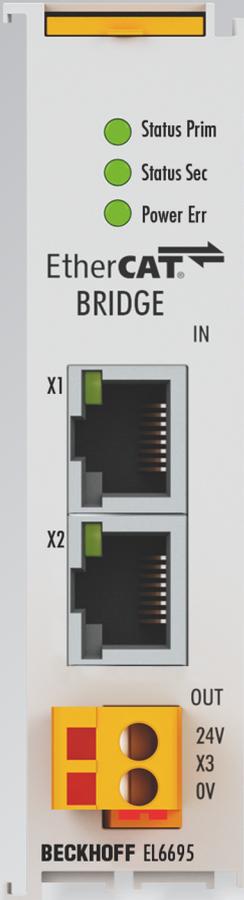


Dokumentation | DE

EL6695

EtherCAT-Bridge-Klemme



Inhaltsverzeichnis

1	Vorwort.....	7
1.1	Hinweise zur Dokumentation	7
1.2	Sicherheitshinweise	8
1.3	Ausgabestände der Dokumentation	9
1.4	Wegweiser durch die Dokumentation	11
1.5	Versionsidentifikation von EtherCAT-Geräten	12
1.5.1	Allgemeine Hinweise zur Kennzeichnung	12
1.5.2	Versionsidentifikation von EL-Klemmen.....	13
1.5.3	Beckhoff Identification Code (BIC).....	14
1.5.4	Elektronischer Zugriff auf den BIC (eBIC).....	15
2	Produktbeschreibung	18
2.1	Einführung.....	18
2.2	Technische Daten	19
2.2.1	Vergleich Beckhoff EtherCAT Datenaustauschgeräte	20
2.3	Betriebsbedingungen zur Installation	21
3	Grundlagen der Kommunikation	22
3.1	EtherCAT-Grundlagen	22
3.2	EtherCAT-Verkabelung - Drahtgebunden	22
3.3	Allgemeine Hinweise zur Watchdog-Einstellung.....	24
3.4	EtherCAT State Machine	25
3.5	CoE-Interface	27
3.6	Distributed Clock	32
4	Montage und Verdrahtung.....	33
4.1	Hinweise zum ESD-Schutz	33
4.2	UL-Hinweise	34
4.3	Hinweis zur Spannungsversorgung	35
4.4	Montage und Demontage - Frontriegelung oben	36
4.5	Einbaulagen	38
4.6	Positionierung von passiven Klemmen	40
4.7	Hinweis zur Schirmung	41
4.8	LEDs und Anschlussbelegung	42
4.9	Diagnose	43
4.10	Entsorgung	44
5	Inbetriebnahme	45
5.1	TwinCAT Quickstart	45
5.1.1	TwinCAT 2	48
5.1.2	TwinCAT 3	58
5.2	TwinCAT Entwicklungsumgebung	72
5.2.1	Installation der TwinCAT Realtime-Treiber	72
5.2.2	Hinweise zur ESI-Gerätebeschreibung	78
5.2.3	OFFLINE Konfigurationserstellung	82
5.2.4	ONLINE Konfigurationserstellung	87
5.2.5	EtherCAT-Teilnehmerkonfiguration.....	95

5.3	Allgemeine Inbetriebnahmehinweise für einen EtherCAT-Slave	105
5.4	CoE Objektübersicht	113
5.4.1	Standard CoE Objekte	113
5.4.2	Klemmenspezifische CoE-Objekte.....	114
5.4.3	Profilspezifische CoE Objekte.....	121
6	Funktion und Betriebsarten	124
6.1	Grundlagen zur Funktion.....	125
6.1.1	Zugriff auf EL6695 über TwinCAT.....	126
6.2	Kompatibilität zur EL6692	128
6.3	State-Maschine EL6695.....	129
6.4	Zyklische Prozessdaten PDO	130
6.4.1	Flusskontrolle	130
6.4.2	Symmetrisches PDO-Mapping.....	134
6.4.3	Selektives PDO-Mapping.....	140
6.4.4	FSoE-Übertragung.....	150
6.5	Lokale Mailbox Protokolle	151
6.5.1	CoE - Can over EtherCAT.....	151
6.6	Übertragungsfähige Mailbox Protokolle	152
6.6.1	AoE Anwendung beim CoE Zugriff	152
6.6.2	EoE - Ethernet over EtherCAT	154
6.6.3	AoE - ADS over EtherCAT	157
6.6.4	FoE – Filetransfer over EtherCAT.....	159
6.6.5	VoE - Vendor specific protocol over EtherCAT	171
6.6.6	SoE - Servo Drive Profile over EtherCAT	171
6.7	Distributed Clocks	172
6.8	Online Scan.....	175
6.9	CompatibilityMode zur EL6692	176
6.9.1	Modulbetrieb mit PDO Mapping und Assignment	176
6.10	Performance Betriebsarten EL6695.....	177
6.10.1	Grundlagen zum PDO Mapping	177
6.10.2	Ohne Init Commands	177
6.10.3	Object description download	177
6.10.4	Geräteemulation.....	177
6.11	Applikationsspezifische Variablendefinition	179
7	Anhang	180
7.1	EtherCAT AL Status Codes	180
7.2	Firmware Kompatibilität.....	181
7.3	Firmware Update EL/ES/ELM/EM/EP/EPP/ERPxxxx	182
7.3.1	Gerätebeschreibung ESI-File/XML	183
7.3.2	Erläuterungen zur Firmware.....	186
7.3.3	Update Controller-Firmware *.efw.....	187
7.3.4	Gleichzeitiges Update mehrerer EtherCAT-Geräte.....	189
7.4	Ergänzung zu Firmware (FW)/ FPGA Update.....	190
7.5	Wiederherstellen des Auslieferungszustandes	193
7.5.1	Allgemeiner Geräte-Reset.....	193

7.5.2 Wiederherstellen des Auslieferungszustandes 193
7.6 Support und Service 195

1 Vorwort

1.1 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, stets die aktuell gültige Dokumentation zu verwenden.

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

Disclaimer

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

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

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

Marken

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

Patente

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



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!

Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

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

Qualifikation des Personals

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

Signalwörter

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

Warnungen vor Personenschäden

GEFAHR

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

WARNUNG

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

VORSICHT

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

Warnung vor Umwelt- oder Sachschäden

HINWEIS

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

Information zum Umgang mit dem Produkt



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

1.3 Ausgabestände der Dokumentation

Version	Kommentar
1.8.0	<ul style="list-style-type: none"> • Update Kapitel „Montage und Verdrahtung“ • Update Kapitel „LEDs und Anschlussbelegung“ • Update Struktur
1.7.0	<ul style="list-style-type: none"> • Update Kapitel „Firmware Update EL/ES/ELM/EM/EPxxxx“ • Update Kapitel „Ergänzung zu Firmware (FW)/ FPGA Update“ • Update Kapitel „Firmware Kompatibilität“ • Update Kapitel „Wiederherstellen des Auslieferungszustandes“ • Kapitel „Wegweiser durch die Dokumentation“ ergänzt (Vorwort) • Kapitel „Empfohlene Trageschiene“ entfernt
1.6.0	<ul style="list-style-type: none"> • Update Kapitel „Anschluss“ • Strukturupdate
1.5.0	<ul style="list-style-type: none"> • Update Kapitel „Technische Daten“ • Update Kapitel „Montage und Verdrahtung“ • Update Kapitel „Diagnose“ • Strukturupdate • Revisionsstand aktualisiert
1.4.6	<ul style="list-style-type: none"> • „CoE Objektübersicht“ im „Anhang“ aktualisiert
1.4.5	<ul style="list-style-type: none"> • Ergänzung im Kapitel „Symmetrisches PDO-Mapping“ • Strukturupdate
1.4.4	<ul style="list-style-type: none"> • Update Kapitel „Symmetrisches PDO-Mapping“ • Strukturupdate
1.4.3	<ul style="list-style-type: none"> • Update Kapitel „Distributed Clocks“ • Strukturupdate
1.4.2	<ul style="list-style-type: none"> • Update Kapitel „Symmetrisches PDO-Mapping“ • Strukturupdate
1.4.1	<ul style="list-style-type: none"> • Strukturupdate
1.4.0	<ul style="list-style-type: none"> • Beispiel zum FoE – Datendurchsatz ergänzt • Diverse Ergänzungen/ Korrekturen • Strukturupdate • Revisionsstand aktualisiert
1.3.0	<ul style="list-style-type: none"> • Update Kapitel „Funktion und Betriebsarten“ (Betriebsart FoE ergänzt)
1.2.0	<ul style="list-style-type: none"> • Update Kapitel „Einleitung“ • Update Kapitel „Funktion und Betriebsarten“ • Kapitel „Anschluss“ ergänzt • Update Kapitel „Diagnose LEDs“
1.1.0	<ul style="list-style-type: none"> • Update Kapitel „Funktion und Betriebsarten“ • Update Kapitel „Technische Daten“ • Strukturupdate • Revisionsstand aktualisiert
1.0.0	<ul style="list-style-type: none"> • Erste Veröffentlichung
0.9.8	<ul style="list-style-type: none"> • Kapitel „Selektives PDO Mapping“ mit „Konvertierung in einen globalen Datentyp“ ergänzt • (Vorläufige Version – Änderungen vorbehalten)
0.9.7	<ul style="list-style-type: none"> • Korrektur im Kapitel „AoE Anwendung beim CoE Zugriff“ durchgeführt
0.9.6	<ul style="list-style-type: none"> • Erweiterung des Kapitels „Funktion und Betriebsarten“

Version	Kommentar
0.9.5	<ul style="list-style-type: none">• Kapitel „AoE Anwendung beim CoE Zugriff“ in das Kapitel „Übertragungsfähige Maibox Protokolle“ eingesetzt
0.9	<ul style="list-style-type: none">• Vorläufige Dokumentation zur Erstserie
0.1	<ul style="list-style-type: none">• Erste Erstellung

1.4 Wegweiser durch die Dokumentation

HINWEIS



Weitere Bestandteile der Dokumentation

Diese Dokumentation beschreibt gerätespezifische Inhalte. Sie ist Bestandteil des modular aufgebauten Dokumentationskonzepts für Beckhoff I/O-Komponenten. Für den Einsatz und sicheren Betrieb des in dieser Dokumentation beschriebenen Gerätes / der in dieser Dokumentation beschriebenen Geräte werden zusätzliche, produktübergreifende Beschreibungen benötigt, die der folgenden Tabelle zu entnehmen sind.

Titel	Beschreibung
EtherCAT System-Dokumentation (PDF)	<ul style="list-style-type: none"> • Systemübersicht • EtherCAT-Grundlagen • Kabel-Redundanz • Hot Connect • Konfiguration von EtherCAT-Geräten
Infrastruktur für EtherCAT/Ethernet (PDF)	Technische Empfehlungen und Hinweise zur Auslegung, Ausfertigung und Prüfung
Software-Deklarationen I/O (PDF)	Open-Source-Software-Deklarationen für Beckhoff-I/O-Komponenten

Die Dokumentationen können auf der Beckhoff-Homepage (www.beckhoff.com) eingesehen und heruntergeladen werden über:

- den Bereich „Dokumentation und Downloads“ der jeweiligen Produktseite,
- den [Downloadfinder](#),
- das [Beckhoff Information System](#).

Sollten Sie Vorschläge oder Anregungen zu unserer Dokumentation haben, schicken Sie uns bitte unter Angabe von Dokumentationstitel und Versionsnummer eine E-Mail an: dokumentation@beckhoff.com

1.5 Versionsidentifikation von EtherCAT-Geräten

1.5.1 Allgemeine Hinweise zur Kennzeichnung

Bezeichnung

Ein Beckhoff EtherCAT-Gerät hat eine 14-stellige technische Bezeichnung, die sich zusammen setzt 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 Januar 2014 außen auf den IP20-Klemmen aufgebracht, siehe Abb. „EL2872 mit Revision 0022 und Seriennummer 01200815“.
- Typ, Version und Revision werden als dezimale Zahlen gelesen, auch wenn sie technisch hexadezimal gespeichert werden.

1.5.2 Versionsidentifikation von EL-Klemmen

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

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

KK - Produktionswoche (Kalenderwoche)

YY - Produktionsjahr

FF - Firmware-Stand

HH - Hardware-Stand

Beispiel mit Seriennummer 12 06 3A 02:

12 - Produktionswoche 12

06 - Produktionsjahr 2006

3A - Firmware-Stand 3A

02 - Hardware-Stand 02



Abb. 1: EL2872 mit Revision 0022 und Seriennummer 01200815

1.5.3 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.

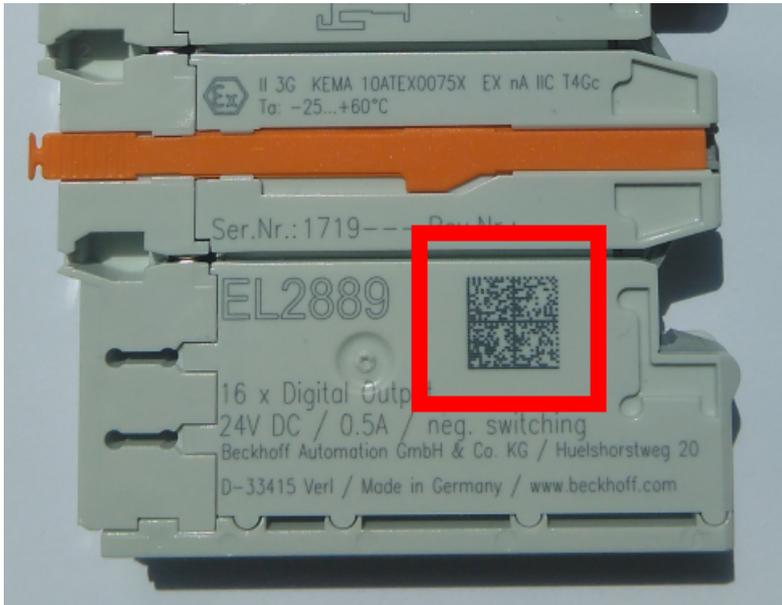


Abb. 2: BIC als Data Matrix Code (DMC, Code-Schema ECC200)

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.

Folgende Informationen sind möglich, die Positionen 1 bis 4 sind immer vorhanden, die weiteren je nach Produktfamilienbedarf:

Pos-Nr.	Art der Information	Erklärung	Datenidentifikator	Anzahl Stellen inkl. Datenidentifikator	Beispiel
1	Beckhoff-Artikelnummer	Beckhoff - Artikelnummer	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	Eindeutige Seriennummer, Hinweis s. u.	SBTN	12	SBTNk4p562d7
3	Artikelbezeichnung	Beckhoff Artikelbezeichnung, z. B. EL1008	1K	32	1KEL1809
4	Menge	Menge in Verpackungseinheit, z. B. 1, 10...	Q	6	Q1
5	Chargennummer	Optional: Produktionsjahr und -woche	2P	14	2P401503180016

Pos-Nr.	Art der Information	Erklärung	Datenidentifikator	Anzahl Stellen inkl. Datenidentifikator	Beispiel
6	ID-/Seriennummer	Optional: vorheriges Seriennummer-System, z. B. bei Safety-Produkten oder kalibrierten Klemmen	51S	12	51S678294
7	Variante	Optional: Produktvarianten-Nummer auf Basis von Standardprodukten	30P	32	30PF971, 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 dem o.a. Beispielwert in Position 6. Die Datenidentifikatoren sind in Fettschrift hervorgehoben:

1P072222**SB**TNk4p562d71**KEL**1809 Q1 **51S**678294

Entsprechend als DMC:



Abb. 3: Beispiel-DMC 1P072222**SB**TNk4p562d71**KEL**1809 Q1 **51S**678294

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.

HINWEIS
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 Dokumentation können keine Ansprüche auf Änderung geltend gemacht werden.

1.5.4 Elektronischer Zugriff auf den BIC (eBIC)

Elektronischer BIC (eBIC)

Der Beckhoff Identification Code (BIC) wird auf Beckhoff-Produkten außen sichtbar aufgebracht. Er soll, wo möglich, auch elektronisch auslesbar sein.

Für die elektronische Auslesung ist die Schnittstelle entscheidend, über die das Produkt angesprochen werden kann.

K-Bus Geräte (IP20, IP67)

Für diese Geräte ist derzeit keine elektronische Speicherung und Auslesung geplant.

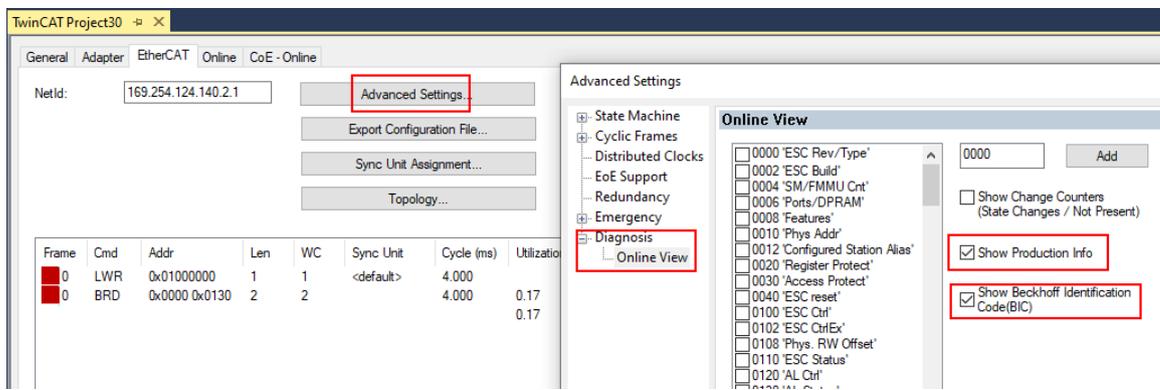
EtherCAT-Geräte (IP20, IP67)

Alle Beckhoff EtherCAT-Geräte haben ein sogenanntes ESI-EEPROM, das die EtherCAT-Identität mit der Revision beinhaltet. Darin wird die EtherCAT-Slave-Information gespeichert, umgangssprachlich auch als ESI/XML-Konfigurationsdatei für den EtherCAT-Master bekannt. Zu den Zusammenhängen siehe die entsprechenden Kapitel im EtherCAT-Systemhandbuch ([Link](#)).

In das ESI-EEPROM wird durch Beckhoff auch die eBIC geschrieben. Die Einführung des eBIC in die Beckhoff-IO-Produktion (Klemmen, Box-Module) erfolgt ab 2020; Stand 2023 ist die Umsetzung weitgehend abgeschlossen.

Anwenderseitig ist die eBIC (wenn vorhanden) wie folgt elektronisch zugänglich:

- Bei allen EtherCAT-Geräten kann der EtherCAT-Master (TwinCAT) den eBIC aus dem ESI-EEPROM auslesen:
 - Ab TwinCAT 3.1 Build 4024.11 kann der eBIC im Online-View angezeigt werden.
 - Dazu unter EtherCAT → Erweiterte Einstellungen → Diagnose das Kontrollkästchen „Show Beckhoff Identification Code (BIC)“ aktivieren:



- Die BTN und Inhalte daraus werden dann angezeigt:

No	Addr	Name	State	CRC	Fw	Hw	Production Data	ItemNo	BTN	Description	Quantity	BatchNo	SerialNo
1	1001	Term 1 (EK1100)	OP	0,0	0	0	---						
2	1002	Term 2 (EL1018)	OP	0,0	0	0	2020 KW36 Fr	072222	k4p562d7	EL1809	1		678294
3	1003	Term 3 (EL3204)	OP	0,0	7	6	2012 KW24 Sa						
4	1004	Term 4 (EL2004)	OP	0,0	0	0	---	072223	k4p562d7	EL2004	1		678295
5	1005	Term 5 (EL1008)	OP	0,0	0	0	---						
6	1006	Term 6 (EL2008)	OP	0,0	0	12	2014 KW14 Mo						
7	1007	Term 7 (EK1110)	OP	0	1	8	2012 KW25 Mo						

- Hinweis: ebenso können wie in der Abbildung zu sehen die seit 2012 programmierten Produktionsdaten HW-Stand, FW-Stand und Produktionsdatum per „Show Production Info“ angezeigt werden.
- Zugriff aus der PLC: Ab TwinCAT 3.1. Build 4024.24 stehen in der Tc2_EtherCAT Library ab v3.3.19.0 die Funktionen *FB_EcReadBIC* und *FB_EcReadBTN* zum Einlesen in die PLC bereit.
- Bei EtherCAT-Geräten mit CoE-Verzeichnis kann zusätzlich das Objekt 0x10E2:01 zur Anzeige der eigenen eBIC vorhanden sein, auch hierauf kann die PLC einfach zugreifen:

- Das Gerät muss zum Zugriff in PREOP/SAFEOP/OP sein

Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	Subindex 001	RO	1P158442SBTN0008jckp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bf277e

- Das Objekt 0x10E2 wird in Bestandsprodukten vorrangig im Zuge einer notwendigen Firmware-Überarbeitung eingeführt.
- Ab TwinCAT 3.1. Build 4024.24 stehen in der Tc2_EtherCAT Library ab v3.3.19.0 die Funktionen *FB_EcCoEReadBIC* und *FB_EcCoEReadBTN* zum Einlesen in die PLC zur Verfügung
- Zur Verarbeitung der BIC/BTN Daten in der PLC stehen noch als Hilfsfunktionen ab TwinCAT 3.1 Build 4024.24 in der *Tc2_Uutilities* zur Verfügung
 - *F_SplitBIC*: Die Funktion zerlegt den BIC sBICValue anhand von bekannten Kennungen in seine Bestandteile und liefert die erkannten Teil-Strings in einer Struktur *ST_SplittedBIC* als Rückgabewert
 - *BIC_TO_BTN*: Die Funktion extrahiert vom BIC die BTN und liefert diese als Rückgabewert
- Hinweis: bei elektronischer Weiterverarbeitung ist die BTN als String(8) zu behandeln, der Identifier „SBTN“ ist nicht Teil der BTN.
- Zum technischen Hintergrund:
Die neue BIC Information wird als Category zusätzlich bei der Geräteproduktion ins ESI-EEPROM geschrieben. Die Struktur des ESI-Inhalts ist durch ETG Spezifikationen weitgehend vorgegeben, demzufolge wird der zusätzliche herstellereigene Inhalt mithilfe einer Category nach ETG.2010 abgelegt. Durch die ID 03 ist für alle EtherCAT-Master vorgegeben, dass sie im Updatefall diese Daten nicht überschreiben bzw. nach einem ESI-Update die Daten wiederherstellen sollen. Die Struktur folgt dem Inhalt des BIC, siehe dort. Damit ergibt sich ein Speicherbedarf von ca. 50..200 Byte im EEPROM.
- Sonderfälle
 - Bei einer hierarchischen Anordnung mehrerer ESC (EtherCAT Slave Controller) in einem Gerät trägt lediglich der oberste ESC die eBIC-Information.
 - Sind mehrere ESC in einem Gerät verbaut die nicht hierarchisch angeordnet sind, tragen alle ESC die eBIC-Information gleich.
 - Besteht das Gerät aus mehreren Sub-Geräten mit eigener Identität, aber nur das TopLevel-Gerät ist über EtherCAT zugänglich, steht im CoE-Objekt-Verzeichnis 0x10E2:01 die eBIC dieses ESC, in 0x10E2:nn folgen die eBIC der Sub-Geräte.

PROFIBUS-, PROFINET-, DeviceNet-Geräte usw.

Für diese Geräte ist derzeit keine elektronische Speicherung und Auslesung geplant.

2 Produktbeschreibung

2.1 Einführung



Abb. 4: EL6695

EtherCAT-Bridge-Klemme

Die EtherCAT-Bridge-Klemme EL6695 ermöglicht den Echtzeit-Datenaustausch zwischen EtherCAT-Strängen und unterschiedlichen Mastern. Auch die asynchrone Kommunikation über diverse Protokolle wird unterstützt. Die Synchronisierung der Distributed-Clocks ist in beide Richtungen möglich. Die EL6695 hebt sich von der weiterhin verfügbaren EL6692 durch flexible CoE-Konfiguration, der Möglichkeit einer Geräteemulation und einer deutlichen Steigerung des Datendurchsatzes ab. Eine komfortable Konfigurationsoberfläche im TwinCAT System Manager „Extension“ steht wie bei der EL6692 zur Verfügung. Die Spannungsversorgung der Sekundärseite (RJ 45) erfolgt über einen externen Anschluss, die Primärseite wird über den E-Bus gespeist. Durch die flexible CoE-Konfiguration kann die Bridge-Klemme auch verwendet werden, um ein unterlagertes IPC-System als EtherCAT-Slave einzubinden – der Anwender kann selbst einen Parametersatz im CoE definieren und das Subsystem nach außen so als eigendefinierten EtherCAT Slave darstellen.

2.2 Technische Daten

Technische Daten	EL6695
Ports	Primäre Seite: E-Bus (Klemmenstrang), sekundäre Seite: 2 x 100 MBit/s Ethernet RJ45, In/Out
Funktion	EtherCAT-Distributed-Clocks-Synchronisierung, Datenaustausch synchron/asynchron
Leitungslänge	Sekundärer Port: max. 100 m 100BASE-TX
Hardwarediagnose	Status-LEDs
Spannungsversorgung Stromaufnahme	Primär: über den E-Bus 400 mA typ. Sekundär: 24 V _{DC} (-15%/+20%), 80 mA typ, steckbar Zum Betrieb muss nur eine der beiden Spannungen anliegen, wenn beide anliegen, verwendet das interne Netzteil bevorzugt die 24 V Versorgung
Distributed-Clocks	Ja, TwinCAT ab TC3.1 b4018.4 wird benötigt
Potenzialtrennung	500 V (E-Bus / Sekundärseite)
Zyklische Prozessdaten	max. 3 kByte je Richtung Hinweis: wieviel zykl. PDO unterstützt werden ist v.a. vom verwendeten EcMaster abhängig Mit TC2.11b2248 oder TC3.1 b4018: 255 Variablen, max. Gesamtgröße MTU (~1400 Byte)
Unterstützte asynchrone Protokolle	CoE, EoE, AoE, FoE, (VoE, SoE)
PDO-Transfargeschwindigkeit	Abhängig von der Betriebsart und Datenmenge (typisch: 10...100 µs)
Minimal EtherCAT Zykluszeit	50 µs
Besondere Eigenschaften	<ul style="list-style-type: none"> • In TwinCAT als externe Referenzuhr nutzbar • Synchroner Datenaustausch • Flexible CoE-Definition / CoE-Geräteemulation • Zur direkten beidseitigen DC-Synchronisierung einsetzbar • PDO Mapping (Symmetrisch oder Selektiv) • ADS-Routing (Bearbeiten von ADS-Anfragen zweier Master) • Übertragung bis zu 5 kByte Nutzdaten per FoE
Zulässige Umgebungstemperatur im Betrieb	0...+55 °C
Zulässige Umgebungstemperatur bei Lagerung	-25...+85 °C
Zulässige relative Feuchte	95 %, ohne Betauung
Schwingungs- / 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
Abmessungen	Breite: 24 mm (beim Aneinanderreihen) Höhe: 100 mm Tiefe: 68 mm
Zulassungen / Kennzeichnungen*	CE, <u>cULus</u> [▶ 34], UKCA, EAC

*) Real zutreffende Zulassungen/Kennzeichnungen siehe seitliches Typenschild (Produktbeschriftung).

2.2.1 Vergleich Beckhoff EtherCAT Datenaustauschgeräte

	EL6695	EL6692	FC1100	FC1121	CXnnnn-B110
Bauform	24mm Klemme		PCIe-Steckkarte		Integriert im embedded-PC
PDO Zyklische Prozessdaten	max. 3 kB je Richtung (¹ Mit TC2.11b2248 oder TC3.1 b4018: 255 Variablen, max. Gesamtgröße MTU (ca.1400 Byte)	max. 480 Byte je Richtung	max. 1024 Byte je Richtung		max. 480 Byte je Richtung
PDO Transfargeschwindigkeit (geräteintern, ohne Buszyklus)	Bsp.: 200 Byte in/200 Byte out: 15 µs typ Bsp.: 1400 Byte in/1400 Byte out: 50 µs typ	typ. 1..4 ms Bsp.: 200 Byte in/200 Byte out: 1 ms typ.	Bsp.: 480 Byte in/480 Byte out: ca. 300 µs (²)		Bsp.: 480 Byte in/480 Byte out: ca. 250 µs auf CX5020 (²)
Unterstützte asynchrone Protokolle	CoE, EoE, AoE, FoE, (VoE, SoE)	CoE, AoE, EoE			
Mailbox	128-1498 Byte	128-1024 Byte	64-1024 Byte		
Mailbox-Voreinstellung (default)	1024 Byte	256 Byte	1024 Byte	512 Byte	
Minimal zulässige EtherCAT Zykluszeit (3)	50 µs (SyncMan Interrupt wird benutzt)	Keine Grenze (SyncMan Interrupt wird nicht benutzt)			
DistributedClocks Synchronisierung	ja				
In TwinCAT als externe Referenzuhr nutzbar	ja				
Besondere Eigenschaften	<ul style="list-style-type: none"> • Synchroner Datenaustausch • Flexible CoE-Definition / CoE Geräteemulation • Zur direkten beidseitigen DC-Synchronisierung einsetzbar • PDO Mapping (Symmetrisch oder Selektiv) • ADS Routing (Bearbeiten von ADS Anfragen zweier Master) • Übertragung bis zu 5 kB Nutzdaten per FoE • Unabhängige Stromversorgung Primär/ Sekundär 	<ul style="list-style-type: none"> • Für DC-Synchronisierung indirekt über EC-Master einsetzbar • ADS Routing 			

¹ Die Anzahl unterstützter zyklische PDO ist abhängig vom verwendeten EtherCAT-Master

² Die Transferwerte hängen erheblich von IPC Plattform und Ansteuerung ab

³ Diese Grenze bedeutet eine Betriebsgrenze des Geräts. Faktisch hängt die minimal sinnvolle Zykluszeit auf beiden Seiten u.a. von der Datengröße (und damit PDO Transferzeit) ab - eine sinnvolle Zykluszeit sollte so gewählt werden, dass die Klemme in jedem Zyklus neue Daten abholen/bringen kann.

2.3 Betriebsbedingungen zur Installation

Stellen Sie sicher, dass die EtherCAT-Klemme EL6695 nur bei den spezifizierten Umgebungsbedingungen (siehe Technische Daten) transportiert, gelagert und betrieben wird!

⚠️ WARNUNG

Unzulässige Betriebsbedingungen beachten!

Die EL6695 darf unter folgenden Betriebsbedingungen nicht eingesetzt werden:

- unter dem Einfluss ionisierender Strahlung
- in korrosivem Umfeld
- in einem Umfeld, das zu unzulässiger Verschmutzung der Busklemme führt

Sicherheitshinweise zur Installation

Lesen Sie vor Installation und Inbetriebnahme der EtherCAT-Klemmen auch die Sicherheitshinweise im Vorwort dieser Dokumentation.

Transportvorgaben / Lagerung

Verwenden Sie zum Transport und bei der Lagerung der EtherCAT-Klemmen die Originalverpackung in der die Klemmen geliefert wurden.

Elektrische Installation

Bitte beachten, dass die EL6695 bevorzugt die 24V-Versorgung benutzt, wenn E-Bus und 24V anliegen. Führen Sie die Inbetriebnahme also bevorzugt ohne angeschlossene 24V-Versorgung durch damit die EL6695 vom E-Bus versorgt wird. Sollte die E-Bus-Versorgungsdimensionierung nicht beachtet worden sein kann es sonst bei Ausfall der 24V-Versorgung und damit der Umschaltung auf E-Bus-Versorgung zu Überbelastung der E-Bus-Versorgung im Klemmenstrang kommen.

Number	Box Name	Address	Type	In Size	Out Size	E-Bus (mA)
1	Box 1 (EL6695)	1001	EL6695	2.0		
2	Term 2 (EK1100)	1002	EK1100			
3	Term 3 (EL6695)	1003	EL6695	2.0		1600
4	Term 4 (EL1008)	1004	EL1008	1.0		1510
5	Term 5 (EL1008)	1005	EL1008	1.0		1420

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 m + 90 m + 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 vier 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 gekreuzte Leitungen (Cross-Over) verwendet werden.

● Empfohlene Kabel



- Es wird empfohlen, die entsprechenden Beckhoff-Komponenten zu verwenden, z. B.
- Kabelsätze ZK1090-9191-xxxx bzw.
 - feldkonfektionierbare RJ45 Stecker ZS1090-0005 oder
 - 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 berechnete, theoretische maximale E-Bus-Strom angezeigt. Eine Unterschreitung wird durch einen 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. 5: 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 EtherCAT-Klemmen sind mit einer Sicherungseinrichtung (Watchdog) ausgestattet, die z. B. bei unterbrochenem Prozessdatenverkehr nach einer voreinstellbaren Zeit die Ausgänge (sofern vorhanden) in einen gegebenenfalls vorgebbaren Zustand schaltet, in Abhängigkeit von Gerät und Einstellung z. B. auf FALSE (aus) oder einen Ausgabewert.

Der EtherCAT Slave Controller verfügt dazu über zwei Watchdogs:

- Sync Manager (SM)-Watchdog (default: 100 ms)
- Process-Data (PDI)-Watchdog (default: 100 ms)

Deren Zeiten werden in TwinCAT wie folgt einzeln parametrisiert:

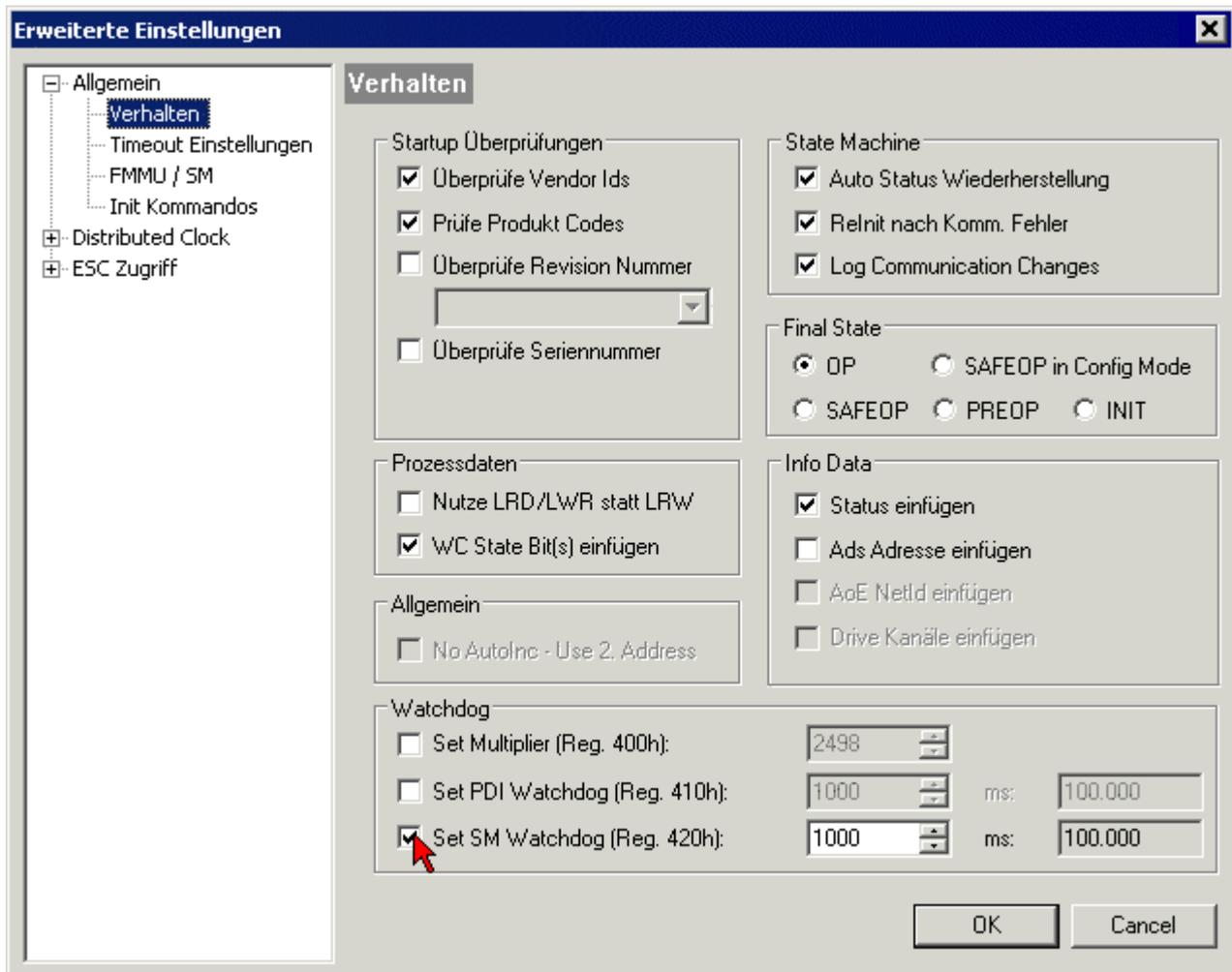


Abb. 6: Karteireiter EtherCAT -> Erweiterte Einstellungen -> Verhalten -> Watchdog

Anmerkungen:

- Das Multiplier-Register 400h (hexadezimal, also 0x0400), ist für beide Watchdogs gültig.
- Jeder Watchdog hat seine eigene Timer-Einstellung 410h bzw. 420h, die zusammen mit dem Multiplier eine resultierende Zeit ergibt.
- Wichtig: die Multiplier-/Timer-Einstellung wird nur dann beim EtherCAT-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.
- Die heruntergeladenen Werte können in den ESC-Registern 400h, 410h und 420h eingesehen werden: ESC Zugriff -> Speicher (ESC Access -> Memory).

SM-Watchdog (SyncManager-Watchdog)

Der SyncManager-Watchdog wird bei jeder erfolgreichen EtherCAT-Prozessdatenkommunikation mit der Klemme zurückgesetzt. Findet z. B. durch eine Leitungsunterbrechung länger als die eingestellte und aktivierte SM-Watchdog-Zeit keine EtherCAT-Prozessdatenkommunikation mit der Klemme statt, löst der Watchdog aus. Der Status der Klemme (in der Regel OP) bleibt davon unberührt. Der Watchdog wird erst wieder durch einen erfolgreichen EtherCAT-Prozessdatenzugriff zurückgesetzt.

Der SyncManager-Watchdog ist also eine Überwachung auf korrekte und rechtzeitige Prozessdatenkommunikation zwischen Master und ESC, die allein auf EtherCAT-Ebene abläuft.

Die maximal mögliche Watchdog-Zeit ist geräteabhängig. Beispielsweise beträgt sie bei „einfachen“ EtherCAT-Slaves (ohne Firmware) mit Watchdog-Ausführung im ESC in der Regel bis zu 170 Sekunden. Bei komplexen EtherCAT-Slaves (mit Firmware) wird die SM-Watchdog-Funktion in der Regel zwar über Register 400h/420h parametrisiert, aber vom Microcontroller (μC) ausgeführt und kann deutlich darunter liegen. Außerdem kann die Ausführung dann einer gewissen Zeitunsicherheit unterliegen. Da der TwinCAT-Dialog ggf. Eingaben bis 65535 zulässt, wird ein Test der gewünschten Watchdog-Zeit empfohlen.

PDI-Watchdog (Process Data Watchdog)

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

PDI 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, nun aber von der Applikationsseite aus betrachtet.

Berechnung

Watchdog-Zeit = $[1/25 \text{ MHz} * (\text{Watchdog-Multiplier} + 2)] * \text{SM/PDI Watchdog}$

Beispiel: Default-Einstellung Multiplier = 2498, SM-Watchdog = 1000 => 100 ms

Der Wert in „Watchdog-Multiplier + 2“ in der oberen Formel entspricht der Anzahl 40ns-Basisticks, die einen Watchdog-Tick darstellen.

⚠ VORSICHT

Ungewolltes Verhalten des Systems möglich!

Die Abschaltung des SM-Watchdog durch SM-Watchdog = 0 funktioniert erst in Klemmen ab Revision -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
- Operational

- Bootstrap

Regulärer Zustand eines jeden EtherCAT-Slaves nach dem Hochlauf ist der Status Operational (OP).

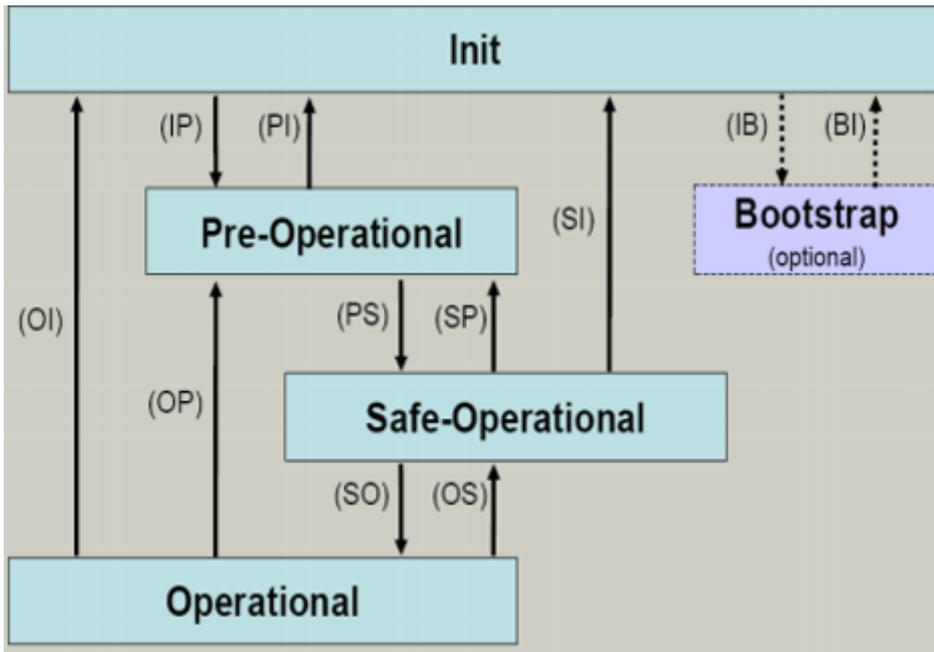


Abb. 7: Zustände der EtherCAT State Machine

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 Prozessdatenkommunikation möglich. Der EtherCAT-Master initialisiert die Sync-Manager-Kanäle für Prozessdaten (ab Sync-Manager-Kanal 2), die Kanäle der Fieldbus Memory Management Unit (FMMU) und, falls der Slave ein konfigurierbares Mapping unterstützt, das Mapping der Prozessdatenobjekte (PDOs) 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 Default-Einstellungen 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. die Einstellungen für die Distributed Clocks korrekt sind. Bevor er den Zustandswechsel quittiert, kopiert der EtherCAT-Slave aktuelle Inputdaten in die entsprechenden Dual Port (DP)-RAM-Bereiche des ESC.

Im Zustand *Safe-Op* ist Mailbox- und Prozessdatenkommunikation 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 Überwachung mittels Watchdog bringt die Ausgänge im ESC-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 Überwachung 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 Mailboxkommunikation 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- und Prozessdatenkommunikation.

3.5 CoE-Interface

Allgemeine Beschreibung

Das CoE-Interface (CAN application protocol over EtherCAT Interface) 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 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-Datentypen 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, Gerätenamen, Abgleichwerte für analoge Messungen 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_{dez})
- Subindex: 0x00...0xFF (0...255_{dez})

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 Identitätsinformationen 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 („Eingänge“ aus Sicht des EtherCAT-Masters)
- 0x7000: Hier liegen die Ausgangs-PDO („Ausgänge“ 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:

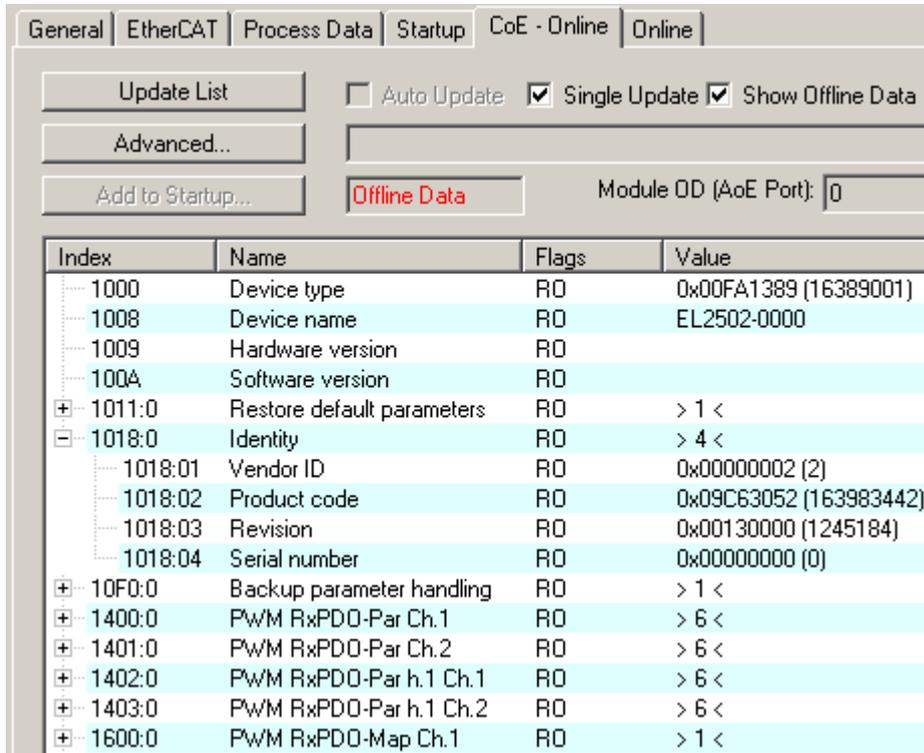


Abb. 8: Karteireiter „CoE-Online“

In der Abbildung „Karteireiter ‚CoE-Online‘“ sind die im Gerät „EL2502“ verfügbaren CoE-Objekte von 0x1000 bis 0x1600 zu sehen, die Subindizes von 0x1018 sind aufgeklappt.

HINWEIS

Veränderungen im CoE-Verzeichnis (CAN over EtherCAT-Verzeichnis), Programmzugriff

Beachten Sie bei Verwendung/Manipulation der CoE-Parameter die allgemeinen CoE-Hinweise im Kapitel „CoE-Interface“ der EtherCAT-System-Dokumentation:

- Startup-Liste führen für den Austauschfall,
- Unterscheidung zwischen Online/Offline Dictionary,
- Vorhandensein aktueller XML-Beschreibung (Download von der [Beckhoff Website](#)),
- "CoE-Reload" zum Zurücksetzen der Veränderungen
- Programmzugriff im Betrieb über die PLC (s. [TwinCAT3 | PLC-Bibliothek: Tc2 EtherCAT](#) und [Beispielprogramm R/W CoE](#))

Datenerhaltung und Funktion „NoCoeStorage“

Einige, insbesondere die vorgesehenen Einstellungsparameter des Slaves, sind veränderlich und beschreibbar,

- über den System Manager (siehe Abb. „Karteireiter ‚CoE-Online‘“) durch Anklicken. Dies bietet sich bei der Inbetriebnahme der Anlage bzw. 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 bzw. PLC über ADS z. B. durch die Bausteine aus der TcEtherCAT.lib Bibliothek. Dies wird für Änderungen während der Anlagenlaufzeit empfohlen oder wenn kein System Manager bzw. Bedienpersonal zur Verfügung steht.

● Datenerhaltung

i Werden online auf dem Slave CoE-Parameter geändert, wird dies in Beckhoff-Geräten üblicherweise ausfallsicher im Gerät (EEPROM) gespeichert. D. h. nach einem Neustart (Re Power) 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 Lebensdauerergrenze des EEPROMs 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 der entsprechenden Dokumentation zu entnehmen.

- Wird diese unterstützt: Die Funktion ist per einmaligem Eintrag des Codeworts 0x12345678 im CoE-Index 0xF008 zu aktivieren. Die Funktion ist solange aktiv, wie das Codewort unverändert bleibt. 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 diese nicht unterstützt: Eine fortlaufende Änderung von CoE-Werten ist angesichts der o.a. Lebensdauerergrenze nicht zulässig.

● Startup-Liste

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

Wenn EtherCAT-Slaves verwendet werden, die lokal CoE-Werte 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.

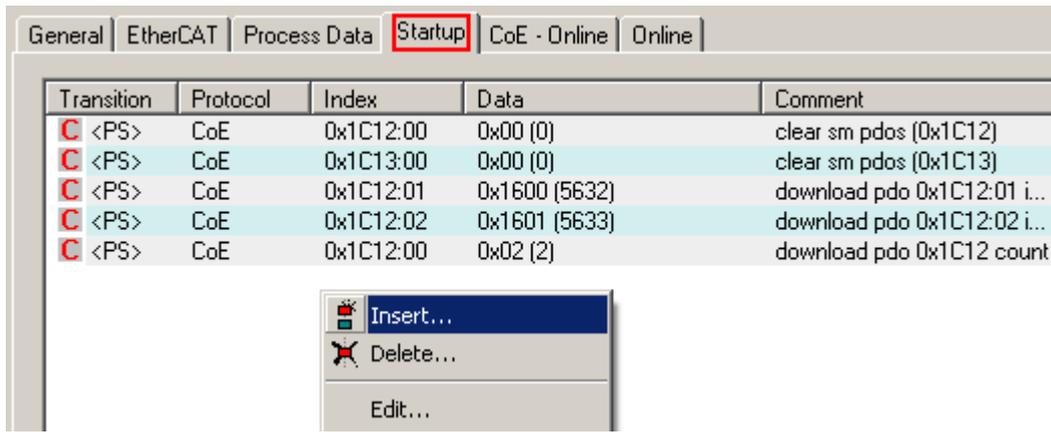


Abb. 9: Startup-Liste im TwinCAT System Manager

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 ebenfalls angelegt werden.

Online- / Offline Verzeichnis

Im Rahmen der Arbeit mit dem TwinCAT System Manager ist zu differenzieren, ob das EtherCAT-Gerät gegenwärtig „verfügbar“ ist, also angeschaltet und über EtherCAT verbunden – somit **online** – oder ob eine Konfiguration **offline** erstellt wird, ohne dass Slaves angeschlossen sind.

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

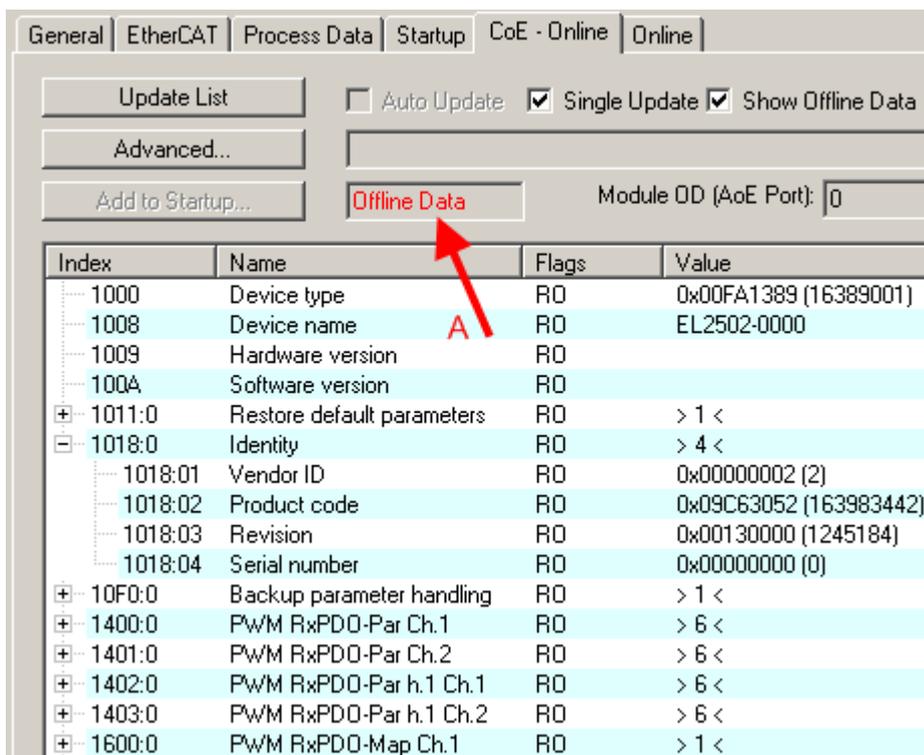


Abb. 10: Offline-Verzeichnis

- 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 im CoE angezeigt.
 - ist ein grünes **Online Data** zu sehen.

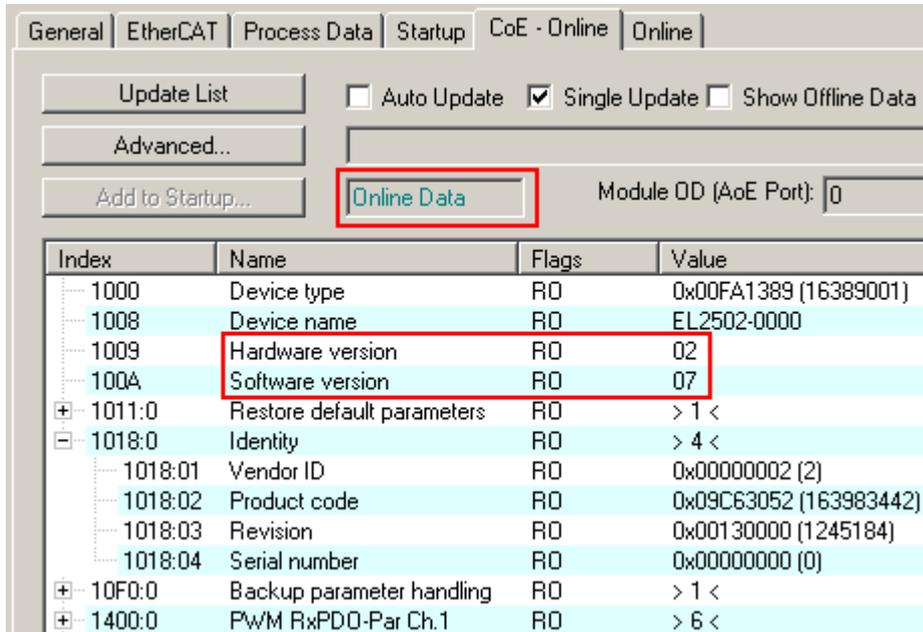


Abb. 11: Online-Verzeichnis

Kanalweise Ordnung

Das CoE-Verzeichnis ist in EtherCAT-Geräten angesiedelt, die meist mehrere funktional gleichwertige Kanäle umfassen; z. B. hat eine vierkanalige Analogeingangsklemme 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_{dez} bzw. 10_{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.

- Beim Umgang mit den Bauteilen ist auf elektrostatische Entladung zu achten; außerdem ist das direkte Berühren der Federkontakte (siehe Abbildung) zu vermeiden.
- Der Kontakt mit hoch isolierenden Stoffen (Kunstfasern, Kunststofffolien etc.) sollte beim gleichzeitigen Umgang mit Komponenten vermieden werden.
- Beim Umgang mit den Komponenten ist auf eine sachgemäße Erdung der Umgebung (Arbeitsplatz, Verpackung und Personen) zu achten.
- Jede Busstation muss auf der rechten Seite mit der Endkappe [EL9011](#) oder [EL9012](#) abgeschlossen werden, um die Schutzart und den ESD-Schutz zu gewährleisten.

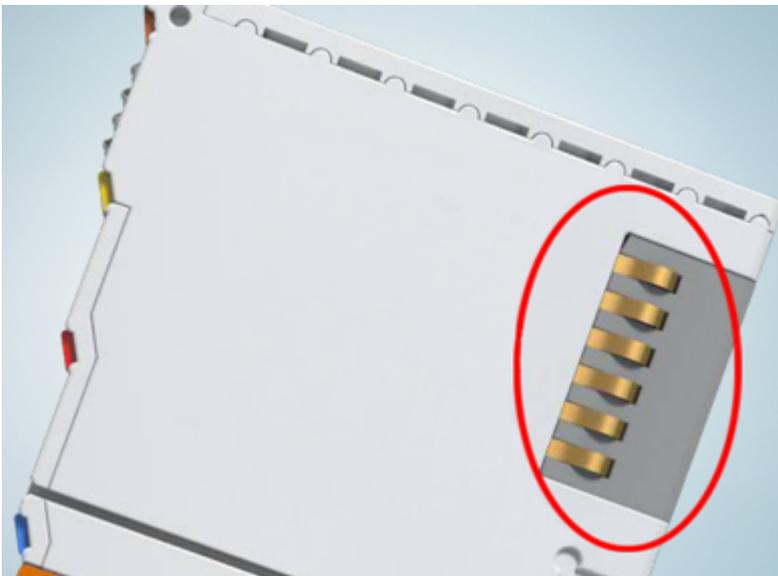


Abb. 12: Federkontakte der Beckhoff I/O-Komponenten

4.2 UL-Hinweise

⚠ VORSICHT	
	<p>Application</p> <p>The modules are intended for use with Beckhoff's UL Listed EtherCAT System only.</p>
⚠ VORSICHT	
	<p>Examination</p> <p>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>
⚠ VORSICHT	
	<p>For devices with Ethernet connectors</p> <p>Not for connection to telecommunication circuits.</p>

Grundlagen

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



4.3 Hinweis zur Spannungsversorgung

WARNUNG

Spannungsversorgung aus SELV- / PELV-Netzteil!

Zur Versorgung dieses Geräts müssen SELV- / PELV-Stromkreise (Sicherheitskleinspannung, "safety extra-low voltage" / Schutzkleinspannung, „protective extra-low voltage“) nach IEC 61010-2-201 verwendet werden.

Hinweise:

- Durch SELV/PELV-Stromkreise entstehen eventuell weitere Vorgaben aus Normen wie IEC 60204-1 et al., zum Beispiel bezüglich Leitungsabstand und -isolierung.
- Eine SELV-Versorgung liefert sichere elektrische Trennung und Begrenzung der Spannung ohne Verbindung zum Schutzleiter, eine PELV-Versorgung benötigt zusätzlich eine sichere Verbindung zum Schutzleiter.

4.4 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.

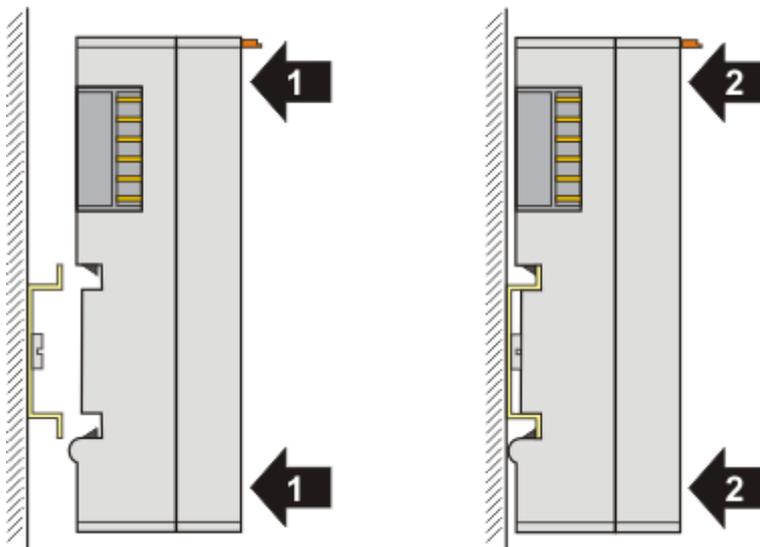
⚠ WARNUNG

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

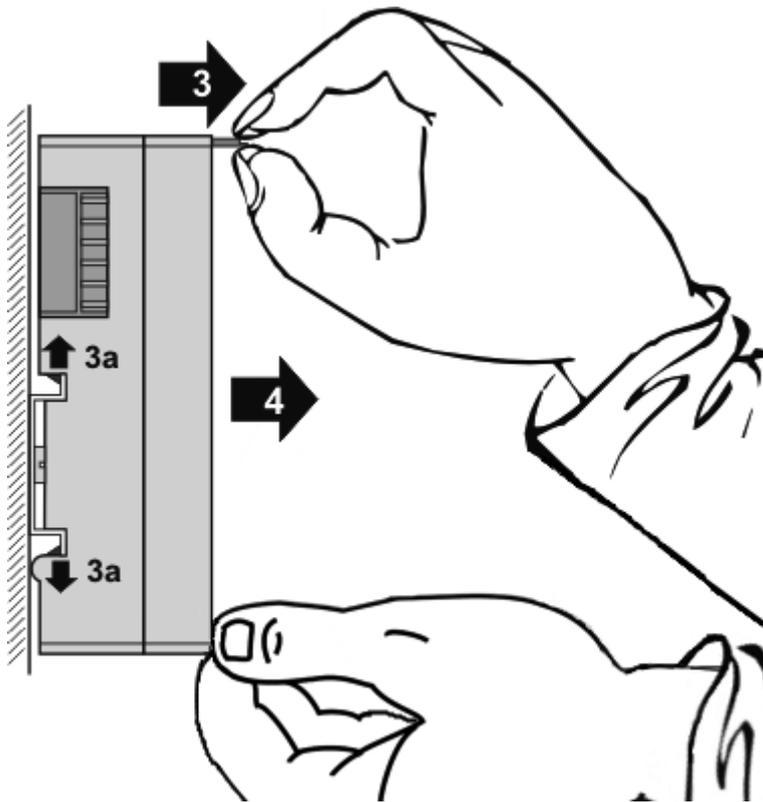


und drücken Sie (1) das Klemmenmodul gegen die Tragschiene, bis es auf der Tragschiene einrastet (2).

- 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

4.5 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 Richtung der Erdbeschleunigung.

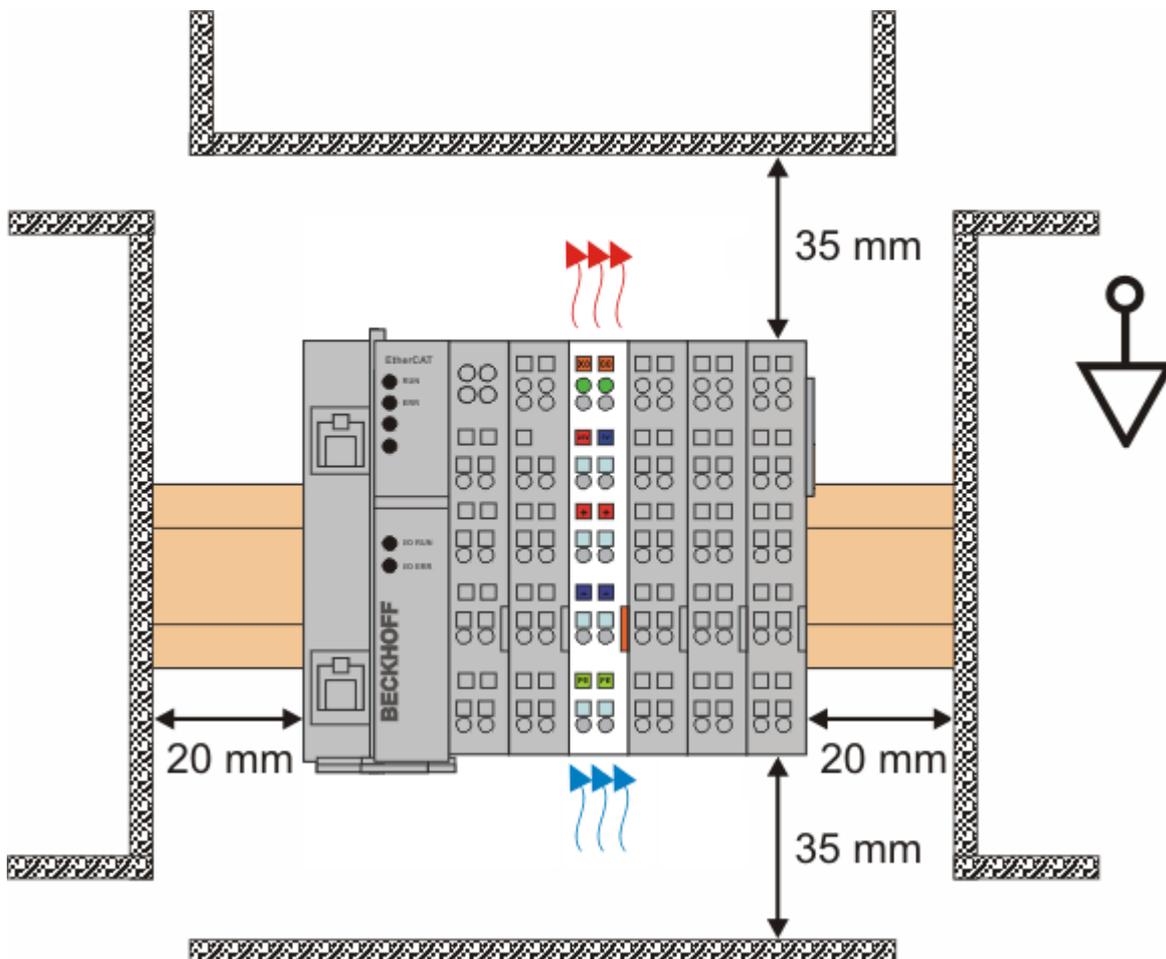


Abb. 13: Empfohlene Abstände bei Standard-Einbaulage

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.

