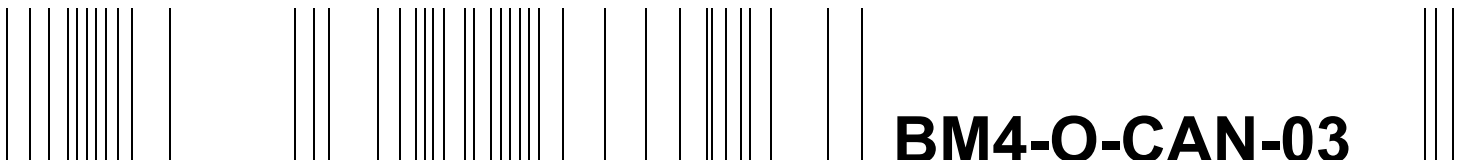


be in motion be in motion



BM4-O-CAN-03

**Optionsmodul CANopen-
Slave für b maXX PLC
Applikationshandbuch**

D

5.03057.02



Titel	Applikationshandbuch
Produkt	Optionsmodul CANopen-Slave für b maXX PLC
Stand	19. Juli 2005
Artikelnummer	376486
Copyright	<p>Dieses Applikationshandbuch darf vom Eigentümer ausschließlich für den internen Gebrauch in beliebiger Anzahl kopiert werden. Für andere Zwecke darf dieses Applikationshandbuch auch auszugsweise weder kopiert noch vervielfältigt werden.</p> <p>Verwertung und Mitteilung von Inhalten dieses Applikationshandbuches sind nicht gestattet.</p> <p>Bezeichnungen bzw. Unternehmenskennzeichen in diesem Applikationshandbuch können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.</p>
Verbindlichkeit	<p>Dieses Applikationshandbuch ist Teil des Gerätes/der Maschine. Dieses Applikationshandbuch muss jederzeit für den Bediener zugänglich und in einem leserlichen Zustand sein. Bei Verkauf/Verlagerung des Gerätes/der Maschine muss dieses Applikationshandbuch vom Besitzer zusammen mit dem Gerät/der Maschine weitergegeben werden. Nach Verkauf des Gerätes/der Maschine sind dieses Original und sämtliche Kopien an den Käufer zu übergeben. Nach Entsorgung oder anderem Nutzungsende sind dieses Original und sämtliche Kopien zu vernichten.</p> <p>Mit der Übergabe des vorliegenden Applikationshandbuches werden entsprechende Applikationshandbücher mit einem früheren Stand außer Kraft gesetzt.</p> <p>Bitte beachten Sie, dass Angaben/Zahlen/Informationen aktuelle Werte zum Druckdatum sind. Zur Ausmessung, Berechnung und Kalkulationen sind diese Angaben nicht rechtlich verbindlich.</p> <p>Die Firma Baumüller Nürnberg GmbH behält sich vor, im Rahmen der eigenen Weiterentwicklung der Produkte die technischen Daten und die Handhabung von Baumüller-Produkten zu ändern.</p> <p>Es kann jedoch keine Gewährleistung bezüglich der Fehlerfreiheit dieses Applikationshandbuches, soweit nicht in den Allgemeinen Verkaufs- und Lieferbedingungen anders beschrieben, übernommen werden.</p>
Hersteller	<p>Baumüller Nürnberg GmbH Ostendstr. 80 - 90 90482 Nürnberg Deutschland Tel. +49 9 11 54 32 - 0 Fax: +49 9 11 54 32 - 1 30 www.baumueller.de</p>



1	Einleitung	5
1.1	Erste Schritte	5
1.2	Verwendete Begriffe	5
1.3	Voraussetzungen	6
2	Grundlegende Sicherheitshinweise	7
2.1	Gefahrenhinweise und Gebote	7
2.1.1	Struktur eines Gefahrenhinweises	8
2.1.2	Verwendete Gefahrenhinweise	9
2.2	Infozeichen	11
2.3	Rechtliche Hinweise	11
2.4	Bestimmungsgemäße Verwendung	11
2.5	Sachwidrige Verwendung	12
2.6	Schutzeinrichtungen	12
2.7	Ausbildung des Personals	13
2.8	Sicherheitsmaßnahmen im Normalbetrieb	13
2.9	Verpflichtung und Haftung	14
2.9.1	Gefahrenhinweise und Sicherheitshinweise beachten	14
2.9.2	Gefahren im Umgang mit diesem Modul	14
2.9.3	Gewährleistung und Haftung	14
3	CANopen	15
3.1	Allgemeines zu CANopen und der Verwendung des Optionsmodul CANopen-Slave	15
3.2	Grundlagen zu CAN- und CANopen-Netzwerken	16
3.2.1	Grundlagen CAN	16
3.2.2	Grundlagen CANopen	17
3.3	Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave	21
3.3.1	Allgemeines zum Datenaustausch	21
3.3.2	Übersicht über den Datenaustausch	22
3.3.3	Durchzuführende Schritte	24
3.3.4	Anlegen eines PROPROG wt II Projekts	24
3.3.5	Gerätekonfiguration mit dem b maXX Konfigurator	27
3.3.6	Konfigurierung des Optionsmodul CANopen-Slave mit dem CANopen-Slave Konfigurator	31
3.3.7	Programmierung der Applikation im PROPROG wt II Projekt	44
3.4	Der Aufbau von CANopen-Telegrammen	76
3.4.1	Allgemeines	76
3.4.2	NMT-Telegramm	76
3.4.3	SDO-Telegramm	77
3.4.4	PDO-Telegramm	80
3.4.5	SYNC-Telegramm	81
3.4.6	Node Guarding Telegramm	82
3.4.7	Heartbeat-Telegramm	83
3.4.8	Emergency-Telegramm	84
Anhang A - Abkürzungen		85
Anhang B - Tabellen		87
B.1	Objekte nach DS 301	87
B.2	Objekte nach DS 405	101
B.3	Beispiel-EDS-Datei	104
B.4	Default Mapping der PDO	104
B.4.1	Default-Mapping für I/O-Module nach DS 401	104
B.4.2	Default-Mapping für Antriebe nach DSP 402	106
B.4.3	Default-Mapping für programmierbare Steuerungen nach DS 405	108



Inhaltsverzeichnis

B.5	Default-EDS-Datei	110
	Anhang C - Bibliothek CANop405_PLC01_20bd02	111
	Index	113

1

EINLEITUNG

Dieses Applikationshandbuch ist ein wichtiger Bestandteil Ihres b maXX 4400 Gerätes; lesen Sie daher nicht zuletzt im Interesse Ihrer eigenen Sicherheit diese Dokumentation komplett durch.

In diesem Kapitel beschreiben wir die ersten Schritte.

1.1 Erste Schritte

- 1 Um das Optionsmodul CANopen-Slave zu programmieren, benötigen Sie folgende Hardware:
Grundgerät b maXX 4400,
Optionsmodul b maXX PLC und
Optionsmodul CANopen-Slave.
Die Hardware muss entsprechend der jeweiligen Betriebsanleitung installiert und betriebsbereit sein.
Um ein funktionsfähiges CANopen-Netzwerk aufzubauen benötigen Sie weiterhin einen CANopen-Master sowie Kommunikationskabel und Abschlussstecker (Terminierstecker) und ggf. weitere CANopen-Slaves (z. B. weitere b maXX 4400 mit Optionsmodul CANopen-Slave oder auch I/O-Module).
- 2 Außerdem benötigen Sie folgende Software:
PROPROG wt II zur Programmierung der b maXX PLC und des Optionsmoduls CANopen-Slave für b maXX PLC und
b maXX Drive-Configurator für PROPROG wt II Version 1.0 (oder höher).

1.2 Verwendete Begriffe

Für das Baumüller-Produkt "BM4-O-CAN-03" (Optionsmodul CANopen-Slave für b maXX PLC) werden wir in dieser Dokumentation auch den Begriff "Steckmodul CANopen-Slave" verwenden. Weiterhin werden die Begriffe CANopen-Slave und Netzwerkknoten für das Optionsmodul CANopen-Slave für b maXX PLC verwendet.

Für die Baumüller-Produkte "BM4-O-ETH-02" (Optionsmodul Ethernet mit CANopen-Master für b maXX PLC) bzw. "BM4-O-CAN-04" (Optionsmodul CANopen-Master für b maXX PLC) werden wir in dieser Dokumentation auch die Begriffe "Steckmodul CANopen-Master" für BM4-O-ETH-02 oder BM4-O-CAN-04 verwenden.

Für das Produkt "Grundgerät b maXX 4400" wird auch der Begriff "b maXX" verwendet.

1.3 Voraussetzungen

Der Regler im Grundgerät wird auch "b maXX Regler" genannt.

Eine Liste der verwendeten Abkürzungen finden Sie in [▶Anhang A - Abkürzungen◀](#) auf Seite 85.

1.3 Voraussetzungen

Dieses Handbuch baut auf das "Applikationshandbuch b maXX PLC" auf und setzt die Kenntnis des Programmiertools PROPROG wt II und des zugehörigen Handbuchs voraus.

GRUNDLEGENDE SICHERHEITS- HINWEISE

Jedes Baumüller-Steckmodul haben wir nach strengen Sicherheitsvorgaben konstruiert und gefertigt. Trotzdem kann die Arbeit mit dem Steckmodul für Sie gefährlich sein.

In diesem Kapitel beschreiben wir Gefahren, die bei der Arbeit mit dem Baumüller-Steckmodul auftreten können. Gefahren verdeutlichen wir mit Symbolen (Icons). Alle in dieser Dokumentation verwendeten Symbole werden wir auflisten und erklären.

Wie Sie sich vor den einzelnen Gefahren im konkreten Fall schützen können, können wir in diesem Kapitel nicht erklären. In diesem Kapitel geben wir ausschließlich allgemeine Schutzmaßnahmen an. Die konkreten Schutzmaßnahmen werden wir in den nachfolgenden Kapiteln immer direkt nach dem Hinweis auf die Gefahr geben.

2.1 Gefahrenhinweise und Gebote



Gefahrenhinweise zeigen Ihnen Gefahren, die zu Verletzungen oder sogar zu Ihrem Tod führen können.

Beachten Sie immer die in dieser Dokumentation angegebenen Gefahrenhinweise.

Eine Gefahr teilen wir immer in eine der drei Gefahrenklassen ein. Jede Gefahrenklasse wird durch eines der folgenden Signalwörter gekennzeichnet:

GEFAHR (DANGER)

- erheblicher Sachschaden
- schwere Körperverletzung
- Tod - **wird** eintreffen

WARNUNG (WARNING)

- erheblicher Sachschaden
- schwere Körperverletzung
- Tod - **kann** eintreffen

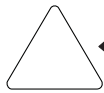
VORSICHT (CAUTION)

- Sachschaden
- leichte bis mittlere Körperverletzung - **kann** eintreffen

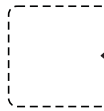
2.1 Gefahrenhinweise und Gebote

2.1.1 Struktur eines Gefahrenhinweises

Die nachfolgenden zwei Beispiele zeigen den prinzipiellen Aufbau eines Gefahrenhinweises. Ein Dreieck wird verwendet, wenn vor einer Gefahr für Lebewesen gewarnt wird. Fehlt das Dreieck, beziehen sich die Gefahrenhinweise ausschließlich auf Sachschäden.



Ein Dreieck zeigt, dass hier eine Gefahr für Lebewesen ist.
Die Farbe der Umrandung zeigt, wie groß die Gefahr ist - je dunkler die Farbe, desto größer ist die Gefahr.



Das Icon im Viereck stellt die Gefahr dar.
Die Farbe der Umrandung zeigt, wie groß die Gefahr ist - je dunkler die Farbe, desto größer ist die Gefahr.



Das Icon im Kreis stellt ein Gebot dar. Dieses Gebot muss der Anwender befolgen.
(Der Kreis ist gestrichelt dargestellt, weil nicht bei jedem Gefahrenhinweis ein Gebot als Icon vorhanden ist.)



Der Kreis zeigt, dass eine Gefahr für Sachschaden existiert.



Das Icon im Viereck stellt die Gefahr dar.
Die Farbe der Umrandung zeigt, wie groß die Gefahr ist - je dunkler die Farbe, desto größer ist die Gefahr. (Das Viereck ist gestrichelt dargestellt, weil nicht bei jedem Gefahrenhinweis die Gefahr als Icon dargestellt wird)

Der Text neben den Icons ist folgendermaßen aufgebaut:

HIER STEHT DAS SIGNALWORT, WELCHES DEN GRAD DER GEFAHR ANZEIGT




Hier schreiben wir, ob eine oder mehrere der untenstehenden Folgen eintreffen, wenn dieser Warnhinweis nicht beachtet wird.


- hier beschreiben wir die möglichen Folgen. Die schlimmste Folge steht ganz rechts.

Hier beschreiben wir die Gefahr.

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.


