

IndraMotion MLC 14VRS

Feldbusschnittstelle CANopen

Anwendungsbeschreibung
R911344487

Ausgabe 01



Titel IndraMotion MLC
14VRS
Feldbusschnittstelle CANopen

Art der Dokumentation Anwendungsbeschreibung

Dokumentations-Type DOK-XLCMLC-FB*CAN**V14-AP01-DE-P

Interner Ablagevermerk RS-214cced2bf5e22fd0a347ea501100e95-1-de-DE-20

Zweck der Dokumentation Die Dokumentation dient der Beschreibung der von den Systemen IndraMotion MLC unterstützten CANbus-Anschaltung.
Der Schwerpunkt dieser Dokumentation liegt in der Konfiguration, Parametrierung, Inbetriebnahme und Diagnose der Anschaltung.
Beschrieben wird ferner die Unterstützung des CANbusses im SPS-Anwenderprogramm (CAA_CiA405.library).

Änderungsverlauf

Ausgabe	Stand	Bemerkung
Ausgabe 01	2016-09	Erstausgabe CANbus für IndraMotion MLC 14VRS (Indra-Control XM2)

Schutzvermerk © Bosch Rexroth AG 2016

Alle Rechte bei Bosch Rexroth AG, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht, bei uns.

Verbindlichkeit Die angegebenen Daten dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne zu verstehen. Änderungen im Inhalt der Dokumentation und Liefermöglichkeiten der Produkte sind vorbehalten.

Redaktion DC-IA/EAI4, BAn, CCI (vha)

Inhaltsverzeichnis

	Seite
1 Zu dieser Dokumentation.....	5
1.1 Gültigkeit der Dokumentation.....	5
1.2 Gliederung der Dokumentation.....	5
1.3 Gebrauch der Sicherheitshinweise.....	6
1.3.1 Aufbau der Sicherheitshinweise.....	6
1.3.2 Erläuterung der Signalwörter und der Signalgrafik.....	6
1.3.3 Verwendete Symbole.....	7
1.3.4 Erläuterung der Signalgrafik auf dem Gerät.....	7
1.4 Erforderliche und ergänzende Dokumentationen.....	7
1.5 Begriffe und Abkürzungen.....	8
2 Features des Feldbusses.....	9
3 Nachinstallation CAN-Gerätebeschreibungen.....	11
4 CANbus Konfigurationseditor.....	15
4.1 CANbus Konfigurationseditor, Allgemeines.....	15
4.2 CANopen Master-Knoten.....	18
4.2.1 Registerkarte 'CANopen Master - Allgemein'.....	18
4.2.2 Registerkarte 'CANopen Master - CANbus Parameter'.....	18
4.2.3 Registerkarte 'CANopen Master - Status'.....	18
4.2.4 Registerkarte 'CANopen Master - Information'.....	19
4.3 CANopen Manager-Knoten.....	19
4.3.1 Registerkarte 'CANopen Manager - Allgemein'.....	19
4.3.2 Registerkarte 'CANopen Manager - Parameter'.....	21
4.3.3 Registerkarte 'CANopen Manager - E/A-Abbild'.....	21
4.3.4 Registerkarte 'CANopen Manager - Status'.....	22
4.3.5 Registerkarte 'CANopen Manager - Information'.....	22
4.4 CANopen Remote Device.....	23
4.4.1 Registerkarte 'CANopen Remote Device - Allgemein'.....	23
4.4.2 Registerkarte 'CANopen Remote Device - PDOs'.....	25
4.4.3 Registerkarte 'CANopen Remote Device - SDOs'.....	27
4.4.4 Registerkarte 'CANopen Remote Device - Parameter'.....	29
4.4.5 Registerkarte 'CANopen Remote Device - E/A-Abbild'.....	29
4.4.6 Registerkarte 'CANopen Remote Device - Status'.....	29
4.4.7 Registerkarte 'CANopen Remote Device - Information'.....	30
4.5 CANopen Modul.....	30
4.5.1 Registerkarte 'CANopen Modul - Parameter'.....	30
4.5.2 Registerkarte 'CANopen Modul - E/A-Abbild'.....	30
4.5.3 Registerkarte 'CANopen Modul - Status'.....	30
4.5.4 Registerkarte 'CANopen Modul - Information'.....	31

Inhaltsverzeichnis

	Seite
5 CANbus Slave-Konfigurationseditor.....	33
5.1 CANopen Slave-Knoten.....	34
5.1.1 Registerkarte 'CANopen Slave - Allgemein'.....	34
5.1.2 Registerkarte 'CANopen Slave - CANbus Parameter'.....	35
5.1.3 Registerkarte 'CANopen Slave - Status'.....	35
5.1.4 Registerkarte 'CANopen Slave - Information'.....	35
5.2 CANopen Local Device.....	36
5.2.1 Registerkarte 'CANopen Local Device - Allgemein'.....	36
5.2.2 Registerkarte 'CANopen Local Device - Parameter'.....	38
5.2.3 Registerkarte 'CANopen Local Device - E/A-Abbild'.....	38
5.2.4 Registerkarte 'CANopen Local Device - Status'.....	39
5.2.5 Registerkarte 'CANopen Local Device - Information'.....	39
6 Gerätedatenbank.....	41
6.1 Gerätedatenbank, Übersicht	41
6.2 Geräte verwalten.....	41
6.2.1 Gerätedatenbank, Dialog	41
6.2.2 Geräte hinzufügen	42
6.2.3 Geräte entfernen	44
7 CAA_CiA405.library.....	45
7.1 CAA_CiA405.Library, Allgemeines.....	45
7.2 Datentypen.....	46
7.2.1 Datentypen der CAA_CiA405.library.....	46
7.2.2 DEVICE_STATE.....	46
7.2.3 SDO_MODE.....	47
7.2.4 TRANSITION_STATE.....	48
7.2.5 CANOPEN_KERNEL_ERROR.....	48
7.2.6 EMCY_ERROR.....	48
7.2.7 DEVICE.....	49
7.2.8 SDO_ERROR.....	49
7.3 Globale Konstanten.....	49
7.3.1 CANOPEN_KERNEL_ERROR_CODES.....	49
7.4 Funktionsbausteine.....	50
7.4.1 Funktionsbausteine der CAA_CiA405.library.....	50
7.4.2 Allgemeine Aspekte zum Aufbau von Funktionsbausteinen.....	51
Datentypen im Vergleich zur Datenlänge.....	51
Timeout.....	51
Zusätzliche Parameter.....	51
Steuermechanismus.....	52
7.4.3 CIA405_CIA405BASE (Verhaltensmodell).....	55
CIA405BASE.....	55
7.4.4 Funktionsbausteine "Lesen und Schreiben von SDO".....	56
SDO_WRITE4.....	56
SDO_WRITE_DATA.....	59

Inhaltsverzeichnis

	Seite
SDO_READ4.....	60
SDO_READ_DATA.....	63
Obsolote Funktionsbausteine.....	64
SDO_WRITE.....	64
SDO_READ.....	66
7.4.5 Funktionsbaustein "Bezeichner von eigenem Knoten".....	67
GET_LOCAL_NODE_ID.....	67
7.4.6 Funktionsbausteine "Abfragestatus".....	68
GET_STATE.....	68
GET_CANOPEN_KERNEL_STATE.....	70
7.4.7 Funktionsbausteine "Netzwerkverwaltung".....	71
NMT.....	71
RECV_EMCY_DEV.....	73
RECV_EMCY.....	75
8 Service und Support.....	79
Index.....	81

1 Zu dieser Dokumentation

1.1 Gültigkeit der Dokumentation

Zielgruppe Diese Dokumentation richtet sich an Anwender, die sich über die CANbus-Anschaltung IndraMotion MLC (IndraControl XM2x) informieren möchten.

Erforderliche Qualifikationen: Person, die aufgrund ihrer fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie Kenntnis der einschlägigen Normen und Bestimmungen die ihr übertragenen Arbeiten beurteilen und mögliche Gefahren erkennen kann.

Anwendungsphasen Diese Dokumentation unterstützt den Anwender in den Phasen

- Konfiguration der Feldbus-Komponenten im IndraWorks,
- Feldbus-Applikation im SPS-Anwenderprogramm.

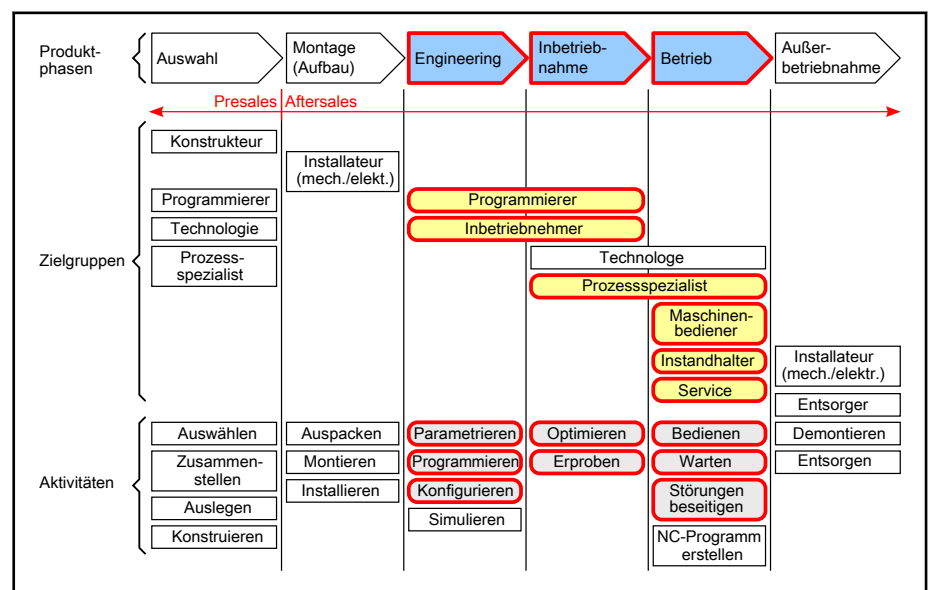


Abb. 1-1: Anwendungsphasen

1.2 Gliederung der Dokumentation

Beschrieben werden

- Eigenschaften des Feldbusses, [Features des Feldbusses, Seite 9](#)
 - Begriffe und Abkürzungen,
 - Features des Feldbusses,
 - Übertragungsarten,
- [Nachinstallation CAN-Gerätebeschreibungen, Seite 11](#)
- Projektierung der Feldbusse:
 - Konfiguration der Master-Anschaltung, [CANbus Konfigurationseditor, Seite 15](#)
 - Konfiguration und Parametrierung von Feldgeräten
 - Inbetriebnahme und Diagnose
 - Konfiguration der Slave-Anschaltung, [CANbus Slave-Konfigurationseditor, Seite 33](#)
- Gerätedatenbank, Installieren zusätzlicher Geräte und Deinstallieren nicht benötigter Geräte, [Gerätedatenbank, Seite 41](#)

Zu dieser Dokumentation

- Unterstützung des CAN-Feldbusses im SPS-Anwenderprogramm, [CAA_CiA405.library](#), Seite 45

1.3 Gebrauch der Sicherheitshinweise

1.3.1 Aufbau der Sicherheitshinweise

Die Sicherheitshinweise sind wie folgt aufgebaut:

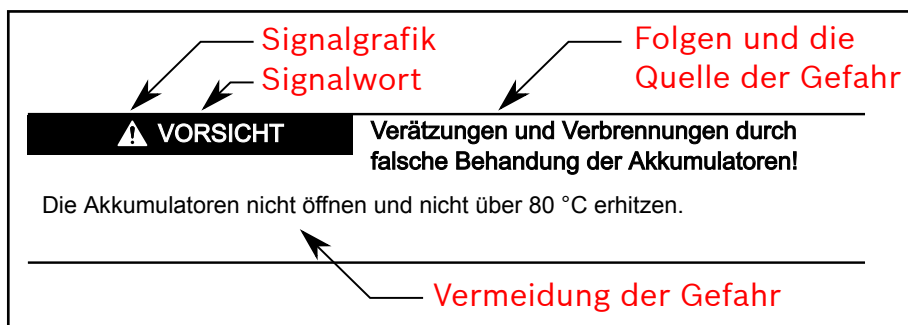


Abb. 1-2: Aufbau der Sicherheitshinweise

1.3.2 Erläuterung der Signalwörter und der Signalgrafik

Die Sicherheitshinweise in der vorliegenden Dokumentation beinhalten bestimmte Signalwörter (Gefahr, Warnung, Vorsicht, Hinweis) und gegebenenfalls eine Signalgrafik (nach ANSI Z535.6-2006).

Das Signalwort soll die Aufmerksamkeit auf den Sicherheitshinweis lenken und bezeichnet die Schwere der Gefährdung.

Die Signalgrafik (Warndreieck mit Ausrufezeichen), welche den Signalwörtern Gefahr, Warnung und Vorsicht vorangestellt wird, weist auf Gefährdungen für Personen hin.

⚠ GEFAHR

Bei Nichtbeachtung dieses Sicherheitshinweises **werden** Tod oder schwere Körperverletzung eintreten.

⚠ WARNUNG

Bei Nichtbeachtung dieses Sicherheitshinweises **können** Tod oder schwere Körperverletzung eintreten.

⚠ VORSICHT

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körperverletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

1.3.3 Verwendete Symbole

Fingerzeige werden wie folgt dargestellt:



Dies ist ein Hinweis.

Tipps werden wie folgt dargestellt:



Dies ist ein Tipp.

1.3.4 Erläuterung der Signalgrafik auf dem Gerät



Beachten Sie vor der Installation und Inbetriebnahme die Dokumentation zu dem Gerät.

1.4 Erforderliche und ergänzende Dokumentationen

Die im Folgenden aufgeführten Dokumente beinhalten zusätzliche Informationen zu diesem Thema.

IndraWorks 14VRS IndraLogic 2G SPS-Programmiersystem

DOK-IWORKS-IL2GPRO*V14-APRS-DE-P, R911343570

Diese Dokumentation beschreibt das SPS-Programmierwerkzeug IndraLogic 2G und seine Verwendung. Beschrieben werden die Basisverwendung, Erste Schritte, die Visualisierung, Menüpunkte und die Editoren.

IndraLogic XLC IndraMotion MLC 14VRS Funktionsbeschreibung

DOK-XLCMLC-FUNC***V14-APRS-DE-P, R911341699

Die Dokumentation dient der Beschreibung der Wizards, Kontextmenüs, Dialoge, Steuerungsinbetriebnahme, Gerätekonfiguration, Funktionalitäten der IndraMotion MLC.

IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G

DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918

Das vorliegende Handbuch beschreibt die systemübergreifenden SPS-Bibliotheken.

IndraWorks 14VRS Feldbusse

DOK-IWORKS-FB*****V14-APRS-DE-P, R911341484

Die Dokumentation dient der Beschreibung der von den Systemen IndraLogic XLC, IndraMotion MLC und IndraMotion MTX unterstützten Feldbus- und lokalen Peripherieanschlüssen. Der Schwerpunkt dieser Dokumentation liegt in der Konfiguration, Parametrierung, Inbetriebnahme und Diagnose der verschiedenen Peripherieanschlüssen. Sie ist die Grundlage für die Online-Hilfe.

IndraControl XFE 01.1 Extension Moduls für XM2x Betriebsanleitung DOK-CONTRL-XFE**EXTMOD-IT02-DE-P, R911345569

Montage- und Betriebsanleitung für Extension Module der XM2x

IndraControl XFE01.1-FB-20 Extension Modul CAN für XM2x Betriebsanleitung DOK-CONTRL-XFE**xxx, R911xxxxxxx

Montage- und Betriebsanleitung für Extension Module der XM2x

In Vorbereitung

Zu dieser Dokumentation

1.5 Begriffe und Abkürzungen

Begriff	Erklärung
PDO	Process Data Object: Eingänge/ Ausgänge, Prozessdaten
SDO	Service Data Object: Konfigurationsdaten
COB-ID	CAN Object Identifiers, Identifizierung des Kommunikationsobjekts als Synchronisationsnachricht
CAN ID	CAN Identifier, 11-bit CAN message identifier, ID zu Beginn jeder CAN-Nachricht auf dem Bus
NMT	Netzwerkmanagement
SYNC	Synchronisationsobjekt (SYNC) Protokoll

Tab. 1-1: *Verwendete Begriffe und Abkürzungen, weitere sind der Norm zu entnehmen.*

Modulare und kompakte Feldbusanschlaltungen

Buskoppler

Feldbus-Anschaltung (Slave),
kompakt

- Bei einem kompakten (nichtmodularen) Slave ist der Modulaufbau fest vorgegeben.

Die Module unterhalb des Slave-Objektknotens sind bereits vollständig vorhanden. In der Bibliothek sind keine weiteren Module vorhanden

modular

- Der Modulaufbau des Slaves ist variabel. Die Module (Klemmen) können individuell – aber nach Bestückungsvorschrift des Gerätes – angeordnet werden.

Modul, Klemme

Kleinste I/O-Einheit zur Bestückung einer Feldbus-Anschaltung.

Gerätebeschreibungsdateien

Die Gerätebeschreibungsdatei beschreibt die Eigenschaften des Feldbusgerätes. In dieser Datei sind alle relevanten Daten enthalten, die sowohl für das Engineering als auch für den Datenaustausch mit dem I/O-Gerät von Bedeutung sind.

Für den CAN-Bus ist folgendes Dateiformat festgelegt:

EDS Datei

Dies sind die Gerätebeschreibungen, mittels derer IndraWorks die CAN-Feldbusgeräte bekannt gemacht werden (EDS: Electronic Data Sheet).

2 Features des Feldbusses

Beschrieben wird die Realisierung einer CANopen-Schnittstelle (Master und Slave) für eine IndraMotion MLC (IndraControl XM) mittels Erweiterungsmodulen.

Die Software- und Firmware-Komponenten sind in der IndraWorks Engineering Suite bzw. MLC-Firmware enthalten.

In der Standard-Installation der IndraWorks Engineering Suite sind die CAN-Komponenten nicht sichtbar. Die CAN-Funktionalität (CAN-Erweiterungsmodul und Konfigurationsdialoge) wird durch die Nachinstallation der Gerätebeschreibungen (Device Descriptions) für das CAN-Erweiterungsmodul, sowie die CAN-Master- und -Slave-Anschaltung für den Anwender verfügbar gemacht.

An den MLC-Steuerungen können bis zu drei CAN-Erweiterungsmodule betrieben werden.

Auf jedem CAN-Erweiterungsmodul stehen zwei CAN-Schnittstellen zur Verfügung, die wahlweise als CANopen-Master oder CANopen-Slave konfiguriert werden können.

Somit sind im Maximalausbau 6 CAN-Schnittstellen an der XM2x verfügbar.



In der IndraWorks-Version 14VRS wird CAN auf Basis eines Erweiterungsmoduls an dem Steuerungstyp IndraControl XM unterstützt, im Unterschied zur IndraWorks-Version 13VRS, in der CAN auf Basis eines Funktionsmoduls an der Steuerung IndraControl Lx5 unterstützt wird.

Eine Kopplung zwischen einer XM (14VRS) und einer Lx5 (13VRS) ist möglich.

Eigenschaften der CANopen-Implementierung allgemein

- CANopen-Stack nach CANopen Communication Profile (CiA 301)
- Zustandsmaschine gemäß CiA 302
- **Error Control:** Heartbeat / Node Guarding
- Anzeige von Diagnosedaten und Emergency-Nachrichten
- Sync Producing/Consuming

Eigenschaften der CANopen-Master-Implementierung

- Auto PDO Mapping für modulare Slaves gemäß CiA 401 (modulabhängig)
- Konfigurierbares Stack-Verhalten
- Funktionsbausteine der CiA405 (CAA_CIA405.lib)
- Unterstützung mehrerer SDO-Kanäle
- Import von CANopen EDS und DCF-Dateien
- Konfigurierbares Stack-Verhalten

Eigenschaften der CANopen-Slave-Implementierung

- Statisches PDO-Mapping
- Definition von I/O-Bereichen mit automatischer Erzeugung des PDO-Mappings
- Definition von SDO-Parameterbereichen
- API zur Beeinflussung der NMT-Zustandsmaschine
- Generierung einer CiA 306- konformen EDS-Datei

3 Nachinstallation CAN-Gerätebeschreibungen

Um ein CAN-Erweiterungsmodul, sowie die CAN-Konfigurationsdialoge (-Editoren) für den Anwender verfügbar zu machen, müssen folgende Gerätebeschreibungen (Device Descriptions) in IndraWorks nachinstalliert werden:

1. EM_CANXM.devdesc.xml
Device Description für zwei CANopen Schnittstellen auf einem XM-Erweiterungsmodul
2. XM2_CANopen_Master.devdesc.xml
Device Description für CANopen Master auf einem XM-Erweiterungsmodul
3. XM2_CANopen_Slave.devdesc.xml
Device Description für CANopen Slave auf einem XM-Erweiterungsmodul
4. EmptyCANModule.devdesc.xml
Device Description für Leermodul am CAN-Knoten des CAN-Funktionsmoduls
5. CanOpenManager.devdesc.xml
Device Description für CANopen Master
6. CanLocalDevice.devdesc.xml
Device Description für CANopen Slave

Die Nachinstallation der "Device Descriptions" erfolgt über das IndraWorks-Menü **Extras ▶ Gerätedatenbank**.

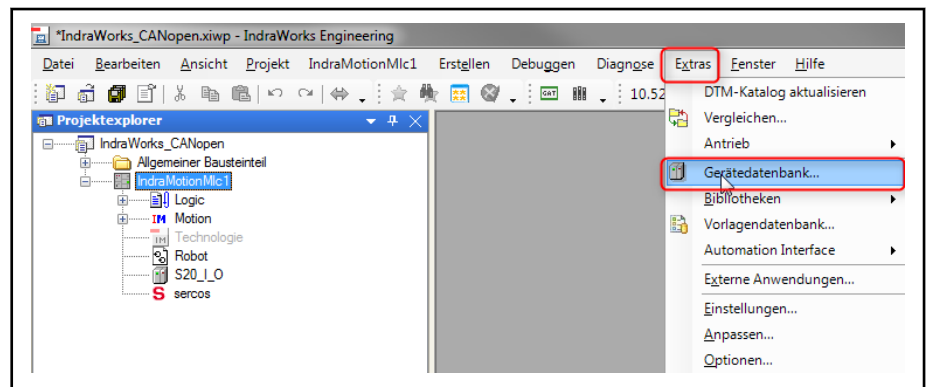


Abb. 3-1: Menüpunkt "Gerätedatenbank"

In dem folgenden Gerätedatenbank-Dialog öffnet sich über den Schaltfläche [Geräte hinzufügen] ein Browser-Fenster, in dem Sie zu dem Ablageort der "Device Descriptions" navigieren können.

Hier können Sie die aufgeführten *.devdesc.XML-Dateien auswählen und mit [Öffnen] in die IW-Gerätedatenbank hinzufügen. Eine Mehrfachmarkierung zur Auswahl ist möglich.



Im Feld "Dateityp" den Typ "Gerätebeschreibungsdateien (*.devdesc.xml)" auswählen!

Nachinstallation CAN-Gerätebeschreibungen

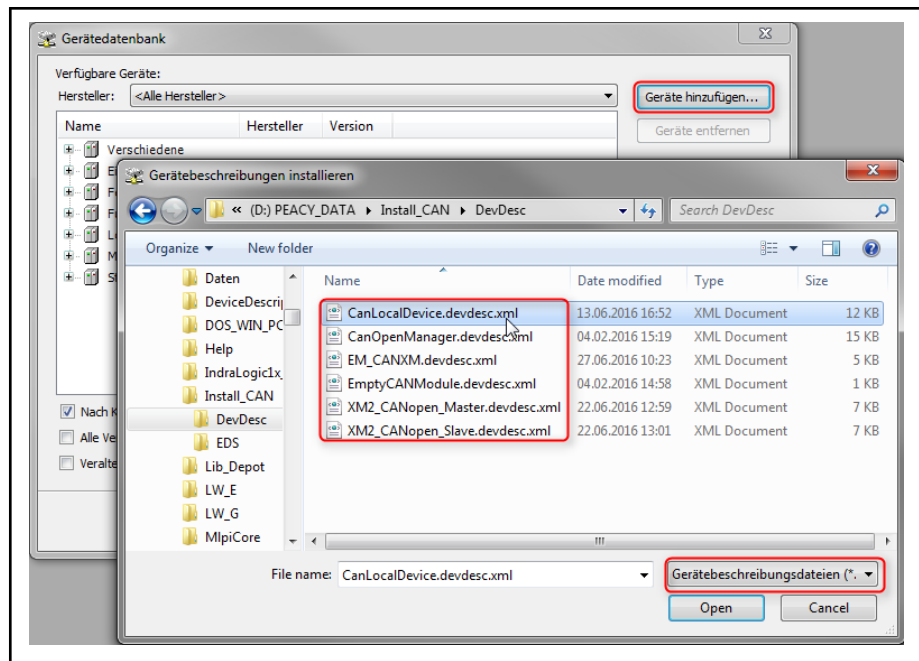


Abb. 3-2: Dialog "Gerätebeschreibungen installieren"

Anschließend erscheinen die Geräte in der IW-Geräte Datenbank:

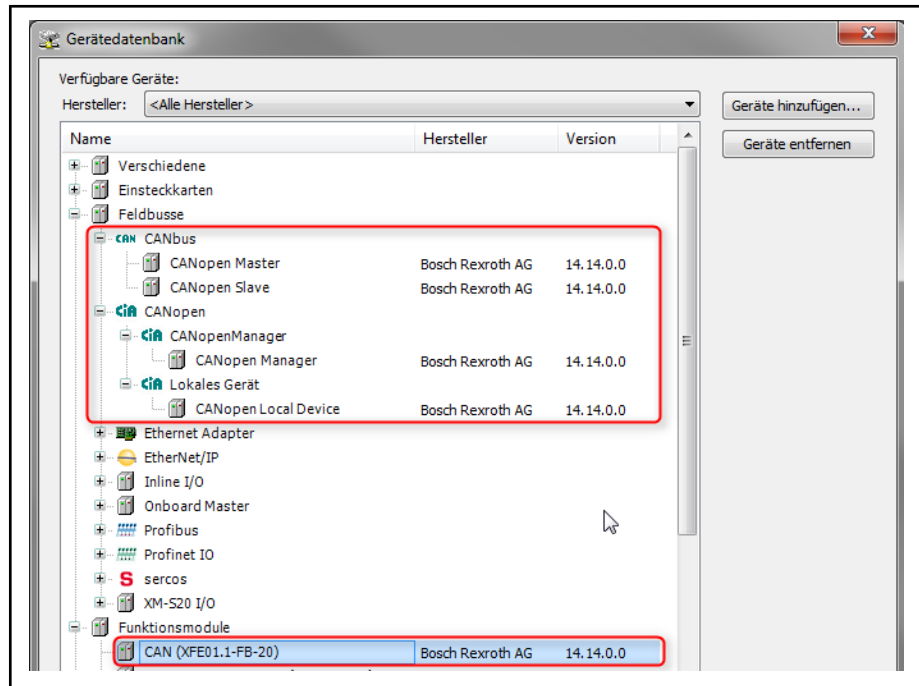


Abb. 3-3: Dialog "Geräte Datenbank"

Import der CAN-Gerätebeschreibungen über IndraWorks-Projektarchiv

Die zur Projektierung einer CAN-Applikation erforderlichen Gerätebeschreibungen werden analog zu Gerätebeschreibungen für Feldbus-Geräte in einem IndraWorks-Projektarchiv gespeichert.

