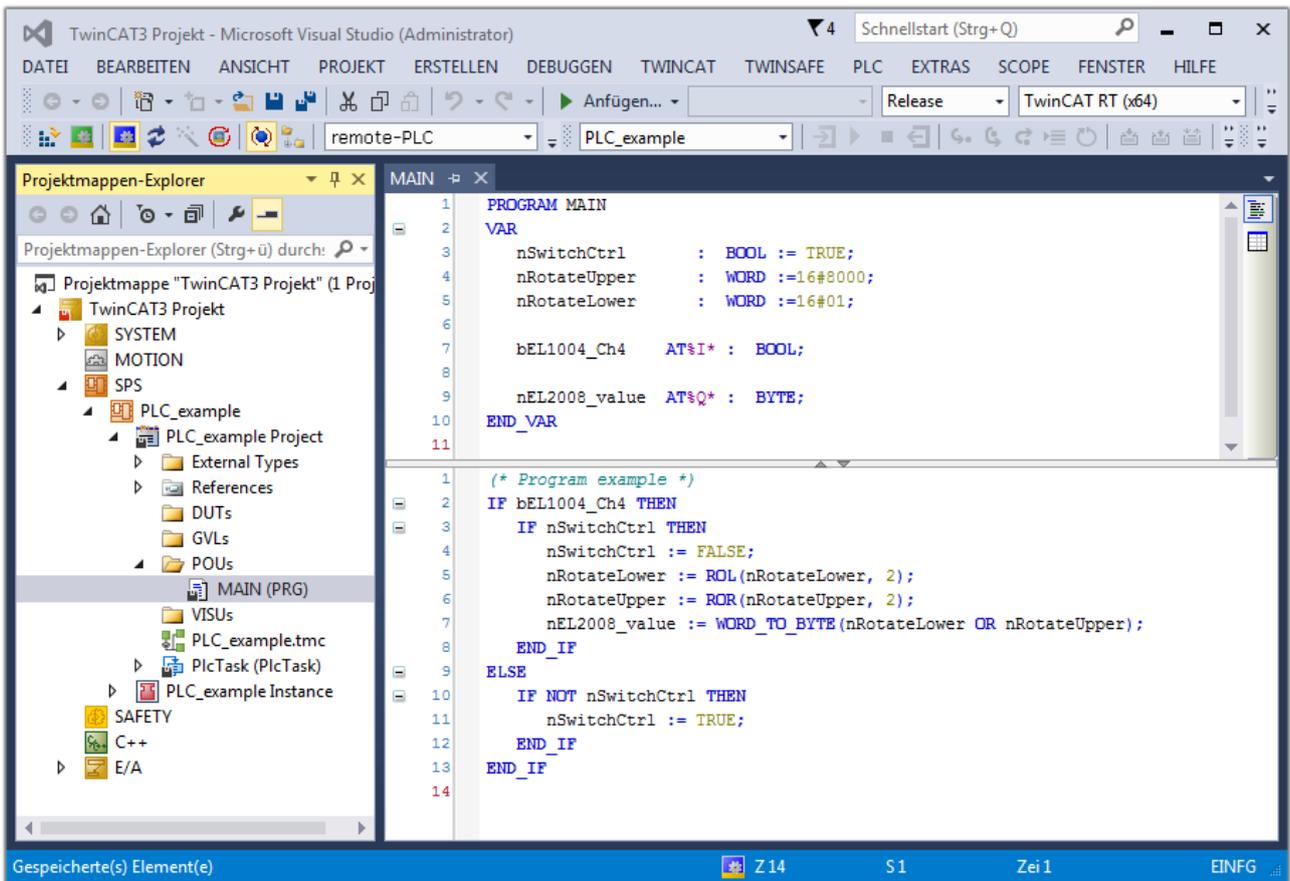


Abb. 73: Beispielprogramm mit Variablen nach einem Kompilervorgang (ohne Variablenanbindung)

Das Steuerprogramm wird nun als Projektmappe erstellt und damit der Kompilervorgang vorgenommen:

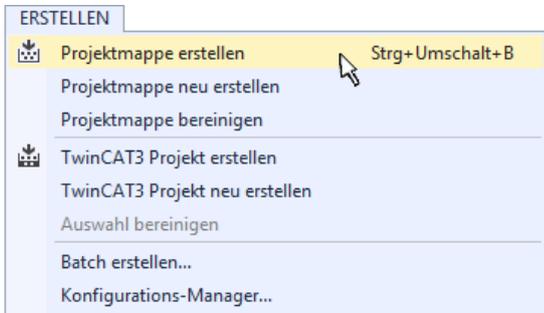
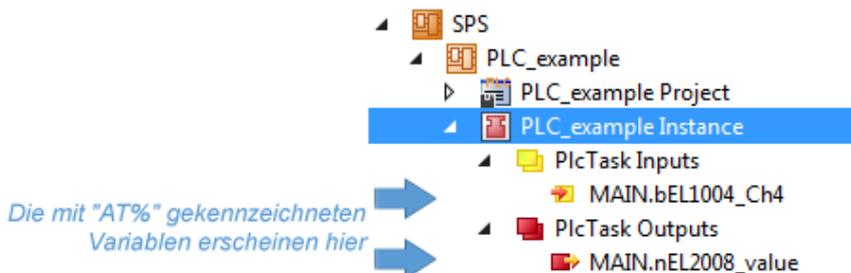


Abb. 74: Kompilierung des Programms starten

Anschließend liegen in den „Zuordnungen“ des Projektmappen-Explorers die folgenden – im ST/ PLC Programm mit „AT%“ gekennzeichneten Variablen vor:



**Variablen Zuordnen**

Über das Menü einer Instanz – Variablen innerhalb des „SPS“ Kontextes wird mittels „Verknüpfung Ändern...“ ein Fenster zur Auswahl eines passenden Prozessobjektes (PDOs) für dessen Verknüpfung geöffnet:

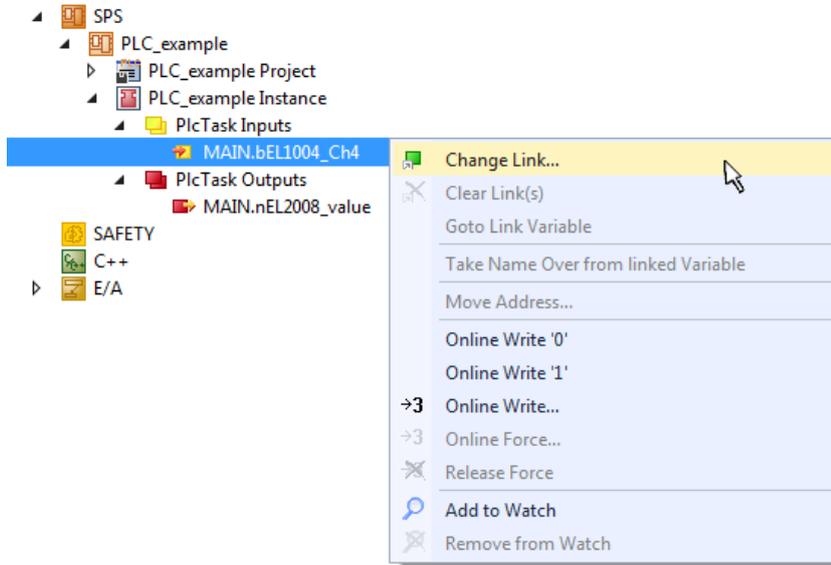


Abb. 75: Erstellen der Verknüpfungen PLC-Variablen zu Prozessobjekten

In dem dadurch geöffneten Fenster kann aus dem SPS-Konfigurationsbaum das Prozessobjekt für die Variable „bEL1004\_Ch4“ vom Typ BOOL selektiert werden:

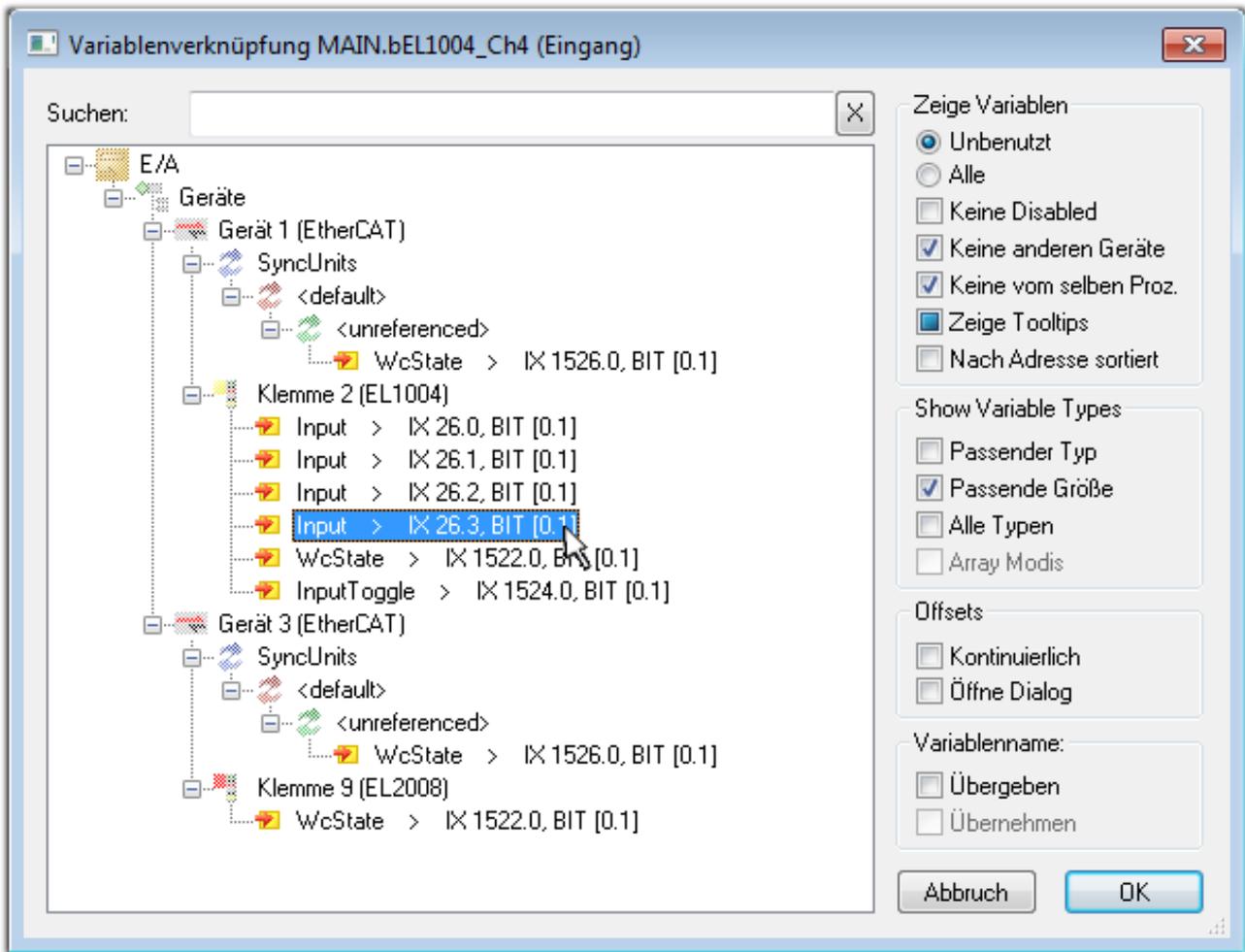


Abb. 76: Auswahl des PDO vom Typ BOOL

Entsprechend der Standarteinstellungen stehen nur bestimmte PDO Objekte zur Auswahl zur Verfügung. In diesem Beispiel wird von der Klemme EL1004 der Eingang von Kanal 4 zur Verknüpfung ausgewählt. Im Gegensatz hierzu muss für das Erstellen der Verknüpfung der Ausgangsvariablen die Checkbox „Alle Typen“ aktiviert werden, um in diesem Fall eine Byte-Variable einen Satz von acht separaten Ausgangsbits zuzuordnen. Die folgende Abbildung zeigt den gesamten Vorgang:

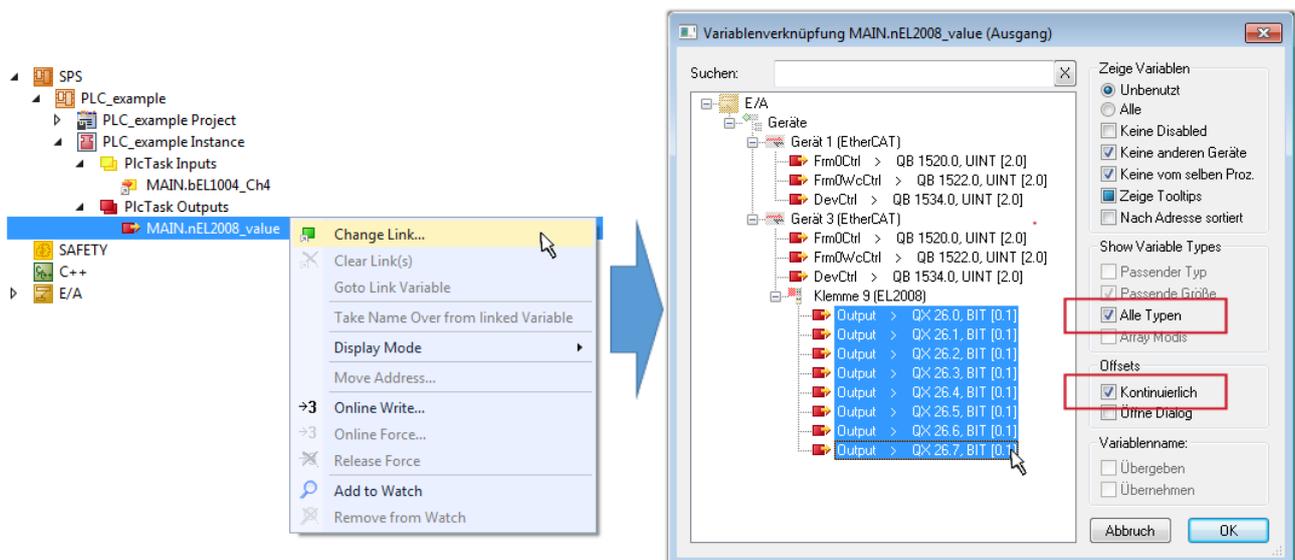


Abb. 77: Auswahl von mehreren PDO gleichzeitig: Aktivierung von „Kontinuierlich“ und „Alle Typen“

Zu sehen ist, dass überdies die Checkbox „Kontinuierlich“ aktiviert wurde. Dies ist dafür vorgesehen, dass die in dem Byte der Variablen „nEL2008\_value“ enthaltenen Bits allen acht ausgewählten Ausgangsbits der Klemme EL2008 der Reihenfolge nach zugeordnet werden sollen. Damit ist es möglich, alle acht Ausgänge der Klemme mit einem Byte entsprechend Bit 0 für Kanal 1 bis Bit 7 für Kanal 8 von der PLC im Programm später anzusprechen. Ein spezielles Symbol (  ) an dem gelben bzw. roten Objekt der Variablen zeigt an, dass hierfür eine Verknüpfung existiert. Die Verknüpfungen können z. B. auch überprüft werden, indem „Goto Link Variable“ aus dem Kontextmenü einer Variable ausgewählt wird. Dann wird automatisch das gegenüberliegende verknüpfte Objekt, in diesem Fall das PDO selektiert:

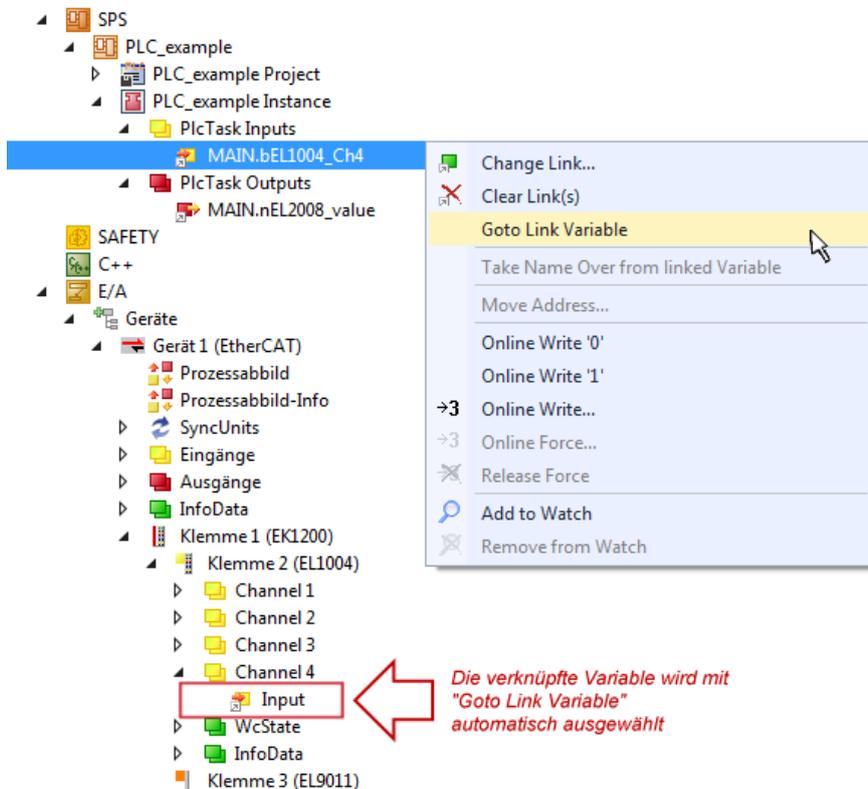


Abb. 78: Anwendung von "Goto Link Variable" am Beispiel von „MAIN.bEL1004\_Ch4“

Der Vorgang zur Erstellung von Verknüpfungen kann auch in umgekehrter Richtung, d. h. von einzelnen PDO ausgehend zu einer Variablen erfolgen. In diesem Beispiel wäre dann allerdings eine komplette Auswahl aller Ausgangsbits der EL2008 nicht möglich, da die Klemme nur einzelne digitale Ausgänge zur Verfügung stellt. Hat eine Klemme einen Byte, Word, Integer oder ein ähnliches PDO, so ist es möglich dies wiederum einen Satz von bit-typisierten Variablen (Typ „BOOL“) zuzuordnen. Auch hier kann ebenso in die andere Richtung ein „Goto Link Variable“ ausgeführt werden, um dann die betreffende Instanz der PLC zu selektieren.

**● Hinweis zur Art der Variablen-Zuordnung**

**i** Diese folgende Art der Variablen Zuordnung kann erst ab der TwinCAT Version V3.1.4024.4 verwendet werden und ist ausschließlich bei Klemmen mit einem Mikrocontroller verfügbar.

In TwinCAT ist es möglich eine Struktur aus den gemappten Prozessdaten einer Klemme zu erzeugen. Von dieser Struktur kann dann in der SPS eine Instanz angelegt werden, so dass aus der SPS direkt auf die Prozessdaten zugegriffen werden kann, ohne eigene Variablen deklarieren zu müssen.

Beispielhaft wird das Vorgehen an der EL3001 1-Kanal-Analog-Eingangsklemme -10...+10 V gezeigt.

1. Zuerst müssen die benötigten Prozessdaten im Reiter „Prozessdaten“ in TwinCAT ausgewählt werden.
2. Anschließend muss der SPS Datentyp im Reiter „PLC“ über die Check-Box generiert werden.
3. Der Datentyp im Feld „Data Type“ kann dann über den „Copy“-Button kopiert werden.

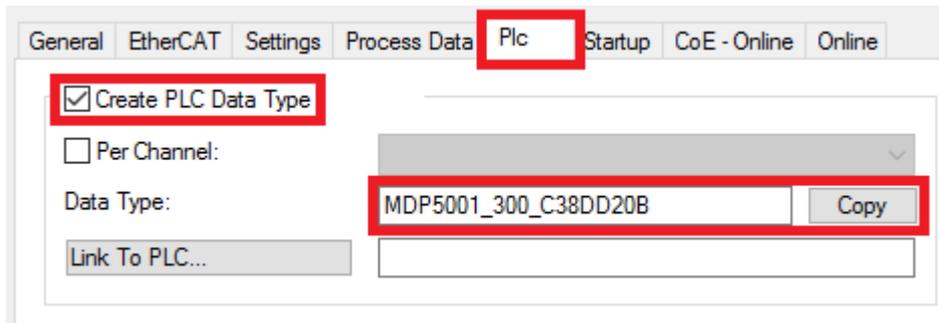


Abb. 79: Erzeugen eines SPS Datentyps

4. In der SPS muss dann eine Instanz der Datenstruktur vom kopierten Datentyp angelegt werden.

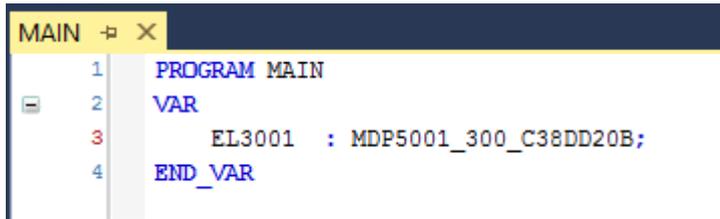


Abb. 80: Instance\_of\_struct

5. Anschließend muss die Projektmappe erstellt werden. Das kann entweder über die Tastenkombination „STRG + Shift + B“ gemacht werden oder über den Reiter „Erstellen“/ „Build“ in TwinCAT.

6. Die Struktur im Reiter „PLC“ der Klemme muss dann mit der angelegten Instanz verknüpft werden.

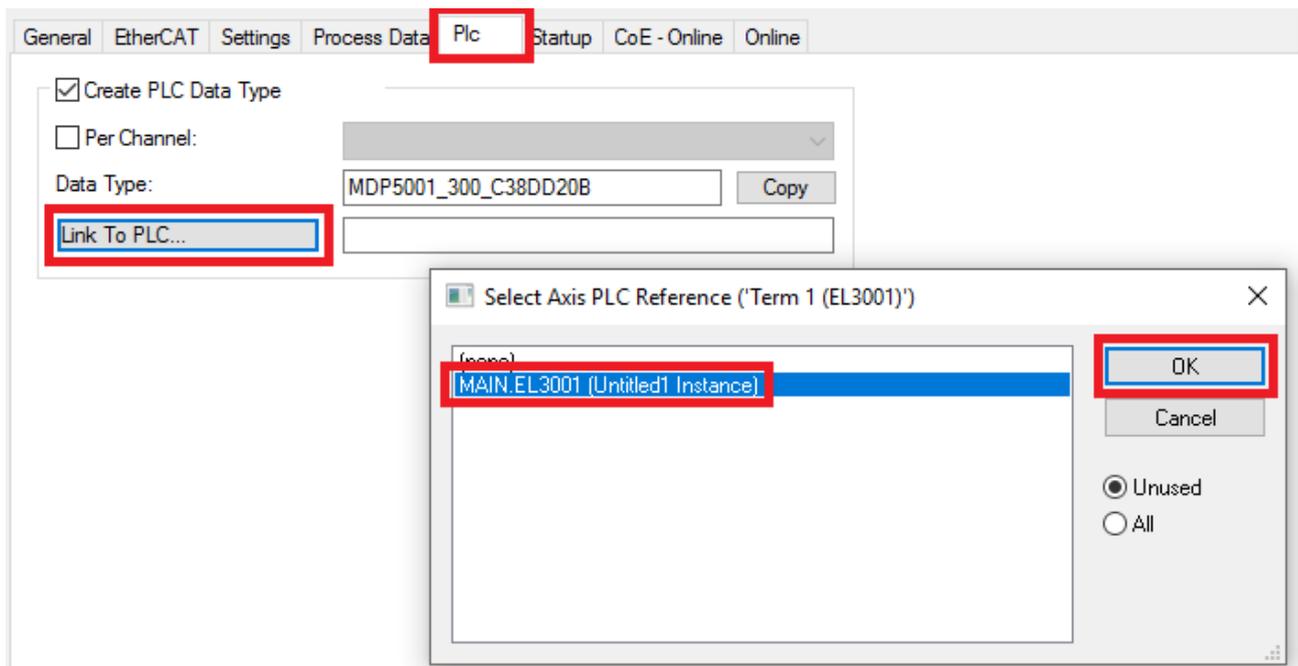


Abb. 81: Verknüpfung der Struktur

7. In der SPS können die Prozessdaten dann über die Struktur im Programmcode gelesen bzw. geschrieben werden.

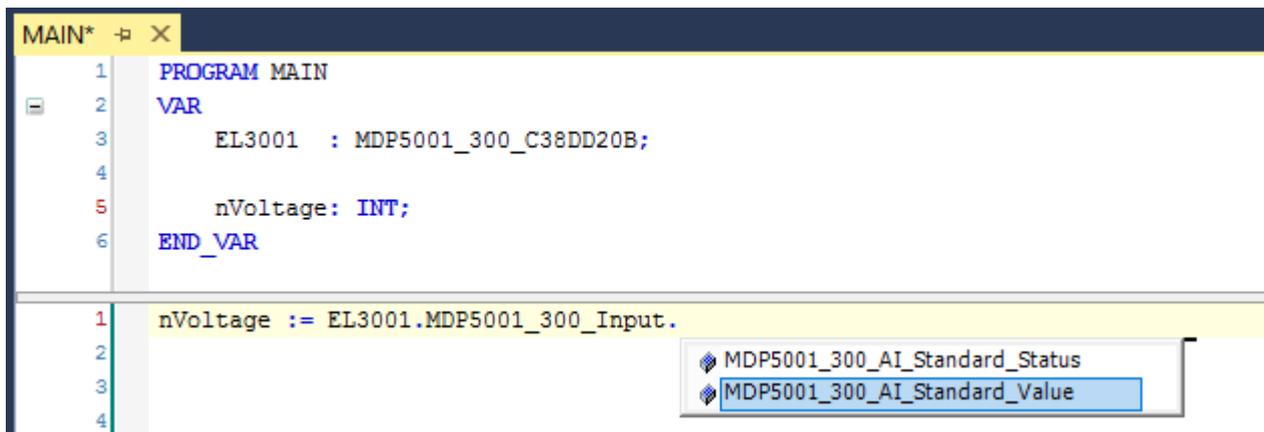


Abb. 82: Lesen einer Variable aus der Struktur der Prozessdaten

### Aktivieren der Konfiguration

Die Zuordnung von PDO zu PLC Variablen hat nun die Verbindung von der Steuerung zu den Ein- und

Ausgängen der Klemmen hergestellt. Nun kann die Konfiguration mit  oder über das Menü unter „TWINCAT“ aktiviert werden, um dadurch Einstellungen der Entwicklungsumgebung auf das Laufzeitsystem zu übertragen. Die darauf folgenden Meldungen „Alte Konfigurationen werden überschrieben!“ sowie „Neustart TwinCAT System in Run Modus“ werden jeweils mit „OK“ bestätigt. Die entsprechenden Zuordnungen sind in dem Projektmappen-Explorer einsehbar:

- ▲  Zuordnungen
  -  PLC\_example Instance - Gerät 3 (EtherCAT) 1
  -  PLC\_example Instance - Gerät 1 (EtherCAT) 1

Einige Sekunden später wird der entsprechende Status des Run Modus mit einem rotierenden Symbol  unten rechts in der Entwicklungsumgebung VS Shell angezeigt. Das PLC System kann daraufhin wie im Folgenden beschrieben gestartet werden.

### Starten der Steuerung

Entweder über die Menüauswahl „PLC“ → „Einloggen“ oder per Klick auf  ist die PLC mit dem Echtzeitsystem zu verbinden und nachfolgend das Steuerprogramm zu laden, um es ausführen lassen zu können. Dies wird entsprechend mit der Meldung „Kein Programm auf der Steuerung! Soll das neue Programm geladen werden?“ bekannt gemacht und ist mit „Ja“ zu beantworten. Die Laufzeitumgebung ist

bereit zum Programmstart mit Klick auf das Symbol , Taste „F5“ oder entsprechend auch über „PLC“ im Menü durch Auswahl von „Start“. Die gestartete Programmierumgebung zeigt sich mit einer Darstellung der Laufzeitwerte von einzelnen Variablen:

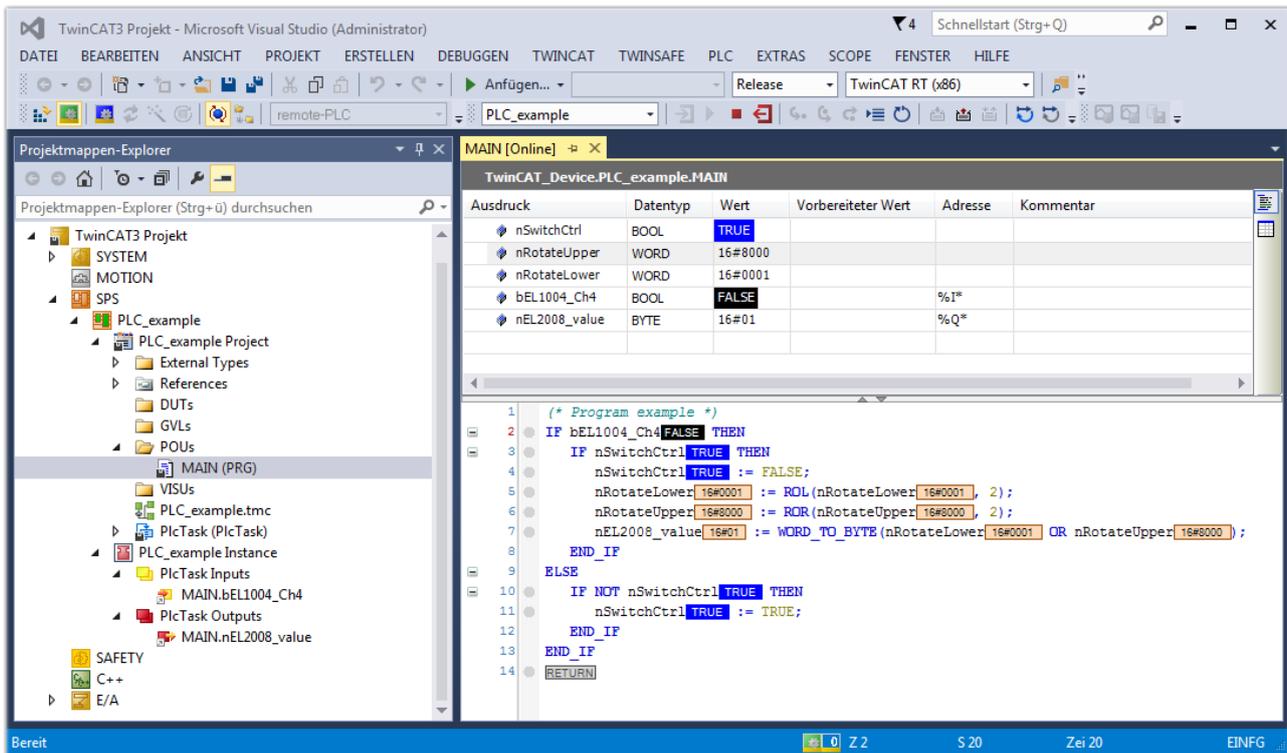


Abb. 83: TwinCAT 3 Entwicklungsumgebung (VS Shell): Logged-in, nach erfolgten Programmstart

Die beiden Bedienelemente zum Stoppen  und Ausloggen  führen je nach Bedarf zu der gewünschten Aktion (entsprechend auch für Stopp „umschalt-Taste + F5“ oder beide Aktionen über das „PLC“ Menü auswählbar).

## 5.2 TwinCAT Entwicklungsumgebung

Die Software zur Automatisierung TwinCAT (The Windows Control and Automation Technology) wird unterschieden in:

- TwinCAT 2: System Manager (Konfiguration) & PLC Control (Programmierung)
- TwinCAT 3: Weiterentwicklung von TwinCAT 2 (Programmierung und Konfiguration erfolgt über eine gemeinsame Entwicklungsumgebung)

### Details:

- **TwinCAT 2:**
  - Verbindet E/A-Geräte und Tasks variablenorientiert
  - Verbindet Tasks zu Tasks variablenorientiert
  - Unterstützt Einheiten auf Bit-Ebene
  - Unterstützt synchrone oder asynchrone Beziehungen
  - Austausch konsistenter Datenbereiche und Prozessabbilder
  - Datenanbindung an NT-Programme mittels offener Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.).
  - Einbettung von IEC 61131-3-Software-SPS, Software- NC und Software-CNC in Windows NT/2000/XP/Vista, Windows 7, NT/XP Embedded, CE
  - Anbindung an alle gängigen Feldbusse
  - Weiteres...

**Zusätzlich bietet:**

- **TwinCAT 3 (eXtended Automation):**
  - Visual-Studio®-Integration
  - Wahl der Programmiersprache
  - Unterstützung der objektorientierten Erweiterung der IEC 61131-3
  - Verwendung von C/C++ als Programmiersprache für Echtzeitanwendungen
  - Anbindung an MATLAB®/Simulink®
  - Offene Schnittstellen für Erweiterbarkeit
  - Flexible Laufzeitumgebung
  - Aktive Unterstützung von Multi-Core- und 64-Bit-Betriebssystemen
  - Automatische Codegenerierung und Projekterstellung mit dem TwinCAT Automation Interface
  - Weiteres...

In den folgenden Kapiteln wird dem Anwender die Inbetriebnahme der TwinCAT Entwicklungsumgebung auf einem PC System der Steuerung sowie die wichtigsten Funktionen einzelner Steuerungselemente erläutert.

Bitte sehen Sie weitere Informationen zu TwinCAT 2 und TwinCAT 3 unter <http://infosys.beckhoff.de/>.

### 5.2.1 Installation TwinCAT Realtime Treiber

Um einen Standard Ethernet Port einer IPC Steuerung mit den nötigen Echtzeitfähigkeiten auszurüsten, ist der Beckhoff Echtzeit Treiber auf diesem Port unter Windows zu installieren.

Dies kann auf mehreren Wegen vorgenommen werden, ein Weg wird hier vorgestellt.

Im System Manager ist über Options → Show realtime Kompatible Geräte die TwinCAT-Übersicht über die lokalen Netzwerkschnittstellen aufzurufen.



Abb. 84: Aufruf im System Manager (TwinCAT 2)

Unter TwinCAT 3 ist dies über das Menü unter „TwinCAT“ erreichbar:

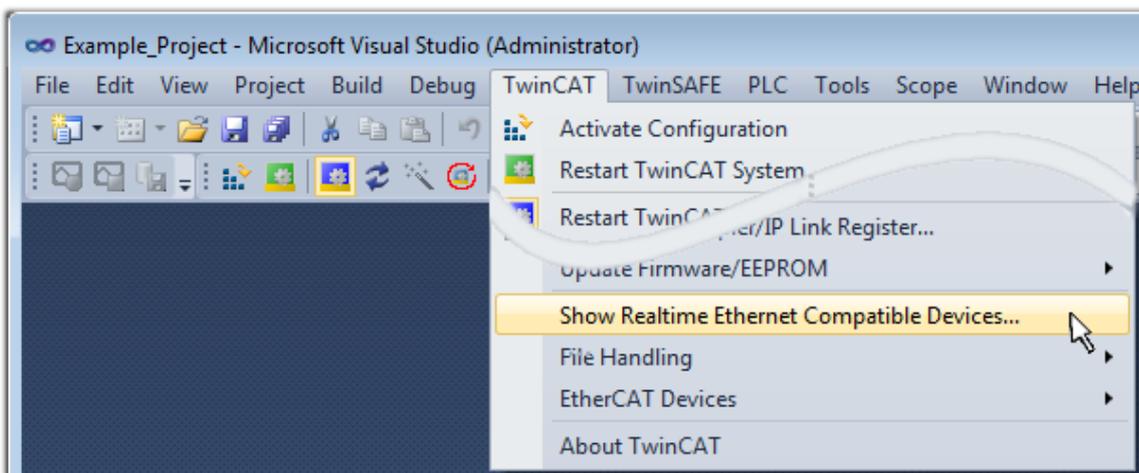


Abb. 85: Aufruf in VS Shell (TwinCAT 3)

Der folgende Dialog erscheint:

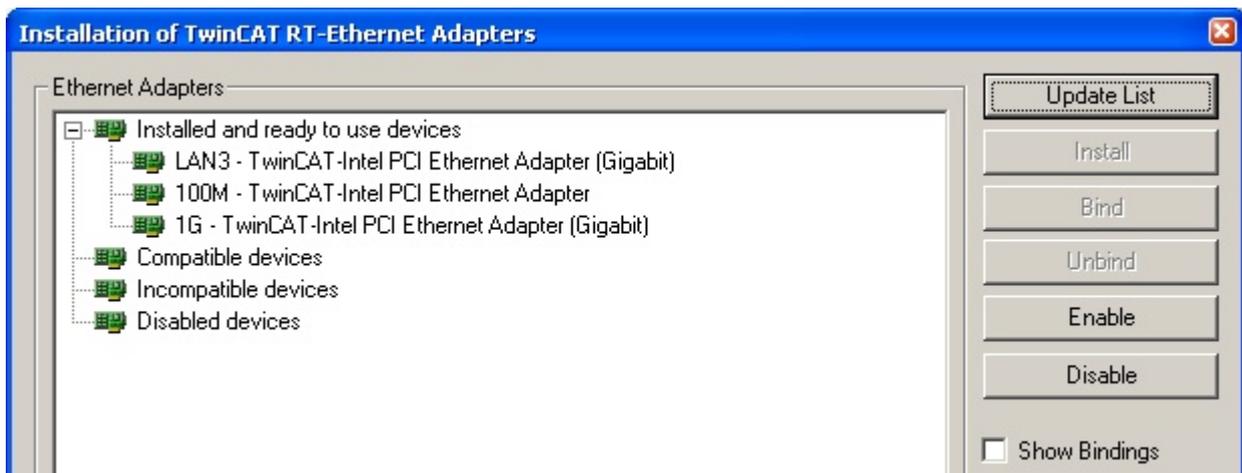


Abb. 86: Übersicht Netzwerkschnittstellen

Hier können nun Schnittstellen, die unter „Kompatible Geräte“ aufgeführt sind, über den „Install“ Button mit dem Treiber belegt werden. Eine Installation des Treibers auf inkompatiblen Devices sollte nicht vorgenommen werden.

Ein Windows-Warnhinweis bezüglich des unsignierten Treibers kann ignoriert werden.

**Alternativ** kann auch wie im Kapitel *Offline Konfigurationserstellung, Abschnitt „Anlegen des Geräts EtherCAT“* [► 97] beschrieben, zunächst ein EtherCAT-Gerät eingetragen werden, um dann über dessen Eigenschaften (Karteireiter „Adapter“, Button „Kompatible Geräte...“) die kompatiblen Ethernet Ports einzusehen:

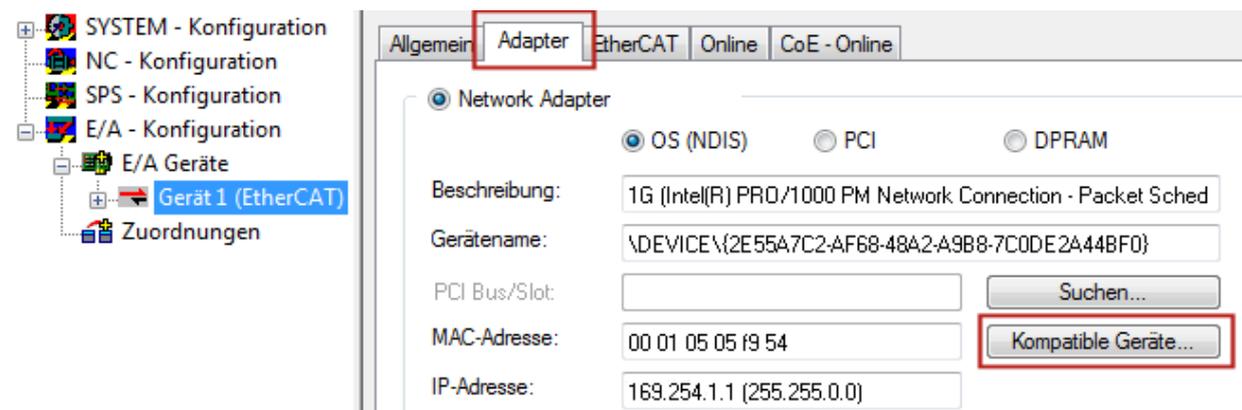
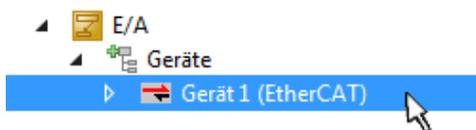


Abb. 87: Eigenschaft von EtherCAT Gerät (TwinCAT 2): Klick auf „Kompatible Geräte...“ von „Adapter“

TwinCAT 3: Die Eigenschaften des EtherCAT-Gerätes können mit Doppelklick auf „Gerät .. (EtherCAT)“ im Projektmappen-Explorer unter „E/A“ geöffnet werden:



Nach der Installation erscheint der Treiber aktiviert in der Windows-Übersicht der einzelnen Netzwerkschnittstelle (Windows Start → Systemsteuerung → Netzwerk)

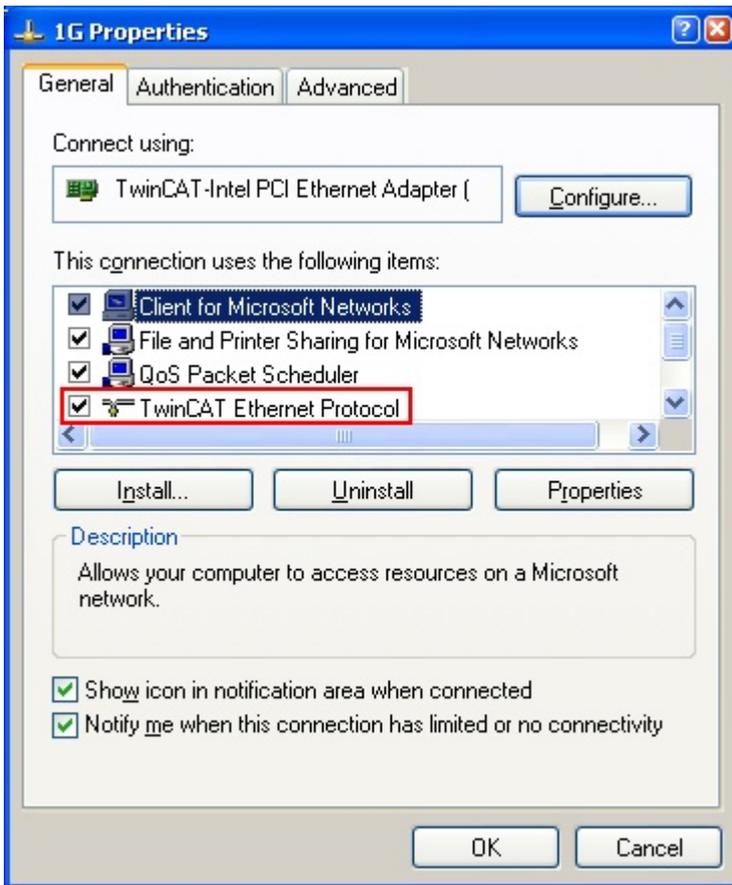


Abb. 88: Windows-Eigenschaften der Netzwerkschnittstelle

Eine korrekte Einstellung des Treibers könnte wie folgt aussehen:

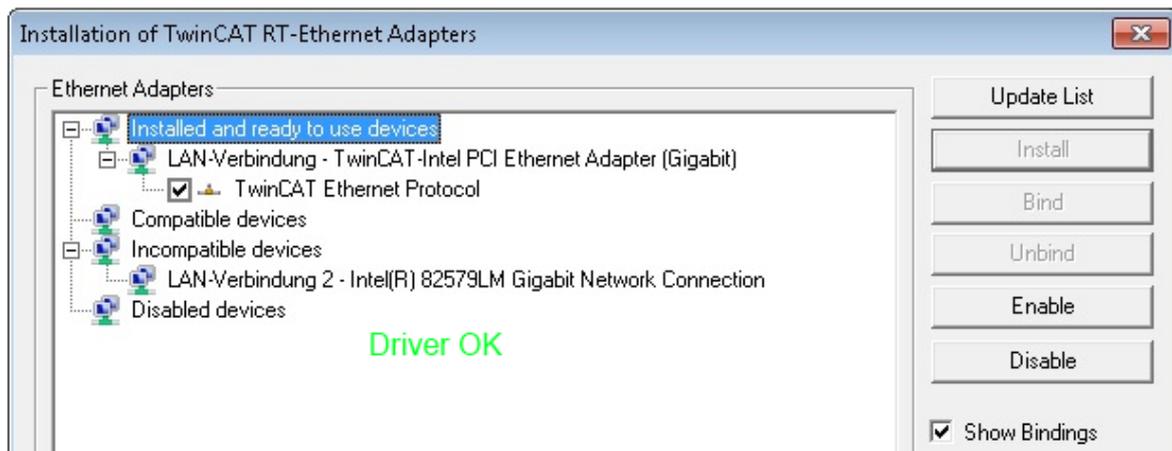


Abb. 89: Beispielhafte korrekte Treiber-Einstellung des Ethernet Ports

Andere mögliche Einstellungen sind zu vermeiden:

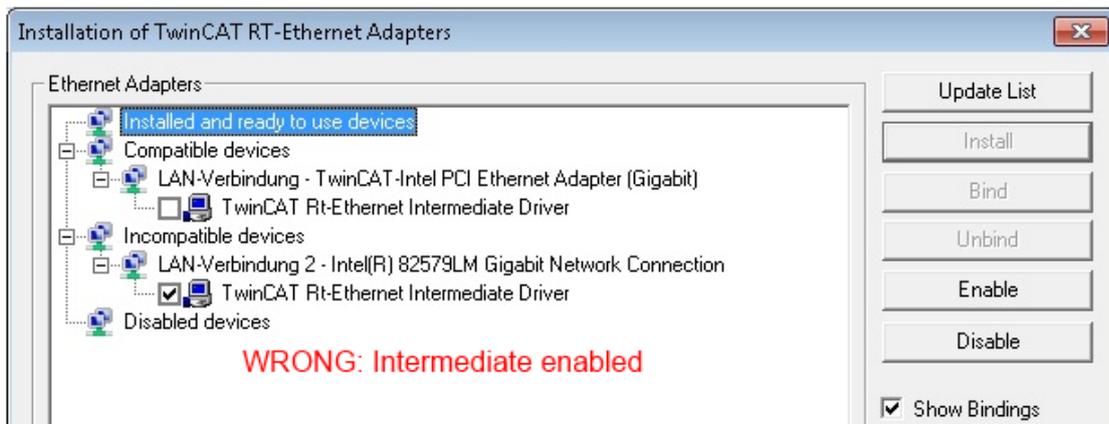
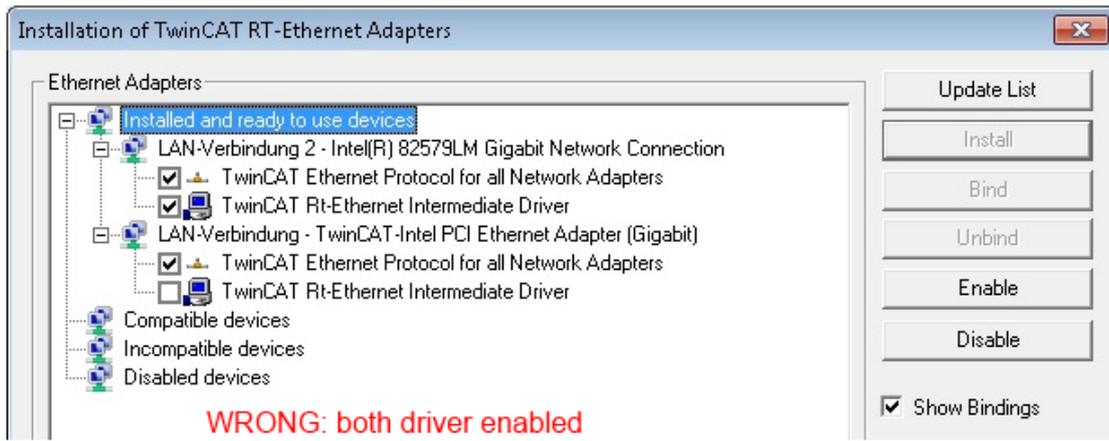


Abb. 90: Fehlerhafte Treiber-Einstellungen des Ethernet Ports

## IP-Adresse des verwendeten Ports

### **i** IP Adresse/DHCP

In den meisten Fällen wird ein Ethernet-Port, der als EtherCAT-Gerät konfiguriert wird, keine allgemeinen IP-Pakete transportieren. Deshalb und für den Fall, dass eine EL6601 oder entsprechende Geräte eingesetzt werden, ist es sinnvoll, über die Treiber-Einstellung „Internet Protocol TCP/IP“ eine feste IP-Adresse für diesen Port zu vergeben und DHCP zu deaktivieren. Dadurch entfällt die Wartezeit, bis sich der DHCP-Client des Ethernet Ports eine Default-IP-Adresse zuteilt, weil er keine Zuteilung eines DHCP-Servers erhält. Als Adressraum empfiehlt sich z. B. 192.168.x.x.

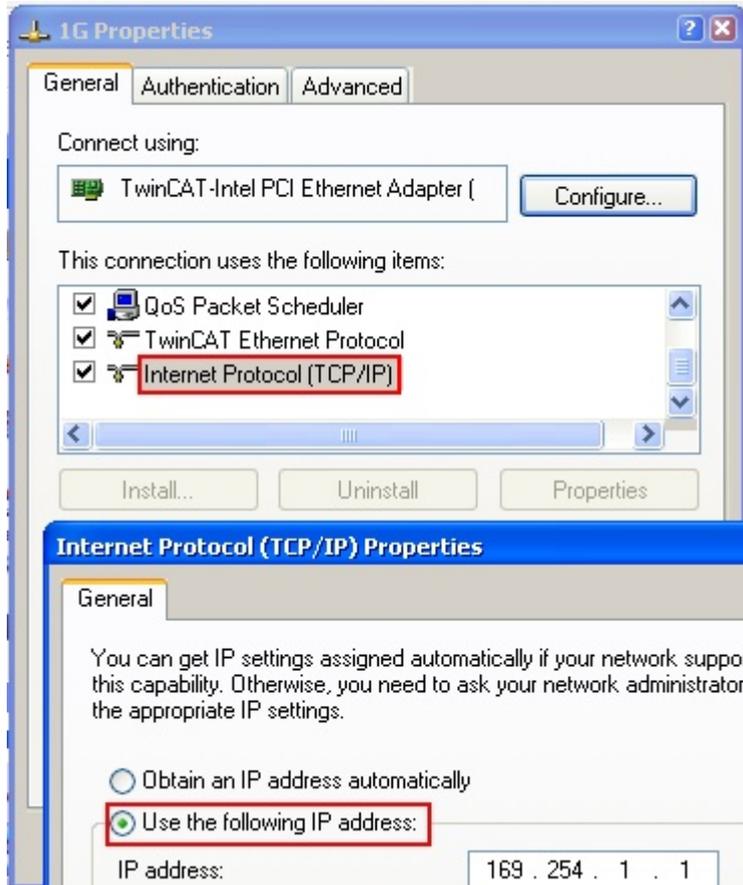


Abb. 91: TCP/IP-Einstellung des Ethernet Ports

## 5.2.2 Hinweise ESI-Gerätebeschreibung

### Installation der neuesten ESI-Device-Description

Der TwinCAT EtherCAT Master/System Manager benötigt zur Konfigurationserstellung im Online- und Offline-Modus die Gerätebeschreibungsdateien der zu verwendeten Geräte. Diese Gerätebeschreibungen sind die so genannten ESI (EtherCAT Slave Information) in Form von XML-Dateien. Diese Dateien können vom jeweiligen Hersteller angefordert werden bzw. werden zum Download bereitgestellt. Eine \*.xml-Datei kann dabei mehrere Gerätebeschreibungen enthalten.

Auf der [Beckhoff Website](#) werden die ESI für Beckhoff EtherCAT Geräte bereitgehalten.

Die ESI-Dateien sind im Installationsverzeichnis von TwinCAT abzulegen.

Standardeinstellungen:

- **TwinCAT 2:** C:\TwinCAT\IO\EtherCAT
- **TwinCAT 3:** C:\TwinCAT\3.1\Config\Io\EtherCAT

Beim Öffnen eines neuen System Manager-Fensters werden die Dateien einmalig eingelesen, wenn sie sich seit dem letzten System Manager-Fenster geändert haben.

TwinCAT bringt bei der Installation den Satz an Beckhoff-ESI-Dateien mit, der zum Erstellungszeitpunkt des TwinCAT builds aktuell war.

Ab TwinCAT 2.11 / TwinCAT 3 kann aus dem System Manager heraus das ESI-Verzeichnis aktualisiert werden, wenn der Programmier-PC mit dem Internet verbunden ist; unter

**TwinCAT 2:** Options → „Update EtherCAT Device Descriptions“

**TwinCAT 3:** TwinCAT → EtherCAT Devices → “Update Device Descriptions (via ETG Website)...”

Hierfür steht der [TwinCAT ESI Updater \[► 96\]](#) zur Verfügung.



#### ESI

Zu den \*.xml-Dateien gehören die so genannten \*.xsd-Dateien, die den Aufbau der ESI-XML-Dateien beschreiben. Bei einem Update der ESI-Gerätebeschreibungen sind deshalb beide Dateiarnten ggf. zu aktualisieren.

### Geräteunterscheidung

EtherCAT Geräte/Slaves werden durch vier Eigenschaften unterschieden, aus denen die vollständige Gerätebezeichnung zusammengesetzt wird. Beispielsweise setzt sich die Gerätebezeichnung „EL2521-0025-1018“ zusammen aus:

- Familienschlüssel „EL“
- Name „2521“
- Typ „0025“
- und Revision „1018“

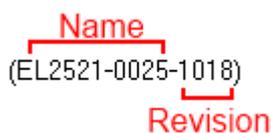


Abb. 92: Gerätebezeichnung: Struktur

Die Bestellbezeichnung aus Typ + Version (hier: EL2521-0010) beschreibt die Funktion des Gerätes. Die Revision gibt den technischen Fortschritt wieder und wird von Beckhoff verwaltet. Prinzipiell kann ein Gerät mit höherer Revision ein Gerät mit niedrigerer Revision ersetzen, wenn z. B. in der Dokumentation nicht anders angegeben. Jeder Revision zugehörig ist eine eigene ESI-Beschreibung. Siehe weitere [Hinweise \[► 10\]](#).

**Online Description**

Wird die EtherCAT Konfiguration online durch Scannen real vorhandener Teilnehmer erstellt (s. Kapitel Online Erstellung) und es liegt zu einem vorgefundenen Slave (ausgezeichnet durch Name und Revision) keine ESI-Beschreibung vor, fragt der System Manager, ob er die im Gerät vorliegende Beschreibung verwenden soll. Der System Manager benötigt in jedem Fall diese Information, um die zyklische und azyklische Kommunikation mit dem Slave richtig einstellen zu können.

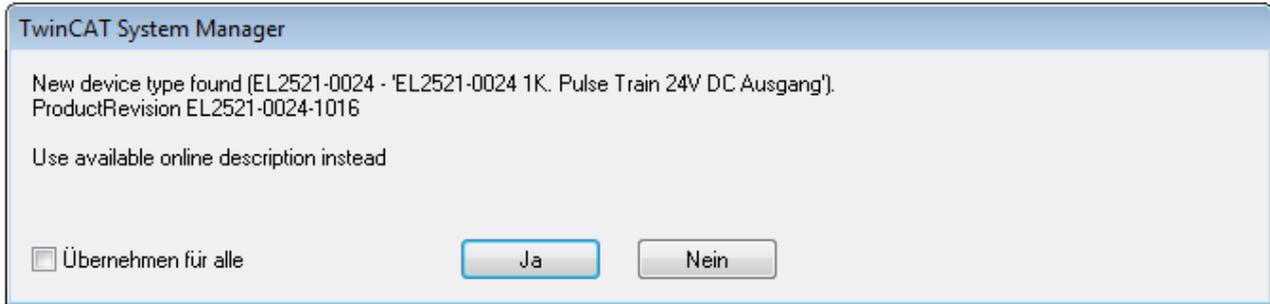


Abb. 93: Hinweisfenster OnlineDescription (TwinCAT 2)

In TwinCAT 3 erscheint ein ähnliches Fenster, das auch das Web-Update anbietet:

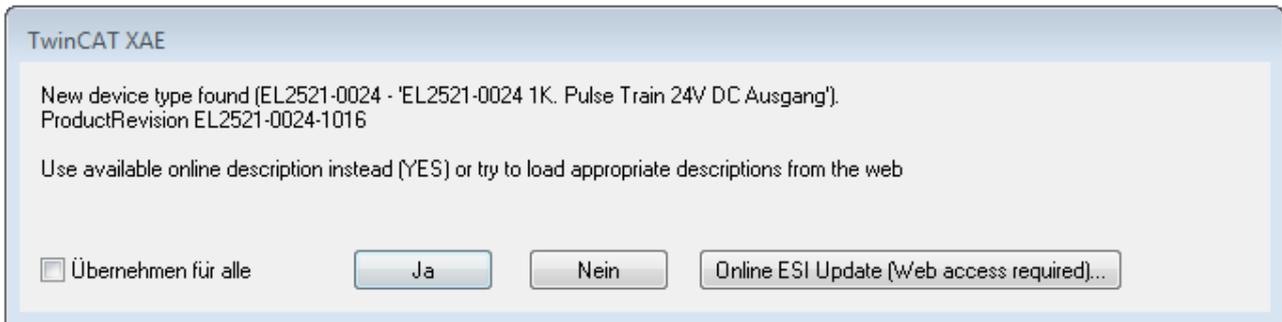


Abb. 94: Hinweisfenster OnlineDescription (TwinCAT 3)

Wenn möglich, ist das Yes abzulehnen und vom Geräte-Hersteller die benötigte ESI anzufordern. Nach Installation der XML/XSD-Datei ist der Konfigurationsvorgang erneut vorzunehmen.

