BECKHOFF New Automation Technology

Dokumentation | DE

EL6731, EL6731-0010

Master/Slave-Klemmen für PROFIBUS

21.01.2024 | Version: 3.6

Inhaltsverzeichnis

1 Vorwort 5
1.1 Hinweise zur Dokumentation 5
1.2 Wegweiser durch die Dokumentation 6
1.3 Sicherheitshinweise 7
1.4 Ausgabestände der Dokumentation 8
1.5 Versionsidentifikation von EtherCAT-Geräten 9
1.5.1 Allgemeine Hinweise zur Kennzeichnung 9
1.5.2 Versionsidentifikation von EL-Klemmen 10
1.5.3 Beckhoff Identification Code (BIC) 11
1.5.4 Elektronischer Zugriff auf den BIC (eBIC) 13

2 Produktbeschreibung 15
2.1 Einführung 15
2.2 Technische Daten 16

3 Montage und Verdrahtung 17
3.1 Hinweise zum ESD-Schutz 17
3.2 Explosionsschutz 18
3.2.1 ATEX - Besondere Bedingungen (erweiterter Temperaturbereich) 18
3.2.2 IECEx - Besondere Bedingungen 20
3.2.3 Weiterführende Dokumentation zu ATEX und IECEx 21
3.3 UL-Hinweise 22
3.4 Einbaulagen 23
3.5 Positionierung von passiven Klemmen 25
3.6 Montage und Demontage - Zughebelentriegelung 26
3.7 Montage und Demontage - Frontentriegelung oben 28
3.8 Hinweis Spannungsversorgung 30
3.9 PROFIBUS-Verkabelung 31
3.10 PROFIBUS-Anschluss 33
3.11 Entsorgung 35

4 Grundlagen der Kommunikation 36
4.1 EtherCAT-Grundlagen 36
4.2 Allgemeine Hinweise zur Watchdog-Einstellung 36
4.3 EtherCAT State Machine 37
4.4 CoE-Interface 39

5 Parametrierung und Inbetriebnahme 44
5.1 EL6731 - PROFIBUS Masterklemme 44
5.1.1 PROFIBUS Protokolle 44
5.1.2 PROFIBUS DP 46
5.1.3 PROFIBUS MC 48
5.1.4 Synchronisierung 49
5.1.5 ADS (azyklische Dienste) 51
5.1.6 TwinCAT (2.1x) System Manager 61
5.1.7 Betrieb von PROFIdrive MC-Antrieben 110
5.1.8 Diagnose und Fehlerbeschreibung 122

5.1.9 Slave-Priorisierung/mehrere DP-Zyklen ... 137
5.2 EL6731-0010 - PROFIBUS Slave-Klemme ... 139
5.2.1 EL6731-0010 - LED Beschreibung ... 142
6 EtherCAT Kommunikation EL6731-00x0 ... 144
6.1 PROFIBUS Master ... 144
6.1.1 State Machine ... 144
6.1.2 Synchronisierung ... 148
6.1.3 Objektbeschreibung und Parametrierung ... 152
6.2 PROFIBUS Slave ... 170
6.2.1 State Machine ... 170
6.2.2 Objektbeschreibung und Parametrierung ... 174
7 Anhang ... 182
7.1 EtherCAT AL Status Codes ... 182
7.2 Firmware Kompatibilität ... 183
7.3 Firmware Update EL/ES/ELM/EM/EP/EPP/ERPxxxx ... 185
7.3.1 Gerätebeschreibung ESI-File/XML ... 186
7.3.2 Erläuterungen zur Firmware ... 189
7.3.3 Update Controller-Firmware *.efw ... 190
7.3.4 FPGA-Firmware *.rbf ... 192
7.3.5 Gleichzeitiges Update mehrerer EtherCAT-Geräte ... 196
7.4 Support und Service ... 197

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 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)	• Systemübersicht • EtherCAT-Grundlagen • Kabel-Redundanz • Hot Connect • Konfiguration von EtherCAT-Geräten
Explosionsschutz für Klemmensysteme (PDF)	Hinweise zum Einsatz der Beckhoff Klemmensysteme in explosionsgefährdeten Bereichen gemäß ATEX und IECEx
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.

1.3 Sicherheitshinweise

Sicherheitsbestimmungen

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

Haftungsausschluss

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

Qualifikation des Personals

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

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.

1.4 Ausgabestände der Dokumentation

Version	Kommentar
3.6	• Update Kapitel „Technische Daten • Update „Parametrierung und Inbetriebnahme” • Update Struktur • Revisionsstand aktualisiert
3.5	• Revisionsstand aktualisiert
3.4	• Kapitel „Empfohlene Tragschiene“ entfernt • Update Kapitel „Technische Daten“ • Kapitel „IECEx - Besondere Bedingungen hinzugefügt“ • Update Struktur
3.3	• Update Kapitel „Technische Daten“ • Update Struktur • Revisionsstand aktualisiert
3.2	• Update Kapitel „ADS (azyklische Dienste)“ • Update Struktur
3.1	• Update Kapitel „Technische Daten“ • Update Struktur
3.0	• Update Kapitel „Technische Daten“ • Update Kapitel „LED-Beschreibung“ • Update Struktur • Revisionsstand aktualisiert
2.9	• Update Kapitel „Technische Daten“ • Update Struktur • Revisionsstand aktualisiert
2.8	• Update Struktur
2.7	• Kapitel „ADS-Fehler-Codes“ aktualisiert • Update Struktur • Revisionsstand aktualisiert
2.6	• Update Kapitel "Technische Daten"
2.5	• Kapitel „DP-State der Slaves“ aktualisiert • Update Struktur • Revisionsstand aktualisiert
2.4	• Kapitel „Master-Diagnosedaten“ aktualisiert • Update Struktur • Revisionsstand aktualisiert
2.3	• Kapitel „Einführung“ aktualisiert • Kapitel „LED-Beschreibung“ aktualisiert • Kapitel „PROFIBUS Anschluss“ aktualisiert
2.2	• Update Kapitel "PROFIBUS Protokolle" und "ADS-Interface" • Kapitel "S5-FDL-Kommunikation" entfernt
2.1	• Update Kapitel "Technische Daten" • Update Kapitel "ADS-Interface" • Kapitel "Hinweise zum ESD-Schutz“ ergänzt • Update Struktur • Revisionsstand aktualisiert
2.0	• Migration
0.1 - 1.7	*archiviert*

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 2014/01 außen auf den IP20-Klemmen aufgebracht, siehe Abb. *„EL5021 EL-Klemme, Standard IP20-IO-Gerät mit Chargennummer und Revisionskennzeichnung (seit 2014/01)“*.
- Typ, Version und Revision werden als dezimale Zahlen gelesen, auch wenn sie technisch hexadezimal gespeichert werden.

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 auf einem Aufkleber angebracht ist. Diese Seriennummer gibt den Bauzustand im Auslieferungszustand an und kennzeichnet somit eine ganze Produktions-Charge, unterscheidet aber nicht die Module einer Charge.

Aufbau der Seriennummer: **KK YY FF HH**	Beispiel mit Seriennummer 12 06 3A 02:
KK - Produktionswoche (Kalenderwoche)	12 - Produktionswoche 12
YY - Produktionsjahr	06 - Produktionsjahr 2006
FF - Firmware-Stand	3A - Firmware-Stand 3A
HH - Hardware-Stand	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.

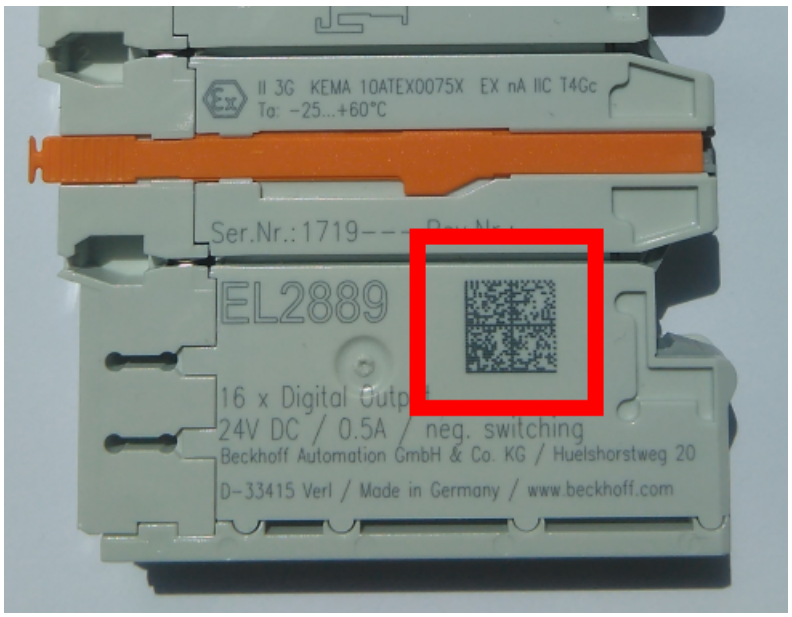


Abb. 2: BIC als Data Matrix Code (DMC, Code-Schema ECC200)

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
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**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

Entsprechend als DMC:

Abb. 3: Beispiel-DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **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. Chargenbezeichungen 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 Information 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 elektronisch angesprochen werden kann.

K-Bus Geräte (IP20, IP67)

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

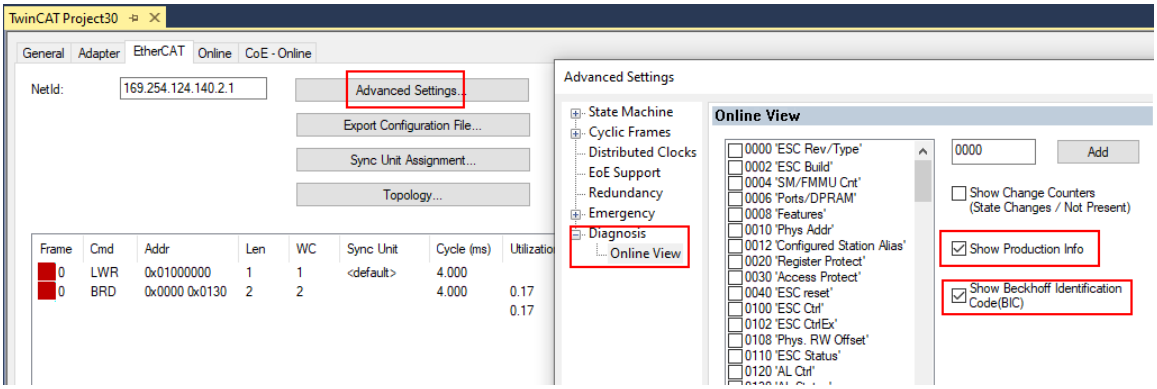
EtherCAT-Geräte (IP20, IP67)

Alle Beckhoff EtherCAT-Geräte haben ein sogenanntes ESI-EEPROM, dass 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).

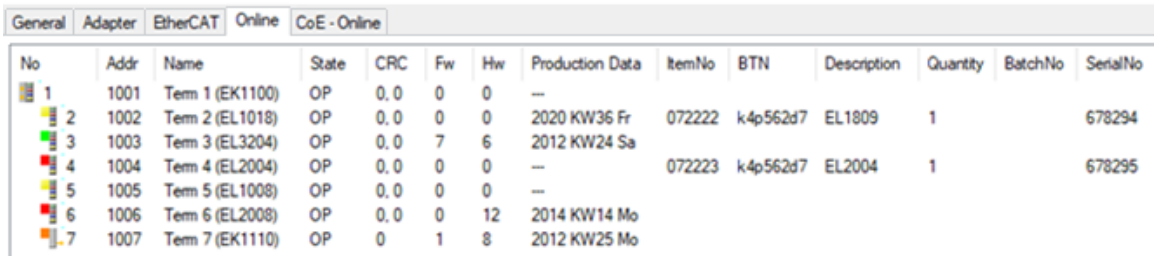
In das ESI-EEPROM wird durch Beckhoff auch die eBIC gespeichert. Die Einführung des eBIC in die Beckhoff IO Produktion (Klemmen, Box-Module) erfolgt ab 2020; Stand 2023 ist die Umsetzung weitgehend abgeschlossen.

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:

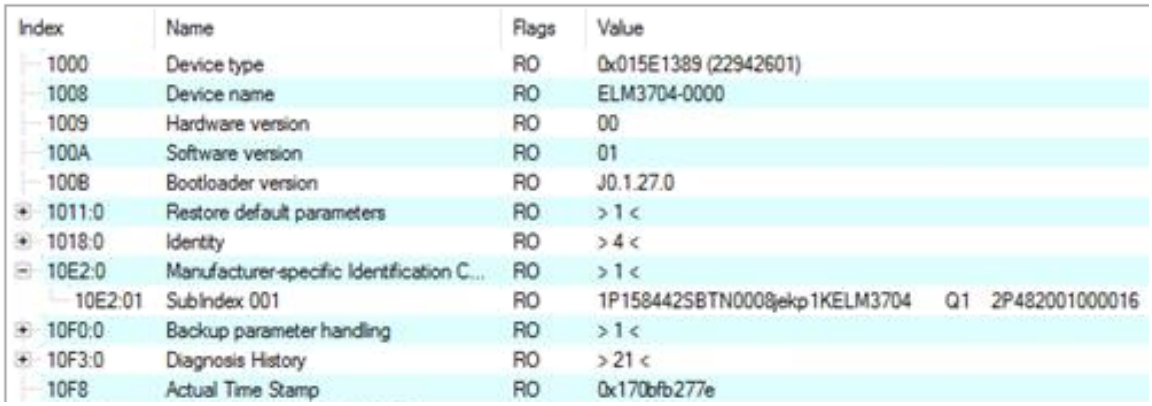


  - Die BTN und Inhalte daraus werden dann angezeigt:



  - 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.

- 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	1P158442SBTN0008jekp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

  - 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_Utilities* zur Verfügung
  - F_SplitBIC: Die Funktion zerlegt den Beckhoff Identification Code (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.
- Technischer 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 herstellerspezifische 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
  - Sind mehrere ESC in einem Gerät verbaut die hierarchisch angeordnet sind, trägt nur der TopLevel 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 des TopLevel-Geräts, 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

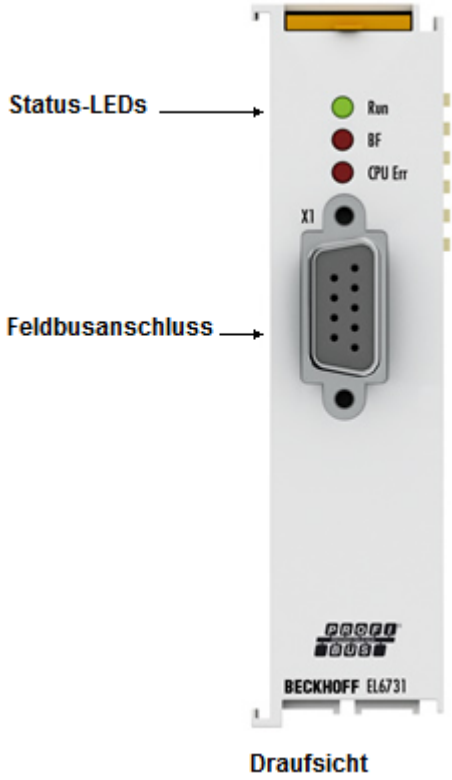


Abb. 4: EL6731

Master- und Slave-Klemmen für PROFIBUS

Die Master- und Slave-Klemmen für PROFIBUS entsprechen der Beckhoff PCI-Karte FC3101. Durch den Anschluss via Ethernet kann im PC auf PCI-Slots verzichtet werden.

Die EL6731 beherrscht das PROFIBUS-Protokoll mit allen Features und ermöglicht im EtherCAT-Klemmenverbund die Integration beliebiger Profibus-Geräte. Dank selbst entwickeltem PROFIBUS-Chip verfügen die Klemmen über den aktuellen Stand der PROFIBUS-Technologie – samt hochpräzisem Isochron-Modus für die Achssteuerung und erweiterten Diagnosemöglichkeiten. Diese Master sind die einzigen, die unterschiedliche Poll-Raten für die Slaves unterstützen.

Eigenschaften:

- Zykluszeiten ab 200 µs möglich
- PROFIBUS DP, PROFIBUS DP-V1, PROFIBUS DP-V2
- Master, Slave und Profibus-Monitor bis 12 MBit/s
- leistungsfähige Parameter- und Diagnoseschnittstellen
- Error-Management für jeden Busteilnehmer frei konfigurierbar
- Einlesen der Buskonfiguration und automatische Zuordnung der GSD-Dateien möglich

2.2 Technische Daten

Technische Daten	EL6731	EL6731-0010
Bus-System	PROFIBUS DP (Standard), PROFIBUS DP-V1 (Cl. 1+2: azyklische Dienste, Alarme), PROFIBUS DP-V2, PROFIBUS MC (Äquidistanz)	
Variante	Master	Slave
Anzahl Feldbuskanäle	1	
Übertragungsrate	9,6 kBit/s bis 12 MBit/s[1) 2)]	
Bus-Interface	1 x Buchse D-Sub, 9-polig, galvanisch entkoppelt	
Busteilnehmer	maximal 125 Slaves mit bis zu 244 Bytes Input-, Output-, Parameter-, Konfigurations-, Diagnosedaten je Slave	
Zykluszeit	durch CDL-Konzept sind unterschiedliche DP-Zykluszeiten je Slave möglich	
Prozessabbild	Summe max.: 1,4 kByte Input- und 1,4 kByte Output-Daten	
Diagnose	Status-LEDs	
Spannungsversorgung	über den E-Bus	
Stromaufnahme aus dem E-Bus	typ. 350 mA (bis Hardwarestand 22)	typ. 350 mA (bis Hardwarestand 20)
	typ. 200 mA ab Hardwarestand 23)	typ. 200 mA (ab Hardwarestand 21)
Potenzialtrennung	500 V (E-Bus / PROFIBUS)	
Konfiguration	mit TwinCAT System-Manager	
Gewicht	ca. 70 g	
Zulässiger Umgebungstemperaturbereich im Betrieb	-25 °C ... +60 °C (erweiterter Temperaturbereich) (für EL6731-0010 ab Hardwarestand 17)	
Zulässiger Umgebungstemperaturbereich bei Lagerung	-40 °C ... +85 °C	
Zulässige relative Luftfeuchtigkeit	95%, keine Betauung	
Abmessungen (B x H x T)	ca. 26 mm x 100 mm x 52 mm (Breite angereiht: 23 mm)	
Montage [▶ 26]	auf 35 mm Tragschiene nach EN 60715	
Vibrations- / Schockfestigkeit	gemäß EN 60068-2-6 / EN 60068-2-27	
EMV-Festigkeit / Aussendung	gemäß EN 61000-6-2 / EN 61000-6-4	
Schutzart	IP20	
Einbaulage	beliebig	
Zulassungen / Kennzeichnungen*)	CE, UKCA, EAC cULus [▶ 22], ATEX [▶ 18], IECEx [▶ 20]	

[1)] Hardware-Stände 29 - 32 [▶ 183] unterstützen 6 MBit/s bei der EL6731 nicht.
[2)] Hardware-Stände 29 - 30 [▶ 183] unterstützen 6 MBit/s bei der EL6731-0010 nicht.

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

Ex-Kennzeichnungen

Standard	Kennzeichnung
ATEX	II 3 G Ex nA IIC T4 Gc
IECEx	Ex nA IIC T4 Gc

3 Montage und Verdrahtung

3.1 Hinweise zum ESD-Schutz

HINWEIS

Zerstörung der Geräte durch elektrostatische Aufladung möglich!

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

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

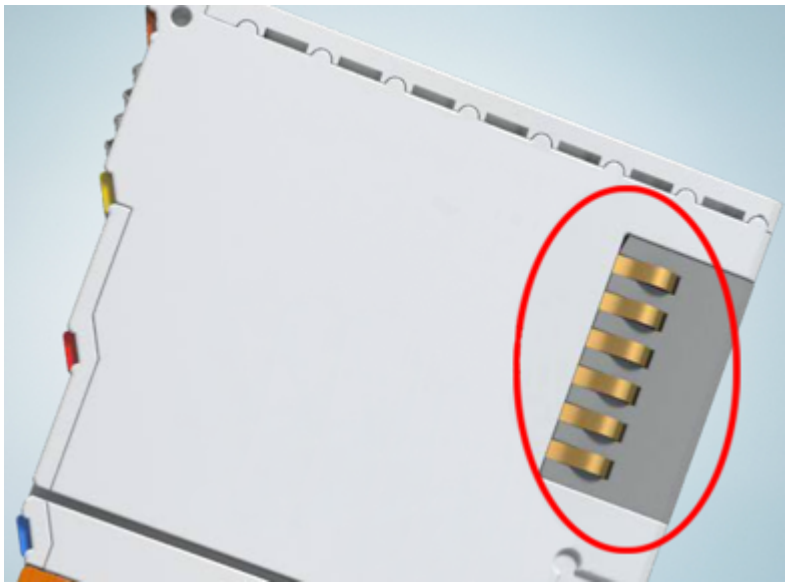


Abb. 5: Federkontakte der Beckhoff I/O-Komponenten

3.2 Explosionsschutz

3.2.1 ATEX - Besondere Bedingungen (erweiterter Temperaturbereich)

⚠ WARNUNG

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

- Die zertifizierten Komponenten sind in ein geeignetes Gehäuse zu errichten, das eine Schutzart von mindestens IP54 gemäß EN 60079-15 gewährleistet! Dabei sind die Umgebungsbedingungen bei der Verwendung zu berücksichtigen!
- Für Staub (nur die Feldbuskomponenten der Zertifikatsnummer KEMA 10ATEX0075 X Issue 9): Das Gerät ist in ein geeignetes Gehäuse einzubauen, das eine Schutzart von IP54 gemäß EN 60079-31 für Gruppe IIIA oder IIIB und IP6X für Gruppe IIIC bietet, wobei die Umgebungsbedingungen, unter denen das Gerät verwendet wird, zu berücksichtigen sind!
- Wenn die Temperaturen bei Nennbetrieb an den Einführungsstellen der Kabel, Leitungen oder Rohrleitungen höher als 70°C oder an den Aderverzweigungsstellen höher als 80°C ist, so müssen Kabel ausgewählt werden, deren Temperaturdaten den tatsächlich gemessenen Temperaturwerten entsprechen!
- Beachten Sie für Beckhoff-Feldbuskomponenten mit erweitertem Temperaturbereich (ET) beim Einsatz in explosionsgefährdeten Bereichen den zulässigen Umgebungstemperaturbereich von -25 bis 60°C!
- Es müssen Maßnahmen zum Schutz gegen Überschreitung der Nennbetriebsspannung durch kurzzeitige Störspannungen um mehr als 40% getroffen werden!
- Die einzelnen Klemmen dürfen nur aus dem Busklemmensystem gezogen oder entfernt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Anschlüsse der zertifizierten Komponenten dürfen nur verbunden oder unterbrochen werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Sicherung der Einspeiseklemmen KL92xx/EL92xx dürfen nur gewechselt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Adresswahlschalter und ID-Switche dürfen nur eingestellt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!

Normen

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

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

Kennzeichnung

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

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

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: -25 ... +60°C
(nur für Feldbuskomponenten mit Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

oder

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

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: -25 ... +60°C
(nur für Feldbuskomponenten mit Zertifikatsnummer KEMA 10ATEX0075 X Issue 9)

3.2.2 IECEx - Besondere Bedingungen

⚠ WARNUNG

Beachten Sie die besonderen Bedingungen für die bestimmungsgemäße Verwendung von Beckhoff-Feldbuskomponenten in explosionsgefährdeten Bereichen!

- Für Gas: Die Komponenten sind in ein geeignetes Gehäuse zu errichten, das gemäß EN 60079-15 eine Schutzart von IP54 gewährleistet! Dabei sind die Umgebungsbedingungen bei der Verwendung zu berücksichtigen!
- Für Staub (nur für Feldbuskomponenten der Zertifikatsnummer IECEx DEK 16.0078X Issue 3): Die Komponenten sind in einem geeigneten Gehäuse zu errichten, das gemäß EN 60079-31 für die Gruppe IIIA oder IIIB eine Schutzart von IP54 oder für die Gruppe IIIC eine Schutzart von IP6X gewährleistet. Dabei sind die Umgebungsbedingungen bei der Verwendung zu berücksichtigen!
- Die Komponenten dürfen nur in einem Bereich mit mindestens Verschmutzungsgrad 2 gemäß IEC 60664-1 verwendet werden!
- Es sind Vorkehrungen zu treffen, um zu verhindern, dass die Nennspannung durch transiente Störungen von mehr als 119 V überschritten wird!
- Wenn die Temperaturen bei Nennbetrieb an den Einführungsstellen der Kabel, Leitungen oder Rohrleitungen höher als 70°C oder an den Aderverzweigungsstellen höher als 80°C ist, so müssen Kabel ausgewählt werden, deren Temperaturdaten den tatsächlich gemessenen Temperaturwerten entsprechen!
- Beachten Sie für Beckhoff-Feldbuskomponenten beim Einsatz in explosionsgefährdeten Bereichen den zulässigen Umgebungstemperaturbereich!
- Die einzelnen Klemmen dürfen nur aus dem Busklemmensystem gezogen oder entfernt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Anschlüsse der zertifizierten Komponenten dürfen nur verbunden oder unterbrochen werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Adresswahlschalter und ID-Switche dürfen nur eingestellt werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!
- Die Frontklappe von zertifizierten Geräten darf nur geöffnet werden, wenn die Versorgungsspannung abgeschaltet wurde bzw. bei Sicherstellung einer nicht-explosionsfähigen Atmosphäre!

Normen

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

- EN 60079-0:2011
- EN 60079-15:2010
- EN 60079-31:2013 (nur für Zertifikatsnummer IECEx DEK 16.0078X Issue 3)

Kennzeichnung

Die gemäß IECEx für den explosionsgefährdeten Bereich zertifizierten Beckhoff-Feldbuskomponenten tragen die folgende Kennzeichnung:

Kennzeichnung für Feldbuskomponenten der Zertifikat-Nr. IECEx DEK 16.0078X Issue 3:	**IECEx DEK 16.0078 X** **Ex nA IIC T4 Gc** **Ex tc IIIC T135°C Dc**
Kennzeichnung für Feldbuskomponenten von Zertifikaten mit späteren Ausgaben:	**IECEx DEK 16.0078 X** **Ex nA IIC T4 Gc**

3.2.3 Weiterführende Dokumentation zu ATEX und IECEx

HINWEIS	
	Weiterführende Dokumentation zum Explosionsschutz gemäß ATEX und IECEx Beachten Sie auch die weiterführende Dokumentation **Explosionsschutz für Klemmensysteme** Hinweise zum Einsatz der Beckhoff Klemmensysteme in explosionsgefährdeten Bereichen gemäß ATEX und IECEx, die Ihnen auf der Beckhoff-Homepage www.beckhoff.de im Download-Bereich Ihres Produktes zum Download zur Verfügung steht!

3.3 UL-Hinweise

⚠ VORSICHT
Application The modules are intended for use with Beckhoff's UL Listed EtherCAT System only.

⚠ VORSICHT
Examination 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).

⚠ VORSICHT
For devices with Ethernet connectors Not for connection to telecommunication circuits.

Grundlagen

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

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

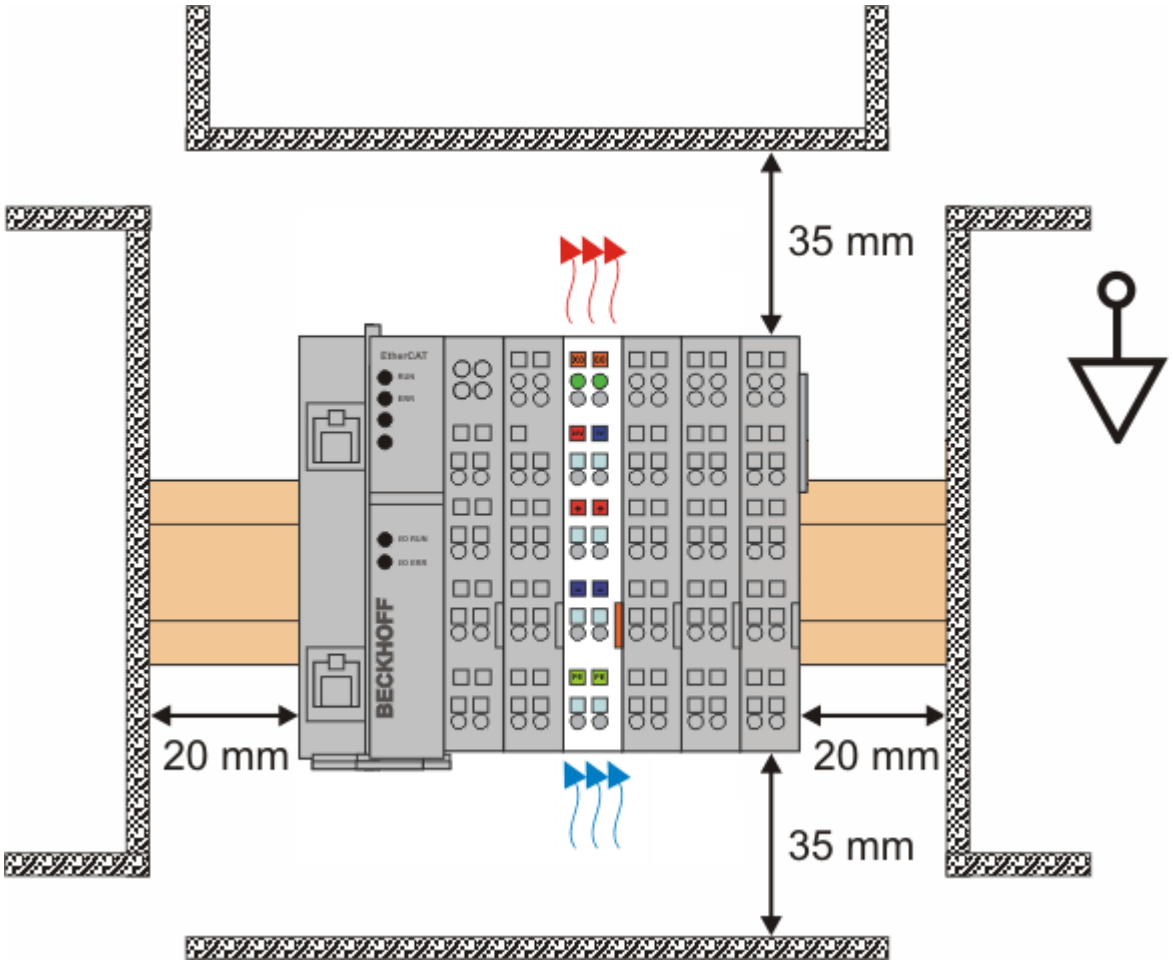


Abb. 6: Empfohlene Abstände bei Standard-Einbaulage

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.

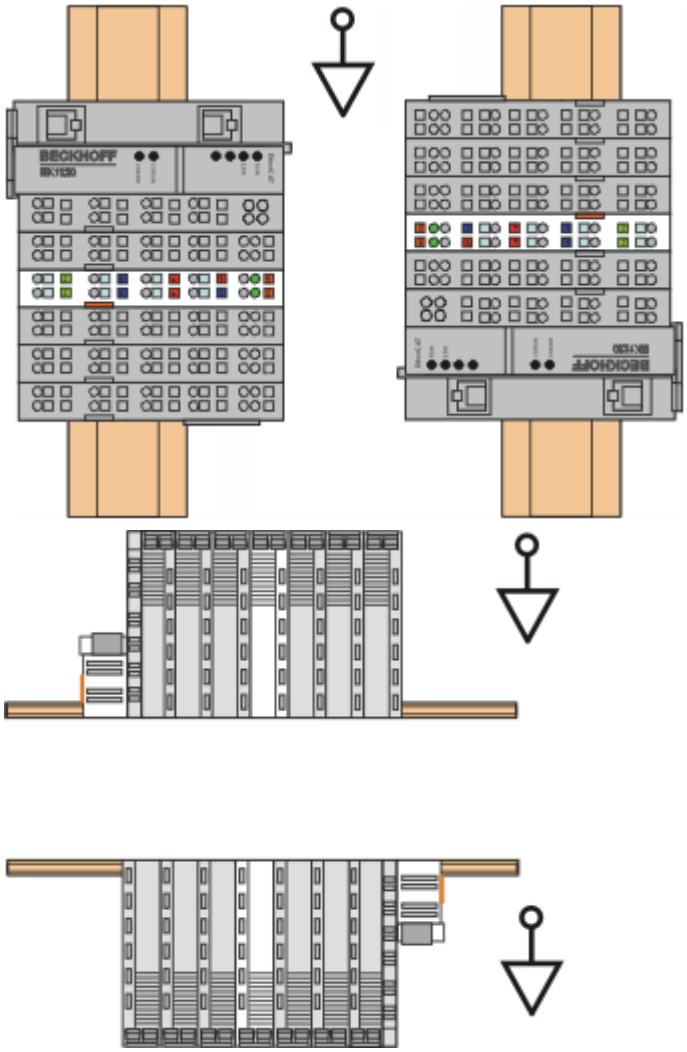


Abb. 7: Weitere Einbaulagen

3.5 Positionierung von passiven Klemmen

Hinweis zur Positionierung von passiven Klemmen im Busklemmenblock

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

Beispiele für die Positionierung von passiven Klemmen (hell eingefärbt)

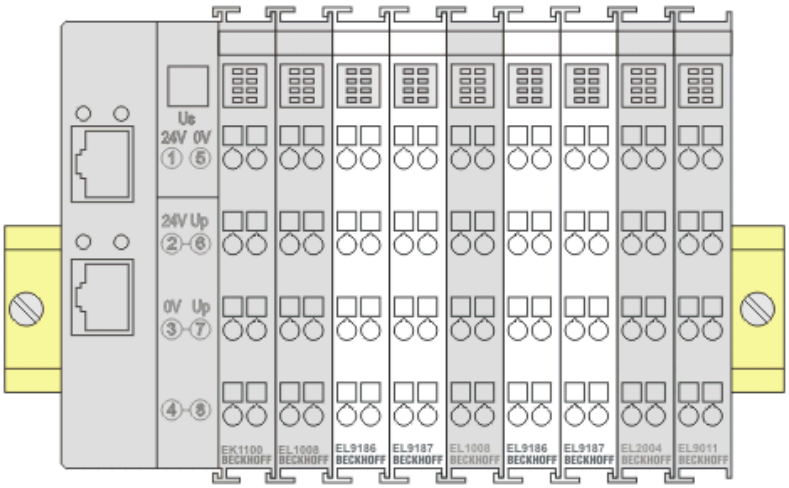


Abb. 8: Korrekte Positionierung

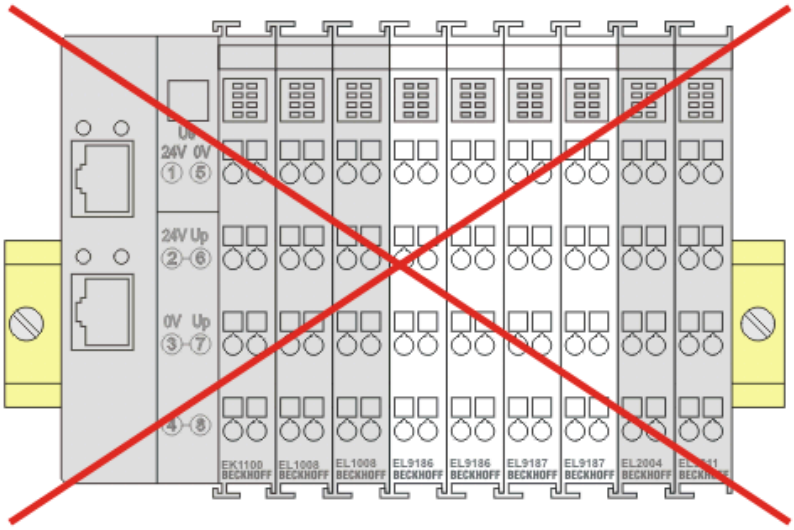


Abb. 9: Inkorrekte Positionierung

3.6 Montage und Demontage - Zughebelentriegelung

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

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.

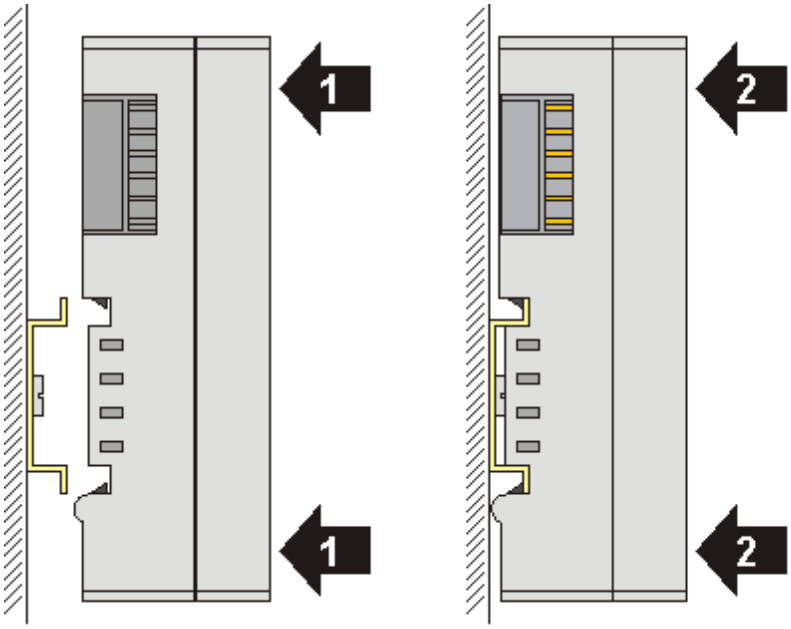
⚠ WARNUNG

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

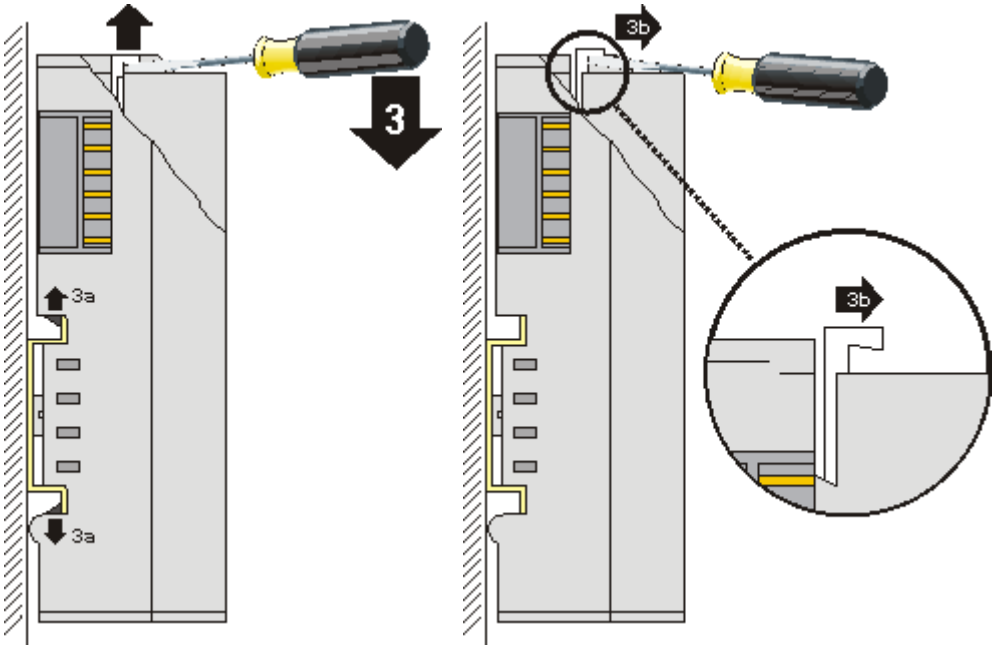


und drücken Sie (1) das Klemmenmodul gegen die Tragschiene, bis es auf der Tragschiene Einrastet (2).

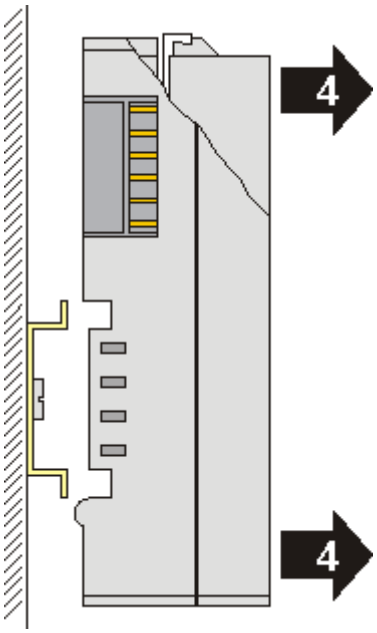
- Schließen Sie die Leitungen an.

Demontage

- Entfernen Sie alle Leitungen. Dank der KM/EM-Steckverbinder müssen Sie hierzu nicht alle Leitungen einzeln entfernen, sondern pro KM/EM-Steckverbinder nur 2 Schrauben lösen um diese abziehen zu können (stehende Verdrahtung)!
- Hebeln Sie auf der linken Seite des Klemmenmoduls mit einem Schraubendreher (3) den Entriegelungshaken nach oben. Dabei
  - ziehen sich über einen internen Mechanismus die beiden Rastnasen (3a) an der Hutschiene ins Klemmenmodul zurück,
  - bewegt sich der Entriegelungshaken nach vorne (3b) und rastet ein



- Bei 32- und 64-kanaligen Klemmenmodulen (KMxxx4 und KMxxx8 bzw. EMxxx4 und EMxxx8) hebeln Sie nun den zweiten Entriegelungshaken auf der rechten Seite des Klemmenmoduls auf die gleiche Weise nach oben.
- Ziehen Sie (4) das Klemmenmodul von der Montagefläche weg.



3.7 Montage und Demontage - Frontentriegelung oben

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

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.

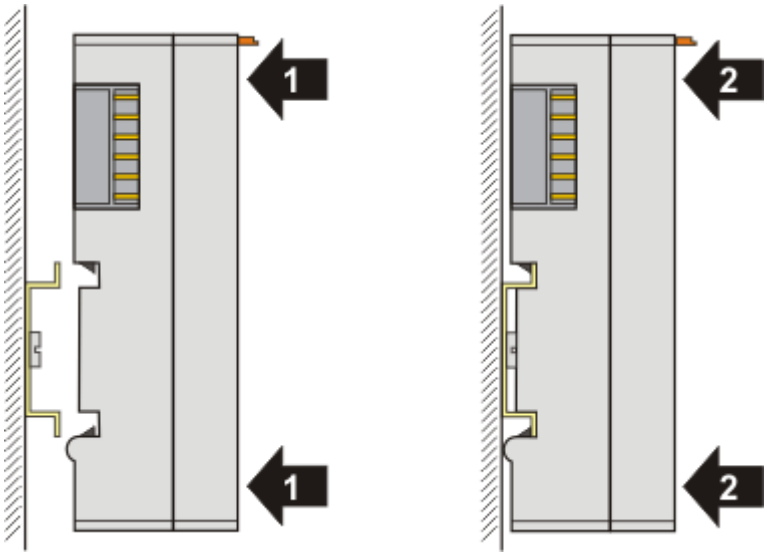
⚠ WARNUNG

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

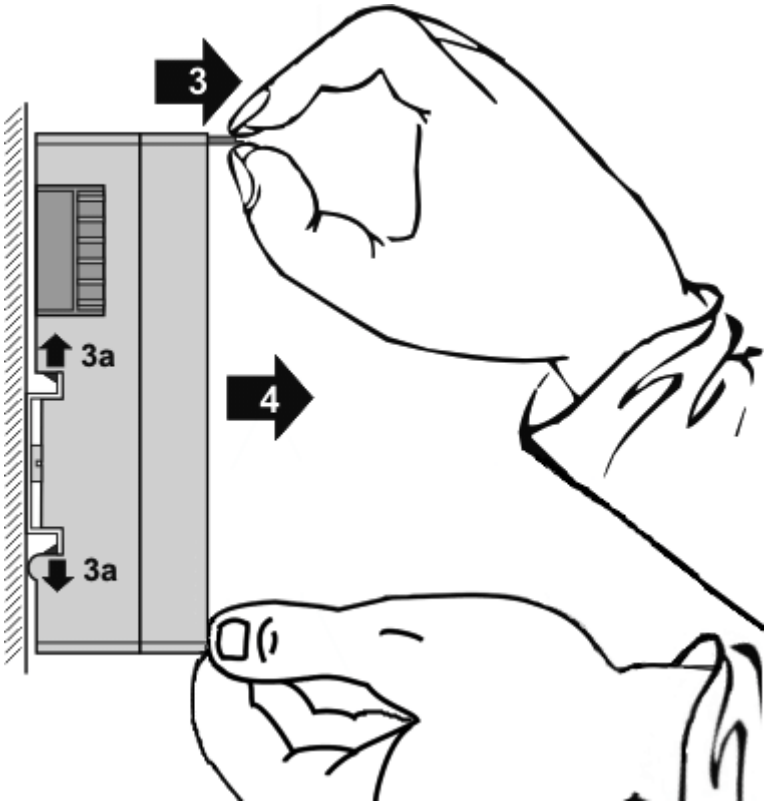


und drücken Sie (1) das Klemmenmodul gegen die Tragschiene, bis es auf der Tragschiene einrastet (2).

- 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

3.8 Hinweis Spannungsversorgung

⚠ WARNUNG

Spannungsversorgung aus SELV/PELV-Netzteil!

Zur Versorgung dieses Geräts müssen SELV/PELV-Stromkreise (Schutzkleinspannung, Sicherheitskleinspannung) 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 (Safety Extra Low Voltage) liefert sichere elektrische Trennung und Begrenzung der Spannung ohne Verbindung zum Schutzleiter,
  eine PELV-Versorgung (Protective Extra Low Voltage) benötigt zusätzlich eine sichere Verbindung zum Schutzleiter.

3.9 PROFIBUS-Verkabelung

Die physikalische Datenübertragung ist in der PROFIBUS-Norm definiert (siehe PROFIBUS Schicht 1: Physical Layer).

Der Einsatzbereich eines Feldbus-Systems wird wesentlich durch die Wahl des Übertragungs-Mediums und der physikalischen Busschnittstelle bestimmt. Neben den Anforderungen an die Übertragungssicherheit sind die Aufwendungen für Beschaffung und Installation des Buskabels von entscheidender Bedeutung. Die PROFIBUS-Norm sieht daher verschiedene Ausprägungen der Übertragungstechnik unter Beibehaltung eines einheitlichen Busprotokolls vor.

Kabelgebundene Übertragung

Diese Version, gemäß dem US-Standard EIA RS-485, wurde als Grundversion für Anwendungen im Bereich der Fertigungstechnik, Gebäudeleittechnik und Antriebstechnik festgelegt. Es wird ein verdrilltes Kupferkabel mit einem Leiterpaar verwendet. Die Abschirmung kann in Abhängigkeit des beabsichtigten Einsatzgebietes (EMV-Gesichtspunkte beachten) entfallen.

Es stehen zwei Leitungstypen mit unterschiedlichen Höchstleitungslängen zur Verfügung (siehe Tabelle RS485).

RS485 - Grundlegende Eigenschaften

RS-485 Übertragungstechnik nach PROFIBUS-Norm	
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden, Stichleitungen sind möglich
Medium	Abgeschirmtes verdrilltes Kabel, Schirmung darf abhängig von den Umgebungsbedingungen (EMV) entfallen
Anzahl der Stationen	32 Stationen in jedem Segment ohne Repeater. Mit Repeater erweiterbar bis 127 Stationen
Max. Bus Länge ohne Repeater	100 m bei 12 MBit/s 200 m bei 1500 KBit/s, bis zu 1,2 km bei 93,75 KBit/s
Max. Bus Länge mit Repeater	Durch Leitungsverstärker (Repeater) kann die max. Buslänge bis zu 10 km vergrößert werden. Die Anzahl der möglichen Repeater ist mindestens 3 und kann je nach Hersteller bis zu 10 betragen.
Übertragungsgeschwindigkeit (in Stufen einstellbar)	9,6 kBit/s; 19,2 kBit/s; 93,75 kBit/s; 187,5 kBit/s; 500 kBit/s; 1500 kBit/s; 12 MBit/s
Steckverbinder	9-Pin D-Sub-Steckverbinder für IP20 M12 Rundsteckverbinder für IP65/67

Verkabelung für PROFIBUS-DP und PROFIBUS-FMS

Beachten Sie die besonderen Anforderungen an das Datenkabel bei Baud-Raten von mehr als 1,5 MBaud. Das richtige Kabel ist Grundvoraussetzung für den störungsfreien Betrieb des Bussystems. Bei der Verwendung des normalen 1,5 MBaud-Kabels kann es durch Reflexionen und zu großer Dämpfung zu erstaunlichen Phänomenen kommen. Zum Beispiel bekommt eine angeschlossene PROFIBUS-Station keine Verbindung, kann diese aber nach Abziehen der benachbarten Station wieder aufnehmen. Oder es kommt zu Übertragungsfehlern, wenn ein bestimmtes Bitmuster übertragen wird. Das kann bedeuten, dass der PROFIBUS ohne Funktion der Anlage störungsfrei arbeitet und nach dem Hochlauf zufällig Busfehler meldet. Eine Reduzierung der Baud-Rate (< 93,75 kBaud) beseitigt das geschilderte Fehlerverhalten.

Führt die Verringerung der Baud-Rate nicht zur Beseitigung des Fehlers, liegt in häufig ein Verdrahtungsfehler vor. Die beiden Datenleitungen sind an einem oder mehreren Steckern gedreht oder Abschlusswiderstände sind nicht oder an falschen Stellen aktiviert.

Empfohlene Kabel

Mit den vorkonfektionierten Kabeln von Beckhoff vereinfacht sich die Installation erheblich! Verdrahtungsfehler werden vermieden und die Inbetriebnahme führt schneller zum Erfolg. Das Beckhoff-Programm umfasst Feldbuskabel, Stromversorgungskabel und Sensorkabel sowie Zubehör wie Abschlusswiderstände und T-Stücke. Ebenso sind jedoch auch feldkonfektionierbare Stecker und Kabel erhältlich.

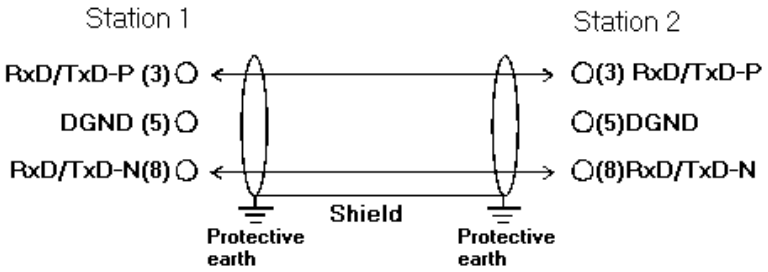


Abb. 10: Belegung des PROFIBUS-Kabels

Abschlusswiderstände

In Systemen mit mehr als zwei Stationen werden alle Teilnehmer parallel verdrahtet. An den Leitungsenden muss das PROFIBUS-Kabel in jedem Fall mit Widerständen abgeschlossen werden, um Reflexionen und damit Übertragungsprobleme zu vermeiden.

Längenausdehnung

Die Busleitung ist in der EN 50170 spezifiziert. Daraus ergibt sich die nachfolgende Längenausdehnung eines Bussegment.

Baud-Rate in kBits/sec	9,6	19,2	93,75	187,5	500	1500	12000
Leitungslänge in m	1200	1200	1200	1000	400	200	100

Stichleitungen bis 1500 kBaud <6,6 m, bei 12 MBaud sollten keine Stichleitungen verwendet werden.

Bussegment

Ein Bussegment besteht aus maximal 32 Teilnehmern. An einem PROFIBUS-Netzwerk sind 126 Teilnehmer erlaubt. Um diese Anzahl zu erreichen sind Repeater erforderlich, die das Signal auffrischen. Dabei wird jeder Repeater wie ein Teilnehmer angesehen.

IP-Link ist das Sub-Bussystem der Feldbus Boxen dessen Topologie eine Ringstruktur ist. In den Koppler Modulen (IP230x-Bxxx oder IP230x-Cxxx) befindet sich ein IP-Link Master, an den bis zu 120 Erweiterungsmodule (IExxxx) angeschlossen werden dürfen. Der Abstand zwischen zwei Modulen darf dabei 5 m nicht überschreiten. Achten Sie bei der Planung und Installation der Module, dass aufgrund der Ringstruktur das letzte Modul wieder am IP-Link Master angeschlossen werden muss.

Einbaurichtlinie

Beachten Sie bei der Montage der Module und beim Verlegen der Leitung die technischen Richtlinien der PROFIBUS-Nutzerorganisation e.V. zu PROFIBUS-DP/FMS (siehe https://www.profibus.de).

Überprüfung der PROFIBUS-Verkabelung

Ein PROFIBUS-Kabel (bzw. ein Kabel-Segment bei Verwendung von Repeatern) kann mit ein paar einfachen Widerstandsmessungen überprüft werden. Dazu sollte das Kabel von allen Stationen abgezogen werden:

1. Widerstand zwischen A und B am Anfang der Leitung: ca. 110 Ohm
2. Widerstand zwischen A und B am Ende der Leitung: ca. 110 Ohm
3. Widerstand zwischen A am Anfang und A am Ende der Leitung: ca. 0 Ohm
4. Widerstand zwischen B am Anfang und B am Ende der Leitung: ca. 0 Ohm
5. Widerstand zwischen Schirm am Anfang und Schirm am Ende der Leitung: ca. 0 Ohm

Falls diese Messungen erfolgreich sind, ist das Kabel in Ordnung. Wenn trotzdem noch Bus-Störungen auftreten, liegt es meistens an EMV-Störungen. Beachten Sie die Installationshinweise der PROFIBUS-Nutzer-Organisation (https://www.profibus.de).

3.10 PROFIBUS-Anschluss

M12-Rundsteckverbinder

Die M12-Buchse ist invers codiert und besitzt 5 Stifte. Stift 1 überträgt 5 V_{DC} und Stift 3 überträgt GND für den aktiven Abschlusswiderstand. Diese dürfen auf keinen Fall für andere Funktionen missbraucht werden, da dies zu Zerstörung des Gerätes führen kann.

Stift 2 und Stift 4 übertragen die Signale des PROFIBUS. Diese dürfen auf keinen Fall getauscht werden, da sonst die Kommunikation gestört ist. Stift 5 ist überträgt den Schirm (Shield) der kapazitiv mit der Grundfläche der Feldbus Box verbunden ist.

Pinbelegung M12 Buchse (-B310)

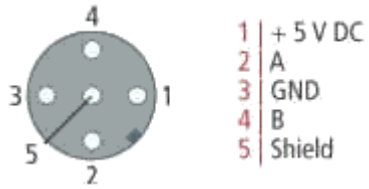


Abb. 11: Pinbelegung M12 Buchse (-B310)

Pinbelegung M12 Buchse/Stecker (-B318)

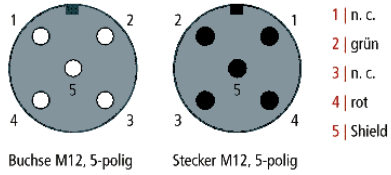


Abb. 12: Pinbelegung M12 Buchse/Stecker (-B318)

Neunpoliger D-Sub

Stift 6 überträgt 5 V_{DC} und Stift 5 GND für den aktiven Abschlusswiderstand. Diese dürfen auf keinen Fall für andere Funktionen missbraucht werden, da dies zu Zerstörung des Gerätes führen kann.

Stift 3 und Stift 8 übertragen die Signale des PROFIBUS. Diese dürfen auf keinen Fall getauscht werden, da sonst die Kommunikation gestört ist.

Pinbelegung der PROFIBUS D-Sub Buchse

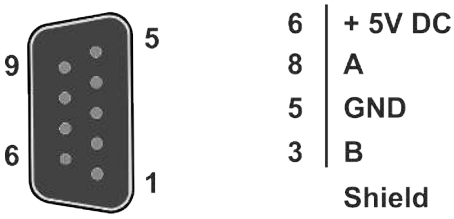


Abb. 13: Pinbelegung der PROFIBUS D-Sub Buchse

Leitungsfarben PROFIBUS

PROFIBUS Leitung	M12	D-Sub
B rot	Stift 4	Stift 3
A grün	Stift 2	Stift 8

Anschluss der Feldbus Box Module

Der Anschluss der Feldbus Box Module erfolgt entweder direkt oder mittels T-Stück (oder Y-Stück).

Die B318 Serie verfügt über jeweils eine Buchse und einen Stecker, d.h. hier wird der PROFIBUS in dem Modul weitergeleitet. Die Versorgungsspannung (+5 V_{DC}) für den Abschluss-Widerstand liegt nur auf der Buchse an. Der Abschluss-Widerstand ZS1000-1610 steht nur als Stecker zur Verfügung!

Die ankommende PROFIBUS-Leitung sollte stets mit einer Buchse enden.

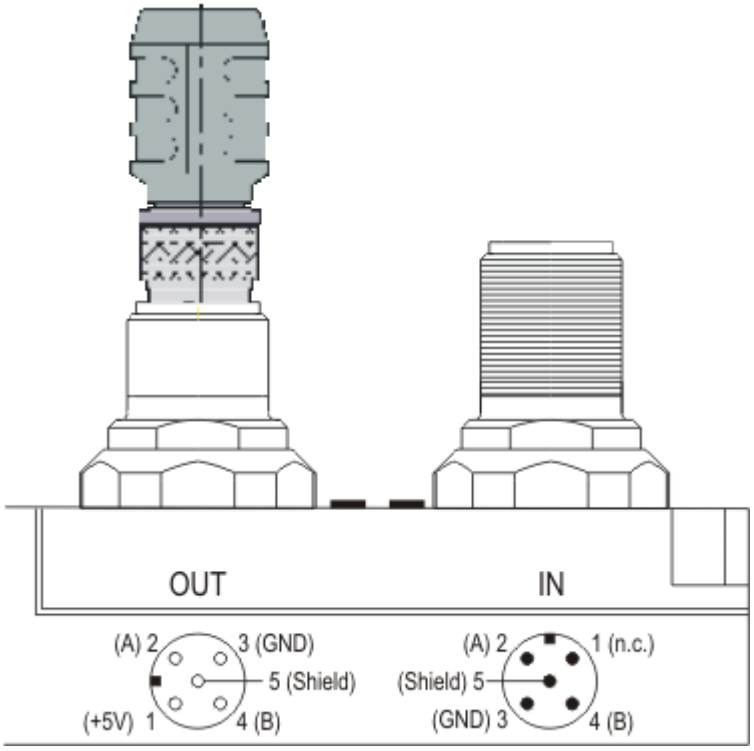


Abb. 14: Pin-Belegung Buchse/Stecker Feldbus Box Module

Es stehen zwei T-Stücke zur Verfügung:

- ZS1031-2600 mit +5 V_{DC} Weiterleitung zur Versorgung des Abschluss-Widerstandes
- ZS1031-2610 ohne +5 V_{DC} Weiterleitung

3.11 Entsorgung

Mit einer durchgestrichenen Abfalltonne gekennzeichnete 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.

4 Grundlagen der Kommunikation

4.1 EtherCAT-Grundlagen

Grundlagen zum Feldbus EtherCAT entnehmen Sie bitte der EtherCAT System-Dokumentation.

4.2 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 (so vorhanden) in einen gegebenenfalls vorgebbaren Zustand schaltet, in Abhängigkeit vom Gerät und Einstellung z. B. auf FALSE (aus) oder einen Ausgabewert.

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

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

Deren Zeiten werden in TwinCAT wie folgt einzeln parametriert:

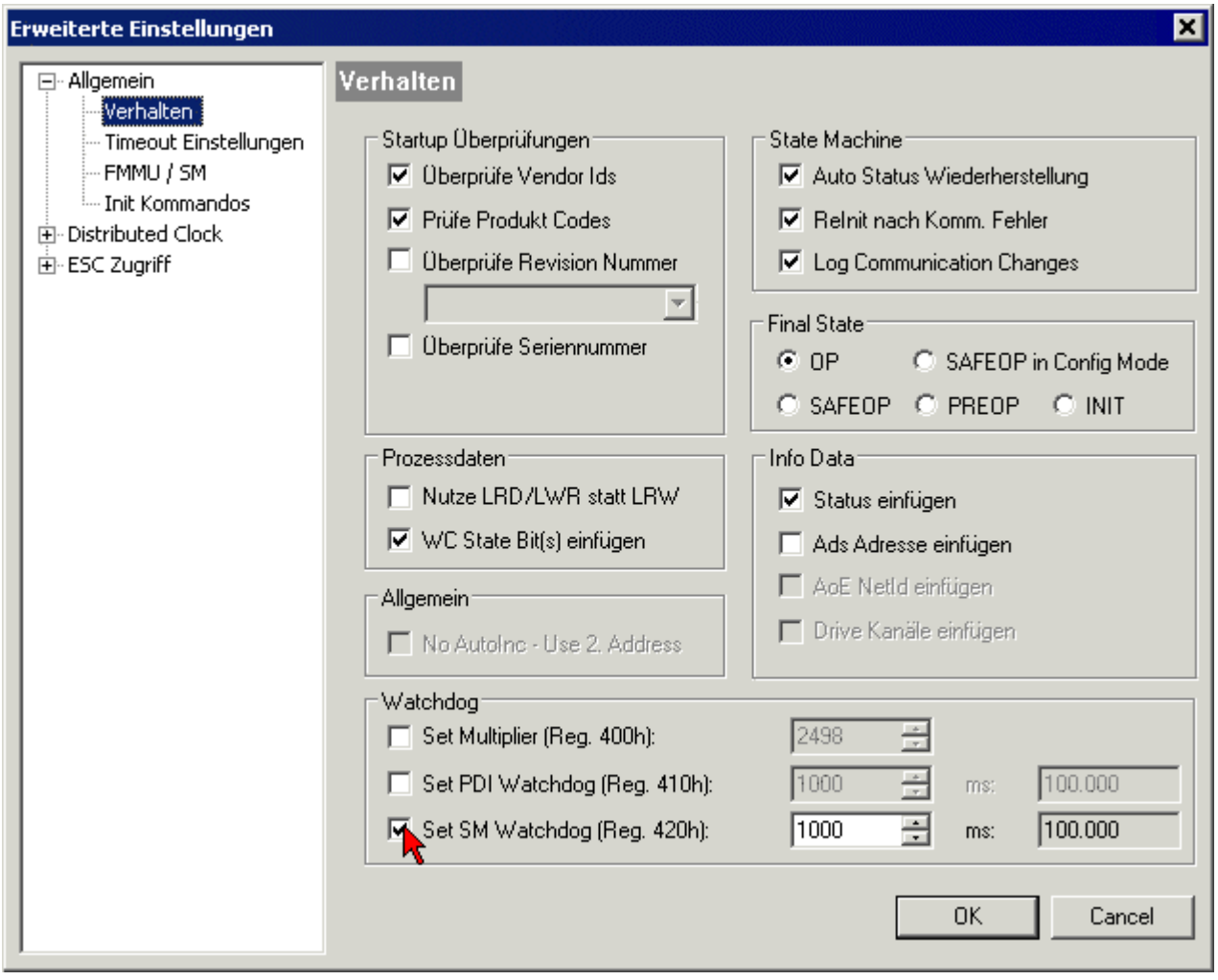


Abb. 15: Karteireiter EtherCAT -> Erweiterte Einstellungen -> Verhalten --> Watchdog

Anmerkungen:

- der Multiplier Register 400h (hexadezimal, also x0400) 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 x0400/0410/0420 eingesehen werden: ESC Access -> Memory

SM-Watchdog (SyncManager-Watchdog)

Der SyncManager-Watchdog wird bei jeder erfolgreichen EtherCAT-Prozessdaten-Kommunikation mit der Klemme zurückgesetzt. Findet z. B. durch eine Leitungsunterbrechung länger als die eingestellte und aktivierte SM-Watchdog-Zeit keine EtherCAT-Prozessdaten-Kommunikation mit der Klemme statt, löst der Watchdog aus. 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 mit dem ESC von der EtherCAT-Seite aus betrachtet.

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 Reg. 400/420 parametriert, aber vom 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-Kommunikation mit dem EtherCAT Slave Controller (ESC) statt, löst dieser Watchdog aus.
PDI (Process Data Interface) ist die interne Schnittstelle des ESC, z. B. zu lokalen Prozessoren im EtherCAT Slave. Mit dem PDI-Watchdog kann diese Kommunikation auf Ausfall überwacht werden.

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

Berechnung

Watchdog-Zeit = [1/25 MHz * (Watchdog-Multiplier + 2)] * PDI/SM Watchdog

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

Der Wert in Multiplier + 2 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 Version -0016. In vorherigen Versionen wird vom Einsatz dieser Betriebsart abgeraten.

⚠ VORSICHT

Beschädigung von Geräten und ungewolltes Verhalten des Systems möglich!

Bei aktiviertem SM-Watchdog und eingetragenem 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!

4.3 EtherCAT State Machine

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

Es werden folgende Zustände unterschieden:

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

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

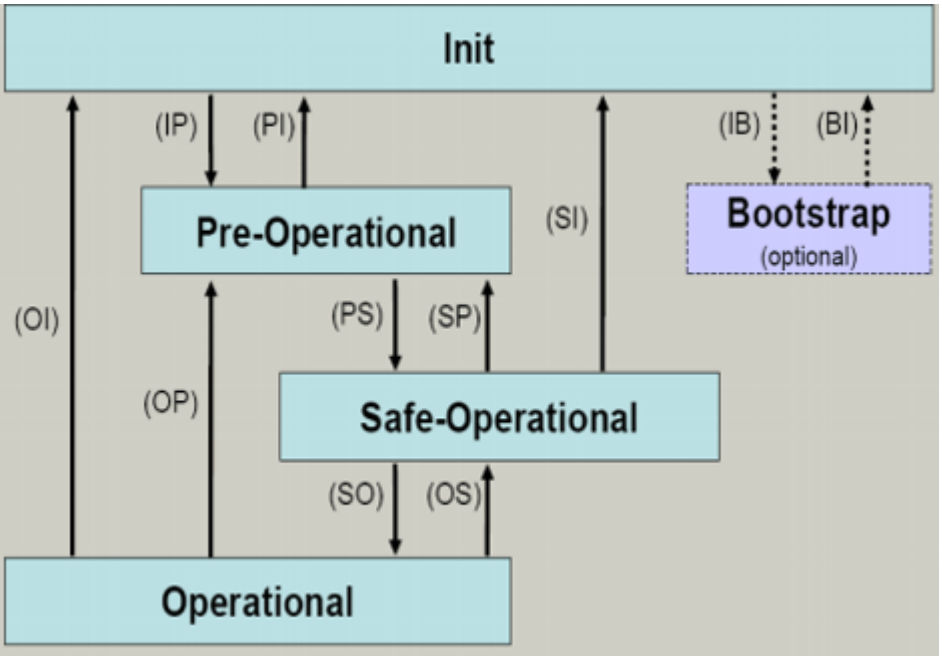


Abb. 16: Zustände der EtherCAT State Machine

Init

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

Pre-Operational (Pre-Op)

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

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

Safe-Operational (Safe-Op)

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

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

Ausgänge im SAFEOP

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

Operational (Op)

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

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

Boot

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

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

4.4 CoE-Interface

Allgemeine Beschreibung

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

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

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

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

- Index: 0x0000…0xFFFF ($0...65535_{dez}$)
- 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 Identitäts-Informationen zum Gerät hinterlegt wie Name, Hersteller, Seriennummer etc. Außerdem liegen hier Angaben über die aktuellen und verfügbaren Prozessdatenkonstellationen.
- 0x8000: hier sind die für den Betrieb erforderlichen funktionsrelevanten Parameter für alle Kanäle zugänglich wie Filtereinstellung oder Ausgabefrequenz.

Weitere wichtige Bereiche sind:

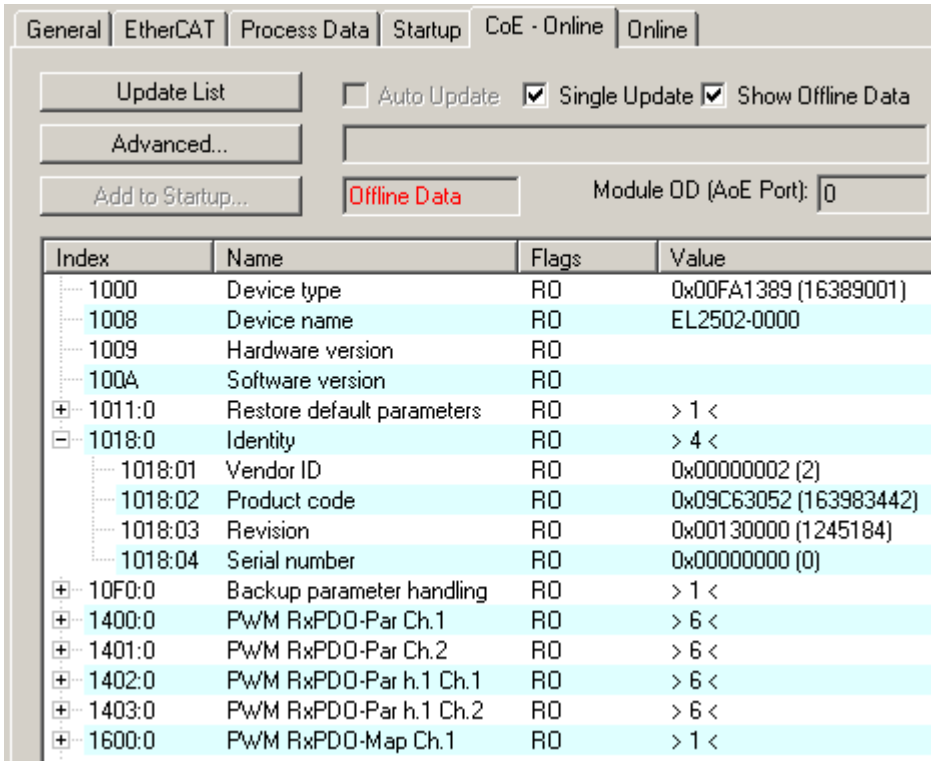
- 0x4000: hier befinden sich bei manchen EtherCAT-Geräten die Kanalparameter. Historisch war dies der erste Parameterbereich, bevor der 0x8000 Bereich eingeführt wurde. EtherCAT Geräte, die früher mit Parametern in 0x4000 ausgerüstet wurden und auf 0x8000 umgestellt wurden, unterstützen aus Kompatibilitätsgründen beide Bereiche und spiegeln intern.