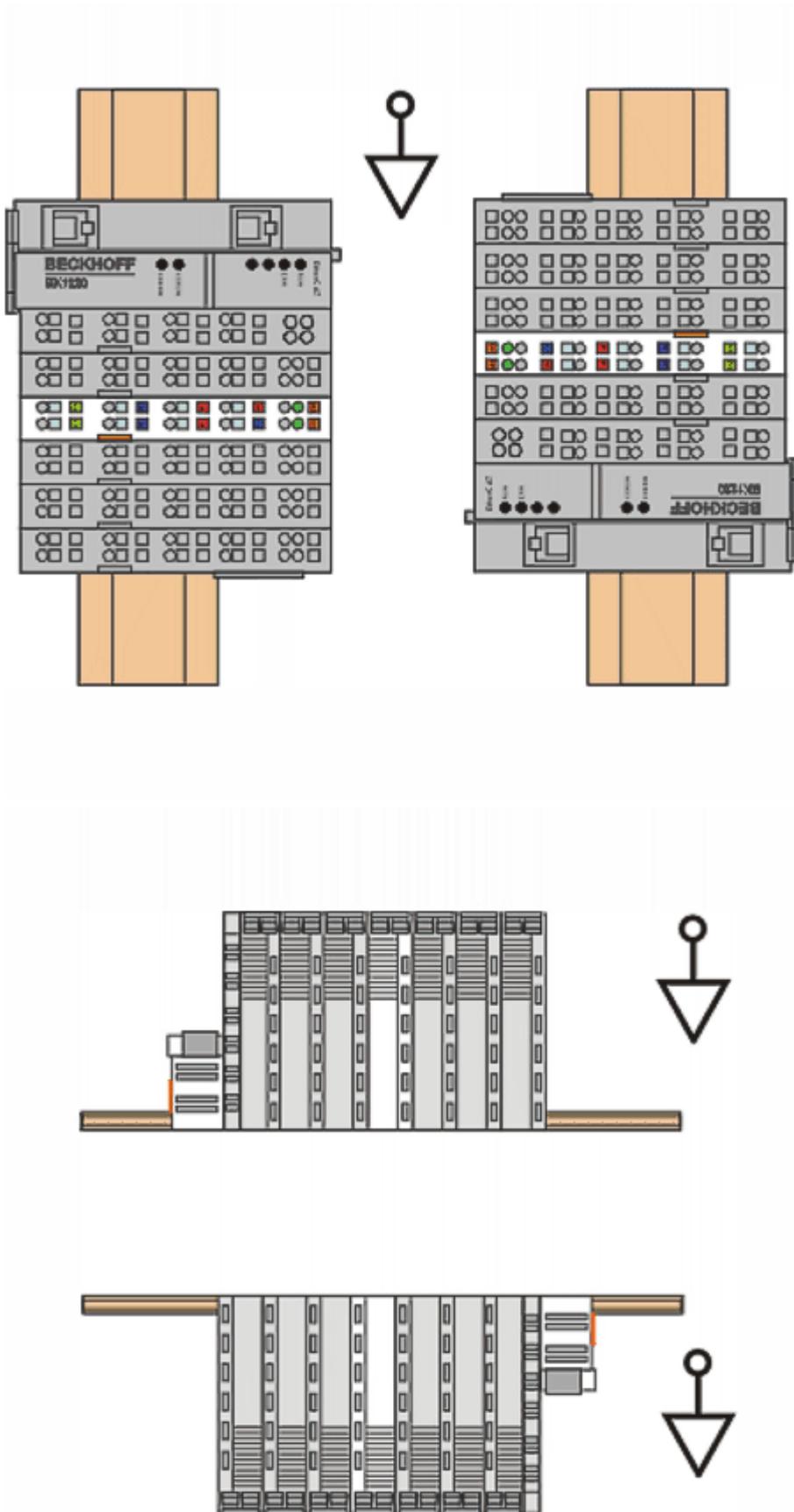


Abb. 14: Weitere Einbaulagen

4.6 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)

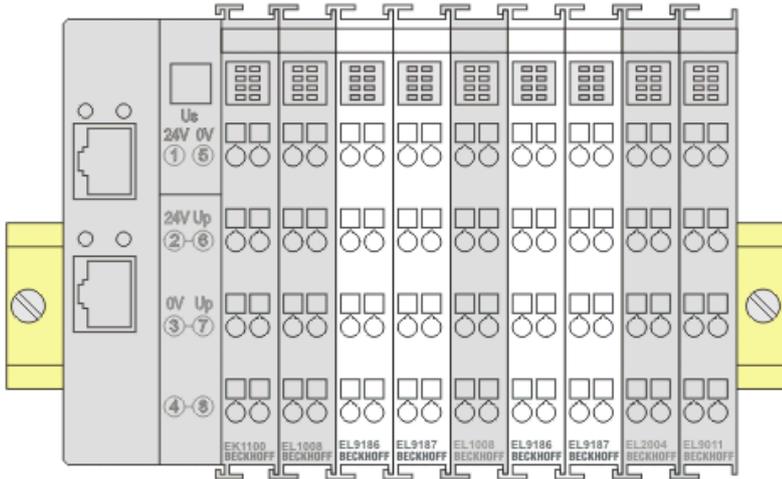


Abb. 15: Korrekte Positionierung

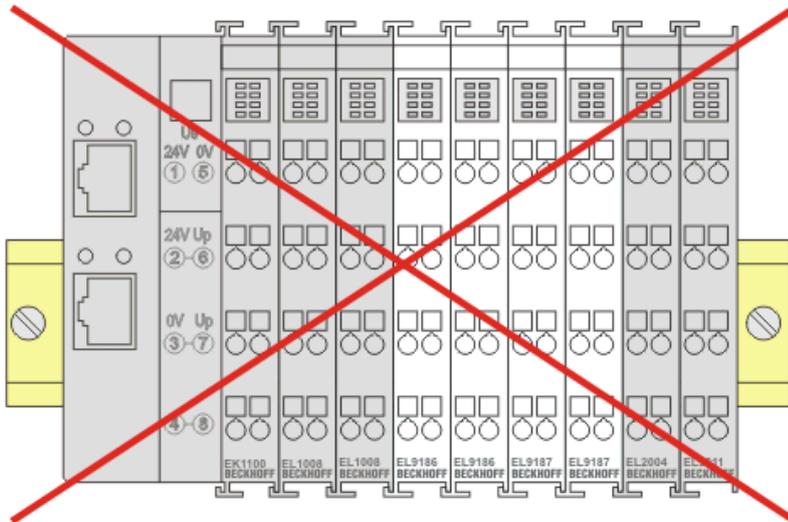


Abb. 16: Inkorrekte Positionierung

4.7 Hinweis zur Schirmung

HINWEIS

Niederohmige Schirmung durch externe Schirmauflage

Die Schirmung der EL6631 ist kapazitiv mit der Hutschiene verbunden.

Wenn eine niederohmige Schirmung gewünscht oder erforderlich ist, muss der Schirm mit einer externen Schirmauflage (z.B. mit dem Klemmbügel [ZB5800](#)) verbunden werden.

4.8 LEDs und Anschlussbelegung

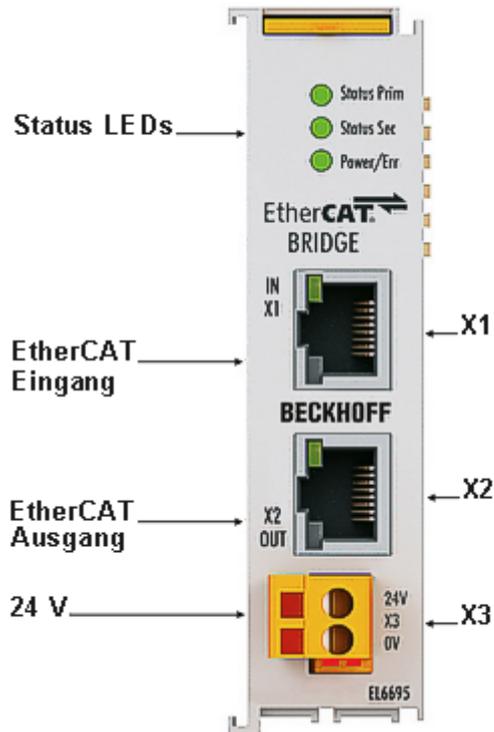


Abb. 17: EL6695 - Anschluss

Anschluss

Anschlusspunkt	Beschreibung
X1	EtherCAT Eingang (RJ45 mit 10BASE-T/100BASE-TX Ethernet)
X2	EtherCAT Ausgang (RJ45 mit 10BASE-T/100BASE-TX Ethernet)
X3	Zweipoliger Buchsenklemmenanschluss (24 V _{DC}), Stromversorgung Sekundärseite

LEDs

LED	Farbe	Bedeutung	
X1/X2	grün	aus	keine Verbindung zum EtherCAT-Netzwerk
		an	Verbindung zum EtherCAT-Netzwerk aufgebaut (LINK)
		flimmernd	Datenaustausch mit Teilnehmern im EtherCAT-Netzwerk (ACT)

● Diagnose-LEDs

i Die Beschreibung der LEDs finden Sie im Kapitel [Diagnose-LEDs](#) [► 43].

4.9 Diagnose

Diagnose LEDs

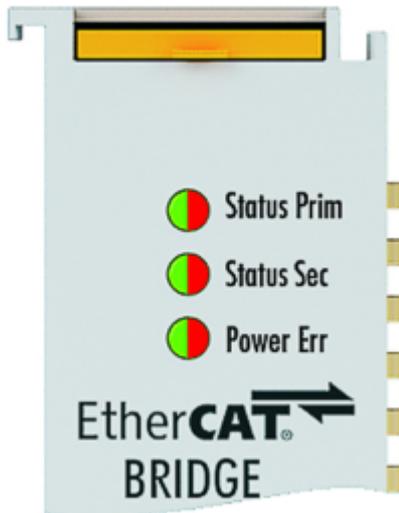


Abb. 18: Anzeige des Betriebszustandes über Diagnose-LEDs

LED	Anzeige	Bedeutung
Status Prim Status Sec EtherCAT State LED (rot/grün), Betriebszustand der jeweiligen Slave Seite	aus	Zustand der <u>EtherCAT State Machine</u> : INIT = Initialisierung der Klemme oder BOOTSTRAP = Funktion für <u>Firmware Updates</u> der Klemme
	grün blinkend	Zustand der EtherCAT State Machine: PREOP = Funktion für Mailbox-Kommunikation und abweichende Standard-Einstellungen gesetzt
	grün Einzelblitz	Zustand der EtherCAT State Machine: SAFEOP = Überprüfung der Kanäle des <u>Sync-Managers</u> und der Distributed Clocks. Ausgänge bleiben im sicheren Zustand
	grün an	Zustand der EtherCAT State Machine: OP = normaler Betriebszustand; Mailbox- und Prozessdatenkommunikation ist möglich
	rot blinkend	Interner Fehler
	rot Einzelblitz	AL status register Fehler
	rot Doppelblitz	Process Data Watchdog timeout
	rot an	PDI Watchdog timeout, oder Selbsttest-Fehler
	aus	Keine Versorgung
Power/Error (Versorgungszustand)	orange	Versorgung nur von 1 Seite, primär oder sekundär
	rot blinkend	FW Update gescheitert, Klemme an <u>Service</u> [► 195] schicken
	grün	Versorgung von beiden Seiten anliegend, 24 V Eingang wird genutzt

4.10 Entsorgung



Die mit einer durchgestrichenen Abfalltonne gekennzeichneten Produkte dürfen nicht in den Hausmüll. Das Gerät gilt bei der Entsorgung als Elektro- und Elektronik-Altgerät. Die nationalen Vorgaben zur Entsorgung von Elektro- und Elektronik-Altgeräten sind zu beachten.

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:

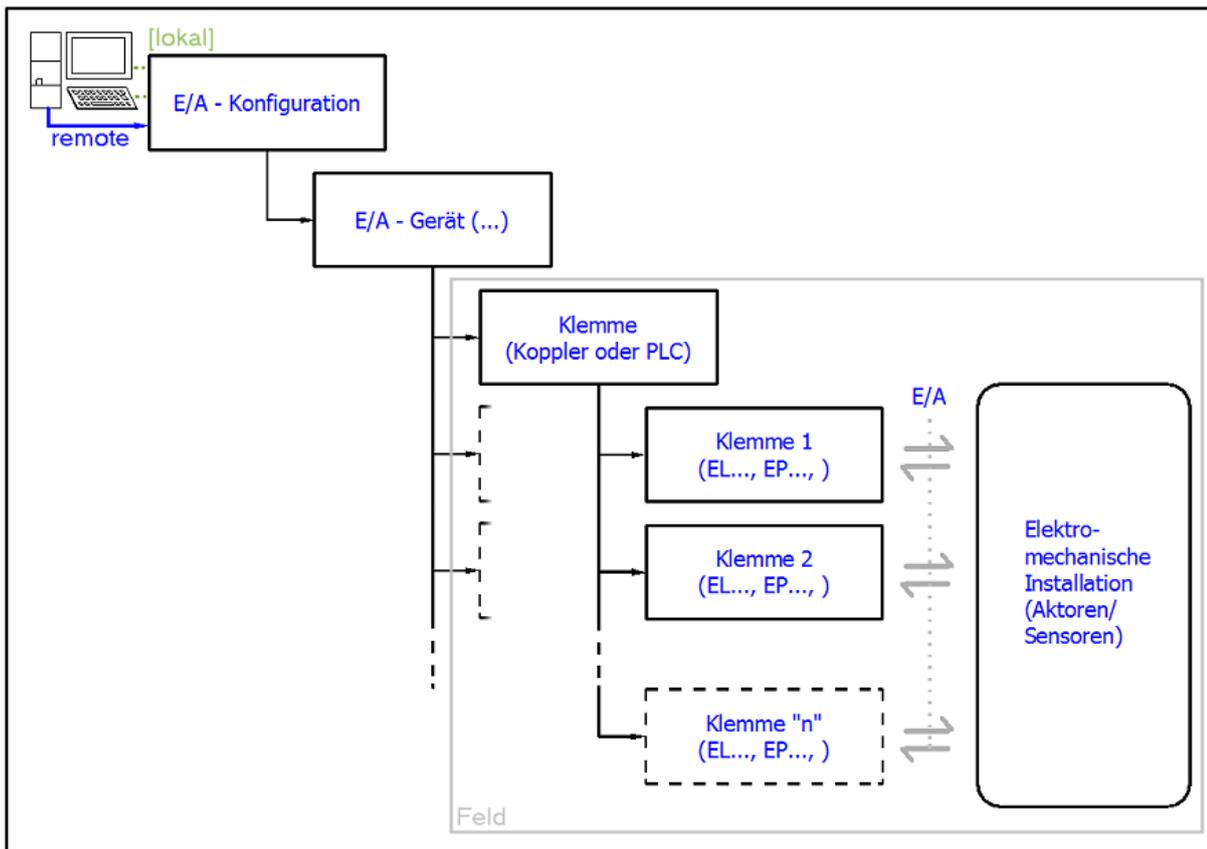


Abb. 19: Bezug von der Anwender Seite (Inbetriebnahme) zur Installation

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_{DC})
- 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_{DC}; 0,5 A)
- (Optional über X000: ein Link zu einen externen PC für die Benutzeroberfläche)

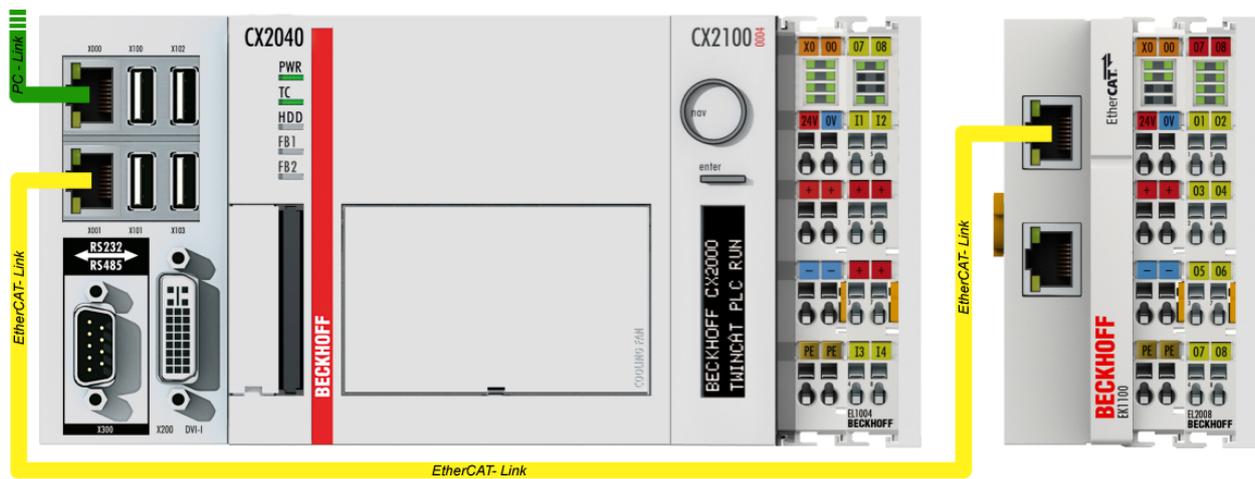


Abb. 20: Aufbau der Steuerung mit Embedded-PC, Eingabe (EL1004) und Ausgabe (EL2008)

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 Managers.

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:

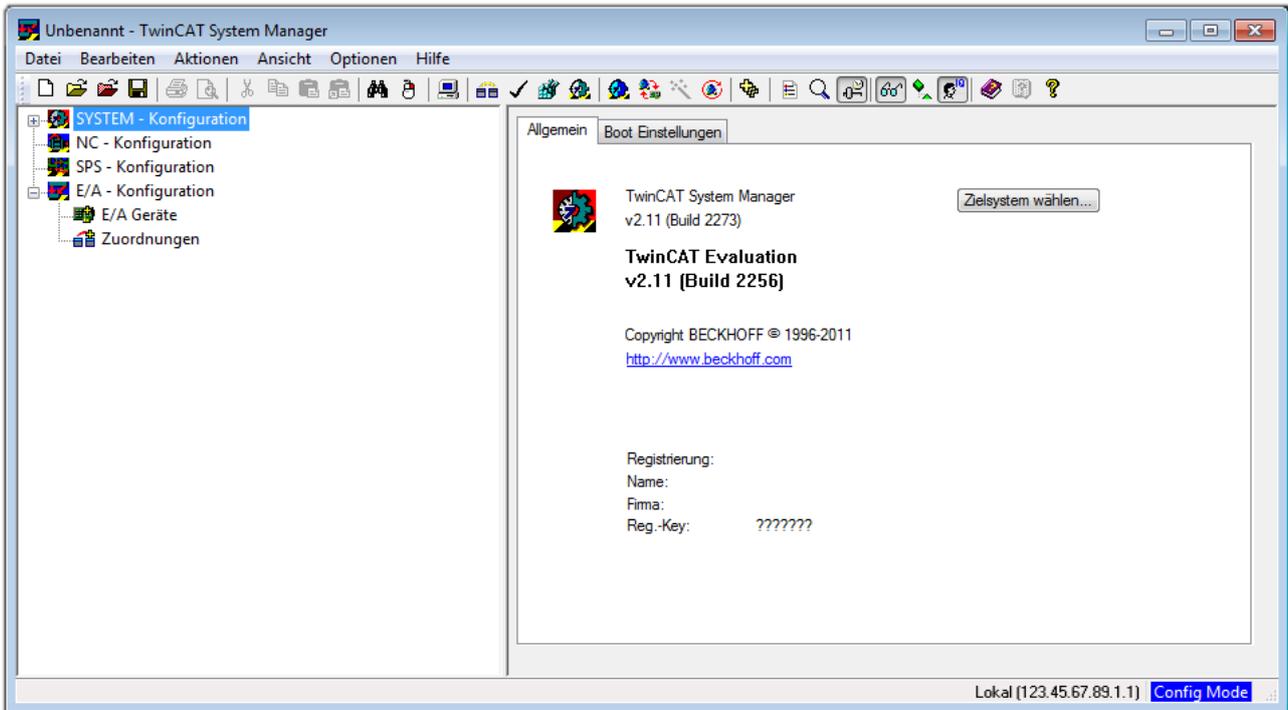


Abb. 21: Initiale Benutzeroberfläche TwinCAT 2

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 [► 50]“ 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:

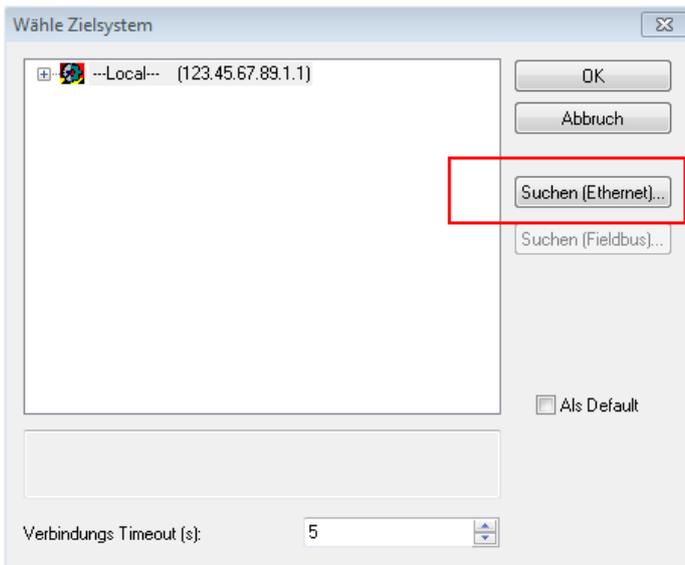


Abb. 22: Wähle Zielsystem

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 Rechnername nicht genau bekannt)
- die bekannte Rechner - IP oder AmsNetId einzutragen

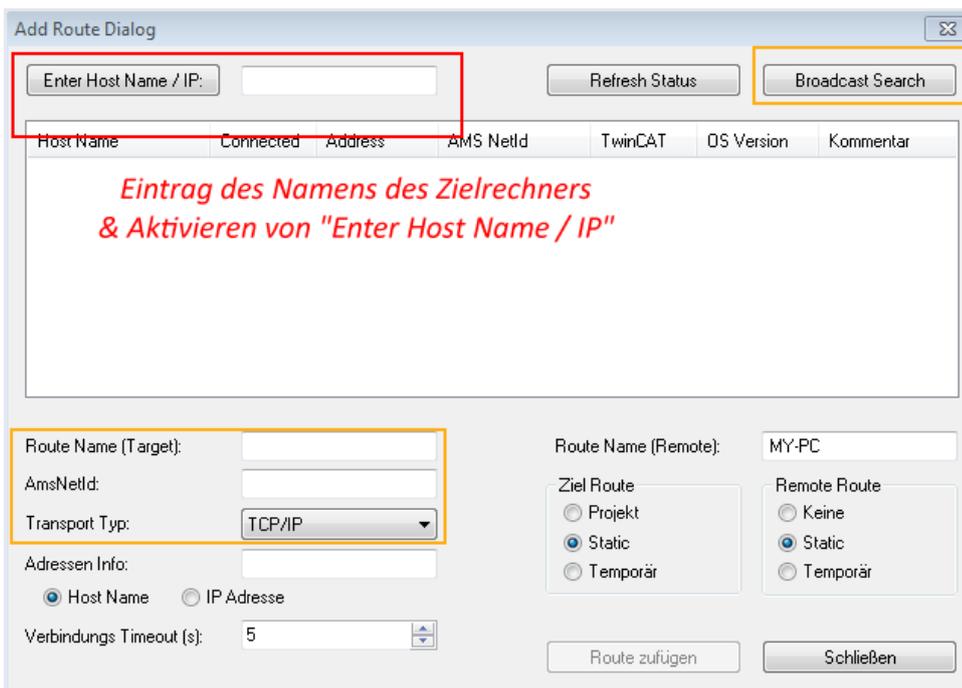
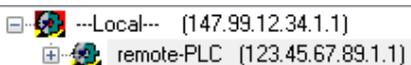


Abb. 23: PLC für den Zugriff des TwinCAT System Managers festlegen: Auswahl des Zielsystems

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.

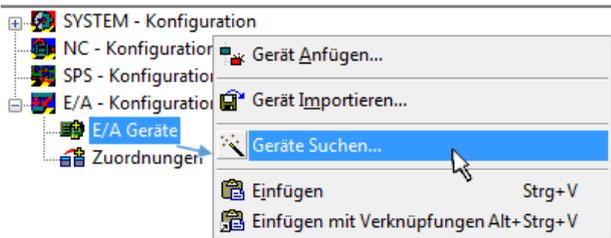


Abb. 24: Auswahl „Gerät Suchen...“

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

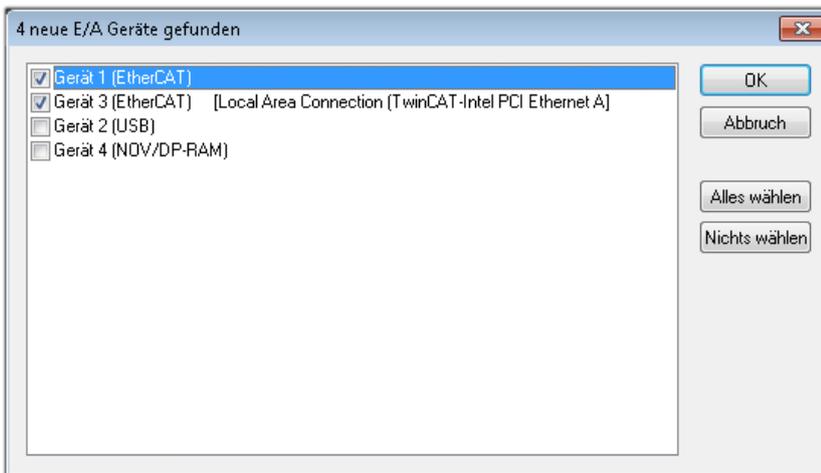


Abb. 25: Automatische Erkennung von E/A-Geräten: Auswahl der einzubindenden Geräte

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](#) [► 46] sieht das Ergebnis wie folgt aus:

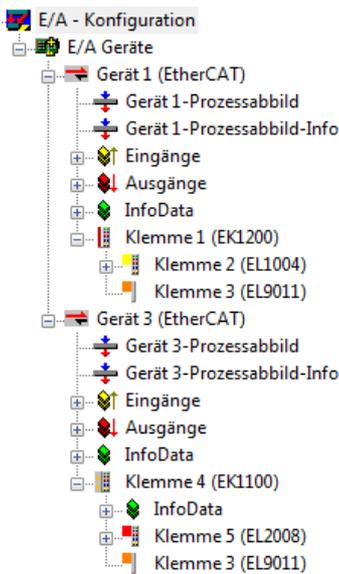


Abb. 26: Abbildung der Konfiguration im TwinCAT 2 System Manager

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:

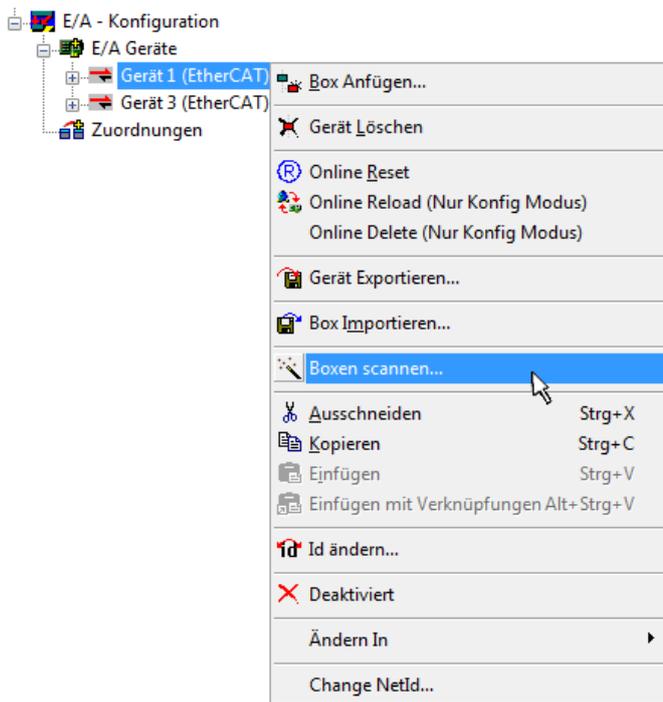


Abb. 27: Einlesen von einzelnen an einem Gerät befindlichen Klemmen

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:

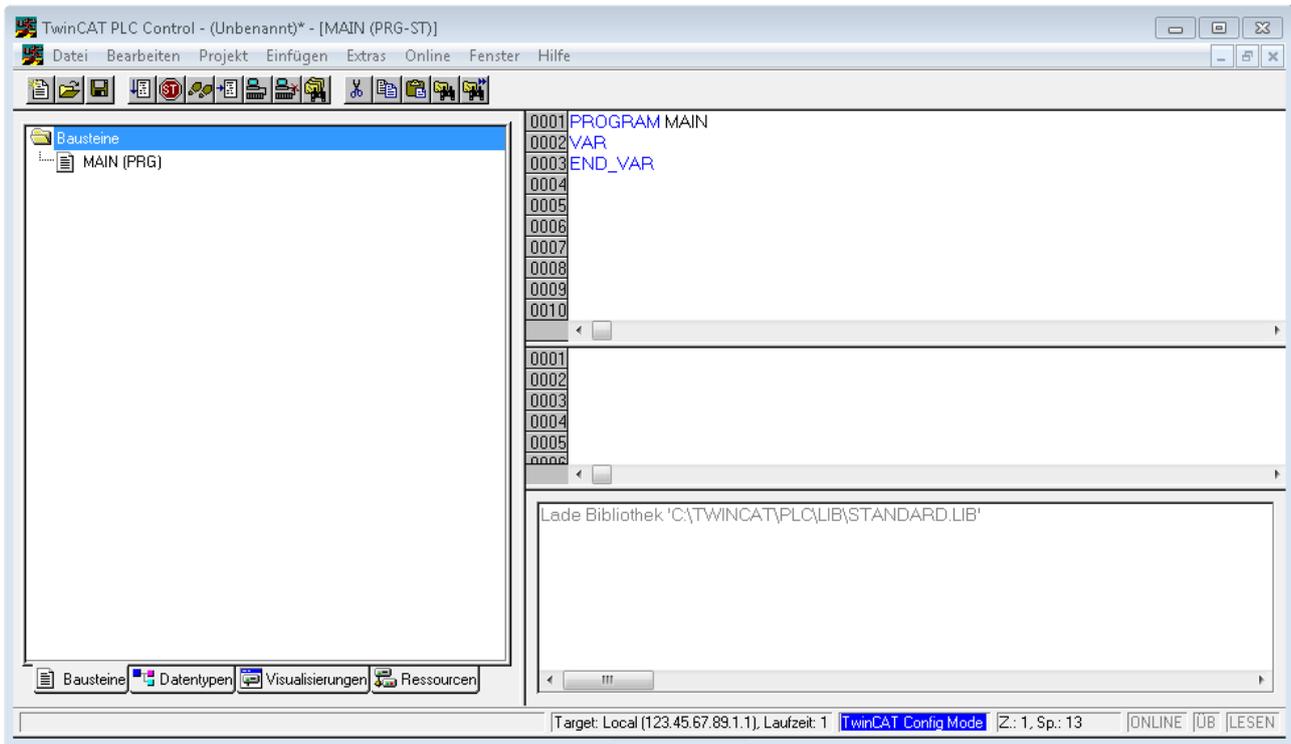


Abb. 28: TwinCAT PLC Control nach dem Start

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

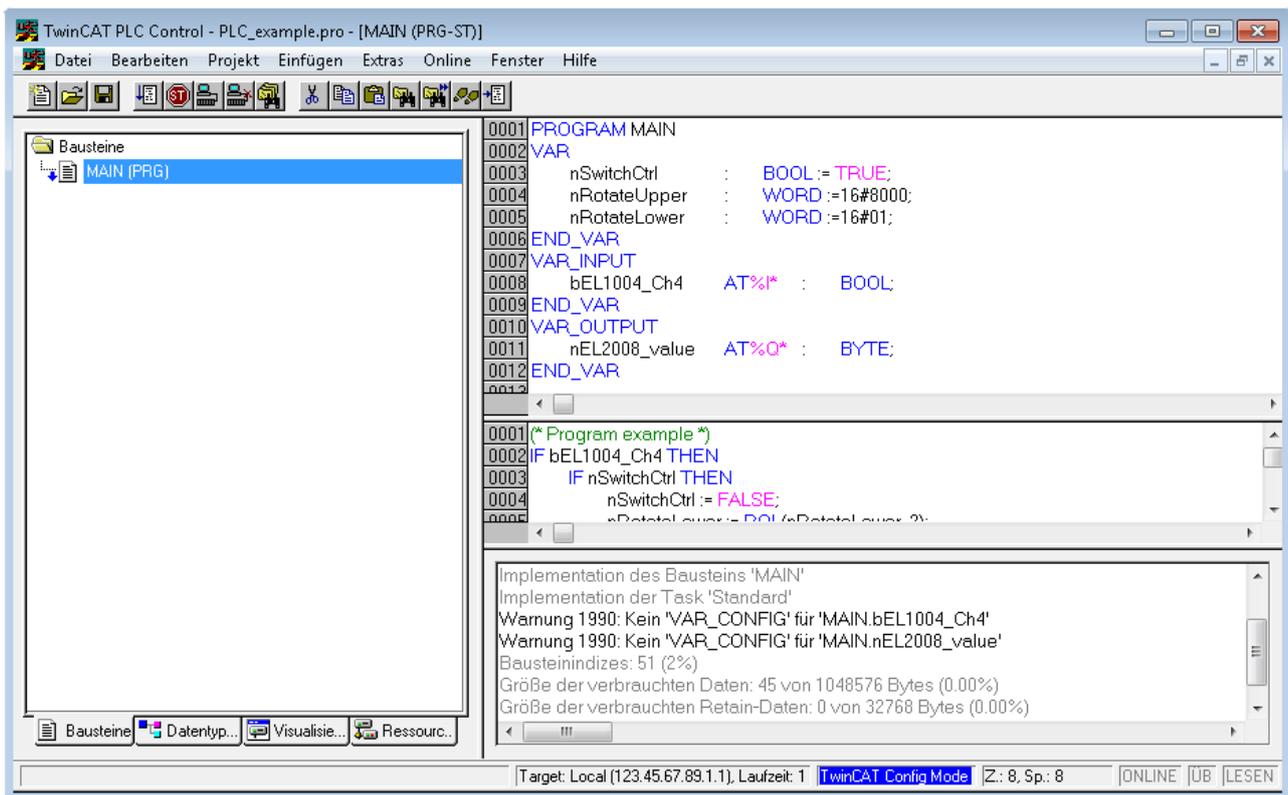


Abb. 29: Beispielprogramm mit Variablen nach einem Kompilervorgang (ohne Variablenanbindung)

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...“:

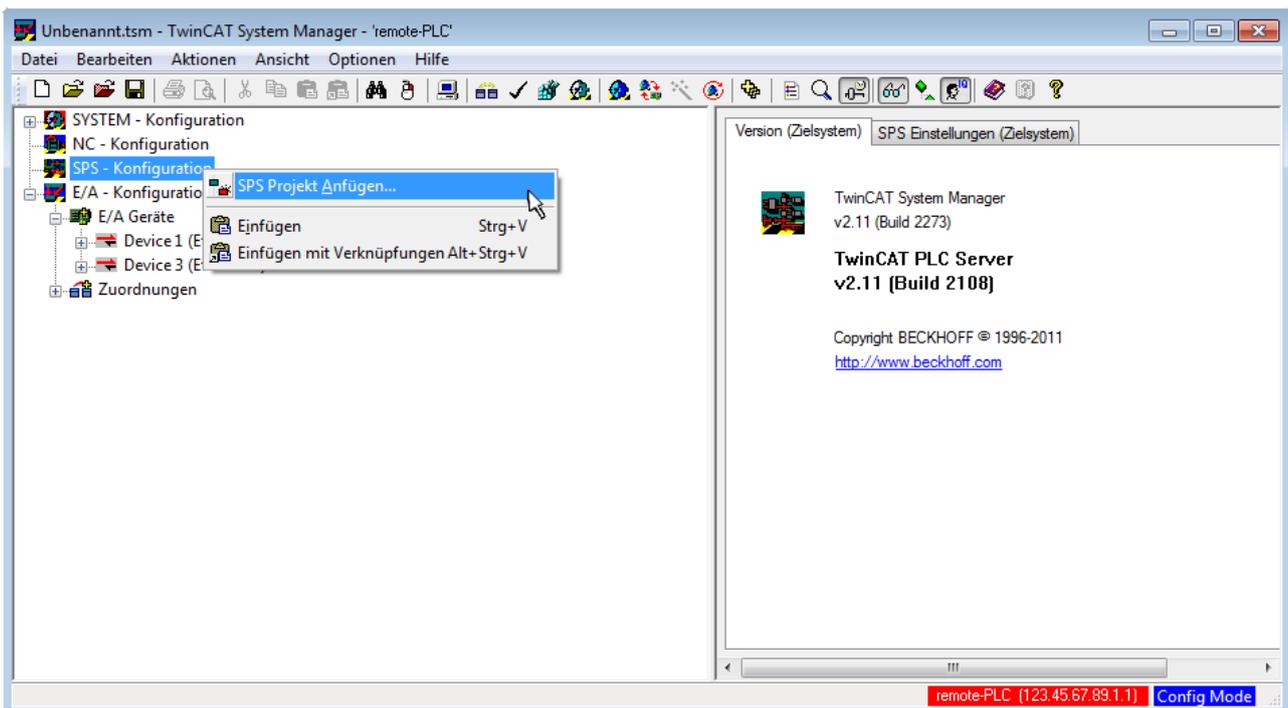


Abb. 30: Hinzufügen des Projektes des TwinCAT PLC Control

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

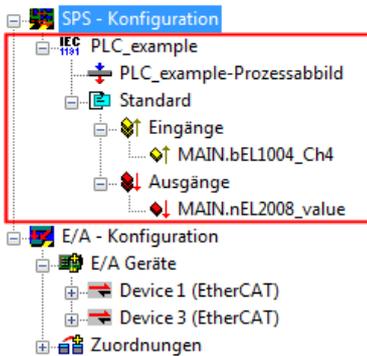


Abb. 31: Eingebundenes PLC-Projekt in der SPS-Konfiguration des System Managers

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:

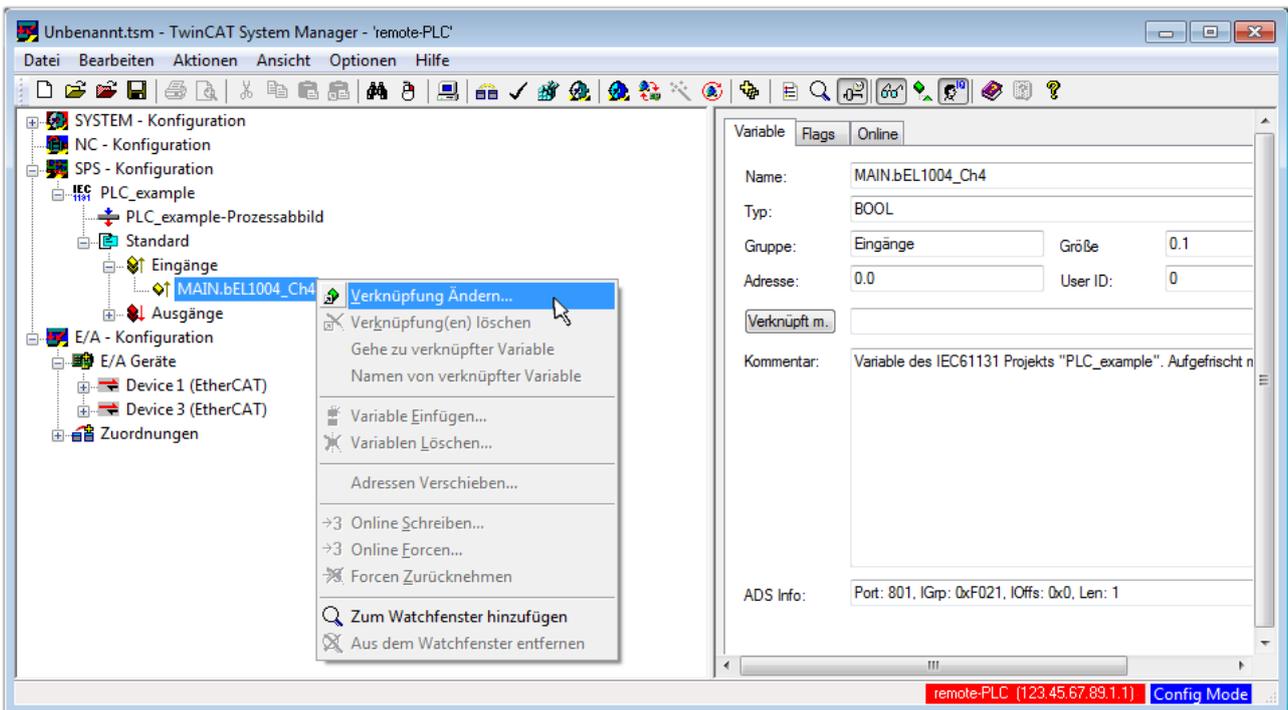


Abb. 32: Erstellen der Verknüpfungen PLC-Variablen zu Prozessobjekten

In dem dadurch geöffneten Fenster kann aus dem SPS-Konfigurationsbaum das Prozessobjekt für die Variable „bEL1004_Ch4“ vom Typ BOOL selektiert werden:

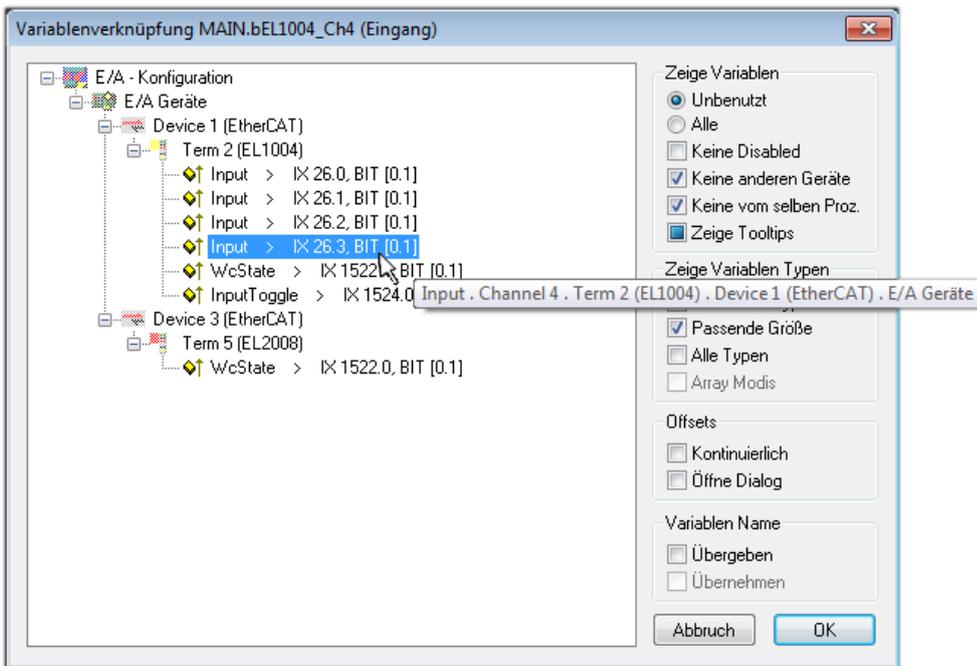


Abb. 33: Auswahl des PDO vom Typ BOOL

Entsprechend der Standardeinstellungen 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:

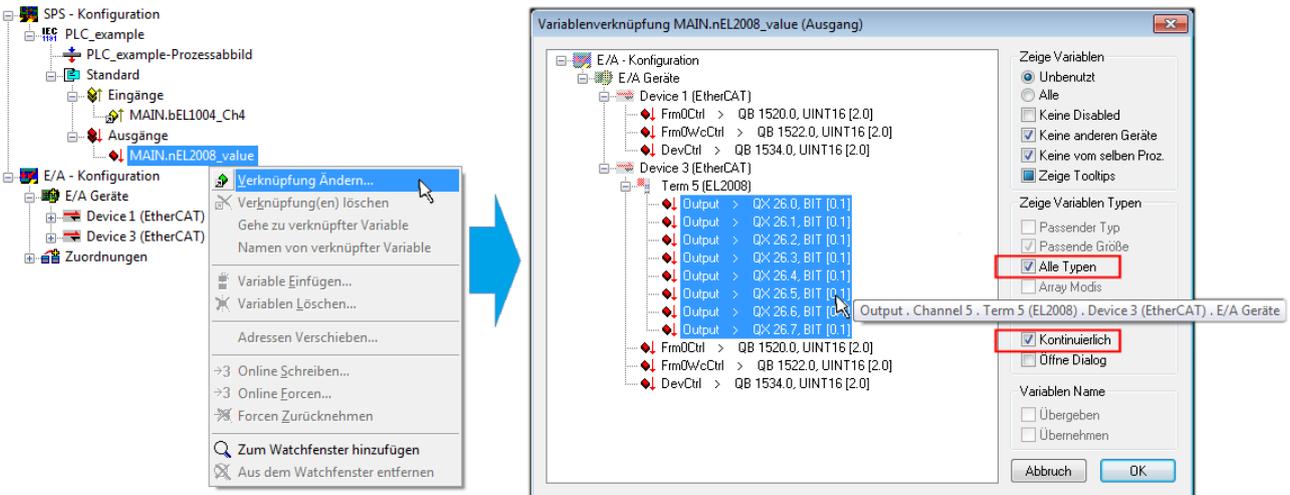


Abb. 34: Auswahl von mehreren PDO gleichzeitig: Aktivierung von „Kontinuierlich“ und „Alle Typen“

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:

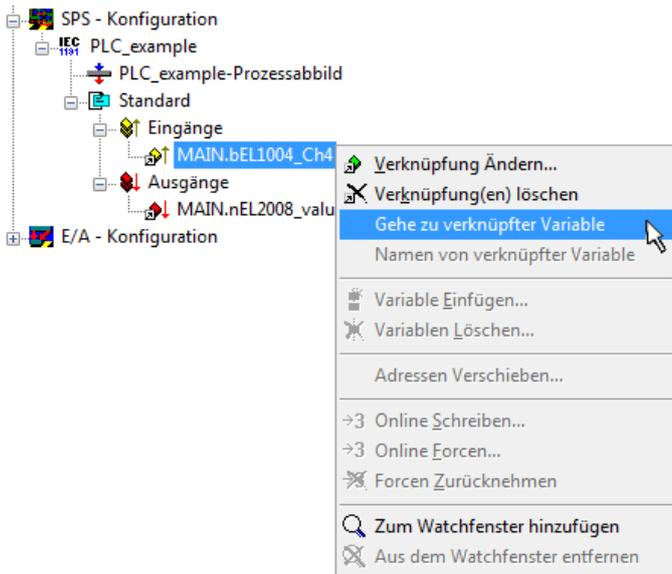


Abb. 35: Anwendung von „Goto Link Variable“ am Beispiel von „MAIN.bEL1004_Ch4“

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 ein Byte, Word, Integer oder ein ähnliches PDO, so ist es möglich dies wiederum einen Satz von bit-typisierten Variablen 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 darauffolgenden 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:

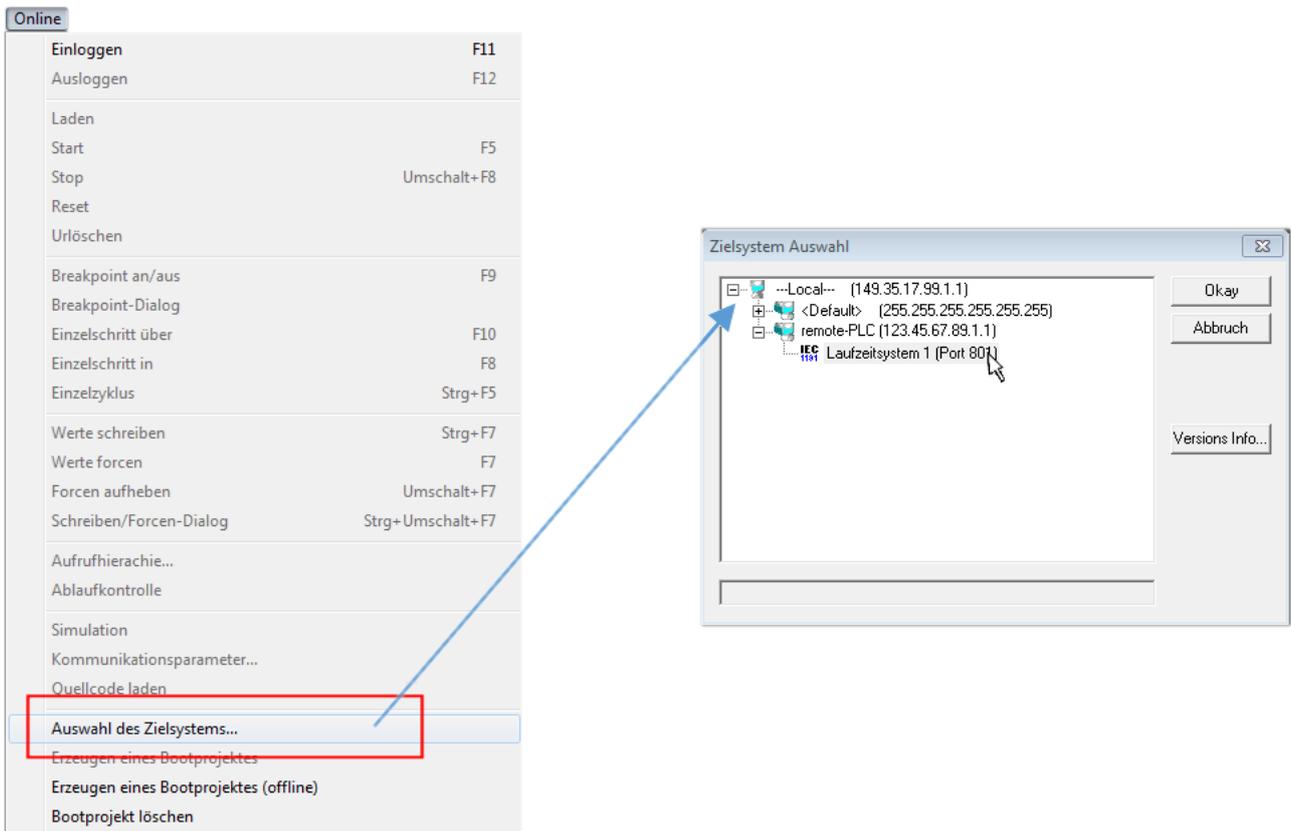


Abb. 36: Auswahl des Zielsystems (remote)

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:

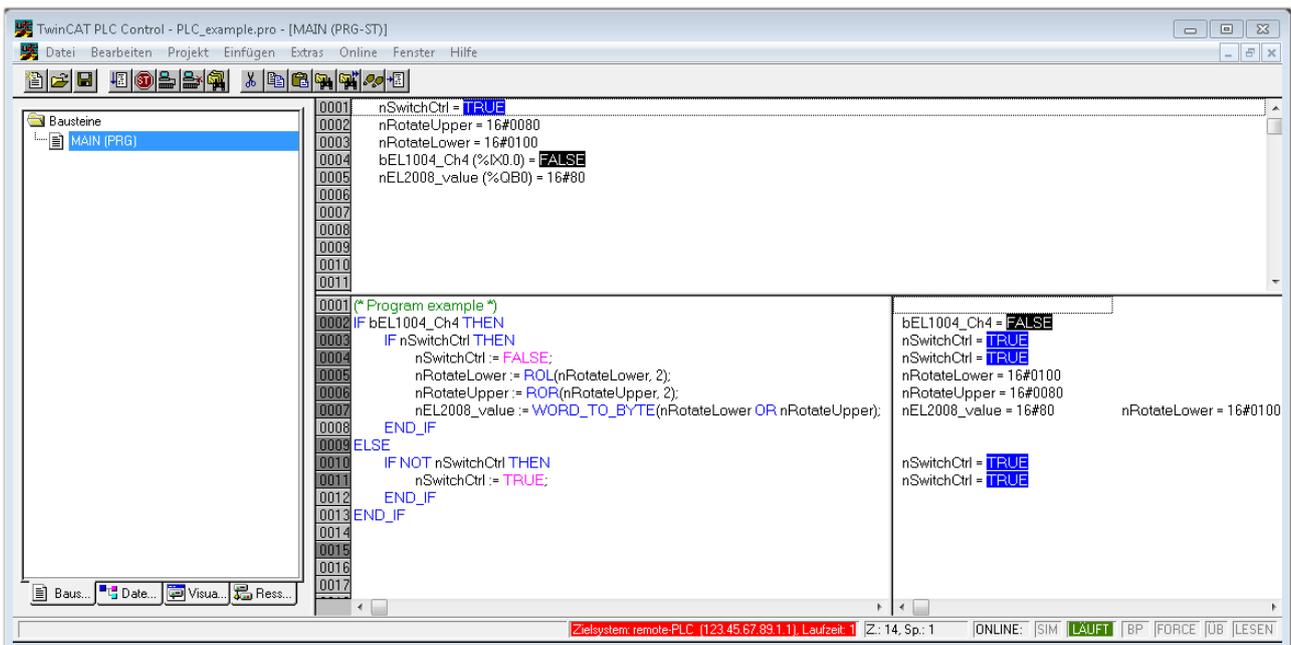


Abb. 37: PLC Control Logged-in, bereit zum Programmstart

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:

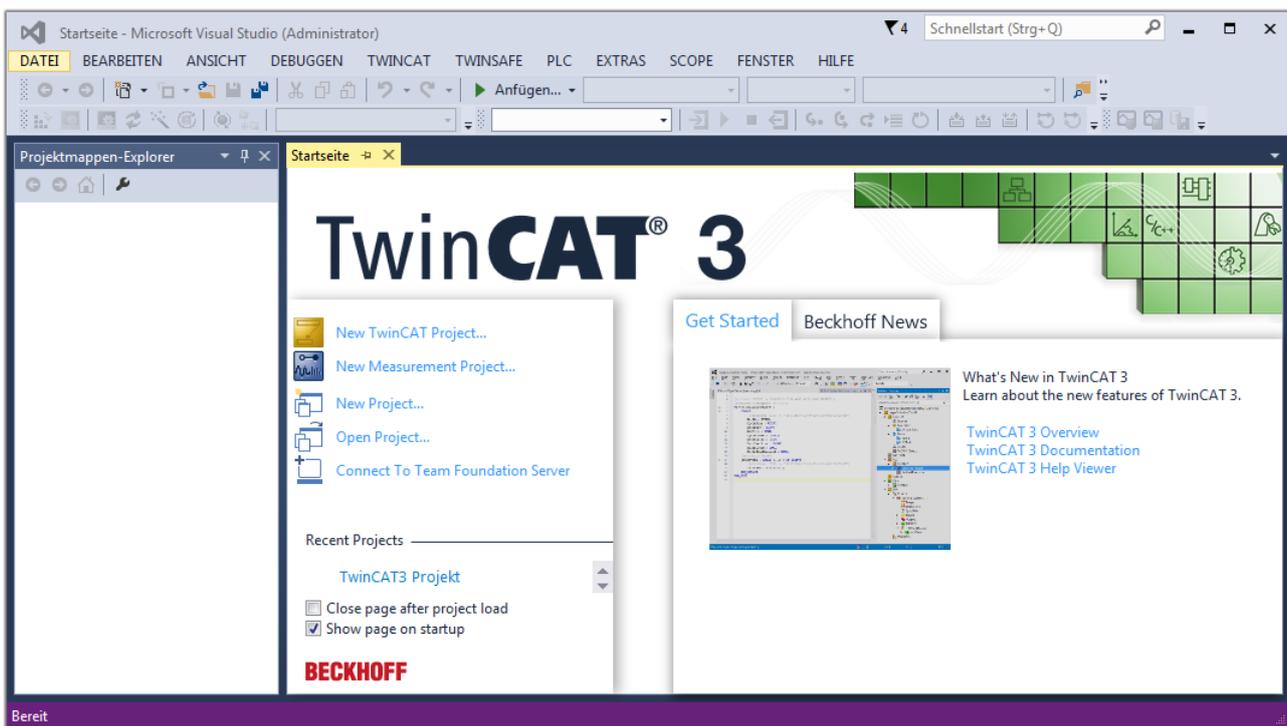


Abb. 38: Initiale Benutzeroberfläche TwinCAT 3

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):

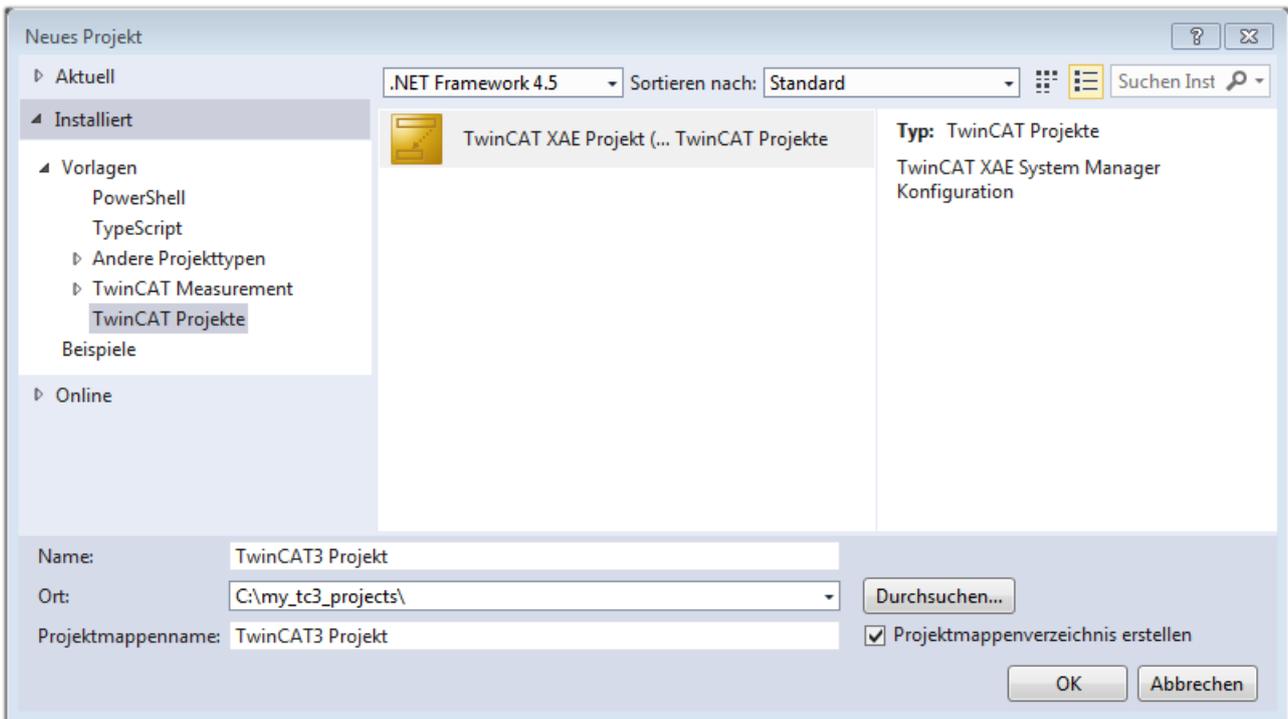


Abb. 39: Neues TwinCAT 3 Projekt erstellen

Im Projektmappen-Explorer liegt sodann das neue Projekt vor:

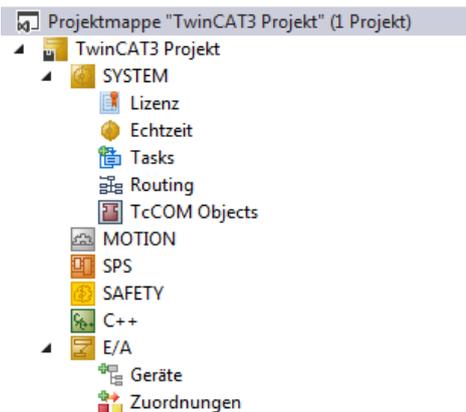
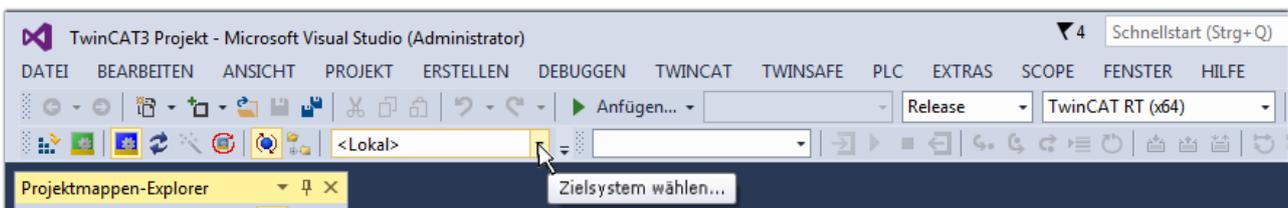


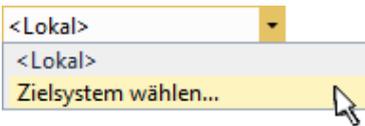
Abb. 40: Neues TwinCAT 3 Projekt im Projektmappen-Explorer

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 |> 61“ 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:

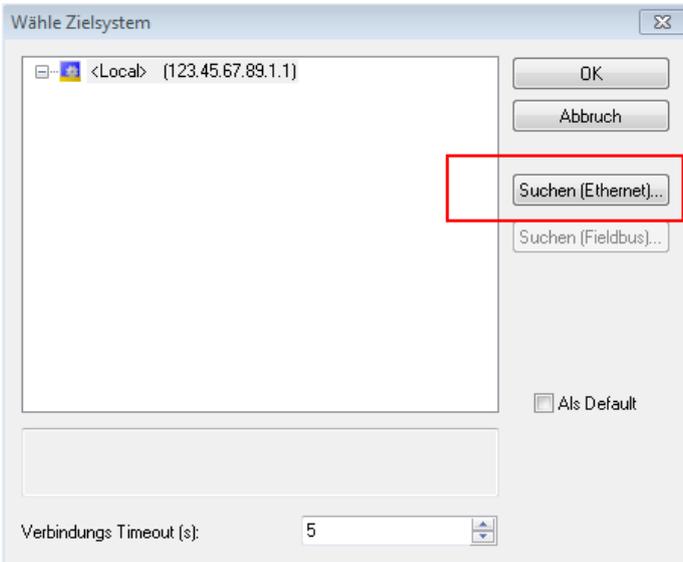
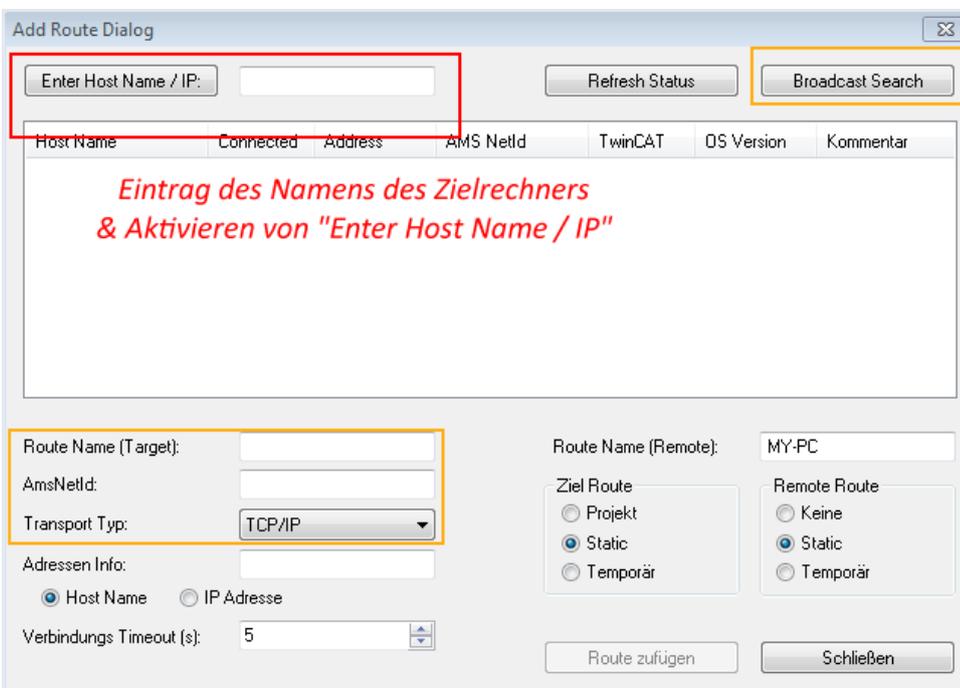


Abb. 41: Auswahldialog: Wähle Zielsystem

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

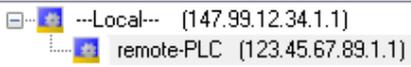
- den bekannten Rechnernamen hinter „Enter Host Name / IP:“ einzutragen (wie rot gekennzeichnet)
- einen „Broadcast Search“ durchzuführen (falls der Rechnernamen nicht genau bekannt)
- die bekannte Rechner - IP oder AmsNetId einzutragen



*Eintrag des Namens des Zielrechners
& Aktivieren von "Enter Host Name / IP"*

Abb. 42: 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.