Beim Wiederherstellen eines IndraWorks-Projektarchivs, das eine CAN-Applikation enthält, werden die erforderlichen CAN-Gerätebeschreibungen automatisch aus dem Archiv heraus in IndraWorks importiert. Die CAN-Komponenten sind anschließend in der IndraWorks-Installation verfügbar, ohne dass sie über den oben beschriebenen Vorgang installiert werden müssen.

Nachinstallation CAN-Gerätebeschreibungen



CAN-Projektarchiv auf Steuerung (z.B. MLC XM2) ablegen. Somit hat ein Service-Techniker nach Wiederherstellen des Projekts die erforderlichen Gerätebeschreibungen zur Verfügung, ohne dass er diese installieren muß!

4 CANbus Konfigurationseditor

4.1 CANbus Konfigurationseditor, Allgemeines

Der CANbus-Konfigurationseditor stellt Dialoge für die Konfiguration von CANopen-Geräten und -Modulen gemäß der Norm CANopen Draft Standard 301 bereit.

Zum Abbilden einer CANbus-Hardware im Projekt werden die entsprechenden Geräteobjekte im Projektextplorer eingefügt und konfiguriert. Dazu stehen dann die speziellen CANbus-Konfigurationsdialoge neben den Standarddialogen des Geräte-Editors zur Verfügung.

Die CANbus-Geräte und -Module erhalten eine Konfiguration, die ihr zeit- und fehlerbezogenes Verhalten während des Datenaustauschs definiert.

Weiterhin wird für jedes Modul das Abbilden ('Mapping') ihrer Eingänge und Ausgänge auf **PDOs** (Process Data Objects) vorgenommen, die für das Senden und Empfangen der Prozessdaten verwendet werden.

Danach wird eine Liste von **SDOs** (Service Data Objects) erzeugt, die alle Konfigurationsdaten - mit Hilfe von Anforderungs- und Antwortdiensten - auf das CANbus Zielsystem übertragen. Dort werden die Kommunikations- und Prozessdaten im "Objektverzeichnis" verwaltet.

Die CANopen-Modulkonfiguration basiert auf den Gerätebeschreibungsdateien der verwendeten Master- und Slave-Geräte und kann vom Anwender in den Konfigurationsdialogen noch angepaßt werden. Beachten Sie die Empfehlung, für Standardapplikationen und modulare Geräte den Autokonfigurationsmodus zu verwenden ([Registerkarte 'CANopen Remote Device - Allgemein', Seite 23](#))!

Folgende Schritte müssen also durchgeführt werden, um eine CANbus-Konfiguration im Projektextplorer des Projekts anzulegen:

1. [Installieren der CANopen-Geräte-dateien, Seite 15](#)
2. Abbilden der CANbus-Struktur durch Einfügen der entsprechenden Geräteobjekte im Geräte-Fenster ([Basis-Hierarchie einer CANbus Konfiguration in IndraLogic / Einfügen der CANbus Geräteobjekte, Seite 15](#)).
3. Überprüfen bzw. Anpassen der Konfigurationsparameter in den Dialogen des Geräte-Editors ([Konfiguration, Seite 16](#)).

Installieren der CANopen-Geräte-dateien

Um die CANbus-Hardware im Gerätebaum des Projekts abbilden zu können, müssen die entsprechenden Gerätebeschreibungsdateien über die Dialoge des Gerätedatenbank auf dem lokalen System installiert werden.

Dies betrifft die Beschreibungsdateien (*.devdesc.xml) für die Basismodule "CANbus" und "CANopen_Manager", wie auch die herstellersizifischen **EDS-Dateien** (Electronic Data Sheet) für die CANopen-Geräte und -Module, die im Bussystem verwendet werden.

Diese Dateien enthalten Parameterdefinitionen und Konfigurationsoptionen eines Moduls ([Gerätedatenbank, Seite 41](#)).

Basis-Hierarchie einer CANbus Konfiguration in IndraLogic / Einfügen der CANbus Geräteobjekte

- Zum Anlegen einer CAN-Konfiguration muß zunächst das Erweiterungsmodul "CAN_XFE01_1_FB_20" unterhalb des Steuerungsknotens eingefügt werden.

Die beiden Kommunikationskanäle des CAN-Erweiterungsmoduls können über die Funktion "Gerät setzen" im Kontextmenü des entsprechenden Knotens jeweils unabhängig voneinander als "CANopen_Master", oder "CANopen Slave" gesetzt werden. Im folgenden wird die Projektierung des Kommunikationskanals 1 als "CANopen Master" beschrieben.

CANbus Konfigurationseditor

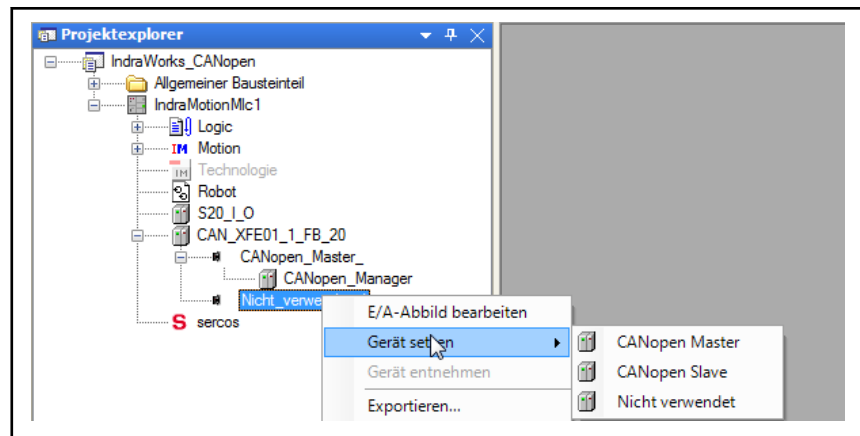


Abb. 4-1: "CANopen_Master_" im Projektexplorer, Kommunikationskanal 1 als CANopen_Master konfiguriert.

- Im CANopen_Master_-Knoten müssen die "Baudrate" für den Datentransfer auf dem Bus und die Nummer des CAN-Netzwerks, das über die CANbus-Schnittstelle angebunden werden soll überprüft bzw. angepasst werden ([Registerkarte 'CANopen Master - Allgemein', Seite 18](#)).
- Unterhalb eines CANopen_Master-Eintrags ist das obligatorische Objekt "CANopen_Manager" vorhanden.

Dieser Manager kann als "Master" des CANbus gesehen werden. Er unterstützt die CANbus-Konfiguration derzeit ausschließlich durch implizite Funktionen und somit ist keine weitere Konfiguration durch den Anwender erforderlich.

Der CANopen Manager erleichtert das Aufsetzen der Verbindung mit dem Netzwerk, speziell durch das Vereinfachen der PDO-Verknüpfungen zwischen den einzelnen Modulen.

- Unterhalb des CANopen Managers werden schließlich die einzelnen CANopen-Geräteobjekte ("Slaves") eingefügt, die mit der Hardware bereitgestellt werden.

Hier sind zwei Typen möglich: nicht-modulare Geräte und modulare Geräte.

Unterhalb von modularen Geräten können weitere CAN-Module eingehängt werden und automatisches I/O-Mapping kann verwendet werden.

Konfiguration Nachdem die CANbus-Hardware im Projektexplorer abgebildet ist, verwenden Sie die Konfigurationsdialoge für die einzelnen Objekte, um die Standardeinstellungen zu prüfen bzw. ggf. anzupassen:

Für das gerade im Baum selektierte Objekt erscheint der passende Konfigurationsdialog im Fenster des Geräte-Editors, wenn sie einen Doppelklick auf den Eintrag ausführen oder den Befehl "Objekt bearbeiten" wählen.

Die Konfigurationsmöglichkeiten eines Geräts oder Moduls hängen immer von den Definitionen in der Gerätebeschreibungsdatei ab. Beachten Sie die allgemeinen Einstellungen für den Geräte-Editor im Optionen-Dialog: Beispielsweise können die generischen Dialoge ausgeblendet werden (**Extras ▶ Optionen ▶ IndraLogic 2G ▶ Geräteditor**).

Ein CANopen-Konfigurationsdialog ist immer mit dem Namen des jeweiligen Moduls/ Geräts betitelt (wie im Projektexplorer verwendet) und kann die folgenden Unterdialoge enthalten:

Für das CANopen Master-Objekt

- [CANopen Master-Knoten, Seite 18](#), (Netzwerknummer, Baudrate)
- [Registerkarte 'CANopen Master - Allgemein', Seite 18](#), (Netzwerknummer, Baudrate)
- [Registerkarte 'CANopen Master - CANbus Parameter', Seite 18](#)
- [Registerkarte 'CANopen Master - Status', Seite 18](#)
- [Registerkarte 'CANopen Master - Information', Seite 19](#)

Für das CANopen Manager-Objekt:

- [CANopen Manager-Knoten, Seite 19](#)
- [Registerkarte 'CANopen Manager - Allgemein', Seite 19](#)
- [Registerkarte 'CANopen Manager - Parameter', Seite 21](#)
- [Registerkarte 'CANopen Manager - E/A-Abbild', Seite 21](#)
- [Registerkarte 'CANopen Manager - Status', Seite 22](#)
- [Registerkarte 'CANopen Manager - Information', Seite 22](#)

Für ein XM_CAN Device:

- [CANopen Remote Device, Seite 23](#) (Haupteinstellungen des Geräts)
- [Registerkarte 'CANopen Remote Device - Allgemein', Seite 23](#) (Haupteinstellungen des Geräts)
- [Registerkarte 'CANopen Remote Device - PDOs', Seite 25](#)
- [Registerkarte 'CANopen Remote Device - SDOs', Seite 27](#)
- [Registerkarte 'CANopen Remote Device - Parameter', Seite 29](#)
- [Registerkarte 'CANopen Remote Device - E/A-Abbild', Seite 29](#)
- [Registerkarte 'CANopen Remote Device - Status', Seite 29](#)
- [Registerkarte 'CANopen Remote Device - Information', Seite 30](#)

Für ein CANopen Modul:

- [Registerkarte 'CANopen Modul - Parameter', Seite 30](#), (Parameter, wie in der EDS-Datei definiert)
- [Registerkarte 'CANopen Modul - E/A-Abbild', Seite 30](#)
- [Registerkarte 'CANopen Modul - Status', Seite 30](#)
- [Registerkarte 'CANopen Modul - Information', Seite 31](#)

CANbus Konfigurationseditor

4.2 CANopen Master-Knoten

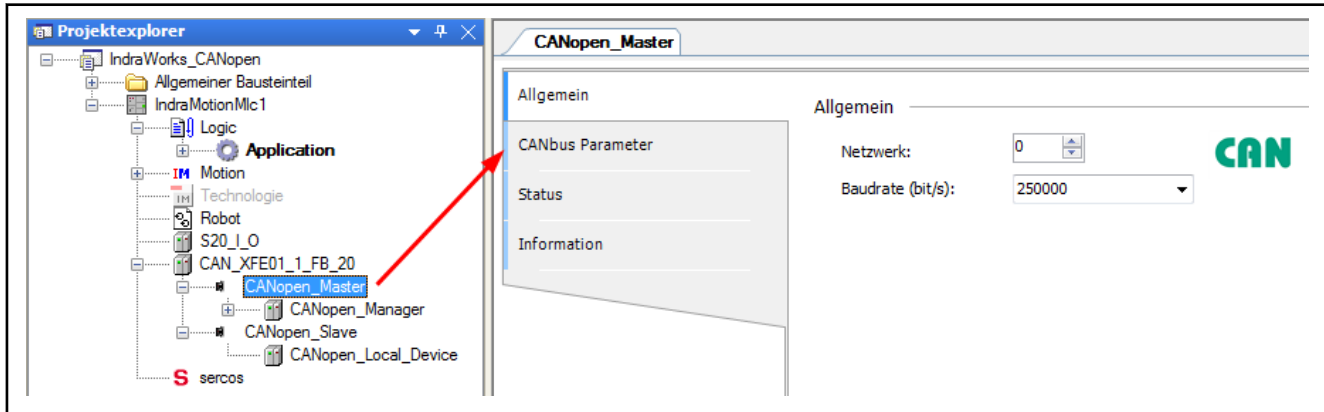


Abb. 4-2: Dialog "CANopen Master"

4.2.1 Registerkarte 'CANopen Master - Allgemein'

Das CANbus-Objekt muss als oberster Knoten einer CANbus Konfiguration in den Projektextplorer eingetragen werden ([CANbus Konfigurationseditor, Seite 15](#)).

Im zugehörigen Konfigurationsdialog wird folgendes definiert:

- **Netzwerk:** Nummer des CAN-Netzwerks, das über die CANbus- Schnittstelle angebunden werden soll.
Erlaubte Werte: 0 bis 63.



Falls an der XM2x mehrere CAN-Erweiterungsmodule und mehrere CAN-Master-/Slave-Schnittstellen projiziert werden, muß für jede CAN-Schnittstelle eine eigene, eindeutige Netzwerk-Nummer eingestellt werden!

- **Baudrate (Bits/s):** Baudrate in Bits pro Sekunden für die Datenübertragung auf dem Bus.
Der Standardwert wird aus der Gerätebeschreibungsdatei (*.devdesc) des CANbus Geräts entnommen.
Die folgenden Baudraten können eingestellt werden:
10000, 20000, 50000, 100000, 125000, 250000, 500000, 800000 und 1000000.
Die Einheit ist bits/s (Bits pro Sekunde) wobei in den Konfigurationsdaten die Baudrate in KBit/s gespeichert wird.

4.2.2 Registerkarte 'CANopen Master - CANbus Parameter'

Dieser Konfigurationsdialog für einen CANbus-Knoten ist derzeit ohne Funktion, da die Gerätebeschreibung außer der Baudrate und Netzwerknummer (NetID) keine weiteren Konfigurationsparameter enthält.

4.2.3 Registerkarte 'CANopen Master - Status'

Dieser Dialog einer CANopen Geräte- oder -Modulkonfiguration zeigt Status-Informationen (z.B. "Running", "Stopped") und spezifische Diagnosemeldungen aus dem Gerät, auch zur verwendeten Karte und dem internen Bussystem ([CANbus Konfigurationseditor, Seite 15](#)).

4.2.4 Registerkarte 'CANopen Master - Information'

Dieser Dialog einer CANopen Geräte- oder -Modulkonfiguration zeigt Informationen zum Gerät bzw. Modul, die aus der Gerätebeschreibung gelesen werden.

Mögliche Angaben: Name, Hersteller, Typ, Versionsnummer, Kategorien, Bestellnummer, Abbildung ([CANbus Konfigurationseditor, Seite 15](#)).

4.3 CANopen Manager-Knoten

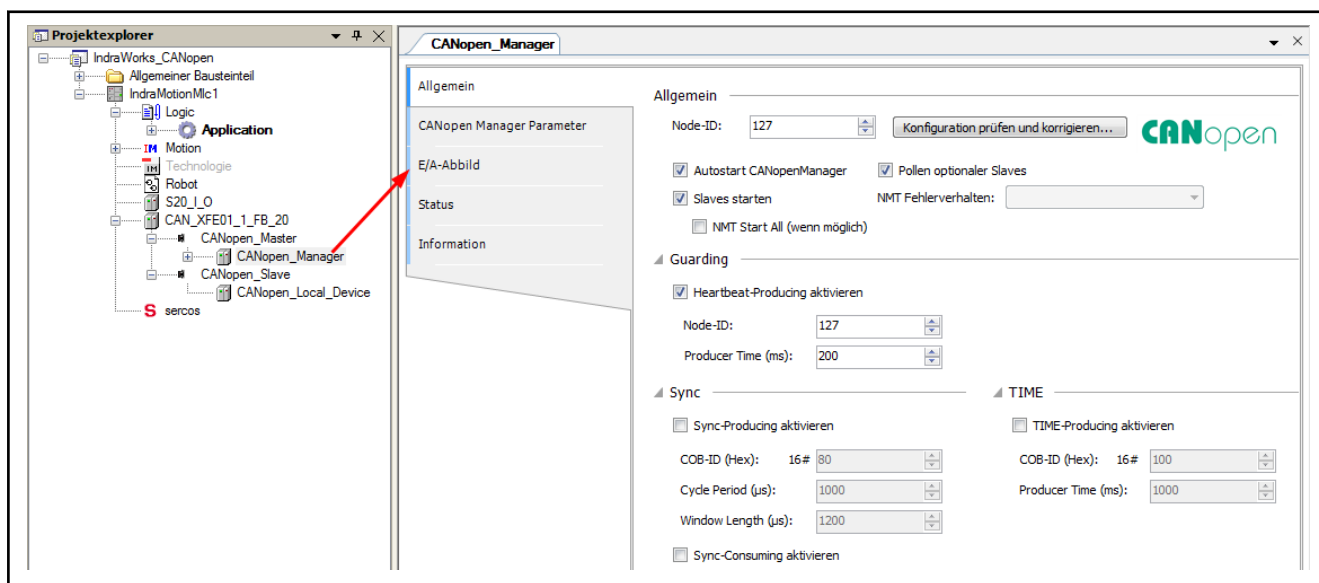


Abb. 4-3: Dialog "CANopen Manager"

4.3.1 Registerkarte 'CANopen Manager - Allgemein'

Allgemein

Node-ID	Die Knotennummer identifiziert den CANopen Manager eindeutig und entspricht der auf dem Modul gesetzten Nummer (Wert zwischen 1 und 127).
Konfiguration prüfen und korrigieren	Öffnet den gleichnamigen Dialog. Details dazu finden Sie unten.
Autostart CANopenManager	☒ : Der CANopen Manager startet automatisch (wechselt auf Mode OPERATIONAL) nachdem alle obligatorischen Slaves bereit sind. ☐ : Sie müssen den CANopen Manager von der Applikation aus starten. Dazu verwenden Sie den CiA405 NMT-Funktionsbaustein. <i>Tipp:</i> Solange der CANopen Manager nicht im Mode OPERATIONAL ist, werden keine PDOs gesendet (Ausgänge aktualisiert).
Pollen optionaler Slaves	☒ : Wenn ein Slave während der Bootsequenz keine Antwort liefert, wird er vom CANopen Manager jede Sekunde angefragt, bis er eine positive Antwort liefert. Das ständige Pollen der Slaves führt zu einer Erhöhung der Buszykluszeit, was zu einer Beeinflussung der Applikation führen kann (speziell bei Motion-Applikationen). Um dieses Verhalten zu vermeiden, können Sie das Pollen deaktivieren. Wenn das Pollen deaktiviert ist, wird ein Slave erst wieder erkannt, wenn er ein Bootup-Telegramm schickt.

CANbus Konfigurationseditor

Slaves starten	☒ : Der CANopen Manager ist für den Start der Slaves verantwortlich. ☐ : Sie müssen die Slaves von der Applikation aus starten. Dazu verwenden Sie den CiA405 NMT-Funktionsbaustein.
NMT Start All (wenn möglich)	Wenn Sie die Option Slaves starten aktiviert haben, startet der CANopen Manager alle Slaves mit einem "NMT Start All"-Befehl. Der "NMT Start All" wird nicht ausgeführt, solange optionale Slaves noch nicht zum Starten bereit sind. In dieser Situation startet der CANopen Manager jeden einzelnen Slave separat. Nur in einem Projekt ohne optionale Slaves kann ein "NMT StartAll" garantiert werden.
NMT Fehlerverhalten	Slave neustarten. Wenn ein Fehler bei der Slave-Überwachung (NMT Error Event) auftritt, wird der Slave automatisch durch den Stack neu gestartet (NMT Reset + SDO-Konfiguration + NMT Start). Slave stoppen. Wenn ein Fehler bei der Slave-Überwachung (NMT Error Event) auftritt, wird der Slave angehalten. Sie müssen dann den Slave über die Applikation mit Hilfe der CiA405 NMT-Funktionsbausteine zurücksetzen.

Guarding

Das Arbeiten mit Heartbeats ist ein alternativer Mechanismus zur Überwachung: Im Gegensatz zu der Nodeguarding-Funktionalität kann er sowohl von Master- als auch von Slave-Knoten ausgeführt werden. Normalerweise sendet der Master Heartbeats zu den Slaves.

Heartbeat-Producing aktivieren	☒ : Der Master sendet Heartbeats. Sie definieren das Zeitintervall in der Producer Time . Wenn Sie nun Slaves mit Heartbeat-Funktionalität anhängen, wird beim Slave ein Heartbeat Consuming-Eintrag für den Master angelegt. Dabei werden die Node-ID und das 1,5-fache Heartbeat-Intervall des Masters übernommen. ☐ : Bei den Slaves wird Node-Guarding aktiviert. Dabei werden die Einstellungen aus der EDS-Datei der Slaves verwendet. Sind dort keine sinnvollen Werte definiert, werden Standardwerte eingetragen. Beachten Sie, dass auch ein CANopen-Slave-Gerät als Heartbeat-Erzeuger konfiguriert werden kann.
Node-ID	Eindeutige Kennzeichnung (1-27) des Heartbeat-Erzeugers auf dem Bus.
Producer Time (ms)	Intervalllänge zwischen aufeinander folgenden Heartbeats in Millisekunden.

Sync

Sync-Producing aktivieren:	☒ : (standardmäßig deaktiviert). Der CANopen Manager sendet Sync-Telegramme. Die synchronen PDOs werden direkt nach der Synchronisationsnachricht verschickt.
COB-ID (Hex)	CAN-ID der SYNC-Nachricht. Möglicher Wertebereich: [1, 2047].
Cycle Period (µs)	Intervalllänge (in Mikrosekunden), nach der die Synchronisationsnachricht verschickt wird.
Window Length (µs)	Länge des Zeitfensters für synchrone PDOs in Mikrosekunden.
Sync-Consuming aktivieren:	☒ : (standardmäßig deaktiviert). Ein anderes Gerät muss die Synchronisationsnachrichten erzeugen, die vom CANopen Manager empfangen werden.



Wenn Sie das Sync-Producing für den CANopen Manager aktivieren, dürfen Sie bei allen anderen Busteilnehmern die Option "Sync-Producing aktivieren" nicht mehr aktivieren.

TIME

TIME-Producing aktivieren:	☒ : (standardmäßig deaktiviert). Der CANopen Manager sendet TIME-Meldungen.
COB-ID (Hex)	Communication Object Identifier, der den Zeitstempel der Meldung kennzeichnet. Standardwerte : [0, 2047], Voreinstellung 16#100.
Producer Time (ms):	Intervall in Millisekunden in dem der Zeitstempel gesendet wird. Dieser Wert muss ein Vielfaches von der Task-Zykluszeit sein. Mögliche Werte [0, 65535].



Das Laufzeitsystem muss "high resolution timestamp" unterstützen. Andernfalls wird eine Fehlermeldung ausgegeben.

Dialog "Überprüfen und Konfiguration reparieren"

Wenn Sie mehrere Geräte unterhalb des CANopen Manager einhängen, können Fehlermeldungen erscheinen, die mehrfach vergebene Node-IDs oder unzulässige COB-IDs melden.

Die Schaltfläche "Konfiguration prüfen und korrigieren" öffnet einen Dialog, in dem Sie diese Konflikte auflösen können.

Bei Konflikten mit Node-IDs oder PDO-COB-IDs können Sie durch den Befehl "Konflikte editieren..." einen Dialog mit detaillierten Angaben öffnen.

Node-ID und COB-ID-Konflikte

Doppelte Knotennummer	Liste aller Geräte, die identische IDs tragen. In den Feldern der Spalte Node-ID können Sie neue Knotennummern für die zugehörigen Geräte eintragen.
Falsche und doppelt zugewiesene PDO COB-IDs	Die aus den Gerätebeschreibungsdateien automatisch erzeugten COB-IDs können unzulässig sein. Alle inkorrekten Einträge werden mit den zugehörigen Gerätenamen, Node-IDs und Index aufgelistet. Sie haben drei Möglichkeiten, die ungültigen COB-IDs zu korrigieren: • Korrektur der angezeigten Formel zur Berechnung der COB-IDs, so dass eine gültige COB-ID resultiert. Das Abändern der Formel können Sie innerhalb des entsprechenden Tabellenfelds vornehmen. • Übernahme des automatischen Vorschlags für die COB-ID durch Klicken auf die entsprechende Schaltfläche. • Akzeptieren aller automatisch erzeugten Vorschläge durch Anklicken der Schaltfläche "Benutze vorgeschlagene COB-ID".

Korrigierte Einträge werden automatisch aus den angezeigten Listen entfernt.

Sie können Timing-Probleme automatisch durch den Befehl "Automatisches Reparieren" lösen. Der Befehl passt alle Timing-Werte auf entsprechende kompatible Werte an (Zeit sollte ein Vielfaches von der Task-Zeit sein).

4.3.2 Registerkarte 'CANopen Manager - Parameter'

Diese Registerkarte zeigt die Parameter des CANopen Managers in tabellarischer Form.

Ein Teil der Parameter ist editierbar (Wertfeld zum Editieren anklicken). Die Schriftfarbe dieser Parameter ist schwarz.

Ist die Schriftfarbe der Felder Wert und Typ grau, werden lediglich die aktuellen Werte angezeigt.