**HINWEIS**

**Veränderung der „üblichen“ Konfiguration durch Scan**

- ✓ für den Fall eines durch Scan entdeckten aber TwinCAT noch unbekanntes Geräts sind zwei Fälle zu unterscheiden. Hier am Beispiel der EL2521-0000 in der Revision 1019:
  - a) für das Gerät EL2521-0000 liegt überhaupt keine ESI vor, weder für die Revision 1019 noch für eine ältere Revision. Dann ist vom Hersteller (hier: Beckhoff) die ESI anzufordern.
  - b) für das Gerät EL2521-0000 liegt eine ESI nur in älterer Revision vor, z. B. 1018 oder 1017. Dann sollte erst betriebsintern überprüft werden, ob die Ersatzteilhaltung überhaupt die Integration der erhöhten Revision in die Konfiguration zulässt. Üblicherweise bringt eine neue/größere Revision auch neue Features mit. Wenn diese nicht genutzt werden sollen, kann ohne Bedenken mit der bisherigen Revision 1018 in der Konfiguration weitergearbeitet werden. Dies drückt auch die Beckhoff Kompatibilitätsregel aus.

Siehe dazu insbesondere das Kapitel „[Allgemeine Hinweise zur Verwendung von Beckhoff EtherCAT IO-Komponenten](#)“ und zur manuellen Konfigurationserstellung das Kapitel „[Offline Konfigurationserstellung](#) [▶ 97]“.

Wird dennoch die Online Description verwendet, liest der System Manager aus dem im EtherCAT Slave befindlichen EEPROM eine Kopie der Gerätebeschreibung aus. Bei komplexen Slaves kann die EEPROM-Größe u. U. nicht ausreichend für die gesamte ESI sein, weshalb im Konfigurator dann eine *unvollständige* ESI vorliegt. Deshalb wird für diesen Fall die Verwendung einer offline ESI-Datei vorrangig empfohlen.

Der System Manager legt bei „online“ erfassten Gerätebeschreibungen in seinem ESI-Verzeichnis eine neue Datei „OnlineDescription0000...xml“ an, die alle online ausgelesenen ESI-Beschreibungen enthält.

OnlineDescriptionCache000000002.xml

Abb. 95: Vom System Manager angelegt OnlineDescription.xml

Soll daraufhin ein Slave manuell in die Konfiguration eingefügt werden, sind „online“ erstellte Slaves durch ein vorangestelltes „>“ Symbol in der Auswahlliste gekennzeichnet (siehe Abbildung *Kennzeichnung einer online erfassten ESI am Beispiel EL2521*).

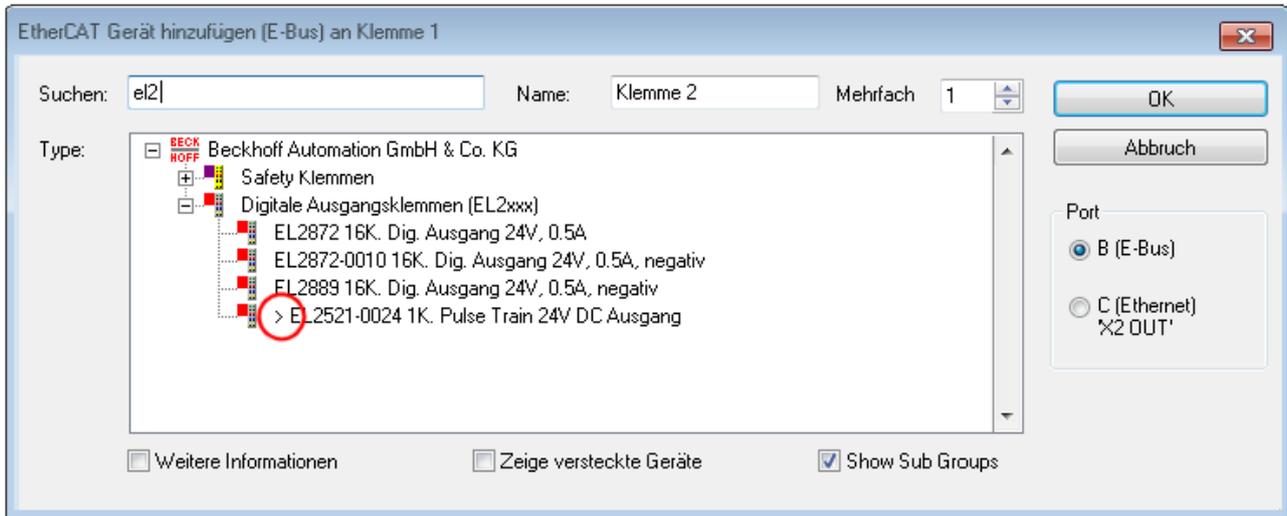


Abb. 96: Kennzeichnung einer online erfassten ESI am Beispiel EL2521

Wurde mit solchen ESI-Daten gearbeitet und liegen später die herstellereigenen Dateien vor, ist die OnlineDescription....xml wie folgt zu löschen:

- alle System Managerfenster schließen
- TwinCAT in Konfig-Mode neu starten
- „OnlineDescription0000...xml“ löschen
- TwinCAT System Manager wieder öffnen

Danach darf diese Datei nicht mehr zu sehen sein, Ordner ggf. mit <F5> aktualisieren.

### **i** OnlineDescription unter TwinCAT 3.x

Zusätzlich zu der oben genannten Datei „OnlineDescription0000...xml“ legt TwinCAT 3.x auch einen so genannten EtherCAT-Cache mit neuentdeckten Geräten an, z. B. unter Windows 7 unter

*C:\User\[USERNAME]\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml*

(Spracheinstellungen des Betriebssystems beachten!)

Diese Datei ist im gleichen Zuge wie die andere Datei zu löschen.

### **Fehlerhafte ESI-Datei**

Liegt eine fehlerhafte ESI-Datei vor die vom System Manager nicht eingelesen werden kann, meldet dies der System Manager durch ein Hinweisfenster.

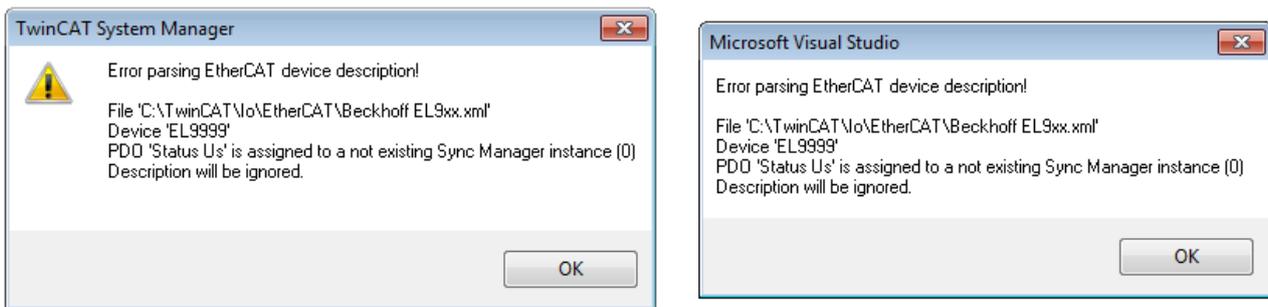


Abb. 97: Hinweisfenster fehlerhafte ESI-Datei (links: TwinCAT 2; rechts: TwinCAT 3)

Ursachen dafür können sein

- Aufbau der \*.xml entspricht nicht der zugehörigen \*.xsd-Datei → prüfen Sie die Ihnen vorliegenden Schemata
- Inhalt kann nicht in eine Gerätebeschreibung übersetzt werden → Es ist der Hersteller der Datei zu kontaktieren

### 5.2.3 TwinCAT ESI Updater

Ab TwinCAT 2.11 kann der System Manager bei Online-Zugang selbst nach aktuellen Beckhoff ESI-Dateien suchen:



Abb. 98: Anwendung des ESI Updater (>=TwinCAT 2.11)

Der Aufruf erfolgt unter:  
„Options“ → „Update EtherCAT Device Descriptions“.

Auswahl bei TwinCAT 3:

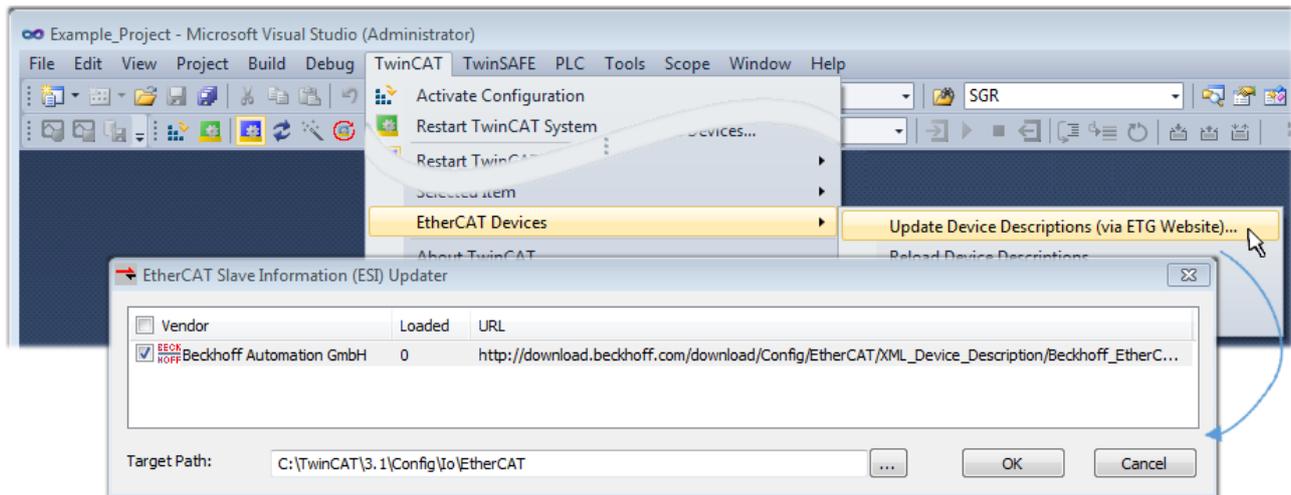


Abb. 99: Anwendung des ESI Updater (TwinCAT 3)

Der ESI Updater ist eine bequeme Möglichkeit, die von den EtherCAT Herstellern bereitgestellten ESIs automatisch über das Internet in das TwinCAT-Verzeichnis zu beziehen (ESI = EtherCAT slave information). Dazu greift TwinCAT auf die bei der ETG hinterlegte zentrale ESI-URL-Verzeichnisliste zu; die Einträge sind dann unveränderbar im Updater-Dialog zu sehen.

Der Aufruf erfolgt unter:  
„TwinCAT“ → „EtherCAT Devices“ → „Update Device Description (via ETG Website)...“.

### 5.2.4 Unterscheidung Online/Offline

Die Unterscheidung Online/Offline bezieht sich auf das Vorhandensein der tatsächlichen I/O-Umgebung (Antriebe, Klemmen, EJ-Module). Wenn die Konfiguration im Vorfeld der Anlagenerstellung z. B. auf einem Laptop als Programmiersystem erstellt werden soll, ist nur die „Offline-Konfiguration“ möglich. Dann müssen alle Komponenten händisch in der Konfiguration z. B. nach Elektro-Planung eingetragen werden.

Ist die vorgesehene Steuerung bereits an das EtherCAT System angeschlossen, alle Komponenten mit Spannung versorgt und die Infrastruktur betriebsbereit, kann die TwinCAT Konfiguration auch vereinfacht durch das so genannte „Scannen“ vom Runtime-System aus erzeugt werden. Dies ist der so genannte Online-Vorgang.

In jedem Fall prüft der EtherCAT Master bei jedem realen Hochlauf, ob die vorgefundenen Slaves der Konfiguration entsprechen. Dieser Test kann in den erweiterten Slave-Einstellungen parametrisiert werden. Siehe hierzu den [Hinweis „Installation der neuesten ESI-XML-Device-Description“](#) [► 92].

#### Zur Konfigurationserstellung

- muss die reale EtherCAT-Hardware (Geräte, Koppler, Antriebe) vorliegen und installiert sein.

- müssen die Geräte/Module über EtherCAT-Kabel bzw. im Klemmenstrang so verbunden sein wie sie später eingesetzt werden sollen.
- müssen die Geräte/Module mit Energie versorgt werden und kommunikationsbereit sein.
- muss TwinCAT auf dem Zielsystem im CONFIG-Modus sein.

**Der Online-Scan-Vorgang setzt sich zusammen aus:**

- Erkennen des EtherCAT-Gerätes [▶\_102] (Ethernet-Port am IPC)
- Erkennen der angeschlossenen EtherCAT-Teilnehmer [▶\_103]. Dieser Schritt kann auch unabhängig vom vorangehenden durchgeführt werden.
- Problembehandlung [▶\_106]

Auch kann der Scan bei bestehender Konfiguration [▶\_107] zum Vergleich durchgeführt werden.

## 5.2.5 OFFLINE Konfigurationserstellung

### Anlegen des Geräts EtherCAT

In einem leeren System Manager Fenster muss zuerst ein EtherCAT Gerät angelegt werden.



Abb. 100: Anfügen eines EtherCAT Device: links TwinCAT 2; rechts TwinCAT 3

Für eine EtherCAT I/O Anwendung mit EtherCAT Slaves ist der „EtherCAT“ Typ auszuwählen. „EtherCAT Automation Protocol via EL6601“ ist für den bisherigen Publisher/Subscriber-Dienst in Kombination mit einer EL6601/EL6614 Klemme auszuwählen.

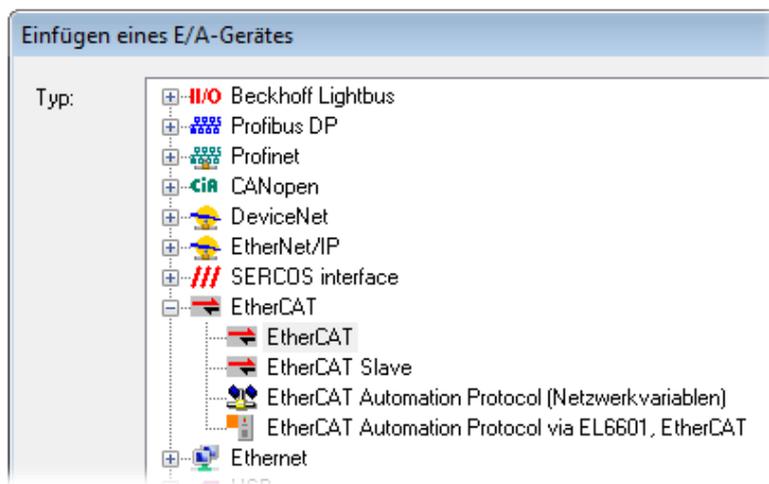


Abb. 101: Auswahl EtherCAT Anschluss (TwinCAT 2.11, TwinCAT 3)

Diesem virtuellen Gerät ist dann ein realer Ethernet Port auf dem Laufzeitsystem zuzuordnen.

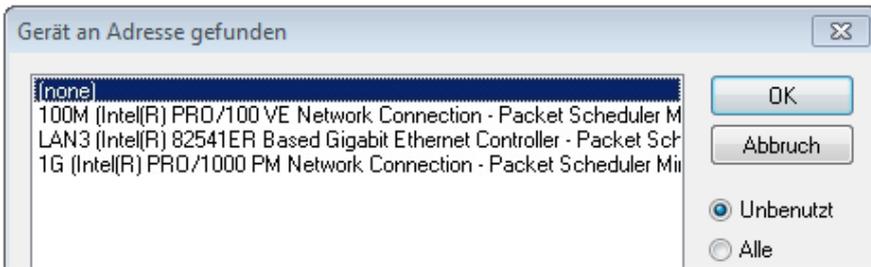


Abb. 102: Auswahl Ethernet Port

Diese Abfrage kann beim Anlegen des EtherCAT-Gerätes automatisch erscheinen, oder die Zuordnung kann später im Eigenschaftendialog gesetzt/geändert werden; siehe Abb. „Eigenschaften EtherCAT Gerät (TwinCAT 2)“.

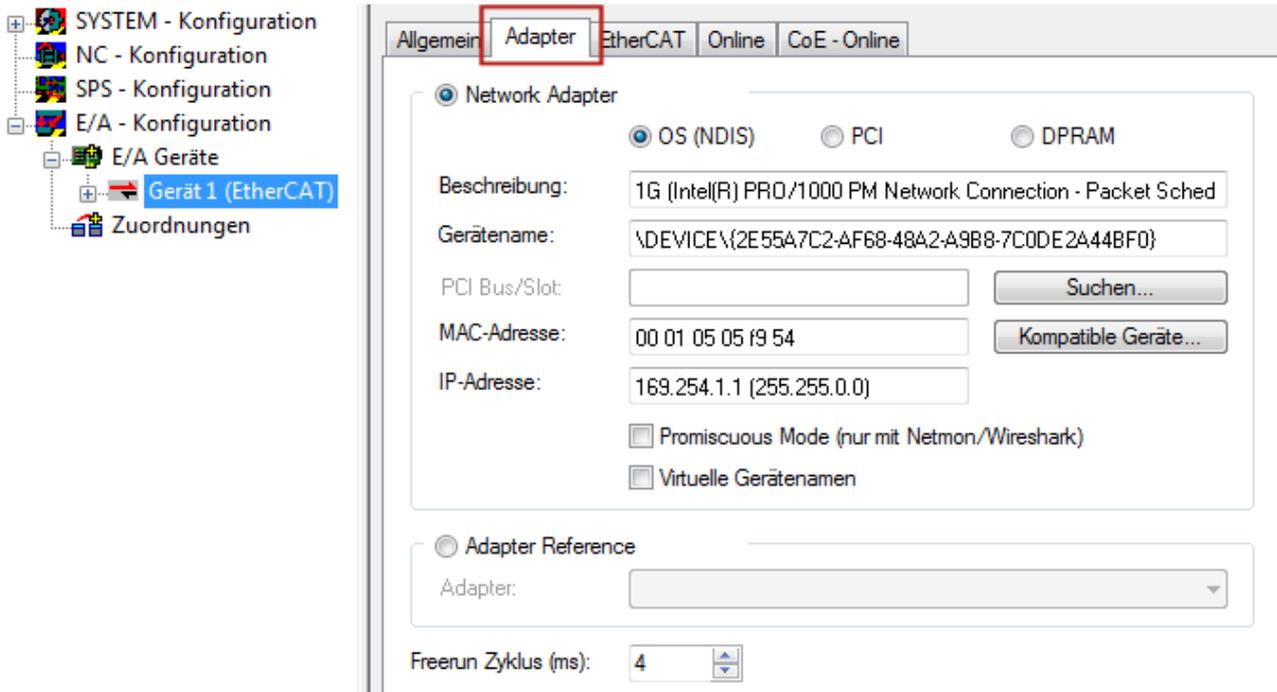
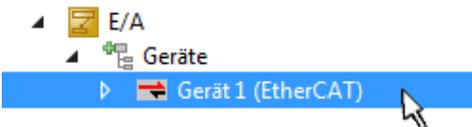


Abb. 103: Eigenschaften EtherCAT Gerät (TwinCAT 2)

TwinCAT 3: Die Eigenschaften des EtherCAT-Gerätes können mit Doppelklick auf „Gerät .. (EtherCAT)“ im Projektmappen-Explorer unter „E/A“ geöffnet werden:



**i Auswahl Ethernet Port**

Es können nur Ethernet Ports für ein EtherCAT Gerät ausgewählt werden, für die der TwinCAT Realtime-Treiber installiert ist. Dies muss für jeden Port getrennt vorgenommen werden. Siehe dazu die entsprechende [Installationsseite](#) [▶ 87](#)].

**Definieren von EtherCAT Slaves**

Durch Rechtsklick auf ein Gerät im Konfigurationsbaum können weitere Geräte angefügt werden.

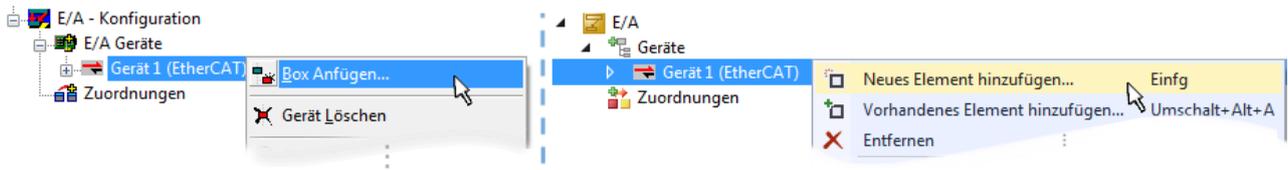


Abb. 104: Anfügen von EtherCAT Geräten (links: TwinCAT 2; rechts: TwinCAT 3)

Es öffnet sich der Dialog zur Auswahl des neuen Gerätes. Es werden nur Geräte angezeigt für die ESI-Dateien hinterlegt sind.

Die Auswahl bietet auch nur Geräte an, die an dem vorher angeklickten Gerät anzufügen sind - dazu wird die an diesem Port mögliche Übertragungsphysik angezeigt (Abb. „Auswahldialog neues EtherCAT Gerät“, A). Es kann sich um kabelgebundene FastEthernet-Ethernet-Physik mit PHY-Übertragung handeln, dann ist wie in Abb. „Auswahldialog neues EtherCAT Gerät“ nur ebenfalls kabelgebundenes Geräte auswählbar. Verfügt das vorangehende Gerät über mehrere freie Ports (z. B. EK1122 oder EK1100), kann auf der rechten Seite (A) der gewünschte Port angewählt werden.

Übersicht Übertragungsphysik

- „Ethernet“: Kabelgebunden 100BASE-TX: EK-Koppler, EP-Boxen, Geräte mit RJ45/M8/M12-Konnectore
- „E-Bus“: LVDS „Klemmenbus“ „EJ-Module“: EL/ES-Klemmen, diverse anreihbare Module

Das Suchfeld erleichtert das Auffinden eines bestimmten Gerätes (ab TwinCAT 2.11 bzw. TwinCAT 3).

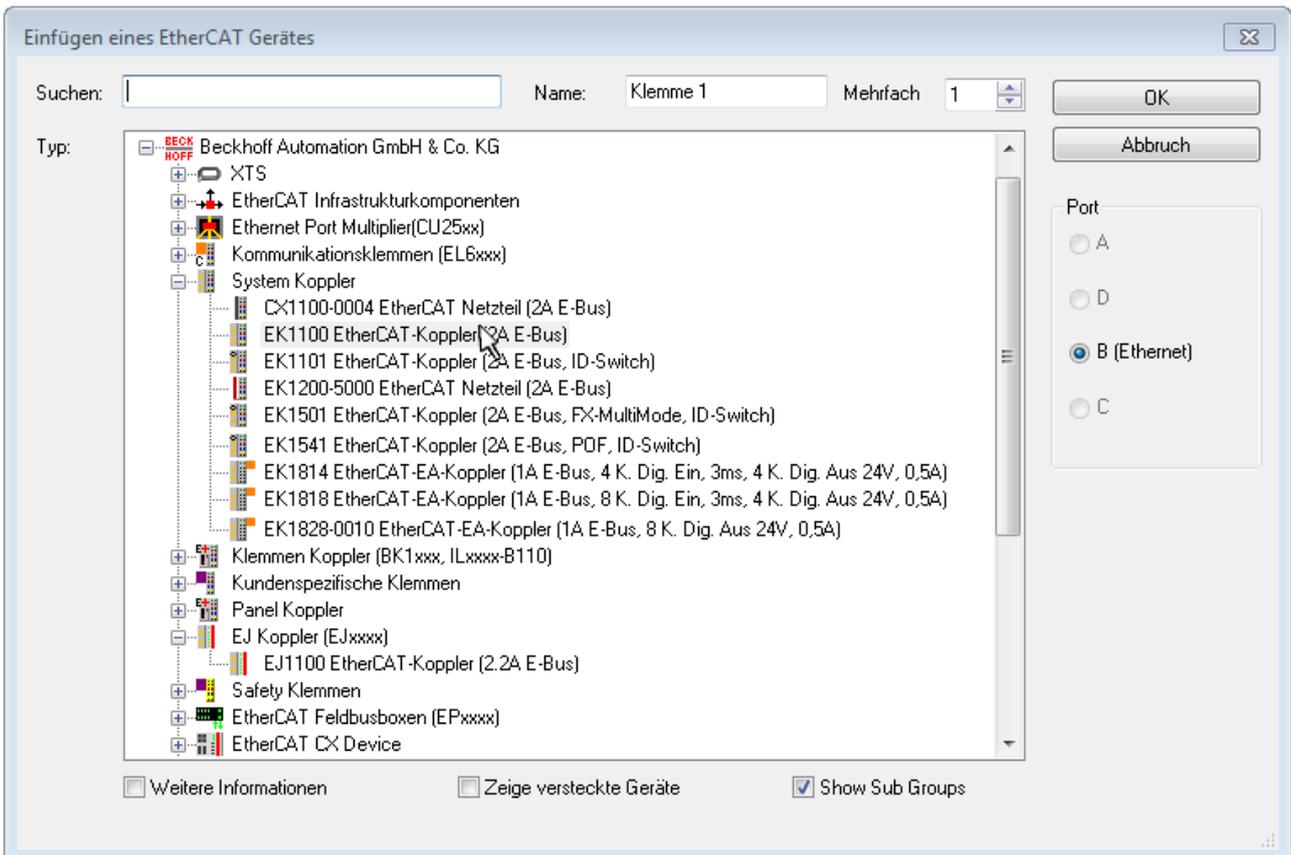


Abb. 105: Auswahldialog neues EtherCAT Gerät

Standardmäßig wird nur der Name/Typ des Gerätes als Auswahlkriterium verwendet. Für eine gezielte Auswahl einer bestimmten Revision des Gerätes kann die Revision als „Extended Information“ eingeblendet werden.

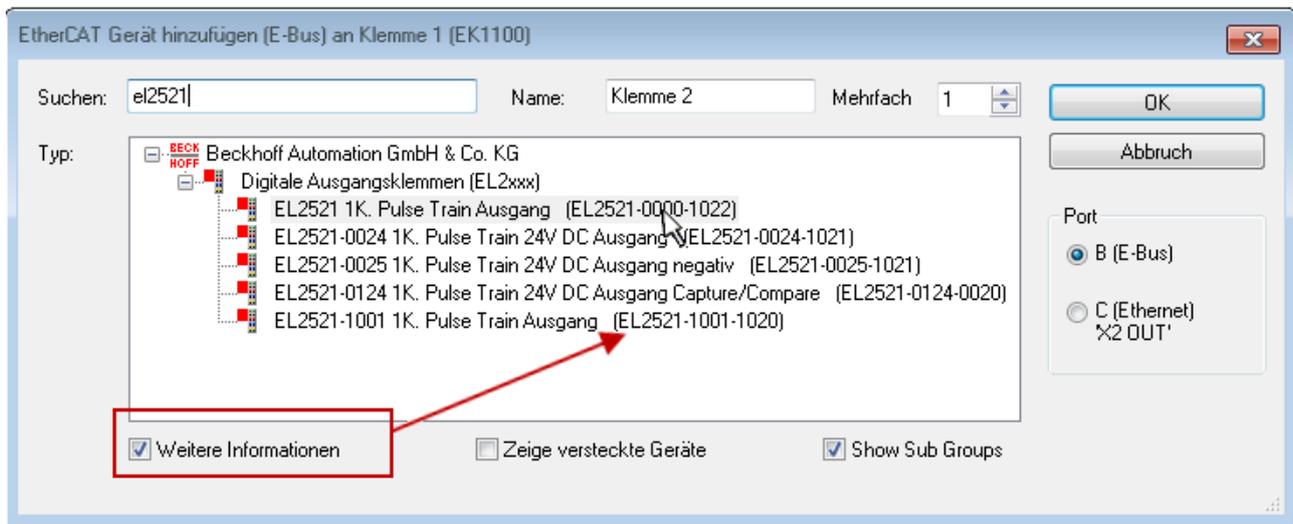


Abb. 106: Anzeige Geräte-Revision

Oft sind aus historischen oder funktionalen Gründen mehrere Revisionen eines Gerätes erzeugt worden, z. B. durch technologische Weiterentwicklung. Zur vereinfachten Anzeige (s. Abb. „Auswahldialog neues EtherCAT Gerät“) wird bei Beckhoff Geräten nur die letzte (=höchste) Revision und damit der letzte Produktionsstand im Auswahldialog angezeigt. Sollen alle im System als ESI-Beschreibungen vorliegenden Revisionen eines Gerätes angezeigt werden, ist die Checkbox „Show Hidden Devices“ zu markieren, s. Abb. „Anzeige vorhergehender Revisionen“.

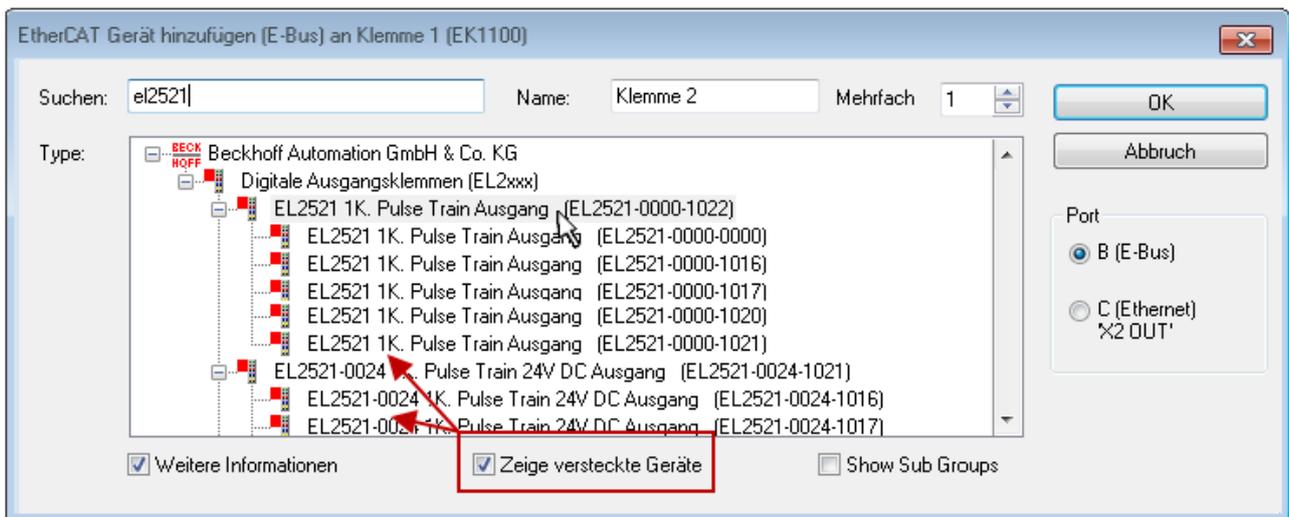


Abb. 107: Anzeige vorhergehender Revisionen

## ● Geräte-Auswahl nach Revision, Kompatibilität

**i** Mit der ESI-Beschreibung wird auch das Prozessabbild, die Art der Kommunikation zwischen Master und Slave/Gerät und ggf. Geräte-Funktionen definiert. Damit muss das reale Gerät (Firmware wenn vorhanden) die Kommunikationsanfragen/-einstellungen des Masters unterstützen. Dies ist abwärtskompatibel der Fall, d. h. neuere Geräte (höhere Revision) sollen es auch unterstützen, wenn der EtherCAT Master sie als eine ältere Revision anspricht. Als Beckhoff-Kompatibilitätsregel für EtherCAT-Klemmen/ Boxen/ EJ-Module ist anzunehmen:

### Geräte-Revision in der Anlage >= Geräte-Revision in der Konfiguration

Dies erlaubt auch den späteren Austausch von Geräten ohne Veränderung der Konfiguration (abweichende Vorgaben bei Antrieben möglich).

**Beispiel**

In der Konfiguration wird eine EL2521-0025-**1018** vorgesehen, dann kann real eine EL2521-0025-**1018** oder höher (-**1019**, -**1020**) eingesetzt werden.

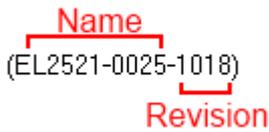


Abb. 108: Name/Revision Klemme

Wenn im TwinCAT System aktuelle ESI-Beschreibungen vorliegen, entspricht der im Auswahldialog als letzte Revision angebotene Stand dem Produktionsstand von Beckhoff. Es wird empfohlen, bei Erstellung einer neuen Konfiguration jeweils diesen letzten Revisionsstand eines Gerätes zu verwenden, wenn aktuell produzierte Beckhoff-Geräte in der realen Applikation verwendet werden. Nur wenn ältere Geräte aus Lagerbeständen in der Applikation verbaut werden sollen, ist es sinnvoll eine ältere Revision einzubinden.

Das Gerät stellt sich dann mit seinem Prozessabbild im Konfigurationsbaum dar und kann nur parametrieren werden: Verlinkung mit der Task, CoE/DC-Einstellungen, PlugIn-Definition, StartUp-Einstellungen, ...

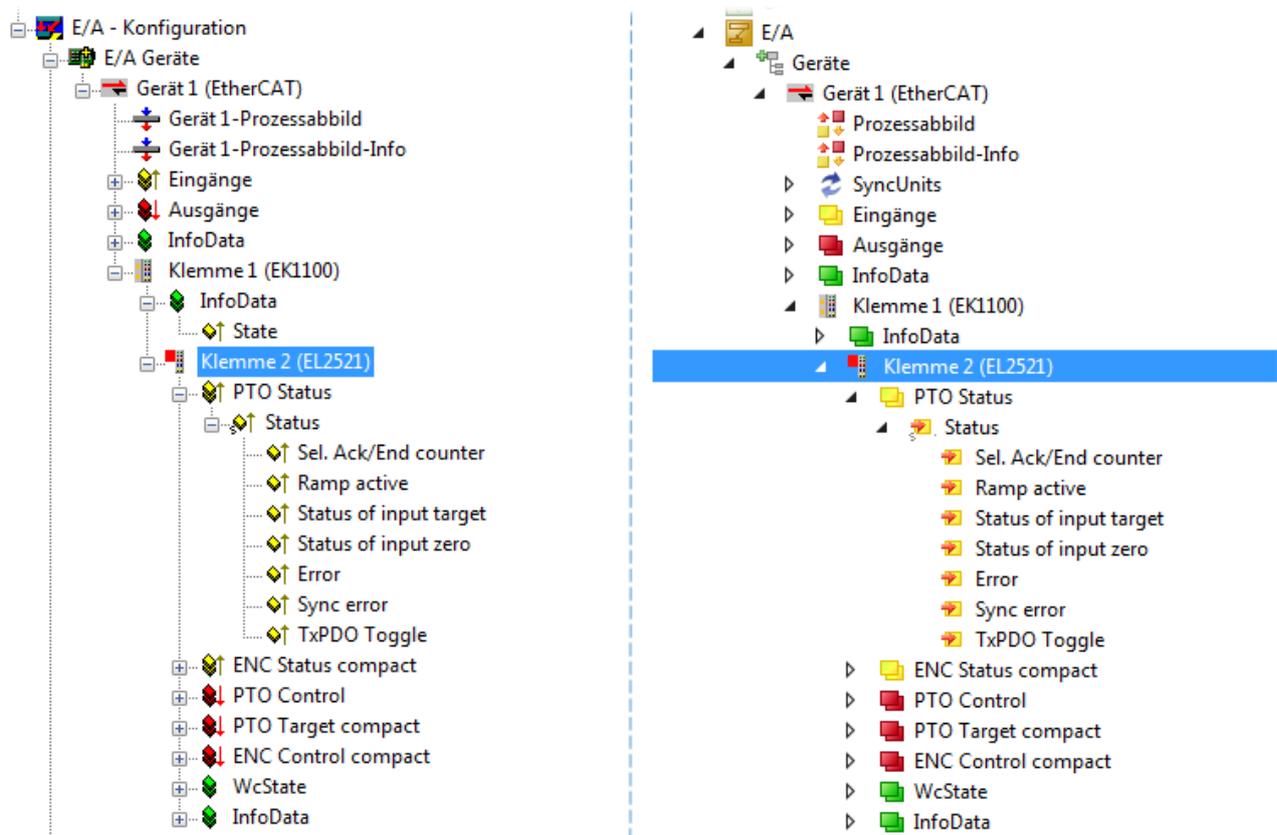


Abb. 109: EtherCAT Klemme im TwinCAT-Baum (links: TwinCAT 2; rechts: TwinCAT 3)

## 5.2.6 ONLINE Konfigurationserstellung

### Erkennen/Scan des Geräts EtherCAT

Befindet sich das TwinCAT-System im CONFIG-Modus, kann online nach Geräten gesucht werden. Erkennbar ist dies durch ein Symbol unten rechts in der Informationsleiste:

- bei TwinCAT 2 durch eine blaue Anzeige „Config Mode“ im System Manager-Fenster:  .
- bei der Benutzeroberfläche der TwinCAT 3 Entwicklungsumgebung durch ein Symbol  .

TwinCAT lässt sich in diesem Modus versetzen:

- TwinCAT 2: durch Auswahl von  aus der Menüleiste oder über „Aktionen“ → „Starten/Restarten von TwinCAT in Konfig-Modus“
- TwinCAT 3: durch Auswahl von  aus der Menüleiste oder über „TWINCAT“ → „Restart TwinCAT (Config Mode)“

### ● Online Scannen im Config Mode

**i** Die Online-Suche im RUN-Modus (produktiver Betrieb) ist nicht möglich. Es ist die Unterscheidung zwischen TwinCAT-Programmiersystem und TwinCAT-Zielsystem zu beachten.

Das TwinCAT 2-Icon () bzw. TwinCAT 3-Icon () in der Windows Taskleiste stellt immer den TwinCAT-Modus des lokalen IPC dar. Im System Manager-Fenster von TwinCAT 2 bzw. in der Benutzeroberfläche von TwinCAT 3 wird dagegen der TwinCAT-Zustand des Zielsystems angezeigt.



Abb. 110: Unterscheidung Lokalsystem/ Zielsystem (links: TwinCAT 2; rechts: TwinCAT 3)

Im Konfigurationsbaum bringt uns ein Rechtsklick auf den General-Punkt „I/O Devices“ zum Such-Dialog.

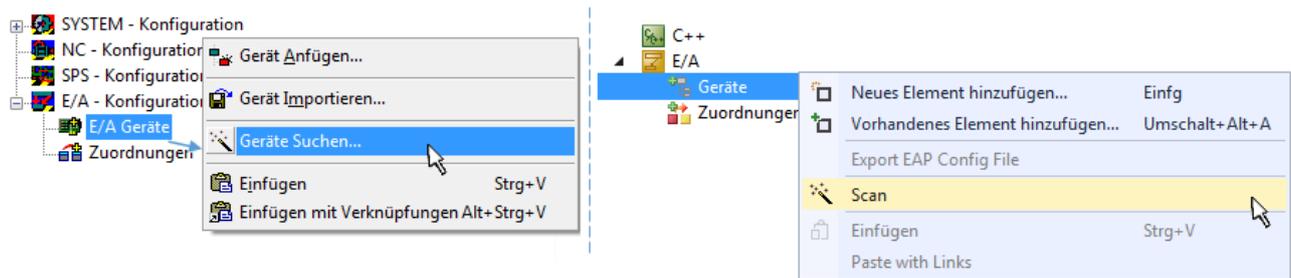


Abb. 111: Scan Devices (links: TwinCAT 2; rechts: TwinCAT 3)

Dieser Scan-Modus versucht nicht nur EtherCAT-Geräte (bzw. die als solche nutzbaren Ethernet-Ports) zu finden, sondern auch NOVRAM, Feldbuskarten, SMB etc. Nicht alle Geräte können jedoch automatisch gefunden werden.

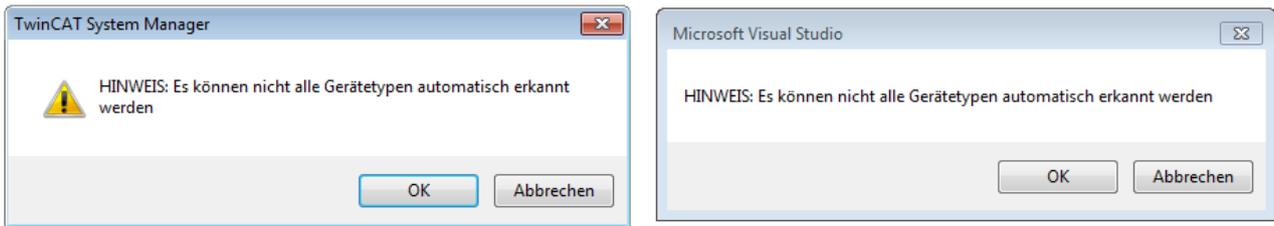


Abb. 112: Hinweis automatischer GeräteScan (links: TwinCAT 2; rechts: TwinCAT 3)

Ethernet Ports mit installierten TwinCAT Realtime-Treiber werden als „RT-Ethernet“ Geräte angezeigt. Testweise wird an diesen Ports ein EtherCAT-Frame verschickt. Erkennt der Scan-Agent an der Antwort, dass ein EtherCAT-Slave angeschlossen ist, wird der Port allerdings gleich als „EtherCAT Device“ angezeigt.

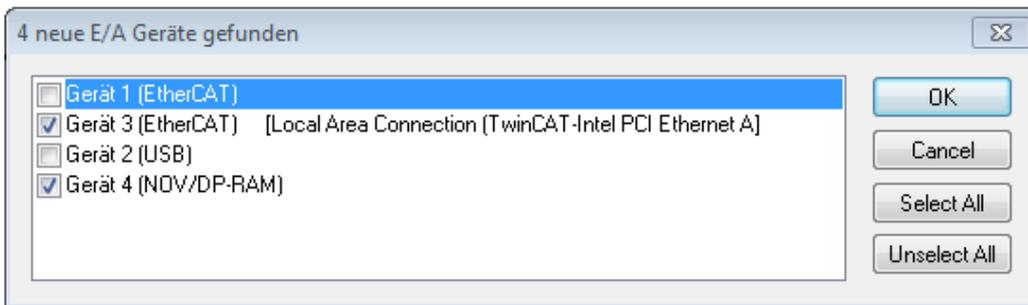


Abb. 113: Erkannte Ethernet-Geräte

Über entsprechende Kontrollkästchen können Geräte ausgewählt werden (wie in der Abb. „Erkannte Ethernet-Geräte“ gezeigt ist z. B. Gerät 3 und Gerät 4 ausgewählt). Für alle angewählten Geräte wird nach Bestätigung „OK“ im nachfolgenden ein Teilnehmer-Scan vorgeschlagen, s. Abb. „Scan-Abfrage nach dem automatischen Anlegen eines EtherCAT Gerätes“.

**● Auswahl Ethernet Port**

**i** Es können nur Ethernet Ports für ein EtherCAT Gerät ausgewählt werden, für die der TwinCAT Realtime-Treiber installiert ist. Dies muss für jeden Port getrennt vorgenommen werden. Siehe dazu die entsprechende [Installationsseite](#) [|> 87](#)].

**Erkennen/Scan der EtherCAT Teilnehmer**

**● Funktionsweise Online Scan**

**i** Beim Scan fragt der Master die Identity Informationen der EtherCAT Slaves aus dem Slave-EEPROM ab. Es werden Name und Revision zur Typbestimmung herangezogen. Die entsprechenden Geräte werden dann in den hinterlegten ESI-Daten gesucht und in dem dort definierten Default-Zustand in den Konfigurationsbaum eingebaut.

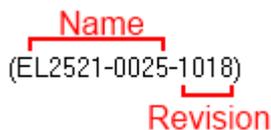


Abb. 114: Beispiel Default-Zustand

## HINWEIS

### Slave-Scan in der Praxis im Serienmaschinenbau

Die Scan-Funktion sollte mit Bedacht angewendet werden. Sie ist ein praktisches und schnelles Werkzeug, um für eine Inbetriebnahme eine Erst-Konfiguration als Arbeitsgrundlage zu erzeugen. Im Serienmaschinenbau bzw. bei Reproduktion der Anlage sollte die Funktion aber nicht mehr zur Konfigurationserstellung verwendet werden sondern ggf. zum [Vergleich](#) [► 107] mit der festgelegten Erst-Konfiguration.

Hintergrund: da Beckhoff aus Gründen der Produktpflege gelegentlich den Revisionsstand der ausgelieferten Produkte erhöht, kann durch einen solchen Scan eine Konfiguration erzeugt werden, die (bei identischem Maschinenaufbau) zwar von der Geräteliste her identisch ist, die jeweilige Geräteversion unterscheiden sich aber ggf. von der Erstkonfiguration.

#### Beispiel:

Firma A baut den Prototyp einer späteren Serienmaschine B. Dazu wird der Prototyp aufgebaut, in TwinCAT ein Scan über die IO-Geräte durchgeführt und somit die Erstkonfiguration "B.tsm" erstellt. An einer beliebigen Stelle sitzt dabei die EtherCAT-Klemme EL2521-0025 in der Revision 1018. Diese wird also so in die TwinCAT-Konfiguration eingebaut:

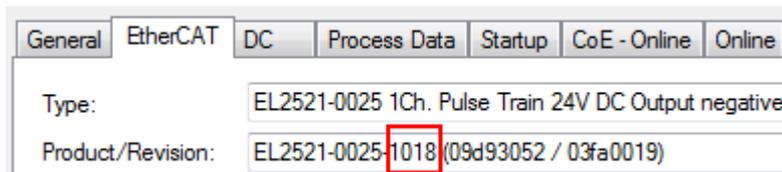


Abb. 115: Einbau EtherCAT-Klemme mit Revision -1018

Ebenso werden in der Prototypentestphase Funktionen und Eigenschaften dieser Klemme durch die Programmierer/Inbetriebnehmer getestet und ggf. genutzt d. h. aus der PLC „B.pro“ oder der NC angesprochen. (sinngemäß gilt das gleiche für die TwinCAT 3-Solution-Dateien).

Nun wird die Prototypenentwicklung abgeschlossen und der Serienbau der Maschine B gestartet, Beckhoff liefert dazu weiterhin die EL2521-0025-0018. Falls die Inbetriebnehmer der Abteilung Serienmaschinenbau immer einen Scan durchführen, entsteht dabei bei jeder Maschine wieder ein eine inhaltsgleiche B-Konfiguration. Ebenso werden eventuell von A weltweit Ersatzteillager für die kommenden Serienmaschinen mit Klemmen EL2521-0025-1018 angelegt.

Nach einiger Zeit erweitert Beckhoff die EL2521-0025 um ein neues Feature C. Deshalb wird die FW geändert, nach außen hin kenntlich durch einen höheren FW-Stand **und eine neue Revision -1019**. Trotzdem unterstützt das neue Gerät natürlich Funktionen und Schnittstellen der Vorgängerversion(en), eine Anpassung von „B.tsm“ oder gar „B.pro“ ist somit nicht nötig. Die Serienmaschinen können weiterhin mit „B.tsm“ und „B.pro“ gebaut werden, zur Kontrolle der aufgebauten Maschine ist ein [vergleichender Scan](#) [► 107] gegen die Erstkonfiguration „B.tsm“ sinnvoll.

Wird nun allerdings in der Abteilung Serienmaschinenbau nicht „B.tsm“ verwendet, sondern wieder ein Scan zur Erstellung der produktiven Konfiguration durchgeführt, wird automatisch die Revision **-1019** erkannt und in die Konfiguration eingebaut:

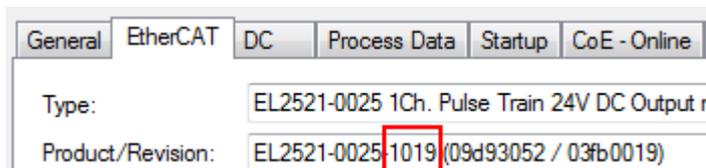


Abb. 116: Erkennen EtherCAT-Klemme mit Revision -1019

Dies wird in der Regel von den Inbetriebnehmern nicht bemerkt. TwinCAT kann ebenfalls nichts melden, da ja quasi eine neue Konfiguration erstellt wird. Es führt nach der Kompatibilitätsregel allerdings dazu, dass in diese Maschine später keine EL2521-0025-**1018** als Ersatzteil eingebaut werden sollen (auch wenn dies in den allermeisten Fällen dennoch funktioniert).

Dazu kommt, dass durch produktionsbegleitende Entwicklung in Firma A das neue Feature C der EL2521-0025-1019 (zum Beispiel ein verbesserter Analogfilter oder ein zusätzliches Prozessdatum zur Diagnose) gerne entdeckt und ohne betriebsinterne Rücksprache genutzt wird. Für die so entstandene neue Konfiguration „B2.tsm“ ist der bisherige Bestand an Ersatzteilgeräten nicht mehr zu verwenden.

Bei etabliertem Serienmaschinenbau sollte der Scan nur noch zu informativen Vergleichszwecken gegen eine definierte Erstkonfiguration durchgeführt werden. Änderungen sind mit Bedacht durchzuführen!

Wurde ein EtherCAT-Device in der Konfiguration angelegt (manuell oder durch Scan), kann das I/O-Feld nach Teilnehmern/Slaves gescannt werden.



Abb. 117: Scan-Abfrage nach dem automatischen Anlegen eines EtherCAT Gerätes (links: TwinCAT 2; rechts TwinCAT 3)

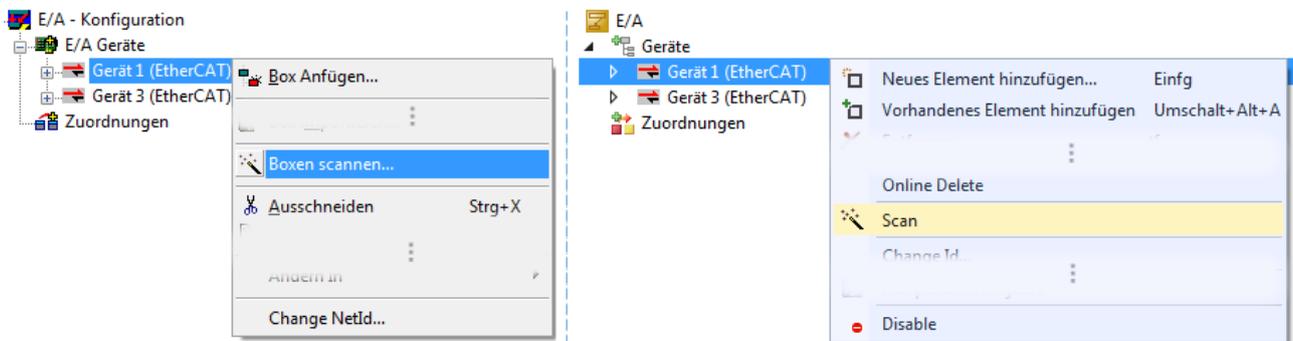


Abb. 118: Manuelles Auslösen des Teilnehmer-Scans auf festgelegtem EtherCAT Device (links: TwinCAT 2; rechts TwinCAT 3)

Im System Manager (TwinCAT 2) bzw. der Benutzeroberfläche (TwinCAT 3) kann der Scan-Ablauf am Ladebalken unten in der Statusleiste verfolgt werden.

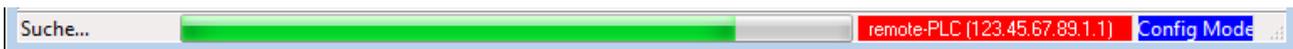


Abb. 119: Scanfortschritt am Beispiel von TwinCAT 2

Die Konfiguration wird aufgebaut und kann danach gleich in den Online-Zustand (OPERATIONAL) versetzt werden.



Abb. 120: Abfrage Config/FreeRun (links: TwinCAT 2; rechts TwinCAT 3)

Im Config/FreeRun-Mode wechselt die System Manager Anzeige blau/rot und das EtherCAT Gerät wird auch ohne aktive Task (NC, PLC) mit der Freilauf-Zykluszeit von 4 ms (Standardeinstellung) betrieben.



Abb. 121: Anzeige des Wechsels zwischen „Free Run“ und „Config Mode“ unten rechts in der Statusleiste



Abb. 122: TwinCAT kann auch durch einen Button in diesen Zustand versetzt werden (links: TwinCAT 2; rechts TwinCAT 3)

Das EtherCAT System sollte sich danach in einem funktionsfähigen zyklischen Betrieb nach Abb. *Beispielhafte Online-Anzeige* befinden.

No	Addr	Name	State	CRC
1	1001	Klemme 1 (EK1100)	OP	0, 0
2	1002	Klemme 2 (EL2008)	OP	0, 0
3	1003	Klemme 3 (EL3751)	SAFEOP	0, 0
4	1004	Klemme 4 (EL2521-0024)	OP	0

Counter	Cyclic	Queued
Send Frames	31713	+ 5645
Frames / sec	500	+ 37
Lost Frames	0	+ 0
Tx/Rx Errors	0	/ 0

Abb. 123: Beispielhafte Online-Anzeige

Zu beachten sind

- alle Slaves sollen im OP-State sein
- der EtherCAT Master soll im „Actual State“ OP sein
- „Frames/sec“ soll der Zykluszeit unter Berücksichtigung der versendeten Frameanzahl sein
- es sollen weder übermäßig „LostFrames“- noch CRC-Fehler auftreten

Die Konfiguration ist nun fertig gestellt. Sie kann auch wie im [manuellen Vorgang \[► 97\]](#) beschrieben verändert werden.

## Problembehandlung

Beim Scannen können verschiedene Effekte auftreten.

- es wird ein **unbekanntes Gerät** entdeckt, d. h. ein EtherCAT Slave für den keine ESI-XML-Beschreibung vorliegt.  
In diesem Fall bietet der System Manager an, die im Gerät eventuell vorliegende ESI auszulesen. Lesen Sie dazu das Kapitel „Hinweise zu ESI/XML“.
- **Teilnehmer werden nicht richtig erkannt**  
Ursachen können sein
  - fehlerhafte Datenverbindungen, es treten Datenverluste während des Scans auf
  - Slave hat ungültige Gerätebeschreibung

Es sind die Verbindungen und Teilnehmer gezielt zu überprüfen, z. B. durch den Emergency Scan.  
Der Scan ist dann erneut vorzunehmen.

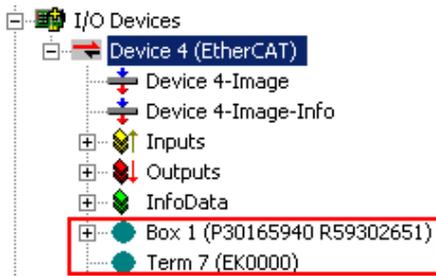


Abb. 124: Fehlerhafte Erkennung

Im System Manager werden solche Geräte evtl. als EK0000 oder unbekannte Geräte angelegt. Ein Betrieb ist nicht möglich bzw. sinnvoll.

**Scan über bestehender Konfiguration**

**HINWEIS**

**Veränderung der Konfiguration nach Vergleich**

Bei diesem Scan werden z. Z. (TwinCAT 2.11 bzw. 3.1) nur die Geräteeigenschaften Vendor (Hersteller), Geräte-Name und Revision verglichen! Ein „ChangeTo“ oder „Copy“ sollte nur im Hinblick auf die Beckhoff IO-Kompatibilitätsregel (s. o.) nur mit Bedacht vorgenommen werden. Das Gerät wird dann in der Konfiguration gegen die vorgefundene Revision ausgetauscht, dies kann Einfluss auf unterstützte Prozessdaten und Funktionen haben.

Wird der Scan bei bestehender Konfiguration angestoßen, kann die reale I/O-Umgebung genau der Konfiguration entsprechen oder differieren. So kann die Konfiguration verglichen werden.



Abb. 125: Identische Konfiguration (links: TwinCAT 2; rechts TwinCAT 3)

Sind Unterschiede feststellbar, werden diese im Korrekturdialog angezeigt, die Konfiguration kann umgehend angepasst werden.

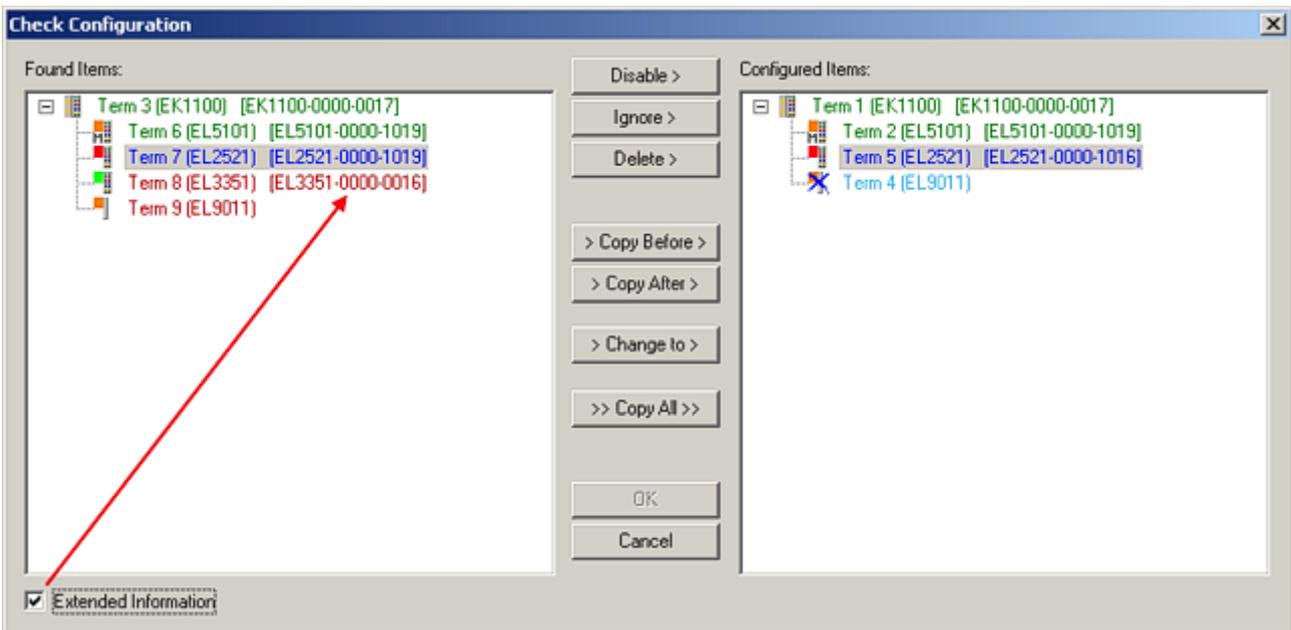


Abb. 126: Korrekturdialog

Die Anzeige der „Extended Information“ wird empfohlen, weil dadurch Unterschiede in der Revision sichtbar werden.