2.1.2 Verwendete Gefahrenhinweise

Steht vor einem Signalwort ein Gefahrzeichen:  oder  oder , dann bezieht sich der Sicherheitshinweis auf Personenschaden.

Steht vor einem Signalwort ein rundes Gefahrzeichen: , dann bezieht sich der Sicherheitshinweis auf Sachschaden.

2.1.2.1 Gefahrenhinweise vor Personenschaden

Zur optischen Unterscheidung verwenden wir für jede Klasse von Gefahrenhinweisen eine eigenen Umrandung für die dreieckigen Gefahrzeichen und die viereckigen Piktogramme.

Für die Gefahrenklasse **GEFAHR** (DANGER) verwenden wir das Gefahrzeichen . Folgende Gefahrenhinweise dieser Gefahrenklasse verwenden wir in dieser Dokumentation.

GEFAHR (DANGER)



Folgendes **wird eintreffen**, wenn Sie diesen Warnhinweis nicht beachten:

- schwere Körperverletzung
- Tod

*Die Gefahr ist: **Elektrizität**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.

GEFAHR (DANGER)




Folgendes **wird eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- schwere Körperverletzung
- Tod

*Die Gefahr ist: **mechanische Einwirkung**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.

Für die Gefahrenklasse **WARNUNG** (WARNING) verwenden wir das Gefahrzeichen . Folgende Gefahrenhinweise dieser Gefahrenklasse verwenden wir in dieser Dokumentation.

WARNUNG (WARNING)




Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- schwere Körperverletzung
- Tod

*Die Gefahr ist: **Elektrizität**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.

Für die Gefahrenklasse **VORSICHT** (CAUTION) verwenden wir das Gefahrzeichen . Folgende Gefahrenhinweise dieser Gefahrenklasse verwenden wir in dieser Dokumentation.

2.1 Gefahrenhinweise und Gebote



VORSICHT (CAUTION)

Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- leichte bis mittlere Körperverletzung

*Die Gefahr ist: **scharfe Kanten**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.



VORSICHT (CAUTION)

Folgendes **kann eintreffen**, wenn Sie diesen Warnhinweis nicht beachten:

- Umweltverschmutzung

*Die Gefahr ist: **unsachgemäße Entsorgung**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.

2.1.2.2 Gefahrenhinweise vor Sachschaden

Steht vor einem Signalwort ein rundes Gefahrzeichen: ⓘ dann bezieht sich der Sicherheitshinweis auf Sachschaden.



VORSICHT (CAUTION)

Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- Sachschaden

*Die Gefahr ist: **elektrostatische Entladung**. Hier wird die Gefahr gegebenenfalls genauer beschrieben.*

Hier beschreiben wir, was Sie tun können, um die Gefahr zu vermeiden.

2.1.2.3 Verwendete Gebotszeichen



Sicherheitshandschuhe tragen



Sicherheitsschuhe tragen

2.2 Infozeichen



HINWEIS

Dieser Hinweis ist eine besonders wichtige Information.

2.3 Rechtliche Hinweise

Diese Dokumentation wendet sich an technisch qualifiziertes Personal, welches speziell ausgebildet ist und gründlich mit allen Warnungen und Instandhaltungsmaßnahmen vertraut ist.

Die Geräte sind nach dem Stand der Technik gefertigt und betriebssicher. Sie lassen sich gefahrlos installieren und in Betrieb setzen und funktionieren problemlos, wenn sichergestellt ist, dass die Hinweise der Dokumentation beachtet werden.

Der Benutzer trägt die Verantwortung für die Durchführung von Service und Inbetriebnahme gemäss den Sicherheitsvorschriften der geltenden Normen und allen anderen relevanten staatlichen oder örtlichen Vorschriften betreffend Leiterdimensionierung und Schutz, Erdung, Trennschalter, Überstromschutz usw.

Für Schäden, die bei der Montage oder beim Anschluss entstehen, haftet der Benutzer.

2.4 Bestimmungsgemäße Verwendung

Sie müssen das Steckmodul immer bestimmungsgemäß verwenden. Untenstehend haben wir einige wichtige Hinweise für Sie zusammengestellt. Die untenstehenden Hinweise sollen Ihnen ein Gefühl für die bestimmungsgemäße Verwendung des Steckmoduls geben. Mit den untenstehenden Hinweisen erheben wir keinen Anspruch auf Vollständigkeit - beachten Sie alle in dieser Betriebsanleitung gegebenen Hinweise.

- Sie dürfen das Steckmodul nur in Geräte der Reihe b maXX 4400 einbauen.
- Projektieren Sie die Anwendung so, dass Sie das Steckmodul immer innerhalb seiner Spezifikationen betreiben.
- Sorgen Sie dafür, dass ausschließlich qualifiziertes Personal mit diesem Steckmodul arbeitet.
- Montieren Sie das Steckmodul nur an dem/den vorgegebenen Steckplatz/Steckplätzen.
- Installieren Sie das Steckmodul so wie in es in dieser Dokumentation vorgegeben ist.
- Sorgen Sie dafür, dass die Anschlüsse immer den vorgegebenen Spezifikationen entsprechen.
- Betreiben Sie das Steckmodul nur, wenn es technisch einwandfrei ist.
- Betreiben Sie das Steckmodul immer in einer Umgebung, wie sie in den „Technischen Daten“ vorgeschrieben ist.
- Betreiben Sie das Steckmodul immer in serienmäßigem Zustand.
Aus Sicherheitsgründen dürfen Sie das Steckmodul nicht umbauen.

- Beachten Sie alle diesbezüglichen Hinweise, falls Sie das Steckmodul lagern.

Sie verwenden das Steckmodul dann bestimmungsgemäß, wenn Sie alle Hinweise und Informationen dieser Betriebsanleitung beachten.

2.5 Sachwidrige Verwendung

Im Folgenden listen wir einige Beispiele sachwidriger Verwendung auf. Die untenstehenden Hinweise sollen Ihnen ein Gefühl dafür geben, was eine sachwidrige Verwendung des Steckmoduls ist. Wir können aber nicht alle erdenklichen sachwidrigen Verwendungen hier auflisten. Alle Verwendungen, bei denen die Hinweise dieser Dokumentation missachtet werden, sind sachwidrig und somit verboten, insbesondere in folgenden Fällen:

- Sie haben das Steckmodul in andere Geräte als die Reihe b maXX 4400 eingebaut.
- Sie haben Hinweise dieser Betriebsanleitung missachtet.
- Sie haben das Steckmodul nicht bestimmungsgemäß verwendet.
- Sie haben das Steckmodul
 - unsachgemäß montiert,
 - unsachgemäß angeschlossen,
 - unsachgemäß in Betrieb genommen,
 - unsachgemäß bedient,
 - von nicht bzw. nicht ausreichend qualifiziertem Personal montieren, anschließen, in Betrieb nehmen und betreiben lassen,
 - überlastet,
- betrieben
 - mit defekten Sicherheitseinrichtungen,
 - mit nicht ordnungsgemäß angebrachten bzw. ohne Sicherheitsvorrichtungen,
 - mit nicht funktionsfähigen Sicherheits- und Schutzvorrichtungen
 - außerhalb der vorgeschriebenen Umgebungsbedingungen
- Sie haben das Steckmodul umgebaut, ohne dass dies schriftlich von der Baumüller Nürnberg GmbH genehmigt wurde.
- Sie haben die Anweisungen bezüglich Wartung in den Komponentenbeschreibungen nicht beachtet.
- Sie haben das Steckmodul unsachgemäß mit Produkten anderer Hersteller kombiniert.
- Sie haben das Antriebssystem mit fehlerhaften und/oder fehlerhaft dokumentierten Produkten anderer Hersteller kombiniert.
- Ihre selbsterstellte Software der PLC enthält Programmierfehler, die zu einer Fehlfunktion führen.

Die „Allgemeinen Verkaufs- und Lieferbedingungen“ Version 1.1 vom 15.02.2002 bzw. die jeweils neueste Version der Firma Baumüller Nürnberg GmbH gelten grundsätzlich. Diese stehen Ihnen spätestens seit Vertragsabschluss zur Verfügung.

2.6 Schutzeinrichtungen

Während des Transports werden die Steckmodule durch ihre Verpackung geschützt. Entnehmen Sie das Steckmodul erst unmittelbar vor der Montage der Transportverpackung.

Die Abdeckhaube des Reglerteils der b maXX Geräte schützt in Schutzklasse IP20 die Steckmodule vor Verschmutzung und Schäden durch statische Entladungen bei Berührungen. Stecken Sie daher nach erfolgter Montage des Steckmoduls die Abdeckhaube wieder auf.

2.7 Ausbildung des Personals



WARNUNG (WARNING)

Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- schwere Körperverletzung
- Tod

Geräte der Firma Baumüller Nürnberg GmbH dürfen ausschließlich von qualifiziertem Personal montiert, installiert, betrieben und gewartet werden.

Qualifiziertes Personal (Fachkräfte) wird folgendermaßen definiert:

Qualifiziertes Personal

Von der Firma Baumüller Nürnberg GmbH autorisierte Elektro-Ingenieure und Elektro-Fachkräfte des Kunden oder Dritter, die Installation und Inbetriebnahme von Baumüller-Antriebssystemen erlernt haben und berechtigt sind, Stromkreise und Geräte gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Qualifiziertes Personal verfügt über eine Ausbildung oder Unterweisung gemäß den örtlich jeweils gültigen Standards der Sicherheitstechnik in Pflege und Gebrauch angemessener Sicherheitsausrüstung.

Anforderungen an das Bedienungs-personal

Die Bedienung des Antriebssystems darf nur von Personen durchgeführt werden, die dafür ausgebildet, eingewiesen und befugt sind.

Störungsbeseitigung, Instandhaltung, Reinigung, Wartung und Austausch dürfen nur durch geschultes oder eingewiesenes Personal durchgeführt werden. Diese Personen müssen die Betriebsanleitung kennen und danach handeln.

Inbetriebnahme und Einweisung dürfen nur vom qualifizierten Personal durchgeführt werden.

2.8 Sicherheitsmaßnahmen im Normalbetrieb

- ▶ Beachten Sie am Aufstellort des Gerätes die gültigen Sicherheitsbestimmungen für die Anlage, in die dieses Gerät eingebaut ist.
- ▶ Versehen Sie das Gerät mit zusätzlichen Überwachungs- und Schutzeinrichtungen, falls Sicherheitsbestimmungen dies fordern.
- ▶ Beachten Sie die Sicherheitsmaßnahmen für das Gerät, in das das Steckmodul eingebaut ist.

2.9 Verpflichtung und Haftung

Damit Sie sicherheitsgerecht mit diesen Optionsmodulen arbeiten können, müssen Sie die Gefahrenhinweise und Sicherheitshinweise dieser Dokumentation kennen und beachten.

2.9.1 Gefahrenhinweise und Sicherheitshinweise beachten

Wir verwenden in dieser Betriebsanleitung optisch einheitliche Sicherheitshinweise, die sie vor Personen- und Sachschäden bewahren sollen.



WARNUNG (WARNING)

Folgendes **kann eintreffen**, wenn Sie diesen Gefahrenhinweis nicht beachten:

- schwere Körperverletzung
- Tod

Alle Personen, die an und mit Geräten der Reihe b maXX arbeiten, müssen bei ihren Arbeiten die Betriebsanleitung verfügbar haben und die hierin enthaltenen Anweisungen und Hinweise - insbesondere die Sicherheitshinweise - beachten.

Außerdem müssen alle Personen, die an diesem Gerät arbeiten, zusätzlich alle Regeln und Vorschriften, die am Einsatzort gelten, kennen und beachten.

2.9.2 Gefahren im Umgang mit diesem Modul

Das Optionsmodul CANopen-Slave wurde nach dem Stand der Technik und unter Einhaltung der geltenden Richtlinien und Normen entwickelt und gefertigt. Dennoch können bei der Verwendung Gefahren entstehen. Eine Übersicht möglicher Gefahren finden Sie im Kapitel [>Grundlegende Sicherheitshinweise<](#) ab Seite 7 .

Weiterhin warnen wir Sie vor der akuten Gefahr an der entsprechenden Stelle in dieser Dokumentation.

2.9.3 Gewährleistung und Haftung

Alle Angaben in dieser Dokumentation sind unverbindliche Kundeninformationen, unterliegen einer ständigen Weiterentwicklung und werden laufend durch unseren permanenten Änderungsdienst aktualisiert.

Gewährleistungs- und Haftungsansprüche gegen die Firma Baumüller Nürnberg GmbH sind ausgeschlossen, wenn insbesondere eine oder mehrere der von uns in [>Sachwidrige Verwendung<](#) ab Seite 12 oder unten aufgeführten Ursachen den Schaden bewirkt hat/haben:

- Eintritt eines Katastrophenfalls durch Fremdkörperwirkung bzw. höhere Gewalt

CANOPEN

In diesem Kapitel finden Sie Informationen zum Datenaustausch über CANopen.

Als CANopen-Slave ist das Optionsmodul BM4-O-CAN-03 verfügbar.