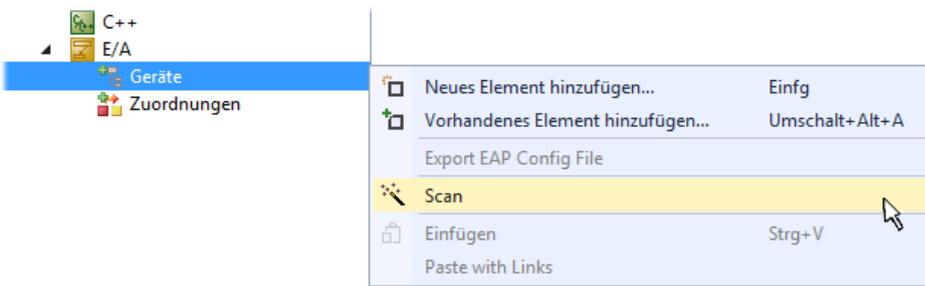


Abb. 43: Auswahl „Scan“

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

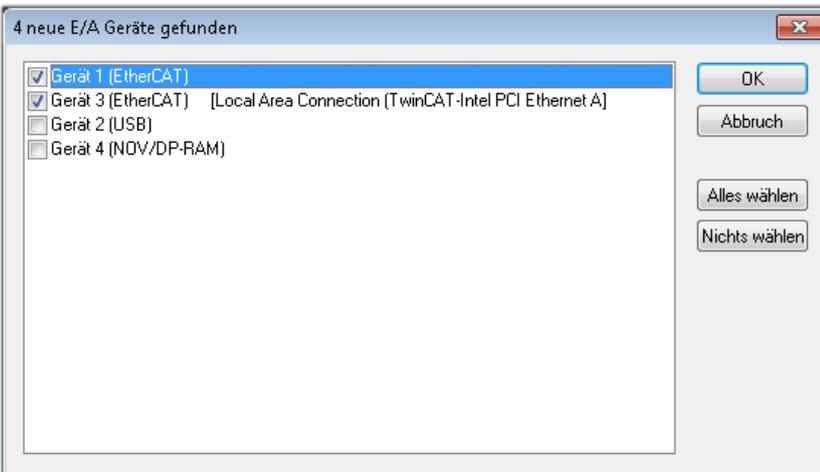


Abb. 44: Automatische Erkennung von E/A-Geräten: Auswahl der einzubindenden Geräte

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](#) [▶ 46] sieht das Ergebnis wie folgt aus:

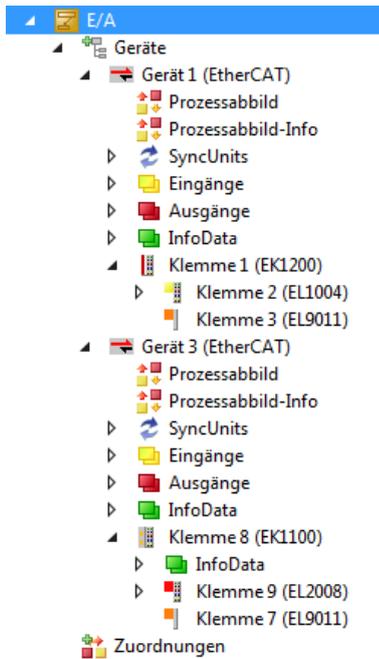


Abb. 45: Abbildung der Konfiguration in VS Shell der TwinCAT 3 Umgebung

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:

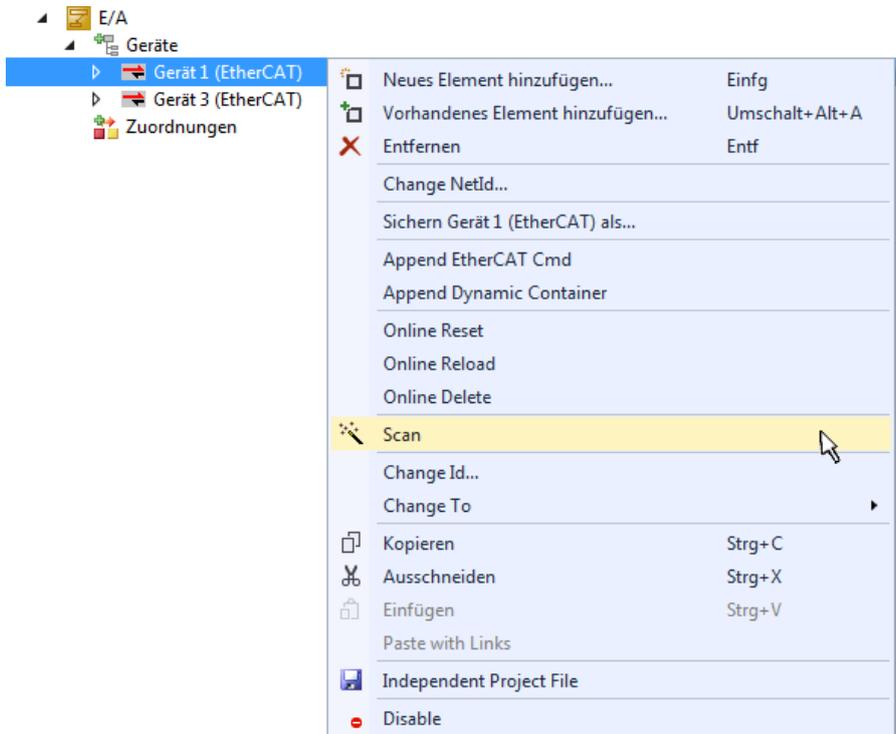


Abb. 46: Einlesen von einzelnen an einem Gerät befindlichen Klemmen

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:

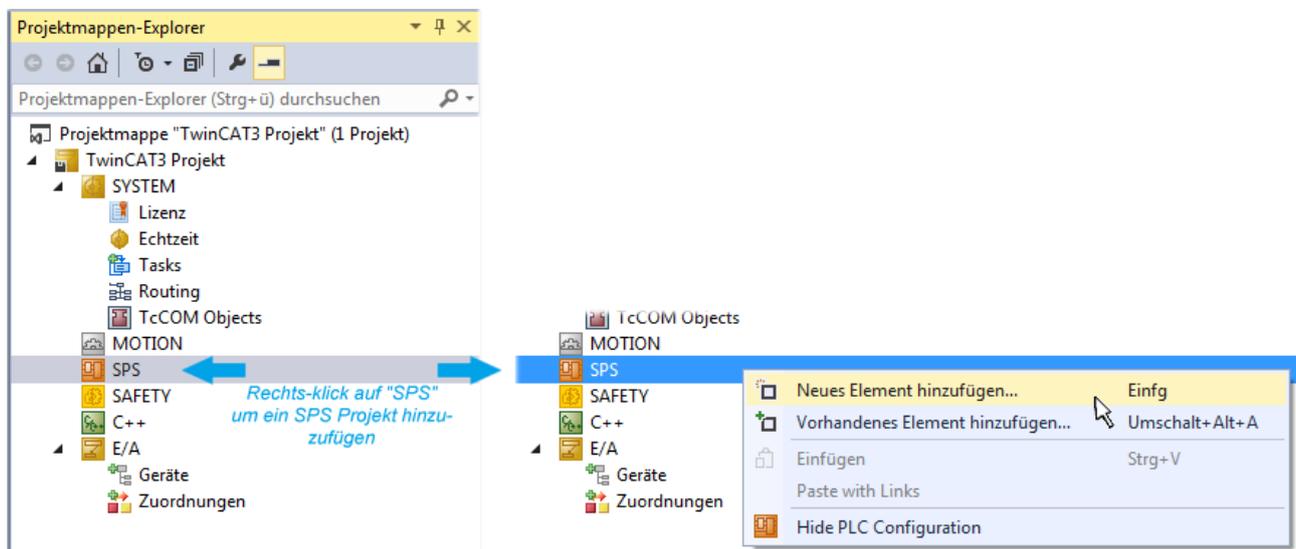


Abb. 47: Einfügen der Programmierumgebung in „SPS“

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

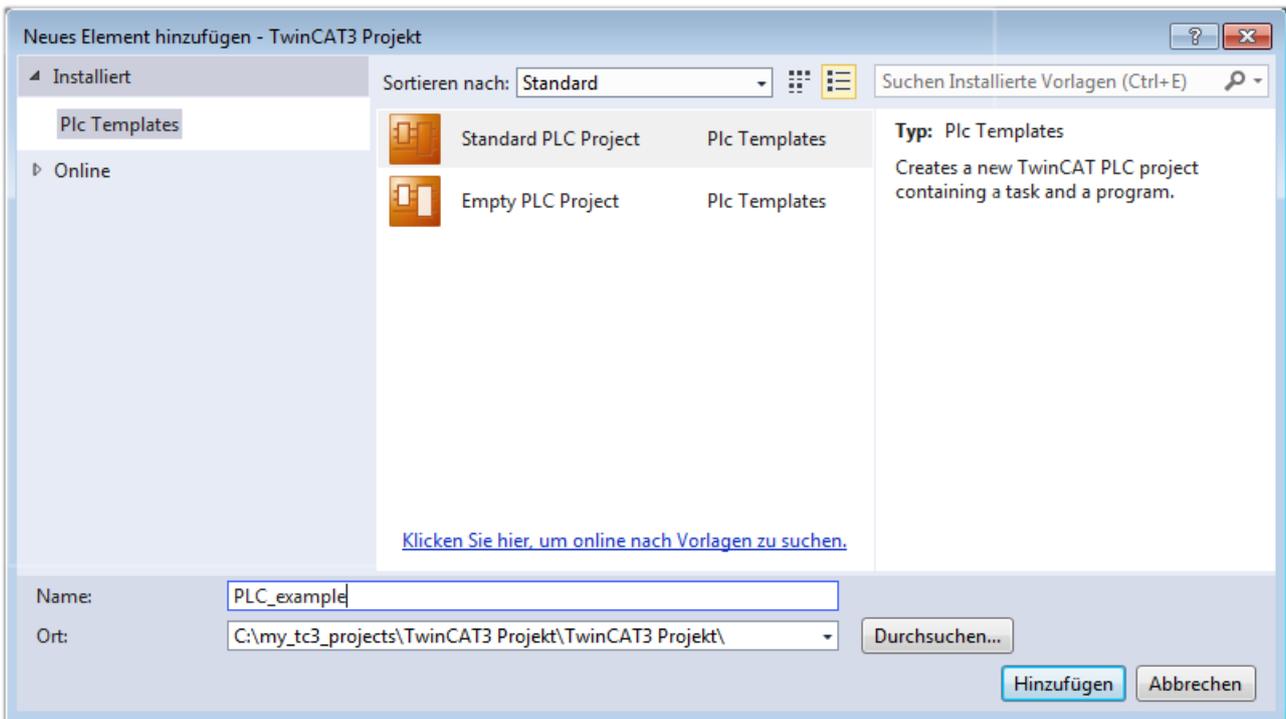


Abb. 48: Festlegen des Namens bzw. Verzeichnisses für die PLC Programmierungsumgebung

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:

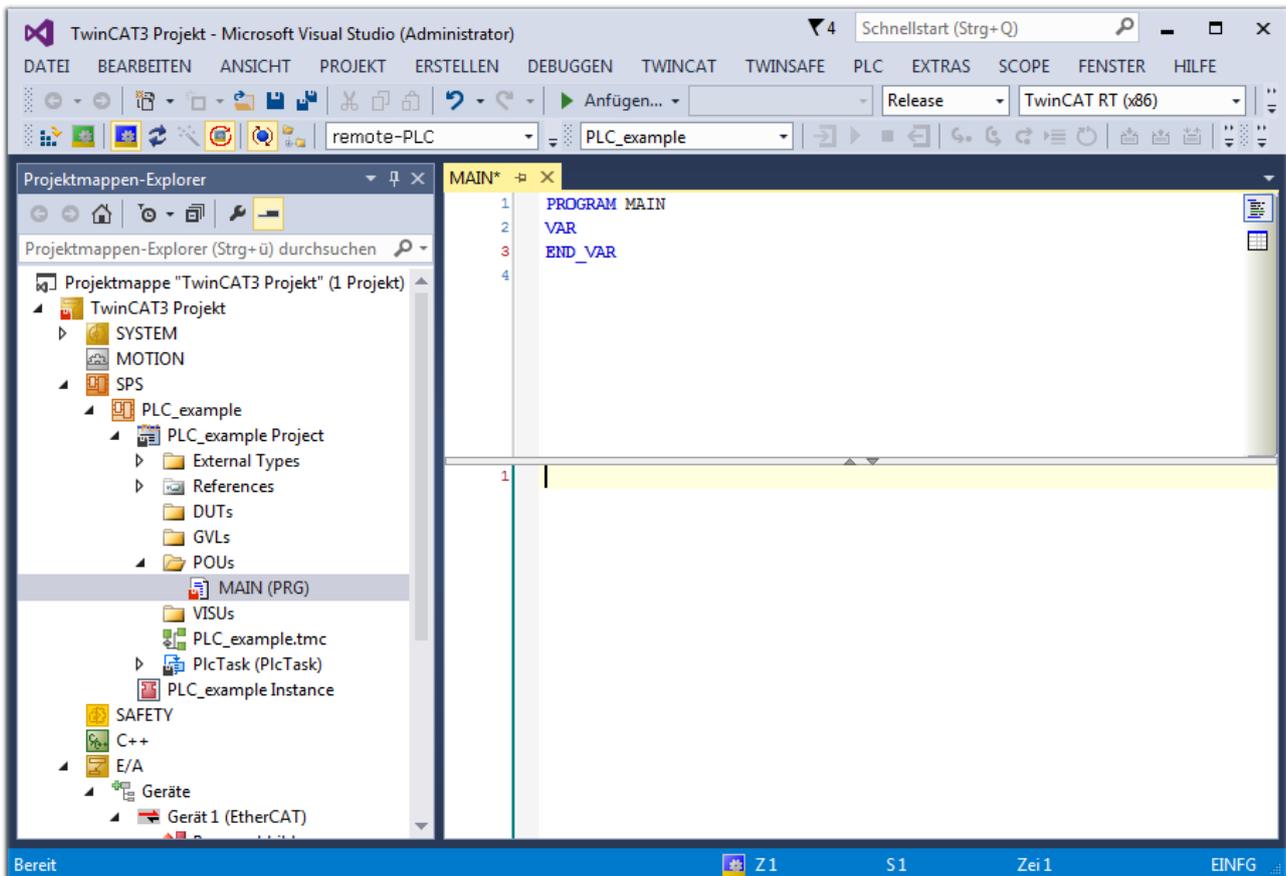


Abb. 49: Initiales Programm „Main“ des Standard PLC Projektes

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

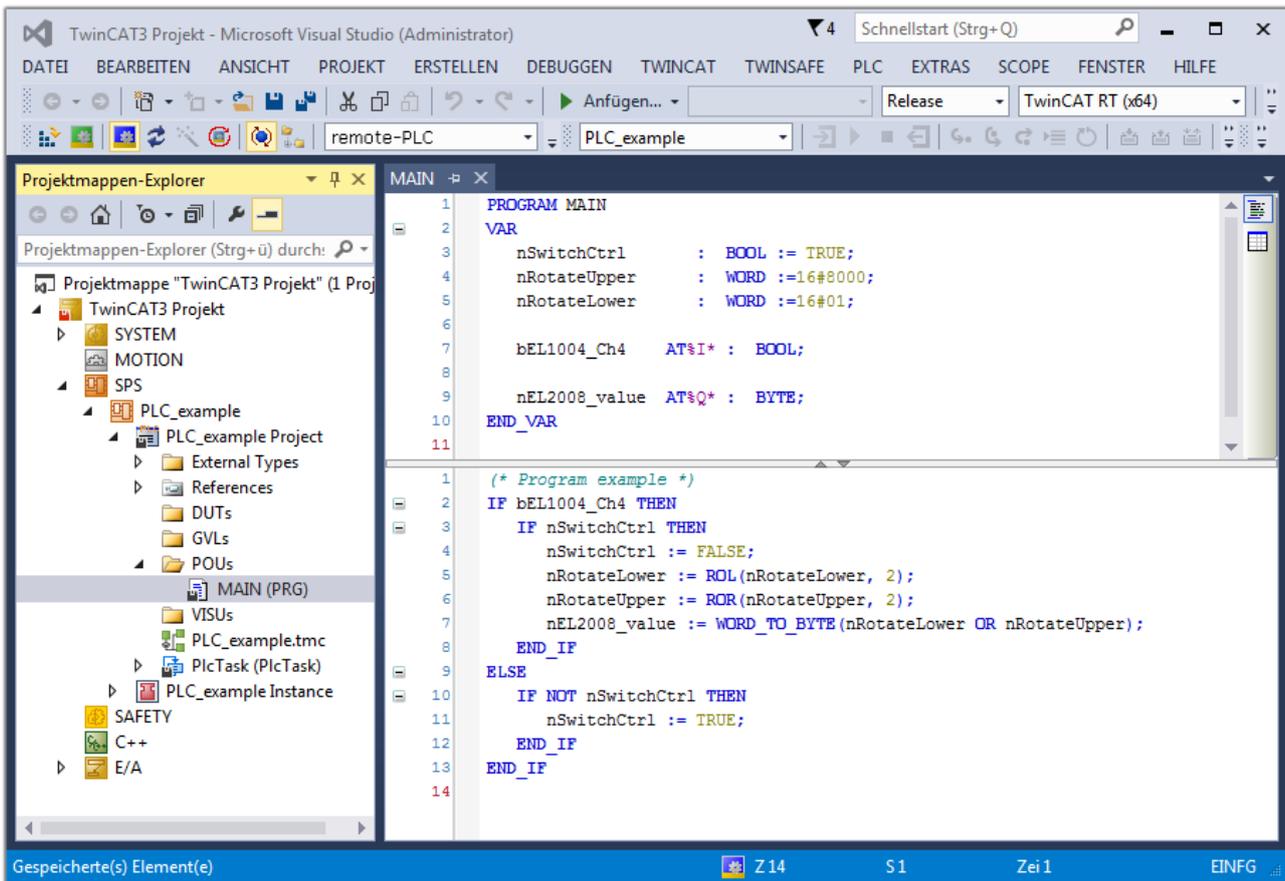


Abb. 50: Beispielprogramm mit Variablen nach einem Kompilervorgang (ohne Variablenanbindung)

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

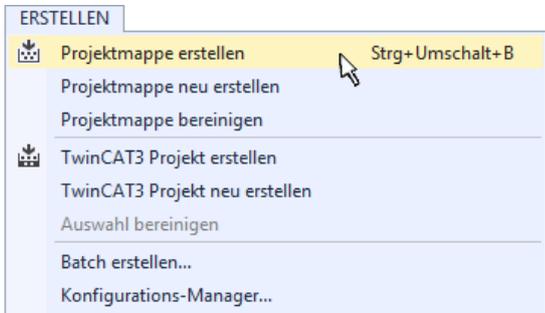
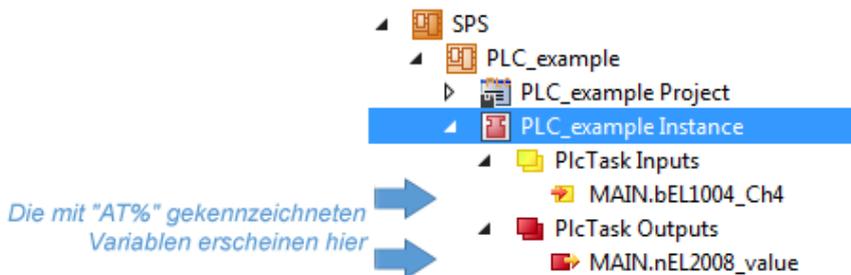


Abb. 51: Kompilierung des Programms starten

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:

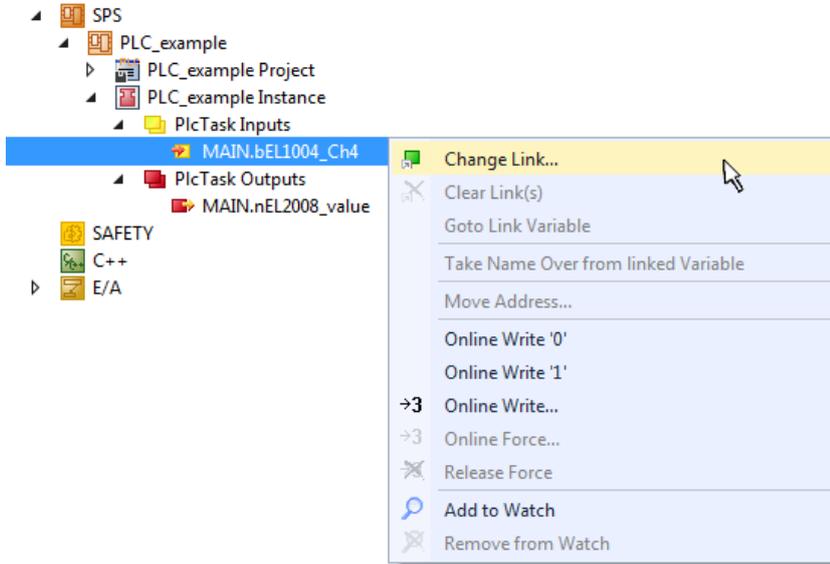


Abb. 52: Erstellen der Verknüpfungen PLC-Variablen zu Prozessobjekten

In dem dadurch geöffneten Fenster kann aus dem SPS-Konfigurationsbaum das Prozessobjekt für die Variable „bEL1004_Ch4“ vom Typ BOOL selektiert werden:

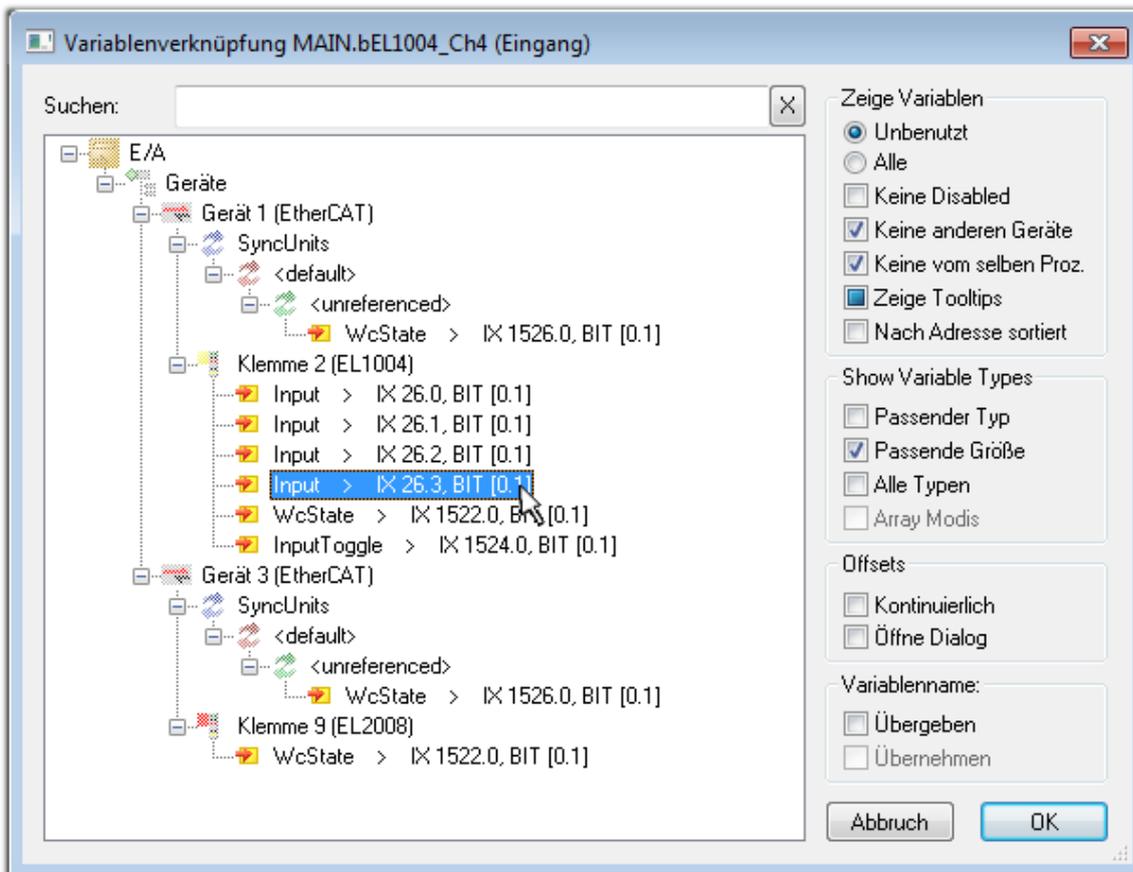


Abb. 53: Auswahl des PDO vom Typ BOOL

Entsprechend der Standardeinstellungen 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:

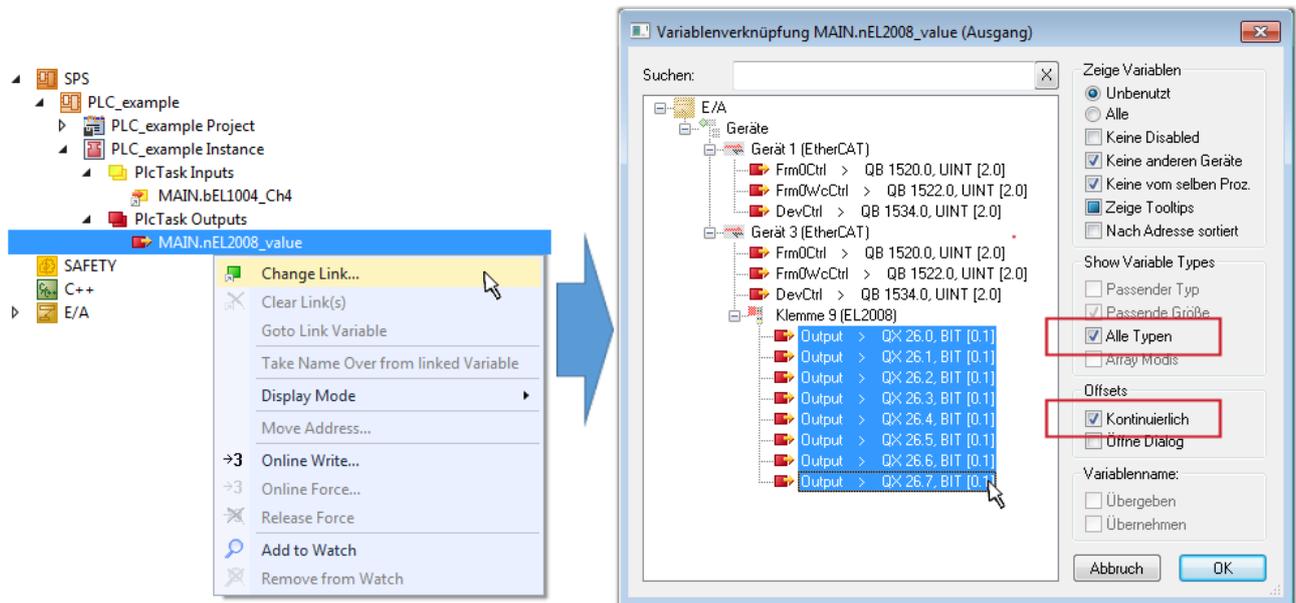


Abb. 54: Auswahl von mehreren PDO gleichzeitig: Aktivierung von „Kontinuierlich“ und „Alle Typen“

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:

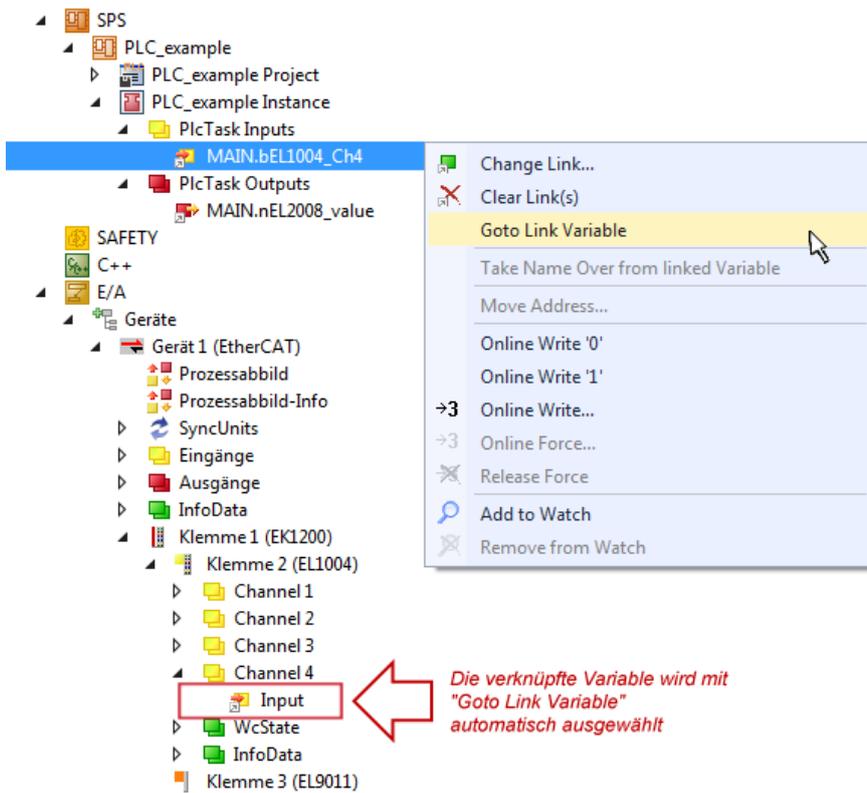


Abb. 55: Anwendung von "Goto Link Variable" am Beispiel von „MAIN.bEL1004_Ch4“

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 ein Byte, Word, Integer oder ein ähnliches PDO, so ist es möglich dies wiederum einen Satz von bit-typisierten Variablen 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.

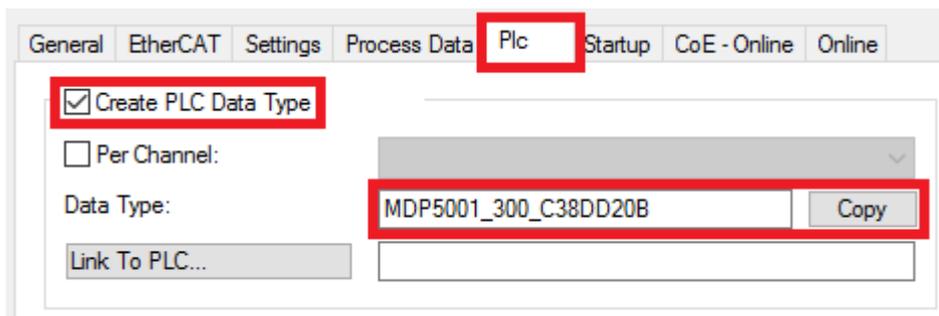


Abb. 56: Erzeugen eines SPS Datentyps

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

```

MAIN  ▸ ×
1  PROGRAM MAIN
2  VAR
3      EL3001 : MDP5001_300_C38DD20B;
4  END_VAR
    
```

Abb. 57: 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.

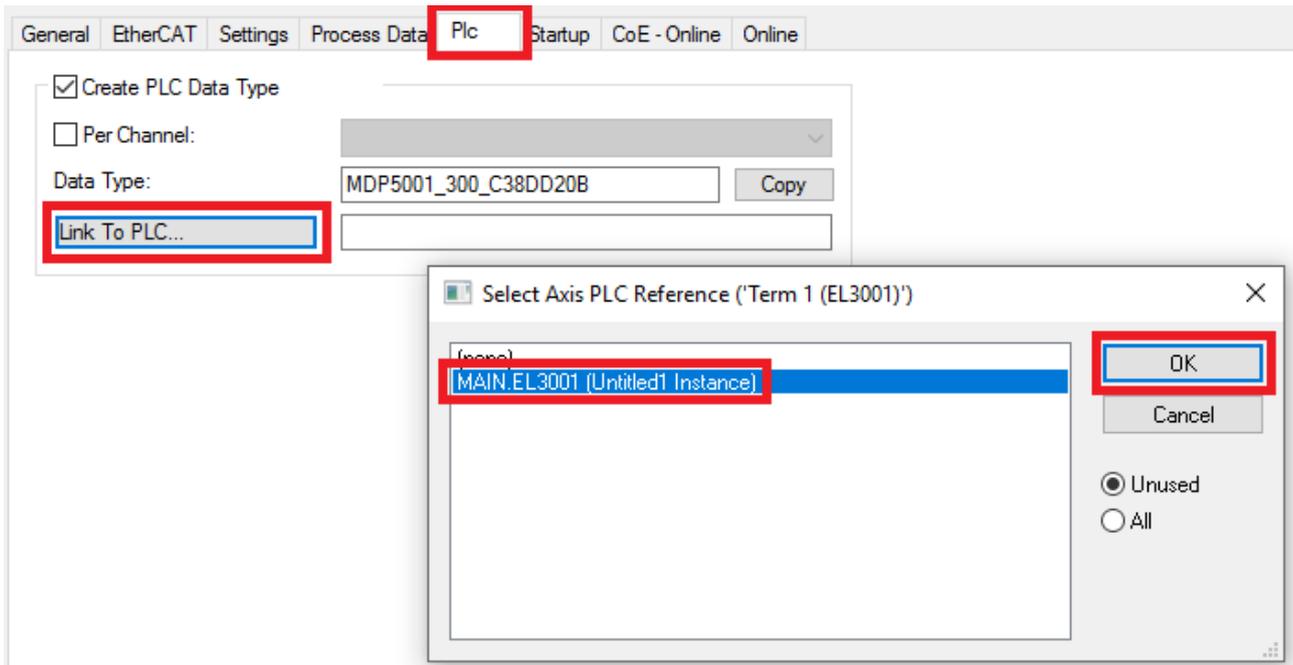


Abb. 58: Verknüpfung der Struktur

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

```

MAIN* ▸ ×
1  PROGRAM MAIN
2  VAR
3      EL3001 : MDP5001_300_C38DD20B;
4
5      nVoltage: INT;
6  END_VAR
    
```

```

1  nVoltage := EL3001.MDP5001_300_Input.
    
```

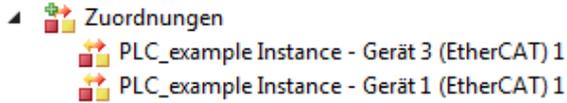
A tooltip is visible over the code, showing the structure members: 'MDP5001_300_AI_Standard_Status' and 'MDP5001_300_AI_Standard_Value'.

Abb. 59: 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:



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:

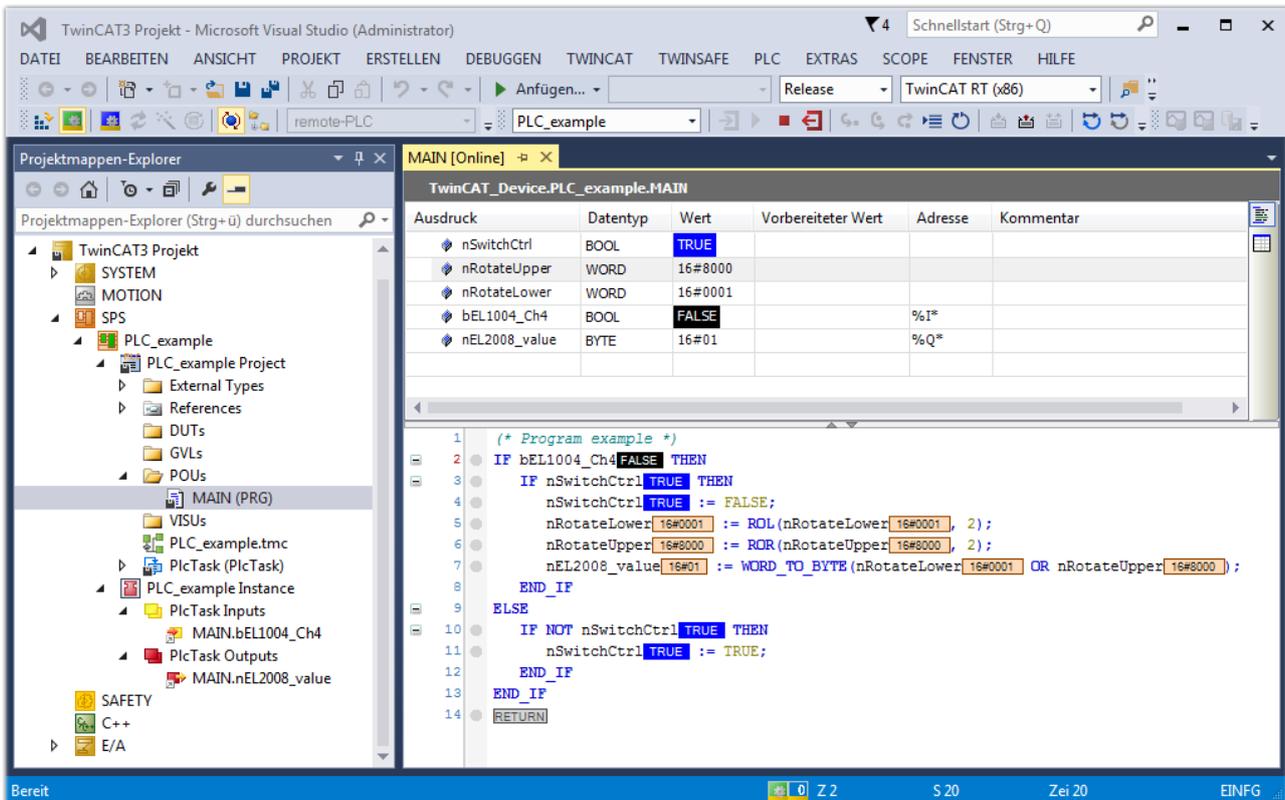


Abb. 60: TwinCAT 3 Entwicklungsumgebung (VS Shell): Logged-in, nach erfolgten Programmstart

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 der 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.

A: Über den TwinCAT Adapter-Dialog

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



Abb. 61: Aufruf im System Manager (TwinCAT 2)

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

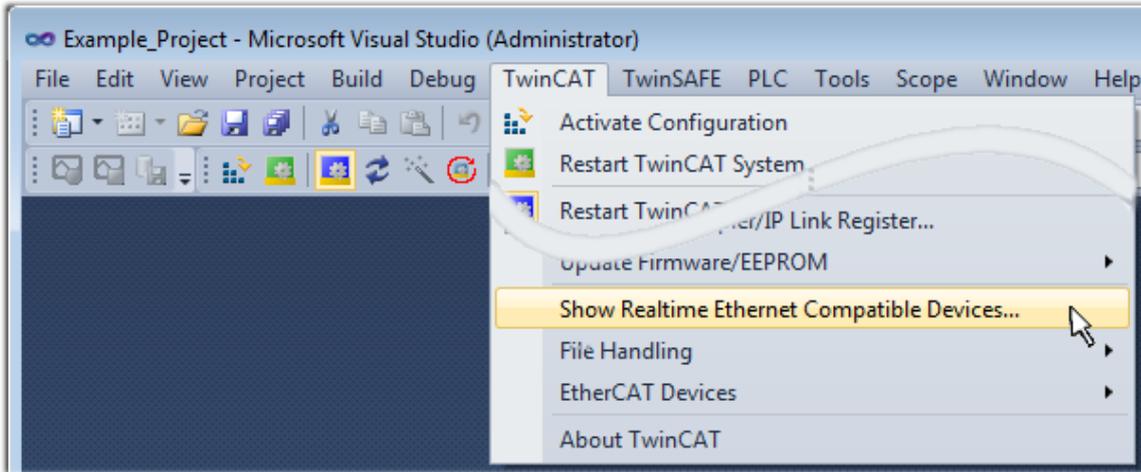


Abb. 62: Aufruf in VS Shell (TwinCAT 3)

B: Über TcRtelInstall.exe im TwinCAT-Verzeichnis

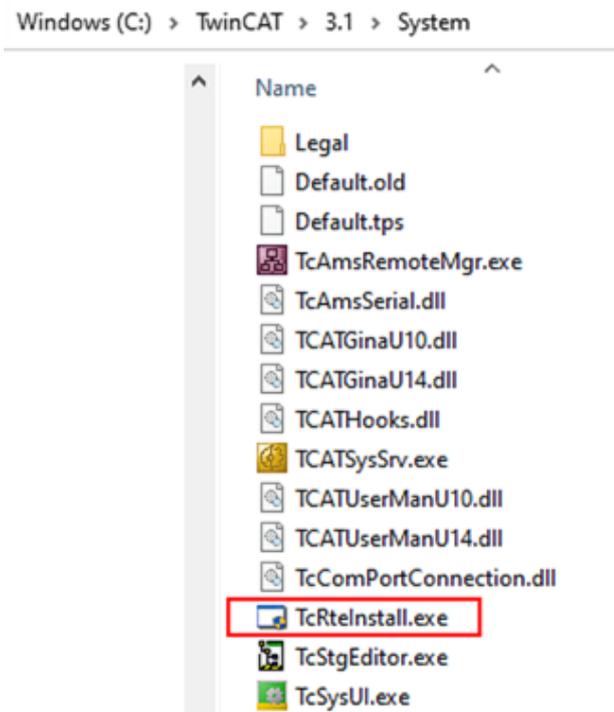


Abb. 63: TcRtelInstall.exe im TwinCAT-Verzeichnis

In beiden Fällen erscheint der folgende Dialog:

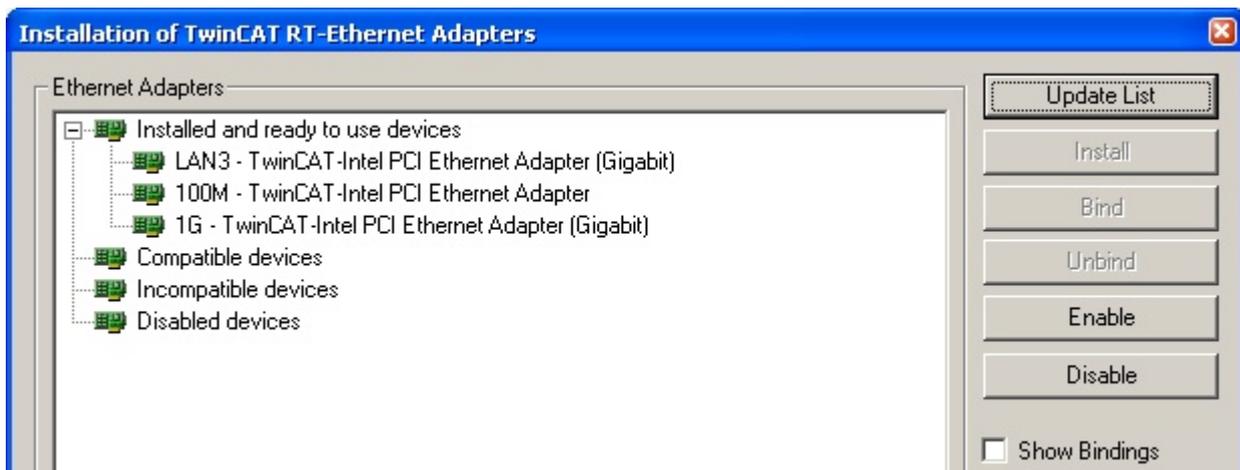


Abb. 64: Übersicht Netzwerkschnittstellen

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“* [► 82] beschrieben, zunächst ein EtherCAT-Gerät eingetragen werden, um dann über dessen Eigenschaften (Karteireiter „Adapter“, Button „Kompatible Geräte...“) die kompatiblen Ethernet Ports einzusehen:

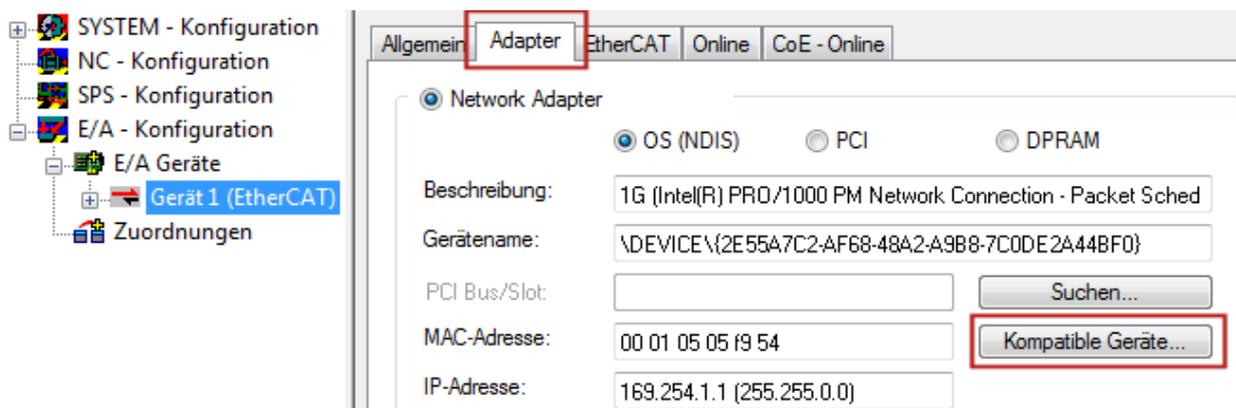
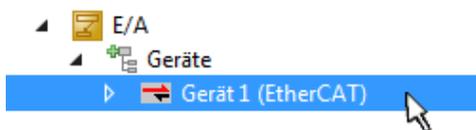


Abb. 65: Eigenschaft von EtherCAT-Gerät (TwinCAT 2): Klick auf „Kompatible Geräte...“ von „Adapter“

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)

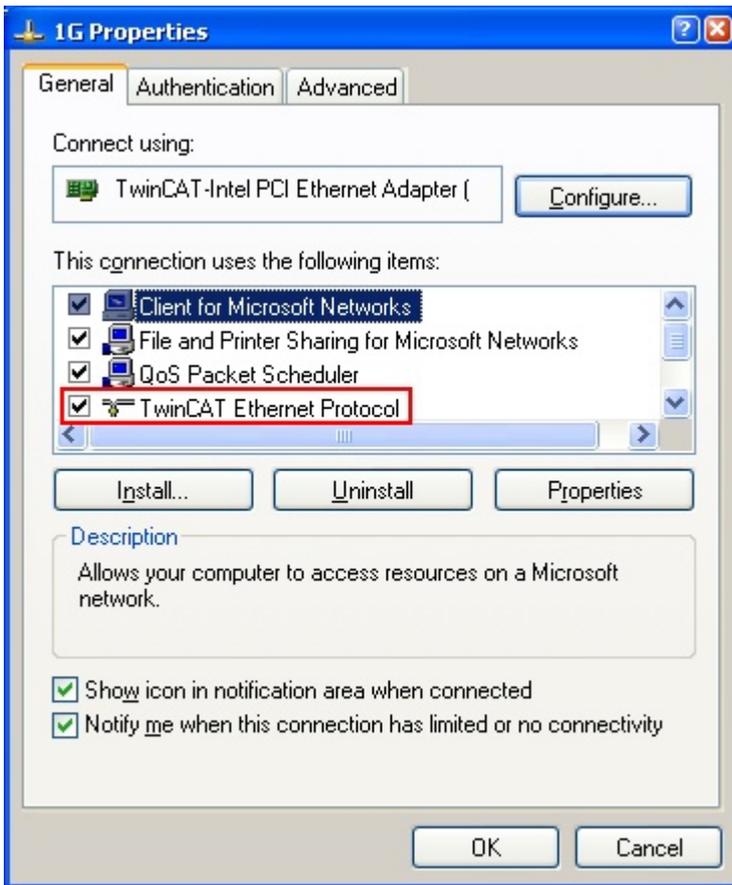


Abb. 66: Windows-Eigenschaften der Netzwerkschnittstelle

Eine korrekte Einstellung des Treibers könnte wie folgt aussehen:

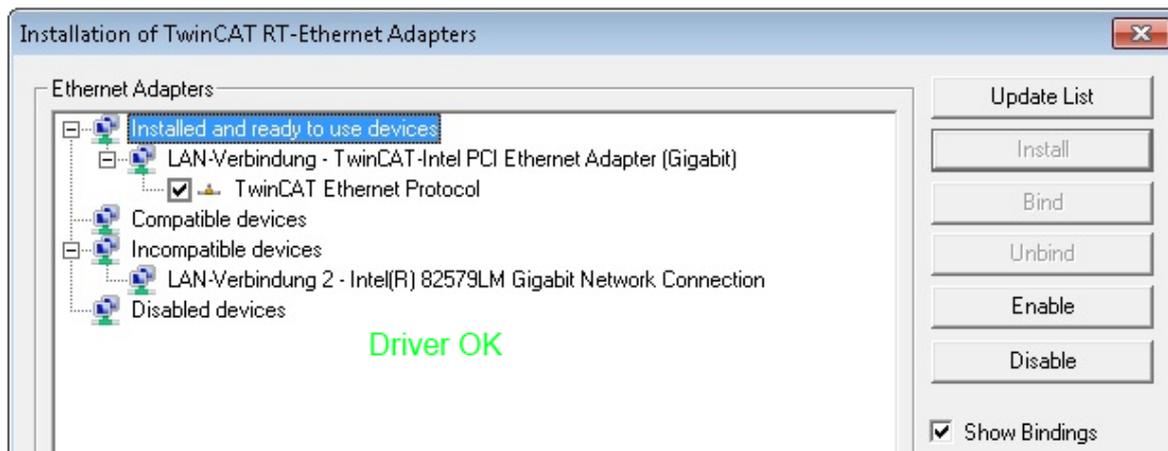


Abb. 67: Beispielhafte korrekte Treiber-Einstellung des Ethernet Ports

Andere mögliche Einstellungen sind zu vermeiden:

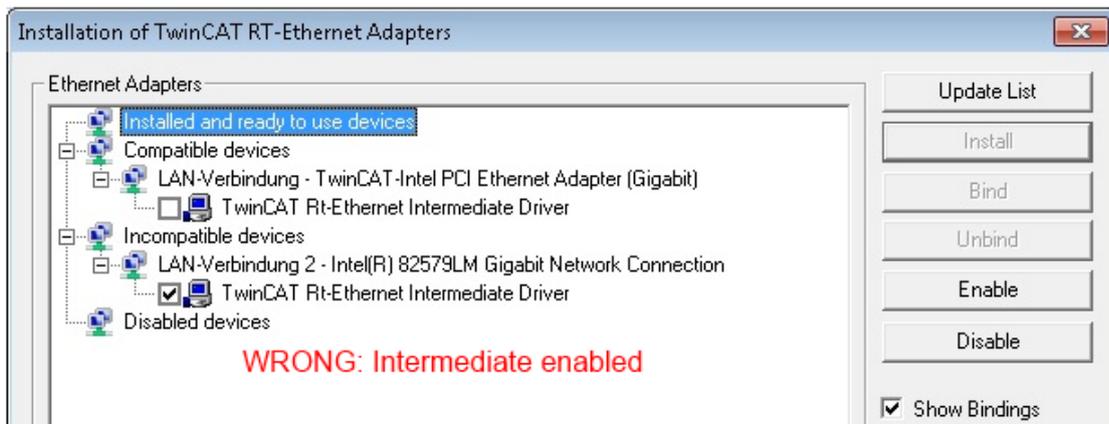
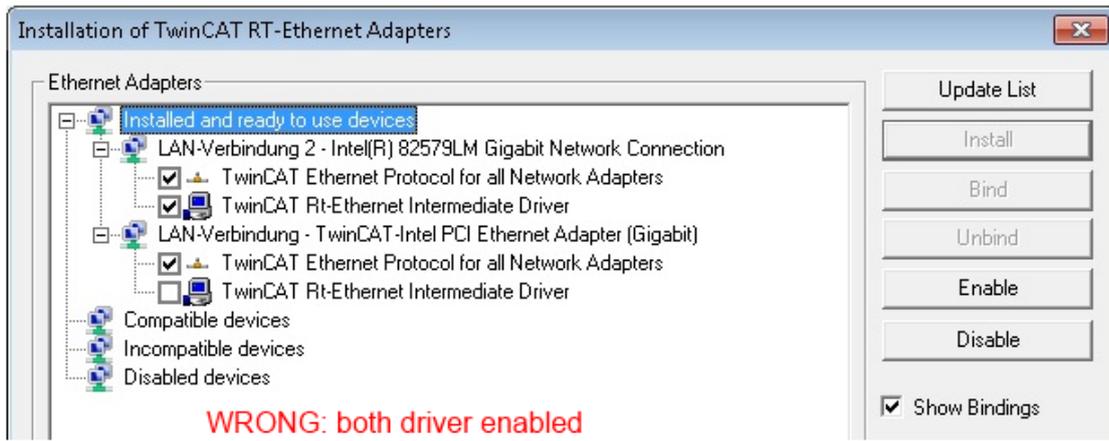


Abb. 68: Fehlerhafte Treiber-Einstellungen des Ethernet Ports

IP-Adresse des verwendeten Ports

● IP-Adresse/DHCP

i 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.

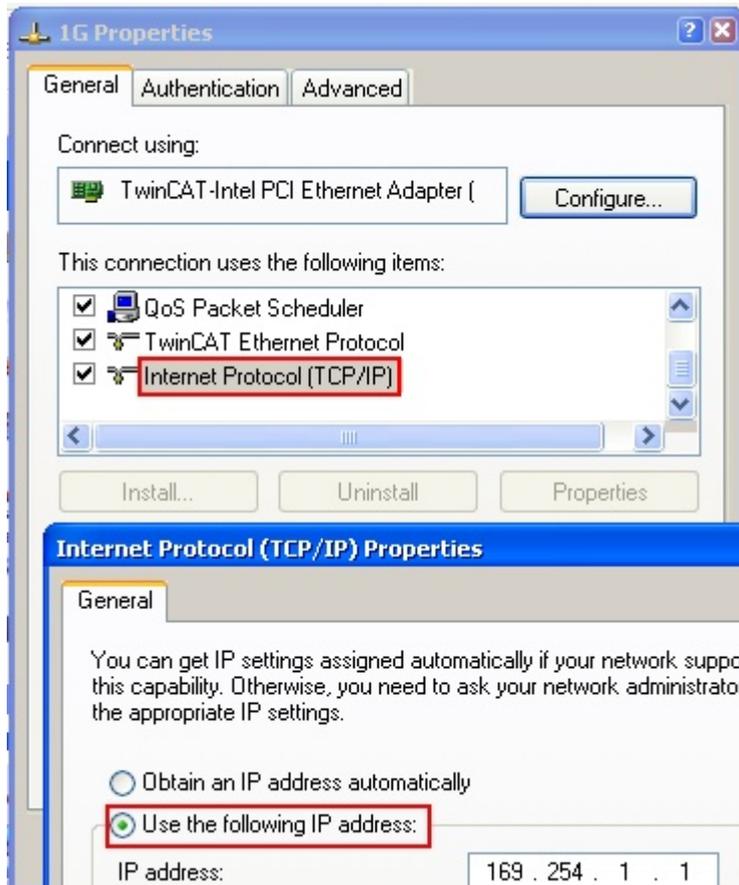


Abb. 69: TCP/IP-Einstellung des Ethernet Ports

5.2.2 Hinweise zur 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 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“

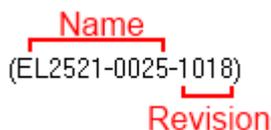


Abb. 70: Gerätebezeichnung: Struktur

Die Bestellbezeichnung aus Typ + Version (hier: EL2521-0025) 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.

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.

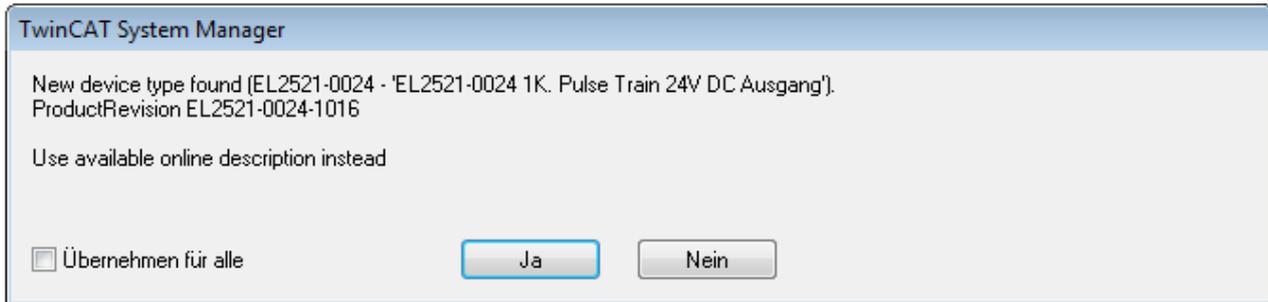


Abb. 71: Hinweisfenster OnlineDescription (TwinCAT 2)

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

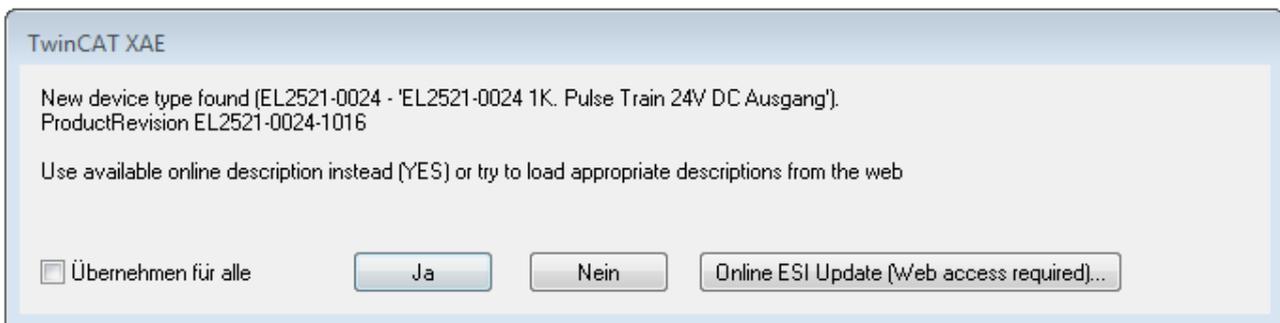


Abb. 72: Hinweisfenster OnlineDescription (TwinCAT 3)

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 \[► 82\]](#)“.

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. 73: 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. 74: 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.

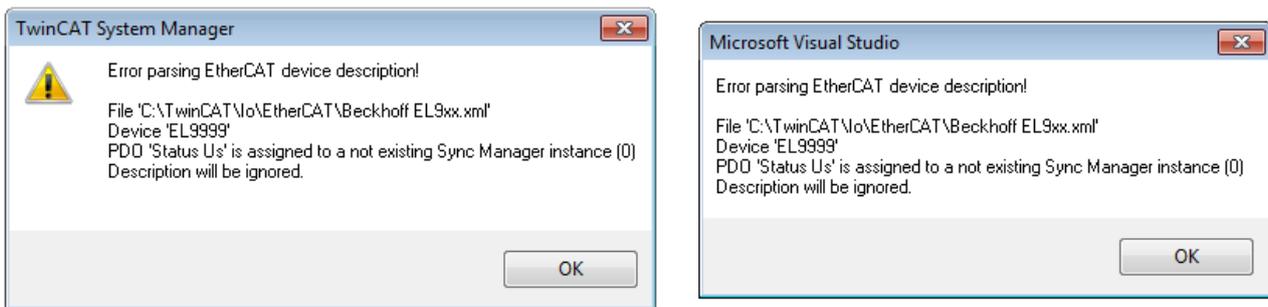


Abb. 75: Hinweisfenster fehlerhafte ESI-Datei (links: TwinCAT 2; rechts: TwinCAT 3)

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 OFFLINE Konfigurationserstellung

Anlegen des Geräts EtherCAT

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

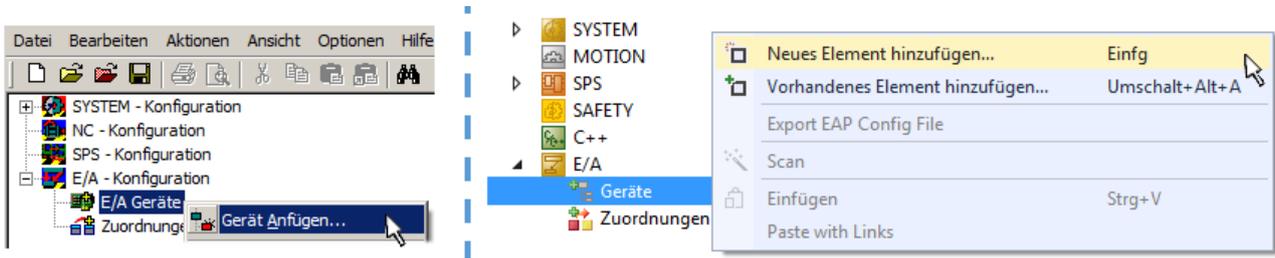


Abb. 76: Anfügen eines EtherCAT Device: links TwinCAT 2; rechts TwinCAT 3

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.

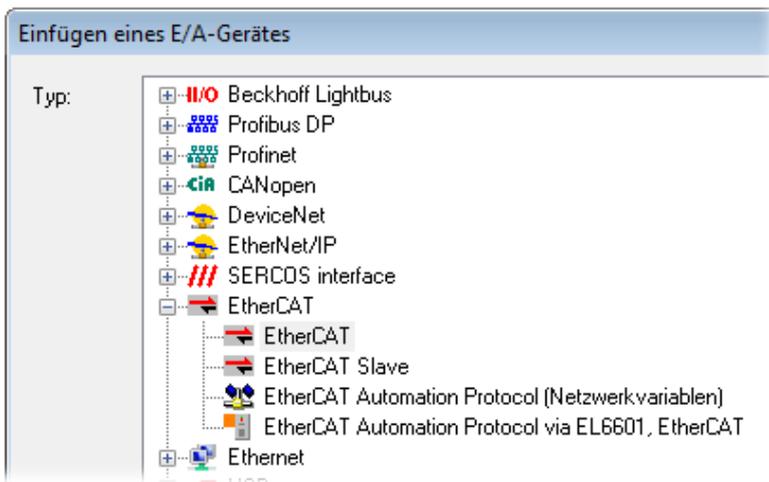


Abb. 77: Auswahl EtherCAT Anschluss (TwinCAT 2.11, TwinCAT 3)

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

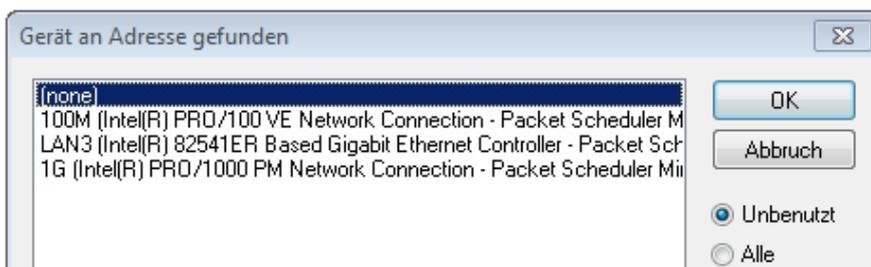


Abb. 78: Auswahl Ethernet Port

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)“.

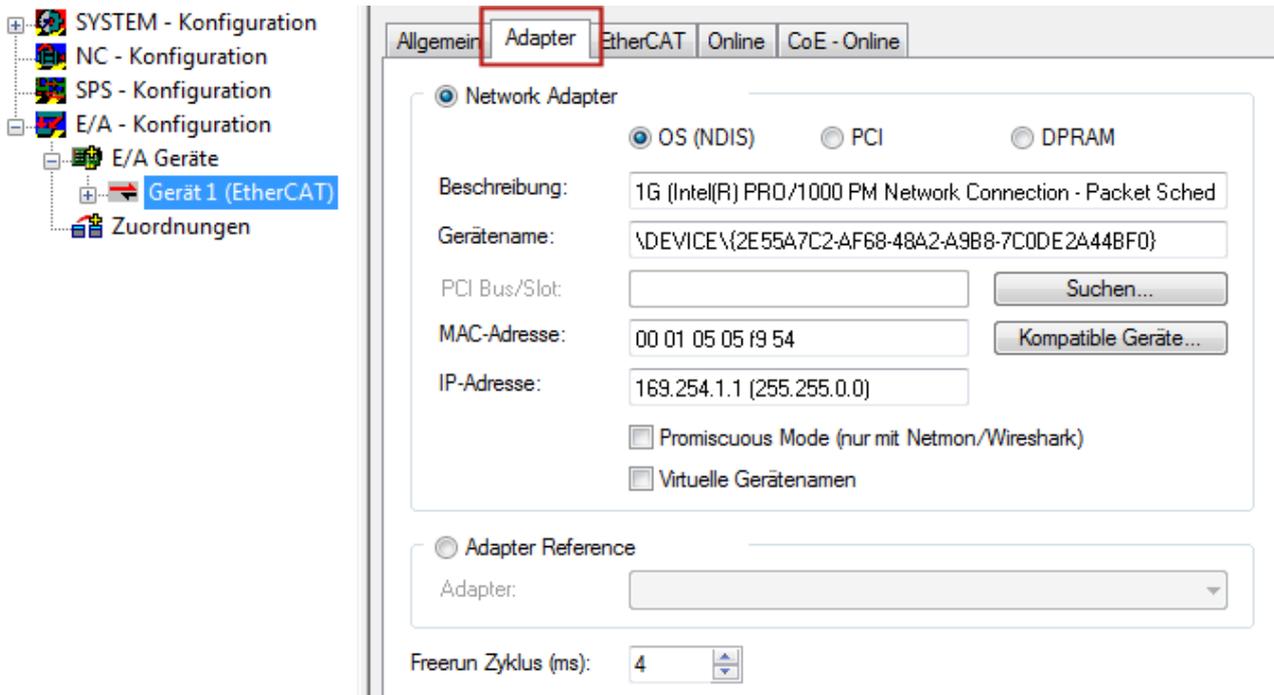


Abb. 79: Eigenschaften EtherCAT-Gerät (TwinCAT 2)

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



i Auswahl des Ethernet-Ports

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](#) [72].

Definieren von EtherCAT-Slaves

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

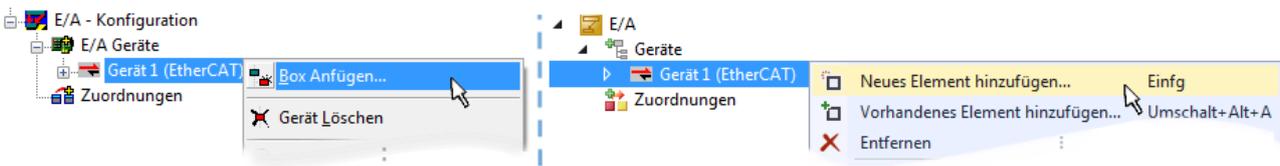


Abb. 80: Anfügen von EtherCAT-Geräten (links: TwinCAT 2; rechts: TwinCAT 3)

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 Fast-Ethernet-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: Koppler, Box-Module, Geräte mit RJ45/M8/M12-Anschluss

- „E-Bus“: LVDS „Klemmenbus“, EtherCAT-Steckmodule (EJ), EtherCAT-Klemmen (EL/ES), diverse anreihbare Module

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

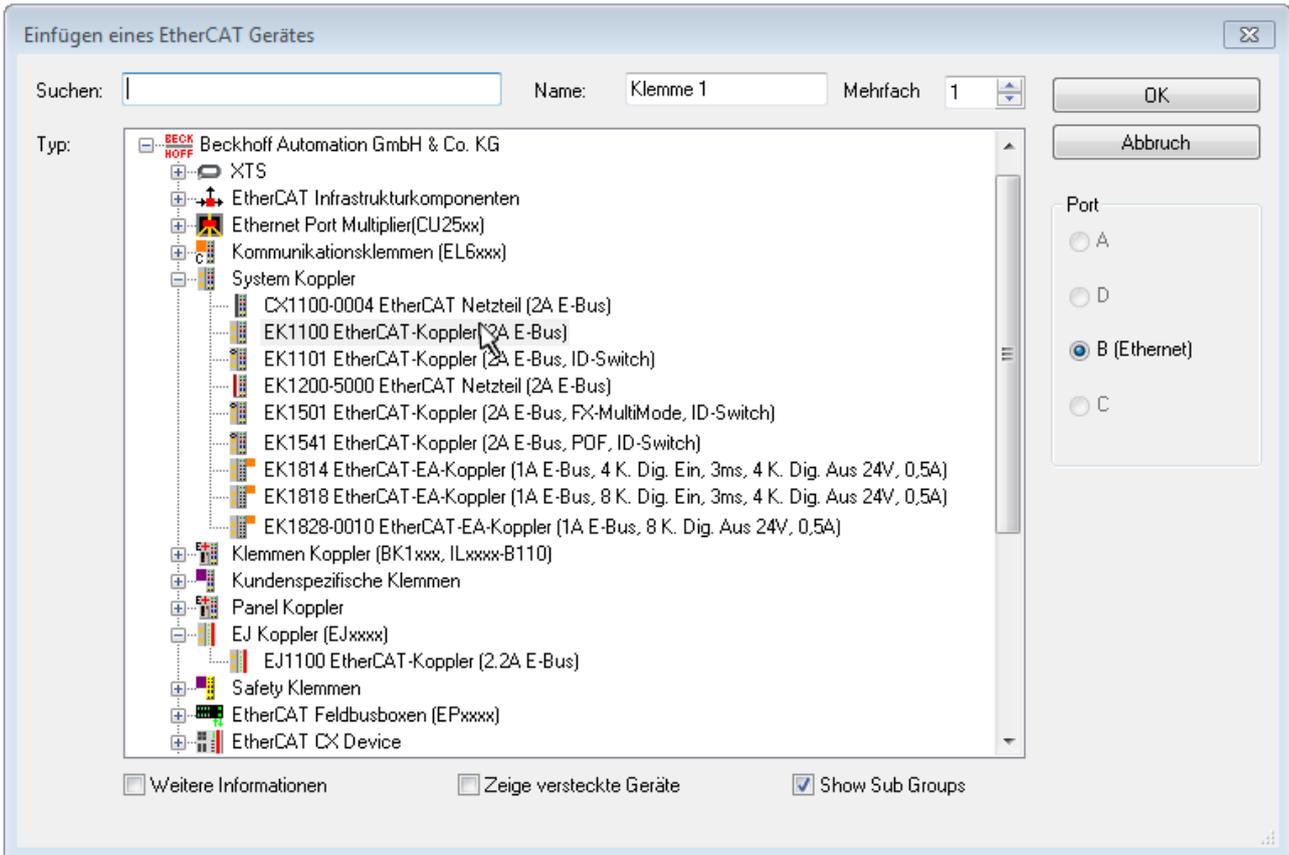


Abb. 81: Auswahldialog neues EtherCAT-Gerät

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.