Farbe	Erläuterung
grün	Dieser EtherCAT Slave findet seine Entsprechung auf der Gegenseite. Typ und Revision stimmen überein.
blau	Dieser EtherCAT Slave ist auf der Gegenseite vorhanden, aber in einer anderen Revision. Diese andere Revision kann andere Default-Einstellungen der Prozessdaten und andere/zusätzliche Funktionen haben. Ist die gefundene Revision > als die konfigurierte Revision, ist der Einsatz unter Berücksichtigung der Kompatibilität möglich. Ist die gefundene Revision < als die konfigurierte Revision, ist der Einsatz vermutlich nicht möglich. Eventuell unterstützt das vorgefundene Gerät nicht alle Funktionen, die der Master von ihm aufgrund der höheren Revision erwartet.
hellblau	Dieser EtherCAT Slave wird ignoriert (Button „Ignore“)
rot	<ul style="list-style-type: none"> <li>Dieser EtherCAT Slave ist auf der Gegenseite nicht vorhanden</li> <li>Er ist vorhanden, aber in einer anderen Revision, die sich auch in den Eigenschaften von der angegebenen unterscheidet. Auch hier gilt dann das Kompatibilitätsprinzip: Ist die gefundene Revision &gt; als die konfigurierte Revision, ist der Einsatz unter Berücksichtigung der Kompatibilität möglich, da Nachfolger-Geräte die Funktionen der Vorgänger-Geräte unterstützen sollen.  Ist die gefundene Revision &lt; als die konfigurierte Revision, ist der Einsatz vermutlich nicht möglich. Eventuell unterstützt das vorgefundene Gerät nicht alle Funktionen, die der Master von ihm aufgrund der höheren Revision erwartet.</li> </ul>

**i Geräte-Auswahl nach Revision, Kompatibilität**

Mit der ESI-Beschreibung wird auch das Prozessabbild, die Art der Kommunikation zwischen Master und Slave/Gerät und ggf. Geräte-Funktionen definiert. Damit muss das reale Gerät (Firmware wenn vorhanden) die Kommunikationsanfragen/-einstellungen des Masters unterstützen. Dies ist abwärtskompatibel der Fall, d. h. neuere Geräte (höhere Revision) sollen es auch unterstützen, wenn der EtherCAT Master sie als eine ältere Revision anspricht. Als Beckhoff-Kompatibilitätsregel für EtherCAT-Klemmen/ Boxen/ EJ-Module ist anzunehmen:

**Geräte-Revision in der Anlage >= Geräte-Revision in der Konfiguration**

Dies erlaubt auch den späteren Austausch von Geräten ohne Veränderung der Konfiguration (abweichende Vorgaben bei Antrieben möglich).

**Beispiel**

In der Konfiguration wird eine EL2521-0025-1018 vorgesehen, dann kann real eine EL2521-0025-1018 oder höher (-1019, -1020) eingesetzt werden.

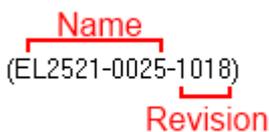


Abb. 127: Name/Revision Klemme

Wenn im TwinCAT System aktuelle ESI-Beschreibungen vorliegen, entspricht der im Auswahldialog als letzte Revision angebotene Stand dem Produktionsstand von Beckhoff. Es wird empfohlen, bei Erstellung einer neuen Konfiguration jeweils diesen letzten Revisionsstand eines Gerätes zu verwenden, wenn aktuell produzierte Beckhoff-Geräte in der realen Applikation verwendet werden. Nur wenn ältere Geräte aus Lagerbeständen in der Applikation verbaut werden sollen, ist es sinnvoll eine ältere Revision einzubinden.

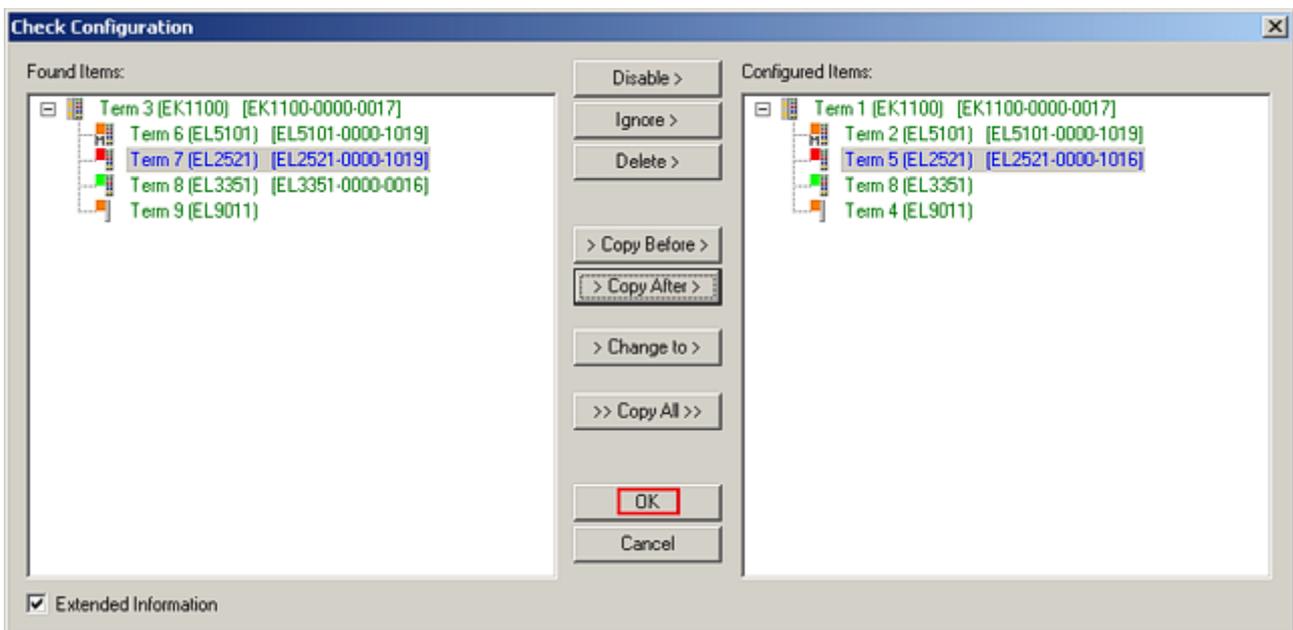


Abb. 128: Korrekturdialog mit Änderungen

Sind alle Änderungen übernommen oder akzeptiert, können sie durch „OK“ in die reale \*.tsm-Konfiguration übernommen werden.

**Change to Compatible Type**

TwinCAT bietet mit „Change to Compatible Type...“ eine Funktion zum Austauschen eines Gerätes unter Beibehaltung der Links in die Task.

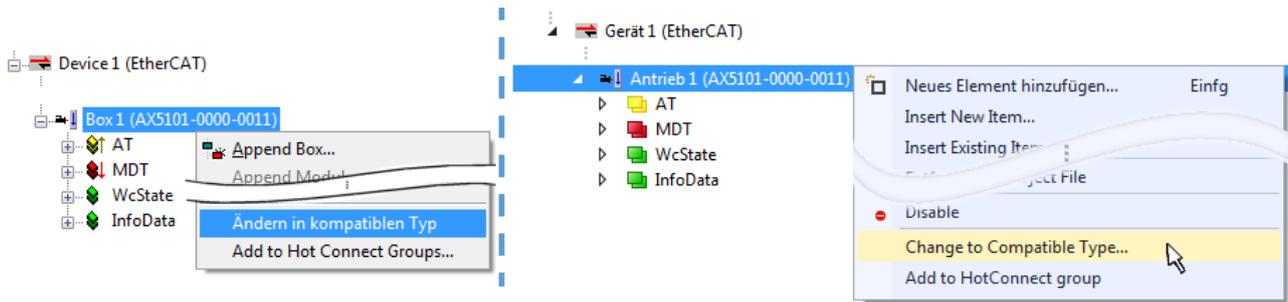


Abb. 129: Dialog „Change to Compatible Type...“ (links: TwinCAT 2; rechts TwinCAT 3)

Diese Funktion ist vorzugsweise auf die AX5000-Geräte anzuwenden.

**Change to Alternative Type**

Der TwinCAT System Manager bietet eine Funktion zum Austauschen eines Gerätes: Change to Alternative Type

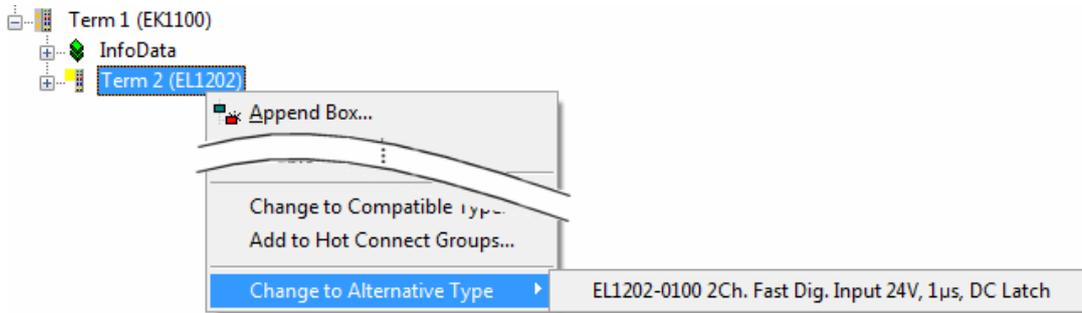


Abb. 130: TwinCAT 2 Dialog Change to Alternative Type

Wenn aufgerufen, sucht der System Manager in der bezogenen Geräte-ESI (hier im Beispiel: EL1202-0000) nach dort enthaltenen Angaben zu kompatiblen Geräten. Die Konfiguration wird geändert und gleichzeitig das ESI-EEPROM überschrieben - deshalb ist dieser Vorgang nur im Online-Zustand (ConfigMode) möglich.

**5.2.7 EtherCAT Teilnehmerkonfiguration**

Klicken Sie im linken Fenster des TwinCAT 2 System Managers bzw. bei der TwinCAT 3 Entwicklungsumgebung im Projektmappen-Explorer auf das Element der Klemme im Baum, die Sie konfigurieren möchten (im Beispiel: Klemme 3: EL3751).

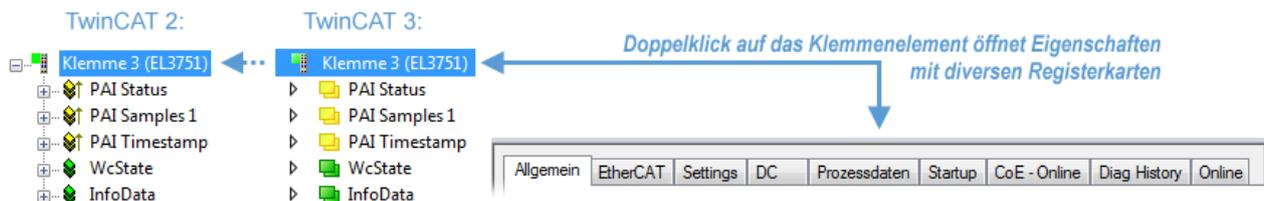


Abb. 131: „Baumzweig“ Element als Klemme EL3751

Im rechten Fenster des System Managers (TwinCAT 2) bzw. der Entwicklungsumgebung (TwinCAT 3) stehen Ihnen nun verschiedene Karteireiter zur Konfiguration der Klemme zur Verfügung. Dabei bestimmt das Maß der Komplexität eines Teilnehmers welche Karteireiter zur Verfügung stehen. So bietet, wie im obigen Beispiel zu sehen, die Klemme EL3751 viele Einstellmöglichkeiten und stellt eine entsprechende Anzahl von Karteireitern zur Verfügung. Im Gegensatz dazu stehen z. B. bei der Klemme EL1004 lediglich die Karteireiter „Allgemein“, „EtherCAT“, „Prozessdaten“ und „Online“ zur Auswahl. Einige Klemmen, wie etwa die EL6695 bieten spezielle Funktionen über einen Karteireiter mit der eigenen Klemmenbezeichnung an, also „EL6695“ in diesem Fall. Ebenfalls wird ein spezieller Karteireiter „Settings“ von Klemmen mit umfangreichen Einstellmöglichkeiten angeboten (z. B. EL3751).

**Karteireiter „Allgemein“**

The screenshot shows the 'Allgemein' tab of a configuration window. At the top, there are several tabs: 'Allgemein', 'EtherCAT', 'Prozessdaten', 'Startup', 'CoE - Online', and 'Online'. The 'Allgemein' tab is active. Below the tabs, there are several input fields: 'Name:' with the value 'Klemme 6 (EL5001)', 'Id:' with the value '6', and 'Typ:' with the value 'EL5001 1K. SSI Encoder'. Below these is a large text area for 'Kommentar:'. At the bottom left, there is a checkbox labeled 'Disabled'. At the bottom right, there is a checkbox labeled 'Symbole erzeugen'.

Abb. 132: Karteireiter „Allgemein“

<b>Name</b>	Name des EtherCAT-Geräts
<b>Id</b>	Laufende Nr. des EtherCAT-Geräts
<b>Typ</b>	Typ des EtherCAT-Geräts
<b>Kommentar</b>	Hier können Sie einen Kommentar (z. B. zum Anlagenteil) hinzufügen.
<b>Disabled</b>	Hier können Sie das EtherCAT-Gerät deaktivieren.
<b>Symbole erzeugen</b>	Nur wenn dieses Kontrollkästchen aktiviert ist, können Sie per ADS auf diesen EtherCAT-Slave zugreifen.

**Karteireiter „EtherCAT“**

The screenshot shows the 'EtherCAT' tab of the same configuration window. The 'Allgemein' tab is now inactive. The 'EtherCAT' tab is active. Below the tabs, there are several input fields: 'Typ:' with the value 'EL5001 1K. SSI Encoder', 'Produkt / Revision:' with the value 'EL5001-0000-0000', 'Auto-Inc-Adresse:' with the value 'FFF8', 'EtherCAT-Adresse:' with a checkbox and the value '1006', and 'Vorgänger-Port:' with the value 'Klemme 5 (EL5001) - B'. To the right of the 'EtherCAT-Adresse:' field is a button labeled 'Weitere Einstellungen...'. At the bottom of the window, there is a URL: <http://www.beckhoff.de/german/default.htm?EtherCAT/EL5001.htm>

Abb. 133: Karteireiter „EtherCAT“

- Typ** Typ des EtherCAT-Geräts
- Product/Revision** Produkt- und Revisions-Nummer des EtherCAT-Geräts
- Auto Inc Adr.** Auto-Inkrement-Adresse des EtherCAT-Geräts. Die Auto-Inkrement-Adresse kann benutzt werden, um jedes EtherCAT-Gerät anhand seiner physikalischen Position im Kommunikationsring zu adressieren. Die Auto-Inkrement-Adressierung wird während der Start-Up-Phase benutzt, wenn der EtherCAT-Master die Adressen an die EtherCAT-Geräte vergibt. Bei der Auto-Inkrement-Adressierung hat der erste EtherCAT-Slave im Ring die Adresse 0000<sub>hex</sub> und für jeden weiteren Folgenden wird die Adresse um 1 verringert (FFFF<sub>hex</sub>, FFFE<sub>hex</sub> usw.).
- EtherCAT Adr.** Feste Adresse eines EtherCAT-Slaves. Diese Adresse wird vom EtherCAT-Master während der Start-Up-Phase vergeben. Um den Default-Wert zu ändern, müssen Sie zuvor das Kontrollkästchen links von dem Eingabefeld markieren.
- Vorgänger Port** Name und Port des EtherCAT-Geräts, an den dieses Gerät angeschlossen ist. Falls es möglich ist, dieses Gerät mit einem anderen zu verbinden, ohne die Reihenfolge der EtherCAT-Geräte im Kommunikationsring zu ändern, dann ist dieses Kombinationsfeld aktiviert und Sie können das EtherCAT-Gerät auswählen, mit dem dieses Gerät verbunden werden soll.
- Weitere Einstellungen** Diese Schaltfläche öffnet die Dialoge für die erweiterten Einstellungen.

Der Link am unteren Rand des Karteireiters führt Sie im Internet auf die Produktseite dieses EtherCAT-Geräts.

**Karteireiter „Prozessdaten“**

Zeigt die (Allgemeine Slave PDO-) Konfiguration der Prozessdaten an. Die Eingangs- und Ausgangsdaten des EtherCAT-Slaves werden als CANopen Prozess-Daten-Objekte (**Process Data Objects, PDO**) dargestellt. Falls der EtherCAT-Slave es unterstützt, ermöglicht dieser Dialog dem Anwender ein PDO über PDO-Zuordnung auszuwählen und den Inhalt des individuellen PDOs zu variieren.

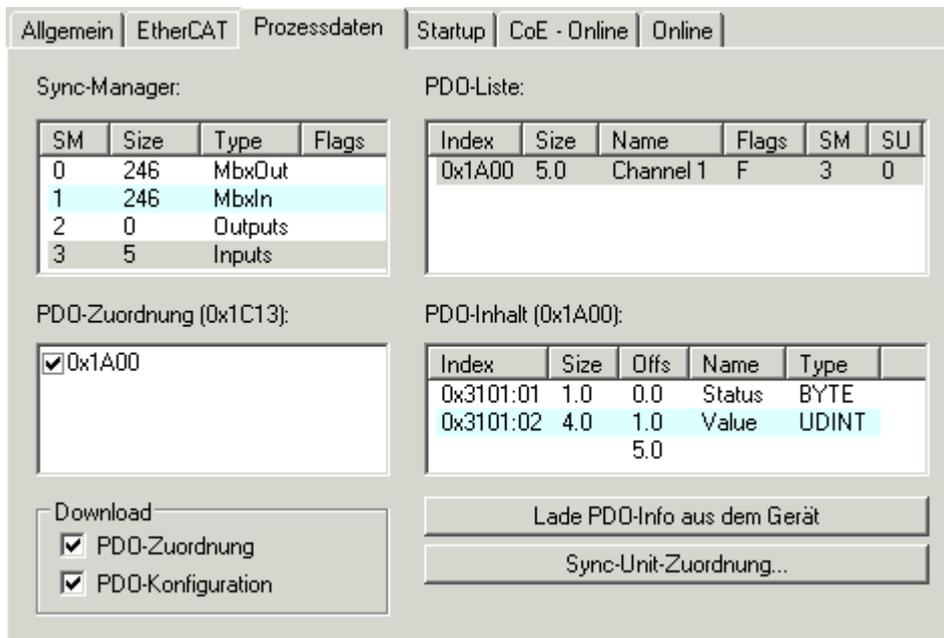


Abb. 134: Karteireiter „Prozessdaten“

Die von einem EtherCAT Slave zyklisch übertragenen Prozessdaten (PDOs) sind die Nutzdaten, die in der Applikation zyklusaktuell erwartet werden oder die an den Slave gesendet werden. Dazu parametriert der EtherCAT Master (Beckhoff TwinCAT) jeden EtherCAT Slave während der Hochlaufphase, um festzulegen, welche Prozessdaten (Größe in Bit/Bytes, Quellort, Übertragungsart) er von oder zu diesem Slave übermitteln möchte. Eine falsche Konfiguration kann einen erfolgreichen Start des Slaves verhindern.

Für Beckhoff EtherCAT Slaves EL, ES, EM, EJ und EP gilt im Allgemeinen:

- Die vom Gerät unterstützten Prozessdaten Input/Output sind in der ESI/XML-Beschreibung herstellerseitig definiert. Der TwinCAT EtherCAT Master verwendet die ESI-Beschreibung zur richtigen Konfiguration des Slaves.
- Wenn vorgesehen, können die Prozessdaten im System Manager verändert werden. Siehe dazu die Gerätedokumentation.  
Solche Veränderungen können sein: Ausblenden eines Kanals, Anzeige von zusätzlichen zyklischen Informationen, Anzeige in 16 Bit statt in 8 Bit Datenumfang usw.
- Die Prozessdateninformationen liegen bei so genannten „intelligenten“ EtherCAT-Geräten ebenfalls im CoE-Verzeichnis vor. Beliebige Veränderungen in diesem CoE-Verzeichnis, die zu abweichenden PDO-Einstellungen führen, verhindern jedoch das erfolgreiche Hochlaufen des Slaves. Es wird davon abgeraten, andere als die vorgesehene Prozessdaten zu konfigurieren, denn die Geräte-Firmware (wenn vorhanden) ist auf diese PDO-Kombinationen abgestimmt.

Ist laut Gerätedokumentation eine Veränderung der Prozessdaten zulässig, kann dies wie folgt vorgenommen werden, s. Abb. Konfigurieren der Prozessdaten.

- A: Wählen Sie das zu konfigurierende Gerät
- B: Wählen Sie im Reiter „Process Data“ den Input- oder Output-Syncmanager (C)
- D: die PDOs können an- bzw. abgewählt werden
- H: die neuen Prozessdaten sind als link-fähige Variablen im System Manager sichtbar  
Nach einem Aktivieren der Konfiguration und TwinCAT-Neustart (bzw. Neustart des EtherCAT Masters) sind die neuen Prozessdaten aktiv.
- E: wenn ein Slave dies unterstützt, können auch Input- und Output-PDO gleichzeitig durch Anwahl eines so genannten PDO-Satzes („Predefined PDO-settings“) verändert werden.

The screenshot shows the 'Process Data' configuration window for an EtherCAT device. The left sidebar shows the device tree with 'Term 2 (EL3162)' selected (A). The 'Sync Manager' table (C) lists SMs 0, 1, 2, and 3 with sizes and types. SM 1 is selected (B). The 'PDO List' table (J) shows indices 0x1A00, 0x1A01, and 0x1A10. A context menu is open over 0x1A01 with 'Edit...' selected (F). The 'PDO Assignment' section (D) shows checkboxes for 0x1A00 and 0x1A01, with 0x1A10 excluded. The 'Download' section (G) has 'PDO Assignment' and 'PDO Configuration' checked. The 'Predefined PDO Assignment' is set to '(none)' (E). The 'PDO Content' table shows indices 0x3101:01 and 0x3101:02. The bottom table (H) shows the resulting process data variables like Status, Value, WcState, State, and AdsAddr.

SM	Size	Type	Flags
0	246	MbxOut	
1	246	MbxIn	
2	0	Outputs	
3	6	Inputs	

Index	Size	Name	Flags	SM	SU
0x1A00	3.0	Channel 1	F		
0x1A01	3.0	Channel 2	F		
0x1A10	4.0	Channels	F		

Index	Assignment
0x1A00	<input checked="" type="checkbox"/>
0x1A01	<input checked="" type="checkbox"/>
0x1A10	<input type="checkbox"/> (excluded by 0x1A01)

Index	Size	Offs	Name
0x3101:01	1.0	0.0	Status
0x3101:02	2.0	1.0	Value
		3.0	

Name	Online	Type	Size	>Addr...	In/Out	User...
Status	0x00 (0)	BYTE	1.0	39.0	Input	0
Value	0x0003 <0.001>	INT	2.0	40.0	Input	0
Status	0x00 (0)	BYTE	1.0	42.0	Input	0
Value	0x0007 <0.002>	INT	2.0	43.0	Input	0
WcState	0	BOOL	0.1	1522.1	Input	0
State	0x0008 (8)	UINT	2.0	1550.0	Input	0
AdsAddr	C0 A8 00 14 05 01 ...	AMSADDRESS	8.0	1552.0	Input	0

Abb. 135: Konfigurieren der Prozessdaten

**Manuelle Veränderung der Prozessdaten**

**I**n der PDO-Übersicht kann laut ESI-Beschreibung ein PDO als „fixed“ mit dem Flag „F“ gekennzeichnet sein (Abb. *Konfigurieren der Prozessdaten*, J). Solche PDOs können prinzipiell nicht in ihrer Zusammenstellung verändert werden, auch wenn TwinCAT den entsprechenden Dialog anbietet („Edit“). Insbesondere können keine beliebigen CoE-Inhalte als zyklische Prozessdaten eingeblendet werden. Dies gilt im Allgemeinen auch für den Fall, dass ein Gerät den Download der PDO Konfiguration „G“ unterstützt. Bei falscher Konfiguration verweigert der EtherCAT Slave üblicherweise den Start und Wechsel in den OP-State. Eine Logger-Meldung wegen „invalid SM cfg“ wird im System Manager ausgegeben: Diese Fehlermeldung „invalid SM IN cfg“ oder „invalid SM OUT cfg“ bietet gleich einen Hinweis auf die Ursache des fehlgeschlagenen Starts.

Eine detaillierte Beschreibung [► 118] befindet sich am Ende dieses Kapitels.

**Karteireiter „Startup“**

Der Karteireiter *Startup* wird angezeigt, wenn der EtherCAT-Slave eine Mailbox hat und das Protokoll *CANopen over EtherCAT* (CoE) oder das Protokoll *Servo drive over EtherCAT* unterstützt. Mit Hilfe dieses Karteireiters können Sie betrachten, welche Download-Requests während des Startups zur Mailbox gesendet werden. Es ist auch möglich neue Mailbox-Requests zur Listenanzeige hinzuzufügen. Die Download-Requests werden in derselben Reihenfolge zum Slave gesendet, wie sie in der Liste angezeigt werden.

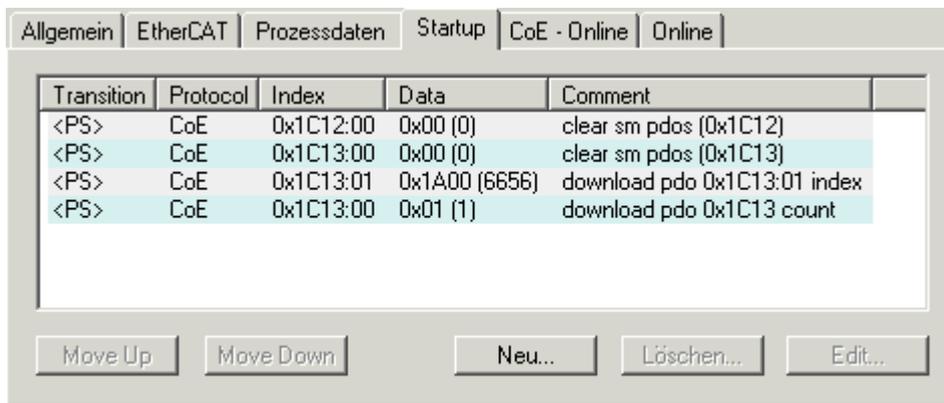


Abb. 136: Karteireiter „Startup“

Spalte	Beschreibung
Transition	Übergang, in den der Request gesendet wird. Dies kann entweder <ul style="list-style-type: none"> <li>• der Übergang von Pre-Operational to Safe-Operational (PS) oder</li> <li>• der Übergang von Safe-Operational to Operational (SO) sein.</li> </ul> Wenn der Übergang in „<>“ eingeschlossen ist (z. B. <PS>), dann ist der Mailbox Request fest und kann vom Anwender nicht geändert oder gelöscht werden.
Protokoll	Art des Mailbox-Protokolls
Index	Index des Objekts
Data	Datum, das zu diesem Objekt heruntergeladen werden soll.
Kommentar	Beschreibung des zu der Mailbox zu sendenden Requests

- Move Up**      Diese Schaltfläche bewegt den markierten Request in der Liste um eine Position nach oben.
- Move Down**      Diese Schaltfläche bewegt den markierten Request in der Liste um eine Position nach unten.
- New**              Diese Schaltfläche fügt einen neuen Mailbox-Download-Request, der während des Startups gesendet werden soll hinzu.
- Delete**            Diese Schaltfläche löscht den markierten Eintrag.
- Edit**                Diese Schaltfläche editiert einen existierenden Request.

**Karteireiter „CoE - Online“**

Wenn der EtherCAT-Slave das Protokoll *CANopen over EtherCAT* (CoE) unterstützt, wird der zusätzliche Karteireiter *CoE - Online* angezeigt. Dieser Dialog listet den Inhalt des Objektverzeichnisses des Slaves auf (SDO-Upload) und erlaubt dem Anwender den Inhalt eines Objekts dieses Verzeichnisses zu ändern. Details zu den Objekten der einzelnen EtherCAT-Geräte finden Sie in den gerätespezifischen Objektbeschreibungen.

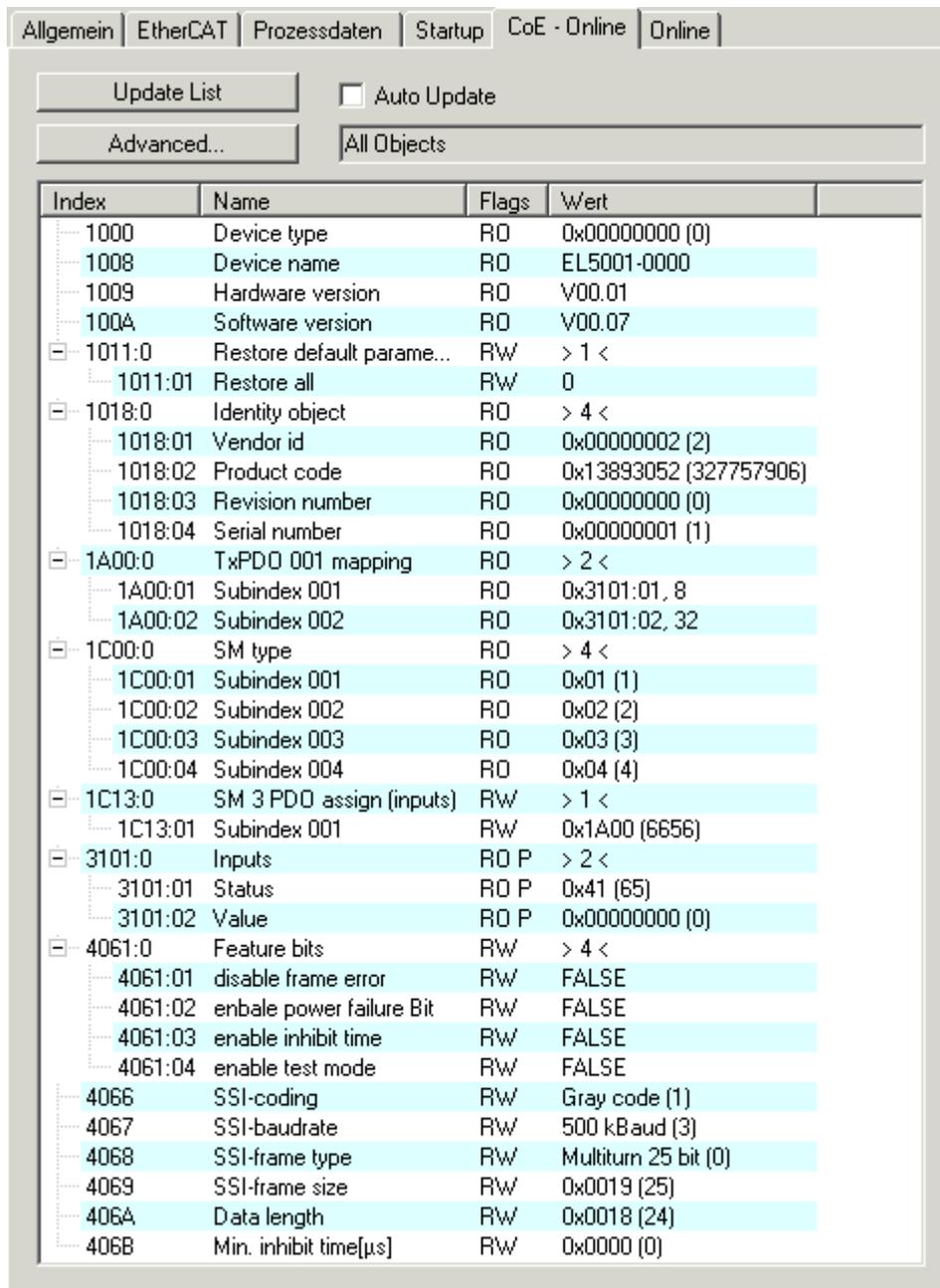


Abb. 137: Karteireiter „CoE - Online“

**Darstellung der Objekt-Liste**

Spalte	Beschreibung	
Index	Index und Subindex des Objekts	
Name	Name des Objekts	
Flags	RW	Das Objekt kann ausgelesen und Daten können in das Objekt geschrieben werden (Read/Write)
	RO	Das Objekt kann ausgelesen werden, es ist aber nicht möglich Daten in das Objekt zu schreiben (Read only)
	P	Ein zusätzliches P kennzeichnet das Objekt als Prozessdatenobjekt.
Wert	Wert des Objekts	

**Update List  
Auto Update**

Die Schaltfläche *Update List* aktualisiert alle Objekte in der Listenanzeige. Wenn dieses Kontrollkästchen angewählt ist, wird der Inhalt der Objekte automatisch aktualisiert.

**Advanced**

Die Schaltfläche *Advanced* öffnet den Dialog *Advanced Settings*. Hier können Sie festlegen, welche Objekte in der Liste angezeigt werden.

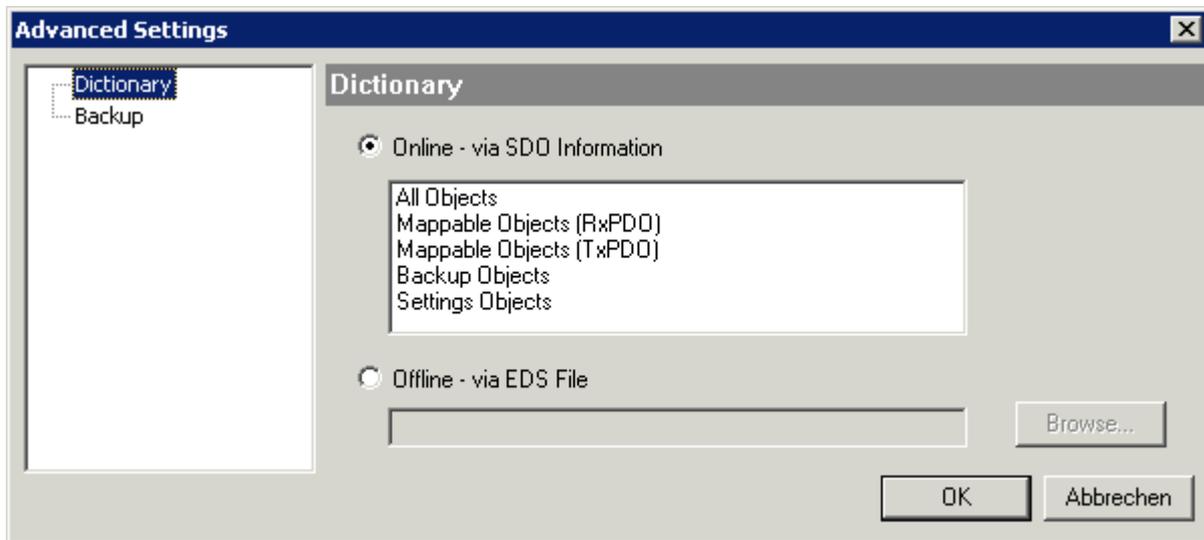


Abb. 138: Dialog „Advanced settings“

**Online - über SDO-  
Information**

Wenn dieses Optionsfeld angewählt ist, wird die Liste der im Objektverzeichnis des Slaves enthaltenen Objekte über SDO-Information aus dem Slave hochgeladen. In der untenstehenden Liste können Sie festlegen welche Objekt-Typen hochgeladen werden sollen.

**Offline - über EDS-Datei**

Wenn dieses Optionsfeld angewählt ist, wird die Liste der im Objektverzeichnis enthaltenen Objekte aus einer EDS-Datei gelesen, die der Anwender bereitstellt.

**Karteireiter „Online“**

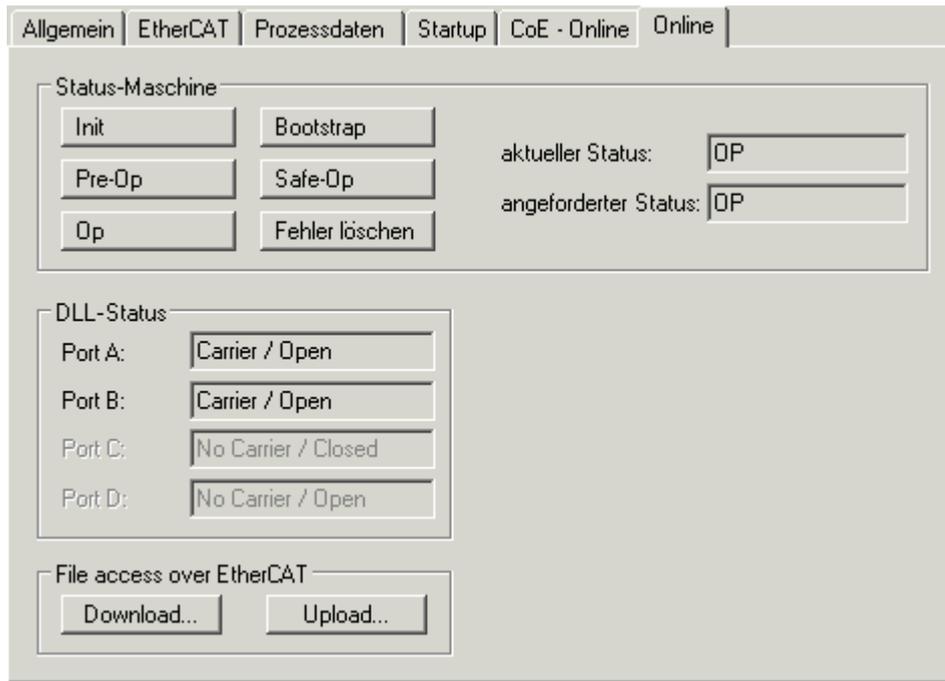


Abb. 139: Karteireiter „Online“

**Status Maschine**

- Init** Diese Schaltfläche versucht das EtherCAT-Gerät auf den Status *Init* zu setzen.
- Pre-Op** Diese Schaltfläche versucht das EtherCAT-Gerät auf den Status *Pre-Operational* zu setzen.
- Op** Diese Schaltfläche versucht das EtherCAT-Gerät auf den Status *Operational* zu setzen.
- Bootstrap** Diese Schaltfläche versucht das EtherCAT-Gerät auf den Status *Bootstrap* zu setzen.
- Safe-Op** Diese Schaltfläche versucht das EtherCAT-Gerät auf den Status *Safe-Operational* zu setzen.
- Fehler löschen** Diese Schaltfläche versucht die Fehleranzeige zu löschen. Wenn ein EtherCAT-Slave beim Statuswechsel versagt, setzt er eine Fehler-Flag.  
Beispiel: ein EtherCAT-Slave ist im Zustand PREOP (Pre-Operational). Nun fordert der Master den Zustand SAFEOP (Safe-Operational) an. Wenn der Slave nun beim Zustandswechsel versagt, setzt er das Fehler-Flag. Der aktuelle Zustand wird nun als ERR PREOP angezeigt. Nach Drücken der Schaltfläche *Fehler löschen* ist das Fehler-Flag gelöscht und der aktuelle Zustand wird wieder als PREOP angezeigt.
- Aktueller Status** Zeigt den aktuellen Status des EtherCAT-Geräts an.
- Angeforderter Status** Zeigt den für das EtherCAT-Gerät angeforderten Status an.

**DLL-Status**

Zeigt den DLL-Status (Data-Link-Layer-Status) der einzelnen Ports des EtherCAT-Slaves an. Der DLL-Status kann vier verschiedene Zustände annehmen:

Status	Beschreibung
No Carrier / Open	Kein Carrier-Signal am Port vorhanden, der Port ist aber offen.
No Carrier / Closed	Kein Carrier-Signal am Port vorhanden und der Port ist geschlossen.
Carrier / Open	Carrier-Signal ist am Port vorhanden und der Port ist offen.
Carrier / Closed	Carrier-Signal ist am Port vorhanden, der Port ist aber geschlossen.

## File Access over EtherCAT

### Download

Mit dieser Schaltfläche können Sie eine Datei zum EtherCAT-Gerät schreiben.

### Upload

Mit dieser Schaltfläche können Sie eine Datei vom EtherCAT-Gerät lesen.

## Karteireiter „DC“ (Distributed Clocks)

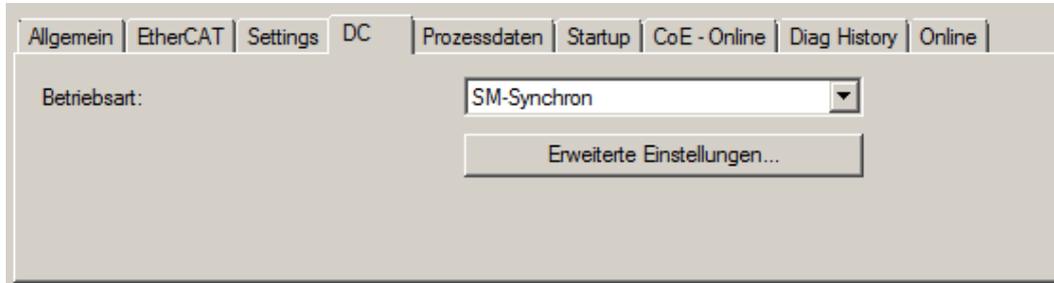


Abb. 140: Karteireiter „DC“ (Distributed Clocks)

### Betriebsart

Auswahlmöglichkeiten (optional):

- FreeRun
- SM-Synchron
- DC-Synchron (Input based)
- DC-Synchron

### Erweiterte Einstellungen...

Erweiterte Einstellungen für die Nachregelung der echtzeitbestimmende TwinCAT-Uhr

Detaillierte Informationen zu Distributed Clocks sind unter <http://infosys.beckhoff.de> angegeben:

**Feldbuskomponenten** → EtherCAT-Klemmen → EtherCAT System Dokumentation → Distributed Clocks

## 5.2.7.1 Detaillierte Beschreibung Karteireiter „Prozessdaten“

### Sync-Manager

Listet die Konfiguration der Sync-Manager (SM) auf.

Wenn das EtherCAT-Gerät eine Mailbox hat, wird der SM0 für den Mailbox-Output (MbxOut) und der SM1 für den Mailbox-Input (MbxIn) benutzt.

Der SM2 wird für die Ausgangsprozessdaten (Outputs) und der SM3 (Inputs) für die Eingangsprozessdaten benutzt.

Wenn ein Eintrag ausgewählt ist, wird die korrespondierende PDO-Zuordnung in der darunter stehenden Liste *PDO-Zuordnung* angezeigt.

### PDO-Zuordnung

PDO-Zuordnung des ausgewählten Sync-Managers. Hier werden alle für diesen Sync-Manager-Typ definierten PDOs aufgelistet:

- Wenn in der Sync-Manager-Liste der Ausgangs-Sync-Manager (Outputs) ausgewählt ist, werden alle RxPDOs angezeigt.
- Wenn in der Sync-Manager-Liste der Eingangs-Sync-Manager (Inputs) ausgewählt ist, werden alle TxPDOs angezeigt.

Die markierten Einträge sind die PDOs, die an der Prozessdatenübertragung teilnehmen. Diese PDOs werden in der Baumdarstellung des System-Managers als Variablen des EtherCAT-Geräts angezeigt. Der Name der Variable ist identisch mit dem Parameter *Name* des PDO, wie er in der PDO-Liste angezeigt wird.

Falls ein Eintrag in der PDO-Zuordnungsliste deaktiviert ist (nicht markiert und ausgegraut), zeigt dies an, dass dieser Eintrag von der PDO-Zuordnung ausgenommen ist. Um ein ausgegrautes PDO auswählen zu können, müssen Sie zuerst das aktuell angewählte PDO abwählen.

**● Aktivierung der PDO-Zuordnung**



- ✓ Wenn Sie die PDO-Zuordnung geändert haben, muss zur Aktivierung der neuen PDO-Zuordnung
  - a) der EtherCAT-Slave einmal den Statusübergang PS (von Pre-Operational zu Safe-Operational) durchlaufen (siehe [Karteireiter Online \[► 117\]](#))
  - b) der System-Manager die EtherCAT-Slaves neu laden

(Schaltfläche  bei TwinCAT 2 bzw.  bei TwinCAT 3)

**PDO-Liste**

Liste aller von diesem EtherCAT-Gerät unterstützten PDOs. Der Inhalt des ausgewählten PDOs wird der Liste *PDO-Content* angezeigt. Durch Doppelklick auf einen Eintrag können Sie die Konfiguration des PDO ändern.

Spalte	Beschreibung	
Index	Index des PDO.	
Size	Größe des PDO in Byte.	
Name	Name des PDO. Wenn dieses PDO einem Sync-Manager zugeordnet ist, erscheint es als Variable des Slaves mit diesem Parameter als Namen.	
Flags	F	Fester Inhalt: Der Inhalt dieses PDO ist fest und kann nicht vom System-Manager geändert werden.
	M	Obligatorisches PDO (Mandatory). Dieses PDO ist zwingend Erforderlich und muss deshalb einem Sync-Manager zugeordnet werden! Als Konsequenz können Sie dieses PDO nicht aus der Liste <i>PDO-Zuordnungen</i> streichen
SM	Sync-Manager, dem dieses PDO zugeordnet ist. Falls dieser Eintrag leer ist, nimmt dieses PDO nicht am Prozessdatenverkehr teil.	
SU	Sync-Unit, der dieses PDO zugeordnet ist.	

**PDO-Inhalt**

Zeigt den Inhalt des PDOs an. Falls das Flag F (fester Inhalt) des PDOs nicht gesetzt ist, können Sie den Inhalt ändern.

**Download**

Falls das Gerät intelligent ist und über eine Mailbox verfügt, können die Konfiguration des PDOs und die PDO-Zuordnungen zum Gerät herunter geladen werden. Dies ist ein optionales Feature, das nicht von allen EtherCAT-Slaves unterstützt wird.

**PDO-Zuordnung**

Falls dieses Kontrollkästchen angewählt ist, wird die PDO-Zuordnung die in der PDO-Zuordnungsliste konfiguriert ist beim Startup zum Gerät herunter geladen. Die notwendigen, zum Gerät zu sendenden Kommandos können in auf dem Karteireiter [Startup \[► 114\]](#) betrachtet werden.

**PDO-Konfiguration**

Falls dieses Kontrollkästchen angewählt ist, wird die Konfiguration des jeweiligen PDOs (wie sie in der PDO-Liste und der Anzeige PDO-Inhalt angezeigt wird) zum EtherCAT-Slave herunter geladen.

## 5.3 Allgemeine Inbetriebnahmehinweise des EtherCAT Slaves

In dieser Übersicht werden in Kurzform einige Aspekte des EtherCAT Slave Betriebs unter TwinCAT behandelt. Ausführliche Informationen dazu sind entsprechenden Fachkapiteln z.B. in der EtherCAT-Systemdokumentation zu entnehmen.

### Diagnose in Echtzeit: WorkingCounter, EtherCAT State und Status

Im Allgemeinen bietet ein EtherCAT Slave mehrere Diagnoseinformationen zur Verarbeitung in der ansteuernden Task an.

Diese Diagnoseinformationen erfassen unterschiedliche Kommunikationsebenen und damit Quellorte und werden deshalb auch unterschiedlich aktualisiert.

Eine Applikation, die auf die Korrektheit und Aktualität von IO-Daten aus einem Feldbus angewiesen ist, muss die entsprechend ihrer unterlagerten Ebenen diagnostisch erfassen.

EtherCAT und der TwinCAT System Manager bieten entsprechend umfassende Diagnoseelemente an. Die Diagnoseelemente, die im laufenden Betrieb (nicht zur Inbetriebnahme) für eine zyklusaktuelle Diagnose aus der steuernden Task hilfreich sind, werden im Folgenden erläutert.

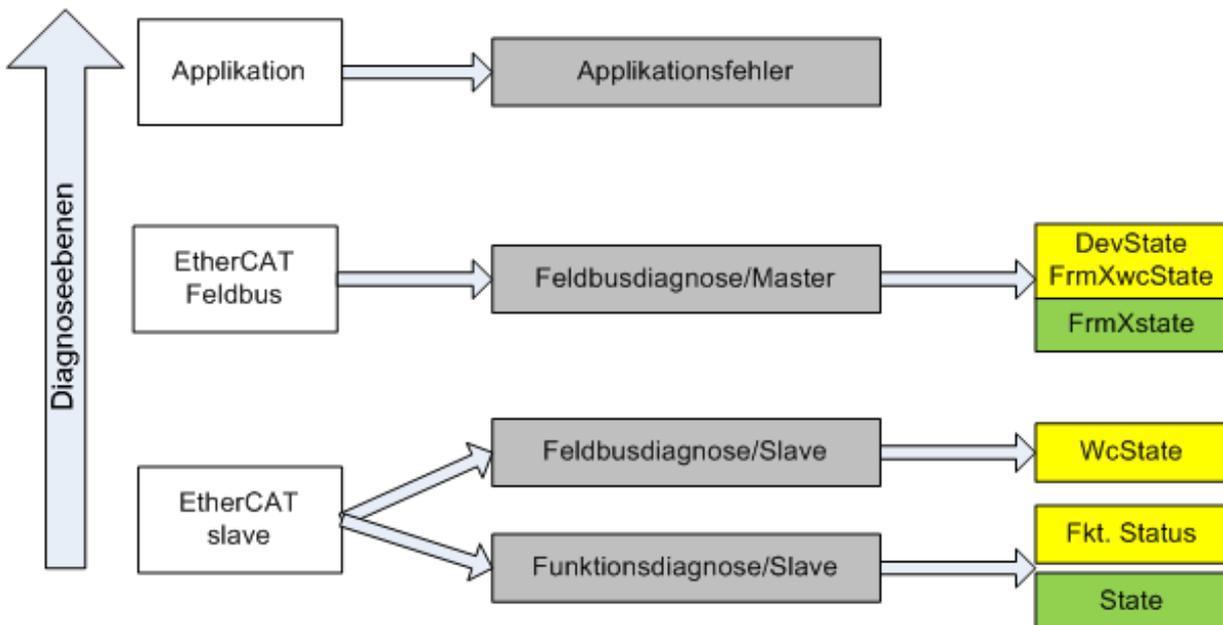


Abb. 141: Auswahl an Diagnoseinformationen eines EtherCAT Slave

Im Allgemeinen verfügt ein EtherCAT Slave über

- slave-typische Kommunikationsdiagnose (Diagnose der erfolgreichen Teilnahme am Prozessdatenaustausch und richtige Betriebsart)  
Diese Diagnose ist für alle Slaves gleich.

als auch über

- kanal-typische Funktionsdiagnose (geräteabhängig)  
Siehe entsprechende Gerätedokumentation

Die Farbgebung in Abb. *Auswahl an Diagnoseinformationen eines EtherCAT Slave* entspricht auch den Variablenfarben im System Manager, siehe Abb. *Grundlegende EtherCAT Slave Diagnose in der PLC*.

Farbe	Bedeutung
gelb	Eingangsvariablen vom Slave zum EtherCAT Master, die in jedem Zyklus aktualisiert werden
rot	Ausgangsvariablen vom Slave zum EtherCAT Master, die in jedem Zyklus aktualisiert werden
grün	Informationsvariablen des EtherCAT Masters, die azyklisch aktualisiert werden d. h. in einem Zyklus eventuell nicht den letztmöglichen Stand abbilden. Deshalb ist ein Auslesen solcher Variablen über ADS sinnvoll.

In Abb. *Grundlegende EtherCAT Slave Diagnose in der PLC* ist eine Beispielimplementation einer grundlegenden EtherCAT Slave Diagnose zu sehen. Dabei wird eine Beckhoff EL3102 (2 kanalige analoge Eingangsklemme) verwendet, da sie sowohl über slave-typische Kommunikationsdiagnose als auch über kanal-spezifische Funktionsdiagnose verfügt. In der PLC sind Strukturen als Eingangsvariablen angelegt, die jeweils dem Prozessabbild entsprechen.

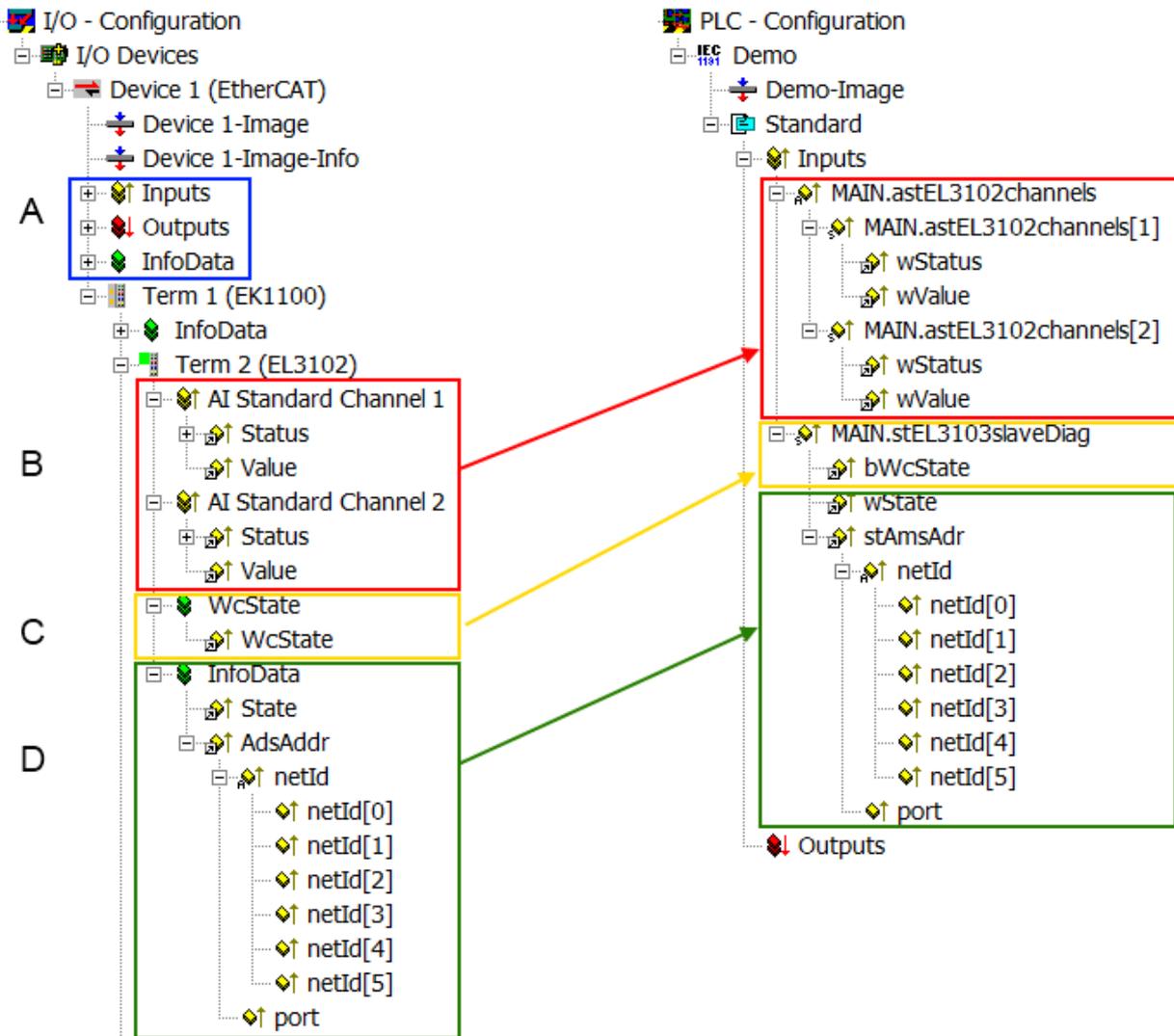


Abb. 142: Grundlegende EtherCAT Slave Diagnose in der PLC

Dabei werden folgende Aspekte abgedeckt:

Kennzeichen	Funktion	Ausprägung	Anwendung/Auswertung
A	Diagnoseinformationen des EtherCAT Master zyklisch aktualisiert (gelb) oder azyklisch bereitgestellt (grün).		Zumindest der DevState ist in der PLC zyklusaktuell auszuwerten. Die Diagnoseinformationen des EtherCAT Master bieten noch weitaus mehr Möglichkeiten, die in der EtherCAT-Systemdokumentation behandelt werden. Einige Stichworte: <ul style="list-style-type: none"> <li>• CoE im Master zur Kommunikation mit/über die Slaves</li> <li>• Funktionen aus <i>TcEtherCAT.lib</i></li> <li>• OnlineScan durchführen</li> </ul>
B	Im gewählten Beispiel (EL3102) umfasst die EL3102 zwei analoge Eingangskanäle, die einen eigenen Funktionsstatus zyklusaktuell übermitteln.	Status <ul style="list-style-type: none"> <li>• die Bitdeutungen sind der Gerätedokumentation zu entnehmen</li> <li>• andere Geräte können mehr oder keine slave-typischen Angaben liefern</li> </ul>	Damit sich die übergeordnete PLC-Task (oder entsprechende Steueranwendungen) auf korrekte Daten verlassen kann, muss dort der Funktionsstatus ausgewertet werden. Deshalb werden solche Informationen zyklusaktuell mit den Prozessdaten bereitgestellt.
C	Für jeden EtherCAT Slave mit zyklischen Prozessdaten zeigt der Master durch einen so genannten Working-Counter an, ob der Slave erfolgreich und störungsfrei am zyklischen Prozessdatenverkehr teilnimmt. Diese elementar wichtige Information wird deshalb im System Manager zyklusaktuell <ol style="list-style-type: none"> <li>1. am EtherCAT Slave als auch inhaltsidentisch</li> <li>2. als Sammelvariable am EtherCAT Master (siehe Punkt A)</li> </ol> zur Verlinkung bereitgestellt.	WcState (Working Counter) 0: gültige Echtzeitkommunikation im letzten Zyklus 1: ungültige Echtzeitkommunikation ggf. Auswirkung auf die Prozessdaten anderer Slaves, die in der gleichen SyncUnit liegen	Damit sich die übergeordnete PLC-Task (oder entsprechende Steueranwendungen) auf korrekte Daten verlassen kann, muss dort der Kommunikationsstatus des EtherCAT Slaves ausgewertet werden. Deshalb werden solche Informationen zyklusaktuell mit den Prozessdaten bereitgestellt.
D	Diagnoseinformationen des EtherCAT Masters, die zwar am Slave zur Verlinkung dargestellt werden, aber tatsächlich vom Master für den jeweiligen Slave ermittelt und dort dargestellt werden. Diese Informationen haben keinen Echtzeit-Charakter weil sie <ul style="list-style-type: none"> <li>• nur selten/nie verändert werden, außer beim Systemstart</li> <li>• selbst auf azyklischem Weg ermittelt werden (z.B. EtherCAT Status)</li> </ul>	State aktueller Status (INIT..OP) des Slaves. Im normalen Betriebszustand muss der Slave im OP (=8) sein. <i>AdsAddr</i> Die ADS-Adresse ist nützlich, um aus der PLC/Task über ADS mit dem EtherCAT Slave zu kommunizieren, z.B. zum Lesen/Schreiben auf das CoE. Die AMS-NetID eines Slaves entspricht der AMS-NetID des EtherCAT Masters, über den <i>port</i> (= EtherCAT Adresse) ist der einzelne Slave ansprechbar.	Informationsvariablen des EtherCAT Masters, die azyklisch aktualisiert werden, d.h. in einem Zyklus eventuell nicht den letztmöglichen Stand abbilden. Deshalb ist ein Auslesen solcher Variablen über ADS möglich.