Als CANopen-Master sind die Optionsmodule BM4-O-ETH-02 und BM4-O-CAN-04 verfügbar.

3.1 Allgemeines zu CANopen und der Verwendung des Optionsmodul CANopen-Slave

CANopen ist eine weit verbreitete Feldbus-Anwendungsschicht, die auf der Basis des Bussystems Controller Area Network (CAN) aufsetzt und von der internationalen CAN-Organisation CAN in Automation e.V. (CiA) veröffentlicht wurde.

Die CANopen Mechanismen und Funktionalitäten werden durch unterschiedliche Profile beschrieben. Das Kommunikationsprofil definiert die Art und Weise des Datenaustausches und allgemeine, für alle CANopen Geräte geltende Festlegungen. In den Geräteprofilen werden die Anwendungs- und Geräte spezifischen Festlegungen, die inhaltliche Bedeutung der Daten und die Gerätefunktionalität beschrieben. Geräteprofile existieren u. a. für Antriebe, I/O-Module, Geber und programmierbare Geräte.

Mit dem Optionsmodul CANopen-Slave für die b maXX PLC können Sie zusammen mit den Funktionsbausteinen der Bibliothek CANop405_PLC01_20bd02 (oder höher) für die Programmieroberfläche PROPROG wt II einen Datenaustausch mit Slave-Funktionalitäten nach Kommunikationsprofil DS 301 umsetzen.

Die Programmierung erfolgt mit den Funktionsbausteinen aus der Bibliothek CANop405_PLC01_20bd02 (oder höher) unter PROPROG wt II. Die Funktionsbausteine sind nach dem Geräteprofil DS 405 umgesetzt. Damit können CANopen- und CANopen-Slave-Funktionen wie

- Prozessdatenaustausch über Prozess Daten Objekte (PDOs) mit hochprioren Identifizieren
- Bedarfsdatenaustausch über Service-Daten-Objekte (SDOs) mit niederprioren Identifizieren
- Synchronisierung (SYNC) des Echtzeitdatenaustausches mit PDOs und SYNC-Überwachung
- Fehler (EMERGENCY): Emergency-Sendeauftrag (Slave --> Master) setzen
- Netzwerküberwachung (NODE GUARDING / HEARTBEAT): für die Ausfallüberwachung von Netzwerkknoten

einfach umgesetzt werden.

Weiterhin werden Netzwerkmanagementkommandos (NMT-Kommandos) wie z. B. Start, Stop und ResetCommunication ausgeführt.

Für den CANopen-Slave stehen die fünf Übertragungsraten 1 MBit/s, 500 kBit/s, 250 kBit/s, 125 kBit/s, und 20 kBit/s zur Verfügung. Das Optionsmodul CANopen-Master kann mit bis zu 32 CANopen-Netzwerkknoten (z. B. I/O-Modulen und Antrieben) kommunizieren.

3.2 Grundlagen zu CAN- und CANopen-Netzwerken

3.2.1 Grundlagen CAN

Ein Feldbussystem auf der Basis des CAN wird in Linienstruktur ausgeführt. Als physikalische Basis der Datenübertragung dient eine Dreidrahtleitung mit den Anschlüssen CAN_High, CAN_Low und CAN_Ground. CAN verwendet eine erdsymmetrische Übertragung, um Gleichtaktstörungen zu unterdrücken. Daher werden Differenzsignale ausgewertet.

Netzwerk

CAN ist ein Multi-Master-Netzwerk. Jeder Teilnehmer kann gleichberechtigt und aktiv auf den Bus zugreifen. CAN verwendet die objektorientierte Adressierung, d. h. die übermittelte Nachricht wird mit einem netzwerkweit festgelegten Identifier gekennzeichnet. Er stellt den codierten Namen der Nachricht dar.

Buszugriff

Der Buszugriff erfolgt über das CSMA/CA-Verfahren (Carrier Sense Multiple Access / Collision Avoidance). Da jeder Teilnehmer das Recht hat, nach Erkennung der notwendigen Busruhe, mit dem Senden seiner Nachricht zu beginnen, können Kollisionen entstehen. Dieses wird durch die bitweise Arbitrierung der zu sendenden Nachrichten vermieden. Dabei werden zwei Buspegel unterschieden, ein dominanter Pegel, logischer Bitwert 0, und ein rezessiver Pegel, logischer Bitwert 1. Im schlimmsten Fall beginnen alle sendewilligen Teilnehmer gleichzeitig mit dem Versenden ihrer Nachricht auf dem Bus. Wird ein rezessives Bit eines Teilnehmers von einem dominanten Bit eines anderen überschrieben, so zieht sich der "rezessive" Knoten vom Bus zurück und versucht nach Erkennung der Busruhe erneut seine Nachricht abzusetzen. Somit ist gewährleistet, dass die wichtigste, höchstpriorie Nachricht (mit dem niedrigsten Identifier) kollisionsfrei und ohne Verzögerung übertragen wird. Aus diesem Grund ist es natürlich notwendig, dass jeder Identifier nur einmal am CAN-Bus vergeben sein darf.

Identifier

Es stehen nach Spezifikation CAN 2.0A 2032 unterschiedliche Identifier zur Verfügung. Jeder Teilnehmer kann unaufgefordert senden (Multi-Master-Fähigkeit). Ein Sender übermittelt seine Nachricht an alle CAN-Knoten (Broadcast), die anhand des Identifiers selbst entscheiden, ob sie die Nachricht weiterverarbeiten oder nicht.

Fehler

In einem CAN-Datentelegramm können bis zu acht Byte Nutzdaten übertragen werden. Zur Fehler- oder Überlastsignalisierung kann ein CAN-Knoten Error- oder Overload-Telegramme senden. Dieses geschieht auf Schicht 2 des OSI/ISO-Referenzmodells, dem Data Link Layer, also unabhängig von der Applikation. Aufgrund einer hochwertigen Fehlererkennung und -behandlung auf Schicht 2 wird eine Hamming-Distanz (Maß der Fehlererkennung) von $HD = 6$ erreicht, d. h. maximal fünf gleichzeitig auftretende Bitfehler innerhalb eines Telegramms werden sicher als Fehler erkannt.

3.2.2 Grundlagen CANopen

CANopen ist ein offenes und damit Hersteller neutrales Feldbussystem, welches auf den Layer 1 und 2 - Definitionen des CAN-Standards aufsetzt

3.2.2.1 CAL-Spezifikation und Profile

In der Internationalen CAN-Organisation CAN in Automation e.V. (CiA) entstand mit der Anwendungsschicht CAL eine allgemeine Beschreibungssprache für CAN Netzwerke. CAL stellt eine Sammlung von Kommunikationsdiensten zur Verfügung ohne deren Anwendung genau festzuschreiben. CANopen nutzt diese Beschreibungssprache. Zum einen wird eine Teilmenge der von CAL angebotenen Kommunikationsdienste zur Definition einer offenen Kommunikationsschnittstelle verwendet. Diese Definitionen sind in den Kommunikationsprofilen festgehalten, wobei hier insbesondere das Profil DS 301 von Bedeutung ist, in welchem das ‚wie‘ der Kommunikation geregelt ist. Zum anderen legt CANopen zusätzlich fest, was die Daten in den jeweiligen Geräteklassen bedeuten.

Diese Definitionen sind in den Geräteprofilen festgehalten, für I/Os ist das z. B. das Profil DS 401 und für Antriebe das Profil DSP 402. Objekte in nach IEC 61131-3 programmierbaren Geräten werden durch das Geräteprofil DS 405 definiert.

Das Optionsmodul CANopen-Slave unterstützt zusammen mit den Funktionsbausteinen aus der Bibliothek CANop405_PLC01_20bd02 (und höher) für PROPROG wt II das Kommunikationsprofil DS 301 und das Geräteprofil DS 405.

3.2.2.2 Kommunikations- und Gerätespezifische Objekte

Bei einem Zugriff über das Netzwerk wird grundsätzlich auf die Objekte eines Netzwerkknotens zugegriffen. Objekte übernehmen verschiedene Aufgaben. Durch Kommunikationsprofil spezifische Objekte wird der Datenaustausch über den CANopen-Feldbus konfiguriert. Als Geräteprofil spezifische Objekte stehen sie in direkter Verbindung zum Gerät. Über sie können die Gerätefunktionen genutzt und verändert werden. Objekte werden immer über einen Index (16 Bit) und einen Subindex (8 Bit) adressiert. Abhängig von der Bedeutung sind verschiedene Bereiche für Objekte festgelegt. Die wichtigsten Anwenderbereiche sind:

Index (hex)	Objekt
1000 - 1FFF	Bereich für das Kommunikationsprofil. Enthält alle Objekte, die für die Kommunikation notwendig sind, z. B. PDO, SDO.
2000 - 5FFF	Bereich für Hersteller spezifische Objekte. Enthält alle Objekte, die nicht innerhalb eines Profiles definiert sind und Hersteller abhängig sind.
6000 - 9FFF	Bereich für das Geräteprofil. Enthält die Objekte des jeweils unterstützten Geräteprofils (DS 401, DSP402, etc.)

In den einzelnen Profilen ist genau festgelegt, welche Objekte vorhanden sein müssen und welche für einen Netzwerkknoten optional sind.

3.2.2.3 Datenaustausch und Objekte des physikalischen Bussystems

Aus Sicht der Kommunikationsbeziehungen zwischen den CANopen-Teilnehmern wird der Datenaustausch beschrieben durch das

- Client - Server Modell: jeder ist berechtigt von einem anderen Teilnehmer Daten anzufordern oder auf diesen zu übertragen

und das

- Producer - Consumer Modell: ein Teilnehmer sendet Daten ohne Anforderung, andere hören mit und können die Daten auswerten.

Eine Master - Slave Beziehung (genau ein CANopen-Teilnehmer organisiert den Datenaustausch) existiert auf Ebene der Kommunikationsbeziehungen nicht.

Unter einem CANopen-Slave versteht man daher nicht einen Slave auf Ebene der Kommunikation sondern auf Ebene der Anwendung. Unter CANopen-Slave Funktionen sind daher hauptsächlich Aufgaben wie Zustandsabfragen sowie Anforderungsfeststellung und- bearbeitung zu sehen.

Der Datenaustausch in CANopen-Netzwerken erfolgt in Form von Telegrammen, mit denen die Nutzdaten übertragen werden bzw. mit denen auf die Objekte eines Netzwerkknotens zugegriffen wird. Jedes Telegramm ist durch einen Identifier gekennzeichnet. Die Telegrammtypen und die dahinter stehenden Mechanismen für den Datenaustausch sind in Gruppen eingeteilt:

- Prozess-Daten-Objekte (PDOs): Echtzeitdatenaustausch mit hochprioren Identifiern und bis zu 8 Byte pro Nachricht. Das Optionsmodul CANopen-Slave kann maximal 4 PDOs empfangen und 4 PDOs senden.
- Service-Daten-Objekte (SDOs): Parameterdatenaustausch mit niederprioren Identifiern und durch Index/Subindex adressierbare Daten. Der Transfertyp beim Optionsmodul CANopen-Slave ist "expedited", d. h. es können bis zu 4 Byte pro Nachricht übertragen werden. Das Optionsmodul CANopen-Slave kann gleichzeitig maximal 1 SDOs empfangen oder senden.
- Netzwerkmanagement (NMT): Spezialobjektart für die Umsetzung von Netzwerkkommunikations-Funktionen wie z. B. Start, Stop und ResetCommunication von Netzwerkknoten. Das Optionsmodul CANopen-Slave setzt die von einem CANopen-Master gesendeten Kommandos um.
- Synchronisierung (SYNC): Spezialobjektart für die Synchronisierung des Echtzeitdatenaustausches mit PDOs. Das Optionsmodul CANopen-Slave synchronisiert seinen

Datenaustausch (über PDO) auf das von einem CANopen-Master gesendete SYNC-Kommando.

- Fehlerbehandlung (EMERGENCY): Spezialobjektart für das Erkennen von Fehlern eines Netzwerkknoten. Das Optionsmodul CANopen-Slave sendet bei einem Fehler ein Emergency-Telegramm.
- Netzwerküberwachung (NODE GUARDING): Spezialobjektart für die Ausfallüberwachung von Netzwerkknoten. Das Optionsmodul CANopen-Slave sendet, auf Anforderung durch den CANopen-Master, zyklisch ein Telegramm. Weiterhin überwacht es die Anforderung dieses Telegramms und erkennt so den Ausfall des NMT-Masters (das ist der CANopen-Master).