4.3.3 Registerkarte 'CANopen Manager - E/A-Abbild'

Die Registerkarte zeigt den Namen der automatisch erzeugten Instanz des Masters unter "IEC-Objekte" in der Spalte "Variable" an.

CANbus Konfigurationseditor

Sie können über die Applikation auf diese Instanz zugreifen. Somit kann beispielsweise ein Neustart, ein Stop oder eine Statusprüfung des Zielgeräts über eine Applikation erfolgen.

Bitte sehen Sie auch hierzu die Beschreibung des allgemeinen E/A-Abbild-Dialogs.

Buszyklusoptionen

Buszyklus-Task	Die Auswahlliste bietet alle Tasks an, die in der Task-Konfiguration der aktiven Applikation definiert sind, wie zum Beispiel 'MainTask'. Bei Zykluseinstellungen des übergeordneten Busses verwenden verwendet IndraLogic die Einstellungen des Vaterknotens. Wenn Sie hier keine Task definieren, gilt die in den SPS-Einstellungen eingestellte Task. Wenn Sie dort ebenfalls keine Task eingestellt haben, dann wird die höchst-priore Task als Buszyklus-Task verwendet.
----------------	--

Registerkarte '<Gerät> E/A-Abbild'

In dieser Registerkarte des generischen Geräteeditors nehmen Sie die Abbildung der Eingangs-, Ausgangs- und Speicheradressen des Steuerungsgeräts auf Variablen der Applikation vor. Damit erstellen Sie das sogenannte "E/A-Abbild" ("Mapping").

IEC Objekte

	Dieser Teil des Dialogs erscheint nur, wenn implizit eine Instanz des Geräteobjekts angelegt wird, auf die in der Applikation zugegriffen werden kann. Dies kann beispielsweise verwendet werden, um einen Bus neu zu starten oder Informationen abzufragen. Ob eine solche Instanz zur Verfügung steht und wie sie verwendet werden kann, hängt vom Gerätetyp ab.
--	---

4.3.4 Registerkarte 'CANopen Manager - Status'

Dieser Dialog einer CANopen-Geräte- oder -Modulkonfiguration zeigt Status-Informationen (z.B. "Running", "Stopped") und spezifische Diagnosemeldungen aus dem Gerät, auch zur verwendeten Karte und dem internen Bussystem ([CANbus Konfigurationseditor, Seite 15](#)).

4.3.5 Registerkarte 'CANopen Manager - Information'

Dieser Dialog einer CANopen-Geräte- oder -Modulkonfiguration zeigt Informationen zum Gerät bzw. Modul, die aus der Gerätebeschreibung gelesen werden ([CANbus Konfigurationseditor, Seite 15](#)).

Mögliche Angaben: Name, Hersteller, Typ, Versionsnummer, Kategorien, Bestellnummer, Abbildung.

4.4 CANopen Remote Device

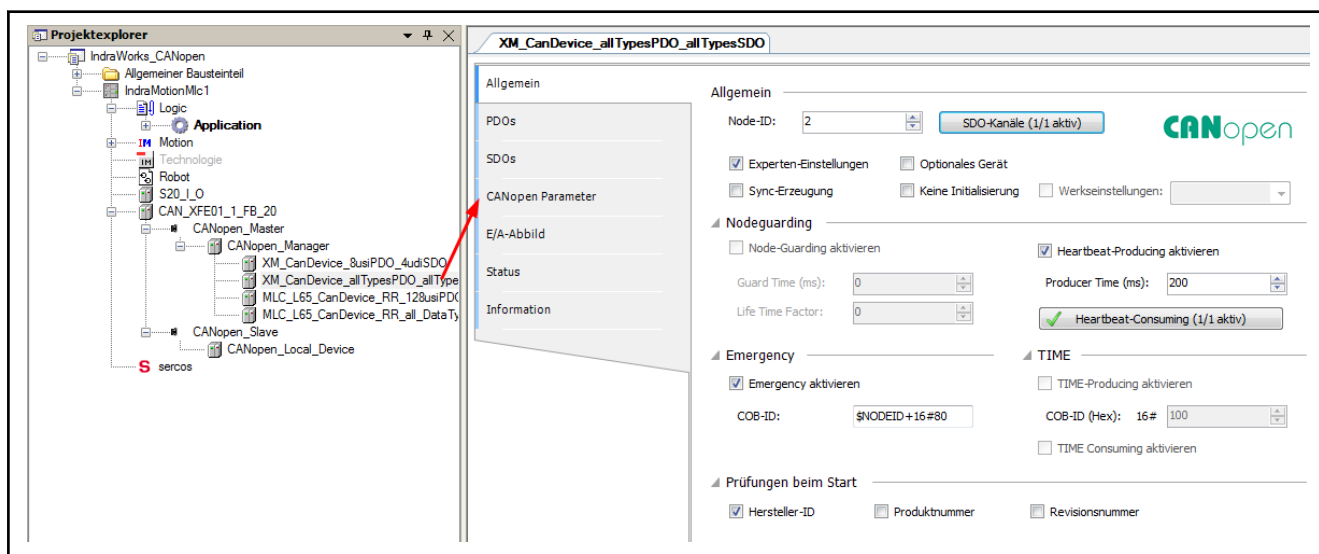


Abb. 4-4: CANopen Remote Device

4.4.1 Registerkarte 'CANopen Remote Device - Allgemein'

... In diesem Dialog eines CANopen Remote Device (Slave) sind die allgemeinen Einstellungen des CANopen-Slaves definiert.

Allgemein

Node-ID	Die Knotennummer identifiziert das CAN-Modul eindeutig. Sie entspricht der auf dem Modul (Hardware) gesetzten Nummer (Wert zwischen 1 und 127). Sie müssen die Node-ID dezimal angeben.
Experteneinstellungen	☒ : Alle Einstellungen werden angezeigt, die von der Gerätebeschreibung (EDS-Datei) für das Gerät vorgegeben sind.
SDO-Kanäle (...)	Über diese Schaltfläche öffnet sich ein Dialog, in dem Sie die in der EDS-Datei vordefinierten SDO-Kanäle aktivieren. Service Data Objects (SDOs) erlauben den Zugriff auf alle Einträge des CANopen-Objektverzeichnisses. Ein SDO erstellt einen Peer-to-Peer-Kommunikationskanal zwischen zwei Geräten (SDO-Server bzw. Client-Kanal).
Optionales Gerät	☒ : Der Slave ist optional und für den Start des CAN-Netzwerks nicht erforderlich.
Sync-Erzeugung	Nur verfügbar, wenn Sie die Option Sync-Erzeugung aktivieren des CANopen Managers deaktiviert haben. ☒ : Die E/A-Übertragung wird auf dem Bus synchronisiert. Der Slave arbeitet als Sync-Erzeuger. Die Parameter des Sync-Intervalls definieren Sie in den Einstellungen des CANopen Managers.
Keine Initialisierung	Diese Option ist für nicht konfigurierbare Slaves, die mit einer bereits gültigen Konfiguration starten ☒ : Der Master schickt keine Konfigurations-SDOs und keinen NMT-Startbefehl an den Slave. PDO-Kommunikation und Überwachung (Heartbeat, Nodeguarding) finden statt, wenn Sie dies im Konfigurator aktiviert haben. Startet der Slave nicht automatisch, so kann der Anwender mittels CiA405 NMT-Funktionsbaustein einen NMT Start-Befehl an den Slave schicken.

CANbus Konfigurationseditor

Werkseinstellungen	Die Verfügbarkeit dieser Option ist abhängig vom Inhalt der Gerätebeschreibungsdatei ☒: Standardmäßig aktiviert. Die Slave-Knoten werden auf die Standardparameter zurückgesetzt, bevor die Konfiguration auf das Gerät geladen wird oder immer dann, wenn der Slave konfiguriert wird. Welche Parameter gesetzt werden können, ist gerätespezifisch. Die konkrete Angabe erfolgt über die Auswahl eines Subindex aus der Auswahlliste. • Sub:001: Alle Parameter werden zurückgesetzt. • Sub:002: Kommunikationsparameter (Index 1000h - 1FFFh herstellerspezifische Kommunikationsparameter) werden zurückgesetzt. • Sub:003: Applikationsparameter (Index 6000h - 9FFFh herstellerspezifische Applikationsparameter) werden zurückgesetzt. • Sub:004 - Sub:127: Herstellerspezifische individuelle Auswahl von Parametern wird zurückgesetzt. • Sub:128 - Sub:254: Reserviert für zukünftige Zwecke.
Autokonfig. PDO-Mapping	Diese Option ist nur für modulare Geräte verfügbar. ☒: Das PDO-Mapping wird automatisch aus den Definitionen in der Gerätebeschreibung erzeugt und kann dann in den beiden Mapping-Dialogen nicht verändert werden. Wenn das automatisch erzeugte Mapping nicht zu Ihrer Applikation passt, können Sie die Option deaktivieren und das Mapping manuell konfigurieren. Die Aktivierung dieser Option wird für Standardapplikationen sehr empfohlen.

Node Guarding

Node Guarding ist ein veralteter Überwachungsmechanismus und sollte nicht mehr verwendet werden, da dieser RTR-Frames verwendet. Wenn möglich sollten Sie immer Heartbeat verwenden. In Einzelfällen, beispielsweise bei älteren Slaves, können Sie jedoch nur Node Guarding verwenden.	
Node Guarding aktivieren	☒: Der CANopen Manager schickt im Intervall Guard Time (ms) eine Nachricht an den Slave. Meldet sich der Slave daraufhin nicht mit der angegebenen Guard COB-ID (Communication Object Identifier), schickt der CANopen Manager diese Meldung erneut so lange, wie Sie unter Life Time Faktor definiert haben oder bis der Slave antwortet. Wenn der Slave nicht antwortet, wird er als "nicht verfügbar" gekennzeichnet.
Guard Time (ms)	Intervall der Nachrichtenversendung (standardmäßig 200 ms)
Life Time Factor	Wenn der Slave nicht antwortet, wird nach der Zeit Life Time Factor multipliziert mit Guard Time ein Node Guarding-Fehler festgestellt.
Heartbeat Producing aktivieren	☒: Das Modul sendet in den bei Producer Time (ms) angegebenen Zeitabständen Heartbeats aus.

CANbus Konfigurationseditor

Producer Time (ms)	Standardmäßig auf Wert 200 gesetzt, sofern keine spezielle Angabe oder die Angabe 0 in der Gerätebeschreibungdatei enthalten ist.
Heartbeat Consuming (...)	Heartbeat-Consuming (..): Öffnet einen Dialog "Heartbeat-Consuming"-Eigenschaften. Darin aktivieren Sie die Slaves, die Sie überwachen wollen. Die Anzahl der möglichen Slaves, die überwacht werden können, sind in der EDS-Datei beschrieben. Dazu müssen Sie das Feld "Aktivieren" aktivieren und die Node-ID der Slaves sowie die gewünschten Werte im Feld Heartbeat Time in Millisekunden eintragen. Dann überwacht der Slave die Heartbeats, die von den betreffenden Slaves (definiert durch die Node-ID) gesendet werden. Wenn keine Heartbeats mehr empfangen werden, schaltet der Slave die E/As ab. Falls ein Slave überwacht wird, wird dies mit einem grünen Haken auf der "Heartbeat Consuming"-Schaltfläche angezeigt. <i>Hinweis:</i> Wenn Sie ein Gerät mit Heartbeat-Funktionalität einfügen, werden seine Heartbeat-Einstellungen automatisch auf die des Masters (CANopen Manager) abgestimmt.

Emergency

Emergency aktivieren	☒ : Der Slave sendet bei internen Fehlern Emergency-Nachrichten mit einer eindeutigen COB-ID. Diese Meldungen können Sie mit Hilfe der Funktionsbausteine der CiA405-Bibliothek lesen. (RECV_EM CY_DEF, RECV_EM CY).
COB-ID	CAN-ID der EM CY-Nachricht. Möglicher Wertebereich: [1, 2047].

TIME

Die Verfügbarkeit dieser Funktion ist von der Gerätebeschreibung abhängig.	
TIME-Producing aktivieren	☒ : Das Gerät sendet TIME-Meldungen.
COB-ID (Hex)	Communication Object Identifier, der den Zeitstempel der Meldung kennzeichnet.
TIME Consuming aktivieren	☒ : Das Gerät verarbeitet TIME-Meldungen.

Prüfungen beim Start

Die entsprechende Information wird aus der Firmware des CANopen Slaves (Objekt 0x1018 identity object) ausgelesen und mit den Informationen aus der EDS-Datei verglichen. Im Falle von Ungleichheit werden die Konfiguration gestoppt und die Slaves nicht gestartet.	
Hersteller-ID	☒ : Überprüfen der Hersteller-ID beim Start
Produktnummer	☒ : Überprüfen der Produktnummer beim Start
Revisionsnummer	☒ : Überprüfen der Revisionsnummer beim Start

4.4.2 Registerkarte 'CANopen Remote Device - PDOs'

Dieser Dialog steht Ihnen nur im Geräteeeditor eines CANopen Slaves der Version V3.5.6.0 oder höher zur Verfügung. Er zeigt alle PDOs und deren Standardeinstellungen. Sie können in diesem Dialog neue Objekte hinzufügen oder bestehende Objekte bearbeiten oder löschen.

In der linken Hälfte stehen die PDOs, die der Slave vom Master empfängt.

In der rechten Seite stehen die PDOs, die der Slave an den Master sendet.

CANbus Konfigurationseditor

PDO hinzufügen	Öffnet den Dialog "PDO auswählen", in dem alle verfügbaren PDOs angezeigt werden. In dem Dialog wählen Sie die PDOs aus, die Sie zu den "Empfangs-PDOs" oder "Sende-PDOs" hinzufügen wollen.
Mapping hinzufügen	Öffnet den Dialog "Element aus dem Objektverzeichnis auswählen". Darin werden Objekte angezeigt, die Sie dem PDO-Mapping hinzufügen können.
Bearbeiten	Wenn Sie ein PDO selektiert haben, öffnet sich der Dialog "PDO-Eigenschaften". Wenn Sie ein PDO-Mapping selektiert haben, öffnet sich der Dialog "Element aus dem Objektverzeichnis auswählen".
Löschen	Löscht die selektierten Objekte aus der Liste
Nach oben	Verschiebt das selektierte Objekt um eine Zeile nach oben
Nach unten	Verschiebt das selektierte Objekt um eine Zeile nach unten

Dialog 'PDO-Eigenschaften'

COB-ID	Jede PDO-Nachricht muss eine COB-ID (Communication Object Identifier) besitzen. Sie können explizite Werte (Beispiel: 16#201) als auch Formeln (Beispiel: \$NODEID +16#200) eingeben.
RTR	Remote Transmission Request. Diese Option steht Ihnen nur bei Sende-PDOs zur Verfügung. ☒ : Sie können das PDO über ein RTR-Frame von außen anfragen.
Sperrzeit (x 100µs)	Sie können dieses Feld nur editieren, wenn das Gerät die Funktionalität unterstützt. Die Sperrzeit ist die minimale Zeit zwischen zwei Nachrichten dieses PDOs. Sie können die Einstellung verwenden, um zu vermeiden, dass PDOs, die bei Änderung des Wertes übertragen werden, zu häufig versendet werden. Default: "0". Mögliche Werte: 0-65535.
Übertragungstyp	• azyklisch - synchron: Das PDO wird synchron bei Änderung übertragen, jedoch nicht periodisch. (default) • zyklisch - synchron: Das PDO wird jeden nten Sync übertragen. • synchron – nur RTR: Nur bei Sende-PDOs verfügbar. Das PDO wird nach einer Synchronisationsnachricht aktualisiert, aber nicht versendet. Übertragen wird es nur auf eine explizite Anfrage hin (Remote Transmission Request) • asynchron – nur RTR: Nur bei Sende-PDOs verfügbar. Das PDO wird aktualisiert und nur auf eine explizite Anfrage hin übertragen (Remote Transmission Request) • asynchron – herstellerspezifisch: Das PDO wird nur nach bestimmten Ereignissen übertragen. • asynchron – Geräteprofil: PDO wird gemäß dem CiA-Geräteprofil übertragen.
Anzahl der Syncs	Nur bei Übertragungstyp zyklisch - synchron . Geben Sie hier an, in welchem Intervall die PDOs versendet werden. Der Wert ist ein Vielfaches der Cycle Period (µs) des CANopen Managers. Default: 1. Mögliche Werte: 1-240. Beispiel: Anzahl der Syncs = 4, Cycle Period = 1000 µs à Übertragungsintervall = 4000 µs.

Ereigniszeit (x 1ms)	Nur bei Übertragungstyp asynchron - herstellerspezifisch und asynchron - Geräteprofil . Sie können dieses Feld nur editieren, wenn das Gerät die Funktionalität unterstützt. Geben Sie hier die Zeitspanne an, die zwischen zwei Übertragungen des PDOs liegen soll (in Millisekunden). Default: "0". Mögliche Werte: 0-65535.
Verarbeitung durch CANopen Manager	☒ : Standardeinstellung. ☐ : Der CANopen Manager verarbeitet das PDO nicht mehr. Es wird nicht mehr gesendet oder empfangen.

Dialog 'Element aus dem Objektverzeichnis auswählen'

Bei modularen Slaves müssen Sie die Option "Autokonfig. PDO-Mapping" deaktivieren, damit Sie das Mapping manuell konfigurieren können.

Die Tabelle zeigt alle Objektverzeichniseinträge aus der EDS-Datei des Geräts. Im Fall von Empfangs-PDOs bietet IndraLogic hier nur Objekte mit Schreibzugriff an (flag = w), im Fall von Sende-PDOs solche mit Lesezugriff.

Name	COB-ID des PDOs oder der Name des "gemappten" Objekts, wie er in der Gerätebeschreibung und im Objektverzeichnis verwendet wird.
Index	Index des Objekts
Subindex	Subindex des Objekts
Zugriffstyp	• RW: lesen und schreiben • RO: nur lesen • WO: nur schreiben • RWW: Schreib und lesbar per SDO; schreibbar per PDO (==> RPDO, Ausgang aus Master Sicht, Eingang aus Slave Sicht). • RWR: Schreib und lesbar per SDO; lesbar per PDO (==> TPDO, Eingang aus Master Sicht, Ausgang aus Slave Sicht). • CONST=Konstante
Datentyp	Datentyp des Objekts
Standardwert	Standardwert des Objekts
Bit-Länge	Länge des Objekts

4.4.3 Registerkarte 'CANopen Remote Device - SDOs'

Während der Initialisierung des CAN-Busses überträgt IndraLogic die aktuellen Konfigurationseinstellungen mit Hilfe von Service Data Objects (SDO) zu den CAN-Modulen.

In dieser Registerkarte konfigurieren Sie die gewünschten SDOs. Sie legen fest, in welcher Reihenfolge die Objekte übertragen werden und was im Falle eines Übertragungsfehlers geschehen soll.

Die Reihenfolge der Objekte in dieser Liste entspricht der Abfolge, in der die SDOs zum Modul übertragen werden.



Wenn Sie die Option "Experteneinstellungen" für das aktuelle Gerät nicht aktiviert haben, werden Ihnen hier nur die benutzerdefinierten SDOs angezeigt.

CANbus Konfigurationseditor

SDO hinzufügen	Öffnet den Dialog "Element aus dem Objektverzeichnis auswählen", in dem alle verfügbaren SDOs angezeigt werden. Das gewählte Objekt wird nach dem selektierten Objekt eingefügt.
Bearbeiten	Öffnet den Dialog "Element aus dem Objektverzeichnis auswählen" und markiert das entsprechende Objekt. Sie können die Parameter des Objekts verändern oder das Objekt durch ein anderes ersetzen.
Löschen	Löscht die selektierten Objekte aus der Liste.
Nach oben	Verschiebt das selektierte Objekt um eine Zeile nach oben
Nach unten	Verschiebt das selektierte Objekt um eine Zeile nach unten
Abbruch bei Fehler	☒ : Wenn ein Fehler für dieses SDO erkannt wird, hält der Stack die Konfigurationsphase des aktuellen Slaves an. Der Slave verbleibt im Status PREOPERATIONAL.
Sprung zu Zeile bei Fehler	☒ : Die Übertragung wird mit dem SDO fortgesetzt, das Sie in der in Spalte "Nächste Zeile" angegeben haben.
Nächste Zeile	Zeilennummer, bei der im Fehlerfall die Abarbeitung fortgesetzt wird.
SDO-Timeout (ms)	Timeout für die SDO-Übertragung. Wenn der Slave nicht in dieser Zeit auf die SDO-Anfrage antwortet, dann wird die Übertragung mit Timeout abgebrochen.
Alle SDOs erzeugen	☒ : Erzeugt für alle schreibbaren Objekte ab Index 16#2000, für die ein Standardwert im EDS angegeben ist, ein SDO. Die Option sollte nur von Experten verwendet werden und ist im Standardfall zu deaktivieren.
Vollständige PDO-Konfiguration schreiben	☒ : Diese Option erzwingt, dass alle PDO-Konfigurations-Objekte geschrieben werden. Damit stellen Sie sicher, dass die Einstellungen im Projekt denen des Slaves entsprechen. ☐ : PDOs werden nicht explizit deaktiviert. Voraussetzung ist, dass die Option Werkseinstellungen in den allgemeinen Einstellungen des Slaves aktiviert ist und die PDOs im EDS ebenfalls deaktiviert sind. Falls die Standardwerte im EDS nicht den "Werkseinstellungen" der Slave Firmware entsprechen, kann dieses Vorgehen Probleme verursachen. In dieser Situation sollten Sie diese Option aktivieren.

Dialog 'Element aus dem Objektverzeichnis auswählen'

Die Tabelle zeigt alle per SDO schreibbaren Objektverzeichniseinträge aus der EDS-Datei des Geräts an, die nicht größer als 4 Byte sind. Bevor Sie ein SDO zur Auswahl im SDO-Dialog hinzufügen, können Sie seine Parameter in den Feldern unterhalb der Tabelle verändern. Somit können Sie auch - durch das Eingeben eines neuen Index/Subindex-Wertes - ein SDO anlegen, das in der EDS-Datei nicht beschrieben ist.

Name	COB-ID des PDOs oder der Name des "gemappten" Objekts, wie er in der Gerätebeschreibung und im Objektverzeichnis verwendet wird.
Index	Index des Objekts
Subindex	Subindex des Objekts
Zugriffstyp	• RW: lesen und schreiben • RO: nur lesen • WO: nur schreiben • RWW: Schreib und lesbar per SDO; schreibbar per PDO (==> RPDO, Ausgang aus Master Sicht, Eingang aus Slave Sicht). • RWR: Schreib und lesbar per SDO; lesbar per PDO (==> TPDO, Eingang aus Master Sicht, Ausgang aus Slave Sicht).

Datentyp	Datentyp des Objekts
Standardwert	Standardwert des Objekts
Bit-Länge	Länge des Objekts
Wert	

4.4.4 Registerkarte 'CANopen Remote Device - Parameter'

Diese Registerkarte zeigt die Parameter des CANopen Remote Device in tabellarischer Form.

Ein Teil der Parameter ist editierbar (Wertfeld zum Editieren anklicken). Die Schriftfarbe dieser Parameter ist schwarz.

Ist die Schriftfarbe der Felder Wert und Typ grau, werden lediglich die aktuellen Werte angezeigt.

4.4.5 Registerkarte 'CANopen Remote Device - E/A-Abbild'

Die Registerkarte zeigt den Namen der automatisch erzeugten Instanz des Slaves unter IEC-Objekte in der Spalte Variable an. Damit können Sie über die Applikation auf diese Instanz zugreifen. Somit kann zum Beispiel ein Neustart, ein Stop oder eine Statusprüfung des Slaves über eine Applikation erfolgen.

In der Registerkarte werden Projektvariablen auf die Ausgänge und Eingänge des Slaves abgebildet.



Bei einem modularen Slave können Sie in diesem Dialog keine Projektvariablen auf E/As abbilden. Dies erfolgt im Editor der untergeordneten Module.

Bitte sehen Sie auch hierzu die Beschreibung des allgemeinen E/A-Abbild-Dialogs.

Mapping zurücksetzen	IndraLogic setzt die Mapping-Einstellungen auf die durch die Gerätebeschreibungsdatei definierten Standardwerte zurück.
Variablen aktualisieren	Festlegung für das Geräteobjekt bezüglich der Aktualisierung der E/A-Variablen. Der Standardwert ist in der Gerätebeschreibung definiert: Einstellungen des übergeordneten Geräts verwenden: Aktualisierung gemäß den Einstellungen des übergeordneten Geräts. Aktiviert 1 (Buszyklustask verwenden, wenn in keiner Task verwendet): IndraLogic aktualisiert die E/A-Variablen in der Buszyklustask, falls sie in keiner anderen Task verwendet werden. Aktiviert 2 (immer in Buszyklustask): IndraLogic aktualisiert alle Variablen in jedem Zyklus der Buszyklustask, unabhängig davon ob sie verwendet werden und ob sie auf einen Ein- oder Ausgangskanal abgebildet sind.

4.4.6 Registerkarte 'CANopen Remote Device - Status'

Dieser Dialog einer CANopen-Geräte- oder -Modulkonfiguration zeigt Status-Informationen (z.B. "Running", "Stopped") und spezifische Diagnosemeldungen aus dem Gerät, auch zur verwendeten Karte und dem internen Bussystem.

CANbus Konfigurationseditor

4.4.7 Registerkarte 'CANopen Remote Device - Information'

Dieser Dialog einer CANopen-Geräte- oder -Modulkonfiguration zeigt Informationen zum Gerät bzw. Modul, die aus der Gerätebeschreibung gelesen werden.

Mögliche Angaben: Name, Hersteller, Typ, Versionsnummer, Kategorien, Bestellnummer, Abbildung.

4.5 CANopen Modul

CANopen Module sind modulare Baugruppen, die Sie unterhalb eines CANopen Remote-Devices einfügen.

4.5.1 Registerkarte 'CANopen Modul - Parameter'

Der zugehörige Dialog ist im Geräte-Editor des CANopen-Managers nur dann verfügbar, wenn die Option "Generische Konfigurationseditoren anzeigen" aktiviert wurde (**Extras ▶ Optionen ▶ IndraLogic 2G ▶ Geräteditor**).

Der Dialog korrespondiert zum Dialog "Parameter", der vom Geräte-Editor auch für andere Gerätemodule bereitgestellt wird.