## HINWEIS

### Diagnoseinformationen

Es wird dringend empfohlen, die angebotenen Diagnoseinformationen auszuwerten um in der Applikation entsprechend reagieren zu können.

### CoE-Parameterverzeichnis

Das CoE-Parameterverzeichnis (CanOpen-over-EtherCAT) dient der Verwaltung von Einstellwerten des jeweiligen Slaves. Bei der Inbetriebnahme eines komplexeren EtherCAT Slaves sind unter Umständen hier Veränderungen vorzunehmen. Zugänglich ist es über den TwinCAT System Manager, s. Abb. *EL3102, CoE-Verzeichnis*:

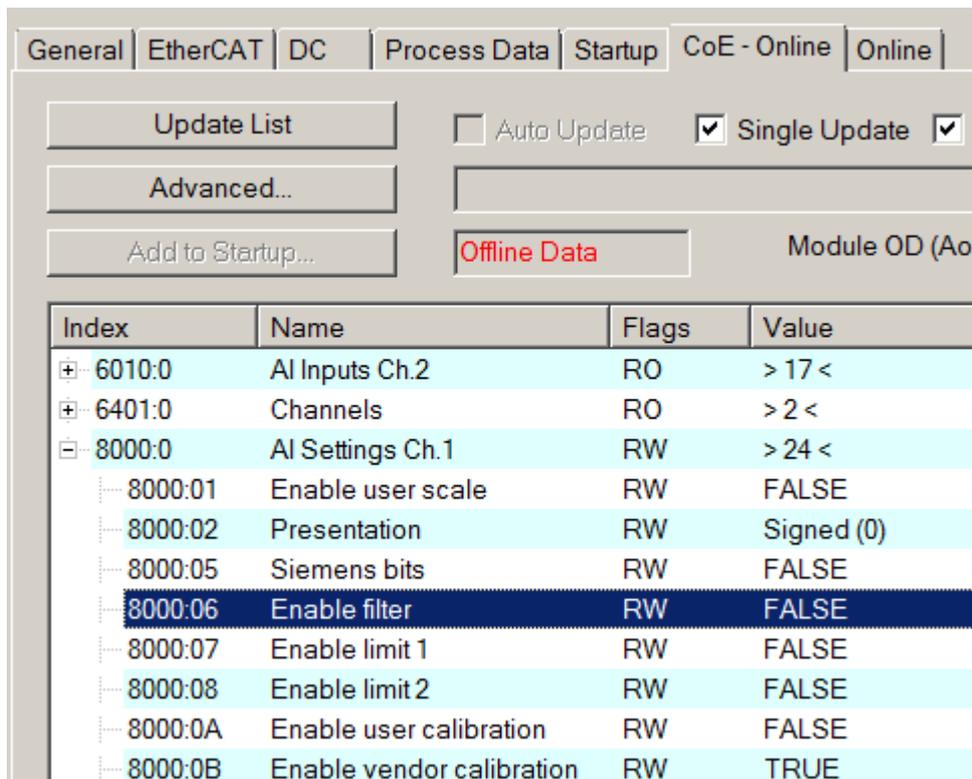


Abb. 143: EL3102, CoE-Verzeichnis

**● EtherCAT-Systemdokumentation**

**i** Es ist die ausführliche Beschreibung in der [EtherCAT-Systemdokumentation](#) (EtherCAT Grundlagen --> CoE Interface) zu beachten!

Einige Hinweise daraus in Kürze:

- Es ist geräteabhängig, ob Veränderungen im Online-Verzeichnis slave-lokal gespeichert werden. EL-Klemmen (außer den EL66xx) verfügen über diese Speichermöglichkeit.
- Es ist vom Anwender die StartUp-Liste mit den Änderungen zu pflegen.

**Inbetriebnahmehilfe im TwinCAT System Manager**

In einem fortschreitenden Prozess werden für EL/EP-EtherCAT Geräte Inbetriebnahmeoberflächen eingeführt. Diese sind in TwinCAT System Managern ab TwinCAT 2.11R2 verfügbar. Sie werden über entsprechend erweiterte ESI-Konfigurationsdateien in den System Manager integriert.

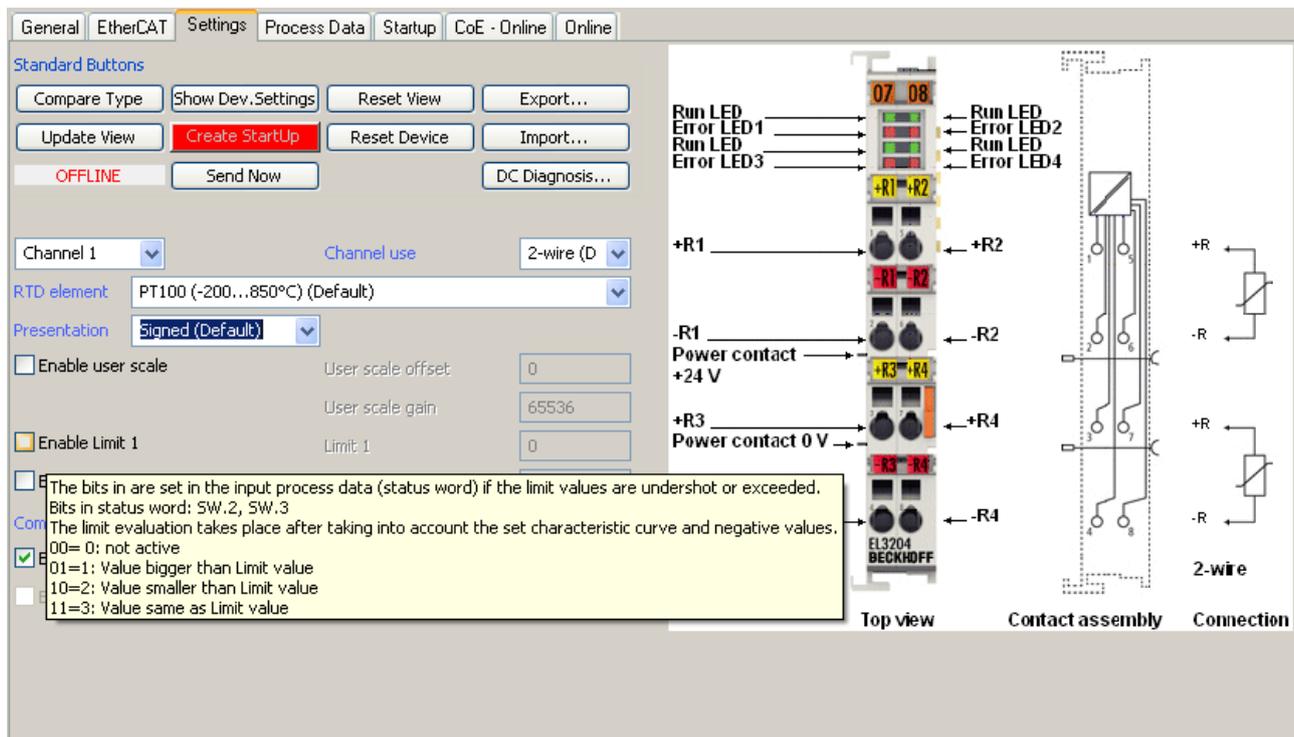


Abb. 144: Beispiel Inbetriebnahmehilfe für eine EL3204

Diese Inbetriebnahme verwaltet zugleich

- CoE-Parameterverzeichnis
- DC/FreeRun-Modus
- die verfügbaren Prozessdatensätze (PDO)

Die dafür bisher nötigen Karteireiter „Process Data“, „DC“, „Startup“ und „CoE-Online“ werden zwar noch angezeigt, es wird aber empfohlen die automatisch generierten Einstellungen durch die Inbetriebnahmehilfe nicht zu verändern, wenn diese verwendet wird.

Das Inbetriebnahme-Tool deckt nicht alle möglichen Einsatzfälle eines EL/EP-Gerätes ab. Sind die Einstellmöglichkeiten nicht ausreichend, können vom Anwender wie bisher DC-, PDO- und CoE-Einstellungen manuell vorgenommen werden.

### EtherCAT State: automatisches Default-Verhalten des TwinCAT System Managers und manuelle Ansteuerung

Ein EtherCAT Slave hat für den ordnungsgemäßen Betrieb nach der Versorgung mit Betriebsspannung die Stati

- INIT
- PREOP
- SAFEOP
- OP

zu durchlaufen. Der EtherCAT Master ordnet diese Zustände an in Abhängigkeit der Initialisierungsroutinen, die zur Inbetriebnahme des Gerätes durch die ES/XML und Anwendereinstellungen (Distributed Clocks (DC), PDO, CoE) definiert sind. Siehe dazu auch Kapitel "Grundlagen der Kommunikation, EtherCAT State Machine [► 28]. Der Hochlauf kann je nach Konfigurationsaufwand und Gesamtkonfiguration bis zu einigen Sekunden dauern.

Auch der EtherCAT Master selbst muss beim Start diese Routinen durchlaufen, bis er in jedem Fall den Zielzustand OP erreicht.

Der vom Anwender beabsichtigte, von TwinCAT beim Start automatisch herbeigeführte Ziel-State kann im System Manager eingestellt werden. Sobald TwinCAT in RUN versetzt wird, wird dann der TwinCAT EtherCAT Master die Zielzustände anfahren.

**Standardeinstellung**

Standardmäßig ist in den erweiterten Einstellungen des EtherCAT Masters gesetzt:

- EtherCAT Master: OP
- Slaves: OP  
Diese Einstellung gilt für alle Slaves zugleich.

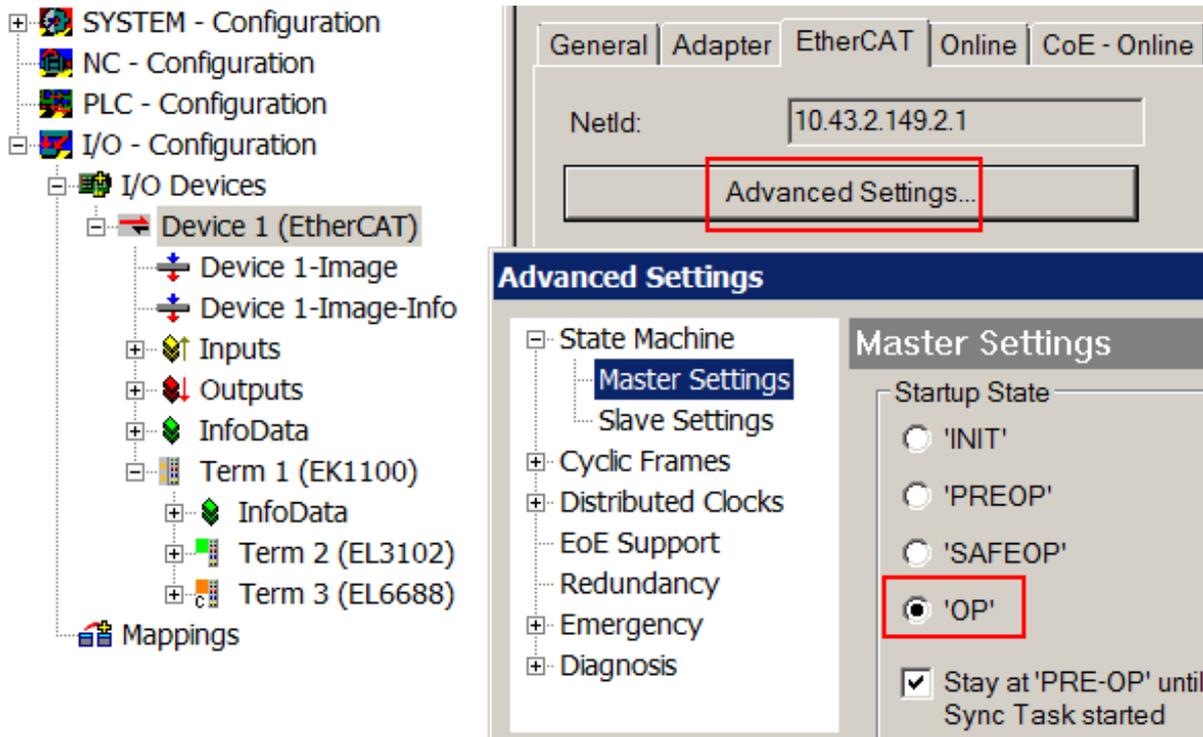


Abb. 145: Default Verhalten System Manager

Zusätzlich kann im Dialog „Erweiterte Einstellung“ beim jeweiligen Slave der Zielzustand eingestellt werden, auch dieser ist standardmäßig OP.

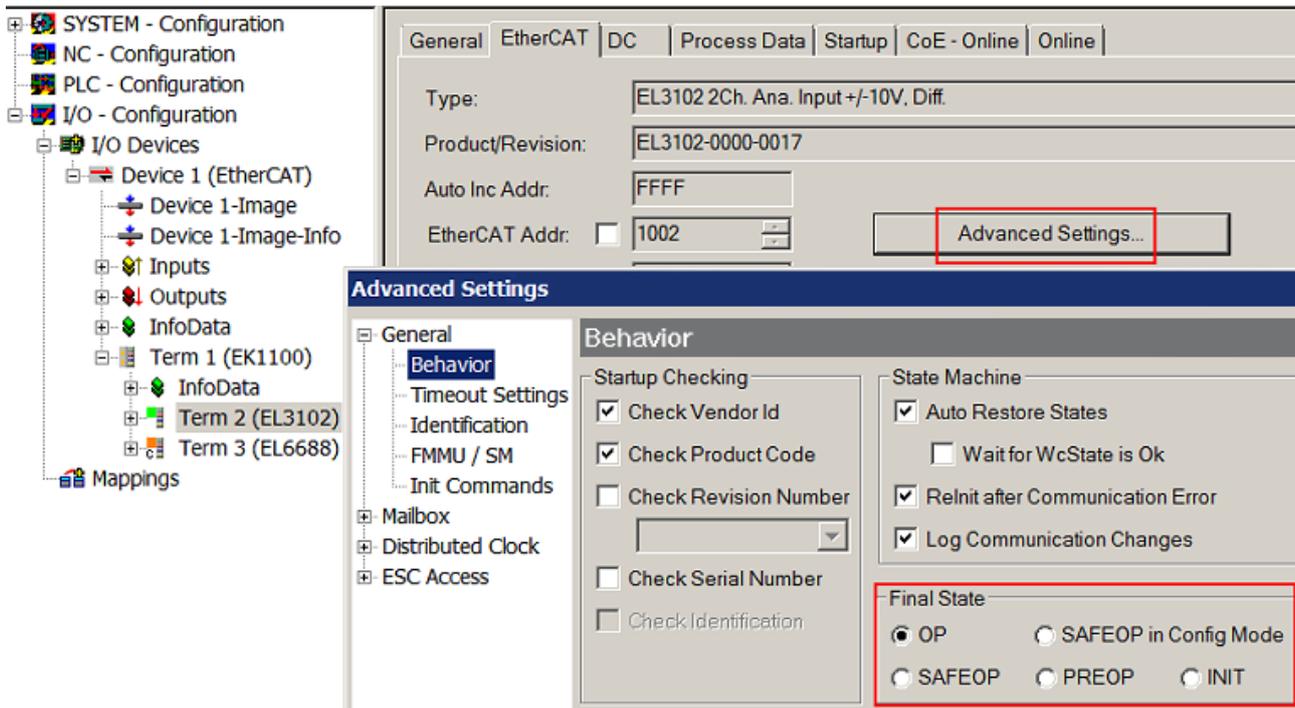


Abb. 146: Default Zielzustand im Slave

**Manuelle Führung**

Aus bestimmten Gründen kann es angebracht sein, aus der Anwendung/Task/PLc die States kontrolliert zu fahren, z. B.

- aus Diagnosegründen
- kontrolliertes Wiederanfahren von Achsen
- ein zeitlich verändertes Startverhalten ist gewünscht

Dann ist es in der PLC-Anwendung sinnvoll, die PLC-Funktionsblöcke aus der standardmäßig vorhandenen *TcEtherCAT.lib* zu nutzen und z. B. mit *FB\_EcSetMasterState* die States kontrolliert anzufahren.

Die Einstellungen im EtherCAT Master sind dann sinnvollerweise für Master und Slave auf INIT zu setzen.

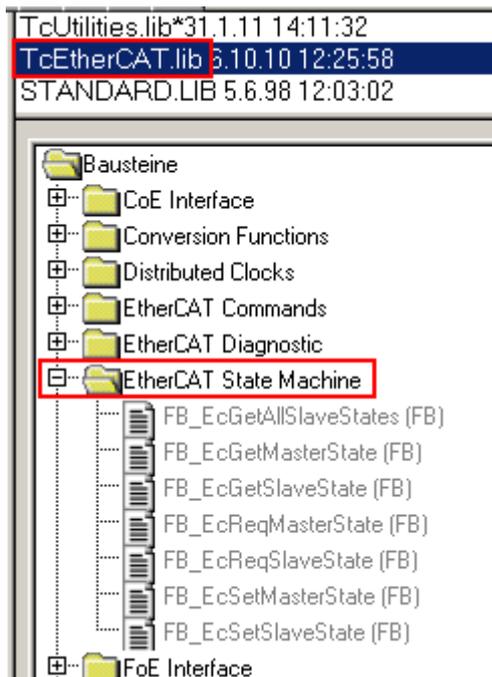


Abb. 147: PLC-Bausteine

**Hinweis E-Bus-Strom**

EL/ES-Klemmen werden im Klemmenstrang auf der Hutschiene an einen Koppler gesetzt. Ein Buskoppler kann die an ihm angefügten EL-Klemmen mit der E-Bus-Systemspannung von 5 V versorgen, i.d.R. ist ein Koppler dabei bis zu 2 A belastbar. Zu jeder EL-Klemme ist die Information, wie viel Strom sie aus der E-Bus-Versorgung benötigt, online und im Katalog verfügbar. Benötigen die angefügten Klemmen mehr Strom als der Koppler liefern kann, sind an entsprechenden Positionen im Klemmenstrang Einspeiseklemmen (z. B. EL9410) zu setzen.

Im TwinCAT System Manager wird der vorberechnete theoretische maximale E-Bus-Strom als Spaltenwert angezeigt. Eine Unterschreitung wird durch negativen Summenbetrag und Ausrufezeichen markiert, vor einer solchen Stelle ist eine Einspeiseklemme zu setzen.

General   Adapter   EtherCAT   Online   CoE - Online						
NetId:		10.43.2.149.2.1		Advanced Settings...		
Number	Box Name	Address	Type	In Size	Out S...	E-Bus (..
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL3102)	1002	EL3102	8.0		1830
3	Term 4 (EL2004)	1003	EL2004		0.4	1730
4	Term 5 (EL2004)	1004	EL2004		0.4	1630
5	Term 6 (EL7031)	1005	EL7031	8.0	8.0	1510
6	Term 7 (EL2808)	1006	EL2808		1.0	1400
7	Term 8 (EL3602)	1007	EL3602	12.0		1210
8	Term 9 (EL3602)	1008	EL3602	12.0		1020
9	Term 10 (EL3602)	1009	EL3602	12.0		830
10	Term 11 (EL3602)	1010	EL3602	12.0		640
11	Term 12 (EL3602)	1011	EL3602	12.0		450
12	Term 13 (EL3602)	1012	EL3602	12.0		260
13	Term 14 (EL3602)	1013	EL3602	12.0		70
14	Term 3 (EL6688)	1014	EL6688	22.0		-240 !

Abb. 148: Unzulässige Überschreitung E-Bus Strom

Ab TwinCAT 2.11 wird bei der Aktivierung einer solchen Konfiguration eine Warnmeldung „E-Bus Power of Terminal...“ im Logger-Fenster ausgegeben:

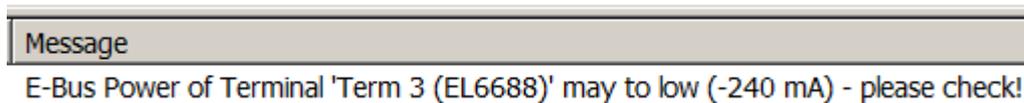


Abb. 149: Warnmeldung E-Bus-Überschreitung

**HINWEIS**

**Achtung! Fehlfunktion möglich!**

Die E-Bus-Versorgung aller EtherCAT-Klemmen eines Klemmenblocks muss aus demselben Massepotential erfolgen!

## 5.4 Betriebsarten und Prozessdaten

### Versionshinweise

Die einkanalige EL60x1 kann auf eine langjährige Modellpflege zurückblicken. Durch Weiterentwicklung der EL6001/EL6021 sind folgende Funktionserweiterungen realisiert worden:

- ab Firmware (FW) 05 / Hardware (HW) 03, (EL6001); Firmware (FW) 04 / Hardware (HW) 03, (EL6021) sind die Objekte zur Statusüberwachung und Parametrierung auch ab Index 0x6000 (Profilspezifische Objekte) verfügbar und in Abhängigkeit der vorliegenden Hardware im TwinCAT System Manager parametrierbar..
- ab Firmware (FW) 06 / Hardware (HW) 03, (EL6021) wird der Kommando-Mode [► 133] unterstützt
- ab Firmware 08 (EL6001) wird das 16-Bit-Prozessdateninterface zum Senden/Empfangen von >8 Bit unterstützt
- ab Firmware 11 (EL6001) sind alle ganzzahligen Baudraten 1000...115200 Baud nutzbar.

Eine EL6001/EL6021 in jeweils älterer Ausführung unterstützt diese Funktionalität nicht!

ab FW/HW	ESI	Bereich Control-/Status-/ Parametrierobjekte
EL6001 01/01 EL6021 01/01	ab EL6001-0000-0000 ab EL6021-0000-0000	Index 0x300n:01 (Control-Word) Index 0x310n:01 (Status-Word) Index 0x4070 (Data bytes in send buffer) Index 0x4071 (Data bytes in receive buffer) Index 0x4072 (Diagnosis) Index 0x4073 (Baudrate) Index 0x4074 (Data frame) Index 0x4075 (Feature bits)
EL6001 05/03 EL6021 04/03	ab EL6001-0000-0016 ab EL6021-0000-0016	zusätzlich zu den oben beschriebenen Objekten: Index 0x6000 (COM inputs) Index 0x7000 (COM outputs) Index 0xA000 (COM Diag data) Index 0x8000 (COM settings)
EL6021 06/03	ab EL6021-0000-0018	zusätzlich zu den oben beschriebenen Objekten: Index 0xB000 (Command)
EL6001 08/03	ab EL6001-0000-0019	zusätzlich zu den oben beschriebenen Objekten: Objekte für 16-Bit-PDO
EL6001 11/11	Ab EL6001-0000-0020	Alle Baudraten 1000 ... 115200 über explicitBaudrate implementiert

Die EL6002/EL6022 verfügen bereits ab Erst-Release mit Revision -0016 die Objekte im 6000er, 7000er und 8000er Bereich.

### ● Kompatibilität im Servicefall



Beispiel: Eine ab Hardware 03 eingesetzte und projektierte EL6001/EL6021 kann nicht durch eine EL6001/EL6021 mit älterem Hardwarestand (< 03) ausgetauscht werden! Der umgekehrte Fall ist problemlos realisierbar!

**i Überwachung der Prozessdaten**

- **WcState**: wenn ≠ 0, dann nimmt dieser EtherCAT Teilnehmer nicht am Prozessdatenverkehr teil
- **State**: wenn ≠ 8, dann ist der EtherCAT Teilnehmer nicht im OP (Operational) Status
- **TxPDO State, SyncError**: wenn ≠ 0, dann liegen keine gültigen Prozessdaten vor, z. B. durch Drahtbruch
- **TxPDO Toggle**: wenn dieses Bit toggelt, liegt ein neuer Satz Prozessdaten vor

**StartUp-Einträge (Hardwarestand < 03)**

**i StartUp-Listeneintrag**

Bei der EL6001/EL6021 mit Hardwarestand < 03 können die StartUp-Einträge nur im Übergang SafeOP -> OP gesetzt werden (S -> O). Die Default Einstellung ist PreOP -> SafeOP (P -> S). Stellen sie bei der Erstellung von StartUp-Einträgen sicher, dass die Checkbox "S -> O" angehakt ist (siehe Abb.)!

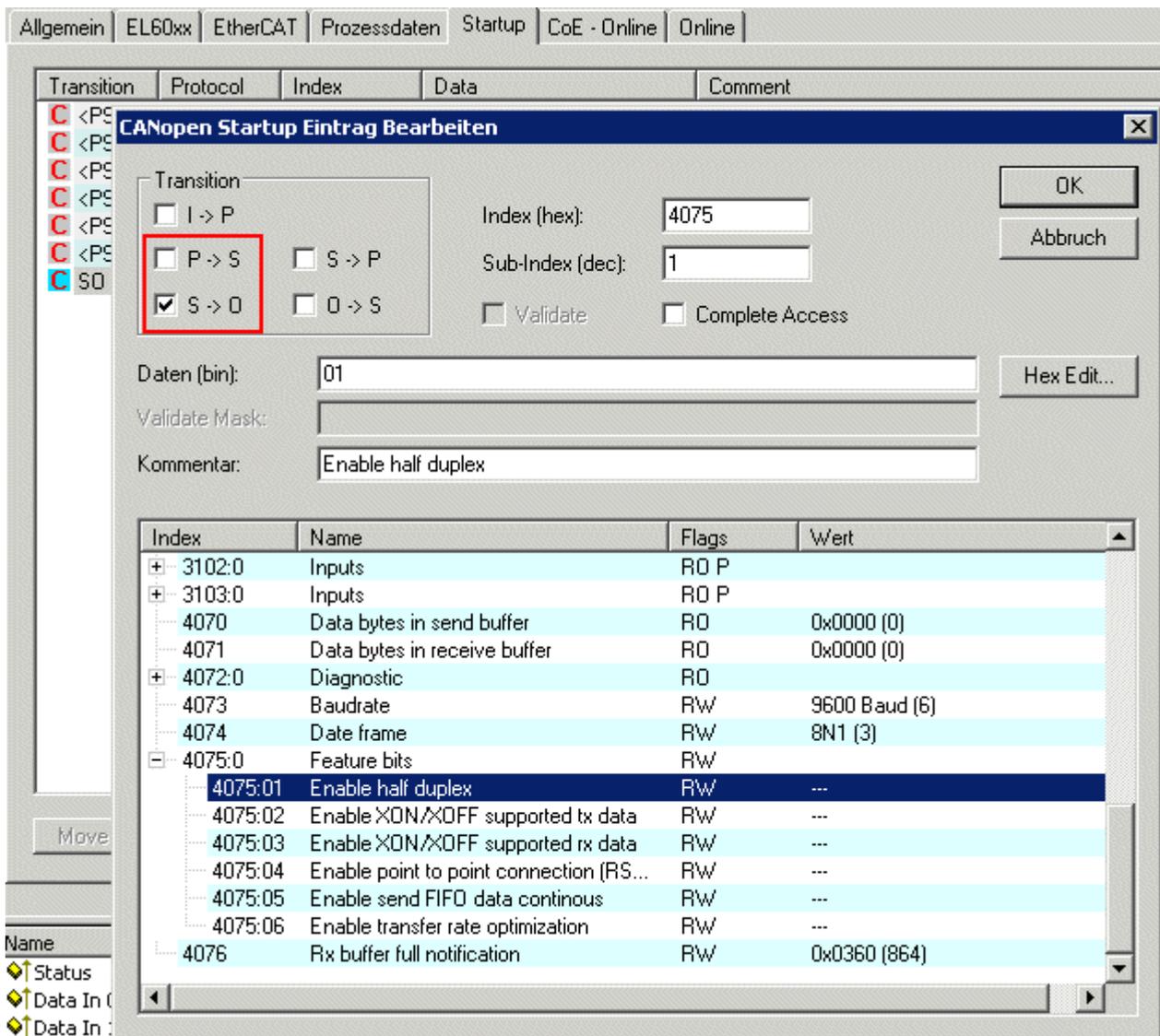


Abb. 150: StartUp-Eintrag mit Übergang S -> O

**Prozessdaten**

Im Auslieferungszustand werden 22 Byte Nutzdaten und 1 Control/Status-Wort übertragen.

Die Prozessdaten werden aus den CoE-Objekten 0x6000 (Inputs) und 0x7000 (Outputs) generiert und sind im Kapitel Objektbeschreibung und Parametrierung [▶ 177] im Einzelnen dargestellt.

**Vergrößertes Prozessabbild 50x 16-Bit**

Für die Kommunikation mit >8 Datenbits ist ein vergrößertes Prozessdateninterface nötig. Falls die Klemme dies unterstützt (siehe [Kommunikationseigenschaften \[► 137\]](#)) kann alternativ zum 22-Byte-Interface ein 50-Word-Interface eingestellt werden (PDOs 0x1605 und 0x1A05). Dies kann mit jeder Codierungsvorschrift (7xx, 8xx) benutzt werden, macht aber nur mit einer Vorschrift >8 bits Sinn, z. B. 9N1. In jedem Fall sind die unteren Bits mit den Datenbits zu belegen, bei 9N1 also die unteren 9 Bits in den Datenwörtern.

**Prozessdaten der EL60x2 ab Revision -0016**

EL60x2 ab Revision -0016

**Sync Manager (SM) - PDO-Zuordnung**

SM2, PDO-Zuordnung 0x1C12				
Index	Index aus- geschlosse- ner PDOs	Größe (Byte.Bit)	Name	PDO Inhalt
0x1600	-	24.0	COM Outputs Channel1	Index 0x7000:01 - Ctrl__Transmit request
				Index 0x7000:02 - Ctrl__Receive accepted
				Index 0x7000:03 - Ctrl__Init request
				Index 0x7000:04 - Ctrl__Send continous
				Index 0x7000:09 - Ctrl__Output length
				Index 0x7000:11 - Data out 0
				--
0x1601	-	24.0	COM Outputs Channel2	Index 0x7010:01
				Index 0x7010:02
				Index 0x7010:03
				Index 0x7010:04
				Index 0x7010:09
				Index 0x7010:11
				--
0x1604 (default)	-	24.0	COM Outputs Channel1	Index 0x7001:01 - Ctrl
				Index 0x7000:11 - Data out 0
				--
				Index 0x7000:26
0x1605 (default)	-	24.0	COM Outputs Channel2	Index 0x7011:01 - Ctrl
				Index 0x7010:11 - Data out 0
				--
				Index 0x7010:26

SM2, PDO-Zuordnung 0x1C13				
Index	Index aus- geschlosse- ner PDOs	Größe (Byte.Bit)	Name	PDO Inhalt
0x1A00	-	24.0	COM Inputs Channel1	Index 0x6000:01 - Status__Transmit accepted
				Index 0x6000:02 - Status__Receive request
				Index 0x6000:03 - Status__Init accepted
				Index 0x6000:04 - Status__Buffer full
				Index 0x6000:05 - Status__Input length
				Index 0x6000:06 - Status__Framing error
				Index 0x6000:07 - Status__Overrun error
				Index 0x6000:09 - Status__Input length
				Index 0x6000:11 - Data in 0
				--
				Index 0x6000:26
0x1A01	-	24.0	COM Inputs Channel2	Index 0x6010:01
				Index 0x6010:02
				Index 0x6010:03
				Index 0x6010:04 - Status__Buffer full
				Index 0x6010:05
				Index 0x6010:06
				Index 0x6010:07
				Index 0x6010:09
				Index 0x6010:11
				--
				Index 0x6010:26
0x1A04(defa ult)	-	24.0	COM Inputs Channel1	Index 0x6001:01
				Index 0x6000:11
				--
				Index 0x6000:26
0x1A05(defa ult)	-	24.0	COM Inputs Channel2	Index 0x6011:01
				Index 0x6010:11
				--
				Index 0x6010:26

**Features und Anwendungshinweise**

Fallweise vorhandene Features werden im Allgemeinen im CoE-Verzeichnis (CAN over EtherCAT, Index 0x80n0) eingestellt.

## ● Parametrierung über das CoE-Verzeichnis (CAN over EtherCAT)

**i** Beachten Sie bei Verwendung/Manipulation der CoE-Parameter die allgemeinen CoE-Hinweise: - StartUp-Liste führen für den Austauschfall - Unterscheidung zwischen Online/Offline Dictionary, Vorhandensein aktueller XML-Beschreibung - "CoE-Reload" zum Zurücksetzen der Veränderungen. Im Besonderen ist bei den EL60xx zu beachten, dass nach Veränderung von Kommunikationseigenschaften (Baudrate, Dataframe, Feature Bits) ein InitRequest über das Control Wort erforderlich ist, um die Änderungen zu übernehmen.

Folgende CoE-Einstellungen aus dem Objekt 0x8000 sind möglich und hier in den Default-Einstellungen wiedergegeben:

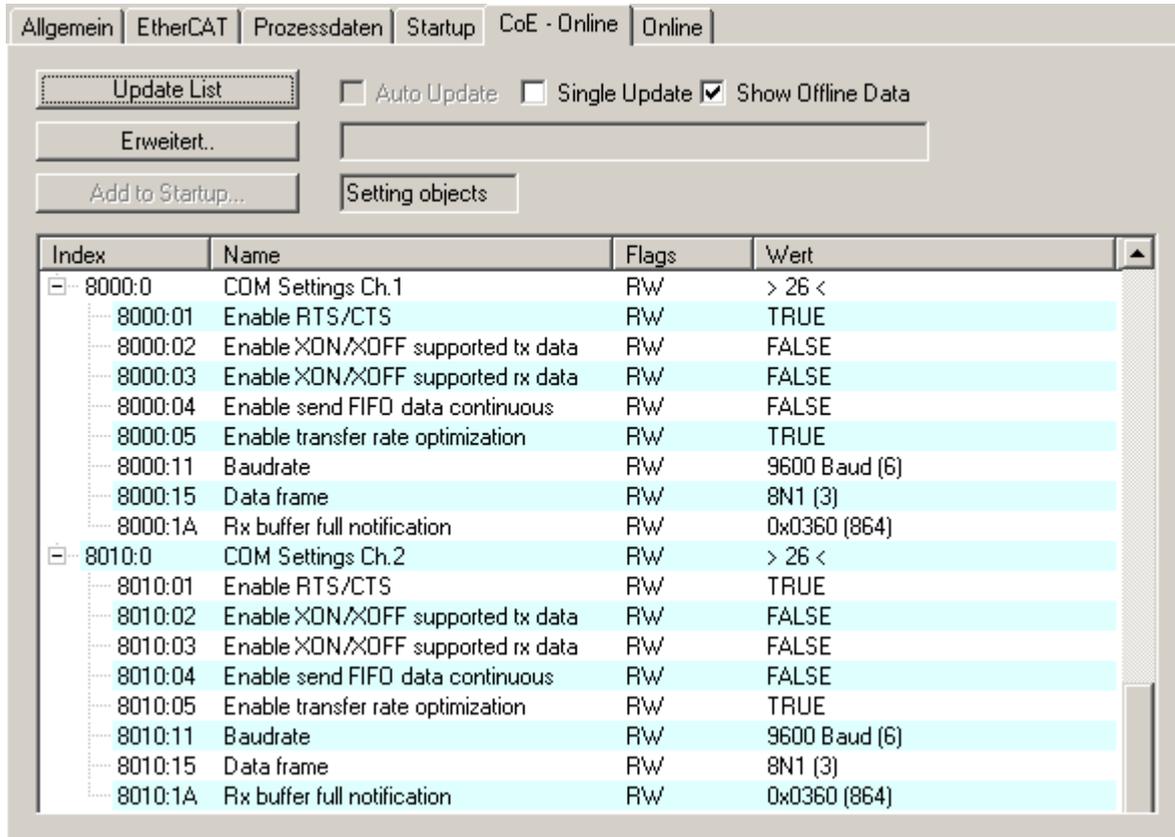


Abb. 151: Reiter "CoE - Online, Klemmen EL60x2"

### Transferraten

Die Klemme verfügt über ein Prozessabbild von 22 Byte Nutzdaten. Es ist maximal jeden zweiten Zyklus möglich, diese 22 Byte zu versenden oder zu empfangen.

Im ersten Zyklus werden die Daten von der Klemme an die Steuerung übertragen. Im zweiten Zyklus muss die Steuerung quittieren, dass sie die Daten übernommen hat.

Bei einer Zykluszeit von 10 ms lassen sich also pro Sekunde 50 mal 22 Byte übertragen.

Bei einem eingestellten Data frame von 8N1 setzt sich jedes gesendete Byte wie aus einem Startbit, acht Datenbits und einem Stoppbit zusammen. Dies entspricht 10 Bit pro Nutzbyte.

Mit den oben erwähnten Einstellungen lässt sich demnach eine kontinuierliche Datentransferrate von:

- $50[1/s] \cdot 22[\text{Byte}] \cdot 10[\text{Bit}] = 11000 \text{ bps}$  erzielen.

Die nächst niedrigere Baudrate ist 9600 Baud. Bei einer Zykluszeit von 10 ms lässt sich demnach eine kontinuierliche Übertragung mit maximal 9600 Baud sicherstellen.

Ab FW11 werden durch CoE 0x8000:1B *explicitBaudrate* alle Baudraten von 1000...115200 Baud unterstützt. Bei der Wahl der Baudrate ist die Kompatibilität der Gegenstelle zur eingestellten Baudrate zu berücksichtigen.

Sollten nur sporadisch geringe Datenmengen gesendet oder empfangen werden (z. B. Barcodescanner) kann die Baudrate auch größer eingestellt werden, bzw. die Zykluszeit vergrößert werden. Falls die Steuerung die Daten nicht schnell genug von der Klemme abholen kann, werden dies im internen Puffer der Klemme zwischengespeichert. Der Puffer für Empfangsdaten hat eine Größe von 864 Byte. Sollte dieser erschöpft sein gehen alle weiteren Daten verloren.

Ein weiteres Szenario wäre, dass die Steuerung deutlich mehr Daten zur Klemme überträgt, als diese Senden kann. Bei einer Eingestellten "Baudrate" von 300 und einem "Data Frame" von 8N1 kann die Klemme nur 30 Byte pro Sekunde senden. Sollten jedoch mehr als dies 30 Byte pro Sekunde eingehen wird auch hier zuerst ein 128 Byte großer Sendepuffer beschrieben. Nachdem dieser gefüllt ist gehen alle weiteren Daten verloren.

### Transferraten Optimierung

Im normalen Betrieb werden empfangene Daten sofort in das Prozessabbild übernommen. Um einen zusammenhängenden Datenstrom zu ermöglichen ist die Option "Enable transfer rate optimization" im Settings-Objekt standardmäßig aktiviert. Durch diesen Schalter werden die Daten zuerst im Empfangspuffer (864 Byte) zwischengespeichert.

Die Daten werden erst in das Prozessabbild kopiert, wenn 16 Bitzeiten lang kein weiteres Zeichen empfangen wurde oder der Puffer voll ist.

### Kontinuierliches Senden von Daten

Normalerweise entscheidet die Klemme EL60xx eigenständig, wann sie die im Puffer befindlichen Datenbytes versendet. Für viele Anwendungen ist ein kontinuierlicher Datenstrom hilfreich. Zu diesem Zweck verfügen die Beckhoff EL60xx Klemmen über die Einstellung "Enable send FIFO data continuous" im Settings-Objekt. Wird dieser Schalter gesetzt

- muss zuerst der interne Sendepuffer (128 Byte) gefüllt werden. Dazu werden die Daten, wie bei einer normalen Übertragung, von der Steuerung an die Klemme gesendet
- Mit einer steigenden Flanke des Bits "Send continuous" wird begonnen die Daten aus dem Puffer zu senden
- Sind die Daten übertragen, so wird dies durch das Setzen des Bits "InitAccepted" von der Klemme an die Steuerung quittiert. "Init accepted" wird mit "Send continuous" zurückgenommen.

Durch diese Einstellung ist es möglich, auch bei langsamen EtherCAT-Zykluszeiten und gleichzeitig hoher Baudrate bis zu 128 Byte ohne lange Wartezeiten zu übertragen.

Die Klemme versucht dabei, so wenig Abstand wie möglich zwischen die Telegramme zu lassen und auf ein Stopbit sofort wieder nahtlos das nächste Startbit anzuschließen. Trotzdem kann es gerade bei hohen Baudraten und 2-kanaligem Betrieb zu kurzen Pausen zwischen Telegrammen kommen. Ein 20-Byte-Datenblock kann so beispielsweise in 2 Blöcke zu 5 und 15 oder 7 und 13 Telegramme oder anderer Aufteilung gesendet werden. Kanal 1 ist bei 2-kanaliger Klemme priorisiert.

### Priorisierung

Da Empfangsdaten i. d. R. nicht wiederholt werden können, haben diese einer höheren Priorität als zu sendende Daten.

Weiterhin sinkt auch hier die Priorität mit steigender Kanalnummer. Die höchste Priorität hat somit der Empfang von Daten auf Kanal 1.

### Kommando-Modus

Die EL6021 unterstützt ab Firmware 06 / Revision -0018 den so genannten Kommando-Modus. Durch eine wahlfreie Kombination bzw. Abfolge von Kommandos können bestimmte Funktionen der Klemme genutzt bzw. gesteuert werden. Aktuell werden dabei folgende Funktionen unterstützt:

- Multi-Data-Frame-Feature: (ab Firmware 06) Wechsel der Kodierung während einer laufenden Datenübertragung beim Senden  
Dazu ist der Sende-Buffer der Klemme mit den zu sendenden Bytes zu füllen (die max. Buffergröße ist hierbei zu beachten!). Sobald das Senden über das Control-Word gestartet wird, werden die ersten n Bytes nach Codierung A, die restlichen Bytes im Buffer mit Codierung B versendet. Dann kann der

Buffer wieder neu befüllt und ebenso versendet werden. Beispiel: das erste Byte wird mit Mark-Parity gesendet, die übrigen Bytes mit Space-Parity.

Ablauf:

- Nach jedem Hochlauf/Neustart des EtherCAT Slaves EL60xx muss diese Parametrierung erneut vorgenommen werden; die Kommandofunktion ist nicht stromausfallsicher gespeichert.
  - Aktivierung des Modus SendContinuous durch Schreiben von 0x1 nach 0x8000:04
  - Aktivierung des Multi-Data-Frame-Feature durch Schreiben von 0x2001 nach Request 0xB000:01  
Kontrolle, dass 0xB000:02 = 0
  - Codierung A vorgeben durch Schreiben von 0x2100 + [data frame code] nach 0xB000:01  
Beispiel: 8E2 = x12 --> Wert x2112 zu schreiben  
Kontrolle, dass 0xB000:02 = 0
  - Anzahl Bytes n vorgeben, die in Codierung A zu senden sind durch Schreiben von 0x2200 + [n] nach 0xB000:01  
Kontrolle, dass 0xB000:02 = 0
  - Codierung B vorgeben durch Schreiben von 0x2300 + [data frame code] nach 0xB000:01  
Kontrolle, dass 0xB000:02 = 0
  - Start des Sendens durch steigende Flanke von Ctrl *SendContinuous*
  - Klemme meldet beendete Übertragung durch *InitAccepted* = 1. *InitAccepted* wird zurückgesetzt mit *SendContinuous*.  
  
CmdNameBedeutung0x2000 + ControlByteCmdControlByteBit 0: Enable MultiDataFrame feature  
Bit 1..7: do not use0x2100 + Valuefirst data frameselect preferred value from data frame table0x2200 + NoOfBytesNo of bytesNo of bytes transfered with first data frame0x2300 + Valuesecond data frameselect preferred value from data frame table  
  
Hinweis: in diesem Modus wird auch die Kodierung 8M1 und 8S1 unterstützt, gleichwohl diese Kodierung nicht über 0x8000:15 auswählbar ist.
- Implementierung weiterer Features auf Anfrage

## Beispiele für die Datenübertragung

### Initialisierung

Vor dem ersten Senden/Empfangen wird die Initialisierung durchgeführt. Dabei wird die Klemme mit den Daten des entsprechenden Settings-Objektes parametrierung.

Vorgehensweise:

1. "Init request" auf 1 setzen
2. Die Erfolgreiche Initialisierung wird von der Klemme durch das Setzen von "Init accepted" bestätigt
3. "Init request" zurücksetzen
4. Die Klemme setzt "Init accepted" auf 0

Die Klemme ist nun zum Datenaustausch bereit.

### Datenübertragung von der Steuerung zur Klemme (2 Zeichen senden)

1. "Output length" auf 2 setzen
2. "Data Out 0" und "Data Out 1" mit Nutzdaten füllen
3. Den Zustand von "Transmit request" umschalten
4. Die Klemme quittiert die Entgegennahme mit einer Zustandsänderung des "Transmit accepted" Bits

### Datenübertragung von der Klemme zur Steuerung (Zeichen empfangen)

1. Die Klemme zeigt durch die Zustandsänderung des "Receive request" Bits an, dass sich neue Daten im Prozessabbild befinden.
2. Die Anzahl der empfangenen Bytes wird in "Input length" abgelegt
3. Die Steuerung quittiert mit einer Zustandsänderung von "Receive request" die Übernahme der Bytes

## 5.5 Hinweise TcVirtualComDriver

Für den Fall, dass die EL60xx mit dem TwinCAT Virtual Serial COM Driver zusammenarbeiten soll, hier einige Hinweise:

Üblicherweise setzt die kundenspezifische übergeordnete Windows-Applikation die gewünschte Eigenschaft der COM-Schnittstelle aus der Applikation heraus, z.B. 2400 Baud und 7N2 Kodierung. Deshalb ist es in der Regel nicht relevant, was an kundenspezifischen Einträgen in den CoE-StartUp-Einträgen, dem Settings-Dialog, des VirtualComPorts oder dem Geräte-CoE vor der Parametrierung eingetragen ist.

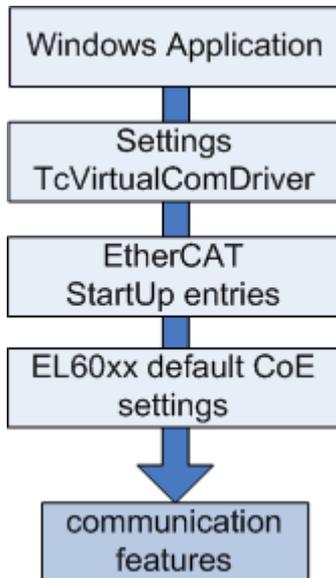


Abb. 152: Die jeweils höhere Ebene gibt (falls vorhanden oder aktiviert) der jeweils darunter liegenden Eben die Kommunikationseigenschaften vor

### HINWEIS

#### TcVirtualComDriver Version

Prüfen Sie, ob eine gewünschte Baudraten/Kodierungs-Kombination von der von Ihnen verwendeten TcVirtualComDriver-Version und EL/EP-Klemme auch unterstützt wird. Ist dies nicht der Fall:

- kann ggf. der COM-Port nicht geöffnet werden
- kann es zu einer ADS-Fehlermeldung im ADS-Logger im Systemmanager kommen
- kann es es zu einer CoE-Fehlermeldung im ADS-Logger kommen

Der TcVirtualComDriver unterstützt ExplicitBaudrate (CoE 0x80n0:1B) ab Version 1.18.

Hat die COM-Applikation die Eigenschaften (wie Baudrate) dann während der Laufzeit gesetzt, kann im Online-CoE die korrekte Einstellung dagegen überprüft werden, s. Abb.

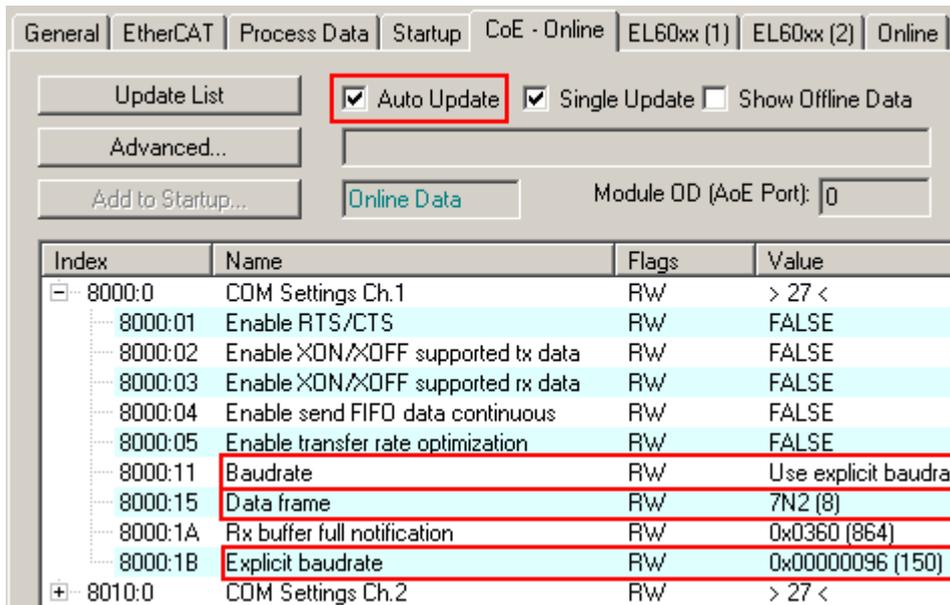


Abb. 153: Prüfen der von der COM-Anwendung gewünschten Einstellungen im CoE

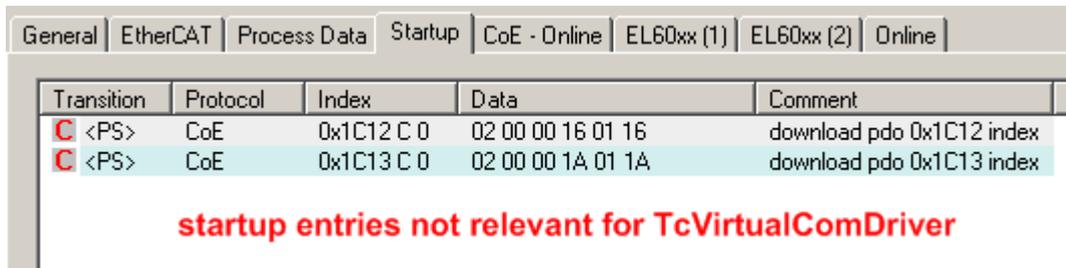


Abb. 154: Default StartUp-Einträge einer EL6002 (Beispiel) - nur die Default-Einträge (hier 0x1C12 und 0x1C13) werden benötigt

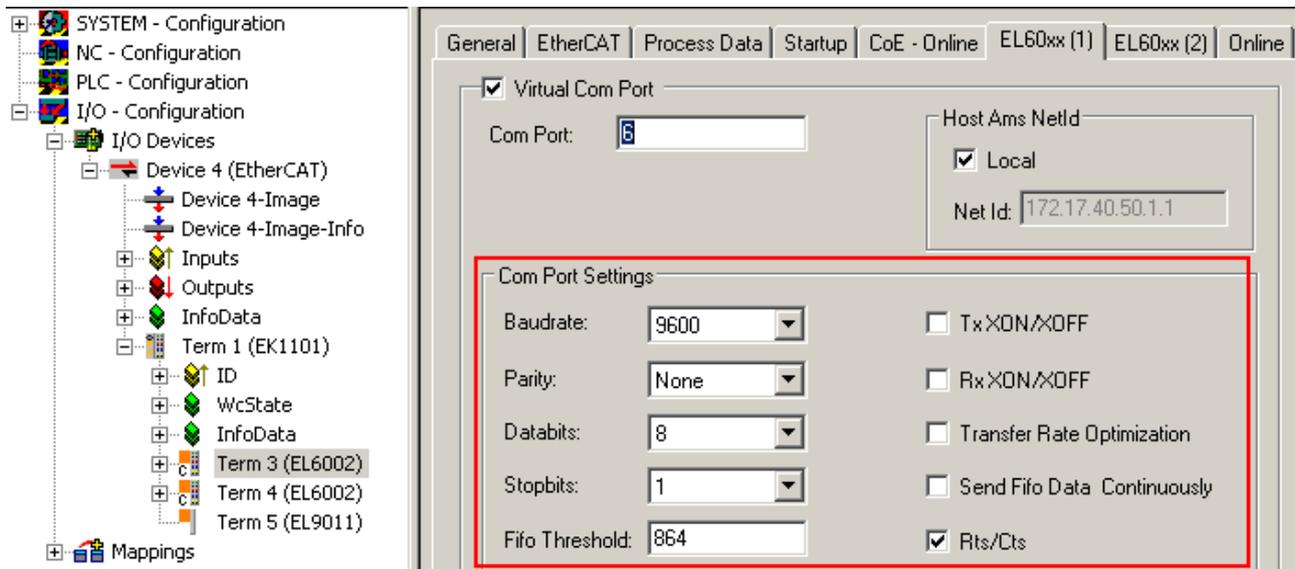


Abb. 155: VirtualComDriver Settings (Beispiel)

Die Settings sind nur relevant, wenn aus der Applikation heraus keine Vorgabe der COM-Eigenschaften erfolgt oder erfolgen kann. Der COM-Port wird mit diesen Default-Einstellungen geöffnet.

*Ausnahme: TransferRateOptimization, SendFifoDataContinuously und FifoThreshold sind nur von TwinCAT aus zu setzen.*

## 5.6 Kommunikationseigenschaften

Zentrale Eigenschaften der EL60xx Geräte zur seriellen Kommunikation sind Baudrate und Datentelegramm-Aufbau/Kodierung auf der Feldseite. Die Geräte unterscheiden sich in den verfügbaren Eigenschaften. Zur besseren Übersicht sind die Geräte hier mit den je nach FW/HW-Stand verfügbaren Parameter aufgelistet.

Überprüfen Sie auf der Beckhoff Webseite, ob eine aktuellere [Dokumentation](#) vorliegt.

Geräteübersicht						
Gerät	verfügbare Baudrate			verfügbare Kodierung		
	Baudrate [baud]	CoE Value [0x80n0:11]	ab FW/HW	Kodierung	CoE Value [0x80n0:15]	ab FW/HW
EL6001	2400	4		7E1	1	
	4800	5		7O1	2	
	9600	6		8N1	3	
	12000	14	ab FW07	8E1	4	
	14400	15	ab FW07	8O1	5	
	19200	7		7E2	9	
	38400	8		7O2	10	
	57600	9		8N2	11	
	115200	10		8E2	12	
	104167	Explicit Baudrate [0x8000:1B]	ab FW08	8O2	13	
				9N1	32 <a href="#">Extended data frame [► 177]</a> [0x8000:1C]	ab FW08
		1000...115200 *)	Explicit Baudrate [0x8000:1B]	Ab FW11		
EL6002	110	<a href="#">Explicit Baudrate [► 135]</a> [0x80n0:1B]	ab FW03	7E1	1	
	150	<a href="#">Explicit Baudrate [► 135]</a> [0x80n0:1B]	ab FW03	7O1	2	
	300	1		8N1	3	
	600	2		8E1	4	
	1200	3		8O1	5	
	2400	4		7N2	8	ab FW03
	4800	5		7E2	9	
	9600	6		7O2	10	
	19200	7		8N2	11	
	38400	8		8E2	12	
	57600	9		8O2	13	
	115200	10				

Geräteübersicht						
Gerät	verfügbare Baudrate			verfügbare Kodierung		
	Baudrate [baud]	CoE Value [0x80n0:11]	ab FW/HW	Kodierung	CoE Value [0x80n0:15]	ab FW/HW
EL6021	2400	4		7E1	1	
	4800	5		7O1	2	
	9600	6		8N1	3	
	19200	7		8E1	4	
	38400	8		8O1	5	
	57600	9		7E2	9	
	115200	10		7O2	10	
				8N2	11	
				8E2	12	
				8O2	13	
				8S1	18	Kommando-Modus ab FW06
EL6022	300	1		7E1	1	
	600	2		7O1	2	
	1200	3		8N1	3	
	2400	4		8E1	4	
	4800	5		8O1	5	
	9600	6		7E2	9	
	19200	7		7O2	10	
	38400	8		8N2	11	
	57600	9		8E2	12	
	115200	10		8O2	13	
	EP6022	300	1		7E1	1
600		2		7O1	2	
1200		3		8N1	3	
2400		4		8E1	4	
4800		5		8O1	5	
9600		6		7E2	9	
19200		7		7O2	10	
38400		8		8N2	11	
57600		9		8E2	12	
115200		10		8O2	13	

\*) Es ist jede ganzzahlige Baudrate als int32 Wert einsetzbar. Der Frequenzfehler über alle Baudraten ist < 0,6%.

## 5.7 LIN Master Feature EL6001

### Beschreibung der „LIN-Master-Unterstützung“

Die EL6001 realisiert eine Busanschaltung für RS232-Pegel. Ab Softwarestand 10 verfügt die EL6001 über eine Zusatzfunktion, um LIN-Telegramme senden und empfangen zu können. Diese Funktion umfasst nicht die formale Abhandlung der Protokollschichten für LIN („Protokollstack“) und auch keine physikalische LIN-Anschaltung, sondern die EL6001 kann in Schreib- und Leserichtung die zeitlich verlängerte Start/Ende-Markierung eines LIN-Telegramms behandeln.

„Oberhalb“ der Klemme ist ein z.B. in PLC realisierter LIN-Stack erforderlich, unterhalb der Klemme ein Pegel-Konverter RS232 <-> 5/12V.

Bitte prüfen Sie vor Einsatz dieser Funktion die Eignung in Ihrem System, da eine vollständige umfassende LIN-Implementierung in der EL6001 nicht gegeben ist!