HINWEIS

In den Profildefinitionen der CiA wird der Begriff "Objekt" auf zwei unterschiedliche Arten verwendet. Zum einen wie in Kapitel [►Kommunikations- und Gerätespezifische Objekte◄](#) auf Seite 17, als eine Art Datum auf das von außen zugegriffen werden kann oder Geräteeigenschaften beschreibt. Zum anderen werden damit auch die unterschiedlichen Telegrammtypen und die dahinter stehenden Mechanismen für den Datenaustausch bezeichnet. Im zweiten Fall können diese Objekte als Objekte des physikalischen CANopen-Bussystems gesehen werden, jedoch nicht als Objekte auf die über Indizes und Subindizes zugegriffen werden kann. Diese Objekte des physikalischen CANopen-Bussystems transportieren die Kommunikations- und Geräte spezifischen Objekte.

Aus diesen Festlegungen ergibt sich auch, dass im Gegensatz zu anderen Feldbussystemen, die Telegramme selbst von den einzelnen Teilnehmern identifiziert werden. Jeder Netzwerkknoten entscheidet selbst, wann er Daten senden möchte und entscheidet auch selbst, welche Telegramme er auswertet. Es gibt allerdings auch die Möglichkeit über sogenannte Remote Telegramme, andere Netzwerkknoten zum Senden von Daten aufzufordern.

3.2.2.4 Vordefinierte Identifier

Um direkt nach einem Boot-Up eine Peer-to-Peer Kommunikation zwischen Master und den übrigen Netzwerkknoten (Slaves) aufbauen zu können, existiert für die Telegramme eine vordefinierte Identifizierung. Diese Identifizierung kann vom Anwender umkonfiguriert werden. Ein Identifier (nach CAN-Definition) teilt sich in Function-Code und Module-ID auf:

Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Function-Code				Module-ID						

Die Module-ID ist gleichbedeutend mit der Knotennummer im Netzwerk. Aus den sieben Bit für die Module-ID ergibt sich je CANopen-Netzwerk eine theoretische maximale Anzahl von 127 Knoten. Aus physikalischen Gründen können mit dem Optionsmodul CANopen-Slave maximal 32 Knoten in einem Netzwerk betrieben werden. Aus den insgesamt 11 Bit ergibt sich die COB-ID (Communication Object Identifier). Folgende Objekte sind vordefiniert:

Broadcast-Objekte (Module-ID = 0):

Objektname	Function Code binär	resultierende COB-ID dezimal	resultierende COB-ID hexadezimal
NMT	0000	0	16#0
SYNC	0001	128	16#80
TIME STAMP ^{a)}	0010	256	16#100

a) Das Time Stamp Objekt wird vom Optionsmodul CANopen-Master nicht unterstützt.

Peer-to-Peer-Objekte:

Objektname	Function Code binär	resultierende COB-ID dezimal	resultierende COB-ID hexadezimal
EMERGENCY	0001	129 - 255	81h - FFh
PDO1 (TX)	0011	385 - 511	181h - 1FFh
PDO1 (RX)	0100	513 - 639	201h - 27Fh
PDO2 (TX)	0101	641 - 767	281h - 2FFh
PDO2 (RX)	0110	769 - 895	301h - 37Fh
PDO3 (TX)	0111	897 - 1023	381h - 3FFh
PDO3 (RX)	1000	1025 - 1151	401h - 47Fh
PDO4 (TX)	1001	1153 - 1279	481h - 4FFh
PDO4 (RX)	1010	1281 - 1407	501h - 57Fh
SDO (TX)	1011	1409 - 1535	581h - 5FFh
SDO (RX)	1100	1537 - 1663	601h - 67Fh
NODE GUARDING	1110	1793 - 1919	701h - 77Fh

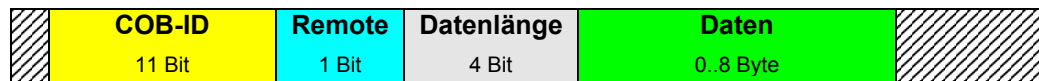
TX steht für transmit (senden) und RX für receive (empfangen). Beachten Sie, dass die Übertragungsrichtung von den Netzwerkknoten und nicht vom Master aus betrachtet wird. D. h. wird vom CANopen-Master z. B. ein PDO gesendet ist dies für das Optionsmodul CANopen-Slave ein RxPDO und muss also mit einem RX-Identifizier auf dem CANopen-Bus übertragen werden. Gleiches gilt für die Module-ID, auch diese ist vom Optionsmodul CANopen-Slave und nicht vom CANopen-Master aus zu betrachten, d. h. die COB-ID enthält die Module-ID des CANopen-Slave und nicht des CANopen-Masters.

Diese Gegebenheiten gelten jedoch nur für die vordefinierte Identifizierung. Nach dem Kommunikationsprofil DS 301 können die COB-IDs für die Bus physikalischen Kommunikationsobjekte vom Anwender für jeden Netzwerkknoten über Geräte Objekte frei festgelegt werden. Somit kann z. B. auch eine Prozessdaten-Kommunikation direkt zwischen zwei (oder mehreren) Netzwerkknoten eingerichtet werden (z. B. zwischen zwei Geräten b maXX 4400 mit b maXX PLC und Optionsmodul CANopen-Slave).

3.2.2.5 Telegrammaufbau und Priorität

Im CAN-Telegramm sind die wichtigsten CANopen-Elemente die COB-ID, das Remote-Bit, die Datenlänge und die Daten. Da die COB-ID dem CAN-Identifizierer entspricht, ergibt sich auf Grund des CAN-Übertragungsverfahrens CSMA/CA (siehe Kapitel [► Grundlagen CAN](#) auf Seite 16), dass niederwertige COB-IDs eine höhere Priorität haben. Aus der vordefinierten Identifizierergabe ergibt sich somit, dass NMT-Telegramme die höchste Priorität haben, gefolgt von SYNC-Telegrammen. Da die COB-ID zudem die Netzwerk-knotennummer enthält, ergibt sich damit auch, dass Telegramme von Netzwerkknoten mit niedriger Knotennummer höhere Priorität haben.

Die wichtigsten Elemente im CANopen-Telegramm sind:



Ist das Remote-Bit gesetzt, werden Daten vom anderen Netzwerkknoten angefordert. Für Daten stehen bis zu 8 Byte zur Verfügung. Die Anzahl der Byte ist jedoch vom Telegrammtyp abhängig. Bei SDO sind dies maximal 4 Byte, die restlichen 4 Byte werden zur Identifizierung der Datenart benötigt. PDOs können alle 8 Byte verwenden. SYNC Telegramme haben keinen Dateninhalt.

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

3.3.1 Allgemeines zum Datenaustausch

In den nachfolgenden Kapiteln wird Ihnen erläutert wie ein Optionsmodul CANopen-Slave in einem CANopen Netzwerk zu verwenden ist.

Es werden Ihnen sowohl die grundlegenden Mechanismen wie sie im Profil DS 301 definiert sind aufgezeigt, als auch die Verwendung der Funktionsbausteine aus der Bibliothek CANop405_PLC01_20bd02 (oder höher) nach Profil DS 405. Die Darstellungen werden durch ein sich Schritt für Schritt aufbauendes Beispiel-Projekt für PROPROG wt II begleitet.

Für die Inbetriebnahme des CANopen-Masters sehen Sie bitte in der zugehörigen Betriebsanleitung nach.



HINWEIS

Das entstehende Beispiel-Projekt ist keinesfalls als voll funktionsfähige Applikation anzusehen. Die Daten, welche über die Beispiele vom CANopen-Slave auf die b maXX PLC und gegebenenfalls weiter auf den b maXX Regler geschrieben werden, haben vom Wert her keine funktionale Bedeutung. Stellen Sie daher unbedingt sicher, dass der Antrieb (Motor) nicht im Betriebsbereiten Zustand ist!

Natürlich bieten sich mit dem Optionsmodul CANopen-Slave und den Funktionsbausteinen weitaus mehr Möglichkeiten, als im Rahmen dieses Handbuchs dargestellt werden können. Es kann auch nicht auf alle Einzelheiten der Funktionsbausteine eingegangen werden. Entnehmen Sie Details zu den Funktionsbausteinen bitte der zugehörigen Online-Hilfe in PROPROG wt II.

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

3.3.2 Übersicht über den Datenaustausch

Der Zugriff auf die Daten eines CANopen-Slaves erfolgt immer über die Objekte. Im Optionsmodul CANopen-Slave für b maXX PLC erfolgt der Zugriff auf die Daten über die profilspezifischen Objekte aus dem CiA Geräteprofil DS 405. Dies sind die Objekte 0xA000 bis 0xA8FF.

Wie der jeweilige Datenzugriff erfolgt und weitere allgemeine Angaben, sind in den kommunikationsspezifischen Objekten festgelegt. Dies sind die Objekte 0x1000 bis 0x1FFF.

Obwohl der b maXX 4400 mit b maXX PLC und Optionsmodul CANopen-Slave ein Antrieb (Drive) ist, wird als CiA Geräteprofil nicht DSP 402 (Drives) sondern DS 405 (IEC 61131-3 Programmable Devices) unterstützt.

Begründung: Der CANopen-Master greift nicht direkt auf den Antrieb (Drive) zu. Der CANopen-Master greift auf die Daten der b maXX PLC (die nach IEC 61131-3 programmiert wird) zu.

Der b maXX 4400 erhält als CANopen-Slave die Daten vom CANopen-Master. Im b maXX 4400 nimmt das Optionsmodul CANopen-Slave die Daten in Empfang und gibt sie an die b maXX PLC weiter. In der b maXX PLC können die Daten weiterverarbeitet werden (z. B. mit einer elektronischen Kurvenscheibe) und werden dann an den b maXX Regler übergeben. In diesem erfolgt dann die eigentliche Regelung und Steuerung des Antriebs (Motor) über das b maXX Leistungsteil.

Im umgekehrten Weg können die Daten zum CANopen-Master übermittelt werden.

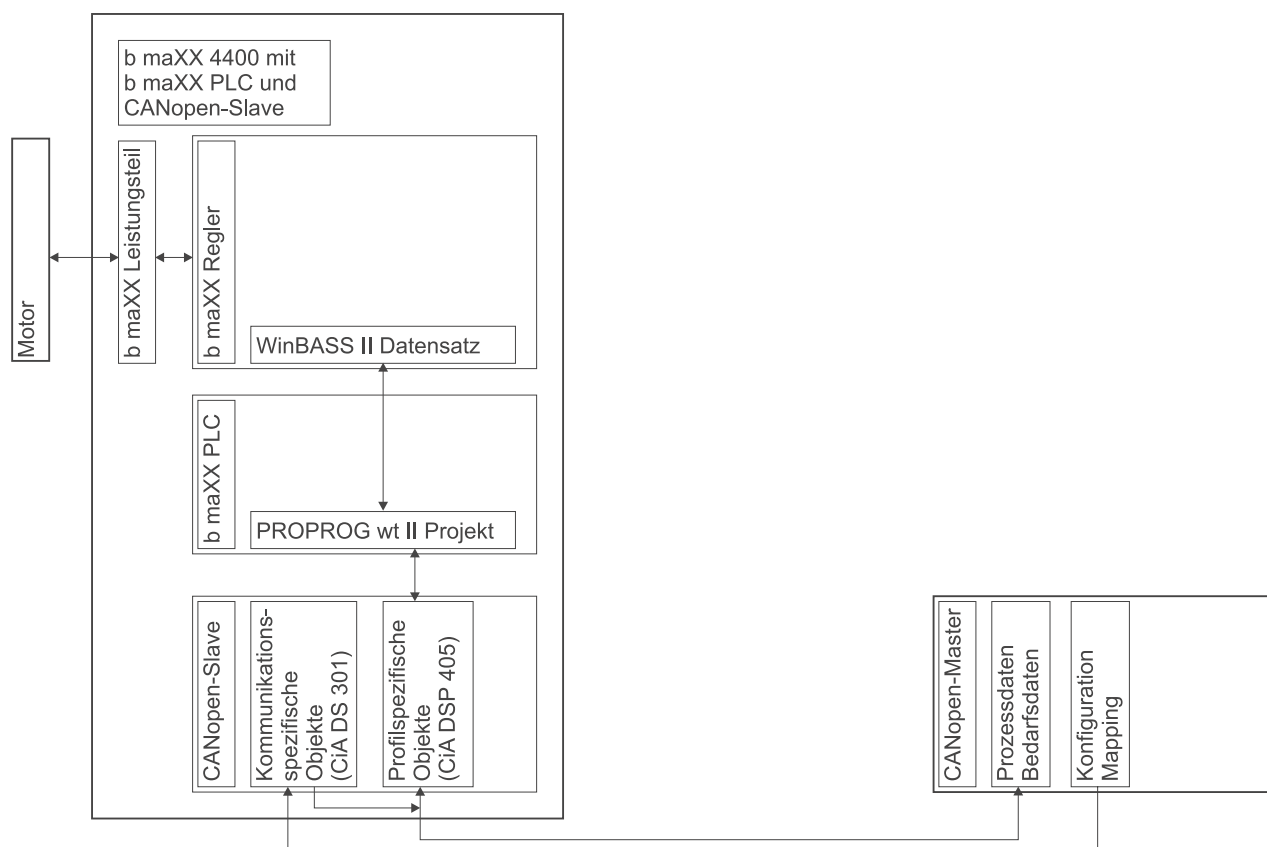


Abbildung 1: Datenfluss CANopen-Master → CANopen-Slave → b maXX PLC → b maXX Regler → Leistungsteil → Motor und zurück

Kommunikation CANopen-Master → CANopen-Slave

Der CANopen-Master kommuniziert mit dem CANopen-Slave über die Prozessdatenobjekte (PDOs) und Bedarfsdatenobjekte (SDOs). Der Inhalt von PDOs und SDOs sind die profilspezifischen Objekte (nach Profil DS 405).