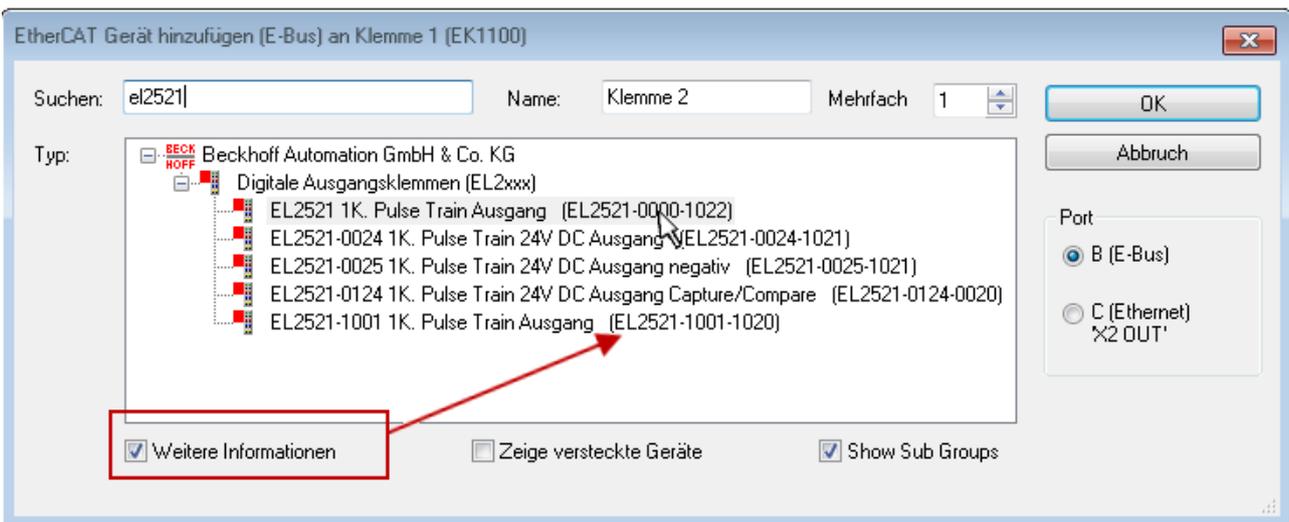


Abb. 82: Anzeige Geräte-Revision

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“.

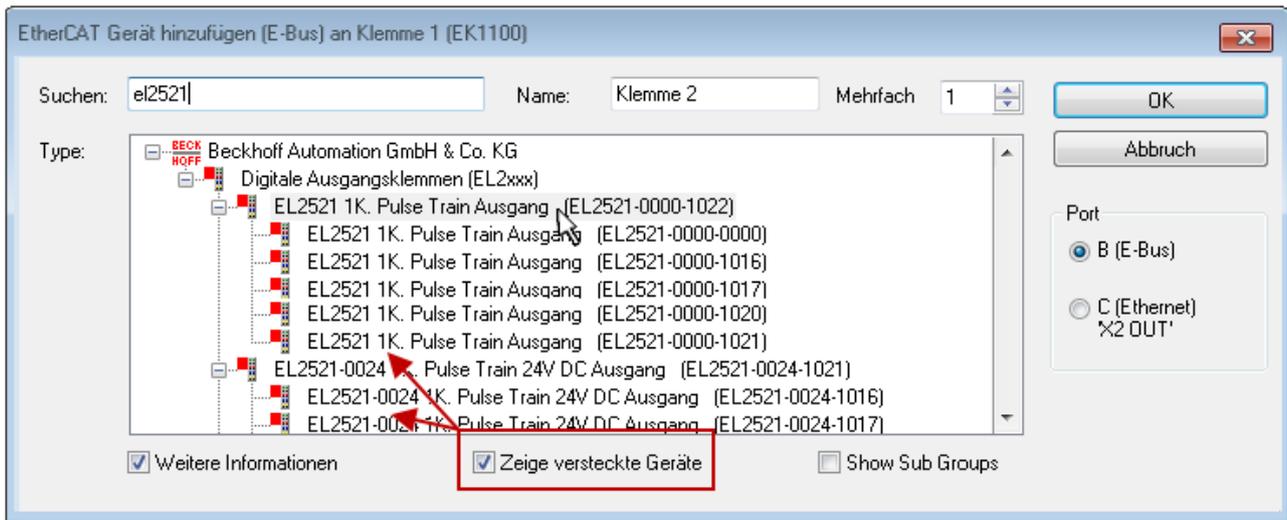


Abb. 83: Anzeige vorhergehender Revisionen

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.

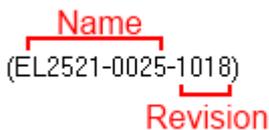


Abb. 84: Name/Revision Klemme

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, ...

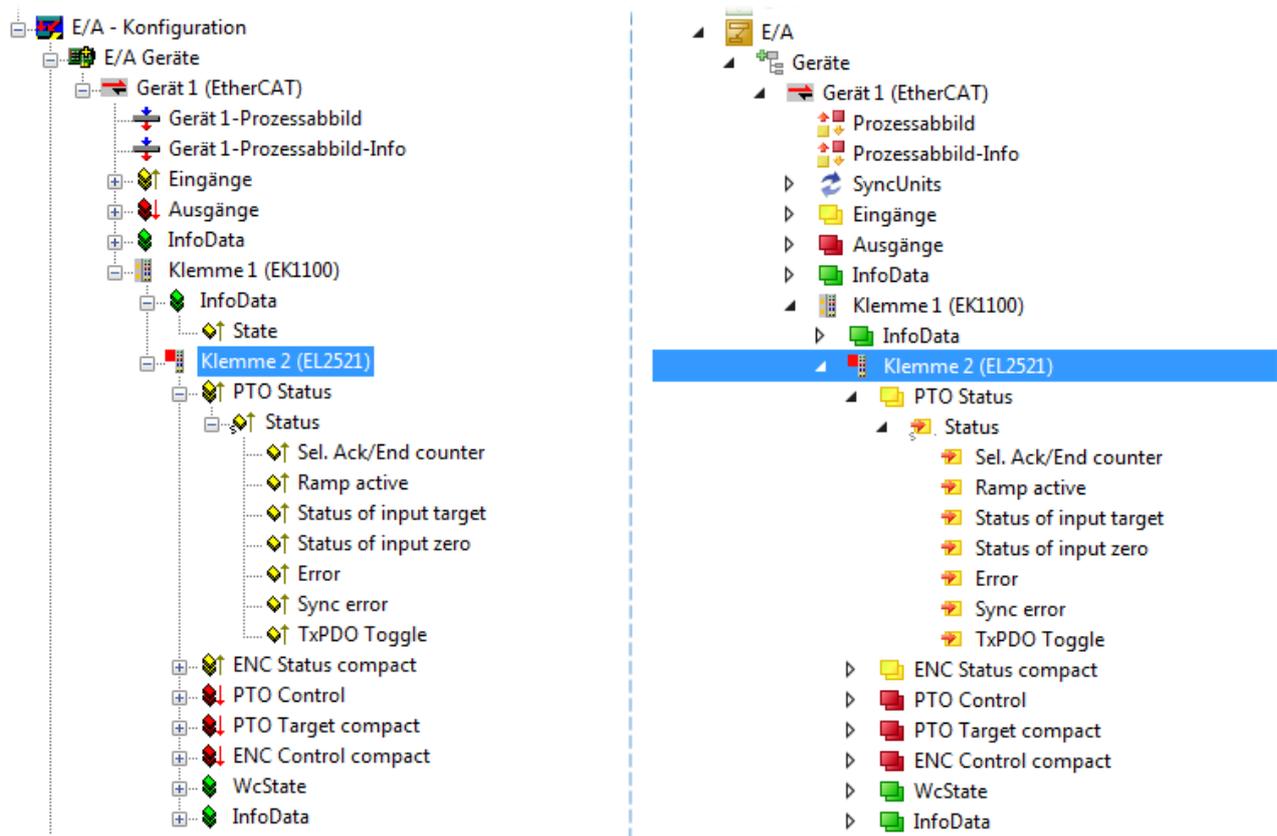


Abb. 85: EtherCAT Klemme im TwinCAT-Baum (links: TwinCAT 2; rechts: TwinCAT 3)

5.2.4 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 Config-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. 86: 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.

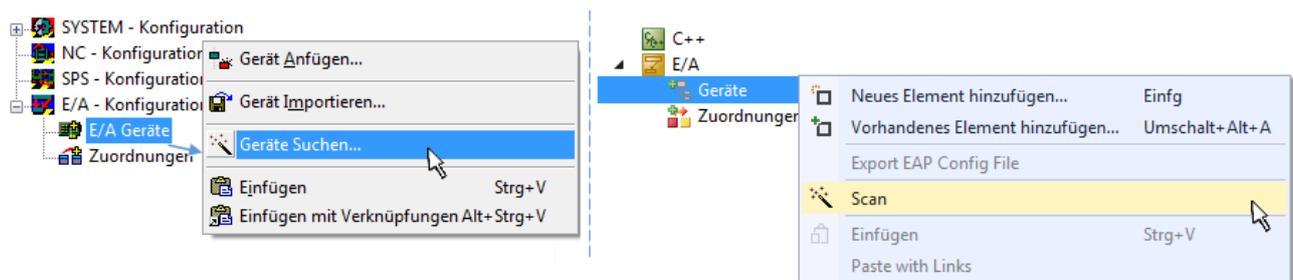


Abb. 87: Scan Devices (links: TwinCAT 2; rechts: TwinCAT 3)

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.

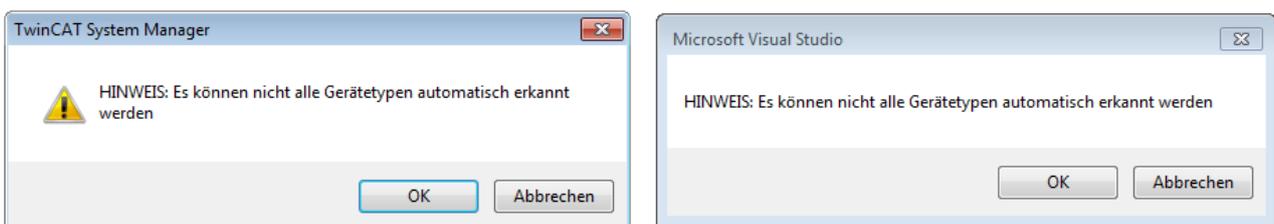


Abb. 88: Hinweis automatischer GeräteScan (links: TwinCAT 2; rechts: TwinCAT 3)

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.

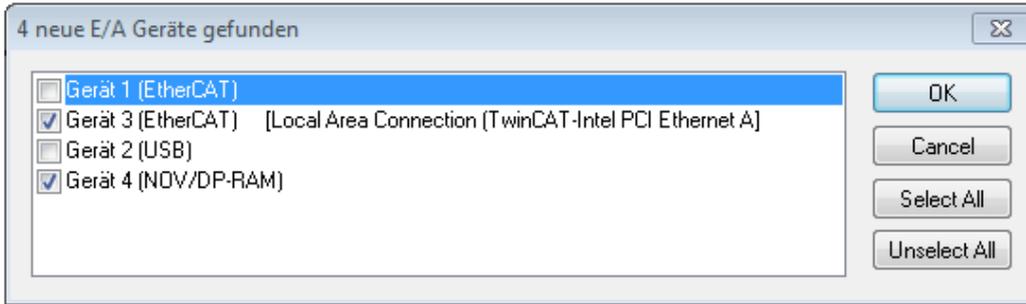


Abb. 89: Erkannte Ethernet-Geräte

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 des Ethernet-Ports

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](#) [► 72].

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.

Name
(EL2521-0025-1018)
Revision

Abb. 90: 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 Serienmaschinebau bzw. bei Reproduktion der Anlage sollte die Funktion aber nicht mehr zur Konfigurationserstellung verwendet werden sondern ggf. zum [Vergleich](#) [► 92] 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:

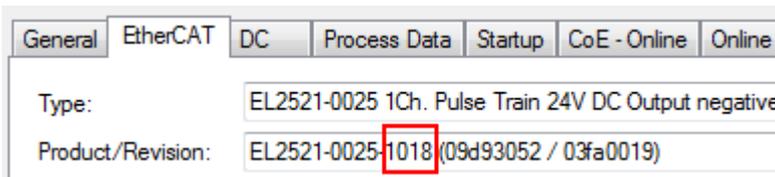


Abb. 91: Einbau EtherCAT-Klemme mit Revision -1018

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 [► 92] 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:

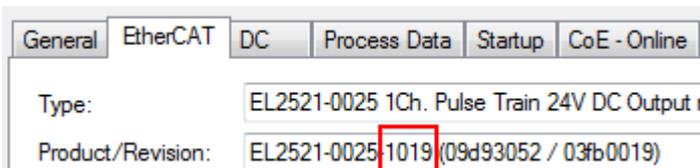


Abb. 92: Erkennen EtherCAT-Klemme mit Revision -1019

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. 93: Scan-Abfrage nach automatischem Anlegen eines EtherCAT-Gerätes (links: TC2; rechts TC3)

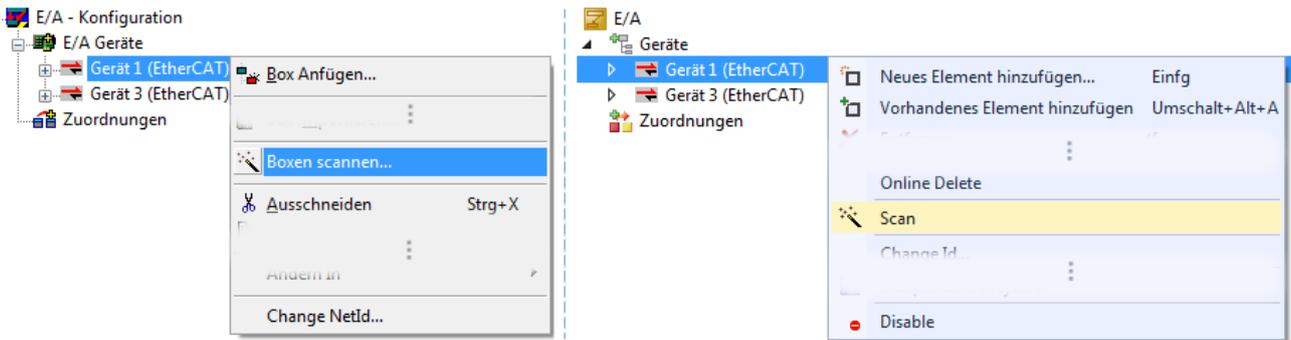


Abb. 94: Manuelles Scannen nach Teilnehmern auf festgelegtem EtherCAT Device (links: TC2; rechts TC3)

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



Abb. 95: Scanfortschritt am Beispiel von TwinCAT 2

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

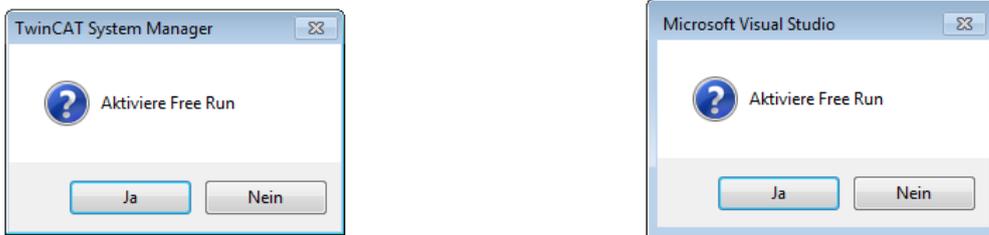


Abb. 96: Abfrage Config/FreeRun (links: TC2; rechts TC3)

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



Abb. 98: TwinCAT kann auch über einen Button in diesen Zustand versetzt werden (links: TC2; rechts TC3)

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

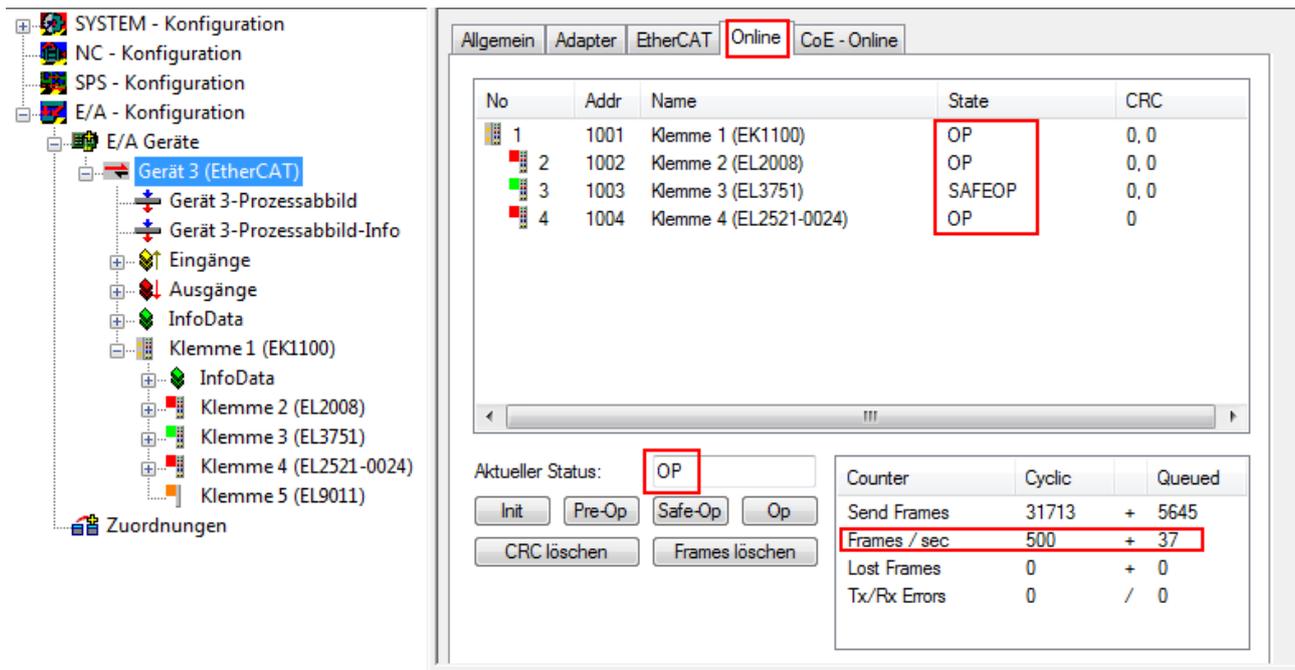


Abb. 99: Beispielhafte Online-Anzeige

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 [▶ 82] 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.

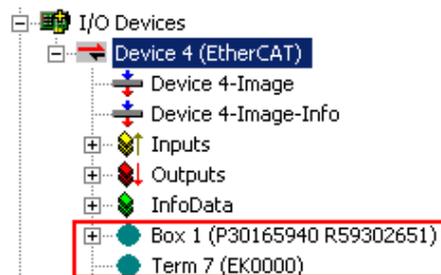


Abb. 100: Fehlerhafte Erkennung

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

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

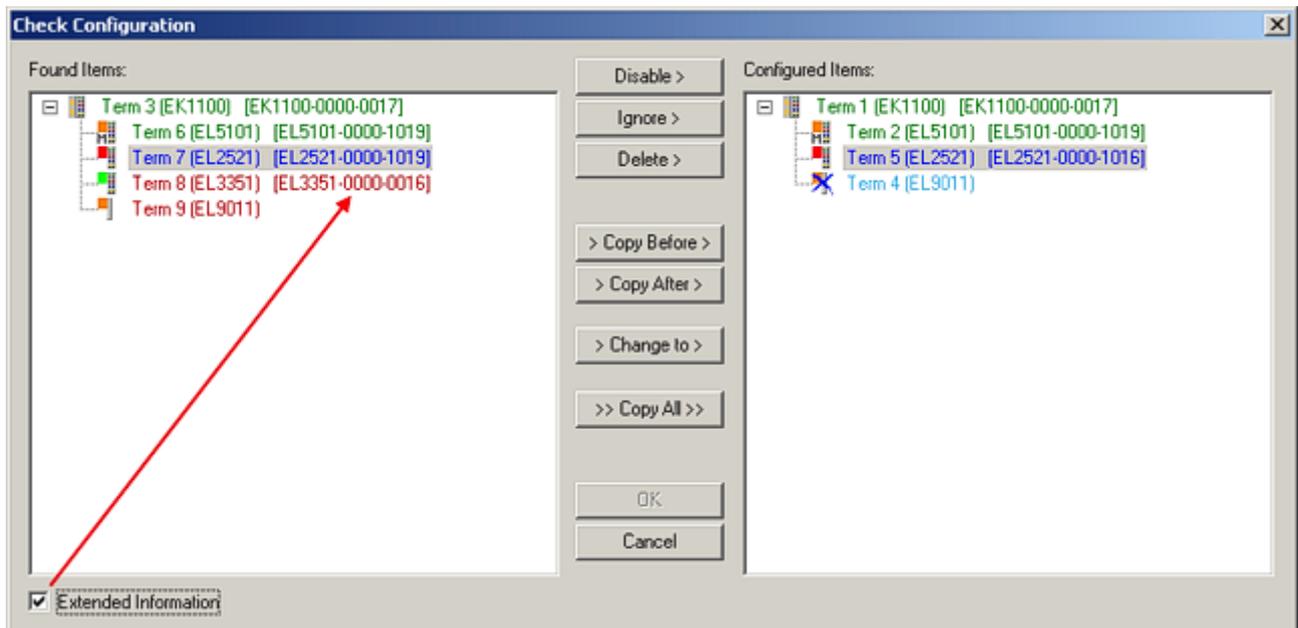


Abb. 102: Korrekturdialog

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.

Farbe	Erläuterung
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"> Dieser EtherCAT-Slave ist auf der Gegenseite nicht vorhanden Er ist vorhanden, aber in einer anderen Revision, die sich auch in den Eigenschaften von der angegebenen unterscheidet. <p>Auch hier gilt dann das Kompatibilitätsprinzip: Ist die gefundene Revision > als die konfigurierte Revision, ist der Einsatz unter Berücksichtigung der Kompatibilität möglich, da Nachfolger-Geräte die Funktionen der Vorgänger-Geräte unterstützen sollen.</p> <p>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.</p>

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.

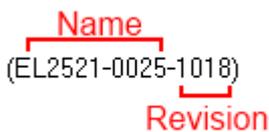


Abb. 103: Name/Revision Klemme

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.

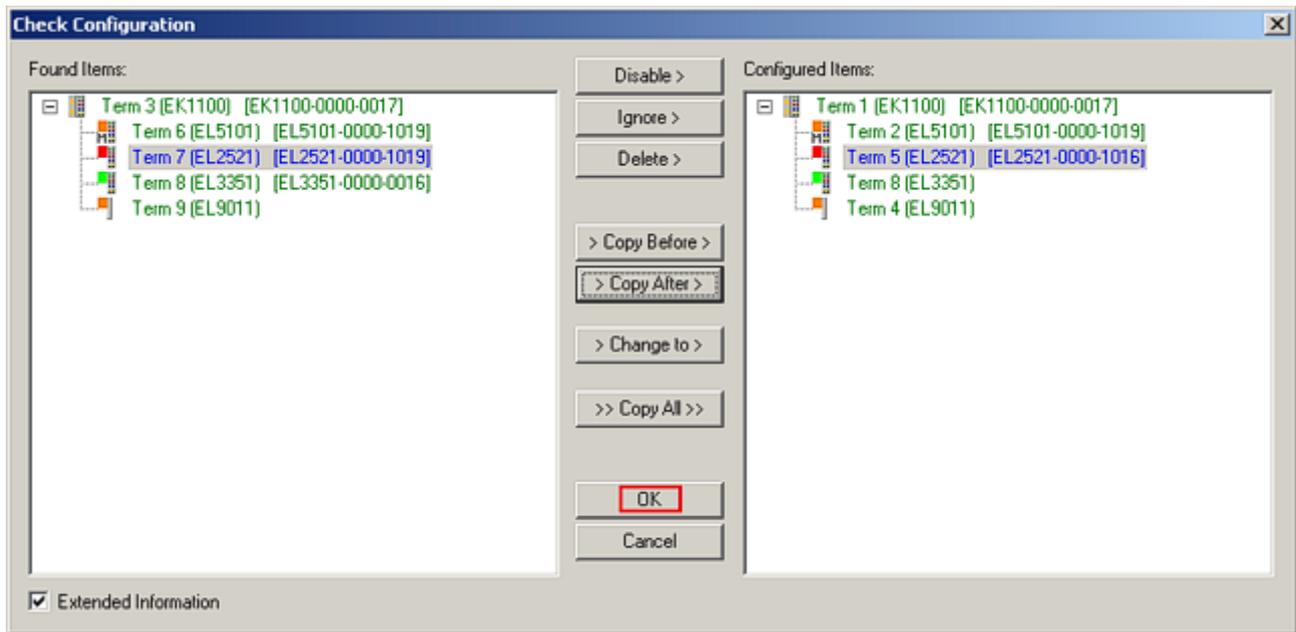


Abb. 104: Korrekturdialog mit Änderungen

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.

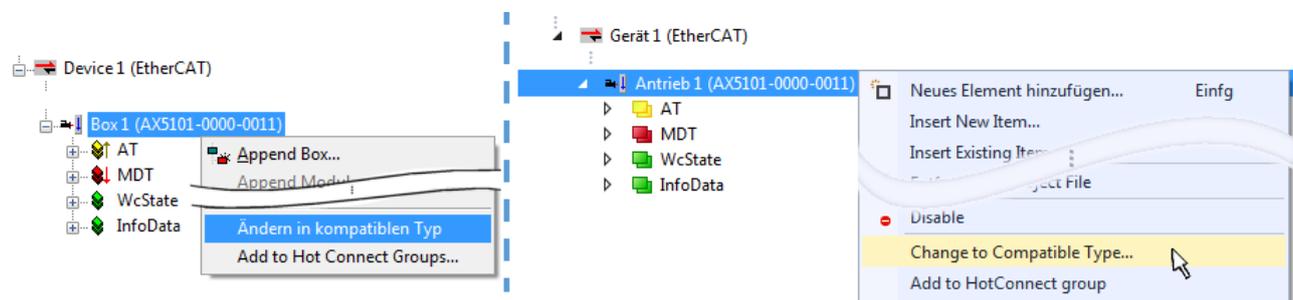


Abb. 105: Dialog „Change to Compatible Type...“ (links: TwinCAT 2; rechts TwinCAT 3)

Folgende Elemente in der ESI eines EtherCAT-Teilnehmers werden von TwinCAT verglichen und als gleich vorausgesetzt, um zu entscheiden, ob ein Gerät als „kompatibel“ angezeigt wird:

- Physics (z.B. RJ45, Ebus...)
- FMMU (zusätzliche sind erlaubt)
- SyncManager (SM, zusätzliche sind erlaubt)
- EoE (Attribute MAC, IP)
- CoE (Attribute SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)
- FoE
- PDO (Prozessdaten: Reihenfolge, SyncUnit SU, SyncManager SM, EntryCount, Entry.Datatype)

Bei Geräten der AX5000-Familie wird diese Funktion intensiv verwendet.

Change to Alternative Type

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

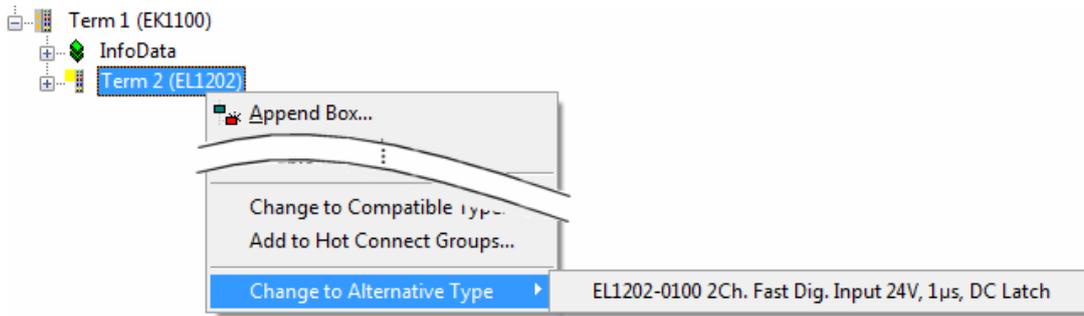


Abb. 106: TwinCAT 2 Dialog Change to Alternative Type

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.

5.2.5 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. 107: „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“

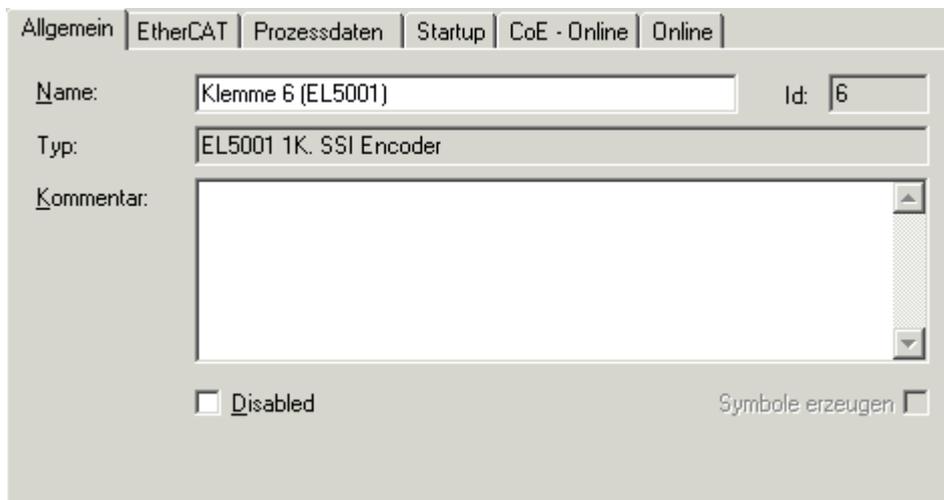


Abb. 108: Karteireiter „Allgemein“

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

Karteireiter „EtherCAT“

Abb. 109: 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_{hex} und für jeden weiteren Folgenden wird die Adresse um 1 verringert ($FFFF_{\text{hex}}$, $FFFE_{\text{hex}}$ usw.).
EtherCAT Adr.	Feste Adresse eines EtherCAT-Slaves. Diese Adresse wird vom EtherCAT-Master während der Startup-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.

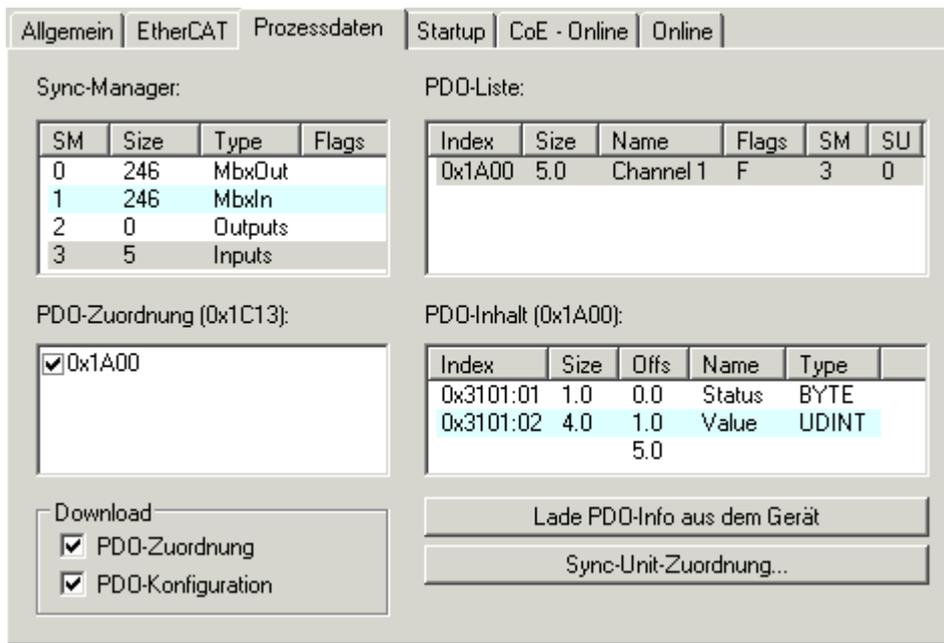


Abb. 110: Karteireiter „Prozessdaten“

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 herstellereitig 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.

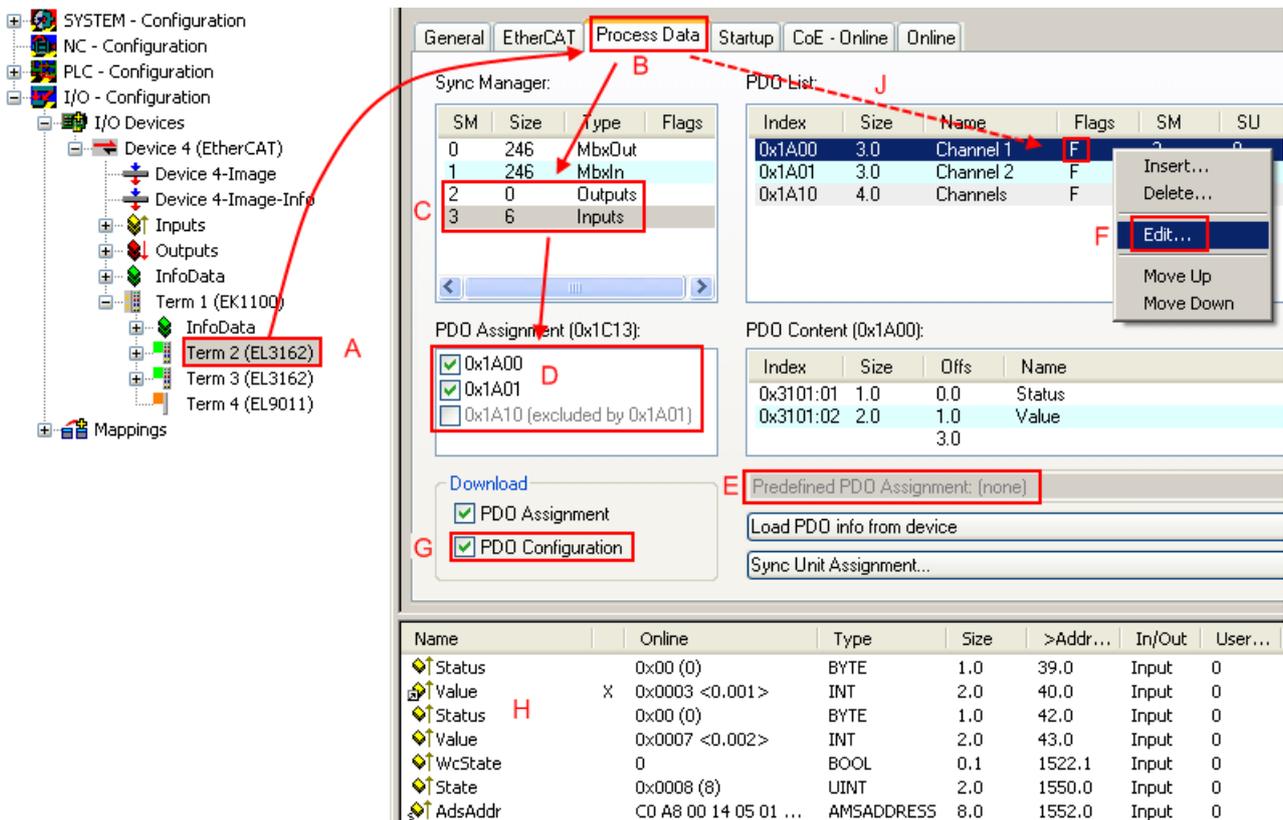


Abb. 111: Konfigurieren der Prozessdaten

Manuelle Veränderung der Prozessdaten

In 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](#) [► 103] 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.

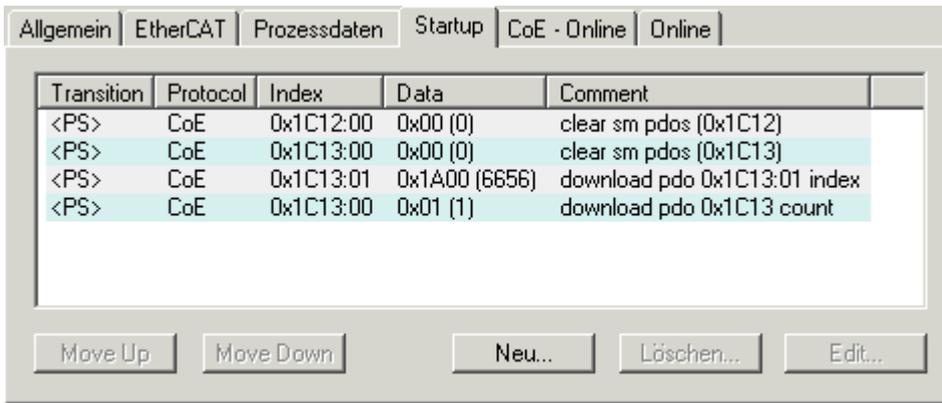


Abb. 112: Karteireiter „Startup“

Spalte	Beschreibung
Transition	Übergang, in den der Request gesendet wird. Dies kann entweder <ul style="list-style-type: none"> • der Übergang von Pre-Operational to Safe-Operational (PS) oder • der Übergang von Safe-Operational to Operational (SO) sein. 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.

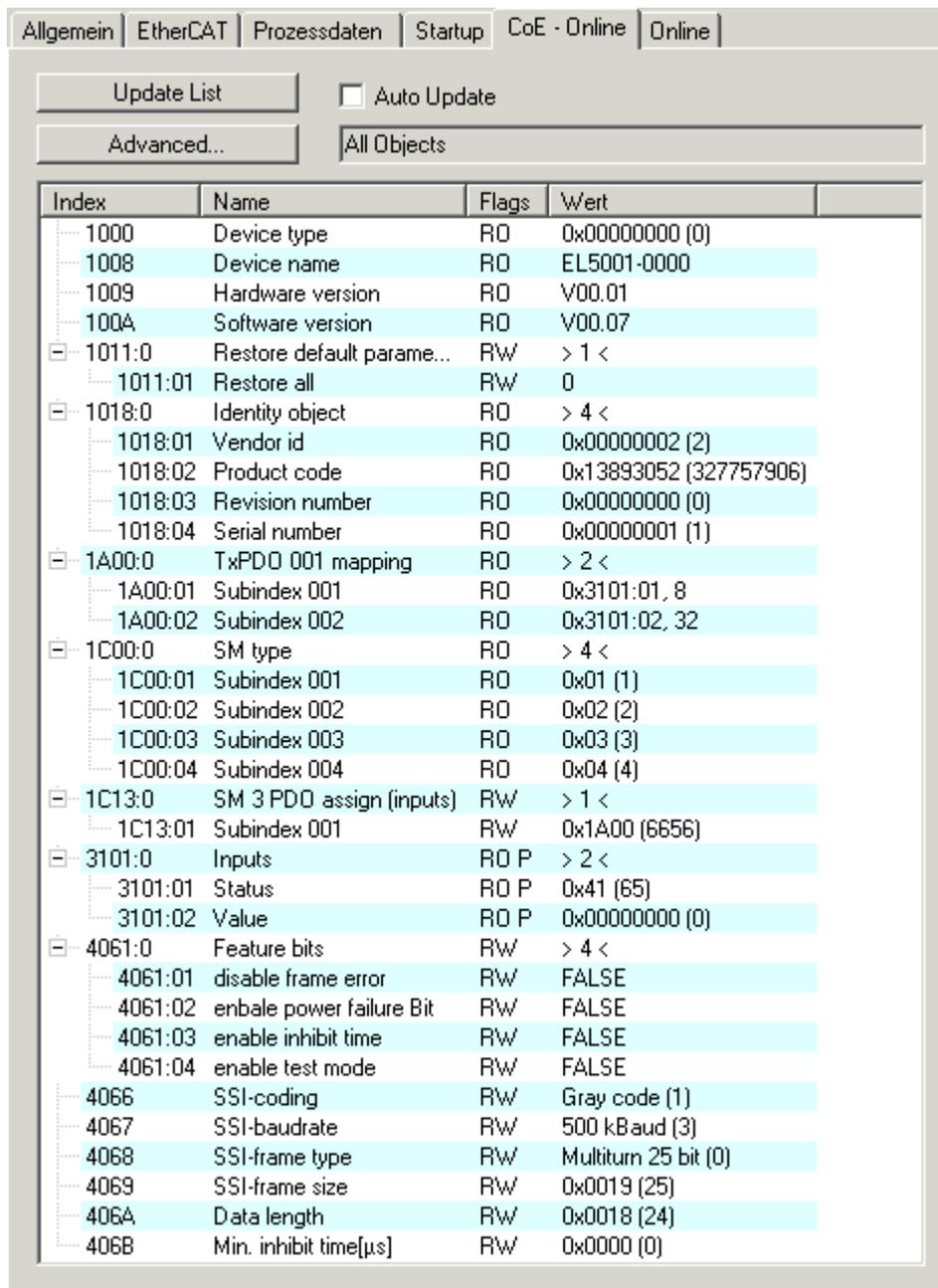


Abb. 113: Karteireiter „CoE - Online“

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.

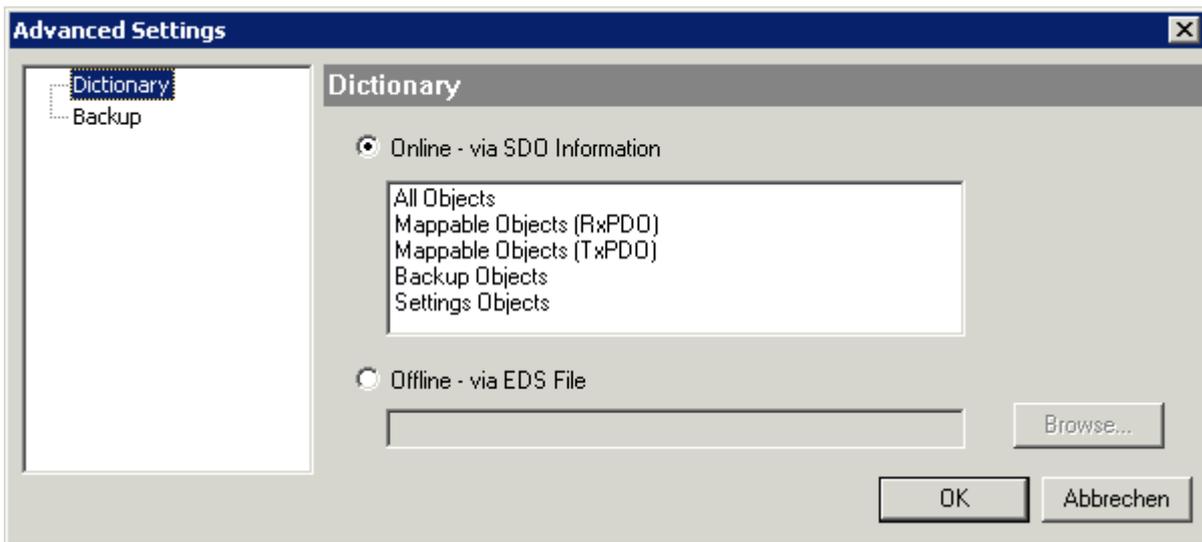


Abb. 114: Dialog „Advanced settings“

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“

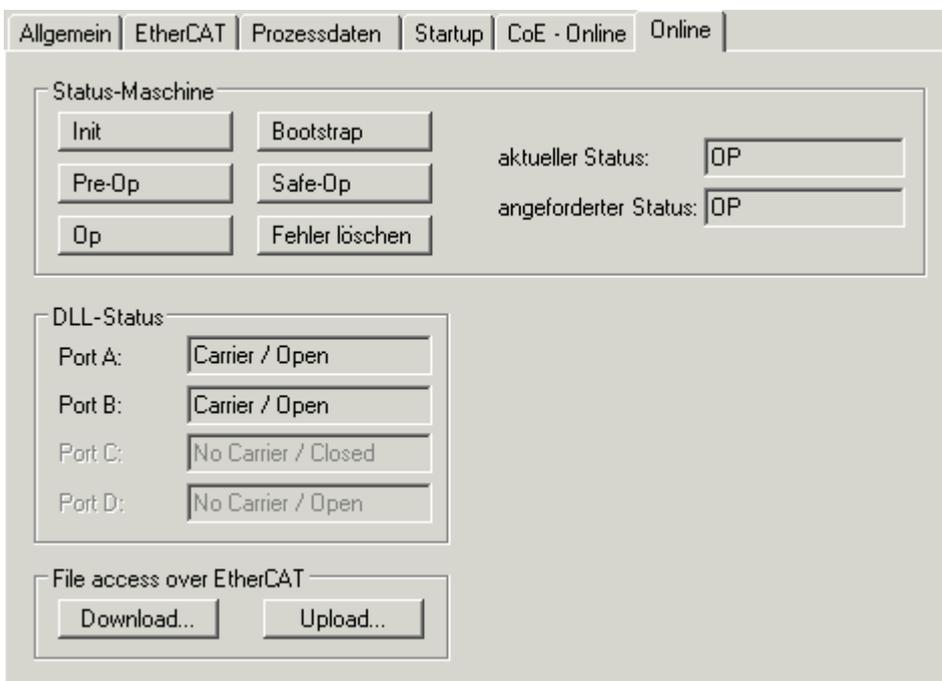


Abb. 115: Karteireiter „Online“

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)

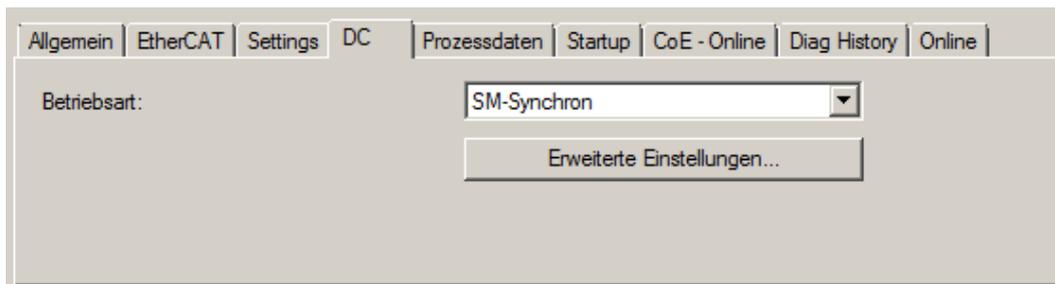


Abb. 116: Karteireiter „DC“ (Distributed Clocks)

- Betriebsart** Auswahlmöglichkeiten (optional):
 - FreeRun
 - SM-Synchron
 - DC-Synchron (Input based)
 - DC-Synchron
- Erweiterte Einstellungen...** Erweiterte Einstellungen für die Nachregelung der echtzeitbestimmenden TwinCAT-Uhr

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

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

5.2.5.1 Detaillierte Beschreibung des Karteireiters „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 \[►_101\]](#))

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.	

Spalte	Beschreibung
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 \[► 98\]](#) 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 für einen EtherCAT-Slave

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 ihr 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.

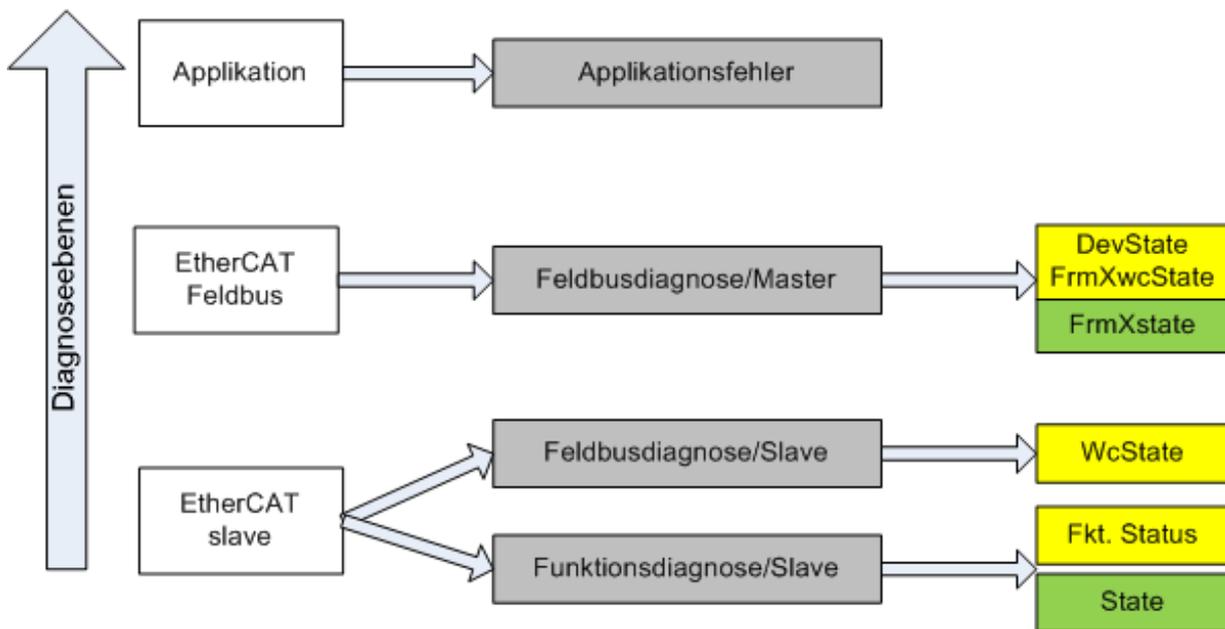


Abb. 117: Auswahl an Diagnoseinformationen eines EtherCAT-Slaves

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-Slaves* 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

Farbe	Bedeutung
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.

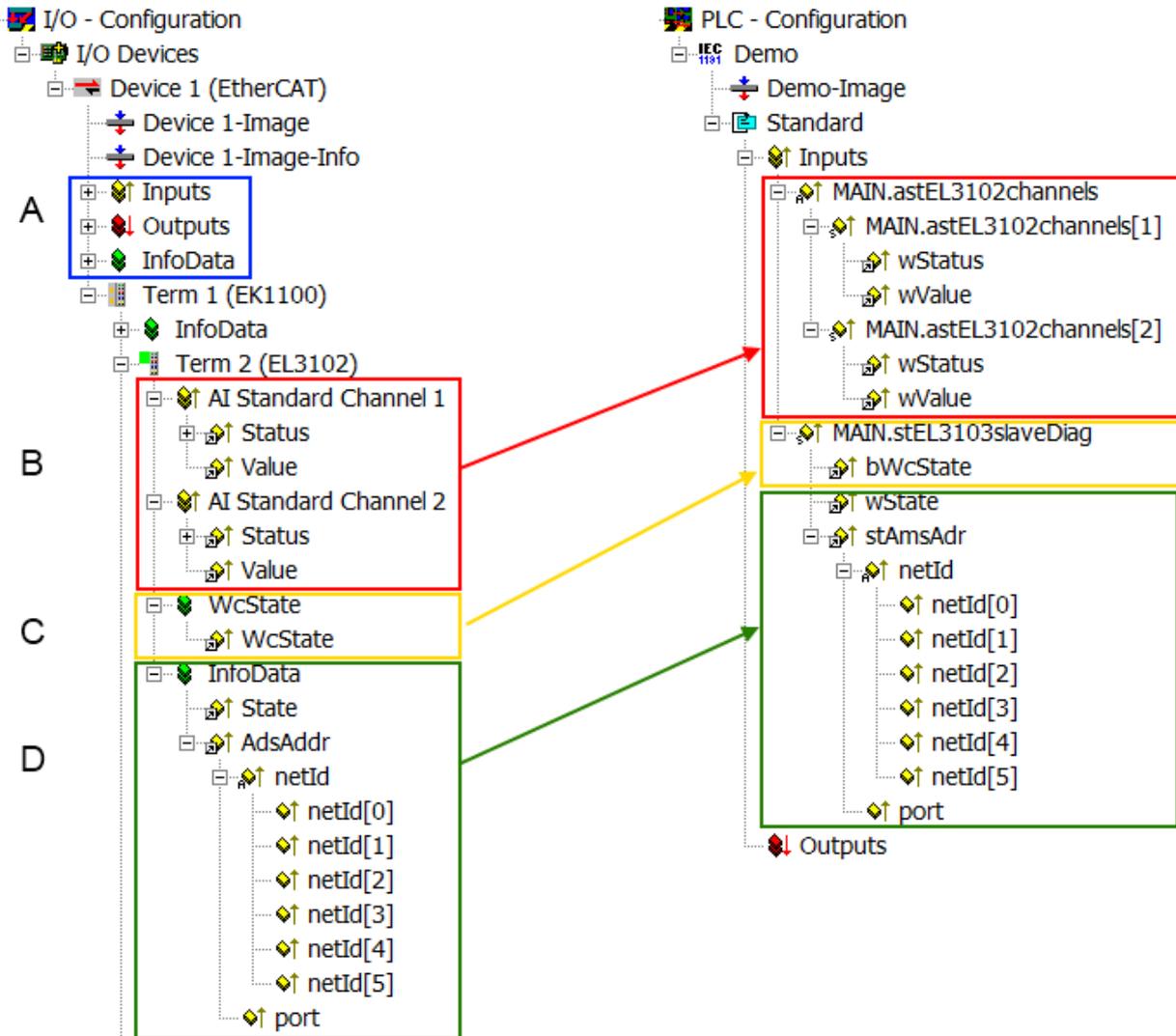


Abb. 118: Grundlegende EtherCAT-Slave Diagnose in der PLC

Dabei werden folgende Aspekte abgedeckt:

Kennzeichen	Funktion	Ausprägung	Anwendung/Auswertung
A	Diagnoseinformationen des EtherCAT-Masters zyklisch aktualisiert (gelb) oder azyklisch bereitgestellt (grün).		Zumindest der DevState ist in der PLC zyklusaktuell auszuwerten. Die Diagnoseinformationen des EtherCAT-Masters bieten noch weitaus mehr Möglichkeiten, die in der EtherCAT-Systemdokumentation behandelt werden. Einige Stichworte: <ul style="list-style-type: none"> • CoE im Master zur Kommunikation mit/über die Slaves • Funktionen aus <i>TcEtherCAT.lib</i>

Kennzeichen	Funktion	Ausprägung	Anwendung/Auswertung
			<ul style="list-style-type: none"> • OnlineScan durchführen
B	Im gewählten Beispiel (EL3102) umfasst die EL3102 zwei analoge Eingangskanäle, die einen eigenen Funktionsstatus zyklusaktuell übermitteln.	<p>Status</p> <ul style="list-style-type: none"> • die Bitdeutungen sind der Gerätedokumentation zu entnehmen • andere Geräte können mehr oder keine slave-typischen Angaben liefern 	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	<p>Für jeden EtherCAT-Slave mit zyklischen Prozessdaten zeigt der Master durch einen so genannten WorkingCounter an, ob der Slave erfolgreich und störungsfrei am zyklischen Prozessdatenverkehr teilnimmt. Diese elementar wichtige Information wird deshalb im System Manager zyklusaktuell</p> <ol style="list-style-type: none"> 1. am EtherCAT-Slave als auch inhaltsidentisch 2. als Sammelvariable am EtherCAT-Master (siehe Punkt A) <p>zur Verlinkung bereitgestellt.</p>	<p>WcState (Working Counter)</p> <p>0: gültige Echtzeitkommunikation im letzten Zyklus</p> <p>1: ungültige Echtzeitkommunikation</p> <p>ggf. Auswirkung auf die Prozessdaten anderer Slaves, die in der gleichen SyncUnit liegen</p>	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	<p>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</p> <ul style="list-style-type: none"> • nur selten/nie verändert werden, außer beim Systemstart • selbst auf azyklischem Weg ermittelt werden (z.B. EtherCAT-Status) 	<p>State</p> <p>aktueller Status (INIT..OP) des Slaves. Im normalen Betriebszustand muss der Slave im OP (=8) sein.</p> <p><i>AdsAddr</i></p> <p>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.</p>	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*:

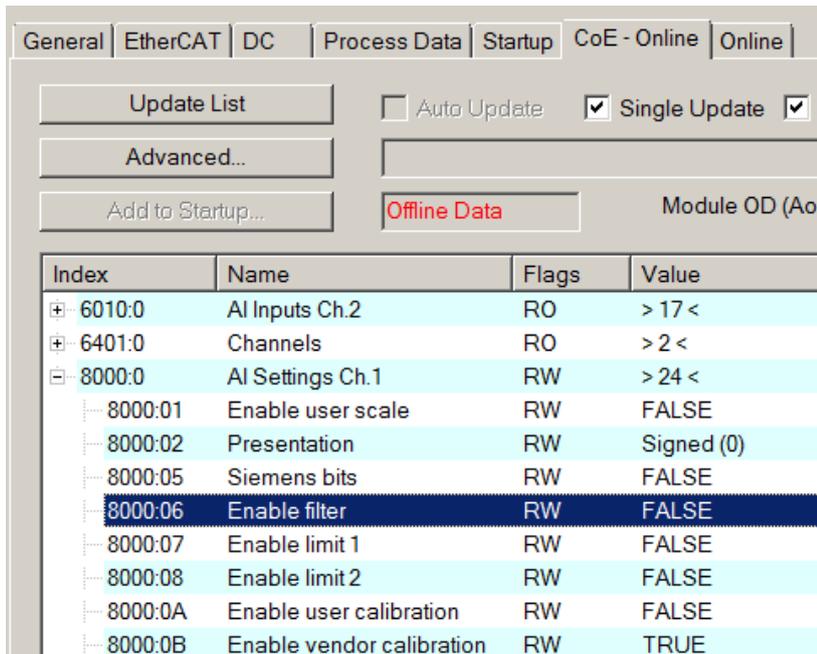


Abb. 119: EL3102, CoE-Verzeichnis

i EtherCAT-Systemdokumentation

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 im TwinCAT System Manager ab TwinCAT 2.11R2 verfügbar. Sie werden über entsprechend erweiterte ESI-Konfigurationsdateien in den System Manager integriert.

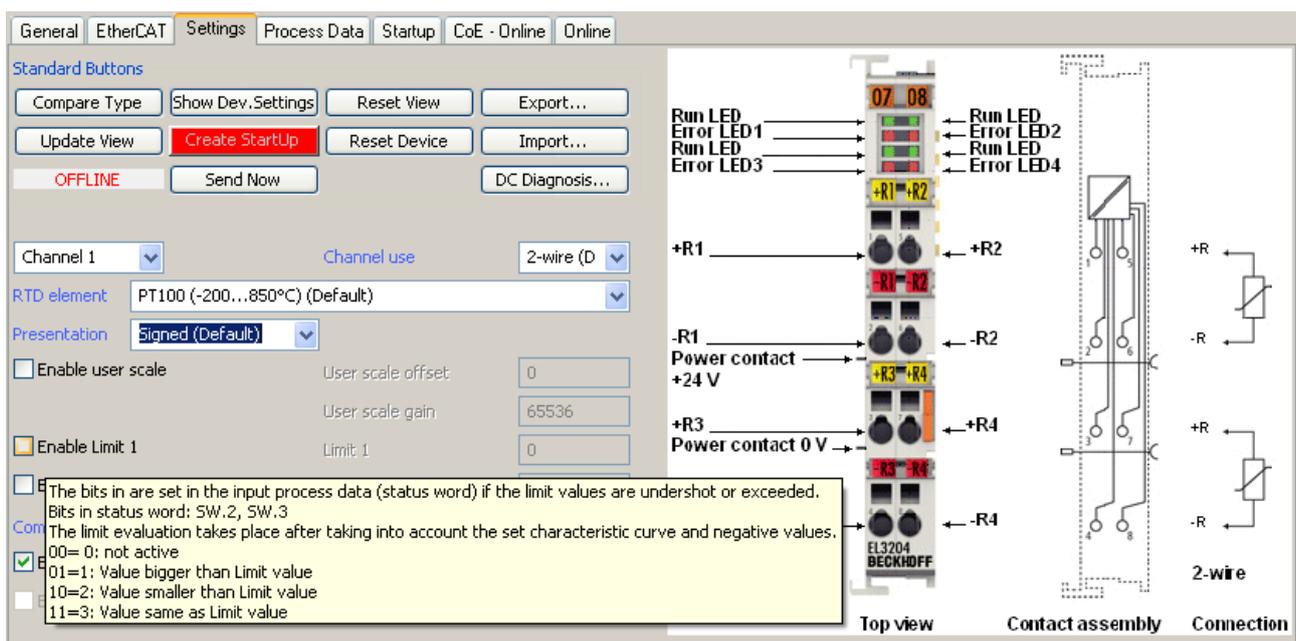


Abb. 120: Beispiel Inbetriebnahmehilfe für eine EL3204

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 Status

- 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 \[► 25\]](#)". 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.

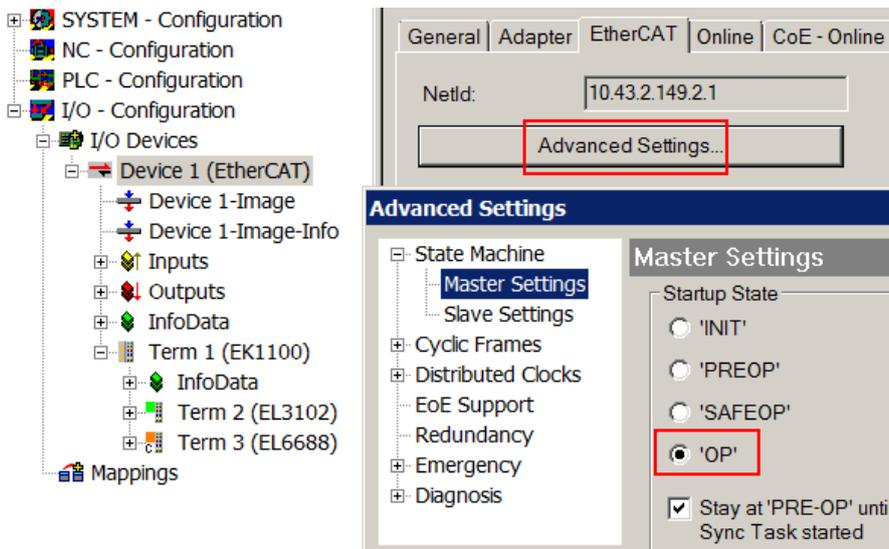


Abb. 121: Default Verhalten System Manager

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

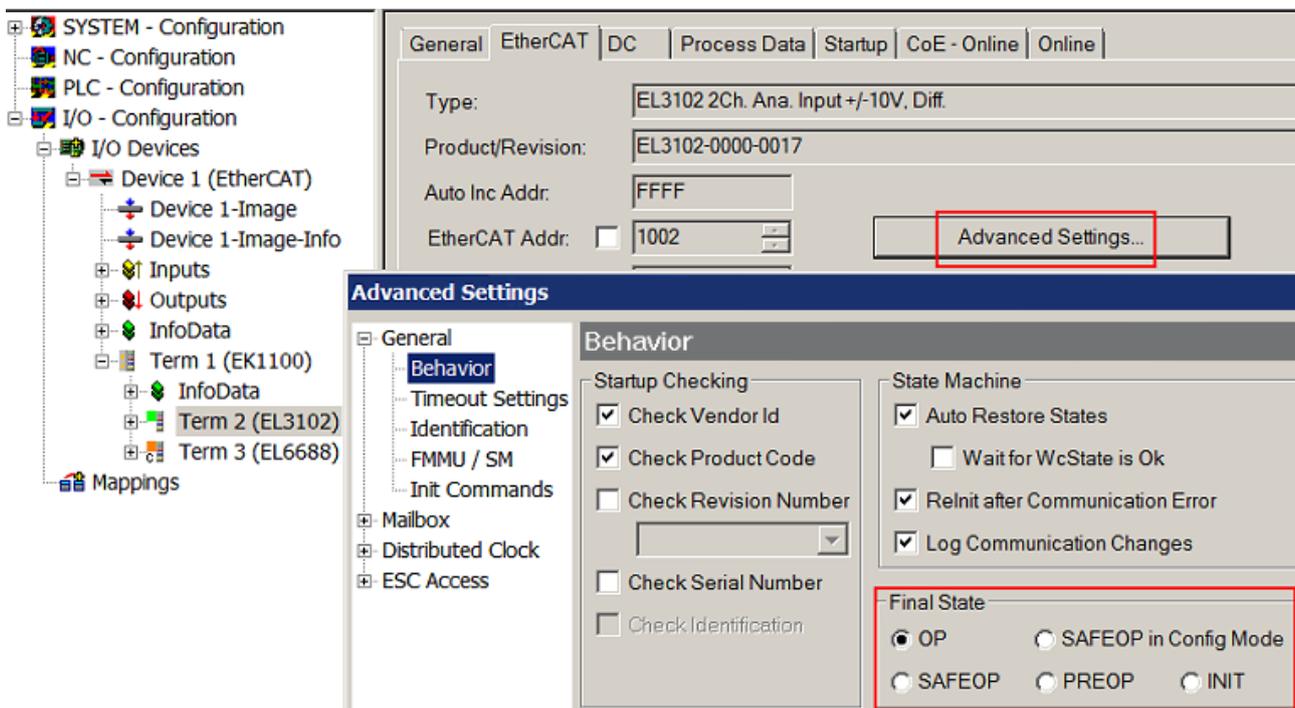


Abb. 122: Default Zielzustand im Slave

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.

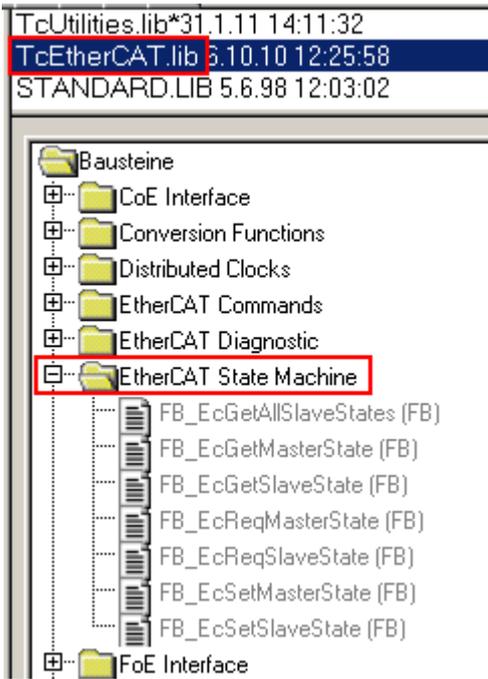


Abb. 123: PLC-Bausteine

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. 124: 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. 125: Warnmeldung E-Bus-Überschreitung

HINWEIS

Fehlfunktion möglich!