- 0x6000: hier liegen die Eingangs-PDO („Eingang“ aus Sicht des EtherCAT-Masters)
- 0x7000: hier liegen die Ausgangs-PDO („Ausgang“ aus Sicht des EtherCAT-Masters)

Verfügbarkeit

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:



Index	Name	Flags	Value
1000	Device type	RO	0x00FA1389 (16389001)
1008	Device name	RO	EL2502-0000
1009	Hardware version	RO	
100A	Software version	RO	
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
1018:01	Vendor ID	RO	0x00000002 (2)
1018:02	Product code	RO	0x09C63052 (163983442)
1018:03	Revision	RO	0x00130000 (1245184)
1018:04	Serial number	RO	0x00000000 (0)
10F0:0	Backup parameter handling	RO	> 1 <
1400:0	PWM RxPDO-Par Ch.1	RO	> 6 <
1401:0	PWM RxPDO-Par Ch.2	RO	> 6 <
1402:0	PWM RxPDO-Par h.1 Ch.1	RO	> 6 <
1403:0	PWM RxPDO-Par h.1 Ch.2	RO	> 6 <
1600:0	PWM RxPDO-Map Ch.1	RO	> 1 <

Abb. 17: Karteireiter „CoE-Online“

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

HINWEIS

Veränderungen im CoE-Verzeichnis (CAN over EtherCAT), 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. Dies kann schreibend/lesend geschehen

- über den System Manager (Abb. Karteireiter „CoE-Online“) durch Anklicken
  Dies bietet sich bei der Inbetriebnahme der Anlage/Slaves an. Klicken Sie auf die entsprechende Zeile des zu parametrierenden Indizes und geben sie einen entsprechenden Wert im „SetValue“-Dialog ein.

- aus der Steuerung/PLC über ADS z. B. durch die Bausteine aus der TcEtherCAT.lib Bibliothek
  Dies wird für Änderungen während der Anlangenlaufzeit empfohlen oder wenn kein System Manager bzw. Bedienpersonal zur Verfügung steht.

Datenerhaltung

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

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

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

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

Startup List

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

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

Empfohlenes Vorgehen bei manueller Veränderung von CoE-Parametern

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

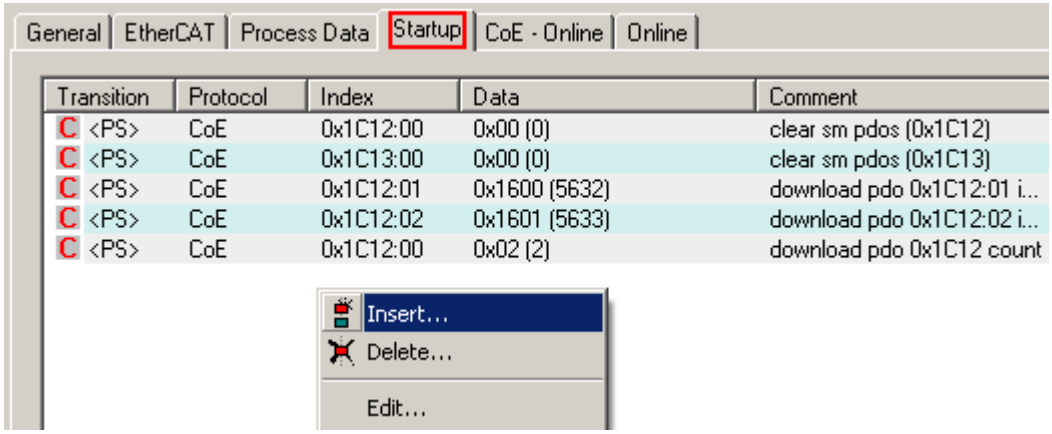


Abb. 18: StartUp-Liste im TwinCAT System Manager

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

Online/Offline Verzeichnis

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

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

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

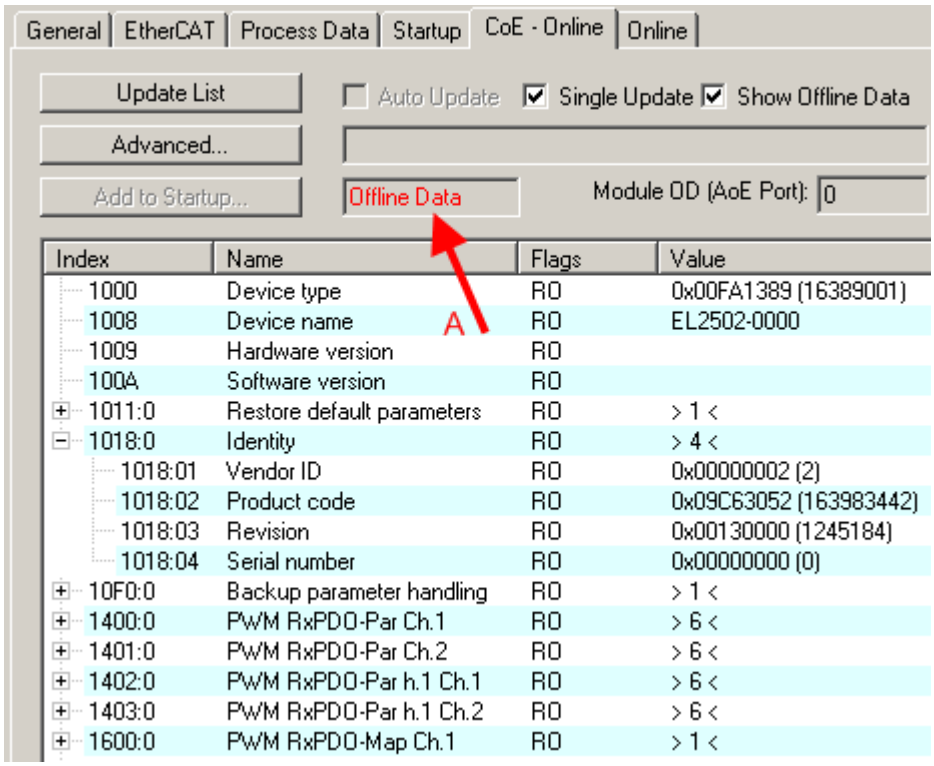


Abb. 19: Offline-Verzeichnis

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

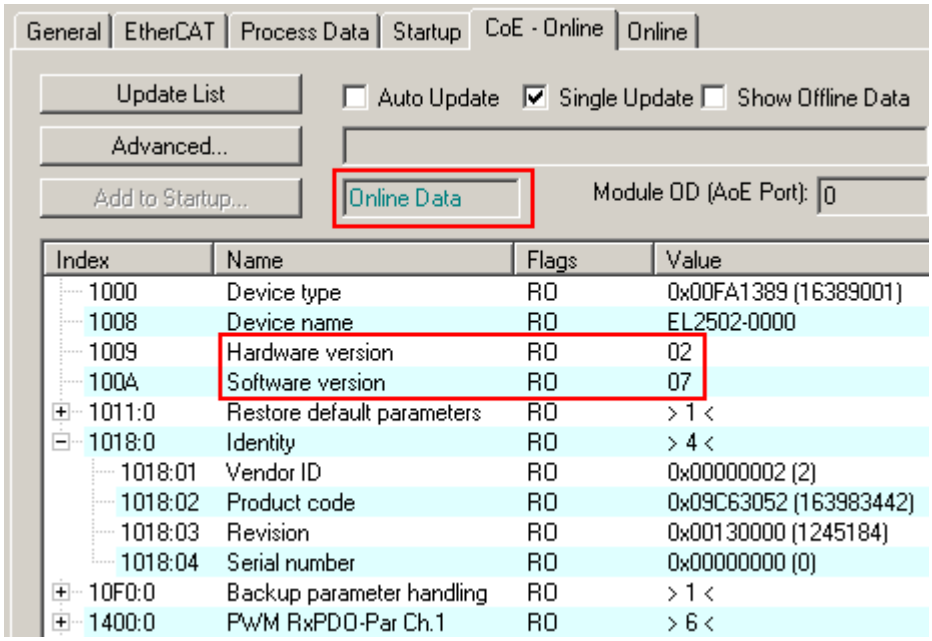


Abb. 20: Online-Verzeichnis

Kanalweise Ordnung

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

Im CoE-System sind für die Menge aller Parameter eines Kanals eigentlich immer 16 Indizes mit jeweils 255 Subindizes ausreichend. Deshalb ist die kanalweise Ordnung in $16_{dez}/10_{hex}$-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.

5 Parametrierung und Inbetriebnahme

5.1 EL6731 - PROFIBUS Masterklemme

5.1.1 PROFIBUS Protokolle

Als Master werden die Protokolle PROFIBUS DP, PROFIBUS DPV1, PROFIBUS DPV2, S5-FDL-AGAG-Kommunikation (nur FC310x) und das PROFIDRIVE-PKW-Interface unterstützt.

PROFIBUS DP

Es folgt eine Übersicht der PROFIBUS-DP-Master-Funktionen:

Funktion	Beschreibung
Standard DP	In dem Kapitel PROFIBUS DP [▶ 46] werden die notwendigen Schritte beschrieben, die zum Aufbau einer DP-Verbindung (Set_Prm - Parameter, Chk_Cfg - Konfiguration) und zum Austausch von Nutzdaten (Data_Exchange) notwendig sind.
Task-Synchronisierung	Im dem Kapitel Synchronisierung [▶ 49] wird beschrieben, wie die TwinCAT-Task mit dem PROFIBUS-Zyklus synchronisiert ist.
Slave-Prioritäten (nur FC310x, FC3151, CX1500-M310, EL6731)	Die Slaves können Telegramme mit unterschiedlichen Zykluszeiten bekommen, in dem Kapitel Slave-Priorisierung/ mehrere DP-Zyklen [▶ 137] sind die notwenigen Einstellungen beschrieben.
mehrere DP-Zyklen (nur FC310x, FC3151, CX1500-M310, EL6731)	Um bei großen Task-Zyklen möglichst neue Inputs zu erhalten, können mehrere DP-Zyklen je Task-Zyklus durchgeführt werden, wie es in dem Kapitel Slave-Priorisierung/mehrere DP-Zyklen [▶ 137] beschrieben ist.
Diagnose	In diesem Kapitel sind die Diagnose [▶ 126]-Möglichkeiten beschrieben.
Fehlerreaktionen	Im Falle eines Fehlers (Slave fällt aus oder Task wird gestoppt) können unterschiedliche Fehlerreaktionen [▶ 123] eingestellt werden.
Sync/Freeze	In dem Kapitel Sync/Freeze [▶ 50] wird Aktivierung des Sync- bzw. Freeze-Kommandos beschrieben.
Upload Configuration	Die am PROFIBUS angeschlossen Slaves können per Upload Configuration [▶ 57] ausgelesen werden.
Master-Redundanz (nur FC310x)	In dem Kapitel Master-Redundanz sind die Einstellungen beschrieben, um einen zweiten Master mit der gleichen Konfiguration als StandBy-Master zu haben (ab TwinCAT 2.9).

PROFIBUS DPV1

Es folgt eine Übersicht der PROFIBUS-DPV1-Master-Funktionen:

Funktion	Beschreibung
MSAC_C1	Die MSAC_C1 [▶ 54]-Verbindung wird zusammen mit der zyklischen Verbindung aufgebaut, es werden die Dienste Read, Write und Data_Transport unterstützt.
MSAC_C2	Die MSAC_C1 [▶ 54]-Verbindung wird unabhängig von der zyklischen Verbindung aufgebaut und kann auch von einem zweiten Master benutzt werden (während der erste über die zyklische bzw. die MSAC_C1-Verbindung mit dem Slave kommuniziert). Es werden die Dienste Initiate, Abort, Read, Write und Data_Transport unterstützt.

PROFIBUS DPV2

Es folgt eine Übersicht der PROFIBUS-DPV2-Master-Funktionen:

Funktion	Beschreibung
Äquidistanz	Die DPV2-Äquidistanz-Funktionalität ist in dem Kapitel PROFIBUS MC [▶ 48] beschrieben.

S5-FDL-AGAG-Kommunikation (nur FC310x)

Die S5-FDL-AGAG-Kommunikation ist in Dokumentation zu FC310x in dem Kapitel S5-FDL beschrieben.

PROFIDRIVE-PKW-Interface

Das PROFIDRIVE-PKW-Protokoll [▶ 57] ist im PROFIBUS-Master implementiert und kann über azyklische ADS-Aufrufe genutzt werden.

5.1.2 PROFIBUS DP

Standard DP-Betrieb

Um den Standard-DP-Betrieb zu konfigurieren, ist im TwinCAT System Manager wie folgt vorzugehen:

DP-Master projektieren

FC310x

Zuerst ist ein E/A-Gerät "Profibus Master FC310x, PCI" zu projektieren (mit rechter Maustaste auf "E/A-Geräte", dann "Gerät anfügen" auswählen). Der entsprechende Kanal ist auf dem Karteireiter "FC310x" und zu suchen ("Search"-Button) und ggf. die Baudrate anzupassen (ist standardmäßig auf 12 MBaud eingestellt).

EL6731

Vorgehensweise wie oben; "Profibus Master EL6731, EtherCAT" projektieren (mit rechter Maustaste auf "E/A-Geräte", dann "Gerät anfügen" auswählen).

DP-Slaves anfügen

Die Beckhoff-Slaves oder Fremd-Geräte sind zu projektieren (es werden alle Slaves automatisch angezeigt (nach Herstellern sortiert), deren GSD-Datei im Unterverzeichnis Profibus des System Managers gespeichert sind, um andere GSD-Dateien einzubinden, ist unter Verschiedenes die "Allgemeine Profibus Box (GSD)" anzuwählen).

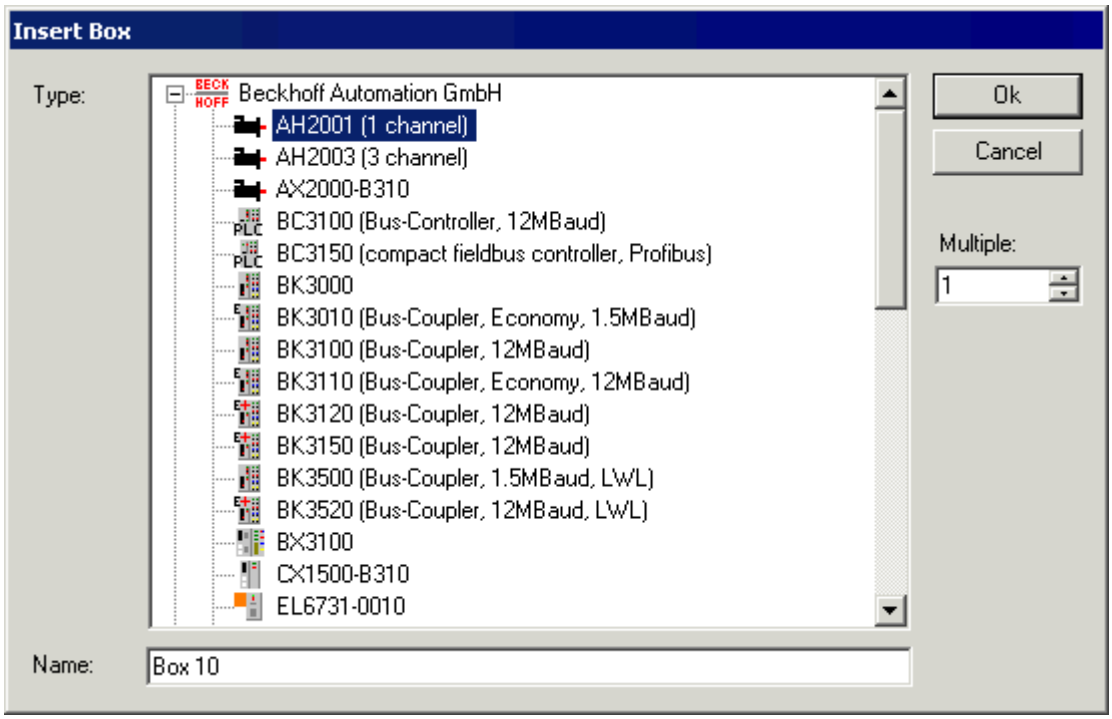


Abb. 21: Anfügen eines DP-Slaves

Bei modularen Slaves sind anschließend noch die Klemmen/IL-Module (Beckhoff-Slaves) bzw. die DP-Module (Fremd-Geräte) anzufügen.

System-Start

TwinCAT Config-Mode

Für den TwinCAT Config-Mode reicht das schon aus, um Daten mit den projektierten Slaves auszutauschen. Der TwinCAT Config-Mode ist dazu zu starten und mit dem "Reload Devices"-Button in der Toolbar die Konfiguration des DP-Masters zu aktivieren. Danach können die Daten der projektierten Slaves aus dem System-Manager auf dem jeweils zugehörigen Variablen-Karteireiter gelesen und beschrieben werden.

TwinCAT Run-Mode

Für den TwinCAT Run-Mode muss noch mindestens eine Variable des PROFIBUS-Masters oder der projektierten Slaves mit einer Task verknüpft werden. Dann ist das Projekt in der Registry zu speichern und das TwinCAT System im Run-Mode zu starten. Der Datenaustausch wird mit den Slaves erst durchgeführt, wenn die zugehörige Task gestartet wurde (wenn mehrere Tasks mit dem PROFIBUS-Master (oder den projektierten Slaves) verknüpft sind, muss die höchstpriore der verknüpften Tasks gestartet sein, damit Datenaustausch mit den Slaves durchgeführt wird).

Busparameter

Die PROFIBUS DP-Busparameter befinden sich im Dialog Bus-Parameter [▶ 95] der über den Karteireiter **"FC310x"** bzw. **"EL6731"** (Button Bus-Parameter (DP)) angewählt wird und sollten nur von sachkundigen Benutzern geändert werden.

5.1.3 PROFIBUS MC

PROFIBUS MC unterscheidet sich von PROFIBUS DP dadurch, dass der PROFIBUS-Zyklus mit einem konstanten Zyklus mit wenigen Mikrosekunden Jitter erfolgt (bei PROFIBUS DP ist der Jitter größer 100 µs) und am Anfang des Zyklus ein Broadcast-Global-Control-Telegramm gesendet wird, auf das sich die MC-Slaves synchronisieren können. Dadurch ist es möglich, Antriebsregelkreise genau mit der NC zu synchronisieren.

Die genaue Synchronisierung hat aber zur Folge, dass Busstörungen, Ausschalten von Slaves, Abziehen von Bussteckern, etc. in der Regel dazu führen, dass die Synchronität zwischen Master und Slave verloren geht, da sich das Bus-Timing dadurch verändert.

DP/MC-Equidistant-Mode

Um die EL6731 mit PROFIBUS MC zu betreiben, muss auf dem Karteireiter **"EL6731"** (für TwinCAT) des Masters der **Operation Mode** "DP/MC(Equidistant)" eingestellt werden. Die Task, die die Äquidistanz-Funktionalität der EL6731 nutzt (in der Regel die NC-Task), sollte die höchste Priorität haben, ansonsten kann es zu Störungen in der Synchronität kommen.

Die EL6731 ist grundsätzlich im Sync-Master Mode; EtherCAT synchronisiert über die Distributed clocks, dabei wird der DC Optimized Mode verwendet. Die Shift Zeit in den Distributed Clocks-Einstellungen des EtherCAT-Masters muss mindestens so groß wie die CalcAndCopy-Time der EL6731 sein. Die CalcAndCopy-Time der EL6731 hängt wiederum von der Anzahl der konfigurierten DP-Slaves ab, kann aber im State OPERATIONAL gemessen werden (Entry 0x1C32:08 auf 1 setzen, dann Entry 0x1C32:06 auslesen). Das folgende Bild zeigt den Zyklus für eine Task mit IO am Taskanfang:

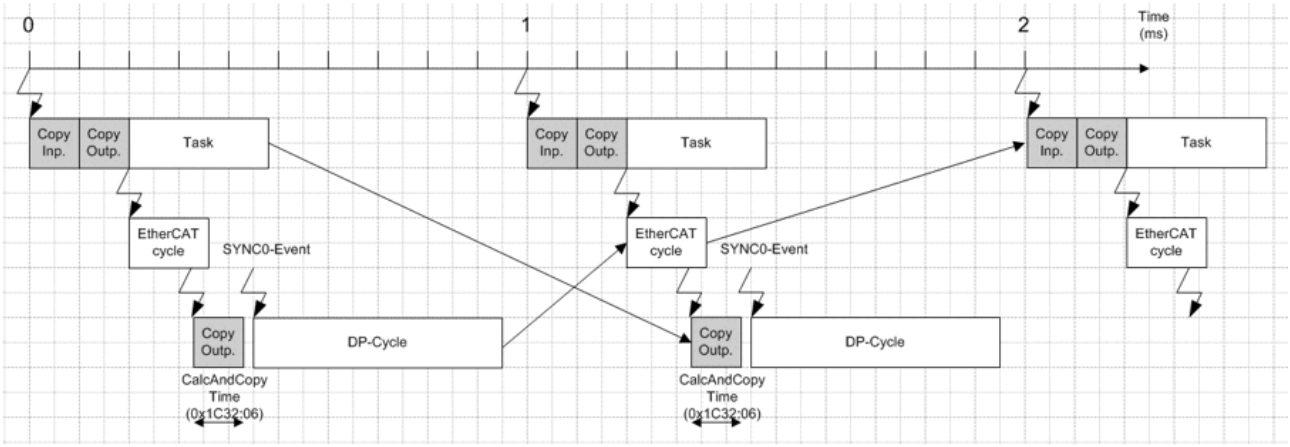


Abb. 22: Task-Zyklus EL6731

Einstellung der Equidistant-Zeiten

Mit dem Button **Calculate MC-Times** (TwinCAT; s. Karteireiter MC [▶ 98]) können alle Equidistant-Parameter automatisch eingestellt werden.

Diagnose der Equidistant-Zeiten

Die Diagnose der Equidistanz-Zeiten kann mit dem Karteireiter MC-Diag [▶ 102] im System Manager bzw. per ADS im Steuerungsprogramm erfolgen (s. Kapitel Master-Diagnose [▶ 126]).

5.1.4 Synchronisierung

5.1.4.1 Übersicht

Im TwinCAT RunMode ist die EL6731 immer mit der höchstprioren Task synchronisiert, mit der Variablen verknüpft sind. Für jede EL6731 wird eigenes EtherCAT Telegramm definiert. Die Zykluszeit der entsprechenden Task wird in der Cycle-Time auf dem Karteireiter "EL6731" des Masters angezeigt, sobald einmal das Mapping erzeugt wurde. Bei der Task kann eingestellt werden, ob das "I/O am Taskanfang" aktualisiert werden soll oder nicht.

I/O am Taskanfang

Wenn bei der Task die Einstellung "I/O am Taskanfang" (Check-Box) angewählt ist (Defaulteinstellung bei NC-Task), erfolgt die Übertragung des EtherCAT Telegramms vor dem Start der Task.

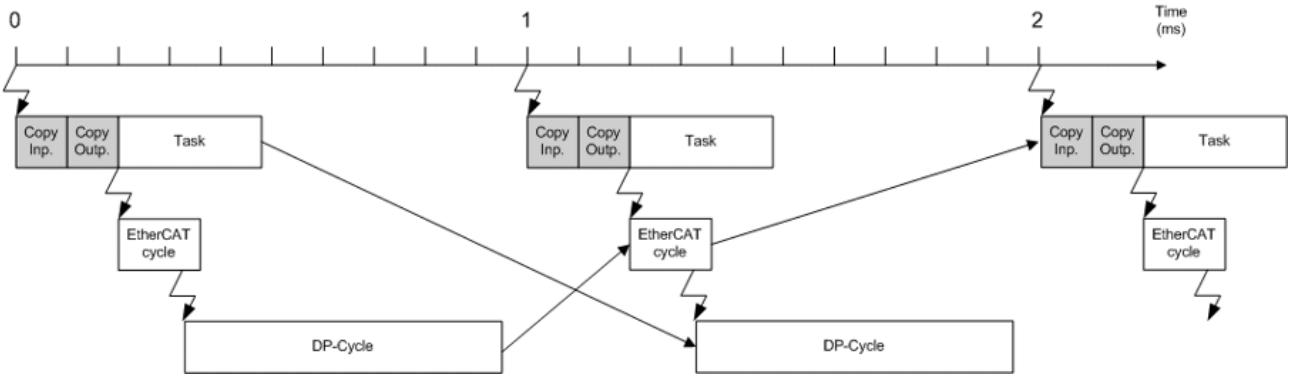


Abb. 23: Zyklus mit IO am Taskanfang

I/O nicht am Taskanfang

Wenn bei der Task die Einstellung "I/O am Taskanfang" (Check-Box) nicht angewählt ist (Defaulteinstellung bei PLC-Task), erfolgt die Übertragung des EtherCAT Telegramms nach dem Ablauf der Task. Die Ausgänge sind daher im Vergleich zu "IO am Taskanfang" um einen Zyklus neuer, das EtherCAT Telegramm jittert aber mit der Laufzeit der Task.

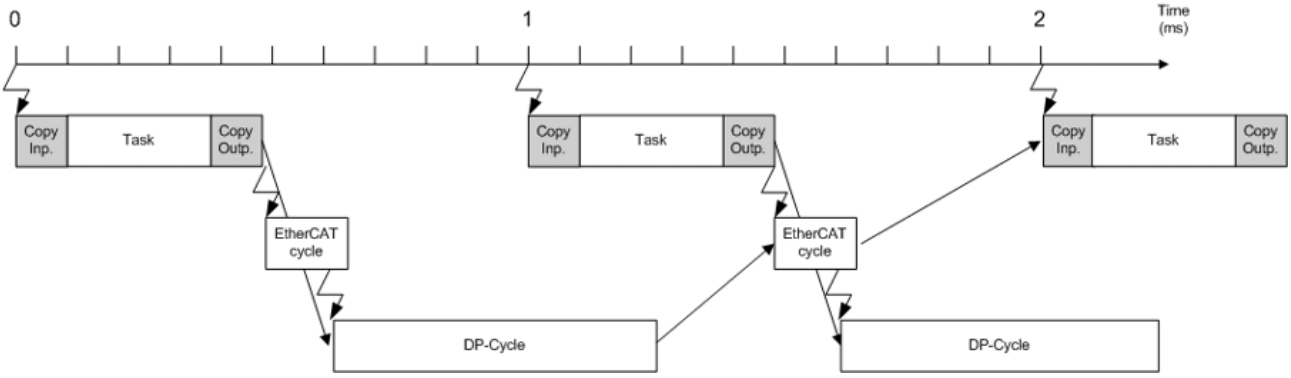


Abb. 24: Zyklus mit IO nicht am Taskanfang

5.1.4.2 Sync/Freeze-Funktionalität

Mit Sync will man bei mehreren Slaves gleichzeitig Ausgänge ausgeben, mit Freeze will man bei mehreren Slaves gleichzeitig Eingänge einlesen.

Der Ablauf in TwinCAT mit FC310x / EL6731und Buskopplern (im K-Bus synchronen Mode) wäre also der folgende (s. Kapitel Synchronisierung [▶ 49]):

- Am Anfang (I/O am Taskanfang) bzw. Ende (I/O nicht am Taskanfang) des Task-Zyklus werden die Ausgänge geschrieben
- Dadurch wird der PROFIBUS-Zyklus gestartet
- Am Anfang des PROFIBUS-Zyklus wird ein Sync/Freeze-Telegramm gesendet
- Dadurch starten die Buskoppler einen K-Bus-Zyklus mit den Ausgängen aus dem letzten Task-Zyklus und übergeben die Eingänge aus dem letzten K-Bus-Zyklus
- Dann sendet der Master zu jedem Slave die aktuellen Ausgänge und holt die übergebenen Eingänge
- Am Anfang des nächsten Task-Zyklus werden die Eingänge gelesen
- usw.

Ausgänge und Eingänge sind also immer einen Zyklus alt.

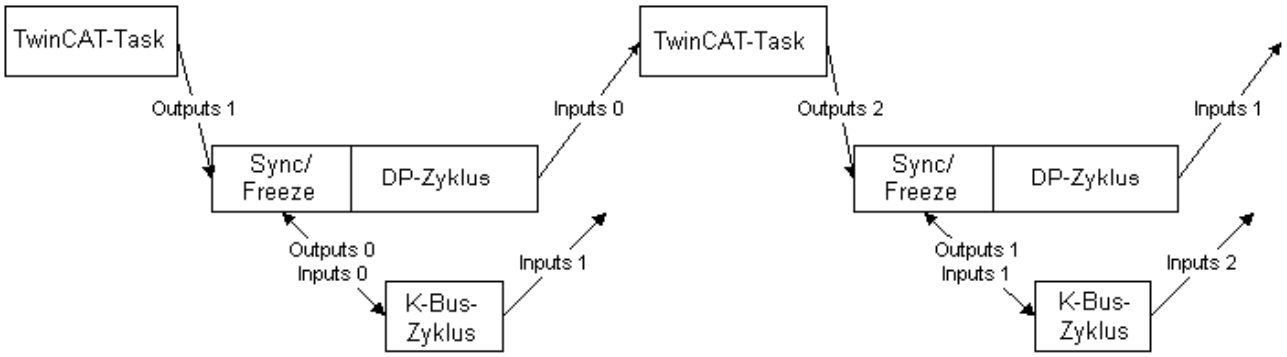


Abb. 25: Zeitliche Abfolge TwinCAT-Task, DP-Zyklus und K-Bus-Zyklus

An der FC310x/EL6731 ist der **Operation-Mode** auf dem Karteireiter **"FC310x"** bzw. **"EL6731"** (für TwinCAT) des Masters auf "DP/MC (Equidistant)" zu stellen, an den Boxen, die per Sync/Freeze betrieben werden sollen, ist auf dem Karteireiter Profibus [▶ 104] des Slaves die Checkbox **Sync/Freeze enable** anzuklicken. Der Master verwendet für die Synchronisierung per Sync/Freeze immer die Gruppe 1.

5.1.5 ADS (azyklische Dienste)

5.1.5.1 ADS-Interface

Sämtliche azyklischen Daten werden mit ADS-Read, ADS-Write oder ADS-Write-Control an die bzw. von der FC310x/EL6731 übertragen. Die FC310x/EL6731 hat eine eigene Net-ID und unterstützt die folgenden Ports:

Port	Beschreibung
200	damit wird die FC310x/EL6731 selbst adressiert, d.h. Daten, die lokal auf der FC310x/EL6731 liegen, bei denen in der Regel kein zusätzlicher Buszugriff nötig ist
0x1000 - 0x107E	damit wird ein angeschlossener PROFIBUS-Teilnehmer adressiert, wobei sich Adresse aus Port-0x1000 berechnet, es wird auch immer ein Buszugriff durchgeführt

ADS-Read

Es folgt eine Übersicht der von der FC310x/EL6731 unterstützten IndexGroup/IndexOffset bei ADS-Read.

IndexGroup bei lokaler Adressierung der FC310x/EL6731 (Port 200)

Index-Group (Lo-Word)	IndexGroup (Hi-Word)	IndexOff-set	Beschreibung FC310x	EL6731
0xF100	0x00	BYTE-Offset innerhalb der Daten	Damit werden die Diagnosedaten der FC310x/EL6731 ausgelesen. Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, enthalten die Daten die im Kapitel Master-Diagnose [▶ 126] beschriebenen Diagnosedaten der FC310x/EL6731. Die FC310x/EL6731 setzt dabei ihr DiagFlag zurück, das dann wieder gesetzt wird, wenn sich die Diagnosedaten der FC310x/EL6731 erneut ändern.	
0xF181	0x00-0x7E	BYTE-Offset innerhalb der Daten	Damit werden die Diagnosedaten eines projektierten DP-Slaves ausgelesen, die Stationsadresse berechnet sich aus dem IndexGroup(Hi-Word). Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, enthalten die Daten die im Kapitel Slave-Diagnose [▶ 127] beschriebenen Diagnosedaten eines projektierten DP-Slaves.	
0xF830	0x8000-0x807E	immer 0	Damit kann festgestellt werden, welche DP-Slaves am PROFIBUS vorhanden sind, unabhängig davon, ob sie projektiert wurden oder nicht, die Stationsadresse berechnet sich aus IndexGroup(Hi-Word)-0x8000. Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, hat der entsprechende DP-Slave korrekt geantwortet, die Daten enthalten die Ident-Nummer des Slaves (BYTE-Offset 0-1) und die ausgelesenen CfgData (ab BYTE-Offset 2) (s. Kapitel Upload Configuration [▶ 57]).	
0xF840	0	0	Damit werden Firmware-Version und Stationsadresse der FC310x/EL6731 ausgelesen. Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, enthalten die Daten die Firmware-Version (BYTE-Offset 0-1) und die Stationsadresse der FC310x/EL6731 (BYTE-Offset 2).	Damit wird Stationsadresse der EL6731 ausgelesen. Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, enthalten die Daten eine Konstante (BYTE-Offset 0-1) und die Stationsadresse der EL6731 (BYTE-Offset 2). Data[0] 0x45 Data[1] 0x23 Data[2] Adresse Data[3] 0x00

PROFIBUS-Stationsadresse über ADSWRITE verändern (nur EL6731 und EL6731-0010)

Wird die Stationsadresse einmal über diesen Befehl gesetzt, wird sie im Flash der Klemme abgelegt. Eine Stationsadresse über die Init-Commands (Objekt 0x8000) wird dann ignoriert. Erst mit dem Wiederherstellen der Default-Parameter (Objekt 0x1011), wird die Adresse in 0x8000:01 (beim PROFIBUS Master 0xF800:01) wieder akzeptiert.

HINWEIS: Die EtherCAT-Klemme muss nach dem Schreiben der Stationsadresse einmal in INIT und zurück in OP gefahren werden, damit sie die neue Adresse übernimmt.
NetId der EL6731 oder EL6731-0010
Port: 200

Index-Group (Lo-Word)	Index-Group (Hi-Word)	Inde-xOffset	Länge (in Byte)	Beschreibung
0xF480	0x00	0 -2	4	Data[0] 0x45 Data[1] 0x23 Data[2] PROFIBUS Adresse < 127 Data[3] 0x00

IndexGroup bei Adressierung eines projektierten PROFIBUS-Teilnehmers (Port 0x1000-0x107E)

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Beschreibung
0x00-0xFF	0x00	0x00-0xFF	Damit wird ein DPV1-Read an den entsprechenden, projektierten DPV1-Slave über eine Class 1-Verbindung gesendet, die DPV1-Slot-Nummer entspricht der IndexGroup, der DPV1-Index entspricht dem IndexOffset. Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, enthalten die Daten die ausgelesenen DPV1-Daten (s. Kapitel DPV1 [▶ 54])
0x100-0x1FF	0x00	0x00-0xFF	Damit wird ein DPV1-Read an den entsprechenden, projektierten DPV1-Slave über eine Class 2-Verbindung gesendet, die DPV1-Slot-Nummer entspricht der IndexGroup - 0x100, der DPV1-Index entspricht dem IndexOffset. Wenn der ADS-Read ohne Fehler (Error-Code = 0) beantwortet wird, enthalten die Daten die ausgelesenen DPV1-Daten (s. Kapitel DPV1 [▶ 54])
0x0000 - 0xFFF	0x10000000 - 0xF0000000	0x00-0xFF	Damit wird ein PKW-Read an den entsprechenden, projektierten PROFIDRIVE-Slave gesendet, die Parameternummer (PNU) steht im Lo-Word der IndexGroup, der Sub-Index bei Zugriff auf ein Array steht im IndexOffset, die angesprochene Achse steht in den Bits 28-31 der IndexGroup (bei einem 1-Achser muss hier eine 1 stehen), in Bit 26,27 kann noch die PKW-Kompatibilität angepasst werden (leider sind nicht alle PROFIDRIVE-Slaves hier kompatibel, s. Kapitel PKW-Interface [▶ 57]).
0	0x01000000	0	Damit wird FDL-Read für die Siemens AG-Kopplung an die entsprechende projektierte FDL-Station gesendet (nur FC310x, s. Dokumentation zu FC310x Kapitel S5-FDL).

ADS-Write

Es folgt eine Übersicht der von der FC310x/EL6731 unterstützten IndexGroup/IndexOffset bei ADS-Write.

IndexGroup bei lokaler Adressierung der FC310x/EL6731 (Port 200)

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Beschreibung
0xF100	0x00	0 -2	Damit werden die Equidistant-Diagnosedaten (IndexOffset = 0), die Repeat-Counter (IndexOffset = 1) oder die NoAnswer-Counter (IndexOffset = 2) der FC310x/EL6731 zurückgesetzt.

IndexGroup bei Adressierung eines projektierten PROFIBUS-Teilnehmers (Port 0x1000-0x107E)

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Beschreibung
0x00-0xFF	0x00	0x00-0xFF	Damit wird ein DPV1-Write an den entsprechenden, projektierten DPV1-Slave über eine Class 1-Verbindung gesendet, die DPV1-Slot-Nummer entspricht der IndexGroup, der DPV1-Index entspricht dem IndexOffset (-> Kapitel DPV1 [▶ 54]).
0x100-0x1FF	0x00	0x00-0xFF	Damit wird ein DPV1-Write an den entsprechenden, projektierten DPV1-Slave über eine Class 2-Verbindung gesendet, die DPV1-Slot-Nummer entspricht der IndexGroup - 0x100, der DPV1-Index entspricht dem IndexOffset (-> Kapitel DPV1 [▶ 54]).
0x400	0x00	0x00	Damit wird ein DPV1-Abort an den entsprechenden, projektierten DPV1-Slave über eine Class 2-Verbindung gesendet, die Abort-Parameter befinden sich in den Daten (-> Kapitel DPV1 [▶ 54]).
0x0000 - 0xFFF	0x10000000 - 0xF0000000	0x00-0xFF	Damit wird ein PKW-Write an den entsprechenden, projektierten PROFIDRIVE-Slave gesendet, die Parameternummer (PNU) steht im Lo-Word der IndexGroup, der Sub-Index bei Zugriff auf ein Array steht im IndexOffset, die angesprochene Achse steht in den Bits 28-31 der IndexGroup (bei einem 1-Achser muss hier eine 1 stehen), in Bit 26,27 kann noch die PKW-Kompatibilität angepasst werden (leider sind nicht alle PROFIDRIVE-Slaves hier kompatibel, -> Kapitel PKW-Interface [▶ 57]).
0	0x01000000	0	Damit wird FDL-Write für die Siemens AG-Kopplung an die entsprechende projektierte FDL-Station gesendet (nur FC310x, s. Dokumentation zu FC310x Kapitel S5-FDL).
0	0x02000000	0	Damit wird ein SetSlaveAddress-Kommando an einen projektierten DP-Slave gesendet, wobei der DP-Slave mit der neuen Stationsadresse projektiert werden muss, die alte Stationsadresse ist dann am BYTE-Offset 0 der ADS-Write-Daten einzutragen. Weiterhin muss unter BYTE-Offset 1 und 2 die Ident-Nummer des Slaves stehen und unter BYTE-Offset 3, ob die Slave später noch einmal geändert werden darf (0) oder nicht (ungleich 0). Insgesamt sind also 4 Bytes ADS-Write-Daten zu übergeben.

ADS-ReadWrite

Es folgt eine Übersicht der von der FC310x/EL6731 unterstützten IndexGroup/IndexOffset bei ADS-ReadWrite.

IndexGroup bei Adressierung eines projektierten PROFIBUS-Teilnehmers (Port 0x1000-0x107E)

IndexGroup (Lo-Word)	IndexGroup (Hi-Word)	IndexOffset	Beschreibung
0x100-0x1FF	0x00	0x00-0xFF	Damit wird ein DPV1-Data_Transport an den entsprechenden, projektierten DPV1-Slave über eine Class 2-Verbindung gesendet, die DPV1-Slot-Nummer entspricht der IndexGroup - 0x100, der DPV1-Index entspricht dem IndexOffset (-> KapitelDPV1 [▶ 54]).
0x200	0x00	0x00	Damit wird ein DPV1-Initiate an den entsprechenden, projektierten DPV1-Slave über eine Class 2-Verbindung gesendet, die Initiate-Parameter befinden sich in den Daten (-> Kapitel DPV1 [▶ 54]).

ADS-WriteControl

Es folgt eine Übersicht der von der FC310x/EL6731 unterstützten ADS-Write-Control-Kommandos.

ADS-WriteControl bei lokaler Adressierung der FC310x/EL6731 (Port 200)

AdsState	DeviceState	State der FC310x/ EL6731	Beschreibung
STOP (6)	0x00	RUN (5)	Damit wird die FC310x/EL6731 gestoppt, d.h. die Prozessdatenverbindungen zu allen DP-Slaves (Data_Exchange) werden abgebaut (mit SetPrm,Unlock).
RUN (5)	0x00	STOP (6)	Damit wird die FC310x/EL6731 nach einem Stoppen erneut gestartet, d.h. die Prozessdatenverbindungen zu allen DP-Slaves (Data_Exchange) werden wieder aufgebaut (normaler DP-Hochlauf).

ADS-WriteControl bei Adressierung eines projektierten PROFIBUS-Teilnehmer (Port 0x1000-0x107E)

AdsState	DeviceState	State der FC310x/ EL6731	Beschreibung
STOP (6)	0x00	RUN (5)	Damit wird der Slave gestoppt, d.h. die Prozessdatenverbindung zu dem entsprechenden DP-Slave (Data_Exchange) wird abgebaut (mit SetPrm, Unlock).
RUN (5)	0x00	STOP (6)	Damit wird der Slave nach einem Stoppen erneut gestartet, d.h. die Prozessdatenverbindung zu dem entsprechenden DP-Slave (Data_Exchange) wird wieder aufgebaut (normaler DP-Hochlauf).

ADS-Fehlercodes

Der 32-Bit-ADS-Fehlercode besteht immer aus einem allgemeinen ADS-Fehlercode (Lo-Word, siehe ADS-Dokumentation) und einem FC310x/EL6731-spezifischen, eindeutigen Fehlercode (Hi-Word, -> Kapitel ADS-Fehlercodes [▶ 135]). Die entsprechende Meldung wird auch im Logger des TwinCAT System Managers textuell angezeigt.

5.1.5.2 PROFIBUS DPV1

Der Master unterstützt auf einer C1-Verbindung die Dienste Read und Write sowie auf einer C2-Verbindung die Dienste Read, Write, Data_Transport, Initiate und Abort.

C1-Verbindung (MSAC-C1)

Die C1-Verbindung ist für den Master reserviert, die mit dem Slave zyklischen Datenaustausch durchführt (C1-Master). Um die C1-Verbindung bei einem Slave nutzen zu können, muss der Slave DPV1 unterstützen (in der GSD-Datei müssen dafür die Zeile "DPV1_Slave = 1" und das Schlüsselwort "C1_Max_Data_Len" mit einer entsprechenden Länge eingetragen sein). Weiterhin ist in der Regel die C1-Funktionalität durch Setzen des Bit 7 von **PrmData**-Byte 0 (s. Karteireiter Profibus [▶ 104] des Slaves) bei dem entsprechenden Slave zu aktivieren (bei Beckhoff-Geräten, die DPV1 unterstützen, passiert das automatisch).

MSAC-C1-Read ist auf ADS-Read, MSAC-C1-Write auf ADS-Write abgebildet:

MSAC-C1 Read

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	Slot-Number (DPV1-Parameter)
IndexOffset	Index (DPV1-Parameter)
Length	Länge der auszulesenden Daten
Data	bei Response: ausgelesene Daten

MSAC-C1 Write

ADS-Write-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	Slot-Number (DPV1-Parameter)
IndexOffset	Index (DPV1-Parameter)
Length	Länge der zu schreibenden Daten
Data	bei Request: zu schreibende Daten

C2-Verbindung (MSAC-C2)

Die C2-Verbindung ist in der Regel für einen zweiten Master (C2-Master) gedacht, der nicht zyklisch mit dem Slave kommuniziert, aber auch der C1-Master kann die C2-Verbindung nutzen. Um die C2-Verbindung bei einem Slave nutzen zu können, muss der Slave DPV1 unterstützen (in der GSD-Datei müssen dafür die Zeile "DPV1_Slave = 1" und das Schlüsselwort "C2_Max_Data_Len" mit einer entsprechenden Länge eingetragen sein).

Die Verbindung wird von dem Master automatisch aufgebaut, sobald ein Read, Write oder Data_Transport-Zugriff gefordert wird, sie kann aber auch explizit per Initiate aufgebaut werden. Beim automatischen Aufbau sendet der Master die zuletzt übergebenen Initiate-Parameter (s. Beschreibung Initiate), wobei er die Initiate-Parameter nach einem TwinCAT Start (Restart) mit 0 initialisiert (Ausnahme ist die Verbindung-Überwachung, die entsprechend des im System Manager eingestellten Wertes (**Watchdog** unter **DPV1-Class 2** auf dem Karteireiter Profibus [▶ 104] des Slaves) initialisiert wird).

Weiterhin muss die C2-Funktionalität durch Anwählen der **Enable**-Check-Box unter **DPV1-Class 2** (s. Karteireiter Profibus [▶ 104] des Slaves) bei jedem Slave, der mit C2-Diensten angesprochen werden soll, aktiviert werden.

Wenn ein anderer Master mit dem Slave zyklischen Datenaustausch macht, muss unter **DP-Class 2** die Einstellung "No cyclic connection" angewählt werden (s. Karteireiter Profibus [▶ 104] des Slaves). Das ist z. B. sinnvoll, um einen BC3100/IL23xx-C310 per PROFIBUS debuggen zu können, obwohl er an einer Fremd-Steuerung hängt.

MSAC-C2-Read ist auf ADS-Read, MSAC-C2-Write auf ADS-Write, MSAC-C2-Data_Transport auf ADS-ReadWrite, MSAC-C2-Initiate auf ADS-ReadWrite und MSAC-C2-Abort auf ADS-Write abgebildet:

MSAC-C2 Read

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	0x100 + Slot-Number (DPV1-Parameter)
IndexOffset	Index (DPV1-Parameter)
Length	Länge der auszulesenden Daten
Data	bei Response: ausgelesene Daten

MSAC-C2 Write

ADS-Write-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	0x100 + Slot-Number (DPV1-Parameter)
IndexOffset	Index (DPV1-Parameter)
Length	Länge der zu schreibenden Daten
Data	bei Request: zu schreibende Daten

MSAC-C2 Data_Transport

ADS-ReadWrite-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	0x100 + Slot-Number (DPV1-Parameter)
IndexOffset	Index (DPV1-Parameter)
Write-Length	Länge der zu schreibenden Daten
Read-Length	Länge der auszulesenden Daten
Data	bei Request: zu schreibende Daten, bei Response: ausgelesene Daten

MSAC-C2 Initiate

Mit dem MSAC-C2-Initiate-Dienst kann die C2-Verbindung zu dem Slave aufgebaut bzw. neue Initiate-Parameter übergeben werden, falls diese schon aufgebaut ist.

ADS-ReadWrite-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	0x200 + Slot-Number (DPV1-Parameter)
IndexOffset	0
Read-Length	Länge der Initiate-Response-Parameter (6)
Write-Length	Länge der Initiate-Request-Parameter (10 - 42)
Data	Initiate-Request-Parameter bzw. Initiate-Response-Parameter

Initiate-Request-Parameter

0x00 - 0x01	Feature_Supported
0x02 - 0x03	Profile_Feature_Supported
0x04 - 0x05	Profile_Ident_number
0x06	sType
0x07	sLen: Länge von sAddr (0 - 16)
0x08	dType
0x09	dLen: Länge von dAddr (0 - 16)
0x0A - 0x19	sAddr
0x1A - 0x29	dAddr

Initiate-Response-Parameter

0x00 - 0x01	Feature_Supported (von Slave empfangener Wert)
0x02 - 0x03	Profile_Feature_Supported (von Slave empfangener Wert)
0x04 - 0x05	Profile_Ident_number (von Slave empfangener Wert)

MSAC-C2 Abort

Mit dem MSAC-C2-Abort-Dienst kann die C2-Verbindung zu dem Slave wieder abgebaut werden.

ADS-Write-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	0x400 + Slot-Number (DPV1-Parameter)
IndexOffset	0
Length	Länge der Abort-Parameter (3)
Data	bei Request: Abort-Parameter

Abort-Parameter

0x00	Reason_Code
0x01 - 0x02	Additional_Detail

5.1.5.3 Upload der Konfiguration

Während des Betriebs kann der PROFIBUS per ADS [▶ 51]-Read nach neuen Geräten gescannt werden:

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	200
IndexGroup	0xzzyyF830 (yy = Stationsadresse, zz = 0: bei Beckhoff-Geräten werden die Tabellen 0,1 und 9 ausgelesen, zz = 0x80: Beckhoff-Geräte liefern die gleichen Informationen wie Fremd-Geräte)
IndexOffset	0
Length	1538
Data	Konfigurationsdaten des Slaves

Falls in der IndexGroup angegeben ist, dass bei Beckhoff-Geräten die Tabellen 0, 1 und 9 ausgelesen werden sollen, werden die folgenden Daten geliefert, sofern es sich um ein Beckhoff-Gerät handelt:

Offset	Beschreibung
0 -1	0
2 -513	Tabelle 0 (enthält u.a. die genauen Koppler-Typ und die Firmware-Version)
514 - 1025	Tabelle 9 (enthält die Koppler-Nummer und die Klemmen-Nummern)
1026 - 1537	Tabelle 1 (nur bei Buscontrollern relevant, enthält u.a. die Zuordnung der Klemmen)

Falls es sich um Fremdgeräte handelt oder in der IndexGroup angegeben ist, dass sich Beckhoff-Geräte genauso verhalten sollen wie Fremd-Geräte, werden die folgenden Informationen in der ADS-Read-Response zurückgeliefert:

Offset	Beschreibung
0 -1	1
2 - 7	DP-Diagnosedaten Byte 0-5 (s. Slave-Diagnose [▶ 127])
8 - 251	DP-Konfigurationsdaten (CfgData [▶ 130])

5.1.5.4 PKW-Interface von PROFIDRIVE-Slaves

Das PKW-Interface ist in der FC310x/EL6731 integriert, es kann dann aus dem Steuerungsprogramm per ADS [▶ 51] darauf zugegriffen werden. PKW-Read wird dabei auf ADS-Read, PKW-Write auf ADS-Write und PKW-Read No Of Array Elements auf ADS-Read abgebildet:

PKW Read

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	Bit 0-11: Parameter-Nummer (PNU)
	Bit 12-25: 0
	Bit 26: 1 = Subindex in Octet 3 (Standard), 0 = Subindex in Octet 4 (Simodrive 611U)
	Bit 27: 1 = ARRAY-Codes werden nicht unterstützt von PROFIDRIVE-Slave
	Bit 28-31: Nummer der Achse (bei 1-Achsen-Modulen immer 1)
IndexOffset	Subindex (bei ARRAY-Zugriffen)
Length	Parameterlänge: 2 oder 4
Data	bei Response: Parameterwert

PKW Write

ADS-Write-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	Bit 0-11: Parameter-Nummer (PNU)
	Bit 12-25: 0
	Bit 26: 1 = Subindex in Octet 3 (Standard), 0 = Subindex in Octet 4 (Simodrive 611U)
	Bit 27: 1 = ARRAY-Codes werden nicht unterstützt von PROFIDRIVE-Slave
	Bit 28-31: Nummer der Achse (bei 1-Achsen-Modulen immer 1)
IndexOffset	Subindex (bei ARRAY-Zugriffen)
Length	Parameterlänge: 2 oder 4
Data	bei Request: Parameterwert

PKW ReadNoOfArrayElements

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	0x1000 + Stationsadresse des Slaves
IndexGroup	Bit 0-11: Parameter-Nummer (PNU)
	Bit 12-15: 0
	Bit 16: 1
	Bit 17-25: 0
	Bit 26: 1 = Subindex in Octet 3 (Standard), 0 = Subindex in Octet 4 (Simodrive 611U)
	Bit 27: 1 = ARRAY-Codes werden nicht unterstützt von PROFIDRIVE-Slave
	Bit 28-31: Nummer der Achse (bei 1-Achsen-Modulen immer 1)
IndexOffset	0
Length	Parameterlänge: 1
Data	bei Response: Anzahl der Array-Elemente des Parameters

5.1.5.5 FDL-Interface

Über das FDL-Interface können per ADS FDL-Telegramme gesendet und empfangene FDL-Telegramme abgeholt werden. Diese Funktionalität wird von der FC31xx ab V02.66 und von der EL6731 ab V01.06 unterstützt.

Um das FDL Interface zu nutzen, muss im Master der Multi-Master Betrieb aktiviert werden.

Klicken sie dazu auf „Bus-Parameter (DP)“:

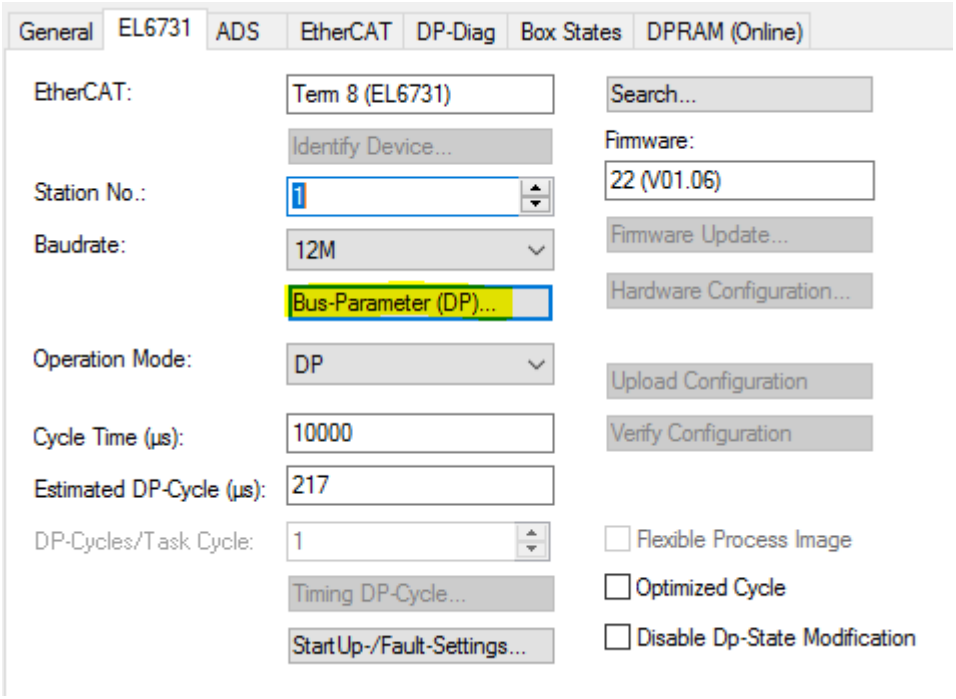


Abb. 26: Karteireiter EL6731, Bus-Parameter (DP)

Aktivieren Sie den Multi-Master Betrieb:

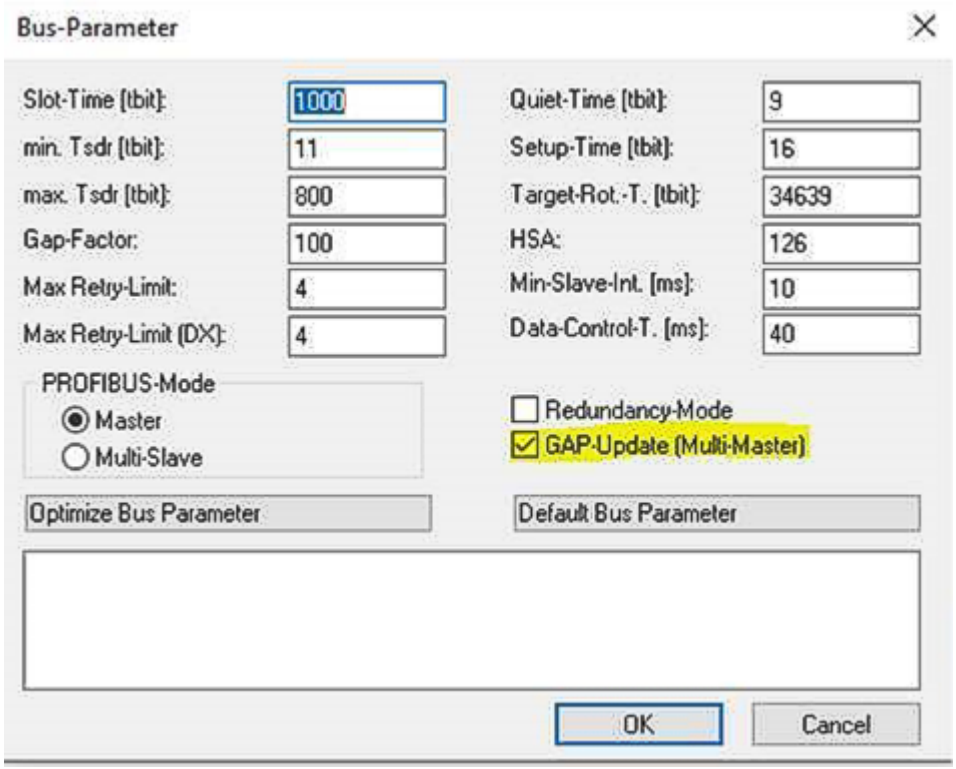


Abb. 27: Multi-Master Betrieb aktivieren

Aktivieren Sie nun per ADS-Write das FDL Interface:

ADS-Write Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	200
IndexGroup	0xF400
IndexOffset	0
Length	1
Data	0xFF (oder die PB eigene Adresse)

Data: 0xFF, es werden alle Daten empfangen, wie auch das Global-Control-Telegramm. Mit der Angabe der eigenen PB Adresse werden nur Daten der eigenen PB Adresse in den Buffer kopiert.

FDL Receive

Mit ADS-Read werden dann entsprechend der Length von einem anderen Master empfangene FDL-Indications oder lokal generierte FDL-Confirmations abgeholt. Für jeden gesendeten FDL-Request wird eine FDL-Confirmation zurückgegeben.

Hier muss der Buffer per ADS READ gepollt werden, um Data zu lesen, ist die Datenlänge 1 mit dem Dateninhalt 0x00 liegen keine Daten im Buffer. Der Buffer ist 0x4000 Byte groß; werden mehr Daten empfangen, gehen diese verloren.

Beispiel um Daten zu schreiben (hier werden 8 Byte Daten vom Master 01 zu Master 02 verschickt):

FDL_DataSend: ARRAY[0..15] OF BYTE :=
[16#01,16#00,16#82,16#81,16#0C,16#0A,16#21,16#20,16#01,16#02,16#03,16#04,16#05,16#06,16#07,16#08];

FDL_DataSend[0]	0x01	Cmd: Request
FDL_DataSend[1]	0x00	Request
FDL_DataSend[2]	0x82	Ziel PB Adresse hier Master Adresse 2
FDL_DataSend[3]	0x81	Ziel PB Adresse hier Master Adresse 1
FDL_DataSend[4]	0x0C	SRD low (Send and Request data)
FDL_DataSend[5]	0x0A	Länge der Daten, ab folgenden Byte gezählt, hier 10 Byte
FDL_DataSend[6]	0x21	SSAP
FDL_DataSend[7]	0x20	SSAP
FDL_DataSend[8..15]	0x01,..0x08	Daten

5.1.6 TwinCAT (2.1x) System Manager

5.1.6.1 Allgemeine Hinweise

5.1.6.1.1 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/.

5.1.6.1.1.1 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.

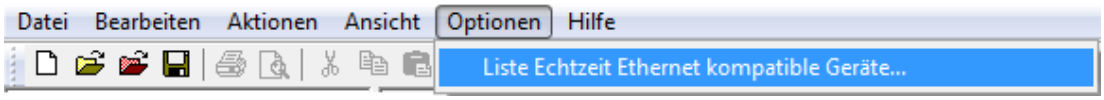


Abb. 28: Aufruf im System Manager (TwinCAT 2)

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

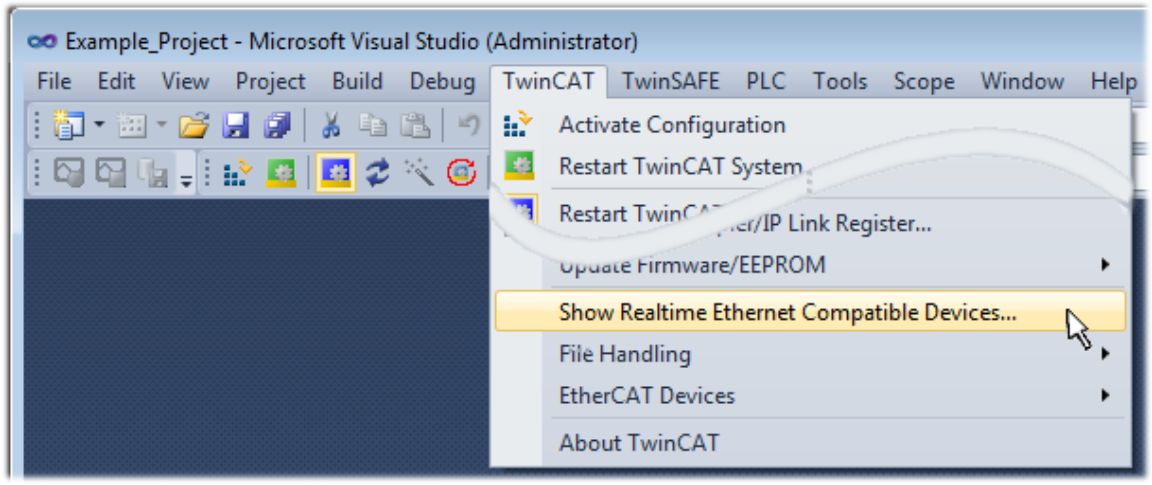


Abb. 29: Aufruf in VS Shell (TwinCAT 3)

B: Über TcRteInstall.exe im TwinCAT-Verzeichnis

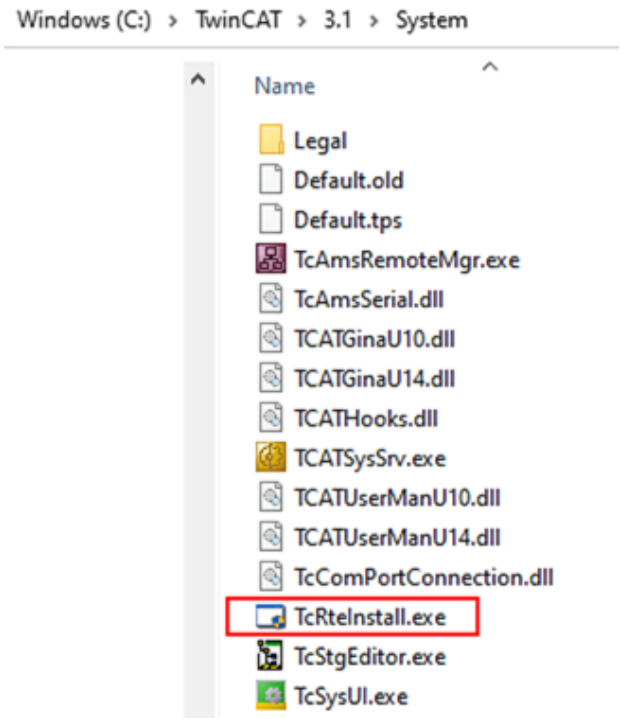


Abb. 30: TcRteInstall.exe im TwinCAT-Verzeichnis

In beiden Fällen erscheint der folgende Dialog:

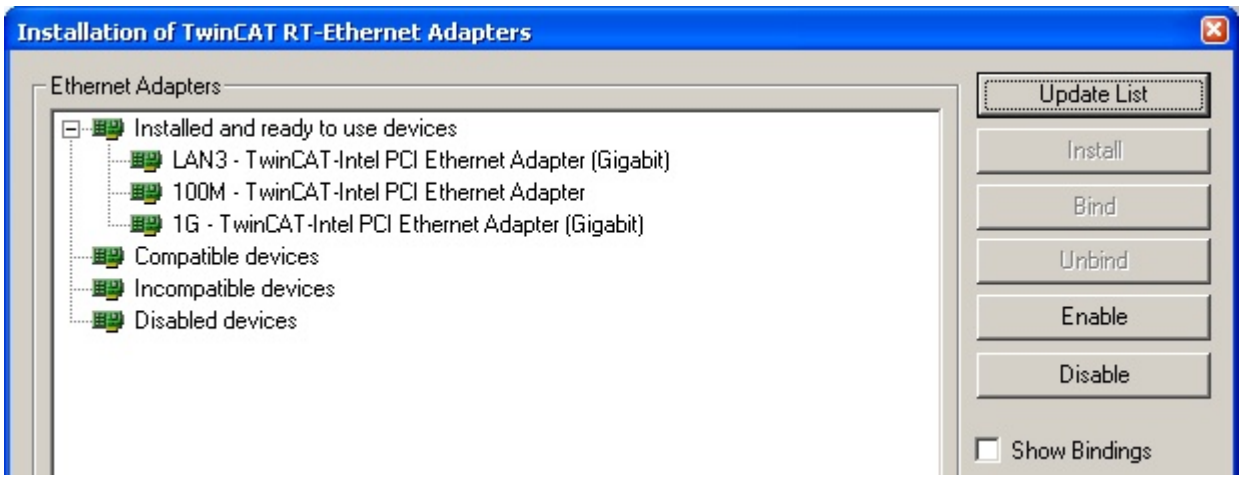


Abb. 31: Übersicht Netzwerkschnittstellen

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

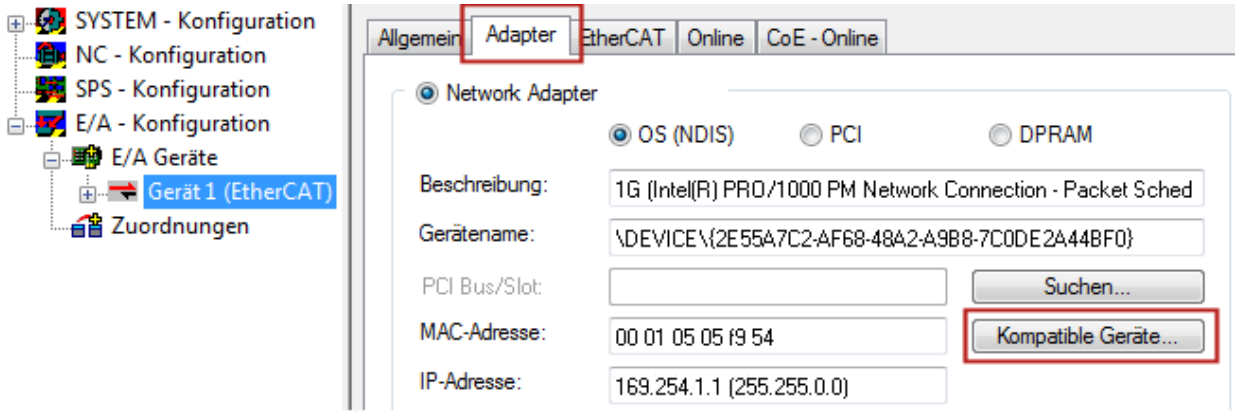
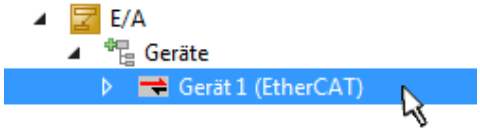


Abb. 32: Eigenschaft von EtherCAT-Gerät (TwinCAT 2): Klick auf „Kompatible Geräte…“ von „Adapter“

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)

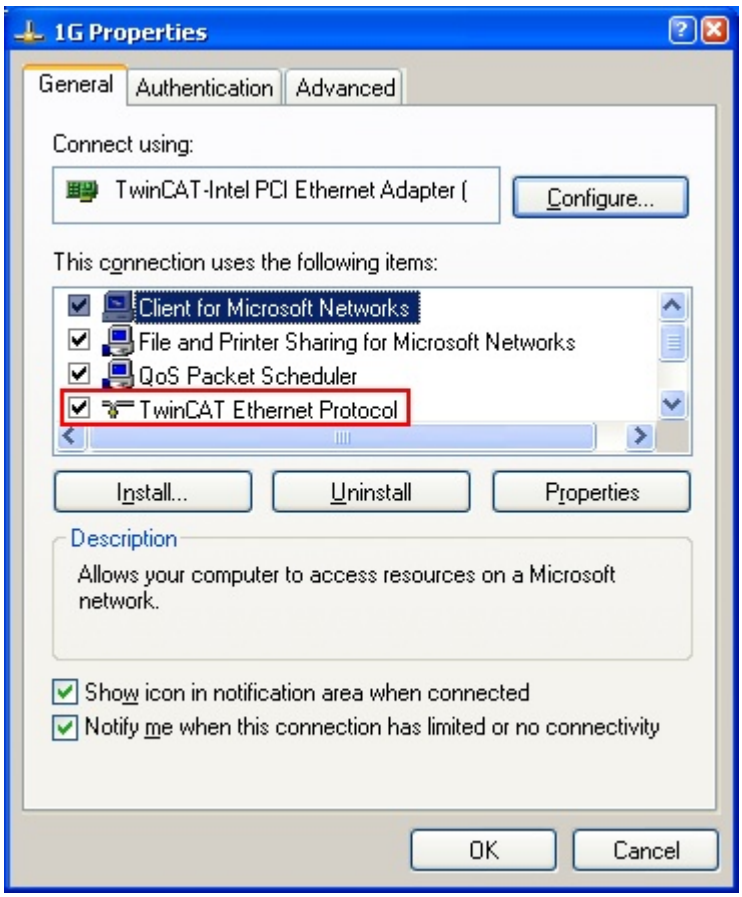


Abb. 33: Windows-Eigenschaften der Netzwerkschnittstelle

Eine korrekte Einstellung des Treibers könnte wie folgt aussehen:

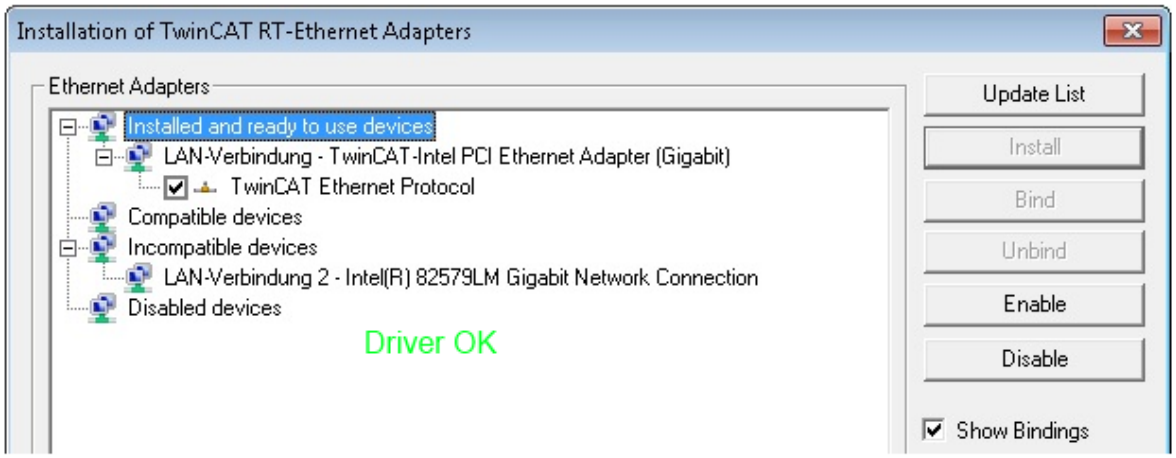


Abb. 34: Beispielhafte korrekte Treiber-Einstellung des Ethernet Ports

Andere mögliche Einstellungen sind zu vermeiden:

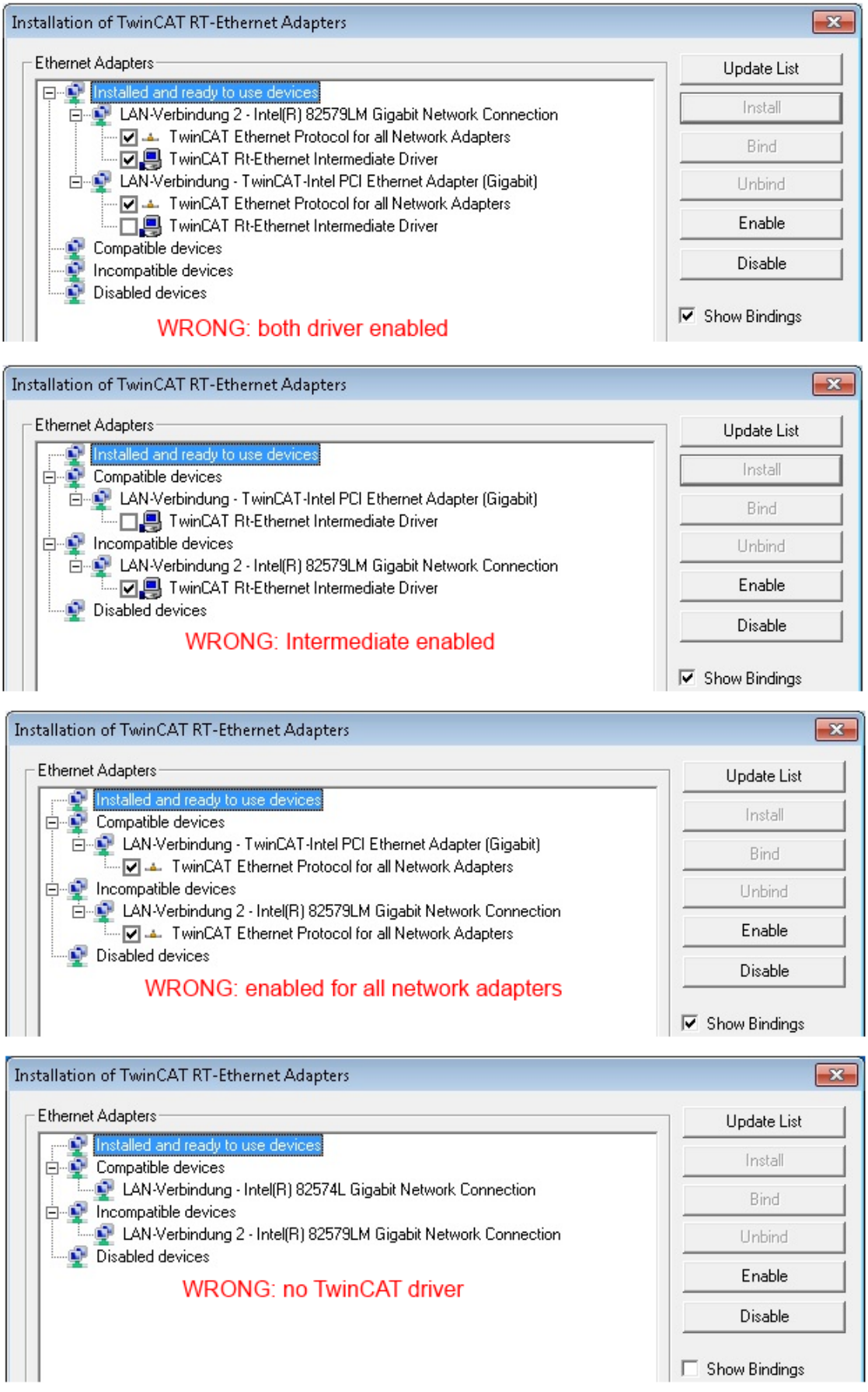


Abb. 35: Fehlerhafte Treiber-Einstellungen des Ethernet Ports

IP-Adresse des verwendeten Ports

IP-Adresse/DHCP

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

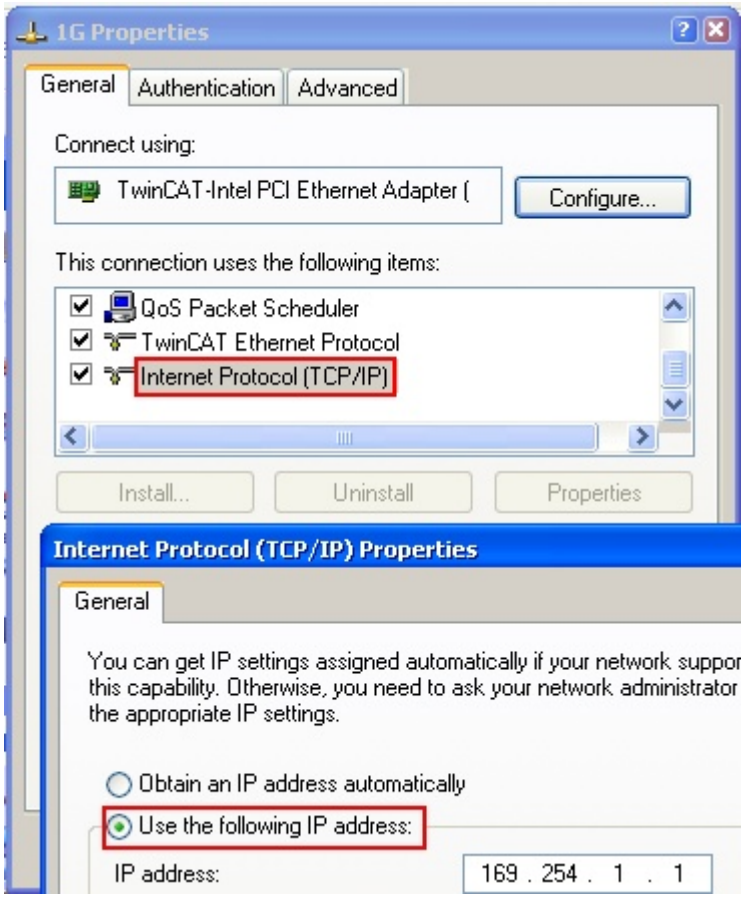


Abb. 36: TCP/IP-Einstellung des Ethernet Ports

5.1.6.1.1.2 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 Dateiarten 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“

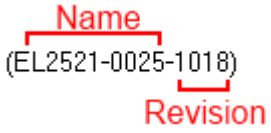


Abb. 37: Gerätebezeichnung: Struktur

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 [▶ 9].