In diesem Dialog einer CANopen-Konfiguration eines CANopen Moduls können die Parameter des Moduls, die aus der Modulbeschreibung entnommen sind, angezeigt und bearbeitet werden. Es handelt es sich um den Standard-Konfigurationsdialog eines Geräte-Editors zur Parametrierung eines Geräts oder Moduls.

4.5.2 Registerkarte 'CANopen Modul - E/A-Abbild'

Die Registerkarte zeigt den Namen der automatisch erzeugten Instanz des Moduls unter "IEC-Objekte" in der Spalte "Variable" an. Damit können Sie über die Applikation auf diese Instanz zugreifen. Somit kann zum Beispiel ein Neustart, ein Stop oder eine Statusprüfung des Moduls über eine Applikation erfolgen.

In der Registerkarte werden Projektvariablen auf die Ausgänge und Eingänge des Moduls abgebildet.

Bitte sehen Sie auch hierzu die Beschreibung des allgemeinen E/A-Abbild-Dialogs.

Mapping zurücksetzen	IndraLogic setzt die Mapping-Einstellungen auf die durch die Gerätebeschreibungsfeld datei definierten Standardwerte zurück.
Variablen aktualisieren	Festlegung für das Geräteobjekt bezüglich der Aktualisierung der E/A-Variablen. Der Standardwert ist in der Gerätebeschreibung definiert: Einstellungen des übergeordneten Geräts verwenden: Aktualisierung gemäß den Einstellungen des übergeordneten Geräts. Aktiviert 1 (Buszyklustask verwenden, wenn in keiner Task verwendet): IndraLogic aktualisiert die E/A-Variablen in der Buszyklustask, falls sie in keiner anderen Task verwendet werden. Aktiviert 2 (immer in Buszyklustask): IndraLogic aktualisiert alle Variablen in jedem Zyklus der Buszyklustask, unabhängig davon ob sie verwendet werden und ob sie auf einen Ein- oder Ausgangskanal abgebildet sind.

4.5.3 Registerkarte 'CANopen Modul - Status'

Dieser Dialog einer CANopen Geräte- oder Modulkonfiguration zeigt Status-Informationen (z.B. "Running", "Stopped") und spezifische Diagnosemeldungen.

gen aus dem Gerät, auch zur verwendeten Karte und dem internen Bussystem.

4.5.4 Registerkarte 'CANopen Modul - Information'

Dieser Dialog einer CANopen Geräte- oder Modulkonfiguration zeigt Informationen zum Gerät bzw. Modul, die aus der Gerätebeschreibung gelesen werden. Mögliche Angaben: Name, Hersteller, Typ, Versionsnummer, Kategorien, Bestellnummer, Abbildung.

5 CANbus Slave-Konfigurationseditor

Der CANbus Slave-Konfigurationseditor stellt Dialoge für die Konfiguration eines lokalen CANbus Slave-Geräts bereit, das im Gerätebaum unterhalb eines CAN-Erweiterungsmoduls angelegt werden kann. Es ermöglicht, das IndraLogic-Laufzeitsystem als CANopen-Slave zu verwenden.

Aus den im Konfigurationseditor gesetzten Parametern kann direkt eine Slave-Gerätebeschreibungsdatei *.eds erzeugt werden, die dann in einem beliebigen Standard-CANopen-Konfigurator, wie beispielsweise IndraLogic verwendet werden kann.

Zur Standard-CANbus-Gerätekonfiguration siehe: [CANbus Konfigurationseditor, Seite 15](#)

Wie für die anderen CAN-Module werden dem Standard-Geräte-Editor auch für die Konfiguration des hier beschriebenen "lokalen" CANbus Slaves spezielle Dialoge hinzugefügt.

Unterstützte Funktionalitäten

Das CANopen-Slave Gerät, das hier konfiguriert werden kann, unterstützt die folgenden Funktionalitäten:

- SDO Server Funktionalität (Nur "Expedited Mode")
- NMT-Client
- Guarding: Heartbeat Producing, Heartbeat Consuming (nur ein "Consumer"-Eintrag), Nodeguarding
- Sync producing/consuming
- Emergency (EMCY)
- Synchrone und asynchrone PDOs (alle Transmission Type spezifiziert in CiA301)
- Inhibit Time, Event Timer für PDOs
- Statisches PDO Mapping
- Max. 16 TxPDOs und 16 RxPDOs
- Lesen und Schreiben von Parametern über SDO

Folgendes wird nicht unterstützt:

- TIME
- Dynamisches PDO Mapping

Hierarchie der CANbus-Konfiguration in IndraLogic /Einfügen des CANbus Slave-Geräts

Der Basisknoten, d.h. der oberste Eintrag im CANbus-Konfigurationsbaum, muss ein CANbus-Objekt sein. Siehe hierzu [CANbus Konfigurationseditor, Seite 15](#)

Zum Anlegen einer CAN-Konfiguration muß zunächst das Erweiterungsmodul "CAN_XFE01_1_FB_20" unterhalb des Steuerungsknotens eingefügt werden.

Das lokale CANopen-Slave-Gerät wird (analog zu dem CANopen-Master) über die Funktion "Gerät setzen" im Kontextmenü des Knotens des CAN-Erweiterungsmoduls angelegt (im Beispiel Kommunikationskanal 2).

CANbus Slave-Konfigurationseditor

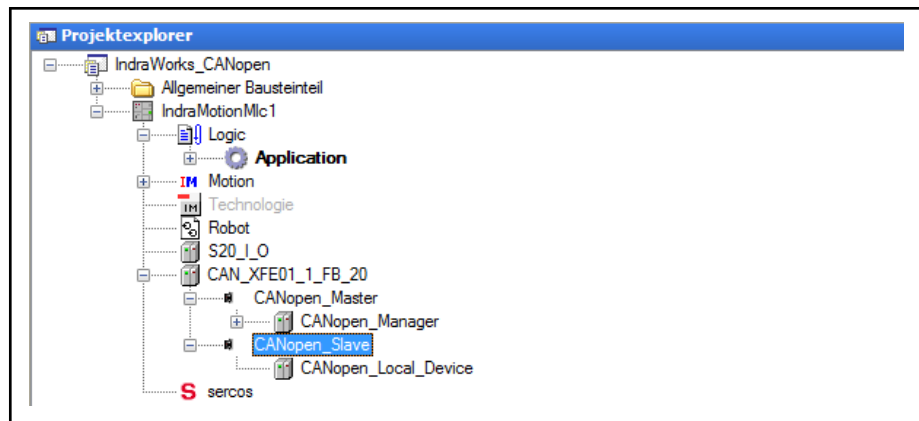


Abb. 5-1: Beispiel: CANbus-Konfigurationsbaum

Konfigurationsdialoge

Für das CANopen-Slave-Objekt

- CANopen Slave-Knoten, Seite 34, (Netzwerknummer, Baudrate)
- Registerkarte 'CANopen Slave - Allgemein', Seite 34, (Netzwerknummer, Baudrate)
- Registerkarte 'CANopen Slave - CANbus Parameter', Seite 35
- Registerkarte 'CANopen Slave - Status', Seite 35
- Registerkarte 'CANopen Slave - Information', Seite 35

Mit einem Doppelklick auf das "CAN_Local_Device"-Objekt im Projektexplorer wird der Konfigurationseditor geöffnet, der die folgenden Unterdialoge für das Einstellen der Slave-Parameter bereitstellt:

- CANopen Local Device, Seite 36
- Registerkarte 'CANopen Local Device - Allgemein', Seite 36
- Registerkarte 'CANopen Local Device - Parameter', Seite 38
- Registerkarte 'CANopen Local Device - E/A-Abbild', Seite 38
- Registerkarte 'CANopen Local Device - Status', Seite 39
- Registerkarte 'CANopen Local Device - Information', Seite 39

5.1 CANopen Slave-Knoten

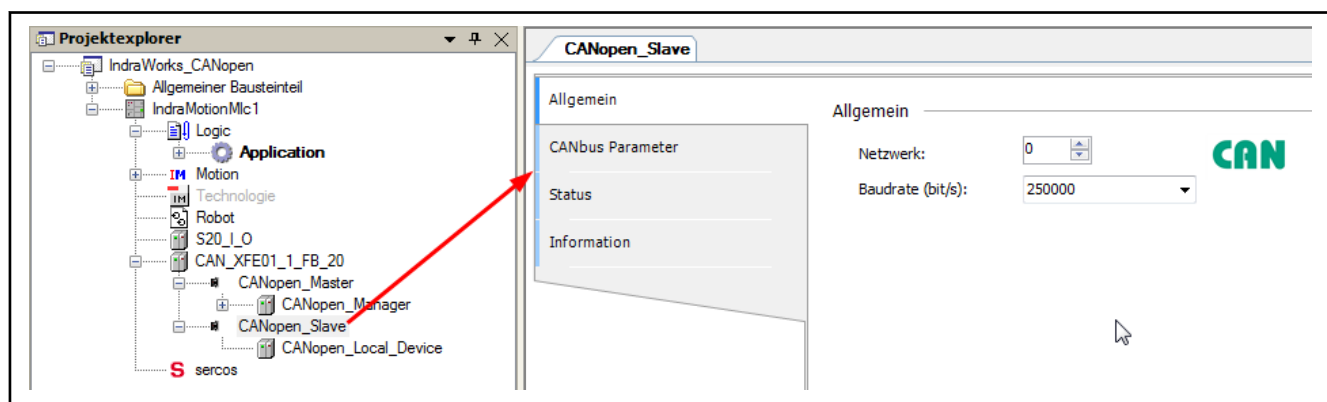


Abb. 5-2: Dialog "CANopen Slave"

5.1.1 Registerkarte 'CANopen Slave - Allgemein'

Das CANbus-Objekt muss als oberster Knoten einer CANbus Konfiguration in den Projektexplorer eingetragen werden ([CANbus Konfigurationseditor](#), Seite 15).

Im zugehörigen Konfigurationsdialog wird folgendes definiert:

- **Netzwerk:** Nummer des CAN-Netzwerks, das über die CANbus- Schnittstelle angebunden werden soll.

Erlaubte Werte: 0 bis 63.



Falls an der XM2x mehrere CAN-Erweiterungsmodule und mehrere CAN-Master-/Slave-Schnittstellen projiziert werden, muß für jede CAN-Schnittstelle eine eigene, eindeutige Netzwerk-Nummer eingestellt werden!

- **Baudrate (Bits/s):** Baudrate in Bits pro Sekunden für die Datenübertragung auf dem Bus.

Der Standardwert wird aus der Gerätebeschreibungsdatei (*.devdesc) des CANbus Geräts entnommen.

Die folgenden Baudraten können eingestellt werden:

10000, 20000, 50000, 100000, 125000, 250000, 500000, 800000 und 1000000.

Die Einheit ist bits/s (Bits pro Sekunde) wobei in den Konfigurationsdaten die Baudrate in KBit/s gespeichert wird.

5.1.2 Registerkarte 'CANopen Slave - CANbus Parameter'

Dieser Konfigurationsdialog für einen CAN-Bus-Knoten ist derzeit ohne Funktion, da die Gerätebeschreibung außer der Baudrate und Netzwerknummer (NetID) keine weiteren Konfigurationsparameter enthält.

5.1.3 Registerkarte 'CANopen Slave - Status'

Dieser Dialog einer CANopen-Geräte- oder -Modulkonfiguration zeigt Status-Informationen (z.B. "Running", "Stopped") und spezifische Diagnosemeldungen aus dem Gerät, auch zur verwendeten Karte und dem internen Bussystem ([CANbus Konfigurationseditor, Seite 15](#)).

5.1.4 Registerkarte 'CANopen Slave - Information'

Dieser Dialog einer CANopen-Geräte- oder -Modulkonfiguration zeigt Informationen zum Gerät bzw. Modul, die aus der Gerätebeschreibung gelesen werden.

Mögliche Angaben: Name, Hersteller, Typ, Versionsnummer, Kategorien, Bestellnummer, Abbildung ([CANbus Konfigurationseditor, Seite 15](#)).

5.2 CANopen Local Device

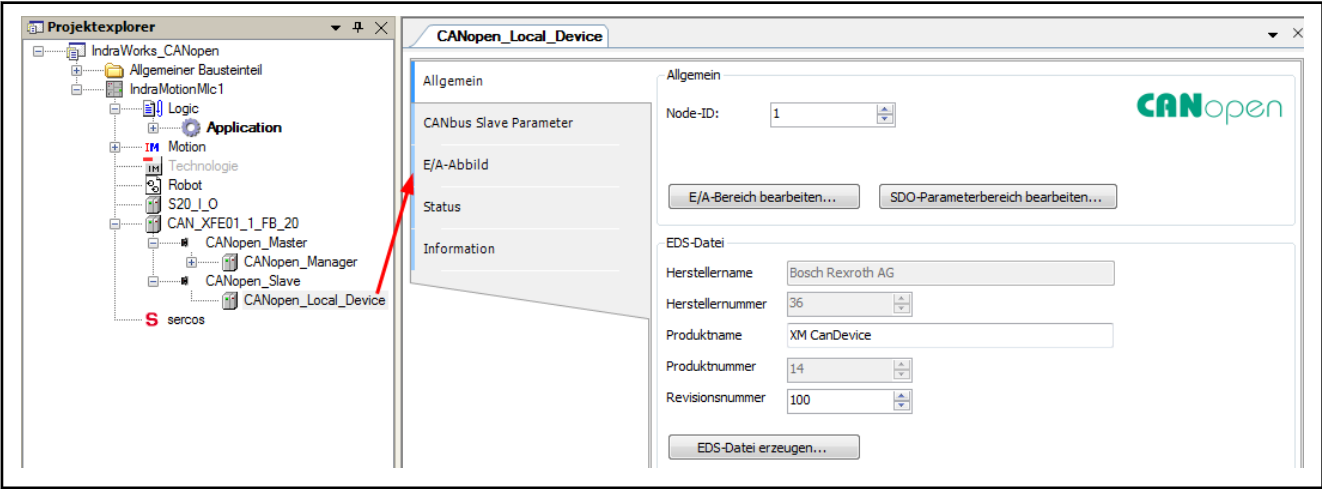


Abb. 5-3: Dialog "CANopen Local Device"

5.2.1 Registerkarte 'CANopen Local Device - Allgemein'

Allgemein

Node-ID	Die Knotennummer identifiziert das CAN-Modul eindeutig (Wert zwischen 1 und 127).
Geräteprofil	Das Geräteprofil, das der CANopen-Slave implementieren soll. Wenn eine Datenbank (*.CODB) für Objektverzeichnisse vorhanden ist, wird die Auswahl im Dialog "Objektverzeichnis" entsprechend der Geräteprofil-ID angepasst. Die Geräteprofil-IDs sind in den Spezifikationen der CIA definiert.
E/A-Bereich bearbeiten	Die Ein- und Ausgänge des CANopen Slaves werden durch E/A-Bereiche repräsentiert. Ein E/A-Bereich ist eine Sammlung von Eingängen und Ausgängen des gleichen Datentyps. Für jeden Eingang / Ausgang werden automatisch ein E/A-Kanal und eine Mapping-Variable angelegt. Die Mapping-Variablen ermöglichen, dass die E/As im Applikationsprogramm gelesen bzw. geschrieben werden können. Schaltfläche "E/A-Bereich bearbeiten..." öffnet den gleichnamigen Dialog zum Definieren der E/A-Bereiche und des entsprechenden Objektverzeichnis-Index des Slaves. Die mögliche Anzahl und Größe der E/A-Bereiche hängt von der Zahl der nicht verwendeten PDOs ab. Die aktuelle Zahl Verwendete TxPDOs und Verwendete RxPDOs wird immer unterhalb der Tabelle angezeigt. Ab Version 14.14.0.0 des CANopen Local_Device sind bis zu 512 PDOs (Tx wie Rx) möglich (ältere Versionen nur max. 16 PDOs).
SDO-Parameterbereich bearbeiten	Öffnet den Dialog "SDO-Parameterbereich bearbeiten". Details dazu finden Sie in der Beschreibung des Dialogs.

EDS-Datei

Herstellernummer	Allgemeine Angabe, die in der EDS-Datei gespeichert werden.
Produktname	
Produktnummer	
Revisionsnummer	
Herstellernummer	

CANbus Slave-Konfigurationseditor

EDS-Datei exportieren	Die EDS-Datei wird mit den angegebenen Einstellungen gemäß Standard CiA DS 306 V1.3 erzeugt und auf Ihrem lokalen Rechner gespeichert.
EDS-Datei importieren	Importieren der Daten aus einer bestehenden EDS-Datei.

Dialog "E/A-Bereich bearbeiten" Der Dialog zeigt die gerade gültigen E/A-Bereiche des Slaves.

Bereich hinzufügen	Öffnet den Dialog "E/A-Bereich hinzufügen". Wenn Sie bereits alle PDOs verwendet haben, ist der Dialog nicht editierbar.
Verwendete TxPDOs	Anzahl aktuell verwendeter TxPDOs / maximal möglicher TxPDOs
Verwendete RxPDOs	Anzahl aktuell verwendeter RxPDOs / maximal möglicher RxPDOs

E/A-Typ	Eingang oder Ausgang
Bereichsname	Ein eindeutiger Standardname wird automatisch eingetragen, kann aber verändert werden.
Objektindex	Index im Objektverzeichnis des Slaves, der dem neuen Objekt zugewiesen wird.
Anzahl	Anzahl der E/As. Die möglichen Werte hängen vom Datentyp und der aktuellen Anzahl der nicht verwendeten PDOs ab.
Datentyp	Auswahl an möglichen Datentypen.
Neues PDO forcen	☒ : (Standard). Jeder neue E/A-Bereich wird auf ein neues PDO abgebildet, auch wenn das vorliegende PDO Mapping noch nicht voll ist. ☐ : Das aktuelle PDO-Mapping wird komplett aufgefüllt (wenn möglich), bevor ein neues PDO angelegt wird.
OK	Aktiviert den neuen E/A-Bereich. Eingangs-/Ausgangs-Kanäle und Mapping-Variablen werden angelegt. Im Objektverzeichnis des Geräts erhält jeder E/A-Bereich einen eigenen Index (Ausgangsbereich: 0x3000 – 0x30FF, Eingangsbereich: 0x3800 – 0x38FF) und jeder einzelne Eingang / Ausgang innerhalb des E/A-Bereichs einen Subindex. Beim Anlegen eines E/A-Bereichs werden automatisch PDOs erzeugt (RxPDOs beginnen mit 0x1400, TxPDOs mit 0x1800). Wenn die Option "Neues PDO forcen" aktiviert ist (s.o.), werden die E/As automatisch nacheinander auf diese neuen PDOs abgebildet.

Dialog 'SDO-Parameterbereichs bearbeiten'

Sie können Lese/Schreib-Parameter (SDO-Parameter) definieren, die von SDOs gelesen bzw. geschrieben werden können. Dazu müssen Sie Parameterbereiche definieren, die jeweils eine Sammlung von Parametern des gleichen Datentyps sind. Für jeden SDO-Parameter werden automatisch ein E/A-Kanal und eine Mapping-Variable angelegt. Die Mapping-Variablen erlauben das Lesen oder Schreiben von E/As mit Hilfe des Applikationsprogramms.

Im Objektverzeichnis des Geräts ist jeder SDO-Parameterbereich mit einem eigenen Index vertreten und jeder einzelne SDO-Parameter innerhalb des Parameterbereichs mit einem Subindex:

- SDO Schreib-Parameterbereich: 0x5000 - 0x50FF
- SDO Lese-Parameterbereich: 0x5800 – 0x58FF

Der Dialog zeigt die aktuell gültigen Parameterbereiche und ihre Objektverzeichnis-Indizes.

CANbus Slave-Konfigurationseditor

Bereich hinzufügen	Öffnet den Dialog Parameterbereich hinzufügen.
--------------------	--

Dialog "Parameterbereich hinzufügen"

Der Dialog ermöglicht Ihnen auf einfache Weise E/As hinzuzufügen.	
SDO-Zugriffstyp	Nur Lesen oder Lesen/Schreiben .
Bereichsname	Ein eindeutiger Name wird automatisch eingetragen, kann aber abgeändert werden.
Objektindex	Index im Objektverzeichnis des Slaves, der dem neuen Objekt zugewiesen wird.
Anzahl	Anzahl der Parameter
Datentyp	Auswahl an möglichen Datentypen

5.2.2 Registerkarte 'CANopen Local Device - Parameter'

Diese Registerkarte zeigt die Parameter des CANopen Local_Device in tabellarischer Form.

Ein Teil der Parameter ist editierbar (Wertfeld zum Editieren anklicken). Die Schriftfarbe dieser Parameter ist schwarz.

Ist die Schriftfarbe der Felder Wert und Typ grau, werden lediglich die aktuellen Werte angezeigt.

5.2.3 Registerkarte 'CANopen Local Device - E/A-Abbild'

Die Registerkarte zeigt den Namen der automatisch erzeugten Instanz des Local_Device unter "IEC-Objekte" in der Spalte "Variable" an.

Sie können über die Applikation auf diese Instanz zugreifen. Somit kann beispielsweise ein Neustart, ein Stop oder eine Statusprüfung des Zielgeräts über eine Applikation erfolgen.

Bitte sehen Sie auch hierzu die Beschreibung des allgemeinen E/A-Abbild-Dialogs.

Buszyklusoptionen

Buszyklus-Task	Die Auswahlliste bietet alle Tasks an, die in der Task-Konfiguration der aktiven Applikation definiert sind, wie zum Beispiel 'MainTask'. Bei Zykluseinstellungen des übergeordneten Busses verwenden verwendet IndraLogic die Einstellungen des Vaterknotens. Wenn Sie hier keine Task definieren, gilt die in den SPS-Einstellungen eingestellte Task. Wenn Sie dort ebenfalls keine Task eingestellt haben, dann wird die höchst-prioritäre Task als Buszyklus-Task verwendet.
----------------	--

Registerkarte '<Gerät> E/A-Abbild'

In dieser Registerkarte des generischen Geräteeditors nehmen Sie die Abbildung der Eingangs-, Ausgangs- und Speicheradressen des Steuerungsgeräts auf Variablen der Applikation vor. Damit erstellen Sie das sogenannte "E/A-Abbild" ("Mapping").

IEC Objekte

	Dieser Teil des Dialogs erscheint nur, wenn implizit eine Instanz des Geräteobjekts angelegt wird, auf die in der Applikation zugegriffen werden kann. Dies kann beispielsweise verwendet werden, um einen Bus neu zu starten oder Informationen abzufragen. Ob eine solche Instanz zur Verfügung steht und wie sie verwendet werden kann, hängt vom Gerätetyp ab.
--	---

5.2.4 Registerkarte 'CANopen Local Device - Status'

Dieser Unterdialog des Konfigurationseditors eines lokalen CANbus Slave-Geräts zeigt Statusinformationen (beispielsweise "Running", "Stopped") und gerätespezifische Diagnosemeldungen, sowie Informationen über die verwendete Karte und das interne Bussystem.

5.2.5 Registerkarte 'CANopen Local Device - Information'

Dieser Dialog in der lokalen CANbus Slave-Gerätekonfiguration zeigt Informationen zum gerade im Projektextplorer selektierten Gerät.

Wenn für das Gerät verfügbar, werden die folgenden allgemeinen Informationen ausgegeben: Name, Hersteller, Typ, Versionsnummer, Kategorien, Bestellnummer, Beschreibung, Bild.

6 Gerätedatenbank

6.1 Gerätedatenbank, Übersicht

Die Gerätedatenbank ist eine Ergänzung zur Gerätebibliothek (IndraWorks, rechte Seite).

Sie enthält die Gerätebeschreibungen, die auf dem lokalen System installiert sind, um sie für Projekte in IndraWorks verfügbar zu machen. In den Gerätebeschreibungsdateien werden Eigenschaften (z.B. Parameter, Hersteller, Gerätenamen usw.) des jeweiligen Gerätes hinterlegt.

Für unterschiedliche Geräte existieren verschiedene Formate

- PROFIBUS DP: *.gs?,
- PROFINET IO: GSDML*.xml,
- EtherNet/IP: EDS*.xml,
- CAN: *.eds
- Sercos III IO: SDDML-Datei, *.xml.
- Gerätebeschreibungsdateien: *devdesc.xml

Die Gerätedatenbank wird bei der IndraWorks-Installation mit installiert.

Das Installieren zusätzlicher Geräte und das Deinstallieren nicht benötigter Geräte kann in der Gerätedatenbank selbst vorgenommen werden.

Für den CAN-Bus ist eine Nachinstallation sowohl der *devdesc.xml als auch der *.eds-Dateien erforderlich.

6.2 Geräte verwalten

6.2.1 Gerätedatenbank, Dialog

Symbol: 

Mit dem Befehl "Gerätedatenbank" können Sie Gerätebeschreibungsdateien installieren und deinstallieren. Klicken Sie im Hauptmenü auf **Extras ► Gerätedatenbank...**, um den Dialog "Gerätedatenbank" zu öffnen:

Gerätedatenbank

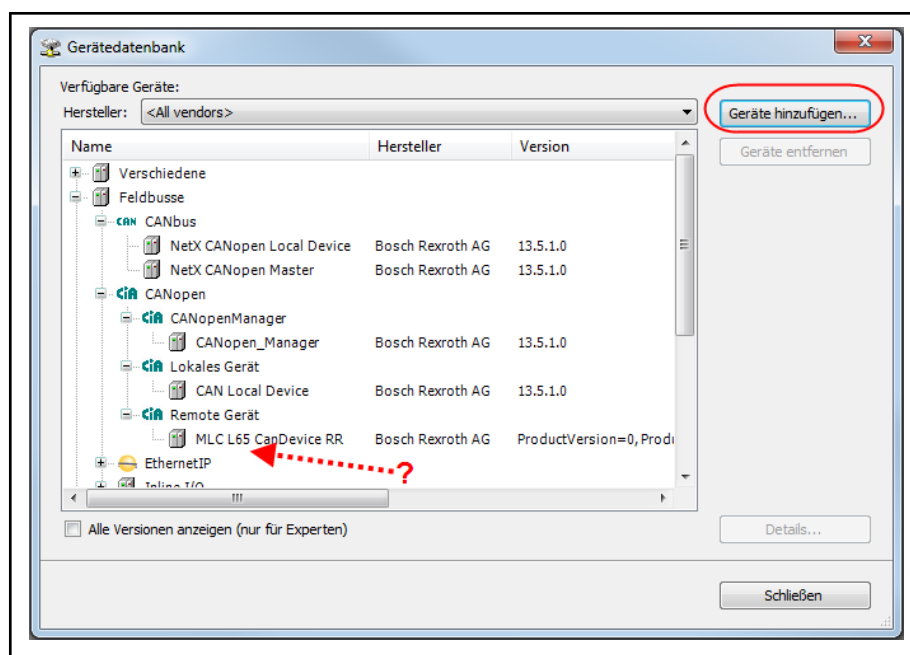


Abb. 6-1: Gerätedatenbank

Verfügbare Geräte

Im Bereich "Verfügbare Geräte" sind die Geräte aufgelistet, die aktuell installiert sind.