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

5.4 CoE Objektübersicht

5.4.1 Standard CoE Objekte

Als Standard CoE Objekte werden die Objekte mit Index (hex) 1000 bis 1018:04 bezeichnet. Dort sind generelle Informationen über das jeweilige Gerät bzw. die jeweilige Klemme gespeichert, wie z.B. Gerätename, Soft- und Hardwareversion sowie Herstellerspezifische Kennungen (Seriennummer, Revision, Vendor ID und Produktcode).

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1000:0	Device type	Geräte-Typ des EtherCAT-Slave: Das Lo-Word enthält das verwendete CoE Profil (5001). Das Hi-Word enthält das Modul Profil entsprechend des Modular Device Profile	UINT32	RO	0x00001389 (5001 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1008:0	Device Name	Geräte-Name des EtherCAT-Slave	STRING	RO	EL6695-0000

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1009:0	Hardware Version	Hardware-Version des EtherCAT-Slave	STRING	RO	02

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
100A:0	Software Version	Firmware -Version des EtherCAT-Slave	STRING	RO	06

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1018:0	Identity	Informationen, um den Slave zu identifizieren	UINT8	RO	0x04 (4 _{dez})
1018:01	Vendor ID	Hersteller-ID des EtherCAT-Slaves	UINT32	RO	0x00000002 (2 _{dez})
1018:02	Product code	Produkt-Code des EtherCAT-Slaves	UINT32	RO	0x1A243052 (438579282 _{dez})
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	0x00000064 (100 _{dez})
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 _{dez})

5.4.2 Klemmenspezifische CoE-Objekte

Ab dem Index (hex) 10F4 beginnen die Klemmenspezifischen CoE-Objekte der Bridge-Klemme EL6695. In den Objekten (hex) 10F4 und 10F5 sind der Status sowie die Einstellungen der externen Synchronisation hinterlegt.

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
10F4:0	External synchronisation status	Information über den Synchronisierungszustand	UINT8	RO	0x13 (19 _{dez})
10F4:01	Sync Mode	Synchronisationsmodus 0 = keine Synchronisierung 1 = Sekundärseite ist Sync Master 2 = Primärseite ist Sync Master 3 = Beide Seiten sind SyncMaster – dieser Zustand ist nicht zulässig und soll durch entsprechende EtherCAT Master Einstellungen geändert werden, indem nur auf eine Seite das PDO Timestamp Mapping aktiviert wird.	BIT2	RO P	0x00 (0 _{dez})
10F4:0E	Control value update toggle	Bit toggelt, wenn ein neuer Control Value bereitsteht	BOOLEAN	RO P	0x00 (0 _{dez})
10F4:0F	Time stamp update toggle	Bit toggelt, wenn neue DC-Daten angeliefert wurden	BOOLEAN	RO P	0x00 (0 _{dez})
10F4:10	External device not connected	0 = Gegenseite ist mit ihrem EtherCAT-Feldbus verbunden 1 = Gegenseite ist nicht mit ihrem EtherCAT-Feldbus verbunden	BOOLEAN	RO P	0x00 (0 _{dez})
10F4:11	Internal time stamp	Distributed Clocks Zeit auf der aktuellen Seite	UINT64	RO P	-
10F4:12	External time stamp	Distributed Clocks Zeit auf der gegenüberliegenden Seite (Remoteseite)	UINT64	RO P	-
10F4:13	Control Value for DC Master Clock	Offset zur Korrektur der niederprioren Reference Clock	INT32	RO P	0x00000000 (0 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
10F5:0	External synchronisation settings	Einstellung zur Synchronisierung der EtherCAT Bridge	UINT8	RO	0x12 (18 _{dez})
10F5:01	Sync master	0: Sync Master ist primärseitig 1: Sync Master ist sekundärseitig	BOOLEAN	RW (PREOP)	0x00 (0 _{dez})
10F5:02	32 Bit time stamps	0: 64-bit Timestamps 1: 32-bit Timestamps	BOOLEAN	RW	0x00 (0 _{dez})
10F5:11	Control Interval (ms)	Intervall in ms zur Berechnung "Control value"	UINT16	RW	0x0000 (0 _{dez})
10F5:12	Additional System Time	Zusätzliche DC-Zeit zur Berechnung "Control value"	UINT64	RW	0x0000000000000000 (0 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1608:0	RxPDO-Map	PDO Mapping RxPDO 1 (PDO-Mapping der deklarierten Ausgangsprozessdaten)	UINT8	RW	0x00 (0 _{dez})
..1608:FF	-	-	-	-	-
...					
161F:00...					
161F:FF					

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1801:0	TxPDO-Par External Sync Compact	PDO Parameter TxPDO 2	UINT8	RO	0x09 (9 _{dez})
1801:06	Exclude TxPDOs	Hier sind die TxPDOs (Index (hex) der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 2 übertragen werden dürfen	OCTET-STRING[4]	RO	02 1A 03 1A
1801:07	TxPDO-State	Der TxPDO State wird gesetzt, wenn die zugehörigen Eingangsdaten nicht korrekt eingelesen werden konnten	BOOLEAN	RO P	0x00 (0 _{dez})
1801:09	TxPDO-Toggle	Das TxPDO Toggle wird mit jedem aktualisieren der zugehörigen Eingangsdaten getoggelt	BOOLEAN	RO P	0x00 (0 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1802:0	TxPDO-Par External Sync	PDO Parameter TxPDO 3	UINT8	RO	0x09 (9 _{dez})
1802:06	Exclude TxPDOs	Hier sind die TxPDOs (Index (hex) der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 3 übertragen werden dürfen	OCTET-STRING[4]	RO	01 1A 03 1A
1802:07	TxPDO-State	Der TxPDO State wird gesetzt, wenn die zugehörigen Eingangsdaten nicht korrekt eingelesen werden konnten	BOOLEAN	RO P	0x00 (0 _{dez})
1802:09	TxPDO-Toggle	Das TxPDO Toggle wird mit jedem aktualisieren der zugehörigen Eingangsdaten getoggelt	BOOLEAN	RO P	0x00 (0 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1803:0	TxPDO-Par External Sync (32 Bit)	PDO Parameter TxPDO 4	UINT8	RO	0x09 (9 _{dez})
1803:06	Exclude TxPDOs	Hier sind die TxPDOs (Index (hex) der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 4 übertragen werden dürfen	OCTET-STRING[4]	RO	01 1A 02 1A

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1803:07	TxPDO-State	Der TxPDO State wird gesetzt, wenn die zugehörigen Eingangsdaten nicht korrekt eingelesen werden konnten	BOOLEAN	RO P	0x00 (0 _{dez})
1803:09	TxPDO-Toggle	Das TxPDO Toggle wird mit jedem aktualisieren der zugehörigen Eingangsdaten getoggelt	BOOLEAN	RO P	0x00 (0 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A01:0	TxPDO-Map External Sync Compact	PDO Mapping TxPDO 1	UINT8	RW	0x05 (5 _{dez})
1A01:01	SubIndex (hex) 001	12 bit align	UINT32	RW	0x0000:00, 12
1A01:02	SubIndex (hex) 002	PDO Mapping entry (object 0x1801, entry 0x09)	UINT32	RW	0x1801:09, 1
1A01:03	SubIndex (hex) 003	PDO Mapping entry (object 0x1801, entry 0x07)	UINT32	RW	0x1801:07, 1
1A01:04	SubIndex (hex) 004	1 bit align	UINT32	RW	0x0000:00, 1
1A01:05	SubIndex (hex) 005	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x10)	UINT32	RW	0x10F4:10, 1

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A02:0	TxPDO-Map External Sync	PDO Mapping TxPDO 2	UINT8	RW	0x09 (9 _{dez})
1A02:01	SubIndex (hex) 001	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x02)	UINT32	RW	0x10F4:01, 2
1A02:02	SubIndex (hex) 002	10 bit align	UINT32	RW	0x0000:00, 10
1A02:03	SubIndex (hex) 003	PDO Mapping entry (object 0x1802, entry 0x09)	UINT32	RW	0x1802:09, 1
1A02:04	SubIndex (hex) 004	PDO Mapping entry (object 0x1802, entry 0x07)	UINT32	RW	0x1802:07, 1
1A02:05	SubIndex (hex) 005	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x0F (Time stamp update toggle))	UINT32	RW	0x10F4:0F, 1
1A02:06	SubIndex (hex) 006	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x10 (External device not connected))	UINT32	RW	0x10F4:10, 1
1A02:07	SubIndex (hex) 007	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x11 (Internal time stamp))	UINT32	RW	0x10F4:11, 64

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A02:08	SubIndex (hex) 008	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x12 (External time stamp))	UINT32	RW	0x10F4:12, 64
1A02:09	SubIndex (hex) 009	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x13 (Reserved))	UINT32	RW	0x10F4:13, 32

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A03:0	TxPDO-Map External Sync	PDO Mapping TxPDO 3	UINT8	RW	0x09 (9 _{dez})
1A03:01	SubIndex (hex) 001	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x02)	UINT32	RW	0x10F4:01, 2
1A03:02	SubIndex (hex) 002	10 bit align	UINT32	RW	0x0000:00, 10
1A03:03	SubIndex (hex) 003	PDO Mapping entry (object 0x1803, entry 0x09)	UINT32	RW	0x1803:09, 1
1A03:04	SubIndex (hex) 004	PDO Mapping entry (object 0x1803, entry 0x07)	UINT32	RW	0x1803:07, 1
1A03:05	SubIndex (hex) 005	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x0F (Time stamp update toggle))	UINT32	RW	0x10F4:0F, 1
1A03:06	SubIndex (hex) 006	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x10 (External device not connected))	UINT32	RW	0x10F4:10, 1
1A03:07	SubIndex (hex) 007	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x11 (Internal time stamp))	UINT32	RW	0x10F4:11, 32
1A03:08	SubIndex (hex) 008	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x12 (External time stamp))	UINT32	RW	0x10F4:12, 32
1A03:09	SubIndex (hex) 009	PDO Mapping entry (object 0x10F4 (External synchronization status), entry 0x13 (Reserved))	UINT32	RW	0x10F4:13, 32

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A04:0	Active TxPDOs-Map	PDO Mapping TxPDO 4	UINT32	RW	0x05 (5 _{dez})
1A04:01	SubIndex (hex) 01	PDO Mapping entry (object 0xF640:01 (Remote Write Cycles u16Count))	UINT32	RW	10 01 40 F6
1A04:02	SubIndex (hex) 02	PDO Mapping entry (object 0xF630:01 Active TxPdo Info PDO 1-8)	UINT32	RW	10 01 30 F6
1A04:03	SubIndex (hex) 03	PDO Mapping entry (object 0xF630:02 Active TxPdo Info PDO 9-16)	UINT32	RW	10 02 30 F6

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A04:04	SubIndex (hex) 04	PDO Mapping entry (object 0xF630:03 Active TxPdo Info PDO 17-24)	UINT32	RW	10 03 30 F6
1A04:05	SubIndex (hex) 05	PDO Mapping entry (object 0xF630:04 Active TxPdo Info PDO 25-32)	UINT32	RW	10 04 30 F6

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A05:0	FoE Info-Map	PDO Mapping TxPDO 5	UINT8	RW	0x01 (1 _{dez})
1A05:01		PDO Mapping entry (object 0xF650:01 Active TxPdo Info PDO 25-32)	UINT32	RW	10 01 50 F6

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default	
1A08:0	TxPDO-Map	PDO Mapping TxPDO 1 (PDO-Mapping der deklarierten Eingangsprozessdaten)	UINT8	RW	0x00 (0 _{dez})	
..1A08:FF	-		-	-	-	
...						
1A1F:00... 1A1F:FF			-			

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C00:0	Sync-manager type	Sync-Manager Type Channels (Mailbox/ Process Data, Read/ Write)	UINT8	RO	0x04 (4 _{dez})
1C00:01	SubIndex (hex) 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1 _{dez})
1C00:02	SubIndex (hex) 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2 _{dez})
1C00:03	SubIndex (hex) 003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3 _{dez})
1C00:04	SubIndex (hex) 004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs Ab FW09: 62 statt 9	UINT8	RW	0x00 (0 _{dez})
1C12:01	SubIndex (hex) 001	1. Zugeordnete RxPDO (enthält den Index (hex) des zugehörigen RxPDO Mapping Objekts)	UINT16	RW	0x0000 (0 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs Ab FW09: 62 statt 9	UINT8	RW	0x01 (1 _{dez})
1C13:01	SubIndex (hex) 001	1. zugeordnete TxPDO (enthält den Index (hex) des zugehörigen TxPDO Mapping Objekts)	UINT16	RW	0x1A01 (6657 _{dez})
1C13:02	SubIndex (hex) 002	2. zugeordnete TxPDO (enthält den Index (hex) des zugehörigen TxPDO Mapping Objekts)	UINT16	RW	0x0000 (0 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C32:0	SM output parameter	Synchronisierungsparameter der Outputs	UINT8	RO	0x20 (32 _{dez})
1C32:01	Sync mode	Aktuelle Synchronisierungsbetriebsart: 0: Free Run 1: Synchron with SM 2 Event 2: DC-Mode - Synchron with SYNC0 Event 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0000 (0 _{dez})
1C32:02	Cycle time	Zykluszeit (in ns): Free Run: Zykluszeit des lokalen Timers Synchron with SM 2 Event: Zykluszeit des Masters DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x00000000 (0 _{dez})
1C32:03	Shift time	Zeit zwischen SYNC0 Event und Ausgabe der Outputs (in ns, nur DC-Mode)	UINT32	RW	0x00000000 (0 _{dez})
1C32:04	Sync modes supported	Unterstützte Synchronisierungsbetriebsarten: Bit 0 = 1: Free Run wird unterstützt Bit 1 = 1: Synchron with SM 2 Event wird unterstützt Bit 2-3 = 01: DC-Mode wird unterstützt Bit 4-5 = 10: Output Shift mit SYNC1 Event (nur DC-Mode) Bit 14 = 1: dynamische Zeiten (Messen durch Beschreiben von 1C32:08)	UINT16	RO	0xC007 (49159 _{dez})
1C32:05	Minimum cycle time	Minimale Zykluszeit (in ns)	UINT32	RO	0x00000000 (0 _{dez})
1C32:06	Calc and copy time	Minimale Zeit zwischen SYNC0 und SYNC1 Event (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 _{dez})
1C32:08	Command	0: Messung der lokalen Zykluszeit wird gestoppt 1: Messung der lokalen Zykluszeit wird gestartet 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	UINT16	RW	0x0000 (0 _{dez})
1C32:09	Maximum delay time	Zeit zwischen SYNC1 Event und Ausgabe der Outputs (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C32:0B	SM event missed counter	Anzahl der ausgefallenen SM-Events im OPERATIONAL (nur im DC Mode)	UINT16	RO	0x0000 (0 _{dez})
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 _{dez})
1C32:0D	Shift too short counter	Anzahl der zu kurzen Abstände zwischen SYNC0 und SYNC1 Event (nur im DC Mode)	UINT16	RO	0x0000 (0 _{dez})
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 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C33:0	SM input parameter	Synchronisierungsparameter der Inputs	UINT8	RO	0x20 (32 _{dez})
1C33:01	Sync mode	Aktuelle Synchronisierungsbetriebsart: 0: Free Run 1: Synchron with SM 3 Event (keine Outputs vorhanden) 2: DC - Synchron with SYNC0 Event 3: DC - Synchron with SYNC1 Event 34: Synchron with SM 2 Event (Outputs vorhanden)	UINT16	RW	0x0000 (0 _{dez})
1C33:02	Cycle time	wie 1C32:02	UINT32	RW	0x00000000 (0 _{dez})
1C33:03	Shift time	Zeit zwischen SYNC0-Event und Einlesen der Inputs (in ns, nur DC-Mode)	UINT32	RW	0x00000000 (0 _{dez})
1C33:04	Sync modes supported	Unterstützte Synchronisierungsbetriebsarten: Bit 0: Free Run wird unterstützt Bit 1: Synchron with SM 2 Event wird unterstützt (Outputs vorhanden) Bit 1: Synchron with SM 3 Event wird unterstützt (keine Outputs vorhanden) Bit 2-3 = 01: DC-Mode wird unterstützt Bit 4-5 = 01: Input Shift durch lokales Ereignis (Outputs vorhanden) Bit 4-5 = 10: Input Shift mit SYNC1 Event (keine Outputs vorhanden) Bit 14 = 1: dynamische Zeiten (Messen durch Beschreiben von 1C32:08 oder 1C33:08)	UINT16	RO	0xC007 (49159 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C33:05	Minimum cycle time	wie 1C32:05	UINT32	RO	0x00000000 (0 _{dez})
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 _{dez})
1C33:08	Command	wie 1C32:08	UINT16	RW	0x0000 (0 _{dez})
1C33:09	Maximum delay time	Zeit zwischen SYNC1-Event und Einlesen der Eingänge (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0 _{dez})
1C33:0B	SM event missed counter	wie 1C32:11	UINT16	RO	0x0000 (0 _{dez})
1C33:0C	Cycle exceeded counter	wie 1C32:12	UINT16	RO	0x0000 (0 _{dez})
1C33:0D	Shift too short counter	wie 1C32:13	UINT16	RO	0x0000 (0 _{dez})
1C33:20	Sync error	wie 1C32:32	BOOLEAN	RO	0x00 (0 _{dez})

5.4.3 Profilspezifische CoE Objekte

i **Auslesen der CoE Objekte 0x6000 und 0x7000**

Es werden beim Auslesen der CoE Objekte 0x6000 und 0x7000 nicht die realen Prozessdaten ausgegeben. Die tatsächlichen Daten können nur indirekt über PDO Zuweisung ausgelesen werden.

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
6000:0	Input Data	Deklarierte Eingangsprozessdaten (dynamisch erzeugt)	UINT8	RO	0x00 (0 _{dez})
6000:01					
...					
6000:FF					

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
7000:0	Output Data	Deklarierte Ausgangsprozessdaten (dynamisch erzeugt)	UINT8	RO	0x00 (0 _{dez})
7000:01					
...					
7000:FF					

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F000:0	Modular device profile	Allgemeine Informationen des Modular Device Profiles	UINT8	RO	0x02 (2 _{dez})
F000:01	Module Index (hex) distance	Index (hex)abstand der Objekte der einzelnen Kanäle	UINT16	RO	0x0010 (16 _{dez})
F000:02	Maximum number of modules	Anzahl der Kanäle	UINT16	RO	0x0001 (1 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F008:0	Code word	z. Zt. reserviert	UINT32	RW	0x00000000 (0 _{dez})

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F010:0	Module list	Max. subIndex (hex)	UINT8	RW	0x01 (1 _{dez})
F010:01	SubIndex (hex) 001	-	UINT32	RW	0x00000000 (0 _{dez})

Aktuell 64 PDOs, 2 Bit pro Tx-PDO

Bit 0 → 1=Mapping vorhanden (p – present)

Bit 1 → 1=Mapping aktiv (a – active)

Index (hex)	Name															
F630	Active TxPdo															
Mapping:	a	p	a	p	a	p	a	p	a	p	a	p	a	p	a	p
0	0															
1	1A07	1A06	1A05	1A04	1A03	1A02	1A01	1A00								
2	1A0F	1A0E	1A0D	1A0C	1A0B	1A0A	1A09	1A08								
3	1A17	1A16	1A15	1A14	1A13	1A12	1A11	1A10								
4	1A1F	1A1E	1A1D	1A1C	1A1B	1A1A	1A19	1A18								
...								
8	1A3F	1A3E	1A3D	1A3C	1A3B	1A3A	1A39	1A38								

Index (hex)	Bit	Bezeichnung	Bedeutung	Kommentar
F800:0	-	Device Config		
F800:01	0x8000	AoE	1 = Protokoll gesperrt	
	0x4000	EoE		
	0x2000	FoE		
	0x1000	SoE		
	0x0800	VoE		
	0x0400	sonstige		
	0x0200	-		Wenn 0 wird geprüft ob das MBX Protokoll ECAT-EEPROM vorhanden ist
	...			
	0x0002	-		Unterdrücke Fehlermeldung
	0x0001	-		1: Wenn nicht geroutet: Fehlermeldung unterdrückt
	...			
0xF800:02	0x0100	FoE Buffer	Freigabe	
	0x2000	Watchdog mode		Der Watchdog der EL6695 wird standardmäßig über das AL Event „Watchdog“ getriggert. Wird 0x2000 einmalig gesetzt, wird dagegen der WriteCycleCounter als Trigger verwendet. Diese Einstellung kann bei Betrieb unter manchen EtherCAT Mastern nötig sein.
	0x0002	OP State	Abschalten Abhängigkeit zur gegenüberliegenden Seite	Abschalten der Abhängigkeit des Betriebszustandes zu der Sekundärseite (nur Geräteemulation)
	0x0001	OP State		Abschalten der Abhängigkeit des Betriebszustandes zu der Primärseite (nur Geräteemulation)
0xF800:03	0x2000		SW Restart	Reboot beim Statewechsel von Any zu Init (A → I)

Index (hex)	Bit	Bezeichnung	Bedeutung	Kommentar
	...			

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F820:0	ADS Server Settings	Destination Net ID / Port	Object	RW	0x02 (2 _{dez})
F820:01	Net ID	Destination Net ID	Array [0..5]	RW	0x00, ..
F820:02	Port	Destination Port	UINT16	RW	0x0000

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F821:0	EL6695 ADS Settings	Source Net ID / Port	Object	RO	0x02 (2 _{dez})
F821:01	Net ID	Source Net ID	Array [0..5]	RO	0x00, ..
F821:02	Port	Source Port	UINT16	RO	0x0000

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
FA20:0	Device Diag	Enthält Diagnose Informationen über Status, CPU und Heap Auslastung, Angaben über gesendete DatenPackets sowie interne Kopierzeit	UINT16	RD	0x1D (29 _{dez})
FA20:01	Status		UINT16	RD	
FA20:02	CPU usage [%]		UINT16	RD	
FA20:03	Heap Usage [%]		UINT16	RD	
FA20:04	AOE Packets		UINT16	RD	
FA20:05	EOE Packets		UINT16	RD	
FA20:06	FOE Packets		UINT16	RD	
FA20:07	SOE Packets		UINT16	RD	
FA20:08	VOE Packets		UINT16	RD	
FA20:09	Other Packets		UINT16	RD	
FA20:0A	Mbx Info		UINT16	RD	
FA20:0B	PD Copy time (my)	Zeit in µs	UINT16	RD	
FA20:0C	PD Copy time (remote)		UINT16	RD	
FA20:0D..	Info 2		UINT16	RD	
FA20:1D	Info 18		UINT16	RD	

6 Funktion und Betriebsarten

Betriebsarten bei EL Klemmen definieren eine grundsätzliche Funktionsweise in Abhängigkeit der Prozessdateneinstellungen.

6.1 Grundlagen zur Funktion

Die EtherCAT-Bridge-Klemme EL6695 ermöglicht den synchronen Echtzeit-Datenaustausch zwischen EtherCAT-Strängen und 2 ggf. unterschiedlichen Mastern. Auch die asynchrone Kommunikation über diverse azyklische Protokolle wie AoE, FoE, EoE, VoE usw. wird unterstützt.

Die EL6695 bietet grundsätzlich 3 Funktionen:

- Synchroner PDO Datenaustausch
- Asynchroner Datenaustausch
- Distributed-Clocks-Synchronisierung

Beim synchronen Datenaustausch der synchronen Prozessdaten werden vom EtherCAT Master vordefinierte Prozessdaten von einer Bridgeseite auf die andere kopiert. Dabei informieren beidseitig Statusvariablen über ausbleibende Daten und den Status der Gegenseite.

Die Synchronisierung der Distributed-Clocks ist in beide Richtungen möglich. Dazu ist zu definieren, welche Seite die übergeordnete Reference Clock enthält, die sog. Grandmaster Clock. Die EL6695 übermittelt dann dem untergeordneten EtherCAT-Master auf der anderen Seite Zeitwerte so dass dieser seine EtherCAT-Systemzeit nachregeln kann, wenn er dieses Feature unterstützt.

Die EL6695 hebt sich von der weiterhin verfügbaren EL6692 durch erweiterte Funktionen wie flexible CoE-Konfiguration, der Möglichkeit einer Geräteemulation und vor allem einer deutlichen Steigerung des Datendurchsatzes ab.

Eine komfortable Konfigurationsoberfläche im TwinCAT System Manager für TwinCAT 3 steht zur Verfügung, für TwinCAT 2 in Vorbereitung. Diese „Extension“ ist aber für den allgemeinen Betrieb nicht zwingend erforderlich, nur für den EmulationMode ist sie Voraussetzung.

Die interne Versorgung der EL6695 bezieht die Energie redundant aus der primären (E-Bus) oder der sekundären Seite (RJ45) – die EL6695 ist also auf beiden EtherCAT-Seiten voll funktionsfähig sobald wenigstens eine der beiden Seiten mit Spannung versorgt wird. Die Spannungsversorgung der Sekundärseite (RJ45) erfolgt über einen externen Anschluss, die Primärseite wird über den E-Bus gespeist. Falls beide Spannungen anliegen, wird die 24V-Versorgung bevorzugt verwendet.

Die EL6695 besteht praktisch aus 2 vollwertigen EtherCAT-Slaves in einem Gehäuse: der EL6695-0000 auf der Primärseite (E-Bus) und der EL6695-0002 auf der Sekundärseite mit Netzkabelanschluss (RJ45).

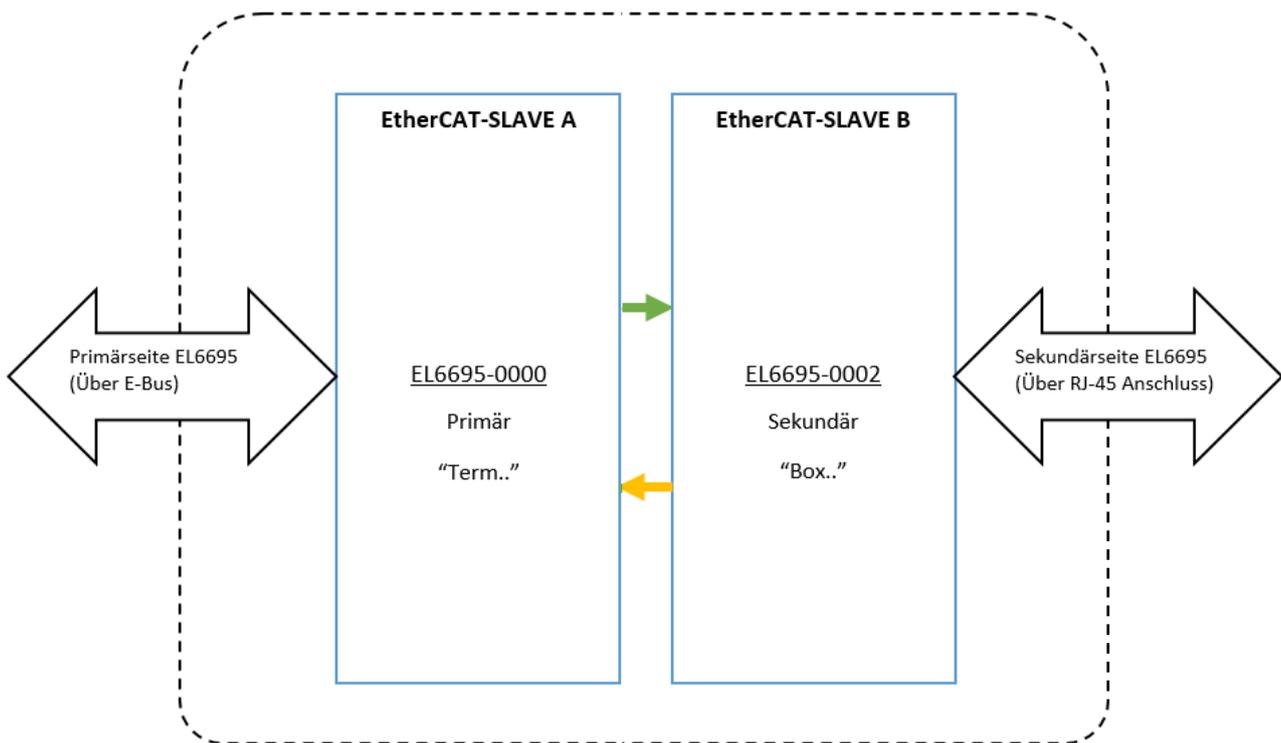


Abb. 126: Basis Funktionalität der Bridge

In Bezug auf die Funktionalität sind beide Bridgeseiten gleichwertig, alle Funktionen können von beiden Seiten aus bedient oder genutzt werden. Dennoch werden die Begriffe „Primär“ für die E-Bus-Seite und „Sekundär“ für die RJ45-Seite aus Lokalisierungsgründen weiterhin verwendet, auch wenn primär/sekundär bei der EL6695 keine Wertigkeit oder Rangfolge ausdrücken soll.

6.1.1 Zugriff auf EL6695 über TwinCAT

Die Konfigurationserstellung bzw. das Einbinden der bridge-Klemme EL6695 in einem TwinCAT Projekt kann entweder über das Kontextmenü durch einen Rechtsklick und Einfügen (**Add New Item..**) erfolgen oder durch einen Rechtsklick auf das jeweilige EtherCAT Device und Auswahl der „Scan“ Funktion aus dem Kontextmenü von Device n (EtherCAT), welche alle angeschlossenen Geräte und Boxen automatisch einliest und sie in die Struktur einbindet (siehe Abbildung):

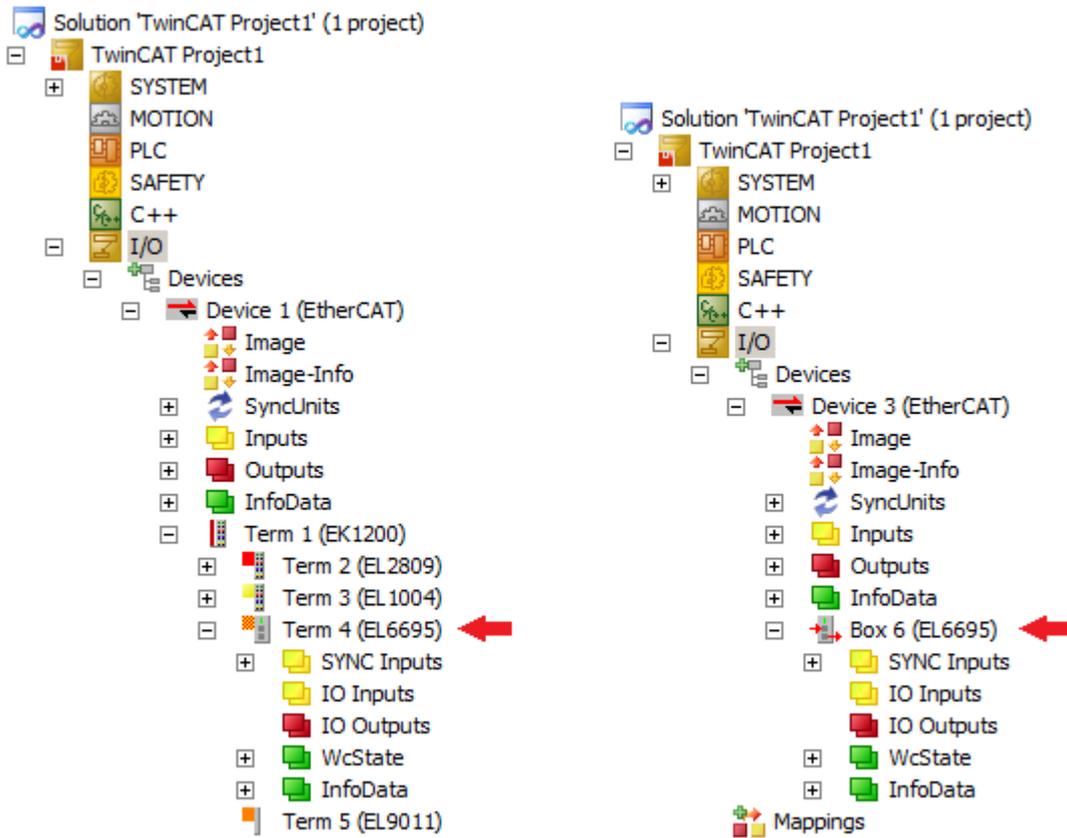


Abb. 127: Nach „scan“: EL6695 im Klemmenstrang (links) und über EtherCAT (per RJ-45 X1) als Box (rechts)

Auf der E-Bus-Seite ist die EL6695 als „Klemme“ im Klemmenstrang zu sehen, über den RJ45 Anschluss wird sie in der Konfiguration als „Box“ dargestellt (vgl. EtherCAT Slave). Der initial-Zustand der bridge-Klemme EL6695 nach dem Einlesen/Erstellen ist:

- Free Run => DC-Unterstützung deaktiviert
- keine Prozessdatenvariablen vorkonfiguriert
- keine Distributed-Clocks Informationen

6.2 Kompatibilität zur EL6692

Die EL6695 ist grundsätzlich als Neuentwicklung zu sehen, die allerdings eine sog. kompatible Betriebsart zur EL6692 unterstützt, in der sie mit denselben Default Prozessdaten (Status/Diagnose) arbeitet. Aufgrund der gänzlich anderen Zeiteigenschaften der EL6695 ist sie aber keinesfalls 1:1 austauschkompatibel mit der EL6692, der Anwender hat in jedem Fall die Funktionsfähigkeit seiner bestehenden Anwendung bei Einsatz der EL6695 zu prüfen.

Durch Anwahl der Diagnosedaten in 0x1A02 oder 0x1A03 können die aus der EL6692 bekannten Diagnosedaten wie SyncMode, TxPDOToggle, TxPDO state, „Timestamp update toggle“ und „External Device not connected“ eingeblendet werden.

Name	Typ	Grö...	>Adr...	Ein/...	User.
◆↑ Sync Mode	BIT2	0.2	39.0	Eing...	0
◆↑ TxPDO toggle	BOOL	0.1	40.4	Eing...	0
◆↑ TxPDO state	BOOL	0.1	40.5	Eing...	0
◆↑ Timestamp update toggle	BOOL	0.1	40.6	Eing...	0
◆↑ External device not connected	BOOL	0.1	40.7	Eing...	0

Abb. 128: EL6695 Kompatibilitätsmodus (default PDO)

6.3 State-Maschine EL6695

Beide Seiten (primär und sekundär) der Bridge-Klemme EL6695 unterstützen die EtherCAT-Stati: INIT, PreOP, SafeOP, OP, BOOTSTRAP.

Status-Änderungen der einen Seite beeinflussen nicht den Status der Gegenseite. Im EmulationMode ist dies nicht so, siehe dort.

● INIT-state

i Eine einseitige „normale“ INIT-State-Anforderung bearbeitet die Bridge-Klemme EL6695 auf der jeweiligen Seite ohne Auswirkung auf die Gegenseite. Nach einer grundlegenden Konfigurationsänderung der Klemme (z.B. durch FW update, Änderung Geräteemulation, Änderung partielles Objektverzeichnis) startet die Klemme auf **beiden** Seiten neu.

6.4 Zyklische Prozessdaten PDO

Zur Differenzierung werden bei der EL6695 zwei Begriffe eingeführt

- „symmetrisches PDO-Mapping“: auf beiden Seiten werden die gleichen Prozessdaten in Größe und Reihenfolge angelegt
- „selektives PDO-Mapping“: eine Seite lädt einen Maximalumfang an Prozessdaten in die EL6695, der Master der anderen Seite wählt daraus nur ein Subset (eine Teilmenge) aus und fordert diese als zyklische Prozessdaten an.

Eckdaten (Stand 2015-06, FW04, TwinCAT 3.1 b4018.4):

- Max. 255 Variablen je Richtung (max. 255 Einträge in 0x1A08:nn und 0x1608:nn durch TwinCAT verwaltet wenn PDO manuell angelegt werden),
- Max. gesamte PDO-Größe: MTU (~1500 Byte) also ein Ethernet Frame je Richtung, dies sind ca. 1408 Byte Nutzdaten. Größere Menge (auf Anfrage) erfordern eine Anpassung in TwinCAT,
- Minimale PDO-Größe ist *ein* Byte, Bit-PDO sind nicht zulässig.

Zur Änderung der Konfiguration ist ein Neustart der jeweiligen EL6695-Seite erforderlich.

Zu jedem 6000er Objekt wird ein entsprechendes 7000er Objekt auf der Gegenseite gesucht – z.B. 6001:05 → 7001:05. Das Objekt auf der Gegenseite (7000er) muss mind. so groß sein wie das 6000er. Insofern ist auch das Teilkopieren einer Teilmenge 7000 → 6000 möglich „selektives PDO Mapping“.

6.4.1 Flusskontrolle

Die Flusskontrolle muss die Applikation übernehmen, es findet in der EL6695 kein Handshake-Mechanismus statt. Es kann allerdings in beiden PDO-Betriebsarten ein Zähler eingeblendet werden, der bei jedem Schreibzugriff der Gegenseite inkrementiert wird: CoE-Objekt 0xF640. Damit dieser Zähler als zyklisches PDO gemappt werden kann, ist er über einen StartUp-Eintrag wie folgt zu mappen:

Um das CoE Objekt 0xF640 mappen zu können, müssen als erstes wie im Kapitel „Grundlagen der Kommunikation“/ „CoE-Interface“, Abschnitt „Online/Offline Verzeichnis“ [► 30] beschrieben, alle online CoE-Objekte angezeigt werden. Anschließend können über den Reiter Process Data die PDO-Daten vom Gerät neu geladen werden, sodass in der PDO-Liste der Eintrag Active TX-PDOs-Map erscheint. Wird nun unter Sync Manager der Eintrag Inputs ausgewählt, kann unter PDO-Assignment der Haken an dem Objekt 0x1A04 gesetzt werden und der Zähler von CoE-Objekt 0xF640 wird ins Prozessabbild gemappt.

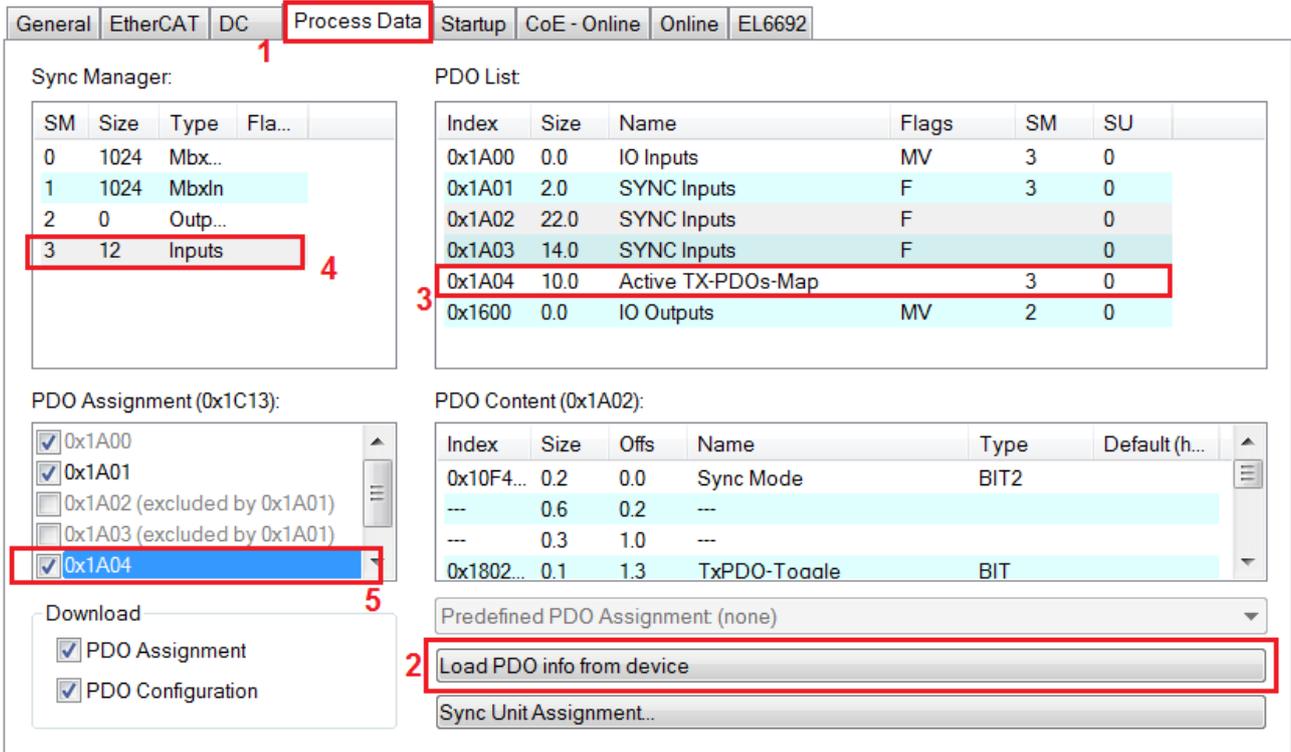


Abb. 129: Laden der PDO-Daten vom Gerät erzeugt den Eintrag Active TX-PDOs-Map

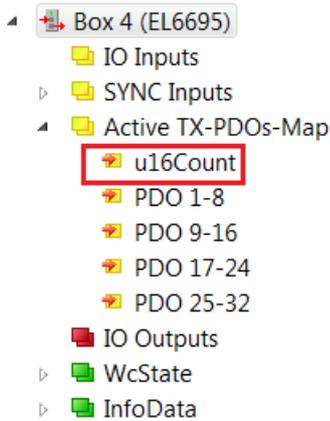
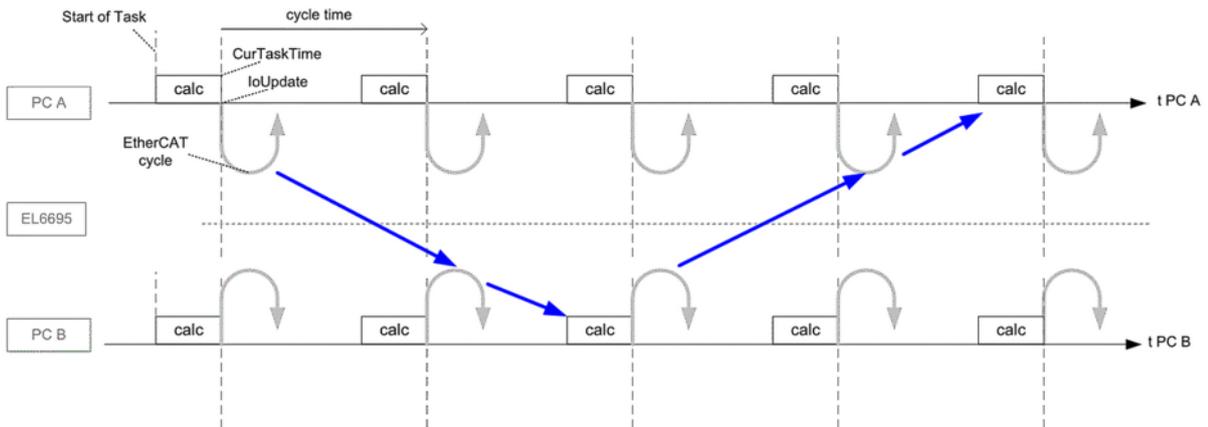


Abb. 130: PDO-Struktur

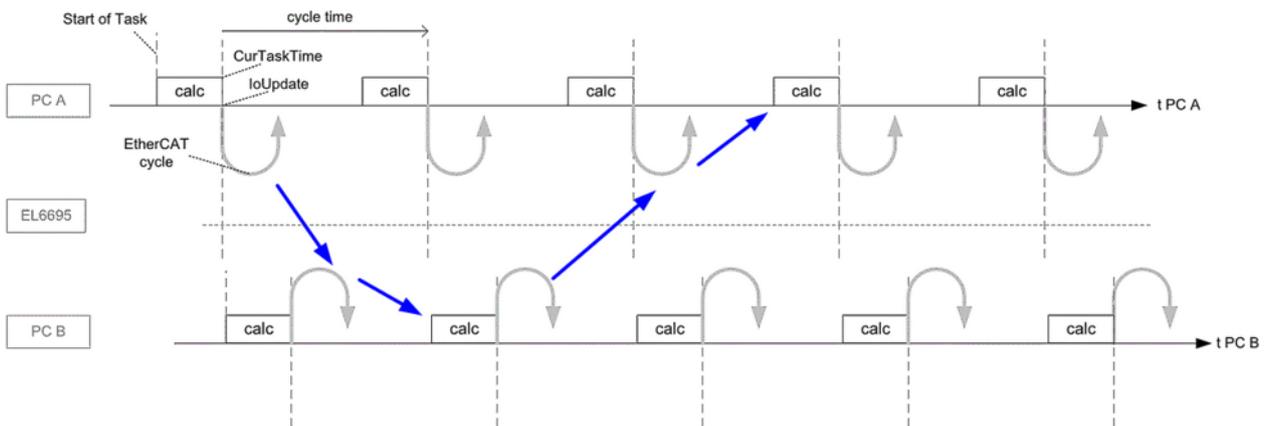
Datendurchsatz (Beispiel)

Im Folgenden ist der Datendurchsatz der EL6695 als Standardkonfiguration und mit zwei Optimierungsmethoden, durch Optimierung durch Synchronisation und durch Optimierung durch separates IO Update inklusive Synchronisation, grafisch dargestellt:

Standard Verhalten I/O am Taskende - durch Jitter der Tasks und unterschiedliche Startzeitpunkte der Tasks zwischen 4 bis 6 Zyklen
 - da die EL6695 schneller als EtherCAT ist, tritt dort keine Verzögerung auf



Optimierung der Synchronisation der Tasks, so dass Task B immer nach Task A das I/O-Update durchläuft
 - somit wäre ein Unterschied von 3 Zyklen möglich



Optimierung durch separates I/O-Update und Synchronisation, Inputs am Anfang, Outputs am Ende
 - somit wäre ein Unterschied von 2 Zyklen möglich

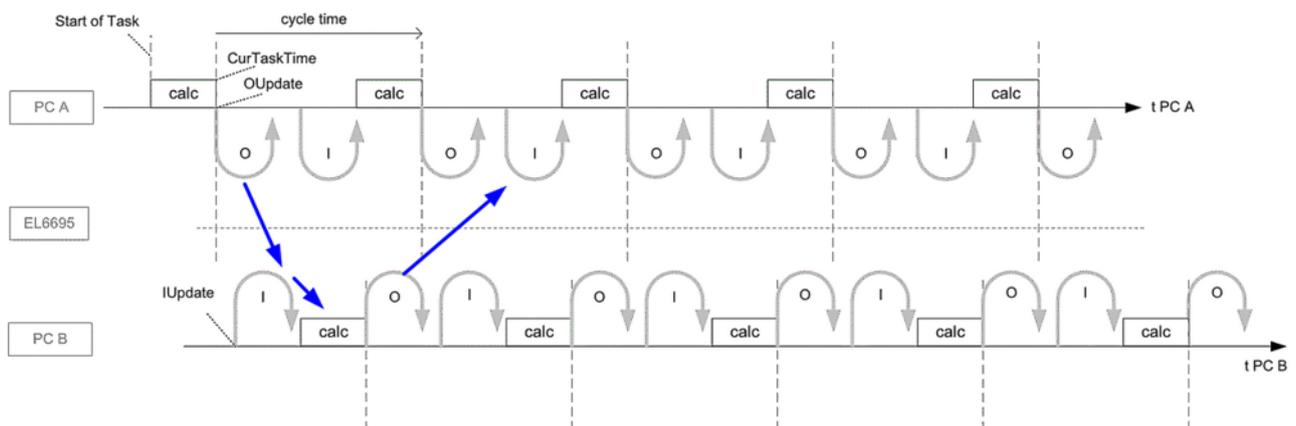


Abb. 131: Datendurchsatz Standard und mit Optimierungen

Um die konfigurierten Prozessdaten von einer EtherCAT-Seite zur anderen zu transportieren, wird eine von der Datenmenge abhängige Zeit benötigt.

Der interne Datentransport wird angestoßen durch das Auslesen einer Seite, vgl. dazu folgender Ablauf anhand der Abb. „EL6695 Ablauf Datentransport“:

- Die SyncManager 2 und 3 arbeiten im 3-Puffer Betrieb:

- Holt ein EtherCAT Master per EtherCAT Frame aus SyncManager 3, Puffer 3 Daten ab (A), startet die EL6695 sofort mit Beendigung dieses Auslesens (B) mit dem Kopieren von neuen Daten der Gegenseite in den nächsten freien Puffer; in diesem Fall Puffer 1. Das dauert die online auslesbare Zeit „Copy Time“. Die Begriffe „my“/ „remote“ beziehen sich dabei auf die „ich“-Seite, je nachdem ob man sich gerade im Online-CoE der primär- oder sekundär-Seite befindet.
- Dort liegen die Daten nun, bis sie von EtherCAT A abgeholt werden.
- Bei langer Zykluszeit kann es vorkommen, dass „relativ alte Daten“ fast einen ganzen EtherCAT-Zyklus dort liegen. Der Transportzeitpunkt kann verschoben werden – durch diese Verzögerung/Delay kann das System so optimiert werden, dass Daten, die von System B in SM2 abgelegt werden, schon kurz danach intern zu Seite A, SM3 transferiert werden und der dann vorbeikommende EtherCAT frame A die Daten relativ frisch abholt. Dabei ist auf ein möglichst jitterarmes System auf Seite A und B zu achten. Das Verschieben kann eingestellt werden über
 - CoE 0x1C33:03 ShiftTime in ns
 - CoE 0xF830:01 nur im Online-CoE verfügbar, delay in μ s

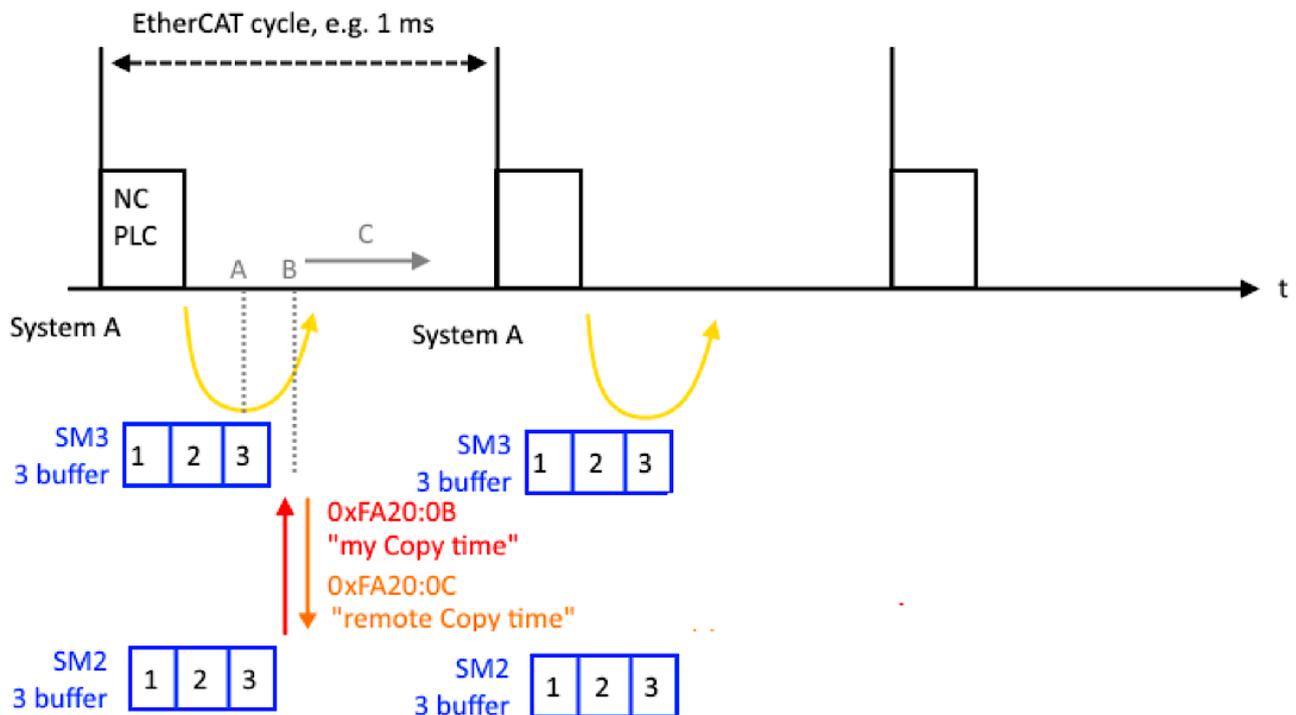


Abb. 132: EL6695 Ablauf Datentransport

Im Folgenden eine exemplarische Messung mit folgendem Aufbau:

- PLC der Primärseite sendet einen Satz PDO
- EL6695 transportiert ihn auf die Gegenseite
- PLC der Primärseite holt die Daten ab und sendet sie ggf. verändert zurück
- EL6695 transportiert sie auf die Gegenseite
- PLC der Primärseite holt die Daten ab, zählt die PLC Zyklen bis hierher

In diesem Beispiel sind keine Optimierungen vorgenommen (PDO Delay, DC Synchronisation).

Anzahl PDO Bytes	Task cycle time	Anzahl Taskzyklen für beide Richtungen (Durchschnittswerte)	Resultierende Übertragungsdauer (eine Richtung)	Kopierzeiten innerhalb der EL6695 der Ein- bzw. Ausgangsdaten; aus CoE Objekt 0xFA20 Device Diag (Durchschnittswerte)
200	50 μ s	4,4	141,1 μ s	14,3 μ s
	100 μ s	3,03	151,5 μ s	14,3 μ s
1400	150 μ s	8,9	667,5 μ s	42,9 μ s
	200 μ s	6,0	600 μ s	42,3 μ s

Soll eine hart gekoppelte zyklische Datenübertragung realisiert werden, empfiehlt sich eine DistributedClocks-Kopplung der beiden EtherCAT-Seiten, um Schwebungen beim Datentransport zu vermeiden.

Hinweis: bei der EL6695 sind faktisch immer zwei Feldbusse mit Ihren Zykluszeiten involviert. Nur bei optimaler Zeitplanung (DC Synchronisierung, ShiftZeiten angepasst) kann in etwa eine so kurze PC A → PC B Transportzeit realisiert werden, wie sie z.B. durch eine direkte Realtime-Ethernet-Strecke mit Publisher/Subscriber möglich ist. Auch dann ist die Optimierung nur für eine Richtung möglich und hat zusätzlich noch die Verzögerung der internen Transportzeit.

6.4.2 Symmetrisches PDO-Mapping

Allgemeine Prozessdaten anlegen / einfacher Datenaustausch (Symmetrisches PDO-Mapping)

Vorgehensweise:

Mit einem Rechtsklick auf **IO Inputs** bzw. **IO Outputs** innerhalb des EL6695 Verzeichnisbaums erscheint das Kontextmenü, in dem **Insert Variable...** (TwinCAT 2) oder **Add New Item** (TwinCAT 3) auszuwählen ist, um damit neue Variablen / Prozessdaten anzulegen (siehe folgende Abbildung).

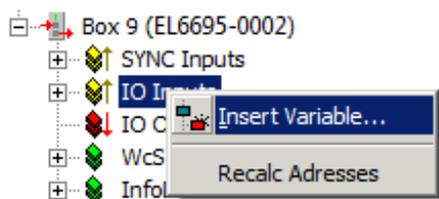


Abb. 133: Kontextmenü für IO Inputs: Einfügen neuer Variablen

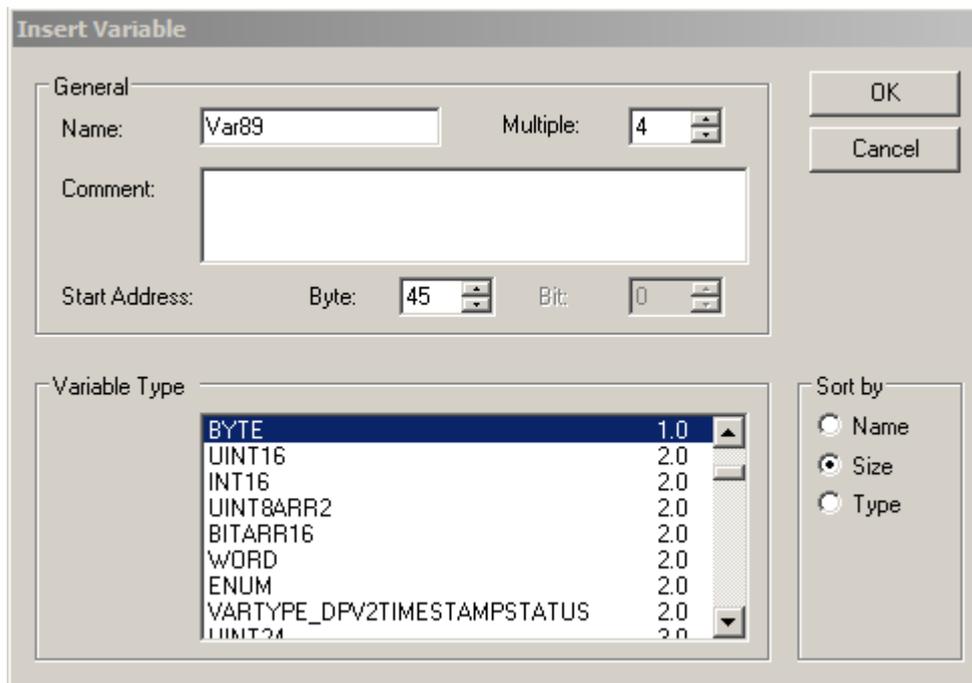


Abb. 134: Dialog zum Einfügen neuer Variablen (hier 4 x BYTE) beginnend mit dem Namen Var89

In diesem Dialog kann nun ein Name für die Variable vergeben werden und der Datentyp kann aus einer Vielzahl von möglichen Typen ausgesucht werden. Zusätzlich lassen sich gleich mehrere Variablen eines Typs durch die **Multiple** Auswahlbox deklarieren und es lässt sich eine definierte Start Adresse einstellen. Bestätigt man obigen Dialog, erhält man das unten zu sehende Prozessabbild.

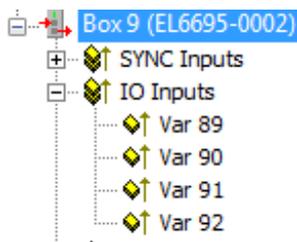


Abb. 135: Neu eingefügte Variablen auf der Sekundärseite der EL6695

Die Variablennamen werden wie hier zu sehen automatisch hochgezählt. Analog werden auch Ausgangsvariablen angelegt.

Auf der Primärseite, also die „Remote Seite“ der EL6695 müssen für einen Datenaustausch nun vier entsprechende Ausgangsvariablen vom gleichen Typ angelegt werden, diese müssen nicht die gleichen Bezeichnung haben, können also auch Var85..Var88 heißen.

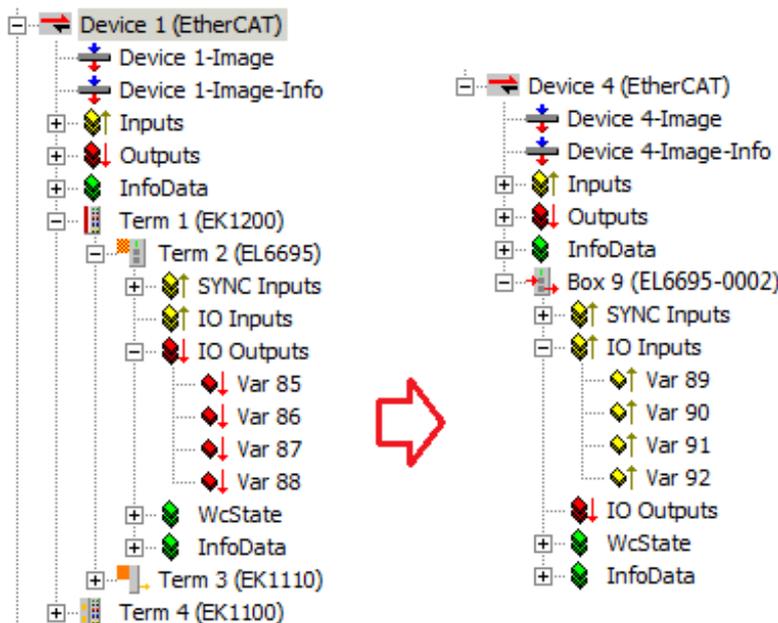


Abb. 136: Anlegen von Prozessdaten

Sind entsprechend auch hier 4 x BYTE Variablen zu den Ausgängen auf der Primärseite der EL6695 hinzugefügt worden, so werden nach Beschreiben dieser Variablen, z.B. über ein PLC Programm, alle Werte kontinuierlich über die vier Eingangsvariablen wiedergegeben. Wird z.B. die Variable Var86 mit einem Wert beschrieben, so wird auch die entsprechenden Ausgangsvariable Var90, im Mapping der Remote Seite der EL6695 mit den gleichen Wert beschrieben (solange die Klemme im OP Zustand ist).

Übertragungsrichtung von zwei Steuerungen PLC1 nach PLC2

I.d.R. wird die Bridge Klemme zum Übertragen von Daten zwischen zwei Steuerungen eingesetzt. Da die Bridge einen Ausgang als einen Eingang und einen Eingang als einen Ausgang abbildet ergibt sich folgende Übertragungsweise:

- Output (PLC1) → Output EL6695 ⇒ Input EL6695-0002 Remote Seite → Input (PLC2)
- Input (PLC1) → Input EL6695 ⇒ Output EL6695-0002 Remote Seite → Output (PLC2)

Die Bridge-Klemme EL6695 „mappt“ jeweils in Byte-Folge einen Ausgangswert mit einen Eingangswert und umgekehrt (ohne sonstige Konfigurationen oder Einstellungen). Daher müssen die Datentypen auf jeder Seite gleich sein.

Automatische Konfiguration über die Konfigurationsoberfläche/Extension

Durch die Extension der **EL6695** mit der Schaltfläche [**Create configuration**] sind die angelegten Variablen auf der gegenüberliegenden Seite online auslesbar und müssen dort nicht explizit angelegt werden. Diese Funktion wird zur Abbildung eines umfangreichen Prozessabbildes empfohlen und steht nur über das [EL6695]-PlugIn Verfügung. Zudem muss die EL6695 von beiden Seiten betriebsbereit, d.h. „online“ zugänglich sein. Im Folgenden wird dazu die Vorgehensweise für TwinCAT 3 gezeigt, in dem auf der Primärseite erstellter Variablen auf die Sekundärseite automatisch „gespiegelt“ werden:

- Vorbedingung: TwinCAT befindet sich im „Free-Run“/ „Config Mode“ (siehe Symbol unten rechts , ) , eine EL6695 befindet sich primärseitig unter einem „Device“ (Term)
- **A)** Auf der Primärseite wurden Variablen angelegt (z.B. 10 Eingangs- und 12 Ausgangsvariablen):

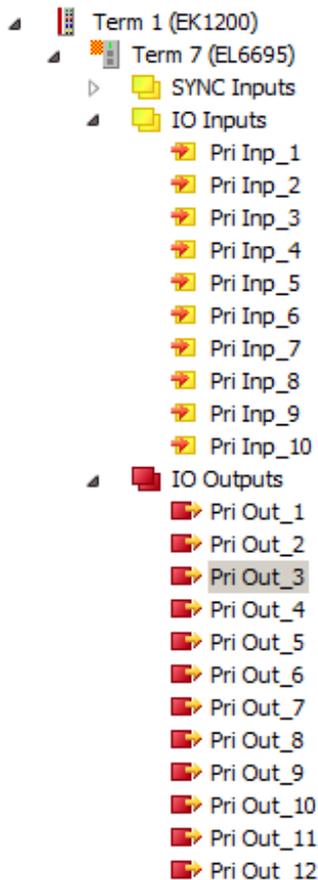


Abb. 137: Beispielhaft angelegte Variablen auf der Primärseite der EL6695

- **B)** Die Klemme ist auf der Primärseite markiert; mit dem Button [Create configuration] unter „Process Data“ im Karteireiter [EL6695] wird nun ein „gespiegelter“ Satz von Variablen für die andere Seite, in diesem Falle die Sekundärseite *intern* von der Klemme, erzeugt; ist daher an dieser Stelle noch nicht sichtbar

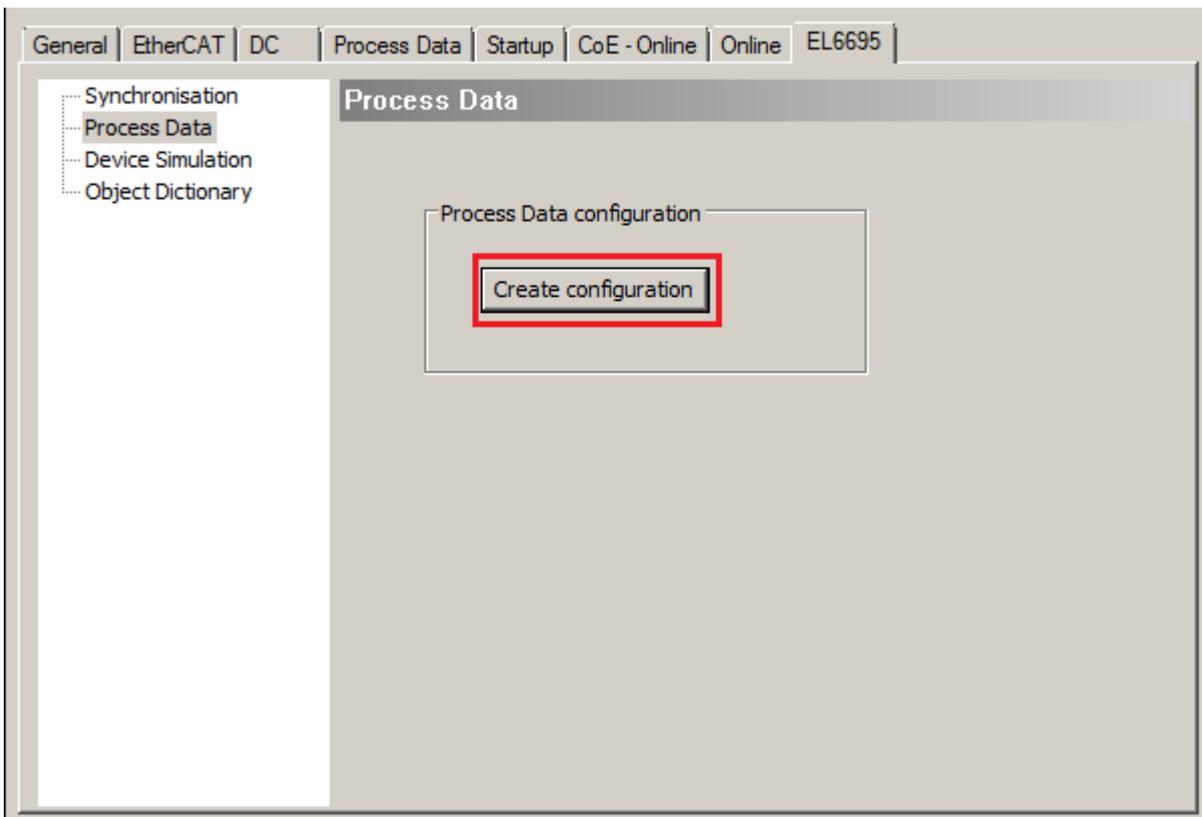
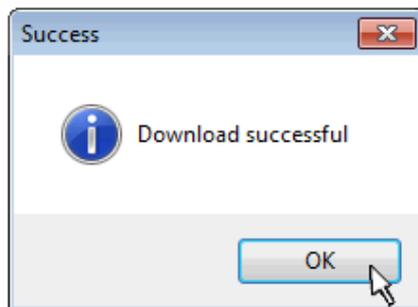
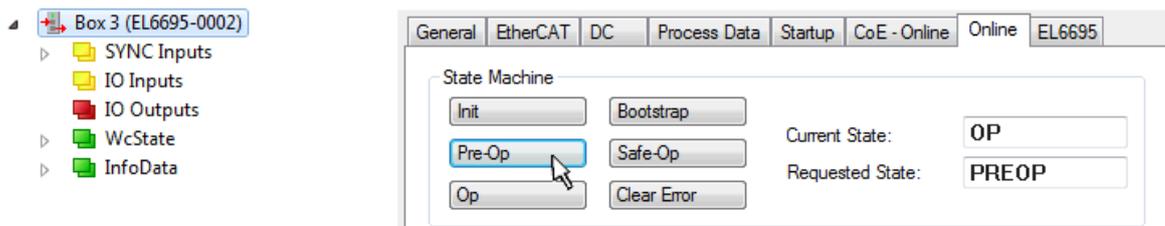


Abb. 138: [Create configuration] unter „Process Data“ im Karteireiter der EL6695

- **C)** Die erfolgreiche Durchführung dieser Aktion wird entsprechend quittiert:



- **D)** Die primäre Seite muss nun einmal über INIT nach PreOp und OP gestellt werden.
- **E)** Die weiteren Schritte erfolgen auf der sekundären Seite. Damit die Klemme nun durch eine interne Initialisierung die Daten der Variablen übernimmt, muss über einen scan-Vorgang auf dem Device der sekundären Seite die Klemme neu eingelesen werden. Ggf. ist sie im Status Error PreOp da die PDO der primären und sekundären Seite nun nicht zusammenpassen. Sie ist dabei manuell bis in den Status PreOp zu fahren - nicht nach OP - damit die StartUp-Liste nicht gesendet wird. Deshalb darf auch FreeRun nicht aktiviert werden.