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.

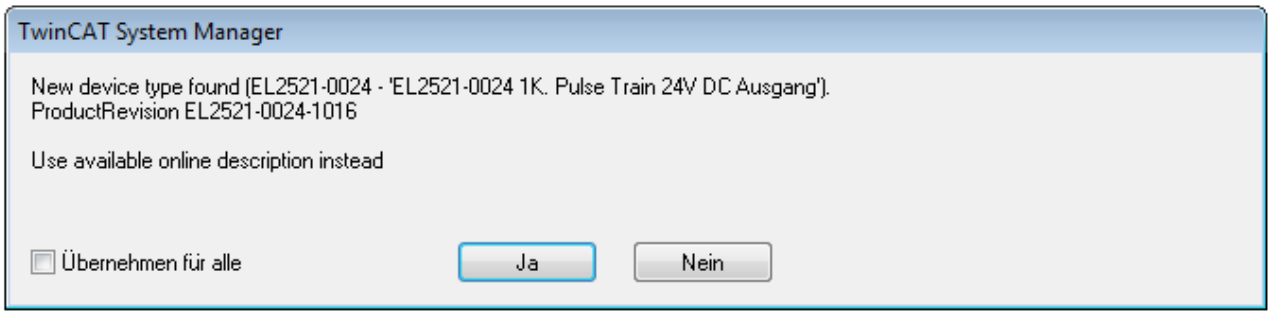


Abb. 38: Hinweisfenster OnlineDescription (TwinCAT 2)

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

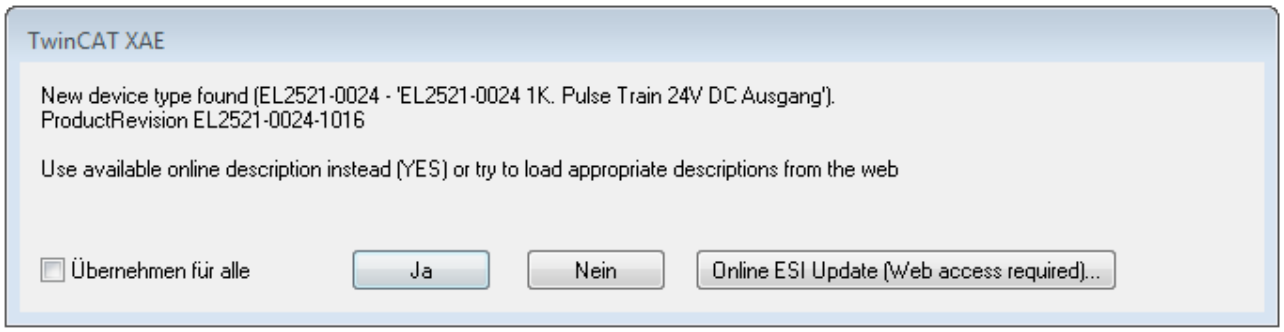


Abb. 39: Hinweisfenster OnlineDescription (TwinCAT 3)

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 unbekannten 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 [▶ 71]“.

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.

OnlineDescriptionCache00000002.xml

Abb. 40: 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. 41: 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.

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.

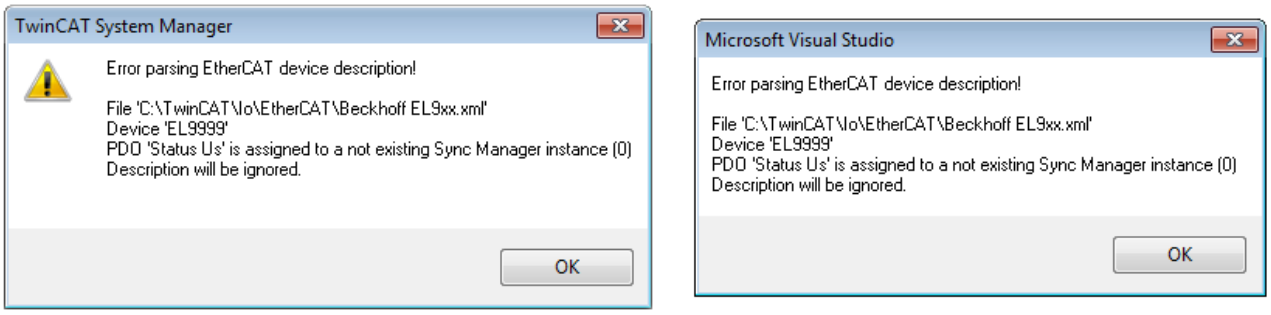


Abb. 42: Hinweisfenster fehlerhafte ESI-Datei (links: TwinCAT 2; rechts: TwinCAT 3)

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

5.1.6.1.1.3 OFFLINE Konfigurationserstellung

Anlegen des Geräts EtherCAT

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

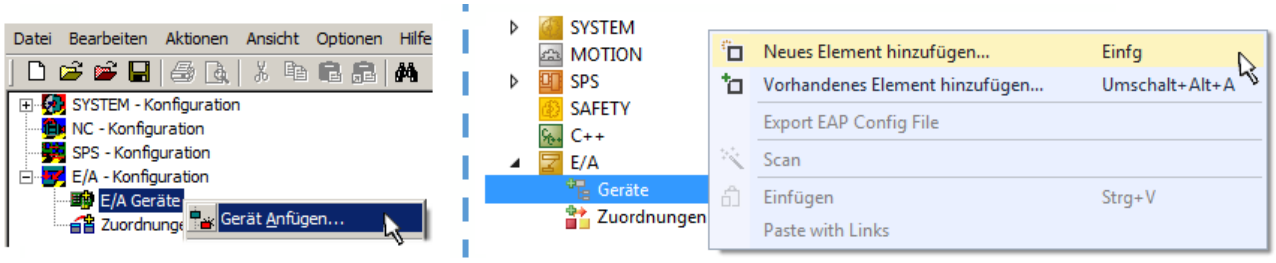


Abb. 43: Anfügen eines EtherCAT Device: links TwinCAT 2; rechts TwinCAT 3

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.

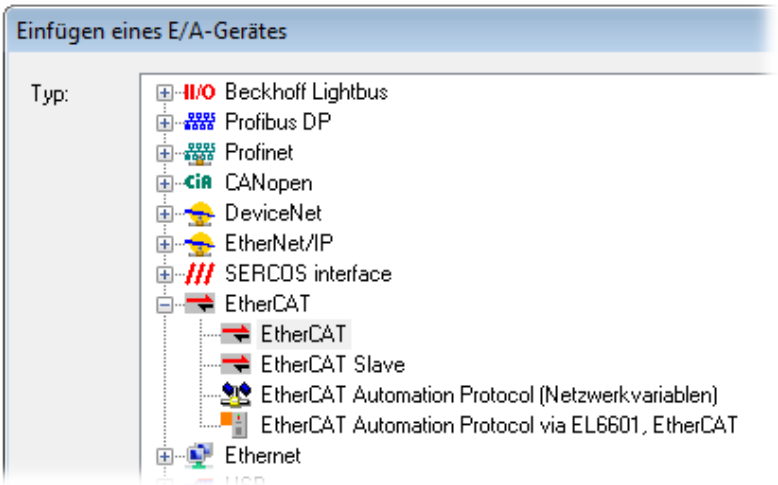


Abb. 44: Auswahl EtherCAT Anschluss (TwinCAT 2.11, TwinCAT 3)

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

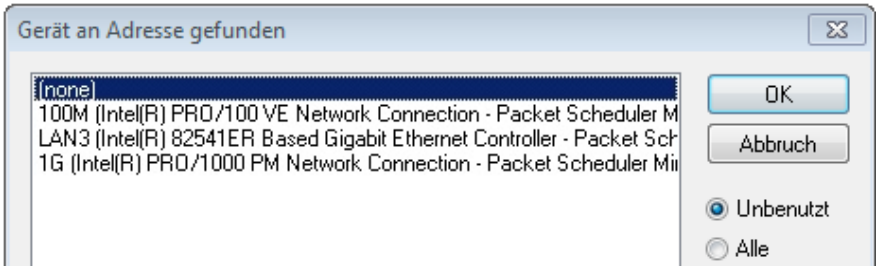


Abb. 45: Auswahl Ethernet Port

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)“.

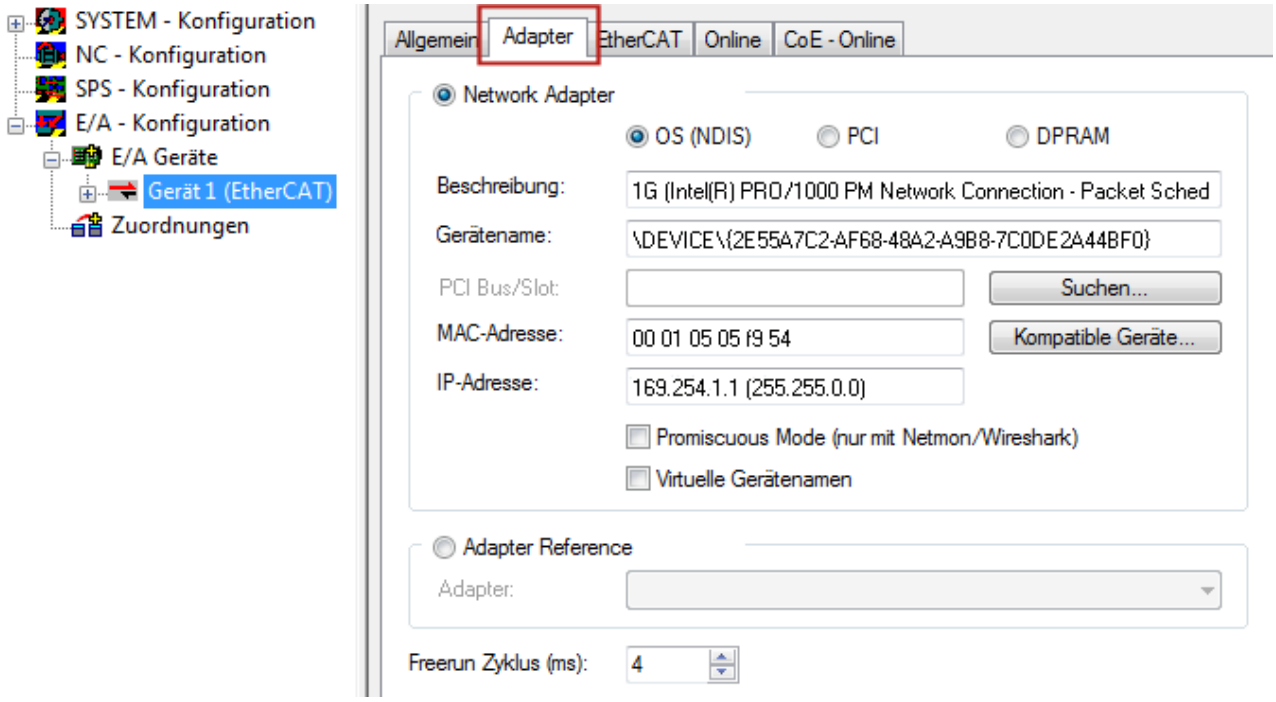
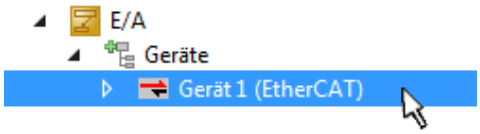


Abb. 46: Eigenschaften EtherCAT-Gerät (TwinCAT 2)

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



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 [▶ 61].

Definieren von EtherCAT Slaves

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

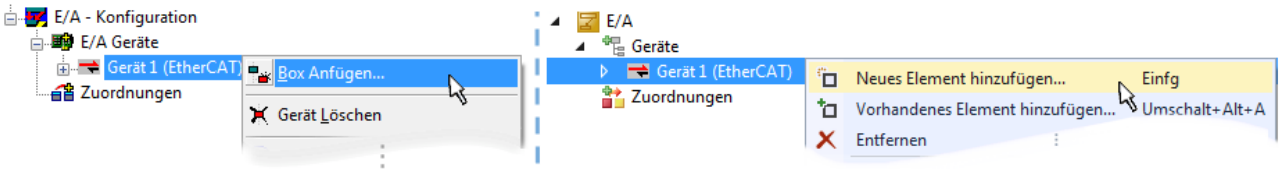


Abb. 47: Anfügen von EtherCAT-Geräten (links: TwinCAT 2; rechts: TwinCAT 3)

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).

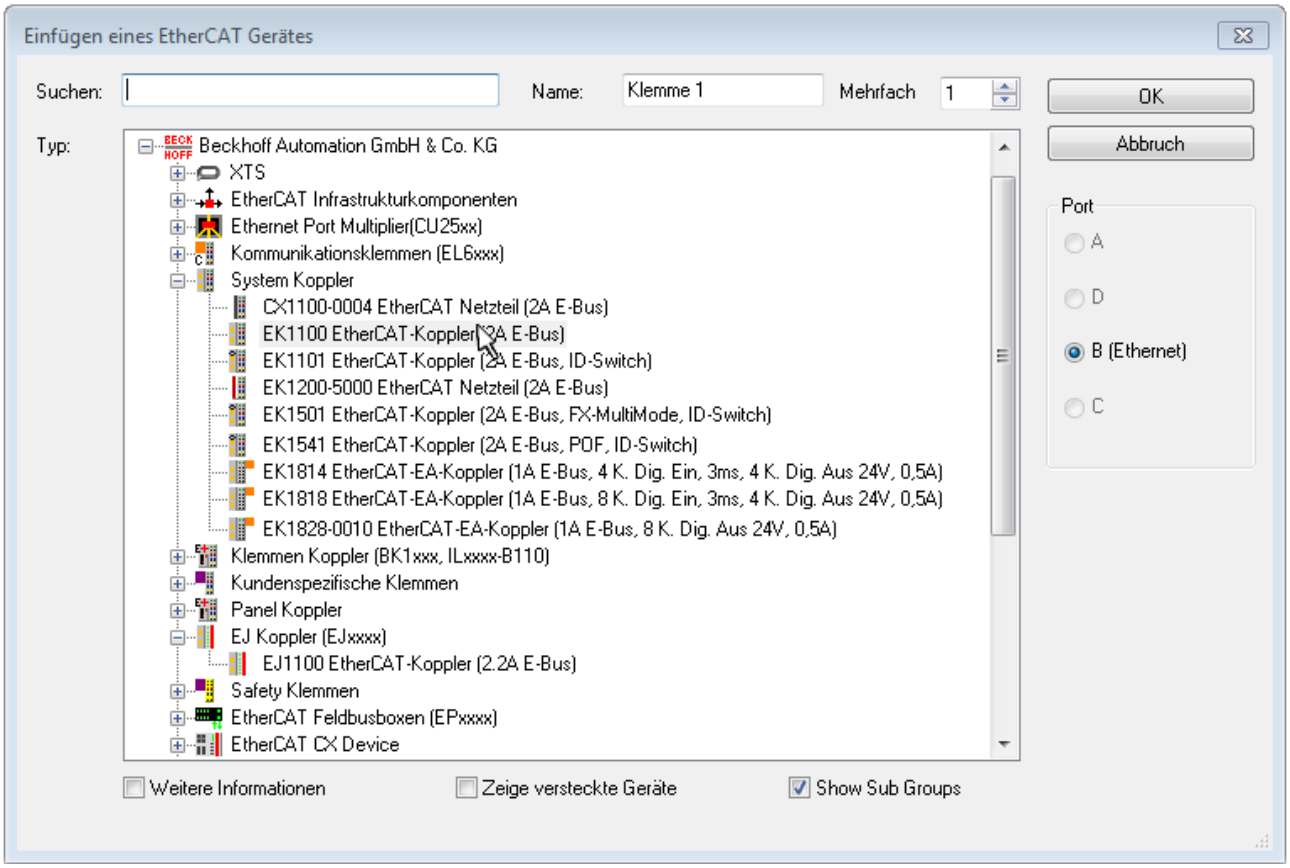


Abb. 48: Auswahldialog neues EtherCAT-Gerät

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

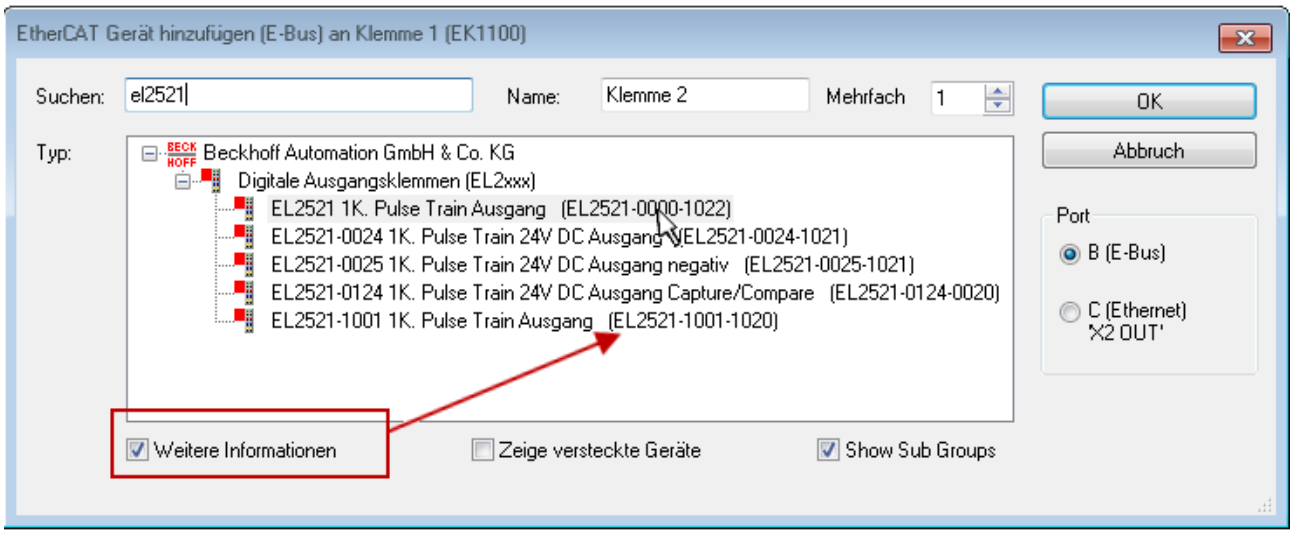


Abb. 49: Anzeige Geräte-Revision

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“.

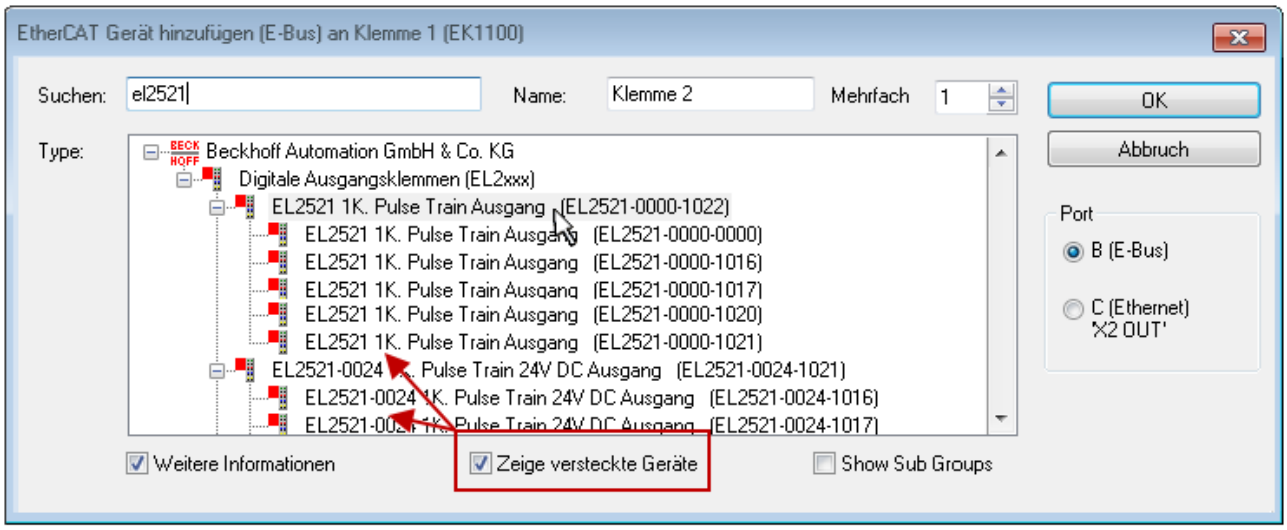


Abb. 50: Anzeige vorhergehender Revisionen

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.

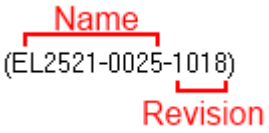


Abb. 51: Name/Revision Klemme

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 parametriert werden: Verlinkung mit der Task, CoE/DC-Einstellungen, PlugIn-Definition, StartUp-Einstellungen, ...

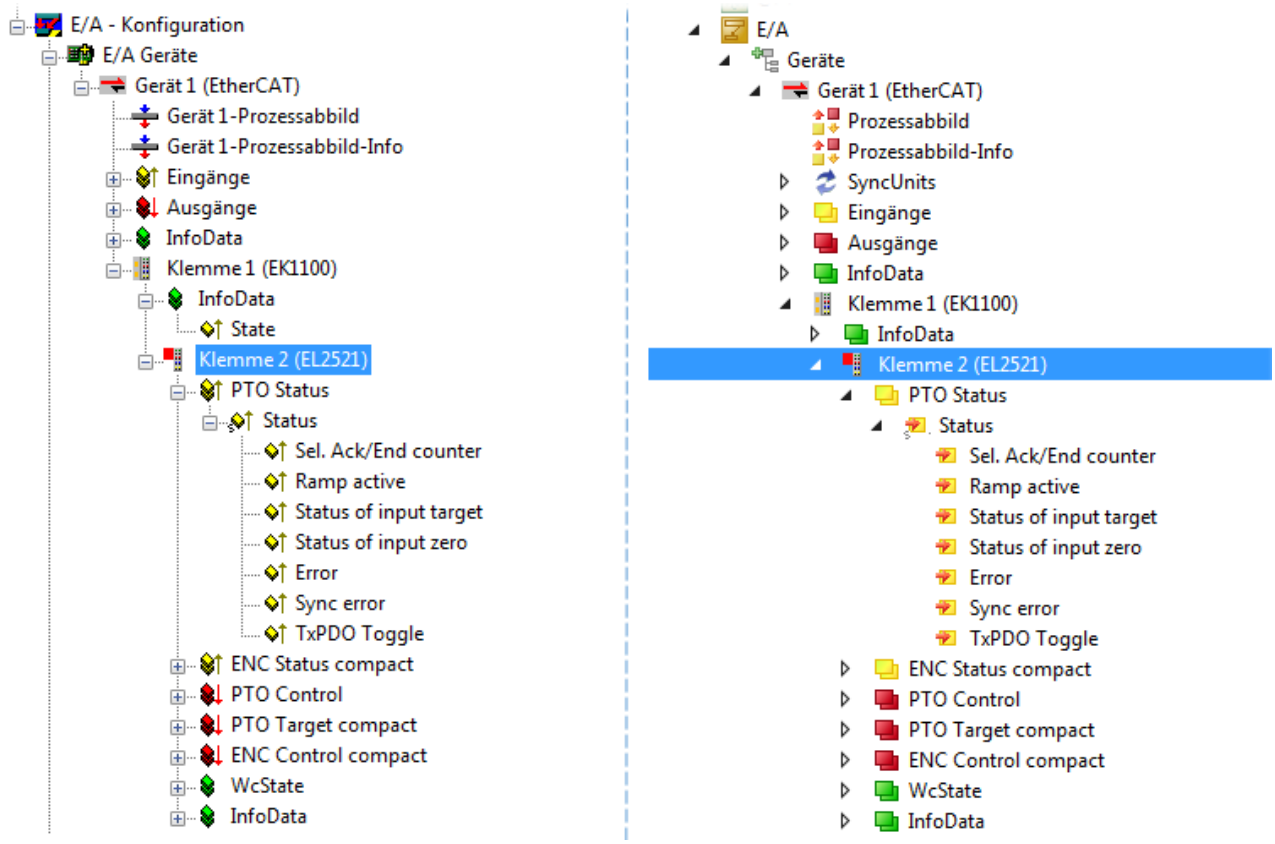


Abb. 52: EtherCAT Klemme im TwinCAT-Baum (links: TwinCAT 2; rechts: TwinCAT 3)

5.1.6.1.1.4 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: Config Mode .
- bei der Benutzeroberfläche der TwinCAT 3 Entwicklungsumgebung durch ein Symbol .

TwinCAT lässt sich in diesem Modus versetzen:

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

Online Scannen im Config Mode

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. 53: 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.

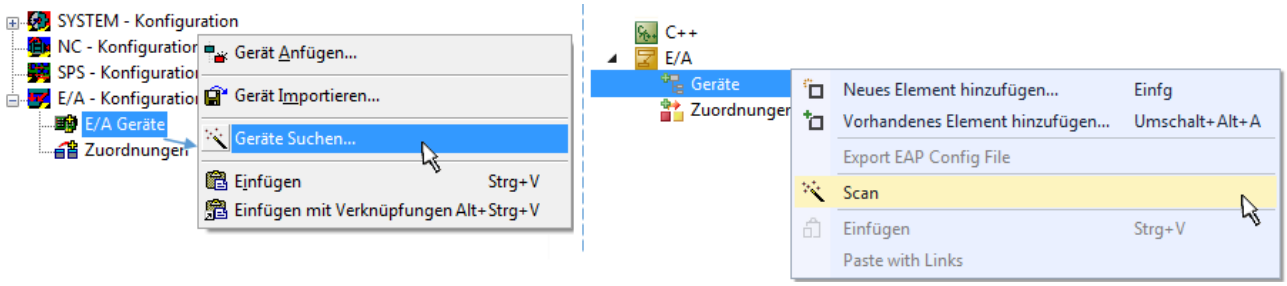


Abb. 54: Scan Devices (links: TwinCAT 2; rechts: TwinCAT 3)

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.

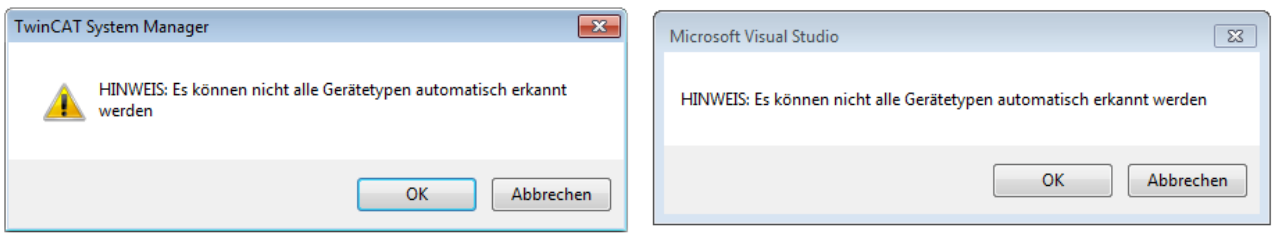


Abb. 55: Hinweis automatischer GeräteScan (links: TwinCAT 2; rechts: TwinCAT 3)

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.

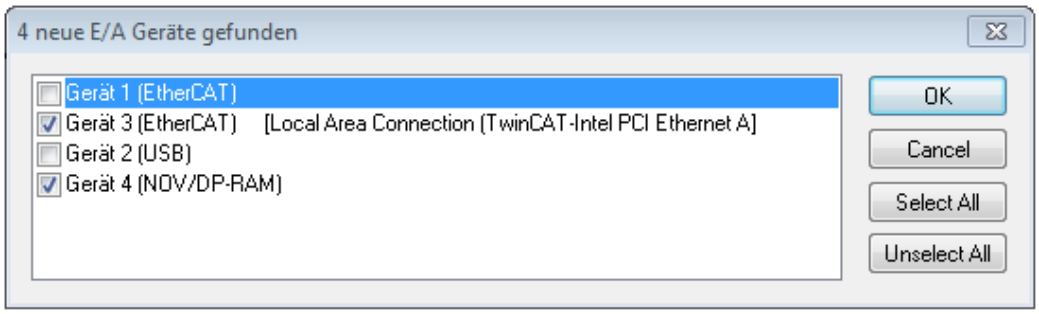


Abb. 56: Erkannte Ethernet-Geräte

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

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 [▶ 61].

Erkennen/Scan der EtherCAT Teilnehmer

Funktionsweise Online Scan

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

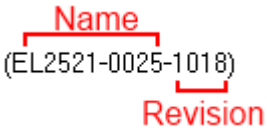


Abb. 57: Beispiel Default-Zustand

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 [▶ 81] 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 Geräterevision 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:

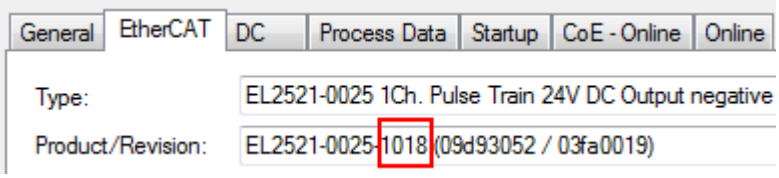


Abb. 58: Einbau EtherCAT-Klemme mit Revision -1018

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** -1**019**. 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 vergleichernder Scan [▶ 81] gegen die Erstkonfiguration „B.tsm“ sinnvoll.

Wird nun allerdings in der Abteilung Seriennmaschinenbau 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:

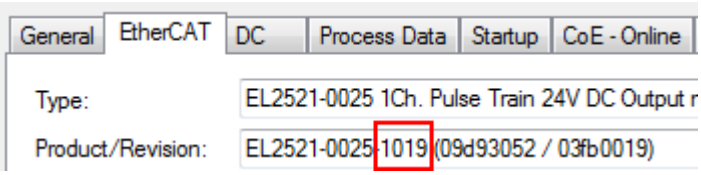


Abb. 59: Erkennen EtherCAT-Klemme mit Revision -1019

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

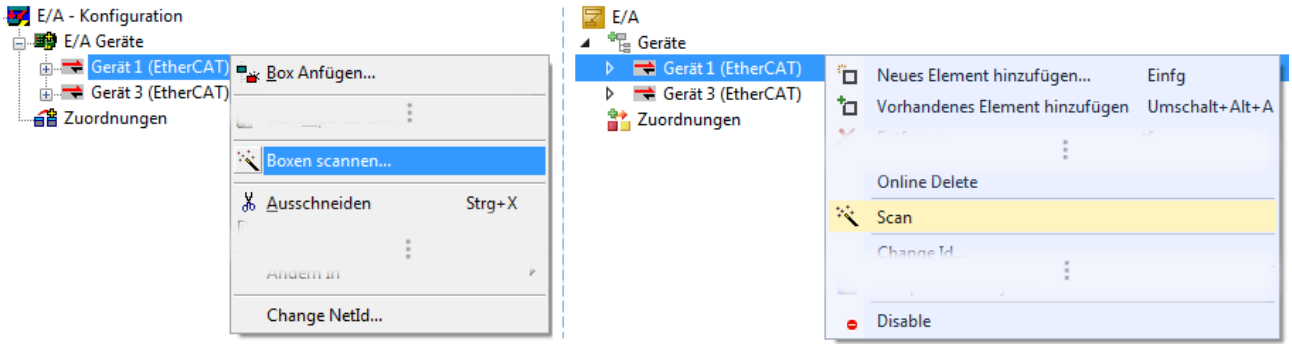


Abb. 61: Manuelles Scannen nach Teilnehmern auf festgelegtem EtherCAT Device (links: TC2; rechts TC3)

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

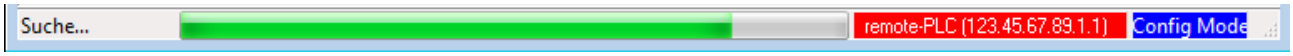


Abb. 62: Scanfortschritt am Beispiel von TwinCAT 2

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

Abb. 63: 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.

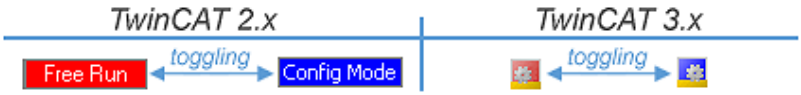


Abb. 64: Anzeige des Wechsels zwischen „Free Run“ und „Config Mode“ unten rechts in der Statusleiste

Abb. 65: 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.

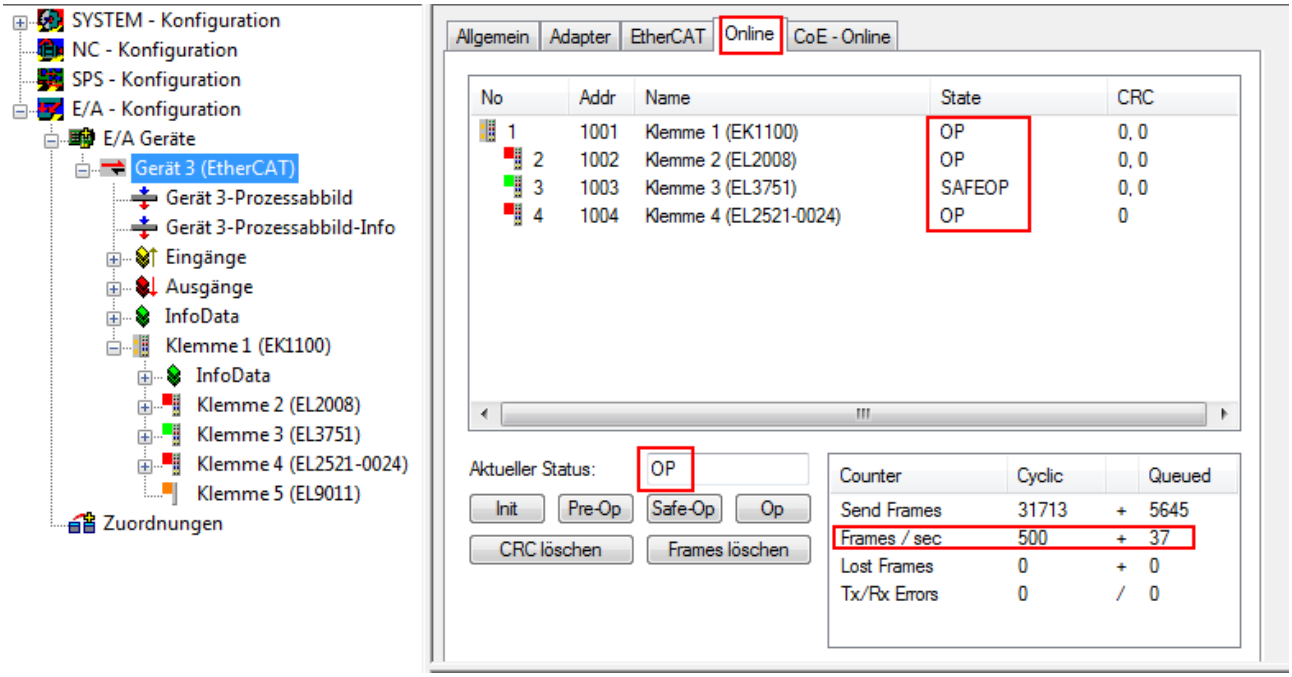


Abb. 66: Beispielhafte Online-Anzeige

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 [▶ 71] 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.

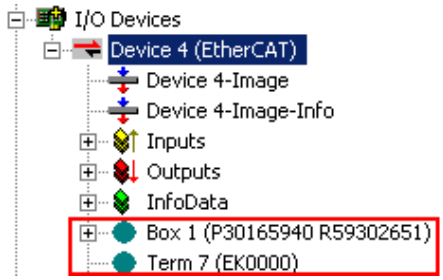


Abb. 67: Fehlerhafte Erkennung

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), Gerätename 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. 68: Identische Konfiguration (links: TwinCAT 2; rechts TwinCAT 3)

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

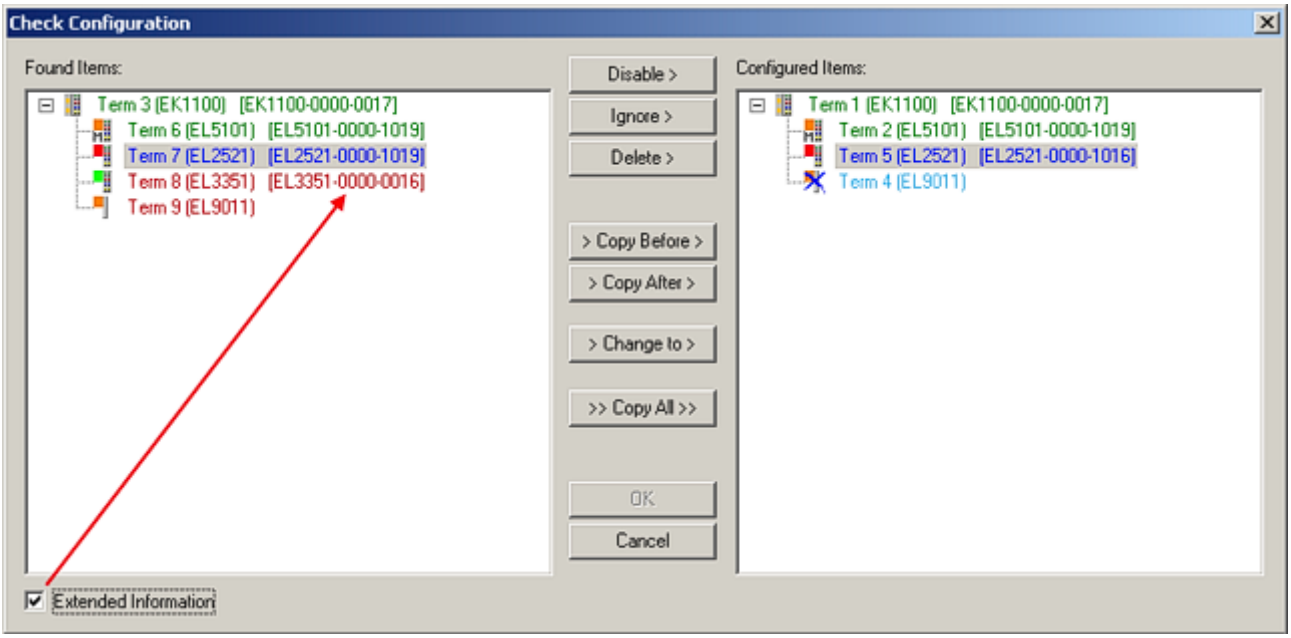


Abb. 69: Korrekturdialog

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

Farbe	Erläuterung
grün	Dieser EtherCAT Slave findet seine Entsprechung auf der Gegenseite. Typ und Revision stimmen überein.
blau	Dieser EtherCAT Slave ist auf der Gegenseite vorhanden, aber in einer anderen Revision. Diese andere Revision kann andere Default-Einstellungen der Prozessdaten und andere/zusätzliche Funktionen haben. Ist die gefundene Revision > als die konfigurierte Revision, ist der Einsatz unter Berücksichtigung der Kompatibilität möglich. Ist die gefundene Revision < als die konfigurierte Revision, ist der Einsatz vermutlich nicht möglich. Eventuell unterstützt das vorgefundene Gerät nicht alle Funktionen, die der Master von ihm aufgrund der höheren Revision erwartet.
hellblau	Dieser EtherCAT Slave wird ignoriert (Button „Ignore“)
rot	• 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. 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. 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.

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.

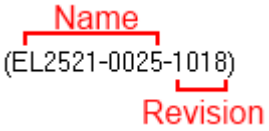


Abb. 70: Name/Revision Klemme

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.

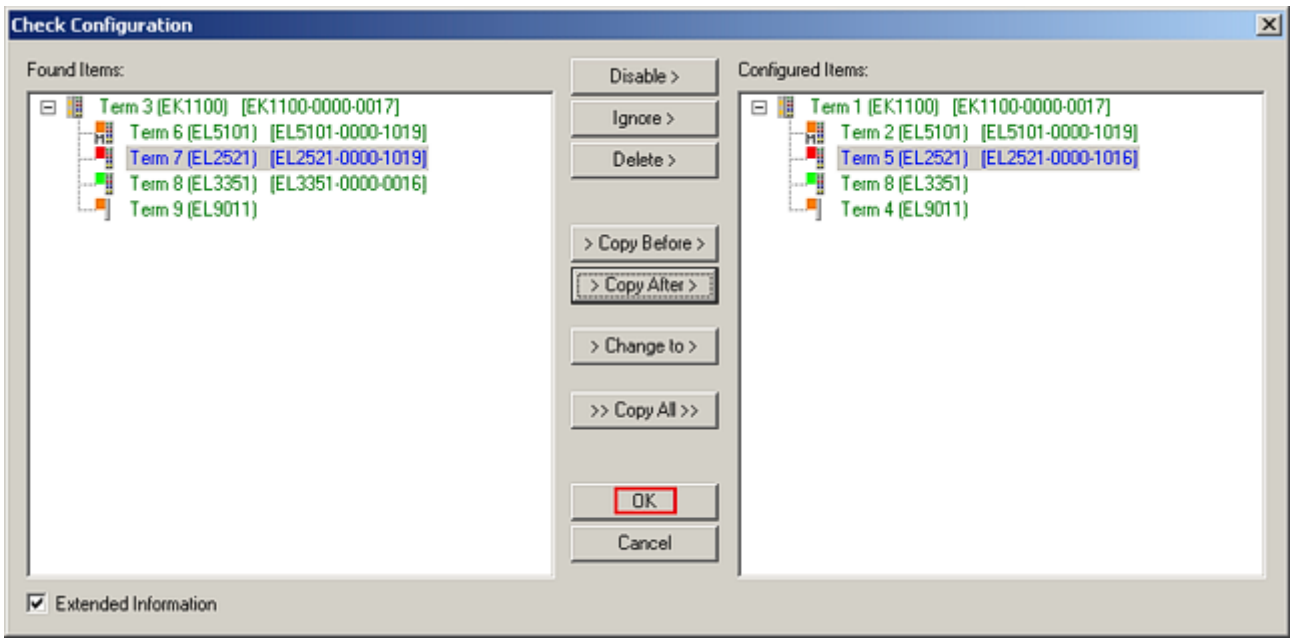


Abb. 71: Korrekturdialog mit Änderungen

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.

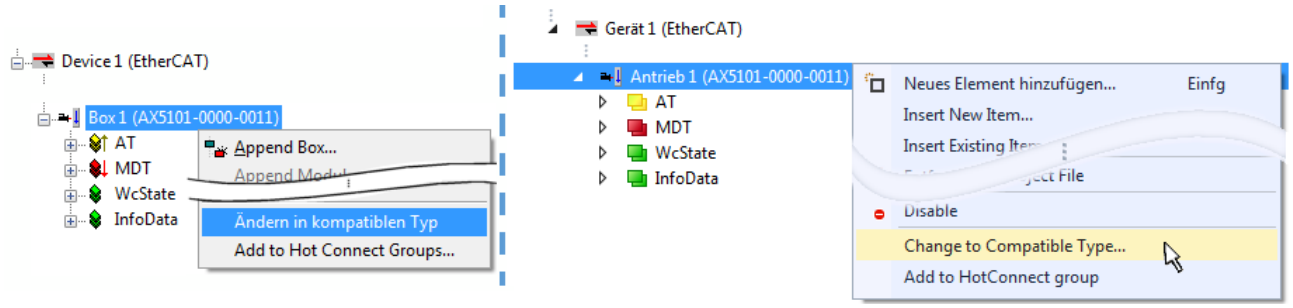


Abb. 72: Dialog „Change to Compatible Type…“ (links: TwinCAT 2; rechts TwinCAT 3)

Folgende Elemente in der ESI eines EtherCAT-Teilenhmers 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

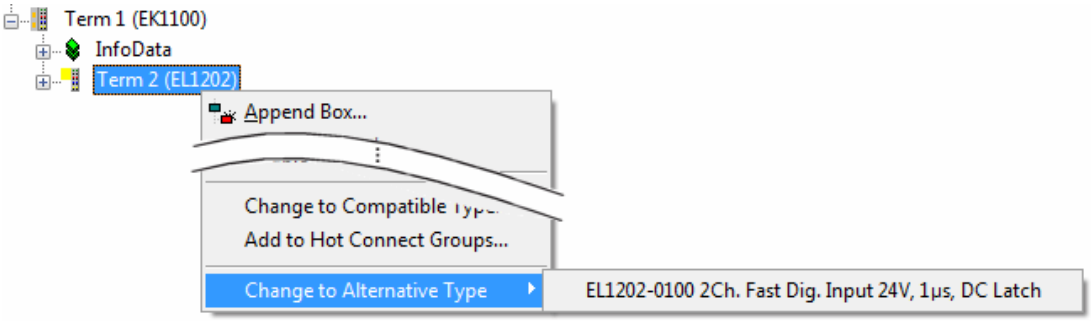


Abb. 73: TwinCAT 2 Dialog Change to Alternative Type

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.

5.1.6.1.1.5 Allgemeine Slave PDO Konfiguration

Die von einem EtherCAT Slave zyklisch übertragenen Prozessdaten (**P**rocess **D**ata **O**bjects, PDO) 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 gilt im Allgemeinen:

- Die vom Gerät unterstützten Prozessdaten Input/Output sind in der ESI/XML-Beschreibung herstellerseitig definiert. Der TwinCAT EtherCAT Master verwendet die ESI-Beschreibung zur richtigen Konfiguration des Slaves.
- Wenn vorgesehen, können die Prozessdaten im Systemmanager 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 den erfolgreichen Hochlauf des Slaves. Es wird abgeraten, andere als die vorgesehene Prozessdaten zu konfigurieren, denn die Geräte-Firmware (wenn vorhanden) ist auf diese PDO-Kombinationen abgestimmt.

Ist lt. 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: im Reiter "Process Data"in der Input- oder Output-Syncmanager zu wählen (C)
- D: die PDOs können an- bzw. abgewählt werden
- H: die neuen Prozessdaten sind als verlinkbare Variablen im Systemmanager 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.

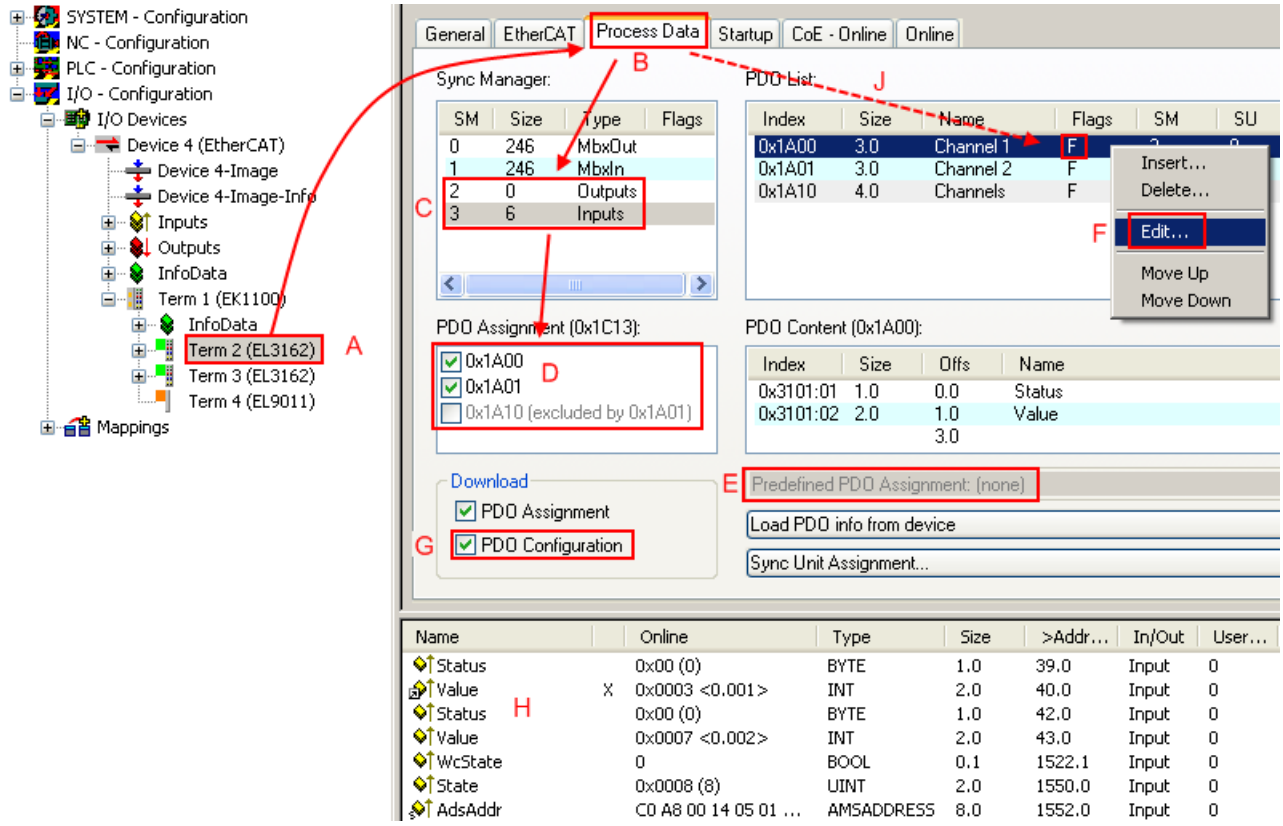


Abb. 74: *Konfigurieren der Prozessdaten*

Manuelle Veränderung der Prozessdaten

In der PDO-Übersicht kann lt. 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 eine 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 Systemmanager ausgegeben:Diese Fehlermeldung "invalid SM IN cfg" oder "invalid SM OUT cfg" bietet gleich einen Hinweis auf die Ursache des fehlgeschlagenen Starts.

5.1.6.1.2 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.

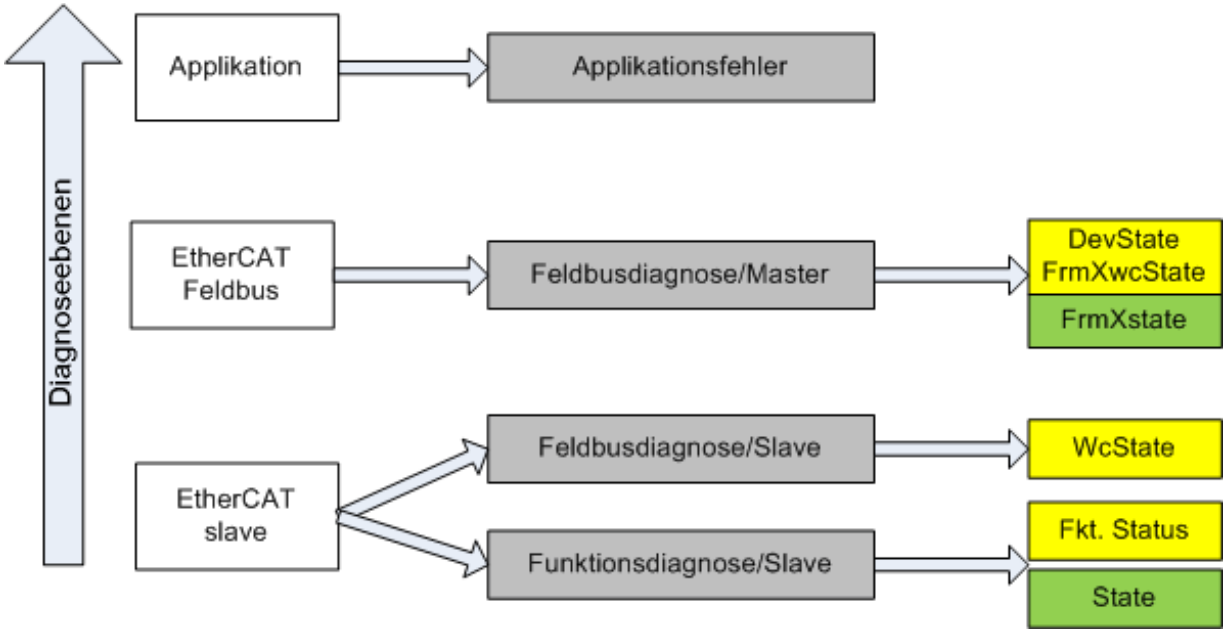


Abb. 75: Auswahl an Diagnoseinformationen eines EtherCAT Slave

Im Allgemeinen verfügt ein EtherCAT Slave über

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

  als auch über
- kanal-typische Funktionsdiagnose (geräteabhängig)
  Siehe entsprechende Gerätedokumentation

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

Farbe	Bedeutung
gelb	Eingangsvariablen vom Slave zum EtherCAT Master, die in jedem Zyklus aktualisiert werden
rot	Ausgangsvariablen vom Slave zum EtherCAT Master, die in jedem Zyklus aktualisiert werden
grün	Informationsvariabeln 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.

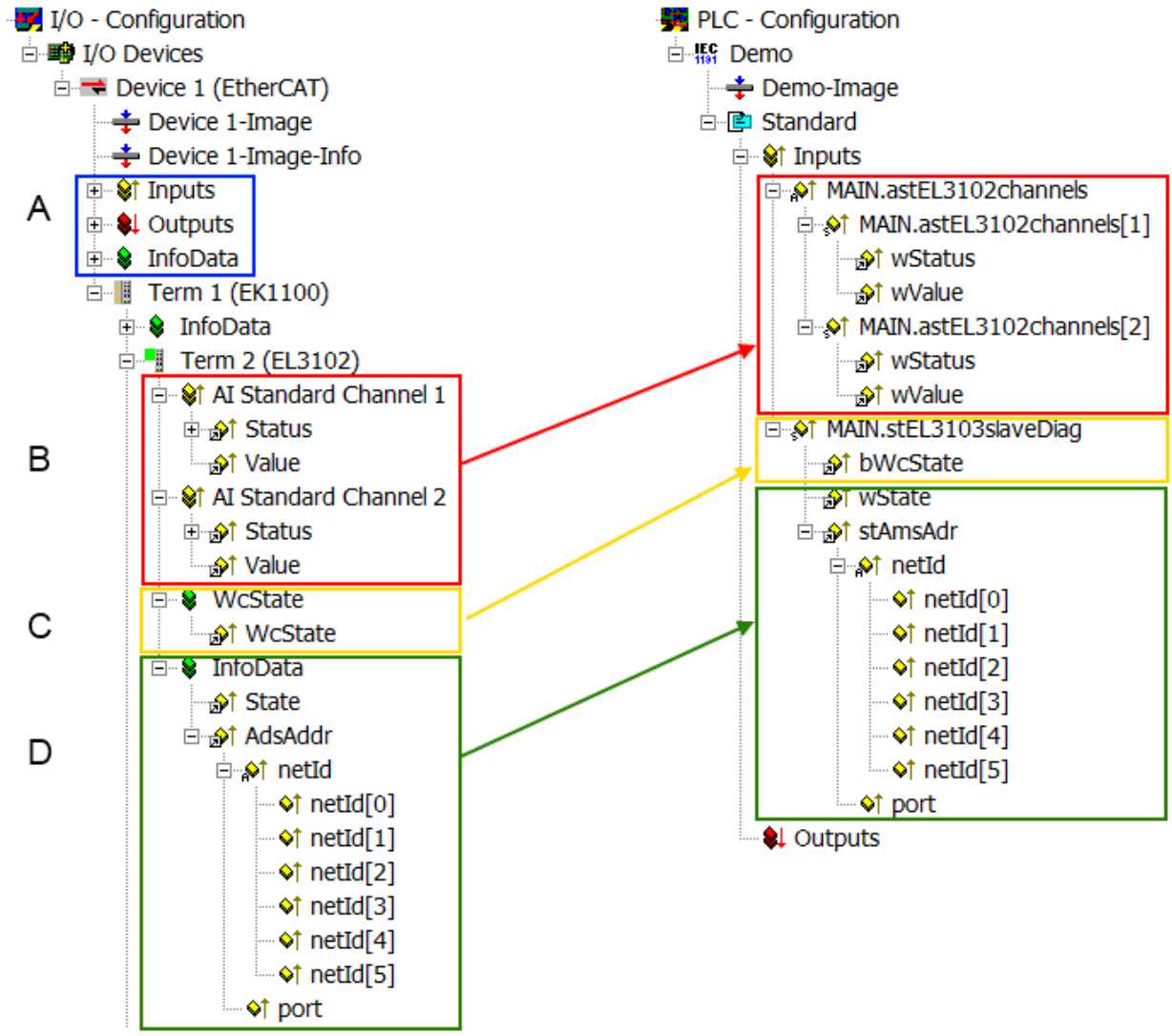


Abb. 76: Grundlegende EtherCAT Slave Diagnose in der PLC

Dabei werden folgende Aspekte abgedeckt:

Kennzeichen	Funktion	Ausprägung	Anwendung/Auswertung
A	Diagnoseinformationen des EtherCAT Master zyklisch aktualisiert (gelb) oder azyklisch bereitgestellt (grün).		Zumindest der DevState ist in der PLC zyklusaktuell auszuwerten. Die Diagnoseinformationen des EtherCAT Master bieten noch weitaus mehr Möglichkeiten, die in der EtherCAT-Systemdokumentation behandelt werden. Einige Stichworte: • CoE im Master zur Kommunikation mit/über die Slaves • Funktionen aus *TcEtherCAT.lib* • OnlineScan durchführen
B	Im gewählten Beispiel (EL3102) umfasst die EL3102 zwei analoge Eingangskanäle, die einen eigenen Funktionsstatus zyklusaktuell übermitteln.	Status • 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	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 1. am EtherCAT Slave als auch inhaltsidentisch 2. als Sammelvariable am EtherCAT Master (siehe Punkt A) zur Verlinkung bereitgestellt.	WcState (Working Counter) 0: gültige Echtzeitkommunikation im letzten Zyklus 1: ungültige Echtzeitkommunikation ggf. Auswirkung auf die Prozessdaten anderer Slaves, die in der gleichen SyncUnit liegen	Damit sich die übergeordnete PLC-Task (oder entsprechende Steueranwendungen) auf korrekte Daten verlassen kann, muss dort der Kommunikationsstatus des EtherCAT Slaves ausgewertet werden. Deshalb werden solche Informationen zyklusaktuell mit den Prozessdaten bereitgestellt.
D	Diagnoseinformationen des EtherCAT Masters, die zwar am Slave zur Verlinkung dargestellt werden, aber tatsächlich vom Master für den jeweiligen Slave ermittelt und dort dargestellt werden. Diese Informationen haben keinen Echtzeit-Charakter weil sie • nur selten/nie verändert werden, außer beim Systemstart • selbst auf azyklischem Weg ermittelt werden (z.B. EtherCAT Status)	State aktueller Status (INIT..OP) des Slaves. Im normalen Betriebszustand muss der Slave im OP (=8) sein. *AdsAddr* 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 *port* (= EtherCAT Adresse) ist der einzelne Slave ansprechbar.	Informationsvariabeln 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*:

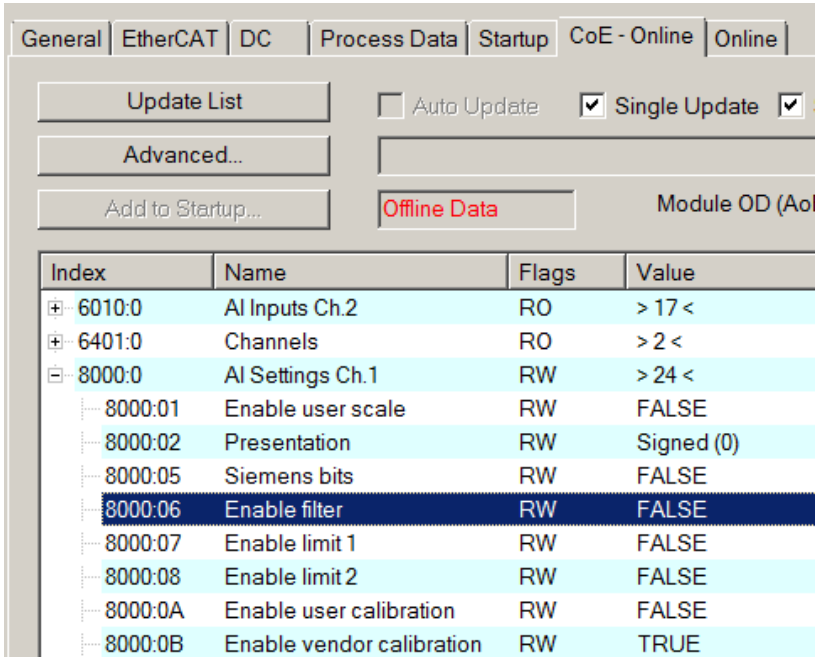


Abb. 77: EL3102, CoE-Verzeichnis

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

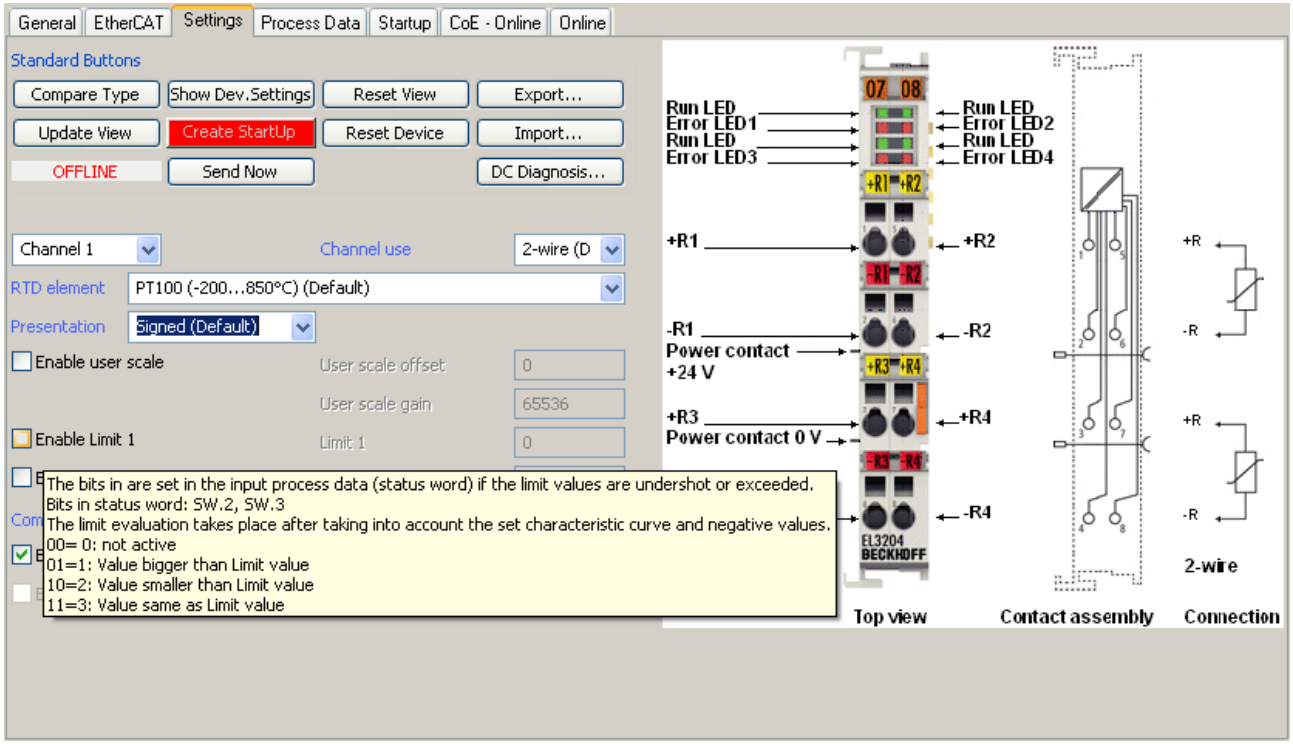


Abb. 78: Beispiel Inbetriebnahmehilfe für eine EL3204

Diese Inbetriebnahme verwaltet zugleich

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

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

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

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

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

- INIT
- PREOP
- SAFEOP
- OP

zu durchlaufen. Der EtherCAT Master ordnet diese Zustände an in Abhängigkeit der Initialisierungsroutinen, die zur Inbetriebnahme des Gerätes durch die ES/XML und Anwendereinstellungen (Distributed Clocks (DC), PDO, CoE) definiert sind. Siehe dazu auch Kapitel "Grundlagen der Kommunikation, EtherCAT State Machine [▶ 37]. 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.