CANopen-Slave

Auf dem CANopen-Slave erfolgt die Umsetzung der PDOs und SDOs auf die entsprechenden Objekte im CANopen-Slave. Hierzu wird auf dem CANopen-Slave während der Initialisierung ein Mapping hinterlegt. Dieses Mapping wird durch den CANopen-Slave-Konfigurator im b maXX Drive-Konfigurator während des Anlegen von Variablen erstellt.

Kommunikation CANopen-Slave → b maXX PLC

Diese profilspezifischen Objekte werden über den CANopen-Slave Konfigurator (im b maXX Drive-Konfigurator; siehe [►Anlegen von Variablen◄](#) ab Seite 34) den Variablen im PROPROG wt II Projekt auf der b maXX PLC zugeordnet.

b maXX PLC

Auf der b maXX PLC erfolgt i. d. R. eine Weiterverarbeitung der Daten (in den Variablen) z. B. mittels elektronischer Kurvenscheibe. Die dabei entstehenden Sollwerte werden an den b maXX Regler übergeben. Die Programmierung des Applikationsprogramms auf der b maXX PLC erfolgt über PROPROG wt II (siehe Handbuch PROPROG wt II und Applikationshandbuch b maXX PLC).

Kommunikation b maXX PLC → b maXX Regler

Die Sollwerte werden von der b maXX PLC an den b maXX Regler übergeben. Im b maXX Regler sind diese Sollwerte dann Parameter wie z. B. der Lagesollwert. Die Programmierung der Kommunikation zwischen b maXX PLC und b maXX Regler erfolgt über PROPROG wt II. Die Beschreibung der Programmierung der Kommunikation ist im Applikationshandbuch b maXX PLC beschrieben und wird in diesem Dokument nicht näher erläutert.

b maXX Regler

Auf dem b maXX Regler erfolgt die Regelung mit den Sollwerten (von der b maXX PLC) und den Istwerten vom Motor (bzw. b maXX Leistungsteil). Die Rahmenbedingungen (z. B. Momentengrenze) für diese Regelung sind parametrierbar.

Die Parametrierung des b maXX Regler erfolgt über WinBASS II (siehe Parameterhandbuch b maXX 4400).

Kommunikation b maXX Regler → b maXX Leistungsteil

Die Stellgrößen des b maXX Regler werden an das b maXX Leistungsteil übergeben.

b maXX Leistungsteil / Motor

Im b maXX Leistungsteil werden die Stellgrößen des b maXX Regler umgesetzt in Strom und Spannung mit denen der Motor betrieben wird. Im b maXX Leistungsteil werden auch Istwerte (wie z. B. Strom-Istwert) erfasst. Andere Istwerte (wie z. B. Motordrehzahl oder Lageistwert) werden auch über den Geber am Motor und das Geber-Modul am b maXX Regler erfasst.

Die Istwerte können über den umgekehrten Weg bis an den CANopen-Master gemeldet werden.

3.3.3 Durchzuführende Schritte

Die Montage, der Anschluss und die Inbetriebnahme von

- Motor
- b maXX 4400
- Optionsmodul b maXX PLC
- Optionsmodul CANopen-Slave

sowie des von Ihnen verwendeten

- CANopen-Master

müssen erfolgreich abgeschlossen sein.

Damit mit dem CANopen-Slave über das CANopen-Netzwerk Daten ausgetauscht werden können müssen folgende Schritte abgearbeitet werden:

- Anlegen eines PROPROG wt II Projekts für die b maXX PLC.
- Gerätekonfiguration mit dem b maXX Konfigurator.
- Konfigurierung des Optionsmodul CANopen-Slave mit dem CANopen-Slave Konfigurator (im b maXX Konfigurator).
 - Ergebnis:
Variablenliste für SDOs und PDOs für das PROPROG wt II Projekt,
EDS-Datei für ein CANopen-Master Konfigurationstool
- Programmierung des Applikationsteils im PROPROG wt II Projekt für die b maXX PLC.
 - Ergebnis:
PROPROG wt II Beispiel-Projekt mit INIT, NMT, EMCY, SDO, PDO.
- Ggf. Einbinden der EDS-Datei in ein CANopen-Master Konfigurationstool.

3.3.4 Anlegen eines PROPROG wt II Projekts

3.3.4.1 Vorgehen beim Anlegen eines Projektes

Um das Optionsmodul CANopen-Slave mit den CANopen-Funktionsbausteinen nach Profil DS 405 verwenden zu können, benötigen Sie ein PROPROG wt II - Projekt für die b maXX PLC. Falls Sie noch kein eigenes Projekt für ihre Applikation angelegt haben, erstellen Sie das Projekt bitte mit der Vorlage *bmaXX4000 mit Konfigurator*. Sie benötigen dazu ein PROPROG wt II Version 3.0 ab Build 262 und den b maXX Drive-Configurator für PROPROG wt II Version 1.0 (oder höher). Die Versionsnummer von PROPROG wt II finden Sie auf der Hülle der Installations CD von PROPROG wt II oder in PROPROG wt II selbst im Menüpunkt ? \ **Info**. Prüfen Sie auch ob die Bibliothek BM_TYPES_20bd05 (oder höher) in Ihrem PROPROG wt II - Projekt vorhanden ist. Sollte dies nicht der Fall sein, binden Sie diese Bibliothek bitte in Ihr Projekt ein. Sie enthält wichtige Datentypen

für CANopen. Binden Sie anschließend die Bibliothek CANop405_PLC01_20bd02 in Ihr Projekt ein.

3.3.4.2 Beispiel: Anlegen des Projektes "CANopenSlave_Example"

Es wurde das Beispiel-Projekt "CANopenSlave_Example" mit der Vorlage *bmaXX4000 mit Drive-Configurator* angelegt und die Bibliotheken BM_TYPES_20bd05 und CANop405_PLC01_20bd02 eingebunden.

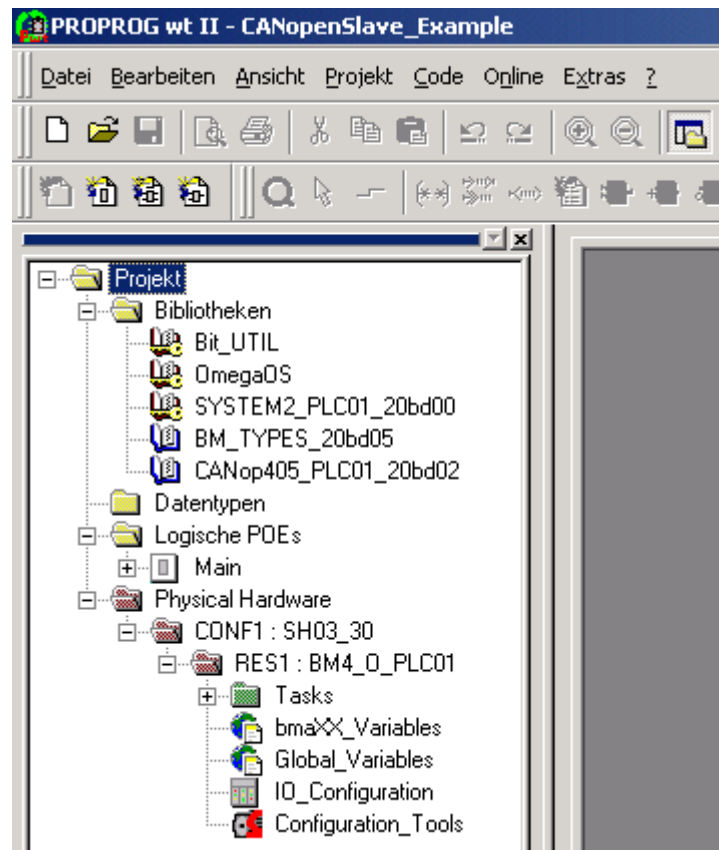


Abbildung 2: Beispiel: Anlegen des Projektes "CANopenSlave_Example"

3.3.4.3 Anlegen einer globalen Variable für den Datenaustausch

Für den Datenaustausch zwischen Optionsmodul BM4-O-CAN-03 und den Funktionsbausteinen wird eine globale Variable benötigt. Sie hat für den Anwender keine weitere Bedeutung. Diese globale Variable ist in Ihrem Projekt bereits angelegt, sofern das Projekt mit der Vorlage *bmaXX4000 mit Drive-Configurator* angelegt wurde. An den Funktionsbausteinen für CANopen wird am Ein-/Ausgang `_CANop405_CTRL` diese globale Variable angeschlossen.

Abhängig vom Steckplatz des Optionsmoduls (Slot G bis M) stehen Ihnen die globalen Variablen `_CANopenSlave_Ctrl_Slot_G` bis `_CANopenSlave_Ctrl_Slot_M` zur Verfügung.

Sie finden diese auch im Arbeitsblatt "Global_Variables":

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

```
(* Option module CANopen-Slave (BM4-O-CAN-03) *)
_CANopenSlave_Ctrl_Slot_G    AT %MB 3.2001792 : CANopen_PLC_SL_BMSTRUCT;
_CANopenSlave_Ctrl_Slot_H    AT %MB 3.3001792 : CANopen_PLC_SL_BMSTRUCT;
_CANopenSlave_Ctrl_Slot_J    AT %MB 3.4001792 : CANopen_PLC_SL_BMSTRUCT;
_CANopenSlave_Ctrl_Slot_K    AT %MB 3.5001792 : CANopen_PLC_SL_BMSTRUCT;
_CANopenSlave_Ctrl_Slot_L    AT %MB 3.6001792 : CANopen_PLC_SL_BMSTRUCT;
_CANopenSlave_Ctrl_Slot_M    AT %MB 3.7001792 : CANopen_PLC_SL_BMSTRUCT;
```

Abbildung 3: Globale Variablen für das Optionsmodul CANopen-Slave abhängig vom Slot

Ist die für den Steckplatz des Optionsmoduls benötigte globale Variable nicht im Projekt vorhanden, legen Sie, je nach Steckplatz (Slot G bis M), die globale Variable `_CANopenSlave_Ctrl_Slot_G` (bis `_CANopenSlave_Ctrl_Slot_M`) vom Datentyp `CANop405_PLC_SL_BMSTRUCT` an. Diese Variable muss als globale Variable deklariert werden und auf die Basisadresse des Optionsmoduls BM4-O-CAN-03 gelegt werden. Die Basisadresse ist abhängig vom Steckplatz:

Steckplatz (Slot)	Basisadresse für CANopen-Master Kommunikation
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792



HINWEIS

Im folgenden wird der Variablenname durch * ersetzt. Weiterhin wird davon ausgegangen, dass das Optionsmodul CANopen-Slave für b maXX PLC im Steckplatz G gesteckt ist.

Auf das Register `*.u_BAUDRATE` greift man demzufolge über

`_CANopenSlave_Ctrl_Slot_G.u_BAUDRATE` zu.

Dabei ist:

`_CANopenSlave_Ctrl_Slot_G` der Variablenname mit der Datentypkurzbezeichnung "_" für Struct
`u_BAUDRATE` das Register, in dem die Baudrate eingetragen ist, mit der das Optionsmodul CANopen-Slave am CAN-Bus arbeitet.

3.3.5 Gerätekonfiguration mit dem b maXX Konfigurator

Im PROPROG wt II Projekt, das Sie (unter [►Anlegen eines PROPROG wt II Projekts◄](#) auf Seite 24) bereits angelegt haben, finden Sie den Button für die Konfiguratoren (Configuration_Tools). Nach dem Doppelklick auf diesen Button öffnet sich das Konfiguratoren-Fenster, u.a. mit dem b maXX Konfigurator (Drive_Configuration).