- **F)** Für den weiteren Ablauf ist zunächst die Objekt-Verzeichnisstruktur der Klemme in TwinCAT zu übertragen. Dies geschieht über den Karteireiter [CoE - Online] über die Auswahl „Erweitert...“ sowie nachfolgender Schritte (1 und 2), wie dargestellt:

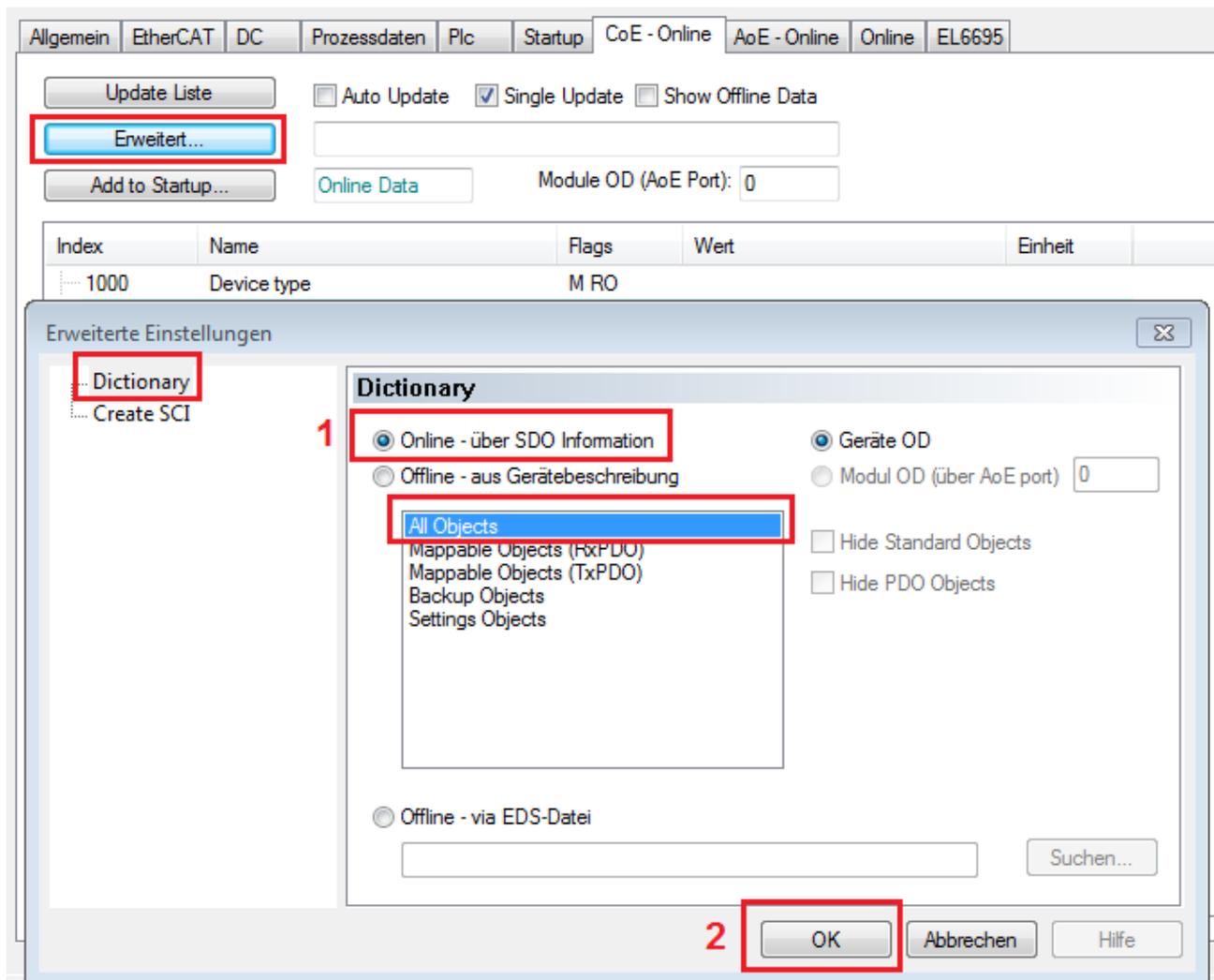


Abb. 139: Laden der Objekt-Verzeichnisstruktur aus der Klemme

- **G)** Unter dem Karteireiter [Prozess Data] wird mittels [Load PDO info from device] die „gespiegelte“ Variablenkonfiguration von der anderen Seite erzeugt und dann im „Solutionexplorer“ (links) sichtbar:

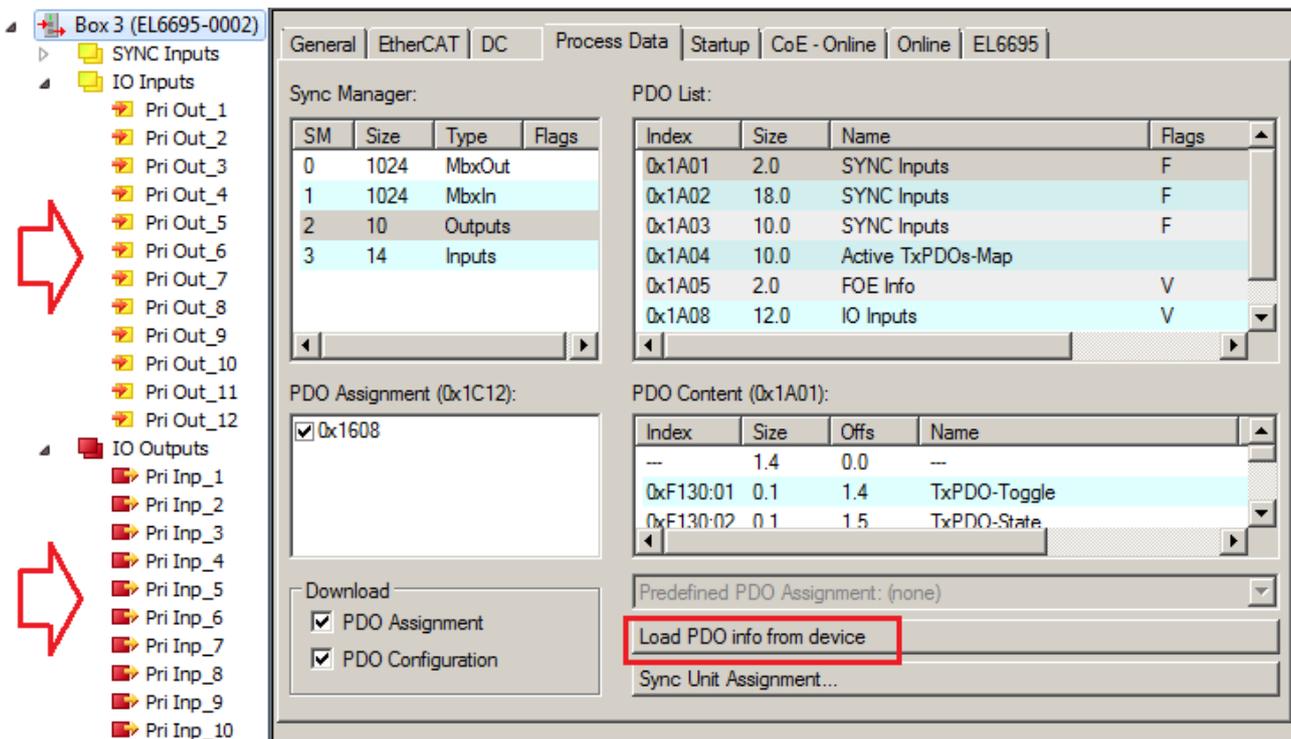


Abb. 140: Variablen der Primärseite mittels „Load PDO info from device“ in gespiegelter Form auf die Sekundärseite übertragen

- **H)** Ggf. Konfiguration mittels „Reload Devices“ () im TwinCAT-Menu neu laden und FreeRun aktiviert werden

Entsprechend dem hier aufgezeigt Vorgang können Variablen ebenso für die Primärseite automatisch erzeugt werden, falls ein I/O-Variablensatz, d.h. eine PDO-Konfiguration auf der Sekundärseite vorliegt.

Einschränkung Variablen

Folgende PDO/Variablentypen sind für dieses Verfahren zu verwenden:
Byte, UINT, UDINT

Nutzung ohne TwinCAT

Steht auf der Gegenseite kein TwinCAT EtherCAT Master mit [Load PDO info from device] zur Verfügung, muss der Mechanismus des PDO Uploads dort nachgebaut werden:

- PDO Assignment Objekte 0x1C12, 0x1C13 auslesen
- anschließend die PDO Mapping Objekte 0x160x (Outputs), 0x1A0x (Inputs) auslesen und entsprechend
- die PDO 0x60xx, 0x70xx auslesen

i Falls ein Scan-Vorgang nicht möglich/nicht erwünscht ist

Um ohne Scan auf der Sekundärseite die Variablen hier erzeugen zu lassen, muss nach [Create configuration], Schritte **B)** und **C)**, die Klemme auf der Primärseite manuell in den INIT- und dann wieder in den OP Zustand versetzt werden (Karteireiter „Online“: Klick auf Init, danach: Safe-Op, Pre-Op, OP). Es kann dann auf der Sekundärseite anstelle eines Scan die entsprechende Variante der EL6695 (-0002) per „addItem“ eingefügt werden. Ansonsten ist der Ablauf wie oben ab **E)** beschrieben.

i Verwendung der Beispielprogramme

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.

→ Beispielkonfiguration: <https://infosys.beckhoff.com/content/1031/el6695/Resources/3521535883.zip>

6.4.3 Selektives PDO-Mapping

Selektives PDO-Mapping

Durch selektives PDO Mapping kann ein Anwender seinen eigenen zyklischen Datenaustausch konfigurieren. Dazu wird auf einer Seite der Klemme ein „vollständiger“ Satz Prozessdaten angelegt. Auf der anderen Seite kann nun ein Teilbereich davon definiert werden, der zyklisch ausgelesen werden soll. Das ist das sog. selektive PDO-Mapping.

Dabei wird das voreingestellte PDO Assignment und Mapping entfernt und der Benutzer muss sich ein eigenes PDO Assignment, sowie Mapping erstellen. Dazu ist es wichtig den Aufbau eines PDO Assignment bzw Mappings zu kennen.

Grundlegend (Standard) gibt es zwei CoE Objekte 0x1C12 „Rx PDO Assign“ und 0x1C13 „Tx PDO Assign“ in denen jeweils die Zuweisungen für den Dateneingang (RxPDO) und den Datenausgang (TxPDO) entsprechend für die Sekundär- und Primärseite definiert sind. In diesen ist ein Verweis auf die CoE Objekte 0x1608 „IO RxPDO Map“ und 0x1A08 „IO TxPDO Map“ in denen das Mapping der In- und Outputs definiert ist. Alle vier dieser CoE Objekte 0x1C12 und 0x1C13 sowie 0x1608 und 0x1A08 sind mit dem RW Flag versehen (RW = read/write). Die beiden Mapping Objekte 0x1608 und 0x1A08 wiederum verweisen initial auf die CoE Objekte 0x7000 „PD Outputs“ und 0x6000 „PD Inputs“ (Process Data Inputs/ Outputs) in denen Prozessdaten (-strukturen) abgelegt werden. „Initial“ bedeutet hierbei die automatische Konfiguration seitens TwinCAT, wenn der Anwender auf der Primär- oder Sekundärseite Ein- oder Ausgangsdaten neu hinzufügt. Das kann auch in den Karteireiter [Startup] nachgesehen werden. Hier werden alle Verknüpfungen der Objektverweise beim Übergang von PREOP zu SAFEOP (P → S) erzeugt. Die beiden „finalen“ CoE Objekte 0x6000 und 0x7000 sind nur „RO“ (read only), da Geräteintern die Kopiervorgänge von Ein- und Ausgangsdaten im OP Zustand durchgeführt werden.

Index	Name	Flags	Value
1000	Device type	RO	0x00001389 (5001)
1008	Device name	RO	EL6695 <SECONDARY>
1009	Hardware version	RO	00
100A	Software version	RO	02
1018:0	Identity	RO	> 4 <
10F4:0	External synchronization status	RO	> 18 <
1608:0	IO RxPDO-Map 8	RW	> 1 <
1608:01	SubIndex 001	RW	0x7000:01, 8
1801:0	TxPDO-Par External Sync Compact	RO	> 6 <
1802:0	TxPDO-Par External Sync	RO	> 6 <
1803:0	TxPDO-Par External Sync(32 Bit)	RO	> 6 <
1A01:0	TxPDO-Map External Sync Compact	RO	> 5 <
1A02:0	TxPDO-Map External Sync	RO	> 8 <
1A03:0	TxPDO-Map External Sync(32 Bit)	RO	> 8 <
1A04:0	Active TxPDOs-Map	RW	> 5 <
1A05:0	FOE Info-Map	RW	> 1 <
1C00:0	Sync manager type	RO	> 4 <
1C12:0	RxPDO assign	RW	> 1 <
1C12:01	SubIndex 001	RW	0x1608 (5640)
1C13:0	TxPDO assign	RW	> 2 <
1C32:0	SM output parameter	RO	> 32 <
1C33:0	SM input parameter	RO	> 32 <
7000:0	PD Outputs	RO	> 1 <
7000:01	SubIndex 001	RO P	0x00 (0)
F000:0	Modular device profile	RO	> 2 <

Abb. 141: Grundlegendes Mapping einer 1 Byte Variable über 0x1C12 bis 0x7000 „PD Outputs“

Übersicht der (initial) Verknüpften Objekte:

- **RxPDO Assign:** 0x1C12 → RxPDO Map 0x1608 → 0x7000 PD Outputs
- **TxPDO Assign:** 0x1C13 → TxPDO Map 0x1A08 → 0x6000 PD Inputs

● Mapping von CoE Objekten

I 0x1608 verweist auf 0x7000 und 0x1A08 auf 0x6000. Die Objekte 0x7000 und 0x6000 sind jeweils auf der anderen Seite der Klemme gespiegelt ansprechbar.

Falls beispielsweise auf der Primärseite ein Objekt 0x7000:0E PD Outputs existiert, so sollte für einen Datenaustausch ein Objekt 0x6000:0E PD Inputs auf der Sekundärseite angelegt sein.

Vorgehensweise unter TwinCAT

Um sich ein selektives Mapping zu erstellen, sollte zunächst die vorhandene PDO Konfiguration mit seinen Zuweisungen deaktiviert werden, indem im Karteireiter „Process Data“ der Klemme, sowohl bei den Out- als auch Inputs und unter Downloads sämtliche Häkchen entfernt werden.

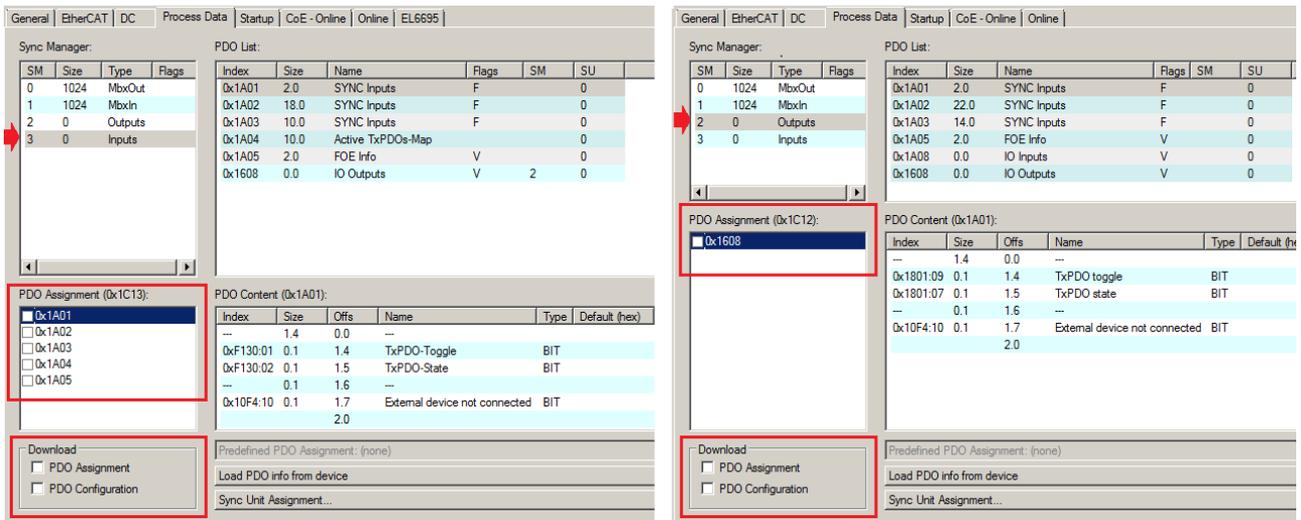


Abb. 142: Prozessdaten beim Selektiven Mapping

Dadurch werden auch die vorhandenen Einträge im Karteireiter Startup gelöscht und die Klemme hat kein Prozessabbild mehr. Dieses wird wie folgt beschrieben in zwei Schritten wieder neu definiert:

Im Karteireiter Startup der Klemme müssen dazu neue Einträge angelegt werden. Dazu ist in diesem Karteireiter unten rechts „New...“ zu betätigen.

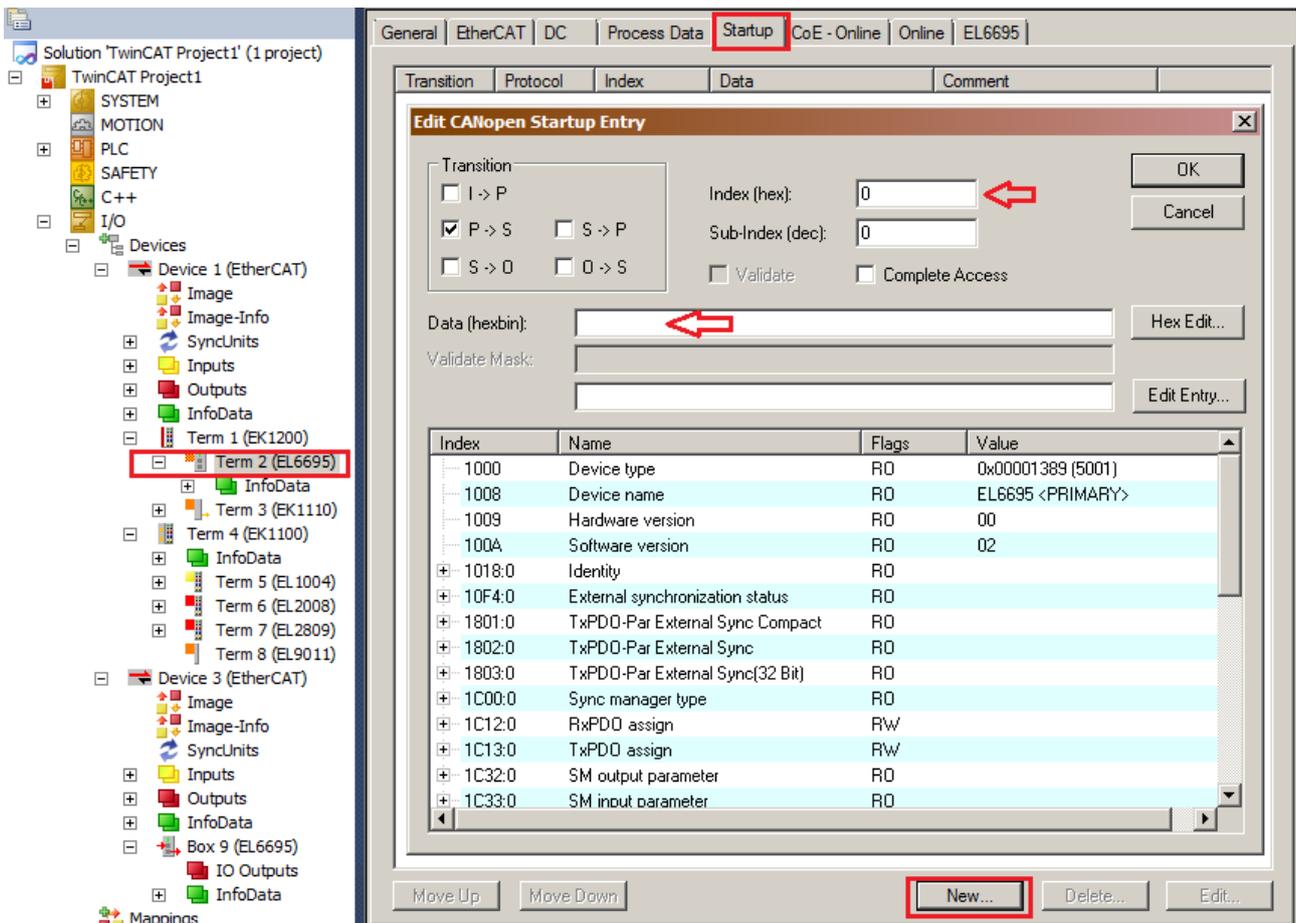


Abb. 143: Anlegen neuer Ausgangsvariablen (in diesem Beispiel: Primärseite der EL6695)

Als Beispiel sollen in diesem Fall insgesamt vier Output Variablen der Größen 8, 16, 32 und wieder 8 angelegt werden [empfohlene Reihenfolge der Vorgehensweise]:

Dazu wird wahlweise (aus dem Bereich 0x1600..0x161F) in diesem Beispiel ein 0x1614 Objekt folgendermaßen eingerichtet (das Häkchen für „Complete Access“ ist zu setzen; als Kommentar z.B. „IO RxPDO-Map“ eintragen):

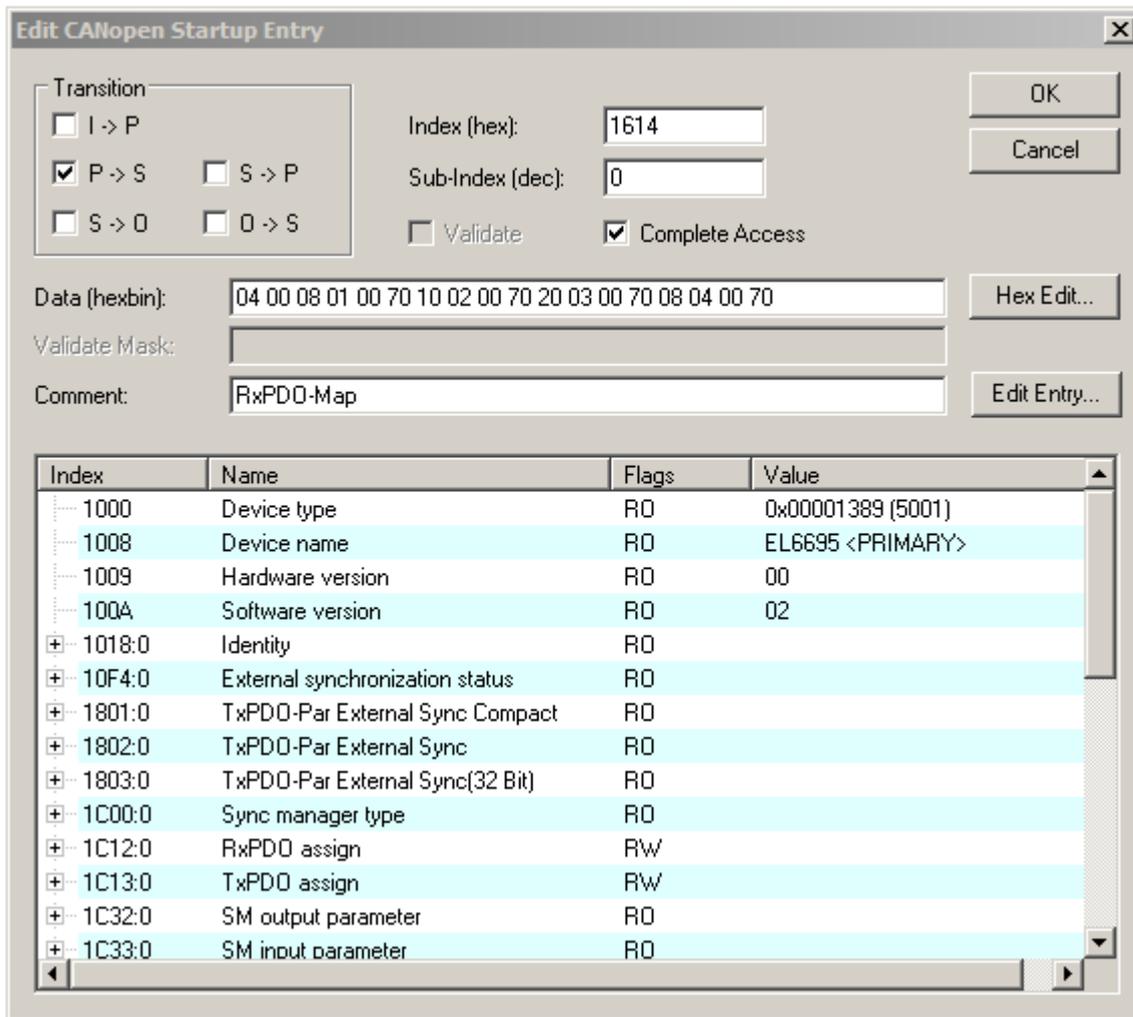


Abb. 144: Einrichten des 0x1614 Objektes

- Einzutragen ist: **04 00 08 01 00 70 10 02 00 70 20 03 00 70 08 04 00 70**

Bedeutung der Eintragsdaten:

04 00	Subindexanzahl
08 01 00 70	Objekt 7000 Index 1 Länge 8
10 02 00 70	Objekt 7000 Index 2 Länge 16
20 03 00 70	Objekt 7000 Index 3 Länge 32
08 04 00 70	Objekt 7000 Index 4 Länge 8

Dies kann z.B. der Datenstruktur eines Programmobjektes oder Struktur repräsentieren. Um diesbezüglich eine Strukturvariable mit einem PDO der EL6695 verknüpfen zu können, ist sie unter TC3 zuvor in einen globalen Datentypen umzuwandeln:

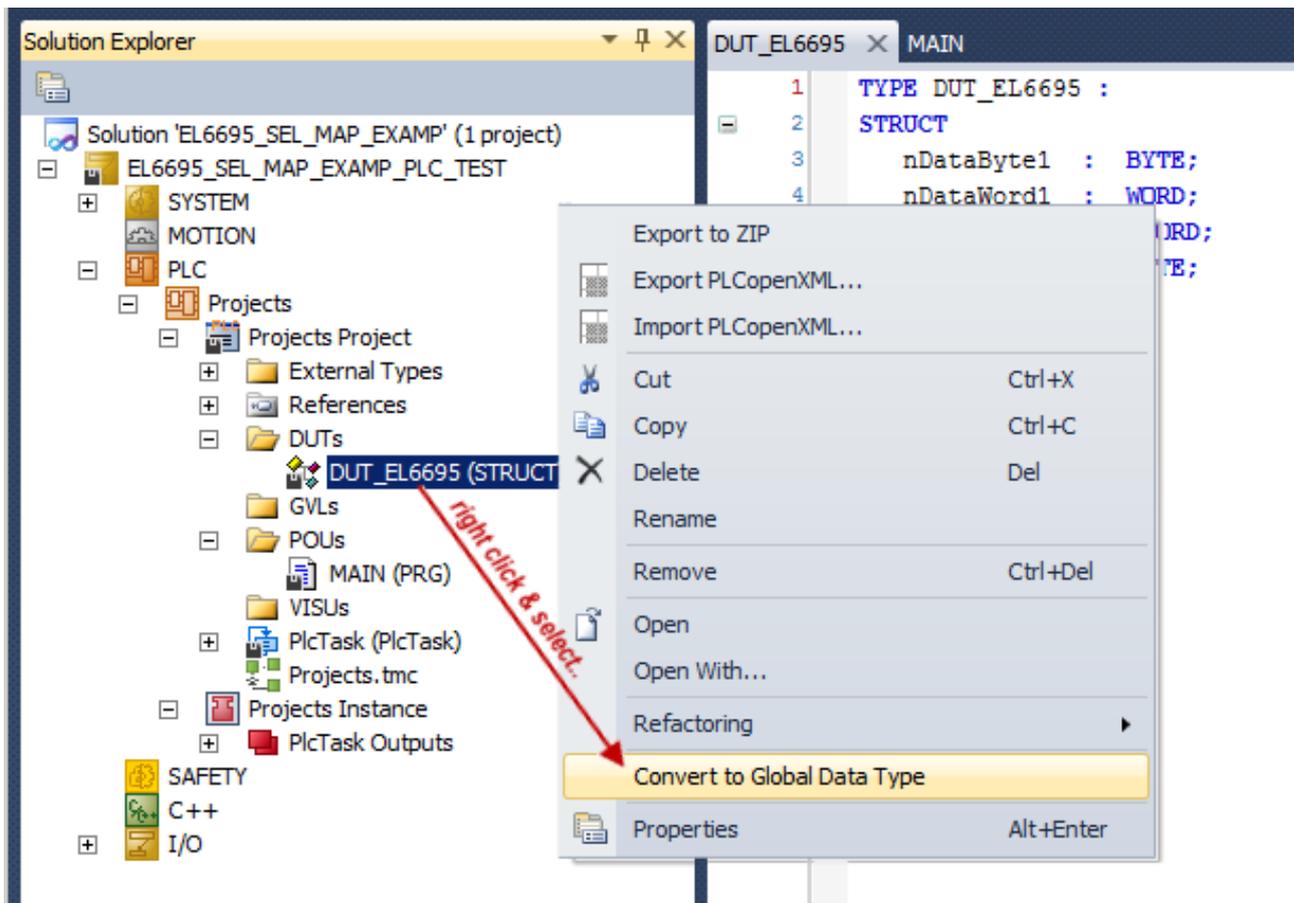


Abb. 145: Konvertierung in einen globalen Datentyp

Im zweiten Schritt sind über den Dialog „Edit CANopen Startup Entry“ nun folgende Einträge gemacht:

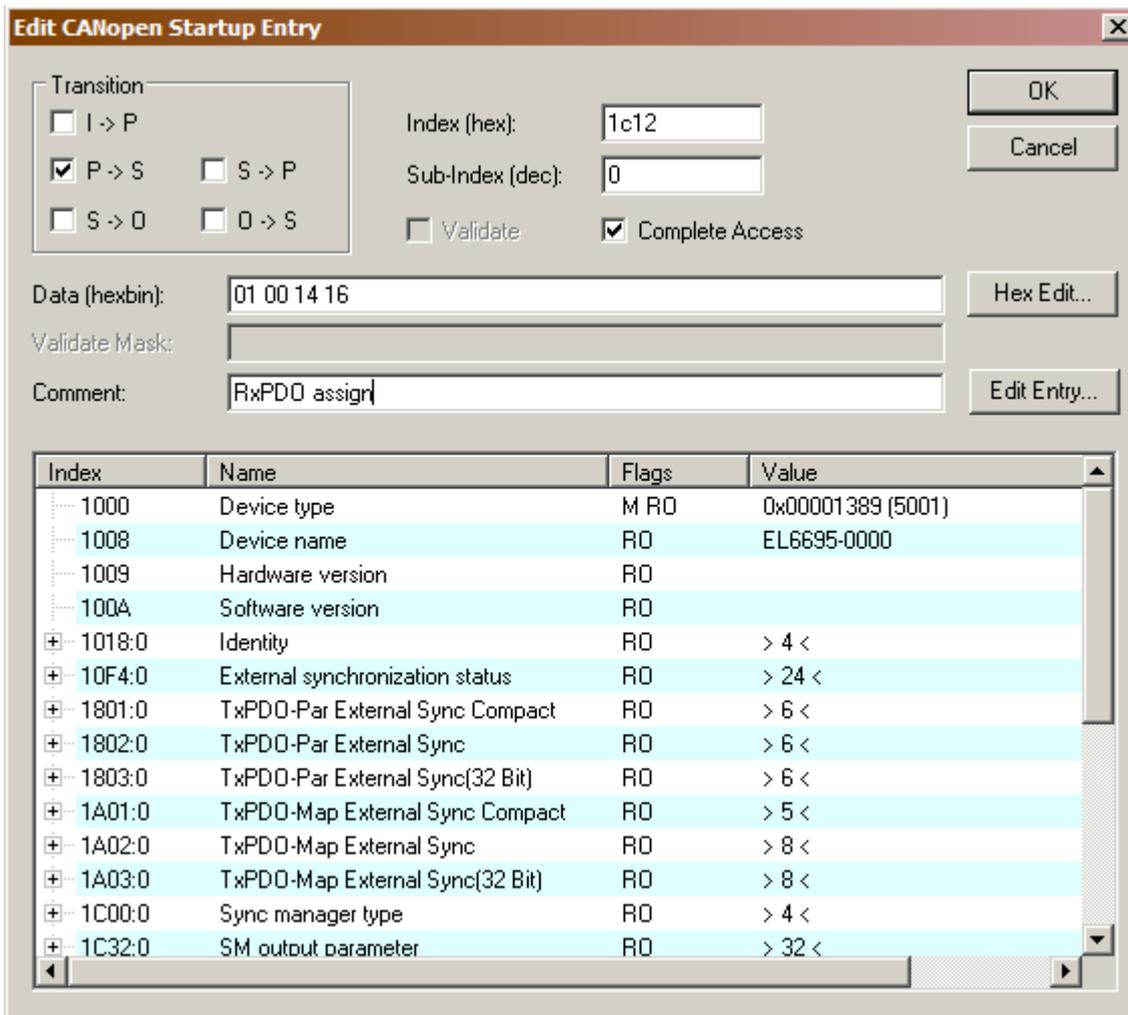


Abb. 146: Anlegen des Objektes 0x1C12 für RxPDO

Der anzulegende CoE Startup Eintrag ist also der Index 0x1C12 mit dem Subindex 0 (Subindex für den Anfang des Objektes). Das Häkchen bei „Complete Access“ ist gesetzt, damit die eingetragenen Werte nicht als ein Datentyp WORD, DWORD etc. interpretiert werden. Die Zuweisungen (Assignments) sollen beim Übergang von PREOP nach SAFEOP erfolgen; dazu bleibt ausschließlich das Häkchen „P → S“ bei Transition gesetzt (oder ist zu setzen). Unter „Comment“ kann ein erklärender Kommentar hinzugefügt werden, der zukünftig in der CoE Übersicht als Name des Objekts dargestellt sein wird. Der Inhalt des CoE Objekts wird in „Data (hexbin)“ angegeben.

Die ersten beiden Blöcke definieren die Anzahl der Subindexe des Objekts (Hi-/Lobyte getauscht: 01 00 → Anzahl Subindex = 1). Darauf folgt der Inhalt des ersten Subindex 01 mit 14 16 → 0x1614.

- Einzutragen ist: **01 00 14 16**

Dies ist nun einer der oben beschrieben Verweise in diesem Fall auf das bereits erstellte „RxPDO Map“ Objekt (0x1614).

Im Karteireiter [Startup] müsste nun folgendes zu sehen sein:

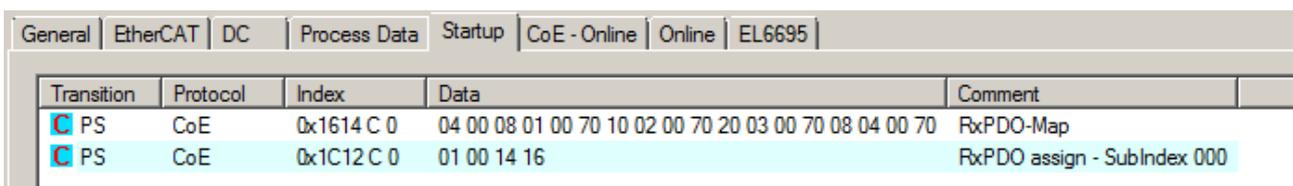


Abb. 147: Neu PDO-Map und Zuweisungs- Objekte auf der Primärseite

Entscheidend ist, dass das „RxPDO assign“ über Objekt 0x1C12 stets am Ende der Liste von RxPDO-Map Einträgen steht. Es kann auch mehrere Verweise enthalten (je nach Bedarf). Ein Datensatz würde dann z.B. so aussehen: 06 00 03 16 1A 16 1E 16 08 16 04 16 ..usw.

Durch nachfolgende Aktionen werden die angelegten Variablen nun in der Konfiguration angezeigt:

1. Auswahl im Menü: [TwinCAT] → [Reload Devices]
2. „Load PDO info from device“ im Karteireiter „Process Data“ der Klemme
3. Anzeigen der CoE Online Objekte mittels „Update List“

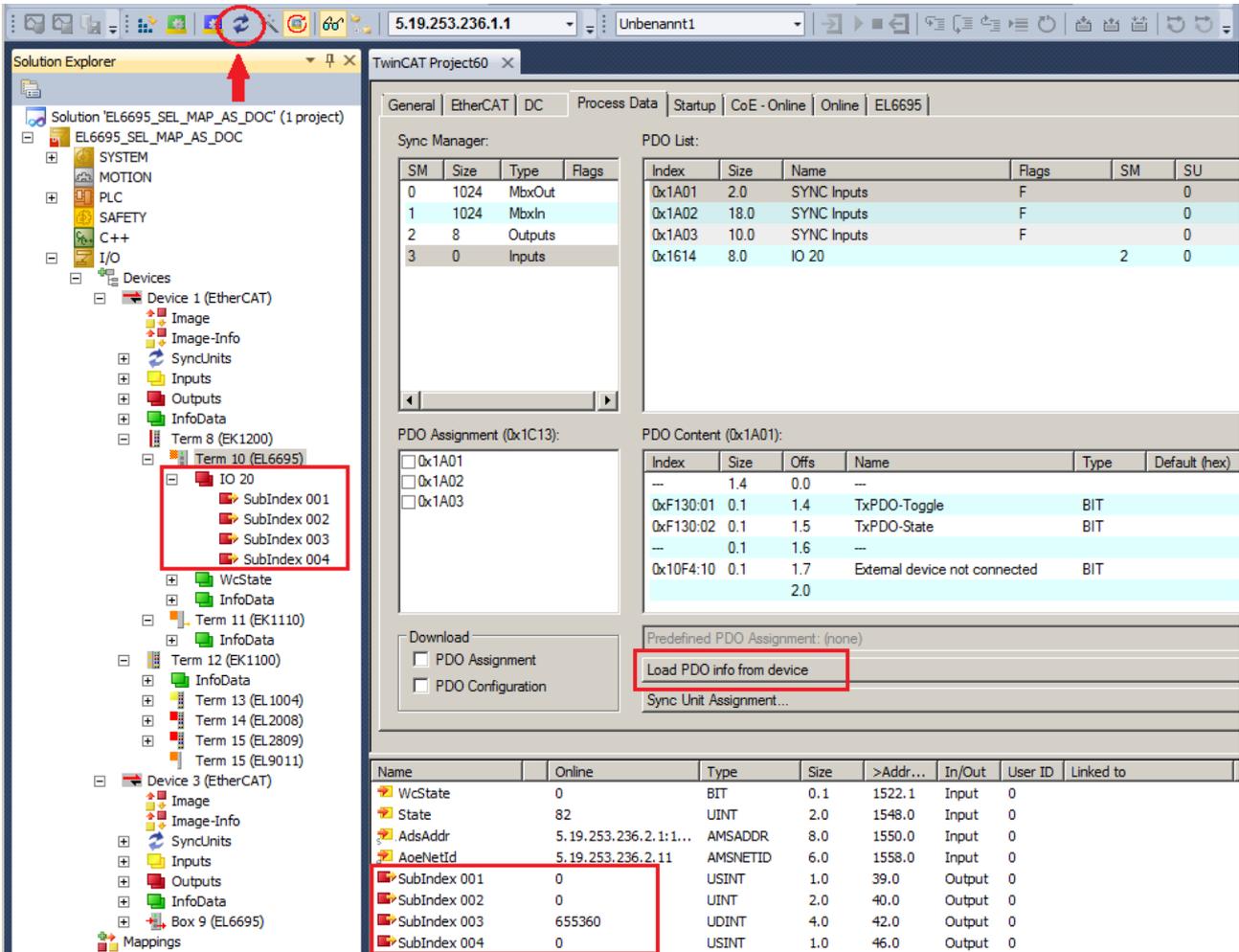


Abb. 148: Neu angelegtes Prozessdaten:Objekt 0x1608

Index	Name	Flags	Value
+ 1803:0	TxPDO-Par External Sync(32 Bit)	RO	> 6 <
+ 1C00:0	Sync manager type	RO	> 4 <
- 1C12:0	RxPDO assign	RW	> 1 <
- 1C12:01	SubIndex 001	RW	0x1614 (5652)
+ 1C13:0	TxPDO assign	RW	> 0 <
+ 1C32:0	SM output parameter	RO	> 32 <
+ 1C33:0	SM input parameter	RO	> 32 <
+ F000:0	Modular device profile	RO	> 2 <
- F008	Code word	RW	0x00000000 (0)
+ F010:0	Module list	RO	> 1 <
+ F130:0	TxPDO-Parameter	RO	> 2 <
+ F630:0	Active TxPdo Info	RO	> 4 <
+ F640:0	Remote Write Cycles	RO	> 3 <
+ F650:0	FOE Info	RO	> 2 <
+ F800:0	Device Config	RO	> 3 <
+ F820:0	ADS Server Settings	RO	> 2 <
+ F821:0	EL6695 ADS Settings	RO	> 2 <
+ FA20:0	Device Diag	RO	> 29 <
- 1614:0	IO RxPDO-Map 20	RW	> 4 <
- 1614:01	SubIndex 001	RW	0x7000:01, 8
- 1614:02	SubIndex 002	RW	0x7000:02, 16
- 1614:03	SubIndex 003	RW	0x7000:03, 32
- 1614:04	SubIndex 004	RW	0x7000:04, 8
+ 1A01:0	TxPDO-Map External Sync Compact	RO	> 5 <
+ 1A02:0	TxPDO-Map External Sync	RO	> 8 <
+ 1A03:0	TxPDO-Map External Sync(32 Bit)	RO	> 8 <
+ 1A04:0	Active TxPDOs-Map	RW	> 5 <
+ 1A05:0	FOE Info-Map	RW	> 1 <
- 7000:0	PD Outputs	RO	> 4 <
- 7000:01	SubIndex 001	RO P	0x00 (0)
- 7000:02	SubIndex 002	RO P	0x0000 (0)
- 7000:03	SubIndex 003	RO P	0x00000000 (0)
- 7000:04	SubIndex 004	RO P	0x00 (0)

Abb. 149: Darstellung [CoE Online] der selektiv angelegten Variablen mit den entsprechenden Verweisen

Erstellen der selektiven PDO Variablen

Auf der gegenüberliegenden Seite der bridge-Klemme EL6695, in diesem Fall die Sekundärseite, können nun wahlweise bestimmte Variablen Auszugsweise „gespiegelt“ werden. Dazu sind in dem „Solution Explorer“ die entsprechenden gewünschten Variablen hinzuzufügen.

In diesem Beispiel soll von der vorhandenen Struktur auf der Primärseite mit einer 1-2-4-1 Byte Struktur nur der 2-Byte und nur der 4-Byte Variablentyp „WORD“ und „DWORD“ als Auszug der gesamten Prozessdaten, wie gezeigt zur Verfügung gestellt werden:

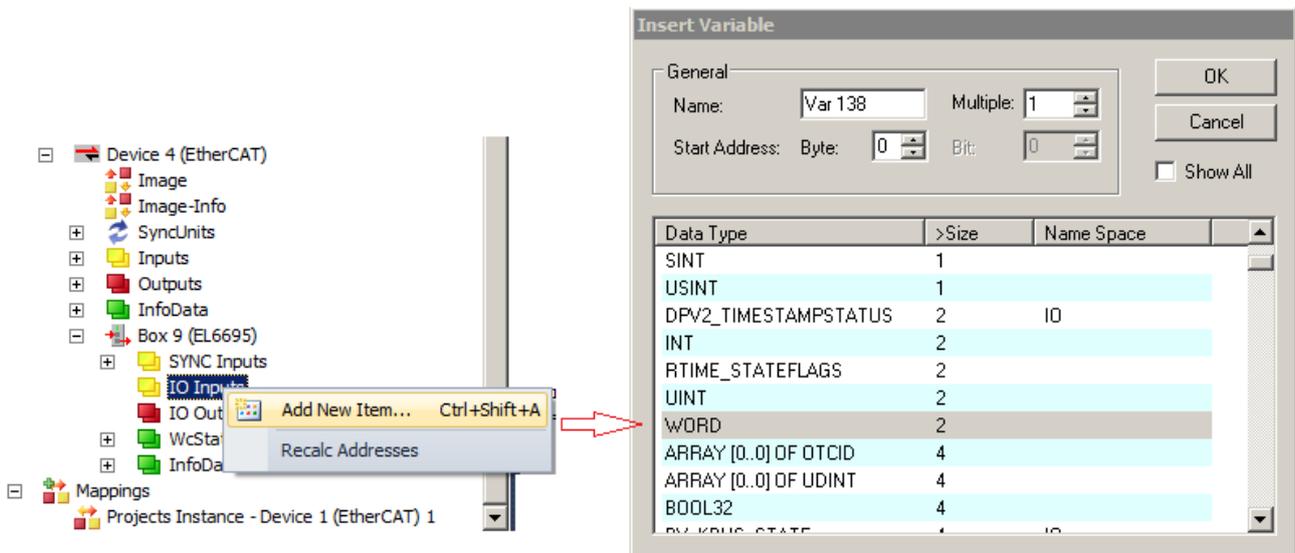


Abb. 150: Hinzufügen selektiver PDO-Variablen

Wie in der Abbildung gezeigt ist auch mit der DWORD-Variablen zu verfahren.

Danach ist wie folgt der Startup-Eintrag wie folgt zu ergänzen, damit die Subindex-Werte 02 für den WORD-Typ und 03 für den DWORD-Typ angepasst werden:

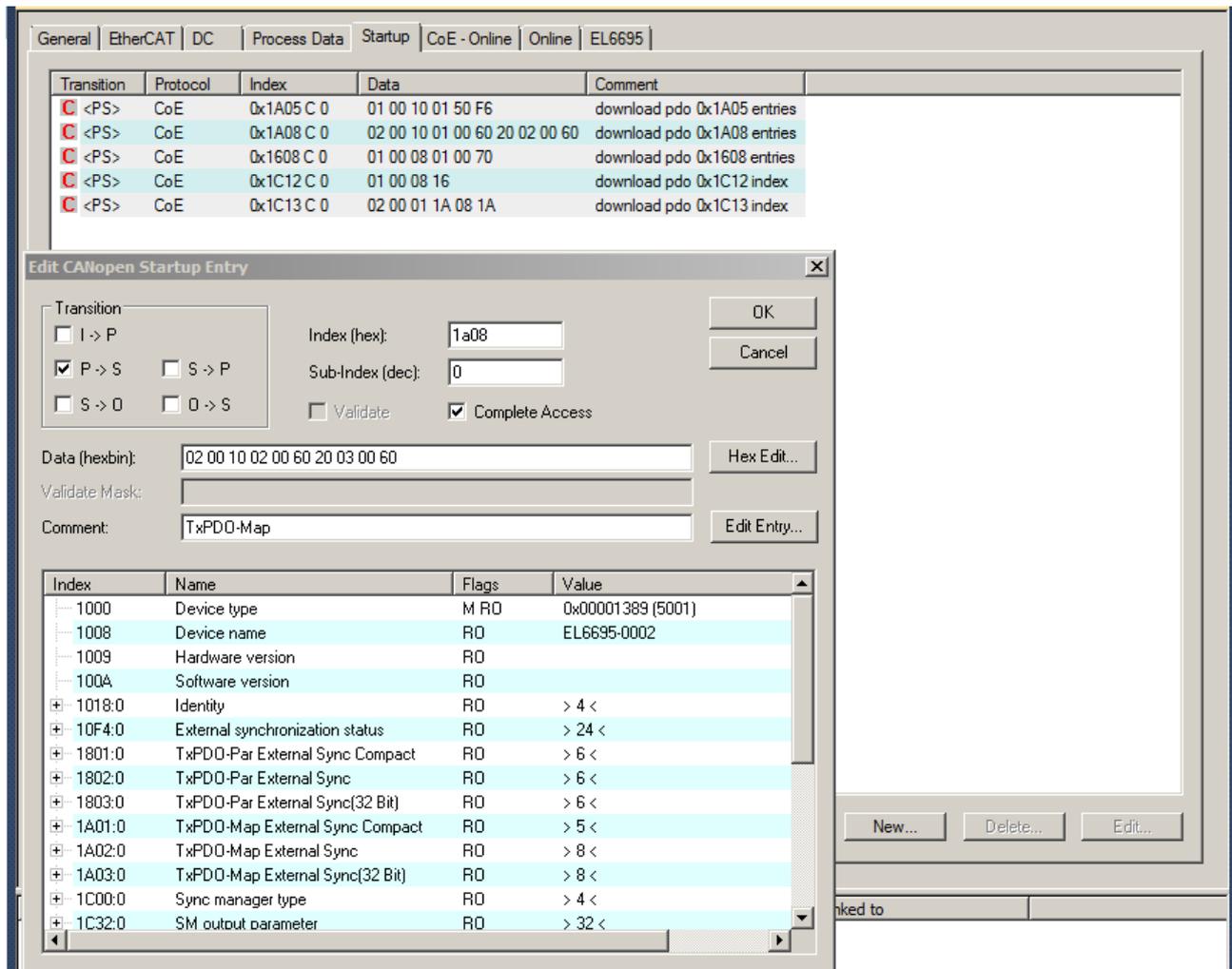


Abb. 151: Startup Eintrag 0x1A08 zum Überschreiben (zur Korrektur der Verweisindex 01 → 02 und 02 → 03)

- Einzutragen ist: **02 00 10 02 00 60 20 03 00 60**

Bedeutung der Eintragsdaten:

02 00	Subindexanzahl
10 02 00 60	Objekt 6000 Index 2 Länge 16
20 03 00 60	Objekt 6000 Index 3 Länge 32

Nach erneuter „Reload“ Anweisung und Aktivierung der Konfiguration stehen das selektive PDO-Mapping nun dem Anwender zur Verfügung und kann von den PLC Programmen verwendet werden. Zu beachten ist allerdings, dass Aufgrund Klemmenspezifischer Systemeigenschaften für den Echtzeitdatenaustausch ein speziellen Puffermechanismus verwendet wird, so dass die Output-Variablen immer mit einem Byte am Anfang und einem Byte am Ende konfiguriert werden müssen, den sog. Start/Stop-Bytes. Verknüpfungen zu PLC Variablen oder Ausgängen und damit eine Daten-Nutzung sind für diese „fillbyte“-Variablen jedoch nicht nötig.

Beim Einsatz der EL6695 unter anderen EtherCAT-Mastern sind die Konfigurationsschritte entsprechend anzupassen.

Erläuterung zur Start/Stop Byte Verwendung

Beckhoff empfiehlt generell, beim selektiven Mapping in den Output-Daten ein Start-Byte am Anfang und ein Stop-Byte am Ende anzulegen. Der technische Grund liegt in der internen Speicherverwaltung der Prozessdaten. Der Zugriff auf das letzte Byte bewirkt dort eine interne Umschaltung auf den nächsten EtherCAT Puffer der Klemme; ab dann kann aus diesem Output-Paket nicht mehr gelesen werden. Wenn die Variablen selektiv „von oben nach unten“ aus dem Output-Buffer gelesen werden, ist dies kein Problem, wenn hingegen (wie vermutlich in den meisten Anwendungen der Fall) nacheinander aus verschiedenen Bereichen gelesen wird und dabei auch das letzte Byte erfasst werden soll, ist von da an der Puffer jedoch gesperrt und es können keine weiteren Daten mehr geholt werden. Ein zusätzliches „leeres“ Byte nach den eigentlichen Prozessdaten verhindert sodann eine unbeabsichtigte Umschaltung des Puffers.

Diese Abbildung zeigt dass in vielen Zugriffsfällen das Start/Stop-Byte gar nicht nötig wäre:

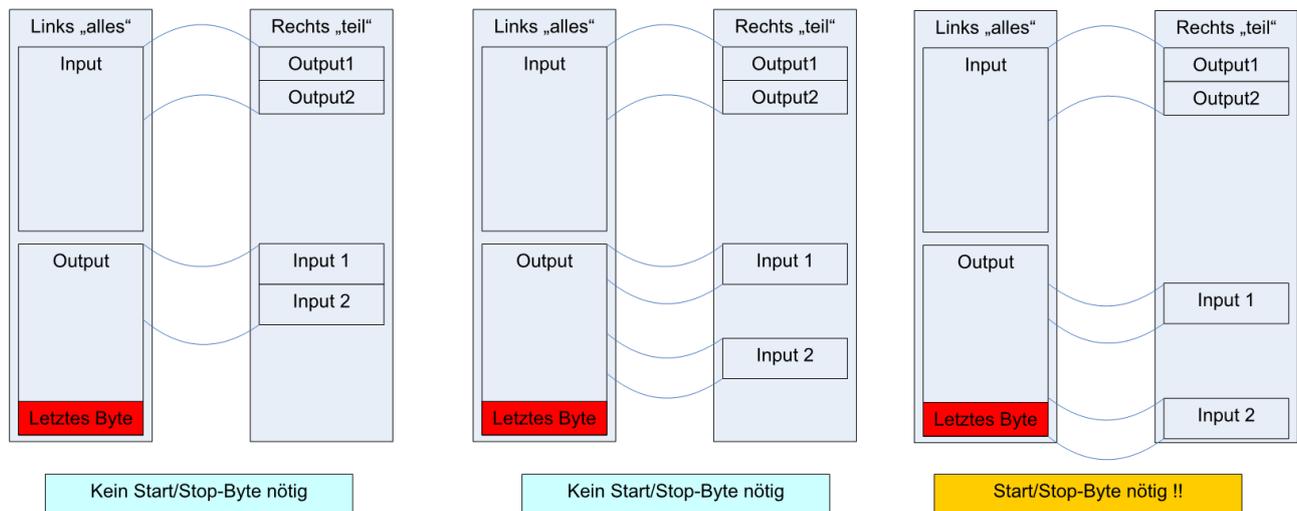


Abb. 152: Start-/ Stop Byte Verwendung

● Einfügen von Start und Stop-Byte

i Damit bei der flexiblen Nutzung der EL6695 während der Lebensdauer der Applikation und eventueller Konfigurationsänderung keine unerwarteten Überraschungen auftreten, empfiehlt Beckhoff generell die Output-Variablen immer mit einem Byte am Anfang und einem Byte am Ende zu konfigurieren.

i Verwendung der Beispielprogramme

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.

→ Beispielprogramme Selektives Mapping auf Anfrage:

<https://infosys.beckhoff.com/content/1031/el6695/Resources/1421824651.zip>

<https://infosys.beckhoff.com/content/1031/el6695/Resources/1421826315.zip>

6.4.4 FSoE-Übertragung

FSoE-Übertragung wird als zyklische Prozessdatenübertragung realisiert und kann auf zwei verschiedenen Wegen umgesetzt werden:

- Erstellen einer ESI mit Safety-Modulen/slots und laden in den EmulationMode, siehe dort
- Die Safety-Connection für ein „External Device“ konfigurieren
- Hinweis: die EL6695 ist kein zulassungspflichtiges sicherheitsgerichtetes Gerät, sondern „nur“ ein Element der Übertragungsstrecke. Soll allerdings der ETG-FSoE-Test gegen ein Safety-Gerät laufen, daß sich „hinter“ der EL6695 versteckt, ist natürlich die PDO-Konfiguration der EL6695 so zu gestalten daß der FSoE-ETG-Test die EL6695 als „Safety“-Gerät erkennt.

Für weiterführende Informationen richten Sie bitte eine Anfrage an den Beckhoff Support, unter Angabe hilfreicher Daten (TwinCAT Version, geforderte Performance Werte, Zykluszeit, Datenmengen etc.)

6.5 Lokale Mailbox Protokolle

Azyklische Daten die lokal von der EL6695 verarbeitet und nicht weitergeleitet werden. Werden i.d.R. über die EtherCAT Mailbox transportiert.

6.5.1 CoE - Can over EtherCAT

CoE - Can over EtherCAT

Verarbeitung: lokal, keine Weiterleitung

Innerhalb eines PLC-Programms (ST) kann über den Funktionsbaustein **FB_EcCoeSdoRead** per SDO (Service Daten Objekt) -Zugriff Daten aus dem Objektverzeichnis eines EtherCAT Slaves ausgelesen werden. Mittels der Parameter nSubIndex und nIndex wird ausgewählt, welches Objekt ausgelesen werden soll. Dieser Baustein gehört zur (standard-) Bibliothek Tc2_EtherCAT (siehe TwinCAT 3 Engineering → PLC → Bibliotheken → TwinCAT 3 PLC Lib: Tc2_EtherCAT → CoE unter <http://infosys.beckhoff.com>).

● 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.

→ Beispielprogramm CoE auf Anfrage:

<https://infosys.beckhoff.com/content/1031/el6695/Resources/1421141259.zip>

6.6 Übertragungsfähige Mailbox Protokolle

6.6.1 AoE Anwendung beim CoE Zugriff

AoE Anwendung beim CoE Zugriff

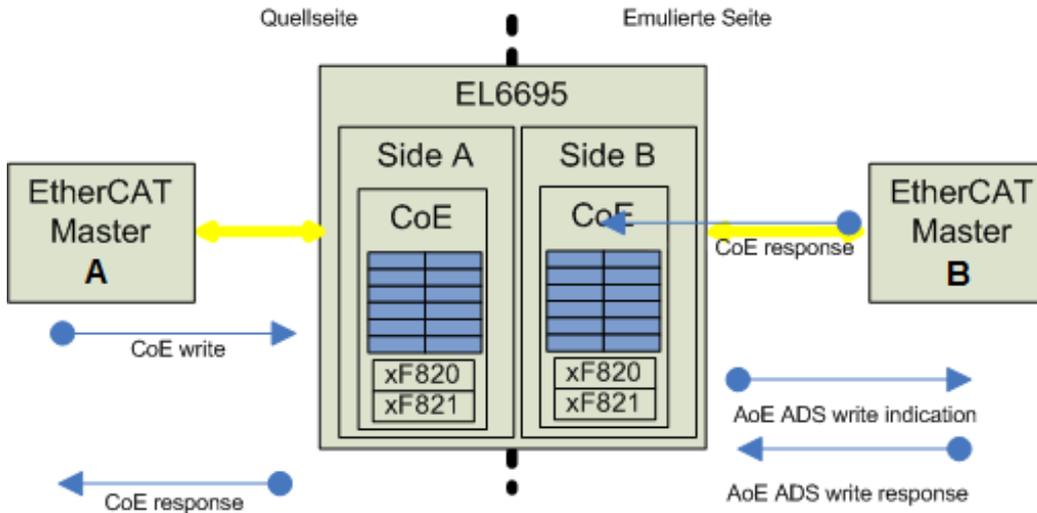


Abb. 153: Darstellung eines CoE Schreibzugriffs mit Bedienung per AoE-ADS

- Die EL6695 trägt auf beiden Seiten ein CoE-Verzeichnis, wobei mind. eine Seite Anwenderspezifisch deklariert ist. Grundsätzlich baut die EL6695 in diesen Modus keine Verbindungen zwischen CoE A und CoE B auf. Die beiden Steuerungen könnten also nicht über CoE miteinander azyklisch Parameter austauschen und ein Schreibzugriff auf der einen Seite kann den anderen Master nicht informieren. Deshalb ist in der EL6695 folgender Datenaustauschmechanismus über Beckhoff ADS implementiert:
 - Annahme: mind. 1 Seite ist ein TwinCAT-System oder ein EtherCAT Master der Beckhoff AoE (ADS over EtherCAT) unterstützt. In diesem Beispiel sei Master B ein solches System, Master A unterstützt nur CoE.
 - Ein Schreib/Lesezugriff von Master A auf CoE A wird nach CoE B übertragen.
 - Die EL6695 entnimmt aus CoE B 0xF820 die Ziel-Ams-NetID und sendet eine AoE ADS write/read Indication an die angegebene Adresse. Dort muss ein Dienst diese Indication aufnehmen und bearbeiten. Dies kann z.B. in der TwinCAT PLC ein FB ADSWRITEIND/ADSREADIND aus der TcSystem.lib sein.
 - Aus der PLC ist dann mit ADSWRITERES/ADSREADRES die Antwort an die EL6695 zurückzusenden. Als Zieladresse ist aus CoE A 0xF821 die eigene AMS Net ID zu verwenden. Diese muss vom EtherCAT Master beim Start als AoE Init Cmd in die EL6695 geladen werden.
 - Die EL6695 kann nun an den Master A die CoE write/read response zurücksenden
 - Bei diesem Vorgang ist die Zeitdauer der Übertragung zu berücksichtigen.
 - Dieser AoE-Mechanismus arbeitet in den CoE-Bereichen wo auch AMS NetIds hinterlegt sind.

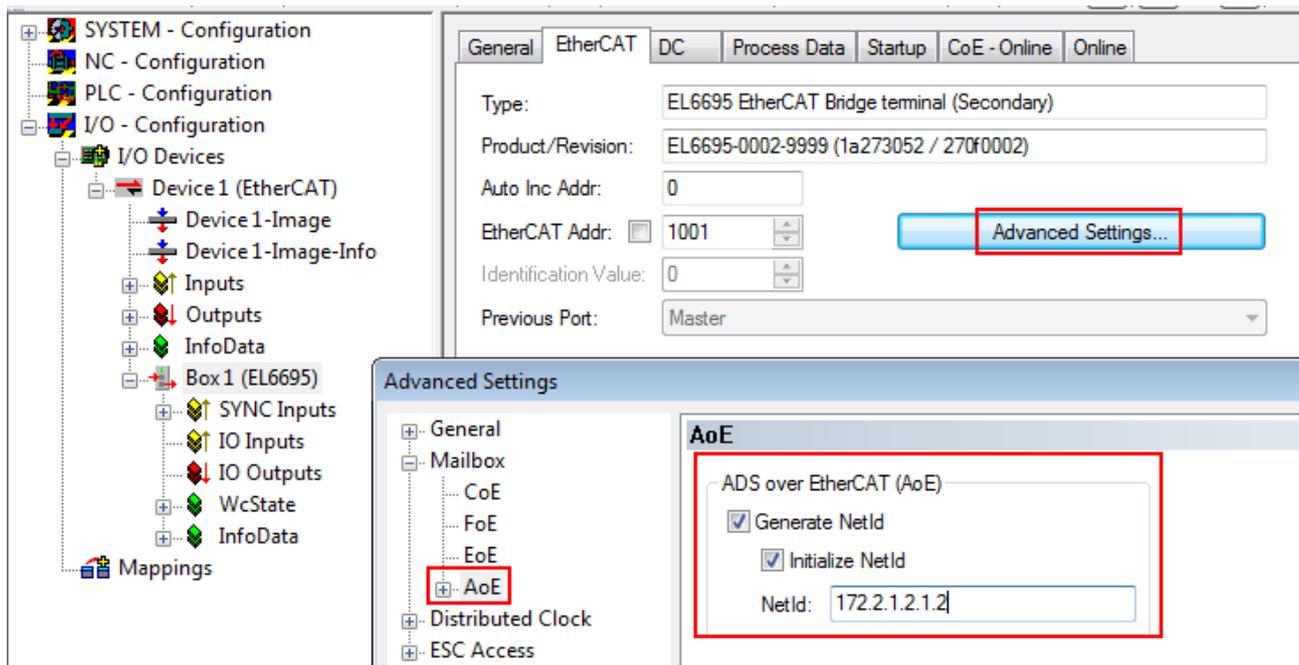


Abb. 154: Übernahme der NetID

● Hinterlegen von AMS Adressen notwendig

i Bei Verwendung eines partielles Objektverzeichnis, bei dem Teile des OD von anderen Teilnehmern ausgetauscht und in das OD eingefügt werden, sind zu den jeweiligen (zusätzlichen) Objekten entsprechende AMS Adressen zu hinterlegen (über das Objekt 0xF820 ▶ 123]).