Ein kompletter LIN-Frame ergibt sich aus eine Master-Anfrage mit direkt nachfolgender Slave-Antwort. Die als Master arbeitende Klemme sendet den Masterframe und empfängt die Daten des Slave. Bei der aktivierten LIN-Funktionalität der EL6001 (ab FW10) wird immer ein „sync-Break“ und ein „sync-Byte“ vor dem „Protected Identifier“ (PID) ausgegeben. Darum kann die Klemme EL6001 zwar auch als Daten empfangender, nicht aber als Daten ausgebender Slave-Knoten eingesetzt werden.

### Aktivierung

Die für LIN erforderliche Zuschaltung eines „sync break“ und „sync-field“ beim Senden kann über das „Command“ Objekt aktiviert werden. Hierzu ist in dem CoE Objekt 0xB000:01 [▶ 173] der Wert 0x3000 einzutragen. Die Antwort ist in dem CoE Objekt 0xB000:03 auslesbar mit den Werten 0x01 0x00 0x00 0x4C 0x49 0x4E. (4C 49 4E = ASCII „LIN“) als Bestätigung der Aktivierung dieser Funktion. Danach ist die Klemme bereit, Frames auf dem LIN-Bus, die das 13-Bit „sync break“ und das „sync-field“ (0x55) enthalten zu empfangen und lediglich die Nachfolgenden Informationen in den Empfangsdaten bereit zu stellen sowie beim Senden diese Kennungen ebenfalls auch auszugeben.

Diese Einstellung wird nicht dauerhaft in der Klemme gespeichert und muss nach jedem neuen Hochfahren erneut geschrieben werden. Um diesen Umstand zu umgehen, bietet sich die Verwendung der Startup [▶ 114]-Liste an.

### Anwenderspezifische Baudrate

In das Objekt „Explicit baudrate“ (0x8000:1B) kann jede bereits verfügbare Baudrate und auch die Baudrate 10417 Baud eingetragen werden.

### Weitere Parameter

Für die korrekte Funktion im LIN-Modus sind die folgenden Einstellungen im CoE Objekt 0x8000 [▶ 153] erforderlich:

Index:Subindex	Name	Wert
0x8000:01	Enable RTS/CTS	FALSE
0x8000:02	Enable XON/XOFF supported tx data	FALSE
0x8000:03	Enable XON/XOFF supported rx data	FALSE
0x8000:04	Enable send FIFO data continuous	FALSE
0x8000:05	Enable transfer rate optimization	TRUE
0x8000:15	Dataframe	3 (8N1)
0x8000:1A	Rx buffer full notification	Siehe <u>Objekte für die Inbetriebnahme</u> [▶ 153]

### Betrieb

Die Verwendung des Control- und Statuswortes bleibt im LIN-Modus unverändert.

Die EL6001 stellt den Nutzdaten im LIN-Modus automatisch das „Break field“ und das „Sync byte field“ voran. Beim Empfangen werden diese beiden Felder ebenfalls automatisch entfernt.

Das Prozessabbild ist wie folgt zu verwenden:

Prozessdatum	Inhalt
Ctrl.Output length	Anzahl der Nutzdatenbytes (n) + 2
Data Out 0	Geschütztes ID Feld (PID = Protected identifier field)
Data Out 1	Daten Byte 1
Data Out n	Daten Byte n
Data Out n+1	Checksumme

### EL6001 LIN Beispiel

Das folgende Beispiel zur LIN-Kommunikation soll verdeutlichen, dass mittels der Zuschaltung der ergänzenden LIN-Funktionalität über das 0xB000 CoE „Command“ Objekt in Kombination mit einer physikalischen Konvertierung von RS-232 auf den LIN-BUS eine Anteilnahme einer PLC Steuerung mittels der EL6001 Klemme in einem LIN-Cluster möglich ist.

### HW-Seitige Schaltung mit Verwendung eines UART-LIN Konverters

Die Umsetzung der physikalischen Schicht als Grundvoraussetzung für eine LIN – Kommunikation umfasst die Bereitstellung eines ein-Draht Busses mit entsprechenden Pegeln 0V / 12V. Hierfür bietet sich ein RS232-LIN Konverter an, der auf einem Sub-D Stecker gebracht wird und auf der anderen Seite drei Pole für Masse, Spannungsversorgung und die elektrische Verbindung zum ein-Draht LIN-Bus bereitstellt.

Die Verwendung des RS232-LIN Konverter erfordert eine Verdrahtung eines Sub-D 9 pol. Steckers an die EL6001 Klemme wie in der Abbildung gezeigt:

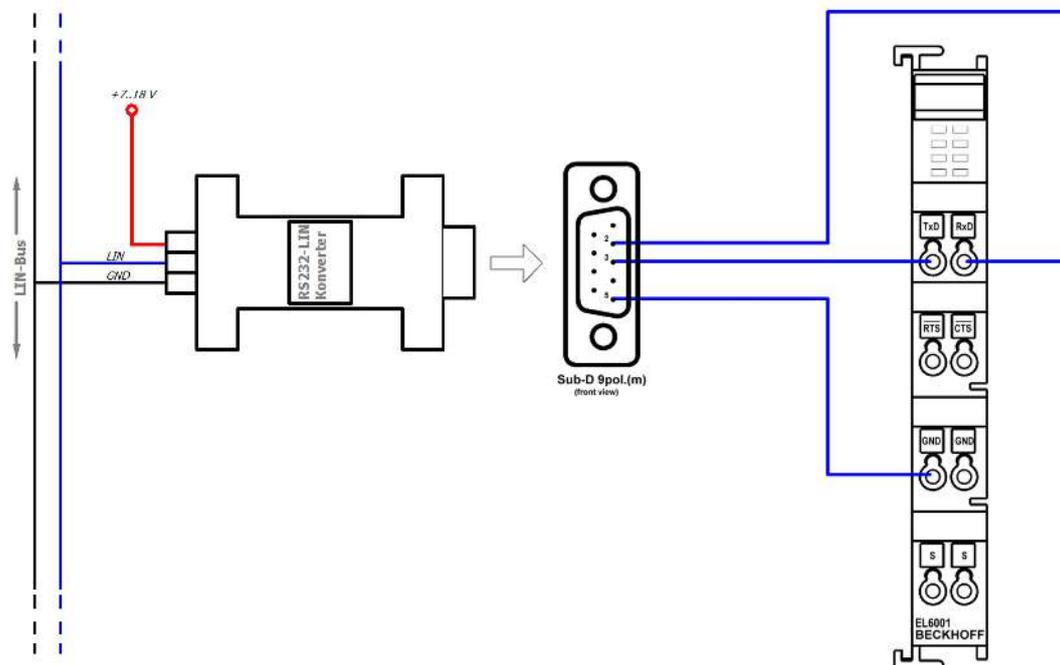


Abb. 156: RS232-LIN Sub-D Stecker angeschlossen an die EL6001 Klemme

Die elektrischen Verbindungen von der Klemme EL6001 zum Sub-D 9 pol. Stecker sind:

- RxD → pin 2
- TxD → pin 3
- GND → pin 5

**Programmierung in ST**

Eine LIN-Kommunikation ist bereits durch eine PLC- „Mastertask“ darstellbar, indem, wie in der üblichen Verwendung des LIN-Bus ein Master ein „Unconditional Frame“ sendet und lediglich die PID (Protocol Identifier) eines bekannten Slave-Knotens auf den LIN-Bus ausgibt und damit i.d.R. Daten von dem jeweiligen Slave anfordert. Im Kapitel Inbetriebnahme ist dazu ein [Programmbeispiel](#) [142] angegeben.

In diesem Beispiel hat der Slave die ID 0x07; der Master sendet also auf dem Bus die ID 0x07 inklusive der berechneten Parität woraus sich das PID-Feld 0x47 ergibt. Die Nachricht auf dem LIN Bus sieht dann wie folgt aus:

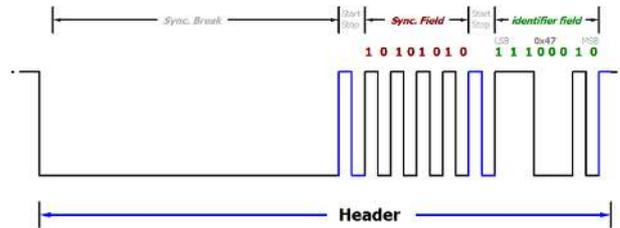


Abb. 157: LIN-Frame Beispiel: Anfrage vom Master an Knoten mit ID 0x07

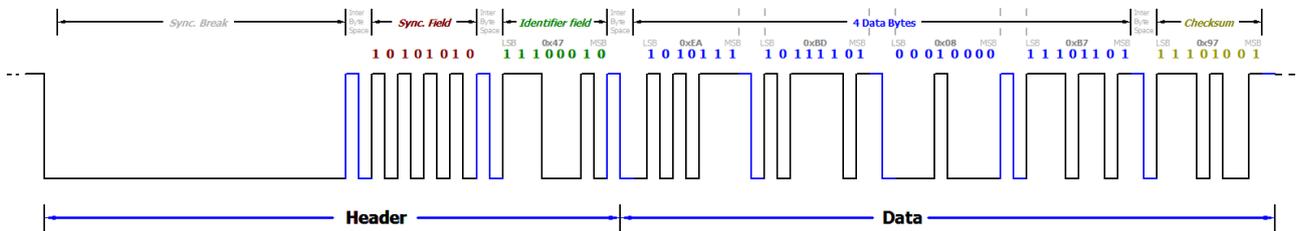


Abb. 158: LIN-Frame Beispiel: ID0x07 mit Daten 0xEA,0xBD,0x08,0xB7 + Checksumme 0x97

Oszilloskop-Aufnahmen LIN Frame mit ID 0x07:

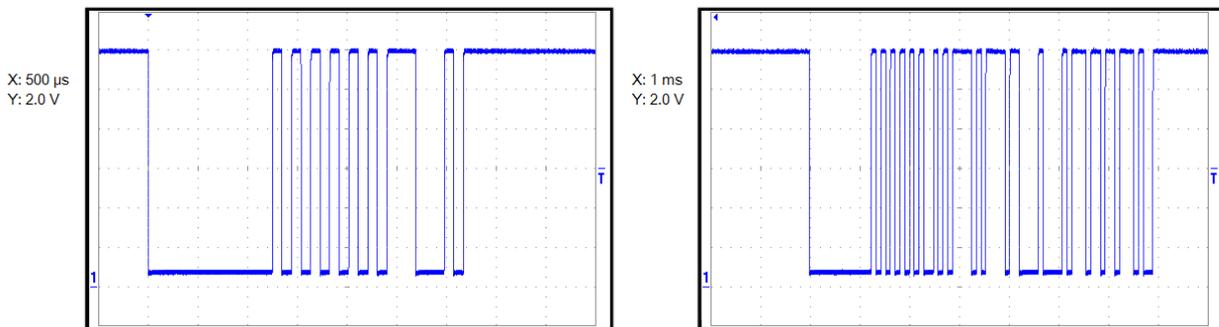


Abb. 159: Links: Anfrage auf dem LIN-Bus mit PID 0x47, rechts: LIN-Frame mit gleicher PID und Daten inkl. Checksumme

## 5.8 Beispielprogramme

### ● Verwendung der Beispielprogramme

**i** Dieses Dokument enthält exemplarische Anwendungen unserer Produkte für bestimmte Einsatzbereiche. Die hier dargestellten Anwendungshinweise beruhen auf den typischen Eigenschaften unserer Produkte und haben ausschließlich Beispielcharakter. Die mit diesem Dokument vermittelten Hinweise beziehen sich ausdrücklich nicht auf spezifische Anwendungsfälle, daher liegt es in der Verantwortung des Anwenders zu prüfen und zu entscheiden, ob das Produkt für den Einsatz in einem bestimmten Anwendungsbereich geeignet ist. Wir übernehmen keine Gewährleistung, dass der in diesem Dokument enthaltene Quellcode vollständig und richtig ist. Wir behalten uns jederzeit eine Änderung der Inhalte dieses Dokuments vor und übernehmen keine Haftung für Irrtümer und fehlenden Angaben.

### 5.8.1 Beispielprogramm 1

 [https://infosys.beckhoff.com/content/1031/el600x\\_el602x/Resources/zip/1719190795.zip](https://infosys.beckhoff.com/content/1031/el600x_el602x/Resources/zip/1719190795.zip)

#### Anschluss eines seriellen Barcodescanners

In diesem Beispiel wird ein Barcode-Lesegerät an der EL6001 angeschlossen. Dabei werden von dem Lesegerät solange Zeichen gelesen, bis das ASCII-Zeichen 0x0D (13<sub>dez</sub>, CR) empfangen wurde.

Daten:

- Schnell Task zur Abwicklung der seriellen Kommunikation: 1 ms Zykluszeit
- Standard SPS Task: 10 ms Zykluszeit
- Barcodescanner an Kanal 1
- TwinCAT 2.11 erforderlich
- Supplement "TwinCAT-PLC-Serial-Communication" erforderlich

Eine detaillierte Beschreibung zur Verwendung der Serial-Communication Library ist im Beckhoff Information System hinterlegt.

Beckhoff Information System -> TwinCAT -> TwinCAT PLC -> TwinCAT Bibliotheken für PC-basierende Systeme -> TwinCAT PLC Library: Serielle Kommunikation

#### Starten des Beispielprogramms

Die Applikationsbeispiele sind mit einem Prüfaufbau getestet und entsprechend beschrieben worden.

Etwaige Abweichungen bei der Einrichtung an realen Applikationen sind möglich.

Für den Prüfaufbau wurde folgende Hardware und Software verwendet:

- TwinCAT-Master-PC mit Betriebssystem Windows XP Professional SP 3, TwinCAT Version 2.11 (Build 1528) und INTEL PRO/100 VE Ethernet-Adapter
- Beckhoff EtherCAT Koppler EK1100, Klemmen EL6001
- Serieller Barcodescanner mit 9 Pol. D-Sub Anschluss

#### Vorgehensweise zum Starten des Programms

- Nach Klick auf den Download-Button speichern Sie das Zip-Archiv lokal auf ihrer Festplatte und entpacken die \*.TSM (Konfigurationsdatei und \*.PRO (PLC-Programmdatei) in einem temporären Arbeitsordner
- Start der \*.TSM-Datei und \*.PRO Datei; der TwinCAT-System Manager und die TwinCAT PLC öffnen sich
- Schließen Sie die Hardware entsprechend Abb. 1 an und verbinden Sie den Ethernet-Adapter ihres PCs mit dem EtherCAT-Koppler (weitere Hinweise hierzu finden sie in den entsprechenden Kopplerhandbüchern)

- Auswahl des lokalen Ethernet-Adapters (ggf. mit Echtzeit-Treiber) unter Systemkonfiguration, E/A - Konfiguration, E/A -Geräte, Gerät (EtherCAT); dann unter Karteireiter "Adapter", "Suchen..." den entsprechenden Adapter auswählen und bestätigen (siehe Abbildungen)

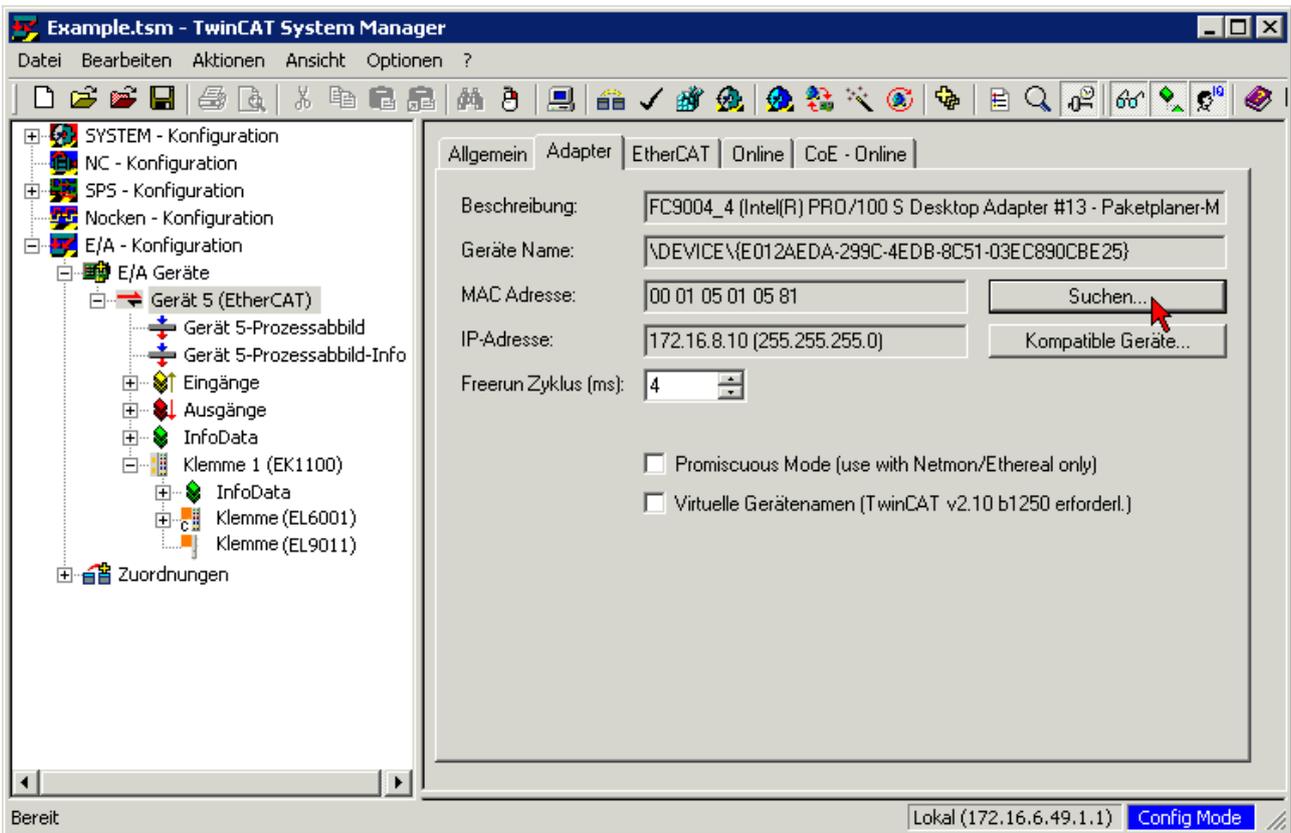


Abb. 160: Suchen des Ethernet-Adapters



Abb. 161: Auswahl und Bestätigung des Ethernet-Adapters

Aktivierung der Konfiguration und bestätigen:

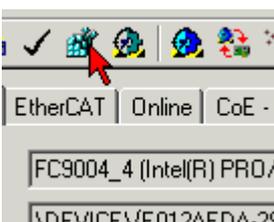


Abb. 162: Aktivierung der Konfiguration



Abb. 163: Konfigurationsaktivierung bestätigen

- Neue Variablenzuordnung bestätigen, Neustart im RUN-Modus:



Abb. 164: Variablenzuordnung erzeugen



Abb. 165: Neustart TwinCAT im RUN-Modus

- In der TwinCAT PLC unter Menü "Projekt" -> "Alles Übersetzen" das Projekt übersetzen:

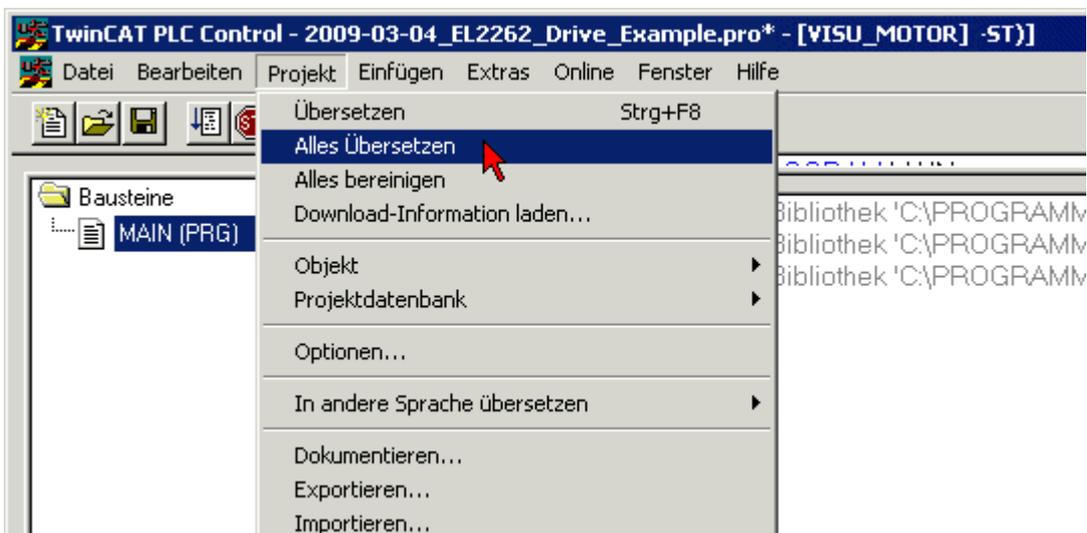


Abb. 166: Projekt übersetzen

- In der TwinCAT PLC: Einloggen mit der Taste "F11", Laden des Programms bestätigen, Start des Programms mit Taste "F5":



Abb. 167: Programmstart bestätigen

- Nachdem das Zeichen 0x13 empfangen wurde, wird der Barcode in "LastBarcode" abgelegt

<pre> 0001 Receive( 0002     Suffix:= '\$0D',           (*RETURN*) 0003     Timeout:= T#1s, 0004     ReceivedString:= Barcode, 0005     RxBuffer:= PxBuffer, 0006     StringReceived=&gt; BarcodeReceived, 0007     Busy=&gt; ReceiveBusy, 0008     Error=&gt; ReceiveErrorID, 0009     RxTimeout=&gt; ReceiveTimeout); 0010 0011 NewBarcode(CLK := Receive.StringReceived); 0012 IF NewBarcode.Q THEN 0013     LastBarcode := Barcode; 0014 END_IF                 </pre>	<pre> Receive.StringRec... = FALSE   Receive.Busy = FALSE Barcode = '544900017888\$R' BarcodeReceived = FALSE ReceiveBusy = FALSE ReceiveErrorID = COMERRO... ReceiveTimeout = FALSE Receive.StringRec... = FALSE NewBarcode.Q = FALSE LastBarcode = '544900017888...' Barcode = '544900017888\$R'                 </pre>
--	---

Abb. 168: Empfangener Barcode

## 5.8.2 Beispielprogramm 2

[https://infosys.beckhoff.com/content/1031/el600x\\_el602x/Resources/zip/1719192971.zip](https://infosys.beckhoff.com/content/1031/el600x_el602x/Resources/zip/1719192971.zip)

### Lesen und Interpretation von Zeitlegrammen

In diesem Beispiel wird ein Weg aufgezeigt, wie verschiedenste serielle Zeitprotokolle in der SPS aufbereitet und interpretiert werden können. Dazu werden IEC61131-PLC-Bausteine vorgestellt, die den in der PLC eintreffenden Bitstrom interpretieren und ggf. die Zeit / Ortsinformationen extrahieren. Mit dieser Information können z.B. Steuerungen synchronisiert oder Bewegungen aufgezeichnet werden.

In diesem Beispiel wird davon ausgegangen, dass die Daten über ein 22-Byte-Interface von einer Beckhoff EL / KL60xx seriellen Datenaustauschklemme geliefert werden.

### Hintergrundinformation

Nicht nur im maritimen Bereich werden Zeit- und Ortsinformationen über serielle Busse transportiert: meist existiert eine Quelle, die über RS232 / RS485, USB oder Ethernet ihre Informationen an untergeordnete Listener/Listener zyklisch oder nach Trigger verteilt.

Weltweit existieren eine sehr große Anzahl von Telegrammformaten zur Verteilung von Zeit- und Ortsinformationen, auch "Sentences" genannt. Solch ein Telegramm besteht aus n Byte Daten und zeichnet sich aus durch:

- Start- und Endezeichen STX, ETX zur Telegramm-Erkennung, ggf. auch mehrere Endezeichen
- eine definierte und feste Länge
- einen definierten Aufbau
- ggf. Checksumme
- ggf. Typbezeichnungen im Sentence

Verschiedenste Organisationen und Firmen haben für spezifische Einsatzzwecke offene oder proprietäre Formate entwickelt. Deshalb werden in diesem Beispiel 2 Beispielimplementationen vorgestellt, die kundenseitig leicht für spezifische andere Protokolle geändert werden können. Die Telegrammformate im Beispiel sind Meinberg Standard und NMEA0183 v2.3 Typ RMC.

## Informationsquellen

Als Datensender im seriellen Bereich werden z. B. GPS- oder Funkuhrengateways verwendet. Diese Geräte empfangen das jeweilige Zeitsignal (GPS über Satellit oder Funkuhr über Langwelle) und setzen es auf die serielle drahtgebundene Übertragung um, z.B. RS232 mit 8N1. Dabei hält das Gateway oft auch eine lokale Taktquelle vor, um bei kurzzeitigem Ausfall des Referenzsignals (GPS, Funksender) über eine gewisse Zeit weiter verlässliche Zeitinformationen verteilen zu können. In Mitteleuropa kann der deutsche DCF77-Sender verwendet werden.

Dabei können aus dem Referenzsignal ggf. noch weitere Informationen verwendet werden:

- GPS: Standortinformationen (W/N und Höhe), anstehende Zeitumstellung DCF77: Wetterinformationen, Großschadenslagen
- DCF77: Wetterinformationen, Großschadenslagen

## Synchronisierung untergeordneter Time-Slaves

Im Allgemeinen sollen untergeordnete Slaves dem Zeitgateway nachgeregelt werden, d.h. sie sollen sich in ihrer Zeit auf synchronisieren. Dazu sind nötig:

- der Offset: die absolute Abweichung der eigenen Uhr zur Referenzzeit des Gateways – diese Information kann z.B. azyklisch seriell übertragen werden wenn bekannt ist, wann die Zeitinformation als gültig anzusehen ist. Der serielle Transportweg aus deinem Beispiel ermöglicht die Gewinnung dieser Offset-Information.
- das Frequenzverhältnis: ein hochgenau zyklisches Signal vom Gateway zu den Slaves erlaubt das Ausregeln von Driftvorgängen und gibt ggf. auch den Zeitpunkt an, wann der obige absolute Offset gültig ist. Beispiel für ein solches Signal ist das weitverbreitete PPS (Pulse per Second).

Wird das serielle Telegramm vom Gateway zyklisch auf den Bus gelegt, kann der Zeitpunkt des ersten Bits oft als PPS-Signal interpretiert werden. In den Bausteinen dieses Beispiels geht diese Information verloren; es wird nur die absolute Zeitinformation ausgewertet.

## Zeitformate

Weit verbreitet sind Zeitlegramme nach dem NMEA0183-Standard. Dabei ist zu beachten:

- Innerhalb der NMEA0183-Standards gibt es aktuell (2009) 8 Versionen von NMEA1083 1.5 bis 4.00 – der Aufbau von Telegrammen kann sich zwischen den Versionen geändert haben
- Allein in NMEA1083 v2.30 sind über 70 verschiedene Formate definiert, dazu können Gerätehersteller eigene Formate erstellen
- Das Telegramm wird Sentence genannt
- Eine TalkerID (2 Zeichen) und eine TypID (3 Zeichen) am Beginn definieren den verwendeten Sentence-Typ
- Es wird eine Checksumme über das Telegramm berechnet
- Informationen sind unter [www.nmea.org](http://www.nmea.org) oder anderweitig online einsichtlich.

Des weiteren existieren viele proprietäre Formate wie Meinberg Standard, Siemens SINEC H1, SAT 1703 oder militärische wie IRIG-Codes (USA).

## Anwendung des Beispielprogramms

Im PLC-Projekt sind 2 FunctionBlocks (FB) enthalten, die exklusiv mit einer EL/KL600x verlinkt werden müssen. Dann holen Sie von der Klemme die empfangenen Bytes ab und interpretieren nach Möglichkeit den Inhalt. Die FB decken ab:

- Meinberg Standard
- NMEA0183 v2.3 Typ RMC

Für andere Zeitformate lassen sich danach eigene Interpretations-FB anhand eines bekannten Telegrammaufbaus erstellen, wenden Sie sich dazu an Ihren Gerätehersteller.

Im Beispiel verwendete Hardware: EK1100 und EL6001 (auch EL600x, KL600x)  
 Der FB der getestet werden soll, muss mit seinen Prozessdaten im Systemmanager mit der Klemme verlinkt werden (22-Byte-Prozessabbild und Control/Status-Word).  
 Im Beispiel ist der NMEA-Baustein verlinkt. Die Baudrate der Klemme muss auf Ihren Sender eingestellt werden.

#### Referenzen:

- [www.beuth.de](http://www.beuth.de), IEC61162: angelehnt an NMEA2000
- <http://gpsinformation.net> privat, über NMEA, viele Formate: <http://www.gpsinformation.org/dale/nmea.htm>  
<http://www.nmea.de/nmea0183datensatze.html>
- <http://www.meinberg.de/german/info/irig.htm>, IRIG-Codes

### 5.8.3 Beispielprogramm 3 (LIN)

Download (TwinCAT 3 Beispielprogramm):



[https://infosys.beckhoff.com/content/1031/el600x\\_el602x/Resources/zip/1805853195.zip](https://infosys.beckhoff.com/content/1031/el600x_el602x/Resources/zip/1805853195.zip)

#### Globale Variablen zum LIN-Master Beispielprogramm

```
VAR_GLOBAL
nSetBaudrate : UINT := 10417; // Für Rx-Delay-Berechnung
(* I/O variables for EL6001 terminal acting as Master*)
COMin_EL6001_MASTER AT %I* : EL6inData22B; (* linked to the EL6001 in the TwinCAT System Manager *)
COMout_EL6001_MASTER AT %Q* : EL6outData22B; (* linked to the EL6001 in the TwinCAT System Manager *)
RxBuffer_MASTER : ComBuffer; (* Receive data buffer; used with all receive function blocks *)
TxBuffer_MASTER : ComBuffer; (* Transmit data buffer; used with all receive function blocks *)
END_VAR
```

#### Datentypen zum LIN-Master-Slave Beispielprogramm

```
TYPE tDataFrame : ARRAY[0..8] OF BYTE; END_TYPE // Datentyp für ein LIN-Frame
```

Funktion zum Hinzufügen von Datenlängeninformation und Paritätsbits:

#### Deklarationsteil:

```
FUNCTION F_ADD_LIN_NODE_PARITY : BYTE
VAR_INPUT
nNodeID:BYTE; // Eingangsvariable: Knoten-Id
nReqLen:BYTE; // Eingangsvariable: Längen-Kennung: 2,4,8 Byte-Frame
END_VAR
VAR
bParity0:BYTE; // Interner Zwischenwert für Parity 0
bParity1:BYTE; // Interner Zwischenwert für Parity 1
nPrepId:BYTE; // Zwischenwert der PID
END_VAR
```

#### Ausführungsteil:

```
nPrepId := nNodeID OR SHL(nReqLen,4);

bParity0 :=
(nNodeID AND 2#0001)
XOR (SHR((nNodeID AND 2#0010), 1))
XOR (SHR((nNodeID AND 2#0100), 2))
XOR (SHR((nNodeID AND 2#0001_0000), 4));

bParity1 := 16#01 AND (NOT(
SHR((nNodeID AND 2#0010), 1)
XOR (SHR((nNodeID AND 2#1000), 3))
XOR (SHR((nNodeID AND 2#0001_0000), 4))
XOR (SHR((nNodeID AND 2#0010_0000), 5))));

F_ADD_LIN_NODE_PARITY := nPrepId OR SHL(bParity0,6) OR SHL(bParity1,7);
```

## Funktion zum Berechnen der Checksumme (herkömmliche Methode)

## Deklarationsteil:

```

FUNCTION F_CALC_LIN_CHKSUM : BYTE
VAR_INPUT
pData:POINTER TO ARRAY[0..10] OF BYTE; // Zeiger auf Datenfeld
nLen:BYTE; // Anzahl Bytes zur Berechnung der Checksumme
END_VAR

VAR
i: BYTE; // interne Zählvariable
wResult:WORD; // Ausgangswert
END_VAR

```

## Ausführungsteil:

```

wResult := BYTE_TO_WORD(pData^[0]);
FOR i := 1 TO (nLen-1) DO
wResult := wResult + BYTE_TO_WORD(pData^[i]);
IF wResult > 255 THEN
wResult := wResult - 255;
END_IF
END_FOR
F_CALC_LIN_CHKSUM := WORD_TO_BYTE(NOT(wResult));

```

Dieser Master-Programmteil sollte von einer entsprechenden separaten Task aufgerufen werden. Hier wird alle 200 ms ein Knoten im LIN-Bus mit der ID 0x07 angefragt.

## Deklarationsteil:

```

PROGRAM EL6001_MASTER
VAR
Timer: TON; // Timer für period. Anfragen vom Master
Send: SendData; // Funktionsblock aus TC2_SerialCom
SendBusy: BOOL; // Flag-Kopie von SendData.Busy
SendErrorID: ComError_t; // Error-ID Kopie
aDataTX:tDataFrame; // Zu sendendes Dataframe
Receive: ReceiveData; // Funktionsblock aus TC2_SerialCom
LastReceivedDataBytes: tDataFrame; // Kopie (Latch) der empfangenen Daten
DataReceived: BOOL; // Flag-Kopie von Empfangsbestätigung
ReceiveBusy: BOOL; // Flag Kopie von Empfangen noch nicht fertig
ReceiveError: BOOL; // Flag Kopie von Empfangsfehler
ReceiveErrorID: ComError_t; // Error-ID Kopie
ReceiveTimeout: BOOL; // Flag Kopie von Empfangs-timeout
ReceiveCounter: UDINT := 0; //Anzahl empfangener Frames
aDataRX:tDataFrame; // Zu empfangenes Dataframe
nDataLen:BYTE := 4; // Festgelegte Datenlänge
nState:BYTE := 0; // Initialer Status (start)
bNodeId_SL1:BYTE := 16#07; // ID des Slave-Knotens
bReqLen_SL1: BYTE:= 0; // 4 Datenbytes optional Wert 2
nRxChecksum:BYTE; // Wertespeicher für empfangene Checksumme
nCalcChecksum:BYTE; // Wertespeicher für berechnete Checksumme
T_ReceiveDelay:TIME; // Wertespeicher für ber. Verzögerungszeit
END_VAR

```

## Ausführungsteil:

```

(*=====
Receive data
*)
CASE nState OF
0:
Timer(IN:=TRUE, PT:=T#0.5S); // Master Requests: Periodische Anfragen
IF Timer.Q THEN
// Setze ID ein:
aDataTX[0] := F_ADD_LIN_NODE_PARITY(bNodeId_SL1, bReqLen_SL1);
LastReceivedDataBytes[0] := aDataTX[0];
// Sende Anfrage an Slave 1 (get Data)
Send(pSendData:= ADR(aDataTX), Length:= 1,
TXbuffer:= TxBuffer_MASTER,
Busy => SendBusy, Error => SendErrorID);
Timer(IN:=FALSE); (* reset timer *)
IF NOT SendBusy THEN // Warten bis Senden abgeschlossen ist
nState := nState + 1;
END_IF
END_IF
1:
// Verzögerungszeit aus 1/Tbaud * Anzahl Bytes * (8 Datenbits + 2 Bit:start-Stop) * 1000ms
T_ReceiveDelay := REAL_TO_TIME((1/DINT_TO_REAL(nSetBaudrate)) * 33 * 1000); // .. für nur 1 Byte
Timer(IN:=TRUE, PT:=T_ReceiveDelay);
IF Timer.Q THEN
// Warten bis ID gesendet wurde
nState := nState + 1;
Timer(IN:=FALSE); (* reset timer *)
END_IF
2:
Receive(
pReceiveData:= ADR(aDataRX),
SizeReceiveData:= (nDataLen + 1),
RXbuffer:= RxBuffer_MASTER,
Timeout:= T#1S,
DataReceived=> DataReceived,
busy=> ReceiveBusy,
Error=> ReceiveErrorID,
RxTimeout=> ReceiveTimeout );
IF DataReceived THEN
//DataReceived := FALSE;
ReceiveCounter := ReceiveCounter + 1;
IF NOT ReceiveBusy THEN
// Vergleiche Checksumme
nRxChecksum := aDataRX[nDataLen];
nCalcChecksum := F_CALC_LIN_CHKSUM(pData := ADR(aDataRX), nLen := nDataLen);
IF(nRxChecksum = nCalcChecksum) THEN
//Antwort empfangen - lösche Datenpuffer:
memset(ADR(LastReceivedDataBytes[1]), 0, (SIZEOF(aDataRX)-1));
// Übernehme Daten wenn Checksumme OK
memcpy(ADR(LastReceivedDataBytes[1]), ADR(aDataRX), (nDataLen +1));
END_IF
nState := 0;
END_IF
ELSE
Timer(IN:=TRUE, PT:=T#0.1S); // Receive-Timeout 100 ms: Keine Daten
IF Timer.Q THEN
nState := 0;
END_IF
END_IF
END_CASE
(*=====
*)

```

## Programm Fast-Task / RS232 Hintergrundkommunikation

Hier ist in diesem Beispiel eine dritte Task mit möglichst geringen „Cycle-Ticks“ anzulegen, die für die Hintergrundkommunikation mit der EL6001 Klemme (als Master) zuständig ist.

### Deklarationsteil:

```
PROGRAM FAST
VAR
(* background communication with the EL6001 as Master device *)
COMportControl_MASTER: SerialLineControl;
COMportControlError_MASTER: BOOL;
COMportControlErrorID_MASTER: ComError_t;
END_VAR
```

### Ausführungsteil:

```
COMportControl_MASTER(
Mode:= SERIALLINEMODE_EL6_22B,
pComIn:= ADR(COMin_EL6001_MASTER), (* I/O data; see global variables *)
pComOut:= ADR(COMout_EL6001_MASTER), (* I/O data; see global variables *)
SizeComIn:= SIZEOF(COMin_EL6001_MASTER), (* I/O data; see global variables *)
TxBuffer:= TxBuffer_MASTER, (* transmit buffer; see global variables *)
RxBuffer:= RxBuffer_MASTER, (* receive buffer; see global variables *)
Error=> COMportControlError_MASTER,
ErrorID=> COMportControlErrorID_MASTER );
```

# 6 Übersicht CoE Objekte EL6001, EL6021

## 6.1 Objektbeschreibung und Parametrierung

### ● EtherCAT XML Device Description

**i** Die Darstellung entspricht der Anzeige der CoE-Objekte aus der EtherCAT XML Device Description. Es wird empfohlen, die entsprechende aktuellste XML-Datei im Download-Bereich auf der [Beckhoff Website](#) herunterzuladen und entsprechend der Installationsanweisungen zu installieren.

### ● Parametrierung

**i** Die Parametrierung der Klemme wird über den [CoE - Online Reiter \[▶ 115\]](#) (mit Doppelklick auf das entsprechende Objekt, s.u.) vorgenommen.

### Einführung

In der CoE-Übersicht sind Objekte mit verschiedenem Einsatzzweck enthalten:

- [Objekte die zu Parametrierung \[▶ 151\]](#) bei der Inbetriebnahme nötig sind
- [Objekte die interne Settings \[▶ 153\]](#) anzeigen und ggf. nicht veränderlich sind
- [Profilspezifische Objekte \[▶ 171\]](#), die die Statusanzeigen der Ein- und Ausgänge darstellen (ab [Hardwarestand 03 \[▶ 195\]](#))

Im Folgenden werden zuerst die im normalen Betrieb benötigten Objekte vorgestellt, dann die für eine vollständige Übersicht noch fehlenden Objekte.

### 6.1.1 Objekte für die Inbetriebnahme

#### Index 0x1011 Restore default parameters

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1011:0	<a href="#">Restore default parameters [▶ 208]</a>	Herstellen der Defaulteinstellungen	UINT8	RO	0x01 (1 <sub>dez</sub> )
1011:01	SubIndex 0x001	Wenn Sie dieses Objekt im Set Value Dialog auf <b>"0x64616F6C"</b> setzen, werden alle Backup Objekte wieder in den Auslieferungszustand gesetzt.	UINT32	RW	0x00000000 (0 <sub>dez</sub> )

#### Index 0x4073 Baudrate

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
4073:0	Baudrate	Detaillierte Hinweise hierzu im Kapitel <a href="#">Kommunikationseigenschaften [▶ 137]</a>	UINT16	RW	0x0006 (6 <sub>dez</sub> )

#### Index 0x4074 Data frame

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
4074:0	Data frame	Detaillierte Hinweise hierzu im Kapitel <a href="#">Kommunikationseigenschaften [▶ 137]</a>	UINT16	RW	0x0003 (3 <sub>dez</sub> )

## Index 0x4075 Feature bits

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
4075:0	Feature bits	Länge dieses Objekts	UINT8	RO	0x06 (6 <sub>dez</sub> )
4075:01	<b>EL6001:</b> Enable RTS/CTS	0: RTS/CTS nicht freigeschaltet 1: RTS/CTS freigeschaltet	BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
	<b>EL6021:</b> Enable half duplex	0: Vollduplexmodus 1: Halbduplexmodus			
4075:02	Enable XON/XOFF supported tx data	0: XON/XOFF wird für Sendedaten nicht unterstützt 1: XON/XOFF wird für Sendedaten unterstützt	BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
4075:03	Enable XON/XOFF supported rx data	0: XON/XOFF wird für Empfangsdaten nicht unterstützt 1: XON/XOFF wird für Empfangsdaten unterstützt	BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
4075:04	<b>EL6001:</b> Enable send FIFO data continuous	0: Kein kontinuierliches Senden der Daten aus dem FIFO 1: Über die Steuerung wird der Sendebuffer gefüllt (bis zu 128 Byte). Mit steigender Flanke des Bits <a href="#">CW.3 [▶_174]</a> im Control-Wort wird der gefüllte Buffer-Inhalt gesendet. Sind die Daten übertragen, so wird dies durch das Setzen des Bits <a href="#">SW.2 [▶_175]</a> im Status-Wort von der Klemme an die Steuerung quittiert. SW.2 wird mit CW.3 zurückgesetzt.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
	<b>EL6021:</b> Enable point to point connection (RS422)	0: Punkt zu Punkt Verbindung deaktiviert 1: Punkt zu Punkt Verbindung aktiviert			
4075:05	<b>EL6001:</b> Enable transfer rate optimization	0: Transferraten-Optimierung ausgeschaltet 1: Transferraten-Optimierung eingeschaltet:  Der Inhalt des Eingangspuffers wird automatisch ins Prozessabbild übernommen, wenn <ul style="list-style-type: none"> <li>nach dem Empfang von Daten für ca. 16 Bitzeiten (also die Zeit, die der Empfang von 2 Bytes gedauert hätte) kein weiteres Byte empfangen wurde.</li> <li>der Eingangspuffer voll ist</li> </ul>	BOOLEAN	RO	0x01 (1 <sub>dez</sub> )
	<b>EL6021:</b> Enable send FIFO data continuous	0: Kein kontinuierliches Senden der Daten aus dem FIFO 1: Über die Steuerung wird der Sendebuffer gefüllt (bis zu 128 Byte). Mit steigender Flanke des Bits <a href="#">CW.3 [▶_174]</a> im Control-Wort wird der gefüllte Buffer-Inhalt gesendet. Sind die Daten übertragen, so wird dies durch das Setzen des Bits <a href="#">SW.2 [▶_175]</a> im Status-Wort von der Klemme an die Steuerung quittiert. SW.2 wird mit CW.3 zurückgesetzt.			
4075:06	<b>nur EL6021:</b> Enable transfer rate optimization	0: Transferraten-Optimierung ausgeschaltet 1: Transferraten-Optimierung eingeschaltet:  Der Inhalt des Eingangspuffers wird automatisch ins Prozessabbild übernommen, wenn <ul style="list-style-type: none"> <li>nach dem Empfang von Daten für ca. 16 Bitzeiten (also die Zeit, die der Empfang von 2 Bytes gedauert hätte) kein weiteres Byte empfangen wurde.</li> <li>das Prozessabbild gefüllt ist</li> </ul>	BOOLEAN	RW	0x01 (1 <sub>dez</sub> )

## Index 0x4076 Rx buffer full notification

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
4076:0	Rx buffer full notification	Der Wert legt die Anzahl der Daten im Empfangs-FIFO fest, ab der im Status-Byte das Bit <a href="#">SW.3 [▶_175]</a> (BUF_F) gesetzt wird.	UINT16	RW	0x0000 (0 <sub>dez</sub> )

**Index 0x8000 COM Settings [ab Hardwarestand 03]**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default		
8000:0	COM Settings	Max. SubIndex (hex)	UINT8	RO	0x26 (38 <sub>dez</sub> )		
8000:01**	Enable RTS/CTS	FALSE	RTS/CTS nicht freigeschaltet		BOOLEAN	RW	0x01 (1 <sub>dez</sub> )
		TRUE	RTS/CTS freigeschaltet				
8000:02	Enable XON/XOFF supported tx data	FALSE	XON/XOFF wird für Sendedaten nicht unterstützt		BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
		TRUE	XON/XOFF wird für Sendedaten unterstützt				
8000:03	Enable XON/XOFF supported rx data	FALSE	XON/XOFF wird für Empfangsdaten nicht unterstützt		BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
		TRUE	XON/XOFF wird für Empfangsdaten unterstützt				
8000:04	Enable send FIFO data continuous	FALSE	Kein kontinuierliches Senden der Daten aus dem FIFO		BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
		TRUE	Kontinuierliches Senden der Daten aus dem FIFO freigeschaltet: Über die Steuerung wird der Sendebuffer gefüllt (bis zu 128 Byte). Mit steigender Flanke des Bits " <u>SendContinious</u> [▶ 172]" wird der gefüllte Buffer-Inhalt gesendet. Sind die Daten übertragen, so wird dies durch das Setzen des Bits " <u>InitAccepted</u> " von der Klemme an die Steuerung quittiert. " <u>InitAccepted</u> [▶ 171]" wird mit " <u>SendContinuous</u> " zurückgesetzt.				
8000:05	Enable transfer rate optimization	FALSE	Transferraten-Optimierung ausgeschaltet		BOOLEAN	RW	0x01 (1 <sub>dez</sub> )
		TRUE	Transferraten-Optimierung eingeschaltet: Der Inhalt des Eingangspuffers wird automatisch ins Prozessabbild übernommen, wenn <ul style="list-style-type: none"> <li>nach dem Empfang von Daten für ca. 16 Bitzeiten (also die Zeit, die der Empfang von 2 Bytes gedauert hätte) kein weiteres Byte empfangen wurde.</li> <li>das Prozessabbild gefüllt ist</li> </ul>				
8000:06** *	Enable half duplex	FALSE	Voll duplex-Modus		BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
		TRUE	Halbduplex-Modus				
8000:07** *	Enable point to point connection (RS422)	FALSE	Punkt zu Punkt Verbindung deaktiviert		BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
		TRUE	Punkt zu Punkt Verbindung aktiviert				
8000:11	Baud rate	Detaillierte Hinweise hierzu im Kapitel <u>Kommunikationseigenschaften</u> [▶ 137]		BIT4	RW	0x06 (6 <sub>dez</sub> )	
8000:15	Data frame	Detaillierte Hinweise hierzu im Kapitel <u>Kommunikationseigenschaften</u> [▶ 137]		BIT4	RW	0x03 (3 <sub>dez</sub> )	
8000:1A	Rx buffer full notification	Der Wert legt die Anzahl der Daten im Empfangs-FIFO fest, ab der das Bit " <u>Buffer full</u> [▶ 172]" gesetzt wird.		UINT16	RW	0x0360 (864 <sub>dez</sub> )	
8000:1B**	Explicit baudrate	In diesem Objekt kann die gewünschte Baudrate direkt als Zahl eingetragen werden. Unterstützt werden nur die unter <u>Kommunikationseigenschaften</u> [▶ 137] genannten Baudraten. Änderungen an diesem Objekt werden auch in die Objekte 0x8000:11 und 0x4073 übernommen		UINT32	RW	0x00000384 (9600 <sub>dez</sub> )	
8000:1C**	Extended data frame	In diesem Objekt können neben den üblichen Datenrahmen auch spezielle Formate ausgewählt werden. (z. B. 9N1). Änderungen in diesem Objekt werden auch in die Objekte 0x8000:15 und 0x4074 übernommen.		ENUM16	RW	0x0003 (3 <sub>dez</sub> )	

\*\* ) nur EL6001

\*\*\* ) nur EL6021

**6.1.2 Standardobjekte (0x1000-0x1FFF)**

Die Standardobjekte haben für alle EtherCAT-Slaves die gleiche Bedeutung.

**Index 0x1000 Device type**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1000:0	Device type	Geräte-Typ des EtherCAT-Slaves: Das Lo-Word enthält das verwendete CoE Profil (5001). Das Hi-Word enthält das Modul Profil entsprechend des Modular Device Profile.	UINT32	RO	0x02581389 (39326601 <sub>dez</sub> )

**Index 0x1008 Device name**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1008:0	Device name	Geräte-Name des EtherCAT-Slave	STRING	RO	EL6001 EL6021

**Index 0x1009 Hardware version**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1009:0	Hardware version	Hardware-Version des EtherCAT-Slaves	STRING	RO	04

**Index 0x100A Software version**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
100A:0	Software version	Firmware-Version des EtherCAT-Slaves	STRING	RO	05

**Index 0x1018 Identity**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1018:0	Identity	Informationen, um den Slave zu identifizieren	UINT8	RO	0x04 (4 <sub>dez</sub> )
1018:01	Vendor ID	Hersteller-ID des EtherCAT-Slaves	UINT32	RO	0x00000002 (2 <sub>dez</sub> )
1018:02	Product code	Produkt-Code des EtherCAT-Slaves	UINT32	RO	0x17853052 (394604626 <sub>dez</sub> )
1018:03	Revision	Revisionsnummer des EtherCAT-Slaves, das Low-Word (Bit 0-15) kennzeichnet die Sonderklemmennummer, das High-Word (Bit 16-31) verweist auf die Gerätebeschreibung	UINT32	RO	0x00100000 (1048576 <sub>dez</sub> )
1018:04	Serial number	Seriennummer des EtherCAT-Slaves, das Low-Byte (Bit 0-7) des Low-Words enthält das Produktionsjahr, das High-Byte (Bit 8-15) des Low-Words enthält die Produktionswoche, das High-Word (Bit 16-31) ist 0	UINT32	RO	0x00000000 (0 <sub>dez</sub> )

**Index 0x10F0 Backup parameter handling**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
10F0:0	Backup parameter handling	Informationen zum standardisierten Laden und Speichern der Backup Entries	UINT8	RO	0x01 (1 <sub>dez</sub> )
10F0:01	Checksum	Checksumme über alle Backup-Entries des EtherCAT-Slaves	UINT32	RO	0x00000000 (0 <sub>dez</sub> )

**Index 0x1400 RxPDO-Par Outputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1400:0	RxPDO-Par Outputs	PDO Parameter RxPDO 1	UINT8	RO	0x06 (6 <sub>dez</sub> )
1400:06	Exclude RxPDOs	Hier sind die RxPDOs (Index 0xder RxPDO Mapping Objekte) angegeben, die nicht zusammen mit RxPDO 1 übertragen werden dürfen	OCTET-STRING[6]	RO	01 16 02 16 04 16

**Index 0x1401 RxPDO-Par Outputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1401:0	RxPDO-Par Outputs	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6 <sub>dez</sub> )
1401:06	Exclude RxPDOs	Hier sind die RxPDOs (Index 0xder RxPDO Mapping Objekte) angegeben, die nicht zusammen mit RxPDO 2 übertragen werden dürfen	OCTET-STRING[6]	RO	00 16 02 16 04 16

**Index 0x1402 RxPDO-Par Outputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1402:0	RxPDO-Par Outputs	PDO Parameter RxPDO 3	UINT8	RO	0x06 (6 <sub>dez</sub> )
1402:06	Exclude RxPDOs	Hier sind die RxPDOs (Index 0xder RxPDO Mapping Objekte) angegeben, die nicht zusammen mit RxPDO 3 übertragen werden dürfen	OCTET-STRING[6]	RO	00 16 01 16 04 16

**Index 0x1600 RxPDO-Map Outputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1600:0	RxPDO-Map Outputs	PDO Mapping RxPDO 1	UINT8	RO	0x04 (4 <sub>dez</sub> )
1600:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3001 ( Outputs), entry 0x01 (Ctrl))	UINT32	RO	0x3001:01, 8
1600:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3001 ( Outputs), entry 0x02 (Data Out 0))	UINT32	RO	0x3001:02, 8
1600:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3001 ( Outputs), entry 0x03 (Data Out 1))	UINT32	RO	0x3001:03, 8
1600:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3001 ( Outputs), entry 0x04 (Data Out 2))	UINT32	RO	0x3001:04, 8

**Index 0x1601 RxPDO-Map Outputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1601:0	RxPDO-Map Outputs	PDO Mapping RxPDO 2	UINT8	RO	0x06 (6 <sub>dez</sub> )
1601:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3002 ( Outputs), entry 0x01 (Ctrl))	UINT32	RO	0x3002:01, 8
1601:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3002 ( Outputs), entry 0x02 (Data Out 0))	UINT32	RO	0x3002:02, 8
1601:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3002 ( Outputs), entry 0x03 (Data Out 1))	UINT32	RO	0x3002:03, 8
1601:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3002 ( Outputs), entry 0x04 (Data Out 2))	UINT32	RO	0x3002:04, 8
1601:05	SubIndex 0x005	5. PDO Mapping entry (object 0x3002 ( Outputs), entry 0x05 (Data Out 3))	UINT32	RO	0x3002:05, 8
1601:06	SubIndex 0x006	6. PDO Mapping entry (object 0x3002 ( Outputs), entry 0x06 (Data Out 4))	UINT32	RO	0x3002:06, 8

**Index 0x1602 RxPDO-Map Outputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1602:0	RxPDO-Map Outputs	PDO Mapping RxPDO 3	UINT8	RO	0x17 (23 <sub>dez</sub> )
1602:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3003 (Outputs), entry 0x01 (Ctrl))	UINT32	RO	0x3003:01, 16
1602:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3003 (Outputs), entry 0x02 (Data Out 0))	UINT32	RO	0x3003:02, 8
1602:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3003 (Outputs), entry 0x03 (Data Out 1))	UINT32	RO	0x3003:03, 8
1602:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3003 (Outputs), entry 0x04 (Data Out 2))	UINT32	RO	0x3003:04, 8
1602:05	SubIndex 0x005	5. PDO Mapping entry (object 0x3003 (Outputs), entry 0x05 (Data Out 3))	UINT32	RO	0x3003:05, 8
1602:06	SubIndex 0x006	6. PDO Mapping entry (object 0x3003 (Outputs), entry 0x06 (Data Out 4))	UINT32	RO	0x3003:06, 8
1602:07	SubIndex 0x007	7. PDO Mapping entry (object 0x3003 (Outputs), entry 0x07 (Data Out 5))	UINT32	RO	0x3003:07, 8
1602:08	SubIndex 0x008	8. PDO Mapping entry (object 0x3003 (Outputs), entry 0x08 (Data Out 6))	UINT32	RO	0x3003:08, 8
1602:09	SubIndex 0x009	9. PDO Mapping entry (object 0x3003 (Outputs), entry 0x09 (Data Out 7))	UINT32	RO	0x3003:09, 8
1602:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0A (Data Out 8))	UINT32	RO	0x3003:0A, 8
1602:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0B (Data Out 9))	UINT32	RO	0x3003:0B, 8
1602:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0C (Data Out 10))	UINT32	RO	0x3003:0C, 8
1602:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0D (Data Out 11))	UINT32	RO	0x3003:0D, 8
1602:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0E (Data Out 12))	UINT32	RO	0x3003:0E, 8
1602:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x3003 (Outputs), entry 0x0F (Data Out 13))	UINT32	RO	0x3003:0F, 8
1602:10	SubIndex 0x016	16. PDO Mapping entry (object 0x3003 (Outputs), entry 0x10 (Data Out 14))	UINT32	RO	0x3003:10, 8
1602:11	SubIndex 0x017	17. PDO Mapping entry (object 0x3003 (Outputs), entry 0x11 (Data Out 15))	UINT32	RO	0x3003:11, 8
1602:12	SubIndex 0x018	18. PDO Mapping entry (object 0x3003 (Outputs), entry 0x12 (Data Out 16))	UINT32	RO	0x3003:12, 8
1602:13	SubIndex 0x019	19. PDO Mapping entry (object 0x3003 (Outputs), entry 0x13 (Data Out 17))	UINT32	RO	0x3003:13, 8
1602:14	SubIndex 0x020	20. PDO Mapping entry (object 0x3003 (Outputs), entry 0x14 (Data Out 18))	UINT32	RO	0x3003:14, 8
1602:15	SubIndex 0x021	21. PDO Mapping entry (object 0x3003 (Outputs), entry 0x15 (Data Out 19))	UINT32	RO	0x3003:15, 8
1602:16	SubIndex 0x022	22. PDO Mapping entry (object 0x3003 (Outputs), entry 0x16 (Data Out 20))	UINT32	RO	0x3003:16, 8
1602:17	SubIndex 0x023	23. PDO Mapping entry (object 0x3003 (Outputs), entry 0x17 (Data Out 21))	UINT32	RO	0x3003:17, 8