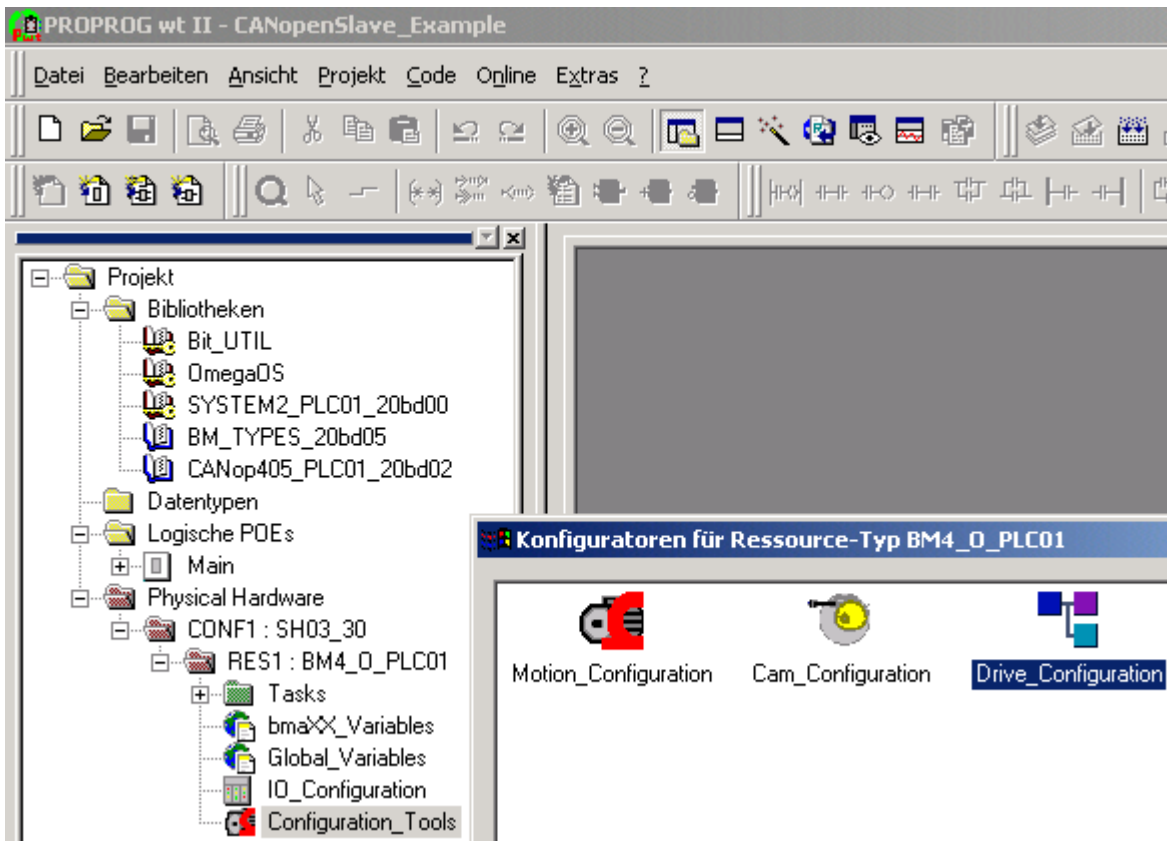


Abbildung 4: Öffnen des b maXX Konfigurators

Bevor Sie den b maXX Konfigurator öffnen, kompilieren Sie bitte Ihr PROPROG wt II Projekt (Menü: Code\Projekt neu erzeugen).

Klicken Sie jetzt auf den Button zum Öffnen des b maXX Konfigurator.

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

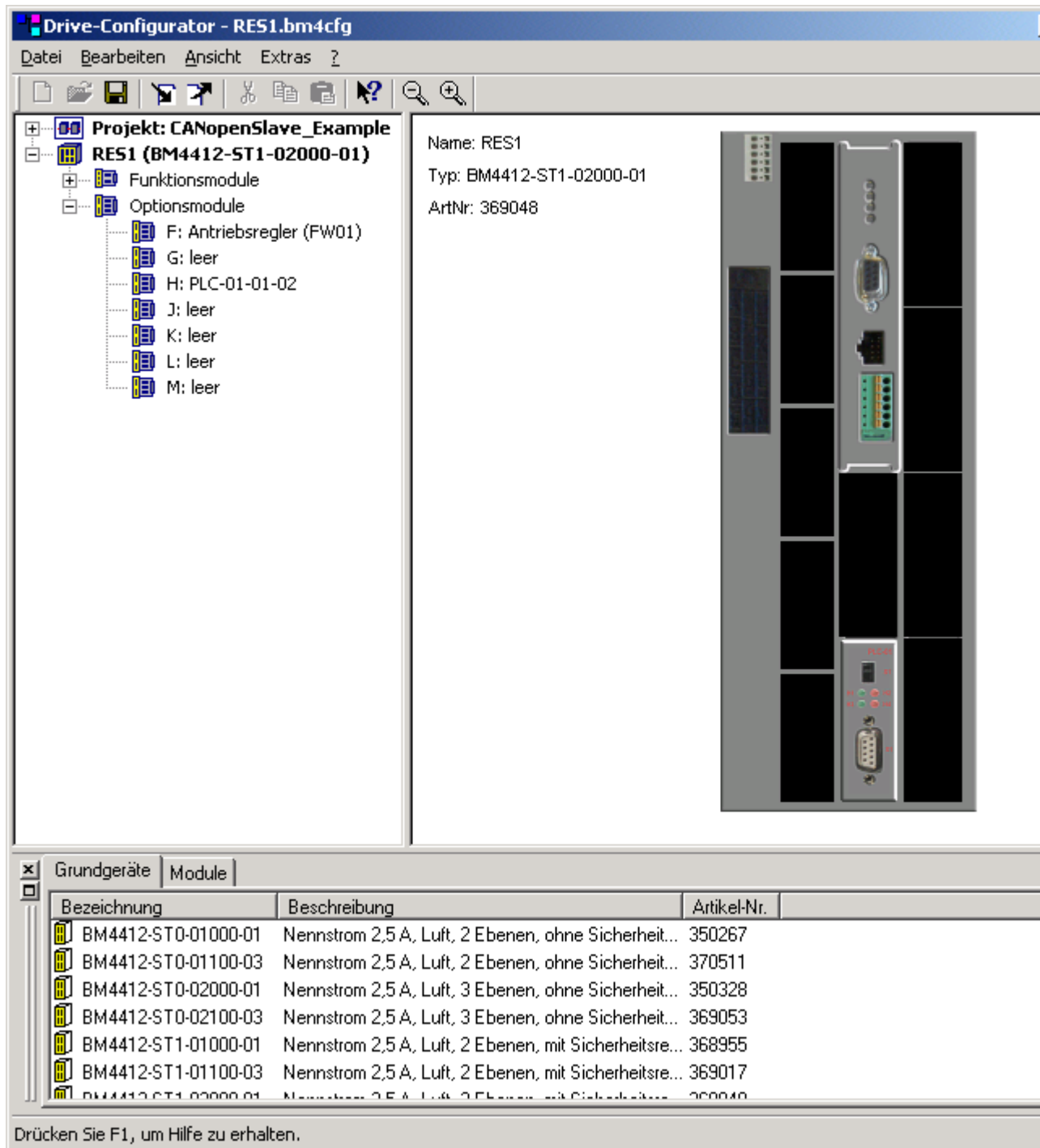


Abbildung 5: b maXX Konfigurator

Beschreibung der Schritte:

- Auswahl b maXX 4000 Grundgerät (inkl. b maXX Regler)
- Auswahl Optionsmodul b maXX PLC
- Auswahl Optionsmodul CANopen-Slave

3.3.5.1 Auswahl b maXX 4000 Grundgerät (inkl. b maXX Regler)

Im Projektbaum des b maXX Konfigurator ist per Default ein b maXX Grundgerät inkl. b maXX Regler eingestellt. In Abbildung 18 ist unter Ressource "RES1" das b maXX Grundgerät BM4412-ST1-02000-01 eingestellt.

- Falls Sie ein anderes b maXX Grundgerät verwenden wollen, wählen Sie über das Kontextmenü der Ressource den Punkt "Hinzufügen/Ändern" an.
- Wählen Sie aus dem Fenster "Auswahl Grundgerätetyp" Ihr b maXX Grundgerät aus.
- Bestätigen Sie die Auswahl mit "OK".

Damit wird das neue b maXX Grundgerät (inkl. b maXX Regler) übernommen und die Grafik angepasst.

Im Beispiel-Projekt wird das b maXX Grundgerät BM4412-ST0-01000-01 verwendet (siehe ► [Abbildung 6](#) ◀ auf Seite 30).

3.3.5.2 Auswahl Optionsmodul b maXX PLC

- Öffnen Sie im Projektbaum des b maXX Konfigurators die Einträge der Optionsmodule ("+" vor "Optionsmodule" anklicken").

Per Default ist eine b maXX PLC in Steckplatz "H" eingestellt. In Abbildung 18 Steckplatz "H" die b maXX PLC BM4-O-PLC-01-01-02 eingestellt.

- Falls Sie eine andere b maXX PLC verwenden wollen, wählen Sie über das Kontextmenü des Steckplatz "H" den Punkt "Hinzufügen/Ändern" an.
- Wählen Sie aus dem Fenster "Optionsmodul für Slot H" Ihre b maXX PLC aus.
- Bestätigen Sie die Auswahl mit "OK".

Damit wird die neue b maXX PLC übernommen und die Grafik angepasst.

Im Beispiel-Projekt wird die b maXX PLC BM4-O-PLC-01-00-02 auf Steckplatz "H" verwendet (siehe ► [Abbildung 6](#) ◀ auf Seite 30).

3.3.5.3 Auswahl Optionsmodul CANopen-Slave

- Wählen Sie über das Kontextmenü des Steckplatz "G" den Punkt "Hinzufügen/Ändern" an.
- Wählen Sie aus dem Fenster "Optionsmodul für Slot G" das Optionsmodul CANopen-Slave für b maXX (BM4-O-CAN-03-00-00) aus.
- Bestätigen Sie die Auswahl mit "OK".

Damit wird das Optionsmodul CANopen-Slave für b maXX PLC übernommen und die Grafik angepasst.

Im Beispiel-Projekt wird das Optionsmodul CANopen-Slave für b maXX PLC BM4-O-CAN-03-00-00 auf Steckplatz "G" verwendet.

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

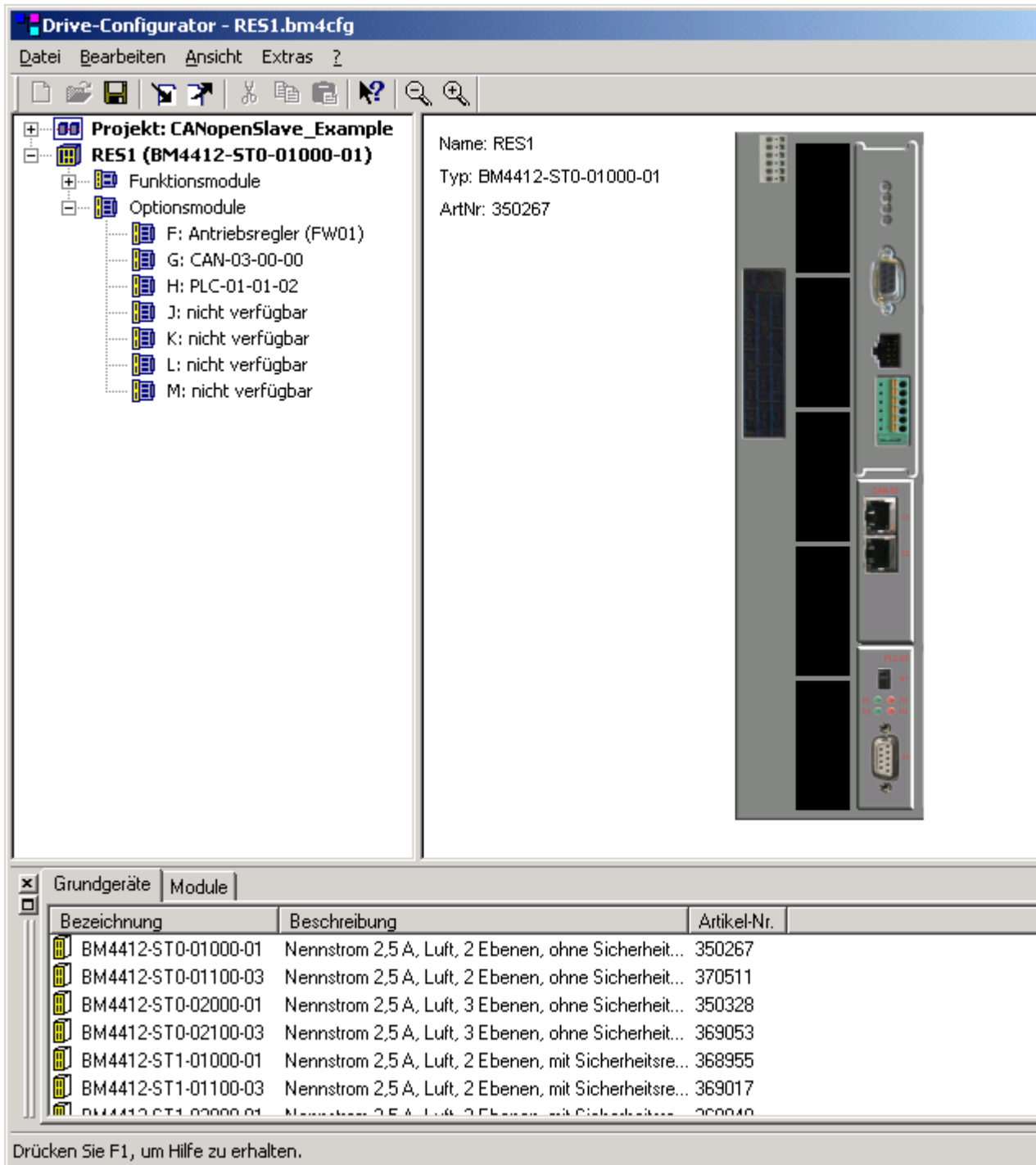


Abbildung 6: b maXX Konfigurator nach der Geräte-Konfiguration

Weiter mit „Konfigurierung des Optionsmodul CANopen-Slave mit dem CANopen-Slave Konfigurator“.