- Modus „partielles Objektverzeichnis“:
 - Teile des OD werden runtergeladen und fügen sich in das OD ein.
- Modus „partielles Objektverzeichnis mit gespiegelten Objekten“:
 - Teile des OD werden in die EL6695 geladen und fügen sich in das OD ein. Außerdem werden im Remote OD gespiegelte Objekte angelegt. Dieser Modus ermöglicht eine direkte Datenverbindung zwischen CoE A und CoE B ohne den o.a. AoE-Mechanismus. Dieser Modus erstreckt sich allerdings nur über Parameter-Bereich 0x8nnn:
 - CoE A 0x8000 – 0x87FF wird gespiegelt auf CoE B 0x8800 – 0x8FFF
 - CoE B 0x8000 – 0x87FF wird gespiegelt auf CoE A 0x8800 – 0x8FFF
 - Die 0x88xy Objekte sind demzufolge „Read Only“ (RO)

● Veränderte CoE Objekte

- i**
- Im CoE Objekt 0x10E1 „OEM Device Name“ befindet sich der Original Name als String „Beckhoff EL6695“. Dies ist nicht veränderbar.
 - Das OD definiert auch die Prozessdaten über 0x1C13 und folgende. Dabei muss die Prozessdatenlänge auf Quell- und emulierter Seite gleich groß sein!
 - Das OD wird in der EL6695 stromausfallsicher gespeichert, max. Größe 128 kByte je Seite.
 - Die Quellseite muss online/präsent sein und somit das Verhalten der EL6695 definiert haben bevor auf die emulierte Seite von Master B zugegriffen werden kann. Sollte Master B zuerst starten, wird die EL6695 mit PreOP_ERR reagieren das die Prozessdaten nicht definiert sind. Nachdem dann Master A gestartet und die EL6695 definiert ist, muss Master B ein clear ERR durchführen und Seite B erneut in OP starten. Das Verhalten ist über das CoE Objekt 0xF800:02 Bit 1 (MASK_RUN_PRI_UNC) und Bit 2 (MASK_RUN_SCND_UNC) einstellbar (siehe Anhang).
- Der Modus „komplett“, „partiell“, „gespiegelt“ wird über den Inhalt der *.coe -Datei in der EL6695 gesetzt

6.6.2 EoE - Ethernet over EtherCAT

Verarbeitung: Weiterleitung

Unterstützung unterschiedlicher Mailbox-Größe primär/sekundärseitig

Da TwinCAT mit der angeschlossenen EtherCAT-Umgebung als virtuelle Netzwerkkarte auftritt, übernimmt Windows das Routing von IP-Frames (z.B: 192.168.2.1) zur EL6695, um die Frames in das gegenüberliegende EtherCAT-System zu transportieren. Grundlagen: siehe Dokumentation zur EL6601/EL6614.

Damit EoE in der EL6695 aktiviert ist, muss im EtherCAT Reiter der Klemme unter den erweiterten Einstellungen unter EoE der Haken bei Virtual Ethernet Port gesetzt sein und Switch Port ausgewählt sein.

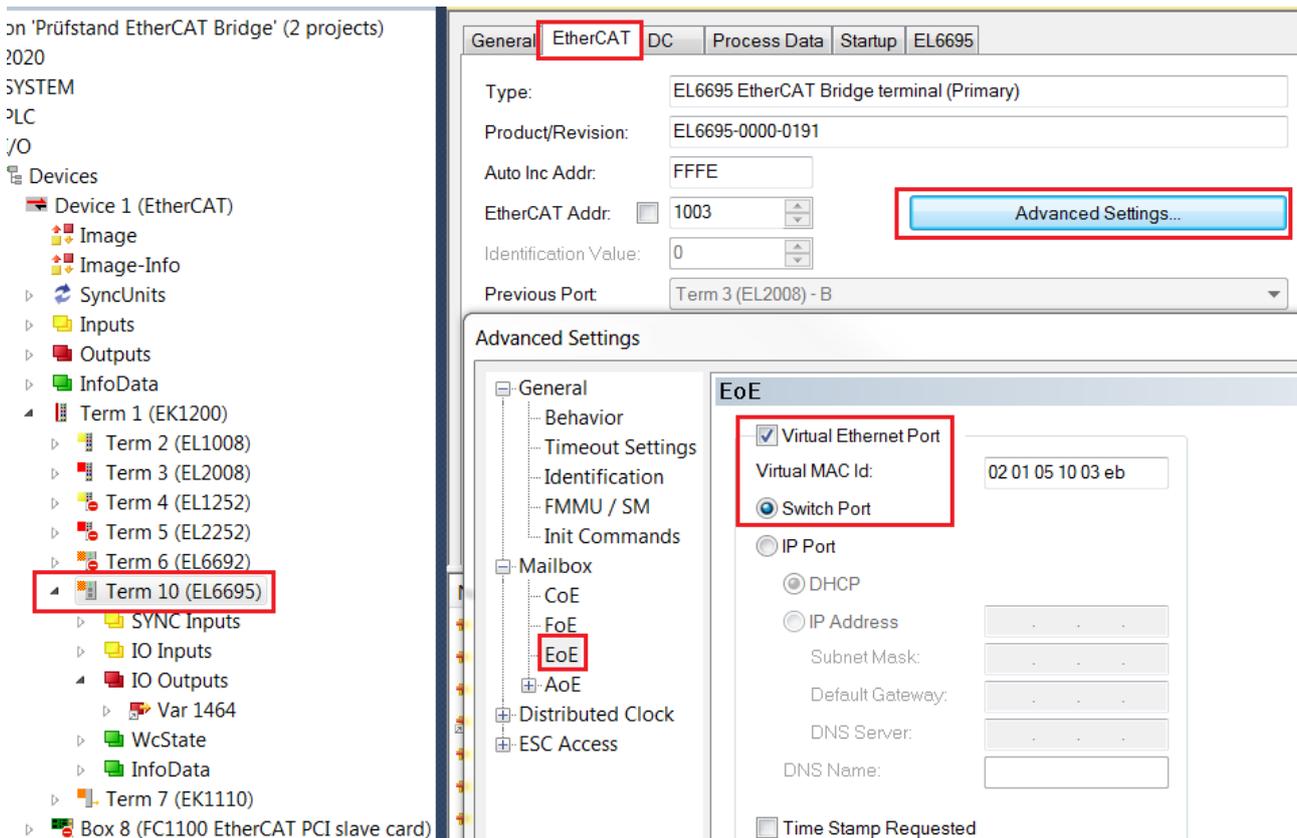


Abb. 155: EoE Konfiguration

i Aktivierung von EoE über Objekt 0xF800 (Device Config)

Die Aktivierung von EoE ist ggf. über das CoE Objekt der Klemme 0xF800 (Device Config) zu überprüfen. Sollte in dem Eintrag 0xF800:01 (Config 1) das Bit 14 gesetzt sein (z.B. der Wert 0x4000 eingetragen sein), so werden EoE-Mailboxprotokolle blockiert (siehe Anhang). Zur Aktivierung ist das Bit auf Null zu setzen bzw. es kann der komplette 16-Bit Wert auf 0x0000 gesetzt werden. Alternativ kann dies auch durch einen Startup Eintrag geschehen (siehe [Abbildung 155](#)).

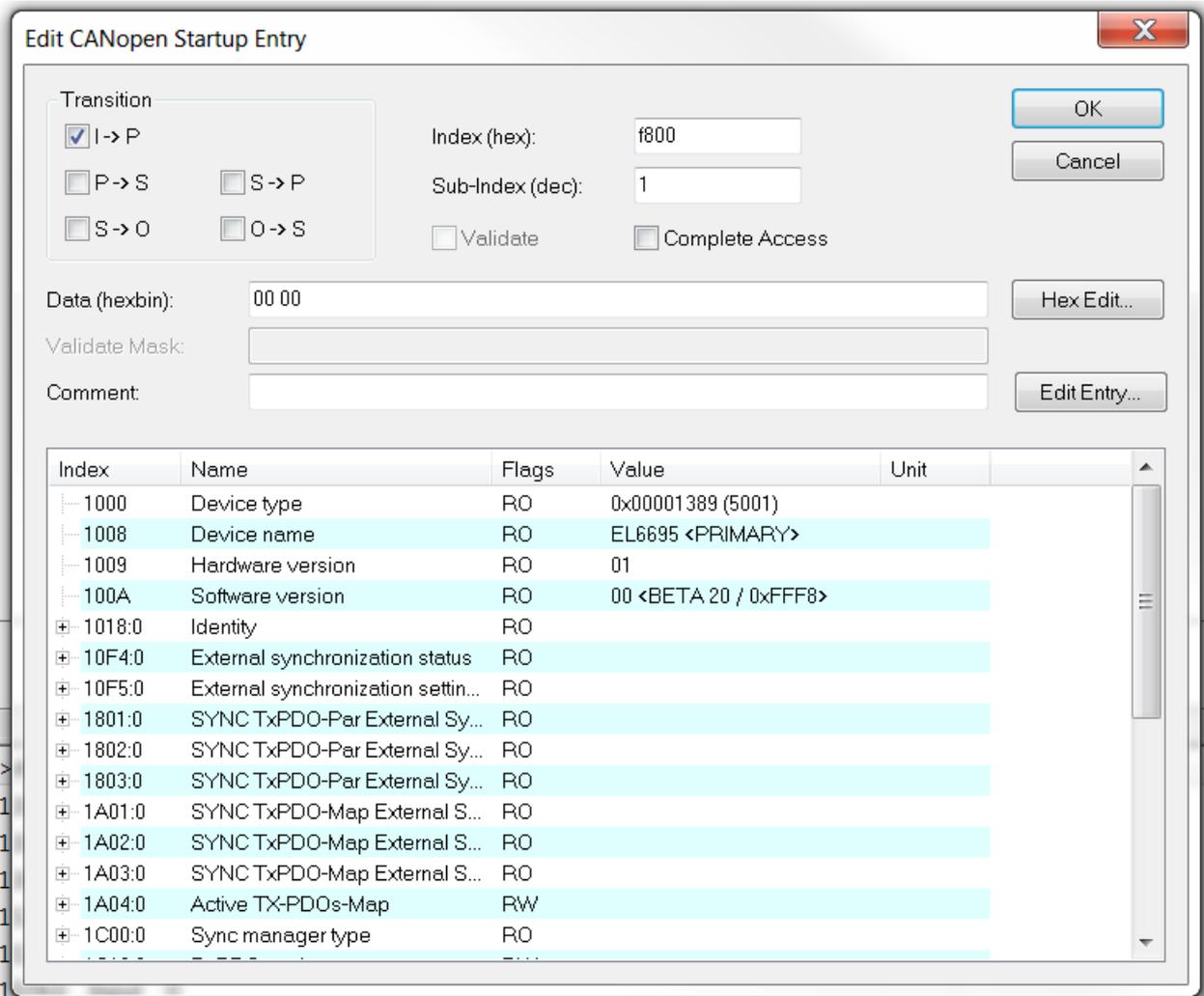


Abb. 156: Prüfung des korrekten Wertes (0x0000) von Objekt 0xF800: EoE Aktiviert

Für eine funktionierende EoE Kommunikation müssen die IP Adressen beider EtherCAT Master im selben Subnetz sein.

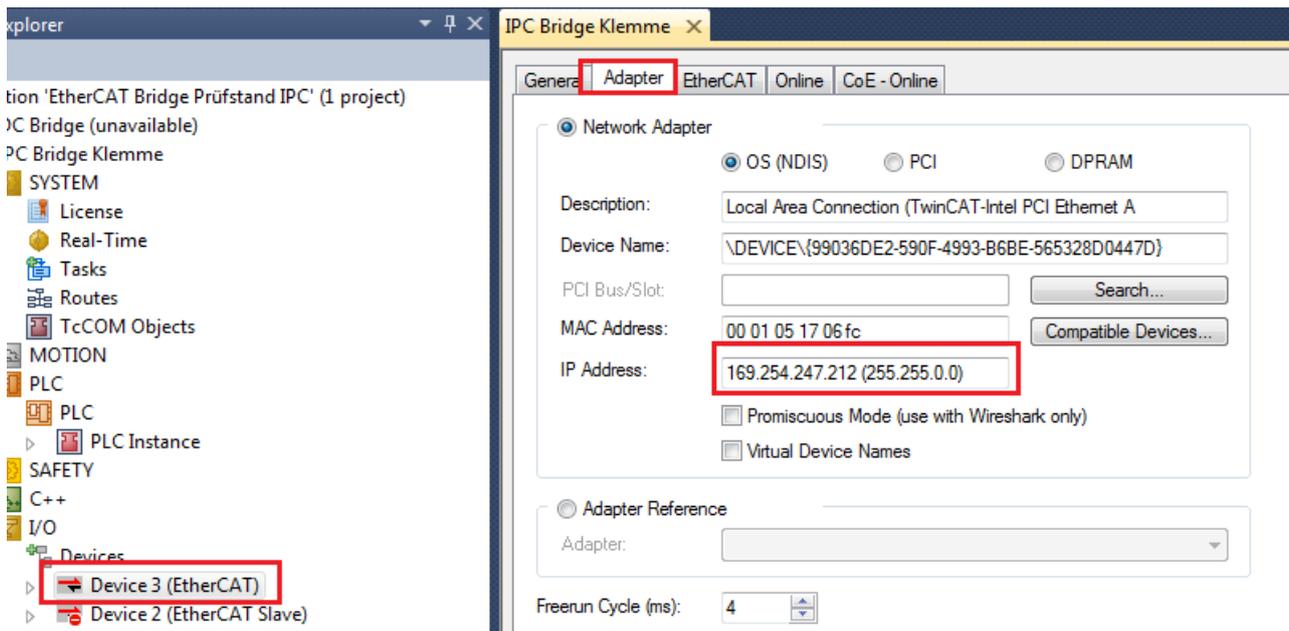


Abb. 157: Prüfung der EoE IP Adresse

Master 1 hat im Beispiel die IP 169.254.247.212 und das Subnetz 255.255.0.0.

Master 2 kann jetzt die IP 169.254.249.204 haben muss aber auch im Subnetz 255.255.0.0 sein.

Anschließend kann man über die Kommandozeile von Master 1 Master 2 über den „ping“-Befehl ansprechen.

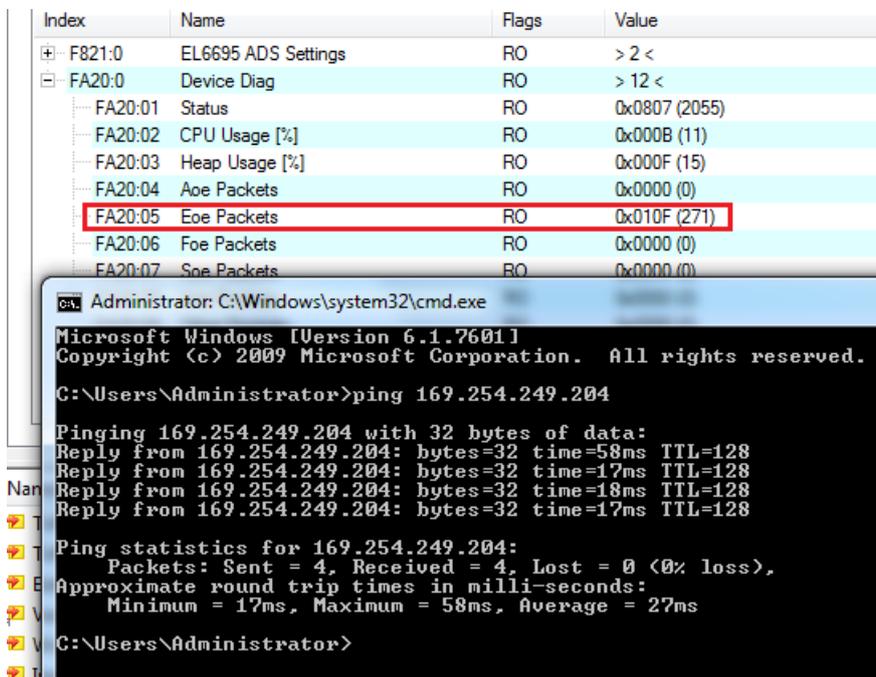


Abb. 158: Ausführung von „ping“ des Master1 nach Master2 über Kommandozeile

In den CoE Objekten der EL6695 werden dann im Objekt 0xFA20:05 die EoE Pakete hochgezählt. Bei diesem Beispiel Ping Befehl werden 4 Pakete gesendet.

6.6.3 AoE - ADS over EtherCAT

AoE - ADS over EtherCAT

Verarbeitung: Weiterleitung

Unterstützung unterschiedlicher Mailbox-Größe primär/sekundärseitig

ADS (Automation Device Specification) ist ein von Beckhoff entwickeltes und offengelegtes Protokoll, um Hard- oder Softwarebasierte Geräte Daten miteinander austauschen lassen zu können. Der Aufbau dieser ADS-Telegramme ist im Beckhoff Information System oder den ETG Standards einsehbar. Die EL6695 kann ADS-Telegramme über AoE (ADS over EtherCAT) direkt, ohne zugrundeliegenden IP-Kanal, zur Gegenseite transportieren. Dabei muss auf beiden Seiten die Kommunikation über AoE aktiviert sein.

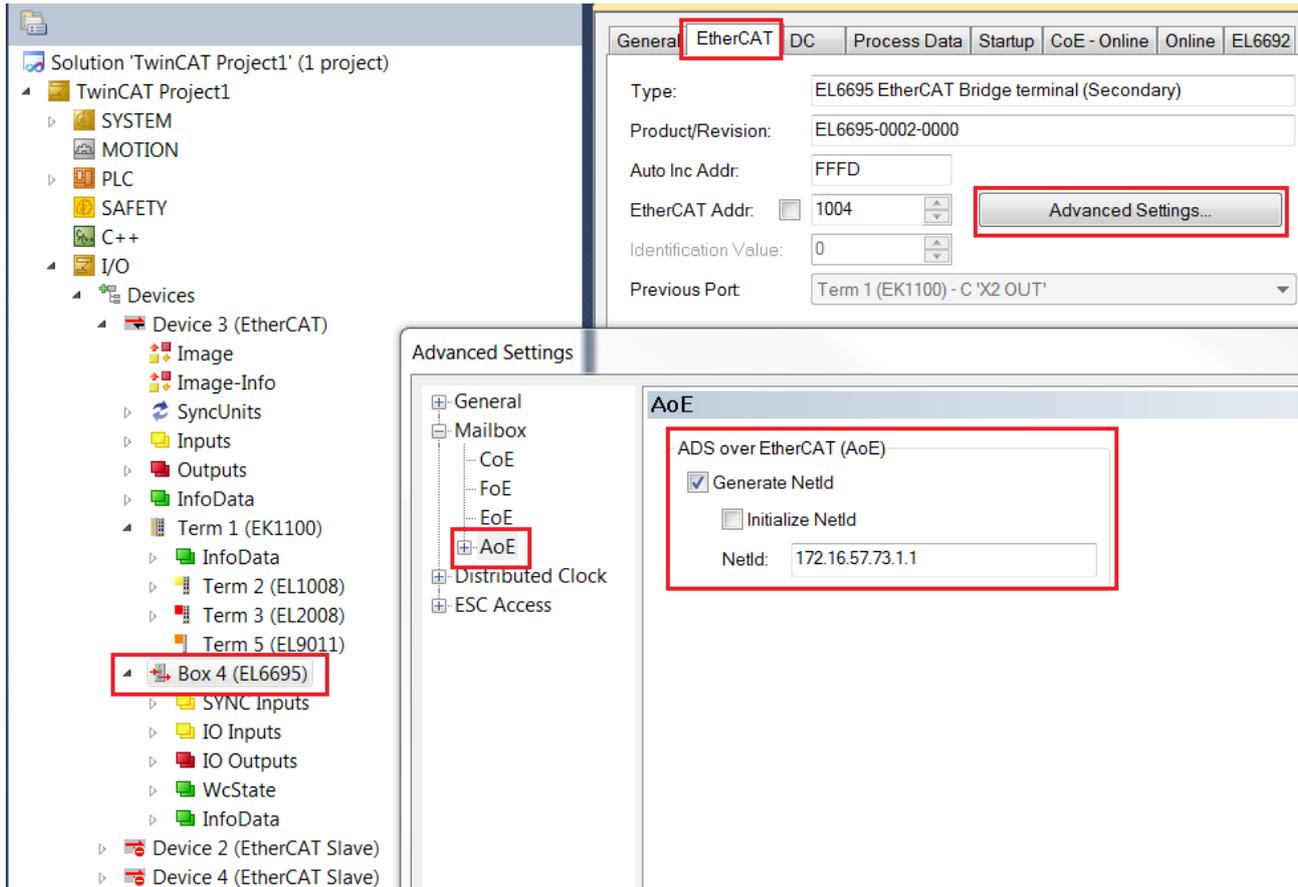


Abb. 159: Einstellungen für das ADS Frame Routing

Damit TwinCAT bekannt ist, wie oder über welchen Kanal es die ADS Frames routen soll, muss als NetID die Adresse des jeweils anderen Systems eingestellt werden. Das Beispiel oben ist somit aus Sicht von PC2.

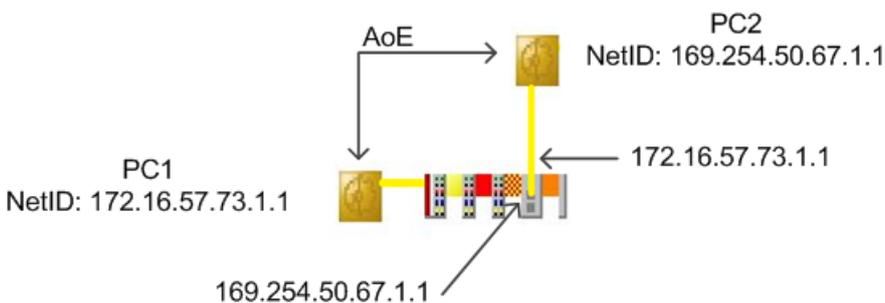


Abb. 160: Einstellung der NetID für AoE – ADS over EtherCAT

i **Verwendung der Beispielprogramme**

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.

→ Beispielprogramm AoE auf Anfrage:

<https://infosys.beckhoff.com/content/1031/el6695/Resources/1421139595.zip>

6.6.4 FoE – Filetransfer over EtherCAT

FoE – Filetransfer over EtherCAT

Hierbei wird im Wesentlichen ein Zugriff auf Daten über zwei referenzierte Speicherbereiche der jeweiligen beteiligten Master (PLCs) ermöglicht. Die Daten können von der Primärseite (E-Bus Seite) zur Sekundärseite (RJ45-Anschluß) oder umgekehrt als ein Art Stream transportiert werden.

Die maximale Größe je Speicher beträgt 32768 Bytes.

Die FoE-Verarbeitung in der EL6695 ist abhängig vom EtherCAT Status:

- BOOT: FoE wird zum lokalen FW-update genutzt
- alle anderen States (INIT, PreOP, SafeOP, OP): in Abhängigkeit vom CoE-Passwort
 - kein Passwort: FoE weiterleiten auf die Gegenstelle
 - mit Passwort: lokales Abspeichern je nach Passwort. Damit werden Sonderfunktionen bedient.

i FoE-Request / kein Passwort

Der FoE-Request muss von der Gegenseite akzeptiert und bedient werden. Die TwinCAT-Versionen (3.10 b4014 und 2.11 b2245) unterstützen dies nicht. Die Mailboxgröße muss hierbei auf beiden Seiten gleich groß sein.

Gepufferte FoE Betriebsart

Bei Einstellung von [MASK_BUFFER_FOE] = Bit 8 in dem Objekt 0xF800 (Device Config), Subindex 02 sind unterschiedliche Mailboxgrößen zulässig.

6.6.4.1 Hinweise für FoE unter TwinCAT

Programmierung in TwinCAT

Für eine Programmierung ist die Bibliothek mit Funktionsbausteinen für den Zugriff auf EtherCAT Master und Slave Geräte, „Tc2_EtherCAT“ einzubinden („Tc2_..“ ist für TwinCAT 3 verwendbar). Die darin enthaltenen folgenden Funktionsbausteine ermöglichen die folgenden Zugriffe für den Daten- bzw. Dateitransport:

- FUNCTION_BLOCK FB_EcFoeOpen (öffnet den Kommunikationsport)
- FUNCTION_BLOCK FB_EcFoeAccess (schreiben/lesen von Daten)
- FUNCTION_BLOCK FB_EcFoeClose (schliessen des Kommunikationsports)

Weitere Angaben hierzu auf der Beckhoff Informationsseite (infosys.beckhoff.com).

Hinweis: der FoE-Transfer kann nicht selbst auf reguläres Dateieinde und somit vollständige Datenübertragung prüfen. Es ist zu empfehlen, die Dateigröße bzw. die Dateivollständigkeit durch andere Kanäle zu übertragen, z.B. prüfbare Meta-Informationen in der Datei selbst.

Ein FoE-Datenaustausch über die EL6695 folgt folgendem Schema:

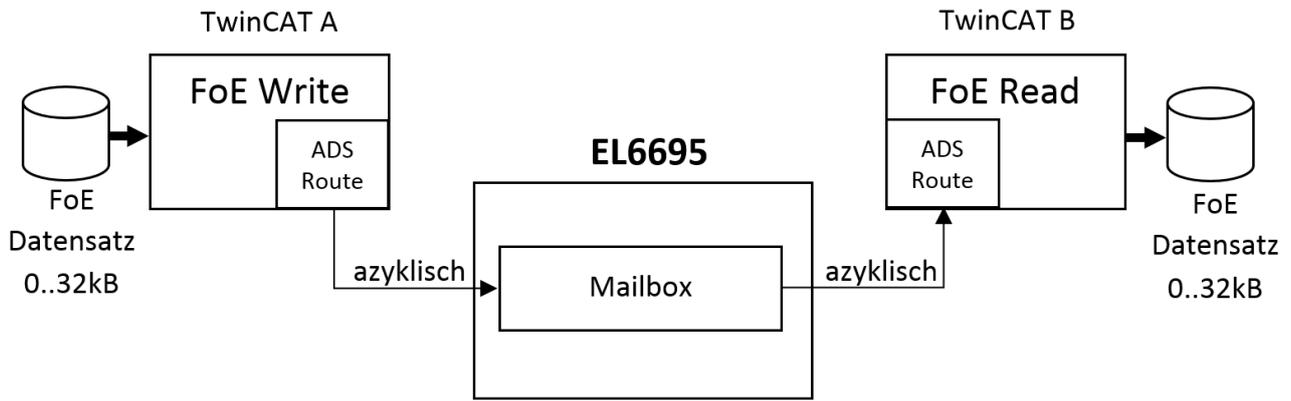


Abb. 161: FoE-Datenaustausch der EL6695

- Es können FoE-Datensätze zu je max. 32 kB über die EL6695 übertragen werden
- Es wird die azyklische EtherCAT-Kommunikation durch die EL6695-Mailbox (default 1024 Byte) genutzt
- Die EL6695 dient nur als Daten-Zwischenspeicher ohne eigene FoE Bearbeitung

6.6.4.2 Beispiel - Konfiguration

Beispiel Konfiguration

Anhand eines Programmierbeispiels soll die Funktionalität von FoE verdeutlicht werden. Die folgende [Abbildung 160](#) zeigt den physikalischen Aufbau mit einem embedded PC CX2040 inkl. Netzteil CX2100-0014 und nachfolgend EL2809, EL1004 und EL6695, wobei der Ethernet-Anschluss X001 vom CX-PC mit dem „oberen“ RJ45-Anschluss X1 der bridge-Klemme EL6695 verbunden ist:

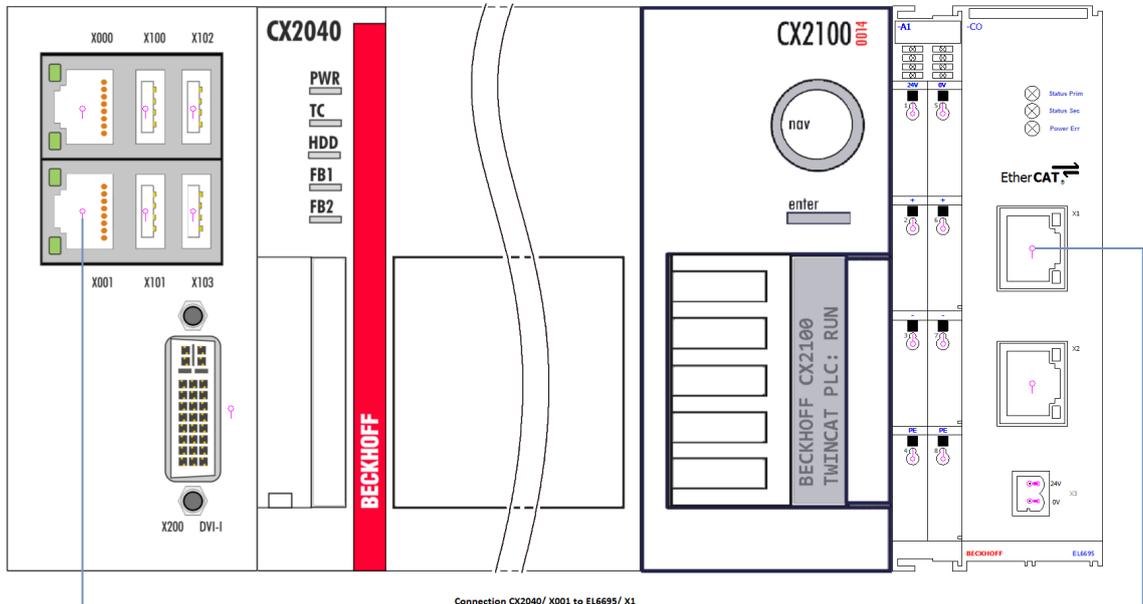


Abb. 162: Aufbau CX2040 inkl. Netzteil CX2100-0014 und EL6695

Im Systemmanager ist im Weiteren das CoE Objekt 0x1A05 hinzuzufügen, um Empfängerseitig anhand des FoE Info Objektes "Data Bytes Pending" erkennen zu können, ob Daten bzw. eine Datei empfangen wurden. Dazu muss die entsprechende Checkbox in den Prozessdaten aktiviert werden. Gezeigt ist dies in der Abbildung nur für die Sekundärseite, die in diesem Beispiel die Empfängerseite darstellt. Die im TwinCAT Beispielprogramm enthaltene öffentlich zugängliche Variable "ScndFoeBytesToRead" ist mit "Data Bytes Pending" im Weiteren zu verknüpfen.

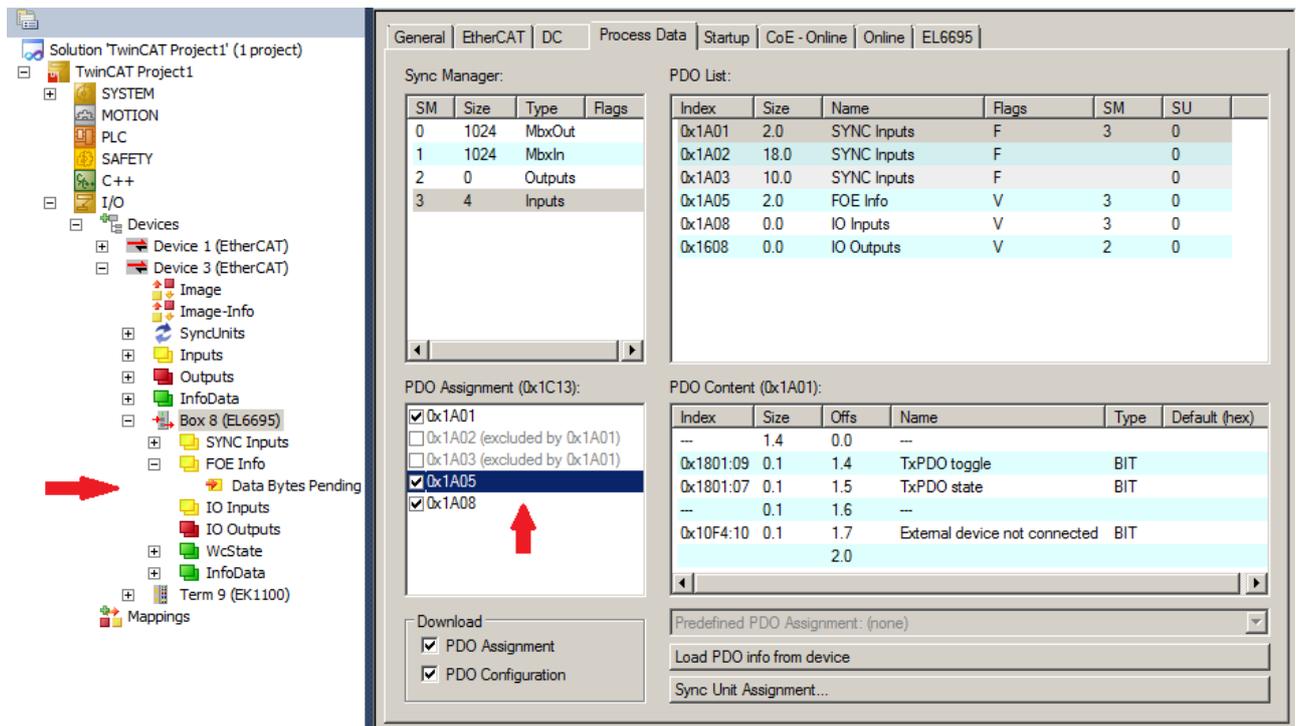


Abb. 163: Hinzufügen von 0x1A05 (Data Bytes Pending) auf der Sekundärseite

Init – Startup Konfiguration

Entsprechend der Konfigurationsdefinition des Objektes 0xF800 (siehe CoE/Parameterverzeichnis – Profilspezifische Objekte) sind für den Übergang vom Init-State zum PreOp State (I → P) in 0xF800 zwei Werte einzutragen:

- Der Wert 0x4000 in 0xF800:01 (um EoE zu deaktivieren)
- Der Wert 0x0100 in 0xF800:02 (spezieller Modus um auf der EL6695 einen FoE Pufferspeicher zu reservieren)

Im Folgenden ist die Vorgehensweise unter Verwendung der Benutzeroberfläche des Systemmanager (TC3.1) gezeigt; ausgewählter Kartenreiter „Startup“ auf der Primär oder Sekundärseite markierte Klemme oder Box (Terminal oder Box):

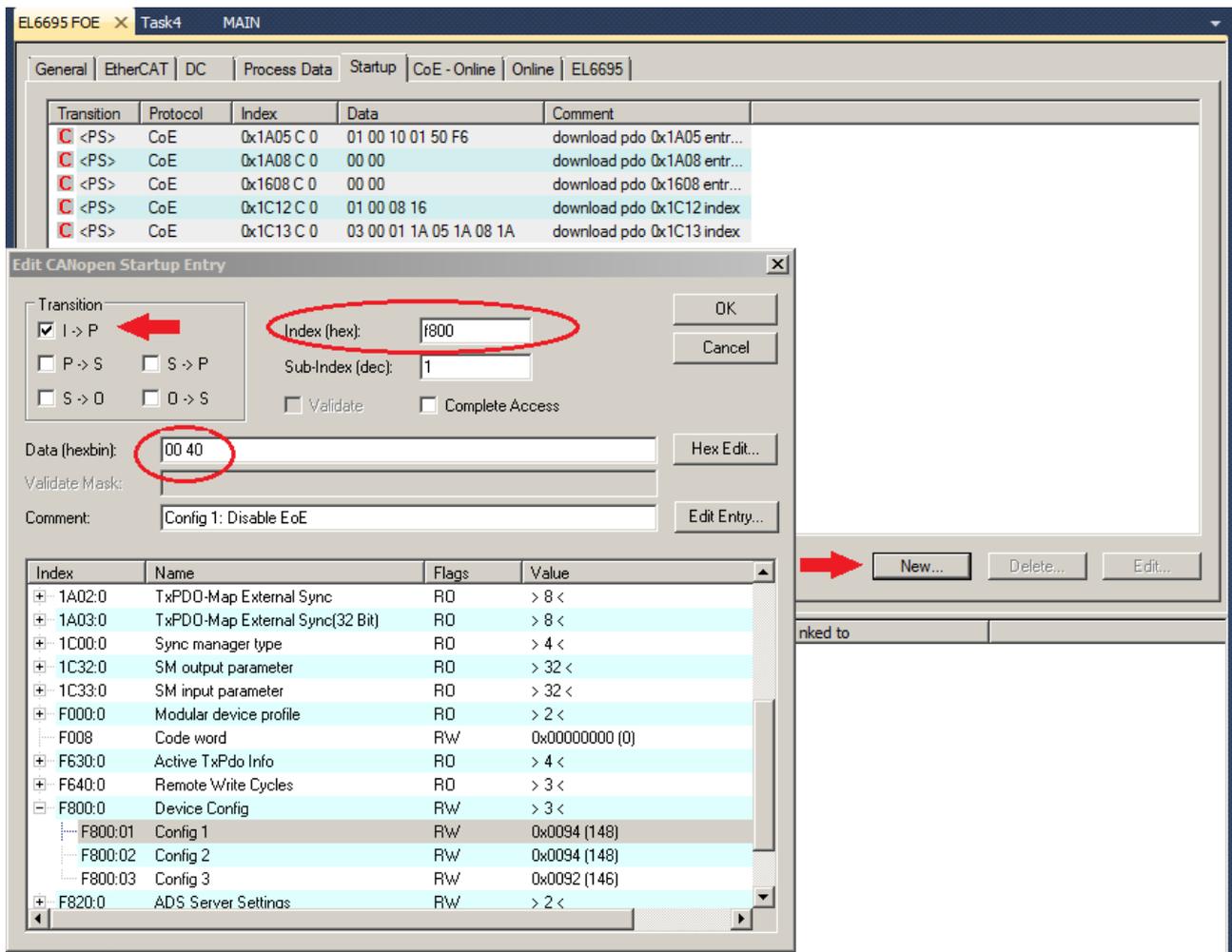


Abb. 164: Eintrag des Wertes 0x4000 in Geräte-Konfiguration 0xF800:01 (Comment = „Config 1: Disable EoE)

Hier ist der Wert 0x4000 (Eintrag: 00 40) für das Objekt 0xF800, Subindex 01 einzutragen, um EoE zu deaktivieren. Dies ist lediglich aufgrund der in diesem Beispiel verwendeten Verbindung von Primär und Sekundärseite der Klemme nötig, da beide Seiten an einem EtherCAT-Master angeschlossen sind.

● Blockieren des EoE Protokolls unter FoE

i Das EoE Protokoll ist durch das Setzen von Bit 14 (0x4000) im Objekt 0xF800:01 auf der Klemme zu blockieren, falls die Primärseite und Sekundärseite an einem EtherCAT-Strang angeschlossen ist, da es sonst zu einer Blockierung der Klemme (aufgrund von wiederholten ARP-Ethernet-Requests) kommt.

Im Weiteren ist der Wert 0x0100 (Eintrag 00 01) für 0xF800:02 einzutragen (Comment = Config 2: Enable FoE Buffer). Ob dies auf Primärseite oder Sekundärseite der Bridge-Klemme EL6695 geschieht ist nicht von Belang, da diese Einstellungen immer auf die jeweils andere Seite mit übernommen werden.

● Änderung von Device Config 0xF800

i Das Objekt 0xF800 ist auf Primär- oder Sekundärseite konfigurierbar und wird für die jeweils andere Seite stets übernommen.

Es ist darauf zu achten, dass die richtige Transition ausgewählt ist: P → S muss aus sein, I → P muss aktiviert sein. Nach dem dies erfolgt ist sollte es wie in der nachfolgenden Abbildung gezeigt aussehen:

General EtherCAT DC Process Data Startup CoE - Online Online EL6695					
Transition	Protocol	Index	Data	Comment	
<PS>	CoE	0x1A05 C 0	01 00 10 01 50 F6	download pdo 0x1A05 entries	
<PS>	CoE	0x1A08 C 0	00 00	download pdo 0x1A08 entries	
<PS>	CoE	0x1608 C 0	00 00	download pdo 0x1608 entries	
<PS>	CoE	0x1C12 C 0	01 00 08 16	download pdo 0x1C12 index	
<PS>	CoE	0x1C13 C 0	03 00 01 1A 05 1A 08 1A	download pdo 0x1C13 index	
IP	CoE	0xF800:01	0x4000 (16384)	Config 1: Disable EoE	
IP	CoE	0xF800:02	0x0100 (256)	Config 2: Enable FoE Buffer	

Abb. 165: Eintrag beider Werte in die Geräte-Konfiguration 0xF800

Erläuterungen zum Programmbeispiel

Dem im Anhang befindlichen Programmbeispiel, dass eine FoE Datenübertragung verdeutlichen soll, liegt das im Folgenden dargestellte Zustandsdiagramm für den Schrieb- und Lesezugriff zugrunde.

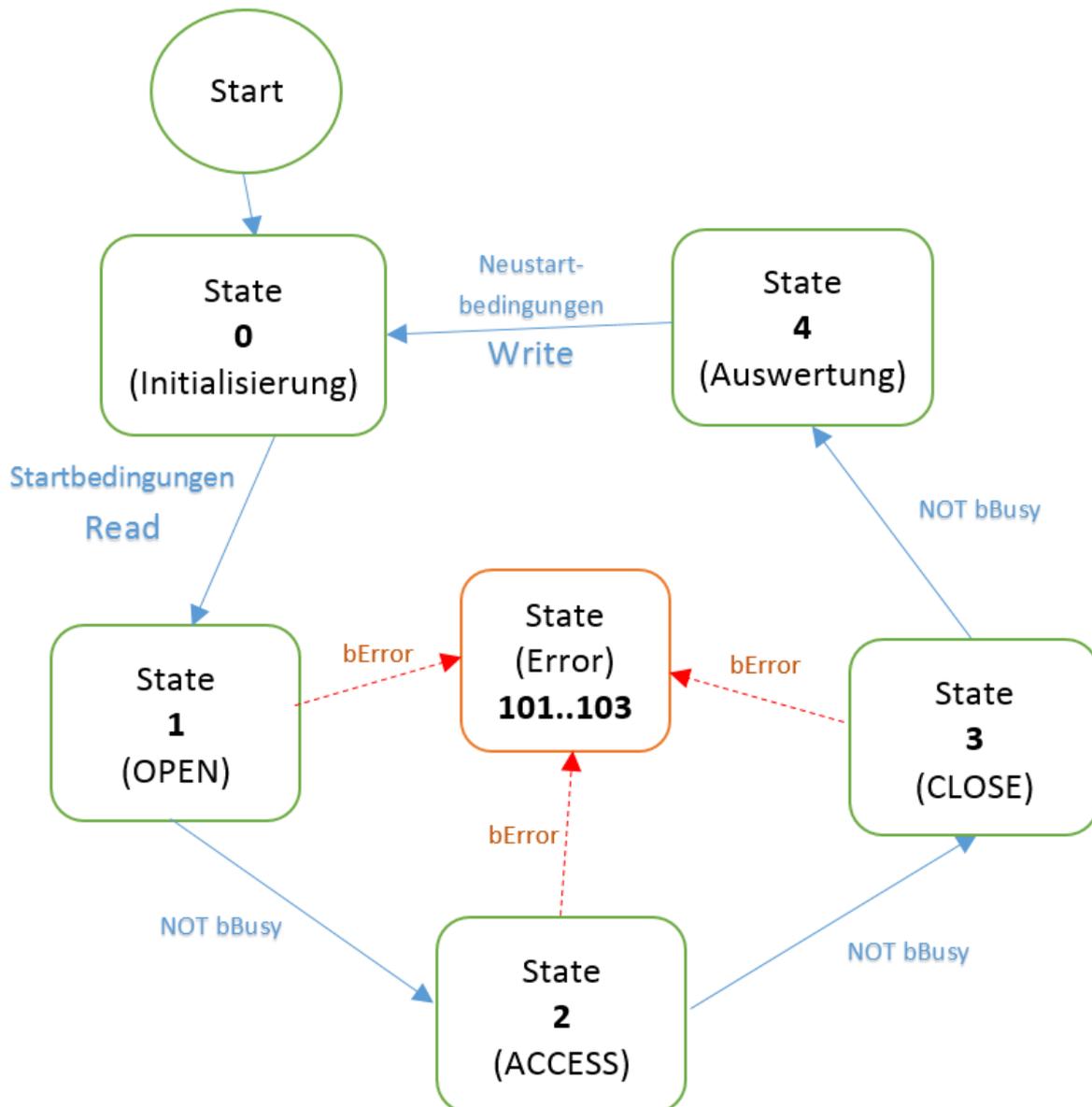


Abb. 166: Zustandsdiagramm zum Beispielprogramm: Schreiben und Lesen von Zufallsdaten per „OPEN“, „ACCESS“ und „CLOSE“

Das im Programm definierte **bEnabled** – Flag ist zur Steuerung des Programms vorgesehen und wird für die Startbedingung bzw. Neustartbedingung verwendet. Dieses Flag kann aufgrund der Zusatzdeklaration AT%I* mit einem „realen“ Eingang einer Eingangsklemme verknüpft werden, um dadurch „von außen“ das Programm z.B. mit einem angeschlossenen Taster (auf +) zu steuern.

- Startbedingung Read: **bEnabled** muss TRUE sein sein (z.B. extern angeschlossenen Taster betätigen); zusätzlich wird „ScndFoeBytesToRead“ auf eingehende Daten geprüft und muss > 0 sein.
- Neustartbedingung Write: **bEnabled** muss wieder FALSE sein (z.B. extern angeschlossenen Taster wieder loslassen).

Das **bEnabled**–Flag wird zudem innerhalb des Zustandes 1 der „Statemachine“ zum Schreiben der Daten auf TRUE geprüft, wodurch erst dann der Schreibvorgang durch ein OPEN in Gang gesetzt wird.

Das **bDataEqual kann** aufgrund der Zusatzdeklaration AT%Q* mit einer Ausgangsvariable einer Klemme verknüpft werden, die Digitale Ausgänge bereitstellt. Daran ist zuletzt zu sehen, ob der korrekte Datentransport erfolgte.

Die einzelnen Zustände, State 0 bis 4 sind als **iWrState** wie folgt programmiert:

- **[iWrState =0]:** Durchführung von Initialisierungen:
 - Setze FALSE → **bDataEqual**
 - Erzeuge Zufallsdaten in **aWrBuffer**
 - Vorbereitung des Aufrufs des Funktionsblocks für OPEN: Setze FALSE → **bExecute**
 - Setze nächsten Zustand: 1 → **iWrState**
- **[iWrState =1]:** Abfrage der Eingangs-Variable **bEnabled**, ob mit Aufrufs des Funktionsblocks für OPEN begonnen werden soll. Wenn **bEnabled** = TRUE, dann erfolgt der Aufruf des Funktionsblocks OPEN mit folgenden Parametern:
 - EC_MasterNetId_Wr → **sNetId**
 - EL6695_WR_EcAddr → **nPort**
 - 16#12345678 → **dwPass**
 - 1 → **eMode** (Kennung zum Schreiben)
 - TRUE → **bExecute**
 - T#10S → **tTimeout**
 - Dieser Aufruf wird solange wiederholt, bis eine Abfrage des **bBusy** – Flags gleich FALSE ist; es erfolgt dann:
 - Setze nächsten Zustand: 2 → **iWrState**; Ist **bBusy** = TRUE, so erfolgt kein Zustandswechsel. Auf die Behandlung der Fehlerfälle durch das Flag **bError** des Funktionsblocks OPEN (für Lesen/Schreiben) wird in diesem Beispiel und auch für alle anderen Funktionsblöcke (ACCESS, CLOSE für Lesen/Schreiben) nicht eingegangen.
 - Vorbereitung des Aufrufs des Funktionsblocks für ACCESS: Setze FALSE → **bExecute**
- **[iWrState =2]:** Es erfolgt der Aufruf des Funktionsblocks ACCESS mit folgenden Parametern:
 - fbWrFoeOpen.hFoe → **hFoe**
 - ADR(aWrBuffer) → **pBuffer**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - Dieser Aufruf wird solange wiederholt, bis eine Abfrage des **bBusy** – Flags gleich FALSE ist; es erfolgt dann:
 - Setze nächsten Zustand: 3 → **iWrState**
 - Vorbereitung des Aufrufs des Funktionsblocks für CLOSE: Setze FALSE → **bExecute**
- **[iWrState =3]:** Es erfolgt der Aufruf des Funktionsblocks CLOSE mit folgenden Parametern:
 - fbWrFoeOpen.hFoe → **hFoe**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - Dieser Aufruf wird solange wiederholt, bis eine Abfrage des **bBusy** – Flags gleich FALSE ist; es erfolgt dann:
 - Setze nächsten Zustand: 4 → **iWrState**
- **[iWrState =4]:** In diesem Zustand wird lediglich die Eingangs-Variable **bEnabled** abgefragt ob sie wieder FALSE ist. Dann beginnt das Programm wieder mit dem Anfangszustand:
 - Wenn **bEnabled** = FALSE dann setze nächsten Zustand: 0 → **iWrState**

Beim nächsten Start werden wieder neue Zufallszahlen erzeugt, die dann beim Vergleich mit den gelesenen Werten wieder übereinstimmen sollten. Hierfür ist das **bDataEqual** Flag vorgesehen, dass dies kenntlich machen soll, wie im Folgenden zu sehen sein wird.

Das Zustandsdiagramm für den Lesezugriff sieht im Prinzip ähnlich aus und hat im Zustand **4** ein Programmabschnitt für den Datenvergleich der Geschriebenen mit den Gelesenen Werten.

Hierbei sind die einzelnen Zustände, State 0..4 als **iRdState** wie folgt programmiert:

- **[iRdState =0]:** In diesem Zustand wird die Eingangs-Variable **bEnabled** abgefragt ob sie TRUE ist. Damit beginnt das Programm mit der Abfrage der auf [FoE Info].Data Bytes Pending (Objekt 0x1A05) verknüpften Eingangsvariable „ScndFoeBytesToRead“ um dadurch festzustellen, ob Daten auf der Empfängerseite „vorliegen“:
 - Abfrage von **ScndFoeBytesToRead** wenn **bDataEqual** = TRUE
 - Setze nächsten Zustand: 1 → **iRdState** wenn **ScndFoeBytesToRead** = TRUE
 - Setze zur Vorbereitung des Aufrufs des Funktionsblocks für OPEN: Setze FALSE → **bExecute** wenn **ScndFoeBytesToRead** = TRUE
- **[iRdState =1]:** Abfrage der Eingangs-Variable **bEnabled**, ob mit Aufrufs des Funktionsblocks für OPEN begonnen werden soll. Wenn **bEnabled** = TRUE, dann erfolgt der Aufruf des Funktionsblocks OPEN mit folgenden Parametern:
 - EC_MasterNetId_Rd → **sNetId**
 - EL6695_RD_EcAddr → **nPort**
 - 16#12345678 → **dwPass**
 - 0 → **eMode** (Kennung zum Lesen)
 - TRUE → **bExecute**
 - T#10S → **tTimeout**
 - Dieser Aufruf wird solange wiederholt, bis eine Abfrage des **bBusy** – Flags gleich FALSE ist; es erfolgt dann:
 - Setze nächsten Zustand: 2 → **iRdState**
 - Vorbereitung des Aufrufs des Funktionsblocks für ACCESS: Setze FALSE → **bExecute**
- **[iRdState =2]:** Es erfolgt der Aufruf des Funktionsblocks ACCESS mit folgenden Parametern:
 - fbRdFoeOpen.hFoe → **hFoe**
 - ADR(aRdBuffer) → **pBuffer**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - Dieser Aufruf wird solange wiederholt, bis eine Abfrage des **bBusy** – Flags gleich FALSE ist; es erfolgt dann:
 - Setze nächsten Zustand: 3 → **iRdState**
 - Vorbereitung des Aufrufs des Funktionsblocks für CLOSE: Setze FALSE → **bExecute**
- **[iRdState =3]:** Es erfolgt der Aufruf des Funktionsblocks CLOSE mit folgenden Parametern:
 - fbRdFoeOpen.hFoe → **hFoe**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - Dieser Aufruf wird solange wiederholt, bis eine Abfrage des **bBusy** – Flags gleich FALSE ist; es erfolgt dann:
 - Setze nächsten Zustand: 4 → **iRdState**
- **[iRdState =4]:** In diesem Zustand erfolgt zuletzt die Prüfung beider Speicherbereiche wobei das Flag **bDataEqual** auf TRUE gesetzt wird, falls beide Speicherbereiche gleich sind. Es ist empfehlenswert, dieses Flag als Ausgangsvariable „sichtbar“ zu machen um sie z.B. über eine Ausgangsklemme auf einen Ausgang zu schalten.
 - Setze nächsten Zustand: 0 → **iRdState**

• 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.

→ Download Beispielprogramm FoE:

<https://infosys.beckhoff.com/content/1031/el6695/Resources/1421822987.zip>

Vorbereitungen zum Starten des Beispielprogramms (tnzip-Datei/TwinCAT 3)

- Nach Klick auf den Download-Button speichern Sie das Zip-Archiv lokal auf ihrer Festplatte und entpacken die *.tnzip-Archivdatei in einem temporären Ordner.

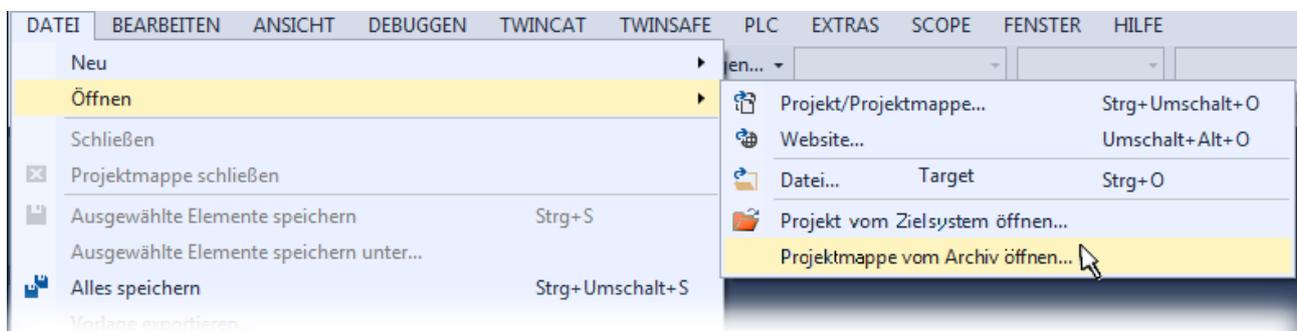


Abb. 167: Öffnen des *.tnzip-Archives

- Wählen Sie die zuvor entpackte .tnzip-Datei (Beispielprogramm) aus.
- Ein weiteres Auswahlfenster öffnet sich: wählen nun Sie das Zielverzeichnis, wo das Projekt gespeichert werden soll.
- Die generelle Vorgehensweise für die Inbetriebnahme der PLC bzw. dem Start des Programms kann u. a. den Klemmen-Dokumentationen oder der EtherCAT-Systemdokumentation entnommen werden.
- Das EtherCAT-Gerät im Beispiel ist in der Regel, zuvor ihrem vorliegenden System bekannt zu machen. Verwenden Sie nach Auswahl des EtherCAT-Gerätes im „Projektmappen-Explorer“ rechtsseitig den Karteireiter „Adapter“ und Klicken „Suchen...“:

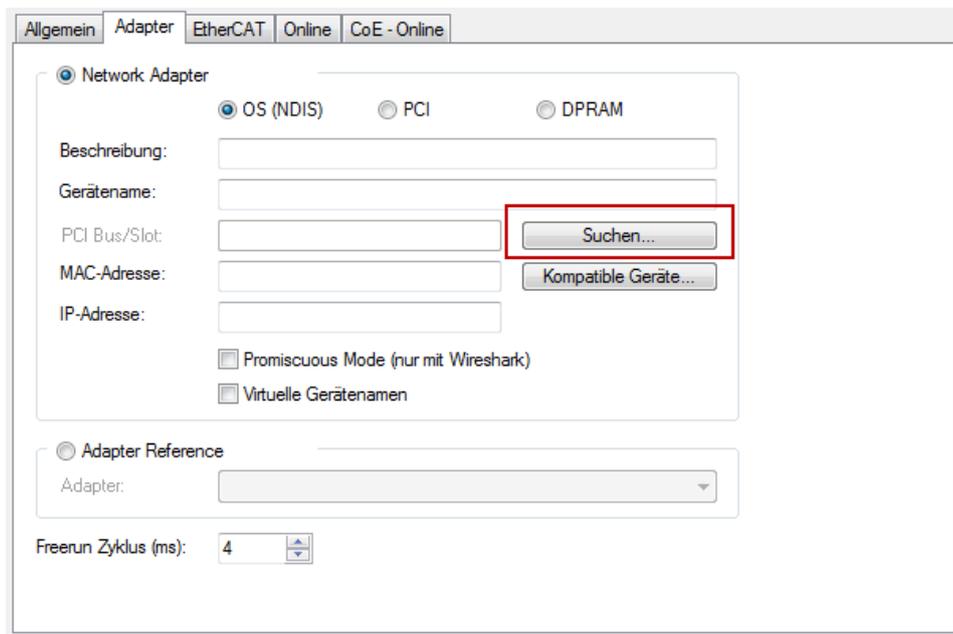
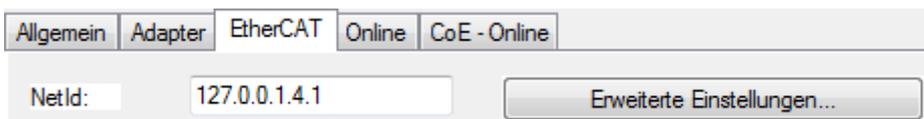


Abb. 168: Suche der bestehenden HW-Konfiguration zur bestehenden EtherCAT-Konfiguration

- Überprüfen der NetId: der Karteireiter „EtherCAT“ des EtherCAT-Gerätes zeigt die konfigurierte NetId:



Diese muss mit den ersten vier Zahlenwerten mit der Projekt-NetId des Zielsystems übereinstimmen. Die NetId des Projektes kann oben in einem Textfeld der TwinCAT-Umgebung eingesehen werden. Ein pull-down Menü kann durch einen Klick rechts im Textfeld geöffnet werden; dort ist zu jedem Rechnernamen eines Zielsystems die NetId in Klammern angegeben.

- Ändern der NetId: mit einem Rechtsklick auf „Gerät EtherCAT“ im Projektmappen-Explorer öffnet sich das Kontextmenü, in dem „Ändern der NetId“ auszuwählen ist. Die ersten vier Zahlen der NetId des Projektes sind einzutragen. die beiden letzten Werte sind in der Regel 4.1.

Beispiel:

- NetId des Projektes: myComputer (123.45.67.89.1.1)
- Eintrag per „Change NetId...“: 123.45.67.89.4.1

6.6.4.3 Beispiel FoE – Datendurchsatz

Im Folgenden ist ein Beispiel für eine FoE-Durchsatzermittlung durch eine EL6695 gegeben. Die Werte sind als exemplarisch für diesen Aufbau und als grobe Richtwerte zu verstehen. Der in der jeweiligen Applikation real erreichbare Durchsatz ist im Bedarfsfall anwenderseitig zu ermitteln.

Das dazu verwendete TwinCAT-Programm ist nicht Bestandteil dieser Dokumentation, es basiert auf dem angegebenen [FoE-Beispielprogramm](#) [► 167].

Der Datendurchsatz wird als Quotient von Datenmenge durch den dafür benötigten Zeitbedarf definiert (Datenmenge pro Zeit). Mit den bereits beschriebenen FoE-Zugriffsbausteinen wird eine gegebene Datenmenge in die EL6695 geschrieben bzw. von ihr gelesen. Die EL6695 ist dafür ausgelegt, Kommunikation zwischen zwei unterschiedliche TwinCAT-Systeme herzustellen. Es ist mit der Klemme aber auch möglich, eine Verbindung der Primär- und Sekundärseite innerhalb eines TwinCAT-Systems aufzubauen indem beispielsweise die Verbindung ihrer Sekundärseite über X1 vom Ausgang X2 des Kopplers EK1100 ausgeht. Dann liegen ebenfalls zwei unabhängige EtherCAT-Stränge in einem TwinCAT-System vor (vgl. hierzu [Beispiel - Konfiguration](#) [► 160]).

Die Zeitmessung erfolgte indirekt über die Anzahl der benötigten Taskzyklen für jeden komplett durchgeführten Lesezugriff (d.h. Anzahl geschriebener Bytes ist gleich Anzahl gelesener Bytes) über den `FB_EcFoeAccess` Baustein (siehe hierzu [Erläuterungen zum Programmbeispiel](#) [► 163]). Das PDO

„DataBytesPending“ der zu lesenden Seite der Klemme ist >0 falls Daten anstehen und kennzeichnet den Beginn des Lesezugriffs. Das Ende des Lesezugriffs ist aus der bekannten geschriebenen Datenmenge ermittelt worden (bei zwei unabhängigen Systemen muss die Dateigröße bekannt sein).

Bei 1 ms Taskzykluszeit ergibt sich für die Berechnung des Datendurchsatzes einfach:
{Datenmenge [Bytes]} / {Anzahl Taskzyklen} = Durchsatz in [kB/s]

Ablauf der Messung

- die Testumgebung ermittelt die Anzahl der von der Klemme benötigten Taskzyklen für einen Lesezugriff. Es wurde wiederholend ein in die Klemme geschriebener Datensatz unterschiedlicher Größe mit unterschiedlichen Datenpaketgrößen komplett wieder ausgelesen. Dabei wurde jeweils die Anzahl der Taskzyklen für die einzelnen Lesezugriffe gezählt.
- Es wurde ebenfalls der FoE-Durchsatz des Testprogramms ohne die zwischengeschaltene EL6695 ermittelt. Damit ergibt sich der „virtuelle“ Zeitbedarf des Testprogramms.
- die tatsächlichen von der Klemme benötigte Anzahl Taskzyklen (und damit der Datendurchsatz) ergibt sich sodann aus der Differenz der benötigten „realen“ Taskzyklen (mit EL6695) und den virtuellen Taskzyklen.
- Der gesamte Durchsatztest umfasst mehrere 1000 Einzeltest mit unterschiedlichen Datenmengen und Datenpaketgrößen.

Konfiguration im Beispiel:

- C6015 + EK1100 + EL6695 (Primärseite) + EL9011
- EK1100 + EL6695-0002 (Sekundärseite)

Eckdaten des Beispiels:

- Taskzykluszeit 1 ms
- Der Testdurchlauf umfasste alle FoE- und Paketgrößen. Da sich der Durchsatz außerhalb festgestellter Eckbereiche nicht mehr wesentlich ändert, wurden für die komprimierte grafische Darstellung im Folgenden folgende Grenzen gewählt:
- Per FoE übertragene Datenmenge (FoE-Dateigröße): von 1167 bis 32767 Bytes in 50 Schritten zu je 632 Bytes pro Schritt
- Datenpakete (also Pakete in die die FoE-Datei aufgeteilt wird): von 30 bis 470 Bytes in 40 Schritten zu je 11 Bytes pro Schritt

Ergebnis:

Datenpaketgröße (Schrittweise inkrementiert) = X-Achse

Datenmenge (Schrittweise inkrementiert) = Y-Achse

Anzahl benötigter Taskzyklen = Z-Achse; entspricht Z(X, Y)

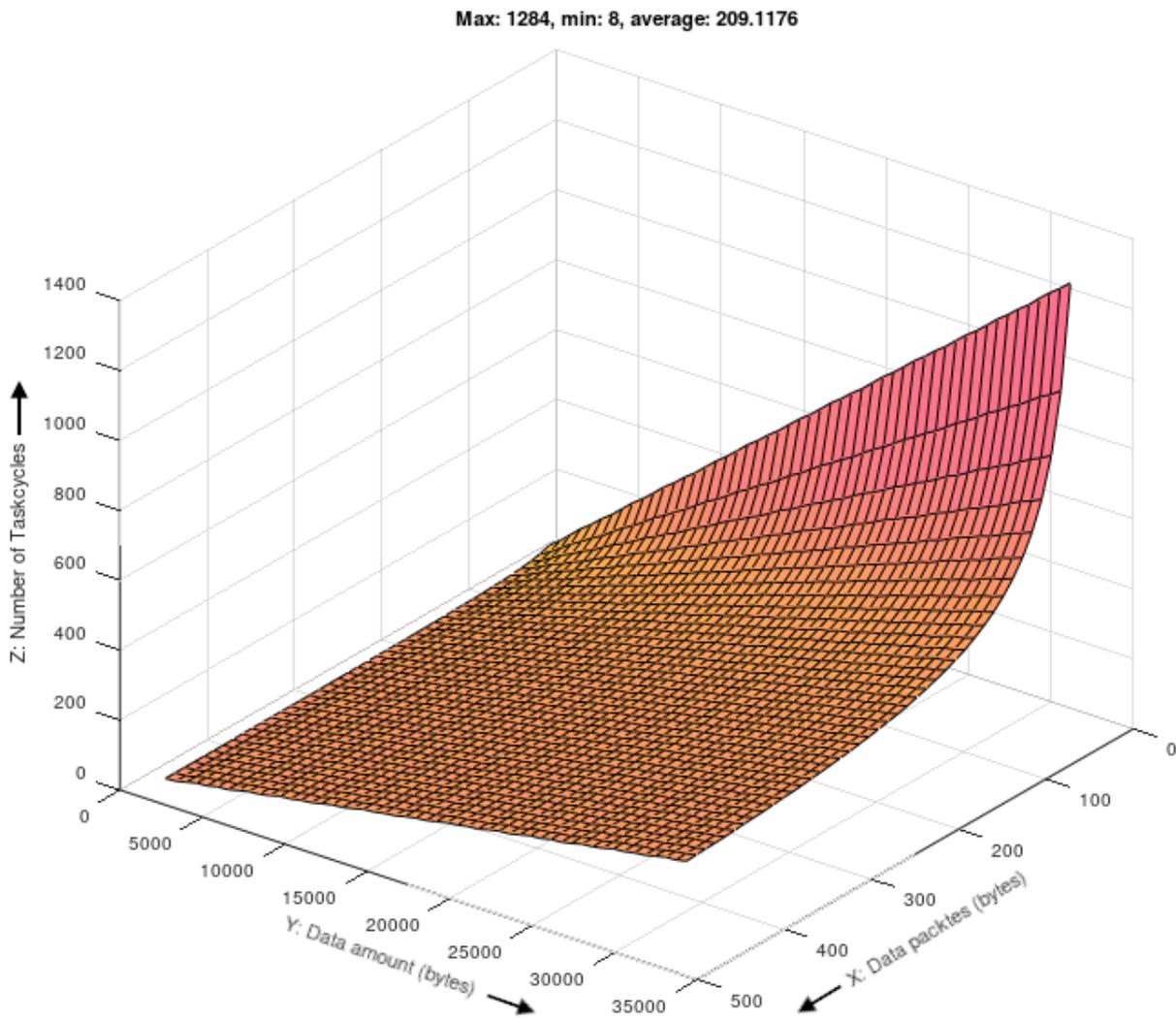


Abb. 169: FoE-Durchsatztest mit EL6695: Anzahl der Taktzyklen in Abhängigkeit von Datenpaketgröße und Datenmenge

Hierin enthalten sind jedoch auch die systembedingten virtuellen Werte aufgrund des sequenziellen Datenzugriffs (s.o.). Werden diese entsprechend ebenfalls aufgezeichnet und für jeden Testpunkt X/Y von den einzelnen Werten der Z-Achse subtrahiert, so ergibt sich ein quasi linearer Diagrammverlauf, welcher keine Abhängigkeit von der Datenpaketgröße mehr hat:

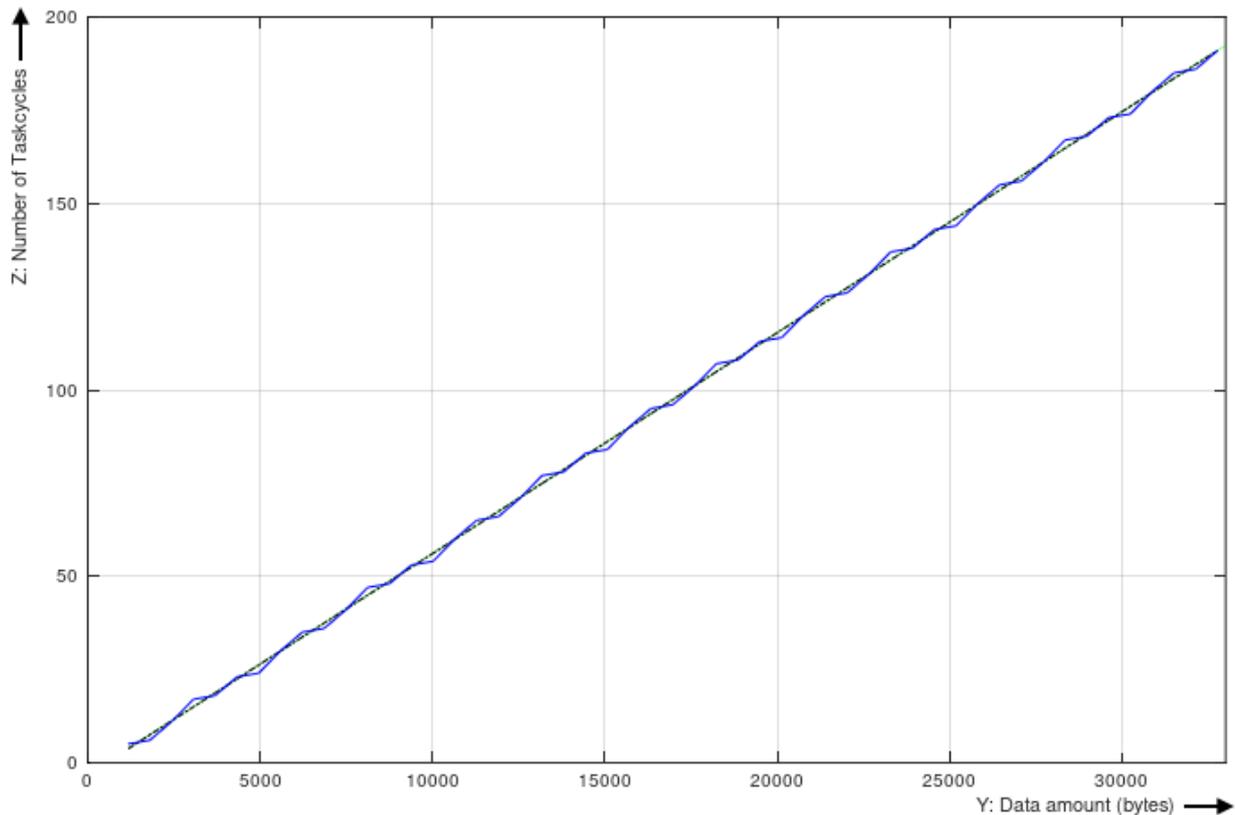


Abb. 170: Zum exemplarischen FoE-Datendurchsatz der EL6695

In der linearen Näherung ergibt dieses FoE-Beispiel einen Durchsatz von rund **171 kB/s**.