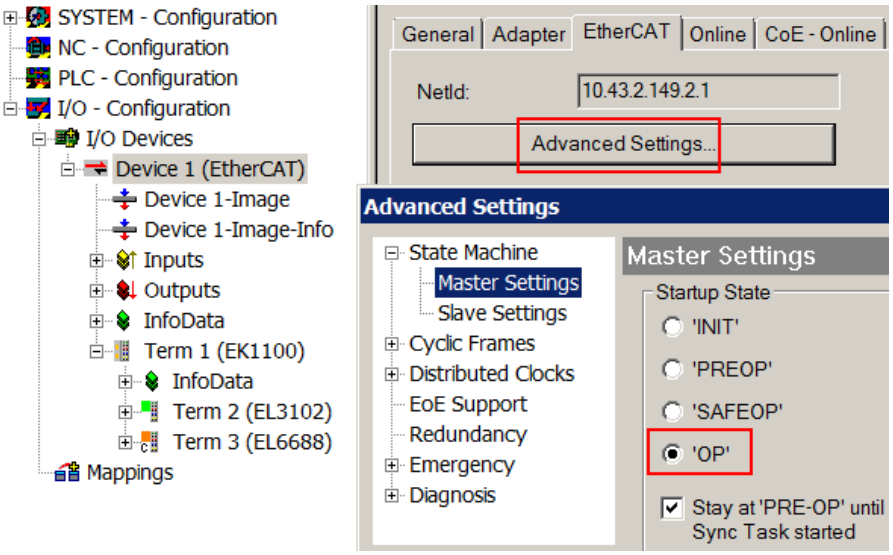


Abb. 79: Default Verhalten System Manager

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

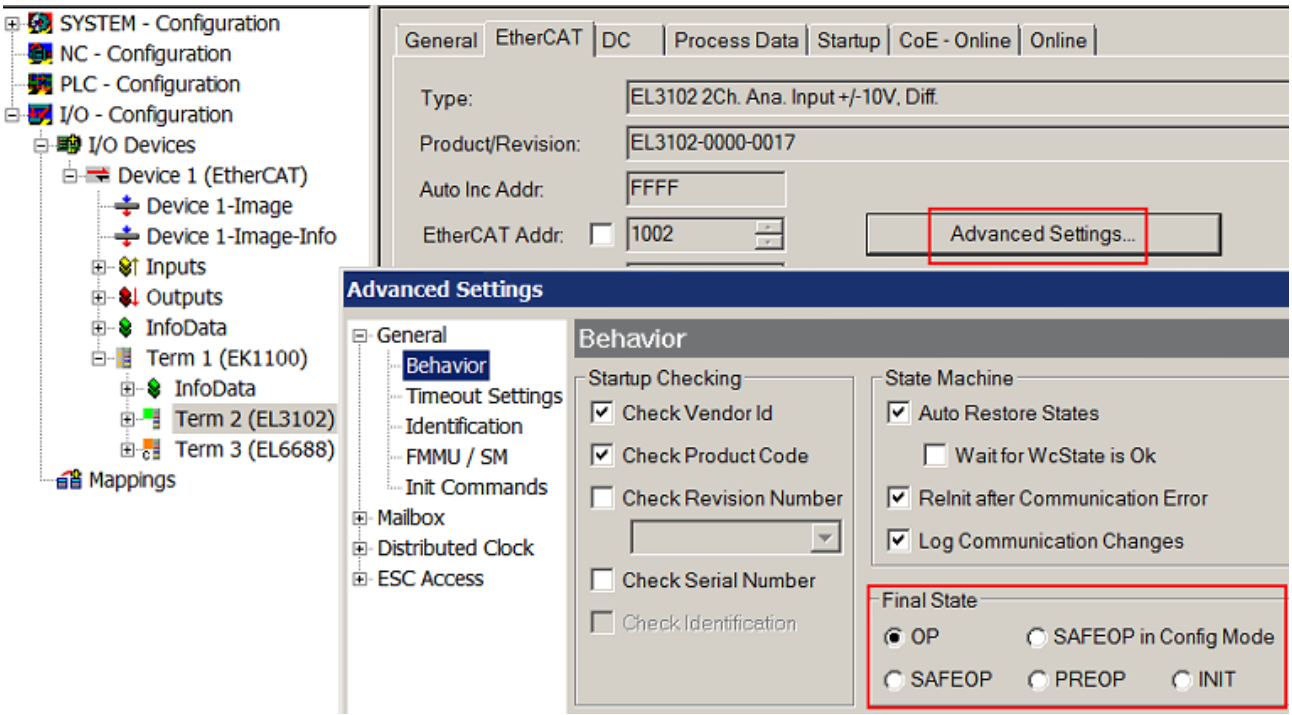


Abb. 80: Default Zielzustand im Slave

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.

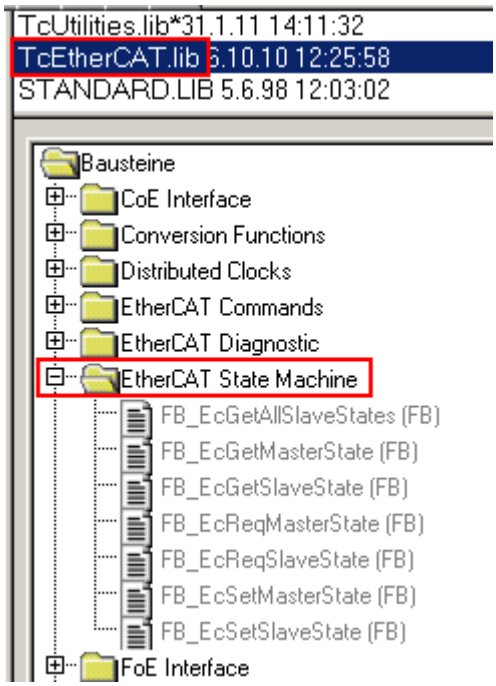
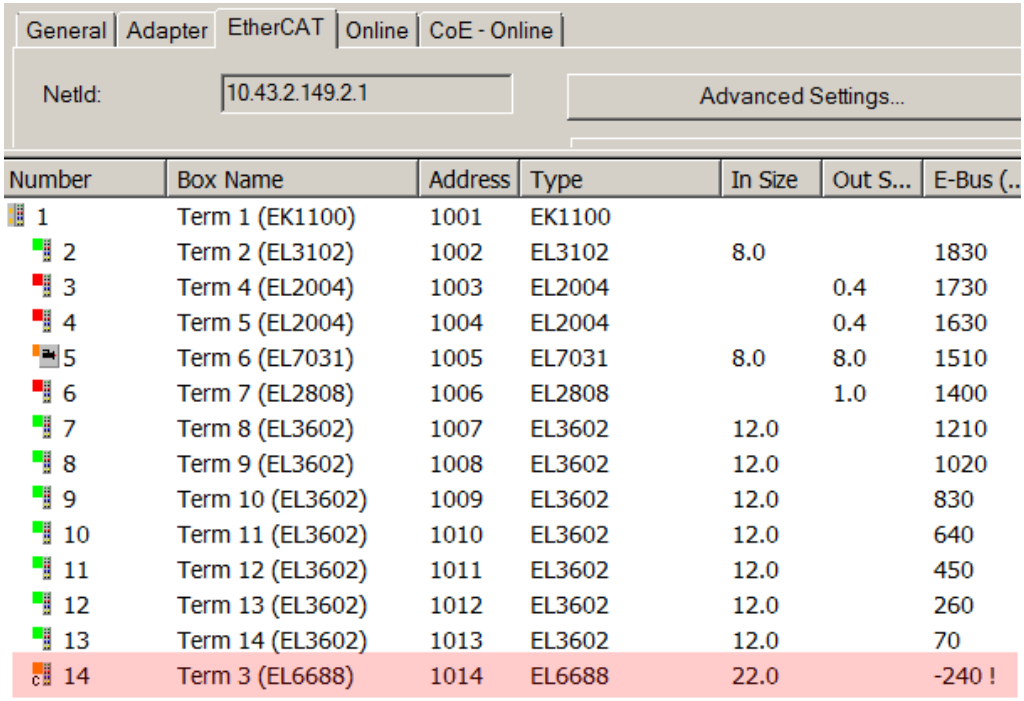


Abb. 81: PLC-Bausteine

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.



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. 82: 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:

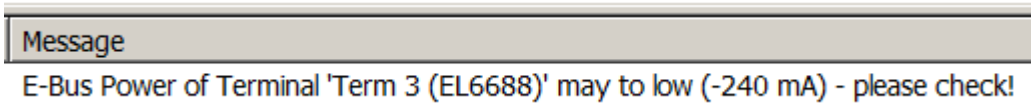


Abb. 83: Warnmeldung E-Bus-Überschreitung

HINWEIS

Achtung! Fehlfunktion möglich!

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

5.1.6.2 TwinCAT-Baum "Gerät EL6731"

5.1.6.2.1 Karteireiter EL6731

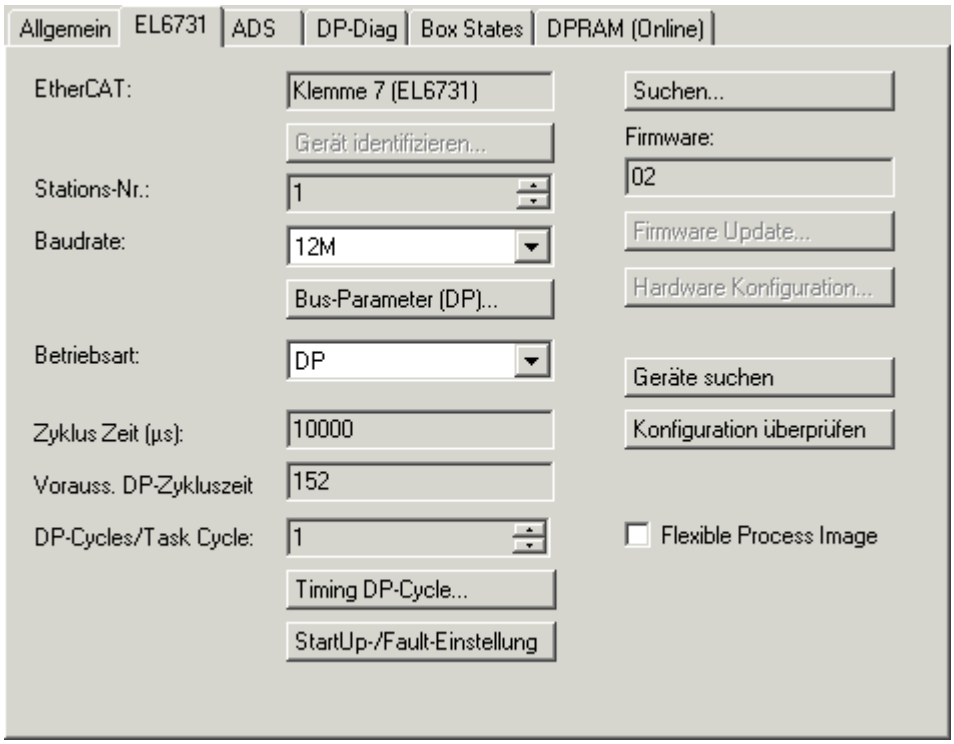


Abb. 84: Karteireiter EL6731

EtherCAT
Bezeichnung der Klemme im Klemmenverbund.

Suchen...
Hierüber werden alle EL6731-Kanäle gesucht, und es kann der gewünschte ausgewählt werden.

Stations-Nr.
Jeder Profibusteilnehmer benötigt eine eindeutige Stations-Nr. - auch der Master.

Baudrate
Hier wird die Profibus Baudrate eingestellt.

Bus-Parameter (DP)...:
Hier wird der Bus-Parameter-Dialog [▶ 95] angewählt.

Betriebsart
Bei allen drei Betriebsarten übernimmt die höchstpriore Task, die mit dem entsprechendem Gerät verknüpft ist, die Ansteuerung des PROFIBUS-Zyklus und ist somit mit dem DP-Zyklus synchronisiert (s. Kapitel Synchronisierung [▶ 49]). Wenn diese Task gestoppt wird oder auf einen Breakpoint läuft, schaltet die EL6731 in den CLEAR-Mode (Ausgänge gehen bei den Slaves auf 0 oder auf sichere Werte) (s. Kapitel Fehlerreaktionen [▶ 123]). Alle anderen Tasks werden asynchron über entsprechende Puffer bedient. Wenn eine dieser Tasks gestoppt wird oder auf einen Breakpoint läuft, kommt in der Regel im System Manager die Meldung, dass der Watchdog des entsprechenden asynchronen Mappings zugeschlagen hat und die entsprechenden Ausgänge werden auf 0 gesetzt. Bei allen Betriebsarten kann je Slave eine Poll-Rate eingestellt werden (im Karteireiter Features [▶ 105] der Box). Die Reihenfolge der Slaves im PROFIBUS-Zyklus entspricht der Reihenfolge, in der sie in dem Baum unter dem FC310x/EL6731-Device hängen. Die Betriebsart "DP" ist für den Standard-DP-Betrieb, die Betriebsarten "DP/MC (Equidistant)" und "Equidistant (no GC)" sind im Kapitel PROFIBUS-MC [▶ 48] beschrieben.

Zyklus Zeit
Anzeige der Zykluszeit der zugehörigen höchstprioren Task.

Vorauss. DP-Zukluszeit
Anzeige der voraussichtlichen PROFIBUS-Zykluszeit.

DP-Cycles/Task-Cycle [nur FC310x]
Hier können mehrere DP-Zyklen bei einem Task-Zyklus eingestellt werden, um möglichst neue Eingänge zu bekommen (s. Kapitel Slave-Priorisierung/mehrere DP-Zyklen [▶ 137]).

Timing DP-Cycle...
Hier wird das Timing des DP-Zyklus angezeigt, das ist insbesondere bei der Verwendung der Slave-Priorisierung [▶ 137] von Vorteil.

StartUp-/Fault-Einstellungen...

Hier wird der Fault-Settings-Dialog [▶ 97] angewählt

Firmware
Hier wird die aktuelle Firmware-Version der EL6731 angezeigt.

Geräte suchen...
Hiermit wird der PROFIBUS gescannt und alle gefunden Geräte werden der EL6731 hinzugefügt. Bei Beckhoff-Boxen wird die Konfiguration genau ausgelesen, bei Fremdgeräten wird versucht, die zugehörige GSD-Datei zu finden.

Konfiguration überprüfen... [nur FC310x]
Hiermit wird der PROFIBUS gescannt und mit den aktuell angefügten Boxen verglichen, Änderungen werden angezeigt.

5.1.6.2.2 Dialog Bus-Parameter

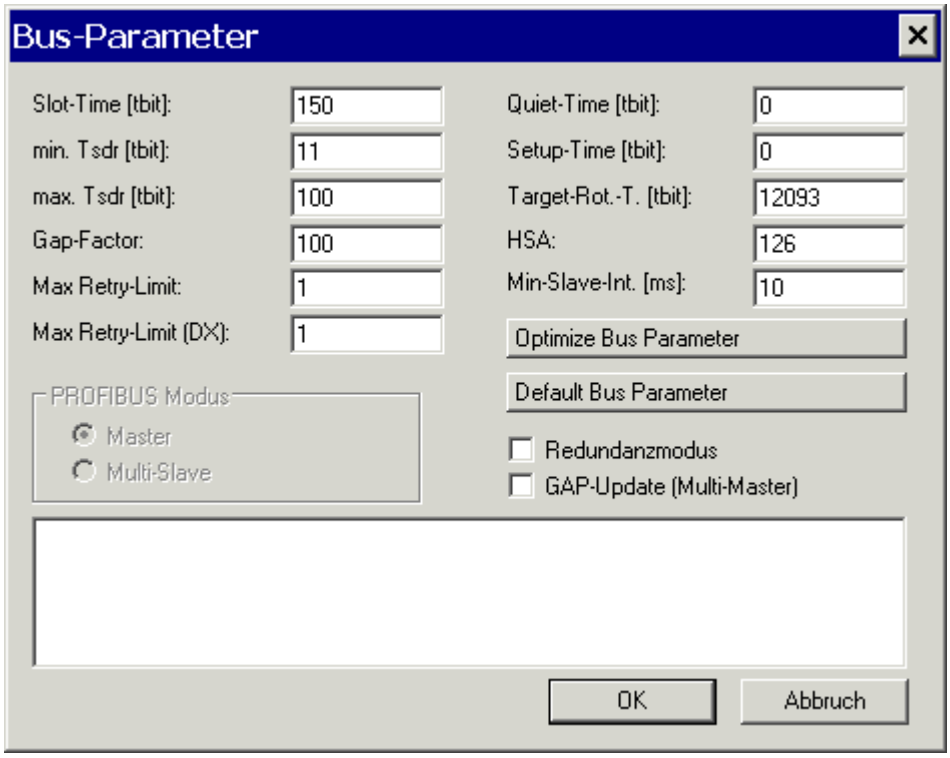


Abb. 85: Dialog Bus-Parameter

Slot-Time
Die Slot-Time gibt an, wie lange der DP-Master auf eine Antwort des DP-Slaves wartet, bevor er eine Wiederholung oder das nächste Telegramm sendet.

min. Tsdr
Die min. Tsdr gibt an, wie lange der DP-Slave mindestens mit einer Antwort wartet. Diese Zeit wird bei allen DP-Slaves während des DP-Start-Ups eingestellt (Wertebereich 11-255 Bit-Zeiten). Die min. Tsdr muss kleiner als die max. Tsdr sein.

max. Tsdr
Die max. Tsdr gibt an, wie lange der DP-Slave höchstens mit einer Antwort warten darf. Diese Zeit wird entsprechend der GSD-Datei-Einträge der DP-Slaves eingestellt. Die max. Tsdr muss kleiner als die Slot-Time sein.

GAP-Factor
Der GAP-Factor bestimmt, wie häufig das GAP-Update durchgeführt wird (falls es aktiviert ist), der Abstand zwischen zwei GAP-Update-Zyklen ist **Gap-Factor x Target-Rot.-T**.

Max-Retry-Limit
Das Max-Retry-Limit gibt an, wie oft ein Telegramm zu wiederholen ist, wenn der adressierte Teilnehmer nicht antwortet. Hier sollte minimal ein Wert von 1 eingestellt werden, damit bei azyklischen Telegrammen im Fehlerfall mindestens einmal wiederholt wird (s. Kapitel Fehlerreaktionen [▶ 123]).

Max-Retry-Limit (DX)
Da das Data_Exchange-Telegramm zyklisch wiederholt wird, könnte man für die Wiederholung des Data_Exchange-Telegrammes hier den Wert 0 eintragen, um im Equidistant-Betrieb den Zyklus auch bei einem nicht antwortenden Teilnehmer einigermaßen konstant zu halten. In diesem Fall sollte aber sinnigerweise auf dem Karteireiter Features [▶ 105] der Box eingestellt werden, dass ein Nicht-Antworten des Slaves nicht zum Verlassen des DATA-EXCH führt. Dass ein Teilnehmer nicht geantwortet hat, kann man immer am DpState [▶ 129] erkennen, der dann für einen Zyklus auf ungleich 0 wäre (s. Kapitel Fehlerreaktionen [▶ 123]).

HSA
Die HSA bestimmt die höchste aktive Adresse, bis zu der das GAP-Update durchgeführt wird (falls es aktiviert ist).

Min. Slave-Int.
Das MinSlaveIntervall gibt an mit welcher minimalen Zykluszeit die DP-StartUp-Telegramme an die DP-Slaves gesendet werden (wird aus den Einstellungen in der GSD-Datei bestimmt).

Redundanz-Modus
Hier kann beim DP-Master der Redundancy-Mode eingestellt werden, in dem Fall hört er nur auf dem Bus mit (s. Kapitel Master-Redundanz).

GAP-Update
Das GAP-Update, mit dem alle Stationen bis zur HSA ab und zu angefragt werden, ob sie vorhanden sind, kann ein- oder ausgeschaltet werden. Das GAP-Update ist eigentlich nur bei Multi-Master-Betrieb von Bedeutung, bei Single-Master-Betrieb hat es zur Folge, dass der Jitter des PROFIBUS-Zyklus größer ist, daher ist es standardmäßig ausgeschaltet.

Optimize Bus-Parameter
Hiermit werden die optimierten Bus-Parameter eingestellt.

Default Bus-Parameter
Hiermit werden die Default-Bus-Parameter eingestellt.

5.1.6.2.3 Dialog Start-Up/Fault-Einstellungen

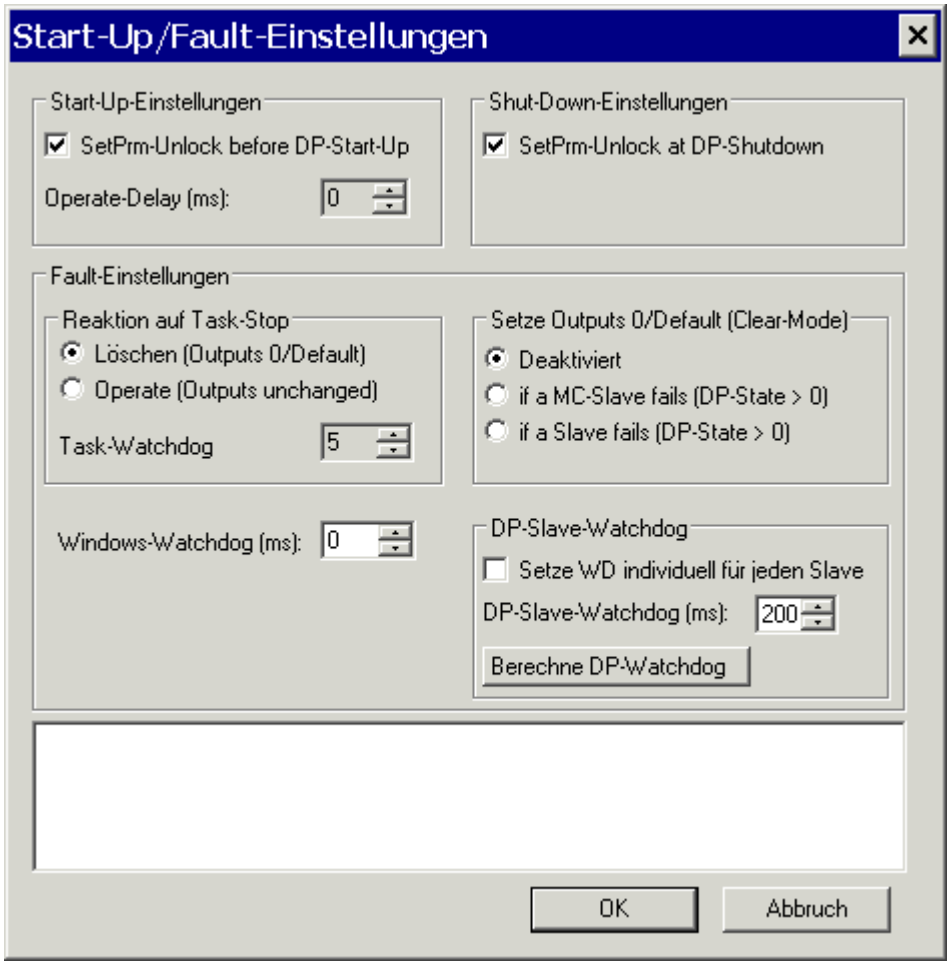


Abb. 86: Dialog Start-Up/Fault-Einstellungen

SetPrm-Unlock before DP-Start-Up
Normalerweise führt der DP-Master beim DP-Start-Up einen Verbindungsabbau der zyklischen Verbindung durch, damit der DP-Slave immer mitbekommt, dass der DP-Master neu gestartet wurde. Im Redundancy-Mode kann es aber gewünscht sein, dass der DP-Slave das gerade nicht mitbekommt, da das Umschalten vom Primary- auf den Redundancy-Master rückwirkungsfrei für den DP-Slave erfolgen soll (s. Kapitel Master-Redundanz).

SetPrm-Unlock at DP-Shutdown
Normalerweise führt der DP-Master beim DP-Shutdown einen Verbindungsabbau der zyklischen Verbindung durch, damit der DP-Slave immer mitbekommt, dass der DP-Master gestoppt wurde. Im Redundancy-Mode kann es aber gewünscht sein, dass der DP-Slave das gerade nicht mitbekommt, da das Umschalten vom Primary- auf den Redundancy-Master rückwirkungsfrei für den DP-Slave erfolgen soll (s. Kapitel Master-Redundanz).

Operate-Delay
Der DP-Master wechselt automatisch bei Beachtung des Auto-Clear-Modes in Operate-Zustand, wenn die Task gestartet wurde. Der Übergang von Clear nach Operate kann mit der Operate-Delay-Zeit verzögert werden. Im Zustand Clear werden alle Ausgänge auf 0 (DP-Slave unterstützt kein Fail_Safe) bzw. auf den Fail_Safe-Wert (DP-Slave unterstützt Fail_Safe) gesetzt, im Zustand Operate haben die Ausgänge den von der Task vorgegebenen Wert.

Reaktion auf Task-STOP
Hier kann eingestellt werden, ob der DP-Master beim PLC-STOP/-Breakpoint die Ausgänge auf 0 setzen und unverändert lassen soll (s. Kapitel Fehlerreaktionen [▶ 123]).

Task-Watchdog (EtherCAT Watchdog)
Der DP-Master wechselt automatisch in den Clear-Mode (Ausgänge der Slaves werden auf 0 bzw. Fail_Safe-Werte gesetzt), wenn er keinen Interrupt mehr von der ihm zugehörigen Task bekommt (z. B.

PLC-Breakpoint oder -Absturz). Hier kann angegeben werden, wie viele fehlende Task-Zyklen toleriert werden, bevor der Master in den Clear-Mode schaltet. Diese Einstellung ist unabhängig von der Einstellung des **Clear-Modes**.

Clear-Mode
Hier kann eingestellt werden, dass der Master in den Zustand "Clear" wechselt bzw. bleibt, solange mindestens ein MC-Slave (Einstellung "if a MC-Slave fails") bzw. irgendein Slave (Einstellung "if a Slave fails"") nicht korrekt antwortet (einen DpState [▶ 129] ungleich 0 hat) (s. Kapitel Fehlerreaktionen [▶ 123]).

Windows-Watchdog (nur FC310x)
Nicht relevant für EL6731.

Set WD individually for each Slave
Hier angewählt werden, ob der WD individuell für jeden Slave eingestellt werden soll (auf dem Karteireiter Profibus [▶ 104] der Box).

DP-Watchdog-Time
Wenn die Check-Box "Set WD individuel für jeden Slave" nicht angeklickt ist, kann hier der DP-Watchdog auf einen einheitlichen Wert für alle Slaves eingestellt werden.

Calculate DP-Slave Watchdog Time
Hier wird die DP-Watchdog-Time aller DP-Slaves auf einen geeigneten Wert gestellt.

5.1.6.2.4 Karteireiter MC

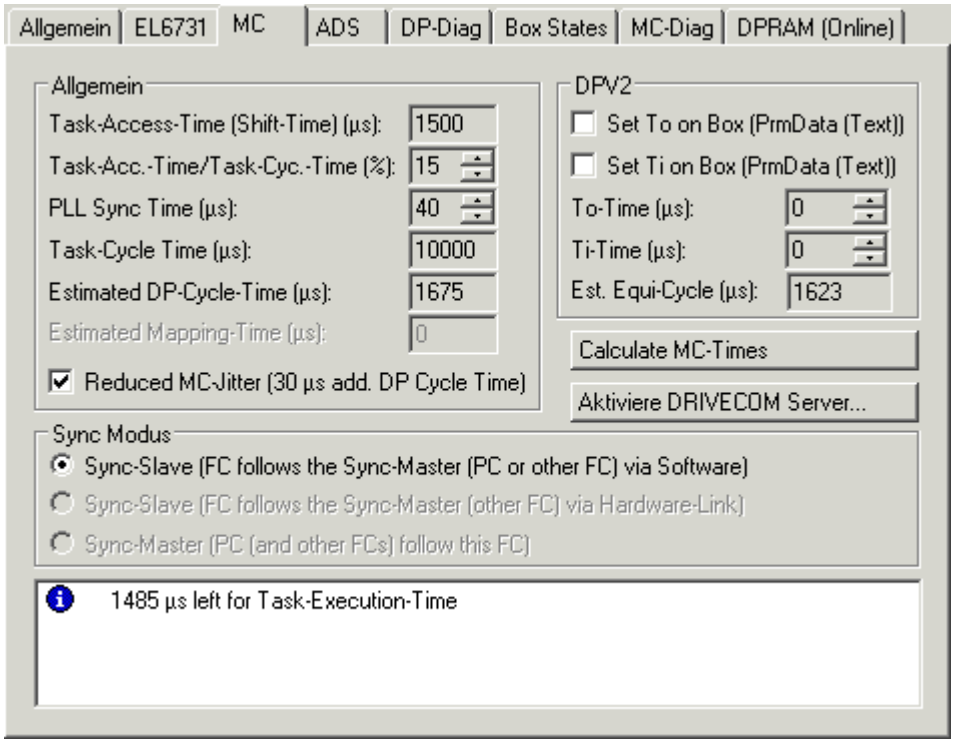


Abb. 87: Karteireiter MC

Task-Access-Time (Shift-Time)
Nicht relevant für EL6731.

Task-Access-Time/Task Cycle-Time
Nicht relevant für EL6731.

PLL-Sync-Time
Nicht relevant für EL6731.

Task Cycle Time
Hier wird die Zykluszeit der zugehörigen höchstprioren Task angezeigt.

Estimated DP-Cycle Tim
Hier wird die voraussichtliche PROFIBUS-Zykluszeit angezeigt.

Set To on Box
Die To-Time kann für jeden Slave individuell auf dem Karteireiter Prm Data(Text) [▶ 108] der Box eingestellt werden.

Set Ti on Box
Die Ti-Time kann für jeden Slave individuell auf dem Karteireiter Prm Data(Text) [▶ 108] der Box eingestellt werden.

To-Time
Wenn die Check-Box **Set To on Box** nicht aktiviert ist, kann die To-Time für alle Slaves einheitlich eingestellt werden.

Ti-Time
Wenn die Check-Box **Set Ti on Box** nicht aktiviert ist, kann die Ti-Time für alle Slaves einheitlich eingestellt werden.

Estimated Equi-Cycle Time
Hier wird der DPV2-Teil der voraussichtliche PROFIBUS-Zykluszeit angezeigt.

Reduced MC-Jitter (bei FC/CX optional, bei EL standardmäßig angewählt)
Der Zyklus der FC/CX wird durch einen lokalen Timer gesteuert, der mit TwinCAT synchronisiert ist. Um die Interruptlatenzzeit auszugleichen (Jitter des DP-Zyklus ca. 960 ns ohne Ausgleich, ca. 320 ns mit Ausgleich), wartet die Interruptroutine immer auf einen bestimmten Timer-Wert, bevor der DP-Zyklus gestartet wird. Durch das Warten wird der Zyklus ca. 30 µs länger.

Calculate MC-Times
Mit diesem Buttom werden alle DPV2-Zeiten automatisch berechnet.

Sync-Mode
Nicht relevant für EL6731; Synchronisierung wird durch die Distributed clocks gesteuert und entspricht daher der Einstellung Sync-Master bei der FC310x.

Aktiviere DRIVECOM Server
Es wird ein XML-File für den DRIVECOM-Server (OPC-Server) generiert, über den z. B. das SimoCon U-Tool über PROFIBUS auf die angeschlossenen 611U-Antriebe zugreifen kann.

5.1.6.2.5 Karteireiter ADS

Die EL6731 ist ein ADS-Device mit einer eigenen Net-ID, die hier verändert werden kann. Alle ADS-Dienste (Diagnose, azyklische Kommunikation), die an die EL6731 gehen, müssen diese Net-ID adressieren.

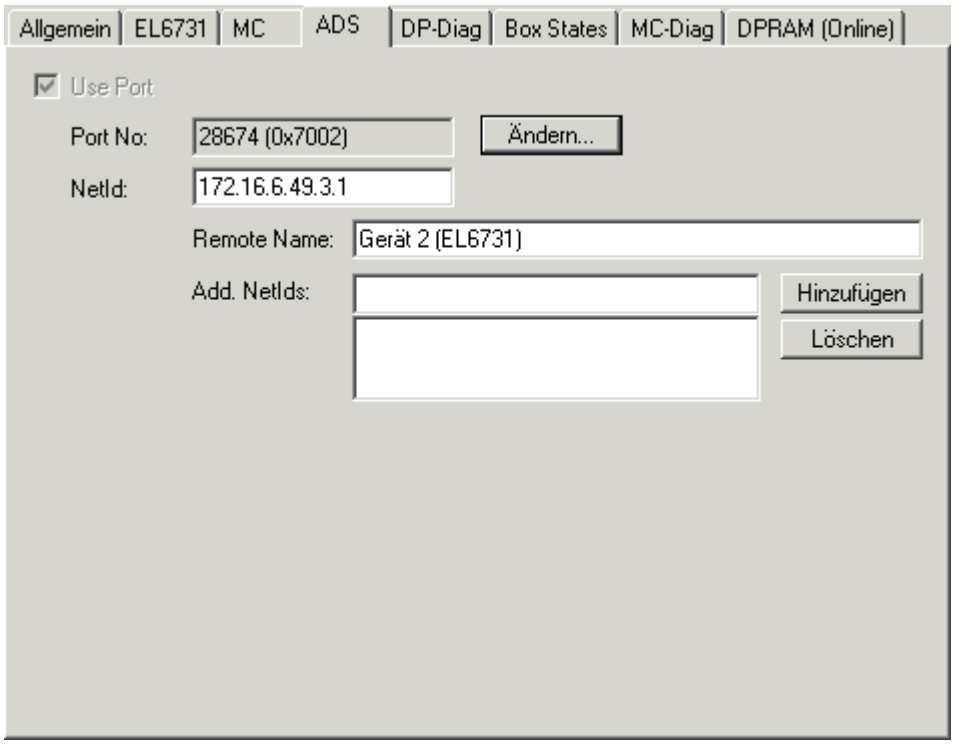


Abb. 88: Karteireiter ADS

5.1.6.2.6 Karteireiter DP-Diag

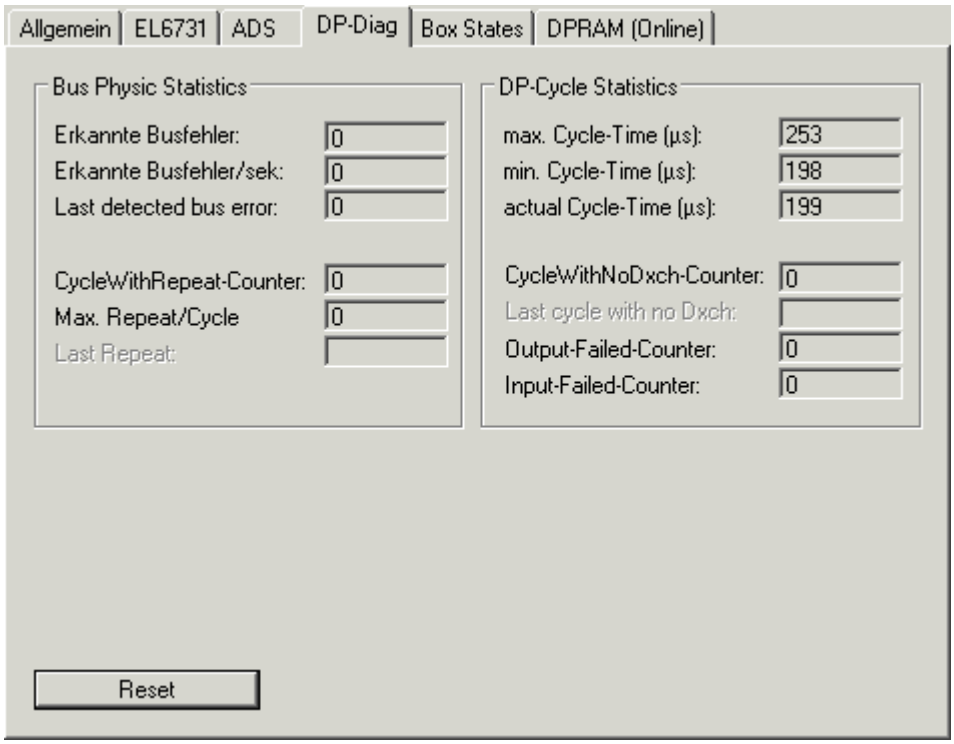


Abb. 89: Karteireiter DP-Diag

Hier werden Busverkablungsprobleme und DP-Zykluszeiten angezeigt:

Erkannte Busfehler
Hier wird die Anzahl der festgestellten Busfehler angezeigt, wenn dieser Zähler ungleich 0 ist, sollte die Verkabelung geprüft werden (falls keine PROFIBUS-Stecker abgezogen oder gesteckt wurden (in der Regel gibt es beim Abziehen oder Stecken von PROFIBUS-Steckern auch kurze Busstörungen)).

CycleWithRepeatCounter
Hier wird die Anzahl der PROFIBUS-Zyklen angezeigt, in denen mindestens einmal ein Telegramm wiederholt wurde. Wiederholungen sind auch ein Anzeichen dafür, dass die Busphysik nicht in Ordnung ist.

Max. Repeat/Cycle
Hier die maximale Anzahl von Wiederholungen innerhalb eines Zyklus angezeigt.

min./max./actual Cycle-Time
Hier werden minimale, maximale und aktuelle DP-Zykluszeit angezeigt, wobei nur Zyklen berücksichtigt werden, in denen alle Slaves im Datenaustausch waren und keine Wiederholungen auftraten.

CycleWithNoDxch-Counter
Zählt hoch, wenn nicht alle Slaves im Datenaustausch sind (d.h. einen DpState ungleich 0 haben)

Output-Failed-Cycle-Counter
Zählt hoch, wenn der DP-Zyklus nicht fertig wurde, bevor der nächste EtherCAT-Zyklus (EL) gestartet ist und alle Slaves im Datenaustausch sind (d.h. einen DpState gleich 0 haben)

5.1.6.2.7 Karteireiter Box States

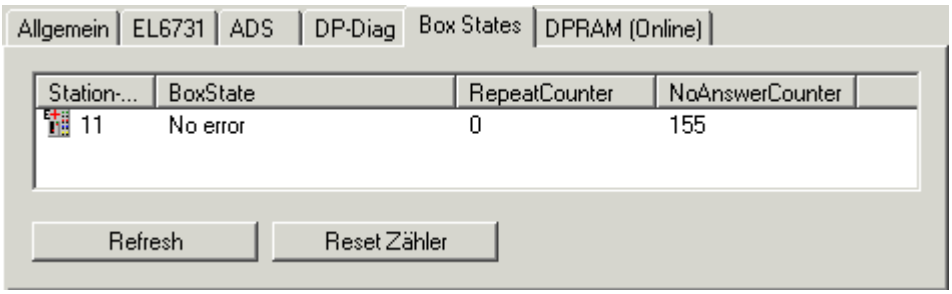


Abb. 90: Karteireiter Box States

Hier wird eine Übersicht aller aktuellen Box-States, der Repeat-Counter (zählt bei jeder Telegrammwiederholung zu dem Slave hoch) und der NoAnswer-Counter (zählt jedes Mal hoch, wenn der Slave nicht geantwortet hat) angezeigt.

5.1.6.2.8 Karteireiter MC-Diag

Es gibt den Karteireiter "MC-Diag", auf dem Online verschiedene Equidistant-Überwachungsparameter angezeigt werden:

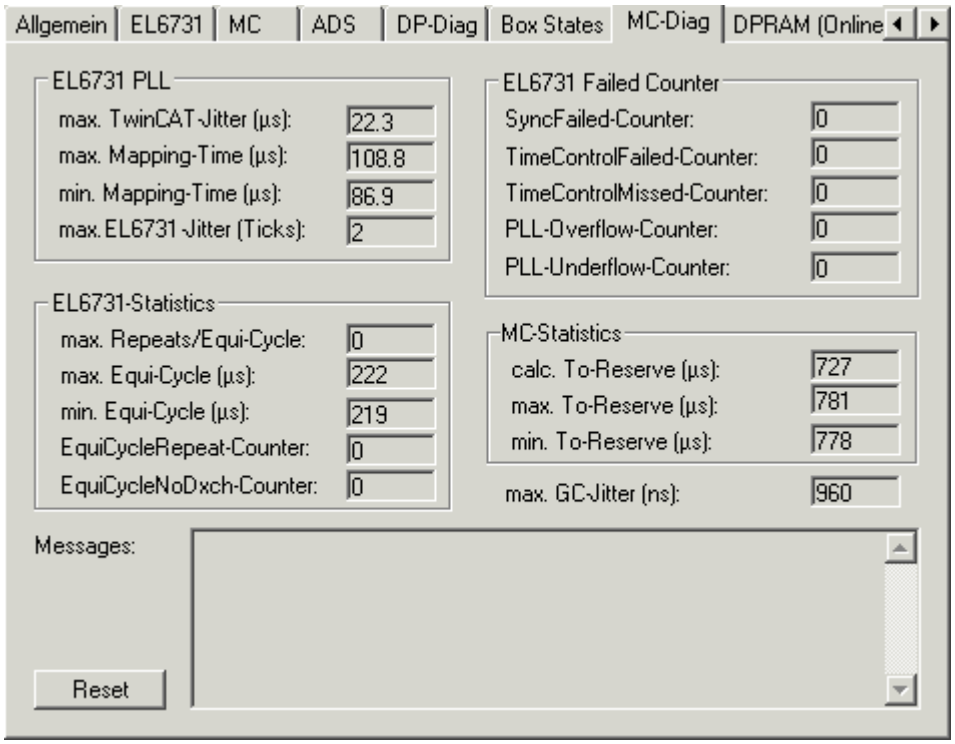


Abb. 91: Karteireiter MC-Diag

EL6731 PLL

Nicht mehr relevant für EL6731.

EL6731-Statistics

Da die PROFIBUS MC-Slaves immer zuerst an die EL6731 angefügt werden sollten, werden diese auch immer vor den DP-Slaves im DP-Zyklus angesprochen, dieser Teil des DP-Zyklus wird im Folgenden als Equi-Cycle bezeichnet. Wenn der Equi-Cycle größer als die To-Zeit der MC-Slaves wird, bekommen die letzten MC-Slaves an der EL6731 in der Regel einen Synchronisationsfehler (Fehler 597 oder 598 bei Simodrive 611U).

max. Repeats/Equi-Cycle

Hier wird angezeigt, wie viele Data_Exchange-Telegramme während eines Equi-Cycles maximal wiederholt wurden, dadurch verlängert sich der Equi-Cycle, normalerweise sollten keine Wiederholungen auftreten (es sein denn, dass Busstecker gezogen oder ein MC-Slave ausgeschaltet wurde)

min./max. Equi-Cycle
Hier wird die minimale bzw. die maximale Equi-Cycle-Time angezeigt.

Equi-Cycle-Repeat-Counter
Hier wird angezeigt, wie oft im Equi-Cycle Telegramm-Wiederholungen auftraten.

Equi-Cycle-NoDxch-Counter

Hier wird angezeigt, wie oft im Equi-Cycle nicht alle MC-Slaves im Datenaustausch waren.

EL6731 Failed Counter

Sync-Failed-Counter
Dieser Zähler zählt hoch, wenn TwinCAT-Task und DP-Zyklus nicht miteinander synchronisiert sind. Das kann beim Hochlauf des TwinCAT Systems passieren, danach sollte dieser Zähler nicht hochzählen. Falls die zugehörige (NC-)Task nicht die höchste Priorität hat, kann dieser Zähler auch hochzählen, das sollte vermieden werden.

Time-Control-Failed-Counter
Dieser Zähler zählt hoch, wenn zum Zeitpunkt des DP-Zyklus-Starts der PROFIBUS nicht frei war. Ursache dafür können Busstörungen, nicht vorhandenen Teilnehmer, ein zweiter Master oder eine zu kleine Safety-Time sein.

PLL-Overflow-/-Underflow-Counter
Nicht relevant für EL6731.

MC-Statistics

Mit der To-Zeit wird bei jedem MC-Slave festgelegt, wann, bezogen auf den DP-Zyklusstart, dieser die vom Master empfangenen Outputs übernehmen soll. Die MC-Slaves können miteinander synchronisiert werden, wenn bei allen MC-Slaves der gleiche Wert für To eingestellt wird. Dieser muss aber mindestens so groß wie die Equi-Cycle-Time plus eine Sicherheitsreserve von ca. 200 µs sein. Die To-Zeit wird bei allen MC-Slaves mit dem Button "Calc. Equi-Times" (s.o.) berechnet.

calc. To-Reserve: Hier steht die kalkulierte To-Reserve (To-Zeit - Equi-Cycle-Time)

min./max. To-Reserve: Hier wird die min. bzw. max. To-Reserve gemessen.

max. GC-Jitter

Hier wird der maximale Jitter des DP-Zyklus gemessen (GC für Global-Control-Telegramm, das immer am Anfang eines Zyklus gesendet wird). Beim Hochlauf kann der Jitter etwas größer sein, im eingeschwungenen Zustand sollte er 1 µs (bei Sync-Mode "Sync-Master") bzw. 2 µs (bei Sync-Mode "Disabled") nicht überschreiten.

5.1.6.3 TwinCAT Baum "Box"

5.1.6.3.1 Karteireiter Profibus

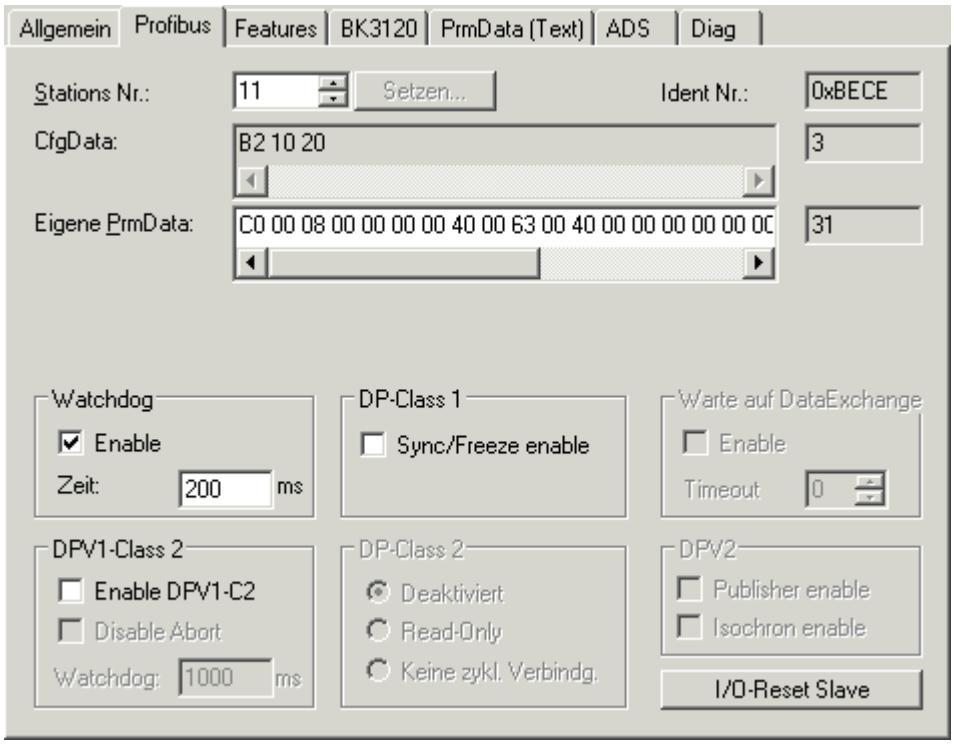


Abb. 92: Karteireiter Profibus

Stations-Nr.
Hier ist die auf dem jeweiligen Slave eingestellte PROFIBUS-Stationsadresse einzustellen. Es gibt Slaves, bei denen die Stationsadresse nicht hardwaremäßig eingestellt werden kann, sondern nur über den SetSlaveAddress-Dienst. In dem Fall ist der Button "Set..." zu drücken, woraufhin ein Dialog aufgeht, mit dem das Senden eines SetSlaveAddress-Telegramms ausgelöst werden kann.

Watchdog
Schaltet den DP-Watchdog ein. Wenn der Slave bei eingeschaltetem Watchdog für die Dauer der Watchdog-Zeit kein DP-Telegramm empfängt, verlässt er automatisch den Datenaustausch. Die minimal einzustellende Watchdog-Zeit hängt von der DP-Zykluszeit ab und sollte größer als der nach der folgenden Formel berechnete Wert sein: **Estimated-Cycle-Time x 10**

Für besonders kritische Ausgänge kann bei DP-Slaves, die eine Watchdog-Base-Time von 1 ms unterstützen (alle Beckhoff-Slaves mit Ausnahme von BK3000 und BK3100 sowie alle Fremd-Geräte in deren GSD-Datei der Eintrag "WD_Base_1ms_supp = 1" steht) ein DP-Watchdog herunter bis zu 2 ms eingestellt werden, allerdings sollte die DP-Watchdog-Time mindestens doppelt so groß wie das Maximum aus **Cycle-Time** und **Estimated Cycle-Time** sein (s. Karteireiter des Masters).

Ident-Nr.
Hier wird die Ident-Nummer aus der GSD-Datei angezeigt

Eigene PrmData
Erlaubt die Profibus-spezifischen Parameterdaten zu editieren, die Größe der aktuellen Parameterdaten wird ebenfalls angezeigt. Die PrmData können aber in der Regel textuell (-> PrmData (Text)) oder bei Beckhoff-DP-Slaves teilweise über den Karteireiter "Beckhoff" eingestellt werden

CfgData
Die aktuellen Konfigurationsdaten (ergeben sich aus den angefügten Modulen bzw. Klemmen) sowie deren Länge werden angezeigt.

Sync/Freeze
In der Betriebsart DP/MC (Equidistant) des Masterskönnen Slaves mit Sync und Freeze [▶ 50] betrieben werden.

DPV1-Class 2
Bei der FC310x/EL6731 kann eine DPV1-Klasse 2-Verbindung zu einem DPV1-Slave aktiviert werden. Mit dem Parameter Timeout wird die Verbindungsüberwachungszeit der Klasse 2-Verbindung eingestellt (s. KapitelDPV1 [▶ 54]).

DP Class 2
Wenn der DP-Slave mit einem anderen Master im Datenaustausch ist, aber trotzdem von TwinCAT azyklisch angesprochen werden soll, ist unter DP-Class 2 "Keine zykl. Verbindung" auszuwählen. Die Einstellung "ReadOnly" wird bisher nicht unterstützt.

I/O-ResetSlave
Mit diesem Button kann bei gestartetem TwinCAT der zyklische Datenaustausch zu dem DP-Slave abgebaut und gleich wieder aufgebaut werden (entspricht einem IO-Reset aber nur für den einen Slave).

5.1.6.3.2 Karteireiter Features

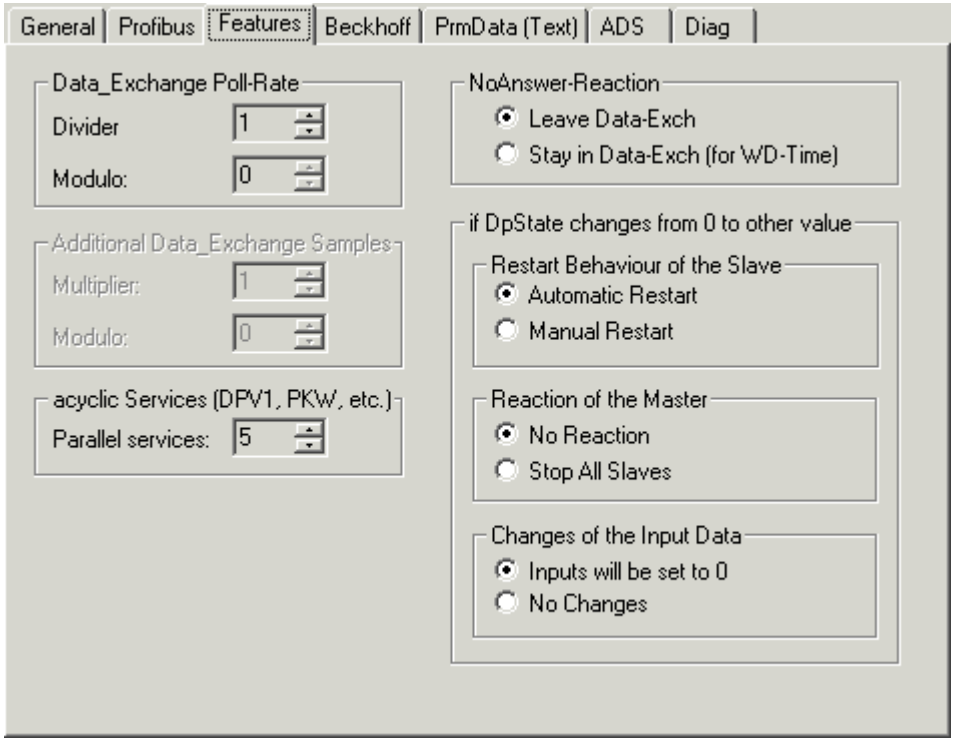


Abb. 93: Karteireiter Features

Data_Exchange Poll-Rate [nur FC310x]
Je Slave kann eine unterschiedliche Polling-Rate (Divider) eingestellt werden, Divider 1 bedeutet, dass der Slave jeden Zyklus gepollt wird, Divider 2 bedeutet jeden 2. Zyklus, usw. Mit dem Modulo können Slaves mit Divider größer 1 auf unterschiedliche Zyklen aufgeteilt werden, um die maximale Zykluszeit zu verkürzen (Divider 2 und Modulo 0 bedeutet, dass der Slave in jedem geraden Zyklus gepollt wird, Divider 2 und Modulo 1 bedeutet, dass der Slave in jedem ungeraden Zyklus gepollt wird) (s. Kapitel Slave-Priorisierung/ mehrere DP-Zyklen [▶ 137]).

Additional Data_Exchange Samples
Es besteht die Möglichkeit, innerhalb eines Task-Zyklus mehrere DP-Zyklen zu fahren. Optional kann jeder Slave dann noch mit unterschiedlichen Outputdaten je DP-Zyklus versorgt und die Inputdaten eines jeden DP-Zyklus an die Steuerung zu übergeben. In diesem Fall gibt es je DP-Zyklus einen eigenen Variablensatz (s. Kapitel Slave-Priorisierung/mehrere DP-Zyklen [▶ 137]).

NoAnswer-Reaction
Je Slave kann angegeben werden, ob dieser weiterhin in Data-Exch bleiben soll, auch wenn er nicht oder fehlerhaft antwortet. In diesem Fall (Stay in Data-Exch) wird der Datenaustausch erst verlassen, wenn der Slave innerhalb der Ansprechüberwachungszeit nie korrekt geantwortet hat (falls der **Watchdog** (s. Karteireiter Profibus [▶ 104] der Box) aktiviert ist, andernfalls wird der Datenaustausch erst beendet, wenn der Slave 65535 mal nicht korrekt geantwortet hat) (s. Kapitel Fehlerreaktionen [▶ 123]).

Restart-Behaviour
Je Slave kann angegeben werden, ob dieser nach einem Verlassen von Data-Exch automatisch wieder hochlaufen oder im Zustand Wait-Prm bleiben soll (s. Kapitel Fehlerreaktionen [▶ 123]).

Reaction of the Master
Je Slave kann angegeben werden, ob dessen Verlassen von Data-Exch zum Stop des PROFIBUS-Zyklus führen soll (FC310x: alle Slaves verlassen den Datenaustausch und gehen nach Wait-Prm, Wiederanlauf nur durch IO-Reset oder Restart des TwinCAT Systems; EL6731: EL geht in den Status PREOP, Wiederanlauf durch Schalten nach Status OP) (s. Kapitel Fehlerreaktionen [▶ 123]).

Changes of the Input Data
Je Slave kann angegeben werden, ob beim Verlassen von Data-Exch (DpState wird ungleich 0) dessen Inputdaten auf 0 gesetzt oder unverändert bleiben sollen (s. Kapitel Fehlerreaktionen [▶ 123]).

acyclic Services
Die Anzahl der parallelen ADS-Dienste zu einer Box kann hier eingestellt werden.

5.1.6.3.3 Karteireiter <Beckhoff>

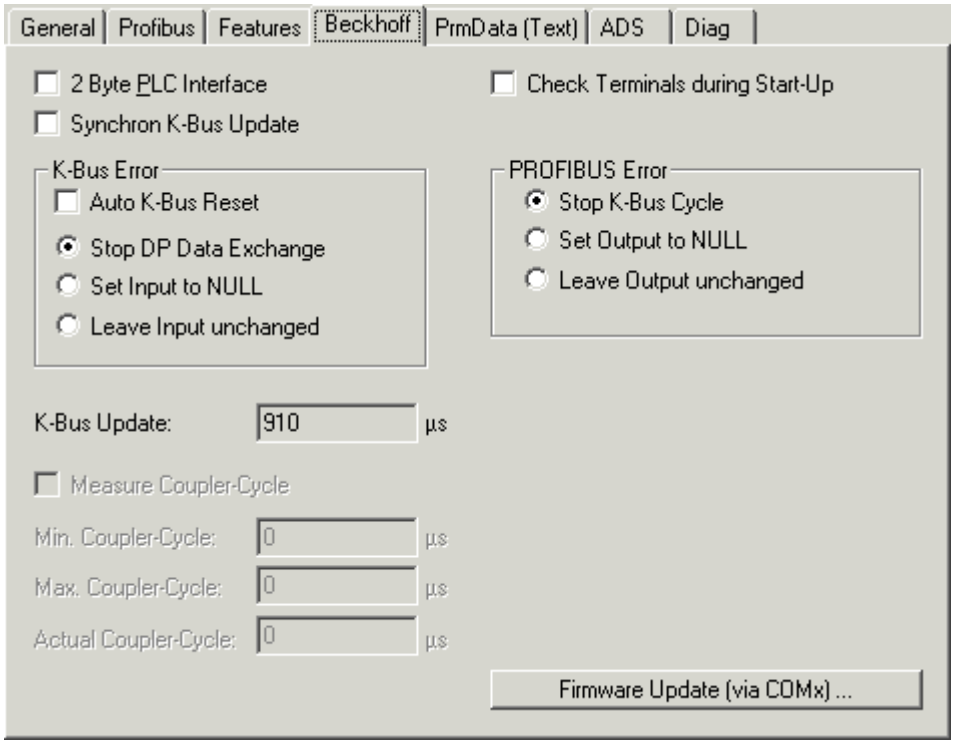


Abb. 94: Karteireiter <Beckhoff>

Firmware Update
Mit diesem Button kann die Firmware eines Beckhoff-DP-Slaves per KS2000-Kabel über die serielle Schnittstelle aktualisiert werden

2 Byte PLC Interface
Schaltet das 2-Byte SPS Interface des Beckhoff-DP-Slaves ein.

Synchron K-Bus-Update
Hier ist bei Buskopplern die erwartete interne Zykluszeit (K-Bus-Zyklus + Übergabe der DP-Puffer angegeben).

Check Terminals during Start-Up
Wenn diese Check-Box aktiviert wird, wird die Tabelle 9 per DPV1-Write an den Koppler übertragen und der Koppler geht nur in den Datenaustausch (DpState = 0), wenn die Einträge übereinstimmen. Dadurch ist eine genauere Prüfung der Klemmen beim Hochlauf möglich, als es mit den PROFIBUS-CfgData getan wird.

K-Bus Error
Hier kann die Reaktion auf einen K-Bus-Fehler angegeben werden (automatischer oder manueller K-Bus Reset, Reaktion in den Inputdaten im Koppler)

PROFIBUS Error
Hier kann die Reaktion auf einen PROFIBUS-Fehler angegeben werden (Reaktion in den Outputdaten im Koppler)

Measure Coupler-Cycle
Hier kann die Zykluszeit auf dem Koppler (DP + K-Bus) gemessen werden.

5.1.6.3.4 Karteireiter Process Data

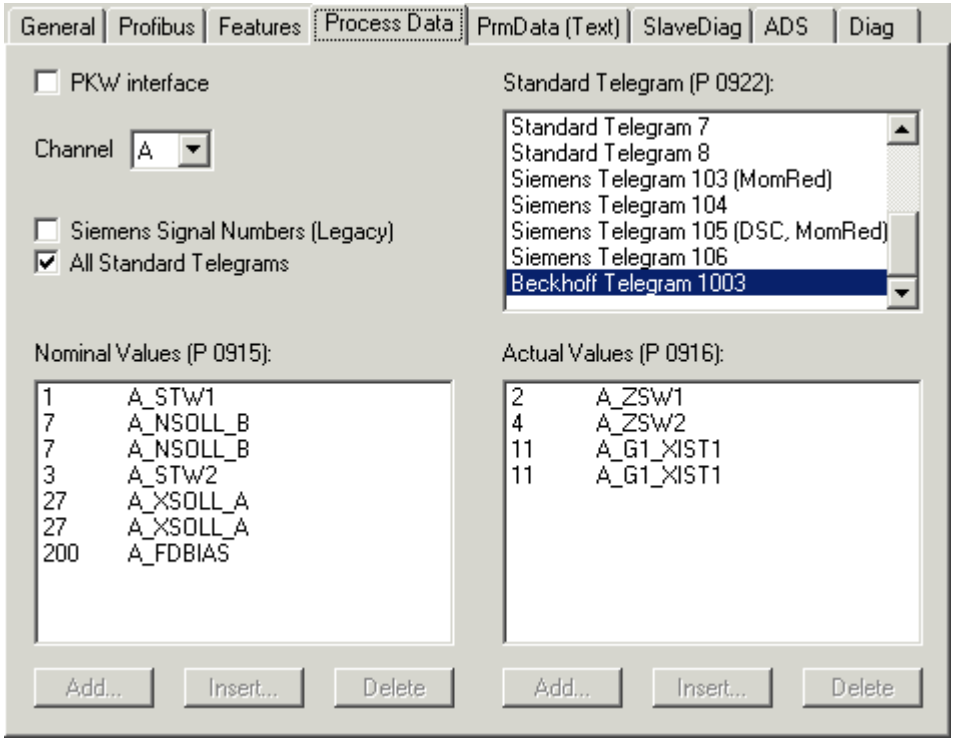


Abb. 95: Karteireiter Process Data

Unter Siemens AG bzw. Profidrive MC können die Boxen Profidrive MC und Profidrive MC (2-fach) ausgewählt werden. Standardmäßig bekommt die Box die Einstellungen für einen Simodrive 611U mit Standardtelegramm 3. Für andere PROFIBUS MC-Slaves ist die Ident-Nummer auf dem Karteireiter Profibus [▶ 104] des Slaves entsprechend zu ändern. Der gewünschte Telegrammtyp kann auf dem Karteireiter **ProcessData** eingestellt werden und muss in der Regel mit dem Parameter 922 des PROFIBUS MC-Slaves übereinstimmen, der über ein herstellerspezifisches Konfigurationstool (SimoCon U bei Simodrive 611U) eingestellt werden kann.

Auf diesem Karteireiter gibt es auch eine Check-Box "PKW-Interface", die das PKW-Interface aktiviert. darüber können online die Parameter der Box auf dem Karteireiter Online angezeigt werden (bisher funktioniert das nur mit dem Simodrive 611U, da ein Parameterfile erforderlich ist, dass der SimoCon U erzeugt). Auf jeden Fall ist es möglich, die Parameter über das PKW-Interface per ADS zu lesen und schreiben (s. Kapitel PKW-Protokoll [▶ 57]).

5.1.6.3.5 Karteireiter PrmData (Text)

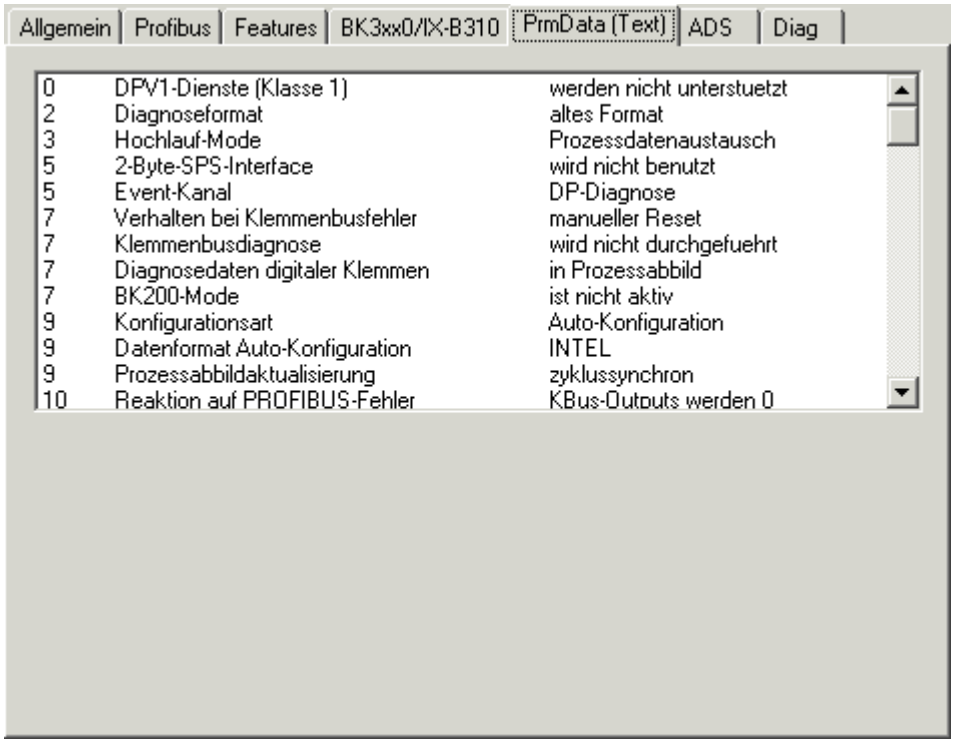


Abb. 96: Karteireiter PrmData (Text)

Durch Anklicken einer Zeile kann der aktuelle Wert verändert werden. Die Beschreibung der jeweiligen Einstellungen ist der Dokumentation des entsprechenden Herstellers zu entnehmen.

5.1.6.3.6 Karteireiter Diag

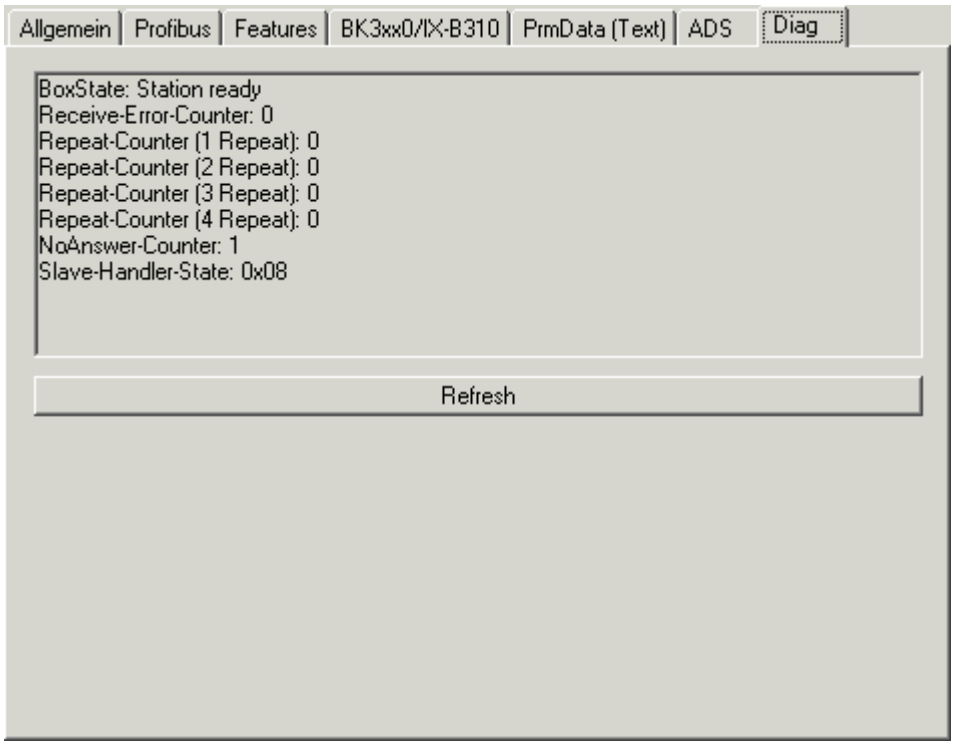


Abb. 97: Karteireiter Diag

Hier können die folgenden Informationen angezeigt werden:

BoxState
Hier wird der aktuelle DpState [▶ 129] angezeigt

Receive-Error-Counter
Anzahl der gestörten Telegramme von dem Slave

Repeat-Counter
Anzahl der notwendigen Wiederholungen aufgrund keiner oder einer gestörten Antwort des Slaves

NoAnswer-Counter
Anzahl der Telegramme, auf die der Slave nicht geantwortet hat

Letzter DPV1-Fehler
Error-Decode, Error-Class, Error-Code und Error-Code 2 (s. Beschreibung DPV1-Fehlercodes [▶ 133])

Bei Beckhoff DP-Slaves werden noch weitere Diagnoseinformationen angezeigt.