**Index 0x1604 COM RxPDO-Map Outputs [ab Hardwarestand 03]**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1604:0	COM RxPDO-Map Outputs	PDO Mapping RxPDO 5	UINT8	RO	0x1C (28 <sub>dez</sub> )
1604:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x01 (Transmit request))	UINT32	RO	0x7000:01, 1
1604:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x02 (Receive accepted))	UINT32	RO	0x7000:02, 1
1604:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x03 (Init request))	UINT32	RO	0x7000:03, 1
1604:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x04 (Send continues))	UINT32	RO	0x7000:04, 1
1604:05	SubIndex 0x005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1604:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x09 (Output length))	UINT32	RO	0x7000:09, 8
1604:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x11 (Data Out 0))	UINT32	RO	0x7000:11, 8
1604:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x12 (Data Out 1))	UINT32	RO	0x7000:12, 8
1604:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x13 (Data Out 2))	UINT32	RO	0x7000:13, 8
1604:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x14 (Data Out 3))	UINT32	RO	0x7000:14, 8
1604:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x15 (Data Out 4))	UINT32	RO	0x7000:15, 8
1604:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x16 (Data Out 5))	UINT32	RO	0x7000:16, 8
1604:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x17 (Data Out 6))	UINT32	RO	0x7000:17, 8
1604:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x18 (Data Out 7))	UINT32	RO	0x7000:18, 8
1604:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x19 (Data Out 8))	UINT32	RO	0x7000:19, 8
1604:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1A (Data Out 9))	UINT32	RO	0x7000:1A, 8
1604:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1B (Data Out 10))	UINT32	RO	0x7000:1B, 8
1604:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1C (Data Out 11))	UINT32	RO	0x7000:1C, 8
1604:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1D (Data Out 12))	UINT32	RO	0x7000:1D, 8
1604:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1E (Data Out 13))	UINT32	RO	0x7000:1E, 8
1604:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x1F (Data Out 14))	UINT32	RO	0x7000:1F, 8
1604:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x20 (Data Out 15))	UINT32	RO	0x7000:20, 8
1604:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x21 (Data Out 16))	UINT32	RO	0x7000:21, 8
1604:18	SubIndex 0x024	24. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x22 (Data Out 17))	UINT32	RO	0x7000:22, 8
1604:19	SubIndex 0x025	25. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x23 (Data Out 18))	UINT32	RO	0x7000:23, 8
1604:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x24 (Data Out 19))	UINT32	RO	0x7000:24, 8
1604:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x25 (Data Out 20))	UINT32	RO	0x7000:25, 8
1604:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x7000 (COM Outputs), entry 0x26 (Data Out 21))	UINT32	RO	0x7000:26, 8

**Index 0x1605 COM ext. outputs**

Index (hex)	Name	Bedeutung	Data type	Flags	Default
1605:0	COM ext. outputs	PDO Mapping RxPDO 6	UINT8	RO	0x38 (56 <sub>dez</sub> )
1605:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x01 (Transmit request))	UINT32	RO	0x7001:01, 1
1605:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x02 (Receive accepted))	UINT32	RO	0x7001:02, 1
1605:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x03 (Init request))	UINT32	RO	0x7001:03, 1
1605:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x04 (Send continues))	UINT32	RO	0x7001:04, 1
1605:05	SubIndex 0x005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1605:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x09 (Output length))	UINT32	RO	0x7001:09, 8
1605:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x11 (Data Out 0))	UINT32	RO	0x7001:11, 16
1605:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x12 (Data Out 1))	UINT32	RO	0x7001:12, 16
1605:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x13 (Data Out 2))	UINT32	RO	0x7001:13, 16
1605:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x14 (Data Out 3))	UINT32	RO	0x7001:14, 16
1605:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x15 (Data Out 4))	UINT32	RO	0x7001:15, 16
1605:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x16 (Data Out 5))	UINT32	RO	0x7001:16, 16
1605:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x17 (Data Out 6))	UINT32	RO	0x7001:17, 16
1605:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x18 (Data Out 7))	UINT32	RO	0x7001:18, 16
1605:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x19 (Data Out 8))	UINT32	RO	0x7001:19, 16
1605:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1A (Data Out 9))	UINT32	RO	0x7001:1A, 16
1605:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1B (Data Out 10))	UINT32	RO	0x7001:1B, 16
1605:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1C (Data Out 11))	UINT32	RO	0x7001:1C, 16
1605:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1D (Data Out 12))	UINT32	RO	0x7001:1D, 16
1605:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1E (Data Out 13))	UINT32	RO	0x7001:1E, 16
1605:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x1F (Data Out 14))	UINT32	RO	0x7001:1F, 16
1605:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x20 (Data Out 15))	UINT32	RO	0x7001:20, 16
1605:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x21 (Data Out 16))	UINT32	RO	0x7001:21, 16
1605:18	SubIndex 0x024	24. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x22 (Data Out 17))	UINT32	RO	0x7001:22, 16
1605:19	SubIndex 0x025	25. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x23 (Data Out 18))	UINT32	RO	0x7001:23, 16
1605:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x24 (Data Out 19))	UINT32	RO	0x7001:24, 16
1605:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x25 (Data Out 20))	UINT32	RO	0x7001:25, 16
1605:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x26 (Data Out 21))	UINT32	RO	0x7001:26, 16
1605:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x27 (Data Out 22))	UINT32	RO	0x7001:27, 16
1605:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x28 (Data Out 23))	UINT32	RO	0x7001:28, 16
1605:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x29 (Data Out 24))	UINT32	RO	0x7001:29, 16

Index (hex)	Name	Bedeutung	Data type	Flags	Default
1605:20	SubIndex 0x032	32. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2A (Data Out 25))	UINT32	RO	0x7001:2A, 16
1605:21	SubIndex 0x033	33. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2B (Data Out 26))	UINT32	RO	0x7001:2B, 16
1605:22	SubIndex 0x034	34. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2C (Data Out 27))	UINT32	RO	0x7001:2C, 16
1605:23	SubIndex 0x035	35. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2D (Data Out 28))	UINT32	RO	0x7001:2D, 16
1605:24	SubIndex 0x036	36. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2E (Data Out 29))	UINT32	RO	0x7001:2E, 16
1605:25	SubIndex 0x037	37. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x2F (Data Out 30))	UINT32	RO	0x7001:2F, 16
1605:26	SubIndex 0x038	38. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x30 (Data Out 31))	UINT32	RO	0x7001:30, 16
1605:27	SubIndex 0x039	39. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x31 (Data Out 32))	UINT32	RO	0x7001:31, 16
1605:28	SubIndex 0x040	40. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x32 (Data Out 33))	UINT32	RO	0x7001:32, 16
1605:29	SubIndex 0x041	41. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x33 (Data Out 34))	UINT32	RO	0x7001:33, 16
1605:2A	SubIndex 0x042	42. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x34 (Data Out 35))	UINT32	RO	0x7001:34, 16
1605:2B	SubIndex 0x043	43. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x35 (Data Out 36))	UINT32	RO	0x7001:35, 16
1605:2C	SubIndex 0x044	44. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x36 (Data Out 37))	UINT32	RO	0x7001:36, 16
1605:2D	SubIndex 0x045	45. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x37 (Data Out 38))	UINT32	RO	0x7001:37, 16
1605:2E	SubIndex 0x046	46. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x38 (Data Out 39))	UINT32	RO	0x7001:38, 16
1605:2F	SubIndex 0x047	47. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x39 (Data Out 40))	UINT32	RO	0x7001:39, 16
1605:30	SubIndex 0x048	48. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3A (Data Out 41))	UINT32	RO	0x7001:3A, 16
1605:31	SubIndex 0x049	49. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3B (Data Out 42))	UINT32	RO	0x7001:3B, 16
1605:32	SubIndex 0x050	50. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3C (Data Out 43))	UINT32	RO	0x7001:3C, 16
1605:33	SubIndex 0x051	51. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3D (Data Out 44))	UINT32	RO	0x7001:3D, 16
1605:34	SubIndex 0x052	52. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3E (Data Out 45))	UINT32	RO	0x7001:3E, 16
1605:35	SubIndex 0x053	53. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x3F (Data Out 46))	UINT32	RO	0x7001:3F, 16
1605:36	SubIndex 0x054	54. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x40 (Data Out 47))	UINT32	RO	0x7001:40, 16
1605:37	SubIndex 0x055	55. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x41 (Data Out 48))	UINT32	RO	0x7001:41, 16
1605:38	SubIndex 0x056	56. PDO Mapping entry (object 0x7001 (COM ext. outputs), entry 0x42 (Data Out 49))	UINT32	RO	0x7001:42, 16

**Index 0x1800 TxPDO-Par Inputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1800:0	TxPDO-Par Inputs	PDO Parameter TxPDO 1	UINT8	RO	0x06 (6 <sub>dez</sub> )
1800:06	Exclude TxPDOs	Hier sind die TxPDOs (Index 0xder TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 1 übertragen werden dürfen	OCTET-STRING[6]	RO	01 1A 02 1A 04 1A

**Index 0x1801 TxPDO-Par Inputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1801:0	TxPDO-Par Inputs	PDO Parameter TxPDO 2	UINT8	RO	0x06 (6 <sub>dez</sub> )
1801:06	Exclude TxPDOs	Hier sind die TxPDOs (Index 0xder TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 2 übertragen werden dürfen	OCTET-STRING[6]	RO	00 1A 02 1A 04 1A

**Index 0x1802 TxPDO-Par Inputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1802:0	TxPDO-Par Inputs	PDO Parameter TxPDO 3	UINT8	RO	0x06 (6 <sub>dez</sub> )
1802:06	Exclude TxPDOs	Hier sind die TxPDOs (Index 0xder TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 3 übertragen werden dürfen	OCTET-STRING[6]	RO	00 1A 01 1A 04 1A

**Index 0x1804 COM TxPDO-Par Inputs**

Index (hex)	Name	Bedeutung	Data type	Flags	Default
1804:0	COM TxPDO-Par Inputs	PDO Parameter TxPDO 5	UINT8	RO	0x06 (6 <sub>dez</sub> )
1804:06	Exclude TxPDOs	Hier sind die TxPDOs (Index 0xder TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 5 übertragen werden dürfen	OCTET-STRING[8]	RO	00 1A 01 1A 02 1A 05 1A

**Index 0x1805 COM ext. inputs**

Index (hex)	Name	Bedeutung	Data type	Flags	Default
1805:0	COM ext. inputs	PDO Parameter TxPDO 6	UINT8	RO	0x06 (6 <sub>dez</sub> )
1805:06	Exclude TxPDOs	Hier sind die TxPDOs (Index 0xder TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 6 übertragen werden dürfen	OCTET-STRING[8]	RO	00 1A 01 1A 02 1A 04 1A

**Index 0x1A00 TxPDO-Map Inputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A00:0	TxPDO-Map Inputs	PDO Mapping TxPDO 1	UINT8	RO	0x04 (4 <sub>dez</sub> )
1A00:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3101 (Inputs), entry 0x01 (Status))	UINT32	RO	0x3101:01, 8
1A00:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3101 (Inputs), entry 0x02 (Data In 0))	UINT32	RO	0x3101:02, 8
1A00:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3101 (Inputs), entry 0x03 (Data In 1))	UINT32	RO	0x3101:03, 8
1A00:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3101 (Inputs), entry 0x04 (Data In 2))	UINT32	RO	0x3101:04, 8

**Index 0x1A01 TxPDO-Map Inputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A01:0	TxPDO-Map Inputs	PDO Mapping TxPDO 2	UINT8	RO	0x06 (6 <sub>dez</sub> )
1A01:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3102 (Inputs), entry 0x01 (Status))	UINT32	RO	0x3102:01, 8
1A01:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3102 (Inputs), entry 0x02 (Data In 0))	UINT32	RO	0x3102:02, 8
1A01:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3102 (Inputs), entry 0x03 (Data In 1))	UINT32	RO	0x3102:03, 8
1A01:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3102 (Inputs), entry 0x04 (Data In 2))	UINT32	RO	0x3102:04, 8
1A01:05	SubIndex 0x005	5. PDO Mapping entry (object 0x3102 (Inputs), entry 0x05 (Data In 3))	UINT32	RO	0x3102:05, 8
1A01:06	SubIndex 0x006	6. PDO Mapping entry (object 0x3102 (Inputs), entry 0x06 (Data In 4))	UINT32	RO	0x3102:06, 8

**Index 0x1A02 TxPDO-Map Inputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A02:0	TxPDO-Map Inputs	PDO Mapping TxPDO 3	UINT8	RO	0x17 (23 <sub>dez</sub> )
1A02:01	SubIndex 0x001	1. PDO Mapping entry (object 0x3103 (Inputs), entry 0x01 (Status))	UINT32	RO	0x3103:01, 16
1A02:02	SubIndex 0x002	2. PDO Mapping entry (object 0x3103 (Inputs), entry 0x02 (Data In 0))	UINT32	RO	0x3103:02, 8
1A02:03	SubIndex 0x003	3. PDO Mapping entry (object 0x3103 (Inputs), entry 0x03 (Data In 1))	UINT32	RO	0x3103:03, 8
1A02:04	SubIndex 0x004	4. PDO Mapping entry (object 0x3103 (Inputs), entry 0x04 (Data In 2))	UINT32	RO	0x3103:04, 8
1A02:05	SubIndex 0x005	5. PDO Mapping entry (object 0x3103 (Inputs), entry 0x05 (Data In 3))	UINT32	RO	0x3103:05, 8
1A02:06	SubIndex 0x006	6. PDO Mapping entry (object 0x3103 (Inputs), entry 0x06 (Data In 4))	UINT32	RO	0x3103:06, 8
1A02:07	SubIndex 0x007	7. PDO Mapping entry (object 0x3103 (Inputs), entry 0x07 (Data In 5))	UINT32	RO	0x3103:07, 8
1A02:08	SubIndex 0x008	8. PDO Mapping entry (object 0x3103 (Inputs), entry 0x08 (Data In 6))	UINT32	RO	0x3103:08, 8
1A02:09	SubIndex 0x009	9. PDO Mapping entry (object 0x3103 (Inputs), entry 0x09 (Data In 7))	UINT32	RO	0x3103:09, 8
1A02:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0A (Data In 8))	UINT32	RO	0x3103:0A, 8
1A02:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0B (Data In 9))	UINT32	RO	0x3103:0B, 8
1A02:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0C (Data In 10))	UINT32	RO	0x3103:0C, 8
1A02:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0D (Data In 11))	UINT32	RO	0x3103:0D, 8
1A02:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0E (Data In 12))	UINT32	RO	0x3103:0E, 8
1A02:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x3103 (Inputs), entry 0x0F (Data In 13))	UINT32	RO	0x3103:0F, 8
1A02:10	SubIndex 0x016	16. PDO Mapping entry (object 0x3103 (Inputs), entry 0x10 (Data In 14))	UINT32	RO	0x3103:10, 8
1A02:11	SubIndex 0x017	17. PDO Mapping entry (object 0x3103 (Inputs), entry 0x11 (Data In 15))	UINT32	RO	0x3103:11, 8
1A02:12	SubIndex 0x018	18. PDO Mapping entry (object 0x3103 (Inputs), entry 0x12 (Data In 16))	UINT32	RO	0x3103:12, 8
1A02:13	SubIndex 0x019	19. PDO Mapping entry (object 0x3103 (Inputs), entry 0x13 (Data In 17))	UINT32	RO	0x3103:13, 8
1A02:14	SubIndex 0x020	20. PDO Mapping entry (object 0x3103 (Inputs), entry 0x14 (Data In 18))	UINT32	RO	0x3103:14, 8
1A02:15	SubIndex 0x021	21. PDO Mapping entry (object 0x3103 (Inputs), entry 0x15 (Data In 19))	UINT32	RO	0x3103:15, 8
1A02:16	SubIndex 0x022	22. PDO Mapping entry (object 0x3103 (Inputs), entry 0x16 (Data In 20))	UINT32	RO	0x3103:16, 8
1A02:17	SubIndex 0x023	23. PDO Mapping entry (object 0x3103 (Inputs), entry 0x17 (Data In 21))	UINT32	RO	0x3103:17, 8

**Index 0x1A04 COM TxPDO-Map Inputs [ab Hardwarestand 03]**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A04:0	COM TxPDO-Map Inputs	PDO Mapping TxPDO 5	UINT8	RO	0x1F (31 <sub>dez</sub> )
1A04:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x01 (Transmit accepted))	UINT32	RO	0x6000:01, 1
1A04:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x02 (Receive request))	UINT32	RO	0x6000:02, 1
1A04:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x03 (Init accepted))	UINT32	RO	0x6000:03, 1
1A04:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x04 (Buffer full))	UINT32	RO	0x6000:04, 1
1A04:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x05 (Parity error))	UINT32	RO	0x6000:05, 1
1A04:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x06 (Framing error))	UINT32	RO	0x6000:06, 1
1A04:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x07 (Overrun error))	UINT32	RO	0x6000:07, 1
1A04:08	SubIndex 0x008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A04:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x09 (Input length))	UINT32	RO	0x6000:09, 8
1A04:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x11 (Data In 0))	UINT32	RO	0x6000:11, 8
1A04:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x12 (Data In 1))	UINT32	RO	0x6000:12, 8
1A04:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x13 (Data In 2))	UINT32	RO	0x6000:13, 8
1A04:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x14 (Data In 3))	UINT32	RO	0x6000:14, 8
1A04:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x15 (Data In 4))	UINT32	RO	0x6000:15, 8
1A04:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x16 (Data In 5))	UINT32	RO	0x6000:16, 8
1A04:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x17 (Data In 6))	UINT32	RO	0x6000:17, 8
1A04:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x18 (Data In 7))	UINT32	RO	0x6000:18, 8
1A04:12	SubIndex 0x018	18. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x19 (Data In 8))	UINT32	RO	0x6000:19, 8
1A04:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1A (Data In 9))	UINT32	RO	0x6000:1A, 8
1A04:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1B (Data In 10))	UINT32	RO	0x6000:1B, 8
1A04:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1C (Data In 11))	UINT32	RO	0x6000:1C, 8
1A04:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1D (Data In 12))	UINT32	RO	0x6000:1D, 8
1A04:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1E (Data In 13))	UINT32	RO	0x6000:1E, 8
1A04:18	SubIndex 0x024	24. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x1F (Data In 14))	UINT32	RO	0x6000:1F, 8
1A04:19	SubIndex 0x025	25. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x20 (Data In 15))	UINT32	RO	0x6000:20, 8
1A04:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x21 (Data In 16))	UINT32	RO	0x6000:21, 8
1A04:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x22 (Data In 17))	UINT32	RO	0x6000:22, 8
1A04:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x23 (Data In 18))	UINT32	RO	0x6000:23, 8
1A04:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x24 (Data In 19))	UINT32	RO	0x6000:24, 8
1A04:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x25 (Data In 20))	UINT32	RO	0x6000:25, 8
1A04:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x6000 (COM Inputs), entry 0x26 (Data In 21))	UINT32	RO	0x6000:26, 8

## Index 0x1A05 COM ext. inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	COM ext. inputs	PDO Mapping TxPDO 6	UINT8	RO	0x3B (59 <sub>dez</sub> )
1A05:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x01 (Transmit accepted))	UINT32	RO	0x6001:01, 1
1A05:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x02 (Receive request))	UINT32	RO	0x6001:02, 1
1A05:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x03 (Init accepted))	UINT32	RO	0x6001:03, 1
1A05:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x04 (Buffer full))	UINT32	RO	0x6001:04, 1
1A05:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x05 (Parity error))	UINT32	RO	0x6001:05, 1
1A05:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x06 (Framing error))	UINT32	RO	0x6001:06, 1
1A05:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x07 (Overrun error))	UINT32	RO	0x6001:07, 1
1A05:08	SubIndex 0x008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A05:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x09 (Input length))	UINT32	RO	0x6001:09, 8
1A05:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x11 (Data In 0))	UINT32	RO	0x6001:11, 16
1A05:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x12 (Data In 1))	UINT32	RO	0x6001:12, 16
1A05:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x13 (Data In 2))	UINT32	RO	0x6001:13, 16
1A05:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x14 (Data In 3))	UINT32	RO	0x6001:14, 16
1A05:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x15 (Data In 4))	UINT32	RO	0x6001:15, 16
1A05:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x16 (Data In 5))	UINT32	RO	0x6001:16, 16
1A05:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x17 (Data In 6))	UINT32	RO	0x6001:17, 16
1A05:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x18 (Data In 7))	UINT32	RO	0x6001:18, 16
1A05:12	SubIndex 0x018	18. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x19 (Data In 8))	UINT32	RO	0x6001:19, 16
1A05:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1A (Data In 9))	UINT32	RO	0x6001:1A, 16
1A05:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1B (Data In 10))	UINT32	RO	0x6001:1B, 16
1A05:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1C (Data In 11))	UINT32	RO	0x6001:1C, 16
1A05:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1D (Data In 12))	UINT32	RO	0x6001:1D, 16
1A05:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1E (Data In 13))	UINT32	RO	0x6001:1E, 16
1A05:18	SubIndex 0x024	24. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x1F (Data In 14))	UINT32	RO	0x6001:1F, 16
1A05:19	SubIndex 0x025	25. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x20 (Data In 15))	UINT32	RO	0x6001:20, 16
1A05:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x21 (Data In 16))	UINT32	RO	0x6001:21, 16
1A05:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x22 (Data In 17))	UINT32	RO	0x6001:22, 16
1A05:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x23 (Data In 18))	UINT32	RO	0x6001:23, 16
1A05:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x24 (Data In 19))	UINT32	RO	0x6001:24, 16
1A05:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x25 (Data In 20))	UINT32	RO	0x6001:25, 16
1A05:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x26 (Data In 21))	UINT32	RO	0x6001:26, 16

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:20	SubIndex 0x032	32. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x27 (Data In 22))	UINT32	RO	0x6001:27, 16
1A05:21	SubIndex 0x033	33. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x28 (Data In 23))	UINT32	RO	0x6001:28, 16
1A05:22	SubIndex 0x034	34. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x29 (Data In 24))	UINT32	RO	0x6001:29, 16
1A05:23	SubIndex 0x035	35. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2A (Data In 25))	UINT32	RO	0x6001:2A, 16
1A05:24	SubIndex 0x036	36. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2B (Data In 26))	UINT32	RO	0x6001:2B, 16
1A05:25	SubIndex 0x037	37. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2C (Data In 27))	UINT32	RO	0x6001:2C, 16
1A05:26	SubIndex 0x038	38. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2D (Data In 28))	UINT32	RO	0x6001:2D, 16
1A05:27	SubIndex 0x039	39. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2E (Data In 29))	UINT32	RO	0x6001:2E, 16
1A05:28	SubIndex 0x040	40. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x2F (Data In 30))	UINT32	RO	0x6001:2F, 16
1A05:29	SubIndex 0x041	41. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x30 (Data In 31))	UINT32	RO	0x6001:30, 16
1A05:2A	SubIndex 0x042	42. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x31 (Data In 32))	UINT32	RO	0x6001:31, 16
1A05:2B	SubIndex 0x043	43. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x32 (Data In 33))	UINT32	RO	0x6001:32, 16
1A05:2C	SubIndex 0x044	44. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x33 (Data In 34))	UINT32	RO	0x6001:33, 16
1A05:2D	SubIndex 0x045	45. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x34 (Data In 35))	UINT32	RO	0x6001:34, 16
1A05:2E	SubIndex 0x046	46. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x35 (Data In 36))	UINT32	RO	0x6001:35, 16
1A05:2F	SubIndex 0x047	47. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x36 (Data In 37))	UINT32	RO	0x6001:36, 16
1A05:30	SubIndex 0x048	48. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x37 (Data In 38))	UINT32	RO	0x6001:37, 16
1A05:31	SubIndex 0x049	49. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x38 (Data In 39))	UINT32	RO	0x6001:38, 16
1A05:32	SubIndex 0x050	50. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x39 (Data In 40))	UINT32	RO	0x6001:39, 16
1A05:33	SubIndex 0x051	51. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3A (Data In 41))	UINT32	RO	0x6001:3A, 16
1A05:34	SubIndex 0x052	52. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3B (Data In 42))	UINT32	RO	0x6001:3B, 16
1A05:35	SubIndex 0x053	53. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3C (Data In 43))	UINT32	RO	0x6001:3C, 16
1A05:36	SubIndex 0x054	54. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3D (Data In 44))	UINT32	RO	0x6001:3D, 16
1A05:37	SubIndex 0x055	55. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3E (Data In 45))	UINT32	RO	0x6001:3E, 16
1A05:38	SubIndex 0x056	56. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x3F (Data In 46))	UINT32	RO	0x6001:3F, 16
1A05:39	SubIndex 0x057	57. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x40 (Data In 47))	UINT32	RO	0x6001:40, 16
1A05:3A	SubIndex 0x058	58. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x41 (Data In 48))	UINT32	RO	0x6001:41, 16
1A05:3B	SubIndex 0x059	59. PDO Mapping entry (object 0x6001 (COM ext. inputs), entry 0x42 (Data In 49))	UINT32	RO	0x6001:42, 16

**Index 0x1C00 Sync manager type**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C00:0	Sync manager type	Benutzung der Sync Manager	UINT8	RO	0x04 (4 <sub>dez</sub> )
1C00:01	SubIndex 0x001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1 <sub>dez</sub> )
1C00:02	SubIndex 0x002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2 <sub>dez</sub> )
1C00:03	SubIndex 0x003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3 <sub>dez</sub> )
1C00:04	SubIndex 0x004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4 <sub>dez</sub> )

**Index 0x1C12 RxPDO assign**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x01 (1 <sub>dez</sub> )
1C12:01	SubIndex 0x001	1. zugeordnete RxPDO (enthält den Index 0xdes zugehörigen RxPDO Mapping Objekts)	UINT16	RW	0x1602 (5634 <sub>dez</sub> )
1C12:02	SubIndex 0x002	2. zugeordnete RxPDO (enthält den Index 0xdes zugehörigen RxPDO Mapping Objekts)	UINT16	RW	0x0000 (0 <sub>dez</sub> )

**Index 0x1C13 TxPDO assign**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x01 (1 <sub>dez</sub> )
1C13:01	SubIndex 0x001	1. zugeordnete TxPDO (enthält den Index 0xdes zugehörigen TxPDO Mapping Objekts)	UINT16	RW	0x1A02 (6658 <sub>dez</sub> )
1C13:02	SubIndex 0x002	2. zugeordnete TxPDO (enthält den Index 0xdes zugehörigen TxPDO Mapping Objekts)	UINT16	RW	0x0000 (0 <sub>dez</sub> )

**Index 0x1C32 SM output parameter [ab Hardwarestand 03]**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C32:0	SM output parameter	Synchronisierungsparameter der Outputs	UINT8	RO	0x20 (32 <sub>dez</sub> )
1C32:01	Sync mode	Aktuelle Synchronisierungsbetriebsart: <ul style="list-style-type: none"> <li>0: Free Run</li> <li>1: Synchron with SM 2 Event</li> <li>2: DC-Mode - Synchron with SYNC0 Event</li> <li>3: DC-Mode - Synchron with SYNC1 Event</li> </ul>	UINT16	RW	0x0001 (1 <sub>dez</sub> )
1C32:02	Cycle time	Zykluszeit (in ns): <ul style="list-style-type: none"> <li>Free Run: Zykluszeit des lokalen Timers</li> <li>Synchron with SM 2 Event: Zykluszeit des Masters</li> <li>DC-Mode: SYNC0/SYNC1 Cycle Time</li> </ul>	UINT32	RW	0x0007A120 (500000 <sub>dez</sub> )
1C32:03	Shift time	Zeit zwischen SYNC0 Event und Ausgabe der Outputs (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C32:04	Sync modes supported	Unterstützte Synchronisierungsbetriebsarten: <ul style="list-style-type: none"> <li>Bit 0 = 1: Free Run wird unterstützt</li> <li>Bit 1 = 1: Synchron with SM 2 Event wird unterstützt</li> <li>Bit 2-3 = 01: DC-Mode wird unterstützt</li> <li>Bit 4-5 = 10: Output Shift mit SYNC1 Event (nur DC-Mode)</li> <li>Bit 14 = 1: dynamische Zeiten (Messen durch Beschreiben von 1C32:08 [▶ 166])</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dez</sub> )
1C32:05	Minimum cycle time	Minimale Zykluszeit (in ns)	UINT32	RO	0x00004E20 (20000 <sub>dez</sub> )

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C32:06	Calc and copy time	Minimale Zeit zwischen SYNC0 und SYNC1 Event (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C32:08	Command	<ul style="list-style-type: none"> <li>0: Messung der lokalen Zykluszeit wird gestoppt</li> <li>1: Messung der lokalen Zykluszeit wird gestartet</li> </ul> Die Entries <a href="#">1C32:03</a> , <a href="#">1C32:05</a> , <a href="#">1C32:06</a> , <a href="#">1C32:09</a> [ <a href="#">▶ 166</a> ], <a href="#">1C33:03</a> , <a href="#">1C33:06</a> , <a href="#">1C33:09</a> [ <a href="#">▶ 167</a> ] werden mit den maximal gemessenen Werten aktualisiert. Wenn erneut gemessen wird, werden die Messwerte zurückgesetzt	UINT16	RW	0x0000 (0 <sub>dez</sub> )
1C32:09	Delay time	Zeit zwischen SYNC1 Event und Ausgabe der Outputs (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C32:0B	SM event missed counter	Anzahl der ausgefallenen SM-Events im OPERATIONAL (nur im DC Mode)	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C32:0C	Cycle exceeded counter	Anzahl der Zykluszeitverletzungen im OPERATIONAL (Zyklus wurde nicht rechtzeitig fertig bzw. der nächste Zyklus kam zu früh)	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C32:0D	Shift too short counter	Anzahl der zu kurzen Abstände zwischen SYNC0 und SYNC1 Event (nur im DC Mode)	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C32:20	Sync error	Im letzten Zyklus war die Synchronisierung nicht korrekt (Ausgänge wurden zu spät ausgegeben, nur im DC Mode)	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )

**Index 0x1C33 SM input parameter [ab Hardwarestand 03]**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C33:0	SM input parameter	Synchronisierungsparameter der Inputs	UINT8	RO	0x20 (32 <sub>dez</sub> )
1C33:01	Sync mode	Aktuelle Synchronisierungsbetriebsart: <ul style="list-style-type: none"> <li>0: Free Run</li> <li>1: Synchron with SM 3 Event (keine Outputs vorhanden)</li> <li>2: DC - Synchron with SYNC0 Event</li> <li>3: DC - Synchron with SYNC1 Event</li> <li>34: Synchron with SM 2 Event (Outputs vorhanden)</li> </ul>	UINT16	RW	0x0022 (34 <sub>dez</sub> )
1C33:02	Cycle time	wie <a href="#">1C32:02</a> [ <a href="#">▶ 166</a> ]	UINT32	RW	0x0007A120 (500000 <sub>dez</sub> )
1C33:03	Shift time	Zeit zwischen SYNC0-Event und Einlesen der Inputs (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C33:04	Sync modes supported	Unterstützte Synchronisierungsbetriebsarten: <ul style="list-style-type: none"> <li>Bit 0: Free Run wird unterstützt</li> <li>Bit 1: Synchron with SM 2 Event wird unterstützt (Outputs vorhanden)</li> <li>Bit 1: Synchron with SM 3 Event wird unterstützt (keine Outputs vorhanden)</li> <li>Bit 2-3 = 01: DC-Mode wird unterstützt</li> <li>Bit 4-5 = 01: Input Shift durch lokales Ereignis (Outputs vorhanden)</li> <li>Bit 4-5 = 10: Input Shift mit SYNC1 Event (keine Outputs vorhanden)</li> <li>Bit 14 = 1: dynamische Zeiten (Messen durch Beschreiben von <a href="#">1C32:08</a> [<a href="#">▶ 166</a>] oder <a href="#">1C33:08</a> [<a href="#">▶ 167</a>])</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dez</sub> )
1C33:05	Minimum cycle time	wie <a href="#">1C32:05</a> [ <a href="#">▶ 166</a> ]	UINT32	RO	0x00004E20 (20000 <sub>dez</sub> )
1C33:06	Calc and copy time	Zeit zwischen Einlesen der Eingänge und Verfügbarkeit der Eingänge für den Master (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C33:08	Command	wie <a href="#">1C32:08</a> [ <a href="#">▶ 166</a> ]	UINT16	RW	0x0000 (0 <sub>dez</sub> )

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C33:09	Delay time	Zeit zwischen SYNC1-Event und Einlesen der Eingänge (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C33:0B	SM event missed counter	wie <a href="#">1C32:11</a> [ <a href="#">▶ 166</a> ]	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C33:0C	Cycle exceeded counter	wie <a href="#">1C32:12</a> [ <a href="#">▶ 166</a> ]	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C33:0D	Shift too short counter	wie <a href="#">1C32:13</a> [ <a href="#">▶ 166</a> ]	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C33:20	Sync error	wie <a href="#">1C32:32</a> [ <a href="#">▶ 166</a> ]	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )

**Index 0x3001 Outputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
3001:0	Outputs	Länge dieses Objekts	UINT8	RO	0x04 (4 <sub>dez</sub> )
3001:01	Ctrl	<a href="#">Control-Byte</a> [ <a href="#">▶ 173</a> ]	UINT8	RO	0x00 (0 <sub>dez</sub> )
3001:02	Data Out 0	Ausgangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )
3001:03	Data Out 1	Ausgangs-Byte 1	UINT8	RO	0x00 (0 <sub>dez</sub> )
3001:04	Data Out 2	Ausgangs-Byte 2	UINT8	RO	0x00 (0 <sub>dez</sub> )

**Index 0x3002 Outputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
3002:0	Outputs	Länge dieses Objekts	UINT8	RO	0x06 (6 <sub>dez</sub> )
3002:01	Ctrl	<a href="#">Control-Byte</a> [ <a href="#">▶ 173</a> ]	UINT8	RO	0x00 (0 <sub>dez</sub> )
3002:02	Data Out 0	Ausgangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )
3002:03	Data Out 1	Ausgangs-Byte 1	UINT8	RO	0x00 (0 <sub>dez</sub> )
3002:04	Data Out 2	Ausgangs-Byte 2	UINT8	RO	0x00 (0 <sub>dez</sub> )
3002:05	Data Out 3	Ausgangs-Byte 3	UINT8	RO	0x00 (0 <sub>dez</sub> )
3002:06	Data Out 4	Ausgangs-Byte 4	UINT8	RO	0x00 (0 <sub>dez</sub> )

**Index 0x3003 Outputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
3003:0	Outputs	Länge dieses Objekts	UINT8	RO	0x17 (23 <sub>dez</sub> )
3003:01	Ctrl	Control-Wort [► 173]	UINT16	RO	0x0000 (0 <sub>dez</sub> )
3003:02	Data Out 0	Ausgangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:03	Data Out 1	Ausgangs-Byte 1	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:04	Data Out 2	Ausgangs-Byte 2	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:05	Data Out 3	Ausgangs-Byte 3	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:06	Data Out 4	Ausgangs-Byte 4	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:07	Data Out 5	Ausgangs-Byte 5	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:08	Data Out 6	Ausgangs-Byte 6	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:09	Data Out 7	Ausgangs-Byte 7	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:0A	Data Out 8	Ausgangs-Byte 8	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:0B	Data Out 9	Ausgangs-Byte 9	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:0C	Data Out 10	Ausgangs-Byte 10	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:0D	Data Out 11	Ausgangs-Byte 11	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:0E	Data Out 12	Ausgangs-Byte 12	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:0F	Data Out 13	Ausgangs-Byte 13	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:10	Data Out 14	Ausgangs-Byte 14	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:11	Data Out 15	Ausgangs-Byte 15	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:12	Data Out 16	Ausgangs-Byte 16	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:13	Data Out 17	Ausgangs-Byte 17	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:14	Data Out 18	Ausgangs-Byte 18	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:15	Data Out 19	Ausgangs-Byte 19	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:16	Data Out 20	Ausgangs-Byte 20	UINT8	RO	0x00 (0 <sub>dez</sub> )
3003:17	Data Out 21	Ausgangs-Byte 21	UINT8	RO	0x00 (0 <sub>dez</sub> )

**Index 0x3101 Inputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
3101:0	Inputs	Länge dieses Objekts	UINT8	RO	0x04 (4 <sub>dez</sub> )
3101:01	Status	Status-Byte [► 173]	UINT8	RO	0x00 (0 <sub>dez</sub> )
3101:02	Data In 0	Eingangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )
3101:03	Data In 1	Eingangs-Byte 1	UINT8	RO	0x00 (0 <sub>dez</sub> )
3101:04	Data In 2	Eingangs-Byte 2	UINT8	RO	0x00 (0 <sub>dez</sub> )

**Index 0x3102 Inputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
3102:0	Inputs	Länge dieses Objekts	UINT8	RO	0x06 (6 <sub>dez</sub> )
3102:01	Status	Status-Byte [► 173]	UINT8	RO	0x00 (0 <sub>dez</sub> )
3102:02	Data In 0	Eingangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )
3102:03	Data In 1	Eingangs-Byte 1	UINT8	RO	0x00 (0 <sub>dez</sub> )
3102:04	Data In 2	Eingangs-Byte 2	UINT8	RO	0x00 (0 <sub>dez</sub> )
3102:05	Data In 3	Eingangs-Byte 3	UINT8	RO	0x00 (0 <sub>dez</sub> )
3102:06	Data In 4	Eingangs-Byte 4	UINT8	RO	0x00 (0 <sub>dez</sub> )

**Index 0x3103 Inputs**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
3103:0	Inputs	Länge dieses Objekts	UINT8	RO	0x17 (23 <sub>dez</sub> )
3103:01	Status	Status-Wort [► 173]	UINT16	RO	0x0000 (0 <sub>dez</sub> )
3103:02	Data In 0	Eingangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:03	Data In 1	Eingangs-Byte 1	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:04	Data In 2	Eingangs-Byte 2	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:05	Data In 3	Eingangs-Byte 3	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:06	Data In 4	Eingangs-Byte 4	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:07	Data In 5	Eingangs-Byte 5	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:08	Data In 6	Eingangs-Byte 6	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:09	Data In 7	Eingangs-Byte 7	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:0A	Data In 8	Eingangs-Byte 8	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:0B	Data In 9	Eingangs-Byte 9	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:0C	Data In 10	Eingangs-Byte 10	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:0D	Data In 11	Eingangs-Byte 11	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:0E	Data In 12	Eingangs-Byte 12	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:0F	Data In 13	Eingangs-Byte 13	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:10	Data In 14	Eingangs-Byte 14	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:11	Data In 15	Eingangs-Byte 15	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:12	Data In 16	Eingangs-Byte 16	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:13	Data In 17	Eingangs-Byte 17	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:14	Data In 18	Eingangs-Byte 18	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:15	Data In 19	Eingangs-Byte 19	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:16	Data In 20	Eingangs-Byte 20	UINT8	RO	0x00 (0 <sub>dez</sub> )
3103:17	Data In 21	Eingangs-Byte 21	UINT8	RO	0x00 (0 <sub>dez</sub> )

**Index 0x4070 Data bytes in send buffer**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
4070:0	Data bytes in send buffer	Anzahl der Datenbytes im Sende-FIFO	UINT16	RO	0x0000 (0 <sub>dez</sub> )

**Index 0x4071 Data bytes in receive buffer**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
4071:0	Data bytes in receive buffer	Anzahl der Datenbytes im Empfangs-FIFO	UINT16	RO	0x0000 (0 <sub>dez</sub> )

**Index 0x4072 Diagnostic**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
4072:0	Diagnostic	Länge dieses Objekts	UINT8	RO	0x05 (5 <sub>dez</sub> )
4072:01	Buffer overflow	Es ist ein Buffer-Overflow aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
4072:02	Parity error	Es ist ein Parity-Error aufgetreten. Das betroffene Datum wird nicht in das Empfangs-FIFO der Klemme geladen und geht verloren.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
4072:03	Framing error	Es ist ein Framing-Error aufgetreten. Das betroffene Datum wird nicht in das Empfangs-FIFO der Klemme geladen und geht verloren.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
4072:04	Overrun error	Es ist ein Overrun-Error aufgetreten. Das betroffene Datum wird nicht in das Empfangs-FIFO der Klemme geladen und geht verloren.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
4072:05	Buffer full	Das Empfangs-FIFO ist voll. Alle ab jetzt eingehenden Daten gehen verloren!	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )

### 6.1.3 Profilspezifische Objekte (0x6000-0xFFFF) [ab Hardwarestand 03]

Die profilspezifischen Objekte haben für alle EtherCAT Slaves, die das Profil 5001 unterstützen, die gleiche Bedeutung.

#### Index 0x6000 COM Inputs

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
6000:0	COM Inputs	Max. SubIndex (hex)	UINT8	RO	0x26 (38 <sub>dez</sub> )
6000:01	Transmit accepted	Die Klemme quittiert die Entgegennahme von Daten mit Zustandsänderung dieses Bits. Erst daraufhin werden neue Daten von der Steuerung zur Klemme übertragen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6000:02	Receive request	Über eine Zustandsänderung dieses Bits teilt die Klemme der Steuerung mit, dass sich die in "Input length [▶_171]" angezeigte Anzahl von Bytes in den DataIn-Bytes befinden. Die Steuerung muss die Entgegennahme der Daten mit Zustandsänderung des <i>ReceiveAccepted</i> [▶_172]-Bits quittieren. Erst daraufhin werden neue Daten von der Klemme zur Steuerung übertragen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6000:03	Init accepted	0 Die Klemme ist wieder für den seriellen Datenaustausch bereit.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
		1 Die Initialisierung wurde von der Klemme ausgeführt.			
6000:04	Buffer full	Das Empfangs-FIFO ist voll. Ab diesem Zeitpunkt gehen alle eingehenden Daten verloren!	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6000:05	Parity error	Es ist ein Parity-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6000:06	Framing error	Es ist ein Framing-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6000:07	Overrun error	Es ist ein Overrun-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6000:09	Input length	Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.	UINT8	RO	0x00 (0 <sub>dez</sub> )
6000:11	Data In 0	Eingangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )
...	...	....	...	...	...
6000:26	Data In 21	Eingangs-Byte 21	UINT8	RO	0x00 (0 <sub>dez</sub> )

#### Index 0x6001 COM ext. inputs

Index (hex)	Name	Bedeutung	Data type	Flags	Default
6001:0	COM ext. inputs		UINT8	RO	0x00 (0 <sub>dez</sub> )
6001:01	Transmit accepted	identisch zu Index 0x6000	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6001:02	Receive request		BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6001:03	Init accepted		BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6001:04	Buffer full		BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6001:05	Parity error		BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6001:06	Framing error		BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6001:07	Overrun error		BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
6001:09	Input length		UINT8	RO	0x00 (0 <sub>dez</sub> )
6001:11	Data In 0		UINT16	RO	0x0000 (0 <sub>dez</sub> )
	....				
6001:42	Data In 49		UINT16	RO	0x0000 (0 <sub>dez</sub> )

## Index 0x7000 COM Outputs

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
7000:0	COM Outputs	Max. SubIndex (hex)	UINT8	RO	0x26 (38 <sub>dez</sub> )
7000:01	Transmit request	Über eine Zustandsänderung dieses Bits teilt die Steuerung der Klemme mit, dass sich die in mit den " <a href="#">Output length</a> [ <a href="#">▶ 172</a> ]" angezeigte Anzahl von Bytes in den DataOut-Bytes befinden. Die Klemme quittiert die Entgegennahme der Daten mit Zustandsänderung des Bits " <a href="#">TransmitAccepted</a> [ <a href="#">▶ 171</a> ]" . Erst daraufhin werden neue Daten von der Steuerung zur Klemme übertragen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
7000:02	Receive accepted	Die Steuerung quittiert die Entgegennahme von Daten mit Zustandsänderung dieses Bits. Erst daraufhin werden neue Daten von der Klemme zur Steuerung übertragen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
7000:03	Init request	0 Die Steuerung fordert von der Klemme wieder die Bereitschaft für den seriellen Datenaustausch. 1 Die Steuerung fordert die Klemme zur Initialisierung auf. Die Sende- und Empfangsfunktionen werden gesperrt, die FIFO-Zeiger werden zurückgesetzt und die Schnittstelle wird mit den Werten des zuständigen Settings-Objektes initialisiert. Die Ausführung der Initialisierung wird von der Klemme mit dem Bit " <a href="#">Init accepted</a> [ <a href="#">▶ 171</a> ]" quittiert.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
7000:04	Send continuous	Kontinuierliches Senden der Daten aus dem FIFO. Über die Steuerung wird der Sendebuffer gefüllt (bis zu 128 Byte). Mit steigender Flanke des Bits wird der gefüllte Buffer-Inhalt gesendet. Sind die Daten übertragen, so wird dies durch das Setzen des Bits " <a href="#">Init accepted</a> [ <a href="#">▶ 171</a> ]" von der Klemme an die Steuerung quittiert. " <a href="#">Init accepted</a> [ <a href="#">▶ 171</a> ]" wird mit " <a href="#">SendContinuous</a> [ <a href="#">▶ 172</a> ]" zurückgenommen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
7000:09	Output length	Anzahl der Ausgangs-Bytes, die für die Übertragung von der Steuerung zur Klemme bereit stehen.	UINT8	RO	0x00 (0 <sub>dez</sub> )
7000:11	Data Out 0	Ausgangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )
...	...	....	...	...	...
7000:26	Data Out 21	Ausgangs-Byte 21	UINT8	RO	0x00 (0 <sub>dez</sub> )

## Index 0x7001 COM ext. outputs

Index (hex)	Name	Bedeutung	Data type	Flags	Default
7001:0	COM ext. outputs		UINT8	RO	0x00 (0 <sub>dez</sub> )
7001:01	Transmit request	Identisch zu Index 0x7000	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
7001:02	Receive accepted		BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
7001:03	Init request		BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
7001:04	Send continuous		BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
7001:09	Output length		UINT8	RO	0x00 (0 <sub>dez</sub> )
7001:11	Data Out 0		UINT16	RO	0x0000 (0 <sub>dez</sub> )
	...		<	<	<
7001:42	Data Out 49		UINT16	RO	0x0000 (0 <sub>dez</sub> )

## Index 0xA000 COM Diag data

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
A000:0	COM Diag data	Max. SubIndex (hex)	UINT8	RO	0x21 (33 <sub>dez</sub> )
A000:01	Buffer overflow	Es ist ein Buffer-Overflow aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A000:02	Parity error	Es ist ein Parity-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A000:03	Framing error	Es ist ein Framing-Error aufgetreten	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A000:04	Overrun error	Es ist ein Overrun-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A000:05	Buffer full	Das Empfangs-FIFO ist voll. Ab diesem Zeitpunkt gehen alle eingehenden Daten verloren!	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A000:11	Data bytes in send buffer	Anzahl der Datenbytes im Sende-FIFO	UINT16	RO	0x0000 (0 <sub>dez</sub> )
A000:21	Data bytes in receive buffer	Anzahl der Datenbytes im Empfangs-FIFO	UINT16	RO	0x0000 (0 <sub>dez</sub> )

**Index 0xB000 Command**

Nur EL6021 ab FW06 und EL6001 ab FW08.

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
B000:0	Command	Max. SubIndex (hex)	UINT8	RO	0x03 (3 <sub>dez</sub> )
B000:01	Request	Über das Request-Objekt können Kommandos an die Klemme abgesetzt werden siehe <a href="#">Kommando-Mode [►_131]</a>	OCTET-STRING[2]	RW	{0}
B000:02	Status	Status des aktuell ausgeführten Kommandos 0: Kommando fehlerfrei ausgeführt 255: Kommando wird ausgeführt	UINT8	RO	0x00 (0 <sub>dez</sub> )
B000:03	Response	Optionaler Rückgabewert des Kommandos Byte 0: siehe B000:02 Byte 1: not used 2-n: Service response Data	OCTET-STRING[6]	RO	{0}

**Index 0xF000 Modular device profile**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F000:0	Modular device profile	Allgemeine Informationen des Modular Device Profiles	UINT8	RO	0x02 (2 <sub>dez</sub> )
F000:01	Module Index Oxdistance	Index (hex)abstand der Objekte der einzelnen Kanäle	UINT16	RO	0x0010 (16 <sub>dez</sub> )
F000:02	Maximum number of modules	Anzahl der Kanäle	UINT16	RO	0x0002 (2 <sub>dez</sub> )

**Index 0xF008 Code word**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F008:0	Code word	reserviert	UINT32	RW	0x00000000 (0 <sub>dez</sub> )

**Index 0xF010 Module list**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F010:0	Module list	Max. SubIndex (hex)	UINT8	RW	0x02 (2 <sub>dez</sub> )
F010:01	SubIndex 0x001	-	UINT32	RW	0x00000000 (0 <sub>dez</sub> )
F010:02	SubIndex 0x002	-	UINT32	RW	0x00000258 (600 <sub>dez</sub> )

## 6.2 Control- und Status-Wort

**Control-Wort**

Das Control-Wort (CW) befindet sich im Ausgangsprozessabbild und wird von der Steuerung zur Klemme übertragen.

Bit	CW.15	CW.14	CW.13	CW.12	CW.11	CW.10	CW.9	CW.8	CW.7	CW.6	CW.5	CW.4	CW.3	CW.2	CW.1	CW.0
Name	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	-	OL2*	OL1*	OL0*	SC	IR	RA	TR

Die Bits CW.15 bis CW.8 werden nur bei Verwendung des großen Prozessabbilds dargestellt. Bei Verwendung des kleinen oder mittleren Prozessabbilds erscheinen dort nur die Bits CW.7 bis CW.0 als Control-Byte! Die Anzahl der Ausgangs-Bytes, die für die Übertragung von der Steuerung zur Klemme bereit stehen werden dann von den Bits OL2\* ... OL0\* angezeigt.

## Legende

Bit	Name	Beschreibung	
CW.15 ... CW.8	OL7 ... OL0 (OutLenght)	1 <sub>dez</sub> ... 22 <sub>dez</sub>	Bei Verwendung des großen Prozessabbilds: Anzahl der Ausgangs-Bytes, die für die Übertragung von der Steuerung zur Klemme bereit stehen.
		-	Bei Verwendung des kleinen/mittleren Prozessabbilds: nicht dargestellt
CW.7	-	0 <sub>bin</sub>	reserviert
CW.6 ... CW.4	OL2* ... OL0* (OutLenght*)	0	Bei Verwendung des großen Prozessabbilds: reserviert
		1 ... 6	Bei Verwendung des mittleren Prozessabbilds: Anzahl der Ausgangs-Bytes, die für die Übertragung von der Steuerung zur Klemme bereit stehen.
		1 ... 4	Bei Verwendung des kleinen Prozessabbilds: Anzahl der Ausgangs-Bytes, die für die Übertragung von der Steuerung zur Klemme bereit stehen.
CW.3	SC (SendContinuous)	rise	Kontinuierliches Senden der Daten aus dem FIFO.  Über die Steuerung wird der Sendebuffer gefüllt (bis zu 128 Byte). Mit steigender Flanke des Bits SC wird der gefüllte Buffer-Inhalt gesendet. Sind die Daten übertragen, so wird dies durch das Setzen des Bits SW.2 von der Klemme an die Steuerung quittiert. SW.2 wird mit CW.3 zurückgenommen.
CW.2	IR (InitRequest)	1 <sub>bin</sub>	Die Steuerung fordert die Klemme zur Initialisierung auf. Die Sende- und Empfangsfunktionen werden gesperrt, die FIFO-Zeiger werden zurückgesetzt und die Schnittstelle wird mit den Werten der zuständigen Objekte (Baud Rate <a href="#">4073</a> [ <a href="#">▶ 151</a> ], Data Frame <a href="#">4074</a> [ <a href="#">▶ 151</a> ], Feature Bits <a href="#">4075</a> [ <a href="#">▶ 152</a> ]) initialisiert. Die Ausführung der Initialisierung wird von der Klemme mit dem Bit <a href="#">SW.2</a> [ <a href="#">▶ 175</a> ] (IA) quittiert.
		0 <sub>bin</sub>	Die Steuerung fordert von der Klemme wieder die Bereitschaft für den seriellen Datenaustausch.
CW.1	RA (ReceiveAccepted)	toggle	Die Steuerung quittiert die Entgegennahme von Daten mit Zustandsänderung dieses Bits. Erst daraufhin werden neue Daten von der Klemme zur Steuerung übertragen.
CW.0	TR (TransmitRequest)	toggle	Über eine Zustandsänderung dieses Bits teilt die Steuerung der Klemme mit, dass sich die in mit den OL-Bits angezeigte Anzahl von Bytes in den DataOut-Bytes befinden. Die Klemme quittiert die Entgegennahme der Daten im Status-Byte mit Zustandsänderung des Bits <a href="#">SW.0</a> [ <a href="#">▶ 175</a> ] (TA). Erst daraufhin werden neue Daten von der Steuerung zur Klemme übertragen.

**Status-Wort**

Das Status-Wort (SW) befindet sich im Eingangsprozessabbild und wird von der Klemme zur Steuerung übertragen.

Bit	SW. 15	SW. 14	SW. 13	SW. 12	SW. 11	SW. 10	SW. 9	SW. 8	SW. 7	SW.6	SW.5	SW.4	SW.3	SW. 2	SW.1	SW.0
<b>Name (kleines und mittleres Prozessabbild)</b>	-	-	-	-	-	-	-	-	-	IL2*	IL1*	IL0*	BUF_F	IA	RR	TA
<b>Name (großes Prozessabbild)</b>	IL7	IL6	IL5	IL4	IL3	IL2	IL1	IL0	-	OVERRUN ERR	FRAMING ERR	PARITY ERR	BUF_F	IA	RR	TA

Bei Verwendung des kleinen oder mittleren Prozessabbilds erscheinen dort nur die Bits SW.7 bis SW.0 als Status-Byte! Die Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen werden dann von den Bits IL2\* ... IL0\* angezeigt.

**Legende**

Bit	Name	Beschreibung	
SW.15 ... SW8	IL7 ... IL0 (InLenght*)	1 <sub>dez</sub> ... 22 <sub>dez</sub>	Bei Verwendung des großen Prozessabbilds: Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.
		-	Bei Verwendung des kleinen/mittleren Prozessabbilds: nicht dargestellt
SW.7	-	0 <sub>bin</sub>	reserviert
SW.6	IL2* (InLenght*)  OVERRUN ERR	0 ... 1	Bei Verwendung des großen Prozessabbilds: Es ist ein Overrun-Error aufgetreten. Das betroffene Datum wird nicht in das Empfangs-FIFO der Klemme geladen und geht verloren.
		1 ... 6	Bei Verwendung des mittleren Prozessabbilds: Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.
		1 ... 4	Bei Verwendung des kleinen Prozessabbilds: Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.
SW.5	IL1* (InLenght*)  FRAMING ERR	0 ... 1	Bei Verwendung des großen Prozessabbilds: Es ist ein Framing-Error aufgetreten. Das betroffene Datum wird nicht in das Empfangs-FIFO der Klemme geladen und geht verloren.
		1 ... 6	Bei Verwendung des mittleren Prozessabbilds: Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.
		1 ... 4	Bei Verwendung des kleinen Prozessabbilds: Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.
SW.4	IL0* (InLenght*)  PARITY ERR	0 ... 1	Bei Verwendung des großen Prozessabbilds: Es ist ein Parity-Error aufgetreten. Das betroffene Datum wird nicht in das Empfangs-FIFO der Klemme geladen und geht verloren
		1 ... 6	Bei Verwendung des mittleren Prozessabbilds: Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.
		1 ... 4	Bei Verwendung des kleinen Prozessabbilds: Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.
SW.3	BUF_F	1 <sub>bin</sub>	Das Empfangs-FIFO ist voll. Alle ab jetzt eingehenden Daten gehen verloren!
SW.2	IA (InitAccepted-Bit)	1 <sub>bin</sub>	Die Initialisierung wurde von der Klemme ausgeführt.
		0 <sub>bin</sub>	Die Klemme ist wieder für den seriellen Datenaustausch bereit.
SW.1	RR (ReceiveRequest)	toggle	Über eine Zustandsänderung dieses Bits teilt die Klemme der Steuerung mit, dass sich die in IL-Bits angezeigte Anzahl von Bytes in den Data-Bytes befinden. Die Steuerung muss die Entgegennahme der Daten im Control-Byte mit Zustandsänderung des Bits <a href="#">CW.1</a> ( <a href="#">▶ 174</a> ) (RA) quittieren. Erst daraufhin werden neue Daten von der Klemme zur Steuerung übertragen.
SW.0	TA (TransmitAccepted)	toggle	Die Klemme quittiert die Entgegennahme von Daten mit Zustandsänderung dieses Bits. Erst daraufhin werden neue Daten von der Steuerung zur Klemme übertragen.

**Beispiele für die Datenübertragung**

Die Beispiele verwenden das große Prozessabbild.

- Initialisierung:

Control-Wort		Status-Wort		Kommentar
CW.15 ... CW.8	CW.7 ... CW.0	SW.15 ... SW.8	SW.7 ... SW.0	
XXXX XXXX <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	Start der Datenübertragung
0000 0000 <sub>bin</sub>	0000 0100 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	Steuerung fordert von der Klemme die Initialisierung
0000 0000 <sub>bin</sub>	0000 0100 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0100 <sub>bin</sub>	Befehl ist ausgeführt: Klemme hat die Initialisierung vollzogen
0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0100 <sub>bin</sub>	Steuerung fordert von der Klemme die Bereitschaft zum seriellen Datenaustausch
0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	Befehl ist ausgeführt: Klemme ist für seriellen Datenaustausch bereit

- Datenübertragung von der Steuerung zur Klemme:

Control-Wort		Status-Wort		Kommentar
CW.15 ... CW.8	CW.7 ... CW.0	SW.15 ... SW.8	SW.7 ... SW.0	
0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x0 <sub>bin</sub>	Start der Datenübertragung
0000 0010 <sub>bin</sub>	0000 0001 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x0 <sub>bin</sub>	Steuerung fordert von der Klemme <b>2</b> Bytes zu senden
0000 0010 <sub>bin</sub>	0000 0001 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x1 <sub>bin</sub>	Befehl ist ausgeführt: Klemme hat <b>2</b> Bytes in Sende-FIFO geladen
0001 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x1 <sub>bin</sub>	Steuerung fordert von der Klemme <b>16</b> Bytes zu senden
0001 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x0 <sub>bin</sub>	Befehl ist ausgeführt: Klemme hat <b>16</b> Bytes in Sende-FIFO geladen

- Datenübertragung von der Klemme zur Steuerung:

Control-Wort		Status-Wort		Kommentar
CW.15 ... CW.8	CW.7 ... CW.0	SW.15 ... SW.8	SW.7 ... SW.0	
XXXX XXXX <sub>bin</sub>	0000 000x <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 000x <sub>bin</sub>	Start der Datenübertragung
XXXX XXXX <sub>bin</sub>	0000 000x <sub>bin</sub>	0000 0011 <sub>bin</sub>	0000 001x <sub>bin</sub>	Klemme fordert von der Steuerung <b>3</b> Bytes zu übernehmen
XXXX XXXX <sub>bin</sub>	0000 001x <sub>bin</sub>	0000 0011 <sub>bin</sub>	0000 001x <sub>bin</sub>	Sie quittieren die Anforderung: Steuerung hat <b>3</b> Bytes aus Empfangs-FIFO übernommen
XXXX XXXX <sub>bin</sub>	0000 001x <sub>bin</sub>	0001 0110 <sub>bin</sub>	0000 000x <sub>bin</sub>	Klemme fordert von der Steuerung <b>22</b> Bytes zu übernehmen
XXXX XXXX <sub>bin</sub>	0000 000x <sub>bin</sub>	0001 0110 <sub>bin</sub>	0000 000x <sub>bin</sub>	Sie quittieren die Anforderung: Steuerung hat <b>22</b> Bytes aus Empfangs-FIFO übernommen

# 7 Übersicht CoE Objekte EL6002, EL6022

## 7.1 Objektbeschreibung und Parametrierung

### **i** EtherCAT XML Device Description

Die Darstellung entspricht der Anzeige der CoE-Objekte aus der EtherCAT XML Device Description. Es wird empfohlen, die entsprechende aktuellste XML-Datei im Download-Bereich auf der [Beckhoff Website](#) herunterzuladen und entsprechend der Installationsanweisungen zu installieren.