3.3.6 Konfigurierung des Optionsmodul CANopen-Slave mit dem CANopen-Slave Konfigurator

In diesem Kapitel erläutern wir Ihnen die Bedienung des CANopen-Slave Konfigurators. Der CANopen-Slave Konfigurator

- generiert Teile des globalen Variablen-Arbeitslatts "bmaXX_Variables" für das PROPROGRAM II Projekt (liest vorher ggf. Teile eines bereits existierenden globalen Variablen-Arbeitslatts "bmaXX_Variables" des PROPROGRAM II Projekts ein)
- generiert die EDS-Datei welche das Gerät "b maXX 4400 mit b maXX PLC und Optionsmodul CANopen-Slave" für CANopen-Master Konfiguratoren beschreibt

Öffnen Sie ggf. den CANopen-Slave Konfigurator aus PROPROGRAM II über Configuration_Tools → Drive_Configuration → "Optionsmodule \ G: CAN-03-00-00" - Kontextmenü Punkt "Konfigurieren".

- Klicken Sie im CANopen-Slave Konfigurator im Bereich CANopen-Anbindung den Button "Einstellungen" an.
- Aktivieren Sie den Radio-Button "CANopen-Anbindung ... der integrierten SPS"

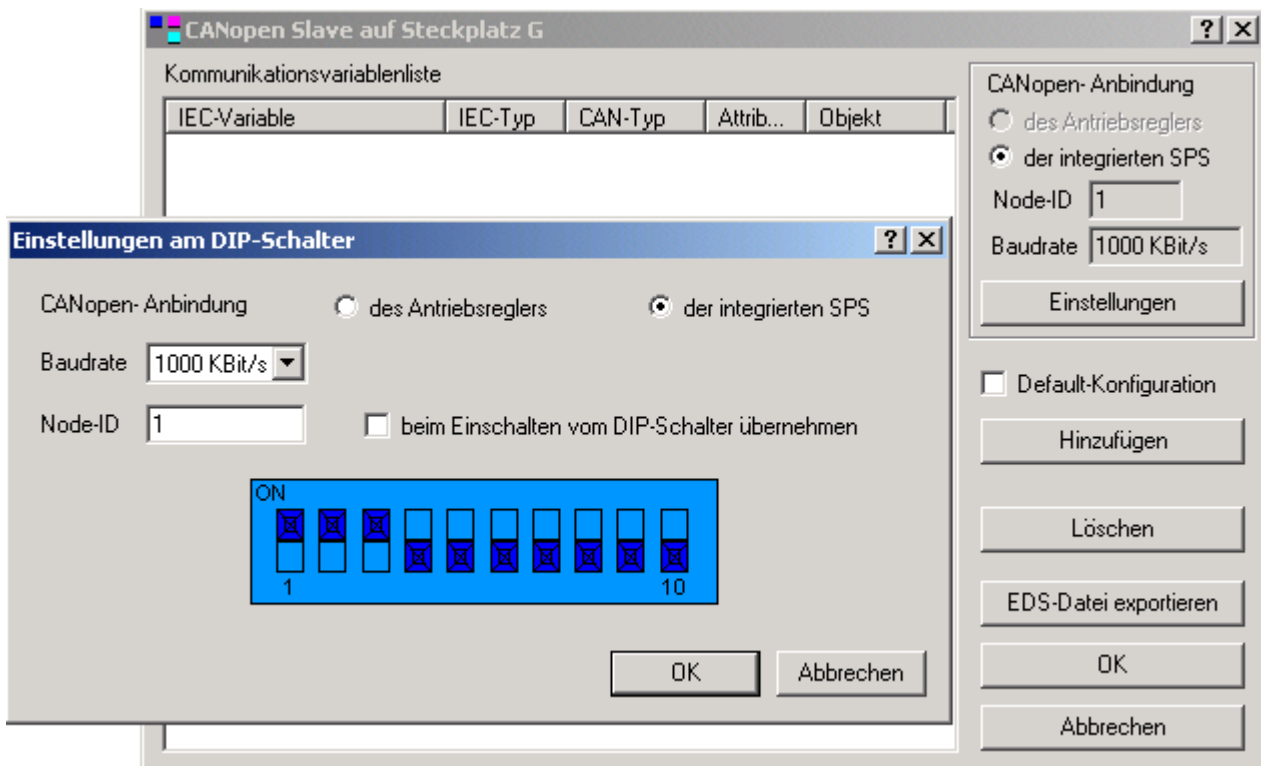


Abbildung 7: Auswahl CANopen-Anbindung der integrierten SPS

Durch die Aktivierung des Radio-Buttons "CANopen-Anbindung ... der integrierten SPS" wird die Anzeige der DIP-Schalter 1, 2 und 3 auf "ON" gestellt. Sie müssen bei der Inbetriebnahme des Optionsmodul CANopen-Slave diese DIP-Schalter auf "ON" stellen, damit Ihr Optionsmodul ein Optionsmodul CANopen-Slave für b maXX PLC ist.

Weiter mit „Einstellen der Baudrate“.

3.3.6.1 Einstellen der Baudrate

Stellen Sie die Baudrate mit der das Optionsmodul CANopen-Slave am CANopen-Bus-system arbeiten soll wie folgt ein:

- Klicken Sie im CANopen-Slave Konfigurator im Bereich CANopen-Anbindung den Button „Einstellungen“ an, sofern das Fenster „Einstellungen am DIP-Schalter“ noch nicht geöffnet ist.
- Klicken Sie in die EditBox „Baudrate“.
- Wählen Sie aus der Liste die gewünschte Baudrate aus.

Im Beispiel ist die Baudrate 500 kBit/s ausgewählt.

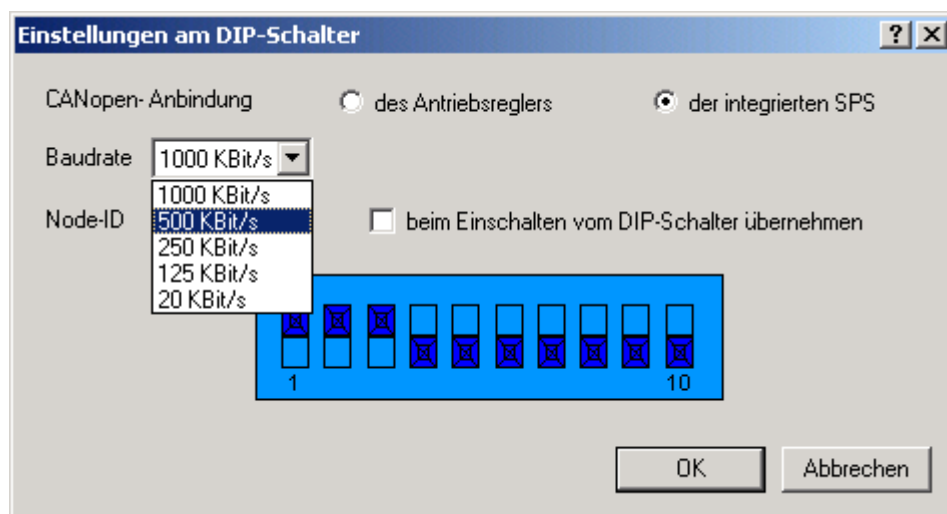


Abbildung 8: Auswahl der Baudrate 500 kbit/s

Die Baudrate steht nach der Konfigurierung des Optionsmodul CANopen-Slave als Variable "us_CoSI_Baudrate" im globalen Variablen-Arbeitslatt "bmaXX_Variables" im PROPROG wt II Projekt zur Verfügung.

Weiter mit „Einstellen der Node-ID“.

3.3.6.2 Einstellen der Node-ID

Stellen Sie die Node-ID (Netzwerkknotenadresse, CANopen-Slave Adresse) mit der sich das Optionsmodul CANopen-Slave am CANopen-Bussystem melden soll wie folgt ein:

- Klicken Sie im CANopen-Slave Konfigurator im Bereich CANopen-Anbindung den Button „Einstellungen“ an, sofern das Fenster „Einstellungen am DIP-Schalter“ noch nicht geöffnet ist.
- Klicken Sie im Fenster „Einstellungen am DIP-Schalter“ in die EditBox "Node-ID".
- Tragen Sie die gewünschte Node-ID ein. Sie können für das Optionsmodul CANopen-Slave eine Node-ID von 1 bis 127 vergeben. Andere Werte sind nicht erlaubt und führen zu einer Fehlermeldung

Im Beispiel ist die Node-ID "5" eingestellt.

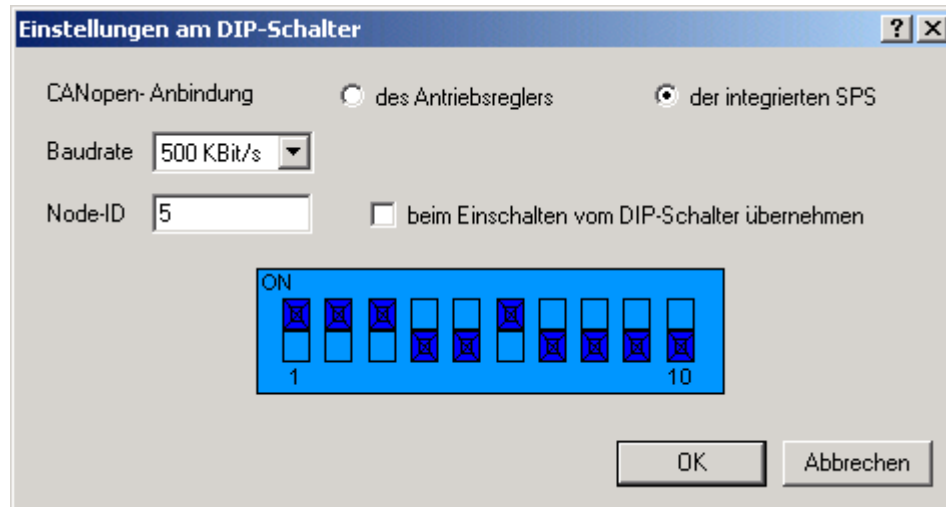


Abbildung 9: Einstellung der Node-ID "5"

Durch das Eintragen einer Node-ID wird die Anzeige der DIP-Schalter 4 bis 10 geändert. Dabei ist zu beachten, dass

$$\text{Node-ID} = \text{"DIP-Schalter 4 bis 10"} + 1$$

ist. D. h. bei Node-ID "5" wird als Wert von "DIP-Schalter 4 bis 10" eine "4" (DIP-Schalter 6 = "ON") angezeigt.

Sie brauchen für das Einstellen der Node-ID nicht die DIP-Schalter 4 bis 10 auf dem Optionsmodul CANopen-Slave einstellen (die Einstellung der Node-ID erfolgt durch den FB CANop405_INIT_SL im PROPROG wt II Projekt).

Alternativ können Sie auch die CheckBox "beim Einschalten vom DIP-Schalter übernehmen" aktivieren. In diesem Fall übernimmt das Optionsmodul CANopen-Slave die Node-ID von den DIP-Schaltern 4 bis 10.

Dabei ist zu beachten, dass

$$\text{Node-ID} = \text{"DIP-Schalter 4 bis 10"} + 1$$

ist. D. h. bei "DIP-Schalter 4 bis 10" = "4" (DIP-Schalter 6 = "ON") ergibt sich als Node-ID der Wert "5".

Die Node-ID steht nach der Konfigurierung des Optionsmodul CANopen-Slave als Variable "us_CoSI_G_NodeID" im globalen Variablen-Arbeitslatt "bmaXX_Variables" im PROPROG wt II Projekt zur Verfügung.

Damit sind die Einstellungen im CANopen-Slave Konfigurator im Bereich CANopen-Anbindung beendet.