Die Geräte sind hierarchisch geordnet, d.h. Steuerungen erscheinen in der obersten Ebene, darunter Funktionsmodule, integrierte Schnittstellen usw.

Die Liste kann durch Wahl des Filters "Hersteller" eingeschränkt werden.

Weiterhin können Sie im Bereich "Verfügbare Geräte" Geräte hinzufügen und Geräte entfernen.

Normalerweise werden die zu Ihrer IndraWorks-Version passenden Geräte gezeigt. Sie können jedoch auch den kompletten Überblick durch "Alle Versionen anzeigen" erhalten.

Haben Sie ein konkretes Gerät ausgewählt, stehen unter "Details" weitere Informationen zur Verfügung.

6.2.2 Geräte hinzufügen

Um Geräte in IndraWorks hinzuzufügen, führen Sie bitte folgende Schritte aus:

1. Klicken Sie im Hauptmenü auf **Extras ► Gerätedatenbank...**, um den Dialog "Gerätedatenbank" zu öffnen.
2. Klicken Sie im Dialog "Gerätedatenbank" auf die Schaltfläche "Geräte hinzufügen".

Es öffnet sich der Dateiauswahl-Dialog "Gerätebeschreibung installieren":

Gerätedatenbank

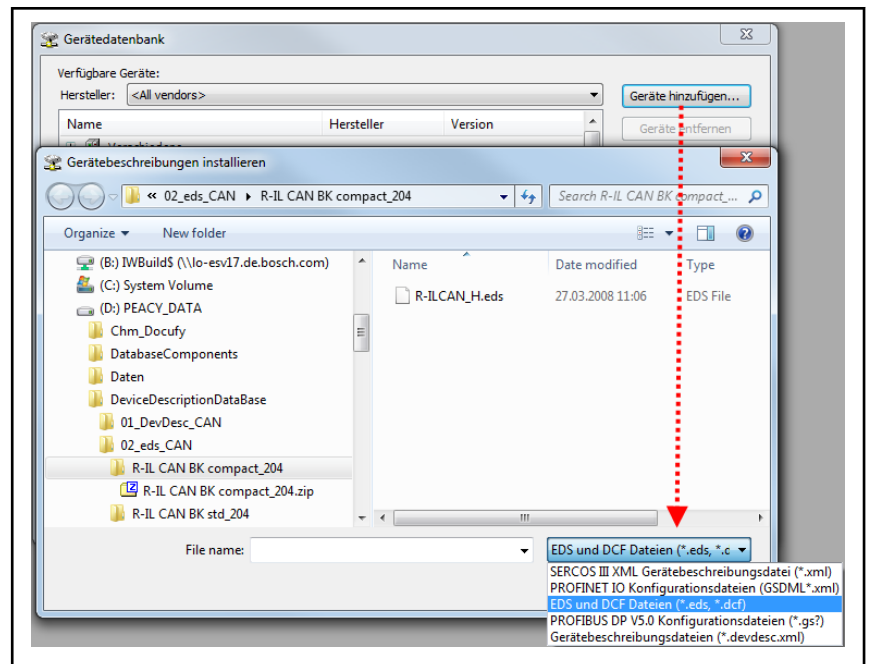


Abb. 6-2: Gerätebeschreibungen installieren

3. Wählen Sie die gewünschten Gerätebeschreibungsdateien, die Sie installieren möchten.



Achten Sie auf den richtigen Dateityp im Auswahlfeld "Dateityp".

4. Starten Sie den Installationsvorgang, indem Sie auf "Öffnen" klicken.
 Nach dem Installationsvorgang erscheinen die importierten Geräte erscheinen unter "Verfügbare Geräte" und sind dort markiert:

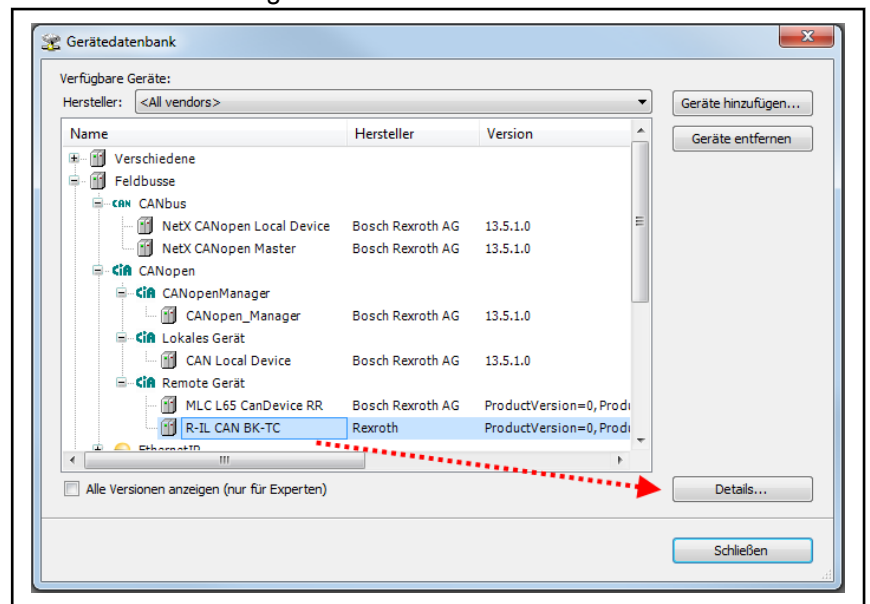


Abb. 6-3: Gerätebeschreibungen installiert

5. Wählen Sie das gewünschte Gerät, und Sie erhalten unter "Details" weitere Informationen.

Gerätedatenbank

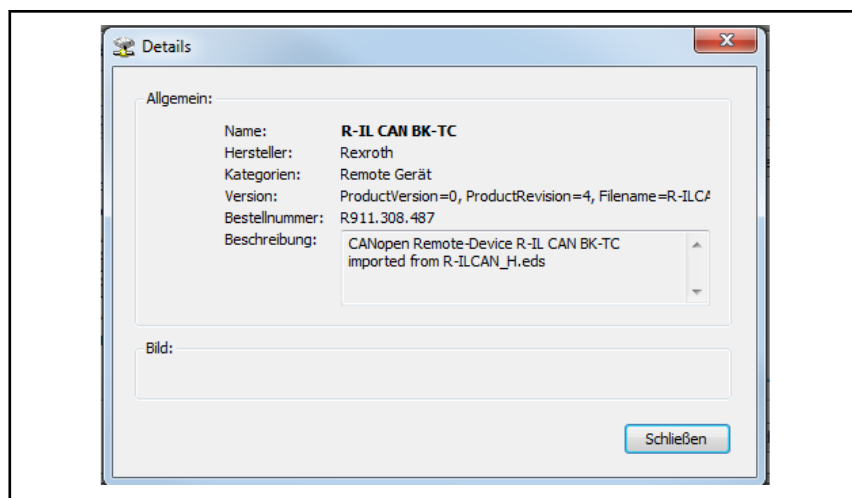


Abb. 6-4: Details zur installierten Gerätebeschreibungsdatei

6.2.3 Geräte entfernen

Um Geräte aus IndraWorks zu entfernen, führen Sie bitte folgende Schritte aus:

1. Klicken Sie im Hauptmenü auf **Extras ► Gerätedatenbank...**, um den Dialog "Gerätedatenbank" zu öffnen.
2. Markieren Sie im Gerätebaum die Geräte, die Sie entfernen möchten.
3. Klicken Sie auf die Schaltfläche "Geräte entfernen".
Es öffnet sich der Dialog "Entfernen von Geräten bestätigen":
4. Klicken Sie auf "Ja", um das markierte Gerät zu entfernen.
Das Gerät wird aus der Gerätedatenbank und aus der Bibliothek entfernt.
5. Beenden Sie den Dialog "Gerätedatenbank" mit "Schließen".

7 CAA_CiA405.library

7.1 CAA_CiA405.Library, Allgemeines

Dieses Dokument verwendet die standardmäßige CANopen-Schnittstelle für IEC 61131-3-programmierbare Geräte wie SPS.

Die Geräte verwenden Kommunikationstechniken, die denjenigen entsprechen, die im CiA Normentwurf DS-301 (Anwendungsebene und Kommunikationsprofil), Normentwurf DSP-302 (Framework für programmierbare CANopen Geräte) und Normentwurf DS-405 (Schnittstelle und Geräteprofil für IEC 61131-3 programmierbare Geräte) beschrieben sind.

Diese Dokumente sollten gleichzeitig mit dieser Beschreibung herangezogen werden.

- Referenzen**
- /DS-301/ CiA Draft Standard DS 301 V4.02 CANopen Application Layer and Communication Profile
 - /DSP-302/ CiA Normentwurf DSP 302 V3.1 CANopen Framework für programmierbare CANopen Geräte
 - /DS-405/ CiA Draft Standard Proposal DS 405 V2.0 CANopen interface and Device Profile for IEC 61131-3 Programmable Devices

Im Allgemeinen implementiert die Erstellung einer Anwendung den Umgang mit bis zu fünf Schnittstellen:

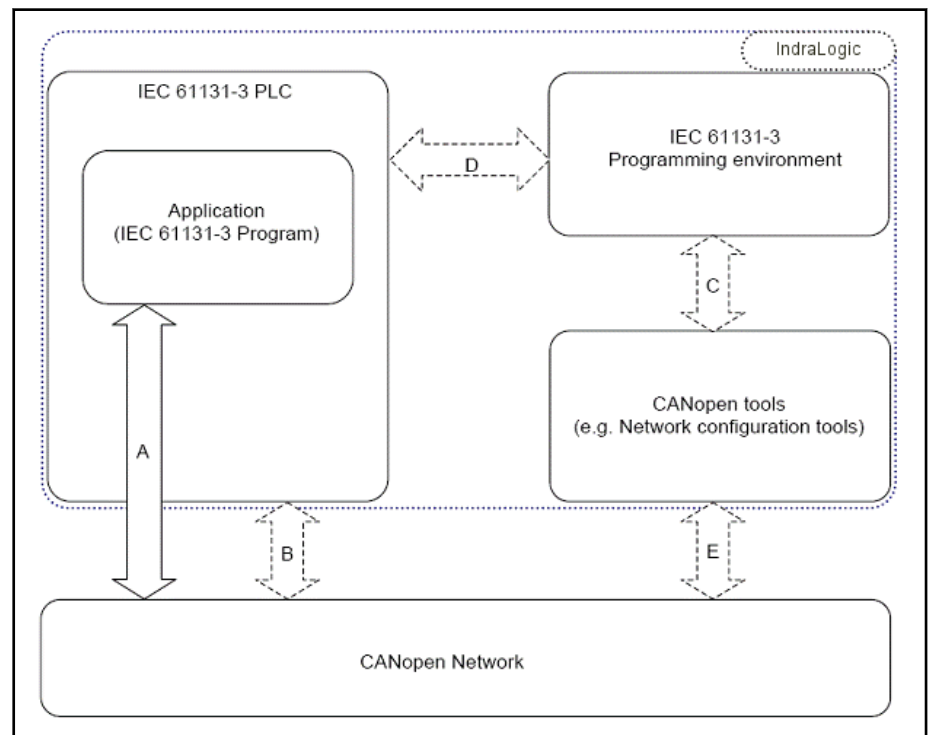


Abb. 7-1: Erzeugung einer Applikation / eines Handling Interface

Diese Beschreibung behandelt nur die Schnittstelle:

A. bedeutet, der Zugriff auf ein CANopen Kommunikationssystem aus einem IEC 61131-3 Programm heraus basierend auf Aufrufen von Funktionsbausteinen

Diese Beschreibung deckt die Schnittstellen (noch) nicht ab:

B. Dienstfunktionen für Fehlersuche, Überwachung und Netzwerkverwaltung

CAA_CiA405.library

C. Schnittstelle zwischen CANopen Werkzeugen und IEC 61131-3 Programmierungsumgebung

D. Herstellerspezifische Schnittstelle zwischen Programmierungsumgebung und SPS-Laufzeit

E. für CANopen Werkzeug spezifische Schnittstelle zwischen Konfigurationswerkzeug und Netzwerk



Die Abbildung beschreibt nicht notwendigerweise die Verwendung von verschiedenen Werkzeugen für Programmierung und Konfiguration.

Ein Werkzeug kann möglicherweise beide Funktionalitäten handhaben und die Schnittstellen überdecken. Ein Beispiel hierfür ist das Programmierwerkzeug IndraLogic (siehe gestrichelte Linie), wo die CANopen Konfiguration und die Laufzeitsoftware in der SPS enthalten ist.

Anwendungsschnittstelle

Jede Anwendung innerhalb einer IEC 61131-3 Ebene muss in der Lage sein, in einem Geräteobjektverzeichnis eines mit CANopen verbundenen Knotens zu lesen oder auf dieses zu schreiben (SDO). Weitere Elemente einer CANopen Kommunikation sind Process Data Objekte (PDO) die im Prozessabbild innerhalb einer IEC Anwendung als Eingänge (PDO empfangen; d.h.: %IX..) oder Ausgänge (PDO übertragen; d.h.: %QW..) unterstützt werden.

Empfangene asynchrone Fehlermeldungen (EMCY) werden in einen Puffer für jeden Knoten registriert. Der Netzwerkstatus wird vor der Netzwerkverwaltung (NMT) kontrolliert. Dank CANopen ist die Node-Guarding Funktionalität, wie auch die synchronisierte Kommunikation für Prozessdaten möglich.

Die vorliegende Beschreibung erläutern die Anforderungen an eine Funktionsbausteinbibliothek für die CANopen-Kommunikation.

7.2 Datentypen

7.2.1 Datentypen der CAA_CiA405.library

Nachstehende Datentypen sind aufgrund von Anforderungen der Bibliothek obligatorisch.

ENUM

- [DEVICE_STATE](#), Seite 46
- [SDO_MODE](#), Seite 47
- [TRANSITION_STATE](#), Seite 48
- [CANOPEN_KERNEL_ERROR](#), Seite 48

STRUCT

- [EMCY_ERROR](#), Seite 48

Typ-Ableitung

- [DEVICE](#), Seite 49
- [SDO_ERROR](#), Seite 49

7.2.2 DEVICE_STATE

Kurzbeschreibung Der Aufzählungstyp `DEVICE_STATE` (INT) beschreibt den Zustand der CANopen Netzwerkschicht, wie in /DS-301/ definiert.

Die Zustände INIT, RESET_COMM, RESET_APP, PRE_OPERATIONAL, STOPPED, OPERATIONAL entsprechen den gleichen Zuständen wie in / DS-301/.

Der Zustand UNKNOWN muss dann verwendet werden, wenn der aktuelle Zustand des Geräts unbekannt ist (z.B. wenn kein Guarding des Geräts durchgeführt wird).

Der Zustand NOT_AVAIL muss dann verwendet werden, wenn bekannt ist, dass das Gerät nicht verfügbar ist (z.B. wenn Guarding durchgeführt wird und das Gerät nicht antwortet).

INIT	0
RESET_COMM	1
RESET_APP	2
PRE_OPERATIONAL	3
STOPPED	4
OPERATIONAL	5
UNKNOWN	6
NOT_AVAIL	7

Tab. 7-1: Elemente des Aufzählungstyps DEVICE_STATE

Der von [GET_STATE](#), Seite 68, gelieferte Zustand ist:

- Wenn Node-Guarding / Heartbeat für das ausgewählte DEVICE nicht aktiviert ist: UNKNOWN (6) (selbst dann wenn, eine Neustartmeldung empfangen wurde)
- Wenn Heartbeat aktiviert ist, aber der NMT-Master nicht der Heartbeat-Consumer ist: UNKNOWN (6).
- Wenn Node-Guarding / Heartbeat aktiviert ist und das Gerät seinen NMT-Status meldet:

berichtigter NMT-Zustand (Wert, in DS301 V4.02 Kap. "Fehlersicherungsprotokoll" definiert)	STATE (Wert, in DS405 V2.0 Kap. "Datentypen" definiert)
Initialisierung (0)	INIT (0)
Pre-Operational (127)	PRE-OPERATIONAL (3)
Operational (5)	OPERATIONAL (5)
Stopped (4)	STOPPED (4)

- Wenn Node-Guarding / Heartbeat aktiv ist und das Gerät seinen NMT Zustand nicht vor dem Timeout korrekt berichtet, z.B. "Node-Life-Time" (Node-Guarding-Protokoll) bzw. "Heartbeat Consumer-Zeit" (Heartbeat-Protokoll): NOT_AVAIL (7)

7.2.3 SDO_MODE

Kurzbeschreibung

Der Parametermodus für [SDO_WRITE](#), Seite 64, ist ein Wert aus folgendem Satz:

AUTO	0	Client wählt den Modus automatisch
EXPEDITED	1	Client verwendet das beschleunigte Protokoll

CAA_CiA405.library

SEGMENTED	2	Client verwendet das segmentierte Übertragungsprotokoll
BLOCK	3	Client verwendet das Blocktransferprotokoll

Tab. 7-2: Elemente des Aufzählungstyps SDO_MODE

Im Modus "AUTO" benutzt der Client das effizienteste Übertragungsprotokoll, um die Daten vom oder auf den Server zu übertragen. Er fragt die Möglichkeiten ab und wählt den besten Weg für beide Parteien.

Das SDO-Protokoll ist in /DS 301/ definiert.

7.2.4 TRANSITION_STATE

Kurzbeschreibung Der Aufzählungstyp (UINT) beschreibt die Zustandsübergänge der CANopen-Netzwerkschicht, wie in /DS-301/ definiert.

STOP_REMOTE_NODE	16#04	
START_REMOTE_NODE	16#05	
RESET_NODE	16#06	
RESET_COMMUNICATION	16#07	
ENTER_PRE_OPERATIONAL	16#7F	
ALL_EXCEPT_NMT_AND_SENDER	16#800	Ein Flag zur Verwendung in Kombination mit anderen Zustandsübergängen (logisches ODER). Es ist zu verwenden, wenn NMT-Kommandos an alle Geräte adressiert sind (DEVICE=0), mit Ausnahme des NMT-Senders und des NMT masters (= CANopenManager).

Tab. 7-3: Elemente des Aufzählungstyps TRANSITION_STATE

7.2.5 CANOPEN_KERNEL_ERROR

Kurzbeschreibung Typ CANOPEN_KERNEL_ERROR beinhaltet Fehlerinformationen bezüglich CANopen Kernel.

Die möglichen Belegungen sind unter "Globale Konstanten" [CANOPEN_KERNEL_ERROR_CODES, Seite 49](#), zusammengestellt.

Name	Typ	Beschreibung
CANOPEN_KERNEL_ERROR	WORD	

Tab. 7-4: Typzuweisung SDO_ERROR

7.2.6 EMCY_ERROR

Kurzbeschreibung Die Struktur EMCY_ERROR beinhaltet Emergency-Fehlerinformationen, wie in /DS-301/ spezifiziert:

Name	Typ	Beschreibung
EMCY_ERROR_CODE	WORD	
ERROR_REGISTER	BYTE	
ERROR_FIELD	ARRAY [1..5] OF BYTE	

Tab. 7-5: Elemente der Struktur EMCY_ERROR

7.2.7 DEVICE

Kurzbeschreibung Typ DEVICE stellt die Node_ID eines Gerätes dar.

Name	Typ	Beschreibung
DEVICE	USINT (0..127)	

Tab. 7-6: Typzuweisung DEVICE

7.2.8 SDO_ERROR

Kurzbeschreibung Typ SDO_ERROR stellt Fehlerinformationen dar, wie in / DS-301/ definiert:

Name	Typ	Beschreibung
SDO_ERROR	UDINT	

Tab. 7-7: Typzuweisung SDO_ERROR

7.3 Globale Konstanten

7.3.1 CANOPEN_KERNEL_ERROR_CODES

Die nachstehende Tabelle enthält mögliche Belegungen des Datentyps [CANOPEN_KERNEL_ERROR](#), Seite 48.

	Wert	Beschreibung
CANOPEN_KERNEL_NO_ERROR	16#0	Kein Fehler Es wurde kein Fehler von CANopen Kernel gemeldet. Der Ausgang DEVICE ist ungültig und kann einen beliebigen Wert enthalten. Nichtsdestotrotz besteht die Möglichkeit, dass eine SDO-Abbruch- oder EMCY-Meldung von irgendeinem Gerät empfangen wurde; siehe SDO_READ , Seite 66, SDO_WRITE , Seite 64, und RECV_EMCI , Seite 75 / RECV_EMCI_DEV , Seite 73
CANOPEN_KERNEL_OTHER_ERROR	16#1	Anderer Fehler Ist ein "further error register"-Ausgang beim aufgerufenen FB vorhanden, enthält dieser Ausgang ausführlichere Informationen, z.B. über die FBs SDO_READ , Seite 66, und SDO_WRITE , Seite 64, an die die Inhalte der SDO-Abbruch-Meldung geliefert wurde. Bei den FBs RECV_EMCI_DEV , Seite 73, und RECV_EMCI , Seite 75, wird der Inhalt der empfangenen EMCYMeldung geliefert. Beim FB GET_CANOPEN_KERNEL_STATE , Seite 70, wird dieser Wert niemals geliefert, weil kein "further error register"-Ausgang verfügbar ist.
CANOPEN_KERNEL_DATA_OVERFLOW	16#2	Datenüberlauf Im Falle von implementierten Sende- oder Empfangspuffer für CAN Objekte besteht die Möglichkeit, dass diese Puffer überlaufen.

CAA_CiA405.library

	Wert	Beschreibung
CANOPEN_KERNEL_TIMEOUT	16#3	Timeout Eine Zeitüberschreitung tritt dann ein, wenn z.B. ein Knoten nicht auf eine SDO-Anfrage antwortet. Die Zeitmessung startet, wenn die Anwendung die Anfrage stellt, d.h. bei steigender Flanke des ENABLE Signals. Siehe auch Zusätzliche Parameter, Seite 51
	16#4.. 16#F	Für weitere SDO-Fehler vorbehalten CIA vorbehalten
CANOPEN_KERNEL_CANBUS_OFF	16#10	CAN-Bus ist in BUSOFF oder im Busalarmzustand Der Knoten ist vom Bus getrennt.
CANOPEN_KERNEL_CAN_ERROR_PASSIVE	16#11	CAN-Fehler passiv Der Knoten ist fehlerpassiv. Der Knoten kann kommunizieren, aber es ist ihm nicht möglich, einen aktiven Fehler-Flag im Falle einer Fehlererkennung zu senden, d.h. es ist ihm nicht möglich, die Buskommunikation zu stören. Darüber hinaus wartet er 8 Bit länger, als ein anderer Knoten, bevor er eine weitere Übertragung startet.
	16#12.. 16#1F	Weiteren internen Fehlern vorbehalten CIA vorbehalten
CANOPEN_INTERNAL_FB_ERROR	16#21	Hersteller spezifisch Bis zu 222 herstellerspezifische Fehler (16#21 – 16#FF). Dies ist der freie Wertebereich, der für Fehlercodes, die von Software der untergeordneten Ebene, z.B. Kernel, Treiber, Dienste, Betriebssystem, ... stammen, verwendet werden soll.
	16#100.. 16#FFFF	CIA vorbehalten

7.4 Funktionsbausteine

7.4.1 Funktionsbausteine der CAA_CiA405.library

Nachfolgend beschrieben werden die standardmäßigen Funktionsbausteine für den Zugriff auf CANopen von IEC 61131-3.

Die Funktionsbausteine sind aufgrund von Anforderungen an die Bibliothek obligatorisch.

Optionale Funktionsbausteine im Sinne der CANopen-Standardisierung werden zusätzlich beschrieben, um alle in /DS-405/ definierten Funktionsbausteine auf eine Stufe zu stellen.

Die folgende Tabelle kann von IEC 61131-3-konformen Systemen dazu verwendet werden, die verdeckten Eigenschaften festzustellen.

Basis-Funktionsbaustein (Verhaltensmodell, Basisklasse)

- [CIA405BASE, Seite 55](#)

Obligatorische Funktionsbausteine

- [RECV_EM CY_DEV, Seite 73](#)
- [RECV_EM CY, Seite 75](#)
- [SDO_WRITE4, Seite 56](#)
- [SDO_WRITE_DATA, Seite 59](#)
- [SDO_READ4, Seite 60](#)
- [SDO_READ_DATA, Seite 63](#)

- [GET_LOCAL_NODE_ID](#), Seite 67
- [GET_STATE](#), Seite 68
- [GET_CANOPEN_KERNEL_STATE](#), Seite 70
- [NMT](#), Seite 71

Optionale Funktionsbausteine

- [SDO_WRITE](#), Seite 64
- [SDO_READ](#), Seite 66

7.4.2 Allgemeine Aspekte zum Aufbau von Funktionsbausteinen

Datentypen im Vergleich zur Datenlänge

Innerhalb von IEC 61131-3 ist es möglich, die Übertragung von Daten beliebiger Länge darzustellen.

Allerdings sind die Schnittstellen zu Funktionsbausteinen, die das erlauben, möglicherweise deutlich schwerer zu verstehen und zu benutzen.

Aus diesem Grund werden hier verschiedene Funktionsbausteine für die Übertragung von jeweils festen (maximalen) Datenmengen, und auch Bausteine für beliebige Längen definiert.

Timeout

Die Schnittstelle ist so konzipiert, dass es möglich ist, die Aufrufe mit einem Zeitgeberblock zu verpacken, um Zeitüberschreitungen zu implementieren.

Zusätzlich ist es erlaubt, dass CANopen-Software der untergeordneten Ebene ihre eigenen Zeitüberschreitungen und Meldungen anwenden, wie z.B. Fehler an den Aufrufer.