5.1.7 Betrieb von PROFIdrive MC-Antrieben

5.1.7.1 E/A-Konfiguration

Im Folgenden wird die Einbindung von PROFIdrive MC-Antrieben (hier ein Servo-Antriebsregler Sinamics S120) in die Beckhoff TwinCAT NC/PTP beschrieben.

Die Kurzanleitung beschreibt ausschließlich den Betrieb von Sychron-Servoachsen. Sie basiert auf TwinCAT NC/PTP Version 2.10 (empfohlen wird der Einsatz von TwinCAT NC/PTP 2.10 Build 1340 oder neuer). Voraussetzung zum Verständnis ist die Kenntnis von Beckhoff TwinCAT NC/PTP.

5.1.7.1.1 Feldbusmaster konfigurieren

Profibus-Feldbusmasterkarte auswählen

Das Vorgehen zum Einfügen eines Profibusgerätes (eine Feldbusmasterkarte/Feldbusmasterklemme EL6731) im TwinCAT System Manager wird als bekannt vorausgesetzt. Es geschieht bei fertiger Installation und Betriebsfähigkeit aller Busteilnehmer entweder durch "Geräte Suchen" (Scannen) im Config-Mode oder über das Kontextmenü im Baum E/A Geräte "Gerät Einfügen" -> Profibus DP. Sollten Sie nicht auf dem Zielgerät konfigurieren (bzw. ohne den Feldbusmaster an Ihrem PC angeschlossen zu haben), kann der Dialog "Gerät an Adresse gefunden" mit "Abbrechen" quittiert werden. Die Adresse der Karte kann dann später bei der Inbetriebnahme gesucht werden.

Abhängig vom gewählten Zielsystem (nur PC, CX, BX, Alle) werden mögliche Geräte angeboten. Das Beispiel bezieht sich auf eine Feldbusmasterkarte FC31xx PCI; bei Verwendung der EL6731 gilt die Beschreibung entsprechend.

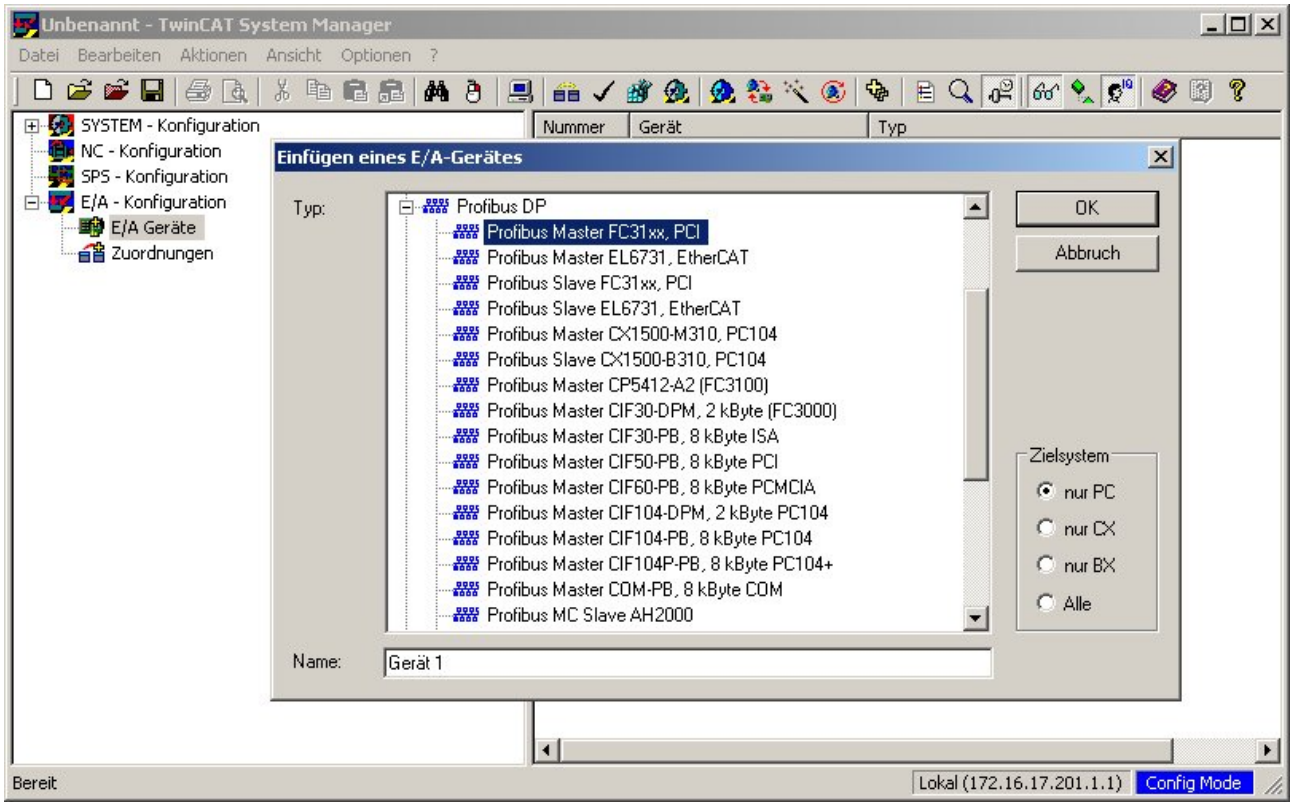


Abb. 98: Einfügen eines E/A Gerätes FC31xx PCI

Bitte wählen Sie das geeignete E/A Gerät aus (hier Profibus Master FC31xx PCI). Der Betrieb an einem Profibus Master EL6731 über EtherCAT ist mit 2 CU getestet.

Profibus DP konfigurieren

Die Profibus-Masterkarte wird über den Kartenreiter FC31xx konfiguriert. Abschließend kann dies nur im Rahmen der Inbetriebsetzung mit laufendem, funktionsfähigen Feldbus und allen konfigurierten Elementen erfolgen:

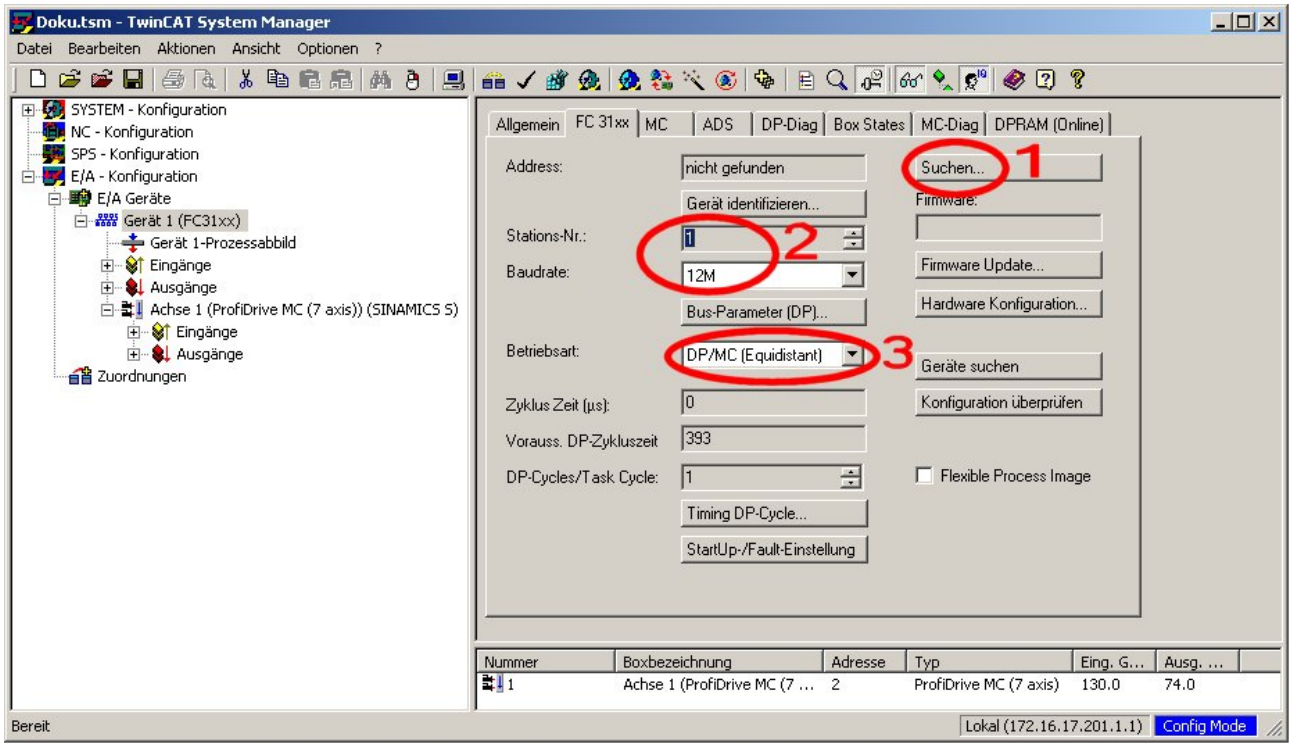


Abb. 99: Konfiguration 1

(1) Die Adresse der Karte muss später bei der Inbetriebnahme gesucht werden. Die Firmwareversion der Profibuskarte FC310x sollte mindestens 2.58, die einer FC3151 mindestens 2.5b sein.
(2) Die Stationsadresse des Feldbusmasters ist auf 1 voreingestellt. Vermeiden sie Adresskonflikte mit einem weiteren Gerät (z. B. einem Programmiergerät mit der STARTER-Software und kontrollieren Sie die Einstellung der Baudrate (12 M).
(3) Die Betriebsart muss DP/MC (Equidistant) sein.

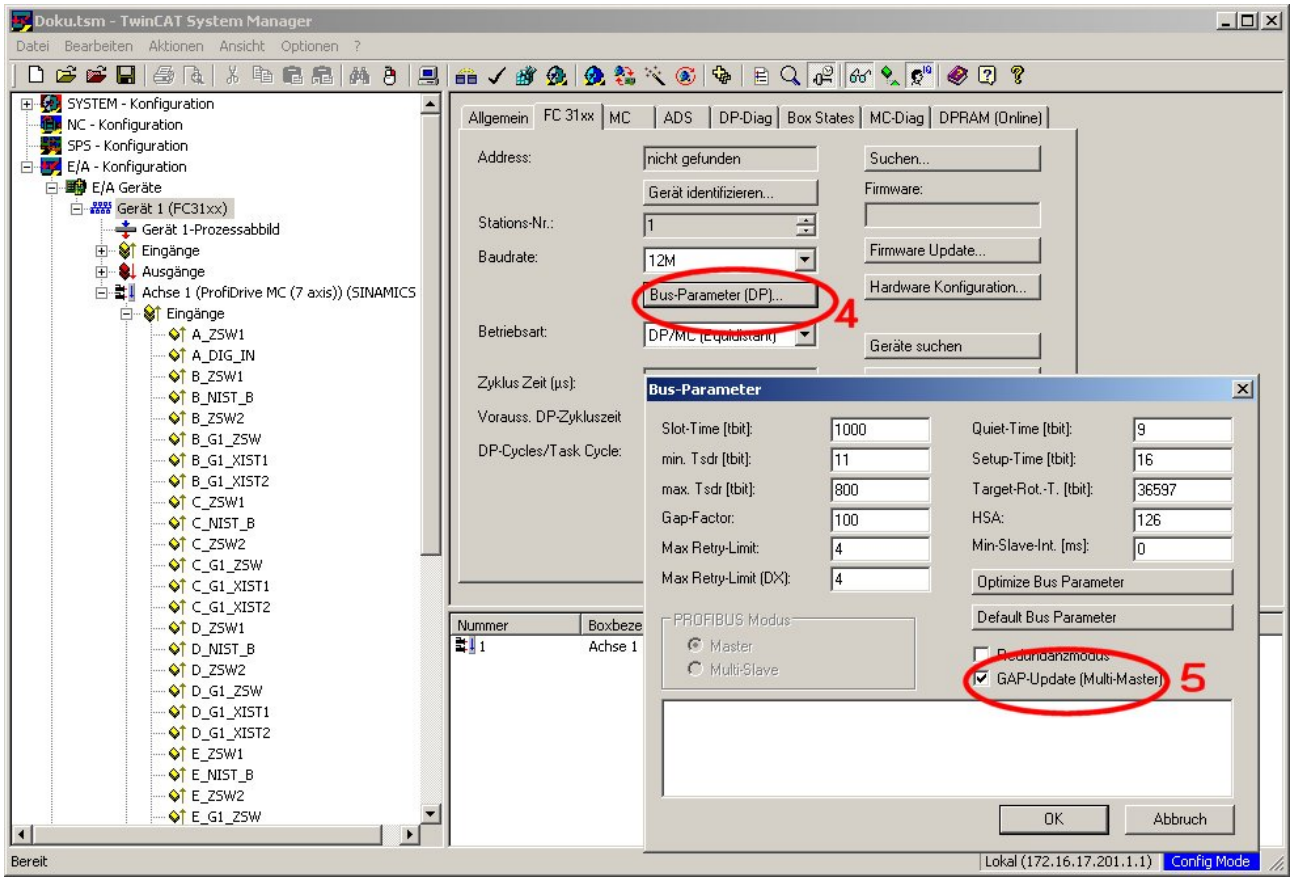


Abb. 100: Konfiguration 2

(4) Mit dem Button Bus Parameter (DP) wird über den Dialog GAP-Update (Multimaster) aktiviert
(5) um den Betrieb eines Programmiergerätes am Profibus zur parallelen Konfiguration der STARTER-Software zu ermöglichen

5.1.7.1.2 Feldbusteilnehmer konfigurieren

Box anfügen

Im Kontextmenu der Profibus-Master-Karte Gerät1 kann "Box Anfügen" (Append Box) gewählt werden. Ausgewählt wird aus dem Ordner PROFIdrive MC (DPV2) (x-fach) Bei der gewählten Anzahl ist zu beachten, dass 1-8 fach ausgewählt werden kann. Dabei zählen jeweils neben den bis zu 6 Antriebsreglern eine CU und ein Active Line Modul extra. Im Beispiel (siebenfach) soll mit einem Verbund aus 6 Achsen und einer CU über den Profibus kommuniziert werden.

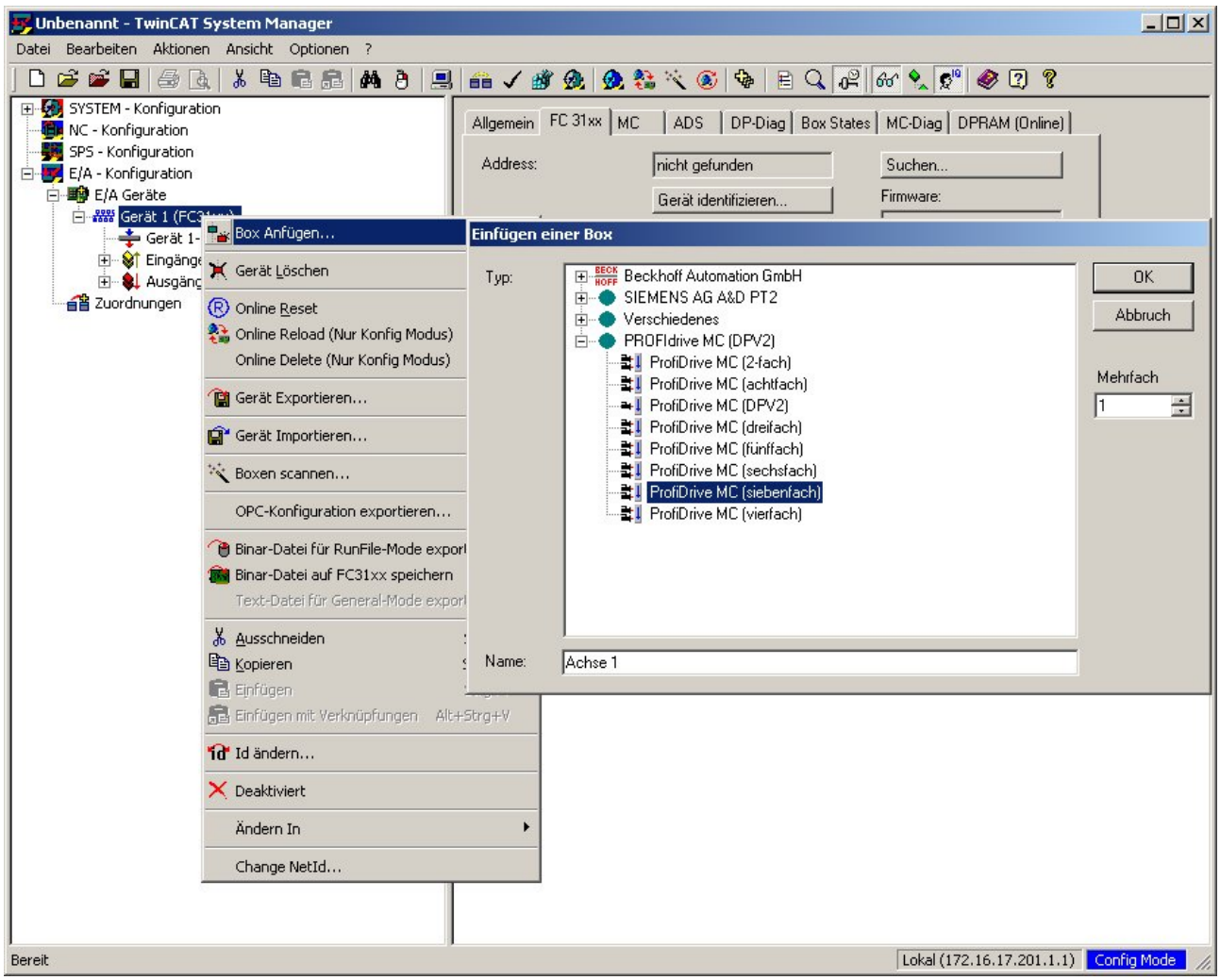


Abb. 101: Einfügen einer Box

GSG-Datei

Bitte wählen Sie aus dem Auswahldialog "Wähle Box" den Eintrag "Sinamics S":
Stellen Sie sicher, dass die Datei Profidrive2.dat und die in dieser Datei genannte GSG-Datei (Editor) sich im Verzeichnis \TwinCAT\IO\Profibus\ befindet

Abb. 102: Auswahldialog

Beispiel Profidrive2.dat:

```
Simodrive 611U=
Sinamics S=si0480e5.gsg
FC310x-MC-Slave=FC310xMC.gsd
```

Es wird dann die entsprechende GSG-Datei eingelesen und die Datenstrukturen werden voreingestellt angelegt. Wenn eine neue, geänderte GSG Datei benutzt werden soll, muss die Box (der Achsverband) neu angelegt werden, Verknüpfungen zu den Softwarestrukturen (siehe Kapitel NC-Konfiguration [▶ 118]) müssen dann neu erzeugt werden.

Profibus-Slave Konfiguration

Die Feldbus-Konfiguration des Achsverbundes wird im Kartenreiter "Profibus eingestell"t:
Die Profibus-Adresse (Stations Nr.: 2 im nachfolgenden Bild) ist entsprechend der in der Antriebskonfiguration (STARTER-Programm) festgelegten Adresse der CU einzustellen

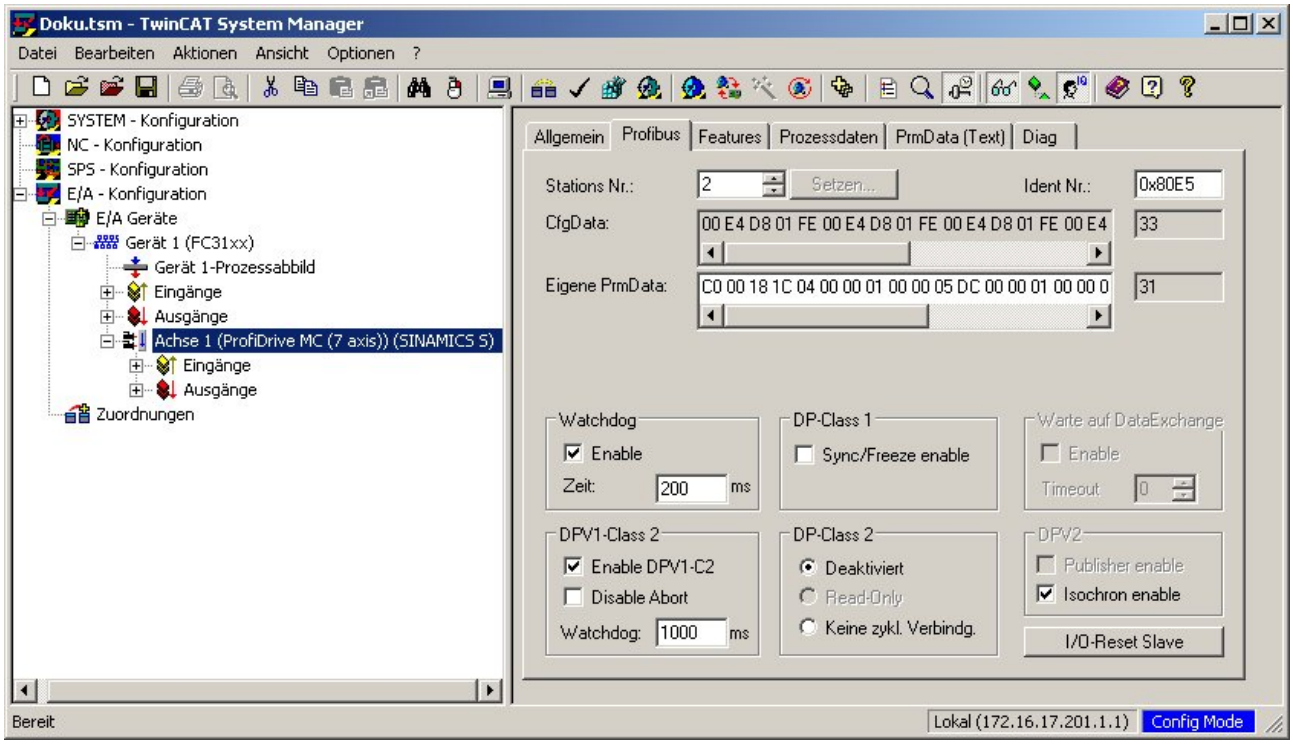


Abb. 103: Karteireiter Profibus, Adresseinstellung

Konfiguration des PROFIdrive Achsverbundes in TwinCAT

Die in der Starter-Software eingestellte Objekt-Nr. (Object-ID) 1-8 entspricht dem Kanal (Channel) A-H. Jedem Kanal muss ein geeignetes Telegramm zugeordnet werden. Im Kartenreiter Prozessdaten werden die einzelnen Antriebe der Achse (des Achsverbundes) per Default-Einstellung auf das Standard-Telegramm 3 eingestellt. Der DSC Betrieb (Telegramm 5) hat sich bei vorausgegangenen Tests im Zusammenhang mit TwinCAT als nicht notwendig herausgestellt. Für die CU ist das Telegramm 390 vorgesehen.

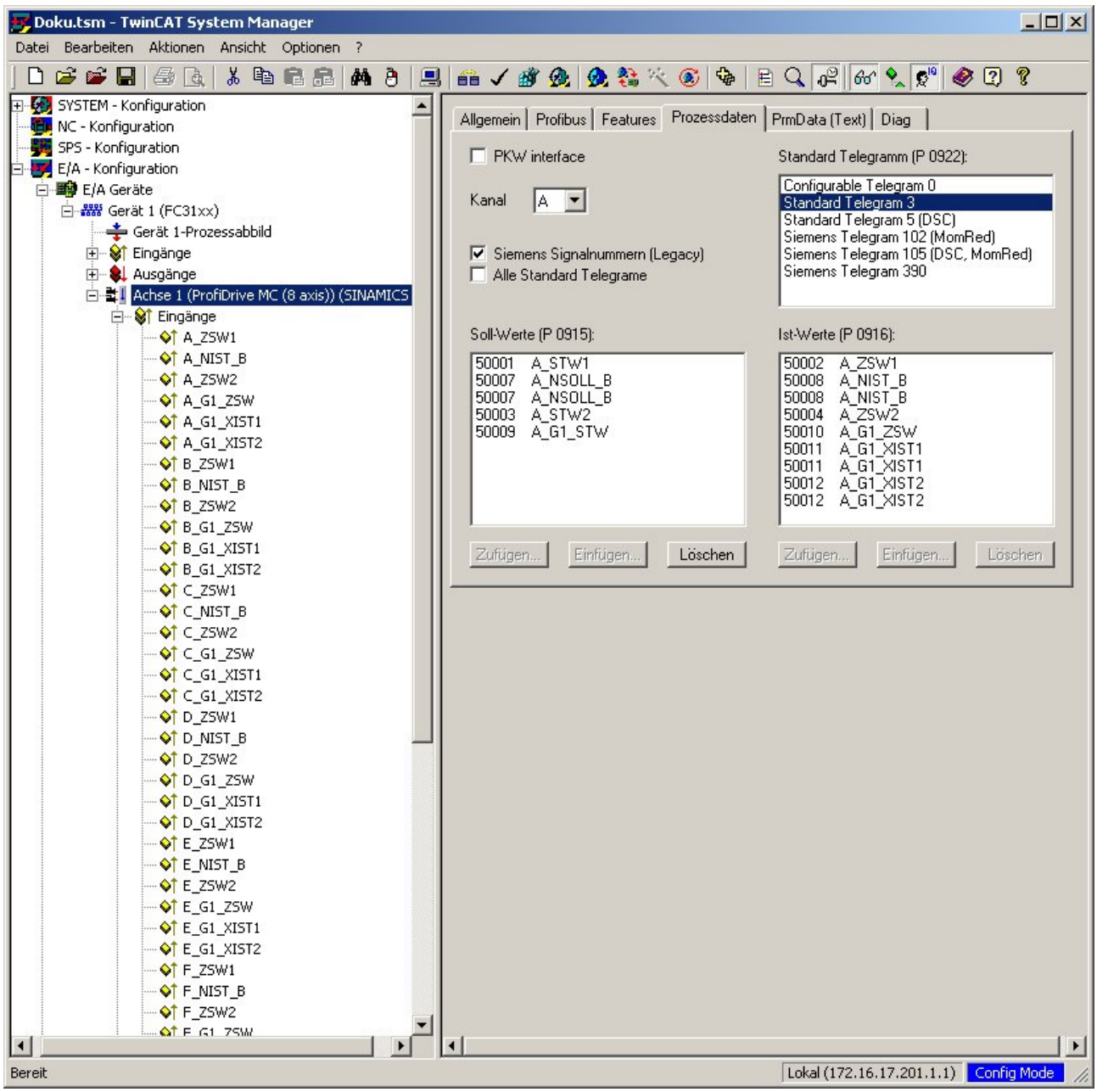


Abb. 104: Karteireiter Prozessdaten, Zuordnung Telegramm

Dieses Telegramm 370 für das Active Line Modul ALM ist bis TwinCAT 2.10, B1319 nicht vordefiniert, kann aber aus dem Configurable Telegram 0 zusammengestellt werden.

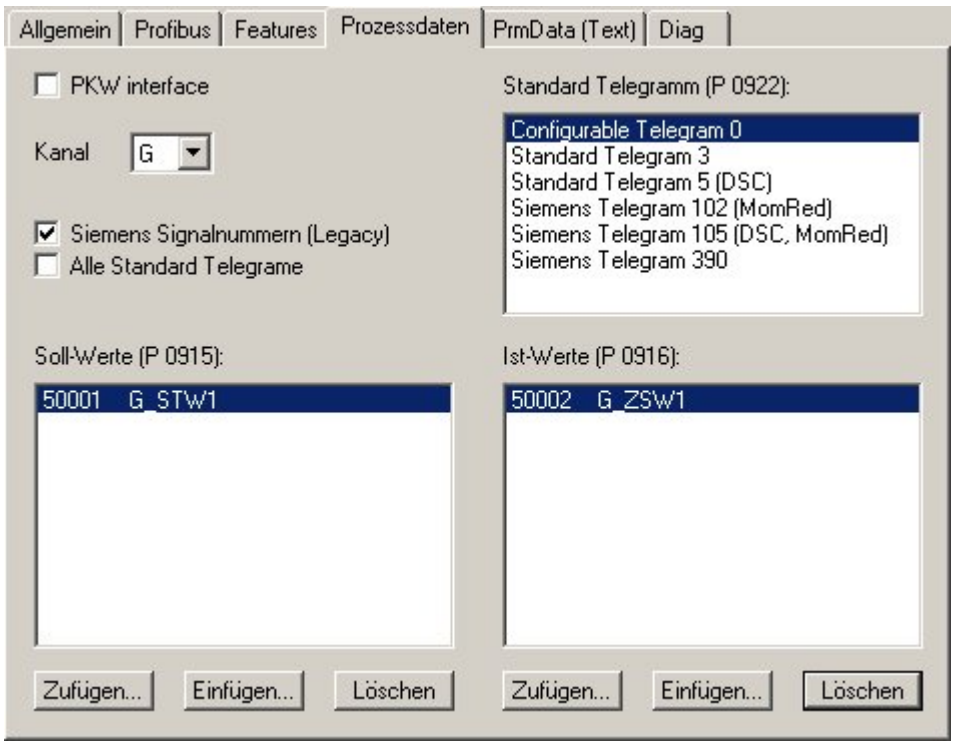


Abb. 105: Karteireiter Prozessdaten, Telegramm bestimmen

Im Beispiel muss der Parameter "PKW-Interface" deaktiviert bleiben, da Sinamics S (anders als Simodrive 611U) das PKW-Interface nicht mehr unterstützt.

Profibus DP Zyklus ermitteln

Nachdem der gesamte Feldbus konfiguriert wurde, betätigen Sie den Button "Calculate MC Times" im Kartenreiter MC der FC31xx.
Die "Estimated DP-Cycle-Time (hier 374 µs)" muss für alle Achsen zusammen kleiner sein, als die im Kapitel NC-Konfiguration [▶ 117] behandelte NC-Task Zeit (z. B. 2 ms). Sie sollten den tatsächlichen Wert später online überprüfen.

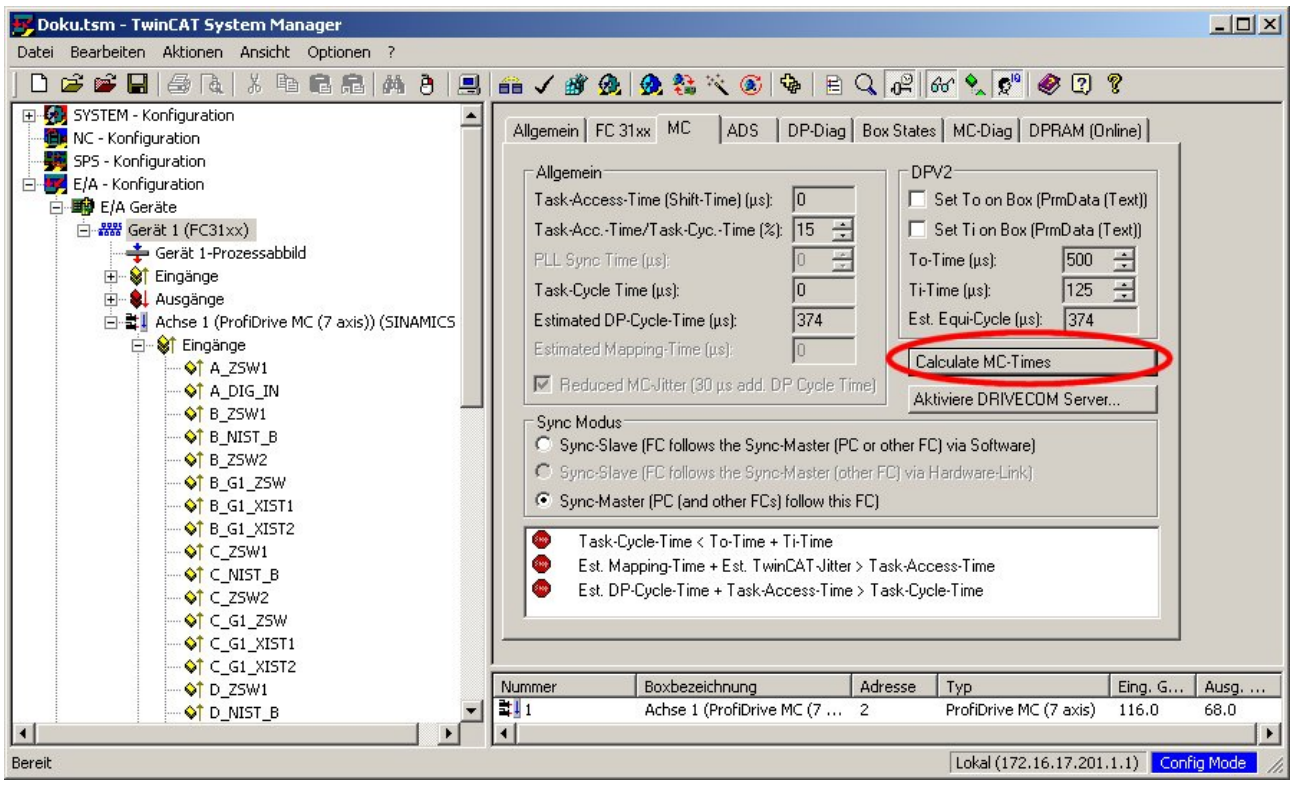


Abb. 106: Karteireiter MC, PB - DP Zyklus ermitteln

5.1.7.2 NC-Konfiguration

NC-Task anfügen und parametrieren

Nach Anfügen einer NC-Task wird die Zykluszeit der Satzausführungs-Task (NC-Task x SAF) parametriert. Die Angabe der Zyklusticks legt die Zykluszeit in Vielfachen der eingestellten TwinCAT Basis Zeit (Standard: 1 ms) der Task fest. Die Zykluszeit ist dann wirksam, wenn Auto-Start angewählt wird. Die Zykluszeit der Task muss größer sein als die erwartete Feldbus-Zykuszeit Estimated DP-Cyle-Time (siehe Kaptiel E/A Konfiguration [▶ 110]). Kleinere Zeiten als 2 ms sind mit Sinamics S120 bisher nicht getestet worden.

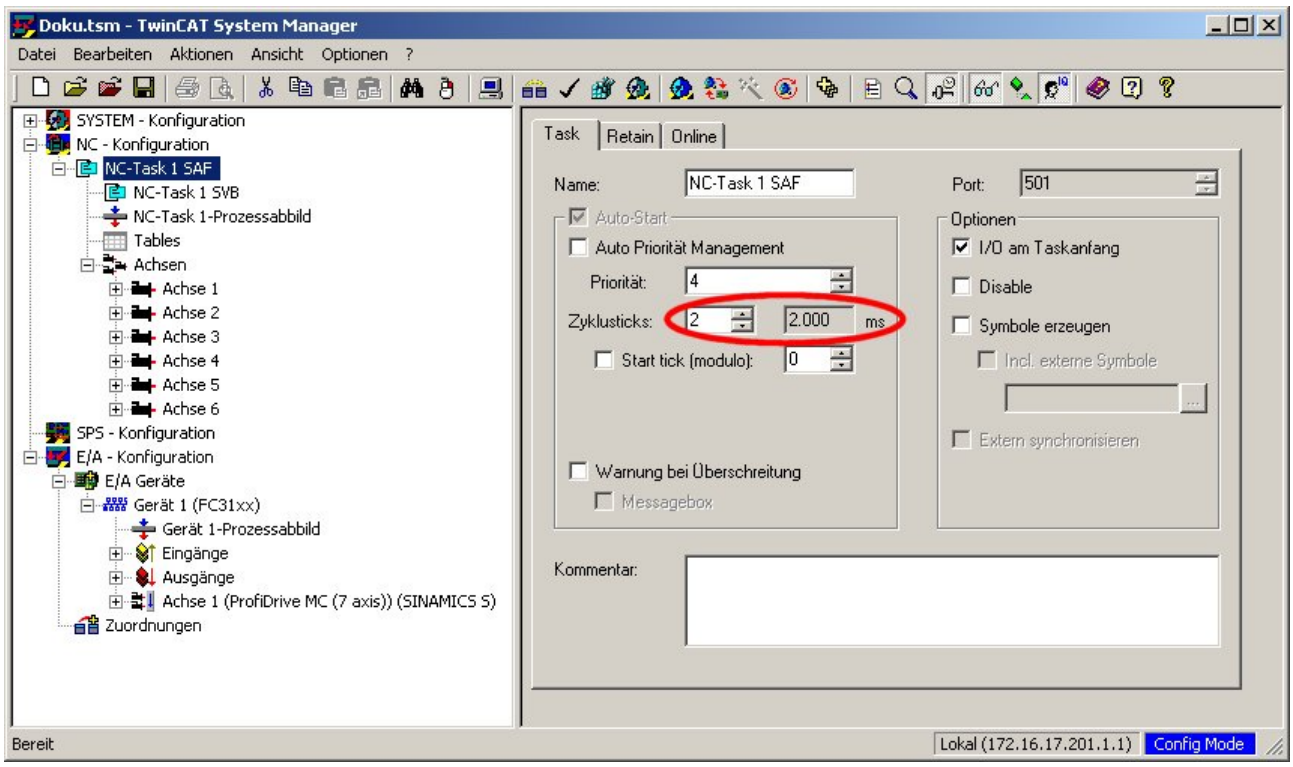


Abb. 107: NC-Task, Zykluszeit

Achsen anfügen und parametrieren

Im Kontextmenü des Symbols "Achsen" fügen Sie die benötigte Anzahl der Achsen ein. Die Namen Achse 1, Achse 2,… sind nur Platzhalter. Üblicherweise werden dabei zur besseren Identifikation die Namen der Funktion in der Software angegeben, bzw. werden diese Strukturen aus einem Vorgängerprojekt übernommen. Im Kartenreiter "Einstellungen" der jeweiligen Einzel-Achse legen Sie den **Achstyp** auf Profidrive MC (DPV2) und die Einheit zur Istpositionserfassung (z. B. mm bei Linearachse) fest. Mit dem Button "Verknüpft mit…" legen Sie fest, welche dieser logischen Achsen (Softwarestrukturen) mit welcher PROFIdrive MC -Einzelachse (Achsverbund x, #A…#H) verbunden wird.

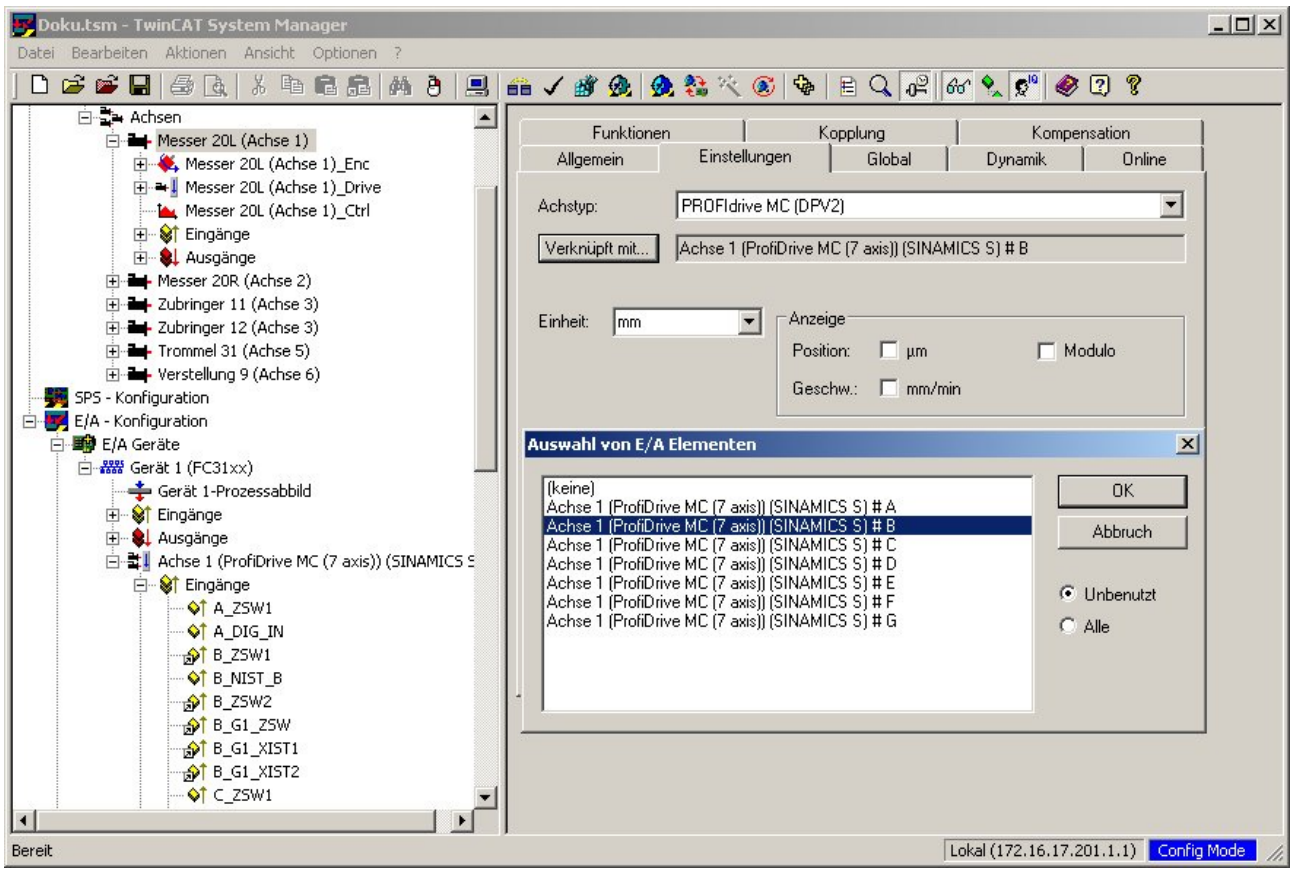


Abb. 108: Konfiguration, Verknüpfung der Achsen

Wenn zum Steuern oder zur Diagnose benötigt, werden CU und Active Line Modul mit anzulegenden Variablen (Strukturen) eines SPS-Projektes verknüpft.

Abschließend Profibus MC Times überprüfen

Nachdem die NC und der Feldbus konfiguriert wurden, betätigen sie im Online-Betrieb noch einmal den Button "Calculate MC Times" im Kartenreiter MC der Masterkarte FC31xx (I/O-Konfiguration) und beachten Sie eventuelle Warnungen. Die "Estimated DP-Cycle-Time" muss für alle Achsen zusammen kleiner sein, als die oben behandelte NC-Task Zeit. Eine Sicherheitsreserve von ca. 10% sollte vorhanden sein.

Parameter von PROFIdrive MC lesen

Der folgende Schritt erfolgt im Rahmen der Inbetriebsetzung und erfordert das Vorhandensein der parametrierten Achsen und einen funktionsfähigen Feldbus zum PROFIdrive MC Achsverbund.

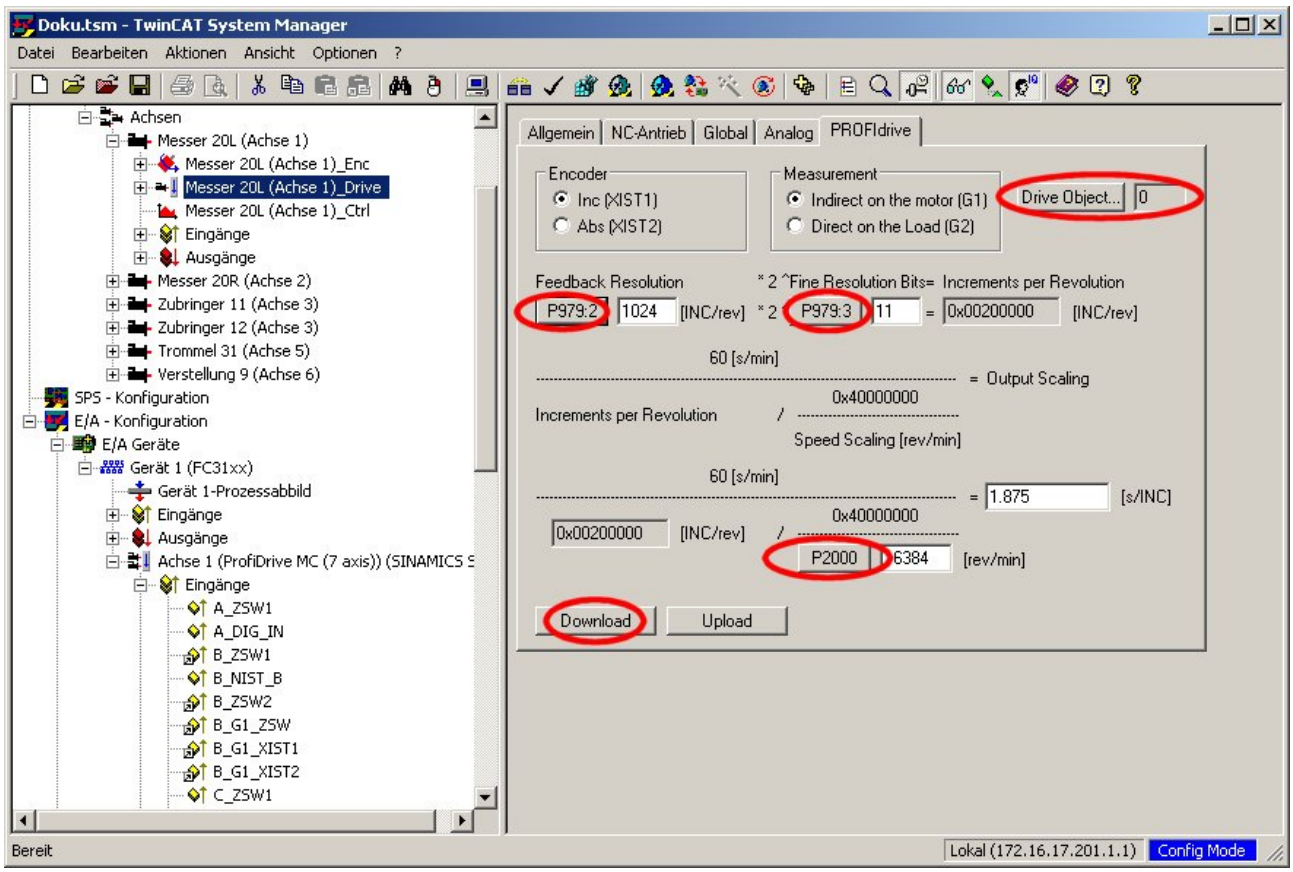


Abb. 109: Karteireiter PROFIdrive, Motorparameter lesen

Im Kartenreiter "PROFIdrive" des Drive der Achse (im Beispiel Messer 20L(Achse 1)_Drive) findet sich ein Dialog, um Motorparameter aus dem Antrieb zu lesen. Dazu muss die Drive Object Nummer aus der STARTER-Software angegeben werden. Dann können mit den entsprechenden Buttons die Geberauflösung über die Parameter P979.2, P979.3 und P2000 gelesen werden. Alternativ können diese Parameter mit der STARTER Software gelesen und manuell in die Eingabefelder geschrieben werden. Die Übernahme der Werte in TwinCAT erfolgt in beiden Fällen mit dem Button "Download".

Zusammen mit den mechanischen Daten (u.A. Übersetzungsverhältnis) der Achse kann aus dem Wert INC/rev (Inkremente pro Umdrehung) der Skalierungsfaktor berechnet werden (im Beispiel der Linearachse in mm/INC).

Formel

Skalierungsfaktor = Weg pro Umdrehung der Abtriebswelle / (Getriebefaktor x Inkremente pro Umdrehung)
Getriebefaktor = Umdrehungen Motorwelle / Umdrehungen Abtriebswelle

Beispiel Förderband:

300 mm Vorschub pro Umdrehung der Welle, Getriebefaktor 10, Geberauflösung 20000H = 2097152DEZ INC/rev Skalierungsfaktor = 300 mm/rev / (10 x 2097152 INC/rev) = 0.000014305 mm/INC

Beispiel Trommel Rotation:

Winkel= 360° pro Umdrehung der Welle, Getriebefaktor 17.08, Übersetzung durch Zahnriemen 64/50, Geberauflösung 20000H = 2097152DEZ INC/rev Skalierungsfaktor = 360°/rev / (17.08 x 64/50 x 2097152 INC/rev) = 0.0000078519 mm/INC

Encoder Parameter Skalierungsfaktor einstellen

Der ermittelte Wert [▶ 119] muss beim Skalierungsfaktor des Encoders im Kartenreiter "Global"angegeben und die Werte durch "Download" übernommen werden

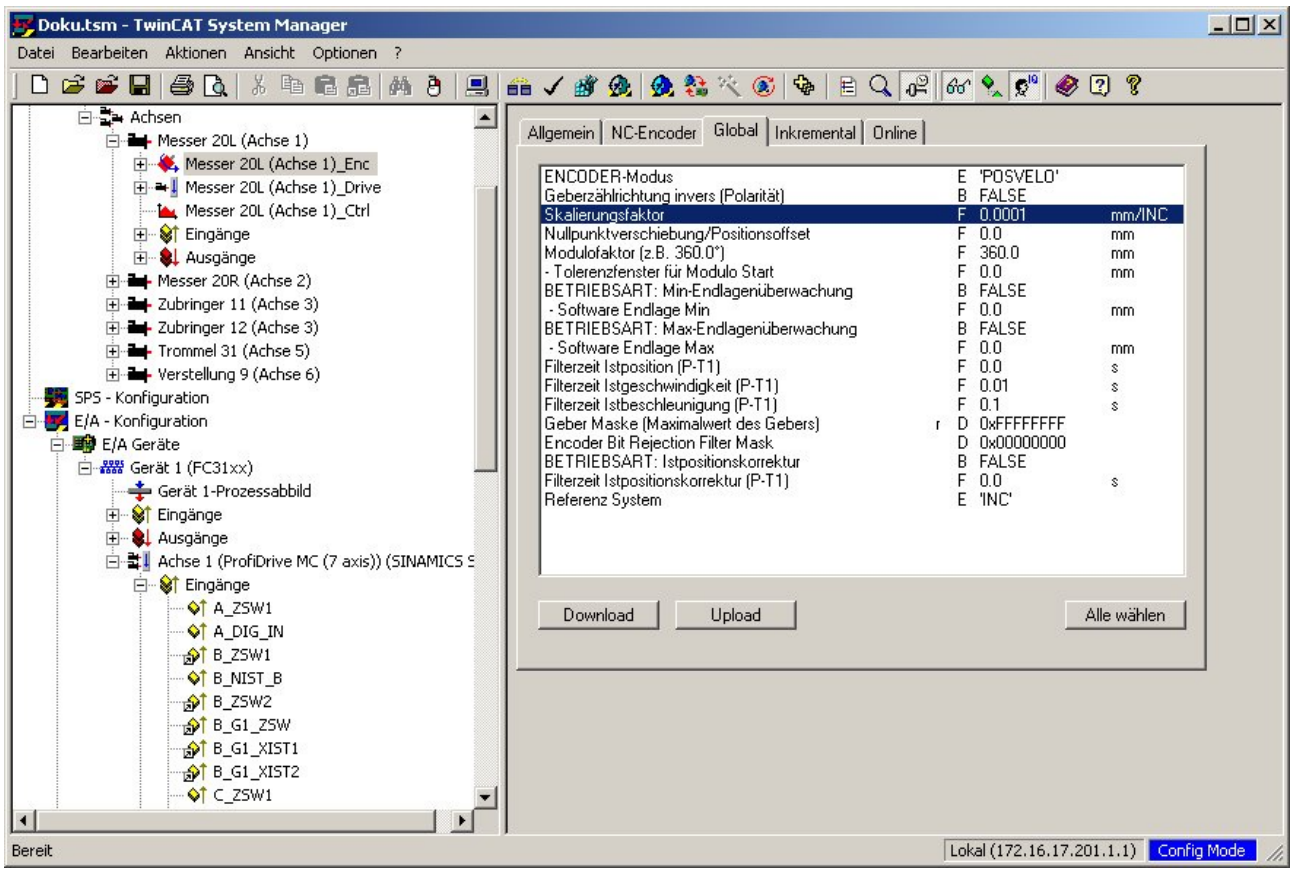


Abb. 110: Karteireiter „Global“, Eingabe Skalierungsfaktor

Alle weiteren Achseinstellungen und Inbetriebnahmeschritte entsprechen dem üblichen Vorgehen.

5.1.8 Diagnose und Fehlerbeschreibung

5.1.8.1 EL6731 - LED Beschreibung

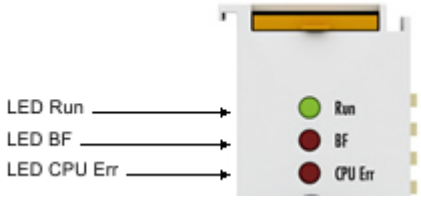


Abb. 111: LEDs

LED Verhalten

Anhand der LED's lassen sich die wichtigsten Zustände der Klemme schnell diagnostizieren:

LED	Farbe	Bedeutung	
RUN	grün	Diese LED gibt den Betriebszustand der Klemme wieder:	
		aus	Zustand der EtherCAT State Machine: **INIT** = Initialisierung der Klemme; **BOOTSTRAP** = Funktion für Firmware-Updates der Klemme
		Einzelblitz	Zustand der EtherCAT State Machine: **PREOP** = Funktion für Mailbox-Kommunikation und abweichende Standard-Einstellungen gesetzt
		Blinken	Zustand der EtherCAT State Machine: **SAFEOP** = Überprüfung der Kanäle des Sync-Managers und der Distributed Clocks. Ausgänge bleiben im sicheren Zustand
		an	Zustand der EtherCAT State Machine: **OP** = normaler Betriebszustand; Mailbox- und Prozessdatenkommunikation ist möglich
BF	rot	aus	DP-Master ist im CLEAR/OPERATE, alle DP-Slaves sind im Data-Exchange
		Blinken	DP-Master ist im CLEAR/OPERATE, mindestens ein DP-Slave ist im Data-Exchange
		an	DP-Master ist im STOP
CPU Err	rot	an	Fehler des Prozessors der EL6731
		Einzelblitz	Prozessor der EL6731startet

5.1.8.2 Fehlerreaktionen

Ausfall eines Slaves

Wenn ein Slave nicht oder gestört antwortet, wiederholt der Master das Telegramm entsprechend des **Max Retry-Limit** (s. Dialog Bus-Parameter [▶ 95]) mehrere Male. Beim Empfang eines gestörten Telegramms wiederholt der Master sofort, im Timeout-Fall hat der Master die **Slot-Time** (s. Dialog Bus-Parameter [▶ 95]) auf eine Antwort des Slaves gewartet. Bei 12 MBaud, einer Slot-Time von 1000 Bit-Zeiten und einem Max Retry-Limit von 4 (Defaultwerte) bei einem Data_Exchange-Telegramm verzögert sich das Senden des folgenden Telegramms um den Wert

TDelay = (4 x ((15 + Anzahl Outputs [Größe in Byte]) x 11 + 1000) - (15 + Anzahl Inputs [Größe in Byte]) x 11)/12 µs

Der DpState [▶ 129] des Slaves wird auf 0x02 (Timeout) bzw. 0x0B (gestörtes Telegramm) gesetzt. Die Auswirkung auf die DP-Verbindung kann eingestellt werden (s.u.).

Normaler DP-Zyklus (12 MBaud, 5 Slaves, im Schnitt 20 Bytes I, 20 Bytes O je Slaves)

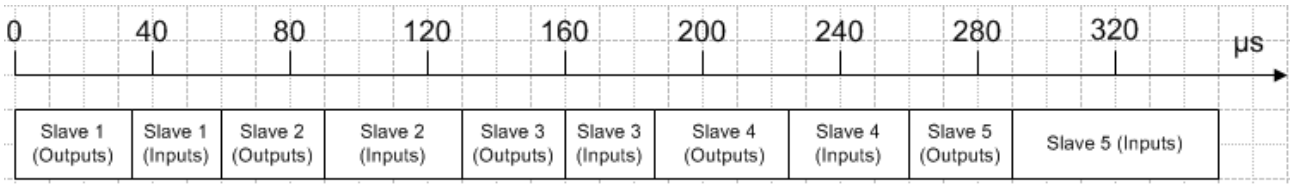


Abb. 112: Diagramm Normaler DP-Zyklus

Erstmalig gestörter DP-Zyklus (Slave 3 antwortet nicht)

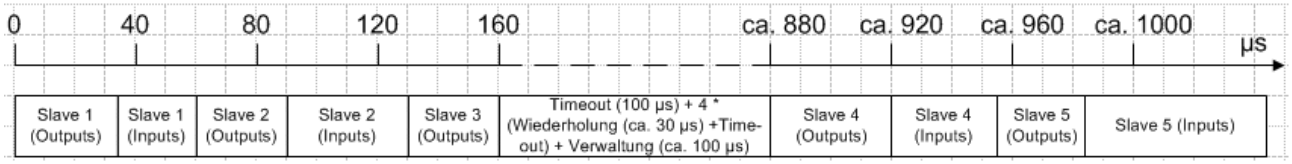


Abb. 113: Diagramm erstmalig gestörter DP-Zyklus

Folgende DP-Zyklen (Slave 3 nicht mehr in der Poll-Liste)

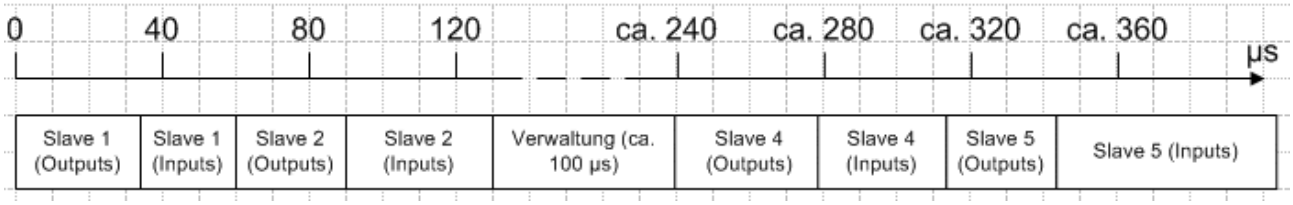


Abb. 114: Diagramm folgende DP-Zyklen

Weiterhin kann es noch passieren, dass der Slave falsch antwortet (z. B. weil aufgrund eines lokalen Ereignisses auf dem Slave die DP-Verbindung abgebaut wurde). In diesem Fall wird das Telegramm nicht wiederholt, sondern mit dem Senden des folgenden Telegramms fortgefahren. Der DpState [▶ 129] wird auf einen Wert ungleich 0 gesetzt, der Slave wird aus der Poll-Liste ausgetragen und im nächsten DP-Zyklus nicht mehr angesprochen (d.h. der Sendezeitpunkt des nachfolgenden Telegramms verändert sich), bis die DP-Verbindung wieder erneut aufgebaut wurde.

Reaktionen im Master

Die Reaktion im Master kann je Slave eingestellt werden (s. Karteireiter Features [▶ 105] des Slaves).

Auswirkung auf die DP-Verbindung (NoAnswer-Reaction), falls Slave nicht oder nicht ungestört antwortet

Hier wird festgelegt, ob der die DP-Verbindung zu dem Slave sofort oder erst nach Ablauf der DP-Watchdog-Zeit (s. Karteireiter Profibus [▶ 104] des Slaves) ohne korrektes Empfangstelegramm abgebaut werden soll.

1. Wenn die DP-Verbindung sofort abgebaut werden soll (Leave Data-Exch, Defaulteinstellung), wird der Slave aus der Poll-Liste ausgetragen und im nächsten DP-Zyklus nicht mehr angesprochen, bis die DP-Verbindung wieder erneut aufgebaut wurde. Um die DP-Verbindung zu dem Slave wieder aufzubauen werden mindestens 7 Telegramme gesendet, der Vorgang dauert in der Regel mindestens 10-20 ms.
2. Wenn die DP-Verbindung erst abgebaut werden soll, wenn innerhalb der DP-Watchdog-Zeit der Slave nicht oder ungestört geantwortet hat (Stay in Data-Exch (for WD-Time)), wird im nächsten Pollzyklus erneut versucht, den Slave anzusprechen, falls der Slave aber nicht antwortet, wird keine Wiederholung gesendet.

Die Einstellung "Stay in Data-Exch (for WD-Time))" (2.) macht dann Sinn, wenn der PROFIBUS-Zyklus auch beim Ausfall eines Slaves zeitlich möglichst konstant ablaufen soll und der Ausfall eines Slave für einen oder mehrere Zyklen toleriert werden kann (z. B. in der Betriebsart DP/MC (Equidistant) [▶ 48]). In diesem Fall sollte die DP-Watchdog-Zeit des Slaves entsprechend der noch tolerierbaren Ausfall-Zeit des Slaves eingestellt und das **Max Retry-Limit (DX)** (s. Dialog Bus-Parameter [▶ 95]) auf 0 gesetzt werden.

Normaler DP-Zyklus (12 MBaud, 5 Slaves, im Schnitt 20 Bytes I, 20 Bytes O je Slaves) im Mode "Stay in Data-Exch (for WD-Time)"

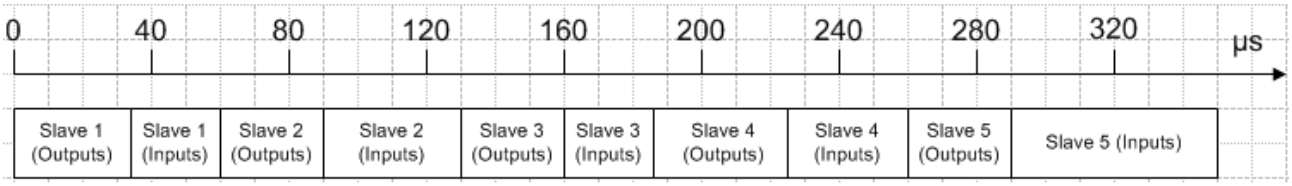


Abb. 115: Diagramm normaler DP-Zyklus

Erster gestörter und folgende DP-Zyklen im Mode "Stay in Data-Exch (for WD-Time)" (Slave 3 antwortet nicht)

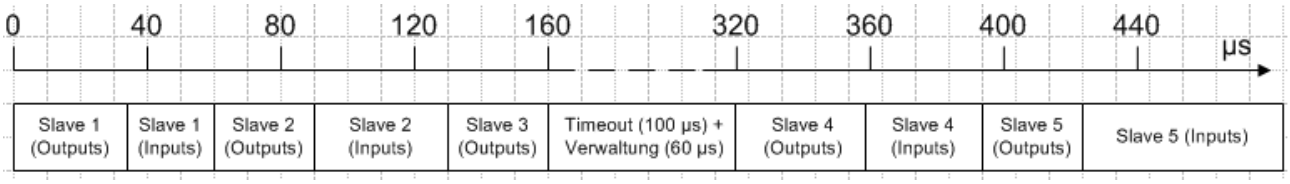


Abb. 116: Diagramm: Erster gestörter und folgende DP-Zyklen im Mode "Stay in Data-Exch (for WD-Time)"

Auswirkung auf die Input-Daten des Slaves (Changes of the Input Data), falls Slave nicht richtig antwortet

Hier wird festgelegt, ob die Input-Daten des Slaves bei dessen Ausfall auf 0 gesetzt ("Inputs will be set to 0", Defaulteinstellung) oder der alte Wert erhalten bleiben soll ("No changes"). In beiden Fällen wird auf jeden Fall der DpState [▶ 129] des Slaves auf einen Wert ungleich 0 gesetzt, so dass die Task immer erkennen kann, ob die Daten gültig sind oder nicht. Wenn ein Slave gestört antwortet, werden die Inputdaten, unabhängig von der Einstellung **Changes of the Input Data** immer auf 0 gesetzt.

Einstellung des Wiederanlauf-Verhaltens des Slaves (Restart Behaviour of the Slave), falls die DP-Verbindung zu dem Slave abgebaut wurde

Hier wird festgelegt, ob die DP-Verbindung zu einem Slave, dessen DP-Verbindung abgebaut wurde, automatisch wieder aufgebaut wird, oder ob dieses manuell durch einen ADS-WriteControl-Aufruf (s. ADS-Interface [▶ 51]) passieren soll.

Auswirkung auf den Zustand des Masters (Reaction of the Master), falls die DP-Verbindung zu dem Slave abgebaut wurde

Hier wird festgelegt, ob der Abbau der DP-Verbindung zu einem Slave ohne weitere Auswirkungen bleiben (No Reaction, Defaulteinstellung) oder ob der Master den STOP-Zustand einnehmen und damit die DP-Verbindungen zu allen Slaves abbauen soll.

Auswirkung auf den Zustand des Masters (Clear Mode), falls die DP-Verbindung zu dem Slave abgebaut wurde

Mit dem **Clear Mode** (s. Dialog Fault-Settings [▶ 97]) kann eingestellt werden, dass der Master in den Zustand "Clear" wechselt bzw. bleibt, solange mindestens ein MC-Slave (Einstellung "Only MC-Slaves") bzw. irgendein Slave (Einstellung "All Slaves") nicht korrekt antwortet (einen DpState [▶ 129] ungleich 0 hat).

Die Einstellung **Reaction of the Master** (s. Karteireiter Features [▶ 105] des Slaves), die im vorangegangenen Kapitel beschrieben wurde, hat gegenüber dem **Auto-Clear Mode** Vorrang, d.h. wenn ein entsprechend eingestellter Slave ausfällt, geht der Master in den Zustand STOP.

Ausfall des Masters

Überwachung in der SPS/IO-Task

Bei dauerhaften Busstörungen kann sich der DP-Zyklus auch bei 12 MBaud auf bis zu 100 ms verlängern. Um den DP-Master zu überwachen, gibt es eine Status-Variable CycleCounter, die in der SPS verknüpft werden kann (s. Kapitel Master-Diagnose [▶ 126]). Diese Variable wird vom Master nach jedem DP-Zyklus inkrementiert, durch Überwachen dieser Variable in der SPS kann also ein Ausfall des Masters festgestellt werden.

Überwachung im Slave

Um den Ausfall des Masters bzw. die Übertragung auf dem PROFIBUS zu überwachen, kann beim Slave ein **Watchdog** (s. Karteireiter Profibus [▶ 104] der Box) aktiviert werden (Defaulteinstellung: Watchdog aktiviert mit 200 ms). Der **Watchdog** muss mindestens zweimal so groß wie das Maximum aus **Estimated Cycle Time** und **Cycle Time** (s. Karteireiter **"FC310x** bzw. **"EL6731"** (für TwinCAT des Masters) eingestellt werden.

Ausfall der SPS/IO-Task (FC310x) oder EtherCAT Unterbrechung (EL6731)

Es wird zwischen den Fällen SPS-Stop, Erreichen eines Breakpoints und Task-Stop [EL6731: EtherCAT-Unterbrechung] (IO-Task, NC-Task wird nur beim System-Stop gestoppt) unterschieden. Beim SPS-Stop werden die Ausgangsdaten von der SPS noch auf 0 gesetzt, beim Erreichen eines Breakpoints bleiben die Daten zunächst unverändert.

Im Master wird mit einer Überwachungszeit (entsprechend der Einstellung **Task-Watchdog** x Task-Zykluszeit auf dem Dialog Fault-Settings [▶ 97]) die Task überwacht. Wenn innerhalb dieser Überwachungszeit keine neue Datenübergabe erfolgt, geht der Master entsprechend der Einstellung **Reaction on PLC-Stop** bzw. **Reaction on Task-Stop** (s. Dialog Fault-Settings [▶ 97]) in den Zustand "Clear" (Ausgänge werden auf 0 bzw. in den sicheren Zustand (Fail_Safe = 1 in der GSD-Datei) gesetzt, Defaulteinstellung) oder bleibt im Zustand "Operate" (Ausgänge behalten den letzten Wert). Die Einstellung "Operate" macht Sinn, wenn beim Erreichen eines Breakpoints in der SPS die Ausgänge nicht abfallen sollen, bei einem SPS-Stop würden die Ausgänge trotzdem auf 0 gesetzt (durch die SPS), auch wenn der Master in "Operate" bleibt. Dabei ist allerdings zu beachten, das die Ausgänge nur genullt werden, wenn der vorangegangene DP-Zyklus rechtzeitig fertig war (s. Kapitel Synchronisierung [▶ 49]), es sollte daher nur während der Inbetriebnahmephase eingestellt werden.

Ausfall des Hosts [nur FC310x]

Um einen Absturz des Hosts (z. B. Blue Screen bei PC) zu überwachen, kann eine **Watchdog-Time** (s. Dialog Fault-Settings [▶ 97]) eingestellt werden. Wenn diese Watchdog-Time abläuft, geht der Master in den Zustand OFFLINE, d.h. die DP-Verbindungen zu allen Slaves werden abgebaut und der Master meldet sich vom PROFIBUS ab, d.h. er führt keine Buszugriffe mehr durch.

Start-Up Verhalten

Beim Start des TwinCAT Systems werden die DP-Verbindungen zu allen Slaves aufgebaut. Solange die höchstpriore zugehörige Task noch nicht gestartet wurde, sendet der Master auch nach dem Aufbau der DP-Verbindung noch keine Data_Exchange-Telegramme sondern Diagnose-Telegramme. Sobald die höchstpriore zugehörige Task einmal Daten übergeben hat und die DP-Verbindung des entsprechenden DP-Slaves aufgebaut ist, sendet der Master zyklisch (mit der höchstprioren zugeordneten Task) je ein Data_Exchange-Telegramm zu den entsprechenden Slaves.