6.6.5 VoE - Vendor specific protocol over EtherCAT

Das VoE ermöglicht die Umsetzung eines Anwenderspezifischen Protokolls um die Geräteinterne Mailbox für einen speziellen, neu definierten Datentransport zu verwenden.

Für weiterführende Informationen richten Sie bitte eine Anfrage an den Beckhoff Support, unter Angabe hilfreicher Daten (TwinCAT Version, geforderte Performance Werte, Zykluszeit, Datenmengen etc.)

6.6.6 SoE - Servo Drive Profile over EtherCAT

Für weiterführende Informationen richten Sie bitte eine Anfrage an den Beckhoff Support, unter Angabe hilfreicher Daten (TwinCAT Version, geforderte Performance Werte, Zykluszeit, Datenmengen etc.)

6.7 Distributed Clocks

i DC-Support im EtherCAT Master

Die DistributedClock-Unit der EL6695 verfügt über die DC system time, aber keine Sync/Latch-Unit. Die entsprechend Initialisierungsroutine wird von TwinCAT3 ab b4018.4, TwinCAT2 ab b2248 unterstützt.

Die EL6695 Konfiguration ist unter dem Karteireiter ersichtlich und hat für den Datenaustausch die default Einstellung „keine Synchronisierung“ („FreeRun“).

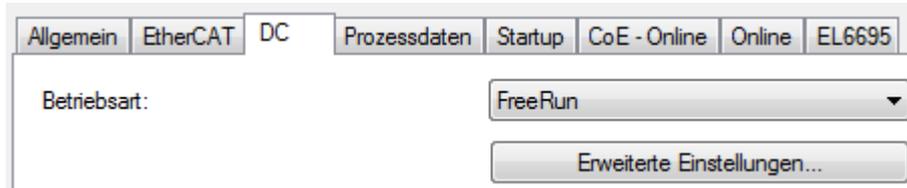


Abb. 171: Distributed Clocks: keine Synchronisierung

Da die EL6695 zwei vollständige EtherCAT Slaves beinhaltet, sind grundsätzlich die beiden Distributed-Clock-Einheiten entsprechend unabhängig voneinander. Die EL6695 unterstützt zwei DC-Synchronisierungsmethoden:

- Sie kann wie die Vorgängerversion EL6692 der SyncSlave-Seite die Internal/External Timestamp Informationen zur Verfügung stellen und so dem dort gelagerten EtherCAT Master die Möglichkeit geben seine Echtzeit/EtherCAT Zyklus nachzuregeln. Es sind beide Regelungsrichtungen möglich.
 - Anwendung unter TwinCAT: EL6695 auf DC-mode setzen damit sie von TwinCAT als Referenz Uhr verwendet werden kann



Abb. 172: Distributed Clocks: Synchronisierung

- Einen Timestamp-PDO einblenden und aktivieren (0x1A02 für 64 Bit oder 0x1A03 für 32 Bit). Dadurch erkennt TwinCAT, dass diese Klemme als externe Referenzuhr benutzt werden kann und liest die Zeitstempel ein.

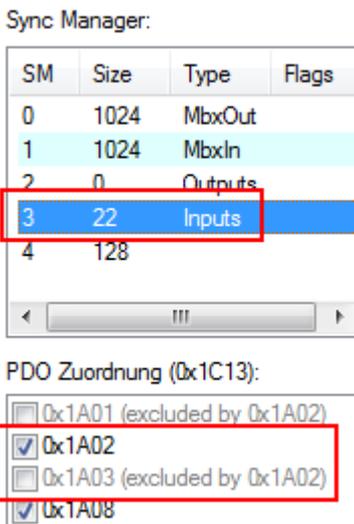


Abb. 173: Timestamp-PDO 0x1A02 oder 0x1A03 einblenden

- In den „EtherCAT DC master settings“ kann dann die externe Referenz Uhr ausgewählt werden:

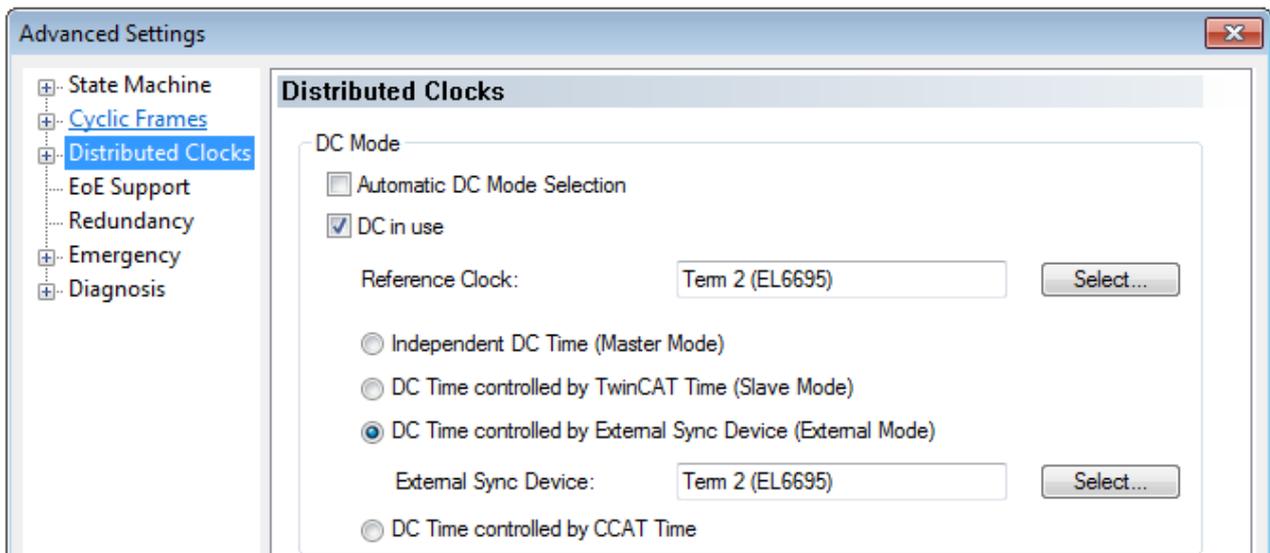


Abb. 174: Distributed Clocks: Auswahl der externen Referenz Uhr (hier auf der Primärseite)

Diese Einstellung ist lediglich auf der „SyncSlave“ –Seite vorzunehmen.

Auf der „SyncMaster“ – Seite ist beispielsweise eine EL3104 (DC-fähig) eingetragen:

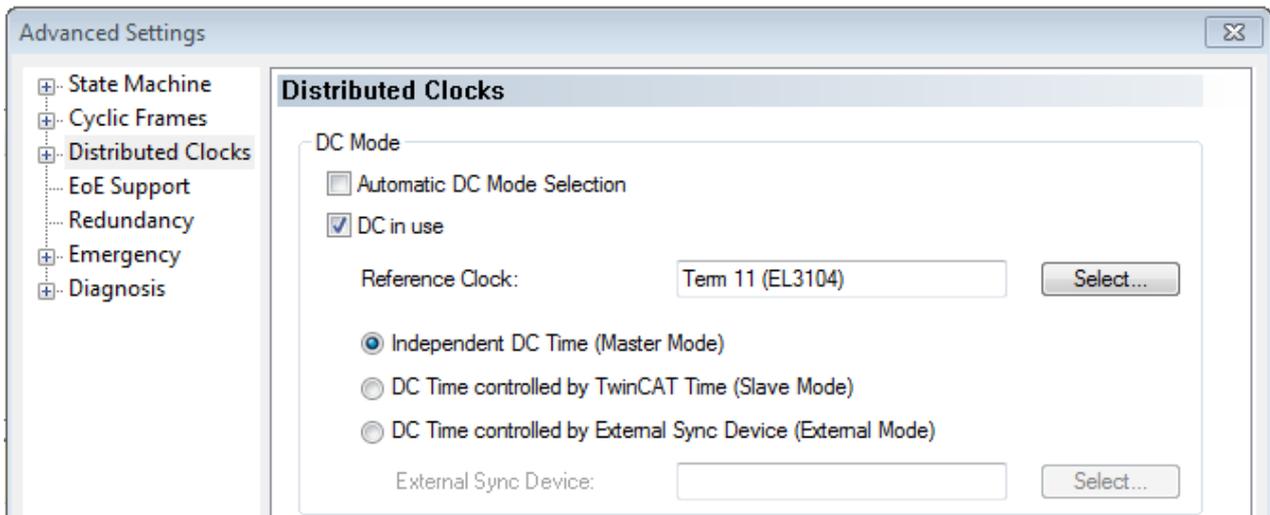


Abb. 175: Distributed Clocks: Auswahl der EL3104 als „SyncMaster“ (hier auf der Sekundärseite)

- Wenn die EL6695 in beiden EtherCAT System der erste DC-fähige Slave ist, kann eine direkte DC-Kopplung der beiden Bridge-Seiten aktiviert werden ohne dass die EtherCAT Master darüber informiert werden müssen. Dann folgen beide EtherCAT-System/Master der EL6695-Zeit, ohne speziell konfiguriert zu werden. Es verbleibt allerdings ein Offset zwischen beiden Systemzeiten mit konstantem Wert.
 - Beide EtherCAT Master setzen von Ihrer Seite aus 0x0920 DC system time offset. Die EL6695 akzeptiert dies jeweils und hält beide ESC in der EL6695 frequenzsynchron.

● Lokalregelungsverfahren nicht unterstützt

i Das in der EL6692 bekannte Lokalregelungsverfahren „ControlValue for DC MasterClock“ wird von der EL6695 nicht unterstützt. EtherCAT Master sollten das genannte Timestamp-Verfahren implementieren.

Da die EL6695 die Zeitinformation der DistributedClock intern nicht benutzt sondern nur an die Gegenseite weiterleitet, ist technologisch eine Verwendung der besonderen DC-Register im ESC nicht nötig bzw. wird nicht unterstützt. Bei vielen EtherCAT Geräten mit DC-Unterstützung werden im Status-Übergang PreOP->SafeOP die DC-Register 0x09A0, 0x0990, 0x0980, 0x09A8 vom Master beschrieben. Die EL6695 verfügt nicht über diese Register weshalb ein Beschreibungsversuch beim Hochlauf zu Fehlermeldungen des EtherCAT Master führt, z.B. „Init Cmd failed: set DC cycle time“.

IP	PS	PI	SP	SO	SI	OS	OP	OI	IB	BI	CMD	Comment
	X										FPWR	set DC cycle time
	X										FPWR	set DC start time
	X										FPWR	set DC activation
	X										FPWR	set DC latch0 cfg

Abb. 176: Distributed Clocks: Fehlermeldungen

Der EtherCAT Master hat deshalb bei der EL6695 entweder diese Register nicht zu beschreiben oder er wertet den `WcState` des Beschreibungskommandos nicht aus. In der EL6695-ESI wird diese DC-Eigenschaft durch das Flag `TimeLoopControlOnly = TRUE` gekennzeichnet, das der EtherCAT Master beachten kann.

6.8 Online Scan

In der EL6695 stehen je Seite 128 kByte Speicherplatz für eine beliebige Datei zur Verfügung, die per FoE geschrieben und gelesen werden kann.

Dies kann zum Abspeichern der Emulations-ESI benutzt werden damit ein Master die vollständige ESI (EtherCAT Slave Information) bekommt und in seiner Konfiguration verwenden kann.

Nachtrag: Beispiel

6.9 CompatibilityMode zur EL6692

Die EL6695 verfügt über einen sog. kompatiblen Modus zur EL6692.

Diese Modi arbeiten wie in der bekannten EL6692. Dadurch kann die EL6695 bei höherer Leistung auch an EtherCAT Mastern und Applikationssoftware betrieben werden die auf die EL6692-Schnittstellen ausgelegt worden sind. In diesen Modi werden vorrangig die gleichen default-Prozessdaten angeboten.

6.9.1 Modulbetrieb mit PDO Mapping und Assignment

Dieser Modulbetrieb ist quasi der standardmäßige Betrieb, auch der neuen EL 6695, und wird im Kapitel Symmetrisches PDO Mapping beschrieben.

6.10 Performance Betriebsarten EL6695

6.10.1 Grundlagen zum PDO Mapping

obsolet

Sehen Sie dazu auch

☰ Symmetrisches PDO-Mapping [▶ 134]

6.10.2 Ohne Init Commands

Dieser Modus ist in erster Linie für Fremdmaster vorgesehen, die nur die reinen Prozessdaten übertragen sollen. Dazu dürfen die Checkbox für das automatische PDO Assignment und Mapping im Reiter Process Data jeweils nicht aktiviert sein (siehe Kapitel [Selektives PDO-Mapping](#) [▶ 140]).

Anschließend können auf der jeweiligen Klemmenseite Variablen deklariert werden. Werden z.B. drei Byte Ausgangsvariablen deklariert und ein Gerätereustart durchgeführt (unter TwinCAT: Reload Devices) ist im jeweiligen CoE Objekt 0x1608 die Struktur des PD Outputs Objekts zu sehen. Sie hat eine Länge von 24 Bit bei drei Byte Variablen.

☰ 1608:0	IO RxPDO-Map 8	RW	> 1 <
└─ 1608:01	SubIndex 001	RW	0x7000:01, 24

Abb. 177: 24 Bit CoE Object 0x1608

Auf der anderen Seite der Klemme müssen für einen Datenaustausch auch das Assignment und Mapping deaktiviert werden, damit die Klemme die Daten interpretieren kann und nicht drei Einträge mit jeweils 8 Bit erwartet, sondern einen Eintrag mit 24 Bit. Zudem muss die Prozessdatenlänge auf der Remote Seite exakt übereinstimmen. So muss in diesem Beispiel auf der Remote Seite, dass Mapping für das CoE Objekt 0x1A01 deaktiviert werden (siehe Kapitel [Symmetrisches PDO Mapping](#) [▶ 134]), da sonst von einer anderen Anzahl Eingangsdaten ausgegangen wird und der Datenaustausch nicht funktioniert.

6.10.3 Object description download

EtherCAT Geräte aus dem Bereich der Kommunikation verfügen über ein CoE-Verzeichnis als eine Übersicht für Objekte und Parameter für betriebsinterne Funktion und Kommunikation. Die bridge-Klemme EL6695 besitzt stets ein entsprechendes Objektverzeichnis (Objectdictionary – OD) im Initialzustand, das für die EL6695 gilt.

Durch komplettes oder teilweises (partielles) verändern des Objektverzeichnisses (OD) ist es möglich, ein anwenderspezifisches Gerät auf der EL6695-Klemme zu definieren. Über die Extension im TwinCAT Systemmanager kann nach Bedarf ein vollständiges oder partielles OD erstellt werden und als *.coe Datei, ein Beckhoff-internes Dateiformat, gesichert werden.

6.10.4 Geräteemulation

Bei der Geräteemulation wird ein von der Bridge-Klemme unterschiedlicher EtherCAT Slave auf einer Bridge-Seite nachgebildet. Physikalisch bleibt dabei die Bridge-Klemme EL6695 der EtherCAT Slave und kann immer auch in diesen Urzustand zurückversetzt werden.

Der zu emulierende EtherCAT Slave wird in Form der ESI auf die gewählte EL6695-Seite geladen. Die EL6695 übernimmt diese Daten und stellt nach außen dann den Slave in Form der Identitätsdaten, PDO und CoE nach. Dies betrifft aber nur die formale Darstellung, das (Zeit)Verhalten kann die EL6695 nicht leisten, dazu müsste auch die Firmware der EL6695 entsprechend geändert werden. Das „Verhalten“ des emulierten Slaves ist über die Gegenseite und die von ihr/zu ihr gelieferten Daten abzubilden.

Da auch im EmulationMode der so dargestellte Slave den EtherCAT ETG ConformanceTest bestehen muß, hat der Anwender Sorge zu tragen dass nur gültige/zertifizierte/zertifizierungsfähige ESI im EmulationMode auf die EL6695 geladen werden. Dazu sind die ETG-Vorschriften ETG1000.6 (Mapping) und ETG2000 (ESI Spezifikation) zu beachten.

Der EmulationMode kann ein- oder sogar beidseitig genutzt werden.

● **Extension (Nutzeroberfläche im TwinCAT)**

i Die sog. Extension (Nutzeroberfläche im TwinCAT) steht beim emulierten Slave nicht mehr zur Verfügung.

Für weiterführende Informationen richten Sie bitte eine Anfrage an den Beckhoff Support, unter Angabe hilfreicher Daten (TwinCAT Version, geforderte Performance Werte, Zykluszeit, Datenmengen etc.)

6.11 Applikationsspezifische Variablendefinition

Die Klemme EL6695 bringt in der default Konfiguration keine Prozessdaten mit, diese sind nach der jeweiligen Applikationsanforderung im Konfigurator anzulegen. Wenn der verwendete EtherCAT Master kein variables PDO mapping unterstützt, kann vom Anwender eine eigene ESI Gerätebeschreibungsdatei erstellt und im jeweiligen Konfigurator verwendet werden.

● **Technisch**

i Als weiterführende Dokumente dazu sind u.a. die ETG-Vorschriften ETG1000.6 (Mapping) und ETG2000 (ESI Specification) zu beachten.

● **Rechtlich**

i Desgleichen müssen Gerätebeschreibungen den EtherCAT ETG ConformanceTest bestehen, der jeweilige Anwender/Ersteller hat Sorge zu tragen, dass diese Geräte Beschreibungen in Abstimmung mit der ETG in Verkehr gebracht werden.

7 Anhang

7.1 EtherCAT AL Status Codes

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

7.2 Firmware Kompatibilität

Beckhoff EtherCAT Geräte werden mit dem aktuell verfügbaren letzten Firmware-Stand ausgeliefert. Dabei bestehen zwingende Abhängigkeiten zwischen Firmware und Hardware; eine Kompatibilität ist nicht in jeder Kombination gegeben. Die unten angegebene Übersicht zeigt auf welchem Hardware-Stand eine Firmware betrieben werden kann.

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 \[► 182\]](#).

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!

EL6695

Hardware (HW)	Firmware (FW)	Revision-Nr	Release-Datum
02 – 13*	03	Primary: EL6695-0000-0001 Secondary: EL6695-0002-0001	2015/02
	04	Primary: EL6695-0000-0002 Secondary: EL6695-0002-0002	2015/06
		Primary: EL6695-0000-0003 Secondary: EL6695-0002-0003	2015/06
	05	Primary: EL6695-0000-0004 Secondary: EL6695-0002-0004	2015/07
	06		2015/08
	07	Primary: EL6695-0000-0005 Secondary: EL6695-0002-0005	2017/10
	08		2018/04
	09	Primary: EL6695-0000-0006,	2021/07
	10		2022/02
	11	Secondary: EL6695-0002-0006	2023/12
	12*		2024/01

*) 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.

7.3 Firmware Update EL/ES/ELM/EM/EP/EPP/ERPxxxx

Dieses Kapitel beschreibt das Geräte-Update für Beckhoff EtherCAT-Slaves der Serien EL/ES, ELM, EM, EK, EP, EPP und ERP. Ein FW-Update sollte nur nach Rücksprache mit dem Beckhoff Support durchgeführt werden.

HINWEIS

Nur TwinCAT 3 Software verwenden!

Ein Firmware-Update von Beckhoff IO Geräten ist ausschließlich mit einer TwinCAT3-Installation durchzuführen. Es empfiehlt sich ein möglichst aktuelles Build, kostenlos zum Download verfügbar auf der [Beckhoff-Website](#).

Zum Firmware-Update kann TwinCAT im sog. FreeRun-Modus betrieben werden, eine kostenpflichtige Lizenz ist dazu nicht nötig.

Das für das Update vorgesehene Gerät kann in der Regel am Einbauort verbleiben; TwinCAT ist jedoch im FreeRun zu betreiben. Zudem ist auf eine störungsfreie EtherCAT Kommunikation zu achten (keine „LostFrames“ etc.).

Andere EtherCAT-Master-Software wie z. B. der EtherCAT-Konfigurator sind nicht zu verwenden, da sie unter Umständen nicht die komplexen Zusammenhänge beim Update von Firmware, EEPROM und ggf. weiteren Gerätebestandteilen unterstützen.

Speicherorte

In einem EtherCAT-Slave werden an bis zu drei Orten Daten für den Betrieb vorgehalten:

- Jeder EtherCAT-Slave hat eine Gerätebeschreibung, bestehend aus Identität (Name, Productcode), Timing-Vorgaben, Kommunikationseinstellungen u. a.
Diese Gerätebeschreibung (ESI; EtherCAT-Slave Information) kann von der Beckhoff Website im Downloadbereich als [Zip-Datei](#) heruntergeladen werden und in EtherCAT-Mastern zur Offline-Konfiguration verwendet werden, z. B. in TwinCAT.
Vor allem aber trägt jeder EtherCAT-Slave seine Gerätebeschreibung (ESI) elektronisch auslesbar in einem lokalen Speicherchip, dem einem sog. **ESI-EEPROM**. Beim Einschalten wird diese Beschreibung einerseits im Slave lokal geladen und teilt ihm seine Kommunikationskonfiguration mit, andererseits kann der EtherCAT-Master den Slave so identifizieren und u. a. die EtherCAT Kommunikation entsprechend einrichten.

HINWEIS

Applikationsspezifisches Beschreiben des ESI-EEPROM

Die ESI wird vom Gerätehersteller nach ETG-Standard entwickelt und für das entsprechende Produkt freigegeben.

- Bedeutung für die ESI-Datei: Eine applikationsseitige Veränderung (also durch den Anwender) ist nicht zulässig.

- Bedeutung für das ESI-EEPROM: Auch wenn technisch eine Beschreibbarkeit gegeben ist, dürfen die ESI-Teile im EEPROM und ggf. noch vorhandene freie Speicherbereiche über den normalen Update-Vorgang hinaus nicht verändert werden. Insbesondere für zyklische Speichervorgänge (Betriebsstundenzähler u. ä.) sind dezidierte Speicherprodukte wie EL6080 oder IPC-eigener NOVRAM zu verwenden.

- Je nach Funktionsumfang und Performance besitzen EtherCAT-Slaves einen oder mehrere lokale Controller zur Verarbeitung von IO-Daten. Das darauf laufende Programm ist die so genannte **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.

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 System Manager 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. ELxxxx-xxxx_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!

7.3.1 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:

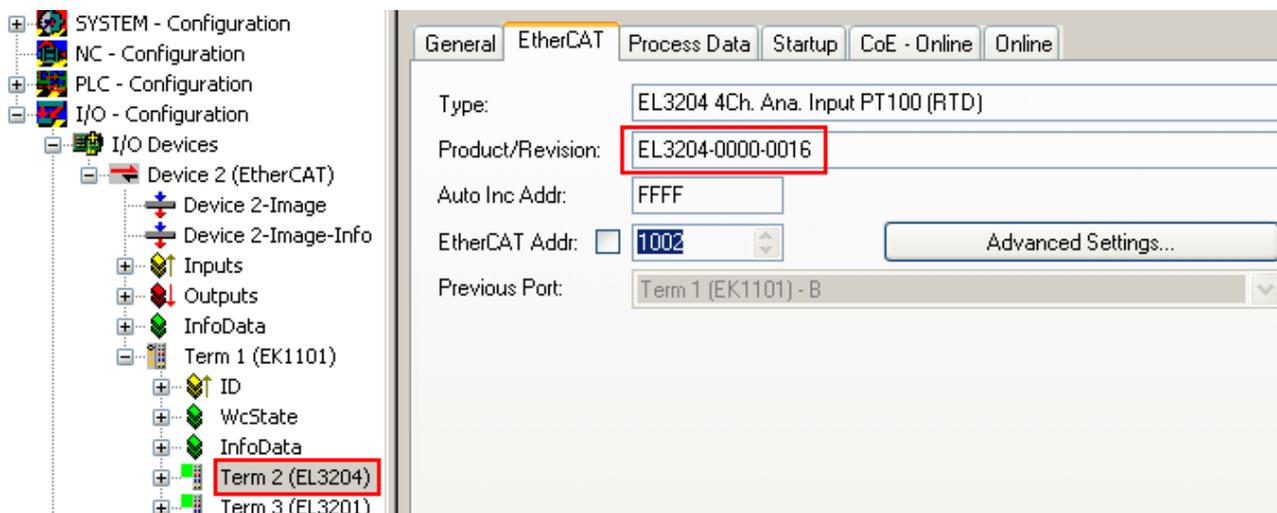


Abb. 178: Geräteerkennung aus Name EL3204-0000 und Revision -0016

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:

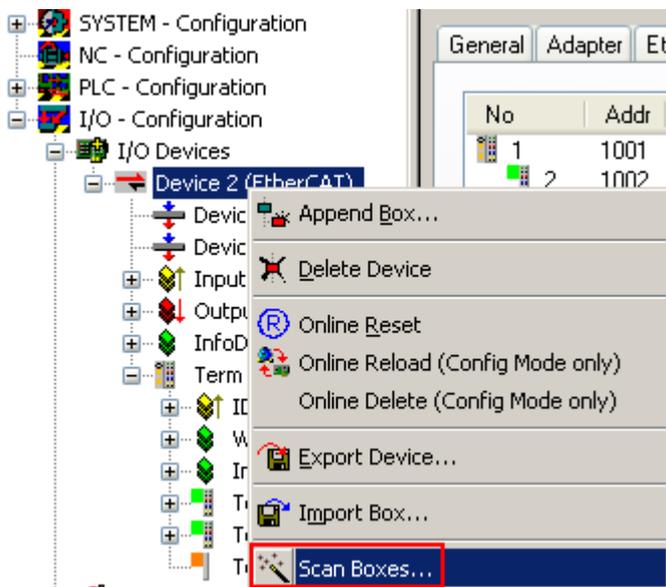


Abb. 179: Rechtsklick auf das EtherCAT-Gerät bewirkt das Scannen des unterlagerten Feldes

Wenn das gefundene Feld mit dem konfigurierten übereinstimmt, erscheint



Abb. 180: Konfiguration identisch

ansonsten erscheint ein Änderungsdialog, um die realen Angaben in die Konfiguration zu übernehmen.

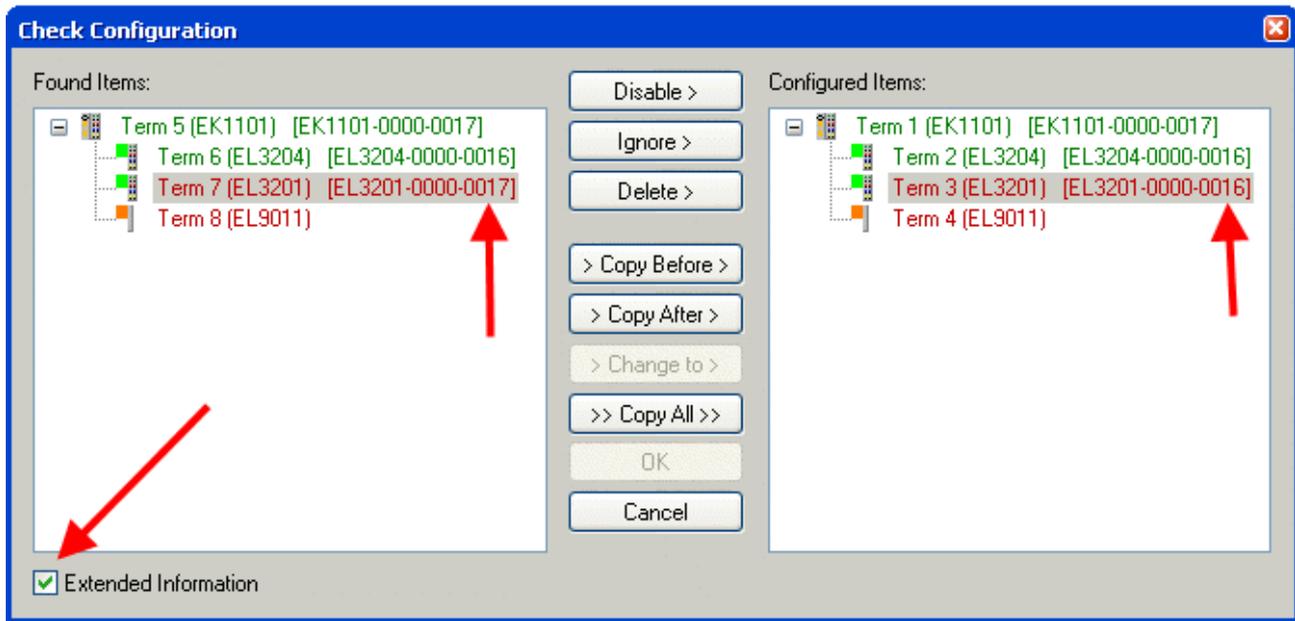


Abb. 181: Änderungsdialog

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*

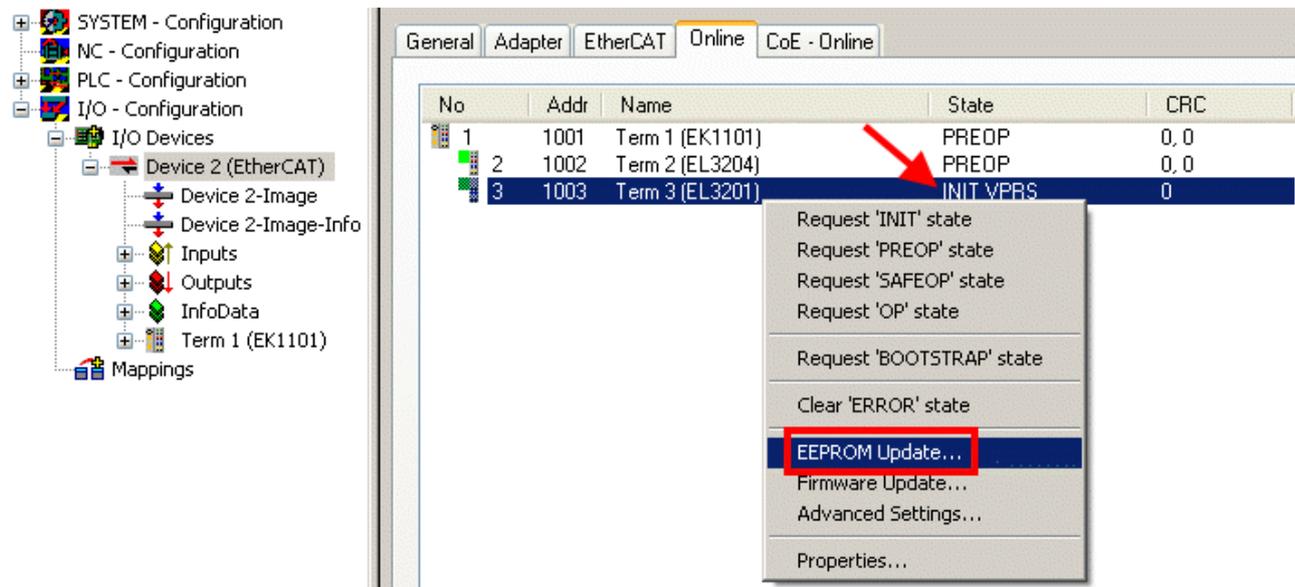


Abb. 182: EEPROM Update

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.

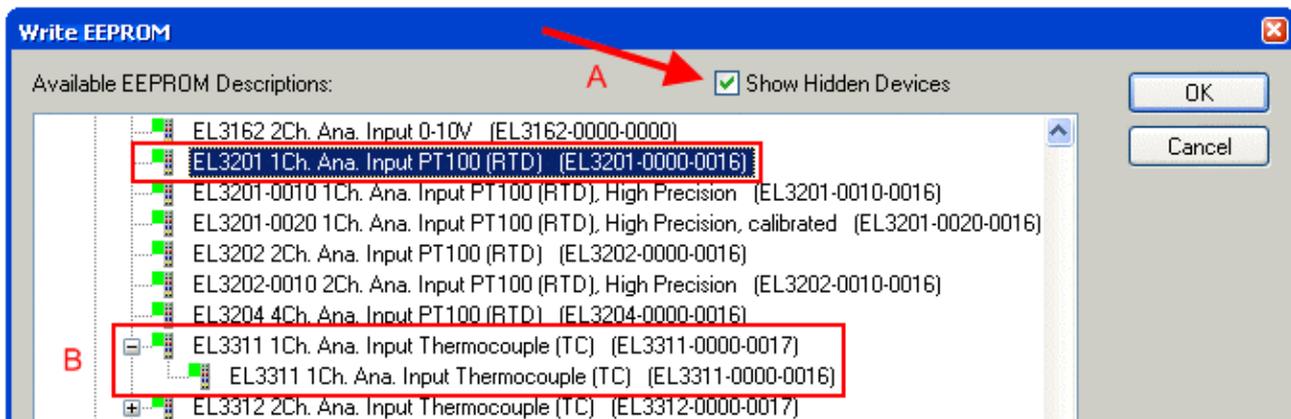


Abb. 183: Auswahl des neuen ESI

Ein Laufbalken im System Manager zeigt den Fortschritt - erst erfolgt das Schreiben, dann das Verifying.

● Änderung erst nach Neustart wirksam

i 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.

7.3.2 Erläuterungen zur Firmware

Versionsbestimmung der Firmware

Versionsbestimmung mit dem TwinCAT 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).

● CoE-Online und Offline-CoE

i 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.

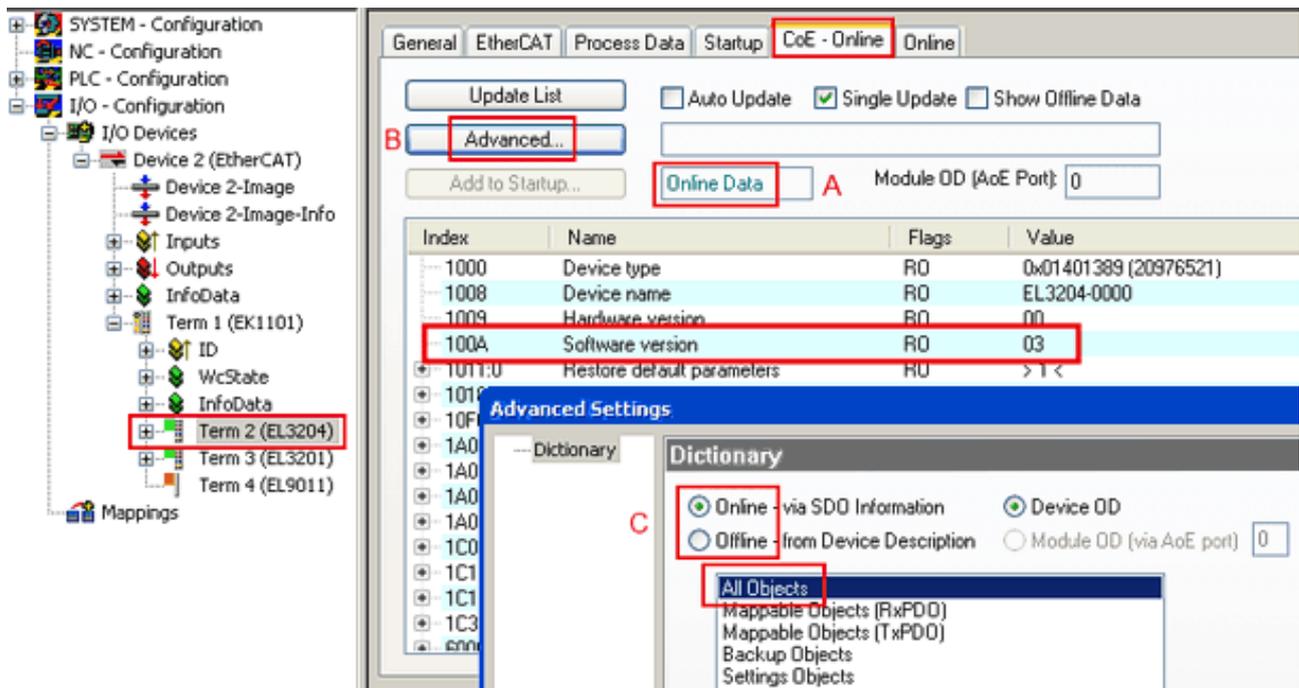


Abb. 184: Anzeige FW-Stand EL3204

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.

7.3.3 Update Controller-Firmware *.efw

● CoE-Verzeichnis

i 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*.

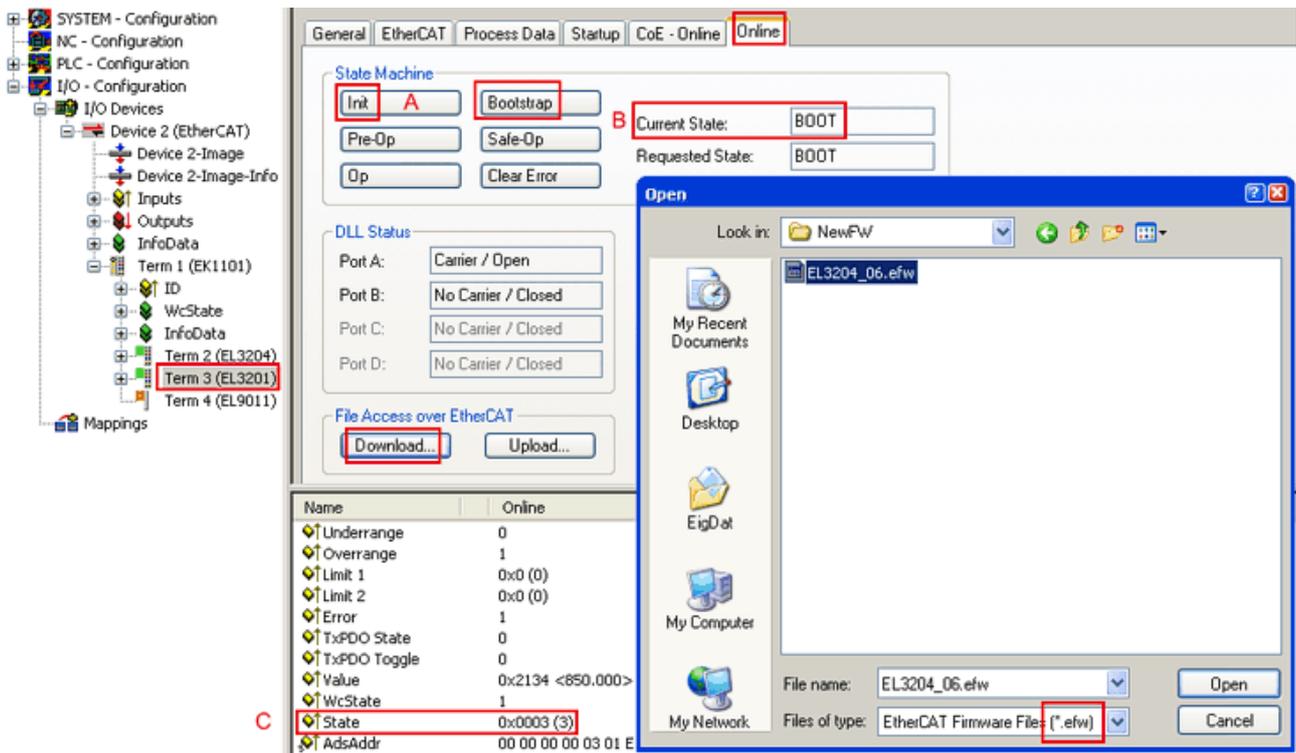
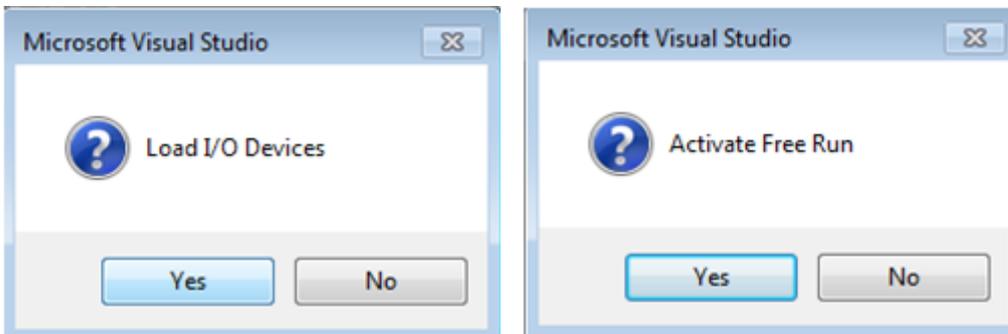


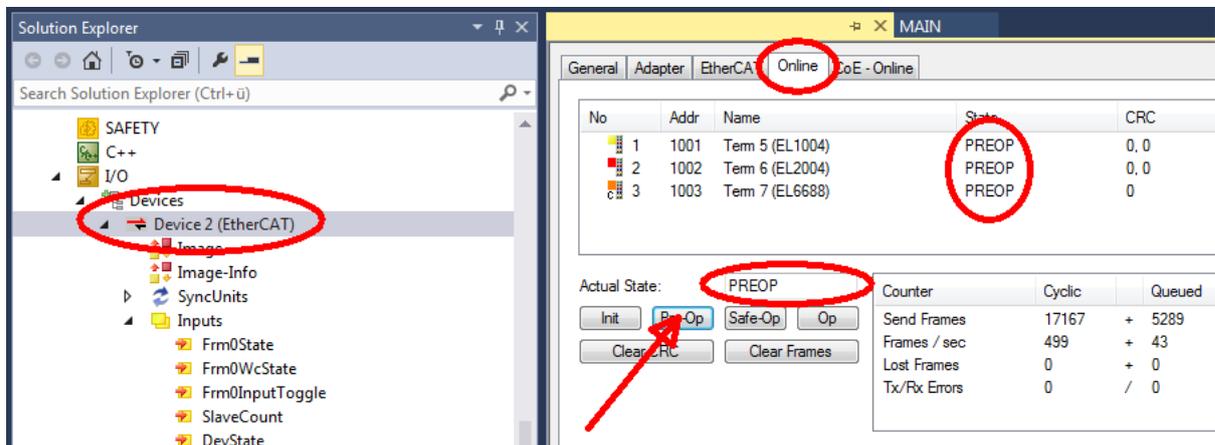
Abb. 185: Firmware Update

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 ≥ 1 ms schalten (default sind im ConfigMode 4 ms). Ein FW-Update während Echtzeitbetrieb ist nicht zu empfehlen.

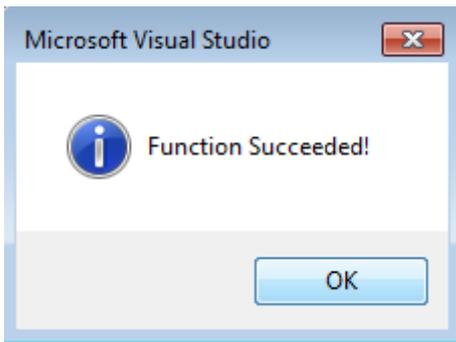


- EtherCAT-Master in PreOP schalten



- 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.

7.3.4 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.

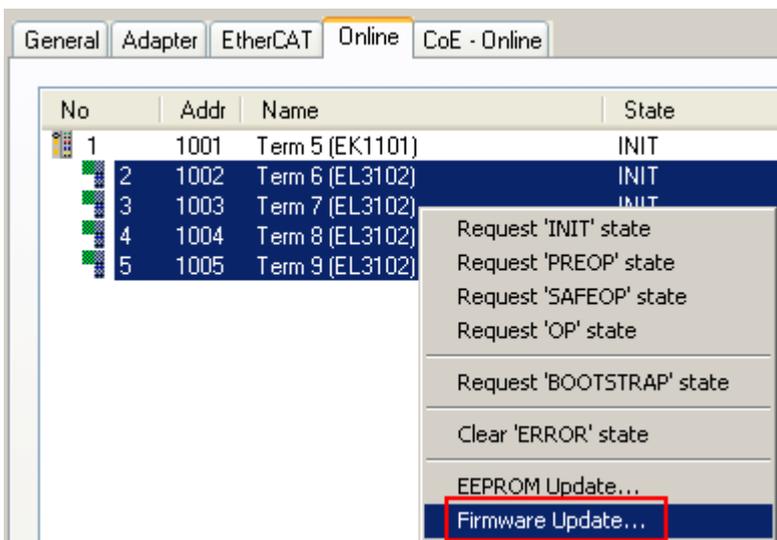


Abb. 186: Mehrfache Selektion und FW-Update

Wählen Sie dazu die betreffenden Slaves aus und führen Sie das Firmware-Update im BOOTSTRAP Modus wie o. a. aus.

7.4 Ergänzung zu Firmware (FW)/ FPGA Update

Ein FW Update kann auf zwei Möglichkeiten erfolgen: Zum einen kann ein Update über den Reiter ‚Online‘ und der Auswahl des Gerätes (Markierung von EL6695) aus der Liste erfolgen und im drop-down-Menü „Firmware Update..“ ausgewählt wird. Eine *.efw Datei kann dann über den entsprechenden Pfad geladen werden; ein Passwort ist dabei nicht nötig, kann aber optional vergeben werden. Anschließend müssen beide Klemmenseiten in den INIT State gesetzt werde, bevor sie wieder in den OP State zu setzen sind.

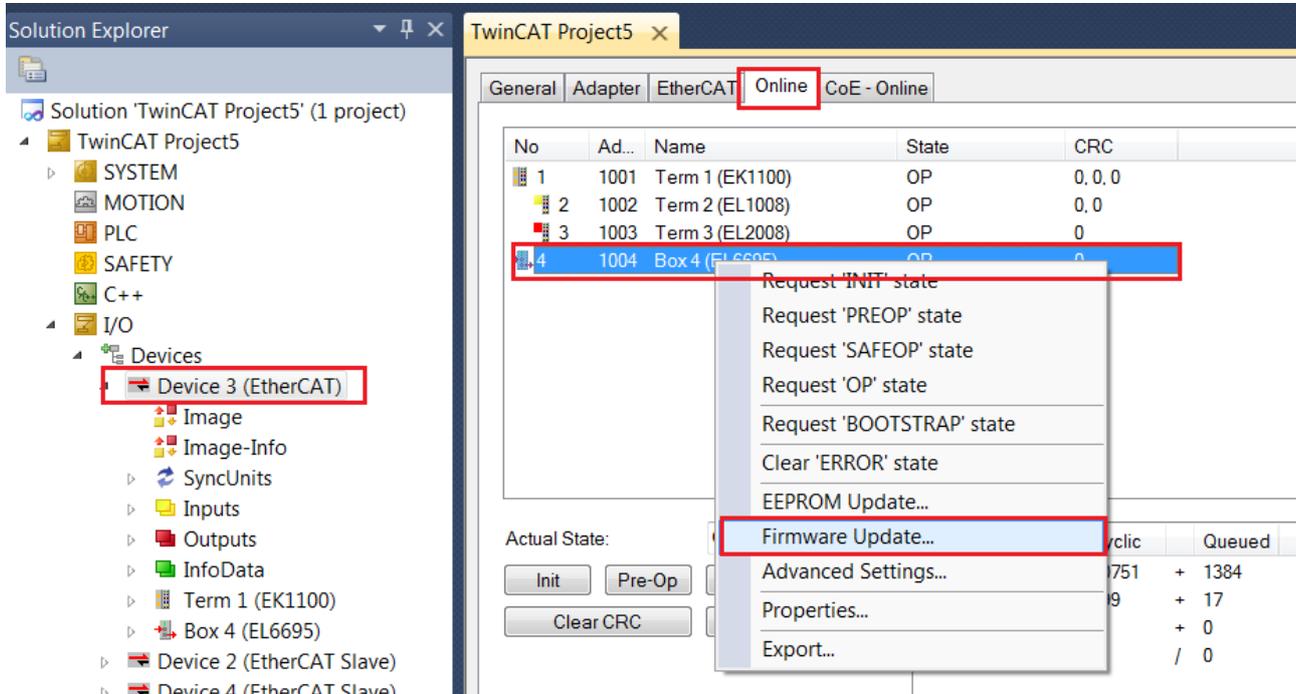


Abb. 187: Firmware Update über Registerkarte Online

Die zweite Möglichkeit ist direkt über die Klemme und den Reiter Online. Dazu muss die Klemme auf der ausgewählten Seite in den State BOOT gesetzt werden.

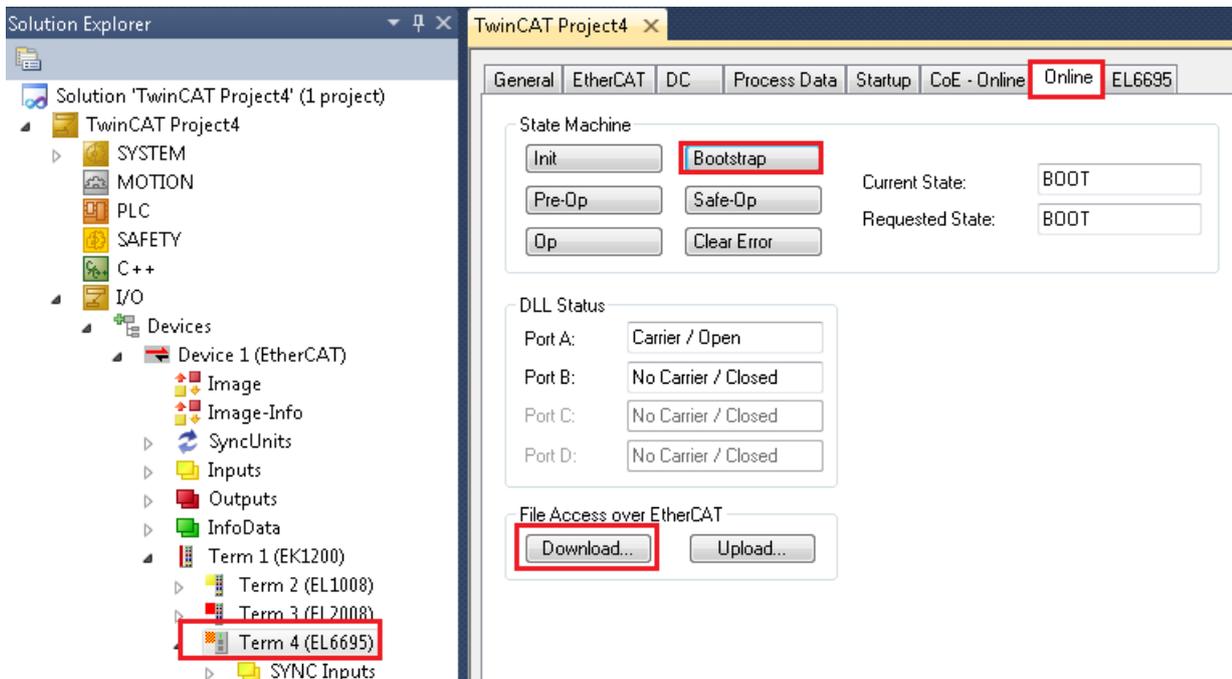


Abb. 188: Setzen des Zustandes „BOOT“

Anschließend kann über die Schaltfläche Download unter File Access over EtherCAT die *.efw-Datei in die Klemme eingespielt werden. Danach die Klemme in INIT und wieder OP setzen. Die andere Seite muss dann ebenfalls neu in INIT und OP gesetzt werden.

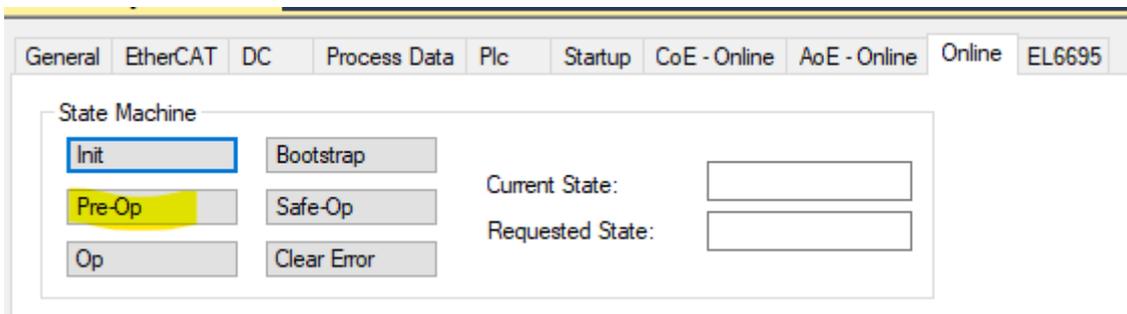
HINWEIS

Bei FW Update zu beachten (Beta FW)

- Es ist **vor** dem Update sicherzustellen, dass der EtherCAT Bus ohne „LostFrames“ oder Linkverlust arbeitet.
- Bei mehreren Klemmen EL6695 im System darf ein FW Update immer nur auf eine Klemme EL6695 „gleichzeitig“ ausgeführt werden. Es kommt sonst zur irreversiblen Beschädigung des Geräts und erfordert einen Austausch der Klemme über den Beckhoff-Service.
- Es darf während des Updates nicht zu einem Verlust der Spannungsversorgung kommen.

Für ein FPGA-Update ist wie folgt vorzugehen:

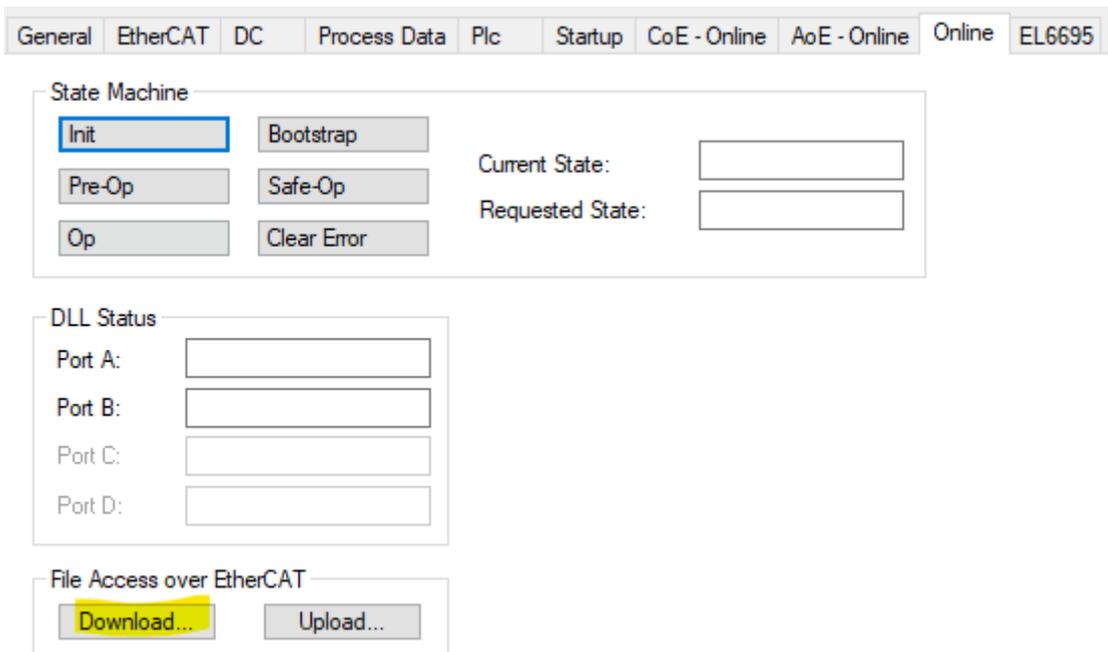
- Voraussetzung: Die EL6695 befindet sich primär oder sekundär im Konfigurationsbaum, TwinCAT ist im ConfigMode/FreeRun
- Klemme nach Pre-OP schalten:



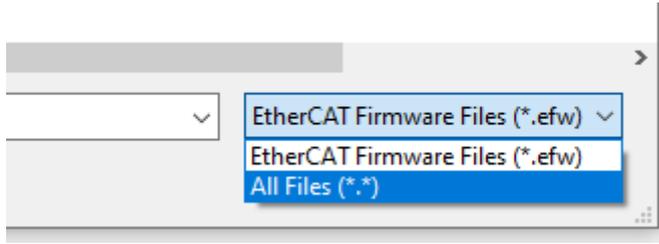
- In CoE-Objekt 0xF008 Code word den Wert 0x1006695 schreiben:

F000:0	Modular device profile	RO	> 2 <
F008	Code word	RW	0x00000000 (0)
F630:0	Active TxPdo Info	RO	> 8 <

- Auswahl Download unter „File Access over EtherCAT“ im Karteireiter „Online“:



- Auswahl der *.rbf FPGA-Datei, „All Files (*.*)“:



- Der Download kann mehrere Minuten in Anspruch nehmen

7.5 Wiederherstellen des Auslieferungszustandes

Die Klemme EL6695 verfügt über zwei Reset-Methoden:

- Per Kommando
- Über das CoE-Objekt 0x1011 (erst ab FW07)

7.5.1 Allgemeiner Geräte-Reset

Die in der Klemme EL6695 gespeicherten Einstellungen können über das Eintragen des Codes 0x33336695 in das CoE Objekt 0xF008 zurückgesetzt werden.

7.5.2 Wiederherstellen des Auslieferungszustandes

Um bei EtherCAT-Geräten („Slaves“) den Auslieferungszustand (Werkseinstellungen) der CoE-Objekte wiederherzustellen, kann per EtherCAT-Master (z. B. TwinCAT) das CoE-Objekt *Restore default parameters*, Subindex 001 verwendet werden (s. Abb. *Auswahl des PDO, Restore default parameters*)

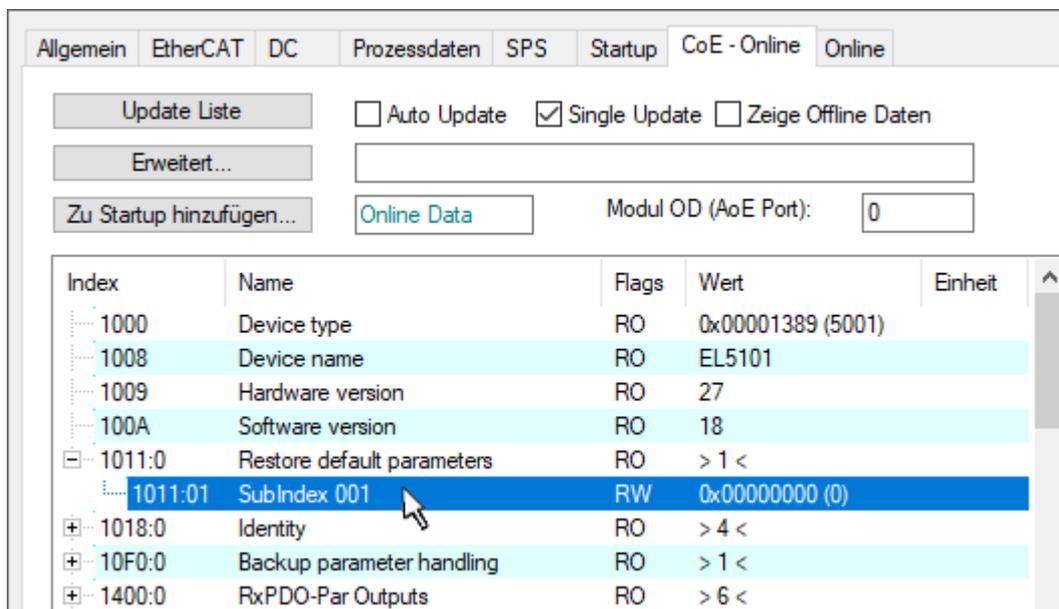


Abb. 189: Auswahl des PDO *Restore default parameters*

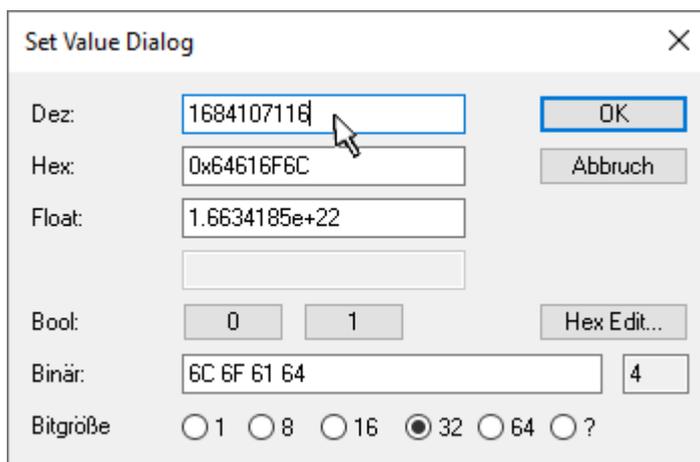


Abb. 190: Eingabe des Restore-Wertes im Set Value Dialog

Durch Doppelklick auf *SubIndex 001* gelangen Sie in den Set Value -Dialog. Tragen Sie im Feld *Dec* den Reset-Wert **1684107116** oder alternativ im Feld *Hex* den Wert **0x64616F6C** ein (ASCII: „load“) und bestätigen Sie mit OK (Abb. *Eingabe des Restore-Wertes im Set Value Dialog*).

- Alle veränderbaren CoE-Einträge werden auf die Default-Werte zurückgesetzt.
- Die Werte können nur erfolgreich zurückgesetzt werden, wenn der Reset auf das Online-CoE, d. h. auf dem Slave direkt angewendet wird. Im Offline-CoE können keine Werte verändert werden.
- TwinCAT muss dazu im Zustand RUN oder CONFIG/FreeRun befinden, d. h. EtherCAT Datenaustausch findet statt. Auf fehlerfreie EtherCAT-Übertragung ist zu achten.
- Es findet keine gesonderte Bestätigung durch den Reset statt. Zur Kontrolle kann zuvor ein veränderbares Objekt umgestellt werden.
- Dieser Reset-Vorgang kann auch als erster Eintrag in die StartUp-Liste des Slaves mit aufgenommen werden, z. B. im Statusübergang PREOP->SAFEOP oder, wie in Abb. *CoE-Reset als StartUp-Eintrag*, bei SAFEOP->OP

Alle Backup-Objekte werden so in den Auslieferungszustand zurückgesetzt.

● Alternativer Restore-Wert

i Bei einigen Klemmen älterer Bauart (FW Erstellung ca. vor 2007) lassen sich die Backup-Objekte mit einem alternativen Restore-Wert umstellen: Dezimalwert: 1819238756, Hexadezimalwert: 0x6C6F6164.

Eine falsche Eingabe des Restore-Wertes zeigt keine Wirkung!

7.6 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 Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unseren Internetseiten: www.beckhoff.com

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Support

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

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

Hotline: +49 5246 963 157
E-Mail: support@beckhoff.com
Internet: www.beckhoff.com/support

Service

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

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

Hotline: +49 5246 963 460
E-Mail: service@beckhoff.com
Internet: www.beckhoff.com/service

Unternehmenszentrale Deutschland

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon: +49 5246 963 0
E-Mail: info@beckhoff.com
Internet: www.beckhoff.com

Mehr Informationen:
www.beckhoff.com/EL6695

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