Zusätzliche Parameter

Möglicherweise unterstützen oder fordern einige Systeme zusätzliche Informationen in den Funktionsbausteinen. So kann z.B. eine SPS mehrere CAN-Kanäle aufweisen.

In diesem Fall benötigen die Funktionsbausteine die Kanalnummer des Dienstes.

Eine weitere Anforderung besteht darin, die Zeit zu begrenzen, die ein Funktionsbaustein benötigt, bis er dem Aufrufer darüber Meldung erstattet, ob der Funktionsbaustein erfolgreich oder nicht erfolgreich ausgeführt wurde. Aus diesem Grunde ist eine Timeout-Eingang (Millisekunden) erforderlich.

Aus diesem Grunde werden diese Parameter den Funktionsbausteinen hinzugefügt.

In /DS405/ ist es erlaubt und innerhalb von CAA_CiA405.Library ist es obligatorisch, den Funktionsbausteinen die folgenden Parameter hinzuzufügen:

NETWORK: USINT;
(*CAN Kanalnummer*)

Bestimmt den CAN-Kanal, auf dem der erforderliche Dienst ausgeführt werden muss (ab Nummer 1). Ein Gerät mit nur einem einzigen CAN-Kanal muss entweder nur die Nummer 1 oder alle Nummern akzeptieren, d.h. muss den Wert dieses Eingangs ignorieren.

CHANNEL: USINT;
(*SDO Kanalnummer*)

Bestimmt den SDO-Kanal, auf dem der erforderliche Dienst ausgeführt werden muss (ab Nummer 1).

CAA_CiA405.library

Ein Gerät mit nur einem einzigen SDO-Kanal muss entweder nur die Nummer 1 oder alle Nummern akzeptieren, d.h. muss den Wert dieses Eingangs ignorieren.

Bei der Verwendung der Nummer 0, wird die "Auto Kanal" Funktion aktiviert, mit dieser wird der erforderliche Dienst automatisch mit dem nächsten freien SDO-Kanal ausgeführt.

Die CAN-Kanalnummer ist nicht die gleiche wie die im CAN-Bus-Konfigurator spezifizierte Net_ID.

Bei der Berechnung der CAN-Kanalnummer muss folgende Formel verwendet werden:

Net_ID + 1

TIMEOUT: UDINT;
(*Millisekunden; maximale Warte-
zeit auf Antwort*)

Die Zeitüberschreitung legt die maximale Wartezeit auf eine Antwort des Funktionsbausteins fest.

Die Zeitmessung beginnt mit der steigenden Flanke an Eingang ENABLE des aufgerufenen Funktionsbausteins.

Wenn die festgelegte maximale Zeit abgelaufen ist, ohne dass eine Antwort für den fragenden Dienst verfügbar ist, dann beendet der Funktionsbaustein die Ausführung und meldet eine Zeitüberschreitung.

Im Falle einer unbegrenzten Antwortzeit wird der Wert 0 in diesem Eingang verwendet.

DEVICE: DEVICE; (*Quelle von
STATE *)

In der CAA_CiA405.Library ist ein weiterer Ausgang des Funktionsbausteins [GET_CANOPEN_KERNEL_STATE](#), Seite 70, definiert:

DEVICE enthält entweder die Gerätenummer des Geräts, mit dem STATE verbunden ist (z.B. im Falle des Empfangs eines unbekannten Datenformats) oder die Nummer des Geräts, wo der Kernel positioniert ist.

MODE: SDO_MODE;
(*Protokolltyp*)

In der CAA_CiA405.Library ist ein weiterer Eingang des Funktionsbausteins [SDO_WRITE](#), Seite 64, definiert:

MODE bestimmt die Methode für Senden/Empfangen von Daten vom Server hinsichtlich SDO-Protokollspezifikation in /CIA301/

Steuermechanismus

Im Allgemeinen besteht die Aufgabe von jedem definierten Funktionsbaustein der CAA_CiA405.Library darin, einen oder mehrere Parameter zu lesen oder zu schreiben.

Solche Parameter können Variablen oder Zustände von an CANopen angeschlossenen Geräten (SDO, EMCY, NMT-Zustand) oder lokalen Knoten (eigene Knotennummer, CANopen Kernel Zustand) sein. Aus diesem Grunde legt jeder Funktionsbaustein im Allgemeinen das gleiche Verhaltensmuster an den Tag.

Ein typisches Verhalten der Signale ENABLE, CONFIRM und ERROR wird beim SDO-Zugriff gezeigt:

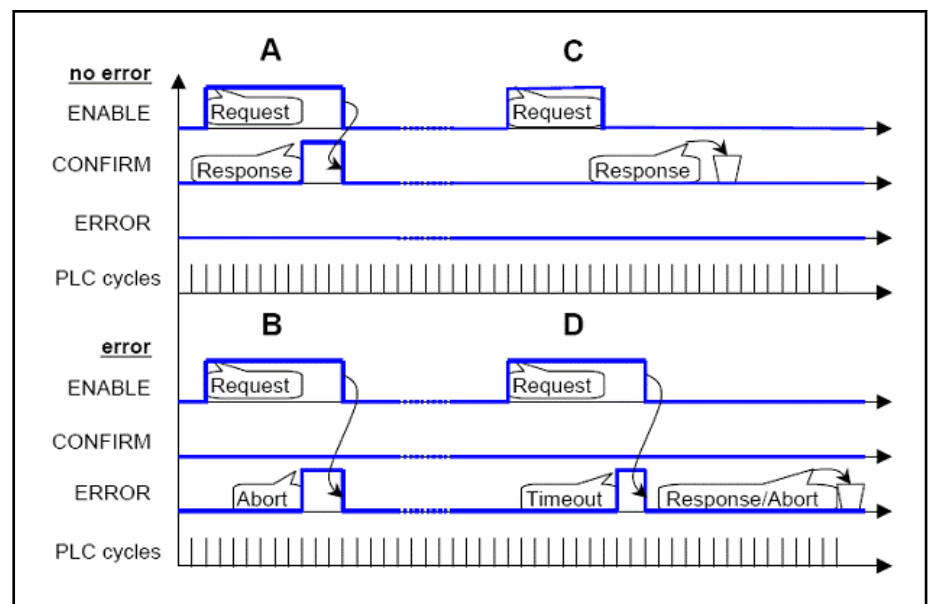


Abb. 7-2: Request/Response Diagramm

- Die Spitze bei Symbol  zeigt den Zeitpunkt, wann das Ereignis auftritt.

Das Symbol  weist darauf hin, dass die Meldung ignoriert wird.

Hohe Pegel bei ENABLE und CONFIRM signalisieren TRUE; niedrige Pegel signalisieren FALSE. Hoher Pegel bei ERROR signalisiert einen Wert > 0; niedriger Pegel signalisiert Wert = 0.

- SPS Zyklen:** Das Diagramm zeigt, dass zwischen dem Signalablauf und den SPS Zyklen kein Zusammenhang besteht, z.B. die Antwort kann sofort oder einige SPS Zyklen später verfügbar sein.
- Timeout:** Die Zeitkontrolle startet mit einer steigenden Flanke bei ENABLE. Ein Timeout-Ereignis tritt auf, wenn keine Antwort verfügbar ist, bevor die Zeitkontrolle den Timeout-Wert erreicht. Wenn Timeout stattfindet, dann ist ERROR = 3 und ERRORINFO enthält gegebenenfalls Fehlerinformation im SDO-Abbruch Datenformat. Wenn keine zusätzliche Fehlerinformation verfügbar ist, dann ist ERRORINFO gleich 0.

Request: Der Zustand no_error wie auch error zeigt, dass das Request Ereignis unmittelbar mit der an Eingang ENABLE erkannten steigenden Flanke auftritt, d.h. es wird an eine untergeordnete Software Ebene übergeben. Die Ausgänge bleiben auf ihre Initialisierungswerte stehen. Die Programmkontrolle wird an den Aufrufer zurückgegeben, wenn die Antwort oder Abbruch nicht sofort verfügbar ist.

A: Response: Sobald eine SDO-Antwort Meldung verfügbar ist, wird der Ausgang CONFIRM auf TRUE gesetzt und bleibt solange TRUE, solange der Funktionsbaustein mit dem Eingang ENABLE=TRUE aufgerufen wird. Wenn der Funktionsbaustein mit ENABLE=FALSE aufgerufen wird, dann wird CONFIRM auf FALSE, sprich Initialisierungswert gesetzt.

B: Abort: Sobald die SDO-Abbruch Meldung verfügbar ist, wird der Ausgang ERROR auf den Wert 1 gesetzt und weitere Fehlerinformationen sind in dem

CAA_CiA405.library

Ausgang ERRORINFO verfügbar ("weitere Fehler Register" /DS405/). Die Ausgänge verbleiben so lange, wie der Funktionsbaustein mit der Eingang ENABLE = TRUE aufgerufen wird. Wenn der Funktionsbaustein mit ENABLE = FALSE aufgerufen wird, dann werden ERROR und ERRORINFO auf 0 gesetzt, d.h. auf deren Initialisierungswerte.

C: ENABLE = FALSE bevor Response oder Abort verfügbar: Sobald der Funktionsbaustein mit ENABLE = FALSE aufgerufen wird, werden die Ausgänge auf deren Initialisierungswerte gesetzt, d.h. CONFIRM = FALSE, ERROR = 0, ERRORINFO = 0. Eine möglicherweise verfügbare Response oder Abort werden ignoriert, genau so wie später eintreffende Response oder Abort.

D: Timeout: Wenn ein Funktionsbaustein aufgerufen wird und es wird eine Zeitüberschreitung erkannt und Response oder Abort sind nicht verfügbar, dann ist ERROR = 3 und ERRORINFO enthält gegebenenfalls Fehlerinformation im SDO-Abbruch Datenformat.

Wenn keine zusätzliche Fehlerinformation verfügbar ist, dann ist ERRORINFO gleich 0.

Flusssteuerung

Der Funktionsbaustein bleibt im Leerlauf, d.h. alle Ausgänge sind auf deren Initialisierungswerte gesetzt und es wird keine Operation ausgeführt, solange der Eingang ENABLE gleich FALSE ist. Die Anzahl Aufrufe ist unerheblich.

Sobald ENABLE eine steigende Flanke realisiert, startet der Funktionsbaustein die Ausführung, d.h. er wird die Eingangswerte lesen und eine Anfrage an die untergeordnete Software stellen, z.B. einen bestimmten Parameter zu lesen oder zu schreiben. Wenn auf die Anfrage eine Antwort unmittelbar verfügbar ist, wie der Rückgabewert nach einem Funktionsaufruf, dann wird der Funktionsbaustein seine Ausgänge sofort setzen und seine Aufgabe abschließen. Zum Beispiel könnte eine Funktion, die die ID des lokalen Knotens erhält, so arbeiten. Ist die Antwort nicht unmittelbar verfügbar, dann gibt der Funktionsbaustein die Programmkontrolle wieder zurück an den Aufrufer, bei Setzung aller Ausgangssignale auf aktiver Zustand, z.B. CONFIRM = FALSE und ERROR = 0.

Bei jedem Aufruf des Funktionsbausteins werden die Verfügbarkeit der Antwort und gegebenenfalls die Zeitüberschreitung überprüft. Eingangsänderungen, außer ENABLE, werden ignoriert.

Wenn die Antwort einen Fehlerzustand meldet oder die Zeitüberschreitung erreicht ist, dann setzt der Funktionsbaustein die Ausgänge auf Fehlerzustand, z.B. CONFIRM = FALSE und ERROR $\neq$ 0 und beendet seine Aufgabe.

Wenn die Antwort Erfolg meldet, dann setzt der Funktionsbaustein die Ausgänge auf Zustandsbestätigung, d.h. CONFIRM = TRUE und ERROR = 0 und, wenn es eine Leseabfrage war, die Datenausgänge auf die aktuellen abgefragten Werte und schließt seine Aufgabe ab.



In /DS405/ ist nicht klar festgelegt, ob der Funktionsbaustein

- a) sofort,
- b) beim nächsten Aufruf oder
- c) nach der nächsten steigenden Flanke des Eingangs ENABLE erneut ausgeführt wird.

Damit andere Funktionsbausteine die Gelegenheit erhalten, auf das selbe Gerät auf dem selben SDO-Kanal zuzugreifen, wird empfohlen, zumindest einen SPS Zyklus (b) abzuwarten, bevor die Ausführung erneut gestartet wird. Dadurch wird verhindert, dass ein Funktionsbaustein, der dauerhaft mit Eingang ENABLE = TRUE aufgerufen wird, die Kommunikation mit einem Gerät blockieren kann.

In den Fällen a) und b) besteht ein möglicher Nachteil darin, dass der Funktionsbaustein nicht auf die Verfügbarkeit einer neuen Antwort, z.B. neue Daten, hinweisen kann. Der Benutzer weiß nicht, wie alt die bereitgestellten Daten sind. Bei Schreibenfragen weiß der Benutzer nicht, ob die Eingangsdaten an das Gerät übertragen werden, bevor diese geändert werden. Des Weiteren ist es unsicher, die gleichen Daten mehrere Male zu übertragen. Aus diesen Gründen ist es ratsam, die Ausführung nicht erneut zu starten, wenn ENABLE gleich TRUE verbleibt (c). Wenn ENABLE gleich FALSE wird, dann werden alle Ausgänge sofort auf deren Initialisierungswerte gesetzt. Eine mögliche verfügbare Antwort wird ignoriert, genau wie eine laufende Abfrage. Eine steigende Flanke von ENABLE startet die Flusssteuerung von Neuem.



In /DS301/ (V 4.02 Kap. 9.2.2.1.7) wird in Bezug auf SDO-Abbruch Folgendes definiert:

"Der Dienst kann jederzeit von beiden, sowohl vom Client, als auch vom Server einer SDO, ausgeführt werden. Wenn der Kunde einer SDO einen ausstehenden Dienst bestätigt hat, dann wird der Abbruch als Bestätigung für diesen Dienst betrachtet."

7.4.3 CIA405_CIA405BASE (Verhaltensmodell)

CIA405BASE

Kurzbeschreibung

Um für alle CIA405-Funktionsbausteine das übliche Verhaltensmodell zu implementieren, wird dieser Funktionsbaustein als eine Basisklasse verwendet. Alle übrigen CIA405-Funktionsbausteine im vorliegenden Dokument erweitern diesen Funktionsbaustein.

Schnittstellenbeschreibung

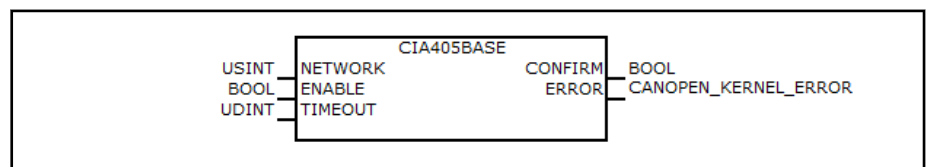


Abb. 7-3:

Schnittstelle des FBs CIA405BASE

CAA_CiA405.library

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	CAN-Kanalnummer; bestimmt auf welchen CAN-Bus ein CiA405-Funktionsbaustein seine Aktion ausführt. MAX_Networks = 64 Die CAN_Kanalnummer ist nicht die gleiche wie die im CAN-Bus-Konfigurator spezifizierte Net_ID. Bei der Berechnung der CAN_Kanalnummer muss folgende Formel verwendet werden: CAN_Kanalnummer := Net_ID + 1
	ENABLE	BOOL	Ausführung frei gegeben
	TIMEOUT	UDINT	Maximale Ausführungszeit in ms 0 bedeutet, dass die Zeitüberschreitung deaktiviert ist
VAR OUTPUT	CONFIRM	BOOL	Erfolgreich ausgeführt Bereich: true / false Initialisierungswert: false
	ERROR	CANOPEN_KERNEL_ERROR	Fehlercode Bereich: 0000 ... FFFF Initialisierungswert: 0 CANOPEN_KERNEL_ERROR, Seite 48

Tab. 7-8: Schnittstelle des FBs CIA405BASE

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NET- WORKS (max. 64)	1	0/1 an ENABLE
TIMEOUT	UDINT	0		0	0/1 an ENABLE

Tab. 7-9: Eingangsverhalten

7.4.4 Funktionsbausteine "Lesen und Schreiben von SDO"

SDO_WRITE4

Kurzbeschreibung Der Funktionsbaustein dient zum Senden der SDO (Service Data Objects).
Der Funktionsbaustein erweitert [CIA405BASE, Seite 55](#).

Schnittstellenbeschreibung

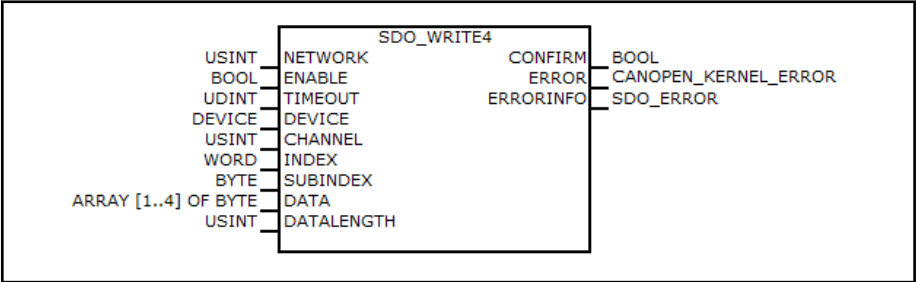


Abb. 7-4: Schnittstelle des FBs SDO_WRITE4

CAA_CiA405.library

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE, Seite 55
	ENABLE	BOOL	siehe CIA405BASE, Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE, Seite 55
	DEVICE	DEVICE	Adressiertes Gerät; 0 bedeutet lokales Gerät DEVICE, Seite 49
	CHANNEL	USINT	SDO Kanalnummer
	INDEX	WORD	Objektindex
	SUBINDEX	BYTE	Objektsubindex
	DATA	ARRAY [1..4] OF BYTE	Objektdaten
	DATALength	USINT	Länge gültiger Daten in Bytes
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE, Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE, Seite 55
	ERRORINFO	SDO_ERROR	SDO-Abbruch-Kode 0 ... 2 ³² -1 SDO_ERROR, Seite 49

Tab. 7-10: Schnittstelle des FBs SDO_WRITE4

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE
DEVICE	DEVICE	0	127	0	0/1 an ENABLE
CHANNEL	USINT	1	128	1	0/1 an ENABLE
INDEX	WORD	16#0000	16#FFFF	16#0000	0/1 an ENABLE
SUBINDEX	BYTE	16#00	16#FF	16#00	0/1 an ENABLE
DATA	ARRAY [1..4] OF BYTE	4(16#00)	4(16#FF)	4(16#00)	0/1 an ENABLE
DATALength	USINT	1	4	1	0/1 an ENABLE

Tab. 7-11: Eingangsverhalten

CAA_CiA405.library

Funktionsbeschreibung

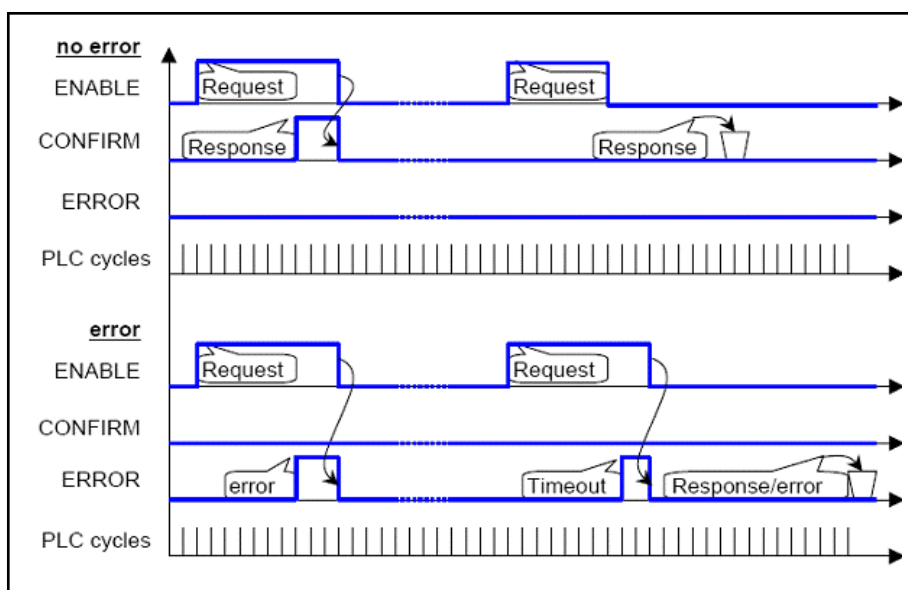


Abb. 7-5: Request/Response Diagramm, Zeitdiagramm für SDO_WRITE4

Nachdem alle Daten den Eingängen zur Verfügung gestellt wurden, wird ENABLE auf TRUE gesetzt (1).

Die SDO wird gesendet und wenn die CANopen-Software Erfolg an den Funktionsbaustein meldet, dann wird der Ausgang CONFIRM in TRUE (2) geändert.

Der Aufrufer wird dies bemerken und ENABLE in FALSE (3) ändern, was wiederum dazu führt, dass der Ausgang CONFIRM zu FALSE (4) wechseln wird.

- Bei steigender Flanke an Eingang ENABLE beginnt der Funktionsbaustein mit dem Sammeln der Eingänge und initiiert die Übertragung der in DATA und DATALENGTH spezifizierten SDO an den mit DEVICE, INDEX und SUBINDEX spezifizierten Empfänger. Der Wert von DEVICE ist auf den Bereich 1 bis 127 eingegrenzt. Zwecks Zugriff auf das lokale Objektwörterbuch ist die Verwendung des Werts 0 für DEVICE zulässig.
- Mit TRUE an Eingang ENABLE wird dem Funktionsbaustein die Fortführung der Ausführung gestattet. Wenn ein Ergebnis auf Anfrage von einer untergeordneten Ebene der CANopen-Software aufgerufen wird, dann werden die Ausgänge CONFIRM, ERROR und ERRORINFO entsprechend gesetzt.
- Bei fallender Flanke des Eingangs ENABLE wird der Funktionsbaustein beendet. Wenn die Übertragung nicht abgeschlossen ist, dann wird sie, falls möglich, abgebrochen. Die Ausgänge CONFIRM und ERROR werden jeweils auf FALSE und "no error" gesetzt.
- Mit einem FALSE an Eingang ENABLE gibt der Funktionsbaustein sofort die Kontrolle zurück und unternimmt keinerlei Aktion.
- Eine genauere Beschreibung des Verhaltens von ENABLE, CONFIRM und ERROR finden Sie in [Steuermechanismus, Seite 52](#).
- Informationen zu den Eingängen NETWORK und TIMEOUT finden Sie in [Zusätzliche Parameter, Seite 51](#).
- Der als ARRAY[1..4] OF BYTE definierter Eingang DATA ist wie nachstehend zu sehen geordnet:

Beispiel Der Wert 16#01020304 (hexadezimal) wird als ARRAY[1..4] OF BYTE := 04, 03, 02, 01 zugewiesen.

Ergebnis und Fehlerbehandlung

Das Ergebnis eines Schreibvorgangs wird (unmittelbar nach dem Schreibauf-ruf oder später) an die Ausgänge CONFIRM, ERROR und ERRORINFO übertragen, mit steigender Flanke entweder auf CONFIRM oder auf ERROR.

Im Falle, wo ERROR gleich 1 ist (anderer Fehler), stellt ERRORINFO mehr spezifische Informationen bezüglich Ursache bereit. Dies ist insbesondere der Fall bei Auftreten eines SDO-Abbruchs. Dann enthält ERRORINFO den Abbruchcode.

Im Allgemeinen kann nicht vorausgesetzt werden, dass diese Funktion syn-chron abgeschlossen wird; es sollte demgegenüber eher möglich sein, dass diese Funktion weiter ausgeführt wird, während der Aufruf von "SDO_WRI-TE4" zurückgegeben wird und das SPS-Programm fortgesetzt wird. Aus die-sem Grund ist das Ergebnis eines Aufrufs möglicherweise erst mehrere Zy-klen nach Anlegen einer steigenden Flanke an ENABLE verfügbar.

SDO_WRITE_DATA

Kurzbeschreibung

Der Funktionsbaustein dient zum Senden der SDO (Service Data Objects).

Dieser Funktionsbaustein kann als Alternative zu den SDO_WRITE_DATAx-Funktionsbausteinen verwendet werden. Er unterstützt die beliebige Daten-länge.

Das Verhalten und Timing entspricht dem von [SDO_WRITE4, Seite 56](#). Der einzige Unterschied besteht im Datentyp der Variablen DATA und DATA-LENGTH.

Der Funktionsbaustein erweitert [CIA405BASE, Seite 55](#).

Schnittstellenbeschreibung

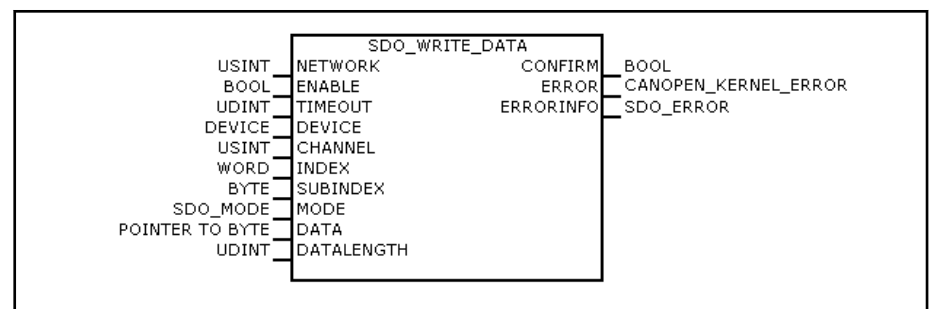


Abb. 7-6: Schnittstelle des FBs SDO_WRITE_DATA

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE, Seite 55
	ENABLE	BOOL	siehe CIA405BASE, Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE, Seite 55
	DEVICE	DEVICE	Adressiertes Gerät; 0 bedeutet lokales Gerät DEVICE, Seite 49
	CHANNEL	USINT	SDO Kanalnummer
	INDEX	WORD	Objektindex
	SUBINDEX	BYTE	Objektsubindex
	MODE	SDO_MODE	Spezifiziert den SDO Mode, der genutzt werden soll SDO_MODE, Seite 47
	DATA	POINTER TO BYTE	Zeiger auf zu schreibende Daten, (Daten sind in little endian byte order)

CAA_CiA405.library

	Name	Typ	Beschreibung
	DATALENGTH	UDINT	Länge gültiger Daten in Bytes
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE , Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE , Seite 55
	ERRORINFO	SDO_ERROR	SDO-Abbruch-Kode 0 ... 2 ³² -1 SDO_ERROR , Seite 49

Tab. 7-12: Schnittstelle des FBs SDO_WRITE_DATA

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE
DEVICE	DEVICE	0	127	0	0/1 an ENABLE
CHANNEL	USINT	1	128	1	0/1 an ENABLE
INDEX	WORD	16#0000	16#FFFF	16#0000	0/1 an ENABLE
SUBINDEX	BYTE	16#00	16#FF	16#00	0/1 an ENABLE
MODE	SDO_MODE	1	...	1	0/1 an ENABLE
DATA	POINTER TO BYTE				0/1 an ENABLE
DATALENGTH	UINT	1	...	1	0/1 an ENABLE

Tab. 7-13: Eingangsverhalten

SDO_READ4

Kurzbeschreibung

Der Funktionsbaustein dient zum Empfangen der SDO (Service Data Objects).