Darüber hinaus kann über die Einstellungen **Operate-Delay** und **Clear Mode** (s. Dialog Fault-Settings [▶ 97]) festgelegt werden, wann der Master vom "Clear"-Zustand (Ausgänge werden auf 0 bzw. in den sicheren Zustand (Fail_Safe = 1 in der GSD-Datei) gesetzt) in den "Operate"-Zustand (Ausgänge entsprechen den von der Task übergebenen Ausgängen) wechselt. Das **Operate-Delay** gibt an, wie lange der Master nach dem erstmaligen Übergeben der Daten mindestens noch im Zustand "Clear" bleiben soll. Mit dem **Clear Mode** wird, wie weiter oben beschrieben, eingestellt, ob der Master beim Ausfall eines allgemeinen bzw. eines MC-Slaves in den Zustand "Clear" wechselt bzw. bleibt.

Shut-Down Verhalten

Beim Stop des TwinCAT Systems wird genauso reagiert, wie weiter oben im Kapitel "Ausfall des Hosts" beschrieben, die DP-Verbindungen aller Slaves werden abgebaut und der Master meldet sich vom Bus ab.

5.1.8.3 Master-Diagnose

Diagnose Eingänge

Die EL6731 verfügt über verschiedene Diagnosevariablen, die den Zustand der Klemme sowie des Profibusses beschreiben und die in der SPS verknüpft werden können:

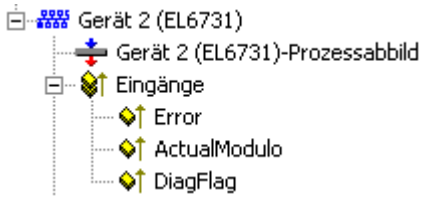


Abb. 117: Diagnose Eingänge EL6731 im TwinCAT-Baum

Error
Zeigt die Anzahl der Slaves an, mit denen kein Datenaustausch im letzten Zyklus durchgeführt werden konnte, nur wenn dieser Wert ungleich 0 ist, muss der BoxState der Slaves überprüft werden

ActualModulo
zeigt das aktuelle Modulo an, diese Variable spielt nur eine Rolle, wenn die Slaves priorisiert sind (s. Kapitel Slave-Priorisierung/mehrere DP-Zyklen [▶ 137])

DiagFlag
Zeigt an, ob sich die Master-Diagnoseinformationen der Karte geändert haben, die dann per ADS [▶ 51] vom Steuerungsprogramm ausgelesen werden, woraufhin die Variable "DiagFlag" wieder zurückgesetzt wird.

Master-Diagnosedaten

Die Master-Diagnosedaten können per ADS [▶ 51] ausgelesen werden:

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	200
IndexGroup	0x0000F100
IndexOffset	Offset innerhalb der Diagnosedaten
Length	Länge der auszulesenden Diagnosedaten
Data	Diagnosedaten

Die Master-Diagnosedaten sind wie folgt aufgebaut:

Offset	Beschreibung
0 -125	BusStatus-Liste, je Stationsadresse 0-125 ein Byte, das den Status der Station enthält (s. BoxState bei den PROFIBUS-Boxen, zusätzlich für nicht projektierte Stationen: 0x80 - nicht vorhanden, 0x81 - Slave, 0x82 - Master nicht bereit für Token-Ring, 0x83 - Master bereit für Token-Ring, 0x84 - Master im Token-Ring)
126 - 127	reserviert
128 - 135	Zustand der EL6731
130 - 131	detected bus errors
136 - 137	Sende-Fehler-Zähler über alle gesendeten Telegramme
138 - 139	Empfangs-Fehler-Zähler über alle empfangenen Telegramme
140 - 255	reserviert für Erweiterungen
256 - 257	Sync-Failed-Counter (s. Karteireiter MC-Diag [▶ 102])
258 - 259	Zyklus-Start-Fehler-Zähler, zählt hoch, wenn der PROFIBUS-Zyklus erneut gestartet wird bevor der alte Zyklus beendet wurde (wird vom TwinCAT-IO-Treiber abgefangen, nur bei kundenspezifischen Treibern möglich)
260 - 261	Time-Control-Failed-Counter (s. Karteireiter MC-Diag [▶ 102])
262 - 263	reserviert für Erweiterungen
264 - 265	minimaler Nachladewert des Echtzeittimers
266 - 267	maximaler Nachladewert des Echtzeittimers (max. Jitter , s. Karteireiter MC-Diag [▶ 102] = max. Nachladewert - min. Nachladewert)
268 - 269	PLL-Overflow-Counter (s. Karteireiter MC-Diag [▶ 102])
270 - 271	PLL-Underflow-Counter (s. Karteireiter MC-Diag [▶ 102])
296 - 297	Output-Failed-Counter
298 - 299	Input-Failed-Counter
300 - 301	actual Cycle-Time (µs)
302 – 303	min. Cycle-Time (µs)
304 - 305	max. Cycle-Time (µs)
306 - 307	CycleWithNoDxch-Counter
308 - 309	CycleWithRepeat-Counter
310- 311	max. Repeat/Cycle

Weitere Hinweise zu den Diagnosedaten

Weitere Hinweise zu den Diagnosedaten finden Sie im Kapitel Karteireiter DP-Diag [▶ 100].

5.1.8.4 Slave-Diagnose

DP-State

Je DP-Slave gibt es eine Status-Variable, die den aktuellen Zustand des DP-Slaves anzeigt. Dieser Status ist ein Echtzeit-Status, d.h. er paßt immer zu den aktuellen Daten des DP-Slaves und kann mit einer PLC-Variable verknüpft werden (-> DpState [▶ 129] des Slaves):

Abb. 118: Eingang DpState im TwinCAT-Baum

Diagnosedaten

Jeder DP-Slave kann während des Data_Exchange-Betriebs azyklisch DP-Diagnosedaten melden. Dabei setzt der Slave in der Antwort des zyklischen Data_Exchange-Telegramms das Diag_Flag, woraufhin der DP-Master automatisch die DP-Diagnosedaten beim Slave ausliest. Der Data-Exchange-Zyklus wird beim Beckhoff-DP-Master dabei nicht beeinflusst, da das DP-Diagnosetelegramm am Ende des zyklischen Data-Exchange-Zyklus (vor dem Anfang des nächsten Zyklus) gesendet wird. Wenn sich die beim Slave ausgelesenen DP-Diagnosedaten gegenüber dem letzten Zustand geändert haben, setzt der DP-Master die Variable "ExtDiagFlag", die mit einer Variable des Steuerungsprogramms verknüpft werden kann.

Die aktuellen Diagnosedaten des DP-Slaves werden im System Manager Karteireiter Diag [▶ 108] des Slaves angezeigt. Außerdem können sie per ADS [▶ 51] vom Steuerungsprogramm ausgelesen werden, woraufhin die Variable "ExtDiagFlag" wieder zurückgesetzt wird:

ADS-Read-Parameter	Bedeutung
Net-ID	Net-ID des Masters (s. Karteireiter ADS [▶ 100] des Devices)
Port	200
IndexGroup	0x00yyF181 (yy = Stationsadresse des Slaves)
IndexOffset	Offset innerhalb der Diagnosedaten
Length	Länge der auszulesenden Diagnosedaten
Data	Diagnosedaten

Die Diagnosedaten beihalten die Slave-Statistiken (32 Bytes) und die vom Slave gesendeten DP-Diagnosedaten (bis zu 244 Bytes) und sind wie folgt aufgebaut:

Offset	Bedeutung
Slave-Statistiken	
0	Receive-Error-Counter (WORD): Anzahl der fehlerhaften Telegramme bei der Kommunikation mit diesem Slave
2	Repeat-Counter[8] (WORD): Die Repeat-Counter zeigen an, wie oft wie viele Repeats gemacht werden mussten. Repeat-Counter[0] zeigt an, wie oft ein Telegramm zu diesem Slave einmal wiederholt werden mußte, Repeat-Counter[1], wie oft ein Telegramm zu diesem Slave zweimal wiederholt werden musste, etc. Die maximale Anzahl der Wiederholungen wird mit dem Parameter **Max Retry-Limit** (s. Dialog Bus-Parameter [▶ 95]) eingestellt, der Wertebereich geht von 0 bis 7, daher gibt es hier 8 Repeat-Counter (für 1 bis 8 Wiederholungen)
18	reserviert für Erweiterungen
20	NoAnswer-Counter (DWORD): Anzahl der Telegramme bei der Kommunikation mit diesem Slave, auf die nicht geantwortet wurde. Wenn ein Slave das erste Mal nicht antwortet, wird entsprechend des **MaxRetryLimit** das Telegramm wiederholt, wenn er auch dann nicht geantwortet hat, wird beim nächsten Mal keine Wiederholung mehr durchgeführt.
24-27	Last-DPV1-Error[4] (BYTE): Hier wird die letzte fehlerhafte DPV1-Antwort eingetragen (Byte 0: DPV1-Dienst (Bit 7 ist gesetzt und zeigt damit einen Fehler an), Byte 1: Error_Decode, Byte 2: Error_Code_1 (Error_Class/Error_Code), Byte 3: Error_Code_2), s. Beschreibung DPV1-Fehlercodes [▶ 133]
27-31	reserviert für Erweiterungen
ab 32	DP-Diagnosedaten [▶ 131]

5.1.8.5 DP-State der Slaves

Wert	Beschreibung
0	No Error - Station ist im Datenaustausch
1	Station deactivated - Slave wurde deaktiviert, tritt zur Zeit nur temporär beim Hochlauf auf
2	Station not exists - Slave antwortet nicht am Bus -> Prüfen, ob Slave eingeschaltet, ob PROFIBUS-Stecker angeschlossen, ob Stationsadresse korrekt oder ob Busverkablung korrekt
3	Master lock - Slave ist mit anderem Master im Datenaustausch -> anderen Master vom Bus nehmen oder Slave mit anderem Master wieder freigeben
4	Invalid slave response - falsche Antwort des Slaves, tritt temporär auf, wenn Slave aufgrund eines lokalen Ereignisses den Datenaustausch beendet hat
5	Parameter fault - Parametrierungsfehler -> Prüfen, ob Buskoppler bzw. GSD-Datei korrekt, ob Stationsadresse korrekt oder ob UserPrmData-Einstellungen korrekt
6	Not supported - DP-Funktion wird nicht unterstützt -> Prüfen, ob GSD-Datei korrekt oder ob Stationsadresse korrekt
7	Config fault - Konfigurationsfehler -> Prüfen, ob die angefügten Klemmen bzw. Module korrekt sind
8	Station not ready -> Station im Hochlauf, wird während des Hochlaufs temporär angezeigt
9	Static diagnosis - Slave meldet statische Diagnose und kann zur Zeit keine gültigen Daten liefern -> Betriebszustand am Slave prüfen
10	Diagnosis overflow - Slave meldet einen Diagnose-Überlauf -> Diagnosedaten (mit ADS-Read, s.u.) und Betriebszustand am Slave prüfen
11	Physical fault - physikalischer Fehler bei Antwort eines Slaves -> Verkablung prüfen
13	Severe Bus fault - schwerer Busfehler -> Verkablung prüfen
14	Telegram fault - Slave antwortet mit einem ungültigen Telegramm -> darf nicht auftreten
15	Station has no ressources -> Slave hat nicht genügend Ressourcen für das Telegramm -> Prüfen, ob GSD-Datei korrekt
16	Service not activated -> temporärer Fehler, falls Slave aufgrund eines lokalen Ereignisses den Datenaustausch beendet, oder prüfen, ob DP-Funktionalität beim Slave disabled ist
17	Unexpected telegram -> unerwartetes Telegramm, kann temporär auftreten, falls zwei PROFIBUS-Netze zusammengesteckt werden oder prüfen, ob Buszeiten bei zweitem Master gleich eingestellt sind.
18	Station ready -> tritt temporär beim Hochlauf auf und solange die Task noch nicht gestartet ist
19	DPV1 StartUp -> tritt temporär nach dem DP-Hochlauf auf, wenn noch Daten per DPV1-Write gesendet werden
31	nur für EtherCAT Gateways: WC-State des zyklischen EtherCAT Frame ist 1
128	FC310x/EL6731 in Slave-Mode, waiting for data transfer -> Slave wurde parametriert und konfiguriert hat aber noch keine Data_Exchange-Telegramm empfangen
129	FC310x/EL6731 in Slave-Mode, waiting for configuration -> Slave wurde parametriert, hat aber noch kein Chk_Cfg-Telegramm empfangen
130	FC310x/EL6731 in Slave-Mode, waiting for parameter -> Slave wurde noch nicht parametriert, wartet auf Set_Prm (Lock)-Telegramm

5.1.8.6 Konfigurationsdaten - CfgData

Die CfgData beschreiben die Struktur und die Länge der Ein- und Ausgangsdaten, die per Data_Exchange zyklisch ausgetauscht werden sollen. Es folgt eine Beschreibung der DP-Konfigurationsdatenbytes

Bit 4-7	Bedeutung
0000B	Modul ohne Daten, Bit 0-3 geben an, wie viel Bytes herstellerspezifische Daten noch folgen
0001B	Eingänge vom Typ Byte ohne Konsistenz, in Bit 0-3 steht die Länge der Eingangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Byte, Bit 0-3 = 1111B entspricht Länge 16 Bytes)
0010B	Ausgänge vom Typ Byte ohne Konsistenz, in Bit 0-3 steht die Länge der Ausgangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Byte, Bit 0-3 = 1111B entspricht Länge 16 Bytes)
0011B	Ein- und Ausgänge vom Typ Byte ohne Konsistenz, in Bit 0-3 steht die Länge der Eingangs- bzw. Ausgangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Byte, Bit 0-3 = 1111B entspricht Länge 16 Bytes)
0100B	spezielles Kennungsformat Eingänge, es folgt ein Byte, das die zugehörigen Eingangsdaten beschreibt (s.u.), Bit 0-3 geben an, wie viel Bytes herstellerspezifische Daten danach noch folgen
0101B	Eingänge vom Typ Word ohne Konsistenz, in Bit 0-3 steht die Länge der Eingangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Word, Bit 0-3 = 1111B entspricht Länge 16 Words)
0110B	Ausgänge vom Typ Word ohne Konsistenz, in Bit 0-3 steht die Länge der Ausgangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Word, Bit 0-3 = 1111B entspricht Länge 16 Words)
0111B	Ein- und Ausgänge vom Typ Word ohne Konsistenz, in Bit 0-3 steht die Länge der Eingangs- bzw. Ausgangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Word, Bit 0-3 = 1111B entspricht Länge 16 Words)
1000B	spezielles Kennungsformat Ausgänge, es folgt ein Byte, das die zugehörigen Ausgangsdaten beschreibt (s.u.), Bit 0-3 geben an, wie viel Bytes herstellerspezifische Daten danach noch folgen
1001B	Eingänge vom Typ Byte mit Konsistenz, in Bit 0-3 steht die Länge der Eingangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Byte, Bit 0-3 = 1111B entspricht Länge 16 Bytes)
1010B	Ausgänge vom Typ Byte mit Konsistenz, in Bit 0-3 steht die Länge der Ausgangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Byte, Bit 0-3 = 1111B entspricht Länge 16 Bytes)
1011B	Ein- und Ausgänge vom Typ Byte mit Konsistenz, in Bit 0-3 steht die Länge der Eingangs- bzw. Ausgangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Byte, Bit 0-3 = 1111B entspricht Länge 16 Bytes)
1100B	spezielles Kennungsformat Ein- und Ausgänge, es folgt erst ein Byte, das die zugehörigen Ausgangsdaten, dann ein Byte, das die zugehörigen Eingangsdaten beschreibt (s.u.), Bit 0-3 geben an, wie viel Bytes herstellerspezifische Daten danach noch folgen
1101B	Eingänge vom Typ Word mit Konsistenz, in Bit 0-3 steht die Länge der Eingangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Word, Bit 0-3 = 1111B entspricht Länge 16 Words)
1110B	Ausgänge vom Typ Word mit Konsistenz, in Bit 0-3 steht die Länge der Ausgangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Word, Bit 0-3 = 1111B entspricht Länge 16 Words)
1111B	Ein- und Ausgänge vom Typ Word mit Konsistenz, in Bit 0-3 steht die Länge der Eingangs- bzw. Ausgangsdaten minus 1 (d.h. Bit 0-3 = 0000B entspricht Länge 1 Word, Bit 0-3 = 1111B entspricht Länge 16 Words)

Falls das erste Byte vom Typ "spezielles Kennungsformat" ist hat das zweite bzw. dritte Byte die folgende Bedeutung:

Bit 6-7	Bedeutung
00B	Typ Byte ohne Konsistenz, in Bit 0-5 steht die Länge der Daten minus 1 (d.h. Bit 0-5 = 000000B entspricht Länge 1 Byte, Bit 0-5 = 111111B entspricht Länge 64 Bytes)
01B	Typ Word ohne Konsistenz, in Bit 0-5 steht die Länge der Daten minus 1 (d.h. Bit 0-5 = 000000B entspricht Länge 1 Word, Bit 0-5 = 111111B entspricht Länge 64 Words)
10B	Typ Byte mit Konsistenz, in Bit 0-5 steht die Länge der Daten minus 1 (d.h. Bit 0-5 = 000000B entspricht Länge 1 Byte, Bit 0-5 = 111111B entspricht Länge 64 Bytes)
11B	Typ Word mit Konsistenz, in Bit 0-5 steht die Länge der Daten minus 1 (d.h. Bit 0-5 = 000000B entspricht Länge 1 Word, Bit 0-5 = 111111B entspricht Länge 64 Words)

5.1.8.7 Diagnosedaten - DiagData

Es folgt eine Beschreibung der DP-Diagnosedaten

Offset	Bedeutung
0x00.0	StationNonExistent: Slave beim letzten Telegramm nicht geantwortet
0x00.1	StationNotReady: Slave verarbeitet noch das Set_Prm bzw. Chk_Cfg-Telegramm
0x00.2	CfgFault: Slave meldet einen Konfigurationsfehler
0x00.3	ExtDiag: Extended DiagData sind vorhanden und gültig
0x00.4	NotSupported: Slave unterstützt ein Feature nicht, das mit Set_Prm oder Global_Control gefordert wurde
0x00.5	InvalidSlaveResponse: Slave antwortet nicht DP-konform
0x00.6	PrmFault: Slave meldet einen Parametrierfehler
0x00.7	MasterLock: Slave ist im Datenaustausch mit einem anderen Master
0x01.0	PrmReq: Slave muss neu parametriert und konfiguriert werden
0x01.1	StatDiag: Slave meldet statische Diagnose bzw. Applikation des DPV1-Slave noch nicht bereit für den Datenaustausch
0x01.2	PROFIBUS-DP-Slave
0x01.3	WdOn: DP-Watchdog ist eingeschaltet
0x01.4	FreezeMode: DP-Slave ist im Freeze-Mode
0x01.5	SyncMode: DP-Slave ist im Sync-Mode
0x01.6	reserviert
0x01.7	Deactivated: DP-Slave wurde deaktiviert
0x02.0	reserviert
0x02.1	reserviert
0x02.2	reserviert
0x02.3	reserviert
0x02.4	reserviert
0x02.5	reserviert
0x02.6	reserviert
0x02.7	ExtDiagOverflow: zu viele Extended DiagData vorhanden
0x03	MasterAdd: Stationsadresse des Masters, der mit dem Slave Datenaustausch macht
0x04,0x05	IdentNumber
ab 0x06	Extended DiagData

Extended DiagData

Bei den Extended DiagData wird zwischen Kennungsdiagnose, Kanaldiagnose und Herstellerspezifischer Diagnose unterschieden, wobei das erste Byte jeweils den Typ der Diagnose und die Länge der dazugehörenden Daten anzeigt. In den Extended DiagData können auch auch mehrere Diagnosetypen nacheinander folgen.

Header-Byte

Bit	Bedeutung
0-5	Länge der zugehörigen Diagnosedaten inklusive Header-Byte
6-7	0 = Herstellerspezifische Diagnose (DPV1 wird nicht unterstützt) bzw. DPV1-Diagnose (DPV1 wird unterstützt (DPV1_Enable = 1) in zugehöriger GSD-Datei)
	1 = Moduldiagnose
	2 = Kanaldiagnose

Herstellerspezifische Diagnose

Der Aufbau der Herstellerspezifischen Diagnose ist der Dokumentation des DP-Slaves zu entnehmen.

DPV1-Diagnose

Bei DP-Slaves, die auch DPV1 unterstützen wird statt der Herstellerspezifischen Diagnose die DPV1-Diagnose gesendet, bei der zwischen Status-Messages und Alarmen unterschieden wird:

Byte	Bedeutung
0	Header-Byte (Bit 6,7 = 0, Bit 0-5 = 4..63)
1	Bit 0-6: Alarm-Typ
	Bit 7: 0: Alarm 1: Status
2	Slot-Number (0-254)
3	Bit 0-1: Alarm-Specifier
	Bit 2: Additional-Acknowledge
	Bit 3-7: Sequence-Number
4-63	herstellerspezifisch (s. Dokumentation des DP-Slaves)

Alarm-Typ

Wert	Bedeutung
0	reserviert
1	Diagnose-Alarm
2	Prozess-Alarm
3	Ziehen-Alarm
4	Stecken-Alarm
5	Status-Alarm
6	Update-Alarm
7-31	reserviert
20-126	herstellerspezifisch (s. Dokumentation des DP-Slaves)
127	reserviert

Moduldiagnose

In der Moduldiagnose gibt es je DP-Modul ein Bit, das anzeigt, ob bei dem entsprechenden DP-Modul eine Diagnose vorliegt.

Byte	Bedeutung
0	Header-Byte (Bit 6,7 = 1, Bit 0-5 = 2..32)
1	Bit 0: 1. DP-Modul hat Diagnose
	Bit 1: 2. DP-Modul hat Diagnose
	...
	Bit 7: 8. DP-Modul hat Diagnose
...	...
31	Bit 0: 241. DP-Modul hat Diagnose
	Bit 1: 242. DP-Modul hat Diagnose
	Bit 2: 243. DP-Modul hat Diagnose
	Bit 3: 244. DP-Modul hat Diagnose (maximal 244 DP-Module möglich)

Kanaldiagnose

Mit der Kanaldiagnose wird die Diagnoseursache eines DP-Moduls genauer beschrieben.

Byte	Bedeutung
0	Header-Byte = 0x83 (3 Bytes inklusive Header, Bit 6,7 = 2)
1	Bit 0-5: Kanal-Nummer
	Bit 6-7: 0 = Reserved, 1 = Input, 2 = Output, 3 = In-/Output
2	Bit 0-4: Error-Typ
	Bit 5-7: Kanal-Typ

Error-Typ

Wert	Bedeutung
0	reserviert
1	Kurzschluss
2	Unterspannung
3	Überspannung
4	Überlast
5	Übertemperatur
6	Leitungsbruch
7	oberer Grenzwert überschritten
8	unterer Grenzwert unterschritten
9	Fehler
10-15	reserviert
16-31	herstellerspezifisch (s. Dokumentation des DP-Slaves)

Kanal-Typ

Wert	Bedeutung
0	beliebiger Typ
1	Bit
2	2 Bit
3	4 Bit
4	Byte
5	Wort
6	2 Worte
7	reserviert

Revision-Number

Der Aufbau der Revision-Number ist der Dokumentation des DP-Slaves zu entnehmen.

5.1.8.8 DPV1-Fehlercodes

Bei einem fehlerhaften DPV1-Zugriff antwortet der Slaves mit 4 Bytes Daten (alle nicht beschriebenen Werte sind nicht in der DPV1-Norm beschrieben und daher dem Handbuch des Slaves zu entnehmen).

Byte 0	DPV1-Dienst
0xD1	Data_Transport
0xD7	Initiate
0xDE	Read
0xDF	Write

Byte 1	Error_Decode
0x80	DPV1
0xFE	FMS
0xFF	HART

Byte 2		Error_Code_1
Error-Class (Bit 4-7)	Error-Code (Bit 0-3)	
0x0A	0x00	Application, Read Error
	0x01	Application, Write Error
	0x02	Application, Module Failure
	0x08	Application, Version Conflict
	0x09	Application, Feature Not Supported
0x0B	0x00	Access, Invalid Index
	0x01	Access, Write Length Error
	0x02	Access, Invalid Slot
	0x03	Access, Type Conflict
	0x04	Access, Invalid Area
	0x05	Access, State Conflict
	0x06	Access, Access Denied
	0x07	Access, Invalid Range
	0x08	Access, Invalid Parameter
	0x09	Access, Invalid Type
0x0C	0x00	Resource, Read Constrain Conflict
	0x01	Resource, Write Constrain Conflict
	0x02	Resource, Busy
	0x03	Resource, Unavailable

5.1.8.9 ADS-Error-Codes

Fehlercode	Bedeutung
0x1129	IndexOffset zu groß beim Lesen der FC310x/EL6731-Diagnosedaten
0x112B	IndexOffset zu groß beim Lesen der Slave-Diagnosedaten
0x112D	ungültige Stationsadresse beim Lesen der Slave-Diagnosedaten
0x2023	ungültiger IndexOffset beim Rücksetzen der FC310x/EL6731-Diagnosedaten
0x2024	ungültige Daten beim Rücksetzen der FC310x/EL6731-Diagnosedaten
0x2025	ungültige Datenlänge beim Rücksetzen der FC310x/EL6731-Diagnosedaten
0x2101	DPV1-C1-Read: zyklische Verbindung zu Slave noch nicht aufgebaut
0x2102	PKW-Read: nur die Datenlängen 2 und 4 sind erlaubt
0x2103	PKW-Read: Slave nicht im Datenaustausch
0x2105	PKW-Read: Slave unterstützt kein PKW
0x2106	PKW-Read: falscher IndexOffset
0x2107	PKW-Read: falsche IndexGroup
0x2109	DPV1-C1-Read: FDL-Fehler (antwortet nicht, etc.)
0x210A	DPV1-C1-Read: Syntax-Fehler (DPV1-Syntax nicht korrekt)
0x210B	DPV1-C1-Read: DPV1-Fehler (4 Byte Fehlercode in den Diagnosedaten des Slave)
0x210C	PKW-Read: Syntax-Fehler
0x210D	PKW-Read: PKW-Fehler
0x210E	PKW-Read: falscher Datentyp
0x210F	DPV1-C1-Write: zyklische Verbindung zu Slave noch nicht aufgebaut
0x2110	PKW-Write: nur die Datenlängen 2 und 4 sind erlaubt
0x2111	PKW-Write: falscher IndexOffset
0x2112	PKW-Write: Slave unterstützt kein PKW
0x2113	PKW-Write: falsche IndexGroup
0x2114	Read allgemein: falsche IndexGroup
0x2115	DPV1-C1-Write: FDL-Fehler (antwortet nicht, etc.)
0x2116	DPV1-C1-Write: Syntax-Fehler (DPV1-Syntax nicht korrekt)
0x2117	DPV1-C1-Write: DPV1-Fehler (4 Byte Fehlercode in den Diagnosedaten des Slave)
0x211C	Read allgemein: falsche IndexGroup
0x211D	SetSlaveAdress: falscher IndexOffset
0x211E	FDL-AGAG-Write: falscher IndexOffset
0x211F	FDL-AGAG-Read: falscher IndexOffset
0x2120	FDL-AGAG-Write: falsche Länge
0x2121	SetSlaveAddress: falsche Länge
0x2122	FDL-AGAG-Read: falsche Länge
0x2131	Write allgemein: falsche indexGroup
0x2132	Write allgemein: falsche indexGroup
0x2137	PKW-Read: WORD empfangen, aber Read-Datenlänge ungleich 2
0x2138	PKW-Read: DWORD empfangen, aber Read-Datenlänge ungleich 4
0x2139	PKW-Read: unbekannte AK empfangen (1,2 oder 7 erwartet)
0x213A	PKW-Read-Array: WORD empfangen, aber Read-Datenlänge ungleich 2
0x213B	PKW-Read-Array: DWORD empfangen, aber Read-Datenlänge ungleich 4
0x213C	PKW-Read-Array: unbekannte AK empfangen (4,5 oder 7 erwartet)
0x213D	PKW-Write-Array: unbekannte AK empfangen (2 oder 7 erwartet)
0x213E	PKW-Write: unbekannte AK empfangen (1 oder 7 erwartet)
0x213F	PKW-Write: unbekannte AK empfangen (2 oder 7 erwartet)
0x2140	PKW-Write-Array: unbekannte AK empfangen (1 oder 7 erwartet)
0x2142	SetSlaveAddress: falscher Parameter beim Setzen der Adresse im Slave-Mode
0x2144	falsche IndexGroup beim ReadWrite
0x2147	DPV1-C2-Initiate: MSAC_C2 ist nicht aktiviert
0x2148	falsche IndexGroup beim Read
0x2149	falsche IndexGroup beim Write

Fehlercode	Bedeutung
0x214E	DPV1-C2-Read: MSAC_C2 ist nicht aktiviert
0x214F	DPV1-C2-Write: MSAC_C2 ist nicht aktiviert
0x2150	DPV1-C2-DataTransport: MSAC_C2 ist nicht aktiviert
0x2151	DPV1-C2-Read: FDL-Fehler (antwortet nicht, etc.)
0x2152	DPV1-C2-Read: Abort der Verbindung
0x2153	DPV1-C2-Read: DPV1-Fehler (4 Byte Fehlercode in den Diagnosedaten des Slave)
0x2154	PKW-ReadNoOfElements: Länge muss gleich 1 sein
0x2155	PKW-ReadNoOfElements: PKW ist nicht aktiviert
0x2156	PKW-ReadNoOfElements: Achs-Nummer ist zu groß
0x2157	PKW-ReadNoOfElements: Slave nicht im Datenaustausch
0x2158	PKW-ReadNoOfElements: unbekannte AK empfangen (6 oder 7 erwartet)
0x215A	DPV1-C2-Write: FDL-Fehler (antwortet nicht, etc.)
0x215B	DPV1-C2-Write: Abort der Verbindung
0x215C	DPV1-C2-Write: DPV1-Fehler (4 Byte Fehlercode in den Diagnosedaten des Slave)
0x215D	DPV1-C2-DataTransport: FDL-Fehler (antwortet nicht, etc.)
0x215E	DPV1-C2-DataTransport: Abort der Verbindung
0x215F	DPV1-C2-DataTransport: DPV1-Fehler (4 Byte Fehlercode in den Diagnosedaten des Slave)
0x2163	DPV1-C2-DataTransport: falscher IndexOffset
0x2600-0x26FF	AK 7 (Fehler) bei PKW-Bearbeitung erhalten, Fehlercode in Lo-Byte
0x2700-0x27FF	Fehler bei DPV1-Bearbeitung, 4 Bytes Fehlercode in den Slave-Diagnosedaten, Byte 3 des Fehlercodes (Error-Class, Error-Code) steht in Lo-Byte

„Nicht erwartete“ Fehlercodes vom PROFIBUS Slave in 0x28xx

Fehlercode	Bedeutung
0x0	OK (wird in der Regel nicht von der Klemme weitergegeben)
0x1	UE = User Error
0x2	RR = No resources
0x3	RS = SAP not enabled
0x8***	DL = Data Low (normal case with DP)
0x9***	NR = No response data ready
0xA	DH = Data High (DP diagnosis pending)
0xC	RDL = Data not received and Data Low
0xD	RDH = Data not received and Data High

***) DL und NR werden von der EL6731 behandelt

5.1.9 Slave-Priorisierung/mehrere DP-Zyklen

Aufteilung der DP-Slaves auf mehrere DP-Zyklen (Priorisierung der Slaves)

Um in Systemen, bei denen einige Slaves sehr schnell gepollt, bei anderen aber eine größere Zykluszeit ausreichend wäre, eine möglichst geringere DP-Zykluszeit zu bekommen, können die Slaves priorisiert werden. Dabei kann je Slave angegeben werden, in jedem wievielten Zyklus (Divider unter **Data-Exch Poll-Rate** auf dem Karteireiter Features [▶ 105] des Slaves) er gepollt werden soll. Weiterhin ist es dann sinnvoll, dass z. B. in dem nachfolgend dargestellten Fall bei 4 Slaves, die jeweils nur in jedem zweiten Zyklus angesprochen werden sollen, auch einstellbar ist, dass jeweils 2 Slaves in dem einen Zyklus und 2 Slaves in dem anderen Zyklus gepollt werden, um die Gesamt-DP-Zyklus möglichst konstant zu halten. Dazu gibt es auf dem Karteireiter Features [▶ 105] des Slaves die Einstellung des **Modulo** unter Data-Exch Poll-Rate, wobei in dem dargestellten Fall Slave 3 und 5 den **Modulo** 0 und Slave 4 und 6 den **Modulo** 1 erhalten würden. Der aktuelle Modulo kann in der Variablen **ActualModulo** gelesen werden, die im Kapitel Master-Diagnose [▶ 126] beschrieben ist.

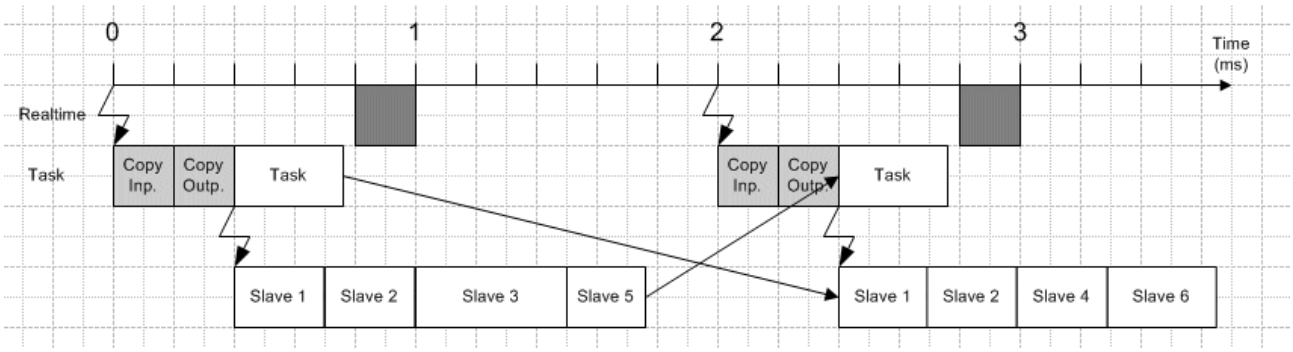


Abb. 119: Diagramm Aufteilung der DP-Slaves auf mehrere DP-Zyklen

Mehrere DP-Zyklen in einem Task-Zyklus

Bei Taskzykluszeiten, die mehr als doppelt so groß wie die DP-Zykluszeit sind, ist es möglich, mehrere DP-Zyklen innerhalb eines Task-Zyklus durchzuführen, um möglichst aktuelle Inputdaten zu bekommen. Dabei wird entsprechend des auf dem Karteireiter **"FC310x"** (für TwinCAT 2.8 bzw. TwinCAT 2.9) des Masters einstellbaren Faktors **Anzahl DP-Zyklen je Task-Zyklus** mit dem Starten des ersten DP-Zyklus ein Timer mit der Zykluszeit (Task-Zykluszeit/(Anzahl DP-Zyklen je Task-Zyklus)) gestartet, über den dann weitere DP-Zyklen gestartet werden. Dabei ist allerdings aufzupassen, dass der letzte DP-Zyklus rechtzeitig (vor dem nächsten Start der Task) beendet ist, da es sonst zu einem (bzw. mehreren, in Abhängigkeit vom Verhältnis Anzahl DP-Zyklen je Task-Zyklus) DP-Zyklusausfall kommt, wie im Kapitel Synchronisierung [▶ 49] beschrieben wurde.

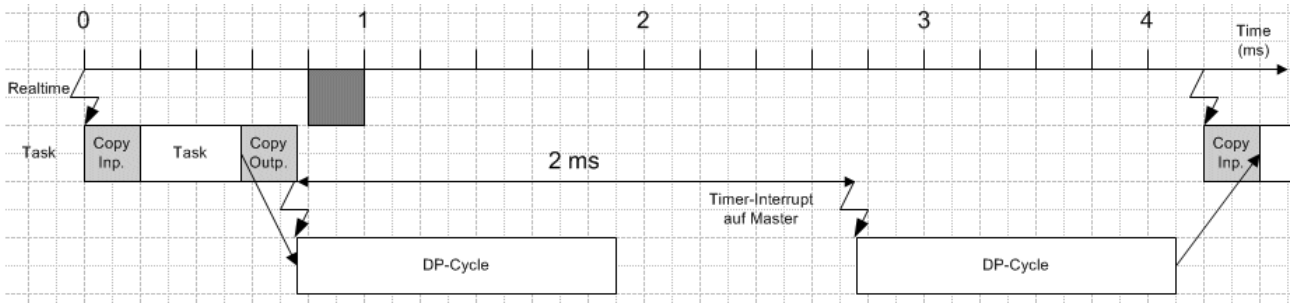


Abb. 120: Diagramm mehrere DP-Zyklen in einem Task-Zyklus

Mehrere Data-Samples innerhalb eines Task-Zyklus

Die beiden beschrieben Funktionalitäten lassen sich jetzt kombinieren, um z. B. einem oder mehreren Slaves in einem 2 ms-Zyklus jede ms neue Daten zu geben bzw. von dem Slave neue Daten zu bekommen, um eine bessere Regelqualität zu erhalten. Für diesen Fall werden die Einstellungen unter **Additional Data_Exchange Samples** auf dem Karteireiter Features [▶ 105] des Slaves statt unter **Data-Exch Poll-Rate** (wie oben beschrieben) vorgenommen.

In dem unten dargestellten Beispiel ist zunächst der Faktor **Anzahl DP-Zyklen je Task-Zyklus** auf dem Karteireiter **"FC310x"** (für TwinCAT 2.8 bzw. TwinCAT 2.9) des Masters auf 2 einzustellen. Damit die Task 2 verschiedene Werte an den Slave senden bzw. 2 verschiedene Werte von dem Slave empfangen kann, ist der entsprechende Slave zweimal im System Manager einzutragen, wobei alle Einstellungen, mit Ausnahme des **Modulo** unter **Additional Data_Exchange Samples** auf dem Karteireiter Features [▶ 105] des Slaves, gleich sein müssen. Dort wäre für den entsprechenden Slave eine 0 bei der einen und eine 1 bei der anderen Box im System Manager anzugeben. Der **Multiplier** unter **Additional Data_Exchange Samples** ist für diesen Slave bei beiden Boxen auf 2 zu stellen, d.h., jede der beiden eingetragenen Boxen wird nur in jedem 2. DP-Zyklus gepollt (der Slave wird dann in jedem DP-Zyklus gepollt, aber er ist ja zweimal eingetragen). Für alle anderen Slaves, die nur einmal innerhalb des Task-Zyklus gepollt werden sollen (die sind natürlich auch nur einmal im System Manager eingetragen), ist der **Multiplier** unter **Additional Data_Exchange Samples** ebenfalls auf 2, mit dem **Modulo** unter **Additional Data_Exchange Samples** können diese Slaves jetzt noch auf die beiden Zyklen aufgeteilt werden. Ein Slave, der zweimal gepollt, aber nur ein Variablenabbild in der Task haben soll, ist nur einmal einzufügen, der **Multiplier** wäre auf 1, der **Modulo** auf 0 eingestellt.

Im vorliegenden Beispiel wären die Slaves 1 und 2 jeweils zweimal im System Manager eingetragen mit den Einstellungen:

- Additional Data_Exchange Samples/Multiplier = 2
- Additional Data_Exchange Samples/Modulo = 0 bzw. 1

Die Slaves 3 und 5 wären nur einmal im System Manager eingetragen und hätten die Einstellungen:

- Additional Data_Exchange Samples/Multiplier = 2
- Additional Data_Exchange Samples/Modulo = 0

Die Slaves 4 und 6 wären ebenfalls nur einmal im System Manager eingetragen mit den Einstellungen:

- Additional Data_Exchange Samples/Multiplier = 2
- Additional Data_Exchange Samples/Modulo = 1

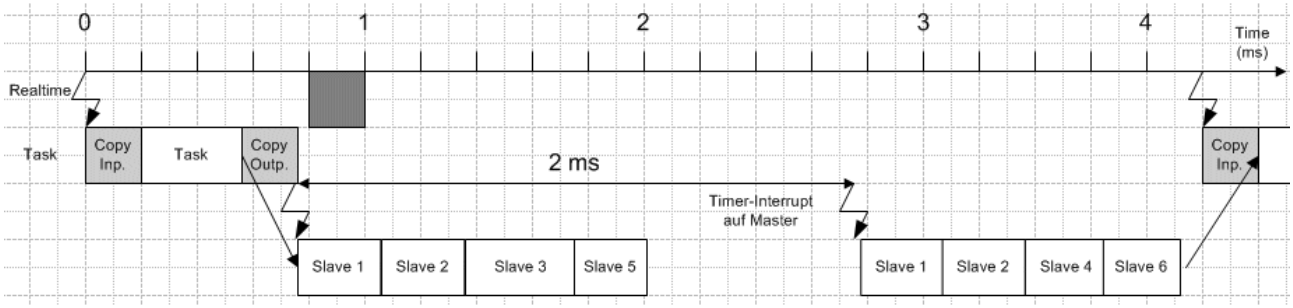


Abb. 121: Diagramm mehrere Data-Samples innerhalb eines Task-Zyklus

Für die Slaves 1 und 2 liegen die Variablen zweimal vor (jeweils 2 Boxen im System Manager), die Variablen der Box mit dem Additional Data_Exchange Samples/Modulo 0 werden zuerst gesendet bzw. empfangen.

5.2 EL6731-0010 - PROFIBUS Slave-Klemme

Die EL6731-0010 unterstützt die Protokolle PROFIBUS DP und PROFIBUS DPV1.

PROFIBUS DP

Um den Slave für den zyklischen DP-Betrieb zu konfigurieren, ist im TwinCAT System Manager wie folgt vorzugehen:

DP-Slave projektieren

Zuerst ist ein E/A-Gerät "Profibus Slave EL6731, EtherCAT" zu projektieren (mit rechter Maustaste auf "E/A-Geräte", dann "Gerät anfügen" auswählen). Dabei werden das Device und eine Box angefügt (dazu muss die GSD-Datei **"EL31095F.GSD"** im Verzeichnis "TwinCAT\Io\Profibus" stehen):

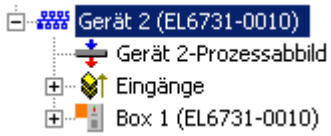


Abb. 122: Anfügen eines „Profibus Slave EL6731, EtherCAT“ im TwinCAT-Baum

Der entsprechende Kanal ist auf dem Karteireiter **"EL6731-0010"** des Devices zu suchen ("Suchen"-Button), die Stationsadresse anzupassen.

Module anfügen

Entsprechend der zyklisch zu übertragenen Daten sind Module an die Box anzufügen (mit rechter Maustaste auf die Box, dann "Module anfügen" auswählen):

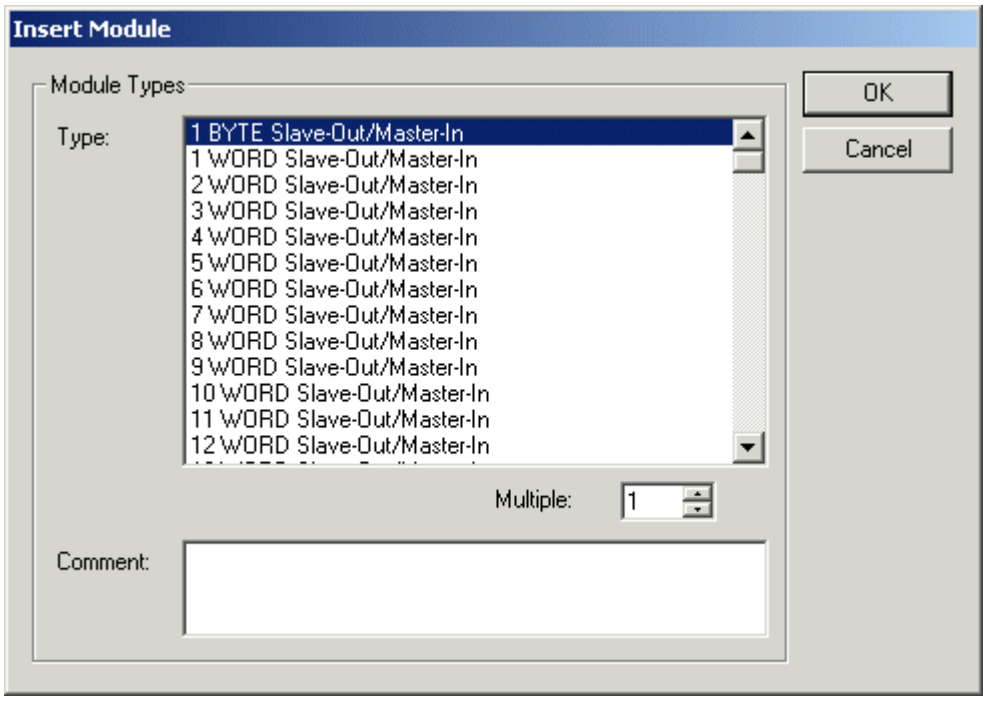


Abb. 123: Anfügen Module

Konfiguration des Masters

Zur Konfiguration des Masters ist die Datei "**EL31095F.GSE**" aus dem Verzeichnis "TwinCAT\Io\Profibus" zu nehmen. Die Module müssen im Master-Konfigurator **in der gleichen Reihenfolge** angefügt werden, wie bei der Konfiguration des Slaves im System Manager.

PROFIBUS DPV1

Der DP-Slave unterstützt eine DPV1-MSAC_C1-Server-Verbindung, die zusammen mit der zyklischen Verbindung aufgebaut wird. Diese kann benutzt werden, um neben den zyklischen Daten noch größere Mengen azyklische Daten übertragen zu können. Ein vom Master empfangenes DPV1-Read-Telegramm wird als ADS-Read-Indication an die PLC gemeldet, ein vom Master empfangenes DPV1-Write-Telegramm wird als ADS-Write-Indication an die PLC gemeldet. Das PLC-Programm ist dann für die Read bzw. Write-Antwort verantwortlich, dazu sind die Funktionen ADS-Read-Response bzw. ADS-Write-Response aufzurufen.

MSAC-C1 Read

Eine DPV1-MSAC_C1-Read-Indication wird wie folgt auf eine ADS-Read-Indication abgebildet:

ADS-Read-Indication-Parameter	Bedeutung
Source-Net-ID (NETID)	Net-ID des Slaves (s. Karteireiter ADS [▶ 100] des Devices)
Source-Port (PORT)	0x200
Invoke-ID (INVOKEID)	eindeutige Nummer, muss bei Response wieder auftauchen
IndexGroup (IDXGRP)	Slot-Number (DPV1-Parameter)
IndexOffset (IDXOFFS)	Index (DPV1-Parameter)
Length (LENGTH)	Länge der auszulesenden Daten

Eine ADS-Read-Response wird wie folgt auf eine DPV1-MSAC_C1-Read-Response abgebildet:

ADS-Read-Response-Parameter	Bedeutung
Destination-Net-ID (NETID)	Net-ID des Slaves (s. Karteireiter ADS [▶ 100] des Devices)
Destination-Port (PORT)	0x200
Invoke-ID (INVOKEID)	eindeutige Nummer, wie bei Indication
Result (RESULT)	Result des Read: 0 = kein Fehler, sonst: Bit 0-15 = Standard-ADS-Fehlercodes, Bit 16-23 = Error_Code_1, Bit 24-31 = Error_Code_2, s. Beschreibung DPV1-Fehlercodes [▶ 133]
Length (LENGTH)	Länge der gelesenen Daten
Data (DATAADDR)	gelesene Daten

MSAC-C1 Write

Eine DPV1-MSAC_C1-Write-Indication wird wie folgt auf eine ADS-Write-Indication abgebildet:

ADS-Write-Indication-Parameter	Bedeutung
Source-Net-ID (NETID)	Net-ID des Slaves (s. Karteireiter ADS [▶ 100] des Devices)
Source-Port (PORT)	0x200
Invoke-ID (INVOKEID)	eindeutige Nummer, muss bei Response wieder auftauchen
IndexGroup (IDXGRP)	Slot-Number (DPV1-Parameter)
IndexOffset (IDXOFFS)	Index (DPV1-Parameter)
Length (LENGTH)	Länge der zu schreibenden Daten
Data (DATAADDR)	zu schreibende Daten

Eine ADS-Read-Response wird wie folgt auf eine DPV1-MSAC_C1-Read-Response abgebildet:

ADS-Read-Response-Parameter	Bedeutung
Destination-Net-ID (NETID)	Net-ID des Slaves (s. Karteireiter ADS [▶ 100] des Devices)
Destination-Port (PORT)	0x200
Invoke-ID (INVOKEID)	eindeutige Nummer, wie bei Indication
Result (RESULT)	Result des Read: 0 = kein Fehler, sonst: Bit 0-15 = Standard-ADS-Fehlercode, Bit 16-23 = Error_Code_1, Bit 24-31 = Error_Code_2, s. Beschreibung DPV1-Fehlercodes [▶ 133]
Length (LENGTH)	Länge der gelesenen Daten

ADS-Interface

In TwinCAT-Systemen kann die Kommunikation auch über ADS erfolgen. Von der Funktionalität entspricht das einer ADS-Verbindung zwischen zwei PCs über Ethernet, nur dass stattdessen die Übertragung über PROFIBUS erfolgt, mit der Ausnahme, dass der Requester, der den ADS-Auftrag anstößt, immer der DP-Master-PC ist. Ein EL6731-DP-Master ist dann mit einem EL6731-DP-Slave verbunden.

Beim DP-Master ist dann auf dem Karteireiter "ADS" der Box das **ADS-Interface** zu aktivieren und die **Net-ID** des DP-Slave-PCs einzutragen:

Abb. 124: DP-Master Karteireiter ADS: Aktivieren des ADS-Interface und Eintragen der Net-ID

Beim DP-Slave ist auf dem Karteireiter "ADS" des Devices die Net-ID des DP-Master-PCs unter **Add. NetIds** einzutragen:

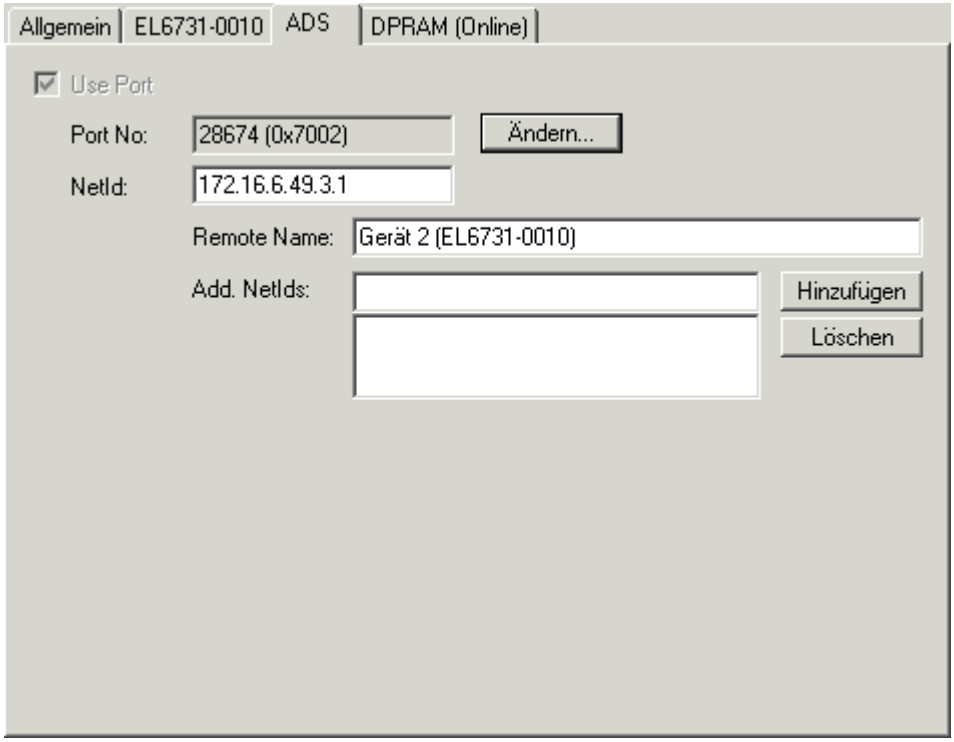


Abb. 125: DP-Slave Karteireiter ADS: Eintragen der DP-Master Net-ID mit „Add. Net-IDs“

5.2.1 EL6731-0010 - LED Beschreibung

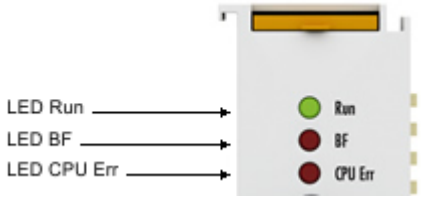


Abb. 126: LED

LED Verhalten

Anhand der LED's lassen sich die wichtigsten Zustände der Klemme schnell diagnostizieren:

LED	Farbe	Bedeutung	
RUN	grün	Diese LED gibt den Betriebszustand der Klemme wieder:	
		aus	Zustand der EtherCAT State Machine: **INIT** = Initialisierung der Klemme; **BOOTSTRAP** = Funktion für Firmware-Updates der Klemme
		Einzelblitz	Zustand der EtherCAT State Machine: **PREOP** = Funktion für Mailbox-Kommunikation und abweichende Standard-Einstellungen gesetzt
		Blinken	Zustand der EtherCAT State Machine: **SAFEOP** = Überprüfung der Kanäle des Sync-Managers und der Distributed Clocks. Ausgänge bleiben im sicheren Zustand
		an	Zustand der EtherCAT State Machine: **OP** = normaler Betriebszustand; Mailbox- und Prozessdatenkommunikation ist möglich
BF	rot	aus	DP-Slave ist im Zustand DataEXCHG und tauscht Daten mit dem Master aus
		Blinken 500ms an/ aus	Der DP-Slave wartet auf Parameter/Konfiguration (WaitPRM / WaitCFG) oder hat einen Fehler und der Datenaustausch ist gestört. Bitte den DP-State und die Diag History der Klemme prüfen.
		an	Es gibt keinen Datenaustausch zwischen Master und Slave
CPU Err	rot	an	Fehler des Prozessors der EL6731-0010
		Einzelblitz	Prozessor der EL6731-0010 startet

6 EtherCAT Kommunikation EL6731-00x0

6.1 PROFIBUS Master

6.1.1 State Machine

Die EL6731 kann auf verschiedene Weisen konfiguriert werden:

1. Konfiguration der EL6731 mit StartUp-SDOs [▶ 145]: Hierbei werden die StartUp-SDOs im EtherCAT Konfigurator berechnet und an den EtherCAT Master übergeben wie es z. B. im TwinCAT System Manager durchgeführt wird.
2. Konfiguration der EL6731 mit Backup Parameter Storage [▶ 147]: Hierbei wird die Konfiguration der DP-Slaves im Flash der EL6731 gespeichert und muss nur einmalig übertragen werden.

Konfiguration der EL6731 mit StartUp-SDOs

Das folgende Flussdiagramm zeigt den Ablauf der Konfiguration der EL6731 mit Start-SDOs:

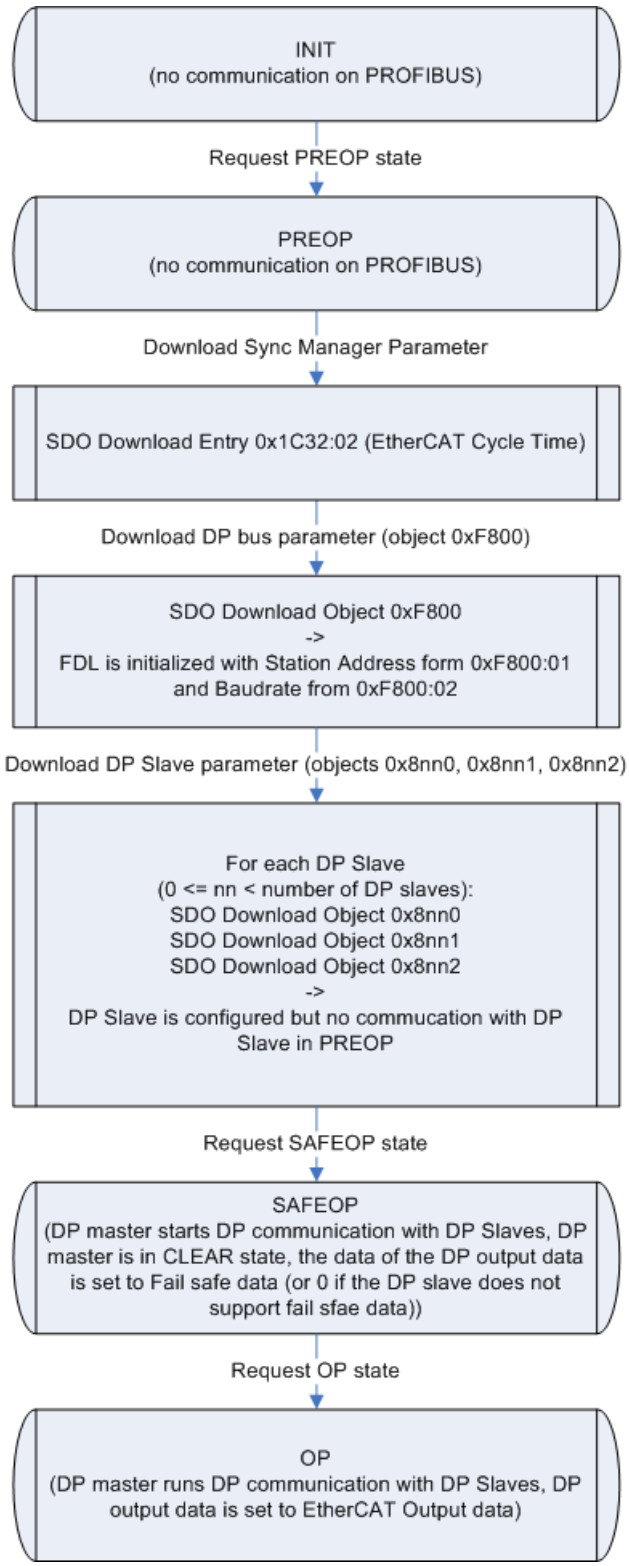


Abb. 127: Flussdiagramm: Ablauf der Konfiguration der EL6731 mit Start-SDOs

Nach einem Power-On befindet sich die EL6731 im Zustand INIT und besitzt keine DP-Konfiguration. Die EL6731 ist nicht am PROFIBUS aktiv.

DP Busparameter

Im Zustand PREOP wird die DP-Konfiguration per SDO-Download durchgeführt. Die zu ladenden Objekte müssen entweder mit Complete-Access oder mit der Konsistenzklammerung (erst Subindex 0 auf 0 setzen, dann Subindex 1-n schreiben, dann Subindex 0 auf n setzen) übertragen werden. Dabei ist zu beachten, dass immer mit dem Objekt 0xF800 [▶ 168] begonnen wird. Nach dem Empfang des Objekts 0xF800 ist die EL6731 am PROFIBUS mit der entsprechenden Baudrate aus 0xF800:02 aktiv (nur FDL, kein DP).

DP Slave Konfiguration

Nach dem Objekt 0xF800 müssen je zu konfigurierendem DP-Slave die Objekte 0x8yy0 [▶ 164], 0x8yy1 [▶ 165] und 0x8yy2 [▶ 165] in dieser Reihenfolge übertragen werden. Je zu konfigurierendem DP-Slave ist yy zu inkrementieren (beginnend bei 0).

PDO Mapping

Je konfiguriertem DP Slave gibt es eine EtherCAT RxPDO (falls der DP-Slave über DP-Outputs verfügt) und eine EtherCAT TxPDO (falls der DP Slave über DP Inputs verfügt). Das PDO-Mapping der EtherCAT PDOs wird von der EL6731 nach dem Download der jeweiligen 0x8yyz-Objekte automatisch berechnet und kann ausgelesen werden. Dabei gehören die PDO-Mappingobjekte 0x16yy und 0x1Ayy zu den Konfigurationsobjekten 0x8yyz. Die PDO-Mappingobjekte können nur mit den Werten beschrieben werden, die die EL6731 selbst berechnet hat. Das Schreiben des PDO-Mappings dient also nur zum Überprüfen des vom EtherCAT-Konfigurator berechneten PDO-Mappings und kann daher weggelassen werden.

PDO Assign

Zusätzlich gibt es noch einige EtherCAT PDOs die Control, Status und Diagnoseinformationen enthalten. Die Auswahl dieser PDOs erfolgt über das PDO-Assign. Dabei ist zu beachten, dass immer alle EtherCAT-PDOs, die den konfigurierten DP-Slaves zugeordnet sind (PDO-Nummer <= 125), im PDO-Assign auftauchen. Bei der Reihenfolge der PDOs im PDO-Assign ist zu beachten, dass mit jedem Entry im entsprechenden PDO-Assign-Objekt der Index der zugeordneten EtherCAT-PDO steigt. Wenn der EtherCAT Master kein PDO-Assign in den StartUp-SDOs überträgt, werden die PDOs 0x1A81 [▶ 156], 0x1A82 [▶ 157], 0x1A83 [▶ 157], 0x1A84 [▶ 157] und 0x1A85 [▶ 158] für Status und Diagnose zugeordnet.

Zyklische DP Kommunikation

Beim Übergang nach SAFEOP überprüft die EL6731 die in den Sync-Manager Kanälen 2 und 3 konfigurierte Länge mit der berechneten Länge aus PDO-Mapping und PDO-Assign. Der Zustand SAFEOP wird nur eingenommen, wenn diese Längen übereinstimmen. Im Zustand SAFEOP startet die EL6731 die Kommunikation mit den konfigurierten DP Slaves. Solange die EL6731 sich in SAFE-OP befindet, werden Fail Safe Data an die DP-Slaves gesendet. Sobald die EL6731 nach OP geschaltet wurde, werden die Daten aus den EtherCAT Outputs an die DP Slaves übertragen.

Konfiguration der EL6731 mit Backup Parameter Storage

Das folgende Flussdiagramm zeigt den Ablauf der Konfiguration der EL6731 mit Backup Parameter Storage:

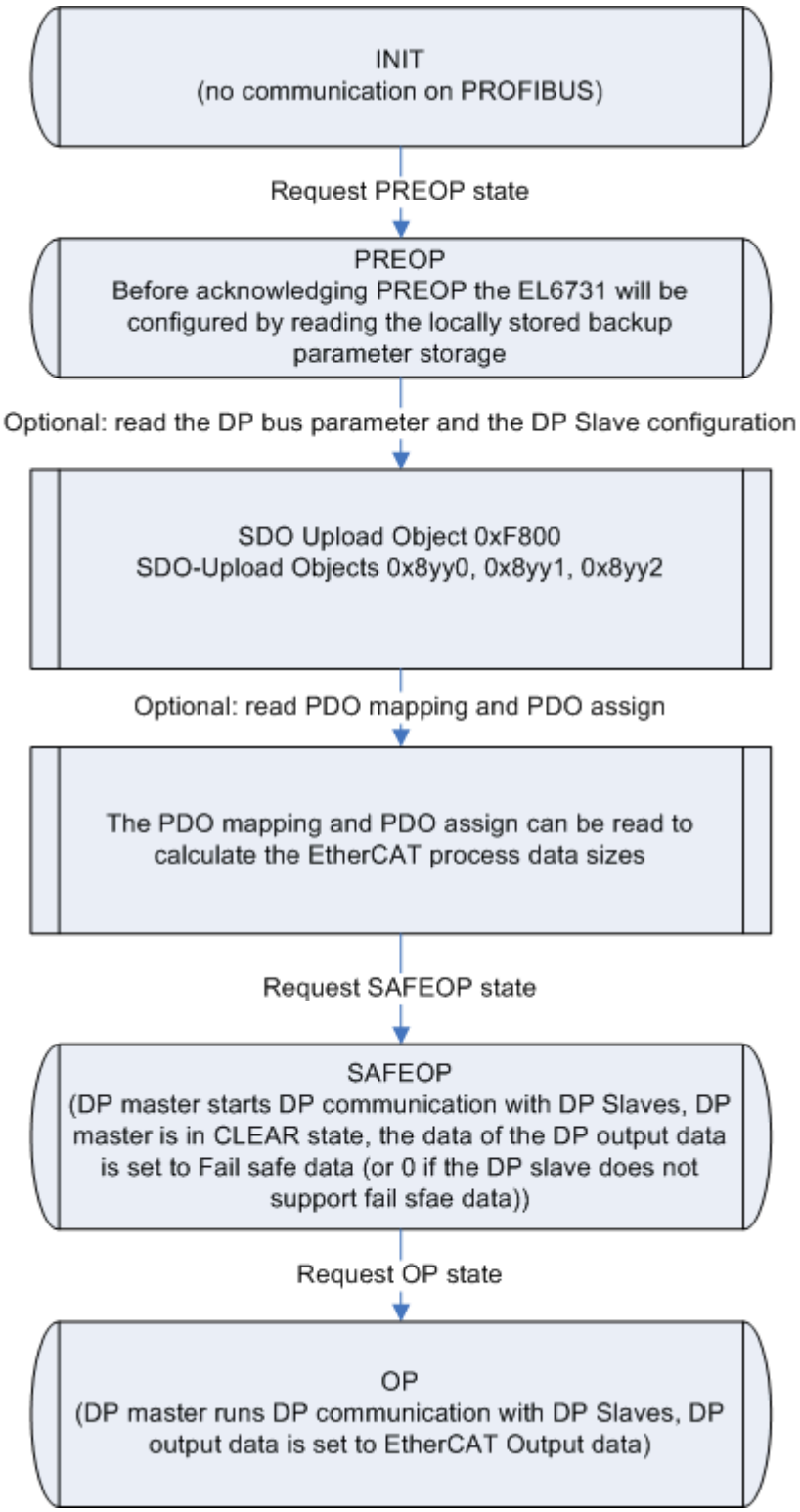


Abb. 128: Flussdiagramm: Ablauf der Konfiguration der EL6731 mit Backup Parameter Storage

Nach einem Power-On befindet sich die EL6731 im Zustand INIT und besitzt keine DP-Konfiguration. Die EL6731 ist nicht am PROFIBUS aktiv.

DP Busparameter / DP Slave Konfiguration

Beim Übergang von INIT nach PREOP wird die im Backup Parameter Storage Objekt 0x10F2 gespeicherte Konfiguration geladenen. Da im Backup Parameter Storage Objekt die StartUp-SDOs aus der Konfiguration der EL6731 mit StartUp-SDOs [▶ 145] gespeichert sind, entspricht der Ablauf dem dort beschriebenen. Zunächst wird also das Objekt 0xF800 mit den gespeicherten Daten beschrieben und die EL6731 ist am PROFIBUS mit der entsprechenden Baudrate aus 0xF800:02 aktiv. Anschließend werden die DP-Slaves entsprechend der gespeicherten DP Slave Konfiguration erzeugt. Wenn der Zustand PREOP quittiert ist, kann die aktuelle DP Konfiguration in den Objekten 0xF800, 0x8yy0, 0x8yy1 und 0x8yy2 ausgelesen werden. Bevor die EL6731 nach SAFE-OP geschaltet wird, muss noch die Zykluszeit des EtherCAT Masters (0x1C32:02) übertragen werden.

PDO Mapping / PDO Assign

Außerdem kann der EtherCAT Master im Zustand PREOP auch das PDO mapping und PDO assign auslesen, um die Längen der EtherCAT process data zu ermitteln.

Erzeugen des Backup Parameter Storage

Das Backup Parameter Storage kann wie folgt erzeugt werden:

1. Download des Objekts 0x10F2 (in PREOP ohne dass vorher StartUp-SDOs gesendet wurden): In diesem Fall werden die empfangenen Daten als Backup Parameter Storage im Flash gespeichert

Löschen des Backup Parameter Storage

Um ein neues Backup Parameter Storage zu laden oder das vorhandene einfach nur zu löschen, ist der Entry 0x1011:01 mit dem Wert 0x64616F6C zu beschreiben.

6.1.2 Synchronisierung

Bei der EL6731 ist den DP-Zyklus mit dem EtherCAT-Zyklus synchronisiert. Die Synchronisierung erfolgt per Default über das Sync Manager 2 Event, falls es keine EtherCAT Output Process Data gibt, über das Sync Manager 3 Event. Alternativ kann die EL6731 auch im Distributed Clocks Mode betrieben werden, dann erfolgt die Synchronisierung über das SYNC0- bzw. das SYNC1-Event.

SM-Synchron

Das folgende Bild zeigt den Ablauf des DP-Zyklusses, wenn keine Distributed Clocks benutzt werden.

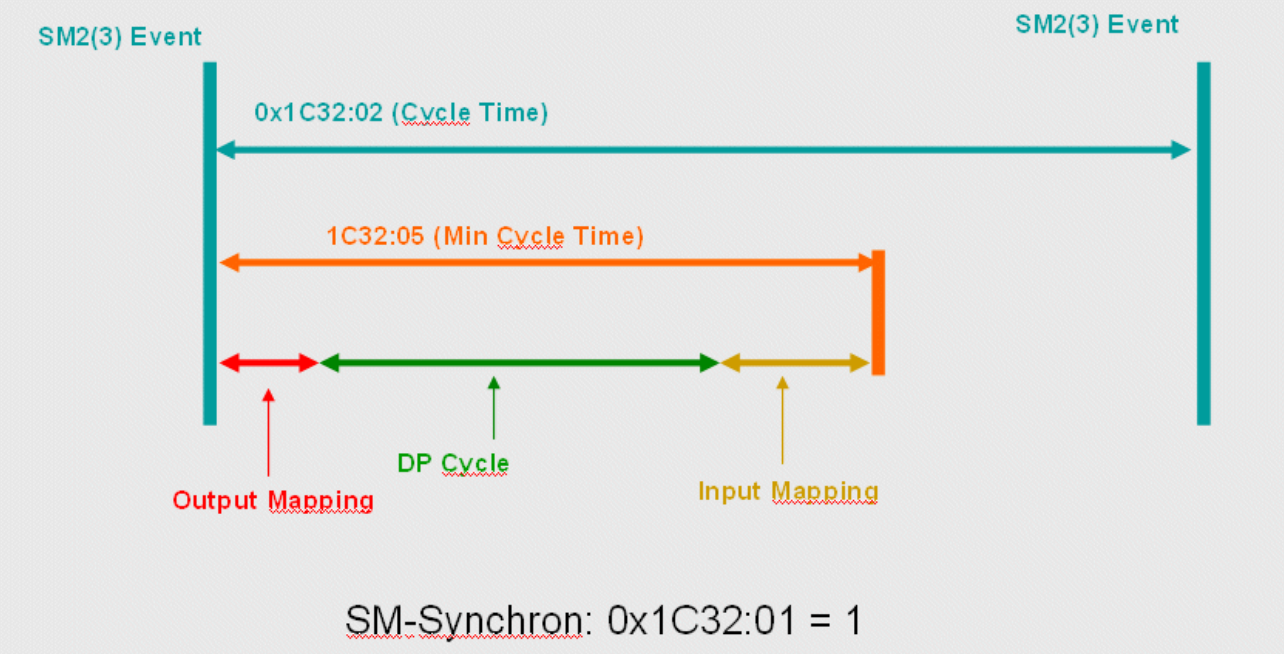


Abb. 129: Ablauf DP-Zyklus ohne Distributed Clocks

Beim Empfang des EtherCAT-Prozessdatentelegramms wird das SM2 Event (SM3, falls keine EtherCAT Output Data konfiguriert sind, also nur DP-Slaves ohne DP-Outputs) vom EtherCAT-Slave Controller generiert und damit der DP-Zyklus gestartet. Dabei wird je konfiguriertem DP-Slave (der sich im Zustand Data-Exchange befindet) ein Data-Exchange-Telegramm mit den über EtherCAT empfangenen Output Process Data gesendet. Die Reihenfolge entspricht dabei die Reihenfolge der Konfiguration, d.h. es wird mit dem in den Objekten 0x800z konfigurierten DP-Slave angefangen. Wenn alle konfigurierten DP-Slaves behandelt wurden, werden die EtherCAT Input Data aktualisiert und der DP-Zyklus ist fertig. Wenn das nächste SM2 (SM3)-Event empfangen wird, bevor der DP-Zyklus fertig ist, wird der Cycle Exceed Counter (0x1C32:0B bzw. 0x1C33:0B) inkrementiert und ein DP-Zyklus ausgelassen.