### **i** Parametrierung

Die Parametrierung der Klemme wird über den [CoE - Online Reiter \[▶ 115\]](#) (mit Doppelklick auf das entsprechende Objekt, s.u.) vorgenommen.

### Einführung

In der CoE-Übersicht sind Objekte mit verschiedenem Einsatzzweck enthalten:

- [Objekte die zu Parametrierung \[▶ 177\]](#) bei der Inbetriebnahme nötig sind
- [Objekte die interne Settings \[▶ 178\]](#) anzeigen und ggf. nicht veränderlich sind
- [Profilspezifische Objekte \[▶ 190\]](#), die die Statusanzeigen der Ein- und Ausgänge darstellen.

Im Folgenden werden zuerst die im normalen Betrieb benötigten Objekte vorgestellt, dann die für eine vollständige Übersicht noch fehlenden Objekte.

### 7.1.1 Objekte für die Inbetriebnahme

#### Index 0x1011 Restore default parameters

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1011:0	<a href="#">Restore default parameters [▶ 208]</a>	Herstellen der Defaulteinstellungen	UINT8	RO	0x01 (1 <sub>dez</sub> )
1011:01	SubIndex 0x001	Wenn Sie dieses Objekt im Set Value Dialog auf "0x64616F6C" setzen, werden alle Backup Objekte wieder in den Auslieferungszustand gesetzt.	UINT32	RW	0x00000000 (0 <sub>dez</sub> )

**Index 0x80n0 COM Settings Ch. 1 (n = 0), Ch. 2 (n = 1)**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
80n0:0	COM Settings Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x1A (26 <sub>dez</sub> )
80n0:01**	Enable RTS/CTS	FALSE RTS/CTS nicht freigeschaltet TRUE RTS/CTS freigeschaltet	BOOLEAN	RW	0x01 (1 <sub>dez</sub> )
80n0:02	Enable XON/XOFF supported tx data	FALSE XON/XOFF wird für Sendedaten nicht unterstützt TRUE XON/XOFF wird für Sendedaten unterstützt	BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
80n0:03	Enable XON/XOFF supported rx data	FALSE XON/XOFF wird für Empfangsdaten nicht unterstützt TRUE XON/XOFF wird für Empfangsdaten unterstützt	BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
80n0:04	Enable send FIFO data continuous	FALSE Kein kontinuierliches Senden der Daten aus dem FIFO TRUE Kontinuierliches Senden der Daten aus dem FIFO freigeschaltet: Über die Steuerung wird der Sendebuffer gefüllt (bis zu 128 Byte). Mit steigender Flanke des Bits "SendContinuous" wird der gefüllte Buffer-Inhalt gesendet. Sind die Daten übertragen, so wird dies durch das Setzen des Bits "InitAccepted" von der Klemme an die Steuerung quittiert. "InitAccepted" wird mit "SendContinuous" zurückgesetzt.	BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
80n0:05	Enable transfer rate optimization	FALSE Transferraten-Optimierung ausgeschaltet TRUE Transferraten-Optimierung eingeschaltet: Der Inhalt des Eingangspuffers wird automatisch ins Prozessabbild übernommen, wenn <ul style="list-style-type: none"> <li>nach dem Empfang von Daten für ca. 16 Bitzeiten (also die Zeit, die der Empfang von 2 Bytes gedauert hätte) kein weiteres Byte empfangen wurde.</li> <li>das Prozessabbild gefüllt ist</li> </ul>	BOOLEAN	RW	0x01 (1 <sub>dez</sub> )
80n0:06** *	Enable half duplex	FALSE Voll duplex-Modus TRUE Halbduplex-Modus	BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
80n0:07** *	Enable point to point connection (RS422)	FALSE Punkt zu Punkt Verbindung deaktiviert TRUE Punkt zu Punkt Verbindung aktiviert	BOOLEAN	RW	0x00 (0 <sub>dez</sub> )
80n0:11	Baud rate	Detaillierte Hinweise hierzu im Kapitel <a href="#">Kommunikationseigenschaften</a> [► 137]	BIT4	RW	0x06 (6 <sub>dez</sub> )
80n0:15	Data frame	Detaillierte Hinweise hierzu im Kapitel <a href="#">Kommunikationseigenschaften</a> [► 137]	BIT4	RW	0x03 (3 <sub>dez</sub> )
80n0:1A	Rx buffer full notification	Der Wert legt die Anzahl der Daten im Empfangs-FIFO fest, ab der das Bit "Buffer full" gesetzt wird.	UINT16	RW	0x0360 (864 <sub>dez</sub> )
80n0:1B** **	Explicit baudrate	Detaillierte Hinweise hierzu im Kapitel <a href="#">TcVirtualComDriver</a> [► 135] und <a href="#">Kommunikationseigenschaften</a> [► 137]	UINT32	RW	0x00002580 (9600 <sub>dez</sub> )

\*\*) nur EL6002

\*\*\*) nur EL6022

\*\*\*\*) nur EL6002 ab [Firmware 03](#) [► 195]**7.1.2 Standardobjekte (0x1000-0x1FFF)**

Die Standardobjekte haben für alle EtherCAT-Slaves die gleiche Bedeutung.

**Index 0x1000 Device type**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1000:0	Device type	Geräte-Typ des EtherCAT-Slaves: Das Lo-Word enthält das verwendete CoE Profil (5001). Das Hi-Word enthält das Modul Profil entsprechend des Modular Device Profile.	UINT32	RO	0x02581389 (39326601 <sub>dez</sub> )

**Index 0x1008 Device name**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1008:0	Device name	Geräte-Name des EtherCAT-Slave	STRING	RO	EL60xx

**Index 0x1009 Hardware version**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1009:0	Hardware version	Hardware-Version des EtherCAT-Slaves	STRING	RO	00

**Index 0x100A Software version**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
100A:0	Software version	Firmware-Version des EtherCAT-Slaves	STRING	RO	01

**Index 0x1018 Identity**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1018:0	Identity	Informationen, um den Slave zu identifizieren	UINT8	RO	0x04 (4 <sub>dez</sub> )
1018:01	Vendor ID	Hersteller-ID des EtherCAT-Slaves	UINT32	RO	0x00000002 (2 <sub>dez</sub> )
1018:02	Product code	Produkt-Code des EtherCAT-Slaves	UINT32	RO	EL6002: 0x17723052 (393359442 <sub>dez</sub> ) EL6022: 0x17863052 (394670162 <sub>dez</sub> )
1018:03	Revision	Revisionsnummer des EtherCAT-Slaves, das Low-Word (Bit 0-15) kennzeichnet die Sonderklemmennummer, das High-Word (Bit 16-31) verweist auf die Gerätebeschreibung	UINT32	RO	0x00100000 (1048576 <sub>dez</sub> )
1018:04	Serial number	Seriennummer des EtherCAT-Slaves, das Low-Byte (Bit 0-7) des Low-Words enthält das Produktionsjahr, das High-Byte (Bit 8-15) des Low-Words enthält die Produktionswoche, das High-Word (Bit 16-31) ist 0	UINT32	RO	0x00000000 (0 <sub>dez</sub> )

**Index 0x10F0 Backup parameter handling**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
10F0:0	Backup parameter handling	Informationen zum standardisierten Laden und Speichern der Backup Entries	UINT8	RO	0x01 (1 <sub>dez</sub> )
10F0:01	Checksum	Checksumme über alle Backup-Entries des EtherCAT-Slaves	UINT32	RO	0x00000000 (0 <sub>dez</sub> )

**Index 0x1600 COM RxPDO-Map Outputs Ch.1**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1600:0	COM RxPDO-Map Outputs Ch.1	PDO Mapping RxPDO 1	UINT8	RO	0x1C (28 <sub>dez</sub> )
1600:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x01 (Transmit request))	BOOLEAN	RO	0x7000:01, 1
1600:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x02 (Receive accepted))	BOOLEAN	RO	0x7000:02, 1
1600:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x03 (Init request))	BOOLEAN	RO	0x7000:03, 1
1600:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x04 (Send continuous))	BOOLEAN	RO	0x7000:04, 1
1600:05	SubIndex 0x005	5. PDO Mapping entry (4 bits align)	Align4	RO	0x0000:00, 4
1600:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x09 (Output length))	UINT8	RO	0x7000:09, 8
1600:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT8	RO	0x7000:11, 8
1600:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT8	RO	0x7000:12, 8
1600:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT8	RO	0x7000:13, 8
1600:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT8	RO	0x7000:14, 8
1600:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT8	RO	0x7000:15, 8
1600:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT8	RO	0x7000:16, 8
1600:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT8	RO	0x7000:17, 8
1600:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT8	RO	0x7000:18, 8
1600:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT8	RO	0x7000:19, 8
1600:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT8	RO	0x7000:1A, 8
1600:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT8	RO	0x7000:1B, 8
1600:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT8	RO	0x7000:1C, 8
1600:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT8	RO	0x7000:1D, 8
1600:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT8	RO	0x7000:1E, 8
1600:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT8	RO	0x7000:1F, 8
1600:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT8	RO	0x7000:20, 8
1600:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT8	RO	0x7000:21, 8
1600:18	SubIndex 0x024	24. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT8	RO	0x7000:22, 8
1600:19	SubIndex 0x025	25. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT8	RO	0x7000:23, 8
1600:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT8	RO	0x7000:24, 8
1600:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT8	RO	0x7000:25, 8
1600:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT8	RO	0x7000:26, 8

**Index 0x1601 COM RxPDO-Map Outputs Ch.2**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1601:0	COM RxPDO-Map Outputs Ch.2	PDO Mapping RxPDO 2	UINT8	RO	0x1C (28 <sub>dez</sub> )
1601:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x01 (Transmit request))	BOOLEAN	RO	0x7010:01, 1
1601:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x02 (Receive accepted))	BOOLEAN	RO	0x7010:02, 1
1601:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x03 (Init request))	BOOLEAN	RO	0x7010:03, 1
1601:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x04 (Send continuous))	BOOLEAN	RO	0x7010:04, 1
1601:05	SubIndex 0x005	5. PDO Mapping entry (4 bits align)	Align4	RO	0x0000:00, 4
1601:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x09 (Output length))	UINT8	RO	0x7010:09, 8
1601:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x11 (Data Out 0))	UINT8	RO	0x7010:11, 8
1601:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x12 (Data Out 1))	UINT8	RO	0x7010:12, 8
1601:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x13 (Data Out 2))	UINT8	RO	0x7010:13, 8
1601:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x14 (Data Out 3))	UINT8	RO	0x7010:14, 8
1601:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x15 (Data Out 4))	UINT8	RO	0x7010:15, 8
1601:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x16 (Data Out 5))	UINT8	RO	0x7010:16, 8
1601:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x17 (Data Out 6))	UINT8	RO	0x7010:17, 8
1601:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x18 (Data Out 7))	UINT8	RO	0x7010:18, 8
1601:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x19 (Data Out 8))	UINT8	RO	0x7010:19, 8
1601:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1A (Data Out 9))	UINT8	RO	0x7010:1A, 8
1601:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1B (Data Out 10))	UINT8	RO	0x7010:1B, 8
1601:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1C (Data Out 11))	UINT8	RO	0x7010:1C, 8
1601:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1D (Data Out 12))	UINT8	RO	0x7010:1D, 8
1601:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1E (Data Out 13))	UINT8	RO	0x7010:1E, 8
1601:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1F (Data Out 14))	UINT8	RO	0x7010:1F, 8
1601:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x20 (Data Out 15))	UINT8	RO	0x7010:20, 8
1601:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x21 (Data Out 16))	UINT8	RO	0x7010:21, 8
1601:18	SubIndex 0x024	24. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x22 (Data Out 17))	UINT8	RO	0x7010:22, 8
1601:19	SubIndex 0x025	25. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x23 (Data Out 18))	UINT8	RO	0x7010:23, 8
1601:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x24 (Data Out 19))	UINT8	RO	0x7010:24, 8
1601:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x25 (Data Out 20))	UINT8	RO	0x7010:25, 8
1601:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x26 (Data Out 21))	UINT8	RO	0x7010:26, 8

**Index 0x1604 COM RxPDO-Map Outputs Ch.1**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1604:0	COM RxPDO-Map Outputs Ch.1	PDO Mapping RxPDO 1	UINT8	RO	0x17 (23 <sub>dez</sub> )
1604:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7001 (Ctrl Ch.1), entry 0x01 (Ctrl))	UINT16	RO	0x7001:01, 16
1604:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x11 (Data Out 0))	UINT8	RO	0x7000:11, 8
1604:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x12 (Data Out 1))	UINT8	RO	0x7000:12, 8
1604:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x13 (Data Out 2))	UINT8	RO	0x7000:13, 8
1604:05	SubIndex 0x005	5. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x14 (Data Out 3))	UINT8	RO	0x7000:14, 8
1604:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x15 (Data Out 4))	UINT8	RO	0x7000:15, 8
1604:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x16 (Data Out 5))	UINT8	RO	0x7000:16, 8
1604:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x17 (Data Out 6))	UINT8	RO	0x7000:17, 8
1604:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x18 (Data Out 7))	UINT8	RO	0x7000:18, 8
1604:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x19 (Data Out 8))	UINT8	RO	0x7000:19, 8
1604:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1A (Data Out 9))	UINT8	RO	0x7000:1A, 8
1604:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1B (Data Out 10))	UINT8	RO	0x7000:1B, 8
1604:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1C (Data Out 11))	UINT8	RO	0x7000:1C, 8
1604:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1D (Data Out 12))	UINT8	RO	0x7000:1D, 8
1604:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1E (Data Out 13))	UINT8	RO	0x7000:1E, 8
1604:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x1F (Data Out 14))	UINT8	RO	0x7000:1F, 8
1604:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x20 (Data Out 15))	UINT8	RO	0x7000:20, 8
1604:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x21 (Data Out 16))	UINT8	RO	0x7000:21, 8
1604:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x22 (Data Out 17))	UINT8	RO	0x7000:22, 8
1604:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x23 (Data Out 18))	UINT8	RO	0x7000:23, 8
1604:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x24 (Data Out 19))	UINT8	RO	0x7000:24, 8
1604:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x25 (Data Out 20))	UINT8	RO	0x7000:25, 8
1604:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7000 (COM Outputs Ch.1), entry 0x26 (Data Out 21))	UINT8	RO	0x7000:26, 8

**Index 0x1605 COM RxPDO-Map Outputs Ch.2**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1605:0	COM RxPDO-Map Outputs Ch.1	PDO Mapping RxPDO 1	UINT8	RO	0x17 (23 <sub>dez</sub> )
1605:01	SubIndex 0x001	1. PDO Mapping entry (object 0x7011 (Ctrl Ch.2), entry 0x01 (Ctrl))	UINT16	RO	0x7011:01, 16
1605:02	SubIndex 0x002	2. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x11 (Data Out 0))	UINT8	RO	0x7010:11, 8
1605:03	SubIndex 0x003	3. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x12 (Data Out 1))	UINT8	RO	0x7010:12, 8
1605:04	SubIndex 0x004	4. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x13 (Data Out 2))	UINT8	RO	0x7010:13, 8
1605:05	SubIndex 0x005	5. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x14 (Data Out 3))	UINT8	RO	0x7010:14, 8
1605:06	SubIndex 0x006	6. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x15 (Data Out 4))	UINT8	RO	0x7010:15, 8
1605:07	SubIndex 0x007	7. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x16 (Data Out 5))	UINT8	RO	0x7010:16, 8
1605:08	SubIndex 0x008	8. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x17 (Data Out 6))	UINT8	RO	0x7010:17, 8
1605:09	SubIndex 0x009	9. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x18 (Data Out 7))	UINT8	RO	0x7010:18, 8
1605:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x19 (Data Out 8))	UINT8	RO	0x7010:19, 8
1605:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1A (Data Out 9))	UINT8	RO	0x7010:1A, 8
1605:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1B (Data Out 10))	UINT8	RO	0x7010:1B, 8
1605:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1C (Data Out 11))	UINT8	RO	0x7010:1C, 8
1605:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1D (Data Out 12))	UINT8	RO	0x7010:1D, 8
1605:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1E (Data Out 13))	UINT8	RO	0x7010:1E, 8
1605:10	SubIndex 0x016	16. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x1F (Data Out 14))	UINT8	RO	0x7010:1F, 8
1605:11	SubIndex 0x017	17. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x20 (Data Out 15))	UINT8	RO	0x7010:20, 8
1605:12	SubIndex 0x018	18. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x21 (Data Out 16))	UINT8	RO	0x7010:21, 8
1605:13	SubIndex 0x019	19. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x22 (Data Out 17))	UINT8	RO	0x7010:22, 8
1605:14	SubIndex 0x020	20. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x23 (Data Out 18))	UINT8	RO	0x7010:23, 8
1605:15	SubIndex 0x021	21. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x24 (Data Out 19))	UINT8	RO	0x7010:24, 8
1605:16	SubIndex 0x022	22. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x25 (Data Out 20))	UINT8	RO	0x7010:25, 8
1605:17	SubIndex 0x023	23. PDO Mapping entry (object 0x7010 (COM Outputs Ch.2), entry 0x26 (Data Out 21))	UINT8	RO	0x7010:26, 8

## Index 0x1A00 COM TxPDO-Map Inputs Ch.1

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A00:0	COM TxPDO-Map Inputs Ch.1	PDO Mapping TxPDO 1	UINT8	RO	0x1F (31 <sub>dez</sub> )
1A00:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x01 (Transmit accepted))	BOOLEAN	RO	0x6000:01, 1
1A00:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x02 (Receive request))	BOOLEAN	RO	0x6000:02, 1
1A00:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x03 (Init accepted))	BOOLEAN	RO	0x6000:03, 1
1A00:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x04 (Buffer full))	BOOLEAN	RO	0x6000:04, 1
1A00:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x05 (Parity error))	BOOLEAN	RO	0x6000:05, 1
1A00:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x06 (Framing error))	BOOLEAN	RO	0x6000:06, 1
1A00:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x07 (Overrun error))	BOOLEAN	RO	0x6000:07, 1
1A00:08	SubIndex 0x008	8. PDO Mapping entry (1 bits align)	Align1	RO	0x0000:00, 1
1A00:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x09 (Input length))	UINT8	RO	0x6000:09, 8
1A00:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT8	RO	0x6000:11, 8
1A00:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT8	RO	0x6000:12, 8
1A00:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT8	RO	0x6000:13, 8
1A00:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT8	RO	0x6000:14, 8
1A00:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT8	RO	0x6000:15, 8
1A00:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT8	RO	0x6000:16, 8
1A00:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT8	RO	0x6000:17, 8
1A00:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT8	RO	0x6000:18, 8
1A00:12	SubIndex 0x018	18. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT8	RO	0x6000:19, 8
1A00:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT8	RO	0x6000:1A, 8
1A00:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT8	RO	0x6000:1B, 8
1A00:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT8	RO	0x6000:1C, 8
1A00:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT8	RO	0x6000:1D, 8
1A00:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT8	RO	0x6000:1E, 8
1A00:18	SubIndex 0x024	24. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT8	RO	0x6000:1F, 8
1A00:19	SubIndex 0x025	25. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT8	RO	0x6000:20, 8
1A00:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT8	RO	0x6000:21, 8
1A00:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT8	RO	0x6000:22, 8
1A00:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT8	RO	0x6000:23, 8
1A00:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT8	RO	0x6000:24, 8
1A00:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT8	RO	0x6000:25, 8
1A00:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT8	RO	0x6000:26, 8

**Index 0x1A01 COM TxPDO-Map Inputs Ch.2**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A01:0	COM TxPDO-Map Inputs Ch.2	PDO Mapping TxPDO 2	UINT8	RO	0x1F (31 <sub>dez</sub> )
1A01:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x01 (Transmit accepted))	BOOLEAN	RO	0x6010:01, 1
1A01:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x02 (Receive request))	BOOLEAN	RO	0x6010:02, 1
1A01:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x03 (Init accepted))	BOOLEAN	RO	0x6010:03, 1
1A01:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x04 (Buffer full))	BOOLEAN	RO	0x6010:04, 1
1A01:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x05 (Parity error))	BOOLEAN	RO	0x6010:05, 1
1A01:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x06 (Framing error))	BOOLEAN	RO	0x6010:06, 1
1A01:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x07 (Overrun error))	BOOLEAN	RO	0x6010:07, 1
1A01:08	SubIndex 0x008	8. PDO Mapping entry (1 bits align)	Align1	RO	0x0000:00, 1
1A01:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x09 (Input length))	UINT8	RO	0x6010:09, 8
1A01:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x11 (Data In 0))	UINT8	RO	0x6010:11, 8
1A01:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x12 (Data In 1))	UINT8	RO	0x6010:12, 8
1A01:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x13 (Data In 2))	UINT8	RO	0x6010:13, 8
1A01:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x14 (Data In 3))	UINT8	RO	0x6010:14, 8
1A01:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x15 (Data In 4))	UINT8	RO	0x6010:15, 8
1A01:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x16 (Data In 5))	UINT8	RO	0x6010:16, 8
1A01:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x17 (Data In 6))	UINT8	RO	0x6010:17, 8
1A01:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x18 (Data In 7))	UINT8	RO	0x6010:18, 8
1A01:12	SubIndex 0x018	18. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x19 (Data In 8))	UINT8	RO	0x6010:19, 8
1A01:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1A (Data In 9))	UINT8	RO	0x6010:1A, 8
1A01:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1B (Data In 10))	UINT8	RO	0x6010:1B, 8
1A01:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1C (Data In 11))	UINT8	RO	0x6010:1C, 8
1A01:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1D (Data In 12))	UINT8	RO	0x6010:1D, 8
1A01:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1E (Data In 13))	UINT8	RO	0x6010:1E, 8
1A01:18	SubIndex 0x024	24. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1F (Data In 14))	UINT8	RO	0x6010:1F, 8
1A01:19	SubIndex 0x025	25. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x20 (Data In 15))	UINT8	RO	0x6010:20, 8
1A01:1A	SubIndex 0x026	26. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x21 (Data In 16))	UINT8	RO	0x6010:21, 8
1A01:1B	SubIndex 0x027	27. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x22 (Data In 17))	UINT8	RO	0x6010:22, 8
1A01:1C	SubIndex 0x028	28. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x23 (Data In 18))	UINT8	RO	0x6010:23, 8
1A01:1D	SubIndex 0x029	29. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x24 (Data In 19))	UINT8	RO	0x6010:24, 8
1A01:1E	SubIndex 0x030	30. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x25 (Data In 20))	UINT8	RO	0x6010:25, 8
1A01:1F	SubIndex 0x031	31. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x26 (Data In 21))	UINT8	RO	0x6010:26, 8

**Index 0x1A04 COM TxPDO-Map Inputs Ch.1**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A04:0	COM TxPDO-Map Inputs Ch.1	PDO Mapping TxPDO 1	UINT8	RO	0x17 (23 <sub>dez</sub> )
1A04:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6001 (Status Ch.1), entry 0x01 (Status))	UINT16	RO	0x6001:01, 16
1A04:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x11 (Data In 0))	UINT8	RO	0x6000:11, 8
1A04:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x12 (Data In 1))	UINT8	RO	0x6000:12, 8
1A04:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x13 (Data In 2))	UINT8	RO	0x6000:13, 8
1A04:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x14 (Data In 3))	UINT8	RO	0x6000:14, 8
1A04:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x15 (Data In 4))	UINT8	RO	0x6000:15, 8
1A04:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x16 (Data In 5))	UINT8	RO	0x6000:16, 8
1A04:08	SubIndex 0x008	8. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x17 (Data In 6))	UINT8	RO	0x6000:17, 8
1A04:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x18 (Data In 7))	UINT8	RO	0x6000:18, 8
1A04:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x19 (Data In 8))	UINT8	RO	0x6000:19, 8
1A04:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1A (Data In 9))	UINT8	RO	0x6000:1A, 8
1A04:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1B (Data In 10))	UINT8	RO	0x6000:1B, 8
1A04:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1C (Data In 11))	UINT8	RO	0x6000:1C, 8
1A04:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1D (Data In 12))	UINT8	RO	0x6000:1D, 8
1A04:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1E (Data In 13))	UINT8	RO	0x6000:1E, 8
1A04:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x1F (Data In 14))	UINT8	RO	0x6000:1F, 8
1A04:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x20 (Data In 15))	UINT8	RO	0x6000:20, 8
1A04:12	SubIndex 0x018	17. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x21 (Data In 16))	UINT8	RO	0x6000:21, 8
1A04:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x22 (Data In 17))	UINT8	RO	0x6000:22, 8
1A04:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x23 (Data In 18))	UINT8	RO	0x6000:23, 8
1A04:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x24 (Data In 19))	UINT8	RO	0x6000:24, 8
1A04:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x25 (Data In 20))	UINT8	RO	0x6000:25, 8
1A04:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6000 (COM Inputs Ch.1), entry 0x26 (Data In 21))	UINT8	RO	0x6000:26, 8

**Index 0x1A05 COM TxPDO-Map Inputs Ch.2**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A05:0	COM TxPDO-Map Inputs Ch.2	PDO Mapping TxPDO 1	UINT8	RO	0x17 (23 <sub>dez</sub> )
1A05:01	SubIndex 0x001	1. PDO Mapping entry (object 0x6011 (Status Ch.2), entry 0x01 (Status))	UINT16	RO	0x6011:01, 16
1A05:02	SubIndex 0x002	2. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x11 (Data In 0))	UINT8	RO	0x6010:11, 8
1A05:03	SubIndex 0x003	3. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x12 (Data In 1))	UINT8	RO	0x6010:12, 8
1A05:04	SubIndex 0x004	4. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x13 (Data In 2))	UINT8	RO	0x6010:13, 8
1A05:05	SubIndex 0x005	5. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x14 (Data In 3))	UINT8	RO	0x6010:14, 8
1A05:06	SubIndex 0x006	6. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x15 (Data In 4))	UINT8	RO	0x6010:15, 8
1A05:07	SubIndex 0x007	7. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x16 (Data In 5))	UINT8	RO	0x6010:16, 8
1A05:08	SubIndex 0x008	8. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x17 (Data In 6))	UINT8	RO	0x6010:17, 8
1A05:09	SubIndex 0x009	9. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x18 (Data In 7))	UINT8	RO	0x6010:18, 8
1A05:0A	SubIndex 0x010	10. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x19 (Data In 8))	UINT8	RO	0x6010:19, 8
1A05:0B	SubIndex 0x011	11. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1A (Data In 9))	UINT8	RO	0x6010:1A, 8
1A05:0C	SubIndex 0x012	12. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1B (Data In 10))	UINT8	RO	0x6010:1B, 8
1A05:0D	SubIndex 0x013	13. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1C (Data In 11))	UINT8	RO	0x6010:1C, 8
1A05:0E	SubIndex 0x014	14. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1D (Data In 12))	UINT8	RO	0x6010:1D, 8
1A05:0F	SubIndex 0x015	15. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1E (Data In 13))	UINT8	RO	0x6010:1E, 8
1A05:10	SubIndex 0x016	16. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x1F (Data In 14))	UINT8	RO	0x6010:1F, 8
1A05:11	SubIndex 0x017	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x20 (Data In 15))	UINT8	RO	0x6010:20, 8
1A05:12	SubIndex 0x018	17. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x21 (Data In 16))	UINT8	RO	0x6010:21, 8
1A05:13	SubIndex 0x019	19. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x22 (Data In 17))	UINT8	RO	0x6010:22, 8
1A05:14	SubIndex 0x020	20. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x23 (Data In 18))	UINT8	RO	0x6010:23, 8
1A05:15	SubIndex 0x021	21. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x24 (Data In 19))	UINT8	RO	0x6010:24, 8
1A05:16	SubIndex 0x022	22. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x25 (Data In 20))	UINT8	RO	0x6010:25, 8
1A05:17	SubIndex 0x023	23. PDO Mapping entry (object 0x6010 (COM Inputs Ch.2), entry 0x26 (Data In 21))	UINT8	RO	0x6010:26, 8

**Index 0x1C00 Sync manager type**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C00:0	Sync manager type	Benutzung der Sync Manager	UINT8	RO	0x04 (4 <sub>dez</sub> )
1C00:01	SubIndex 0x001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1 <sub>dez</sub> )
1C00:02	SubIndex 0x002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2 <sub>dez</sub> )
1C00:03	SubIndex 0x003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3 <sub>dez</sub> )
1C00:04	SubIndex 0x004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4 <sub>dez</sub> )

**Index 0x1C12 RxPDO assign**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RO	0x04 (4 <sub>dez</sub> )
1C12:01	SubIndex 0x001	1. zugeordnete RxPDO (enthält den Index 0xdes zugehörigen RxPDO Mapping Objekts)	UINT16	RO	0x1600 (5632 <sub>dez</sub> )
1C12:02	SubIndex 0x002	2. zugeordnete RxPDO (enthält den Index 0xdes zugehörigen RxPDO Mapping Objekts)	UINT16	RO	0x1601 (5633 <sub>dez</sub> )

**Index 0x1C13 TxPDO assign**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RO	0x04 (4 <sub>dez</sub> )
1C13:01	SubIndex 0x001	1. zugeordnete TxPDO (enthält den Index 0xdes zugehörigen TxPDO Mapping Objekts)	UINT16	RO	0x1A00 (6656 <sub>dez</sub> )
1C13:02	SubIndex 0x002	2. zugeordnete TxPDO (enthält den Index 0xdes zugehörigen TxPDO Mapping Objekts)	UINT16	RO	0x1A01 (6657 <sub>dez</sub> )

**Index 0x1C32 SM output parameter**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C32:0	SM output parameter	Synchronisierungsparameter der Outputs	UINT8	RO	0x20 (32 <sub>dez</sub> )
1C32:01	Sync mode	Aktuelle Synchronisierungsbetriebsart: <ul style="list-style-type: none"> <li>• 0: Free Run</li> <li>• 1: Synchron with SM 2 Event</li> <li>• 2: DC-Mode - Synchron with SYNC0 Event</li> <li>• 3: DC-Mode - Synchron with SYNC1 Event</li> </ul>	UINT16	RW	0x0000 (0 <sub>dez</sub> )
1C32:02	Cycle time	Zykluszeit (in ns): <ul style="list-style-type: none"> <li>• Free Run: Zykluszeit des lokalen Timers</li> <li>• Synchron with SM 2 Event: Zykluszeit des Masters</li> <li>• DC-Mode: SYNC0/SYNC1 Cycle Time</li> </ul>	UINT32	RW	0x0003D090 (250000 <sub>dez</sub> )
1C32:03	Shift time	Zeit zwischen SYNC0 Event und Ausgabe der Outputs (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C32:04	Sync modes supported	Unterstützte Synchronisierungsbetriebsarten: <ul style="list-style-type: none"> <li>• Bit 0 = 1: Free Run wird unterstützt</li> <li>• Bit 1 = 1: Synchron with SM 2 Event wird unterstützt</li> <li>• Bit 2-3 = 01: DC-Mode wird unterstützt</li> <li>• Bit 4-5 = 10: Output Shift mit SYNC1 Event (nur DC-Mode)</li> <li>• Bit 14 = 1: dynamische Zeiten (Messen durch Beschreiben von 1C32:08)</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dez</sub> )
1C32:05	Minimum cycle time	Minimale Zykluszeit (in ns)	UINT32	RO	0x00004E20 (20000 <sub>dez</sub> )
1C32:06	Calc and copy time	Minimale Zeit zwischen SYNC0 und SYNC1 Event (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C32:08	Command	<ul style="list-style-type: none"> <li>• 0: Messung der lokalen Zykluszeit wird gestoppt</li> <li>• 1: Messung der lokalen Zykluszeit wird gestartet</li> </ul> <p>Die Entries 1C32:03, 1C32:05, 1C32:06, 1C32:09, 1C33:03, 1C33:06, 1C33:09 werden mit den maximal gemessenen Werten aktualisiert.                      Wenn erneut gemessen wird, werden die Messwerte zurückgesetzt</p>	UINT16	RW	0x0000 (0 <sub>dez</sub> )
1C32:09	Delay time	Zeit zwischen SYNC1 Event und Ausgabe der Outputs (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C32:0B	SM event missed counter	Anzahl der ausgefallenen SM-Events im OPERATIONAL (nur im DC Mode)	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C32:0C	Cycle exceeded counter	Anzahl der Zykluszeitverletzungen im OPERATIONAL (Zyklus wurde nicht rechtzeitig fertig bzw. der nächste Zyklus kam zu früh)	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C32:0D	Shift too short counter	Anzahl der zu kurzen Abstände zwischen SYNC0 und SYNC1 Event (nur im DC Mode)	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C32:20	Sync error	Im letzten Zyklus war die Synchronisierung nicht korrekt (Ausgänge wurden zu spät ausgegeben, nur im DC Mode)	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )

**Index 0x1C33 SM input parameter**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C33:0	SM input parameter	Synchronisierungsparameter der Inputs	UINT8	RO	0x20 (32 <sub>dez</sub> )
1C33:01	Sync mode	Aktuelle Synchronisierungsbetriebsart: <ul style="list-style-type: none"> <li>• 0: Free Run</li> <li>• 1: Synchron with SM 3 Event (keine Outputs vorhanden)</li> <li>• 2: DC - Synchron with SYNC0 Event</li> <li>• 3: DC - Synchron with SYNC1 Event</li> <li>• 34: Synchron with SM 2 Event (Outputs vorhanden)</li> </ul>	UINT16	RW	0x0000 (0 <sub>dez</sub> )
1C33:02	Cycle time	wie 1C32:02	UINT32	RW	0x0003D090 (250000 <sub>dez</sub> )
1C33:03	Shift time	Zeit zwischen SYNC0-Event und Einlesen der Inputs (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C33:04	Sync modes supported	Unterstützte Synchronisierungsbetriebsarten: <ul style="list-style-type: none"> <li>• Bit 0: Free Run wird unterstützt</li> <li>• Bit 1: Synchron with SM 2 Event wird unterstützt (Outputs vorhanden)</li> <li>• Bit 1: Synchron with SM 3 Event wird unterstützt (keine Outputs vorhanden)</li> <li>• Bit 2-3 = 01: DC-Mode wird unterstützt</li> <li>• Bit 4-5 = 01: Input Shift durch lokales Ereignis (Outputs vorhanden)</li> <li>• Bit 4-5 = 10: Input Shift mit SYNC1 Event (keine Outputs vorhanden)</li> <li>• Bit 14 = 1: dynamische Zeiten (Messen durch Beschreiben von 1C32:08 oder 1C33:08)</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dez</sub> )
1C33:05	Minimum cycle time	wie 1C32:05	UINT32	RO	0x00004E20 (20000 <sub>dez</sub> )
1C33:06	Calc and copy time	Zeit zwischen Einlesen der Eingänge und Verfügbarkeit der Eingänge für den Master (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C33:08	Command	wie 1C32:08	UINT16	RW	0x0000 (0 <sub>dez</sub> )
1C33:09	Delay time	Zeit zwischen SYNC1-Event und Einlesen der Eingänge (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 <sub>dez</sub> )
1C33:0B	SM event missed counter	wie 1C32:11	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C33:0C	Cycle exceeded counter	wie 1C32:12	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C33:0D	Shift too short counter	wie 1C32:13	UINT16	RO	0x0000 (0 <sub>dez</sub> )
1C33:20	Sync error	wie 1C32:32	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )

**7.1.3 Profilspezifische Objekte (0x6000-0xFFFF) [ab Hardwarestand 03]**

Die profilspezifischen Objekte haben für alle EtherCAT Slaves, die das Profil 5001 unterstützen, die gleiche Bedeutung.

**Index 0x60n0 COM Inputs Ch. 1 (n = 0), Ch. 2 (n = 1)**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default	
60n0:0	COM Inputs Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x26 (38 <sub>dez</sub> )	
60n0:01	Transmit accepted	Die Klemme quittiert die Entgegennahme von Daten mit Zustandsänderung dieses Bits. Erst daraufhin werden neue Daten von der Steuerung zur Klemme übertragen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )	
60n0:02	Receive request	Über eine Zustandsänderung dieses Bits teilt die Klemme der Steuerung mit, dass sich die in "Input length [▶_191]" angezeigte Anzahl von Bytes in den DataIn-Bytes befinden. Die Steuerung muss die Entgegennahme der Daten mit Zustandsänderung des ReceiveAccepted [▶_192]-Bits quittieren. Erst daraufhin werden neue Daten von der Klemme zur Steuerung übertragen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )	
60n0:03	Init accepted	0	Die Klemme ist wieder für den seriellen Datenaustausch bereit.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
		1				
60n0:04	Buffer full	Das Empfangs-FIFO ist voll. Ab diesem Zeitpunkt gehen alle eingehenden Daten verloren!	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )	
60n0:05	Parity error	Es ist ein Parity-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )	
60n0:06	Framing error	Es ist ein Framing-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )	
60n0:07	Overrun error	Es ist ein Overrun-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )	
60n0:09	Input length	Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.	UINT8	RO	0x00 (0 <sub>dez</sub> )	
60n0:11	Data In 0	Eingangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )	
...	...	....	...	...	...	
60n0:26	Data In 21	Eingangs-Byte 21	UINT8	RO	0x00 (0 <sub>dez</sub> )	

**Index 0x60n1 Status Ch. 1 (n = 0), Ch. 2 (n = 1)**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
60n1:0	Status Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x01 (01 <sub>dez</sub> )
60n1:01	Status	Status-Wort	UINT16	RO	0x00 (0 <sub>dez</sub> )

**Index 0x70n0 COM Outputs Ch. 1 (n = 0), Ch. 2 (n = 1)**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
70n0:0	COM Outputs Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x26 (38 <sub>dez</sub> )
70n0:01	Transmit request	Über eine Zustandsänderung dieses Bits teilt die Steuerung der Klemme mit, dass sich die in mit den " <u>Output length</u> [▶_192]" angezeigte Anzahl von Bytes in den DataOut-Bytes befinden. Die Klemme quittiert die Entgegennahme der Daten mit Zustandsänderung des Bits " <u>TransmitAccepted</u> [▶_191]". Erst daraufhin werden neue Daten von der Steuerung zur Klemme übertragen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
70n0:02	Receive accepted	Die Steuerung quittiert die Entgegennahme von Daten mit Zustandsänderung dieses Bits. Erst daraufhin werden neue Daten von der Klemme zur Steuerung übertragen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
70n0:03	Init request	0 Die Steuerung fordert von der Klemme wieder die Bereitschaft für den seriellen Datenaustausch. 1 Die Steuerung fordert die Klemme zur Initialisierung auf. Die Sende- und Empfangsfunktionen werden gesperrt, die FIFO-Zeiger werden zurückgesetzt und die Schnittstelle wird mit den Werten des zuständigen Settings-Objektes initialisiert. Die Ausführung der Initialisierung wird von der Klemme mit dem Bit " <u>Init accepted</u> [▶_191]" quittiert.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
70n0:04	Send continuous	Kontinuierliches Senden der Daten aus dem FIFO. Über die Steuerung wird der Sendebuffer gefüllt (bis zu 128 Byte). Mit steigender Flanke des Bits wird der gefüllte Buffer-Inhalt gesendet. Sind die Daten übertragen, so wird dies durch das Setzen des Bits " <u>Init accepted</u> [▶_191]" von der Klemme an die Steuerung quittiert. " <u>Init accepted</u> [▶_191]" wird mit " <u>SendContinuous</u> [▶_192]" zurückgenommen.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
70n0:09	Output length	Anzahl der Ausgangs-Bytes, die für die Übertragung von der Steuerung zur Klemme bereit stehen.	UINT8	RO	0x00 (0 <sub>dez</sub> )
70n0:11	Data Out 0	Ausgangs-Byte 0	UINT8	RO	0x00 (0 <sub>dez</sub> )
...	...	....	...	...	...
70n0:26	Data Out 21	Ausgangs-Byte 21	UINT8	RO	0x00 (0 <sub>dez</sub> )

**Index 0x70n1 Ctrl Ch. 1 (n = 0), Ch. 2 (n = 1)**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
70n1:0	Ctrl Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x01 (01 <sub>dez</sub> )
70n1:01	Status	Control-Wort	UINT16	RO	0x00 (0 <sub>dez</sub> )

**Index 0xA0n0 COM Diag data Ch. 1 (n = 0), Ch. 2 (n = 1)**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
A0n0:0	COM Diag data Ch. 1 + Ch. 2	Max. SubIndex (hex)	UINT8	RO	0x12 (18 <sub>dez</sub> )
A0n0:01	Buffer overflow	Es ist ein Buffer-Overflow aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A0n0:02	Parity error	Es ist ein Parity-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A0n0:03	Framing error	Es ist ein Framing-Error aufgetreten	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A0n0:04	Overrun error	Es ist ein Overrun-Error aufgetreten.	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A0n0:05	Buffer full	Das Empfangs-FIFO ist voll. Ab diesem Zeitpunkt gehen alle eingehenden Daten verloren!	BOOLEAN	RO	0x00 (0 <sub>dez</sub> )
A0n0:11	Data bytes in send buffer	Anzahl der Datenbytes im Sende-FIFO	UINT16	RO	0x0000 (0 <sub>dez</sub> )
A0n0:21	Data bytes in receive buffer	Anzahl der Datenbytes im Empfangs-FIFO	UINT16	RO	0x0000 (0 <sub>dez</sub> )

**Index 0xF000 Modular device profile**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F000:0	Modular device profile	Allgemeine Informationen des Modular Device Profiles	UINT8	RO	0x02 (2 <sub>dez</sub> )
F000:01	Module Index 0xdistance	Index (hex)abstand der Objekte der einzelnen Kanäle	UINT16	RO	0x0010 (16 <sub>dez</sub> )
F000:02	Maximum number of modules	Anzahl der Kanäle	UINT16	RO	0x0004 (4 <sub>dez</sub> )

**Index 0xF008 Code word**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F008:0	Code word	reserviert	UINT32	RW	0x00000000 (0 <sub>dez</sub> )

**Index 0xF010 Module list**

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F010:0	Module list	Max. SubIndex (hex)	UINT8	RW	0x04 (4 <sub>dez</sub> )
F010:01	SubIndex 0x001	-	UINT32	RW	0x00000258 (600 <sub>dez</sub> )
F010:02	SubIndex 0x002	-	UINT32	RW	0x00000258 (600 <sub>dez</sub> )

## 7.2 Control- und Status-Daten

Die Control- und Status-Daten befinden sich in den ersten 16 Bit des Eingangs- und Ausgangsprozessabbildes. Über sie wird die Kommunikation zwischen der Klemme und der Steuerung kontrolliert.

**Status-Daten**

Bit-Position	Name	Bedeutung	Data type
0	Transmit accepted	Die Klemme quittiert die Entgegennahme von Daten mit Zustandsänderung dieses Bits. Erst daraufhin werden neue Daten von der Steuerung zur Klemme übertragen.	BOOLEAN
1	Receive request	Über eine Zustandsänderung dieses Bits teilt die Klemme der Steuerung mit, dass sich die in "Input length" angezeigte Anzahl von Bytes in den DataIn-Bytes befinden. Die Steuerung muss die Entgegennahme der Daten mit Zustandsänderung des ReceiveAccepted-Bits quittieren. Erst daraufhin werden neue Daten von der Klemme zur Steuerung übertragen.	BOOLEAN
2	Init accepted	0 Die Klemme ist wieder für den seriellen Datenaustausch bereit.	BOOLEAN
		1 Die Initialisierung wurde von der Klemme ausgeführt.	
3	Buffer full	Das Empfangs-FIFO ist voll. Ab diesem Zeitpunkt gehen alle eingehenden Daten verloren!	BOOLEAN
4	Parity error	Es ist ein Parity-Error aufgetreten.	BOOLEAN
5	Framing error	Es ist ein Framing-Error aufgetreten.	BOOLEAN
6	Overrun error	Es ist ein Overrun-Error aufgetreten.	BOOLEAN
7	-		
8...15	Input length	Anzahl der Eingangs-Bytes, die für die Übertragung von der Klemme zur Steuerung bereit stehen.	UINT8

**Control-Daten**

Bit-Position	Name	Bedeutung	Data type
0	Transmit request	Über eine Zustandsänderung dieses Bits teilt die Steuerung der Klemme mit, dass sich die in mit den "Output length" angezeigte Anzahl von Bytes in den DataOut-Bytes befinden. Die Klemme quittiert die Entgegennahme der Daten mit Zustandsänderung des Bits "TransmitAccepted". Erst daraufhin werden neue Daten von der Steuerung zur Klemme übertragen.	BOOLEAN
1	Receive accepted	Die Steuerung quittiert die Entgegennahme von Daten mit Zustandsänderung dieses Bits. Erst daraufhin werden neue Daten von der Klemme zur Steuerung übertragen.	BOOLEAN
2	Init request	0 Die Steuerung fordert von der Klemme wieder die Bereitschaft für den seriellen Datenaustausch.	BOOLEAN
		1 Die Steuerung fordert die Klemme zur Initialisierung auf. Die Sende- und Empfangsfunktionen werden gesperrt, die FIFO-Zeiger werden zurückgesetzt und die Schnittstelle wird mit den Werten des zuständigen Settings-Objektes initialisiert. Die Ausführung der Initialisierung wird von der Klemme mit dem Bit "Init accepted" quittiert.	
3	Send continuous	Kontinuierliches Senden der Daten aus dem FIFO. Über die Steuerung wird der Sendebuffer gefüllt (bis zu 128 Byte). Mit steigender Flanke des Bits wird der gefüllte Buffer-Inhalt gesendet. Sind die Daten übertragen, so wird dies durch das Setzen des Bits "Init accepted" von der Klemme an die Steuerung quittiert. "Init accepted" wird mit "SendContinuous" zurückgenommen.	BOOLEAN
4...7	-		
8...15	Output length	Anzahl der Ausgangs-Bytes, die für die Übertragung von der Steuerung zur Klemme bereit stehen.	UINT8

**PDO Zuordnung**

Die Klemme stellt für jeden Kanal zwei Ein-/Ausgangsprozessabbilder zur Verfügung. Diese unterscheiden sich lediglich durch die Darstellung der Control/Status-Daten.

Bei der bitweisen Darstellung werden die Daten wie in den obigen Tabellen zur Verfügung gestellt. Bei der wortweisen Darstellung werden die ersten 16 Bit der Prozessdaten in einem Wort zusammengefasst.

Standardmäßig beginnen die Prozessdatenobjekte mit einem Control/Status-Wort. Für die Verwendung der Library "TwinCAT-PLC-Serial-Communication" wird diese PDO-Zuordnung benötigt.

Status Inputs	Control Outputs	Darstellung
0x1A00	0x1600	bitweise Ch.1
0x1A01	0x1601	bitweise Ch.2
0x1A02	0x1602	bitweise Ch.3
0x1A03	0x1603	bitweise Ch.4
0x1A04	0x1604	wortweise Ch 1
0x1A05	0x1605	wortweise Ch 2
0x1A06	0x1606	wortweise Ch 3
0x1A07	0x1607	wortweise Ch 4

## 8 Anhang

### 8.1 EtherCAT AL Status Codes

Detaillierte Informationen hierzu entnehmen Sie bitte der vollständigen [EtherCAT-Systembeschreibung](#).

### 8.2 Firmware Kompatibilität

**Anmerkung**

- Es wird empfohlen, die für die jeweilige Hardware letztmögliche Firmware einzusetzen.
- Ein Anspruch auf ein kostenfreies Firmware-Update bei ausgelieferten Produkten durch Beckhoff gegenüber dem Kunden besteht nicht.

**HINWEIS**

**Beschädigung des Gerätes möglich**

Beachten Sie die Hinweise zum Firmware Update auf der [gesonderten Seite \[► 196\]](#).

Wird ein Gerät in den BOOTSTRAP-Mode zum Firmware-Update versetzt, prüft es u.U. beim Download nicht, ob die neue Firmware geeignet ist.

Dadurch kann es zur Beschädigung des Gerätes kommen!

Vergewissern Sie sich daher immer, ob die Firmware für den Hardware-Stand des Gerätes geeignet ist!

<b>EL6001</b>			
<b>Hardware (HW)</b>	<b>Firmware (FW)</b>	<b>Revision Nr.</b>	<b>Release-Datum</b>
00 - 02	01		05/2006
	02		12/2006
	03		04/2008
	04		04/2008
04 - 13*	05	EL6001-0000-0016	12/2009
	06		04/2010
	07	EL6001-0000-0017	03/2011
		EL6001-0000-0018	10/2012
	08	EL6001-0000-0019	05/2014
	09	EL6001-0000-0020	08/2014
	10		05/2015
11*		06/2017	

<b>EL6002</b>			
<b>Hardware (HW)</b>	<b>Firmware (FW)</b>	<b>Revision Nr.</b>	<b>Release-Datum</b>
00 – 16*	01	EL6002-0000-0016	12/2009
	02		06/2010
	03	EL6002-0000-0017	11/2012
		EL6002-0000-0018	08/2013
	04*	EL6002-0000-0019	05/2014

EL6021			
Hardware (HW)	Firmware (FW)	Revision Nr.	Release-Datum
00 - 02	01		05/2005
	02		12/2006
	03		04/2008
03 - 13*	04	EL6021-0000-0016	11/2009
	05		04/2010
		EL6021-0000-0017	10/2012
	06	EL6021-0000-0018	08/2013
	07	EL6021-0000-0019	06/2014
	08	EL6021-0000-0020	10/2014
	09*		01/2020

EL6022				
Hardware (HW)	Firmware (FW)	Revision Nr.	Release-Datum	
00 - 17*	01	EL6022-0000-0016	01/2010	
	02		06/2010	
	03*			09/2011
		EL6022-0000-0017	08/2012	
		EL6022-0000-0018	08/2013	
		EL6022-0000-0019	03/2015	

\*) Zum Zeitpunkt der Erstellung dieser Dokumentation ist dies der aktuelle kompatible Firmware/Hardware-Stand. Überprüfen Sie auf der Beckhoff Webseite, ob eine aktuellere Dokumentation vorliegt.

### 8.3 Firmware Update EL/ES/ELM/EM/EPxxxx

Dieses Kapitel beschreibt das Geräte-Update für Beckhoff EtherCAT Slaves der Serien EL/ES, ELM, EM, EK und EP. Ein FW-Update sollte nur nach Rücksprache mit dem Beckhoff Support durchgeführt werden.

#### Speicherorte

In einem EtherCAT-Slave werden an bis zu drei Orten Daten für den Betrieb vorgehalten:

- Je nach Funktionsumfang und Performance besitzen EtherCAT Slaves einen oder mehrere lokale Controller zur Verarbeitung von IO-Daten. Das darauf laufende Programm ist die sog. **Firmware** im Format \*.efw.
- In bestimmten EtherCAT Slaves kann auch die EtherCAT Kommunikation in diesen Controller integriert sein. Dann ist der Controller meist ein so genannter **FPGA**-Chip mit der \*.rbf-Firmware.
- Darüber hinaus besitzt jeder EtherCAT Slave einen Speicherchip, um seine eigene Gerätebeschreibung (ESI; EtherCAT Slave Information) zu speichern, in einem sog. **ESI-EEPROM**. Beim Einschalten wird diese Beschreibung geladen und u. a. die EtherCAT Kommunikation entsprechend eingerichtet. Die Gerätebeschreibung kann von der Beckhoff Website (<http://www.beckhoff.de>) im Downloadbereich heruntergeladen werden. Dort sind alle ESI-Dateien als Zip-Datei zugänglich.

Kundenseitig zugänglich sind diese Daten nur über den Feldbus EtherCAT und seine Kommunikationsmechanismen. Beim Update oder Auslesen dieser Daten ist insbesondere die azyklische Mailbox-Kommunikation oder der Registerzugriff auf den ESC in Benutzung.

Der TwinCAT Systemmanager bietet Mechanismen, um alle drei Teile mit neuen Daten programmieren zu können, wenn der Slave dafür vorgesehen ist. Es findet üblicherweise keine Kontrolle durch den Slave statt, ob die neuen Daten für ihn geeignet sind, ggf. ist ein Weiterbetrieb nicht mehr möglich.

**Vereinfachtes Update per Bundle-Firmware**

Bequemer ist der Update per sog. **Bundle-Firmware**: hier sind die Controller-Firmware und die ESI-Beschreibung in einer \*.efw-Datei zusammengefasst, beim Update wird in der Klemme sowohl die Firmware, als auch die ESI verändert. Dazu ist erforderlich

- dass die Firmware in dem gepackten Format vorliegt: erkenntlich an dem Dateinamen der auch die Revisionsnummer enthält, z. B. ELxxx-xxx\_REV0016\_SW01.efw
- dass im Download-Dialog das Passwort=1 angegeben wird. Bei Passwort=0 (default Einstellung) wird nur das Firmware-Update durchgeführt, ohne ESI-Update.
- dass das Gerät diese Funktion unterstützt. Die Funktion kann in der Regel nicht nachgerüstet werden, sie wird Bestandteil vieler Neuentwicklungen ab Baujahr 2016.

Nach dem Update sollte eine Erfolgskontrolle durchgeführt werden

- ESI/Revision: z. B. durch einen Online-Scan im TwinCAT ConfigMode/FreeRun – dadurch wird die Revision bequem ermittelt
- Firmware: z. B. durch einen Blick ins Online-CoE des Gerätes

**HINWEIS**

**Beschädigung des Gerätes möglich!**

- ✓ Beim Herunterladen von neuen Gerätedateien ist zu beachten

- Das Herunterladen der Firmware auf ein EtherCAT-Gerät darf nicht unterbrochen werden.
- Eine einwandfreie EtherCAT-Kommunikation muss sichergestellt sein, CRC-Fehler oder LostFrames dürfen nicht auftreten.
- Die Spannungsversorgung muss ausreichend dimensioniert, die Pegel entsprechend der Vorgabe sein.

⇒ Bei Störungen während des Updatevorgangs kann das EtherCAT-Gerät ggf. nur vom Hersteller wieder in Betrieb genommen werden!

**8.3.1 Gerätebeschreibung ESI-File/XML**

**HINWEIS**

**ACHTUNG bei Update der ESI-Beschreibung/EEPROM**

Manche Slaves haben Abgleich- und Konfigurationsdaten aus der Produktion im EEPROM abgelegt. Diese werden bei einem Update unwiederbringlich überschrieben.

Die Gerätebeschreibung ESI wird auf dem Slave lokal gespeichert und beim Start geladen. Jede Gerätebeschreibung hat eine eindeutige Kennung aus Slave-Name (9-stellig) und Revision-Nummer (4-stellig). Jeder im System Manager konfigurierte Slave zeigt seine Kennung im EtherCAT-Reiter:

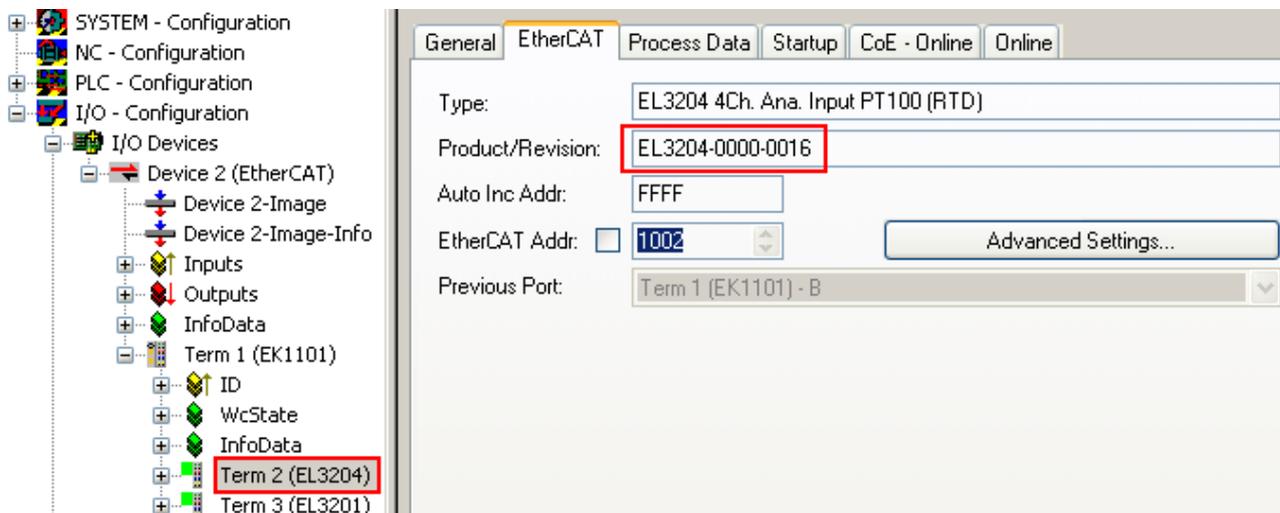


Abb. 169: Geräteerkennung aus Name EL3204-0000 und Revision -0016

Die konfigurierte Kennung muss kompatibel sein mit der tatsächlich als Hardware eingesetzten Gerätebeschreibung, d. h. der Beschreibung die der Slave (hier: EL3204) beim Start geladen hat. Üblicherweise muss dazu die konfigurierte Revision gleich oder niedriger der tatsächlich im Klemmenverbund befindlichen sein.

Weitere Hinweise hierzu entnehmen Sie bitte der [EtherCAT System-Dokumentation](#).

### **i** Update von XML/ESI-Beschreibung

Die Geräteversion steht in engem Zusammenhang mit der verwendeten Firmware bzw. Hardware. Nicht kompatible Kombinationen führen mindestens zu Fehlfunktionen oder sogar zur endgültigen Außerbetriebsetzung des Gerätes. Ein entsprechendes Update sollte nur in Rücksprache mit dem Beckhoff Support ausgeführt werden.