Der Funktionsbaustein erweitert [CIA405BASE](#), Seite 55.

Schnittstellenbeschreibung

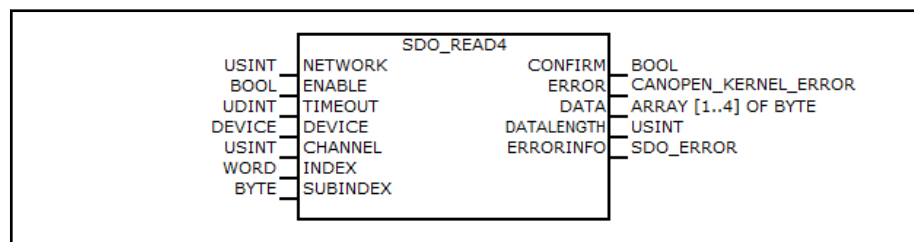


Abb. 7-7: Schnittstelle des FBs SDO_READ4

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE , Seite 55
	ENABLE	BOOL	siehe CIA405BASE , Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE , Seite 55

	Name	Typ	Beschreibung
	DEVICE	DEVICE	Adressiertes Gerät; 0 bedeutet lokales Gerät DEVICE, Seite 49
	CHANNEL	USINT	SDO Kanalnummer
	INDEX	WORD	Objektindex
	SUBINDEX	BYTE	Objektsubindex
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE, Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE, Seite 55
	DATA	ARRAY [1..4] OF BYTE	BYTE-ARRAY in das die Daten geschrieben werden. Die Daten sind in CANopen Byte Order (= Little Endian) abgelegt.
	DATALength	USINT	Länge gültiger Daten in Bytes
	ERRORINFO	SDO_ERROR	SDO-Abbruch-Kode 0 ... $2^{32}-1$ SDO_ERROR, Seite 49

Tab. 7-14: Schnittstelle des FBs SDO_READ4

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE
DEVICE	DEVICE	0	127	0	0/1 an ENABLE
CHANNEL	USINT	1	128	1	0/1 an ENABLE
INDEX	WORD	16#0000	16#FFFF	16#0000	0/1 an ENABLE
SUBINDEX	BYTE	16#00	16#FF	16#00	0/1 an ENABLE

Tab. 7-15: Eingangsverhalten

CAA_CiA405.library

Funktionsbeschreibung

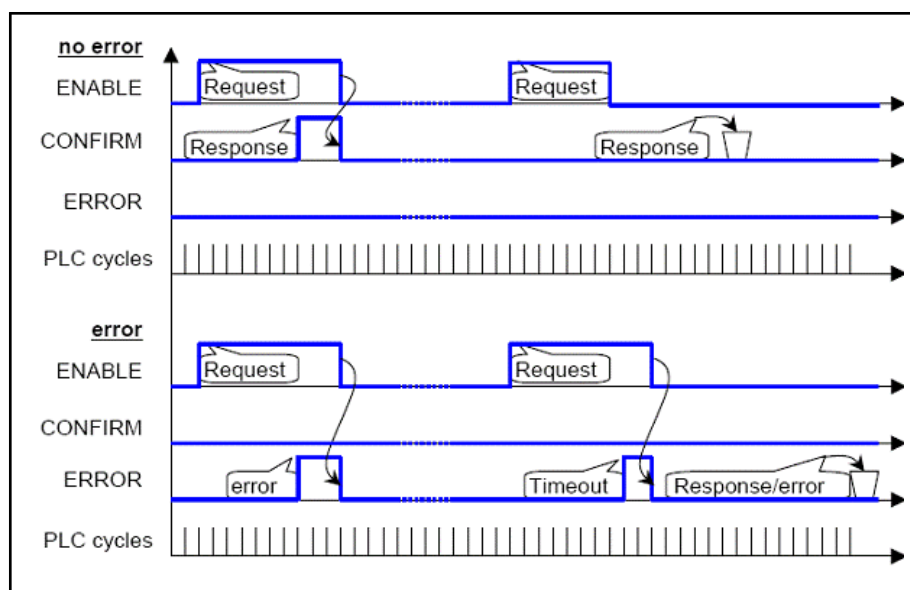


Abb. 7-8: Request/Response Diagramm, Zeitdiagramm für SDO_READ4

Nachdem alle Daten den Eingängen zur Verfügung gestellt wurden, wird ENABLE auf TRUE gesetzt (1).

Die SDO wird empfangen, und wenn die CANopen-Software Erfolg an den Funktionsbaustein meldet, dann wird der Ausgang CONFIRM in TRUE (2) geändert.

Der Aufrufer wird dies bemerken und ENABLE in FALSE (3) ändern, was wiederum dazu führt, dass der Ausgang CONFIRM zu FALSE (4) wechseln wird.

- Bei steigender Flanke an Eingang ENABLE beginnt der Funktionsbaustein mit dem Sammeln der Eingänge und initiiert die Übertragung der mit DEVICE, INDEX und SUBINDEX spezifizierten SDO. Der Wert von DEVICE ist auf den Bereich 1 bis 127 eingegrenzt. Zwecks Zugriff auf das lokale Objektwörterbuch ist die Verwendung des Werts 0 für DEVICE zulässig.
- Mit TRUE an Eingang ENABLE wird dem Funktionsbaustein die Fortführung der Ausführung gestattet. Wenn ein Ergebnis auf Anfrage von einer untergeordneten Ebene der CANopen-Software aufgerufen wird, dann werden die Ausgänge CONFIRM, DATA und DATALENGTH (wenn erfolgreich) oder ERROR und ERRORINFO (wenn fehlgeschlagen) entsprechend gesetzt.
- Bei fallender Flanke des Eingangs ENABLE wird der Funktionsbaustein beendet. Wenn die Übertragung nicht abgeschlossen ist, dann wird sie, falls möglich, abgebrochen. Die Ausgänge CONFIRM und ERROR werden jeweils auf FALSE und "no error" gesetzt.
- Mit einem FALSE an Eingang ENABLE gibt der Funktionsbaustein sofort die Steuerung zurück und unternimmt keinerlei Aktivitäten.
- Eine genauere Beschreibung des Verhaltens von ENABLE, CONFIRM und ERROR finden Sie in [Steuermechanismus, Seite 52](#).
- Informationen zu den Eingängen NETWORK und TIMEOUT finden Sie in [Zusätzliche Parameter, Seite 51](#).
- Der als ARRAY[1..4] OF BYTE definierter Ausgang DATA ist wie nachstehend zu sehen geordnet:

Beispiel Der Wert 16#01020304 (hexadezimal) wird als ARRAY[1..4] OF BYTE := 04, 03, 02, 01 zugewiesen.

SDO_READ_DATA

Kurzbeschreibung Der Funktionsbaustein dient zum Empfangen der SDO (Service Data Objects).

Dieser Funktionsbaustein kann als Alternative zu den SDO_READx-Funktionsbausteinen verwendet werden. Er unterstützt die beliebige Datenlänge. Das Verhalten und Timing entspricht dem von [SDO_READ4, Seite 60](#). Der einzige Unterschied besteht im Datentyp der Variablen DATA und DATALENGTH sowie in der Möglichkeit die Art der Abspeicherung (MODE) zu beeinflussen.

Der Funktionsbaustein erweitert [CIA405BASE, Seite 55](#).

Schnittstellenbeschreibung

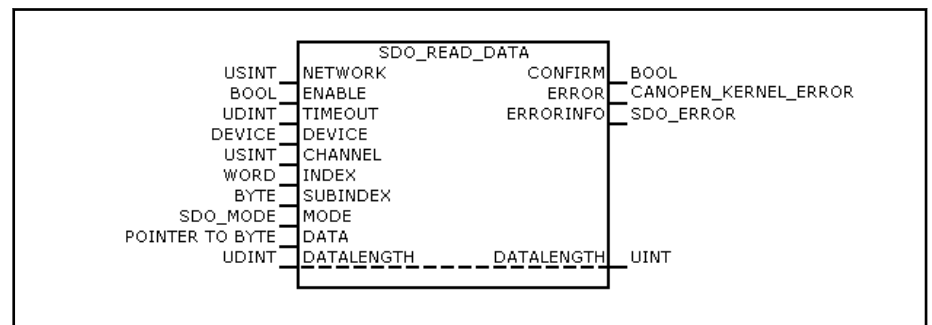


Abb. 7-9: Schnittstelle des FBs SDO_READ_DATA

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE, Seite 55
	ENABLE	BOOL	siehe CIA405BASE, Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE, Seite 55
	DEVICE	DEVICE	Adressiertes Gerät; 0 bedeutet lokales Gerät DEVICE, Seite 49
	CHANNEL	USINT	SDO Kanalnummer
	INDEX	WORD	Objektindex
	SUBINDEX	BYTE	Objektsubindex
	MODE	SDO_MODE	Spezifiziert den SDO Mode, der genutzt werden soll SDO_MODE, Seite 47
	DATA	POINTER TO BYTE	Zeiger auf den Speicherbereich auf den die gelesenen Daten geschrieben werden sollen. Die Daten werden in CANopen Byte Order (= Little Endian) abgelegt.
VAR_OUTPUT	CONFIRM	BOOL	siehe CIA405BASE, Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE, Seite 55

CAA_CiA405.library

	Name	Typ	Beschreibung
	ERRORINFO	SDO_ERROR	SDO-Abbruch-Kode 0 ... 2 ³² -1 SDO_ERROR, Seite 49
VAR_INOUT	DATALength	UDINT	Eingang: Fassungsvermögen des Puffers Ausgang: Länge gültiger Daten in Bytes Initialisierungswert: 0

Tab. 7-16: Schnittstelle des FBs SDO_READ_DATA

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NET- WORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE
DEVICE	DEVICE	0	127	0	0/1 an ENABLE
CHANNEL	USINT	1	128	1	0/1 an ENABLE
INDEX	WORD	16#0000	16#FFFF	16#0000	0/1 an ENABLE
SUBINDEX	BYTE	16#00	16#FF	16#00	0/1 an ENABLE

Tab. 7-17: Eingangsverhalten

Obsolater Funktionsbaustein

SDO_WRITE

Kurzbeschreibung



Dieser Funktionsbaustein ist obsolet. Er unterstützt nur Daten bis 64kb.
Bitte nutzen Sie stattdessen den neuen Funktionsbaustein [SDO_WRITE_DATA, Seite 59](#).

Der Funktionsbaustein dient zum Senden der SDO (Service Data Objects).
Dieser Funktionsbaustein kann als Alternative zu den SDO_WRITEx-Funktionsbausteinen verwendet werden. Er unterstützt die beliebige Datenlänge.
Das Verhalten und Timing entspricht dem von [SDO_WRITE4, Seite 56](#). Der einzige Unterschied besteht im Datentyp der Variablen DATA und DATA-LENGTH.
Der Funktionsbaustein erweitert [CIA405BASE, Seite 55](#).

Schnittstellenbeschreibung

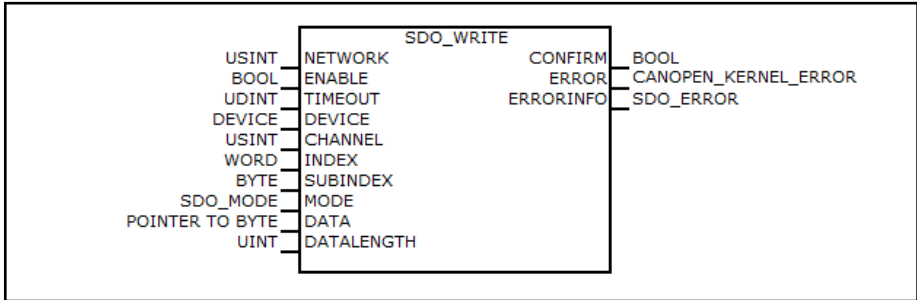


Abb. 7-10: Schnittstelle des FBs SDO_WRITE

CAA_CiA405.library

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE, Seite 55
	ENABLE	BOOL	siehe CIA405BASE, Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE, Seite 55
	DEVICE	DEVICE	Adressiertes Gerät; 0 bedeutet lokales Gerät DEVICE, Seite 49
	CHANNEL	USINT	SDO Kanalnummer
	INDEX	WORD	Objektindex
	SUBINDEX	BYTE	Objektsubindex
	MODE	SDO_MODE	SDO_MODE, Seite 47
	DATA	POINTER TO BYTE	Objektdaten
	DATALength	UINT	Länge gültiger Daten in Bytes
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE, Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE, Seite 55
	ERRORINFO	SDO_ERROR	SDO-Abbruch-Kode 0 ... $2^{32}-1$ SDO_ERROR, Seite 49

Tab. 7-18: Schnittstelle des FBs SDO_WRITE

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE
DEVICE	DEVICE	0	127	0	0/1 an ENABLE
CHANNEL	USINT	1	128	1	0/1 an ENABLE
INDEX	WORD	16#0000	16#FFFF	16#0000	0/1 an ENABLE
SUBINDEX	BYTE	16#00	16#FF	16#00	0/1 an ENABLE
MODE	SDO_MODE	1	...	1	0/1 an ENABLE
DATA	POINTER TO BYTE				0/1 an ENABLE
DATALength	UINT	1	...	1	0/1 an ENABLE

Tab. 7-19: Eingangsverhalten

CAA CiA405.library

SDO READ

Kurzbeschreibung



Dieser Funktionsbaustein ist obsolet. Er unterstützt nur Daten bis 64kb und keinen Block Transfer.

Bitte nutzen Sie stattdessen den neuen Funktionsbaustein [SDO READ DATA](#), Seite 63.

Der Funktionsbaustein dient zum Empfangen der SDO (Service Data Objects).

Dieser Funktionsbaustein kann als Alternative zu den SDO_READx-Funktionsbausteinen verwendet werden. Er unterstützt die beliebige Datenlänge. Das Verhalten und Timing entspricht dem von [SDO_READ4, Seite 60](#). Der einzige Unterschied besteht im Datentyp der Variablen DATA und DATA-LENGTH.

Der Funktionsbaustein erweitert CIA405BASE. Seite 55.

Schnittstellenbeschreibung

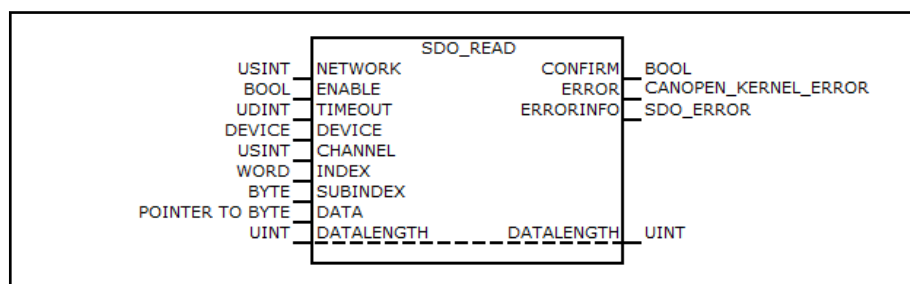


Abb. 7-11: Schnittstelle des FBs SDO READ

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE, Seite 55
	ENABLE	BOOL	siehe CIA405BASE, Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE, Seite 55
	DEVICE	DEVICE	Adressiertes Gerät; 0 bedeutet lokales Gerät DEVICE, Seite 49
	CHANNEL	USINT	SDO Kanalnummer
	INDEX	WORD	Objektindex
	SUBINDEX	BYTE	Objektsubindex
	DATA	POINTER TO BYTE	Zeiger auf den Speicherbereich auf den die gelesenen Daten geschrieben werden sollen. Die Daten werden in CANopen Byte Order (= Little Endian) abgelegt.
VAR_OUTPUT	CONFIRM	BOOL	siehe CIA405BASE, Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE, Seite 55

	Name	Typ	Beschreibung
	ERRORINFO	SDO_ERROR	SDO-Abbruch-Kode 0 ... $2^{32}-1$ SDO_ERROR, Seite 49
VAR_INOUT	DATALength	UINT	Eingang: Fassungsvermögen des Puffers Ausgang: Länge gültiger Daten in Bytes Initialisierungswert: 0

Tab. 7-20: Schnittstelle des FBs SDO_READ

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE
DEVICE	DEVICE	0	127	0	0/1 an ENABLE
CHANNEL	USINT	1	128	1	0/1 an ENABLE
INDEX	WORD	16#0000	16#FFFF	16#0000	0/1 an ENABLE
SUBINDEX	BYTE	16#00	16#FF	16#00	0/1 an ENABLE

Tab. 7-21: Eingangsverhalten

7.4.5 Funktionsbaustein "Bezeichner von eigenem Knoten"

GET_LOCAL_NODE_ID

Kurzbeschreibung

Funktionsbaustein GET_LOCAL_NODE_ID gibt die eigene Knoten-ID zurück.
 Der Funktionsbaustein erweitert [CIA405BASE, Seite 55](#).

Schnittstellenbeschreibung

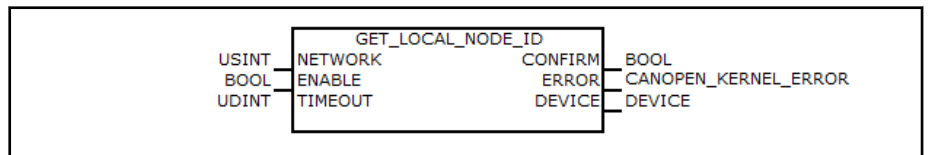


Abb. 7-12: Schnittstelle des FBs GET_LOCAL_NODE_ID

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE, Seite 55
	ENABLE	BOOL	siehe CIA405BASE, Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE, Seite 55
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE, Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE, Seite 55
	DEVICE	DEVICE	ID von lokalem Knoten; 0 bedeutet Fehler DEVICE, Seite 49

Tab. 7-22: Schnittstelle des FBs GET_LOCAL_NODE_ID

CAA_CiA405.library

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE

Tab. 7-23: Eingangsverhalten

Funktionsbeschreibung

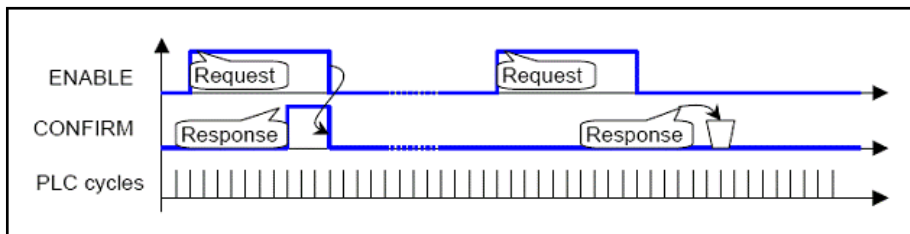


Abb. 7-13: Request/Response Diagramm; Zeitdiagramm des FBs GET_LOCAL_NODE_ID

- Datenausgang DEVICE ist nur gültig, wenn CONFIRM gleich TRUE ist. Wenn CONFIRM gleich FALSE ist, dann ist der Datenausgang ungültig und kann einen beliebigen Wert enthalten.
- Die Anfrage für Daten wird nur einmal mit steigender Flanke auf ENABLE initiiert. Eine weitere Anfrage wird nicht initiiert, bevor nicht eine weitere steigende Flanke auf ENABLE erkannt wird.
- Für den Fall eines Fehlers im Verlauf der Ausführung des Funktionsbausteins, ist dieser angewiesen, die ungültige Knotennummer, DEVICE=0 und ERROR=0 auszugeben.
- Eine genauere Beschreibung des Verhaltens von ENABLE, CONFIRM und ERROR finden Sie in [Steuermechanismus, Seite 52](#).
- Informationen zu den Eingängen NETWORK und TIMEOUT finden Sie in [Zusätzliche Parameter, Seite 51](#).

7.4.6 Funktionsbausteine "Abfragestatus"

GET_STATE

Kurzbeschreibung

Funktionsbaustein gibt den aktuellen Zustand eines CANopen-Netzwerkgeräts zurück.

Der Funktionsbaustein erweitert [CIA405BASE, Seite 55](#).

Schnittstellenbeschreibung

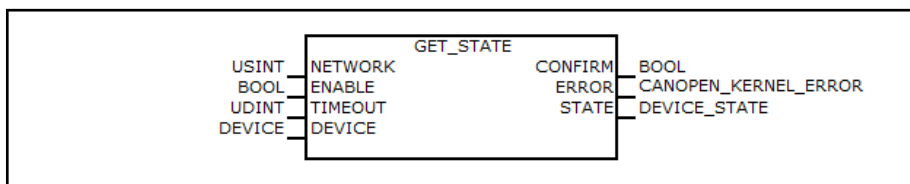


Abb. 7-14: Schnittstelle des FBs GET_STATE

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE, Seite 55
	ENABLE	BOOL	siehe CIA405BASE, Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE, Seite 55

	Name	Typ	Beschreibung
	DEVICE	DEVICE	Adressiertes Gerät; 0 bedeutet lokales Gerät DEVICE, Seite 49
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE, Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE, Seite 55
	STATE	DEVICE_STATE	Gerätezustand DEVICE_STATE, Seite 46

Tab. 7-24: Schnittstelle des FBs GET_STATE

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE
DEVICE	DEVICE	0	127	0	0/1 an ENABLE

Tab. 7-25: Eingangsverhalten

Funktionsbeschreibung

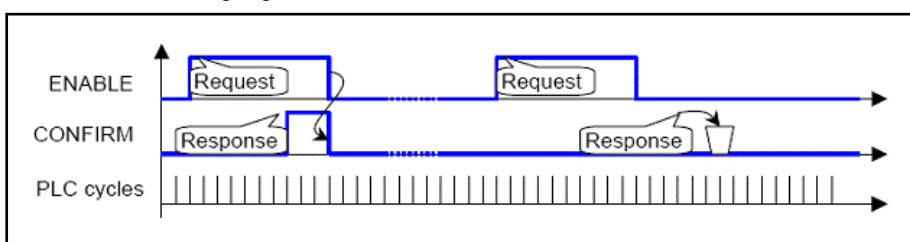


Abb. 7-15: Request/Response Diagramm, Zeitdiagramm des FBs GET_STATE

Wenn der Zustand nicht bekannt ist, dann gibt der Funktionsbaustein die Steuerung zurück mit dem Zustand UNKNOWN. Dies ist der Fall, wenn das Gerät nicht überwacht wird und der Heartbeat nicht läuft.

Wird die Überwachung (Guarding) vorgenommen und das Gerät antwortet nicht, oder ist aus anderen Gründen bekannt, dass das Gerät nicht verfügbar ist, dann gibt die Funktion den Zustand NOT_AVAIL zurück.

Wenn DEVICE gleich 0 oder gleich der eigenen Knoten-ID ist, dann gibt der Funktionsbaustein den Zustand des lokalen Kommunikationsprozesses zurück.

- Datenausgang STATE ist nur gültig, wenn CONFIRM gleich TRUE ist. Wenn CONFIRM gleich FALSE ist, dann ist der Datenausgang ungültig und kann einen beliebigen Wert enthalten.
- Die Anfrage für Daten wird nur einmal mit steigender Flanke auf ENABLE initiiert. Eine weitere Anfrage wird nicht initiiert bevor nicht eine weitere steigende Flanke auf ENABLE erkannt wird.
- Eine genauere Beschreibung des Verhaltens von ENABLE, CONFIRM und ERROR finden Sie in [Steuermechanismus, Seite 52](#).
- Informationen zu den Eingängen NETWORK und TIMEOUT finden Sie in [Zusätzliche Parameter, Seite 51](#).

CAA_CiA405.library

- Für den Fall eines Fehler während der Ausführung des Funktionsbausteins ist dieser angewiesen, STATE = UNBEKANNT (6) mit CONFIRM = TRUE und ERROR = 0 auszugeben.

GET_CANOPEN_KERNEL_STATE

Kurzbeschreibung Der Funktionsbaustein gibt den aktuellen Zustand des CANopen-Kernel zurück.

Der Funktionsbaustein erweitert [CIA405BASE](#), Seite 55.

Schnittstellenbeschreibung

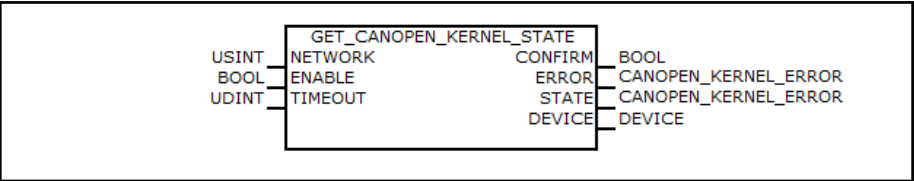


Abb. 7-16: Schnittstelle des FBs GET_CANOPEN_KERNEL_STATE

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE , Seite 55
	ENABLE	BOOL	siehe CIA405BASE , Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE , Seite 55
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE , Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE , Seite 55
	STATE	CANOPEN_KERNEL_ERROR	Kernel-Zustand CANOPEN_KERNEL_ERROR , Seite 48
	DEVICE	DEVICE	Weitere mit STATE verbundene Geräteinformation; 0 bedeutet lokales Gerät DEVICE , Seite 49

Tab. 7-26: Schnittstelle des FBs GET_CANOPEN_KERNEL_STATE

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE

Tab. 7-27: Eingangsverhalten

Funktionsbeschreibung

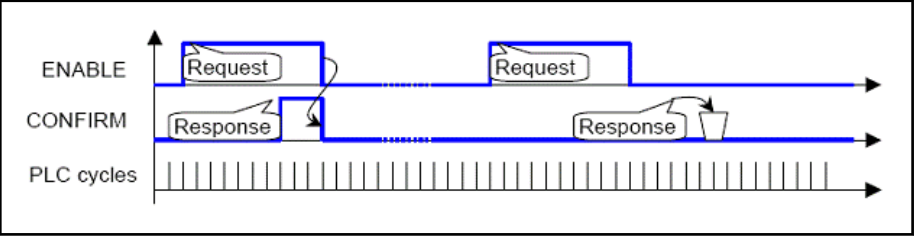


Abb. 7-17: Request/Response Diagramm, Zeitdiagramm des FBs GET_CANOPEN_KERNEL_STATE

- Datenausgang STATE ist nur gültig, wenn CONFIRM gleich TRUE ist.