SYNC0-Synchron

Der DP-Zyklus wird durch das SM2 (SM3)-Event gestartet. Vor dem Senden des ersten Telegramms wird auf das SYNC0-Event gewartet, so dass das Senden des Global-Control-Telegramms mit einem Jitter von maximal 500 ns erfolgt. Der weitere Ablauf des DP-Zyklus entspricht dem bei der Synchronisation ohne Distributed Clocks.

Übertragung der Process Data mit LRW Telegramm (Separate Input Update = FALSE)

Das folgende Bild zeigt den Ablauf des DP-Zyklus und die Bedeutung der Sync Manager Parameter, wenn der DP-Zyklus mit Distributed Clocks über SM- und SYNC0-Event gesteuert wird.

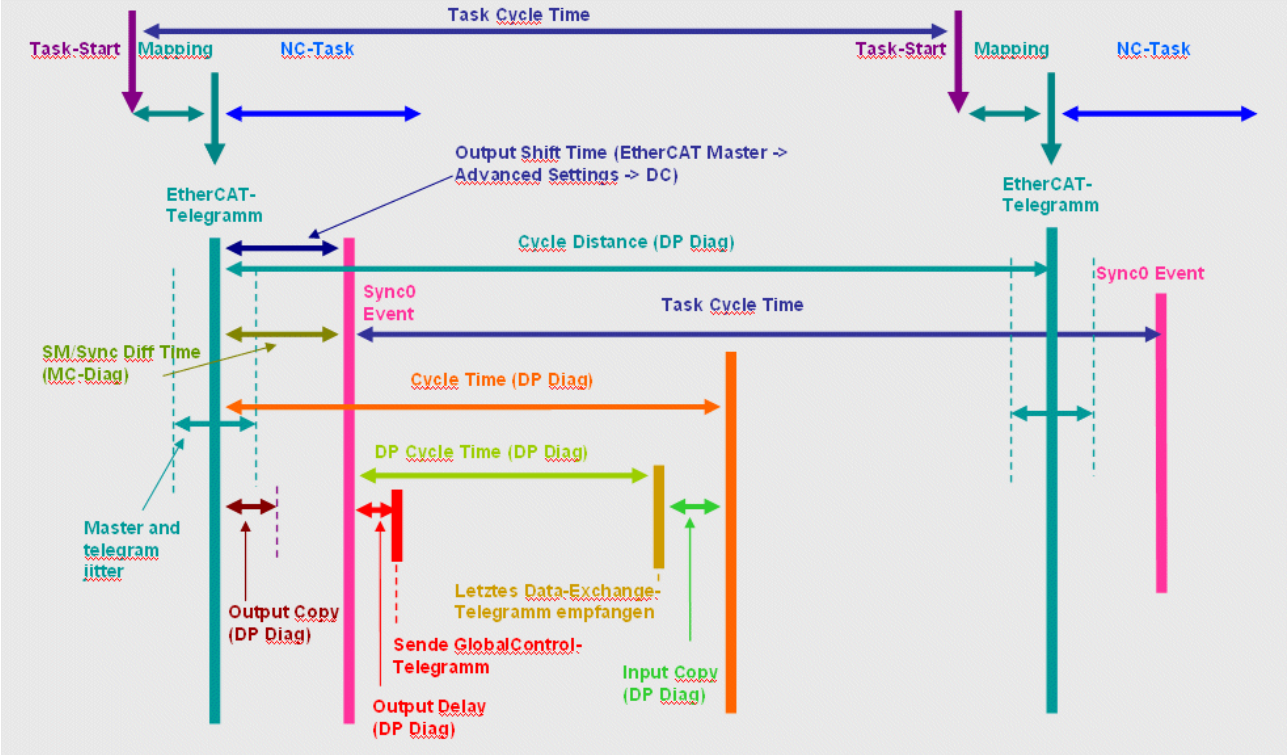


Abb. 130: Ablauf DP-Zyklus mit Distributed Clocks und Steuerung über SM- und SYNC0-Event

Übertragung der Output Data mit LWR und der Input Data mit LRD Telegramm (Separate Input Update = TRUE, Task Cycle Time = Base Time)

Falls EtherCAT Outputs und Inputs mit getrennten Telegrammen übertragen werden, damit die Inputs möglichst aktuell sind (Separate Input Update bei der zugehörigen TwinCAT-Task im TwinCAT System Manager anklicken), ist weniger Luft für den DP-Zyklus. Wenn die Task Cycle Time (= EtherCAT Master Cycle Time) gleich der TwinCAT Realtime Base Time ist, wird entsprechend des eingestellten CPU-Limits das LRD-Telegramm gesendet, mit dem die Inputs möglichst spät eingelesen werden.

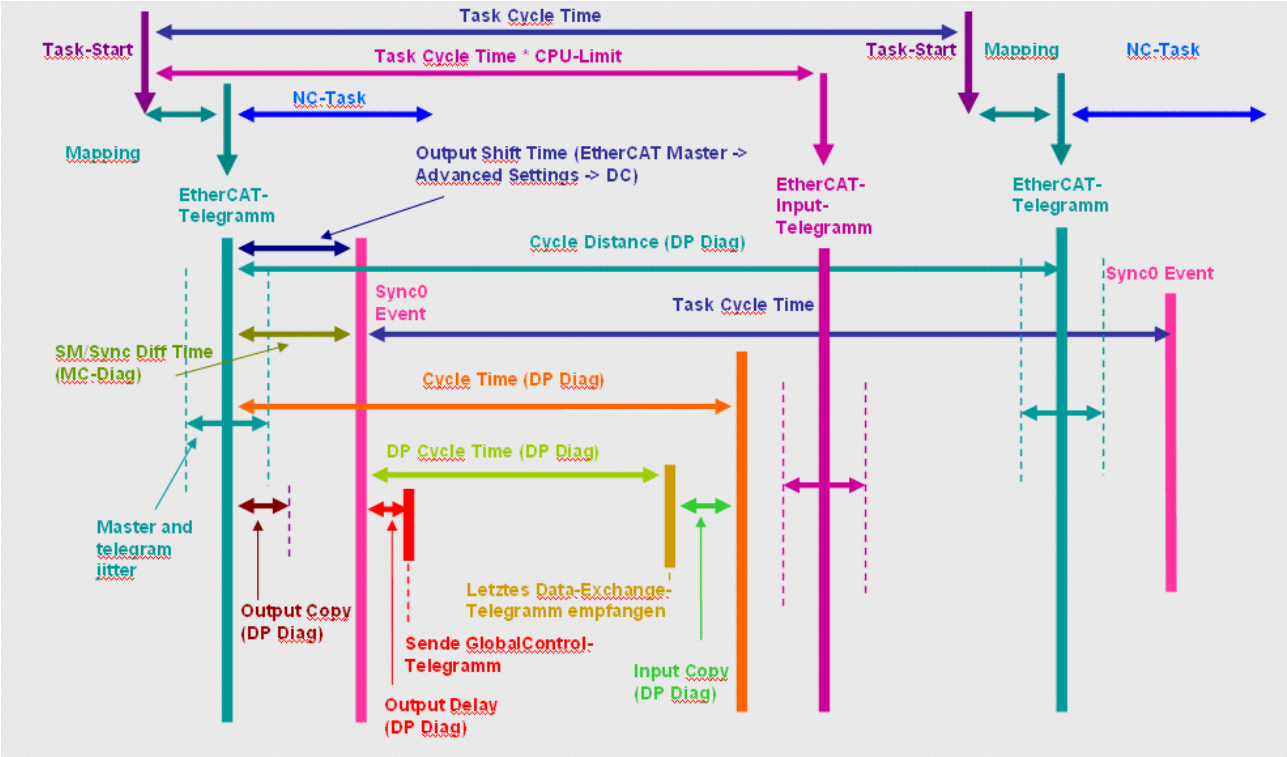


Abb. 131: Übertragung Output Data mit LWR, Input Data mit LRD Telegramm (Separate Input Update = TRUE, Task Cycle Time = Base Time)

Übertragung der Output Data mit LWR und der Input Data mit LRD Telegramm (Separate Input Update = TRUE, Task Cycle Time = 4*Base Time, Pre ticks = 1)

Wenn die Task Cycle Time (= EtherCAT Master Cycle Time) größer als die TwinCAT Realtime Base Time ist, wird entsprechend des eingestellten Pre ticks das LRD-Telegramm zu einem vorherigen Base Time Zeitpunkt gesendet.

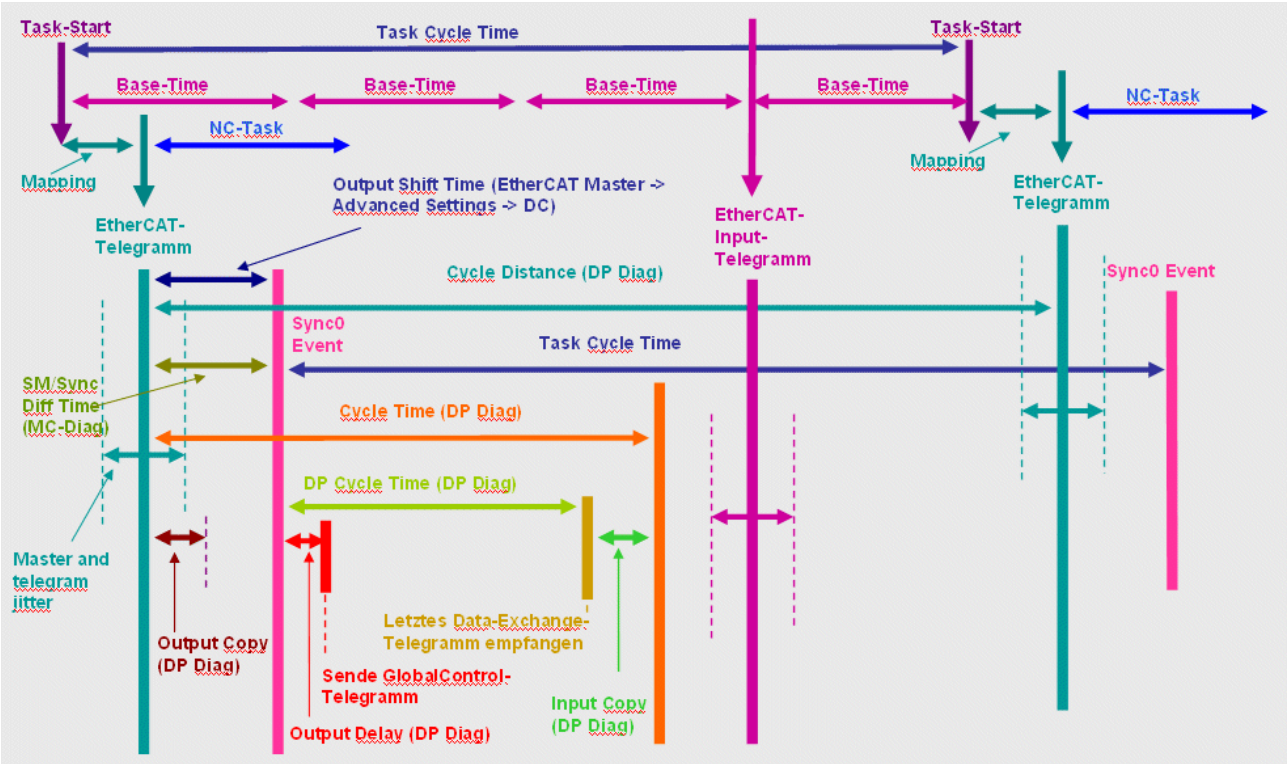


Abb. 132: Übertragung Output Data mit LWR, Input Data mit LRD Telegramm (Separate Input Update = TRUE, Task Cycle Time = 4*Base Time, Pre ticks = 1)

SYNC1-Synchron

Der DP-Zyklus wird durch den SYNC0-Event gestartet. Vor dem Senden des ersten Telegramms wird auf den SYNC1-Event gewartet, so dass das Senden des Global-Control-Telegramms mit einem Jitter von maximal 500ns erfolgt. Der weitere Ablauf des DP-Zyklus entspricht dem bei der Synchronisation ohne Distributed Clocks.

Die Einstellungen für Separate Input Update gelten wie bei der Synchronisierung mit SYNC0, daher ist hier nur der Fall ohne Separate Input Update dargestellt:

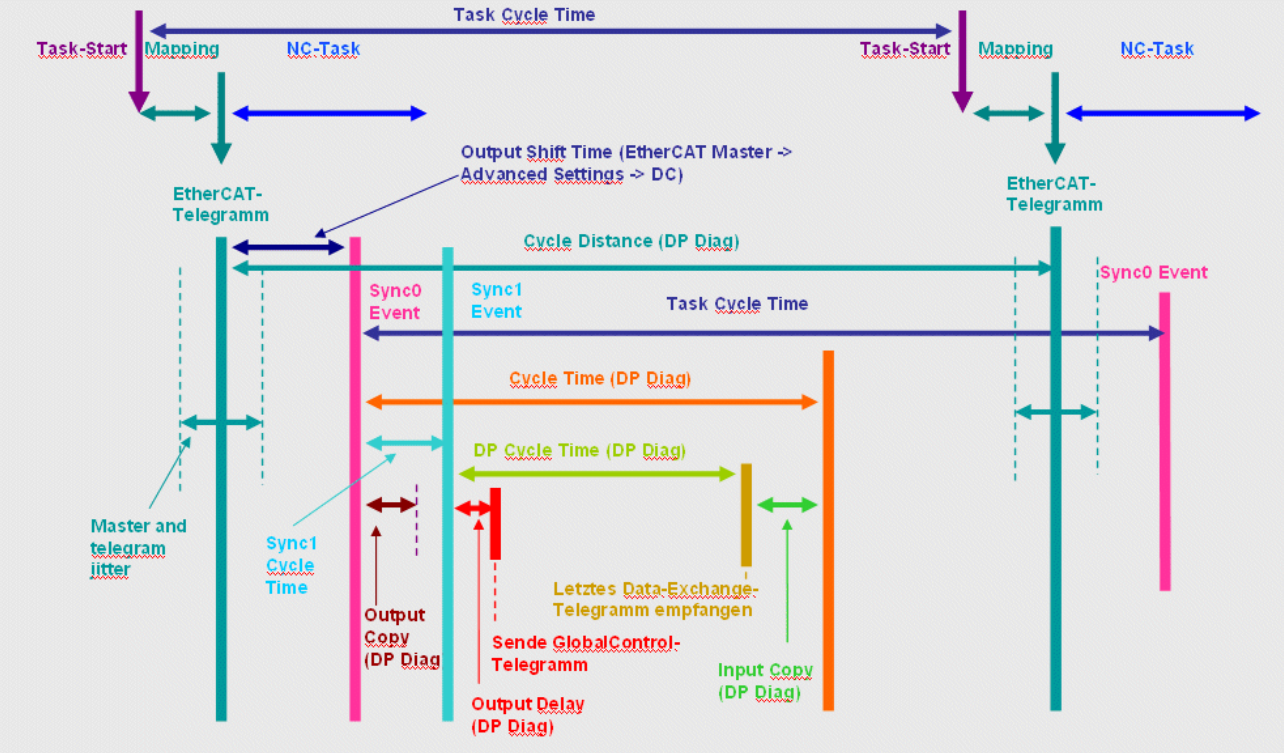


Abb. 133: Starten des DP-Zyklus durch SYNC0, Senden des ersten Telegrams nach SYNC1-Event

6.1.3 Objektbeschreibung und Parametrierung

EtherCAT XML Device Description

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

Parametrierung über das CoE-Verzeichnis (CAN over EtherCAT)

Die Parametrierung des EtherCAT Gerätes wird über den CoE-Online Reiter (mit Doppelklick auf das entsprechende Objekt) bzw. über den Prozessdatenreiter (Zuordnung der PDOs) vorgenommen. Beachten Sie bei Verwendung/Manipulation der CoE-Parameter die allgemeinen CoE-Hinweise [▶ 39]:

- StartUp-Liste führen für den Austauschfall
- Unterscheidung zwischen Online/Offline Dictionary, Vorhandensein aktueller XML-Beschreibung
- „CoE-Reload“ zum Zurücksetzen der Veränderungen

6.1.3.1 Standardobjekte (0x1000-0x1FFF)

Die Standardobjekte haben für alle EtherCAT-Slaves die gleiche Bedeutung.

Index 1000 Device type

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1000:0	Device type	Geräte-Typ des EtherCAT-Slaves: Das Lo-Word enthält das verwendete CoE Profil (5001). Das Hi-Word enthält das Modul Profil entsprechend des Modular Device Profile.	UINT32	RO	0x0C1C1389 (203166601_{dez})

Index 1008 Device name

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1008:0	Device name	Geräte-Name des EtherCAT-Slave	STRING	RO	EL6731

Index 1009 Hardware version

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1009:0	Hardware version	Hardware-Version des EtherCAT-Slaves	STRING	RO	06

Index 100A Software version

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
100A:0	Software version	Firmware-Version des EtherCAT-Slaves	STRING	RO	10

Index 1011 Restore default parameters

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1011:0	Restore default parameters	Herstellen der Defaulteinstellungen	UINT8	RO	0x01 (1_{dez})
1011:01	SubIndex 001	Wenn Sie dieses Objekt im Set Value Dialog auf **"0x64616F6C"** setzen, werden alle Backup Objekte wieder in den Auslieferungszustand gesetzt.	UINT32	RW	0x00000000 (0_{dez})

Index 1018 Identity

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	0x1A4B3052 (441135186_{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	0x00000000 (0_{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})

Index 10F0 Backup parameter handling

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
10F0:0	Backup parameter handling	Informationen zum standardisierten Laden und Speichern der Backup Entries	UINT8	RO	0x01 (1_{dez})
10F0:01	Checksum	Checksumme über das Backup Parameter Storage (Objekt 0x10F2 [▶ 153], Word 2-3)	UINT32	RO	0x00000000 (0_{dez})

Index 10F2 Backup parameter storage

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
10F2:0	Backup parameter storage	Wenn dieses Objekt verwendet wird, dürfen keine StartUp-SDOs im Zustand PREOP gesendet werden, da das Backup Parameter Storage die kompletten StartUp-SDOs enthält (s. Konfiguration der EL6731 mit Backup Parameter Storage [▶ 147]). 5s nach dem Flashen des Backup Parameter Storage wird die EL6731 neu gebootet (schaltet nach INIT mit AL-Status-Code = 0x60). die Daten haben die folgende Bedeutung:	OCTET-STRING[n]	RW	

Word-Offset	Beschreibung
0	Kommando: mit 0xC0DE werden die empfangenen Daten im Flash gespeichert
1	Länge der Daten ab Word-Offset 4 in Bytes
2-3	Checksumme, die lokal berechnet wird
4	Index des Objekts der 1. StartUp-SDO
5	len1: Länge des Objekts der 1.StartUp-SDO als CompleteAccess (ab SubIndex 0) in Bytes
6-n1	Daten des Objekts der 1. StartUp-SDO als CompleteAccess (n1 = 2*((len1+1)/2)+5)
n1+1	Index des Objekts der 2. StartUp-SDO
n1+2	len2: Länge des Objekts der 2.StartUp-SDO als CompleteAccess (ab SubIndex 0) in Bytes
(n1+3)-n2	Daten des Objekts der 2. StartUp-SDO als CompleteAccess (n2 = 2*((len2+1)/ 2)+n1+2)
...	
m	Index des Objekts der 3. StartUp-SDO
m+1	len3: Länge des Objekts der 3.StartUp-SDO als CompleteAccess (ab SubIndex 0) in Bytes
(m+2)-n3	Daten des Objekts der 3. StartUp-SDO als CompleteAccess (n3 = 2*((len3+1)/2)+m+1)

Index 10F3 Diagnosis History

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
10F3:0	Diagnosis History	Maximaler Subindex	UINT8	RO	0x45 (69_{dez})
10F3:01	Maximum Messages	Maximale Anzahl der gespeicherten Nachrichten Es können maximal 50 Nachrichten gespeichert werden	UINT8	RO	0x00 (0_{dez})
10F3:02	Newest Message	Subindex der neusten Nachricht	UINT8	RO	0x00 (0_{dez})
10F3:03	Newest Acknowledged Message	Subindex der letzten bestätigten Nachricht	UINT8	RW	0x00 (0_{dez})
10F3:04	New Messages Available	Zeigt an, wenn eine neue Nachricht verfügbar ist	BOOLEAN	RO	0x00 (0_{dez})
10F3:05	Flags	ungenutzt	UINT16	RW	0x0000 (0_{dez})
10F3:06	Diagnosis Message 001	Nachricht 1	OCTET-STRING[20]	RO	{0}
...	...	...	...	...	...
10F3:45	Diagnosis Message 064	Nachricht 64	OCTET-STRING[20]	RO	{0}

Index 10F8 Actual Time Stamp

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
10F8:0	Actual Time Stamp	Zeitstempel	UINT64	RO	

Index 1600-167C RxPDO-Map DP-Slave yyy

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1600+n:0	RxPDO-Map DP-Slave yyy	Je konfiguriertem DP-Slave gibt es eine RxPDO, die die DP Output Data des DP Slaves enthält. Je in Objekt 0x8002 [▶ 165]+(n*16) konfiguriertem DP-Modul mit DP Output Data gibt es einen Entry in dem RxPDO-Mapping Objekt 0x1600+n. Falls ein DP-Slave keine DP Output Data enthält, dann existiert das PDO-Mapping-Objekt 0x1600+n nicht. Diese PDOs sind mandatory und müssen abhängig von den konfigurierten DP Slaves immer im PDO-Assign Objekt 0x1C12 [▶ 159] enthalten sein. SubIndex 0 enthält die Anzahl der DP Module mit DP Output Data des (n+1). konfigurierten DP-Slave.	UINT8	RO	
(1600+n):01		erstes DP-Modul mit DP Output Data des (n+1). konfigurierten DP Slaves	UINT32	RO	
...		..			
(1600+n):FF		letztes DP-Modul mit DP Output Data des (n+1). konfigurierten DP Slaves	UINT32	RO	

Index 167F DPM RxPDO-Map Control

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
167F:0	DPM RxPDO-Map Control	PDO Mapping RxPDO 128	UINT8	RO	0x02 (2_{dez})
167F:01	SubIndex 001	1. PDO Mapping entry (object 0xF200, entry 0x01)	UINT32	RO	0xF200:01, 1
167F:02	SubIndex 002	2. PDO Mapping entry (15 bits align)	UINT32	RO	0x0000:00, 15

Index 187F DPM TxPDO-Par Status PDO

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
187F:0	DPM TxPDO-Par Status PDO	PDO Parameter TxPDO 128	UINT8	RO	0x06 (6_{dez})
187F:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 128 übertragen werden dürfen	OCTET-STRING[10]	RO	80 1A 00 00 00 00 00 00 00 00

Index 1880 DPM TxPDO-Par Status PDO

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1880:0	DPM TxPDO-Par Status PDO	PDO Parameter TxPDO 129	UINT8	RO	0x06 (6_{dez})
1880:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 129 übertragen werden dürfen	OCTET-STRING[10]	RO	81 1A 82 1A 83 1A 84 1A 85 1A

Index 1881 DPM TxPDO-Par PDO State

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1881:0	DPM TxPDO-Par PDO State	PDO Parameter TxPDO 130	UINT8	RO	0x06 (6_{dez})
1881:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 130 übertragen werden dürfen	OCTET-STRING[10]	RO	80 1A 00 00 00 00 00 00 00 00

Index 1882 DPM TxPDO-Par PDO Toggle

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1882:0	DPM TxPDO-Par PDO Toggle	PDO Parameter TxPDO 131	UINT8	RO	0x06 (6_{dez})
1882:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 131 übertragen werden dürfen	OCTET-STRING[10]	RO	80 1A 00 00 00 00 00 00 00 00

Index 1883 DPM TxPDO-Par DP Diag

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1883:0	DPM TxPDO-Par DP Diag	PDO Parameter TxPDO 132	UINT8	RO	0x06 (6_{dez})
1883:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 132 übertragen werden dürfen	OCTET-STRING[10]	RO	80 1A 00 00 00 00 00 00 00 00

Index 1884 DPM TxPDO-Par DP Status

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1884:0	DPM TxPDO-Par DP Status	PDO Parameter TxPDO 133	UINT8	RO	0x06 (6_{dez})
1884:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 133 übertragen werden dürfen	OCTET-STRING[10]	RO	80 1A 00 00 00 00 00 00 00 00

Index 1885 DPM TxPDO-Par DP-Master Diag

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1885:0	DPM TxPDO-Par DP-Master Diag	PDO Parameter TxPDO 134	UINT8	RO	0x06 (6_{dez})
1885:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 134 übertragen werden dürfen	OCTET-STRING[10]	RO	80 1A 00 00 00 00 00 00 00 00

Index 1886 DPM TxPDO-Par Cycle Statistics

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1886:0	DPM TxPDO-Par Cycle Statistics	PDO Parameter TxPDO 135	UINT8	RO	0x06 (6_{dez})
1886:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 135 übertragen werden dürfen	OCTET-STRING[10]	RO	80 1A 87 1A 00 00 00 00 00 00

Index 1887 DPM TxPDO-Par DC Cyclic Statistics

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1887:0	DPM TxPDO-Par DC Cyclic Statistics	PDO Parameter TxPDO 136	UINT8	RO	0x06 (6_{dez})
1887:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 136 übertragen werden dürfen	OCTET-STRING[10]	RO	80 1A 86 1A 00 00 00 00 00 00

Index 1A00-1A7C DPS TxPDO-Map Slave yyy

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A00+n:0	DPS TxPDO-Map Slave yyy	Je konfiguriertem DP-Slave gibt es eine TxPDO, die die DP Input Data des DP Slaves enthält. Je in Objekt 0x8002 [▶ 165]+(n*16) konfiguriertem DP-Modul mit DP Input Data gibt es einen Entry in dem TxPDO-Mapping Objekt 0x1A00+n. Falls ein DP-Slave keine DP Input Data enthält, dann existiert das PDO-Mapping-Objekt 0x1A00+n nicht. Diese PDOs sind mandatory und müssen abhängig von den konfigurierten DP Slaves immer im PDO-Assign Objekt 0x1C13 [▶ 159] enthalten sein. SubIndex 0 enthält die Anzahl der DP Module mit DP Input Data des (n+1). konfigurierten DP-Slave.	UINT8	RO	
(1A00+n):01		erstes DP-Modul mit DP Input Data des (n+1). konfigurierten DP Slaves	UINT32	RO	
...		..			
(1A00+n):FF		letztes DP-Modul mit DP Input Data des (n+1). konfigurierten DP Slaves	UINT32	RO	

Index 1A7F DPM TxPDO-Map Status PDO

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A7F:0	DPM TxPDO-Map Status PDO	PDO Mapping TxPDO 128	UINT8	RO	0x03 (3_{dez})
1A7F:01	SubIndex 001	1. PDO Mapping entry (object 0xF100, entry 0x01)	UINT32	RO	0xF100:01, 8
1A7F:02	SubIndex 002	2. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1A7F:03	SubIndex 003	3. PDO Mapping entry (object 0xF100, entry 0x0F)	UINT32	RO	0xF100:0F, 1

Index 1A80 DPM TxPDO-Map PDO Status

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A82:0	DPM TxPDO-Map PDO Status	Diese PDO ist zugeordnet, wenn das legacy mapping (kein MDP) aktiv ist	UINT8	RO	
1A82:01		erster PDO-Mapping Entry beim Legacy Mapping	UINT32	RO	
...					
1A82:m		letzter PDO-Mapping Entry beim Legacy Mapping	UINT32	RO	

Index 1A81 DPM TxPDO-Map PDO State

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A81:0	DPM TxPDO-Map PDO State	In dieser PDO gibt es je konfiguriertem DP Slave gibt ein Bit, das gesetzt ist, wenn sich dieser nicht im Datenaustausch befindet (eine genauere Fehlerursache steht für den m. konfigurierten DP Slave in 0xF102 [▶ 167]:m). Wenn das Bit gesetzt ist, sind die Daten der zugehörigen TxPDO m zu ignorieren. Diese PDO ist optional.	UINT8	RO	
1A81:01		PDO State des ersten konfigurierten DP Slave (konfiguriert über die Objekte 0x800z)	UINT32	RO	
...					
1A81:m		PDO State des letzten (m.) konfigurierten DP Slave (konfiguriert über die Objekte 0x800z+(m-1)*16 (1 <= m <= 127))	UINT32	RO	

Index 1A82 DPM TxPDO-Map PDO Toggle

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A82:0	DPM TxPDO-Map PDO Toggle	In dieser PDO gibt es je konfiguriertem DP Slave gibt ein Bit, das jedes mal getoggelt, wenn neue DP Input Data von dem DP Slave empfangen und in den EtherCAT Input Data aktualisiert wurden. Diese PDO ist optional.	UINT8	RO	
1A82:01		PDO Toggle des ersten konfigurierten DP Slave (konfiguriert über die Objekte 0x800z)	UINT32	RO	
...					
1A82:m		PDO Toggle des letzten (m.) konfigurierten DP Slave (konfiguriert über die Objekte 0x800z+(m-1)*16 (1 <= m <= 127))	UINT32	RO	

Index 1A83 DPM TxPDO-Map DP Status

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A83:0	DPM TxPDO-Map DP Status	In dieser PDO gibt es je konfiguriertem DP Slave gibt ein Byte, das den Kommunikationsstatus (Objekt 0xF102 [▶ 167]) zu dem DP Slave enthält. Diese PDO ist optional.	UINT8	RO	Anzahl der konfigurierten DP Slaves
1A83:01		Node State des ersten konfigurierten DP Slave (konfiguriert über die Objekte 0x800z)	UINT32	RO	0xF102:01,8
...					
1A83:m		Node State des letzten (m.) konfigurierten DP Slave (konfiguriert über die Objekte 0x800z+(m-1)*16 (1 <= m <= 127))	UINT32	RO	0xF102:m,8

Index 1A84 DPM TxPDO-Map DP Diag

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A84:0	DPM TxPDO-Map DP Diag	In dieser PDO gibt es je konfiguriertem DP Slave gibt ein Bit, das gesetzt ist, wenn sich Diagnoseinformationen (Objekt 0xF103) geändert haben. Diese PDO ist optional.	UINT8	RO	Anzahl der konfigurierten DP Slaves
1A84:01		Diag Flag des ersten konfigurierten DP Slave (konfiguriert über die Objekte 0x800z)	UINT32	RO	0xF103:01,1
...					
1A84:FF		Diag Flag des letzten (m.) konfigurierten DP Slave (konfiguriert über die Objekte 0x800z+(m-1)*16 (1 <= m <= 127))	UINT32	RO	0xF103:m,1

Index 1A85 DPM TxPDO-Map DP-Master Diag

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A85:0	DPM TxPDO-Map DP-Master Diag	PDO Mapping TxPDO 134 enthält den DP Master Status (Objekt 0xF101 [▶ 166])	UINT8	RW	0x0D (13_{dez})
1A85:01	SubIndex 001	1. PDO Mapping entry (object 0xF101 (DP Master Status data), entry 0x01 (Bus Error Counter))	UINT32	RW	0xF101:01, 16
1A85:02	SubIndex 002	2. PDO Mapping entry (object 0xF101 (DP Master Status data), entry 0x02 (Cycle Counter))	UINT32	RW	0xF101:02, 16
1A85:03	SubIndex 003	3. PDO Mapping entry (object 0xF101 (DP Master Status data), entry 0x03 (Slave Status Counter))	UINT32	RW	0xF101:03, 16
1A85:04	SubIndex 004	4. PDO Mapping entry (object 0xF101 (DP Master Status data), entry 0x04 (Cycle Time))	UINT32	RW	0xF101:04, 16
1A85:05	SubIndex 005	5. PDO Mapping entry (object 0xF101 (DP Master Status data), entry 0x05 (Repeat Counter))	UINT32	RW	0xF101:05, 16
1A85:06	SubIndex 006	6. PDO Mapping entry (16 bits align)	UINT32	RW	0x0000:00, 16
1A85:07	SubIndex 007	7. PDO Mapping entry (16 bits align)	UINT32	RW	0x0000:00, 16
1A85:08	SubIndex 008	8. PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8
1A85:09	SubIndex 009	9. PDO Mapping entry (4 bits align)	UINT32	RW	0x0000:00, 4
1A85:0A	SubIndex 010	10. PDO Mapping entry (object 0xF101 (DP Master Status data), entry 0x14 (Device Diag))	UINT32	RW	0xF101:14, 1
1A85:0B	SubIndex 011	11. PDO Mapping entry (object 0xF101 (DP Master Status data), entry 0x15 (Sync Error))	UINT32	RW	0xF101:15, 1
1A85:0C	SubIndex 012	12. PDO Mapping entry (object 0xF101 (DP Master Status data), entry 0x16 (Cycle Toggle))	UINT32	RW	0xF101:16, 1
1A85:0D	SubIndex 013	13. PDO Mapping entry (object 0xF101 (DP Master Status data), entry 0x17 (Cycle State))	UINT32	RW	0xF101:17, 1

Index 1A86 DPM TxPDO-Map Cycle Statistics

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A86:0	DPM TxPDO-Map Cycle Statistics	PDO Mapping TxPDO 135 enthält die DP Master Cycle Statistics (Objekt 0xF10F [▶ 168]) für den SM-synchronen Betrieb	UINT8	RW	0x05 (5_{dez})
1A86:01	SubIndex 001	1. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x03 (Cycle Time))	UINT32	RW	0xF10F:03, 16
1A86:02	SubIndex 002	2. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x04 (Cycle Distance Time))	UINT32	RW	0xF10F:04, 16
1A86:03	SubIndex 003	3. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x05 (Output Calc And Copy Time))	UINT32	RW	0xF10F:05, 16
1A86:04	SubIndex 004	4. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x06 (Input Calc And Copy Time))	UINT32	RW	0xF10F:06, 16
1A86:05	SubIndex 005	5. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x07 (Output Delay Time))	UINT32	RW	0xF10F:07, 16

Index 1A87 DPM TxPDO-Map DC Cyclic Statistics

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A87:0	DPM TxPDO-Map DC Cyclic Statistics	PDO Mapping TxPDO 136 die DP Master Cycle Statistics (Objekt 0xF10F [▶ 168]) für den DC-Betrieb	UINT8	RW	0x06 (6_{dez})
1A87:01	SubIndex 001	1. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x03 (Cycle Time))	UINT32	RW	0xF10F:03, 16
1A87:02	SubIndex 002	2. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x04 (Cycle Distance Time))	UINT32	RW	0xF10F:04, 16
1A87:03	SubIndex 003	3. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x05 (Output Calc And Copy Time))	UINT32	RW	0xF10F:05, 16
1A87:04	SubIndex 004	4. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x06 (Input Calc And Copy Time))	UINT32	RW	0xF10F:06, 16
1A87:05	SubIndex 005	5. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x07 (Output Delay Time))	UINT32	RW	0xF10F:07, 32
1A87:06	SubIndex 006	6. PDO Mapping entry (object 0xF10F (Cycle Statistic data), entry 0x08 (SM/SYNC Event Distance Time))	UINT32	RW	0xF10F:08, 32

Index 1A88 DPM TxPDO-Map Redundancy Status

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A88:0	DPM TxPDO-Map Redundancy Status	PDO Mapping TxPDO 137	UINT8	RO	0x05 (5_{dez})
1A88:01	SubIndex 001	1. PDO Mapping entry (object 0xF118, entry 0x01)	UINT32	RO	0xF118:01, 1
1A88:02	SubIndex 002	2. PDO Mapping entry (15 bits align)	UINT32	RO	0x0000:00, 15
1A88:03	SubIndex 003	3. PDO Mapping entry (object 0xF118, entry 0x11)	UINT32	RO	0xF118:11, 16
1A88:04	SubIndex 004	4. PDO Mapping entry (object 0xF118, entry 0x12)	UINT32	RO	0xF118:12, 16
1A88:05	SubIndex 005	5. PDO Mapping entry (object 0xF118, entry 0x13)	UINT32	RO	0xF118:13, 16

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs: die RxPDOs müssen in der Reihenfolge ihrer Indexe assigned werden. Die RxPDOs der konfigurierten DP-Slaves (0x1600 [▶ 154]-0x167E) müssen assigned werden, wenn das Objekt 0x1C12 in den StartUp-SDOs übertragen wird.	UINT8	RW	0x00 (0_{dez})
1C12:01		1. zugeordnete RxPDO (enthält den Index des zugehörigen RxPDO Mapping Objekts)			
...					
1C12:7E		126. zugeordnete RxPDO (enthält den Index des zugehörigen RxPDO Mapping Objekts)			

Index 1C13 TxPDO assign

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs: die TxPDOs müssen in der Reihenfolge ihrer Indexe assigned werden. Die TxPDOs der konfigurierten DP Slaves (0x1A00 [▶ 156]-0x1A7C) müssen assigned werden, wenn das Objekt 0x1C13 in den StartUp-SDOs übertragen wird. Über das TxPDO Assign kann dann noch entschieden werden, ob die TxPDOs PDO State (Index 0x1A81 [▶ 156]), PDO Toggle (Index 0x1A82 [▶ 157]), DP Diag (Index 0x1A83 [▶ 157]), DP Status (Index 0x1A84 [▶ 157]), DP Master Diag (Index 0x1A85 [▶ 157]), Cycle Statistics (Index 0x1A86 [▶ 158]) und DC Cycle Statistics (Index 0x1A87 [▶ 158]) in den EtherCAT Input Data übertragen werden.	UINT8	RW	
1C13:01		1. zugeordnete TxPDO (enthält den Index des zugehörigen TxPDO Mapping Objekts)	UINT16	RW	
...					
1C13:86		134. zugeordnete TxPDO (enthält den Index des zugehörigen TxPDO Mapping Objekts)	UINT16	RW	

Index 1C32 SM output parameter

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 (SM-Synchron) • 2: DC-Mode - Synchron with SYNC0 Event (DC-SYNC0-Synchron) • 3: DC-Mode - Synchron with SYNC1 Event (DC-SYNC1-Synchron)	UINT16	RW	0x0001 (1_{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 Cycle Time (i.d.R. auch Zykluszeit des Masters)	UINT32	RW	0x000F4240 (1000000_{dez})
1C32:03	Shift time	nur DC-Mode (wie 0x1C32:09 [▶ 160])	UINT32	RO	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 [▶ 160])	UINT16	RO	0x440B (17419_{dez})
1C32:05	Minimum cycle time	Dieser Entry enthält die gemessene Zykluszeit (in ns) des DP Zyklus inklusive des Updates der EtherCAT Process Data, hier steht der maximal gemessene Wert (Summe aus 0x1C32:06+0x1C33:09+0x1C33:06). Wenn der CycleExceeded-Counter (0x1C32:0C oder 0x1C33:0C) hoch zählt ist die in 0x1C32:02 eingestellte Zykluszeit zu klein für den konfigurierten DP-Busaufbau.	UINT32	RO	0x0003CFB1 (249777_{dez})
1C32:06	Calc and copy time	Dieser Entry enthält die gemessene Zeit nach dem ersten Event (SM2 (SM-Synchron- oder DC-SYNC0-Synchron) oder SYNC0 (DC-SYNC1-Synchron)) bis zum Starten des DP-Zyklus (SM-Synchron) bzw. nach der das zweite Event (SYNC0 (DC-SYNC0-Synchron) bzw. SYNC1 (DC-SYNC1-Synchron)) frühestens kommen darf. Bei der Einstellung DC-SYNC0-Synchron ist diese Zeit ggf. bei der SYNC Shift Time für Outputs des EtherCAT Masters mit zu berücksichtigen. Bei der Einstellung DC SYNC1-Synchron ist diese Zeit ggf. bei der SYNC1 Shift Time der EL6731 mit zu berücksichtigen. Wenn der ShiftTooShort-Counter (0x1C32:0D) hoch zählt, ist die entsprechende Shift Time zu klein eingestellt.	UINT32	RO	0x00000000 (0_{dez})
1C32:07	Minimum delay time		UINT32	RO	0x00000000 (0_{dez})
1C32:08	Command	• 0: Messung der lokalen Zykluszeit wird gestoppt • 1: Messung der lokalen Zykluszeit wird gestartet Die Entries 0x1C32:03 [▶ 160], 0x1C32:05 [▶ 160], 0x1C32:06 [▶ 160], 0x1C32:09 [▶ 160], 0x1C33:03 [▶ 162], 0x1C33:06 [▶ 160], 0x1C33:09 [▶ 162] werden mit den maximal gemessenen Werten aktualisiert. Wenn erneut gemessen wird, werden die Messwerte zurückgesetzt	UINT16	RW	0x0000 (0_{dez})
1C32:09	Delay time	Dieser Entry enthält die Zeit nach dem zweiten Event (SYNC0-Event (DC-SYNC0-Synchron) bzw. SYNC1 Event (DC-SYNC1-Synchron)) und dem Zeitpunkt und Ausgabe der Outputs (in ns, nur DC-Mode)	UINT32	RO	0x00000000 (0_{dez})

Index 1C32 SM output parameter

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C32:0B	SM event missed counter	Dieser Entry enthält die Anzahl der ausgefallenen SM-Events im OPERATIONAL (nur im DC Mode). Der Zähler wird beim Übergang von SAFEOP nach OP automatisch zurückgesetzt. Wenn er hoch zählt, ist die SYNC Shift Time für Outputs beim EtherCAT Master zu klein.	UINT16	RO	0x0000 (0_{dez})
1C32:0C	Cycle exceeded counter	Dieser Entry enthält die Anzahl der Zykluszeitverletzungen im OPERATIONAL (Zyklus wurde nicht rechtzeitig fertig bzw. der nächste Zyklus kam zu früh). Der Zähler wird beim Übergang von SAFEOP nach OP automatisch zurückgesetzt. Wenn er hoch zählt, ist die Zykluszeit des EtherCAT Masters zu klein.	UINT16	RO	0x0000 (0_{dez})
1C32:0D	Shift too short counter	Dieser Entry enthält die Anzahl der zu kurzen Abstände zwischen SM2 und SYNC 0 Event (DC-SYNC0) bzw. zwischen SYNC0 und SYNC1 Event (DC-SYNC1). Der Zähler wird beim Übergang von SAFEOP nach OP automatisch zurückgesetzt. Wenn er hoch zählt, ist die SYNC Shift Time für Outputs beim EtherCAT Master (DC-SYNC0) bzw. die SYNC1 Shift Time bei der EL6731 (DC-SYNC1) zu klein.	UINT16	RO	0x0000 (0_{dez})
1C32:0F	Minimum Cycle Distance				0x00000000 (0_{dez})
1C32:10	Maximum Cycle Distance				0x00000000 (0_{dez})
1C32:11	Minimum SM SYNC Distance				0x00000000 (0_{dez})
1C32:12	Maximum SM SYNC Distance				0x00000000 (0_{dez})
1C32:13	Application Cycle Exceeded Counter				0x0000 (0_{dez})
1C32:14	Frame repeat time				0x00000000 (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 1C33 SM input parameter

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	0x0022 (34_{dez})
1C33:02	Cycle time	wie 0x1C32:02 [▶ 160]	UINT32	RW	0x000F4240 (1000000_{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 0x1C32:08 [▶ 160] oder 0x1C33:08 [▶ 162])	UINT16	RO	0x440B (17419_{dez})
1C33:05	Minimum cycle time	wie 0x1C32:05 [▶ 160]	UINT32	RO	0x0003CFB1 (249777_{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})
1C33:07	Minimum delay time		UINT32	RO	0x00000000 (0_{dez})
1C33:08	Command	wie 0x1C32:08 [▶ 160]	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 0x1C32:11 [▶ 160]	UINT16	RO	0x0000 (0_{dez})
1C33:0C	Cycle exceeded counter	wie 0x1C32:12 [▶ 160]	UINT16	RO	0x0000 (0_{dez})
1C33:0D	Shift too short counter	wie 0x1C32:13 [▶ 160]	UINT16	RO	0x0000 (0_{dez})
1C32:0F	Minimum Cycle Distance		UINT32	RO	0x00000000 (0_{dez})
1C32:10	Maximum Cycle Distance		UINT32	RO	0x00000000 (0_{dez})
1C32:11	Minimum SM SYNC Distance		UINT32	RO	0x00000000 (0_{dez})
1C32:12	Maximum SM SYNC Distance		UINT32	RO	0x00000000 (0_{dez})
1C32:13	Application Cycle Exceeded Counter		UINT16	RO	0x0000 (0_{dez})
1C32:14	Frame repeat time		UINT32	RW	0x00000000 (0_{dez})
1C33:20	Sync error	wie 0x1C32:32 [▶ 160]	BOOLEAN	RO	0x00 (0_{dez})

6.1.3.2 Profilspezifische Objekte (0x6000-0xFFFF)

Die profilspezifischen Objekte haben für alle EtherCAT Slaves, die das Profil 5001 unterstützen, die gleiche Bedeutung.

Index 6000-67C0 DP Inputs Slave yyy

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
6000+n*16: 0	DP Inputs Slave yyy	Je konfiguriertem DP-Slave gibt es dieses Objekt, das die DP Input Data des DP Slaves enthält. Je in Objekt 0x8002 [▶ 165]+(n*16) konfiguriertem DP-Modul mit DP Input Data gibt es einen Entry in dem Input Data Objekt 0x6000+(n*16). DP Modul m entspricht dabei Subindex m, falls ein DP-Modul keine DP Input Data enthält, existiert dieser Subindex nicht. Falls ein DP-Slave keine DP Input Data enthält, dann existiert das Objekt 0x6000+(n*16) nicht. SubIndex 0 enthält die Anzahl der DP Module des (n+1). konfigurierten DP-Slave.	UINT8	RO	
(6000+n*16) :01		falls erstes DP-Modul DP Input Data hat: DP Input Data des ersten DP Moduls	OCTET-STRING	RO	
...					
(6000+n*16) :m		falls m. DP-Modul DP Input Data hat: DP Input Data des m. DP Moduls	OCTET-STRING	RO	

Index 7000-77C0 DP Outputs Slave yyy

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
7000+n*16: 0	DP Outputs Slave yyy	Je konfiguriertem DP-Slave gibt es dieses Objekt, das die DP Output Data des DP Slaves enthält. Je in Objekt 0x8002 [▶ 165]+(n*16) konfiguriertem DP-Modul mit DP Output Data gibt es einen Entry in dem Input Data Objekt 0x7000+(n*16). DP Modul m entspricht dabei Subindex m, falls ein DP-Modul keine DP Output Data enthält, existiert dieser Subindex nicht. Falls ein DP-Slave keine DP Output Data enthält, dann existiert das Objekt 0x7000+(n*16) nicht. SubIndex 0 enthält die Anzahl der DP Module des (n+1). konfigurierten DP-Slave.	UINT8	RO	
(7000+n*16) :01		falls erstes DP-Modul DP Output Data hat: DP Output Data des ersten DP Moduls	OCTET-STRING	RO	
...					
(7000+n*16) :F4		falls m. DP-Modul DP Output Data hat: DP Output Data des m. DP Moduls	OCTET-STRING	RO	

Index 8000-87C0 DP Slave Parameter Set Slave yyy

Index (hex)	Name	Bedeutung		Datentyp	Flags	Default
8000+n*16: 0	DP Slave Parameter Set Slave yyy	Dieses Objekt enthält die DP Konfiguration des (n+1). konfigurierten DP Slave (0 <= n < 125). Das Objekt ist mit Complete Access zu übertragen oder es muss erst SubIndex 0 auf 0 gesetzt , dann die einzelnen SubIndexe übertragen (nicht vorhandene SubIndexe bzw. Lücken sind dabei auszulassen) und schließlich SubIndex 0 auf den richtigen Wert gesetzt werden.		UINT8	RW	0x33 (51_{dez})
(8000+n*16) :01	Station Address	DP Stationsadresse des DP Slaves, erlaubte Werte: 0-125, damit wird automatisch der Entry 0xF020: (n+1) aktualisiert		UINT16	RW	0x0000 (0_{dez})
(8000+n*16) :04	Device Type	DP Ident Number des DP Slaves		UINT32	RW	0x00000000 (0_{dez})
(8000+n*16) :1D	Network Flags	reserviert für AMS über DP		UINT16	RW	0x0000 (0_{dez})
(8000+n*16) :1E	Network Port	reserviert für AMS über DP		UINT16	RW	0x0000 (0_{dez})
(8000+n*16) :1F	Network Segment Address	reserviert für AMS über DP		OCTET-STRING[6]	RW	{0}
(8000+n*16) :20	Sl_Flag	Sl_flag		UINT8	RW	0x80 (128_{dez})
		Bit 0-2	Reserve, muss 0 sein			
		Bit 3	DPV1-Kommunikation für DP Slave aktivieren			
		Bit 4	Data-Exchange Telegramm wird als DXB-Broadcast gesendet			
		Bit 5	FailSafe-Funktionalität (Data-Exchange ohne Daten im Zustand CLEAR (SAFE-OP) senden) ist aktiv			
		Bit 6	Reserve, muss 0 sein			
		Bit 7	Reserve, muss 1 sein			
(8000+n*16) :21	Slave_Type	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
(8000+n*16) :22	Max_Diag_Data_Len	maximale Länge der DP DiagData des DP Slaves (6 <= Max_Diag_Data_Len <= 244)		UINT8	RW	
(8000+n*16) :23	Max_Alarm_Len	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
(8000+n*16) :24	Max_Channel_Data_Length	maximale Länge der DPV1 Telegramme des DP Slaves (4 <= Max_Channel_Data_Length <= 244)		UINT8	RW	
(8000+n*16) :25	Diag_Upd_Delay	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
(8000+n*16) :26	Alarm_Mode	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
(8000+n*16) :27	Add_Sl_Flag	Reserve, muss 1 sein		UINT8	RW	0x01 (1_{dez})
(8000+n*16) :28	C1_Timeout	Timeout für das warten auf eine DPV1-Response (in 10 ms)		UINT16	RW	
(8000+n*16) :29	Number of tolerated Data Exchange with no answer	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
(8000+n*16) :2A	Parallel AoE Services	Anzahl der parallelen AoE services (z. B. für DPV1), der Wert 0 entspricht 5 parallele services		UINT8	RW	0x00 (0_{dez})
(8000+n*16) :2B	Reaction on no answer	0	Data-Exchange wird verlassen	BIT1	RW	0x00 (0_{dez})
		1	DP Slave bleibt in Data-Exchange,wenn innerhalb der DP Watchdogzeit eine gültige Data-Exchange-Response empfangen wird			
(8000+n*16) :2C	Restart behaviour after DP fault	0	DP Slave wird automatisch neu gestartet (Set_Prm-Unlock, dann Slave_Diag, Set_Prm-Lock, Chk_Cfg, Slave_Diag)	BIT1	RW	0x00 (0_{dez})
		1	DP Slave muss manuellper AoE neu gestartet werden			

Index 8000-87C0 DP Slave Parameter Set Slave yyy

Index (hex)	Name	Bedeutung		Datentyp	Flags	Default
(8000+n*16):2D	Master reaction after DP fault	0	keine Reaktion	BIT1	RW	0x00 (0_{dez})
		1	Datenaustausch mit allen DP-Slaves wird gestoppt (Set_Prm-Unlock an alle Slaves), DP-Kommunikation muss per AoE wieder neu gestartet werden			
(8000+n*16):2E	Changes of DP Inputs after DP fault	0	Daten der TxPDO n werden auf 0 gesetzt	BIT1	RW	0x00 (0_{dez})
		1	Daten der TxPDO n werden bleiben unverändert			
(8000+n*16):2F	PKW supported	Reserve, muss 0 sein		BOOLEAN	RW	0x00 (0_{dez})
(8000+n*16):30	FDL only	Reserve, muss 0 sein		BOOLEAN	RW	0x00 (0_{dez})
(8000+n*16):31	Watchdog Base 1 ms	FALSE	Die Basis für den DP watchdog ist 10 ms	BOOLEAN	RW	0x00 (0_{dez})
		TRUE	Die Basis für den DP watchdog ist 1 ms			
(8000+n*16):33	Cycle Access Divider	Reserve, muss 1 sein		UINT8	RW	0x01 (1_{dez})
(8000+n*16):34	Cycle Access Modulo	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
(8000+n*16):35	Vendor specific reserved	Reserve, muss 0 sein		OCTET-STRING[25]	RW	{0}

Index 8001-87C1 DP PrmData Slave yyy

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
8001+n*16:0	DP PrmData Slave yyy	Dieses Objekt enthält die DP Prm Data des (n+1). konfigurierten DP Slave.	OCTET-STRING[244]	RW	{0}

Index 8002-87C2 DP CfgData Slave yyy

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
8002+n*16:0	DP CfgData Slave yyy	Dieses Objekt enthält die DP Cfg Data des (n+1). konfigurierten DP Slave.	OCTET-STRING[244]	RW	{0}

Index A000-A7C0DP Status data Slave yyy

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
A000+n*16:0	DP Status data Slave.000	Dieses Objekt enthält die Statusdaten des (n+1). konfigurierten DP Slave.	UINT8	RO	0x04 (4_{dez})
(A000+n*16):01	DP state	Zustand der DP Verbindung (identisch mit Entry 0xF102 [▶ 167]:(n+1))	UINT8	RO	0x00 (0_{dez})
(A000+n*16):02	Ext Diag	Dieser Entry zeigt an, ob sich die Diagnoseinformation in Objekt 0xA002+(n*16) geändert haben	BOOLEAN	RO	0x00 (0_{dez})
(A000+n*16):03	Repeat Counter	Dieser Entry zählt mit jedem Wiederholungstelegramm zu dem DP Slave hoch	UINT32	RO	0x00000000 (0_{dez})
(A000+n*16):04	No Answer Counter	Dieser Entry zählt mit jedem Telegramm zu dem DP Slave hoch, zu dem keine Response empfangen wird	UINT32	RO	0x00000000 (0_{dez})

Index A002-A07E DP Diag data Slave.000

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
A002+n*16:0	DP Diag data Slave.000	Dieses Objekt enthält die DP DiagData des (n+1). konfigurierten DP Slave.	OCTET-STRING[244]	RO	{0}

Index F000 Modular device profile

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F000:0	Modular device profile	Allgemeine Informationen des Modular Device Profiles	UINT8	RO	0x02 (2_{dez})
F000:01	Module index distance	Indexabstand der Objekte der einzelnen Kanäle	UINT16	RO	0x0010 (16_{dez})
F000:02	Maximum number of modules	Anzahl der Kanäle	UINT16	RO	0x007D (125_{dez})
F000:03	General Configuration Entries	Allgemeine Konfigurationseinträge	UINT32	RO	0x70000009 (1879048201_{dez})

Index F008 Code word

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

Index F010 Module list

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F010:0	Module list	Max. Subindex	UINT8	RW	0x00 (0_{dez})
F010:01		reserviert			
...		reserviert			
F010:7D		reserviert			

Index F101 DP Master Status data

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F101:0	DP Master Status data	Dieses Objekt enthält die DP Master Statusdaten, die in die TxPDO 134 (Index 0x1A85 [▶ 158]) gemappt wird	UINT8	RO	0x17 (23_{dez})
F101:01	Bus Error Counter	zählt jeder Busstörung hoch	UINT16	RO	0x0000 (0_{dez})
F101:02	Cycle Counter	zählt mit jedem DP-Zyklus hoch	UINT16	RO	0x0000 (0_{dez})
F101:03	Slave Status Counter	Anzahl der DP Slaves, die im vorangegangenen Zyklus nicht im Datenaustausch waren	UINT16	RO	0x0000 (0_{dez})
F101:04	Cycle Time	Zykluszeit des vorangegangenen DP-Zyklusses in 1/9 µs	UINT16	RO	0x0000 (0_{dez})
F101:05	Repeat Counter	Anzahl der Wiederholungen im vorangegangenen DP Zyklus	UINT16	RO	0x0000 (0_{dez})
F101:14	Device Diag	zeigt an, ob Diagnoseinformation beim Master vorliegen, die per AoE abgeholt werden können	BOOLEAN	RO	0x00 (0_{dez})
F101:15	Sync Error	ist gesetzt, wenn im Distributed Clocks Betrieb im vorangegangenen Zyklus ein Synchronisierungsfehler aufgetreten ist	BOOLEAN	RO	0x00 (0_{dez})
F101:16	Cycle Toggle	toggelt mit jedem DP-zyklus	BOOLEAN	RO	0x00 (0_{dez})
F101:17	Cycle State	ist gesetzt, wenn mindestens ein DP Slave nicht im datenaustausch ist	BOOLEAN	RO	0x00 (0_{dez})

Index F102 DP Slave Status data

Index	Name	Bedeutung		Datentyp	Flags	Default
F102:0	DP Slave Status data	Dieses Objekt enthält den Kommnikationsstatus zu den DP-Slaves, der in die TxPDO 133 (Index 0x1A84 [▶ 157]) gemappt wird		UINT8	RO	
F102:01		Kommunikationsstatus erster konfigurierter DP-Slave (in den Objekten 0x8000 [▶ 164], 0x8001 [▶ 165], 0x8002 [▶ 165])		UINT8	RO	
		0	Slave ist im Data-Exchange, DP-Inputs sind aktuell			
		1	Slave ist deaktiviert (per AoE)			
		2	Slave antwortet nicht			
		3	Slave ist im Datenaustausch mit einem anderen Master			
		4	falsche Antwort des Slaves (z. B. Service not activated bei Data-Exchange)			
		5	Slave meldet Parametrierfehler (Fehler im SetPrm-Telegramm (i.d.R. Ident-Number oder UserPrmData))			
		6	Slave meldet, dass eine DP-Funktion nicht unterstützt wird			
		7	Slave meldet Konfigurationsfehler 8Fehler im ChkCfg-Telegramm (i.d.R. falsche DP-Module konfiguriert))			
		8	Slave ist im DP-Hochlauf, aber noch nicht bereit für den Datenaustausch			
		9	Slave meldet statische Diagnose			
		11	Antwort mit Busfehler (z. B. Parity-oder Checksummenfehler)			
		14	Antwort mit Telegrammfehler (z. B. Requestbit gesetzt)			
		15	Slave meldet keine Resourcen (z. B. Länge der PrmData oder CfgData ist zu groß)			
		16	Slave meldet, dass der DP-Dienst nicht aktiviert ist			
		17	unerwartetes Telegramm empfangen (z. B. Token, wenn auf Antwort eines Slaves gewartet wird)			
		18	Slave ist bereit für den Datenaustausch, es werden aber noch keine Prozessdaten über EtherCAT ausgetauscht			
...						
F102:7D		Kommunikationsstatus m. konfigurierter DP-Slave		UINT8	RO	

Index F10F Cycle Statistic data

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F10F:0	Cycle Statistic data	Dieses Objekt enthält die gemessenen Zeiten des jeweils letzten Zyklus. Sie können mit der TxPDO 135 (0x1A86 [▶ 158]) bzw. der TxPDO 136 (0x1A87 [▶ 158]) zyklisch übertragen werden und enthalten dann immer die Werte für den vorangegangenen Zyklus.	UINT8	RO	0x08 (8_{dez})
F10F:01	Multiplier	Mit diesem Wert müssen die Entries 0xF10F:03, 0xF10F:04, 0xF10F:05, 0xF10F:06 multipliziert werden, um diese Entries in die Einheit 100 ns umzurechnen	UINT16	RO	0x0000 (0_{dez})
F10F:02	Divider	Mit diesem Wert müssen die Entries 0xF10F:03, 0xF10F:04, 0xF10F:05, 0xF10F:06 dividiert werden, um diese Entries in die Einheit 100 ns umzurechnen	UINT16	RO	0x0000 (0_{dez})
F10F:03	Cycle Time	Zykluszeit (0x1C32:05 [▶ 160]) des vorangegangenen DP-Zyklus in lokalen Ticks	UINT16	RO	0x0000 (0_{dez})
F10F:04	Cycle Distance Time	Abstand zwischen den letzten beiden SM2-Events (nur wenn 0x1C32:01 [▶ 160]=1 oder 0x1C32:01 = 2)	UINT16	RO	0x0000 (0_{dez})
F10F:05	Output Calc And Copy Time	Output Calc And Copy Time (0x1C32:06 [▶ 160]) des vorangegangenen DP-Zyklus in lokalen Ticks.	UINT16	RO	0x0000 (0_{dez})
F10F:06	Input Calc And Copy Time	Input Calc And Copy Time (0x1C33:06) des vorangegangenen DP-Zyklus in lokalen Ticks.	UINT16	RO	0x0000 (0_{dez})
F10F:07	Output Delay Time	Output Delay Time (0x1C32:09 [▶ 160]) des vorangegangenen Zyklus in ns	UINT32	RO	0x00000000 (0_{dez})
F10F:08	SM/SYNC Event Distance Time	SM/SYNC0 Event Distance Time (Abstand zwischen SM2 und SYNC0 Event) des vorangegangenen Zyklus in ns (muss über das Register 0x816 im ET1100 aktiviert werden).	INT32	RO	0x00000000 (0_{dez})

Index F800 DP Bus Parameter Set

Index (hex)	Name	Bedeutung		Datentyp	Flags	Default
F800:0	DP Bus Parameter Set	Dieses Objekt enthält die DP-Bus-Parameter. Das Objekt ist mit Complete Access zu übertragen oder es muss erst SubIndex 0 auf 0 gesetzt, dann die einzelnen SubIndexe übertragen (nicht vorhandene SubIndexe bzw. Lücken sind dabei auszulassen) und schließlich SubIndex 0 auf den richtigen Wert gesetzt werden.		UINT8	RW	0x18 (24_{dez})
F800:01	DL_Add	Stationsadresse des DP-Masters		UINT8	RW	0x00 (0_{dez})
F800:02	Data_rate	Baudrate		UINT8	RW	0x00 (0_{dez})
		0	9.6 kBaud			
		1	19.2 kBaud			
		2	93.75 kBaud			
		3	187.5 kBaud			
		4	500 kBaud			
		6	1.5 MBaud			
		7	3 MBaud			
		8	6 MBaud			
		9	12 MBaud			
F800:03	Tsl	Slot-Time in Bitzeiten		UINT16	RW	*
F800:04	min Tsdr	Min. TSDR in Bitzeiten		UINT16	RW	0x0B (11_{dez})
F800:05	max Tsdr	Max. TSDR in Bitzeiten		UINT16	RW	*
F800:06	Tqui	Quiet Time in Bitzeiten		UINT8	RW	*
F800:07	Tset	Setup Time in Bitzeiten		UINT8	RW	*
F800:08	Ttr	Target Token Rotation Time in Bitzeiten		UINT32	RW	*
F800:09	G	GAP Update Factor (1-100)		UINT8	RW	0x64 (100_{dez})
F800:0A	HSA	höchste Masteradresse (0-126)		UINT8	RW	0x7E (126_{dez})
F800:0B	max_retry_limit	Maximale Anzahl Wiederholungen (1-8)		UINT8	RW	*
F800:0C	Bp_Flag	Bit 0	Single Master operation (es wird kein GAP-Update gesendet)	UINT8	RW	0x00 (0_{dez})
		Bit 1-6	Reserve, muss 0 sein			
		Bit 8	Auto-Clear Modus ist an (DP-Master geht nach CLEAR, wenn ein DP-Slave nicht im Datenaustausch)			

Index F800 DP Bus Parameter Set

Index (hex)	Name	Bedeutung		Datentyp	Flags	Default
F800:0D	Min_Slave_Interval	Zykluszeit für DP-Telegramme (außer Data-Exchange (läuft synchron zum EtherCAT-Master))		UINT16	RW	*
F800:0E	Poll_Timeout	Reserve, muss 0 sein		UINT16	RW	0x0000 (0_{dez})
F800:0F	Data_Control_Time	Zykluszeit des Global_Control-Telegramms		UINT16	RW	*
F800:10	Alarm_Max	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
F800:11	Max_User_Global_control	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
F800:12	Max Retry Limit (Data-Exchange)	Maximale Anzahl Wiederholungen Data-Exchange		BIT4	RW	*
F800:13	Send Set-Prm-Unlock	Reserve, muss 0 sein		BIT2	RW	0x00 (0_{dez})
F800:14	Auto Clear Mode	Reserve, muss 0 sein		BIT2	RW	0x00 (0_{dez})
F800:15	Operate Delay (in 100 ms)	Das automatische Schalten nach OPERATE im Zustand OP wird entsprechend dieser Einstellung verzögert		UINT8	RW	0x00 (0_{dez})
F800:16	Safety Time (in us)	Reserve, muss 0 sein		UINT16	RW	0x0000 (0_{dez})
F800:17	Flags	Reserve, muss 0 sein		UINT16	RW	0x0000 (0_{dez})
F800:18	Watchdog Reaction	0	DP Master in CLEAR Mode	OCTET-STRING[26]	RW	{0}
		1	DP Master in OPERATE Mode			
		2	DP Master in STOP Mode			
F800:1A	Vendor specific	Reserve, muss 0 sein		OCTET-STRING[26]	RW	

Index F920 AoE Settings

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F920:0	AoE Settings	Max. Subindex	UINT8	RO	0x01 (1_{dez})
F920:01	Local AoE Net ID	Lokale AoE Net ID	OCTET-STRING[6]	RO	{0}

* Diese Parameter hängen von der Baudrate ab, in der folgenden Tabelle sind die Defaultwerte für die jeweilige Baudrate angegeben:

Data_rate	Tsl	max. Tsdr	Tqui	Tset	Ttr	max_retry_limit	Min_Slave_Interval	Data_Control_Time	Max Retry (DX)
9 (12 MBaud)	1000	800	9	16	12000	4	10	1	4
8 (6 MBaud)	600	450	6	8	12000	3	20	2	3
7 (3 MBaud)	400	250	3	4	12000	2	40	4	2
6 (1.5 MBaud)	300	150	0	1	12000	1	80	8	1
4 (500 kBaud)	200	100	0	1	12000	1	200	20	1
3 (187.5 kBaud)	150	100	0	1	12000	1	400	40	1
2 (93.75 kBaud)	150	100	0	1	12000	1	800	80	1
1 (19.2 kBaud)	150	100	0	1	12000	1	4000	400	1
0 (9.6 kBaud)	150	100	0	1	12000	1	8000	800	1

6.2 PROFIBUS Slave

6.2.1 State Machine

Die EL6731-0010 kann auf verschiedene Weisen konfiguriert werden:

1. Konfiguration der EL6731-0010 mit StartUp-SDOs [▶ 170]: Hierbei werden die StartUp-SDOs im EtherCAT Konfigurator berechnet und an den EtherCAT Master übergeben wie es z. B. im TwinCAT System Manager durchgeführt wird.
2. Konfiguration der EL6731-0010 mit Backup Parameter Storage [▶ 173]: Hierbei wird die Konfiguration der PROFIBUS-Slaves im Flash der EL6731-0010 gespeichert und muss nur einmalig übertragen werden.