### Anzeige der Slave-Kennung ESI

Der einfachste Weg die Übereinstimmung von konfigurierter und tatsächlicher Gerätebeschreibung festzustellen, ist im TwinCAT-Modus Config/FreeRun das Scannen der EtherCAT-Boxen auszuführen:

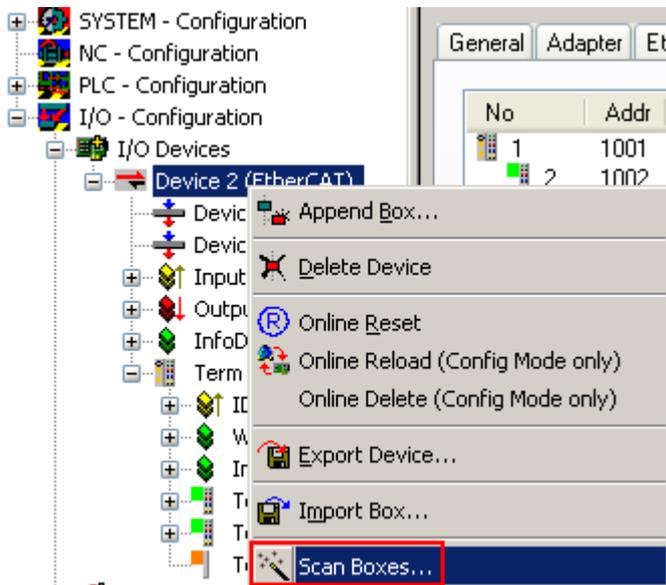


Abb. 170: Rechtsklick auf das EtherCAT Gerät bewirkt das Scannen des unterlagerten Feldes

Wenn das gefundene Feld mit dem konfigurierten übereinstimmt, erscheint



Abb. 171: Konfiguration identisch

ansonsten erscheint ein Änderungsdialog, um die realen Angaben in die Konfiguration zu übernehmen.

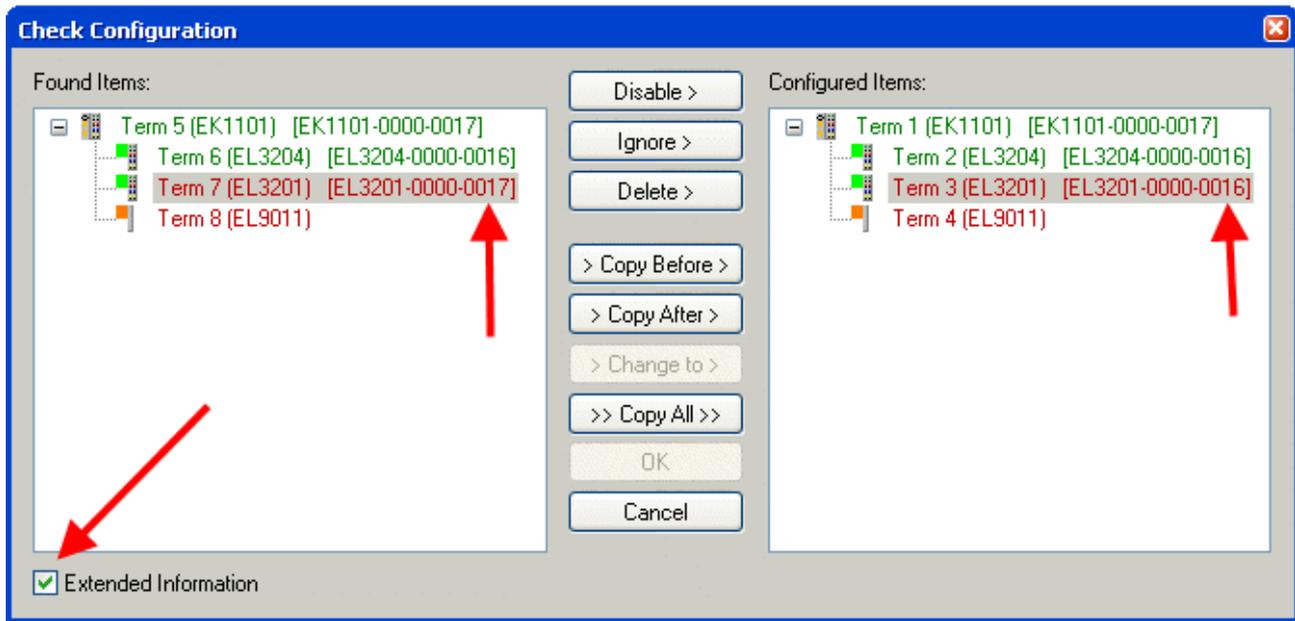


Abb. 172: Änderungsdialog

In diesem Beispiel in Abb. *Änderungsdialog*, wurde eine EL3201-0000-0017 vorgefunden, während eine EL3201-0000-0016 konfiguriert wurde. In diesem Fall bietet es sich an, mit dem *Copy Before*-Button die Konfiguration anzupassen. Die Checkbox *Extended Information* muss gesetzt werden, um die Revision angezeigt zu bekommen.

**Änderung der Slave-Kennung ESI**

Die ESI/EEPROM-Kennung kann unter TwinCAT wie folgt aktualisiert werden:

- Es muss eine einwandfreie EtherCAT-Kommunikation zum Slave hergestellt werden
- Der State des Slave ist unerheblich
- Rechtsklick auf den Slave in der Online-Anzeige führt zum Dialog *EEPROM Update*, Abb. *EEPROM Update*

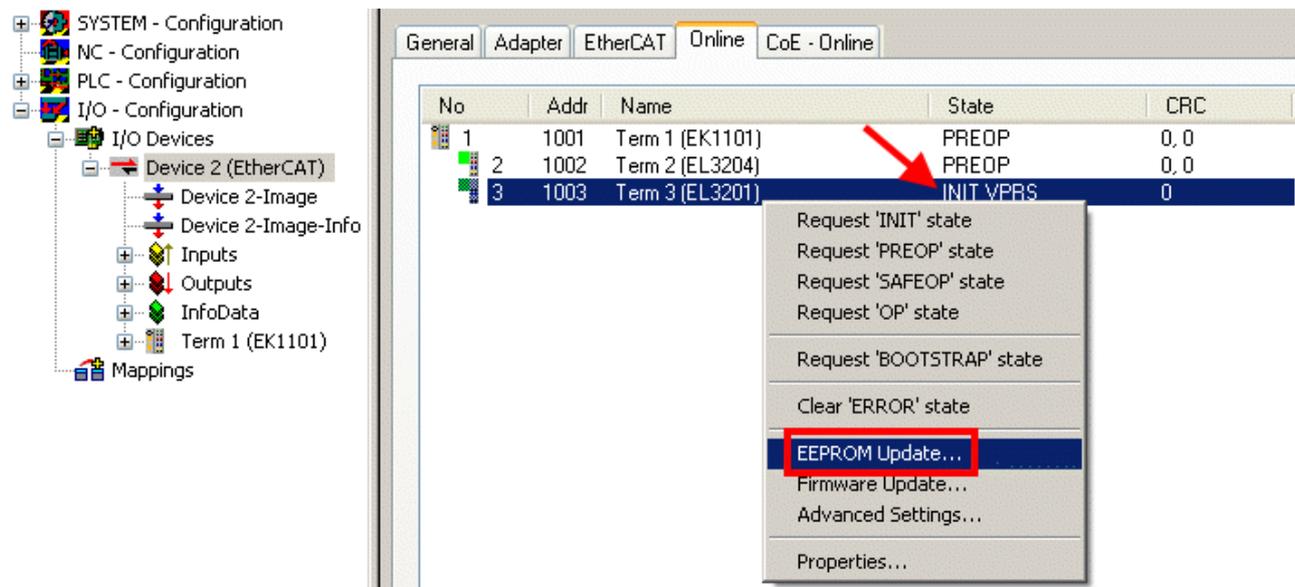


Abb. 173: EEPROM Update

Im folgenden Dialog wird die neue ESI-Beschreibung ausgewählt, s. Abb. *Auswahl des neuen ESI*. Die CheckBox *Show Hidden Devices* zeigt auch ältere, normalerweise ausgeblendete Ausgaben eines Slave.

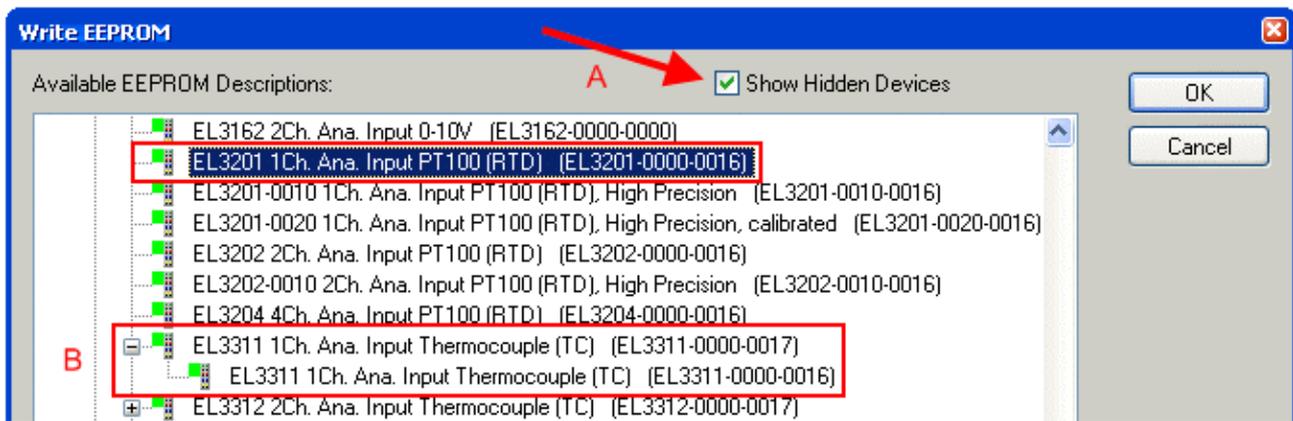


Abb. 174: Auswahl des neuen ESI

Ein Laufbalken im System Manager zeigt den Fortschritt - erst erfolgt das Schreiben, dann das Verifying.

### **i** Änderung erst nach Neustart wirksam

Die meisten EtherCAT-Geräte lesen eine geänderte ESI-Beschreibung umgehend bzw. nach dem Aufstarten aus dem INIT ein. Einige Kommunikationseinstellungen wie z. B. Distributed Clocks werden jedoch erst bei PowerOn gelesen. Deshalb ist ein kurzes Abschalten des EtherCAT Slave nötig, damit die Änderung wirksam wird.

## 8.3.2 Erläuterungen zur Firmware

### Versionsbestimmung der Firmware

#### Versionsbestimmung nach Laseraufdruck

Auf einem Beckhoff EtherCAT Slave ist eine Seriennummer aufgelasert. Der Aufbau der Seriennummer lautet: **KK YY FF HH**

KK - Produktionswoche (Kalenderwoche)

YY - Produktionsjahr

FF - Firmware-Stand

HH - Hardware-Stand

Beispiel mit Ser. Nr.: 12 10 03 02:

12 - Produktionswoche 12

10 - Produktionsjahr 2010

03 - Firmware-Stand 03

02 - Hardware-Stand 02

#### Versionsbestimmung mit dem System-Manager

Der TwinCAT System-Manager zeigt die Version der Controller-Firmware an, wenn der Slave online für den Master zugänglich ist. Klicken Sie hierzu auf die E-Bus-Klemme deren Controller-Firmware Sie überprüfen möchten (im Beispiel Klemme 2 (EL3204) und wählen Sie den Karteireiter *CoE-Online* (CAN over EtherCAT).

**i CoE-Online und Offline-CoE**

Es existieren zwei CoE-Verzeichnisse:

- **online:** es wird im EtherCAT Slave vom Controller angeboten, wenn der EtherCAT Slave dies unterstützt. Dieses CoE-Verzeichnis kann nur bei angeschlossenem und betriebsbereitem Slave angezeigt werden.
- **offline:** in der EtherCAT Slave Information ESI/XML kann der Default-Inhalt des CoE enthalten sein. Dieses CoE-Verzeichnis kann nur angezeigt werden, wenn es in der ESI (z. B. „Beckhoff EL5xx.xml“) enthalten ist.

Die Umschaltung zwischen beiden Ansichten kann über den Button *Advanced* vorgenommen werden.

In Abb. *Anzeige FW-Stand EL3204* wird der FW-Stand der markierten EL3204 in CoE-Eintrag 0x100A mit 03 angezeigt.

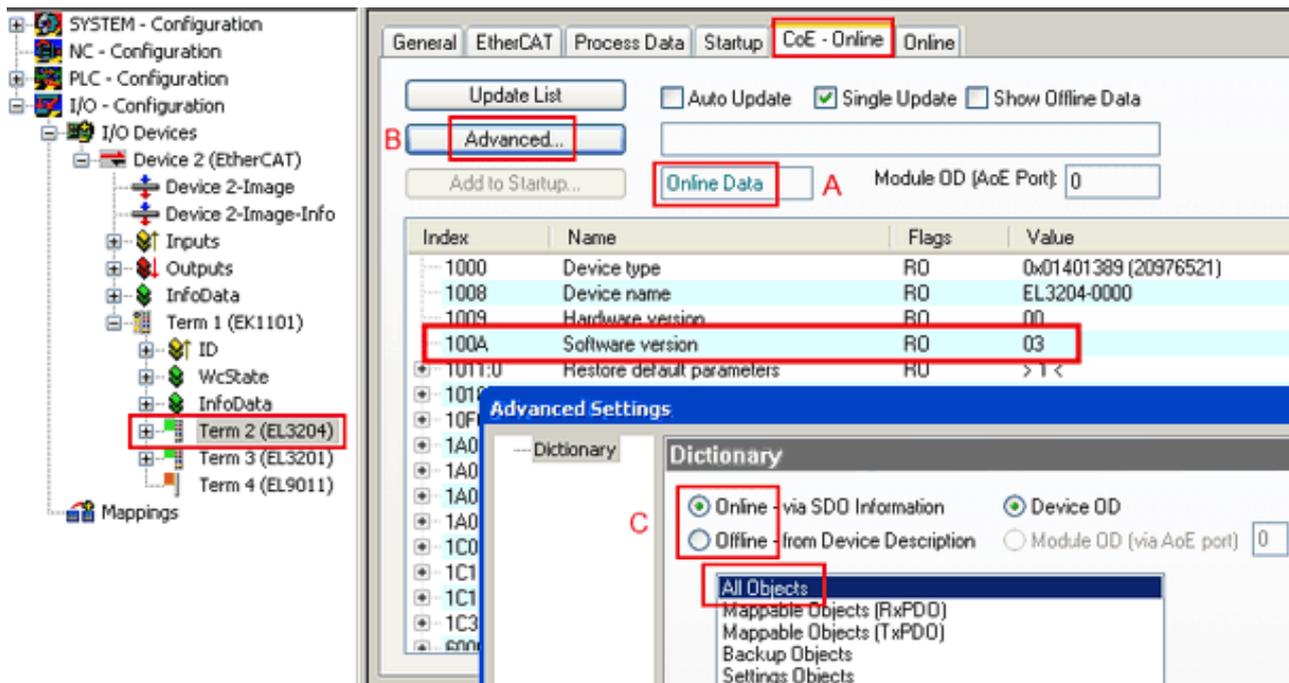


Abb. 175: Anzeige FW-Stand EL3204

TwinCAT 2.11 zeigt in (A) an, dass aktuell das Online-CoE-Verzeichnis angezeigt wird. Ist dies nicht der Fall, kann durch die erweiterten Einstellungen (B) durch *Online* und Doppelklick auf *All Objects* das Online-Verzeichnis geladen werden.

**8.3.3 Update Controller-Firmware \*.efw**

**i CoE-Verzeichnis**

Das Online-CoE-Verzeichnis wird vom Controller verwaltet und in einem eigenen EEPROM gespeichert. Es wird durch ein FW-Update im allgemeinen nicht verändert.

Um die Controller-Firmware eines Slave zu aktualisieren, wechseln Sie zum Karteireiter *Online*, s. Abb. *Firmware Update*.

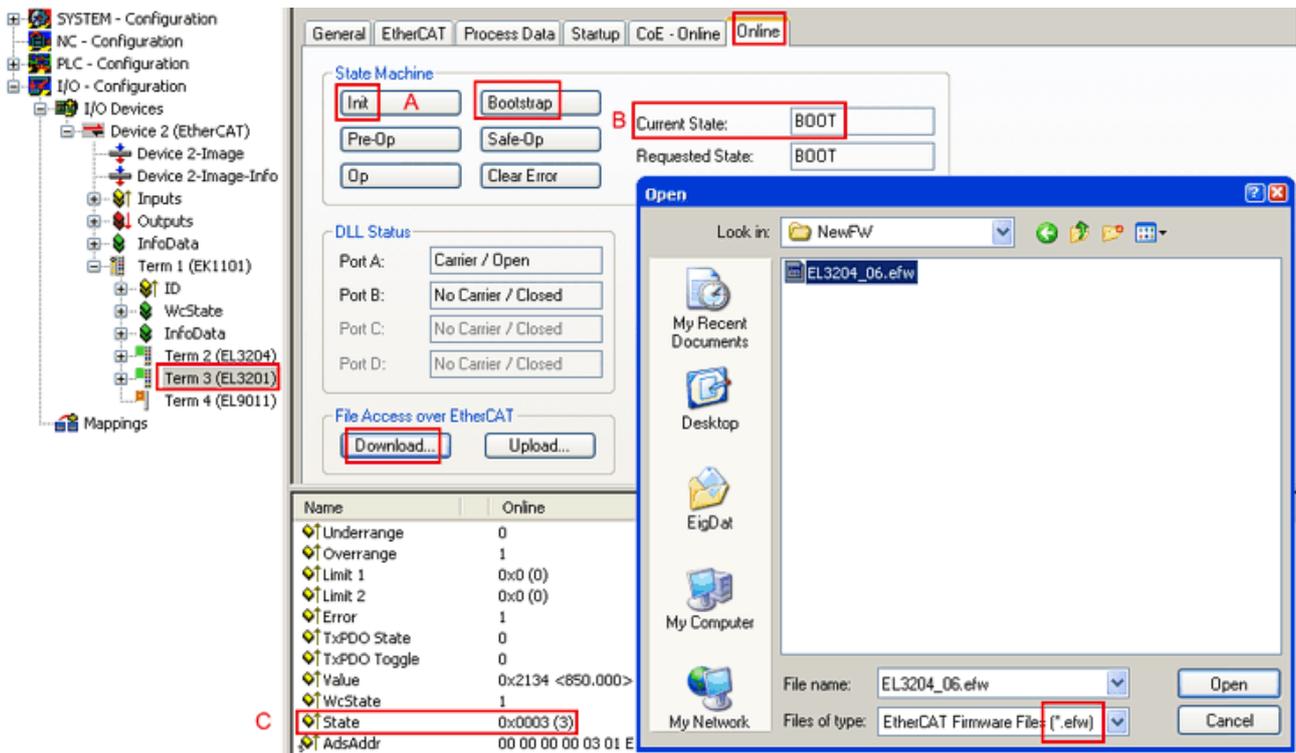
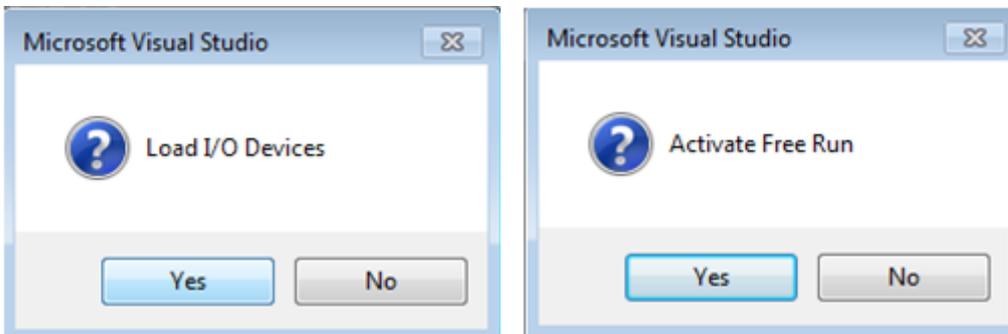


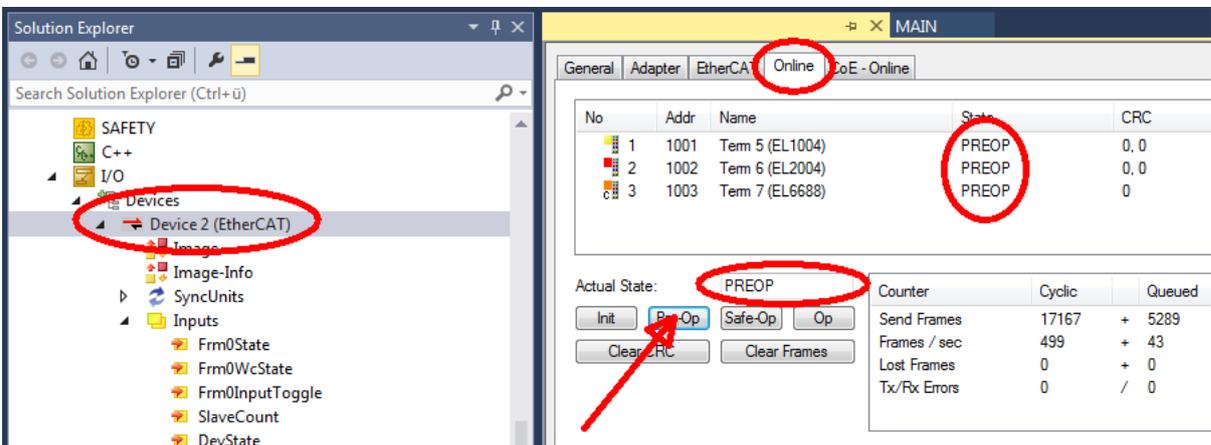
Abb. 176: Firmware Update

Es ist folgender Ablauf einzuhalten, wenn keine anderen Angaben z. B. durch den Beckhoff Support vorliegen. Gültig für TwinCAT 2 und 3 als EtherCAT Master.

- TwinCAT System in ConfigMode/FreeRun mit Zykluszeit  $\geq 1$  ms schalten (default sind im ConfigMode 4 ms). Ein FW-Update während Echtzeitbetrieb ist nicht zu empfehlen.

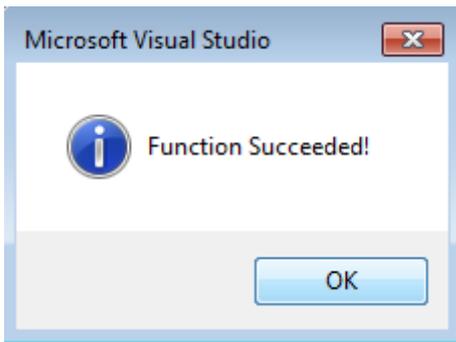


- EtherCAT Master in PreOP schalten



- Slave in INIT schalten (A)
- Slave in BOOTSTRAP schalten

- Kontrolle des aktuellen Status (B, C)
- Download der neuen \*efw-Datei, abwarten bis beendet. Ein Passwort wird in der Regel nicht benötigt.



- Nach Beendigung des Download in INIT schalten, dann in PreOP
- Slave kurz stromlos schalten (nicht unter Spannung ziehen!)
- Im CoE 0x100A kontrollieren ob der FW-Stand korrekt übernommen wurde.

### 8.3.4 FPGA-Firmware \*.rbf

Falls ein FPGA-Chip die EtherCAT-Kommunikation übernimmt, kann ggf. mit einer \*.rbf-Datei ein Update durchgeführt werden.

- Controller-Firmware für die Aufbereitung der E/A-Signale
- FPGA-Firmware für die EtherCAT-Kommunikation (nur für Klemmen mit FPGA)

Die in der Seriennummer der Klemme enthaltene Firmware-Versionsnummer beinhaltet beide Firmware-Teile. Wenn auch nur eine dieser Firmware-Komponenten verändert wird, dann wird diese Versionsnummer fortgeschrieben.

#### Versionsbestimmung mit dem System-Manager

Der TwinCAT System-Manager zeigt die Version der FPGA-Firmware an. Klicken Sie hierzu auf die Ethernet-Karte Ihres EtherCAT-Stranges (im Beispiel Gerät 2) und wählen Sie den Karteireiter *Online*.

Die Spalte *Reg:0002* zeigt die Firmware-Version der einzelnen EtherCAT-Geräte in hexadezimaler und dezimaler Darstellung an.

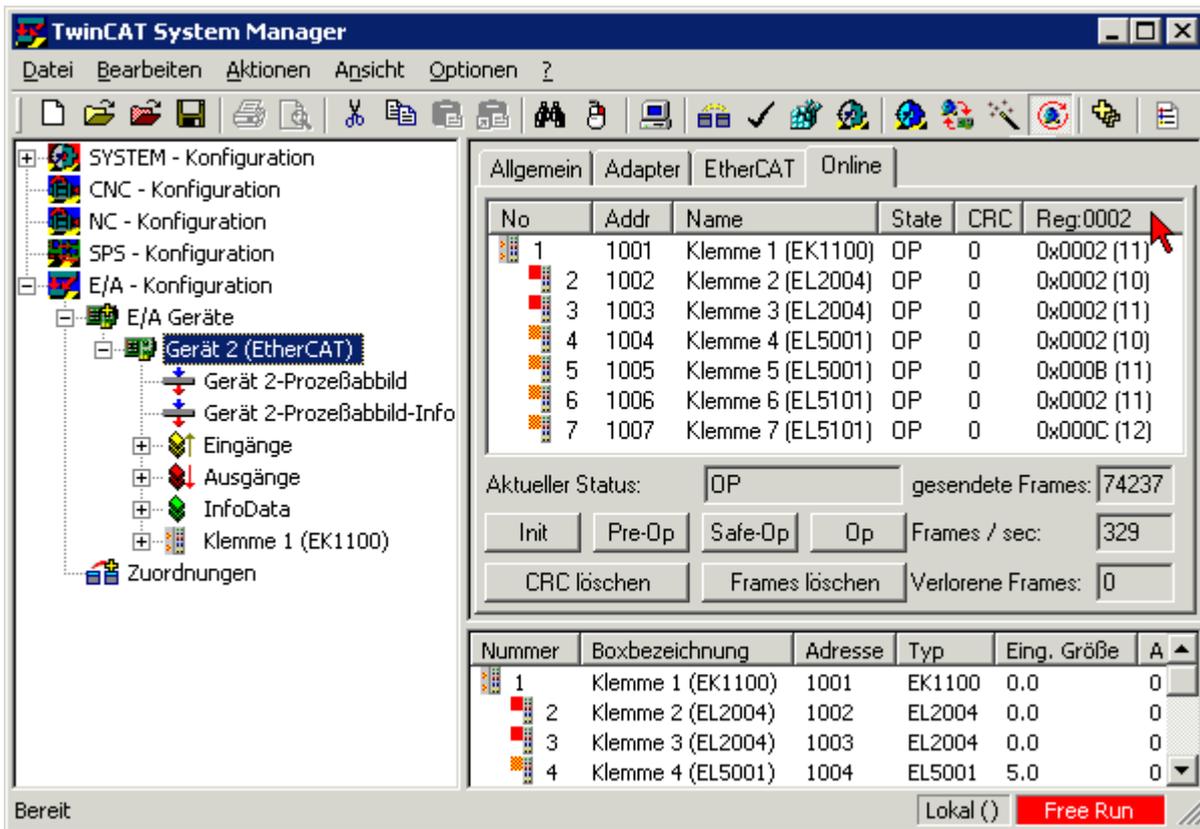
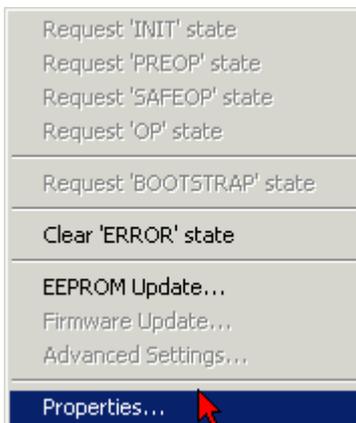


Abb. 177: Versionsbestimmung FPGA-Firmware

Falls die Spalte *Reg:0002* nicht angezeigt wird, klicken sie mit der rechten Maustaste auf den Tabellenkopf und wählen im erscheinenden Kontextmenü, den Menüpunkt *Properties*.

Abb. 178: Kontextmenu *Eigenschaften (Properties)*

In dem folgenden Dialog *Advanced Settings* können Sie festlegen, welche Spalten angezeigt werden sollen. Markieren Sie dort unter *Diagnose/Online Anzeige* das Kontrollkästchen vor *'0002 ETxxxx Build'* um die Anzeige der FPGA-Firmware-Version zu aktivieren.

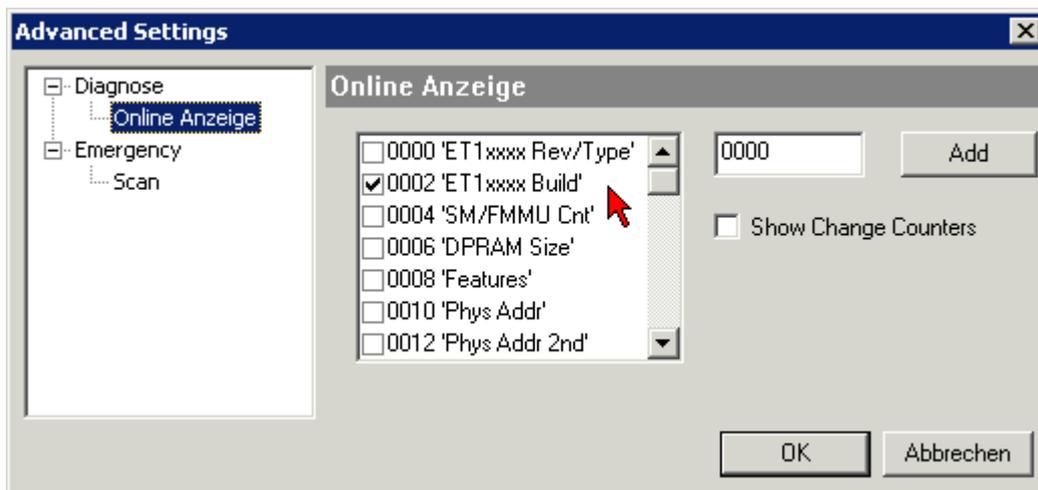


Abb. 179: Dialog *Advanced settings*

### Update

Für das Update der FPGA-Firmware

- eines EtherCAT-Kopplers, muss auf diesem Koppler mindestens die FPGA-Firmware-Version 11 vorhanden sein.
- einer E-Bus-Klemme, muss auf dieser Klemme mindestens die FPGA-Firmware-Version 10 vorhanden sein.

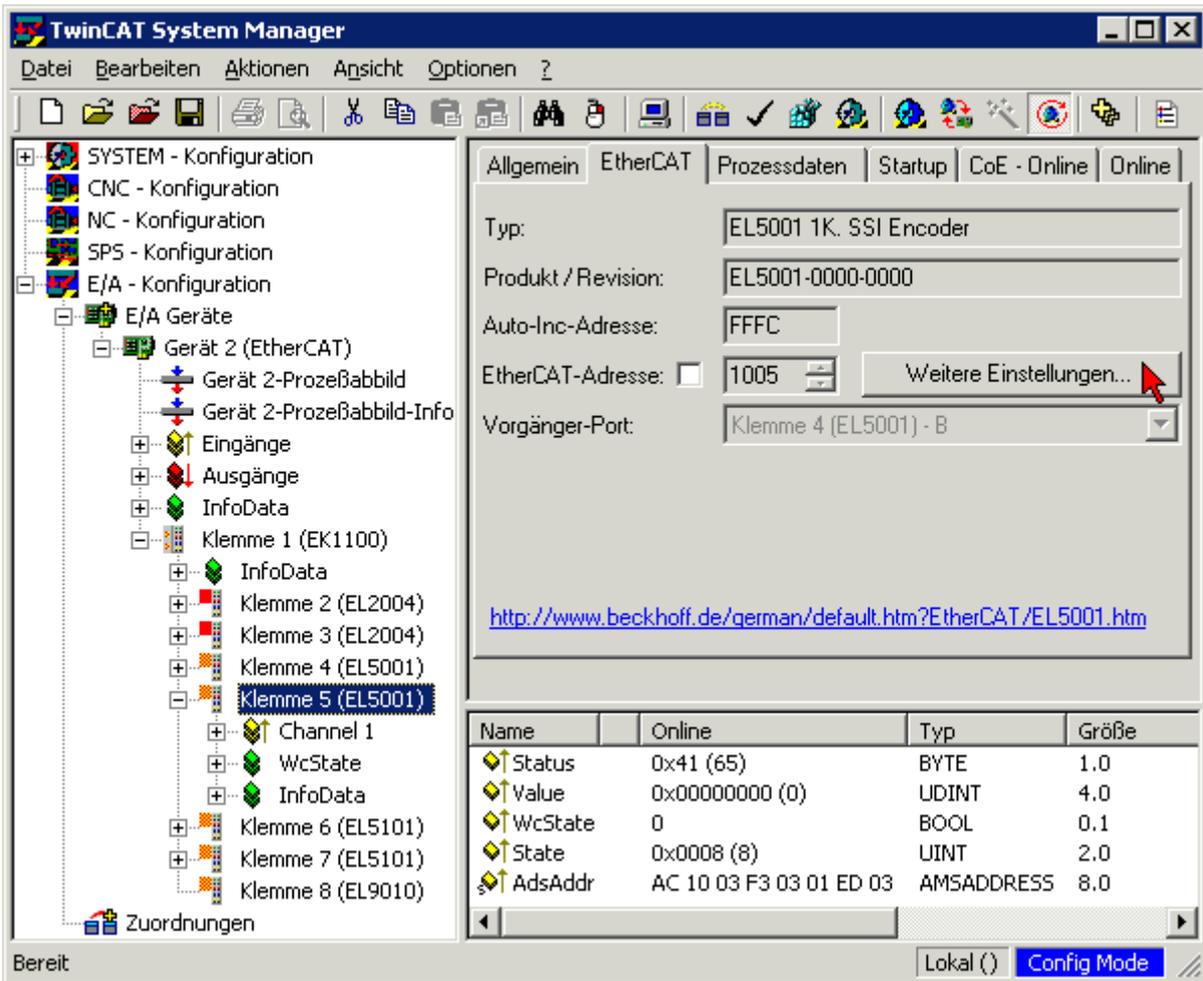
Ältere Firmware-Stände können nur vom Hersteller aktualisiert werden!

### Update eines EtherCAT-Geräts

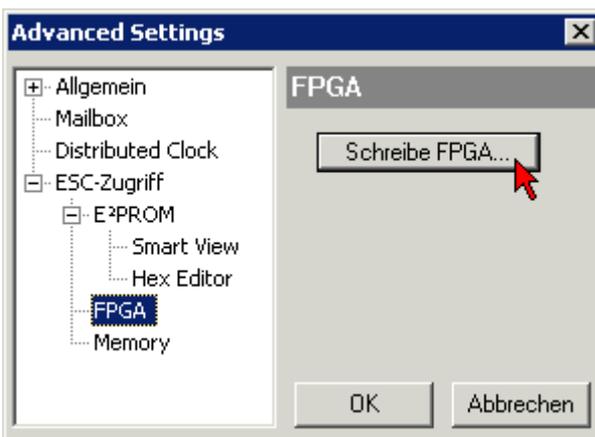
Es ist folgender Ablauf einzuhalten, wenn keine anderen Angaben z. B. durch den Beckhoff Support vorliegen:

- TwinCAT System in ConfigMode/FreeRun mit Zykluszeit  $\geq 1$  ms schalten (default sind im ConfigMode 4 ms). Ein FW-Update während Echtzeitbetrieb ist nicht zu empfehlen.

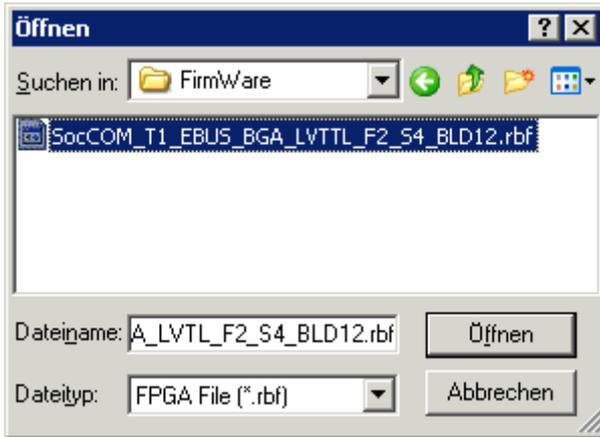
- Wählen Sie im TwinCAT System-Manager die Klemme an, deren FPGA-Firmware Sie aktualisieren möchten (im Beispiel: Klemme 5: EL5001) und klicken Sie auf dem Karteireiter *EtherCAT* auf die Schaltfläche *Weitere Einstellungen*:



- Im folgenden Dialog *Advanced Settings* klicken Sie im Menüpunkt *ESC-Zugriff/E<sup>2</sup>PROM/FPGA* auf die Schaltfläche *Schreibe FPGA*:



- Wählen Sie die Datei (\*.rbf) mit der neuen FPGA-Firmware aus und übertragen Sie diese zum EtherCAT-Gerät:



- Abwarten bis zum Ende des Downloads
- Slave kurz stromlos schalten (nicht unter Spannung ziehen!). Um die neue FPGA-Firmware zu aktivieren ist ein Neustart (Aus- und Wiedereinschalten der Spannungsversorgung) des EtherCAT-Geräts erforderlich
- Kontrolle des neuen FPGA-Standes

**HINWEIS**

**Beschädigung des Gerätes möglich!**

Das Herunterladen der Firmware auf ein EtherCAT-Gerät dürfen Sie auf keinen Fall unterbrechen! Wenn Sie diesen Vorgang abbrechen, dabei die Versorgungsspannung ausschalten oder die Ethernet-Verbindung unterbrechen, kann das EtherCAT-Gerät nur vom Hersteller wieder in Betrieb genommen werden!

**8.3.5 Gleichzeitiges Update mehrerer EtherCAT-Geräte**

Die Firmware von mehreren Geräten kann gleichzeitig aktualisiert werden, ebenso wie die ESI-Beschreibung. Voraussetzung hierfür ist, dass für diese Geräte die gleiche Firmware-Datei/ESI gilt.

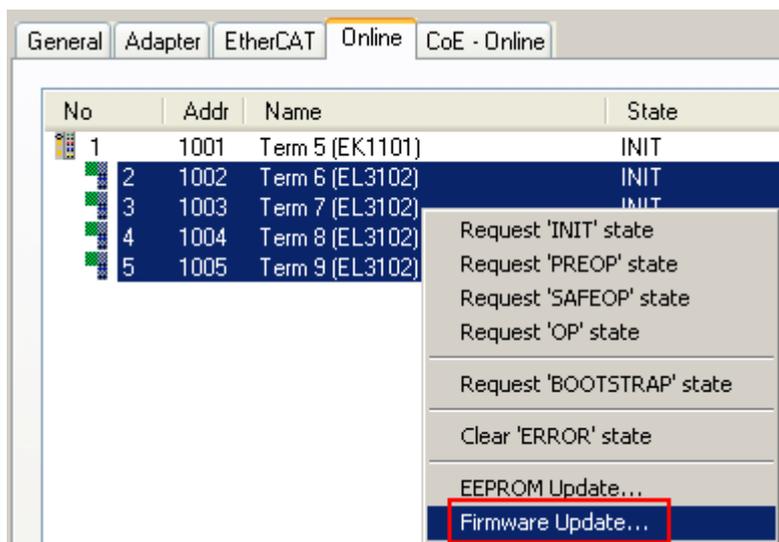


Abb. 180: Mehrfache Selektion und FW-Update

Wählen Sie dazu die betreffenden Slaves aus und führen Sie das Firmware-Update im BOOTSTRAP Modus wie o. a. aus.

## 8.4 Wiederherstellen des Auslieferungszustandes

Um den Auslieferungszustand der Backup-Objekte bei den ELxxx-Klemmen wiederherzustellen, kann im TwinCAT System Manger (Config-Modus) das CoE-Objekt *Restore default parameters*, Subindex 001 angewählt werden (s. Abb. *Auswahl des PDO, Restore default parameters*)

The screenshot shows the 'CoE - Online' tab in the TwinCAT System Manager. A table lists various PDO objects. The object '1011:01 SubIndex 001' is selected, and a red arrow points to it. Below the table is a detailed view of the selected object's properties.

Index	Name	Flags	Wert
1000	Device type	RO	0x00001389 (5001)
1008	Device name	RO	EL5101
1009	Hardware version	RO	09
100A	Software version	RO	10
1011:0	Restore default parameters	RO	> 1 <
1011:01	SubIndex 001	RW	0x00000000 (0)
1018:0	Identity	RO	> 4 <

Name	Typ	Größe	>Adre...	Ein/Aus	User ID	Verknüpft mit
Status	USINT	1.0	26.0	Eingang	0	
Value	UINT	2.0	27.0	Eingang	0	
Latch	UINT	2.0	29.0	Eingang	0	
WcState	BOOL	0.1	1522.0	Eingang	0	
State	UINT	2.0	1550.0	Eingang	0	
AdsAddr	AMSADDRESS	8.0	1552.0	Eingang	0	
netId	ARRAY Tn....	6.0	1552.0	Finnann	0	

Abb. 181: Auswahl des PDO *Restore default parameters*

Durch Doppelklick auf *SubIndex 001* gelangen Sie in den Set Value -Dialog. Tragen Sie im Feld *Dec* den Wert **1684107116** oder alternativ im Feld *Hex* den Wert **0x64616F6C** ein und bestätigen Sie mit OK (Abb. *Eingabe des Restore-Wertes im Set Value Dialog*).

Alle Backup-Objekte werden so in den Auslieferungszustand zurückgesetzt.

The 'Set Value Dialog' window is shown with the following fields and values:

- Dec: 1684107116
- Hex: 0x64616F6C
- Float: 1684107116
- Bool: 0
- Binär: 6C 6F 61 64
- Bitgröße: 32 (selected)

The 'OK' button is highlighted with a red arrow.

Abb. 182: Eingabe des Restore-Wertes im Set Value Dialog

### ● Alternativer Restore-Wert

**I** Bei einigen Klemmen älterer Bauart lassen sich die Backup-Objekte mit einem alternativen Restore-Wert umstellen: Dezimalwert: 1819238756, Hexadezimalwert: 0x6C6F6164. Eine falsche Eingabe des Restore-Wertes zeigt keine Wirkung!

## 8.5 Support und Service

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

### 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.

# Abbildungsverzeichnis

Abb. 1	EL5021 EL-Klemme, Standard IP20-IO-Gerät mit Seriennummer/ Chargennummer und Revisionskennzeichnung (seit 2014/01)	11
Abb. 2	EK1100 EtherCAT Koppler, Standard IP20-IO-Gerät mit Seriennummer/ Chargennummer	12
Abb. 3	CU2016 Switch mit Seriennummer/ Chargennummer	12
Abb. 4	EL3202-0020 mit Seriennummer/ Chargennummer 26131006 und eindeutiger ID-Nummer 204418	12
Abb. 5	EP1258-00001 IP67 EtherCAT Box mit Chargennummer/ DateCode 22090101 und eindeutiger Seriennummer 158102	13
Abb. 6	EP1908-0002 IP67 EtherCAT Safety Box mit Chargennummer/ DateCode 071201FF und eindeutiger Seriennummer 00346070	13
Abb. 7	EL2904 IP20 Safety Klemme mit Chargennummer/ DateCode 50110302 und eindeutiger Seriennummer 00331701	13
Abb. 8	ELM3604-0002 Klemme mit eindeutiger ID-Nummer (QR Code) 100001051 und Seriennummer/ Chargennummer 44160201	13
Abb. 9	BIC als Data Matrix Code (DMC, Code-Schema ECC200)	14
Abb. 10	Pegel Schnittstellen RS232, RS485/422	17
Abb. 11	Terminierung RS422	18
Abb. 12	Terminierung RS485	18
Abb. 13	Schirmanschluss in EL60xx	19
Abb. 14	System Manager Stromberechnung	26
Abb. 15	Karteireiter EtherCAT -> Erweiterte Einstellungen -> Verhalten --> Watchdog	27
Abb. 16	Zustände der EtherCAT State Machine	29
Abb. 17	Karteireiter „CoE-Online“	31
Abb. 18	StartUp-Liste im TwinCAT System Manager	32
Abb. 19	Offline-Verzeichnis	33
Abb. 20	Online-Verzeichnis	34
Abb. 21	Federkontakte der Beckhoff I/O-Komponenten	36
Abb. 22	Montage auf Tragschiene	37
Abb. 23	Demontage von Tragschiene	38
Abb. 24	Linksseitiger Powerkontakt	39
Abb. 25	Standardverdrahtung	40
Abb. 26	Steckbare Verdrahtung	40
Abb. 27	High-Density-Klemmen	40
Abb. 28	Anschluss einer Leitung an eine Klemmstelle	42
Abb. 29	Korrekte Positionierung	44
Abb. 30	Inkorrekte Positionierung	44
Abb. 31	EL6001, EL6021 - LEDs und Anschlussbelegung	45
Abb. 32	EL6001, EL6021 - LEDs und Anschlussbelegung	46
Abb. 33	Anschluss bei RS422 Übertragung	47
Abb. 34	Anschluss bei RS485 Übertragung	47
Abb. 35	EL6002, EL6022 - LEDs	50
Abb. 36	Anschluss bei RS422 Übertragung	52
Abb. 37	Anschluss bei RS485 Übertragung	52
Abb. 38	Korrekte Positionierung	53
Abb. 39	Inkorrekte Positionierung	53
Abb. 40	Empfohlene Abstände bei Standard-Einbaulage	55

Abb. 41	Weitere Einbaulagen .....	56
Abb. 42	Bezug von der Anwender Seite (Inbetriebnahme) zur Installation.....	61
Abb. 43	Aufbau der Steuerung mit Embedded-PC, Eingabe (EL1004) und Ausgabe (EL2008) .....	62
Abb. 44	Initiale Benutzeroberfläche TwinCAT 2 .....	63
Abb. 45	Wähle Zielsystem .....	64
Abb. 46	PLC für den Zugriff des TwinCAT System Managers festlegen: Auswahl des Zielsystems.....	64
Abb. 47	Auswahl „Gerät Suchen.“ .....	65
Abb. 48	Automatische Erkennung von E/A Geräten: Auswahl der einzubindenden Geräte.....	65
Abb. 49	Abbildung der Konfiguration im TwinCAT 2 System Manager.....	66
Abb. 50	Einlesen von einzelnen an einem Gerät befindlichen Klemmen.....	66
Abb. 51	TwinCAT PLC Control nach dem Start .....	67
Abb. 52	Beispielprogramm mit Variablen nach einem Kompilervorgang (ohne Variablenanbindung).....	68
Abb. 53	Hinzufügen des Projektes des TwinCAT PLC Control.....	68
Abb. 54	Eingebundenes PLC Projekt in der SPS- Konfiguration des System Managers .....	69
Abb. 55	Erstellen der Verknüpfungen PLC-Variablen zu Prozessobjekten .....	69
Abb. 56	Auswahl des PDO vom Typ BOOL.....	70
Abb. 57	Auswahl von mehreren PDO gleichzeitig: Aktivierung von „Kontinuierlich“ und „Alle Typen“ .....	70
Abb. 58	Anwendung von „Goto Link Variable“ am Beispiel von „MAIN.bEL1004_Ch4“ .....	71
Abb. 59	Auswahl des Zielsystems (remote).....	72
Abb. 60	PLC Control Logged-in, bereit zum Programmstart.....	72
Abb. 61	Initiale Benutzeroberfläche TwinCAT 3 .....	73
Abb. 62	Neues TwinCAT 3 Projekt erstellen.....	74
Abb. 63	Neues TwinCAT 3 Projekt im Projektmappen-Explorer .....	74
Abb. 64	Auswahldialog: Wähle Zielsystem .....	75
Abb. 65	PLC für den Zugriff des TwinCAT System Managers festlegen: Auswahl des Zielsystems.....	75
Abb. 66	Auswahl „Scan“ .....	76
Abb. 67	Automatische Erkennung von E/A Geräten: Auswahl der einzubindenden Geräte.....	76
Abb. 68	Abbildung der Konfiguration in VS Shell der TwinCAT 3 Umgebung .....	77
Abb. 69	Einlesen von einzelnen an einem Gerät befindlichen Klemmen.....	77
Abb. 70	Einfügen der Programmierumgebung in „SPS“ .....	78
Abb. 71	Festlegen des Namens bzw. Verzeichnisses für die PLC Programmierumgebung .....	79
Abb. 72	Initiales Programm „Main“ des Standard PLC Projektes .....	79
Abb. 73	Beispielprogramm mit Variablen nach einem Kompilervorgang (ohne Variablenanbindung).....	80
Abb. 74	Kompilierung des Programms starten .....	80
Abb. 75	Erstellen der Verknüpfungen PLC-Variablen zu Prozessobjekten .....	81
Abb. 76	Auswahl des PDO vom Typ BOOL.....	82
Abb. 77	Auswahl von mehreren PDO gleichzeitig: Aktivierung von „Kontinuierlich“ und „Alle Typen“ .....	82
Abb. 78	Anwendung von "Goto Link Variable" am Beispiel von „MAIN.bEL1004_Ch4“ .....	83
Abb. 79	Erzeugen eines SPS Datentyps .....	84
Abb. 80	Instance_of_struct .....	84
Abb. 81	Verknüpfung der Struktur.....	84
Abb. 82	Lesen einer Variable aus der Struktur der Prozessdaten .....	85
Abb. 83	TwinCAT 3 Entwicklungsumgebung (VS Shell): Logged-in, nach erfolgten Programmstart .....	86
Abb. 84	Aufruf im System Manager (TwinCAT 2).....	87
Abb. 85	Aufruf in VS Shell (TwinCAT 3) .....	87
Abb. 86	Übersicht Netzwerkschnittstellen.....	88

Abb. 87	Eigenschaft von EtherCAT Gerät (TwinCAT 2): Klick auf „Kompatible Geräte...“ von „Adapter“	88
Abb. 88	Windows-Eigenschaften der Netzwerkschnittstelle	89
Abb. 89	Beispielhafte korrekte Treiber-Einstellung des Ethernet Ports	89
Abb. 90	Fehlerhafte Treiber-Einstellungen des Ethernet Ports	90
Abb. 91	TCP/IP-Einstellung des Ethernet Ports	91
Abb. 92	Gerätebezeichnung: Struktur	92
Abb. 93	Hinweisfenster OnlineDescription (TwinCAT 2)	93
Abb. 94	Hinweisfenster OnlineDescription (TwinCAT 3)	93
Abb. 95	Vom System Manager angelegt OnlineDescription.xml	94
Abb. 96	Kennzeichnung einer online erfassten ESI am Beispiel EL2521	94
Abb. 97	Hinweisfenster fehlerhafte ESI-Datei (links: TwinCAT 2; rechts: TwinCAT 3)	95
Abb. 98	Anwendung des ESI Updater (>=TwinCAT 2.11)	96
Abb. 99	Anwendung des ESI Updater (TwinCAT 3)	96
Abb. 100	Anfügen eines EtherCAT Device: links TwinCAT 2; rechts TwinCAT 3	97
Abb. 101	Auswahl EtherCAT Anschluss (TwinCAT 2.11, TwinCAT 3)	97
Abb. 102	Auswahl Ethernet Port	98
Abb. 103	Eigenschaften EtherCAT Gerät (TwinCAT 2)	98
Abb. 104	Anfügen von EtherCAT Geräten (links: TwinCAT 2; rechts: TwinCAT 3)	99
Abb. 105	Auswahldialog neues EtherCAT Gerät	99
Abb. 106	Anzeige Geräte-Revision	100
Abb. 107	Anzeige vorhergehender Revisionen	100
Abb. 108	Name/Revision Klemme	101
Abb. 109	EtherCAT Klemme im TwinCAT-Baum (links: TwinCAT 2; rechts: TwinCAT 3)	101
Abb. 110	Unterscheidung Lokalsystem/ Zielsystem (links: TwinCAT 2; rechts: TwinCAT 3)	102
Abb. 111	Scan Devices (links: TwinCAT 2; rechts: TwinCAT 3)	102
Abb. 112	Hinweis automatischer GeräteScan (links: TwinCAT 2; rechts: TwinCAT 3)	103
Abb. 113	Erkannte Ethernet-Geräte	103
Abb. 114	Beispiel Default-Zustand	103
Abb. 115	Einbau EtherCAT-Klemme mit Revision -1018	104
Abb. 116	Erkennen EtherCAT-Klemme mit Revision -1019	104
Abb. 117	Scan-Abfrage nach dem automatischen Anlegen eines EtherCAT Gerätes (links: TwinCAT 2; rechts TwinCAT 3)	105
Abb. 118	Manuelles Auslösen des Teilnehmer-Scans auf festgelegtem EtherCAT Device (links: TwinCAT 2; rechts TwinCAT 3)	105
Abb. 119	Scanfortschritt am Beispiel von TwinCAT 2	105
Abb. 120	Abfrage Config/FreeRun (links: TwinCAT 2; rechts TwinCAT 3)	105
Abb. 121	Anzeige des Wechsels zwischen „Free Run“ und „Config Mode“ unten rechts in der Statusleiste	106
Abb. 122	TwinCAT kann auch durch einen Button in diesen Zustand versetzt werden (links: TwinCAT 2; rechts TwinCAT 3)	106
Abb. 123	Beispielhafte Online-Anzeige	106
Abb. 124	Fehlerhafte Erkennung	107
Abb. 125	Identische Konfiguration (links: TwinCAT 2; rechts TwinCAT 3)	107
Abb. 126	Korrekturdialog	108
Abb. 127	Name/Revision Klemme	109
Abb. 128	Korrekturdialog mit Änderungen	109
Abb. 129	Dialog „Change to Compatible Type...“ (links: TwinCAT 2; rechts TwinCAT 3)	110

Abb. 130 TwinCAT 2 Dialog Change to Alternative Type ..... 110

Abb. 131 „Baumzweig“ Element als Klemme EL3751 ..... 110

Abb. 132 Karteireiter „Allgemein“ ..... 111

Abb. 133 Karteireiter „EtherCAT“ ..... 111

Abb. 134 Karteireiter „Prozessdaten“ ..... 112

Abb. 135 Konfigurieren der Prozessdaten ..... 113

Abb. 136 Karteireiter „Startup“ ..... 114

Abb. 137 Karteireiter „CoE - Online“ ..... 115

Abb. 138 Dialog „Advanced settings“ ..... 116

Abb. 139 Karteireiter „Online“ ..... 117

Abb. 140 Karteireiter „DC“ (Distributed Clocks) ..... 118

Abb. 141 Auswahl an Diagnoseinformationen eines EtherCAT Slave ..... 120

Abb. 142 Grundlegende EtherCAT Slave Diagnose in der PLC ..... 121

Abb. 143 EL3102, CoE-Verzeichnis..... 123

Abb. 144 Beispiel Inbetriebnahmehilfe für eine EL3204 ..... 124

Abb. 145 Default Verhalten System Manager ..... 125

Abb. 146 Default Zielzustand im Slave ..... 126

Abb. 147 PLC-Bausteine..... 126

Abb. 148 Unzulässige Überschreitung E-Bus Strom ..... 127

Abb. 149 Warnmeldung E-Bus-Überschreitung ..... 127

Abb. 150 StartUp-Eintrag mit Übergang S -> O ..... 129

Abb. 151 Reiter "CoE - Online, Klemmen EL60x2" ..... 132

Abb. 152 Die jeweils höhere Ebene gibt (falls vorhanden oder aktiviert) der jeweils darunter liegenden Eben die Kommunikationseigenschaften vor ..... 135

Abb. 153 Prüfen der von der COM-Anwendung gewünschten Einstellungen im CoE..... 136

Abb. 154 Default StartUp-Einträge einer EL6002 (Beispiel) - nur die Default-Einträge (hier 0x1C12 und 0x1C13) werden benötigt..... 136

Abb. 155 VirtualComDriver Settings (Beispiel) ..... 136

Abb. 156 RS232-LIN Sub-D Stecker angeschlossen an die EL6001 Klemme ..... 140

Abb. 157 LIN-Frame Beispiel: Anfrage vom Master an Knoten mit ID 0x07 ..... 141

Abb. 158 LIN-Frame Beispiel: ID0x07 mit Daten 0xEA,0xBD,0x08,0xB7 + Checksumme 0x97 ..... 141

Abb. 159 Links: Anfrage auf dem LIN-Bus mit PID 0x47, rechts: LIN-Frame mit gleicher PID und Daten inkl. Checksumme ..... 141

Abb. 160 Suchen des Ethernet-Adapters..... 143

Abb. 161 Auswahl und Bestätigung des Ethernet-Adapters ..... 143

Abb. 162 Aktivierung der Konfiguration..... 143

Abb. 163 Konfigurationsaktivierung bestätigen..... 144

Abb. 164 Variablenzuordnung erzeugen..... 144

Abb. 165 Neustart TwinCAT im RUN-Modus..... 144

Abb. 166 Projekt übersetzen..... 144

Abb. 167 Programmstart bestätigen ..... 145

Abb. 168 Empfangener Barcode ..... 145

Abb. 169 Gerätekennung aus Name EL3204-0000 und Revision -0016 ..... 197

Abb. 170 Rechtsklick auf das EtherCAT Gerät bewirkt das Scannen des unterlagerten Feldes ..... 198

Abb. 171 Konfiguration identisch ..... 198

Abb. 172 Änderungsdialog..... 199

Abb. 173 EEPROM Update..... 199

Abb. 174 Auswahl des neuen ESI.....	200
Abb. 175 Anzeige FW-Stand EL3204 .....	201
Abb. 176 Firmware Update .....	202
Abb. 177 Versionsbestimmung FPGA-Firmware .....	204
Abb. 178 Kontextmenu Eigenschaften (Properties).....	204
Abb. 179 Dialog Advanced settings .....	205
Abb. 180 Mehrfache Selektion und FW-Update .....	207
Abb. 181 Auswahl des PDO Restore default parameters .....	208
Abb. 182 Eingabe des Restore-Wertes im Set Value Dialog.....	208



Mehr Informationen:

[www.beckhoff.de/german/ethercat/sonder\\_el6xxx.htm](http://www.beckhoff.de/german/ethercat/sonder_el6xxx.htm)

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)