Wenn CONFIRM gleich FALSE ist, dann ist der Datenausgang ungültig und kann einen beliebigen Wert enthalten.

- Die Anfrage für Daten wird nur einmal mit steigender Flanke auf ENABLE initiiert. Eine weitere Anfrage wird nicht initiiert bevor nicht eine weitere steigende Flanke auf ENABLE erkannt wird.
- Eine genauere Beschreibung des Verhaltens von ENABLE, CONFIRM und ERROR finden Sie in [Steuermechanismus, Seite 52](#).
- Informationen zu den Eingängen NETWORK und TIMEOUT finden Sie in [Zusätzliche Parameter, Seite 51](#).
- Für den Fall eines Fehler während der Ausführung des Funktionsbausteins ist dieser angewiesen, DEVICE = 0 (lokales Gerät) mit STATE <> 0 und ERROR = 0 auszugeben.
- Der Funktionsbaustein berichtet jeden Fehler, ausgenommen
 - a) SDO-Abbruch-Angaben, die von einem SDO-Clients erzeugt werden und
 - b) von einem CAN-Gerät empfangene EMCY-Meldungen.

SDO-Abbruch Angaben werden von den Funktionsbausteinen [SDO_READ, Seite 66](#), und [SDO_WRITE, Seite 64](#) behandelt.

EMCY Meldungen werden von den Funktionsbausteinen [RECV_EMCY, Seite 75](#) und [RECV_EMCY_DEV, Seite 73](#), behandelt.
- Sobald der STATE berichtet wird, wird dies bestätigt an die Fehlerbehandlung der untergeordnete Ebene. Das bedeutet z.B. die Fehlermeldung im Speicher zu löschen, um den Eingang einer anderen Fehlermeldung, die bei der nächsten Datenabfrage berichtet wird, freizugeben.

7.4.7 Funktionsbausteine "Netzwerkverwaltung"

NMT

Kurzbeschreibung

Der Funktionsbaustein steuert die Netzwerkverwaltungsfunktionen für einen oder für alle CANopen-Knoten.

Der Funktionsbaustein erweitert [CIA405BASE, Seite 55](#).

Schnittstellenbeschreibung

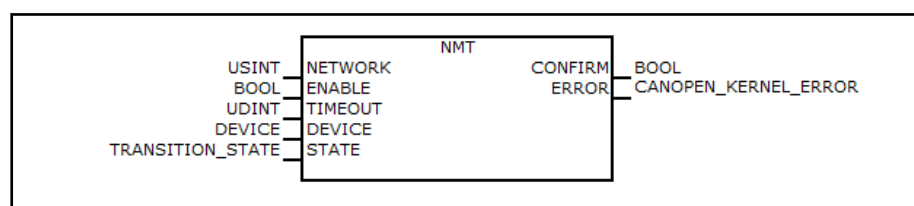


Abb. 7-18: Schnittstelle des FBs NMT

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE, Seite 55
	ENABLE	BOOL	siehe CIA405BASE, Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE, Seite 55
	DEVICE	DEVICE	Adressiertes Gerät; 0 bedeutet lokales Gerät DEVICE, Seite 49
	STATE	TRANSITION_STATE	Zustandsübergänge der CANopen-Netzwerkschicht TRANSITION_STATE, Seite 48

CAA_CiA405.library

	Name	Typ	Beschreibung
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE, Seite 55
	ERROR	CANOPEN NEL_ERROR	siehe CIA405BASE, Seite 55

Tab. 7-28: Schnittstelle des FBs NMT

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NET- WORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE
DEVICE	DEVICE	0	127	0	0/1 an ENABLE
STATE	TRANSITI- ON_STATE	16#00	16#8FF	16#00	0/1 an ENABLE

Tab. 7-29: Eingangsverhalten

Funktionsbeschreibung

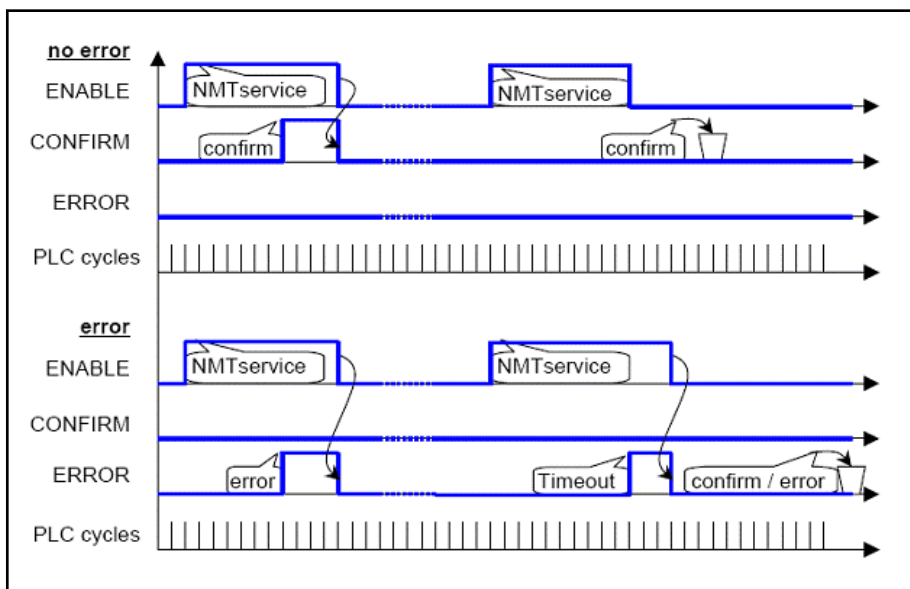


Abb. 7-19: NMT-Service-Diagramm, Zeitdiagramm des FBs NMT

- Eine genauere Beschreibung des Verhaltens von ENABLE, CONFIRM und ERROR finden Sie in [Steuermechanismus, Seite 52](#).
- Informationen zu den Eingängen NETWORK und TIMEOUT finden Sie in [Zusätzliche Parameter, Seite 51](#).
- **NMTservice:** Das NMT-Service-Ereignis ist die Anfrage an den adressierten Knoten (DEVICE=0 bedeutet: Alle Knoten) die gewünschte Zustandsänderung vorzunehmen.

Bei der Verwendung von **DEVICE = 0** kann es zum Fehler kommen, wenn ein NMT Master oder Device nicht alle Befehle unterstützt. In diesem Fall kann das Flag **TRANSITION_STATE.ALL_EXCEPT-NMT-AND-SENDER** für den Eingang **STATE** gesetzt werden, wenn der NMT Befehl nicht den NMT Master und das sendende Device adressiert werden soll.

- **confirm:** Das Ereignis confirm ist die positive Bestätigung, dass der Dienst fehlerfrei abgearbeitet wurde, z.B. der CANopen Kernel gibt no_error an den Funktionsbaustein zurück.
- **error:** Das Ereignis error ist die negative Bestätigung, dass der Dienst nicht fehlerfrei abgearbeitet wurde, z.B. der CANopen Kernel gibt einen Fehler an den Funktionsbaustein zurück.
- **Timeout:** Die Zeitkontrolle startet mit einer steigenden Flanke bei ENABLE. Ein Timeout Ereignis tritt auf, wenn keine Bestätigung an den Funktionsbaustein zurückgegeben wurde, bevor die Zeitkontrolle den Timeout-Wert erreicht. Bei Auftreten von Timeout, ERROR = 3.

RECV_EMCY_DEV

Kurzbeschreibung

Der Funktionsbaustein überprüft, ob ein Notfallobjekt von DEVICE empfangen wurde.

Der Funktionsbaustein erweitert [CIA405BASE](#), Seite 55.

Schnittstellenbeschreibung

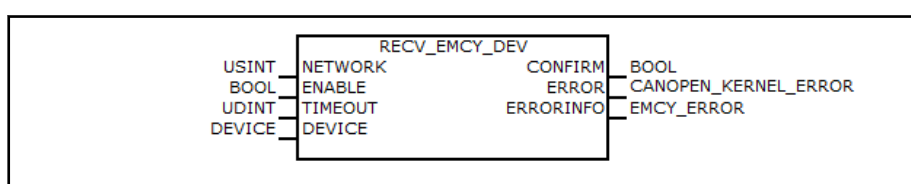


Abb. 7-20: Schnittstelle des FBs RECV_EMCY_DEV

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE , Seite 55
	ENABLE	BOOL	siehe CIA405BASE , Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE , Seite 55
	DEVICE	DEVICE	Adressiertes Gerät; 0 bedeutet lokales Gerät DEVICE , Seite 49
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE , Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE , Seite 55
	ERRORINFO	EMCY_ERROR	EMCY Code EMCY_ERROR , Seite 48

Tab. 7-30: Schnittstelle des FBs RECV_EMCY_DEV

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE
DEVICE	DEVICE	0	127	0	0/1 an ENABLE

Tab. 7-31: Eingangsverhalten

CAA_CiA405.library

Funktionsbeschreibung

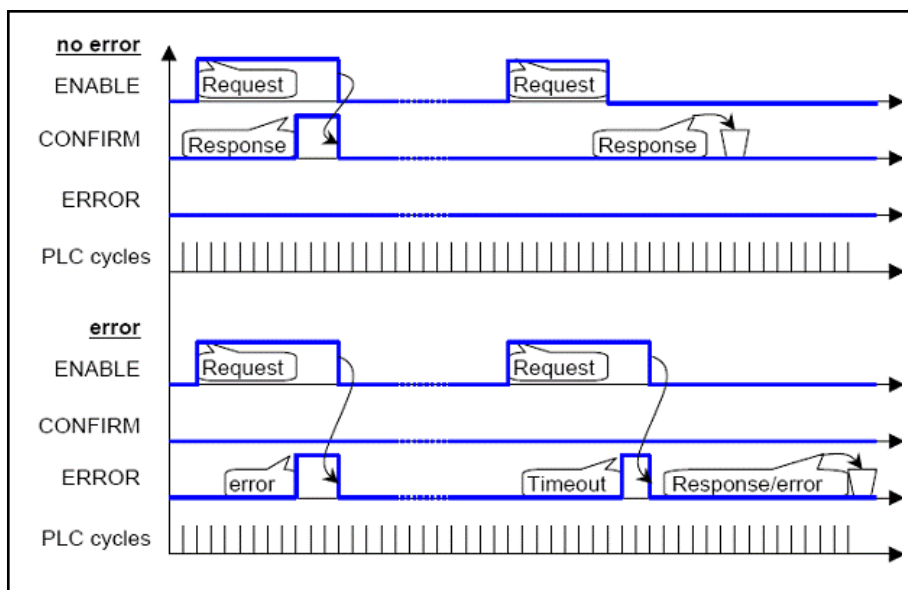


Abb. 7-21: Request/Response Diagramm, Zeitdiagramm des FBs REC_V_EMCY_DEV

Der FB überprüft, ob ein Notfallobjekt (/ DS-301/) von DEVICE empfangen wurde.

Wenn der Funktionsbaustein seine Aktion ohne Fehler ausgeführt hat, dann wird der Ausgang CONFIRM auf TRUE und ERROR auf FALSE gesetzt.

ERRORINFO enthält die Informationen über den empfangenen Notfall.

Es wurde ein Notfall empfangen, wenn der Fehlercode von ERRORINFO verschieden von 0 ist. Ein Fehlercode mit Wert 0 bedeutet, dass das Gerät derzeit keinen Notfall hat.

Ein empfangener Notfall wird nicht gelöscht und wird mit jedem Aufruf des Funktionsbausteins übergeben, bis das Gerät eine Notfalloberstellung gesendet hat.

Wenn ein Fehler während der Überprüfung eines Notfalls auftritt, dann wird CONFIRM auf FALSE und ERROR auf den entsprechenden Fehlerwert gesetzt.

- Eine genauere Beschreibung des Verhaltens von ENABLE, CONFIRM und ERROR finden Sie in [Steuermechanismus, Seite 52](#).
- Informationen zu den Eingängen NETWORK und TIMEOUT finden Sie in [Zusätzliche Parameter, Seite 51](#).
- **Response:** Dieses Ereignis ist die positive Bestätigung, dass die Anfrage ohne Fehler abgearbeitet wurde, z.B. ist der Wert von EMCY an dem Ausgang ERRORINFO verfügbar.

Wurde kein EMCY empfangen, ERRORINFO = 0, d.h. alle Elemente der Struktur [EMCY_ERROR, Seite 48](#), sind Null.

- **error:** Dieses Ereignis ist die negative Bestätigung und bedeutet, dass die Anfrage nicht ohne Fehler abgearbeitet wurde, z.B. gibt CANopen Kernel beim Zugriff auf den Wert von EMCY einen Fehler an den Funktionsbaustein zurück. In diesem Falle ist ERRORINFO = 0.

Siehe auch [CANOPEN_KERNEL_ERROR, Seite 48](#)

- **Timeout:** Die Zeitkontrolle startet mit einer steigenden Flanke bei ENABLE. Ein Timeout Ereignis tritt auf, wenn keine Bestätigung an den

Funktionsbaustein zurückgegeben wurde, bevor die Zeitkontrolle den Timeout-Wert erreicht.

Beim Auftreten von Timeout, ERROR = 3 und ERRORINFO = 0.



Der Funktionsbaustein sollte den EMCY des überprüften Geräts bei jedem Aufruf übergeben.

Oft enthält der früheste EMCY die ursprünglichen Fehlerinformationen; spätere EMCYs ergeben sich oft aus Folgefehlern.

Eine untergeordnete Ebene der Software kann so arbeiten, dass eine eingehende EMCY nicht die EMCYs überschreibt, die verschieden von no_error (gleich „Reset Error“) sind.

Andererseits überschreibt eine eingehende no_error-EMCY in jedem Falle die gespeicherte EMCY.

RECV_EMCY

Kurzbeschreibung

Der Funktionsbaustein überprüft, ob ein Notfallobjekt von irgendeinem DEVICE empfangen wurde.

Der Funktionsbaustein erweitert [CIA405BASE](#), Seite 55.

Schnittstellenbeschreibung

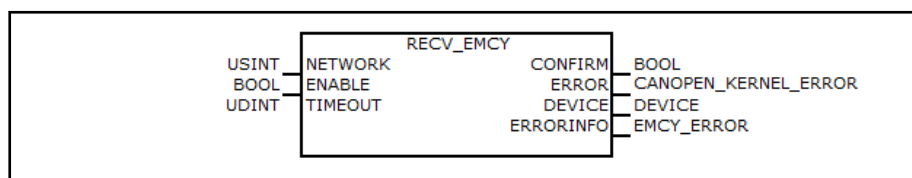


Abb. 7-22: Schnittstelle des FBs RECV_EMCY

	Name	Typ	Beschreibung
VAR_INPUT	NETWORK	USINT	siehe CIA405BASE , Seite 55
	ENABLE	BOOL	siehe CIA405BASE , Seite 55
	TIMEOUT	UDINT	siehe CIA405BASE , Seite 55
VAR OUTPUT	CONFIRM	BOOL	siehe CIA405BASE , Seite 55
	ERROR	CANOPEN_KERNEL_ERROR	siehe CIA405BASE , Seite 55
	DEVICE	DEVICE	Gerät, welches Notfallobjekt empfangen hat 0 bedeutet lokales Gerät DEVICE , Seite 49
	ERRORINFO	EMCY_ERROR	EMCY Code EMCY_ERROR , Seite 48

Tab. 7-32: Schnittstelle des FBs RECV_EMCY

Name	Typ	Min-Wert	Max-Wert	Default-Wert	Übernahme
ENABLE	BOOL	FALSE	TRUE	FALSE	
NETWORK	USINT	1	MAX_NETWORKS	1	0/1 an ENABLE
TIMEOUT	UDINT	0	...	0	0/1 an ENABLE

Tab. 7-33: Eingangsverhalten

CAA_CiA405.library

Funktionsbeschreibung

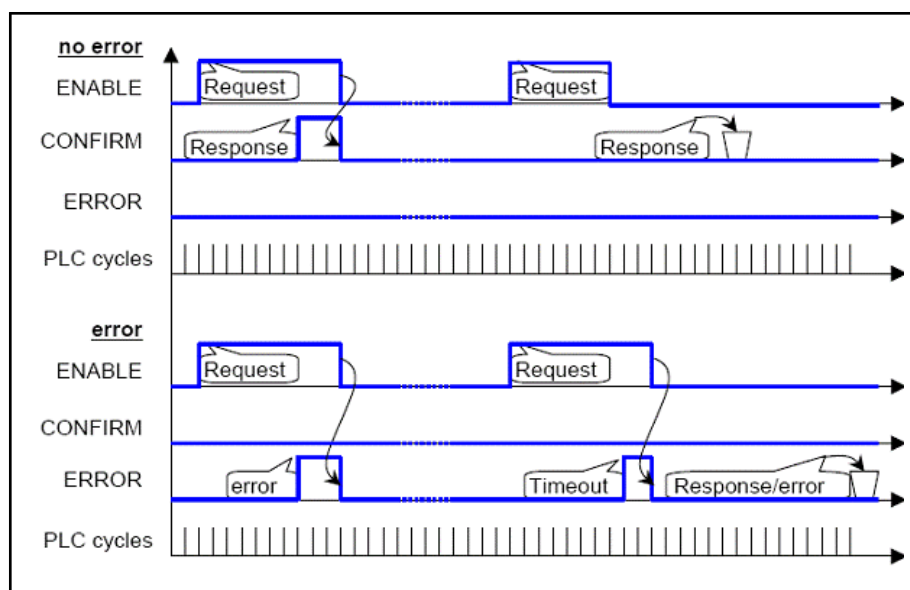


Abb. 7-23: Request/Response Diagramm, Zeitdiagramm des FBs REC_V_EMCY

Der FB überprüft, ob ein Notfallobjekt (/ DS-301/) von irgendeinem DEVICE empfangen wurde.

Wenn der Funktionsbaustein seine Aktion ohne Fehler ausgeführt hat, dann wird der Ausgang CONFIRM auf TRUE und ERROR auf FALSE gesetzt.

ERRORINFO enthält die Information bezüglich des empfangenen Notfalls und DEVICE wird auf die ID des Geräts gesetzt, das das Notfallobjekt gesendet hat.

Es wurde ein Notfall empfangen, wenn beide, DEVICE und der Fehlercode von ERRORINFO, verschieden von 0 sind.

Wenn ein Fehler während der Überprüfung eines Notfalls auftritt, dann wird CONFIRM auf FALSE und ERROR auf den entsprechenden Fehlerwert gesetzt.

- Eine genauere Beschreibung des Verhaltens von ENABLE, CONFIRM und ERROR finden Sie in [Steuermechanismus, Seite 52](#).
- Informationen zu den Eingängen NETWORK und TIMEOUT finden Sie in [Zusätzliche Parameter, Seite 51](#).
-

Response: Dieses Ereignis ist die positive Bestätigung, dass die Anfrage ohne Fehler abgearbeitet wurde, z.B. ist der vom DEVICE empfangene Wert von EMCY an dem Ausgang ERRORINFO verfügbar.

Wurde kein EMCY empfangen, ERRORINFO = 0, d.h. alle Elemente der Struktur [EMCY_ERROR, Seite 48](#), sind Null.

error: Dieses Ereignis ist die negative Bestätigung und bedeutet, dass die Anfrage nicht ohne Fehler abgearbeitet wurde, z.B. gibt CANopen Kernel beim Zugriff auf den Wert von EMCY einen Fehler an den Funktionsbaustein zurück. In diesem Falle ist ERRORINFO = 0.

Timeout: Die Zeitkontrolle startet mit einer steigenden Flanke bei ENABLE. Ein Timeout Ereignis tritt auf, wenn keine Bestätigung an den Funktionsbaustein zurückgegeben wurde, bevor die Zeitkontrolle den Timeout-Wert erreicht.

Beim Auftreten von Timeout, ERROR = 3 und ERRORINFO = 0.



Um alle ausstehenden EMCY Meldungen mit jedem Aufruf des Funktionsbausteins in einer Schleife zu übergeben und zudem mindestens einmal eine "Reset Error"-EMCY zu melden, kann eine untergeordnete Ebene der Software so arbeiten, dass für jedes Gerät ein Speicher verwaltet wird für empfangene EMCY Meldungen, so wie in der Empfehlung für [RECV_EMCY_DEV](#), [Seite 73](#), beschrieben.

Der Funktionsbaustein RECV_EMCY scannt die EMCY Speicher in einer sich über alle bestehenden Geräte erstreckenden Schleife.

Wird im Verlauf dieses Scans ein Gerät ausfindig gemacht, in dessen EMCY-Speicher eine EMCY gefunden wird, die von no_error abweicht, dann wird die EMCY übergeben.

Wird, andernfalls, ein Gerät ausfindig gemacht, für das eine von no_error abweichende EMCY im Verlauf des letzten Scans übergeben wurde und dessen EMCY-Speicher nun no_error enthält, dann wird diese no_error- EMCY übergeben.

Jedes Gerät, für das zuletzt ein no_error übergeben wurde, wird im Verlauf des Scans übersprungen, so dass eine no_error-EMCY nur einmal übergeben wird.

Bei der nächsten Ausführung wird die Suche fortgesetzt. Um darüber informiert zu sein, dass eine ausstehende EMCY mittels Empfang von "Reset Error" seitens eines Geräts gelöscht wurde, wird diese "Reset Error"-EMCY nur einmal übergeben, d.h. DEVICE > 0 mit ERRORINFO = 0.

Um darüber informiert zu sein, dass alle Geräte fehlerfrei sind, übergibt der Funktionsbaustein DEVICE = 0 und ERRORINFO = 0 nach einem vollständigen Scan aller EMCY-Speicher von allen bestehenden Geräten.

8 Service und Support

Für Ihre schnelle und optimale Unterstützung verfügen wir über ein dichtes weltweites Servicenetz. Unsere Experten stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr – auch an Wochenenden und Feiertagen**.

Service Deutschland Unser technologieorientiertes Competence Center in Lohr deckt alle Belange rund um den Service für elektrische Antriebe und Steuerungen ab.

Sie erreichen unsere **Service-Hotline** und unseren **Service-Helpdesk** unter:

Telefon: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
E-Mail: service.svc@boschrexroth.de
Internet: <http://www.boschrexroth.com>

Auf unseren Internetseiten finden Sie ergänzende Hinweise zu Service, Reparatur (z. B. Anlieferadressen) und Training.

Service weltweit Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.

Vorbereitung der Informationen Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- Eine detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Ihre Kontaktdaten (Telefon-, Faxnummer und E-Mail-Adresse)

Index

A

Abkürzungen.....	8
ANSI Z535.6-2006.....	6

B

Bibliotheken	
CAA_CiA405.library.....	45
Buskoppler.....	8

C

CAA_CiA405.library.....	45
CANbus Konfigurationseditor.....	15
CANopen Local Device.....	36
Allgemein.....	36
E/A-Abbild.....	38
Information.....	39
Parameter.....	38
Status.....	39
CANopen Manager	
Information.....	22
Status.....	22
CANopen Master.....	18
CANbus Parameter.....	18
Information.....	19
Status.....	18
CANopen Modul.....	30
E/A-Abbild.....	30
Information.....	31
Parameter.....	30
Status.....	30
CANopen Remote Device.....	23
E/A-Abbild.....	29
Information.....	30
Parameter.....	29
PDOs.....	25
SDOs.....	27
Status.....	29
CANopen Slave.....	34
CANbus Parameter.....	35
Information.....	35
Status.....	35
CANOPEN_KERNEL_ERROR.....	48
CANOPEN_KERNEL_ERROR_CODES.....	49
CIA405BASE.....	55

D

Darstellung von Informationen	
Begriffe und Abkürzungen.....	8
DEVICE.....	49
DEVICE_STATE.....	46

E

EDS Datei.....	8
EMCY_ERROR.....	48

G

Gefahrenhinweise.....	6
Gerätedatenbank	
Geräte entfernen.....	44
Geräte hinzufügen.....	42
Gerätedatenbank Dialog.....	41
Gerätedatenbank Übersicht.....	41
Gerätedatenbank verwalten.....	41
GET_CANOPEN_KERNEL_STATE.....	70
GET_LOCAL_NODE_ID.....	67
GET_STATE.....	68
Gliederung der Dokumentation.....	5

H

Helpdesk.....	79
Hotline.....	79

K

Klemme.....	8
Kommunikation und Feldbus	
Features des Feldbusses.....	9

M

Modul.....	8
------------	---

N

Nachinstallation	
CAN-Gerätebeschreibungen.....	11
NMT.....	71

R

RECV_EMCY.....	75
RECV_EMCY_DEV.....	73

S

SDO_ERROR.....	49
SDO_MODE.....	47
SDO_READ.....	66
SDO_READ_DATA.....	63
SDO_READ4.....	60
SDO_WRITE.....	64
SDO_WRITE_DATA.....	59
SDO_WRITE4.....	56
Service-Hotline.....	79
Sicherheitshinweise.....	6
Signalgrafik.....	6
Signalwörter.....	6
Support.....	79
Symbole.....	7

T

TRANSITION_STATE.....	48
-----------------------	----

Index

W

Warnhinweise..... 6

Z

Zu dieser Dokumentation..... 5
 Gültigkeit der Dokumentation..... 5

Notizen

Bosch Rexroth AG

Electric Drives and Controls

Postfach 13 57

97803 Lohr, Deutschland

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Deutschland

Tel. +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911344487