Konfiguration der EL6731-0010 mit StartUp-SDOs

Das folgende Flussdiagramm zeigt den Ablauf der Konfiguration der EL6731-0010 mit Start-SDOs:

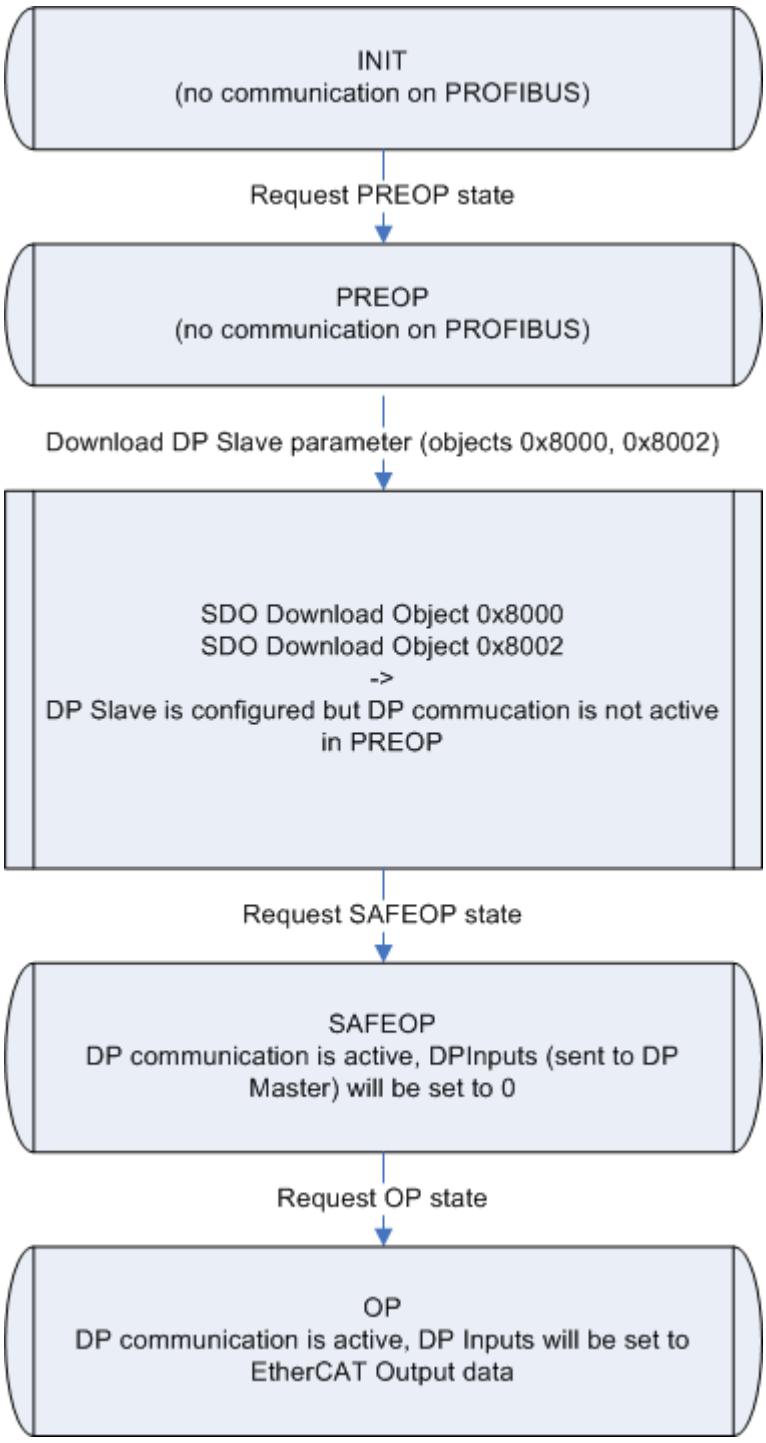


Abb. 134: Flussdiagramm: Ablauf der Konfiguration der EL6731-0010 mit Start-SDOs

Nach einem Power-On befindet sich die EL6731-0010 im Zustand INIT und besitzt keine DP-Konfiguration. Die EL6731-0010 ist nicht am PROFIBUS aktiv.

DP Slave Konfiguration

Im Zustand PREOP wird die DP-Konfiguration per SDO-Download durchgeführt. Die zu ladenden Objekte müssen entweder mit Complete-Access oder mit der Konsistenzklammerung (erst Subindex 0 auf 0 setzen, dann Subindex 1-n schreiben, dann Subindex 0 auf n setzen) übertragen werden. Dabei ist zu beachten, dass immer mit dem Objekt 0x8000 [▶ 180] begonnen wird. Die DP CfgData sind mit dem Objekt 0x8002 [▶ 180] zu übertragen. Nach dem Empfang des Objekts 0x8000 ist die EL6731-0010 am PROFIBUS aktiv (nur FDL, kein DP). Die EL6731-0010 unterstützt die automatische Baudratensuche, daher muss keine Baudrate konfiguriert werden.

PDO Mapping

Es gibt eine EtherCAT RxPDO (falls der DP-Slave über DP-Inputs (wird an den DP-Master gesendet) verfügt) und eine EtherCAT TxPDO (falls der DP Slave über DP Outputs (wird vom DP Master empfangen) verfügt). Das PDO-Mapping der EtherCAT PDOs wird von der EL6731-0010 nach dem Download der jeweiligen 0x8yyz-Objekte automatisch berechnet und kann ausgelesen werden. Die PDO-Mappingobjekte können nur mit den Werten beschrieben werden, die die EL6731-0010 selbst berechnet hat. Das Schreiben des PDO-Mappings dient also nur zum Überprüfen des vom EtherCAT-Konfigurator berechneten PDO-Mappings und kann daher weggelassen werden.

PDO Assign

Zusätzlich gibt es noch eine EtherCAT PDOs die Statusinformationen enthält. Die Auswahl dieser PDO erfolgt über das PDO-Assign. Bei der Reihenfolge der PDOs im PDO-Assign ist zu beachten, dass mit jedem Entry im entsprechenden PDO-Assign-Objekt der Index der zugeordneten EtherCAT-PDO steigt. Wenn der EtherCAT Master kein PDO-Assign in den StartUp-SDOs überträgt, wird die PDO 0x1A7F [▶ 177] für den Status zugeordnet.

Zyklische DP Kommunikation

Beim Übergang nach SAFEOP überprüft die EL6731 die in den Sync-Manager Kanälen 2 und 3 konfigurierte Länge mit der berechneten Länge aus PDO-Mapping und PDO-Assign. Der Zustand SAFEOP wird nur eingenommen, wenn diese Längen übereinstimmen. Im Zustand SAFEOP ist die DP-Funktionalität der EL6731-0010 aktiv. Solange die EL6731-0010 sich in SAFE-OP befindet, werden die DP Input Data, die an den DP-Master gesendet werden, auf 0 gesetzt. Sobald die EL6731-0010 nach OP geschaltet wurde, werden die Daten aus den EtherCAT Outputs an den DP Master übertragen.

Die EL6731-0010 läuft freilaufend zum EtherCAT Zyklus.

Konfiguration der EL6731-0010 mit Backup Parameter Storage

Das folgende Flussdiagramm zeigt den Ablauf der Konfiguration der EL6731-0010 mit Backup Parameter Storage:

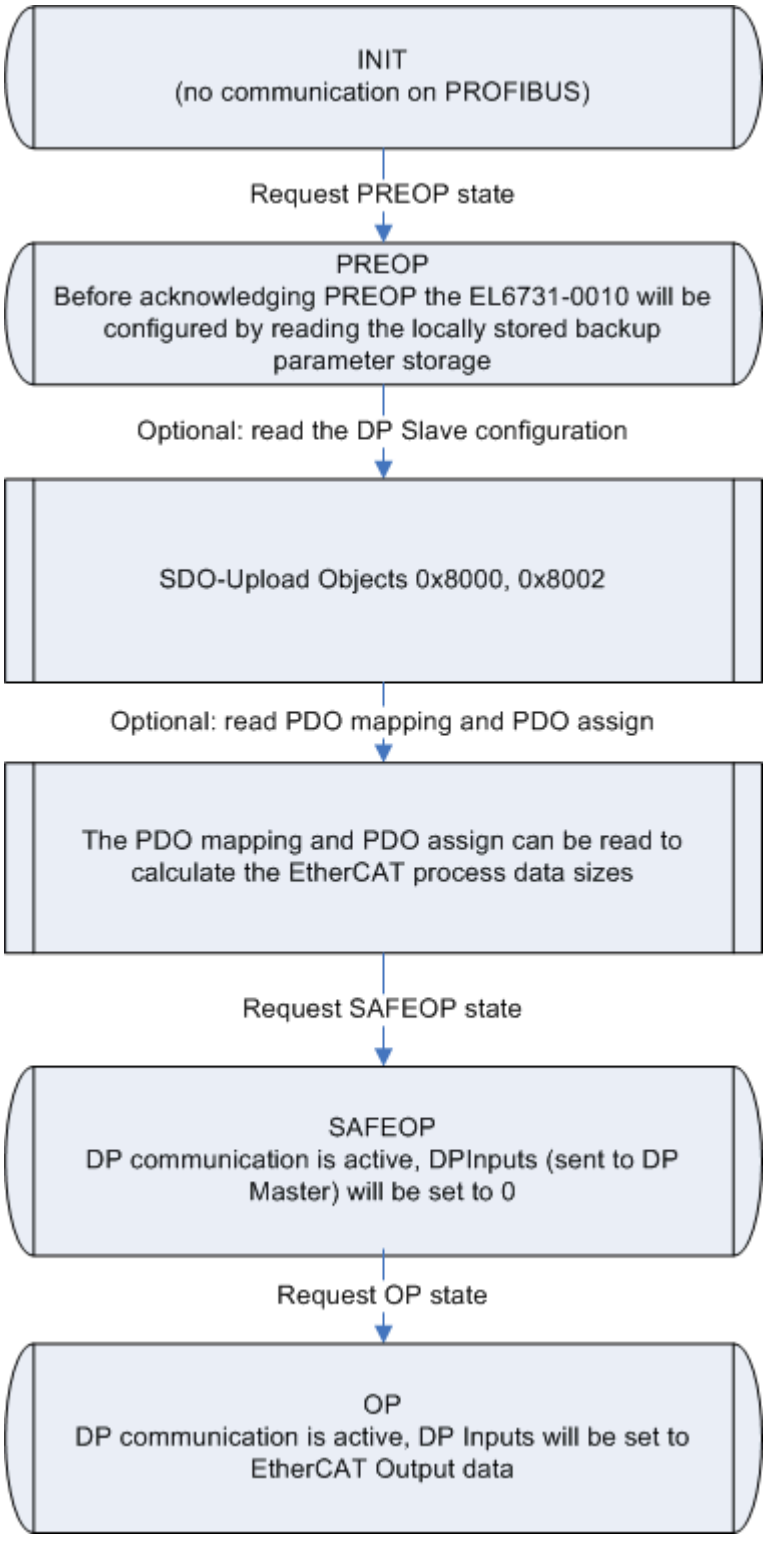


Abb. 135: Flussdiagramm: Ablauf der Konfiguration der EL6731-0010 mit Backup Parameter Storage

Nach einem Power-On befindet sich die EL6731-0010 im Zustand INIT und besitzt keine DP-Konfiguration. Die EL6731-0010 ist nicht am PROFIBUS aktiv.

DP Busparameter / DP Slave Konfiguration

Beim Übergang von INIT nach PREOP wird die im Backup Parameter Storage Objekt 0x10F2 gespeicherte Konfiguration geladen. Da im Backup Parameter Storage Objekt die StartUp-SDOs aus der Konfiguration der EL6731-0010 mit StartUp-SDOs [▶ 170] gespeichert sind, entspricht der Ablauf dem dort beschriebenen. Zunächst wird also das Objekt 0x8000 mit den gespeicherten Daten beschrieben und die EL6731-0010 ist am PROFIBUS mit automatischer Baudratensuche aktiv. Anschließend DP Slave Konfiguration in dem Objekt 0x8002 erzeugt. Wenn der Zustand PREOP quittiert ist, kann die aktuelle DP Konfiguration in den Objekten 0x8000 und 0x8002 ausgelesen werden.

PDO Mapping / PDO Assign

Außerdem kann der EtherCAT Master im Zustand PREOP auch das PDO mapping und PDO assign auslesen, um die Längen der EtherCAT process data zu ermitteln.

Erzeugen des Backup Parameter Storage

Das Backup Parameter Storage kann wie folgt erzeugt werden:

1. Download des Objekts 0x10F2 (in PREOP ohne dass vorher StartUp-SDOs gesendet wurden): In diesem Fall werden die empfangenen Daten als Backup Parameter Storage im Flash gespeichert

Löschen des Backup Parameter Storage

Um ein neues Backup Parameter Storage zu laden oder das vorhandene einfach nur zu löschen, ist der Entry 0x1011:01 mit dem Wert 0x64616F6C zu beschreiben.

6.2.2 Objektbeschreibung und Parametrierung

EtherCAT XML Device Description

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

Parametrierung über das CoE-Verzeichnis (CAN over EtherCAT)

Die Parametrierung des EtherCAT Gerätes wird über den CoE-Online Reiter (mit Doppelklick auf das entsprechende Objekt) bzw. über den Prozessdatenreiter (Zuordnung der PDOs) vorgenommen. Beachten Sie bei Verwendung/Manipulation der CoE-Parameter die allgemeinen CoE-Hinweise [▶ 39]:

- StartUp-Liste führen für den Austauschfall
- Unterscheidung zwischen Online/Offline Dictionary, Vorhandensein aktueller XML-Beschreibung
- „CoE-Reload“ zum Zurücksetzen der Veränderungen

6.2.2.1 Standardobjekte (0x1000-0x1FFF)

Die Standardobjekte haben für alle EtherCAT-Slaves die gleiche Bedeutung.

Index 1000 Device type

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1000:0	Device type	Geräte-Typ des EtherCAT-Slaves: Das Lo-Word enthält das verwendete CoE Profil (5001). Das Hi-Word enthält das Modul Profil entsprechend des Modular Device Profile.	UINT32	RO	0x0C1D1389 (203232137_{dez})

Index 1008 Device name

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1008:0	Device name	Geräte-Name des EtherCAT-Slave	STRING	RO	EL6731-0010

Index 1009 Hardware version

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

Index 100A Software version

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

Index 1011 Restore default parameters

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1011:0	Restore default parameters	Herstellen der Defaulteinstellungen	UINT8	RO	
1011:01	SubIndex 001	Wenn Sie dieses Objekt im Set Value Dialog auf **"0x64616F6C"** setzen, werden alle Backup Objekte wieder in den Auslieferungszustand gesetzt.	UINT32	RW	

Index 1018 Identity

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	0x1A4B3052 (441135186_{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	
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	

Index 10F0 Backup parameter handling

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
10F0:0	Backup parameter handling	Informationen zum standardisierten Laden und Speichern der Backup Entries	UINT8	RO	
10F0:01	Checksum	Checksumme über das Backup Parameter Storage (Objekt 0x10F2 [▶ 176], Word 2-3)	UINT32	RO	

Index 10F2 Backup parameter storage

Index (hex)	Name	Bedeutung		Datentyp	Flags	Default
10F2:0	Backup parameter storage	Wenn dieses Objekt verwendet wird, dürfen keine StartUp-SDOs im Zustand PREOP gesendet werden, da das Backup Parameter Storage die kompletten StartUp-SDOs enthält (s. Konfiguration der EL6731-0010 mit Backup Parameter Storage). 5 s nach dem Flaschen des Backup Parameter Storage wird die EL6731-0010 neu gebootet (schaltet nach INIT mit AL-Status-Code = 0x60). Die Daten haben die folgende Bedeutung:		OCTET-STRING[n]	RW	
		Word-Offset	Beschreibung			
		0	Kommando: mit 0xC0DE werden die empfangenen Daten im Flash gespeichert			
		1	Länge der Daten ab Word-Offset 4 in Bytes			
		2-3	Checksumme, die lokal berechnet wird			
		4	Index des Objekts der 1. StartUp-SDO			
		5	len1: Länge des Objekts der 1.StartUp-SDO als CompleteAccess (ab SubIndex 0) in Bytes			
		6-n1	Daten des Objekts der 1. StartUp-SDO als CompleteAccess (n1 = 2*((len1+1)/2)+5)			
		n1+1	Index des Objekts der 2. StartUp-SDO			
		n1+2	len2: Länge des Objekts der 2.StartUp-SDO als CompleteAccess (ab SubIndex 0) in Bytes			
		(n1+3)-n2	Daten des Objekts der 2. StartUp-SDO als CompleteAccess (n2 = 2*((len2+1)/2)+n1+2)			
		...				
		m	Index des Objekts der 3. StartUp-SDO			
		m+1	len3: Länge des Objekts der 3.StartUp-SDO als CompleteAccess (ab SubIndex 0) in Bytes			
		(m+2)-n3	Daten des Objekts der 3. StartUp-SDO als CompleteAccess (n3 = 2*((len3+1)/2)+m+1)			

Index 1600 DPS RxPDO-Map Slave

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1600:0	DPS RxPDO-Map Slave	Diese RxPDO enthält die DP Input Data des DP Slaves, die an den DP Master gesendet werden. Je in Objekt 0x8002 [▶ 180] konfiguriertem DP-Modul mit DP Input Data gibt es einen Entry in dem RxPDO-Mapping Objekt 0x1600. Falls der DP-Slave keine DP Input Data enthält, dann existiert das PDO-Mapping-Objekt 0x1600 nicht. Diese PDO ist mandatory und muss immer im PDO-Assign Objekt 0x1C12 [▶ 177] enthalten sein. SubIndex 0 enthält die Anzahl der DP Module mit DP Input Data des DP-Slaves.	UINT8	RO	
1600:01		erstes DP-Modul mit DP Input Data des DP Slaves	UINT32	RO	
...		..			
1600:m		letztes DP-Modul mit DP Input Data des DP Slaves	UINT32	RO	

Index 187F TxPDO-Par Status PDO

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
187F:0	TxPDO-Par Status PDO	PDO Parameter TxPDO 129	UINT8	RO	0x06 (6_{dez})
187F:06	Exclude TxPDOs	Hier sind die TxPDOs (Index der TxPDO Mapping Objekte) angegeben, die nicht zusammen mit TxPDO 128 übertragen werden dürfen	OCTET-STRING[10]	RO	80 1A 00 00 00 00 00 00 00 00
187F:07	TxPDO State	Der TxPDO State ist gesetzt, wenn der DP-Slave nicht im Datenaustausch ist	BOOLEAN	RO	
187F:09	TxPDO Toggle	Das TxPDO Toggle wird	BOOLEAN	RO	

Index 1A00 DPS TxPDO-Map Slave

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A00:0	DPS TxPDO-Map Slave	Diese TxPDO enthält die DP Output Data des DP Slaves, die vom DP Master empfangen wurden. Je in Objekt 0x8002 [▶ 180] konfiguriertem DP-Modul mit DP Output Data gibt es einen Entry in dem TxPDO-Mapping Objekt 0x1A00. Falls der DP-Slave keine DP Output Data enthält, dann existiert das PDO-Mapping-Objekt 0x1A00 nicht. Diese PDO ist mandatory und muss immer im PDO-Assign Objekt 0x1C13 [▶ 177] enthalten sein. SubIndex 0 enthält die Anzahl der DP Module mit DP Output Data des DP-Slaves.	UINT8	RO	
1A00:01		erstes DP-Modul mit DP Output Data des DP Slaves	UINT32	RO	
...		..			
1A00:m		letztes DP-Modul mit DP Output Data des DP Slaves	UINT32	RO	

Index 1A7F TxPDO-Map Status PDO

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1A7F:0	TxPDO-Map Status PDO	Diese PDO enthält die Statusinformationen des DP-Slaves (Objekt 0xF100 [▶ 181])	UINT8	RO	4
1A7F:01		1. PDO Mapping entry (object 0xF100 (DP Status), entry 0x01 (DP Communication Status))	UINT32	RO	0xF100:01, 08
1A7F:02		2. PDO Mapping entry (6 Bit Align)	UINT32	RO	0x0000:00, 06
1A7F:03		3. PDO Mapping entry (object 0xF100 (DP Status), entry 0x0F (TxPDO State))	UINT32	RO	0xF100:0F, 01
1A7F:04		4. PDO Mapping entry (object 0x1800 (TxPDO-Par Status PDO), entry 0x09 (TxPDO Toggle))	UINT32	RO	0x1800:09, 01

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs: wenn der DP-Slave über DP Inputs verfügt, die an den DP-Master gesendet werden, enthält das RxPDO Assign die RxPDO 0x1600, ansonsten ist es leer (Subindex 0 = 0)	UINT8	RW	0x01 (1_{dez})
1C12:01		1. zugeordnete RxPDO (Index 0x1600)	UINT16	RW	0x1600 (5632_{dez})

Index 1C13 TxPDO assign

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs: wenn der DP-Slave über DP Outputs verfügt, die vom DP-Master empfangen werden, enthält das TxPDO Assign die TxPDOs 0x1A00 und 0x1A7F, ansonsten nur die TxPDO 0x1A7F	UINT8	RW	0x02 (2_{dez})
1C13:01		1. zugeordnete TxPDO (Index 0x1A00)	UINT16	RW	0x1A00 (6656_{dez})
1C13:02		2. zugeordnete TxPDO (Index 0x1A7F)	UINT16	RW	0x1A7F (6783_{dez})

Index 1C32 SM output parameter

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	UINT16	RW	0x0000 (0_{dez})
1C32:02	Cycle time	Zykluszeit (in ns): • Zykluszeit des EtherCAT Masters	UINT32	RW	0x000F4240 (1000000_{dez})
1C32:03	Shift time	wird nicht unterstützt	UINT32	RO	0x00000000 (0_{dez})
1C32:04	Sync modes supported	Unterstützte Synchronisierungsbetriebsarten: • Bit 1 = 1: Synchron with SM 2 Event wird unterstützt • Bit 2-3 = 01: DC-Mode wird unterstützt • Bit 14 = 1: dynamische Zeiten (Messen durch Beschreiben von 0x1C32:08 [▶ 178])	UINT16	RO	0x0001 (1_{dez})
1C32:05	Minimum cycle time	wird nicht unterstützt	UINT32	RO	0x00000000 (0_{dez})
1C32:06	Calc and copy time	wird nicht unterstützt	UINT32	RO	0x00000000 (0_{dez})
1C32:08	Command	wird nicht unterstützt	UINT16	RW	0x0000 (0_{dez})
1C32:09	Delay time	wird nicht unterstützt	UINT32	RO	0x00000000 (0_{dez})
1C32:0B	SM event missed counter	wird nicht unterstützt	UINT16	RO	0x0000 (0_{dez})
1C32:0C	Cycle exceeded counter	wird nicht unterstützt	UINT16	RO	0x0000 (0_{dez})
1C32:0D	Shift too short counter	wird nicht unterstützt	UINT16	RO	0x0000 (0_{dez})
1C32:20	Sync error	wird nicht unterstützt	BOOLEAN	RO	FALSE

Index 1C33 SM input parameter

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
1C33:0	SM input parameter	Synchronisierungsparameter der Inputs (wie 0x1C32 [▶ 178])	UINT8	RO	0x20 (32_{dez})
1C33:01	Sync mode	wie 0x1C32:01 [▶ 178]	UINT16	RO	0x0000 (0_{dez})
...					
1C33:20	Sync error	wie 0x1C32:32 [▶ 178]	BOOLEAN	RO	0x00 (0_{dez})

6.2.2.2 Profilspezifische Objekte (0x6000-0xFFFF)

Die profilspezifischen Objekte haben für alle EtherCAT Slaves, die das Profil 5001 unterstützen, die gleiche Bedeutung.

Index 6000 Input Data

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
6000:0	Input Data	Dieses Objekt enthält die DP Output Data des DP Slaves (werden vom DP-Master empfangen). Je in Objekt 0x8002 [▶ 180] konfiguriertem DP-Modul mit DP Output Data gibt es einen Entry in dem Objekt 0x6000. DP Modul m entspricht dabei Subindex m, falls ein DP-Modul keine DP Output Data enthält, existiert dieser Subindex nicht. Falls der DP-Slave keine DP Output Data enthält, dann existiert das Objekt 0x6000 nicht. SubIndex 0 enthält die Anzahl der DP Module des DP-Slaves.	UINT8	RO	
6000:01		falls erstes DP-Modul DP Output Data hat: DP Output Data des ersten DP Moduls	OCTET-STRING	RO	
...					
6000:m		falls m. DP-Modul DP Output Data hat: DP Output Data des m. DP Moduls	OCTET-STRING	RO	

Index 7000 Output Data

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
7000:0	Output Data	Dieses Objekt enthält die DP Input Data des DP Slaves (werden an den DP-Master gesendet). Je in Objekt 0x8002 [▶ 180] konfiguriertem DP-Modul mit DP Input Data gibt es einen Entry in dem Objekt 0x7000. DP Modul m entspricht dabei Subindex m, falls ein DP-Modul keine DP Input Data enthält, existiert dieser Subindex nicht. Falls der DP-Slave keine DP Input Data enthält, dann existiert das Objekt 0x7000 nicht. SubIndex 0 enthält die Anzahl der DP Module des DP-Slaves.	UINT8	RO	
7000:01		falls erstes DP-Modul DP input Data hat: DP Input Data des ersten DP Moduls	OCTET-STRING	RO	
...					
7000:m		falls m. DP-Modul DP input Data hat: DP Input Data des m. DP Moduls	OCTET-STRING	RO	

Index 8000 DP Slave Parameter Set Slave

Index (hex)	Name	Bedeutung		Datentyp	Flags	Default
8000:0	DP Slave Parameter Set Slave yyy	Dieses Objekt enthält die DP Konfiguration des (n+1). konfigurierten DP Slave (0 <= n < 125). Das Objekt ist mit Complete Access zu übertragen oder es muss erst SubIndex 0 auf 0 gesetzt, dann die einzelnen SubIndexe übertragen (nicht vorhandene SubIndexe bzw. Lücken sind dabei auszulassen) und schließlich SubIndex 0 auf den richtigen Wert gesetzt werden.		UINT8	RW	0x33 (51_{dez})
8000:01	Station Address	DP Stationsadresse des DP Slaves, erlaubte Werte: 0-125		UINT16	RW	
8000:04	Device Type	DP Ident Number des DP Slaves		UINT32	RW	
8000:1D	Network Flags	reserviert für AMS über DP		UINT16	RW	0x0000 (0_{dez})
8000:1E	Network Port	reserviert für AMS über DP		UINT16	RW	0x0000 (0_{dez})
8000:1F	Network Segment Address	reserviert für AMS über DP		OCTET-STRING[6]	RW	{0}
8000:20	Sl_Flag	Sl_flag		UINT8	RW	0xA8 (168_{dez})
		Bit 0-2	Reserve, muss 0 sein			
		Bit 3	Reserve, muss 1 sein			
		Bit 4	Reserve, muss 0 sein			
		Bit 5	Reserve, muss 1 sein			
		Bit 6	Reserve, muss 0 sein			
		Bit 7	Reserve, muss 1 sein			
8000:21	Slave_Type	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
8000:22	Max_Diag_Data_Len	Reserve, muss 244 sein		UINT8	RW	0xF4(244_{dez})
8000:23	Max_Alarm_Len	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
8000:24	Max_Channel_Data_Length	Reserve, muss 240 sein		UINT8	RW	0xF0(240_{dez})
8000:25	Diag_Upd_Delay	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
8000:26	Alarm_Mode	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
8000:27	Add_Sl_Flag	Reserve, muss 1 sein		UINT8	RW	0x01 (1_{dez})
8000:28	C1_Timeout	Reserve, muss 1000 sein		UINT16	RW	0x03E8 (1000_{dez})
8000:29	Number of tolerated Data Exchange with no answer	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
8000:2A	Parallel AoE Services	Reserve, muss 5 sein		UINT8	RW	0x05 (5_{dez})
8000:2B	Reaction on no answer	Reserve, muss 0 sein		BIT1	RW	0x00 (0_{dez})
8000:2C	Restart behaviour after DP fault	Reserve, muss 0 sein		BIT1	RW	0x00 (0_{dez})
8000:2D	Master reaction after DP fault	Reserve, muss 0 sein		BIT1	RW	0x00 (0_{dez})
8000:2E	Changes of DP Inputs after DP fault	0	Daten der TxPDO n werden auf 0 gesetzt	BIT1	RW	
		1	Daten der TxPDO n werden bleiben unverändert			
8000:2F	PKW supported	Reserve, muss 0 sein		BOOLEAN	RW	FALSE
8000:30	FDL only	Reserve, muss 0 sein		BOOLEAN	RW	FALSE
8000:31	Watchdog Base 1 ms	Reserve, muss 0 sein		BOOLEAN	RW	FALSE
8000:33	Cycle Access Divider	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
8000:34	Cycle Access Modulo	Reserve, muss 0 sein		UINT8	RW	0x00 (0_{dez})
8000:35	Vendor specific reserved	Reserve, muss 0 sein		OCTET-STRING[25]	RW	{0}

Index 8002 DP CfgData Slave

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
8002:0	DP CfgData Slave	Dieses Objekt enthält die DP Cfg Data des DP Slaves.	OCTET-STRING[244]	RW	{0}

Index F000 Modular device profile

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F000:0	Modular device profile	Allgemeine Informationen des Modular Device Profiles	UINT8	RO	0x02 (2_{dez})
F000:01	Module index distance	Indexabstand 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 F100 DP Status

Index (hex)	Name	Bedeutung		Datentyp	Flags	Default
F100:0	DP Status	Dieses Objekt enthält die DP Statusdaten, die in die TxPDO 128 (Index 0x1A7F [▶ 177]) gemappt sind		UINT8	RO	
F100:01	DP Communication Status	0	Slave ist im Datenaustausch	UINT8	RO	
		128	Slave ist bereit für den Datenaustausch			
		129	Slave wartet auf das ChkCfg-Telegramm			
		130	Slave wartet auf das SetPrm-Telegramm			
F100:0E	Sync Error	ist gesetzt, wenn im Distributed Clocks Betrieb im vorangegangenen Zyklus ein Synchronisierungsfehler aufgetreten ist		BOOLEAN	RO	
F100:10	TxPDO State	ist gesetzt, wenn der DP Slave nicht im Datenaustausch ist		BOOLEAN	RO	
F100:0F	TxPDO Toggle	toggelt wenn nach einem EtherCAT Input Update erstmalig neue vom DP-Master empfangene DP Outputs in den EtherCAT Input Buffer eingetragen wurden.		BOOLEAN	RO	

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 [▶ 185]. 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!

EL6731-0000			
Hardware (HW)	**Firmware**	**Revision-Nr.**	**Release - Datum**
06 - 07	07	EL6731-0000-0016	2008/03
	08	EL6731-0000-0017	2008/04
	09	EL6731-0000-0018	2008/11
08 - 13	10	EL6731-0000-0020	2009/04
		EL6731-0000-0021	2012/01
14 - 22	11		2012/02
	12		2012/07
		EL6731-0000-0022	2012/10
		EL6731-0000-0023	2013/11
	13	EL6731-0000-0024	2014/07
23 - 35*	14	EL6731-0000-0025	2014/06
	15		2014/10
	17		2016/03
	18		2018/07
	19		2020/02
	20		2020/07
	21		2020/07
	22		2021/01
	23*		2022/02

EL6731-0010			
Hardware (HW)	**Firmware**	**Revision-Nr.**	**Release - Datum**
05 - 07	04	EL6731-0010-0016	2008/05
	05		2008/09
	06	EL6731-0010-0017	2008/11
07	07		2008/12
08 - 20	08		2009/09
	09		2010/10
		EL6731-0010-0018	2012/10
		EL6731-0010-0019	2013/11
21 – 33*	11	EL6731-0010-0021	2014/06
	12		2014/10
	13		2018/07
	14		2020/07
	15*		2020/07

*) 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 Systemmanager bietet Mechanismen, um alle drei Teile mit neuen Daten programmieren zu können, wenn der Slave dafür vorgesehen ist. Es findet üblicherweise keine Kontrolle durch den Slave statt, ob die neuen Daten für ihn geeignet sind, ggf. ist ein Weiterbetrieb nicht mehr möglich.

Vereinfachtes Update per Bundle-Firmware

Bequemer ist der Update per sog. **Bundle-Firmware**: hier sind die Controller-Firmware und die ESI-Beschreibung in einer *.efw-Datei zusammengefasst, beim Update wird in der Klemme sowohl die Firmware, als auch die ESI verändert. Dazu ist erforderlich

- dass die Firmware in dem gepackten Format vorliegt: erkenntlich an dem Dateinamen der auch die Revisionsnummer enthält, z. B. 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

a) Das Herunterladen der Firmware auf ein EtherCAT-Gerät darf nicht unterbrochen werden.

b) Eine einwandfreie EtherCAT-Kommunikation muss sichergestellt sein, CRC-Fehler oder LostFrames dürfen nicht auftreten.

c) 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:

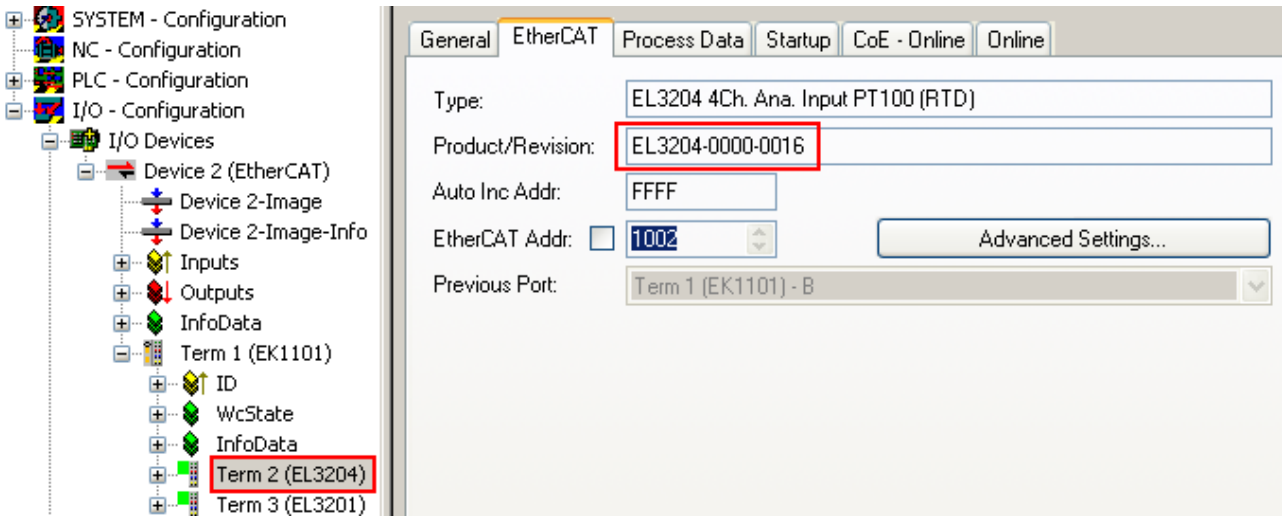


Abb. 136: Gerätekennung aus Name EL3204-0000 und Revision -0016

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.

Update von XML/ESI-Beschreibung

Die Geräterevision 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:

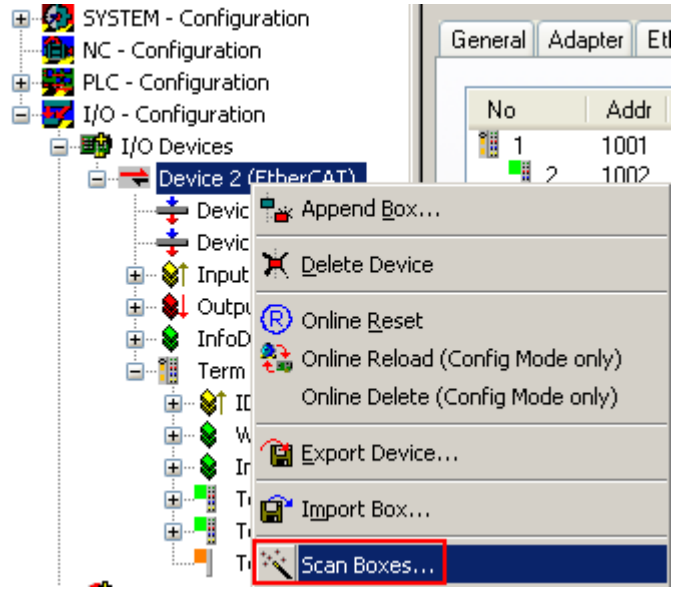


Abb. 137: Rechtsklick auf das EtherCAT Gerät bewirkt das Scannen des unterlagerten Feldes

Wenn das gefundene Feld mit dem konfigurierten übereinstimmt, erscheint

Abb. 138: Konfiguration identisch

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

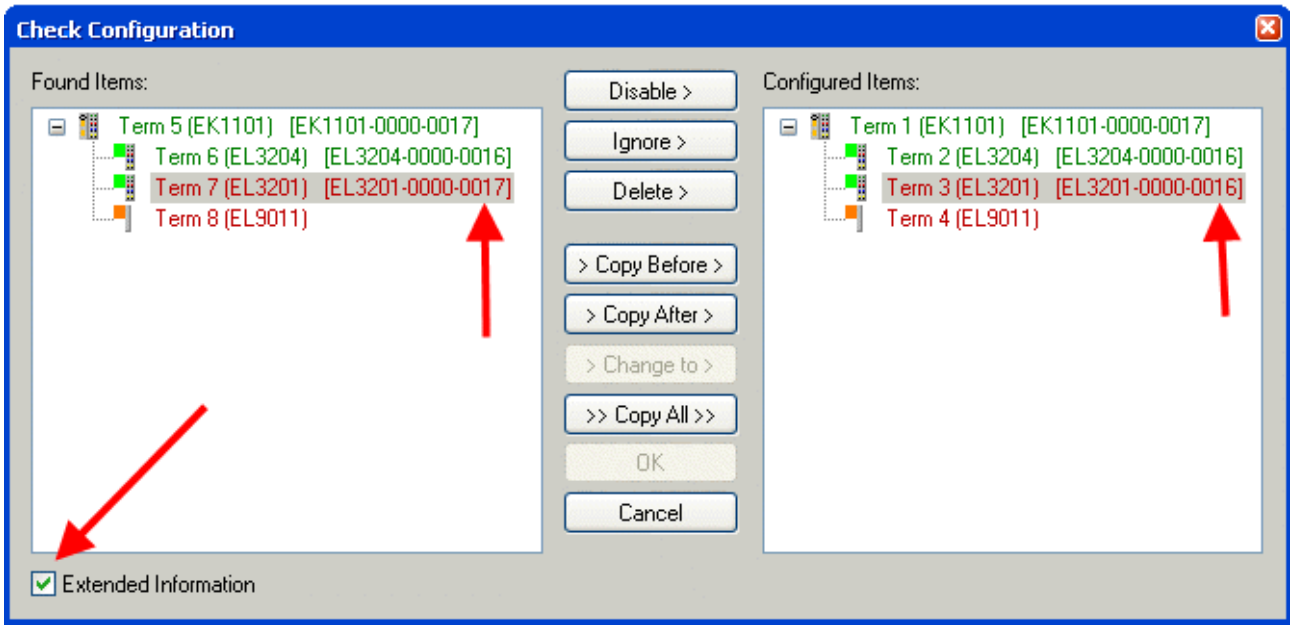


Abb. 139: Änderungsdialog

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*

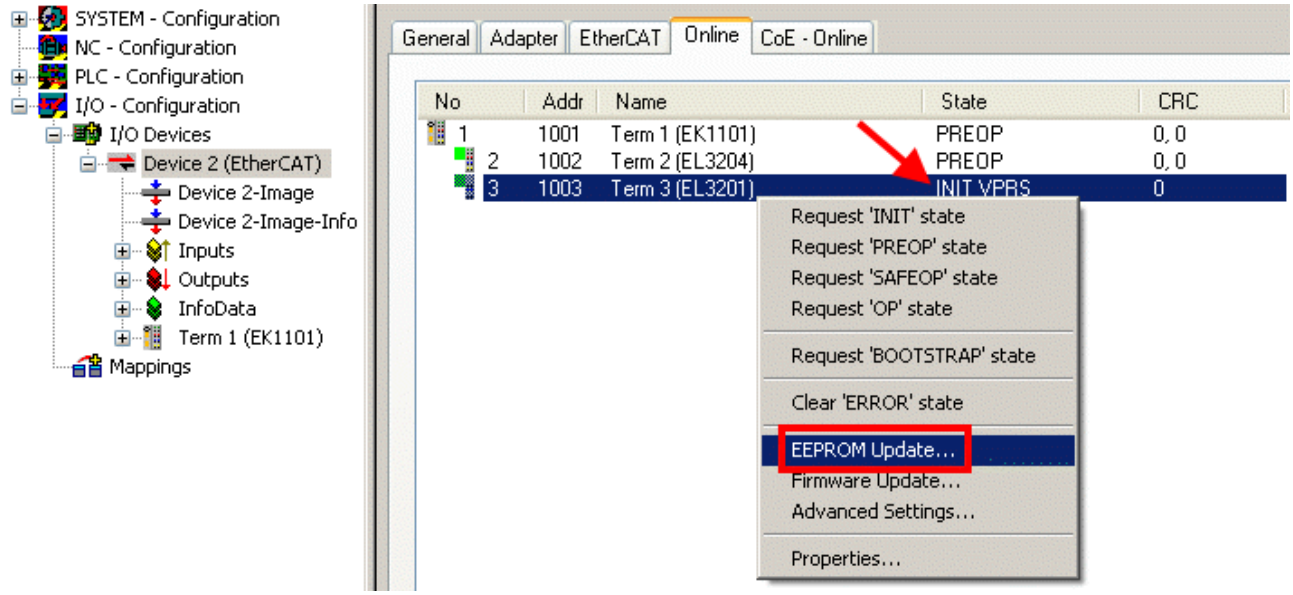


Abb. 140: EEPROM Update

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.

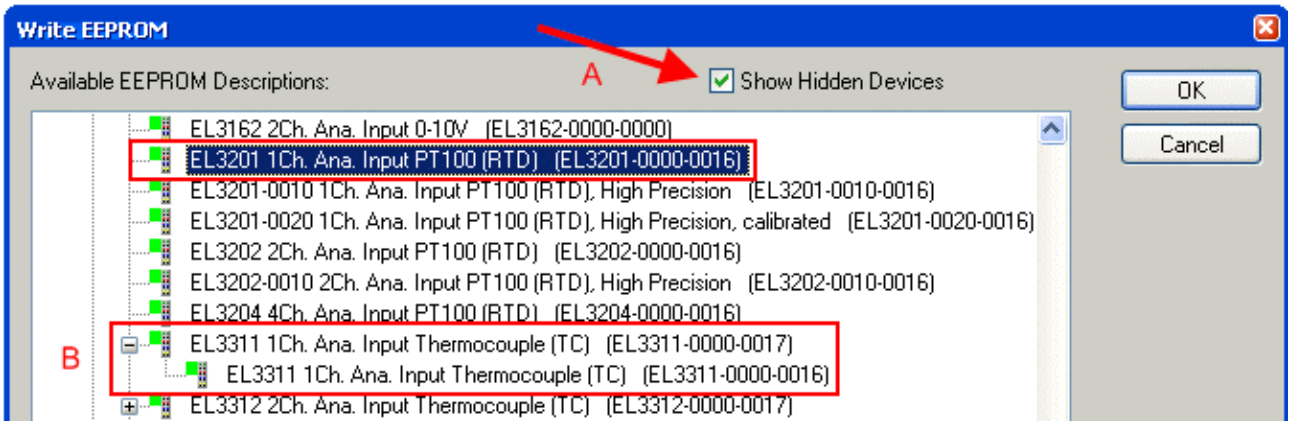


Abb. 141: Auswahl des neuen ESI

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

Änderung erst nach Neustart wirksam

Die meisten EtherCAT-Geräte lesen eine geänderte ESI-Beschreibung umgehend bzw. nach dem Aufstarten aus dem INIT ein. Einige Kommunikationseinstellungen wie z. B. Distributed Clocks werden jedoch erst bei PowerOn gelesen. Deshalb ist ein kurzes Abschalten des EtherCAT Slave nötig, damit die Änderung wirksam wird.

7.3.2 Erläuterungen zur Firmware

Versionsbestimmung der Firmware

Versionsbestimmung mit dem System-Manager

Der TwinCAT System-Manager zeigt die Version der Controller-Firmware an, wenn der Slave online für den Master zugänglich ist. Klicken Sie hierzu auf die E-Bus-Klemme deren Controller-Firmware Sie überprüfen möchten (im Beispiel Klemme 2 (EL3204) und wählen Sie den Karteireiter *CoE-Online* (CAN over EtherCAT).

CoE-Online und Offline-CoE

Es existieren zwei CoE-Verzeichnisse:
• **online:** es wird im EtherCAT Slave vom Controller angeboten, wenn der EtherCAT Slave dies unterstützt. Dieses CoE-Verzeichnis kann nur bei angeschlossenem und betriebsbereitem Slave angezeigt werden.
• **offline**: in der EtherCAT Slave Information ESI/XML kann der Default-Inhalt des CoE enthalten sein. Dieses CoE-Verzeichnis kann nur angezeigt werden, wenn es in der ESI (z. B. „Beckhoff EL5xxx.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.

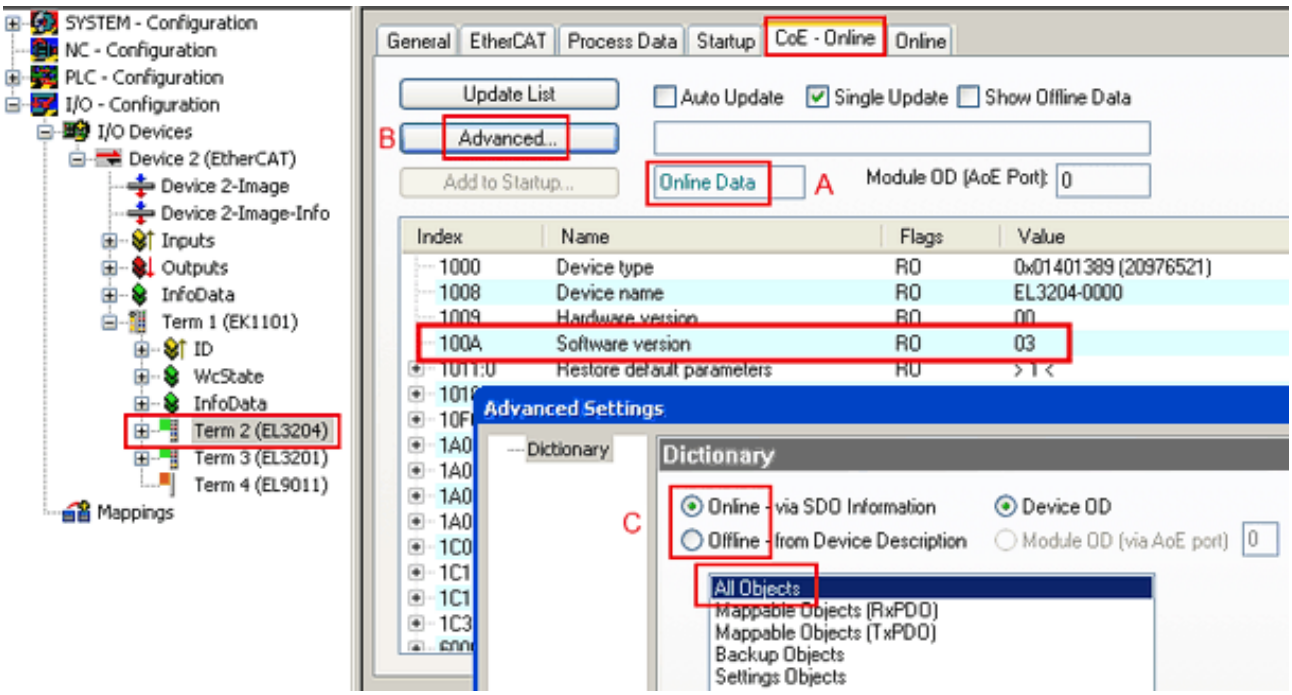


Abb. 142: Anzeige FW-Stand EL3204

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

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*.

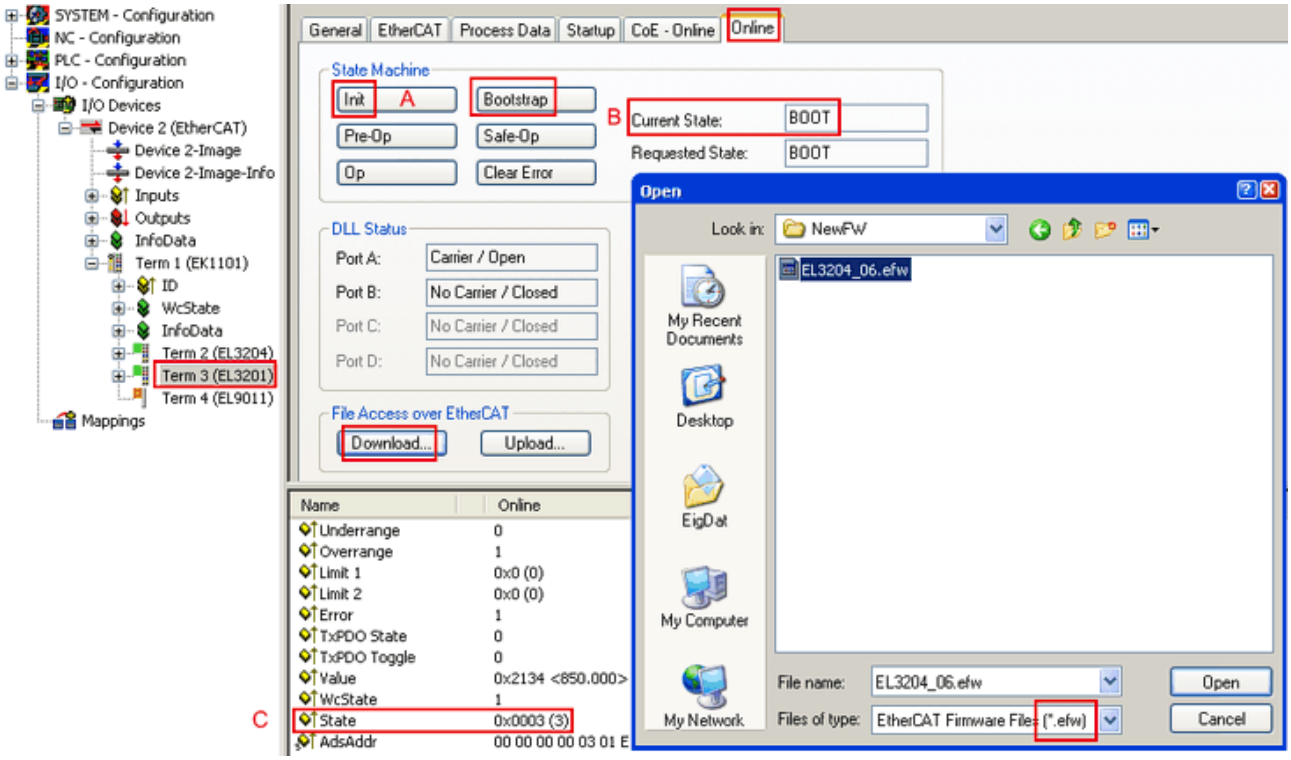
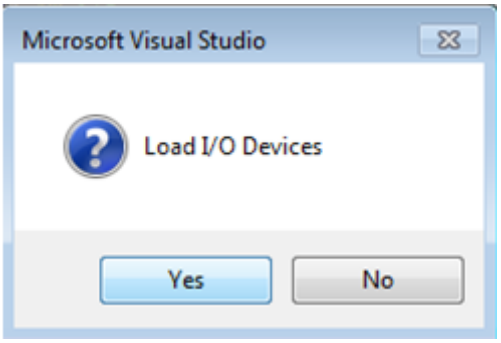
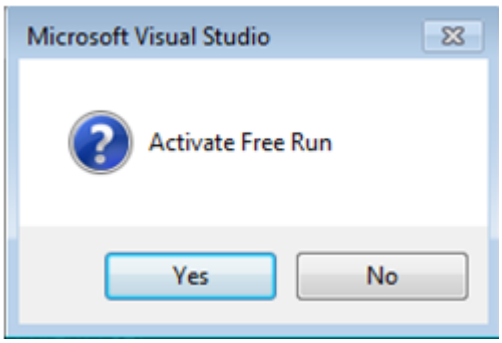


Abb. 143: Firmware Update

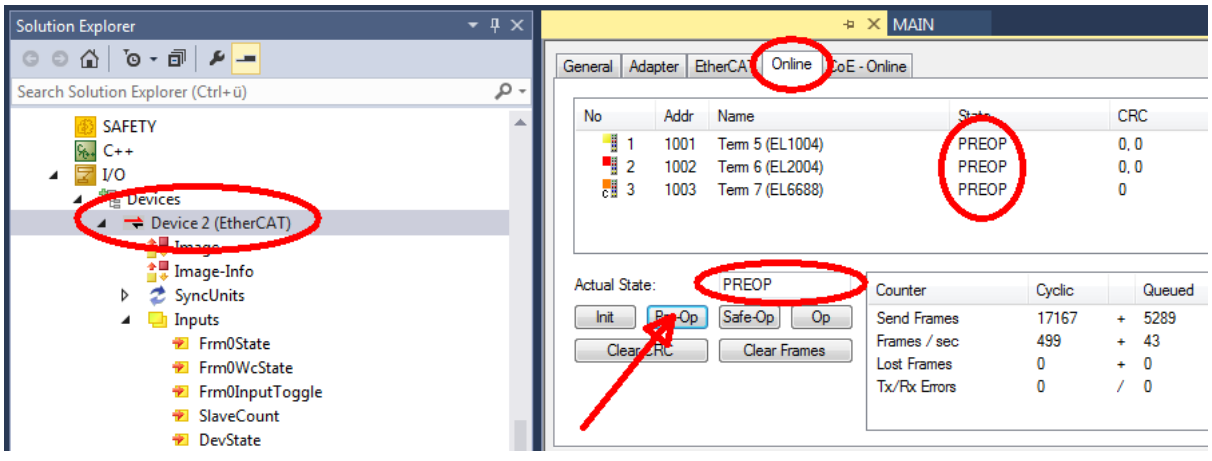
Es ist folgender Ablauf einzuhalten, wenn keine anderen Angaben z. B. durch den Beckhoff Support vorliegen. Gültig für TwinCAT 2 und 3 als EtherCAT Master.

- TwinCAT System in ConfigMode/FreeRun mit Zykluszeit >= 1ms schalten (default sind im ConfigMode 4 ms). Ein FW-Update während Echtzeitbetrieb ist nicht zu empfehlen.



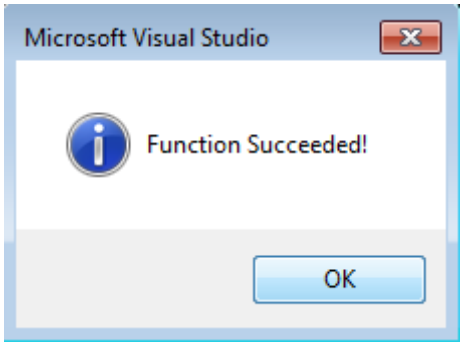


- EtherCAT Master in PreOP schalten



- 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 FPGA-Firmware *.rbf

Falls ein FPGA-Chip die EtherCAT-Kommunikation übernimmt, kann ggf. mit einer *.rbf-Datei ein Update durchgeführt werden.

- Controller-Firmware für die Aufbereitung der E/A-Signale
- FPGA-Firmware für die EtherCAT-Kommunikation (nur für Klemmen mit FPGA)

Die in der Seriennummer der Klemme enthaltene Firmware-Versionsnummer beinhaltet beide Firmware-Teile. Wenn auch nur eine dieser Firmware-Komponenten verändert wird, dann wird diese Versionsnummer fortgeschrieben.

Versionsbestimmung mit dem System-Manager

Der TwinCAT System-Manager zeigt die Version der FPGA-Firmware an. Klicken Sie hierzu auf die Ethernet-Karte Ihres EtherCAT-Stranges (im Beispiel Gerät 2) und wählen Sie den Karteireiter *Online*.

Die Spalte *Reg:0002* zeigt die Firmware-Version der einzelnen EtherCAT-Geräte in hexadezimaler und dezimaler Darstellung an.

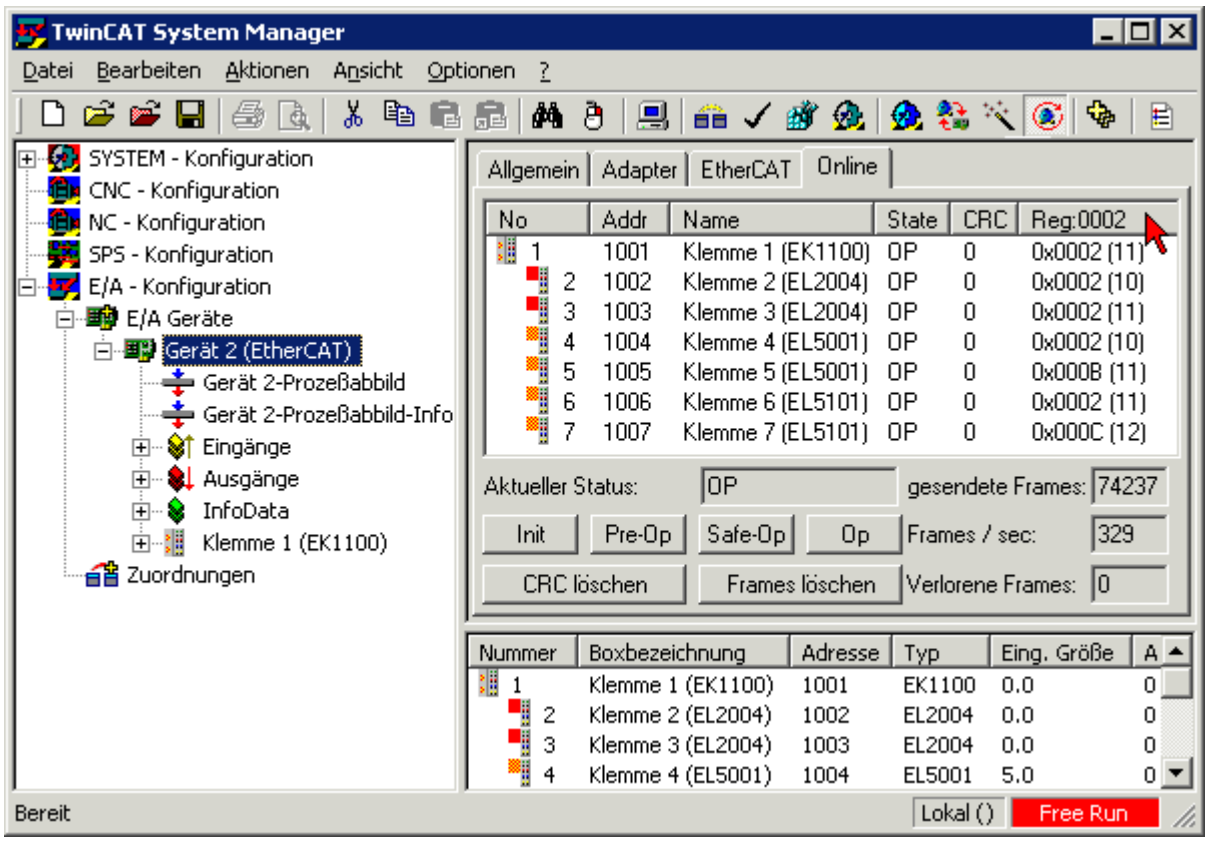


Abb. 144: Versionsbestimmung FPGA-Firmware

Falls die Spalte *Reg:0002* nicht angezeigt wird, klicken sie mit der rechten Maustaste auf den Tabellenkopf und wählen im erscheinenden Kontextmenü, den Menüpunkt *Properties*.

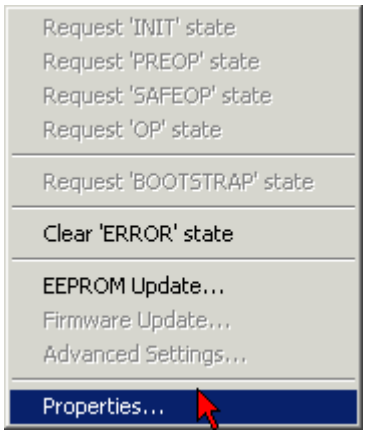


Abb. 145: Kontextmenu *Eigenschaften (Properties)*

In dem folgenden Dialog *Advanced Settings* können Sie festlegen, welche Spalten angezeigt werden sollen. Markieren Sie dort unter *Diagnose/***Online Anzeige** das Kontrollkästchen vor *'0002 ETxxxx Build'* um die Anzeige der FPGA-Firmware-Version zu aktivieren.

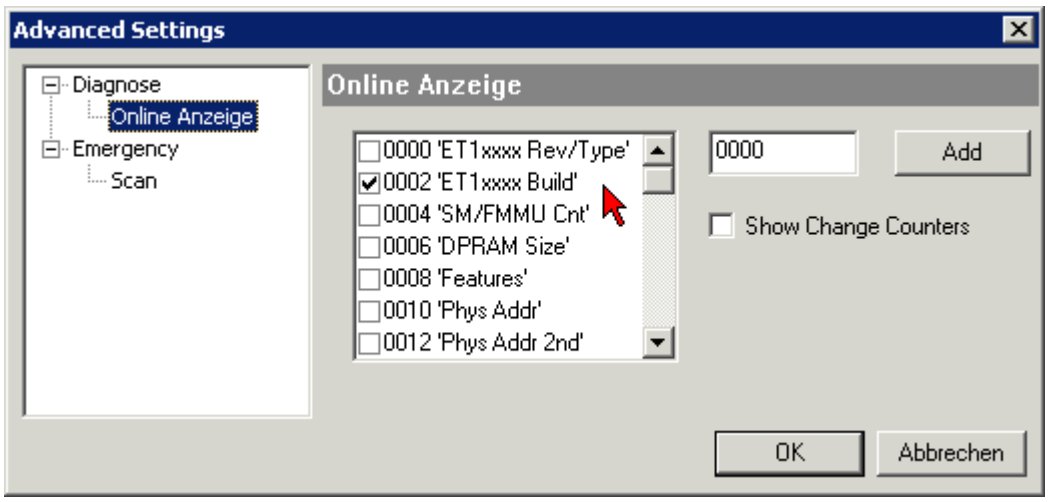


Abb. 146: Dialog *Advanced settings*

Update

Für das Update der FPGA-Firmware

- eines EtherCAT-Kopplers, muss auf diesem Koppler mindestens die FPGA-Firmware-Version 11 vorhanden sein.
- einer E-Bus-Klemme, muss auf dieser Klemme mindestens die FPGA-Firmware-Version 10 vorhanden sein.

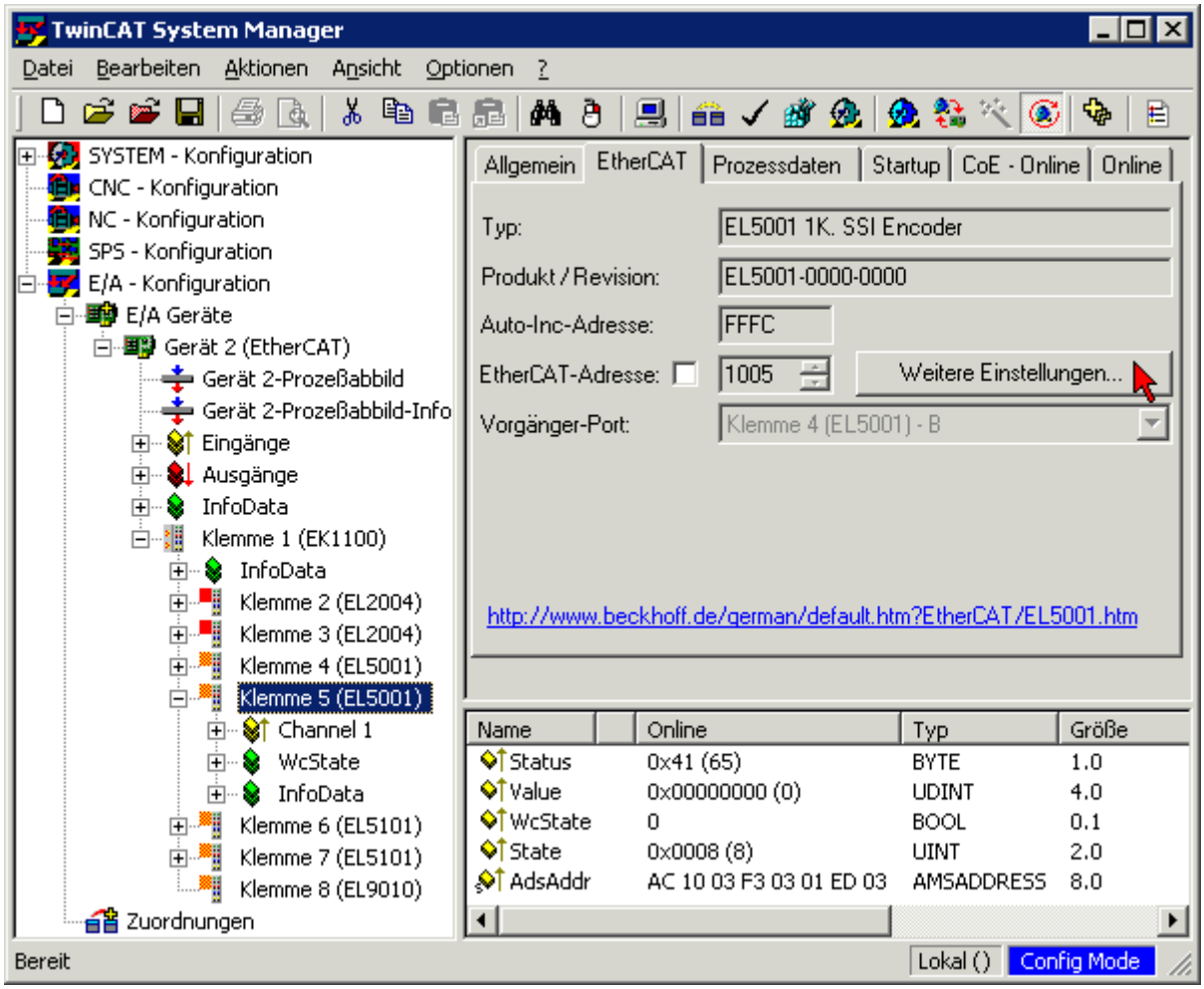
Ältere Firmware-Stände können nur vom Hersteller aktualisiert werden!

Update eines EtherCAT-Geräts

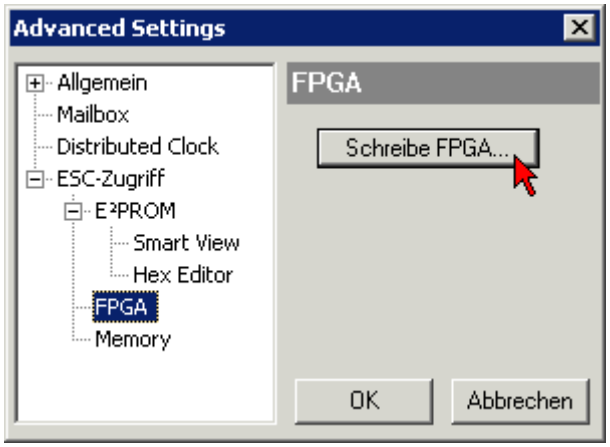
Es ist folgender Ablauf einzuhalten, wenn keine anderen Angaben z. B. durch den Beckhoff Support vorliegen:

- TwinCAT System in ConfigMode/FreeRun mit Zykluszeit >= 1 ms schalten (default sind im ConfigMode 4 ms). Ein FW-Update während Echtzeitbetrieb ist nicht zu empfehlen.

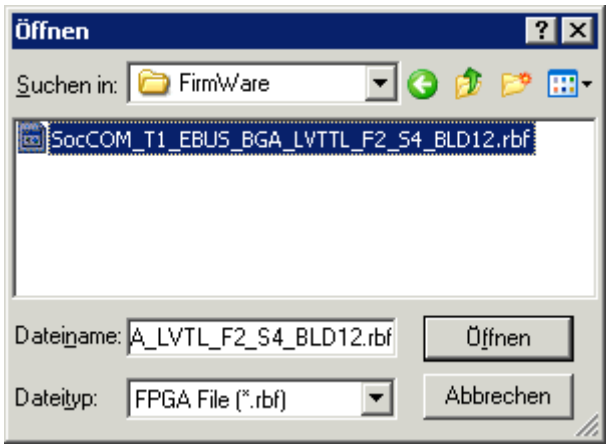
- Wählen Sie im TwinCAT System-Manager die Klemme an, deren FPGA-Firmware Sie aktualisieren möchten (im Beispiel: Klemme 5: EL5001) und klicken Sie auf dem Karteireiter *EtherCAT* auf die Schaltfläche *Weitere Einstellungen*:



- Im folgenden Dialog *Advanced Settings* klicken Sie im Menüpunkt *ESC-Zugriff/E²PROM*/FPGA auf die Schaltfläche *Schreibe FPGA*:



- Wählen Sie die Datei (*.rbf) mit der neuen FPGA-Firmware aus und übertragen Sie diese zum EtherCAT-Gerät:



- Abwarten bis zum Ende des Downloads
- Slave kurz stromlos schalten (nicht unter Spannung ziehen!). Um die neue FPGA-Firmware zu aktivieren ist ein Neustart (Aus- und Wiedereinschalten der Spannungsversorgung) des EtherCAT-Geräts erforderlich
- Kontrolle des neuen FPGA-Standes

HINWEIS

Beschädigung des Gerätes möglich!

Das Herunterladen der Firmware auf ein EtherCAT-Gerät dürfen Sie auf keinen Fall unterbrechen! Wenn Sie diesen Vorgang abbrechen, dabei die Versorgungsspannung ausschalten oder die Ethernet-Verbindung unterbrechen, kann das EtherCAT-Gerät nur vom Hersteller wieder in Betrieb genommen werden!

7.3.5 Gleichzeitiges Update mehrerer EtherCAT-Geräte

Die Firmware von mehreren Geräten kann gleichzeitig aktualisiert werden, ebenso wie die ESI-Beschreibung. Voraussetzung hierfür ist, dass für diese Geräte die gleiche Firmware-Datei/ESI gilt.

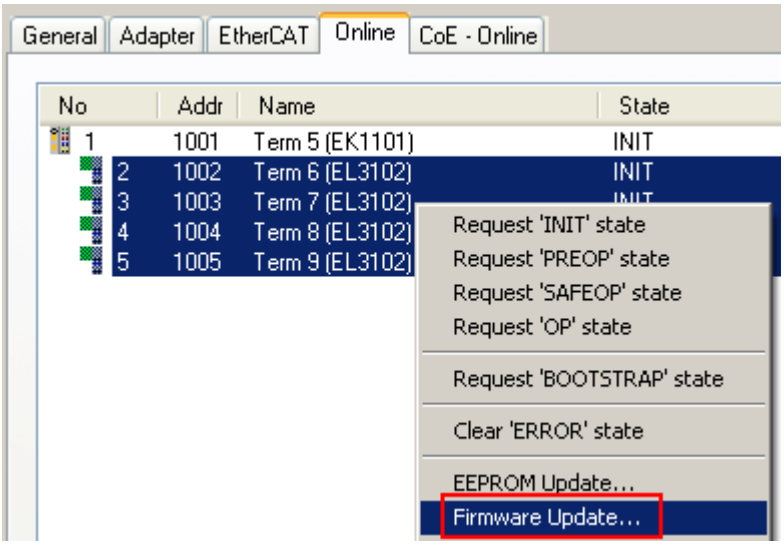


Abb. 147: Mehrfache Selektion und FW-Update

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

7.4 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.de/EL6731

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