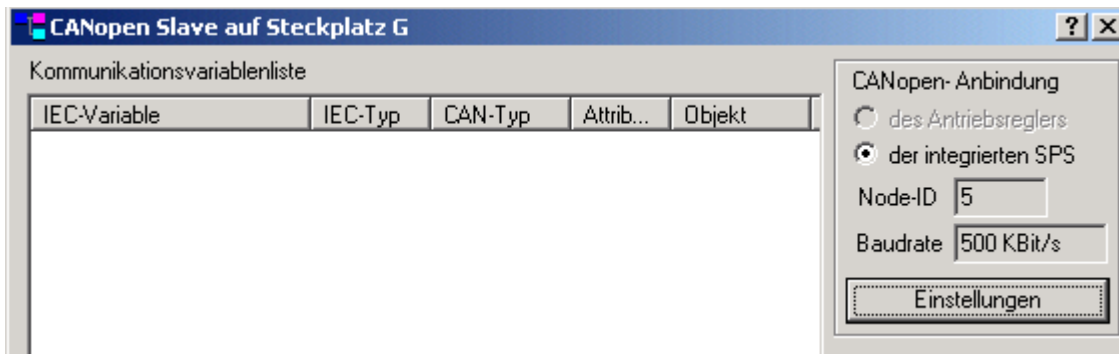


Abbildung 10: Einstellungen im Bereich CANopen-Anbindung

3.3.6.3 Anlegen von Variablen

In diesem Kapitel erläutern wir Ihnen wie Sie eine IEC 61131-3 Variable anlegen und einem CANopen-Objekt zuordnen. Diese Variable steht Ihnen dann im PROPROGRAMM II Projekt, im globalen Variablen-Arbeitsblatt "bmaXX_Variablen", für die Programmierung Ihrer Applikation zur Verfügung. Weiterhin steht Ihnen das CANopen-Objekt (das der Variablen zugeordnet ist) für die Kommunikation über CANopen zur Verfügung.

- ro (read only / nur lesbar):
Die b maXX PLC übergibt die Daten aus den Variablen an das Optionsmodul CANopen-Slave und dieses sendet die Daten an den CANopen-Master. Aus der Sicht des CANopen-Masters sind diese Daten die Istwerte vom b maXX, also Daten die gelesen werden.
- rw (read and write / schreib- u. lesbar)
Der CANopen-Master sendet die Daten an das Optionsmodul CANopen-Slave und dieses übergibt die Daten an die b maXX PLC. Auf der b maXX PLC stehen die Daten in den Variablen zur Verfügung. Aus der Sicht des CANopen-Masters sind diese Daten die Sollwerte zum b maXX, also Daten die geschrieben werden.

Sie legen ein Variable an indem Sie im CANopen-Slave Konfigurator auf den Button "Hinzufügen" klicken.

- Klicken Sie auf den Button "Hinzufügen"

Das Fenster "Einstellungen Kommunikations-Variable" wird geöffnet.

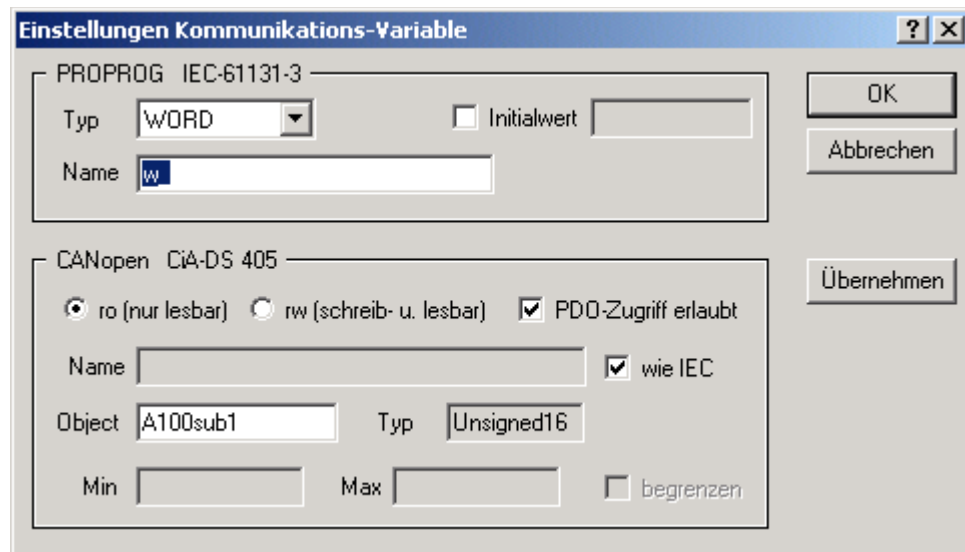


Abbildung 11: Öffnen des Fensters "Einstellungen Kommunikations-Variablen"

Das Fenster ist in die Bereiche "PROPROG IEC 61131-3" und "CANopen CiA DS 405" unterteilt.

Bereich PROPROG IEC 61131-3

Wählen Sie im Bereich PROPROG IEC 61131-3 in der EditBox "Typ" den Datentyp der Variablen aus. Zur Auswahl stehen Ihnen hier die IEC 61131-3 Datentypen BOOL, SINT, USINT, INT, UINT, DINT, UDINT, BYTE, WORD, DWORD und REAL. Für unser Beispiel wählen Sie bitte den Datentyp UDINT.

- Klicken Sie in die EditBox "Typ"
- Wählen Sie aus der Liste den gewünschten Datentyp aus.

Im Beispiel ist der Datentyp UDINT ausgewählt.

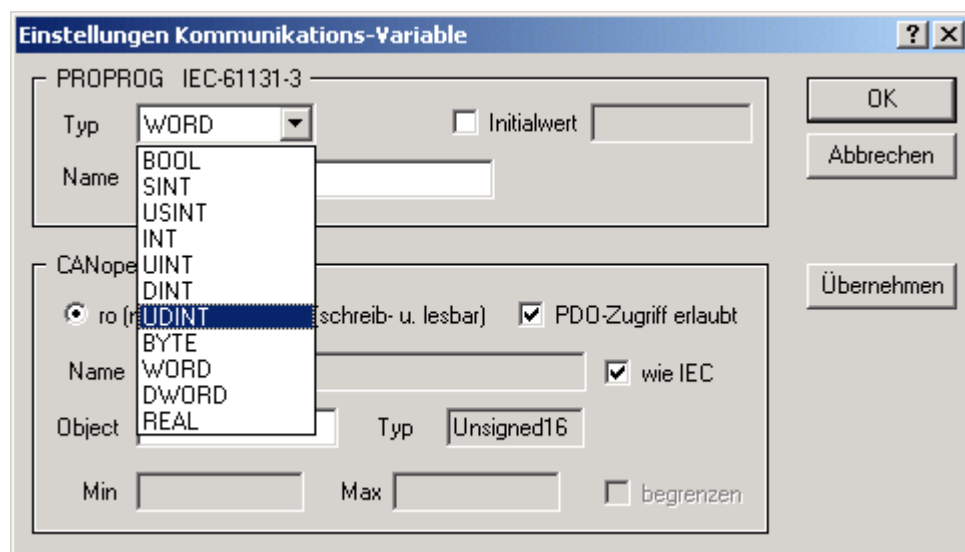


Abbildung 12: Datentyp auswählen

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

In der EditBox "Name" erscheint dann die entsprechende Datentyp-Kurzbezeichnung für die Variable. Sie können nun in der EditBox "Name" den Variablen-Namen (nach der Datentyp-Kurzbezeichnung) angeben.

- Geben Sie in der EditBox "Name" den gewünschten Variablen-Namen an.

Im Beispiel ist der Variablenname "ud_MyReadVar1" angegeben.

The screenshot shows a dialog box titled "Einstellungen Kommunikations-Variablen". It is divided into two main sections: "PROPROG IEC-61131-3" and "CANopen CiA-DS 405". In the "PROPROG" section, the "Typ" dropdown is set to "UDINT", and the "Name" text box contains "ud_MyReadVar1". There is an unchecked checkbox for "Initialwert" next to an empty text box. In the "CANopen" section, the "ro (nur lesbar)" radio button is selected, and the "PDD-Zugriff erlaubt" checkbox is checked. The "Name" text box also contains "ud_MyReadVar1", and the "wie IEC" checkbox is checked. The "Object" text box contains "A200sub1" and the "Typ" dropdown is set to "Unsigned32". There are also empty "Min" and "Max" text boxes and an unchecked "begrenzen" checkbox. On the right side, there are buttons for "OK", "Abbrechen", and "Übernehmen".

Abbildung 13: Variablen-Namen vergeben

In der EditBox Initialwert können Sie einen Startwert für Ihre Variable vergeben.

- Klicken Sie die CheckBox (links von der EditBox "Initialwert") an.
- Tragen Sie in der EditBox "Initialwert" den Startwert für die Variable ein.

This screenshot is identical to the previous one, but with the "Initialwert" checkbox checked and the "Initialwert" text box containing the value "1000".

Abbildung 14: Initialisierungswert vorgeben

Damit ist das Anlegen der Variablen im Bereich "PROPROG IEC 61131-3" beendet und nun erfolgt die Zuordnung der Variablen zu einem CANopen-Objekt im Bereich "CANopen CiA DS 405".

Bereich CANopen CiA DS 405

In diesem Bereich wird der Variablen aus dem Bereich "PROPROG IEC 61131-3" ein CANopen-Objekt zugeordnet. Deshalb wird in diesem Bereich anstelle des Begriffs "Variable" der Begriff "Objekt" verwendet.

Mit den RadioButtons "ro (nur lesbar)" und "rw (schreib- u. lesbar)" legen Sie die Übertragungsrichtung des Objekts fest.

Wählen Sie "rw (schreib- u. lesbar)", wird das Objekt über den CANopen-Bus empfangen und in die Variable auf der b maXX PLC geschrieben. Das Objekt kann nur einer RxPDO des Optionsmodul CANopen-Slave zugeordnet sein.

Wählen Sie "ro (nur lesbar)", wird die Variable von der b maXX PLC gelesen und als Objekt auf den CANopen-Bus gesendet. Das Objekt kann nur einer TxPDO des Optionsmodul CANopen-Slave zugeordnet sein.

Mit der CheckBox "PDO-Zugriff erlaubt" legen Sie fest, ob dieses Objekt in PDOs verwendet werden kann, d. h. ob für dieses Objekt ein Mapping in PDOs erlaubt ist. Dieser Eintrag ist für die EDS-Datei notwendig.

- Klicken Sie den RadioButton "ro (nur lesbar)" an.
- Klicken Sie die CheckBox "PDO-Zugriff erlauben" an.

The screenshot shows a dialog box titled "Einstellungen Kommunikations-Variablen". It is divided into two main sections. The top section, labeled "PROPROG IEC-61131-3", contains a dropdown menu for "Typ" set to "UDINT", a checked checkbox for "Initialwert" with a value of "1000", and a text input field for "Name" containing "ud_MyReadVar1". The bottom section, labeled "CANopen CiA-DS 405", contains two radio buttons: "ro (nur lesbar)" is selected, and "rw (schreib- u. lesbar)" is unselected. There is a checked checkbox for "PDO-Zugriff erlaubt" and another checked checkbox for "wie IEC". Below these are text input fields for "Name" (containing "ud_MyReadVar1"), "Object" (containing "A200sub1"), and "Typ" (containing "Unsigned32"). At the bottom of this section are two empty text input fields for "Min" and "Max", and an unchecked checkbox for "begrenzen". On the right side of the dialog, there are three buttons: "OK", "Abbrechen", and "Übernehmen".

Abbildung 15: PDO-Einstellungen Variable "ud_MyReadVar1"

Sie können dem Objekt einen symbolischen Namen geben. Der Übersichtlichkeit halber ist der symbolische Name des Objekts gleich dem Variablennamen (aus dem Bereich "PROPROG IEC 61131-3"). Wenn Sie den symbolischen Namen des Objekts ändern wollen, schalten Sie die CheckBox "wie IEC" aus und vergeben dann in der EditBox "Name" den neuen symbolischen Namen für das Objekt. Im Beispiel wird der symbolische Name des Objekts nicht geändert.

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

In der EditBox "Object" wird der Objekt-Index und der Objekt-Subindex des (CANopen-) Objekts eingetragen. Der CANopen-Slave Konfigurator trägt automatisch den nächsten freien Objekt-Index und Objekt-Subindex gemäß dem CiA Geräte-Profil DS 405 ein (siehe Anhang [►B.2 Objekte nach DS 405◄](#) ab Seite 101). Sie können auch andere Objekt-Indizes und/oder Objekt-Subindizes nach DS 405 eintragen. Eine Überprüfung ob die Objekt-Indizes und Objekt-Subindizes bereits verwendet werden findet ebenso statt wie die Überprüfung ob die Objekt-Indizes zum gewählten Datentyp gemäß DS 405 passen. Im Beispiel wird der vom CANopen-Slave Konfigurator vorgeschlagene Wert, Objekt-Index A200h und Objekt-Subindex 1 (→ A200sub1) nicht geändert.

Die CheckBox "Begrenzen" und die EditBoxes "Min" und "Max" werden bei "ro (nur lesbar)"-Objekten nicht verwendet.

Damit ist das Zuordnen der Variablen aus dem Bereich "PROPROG IEC 61131-3" zu einem CANopen-Objekt im Bereich "CANopen CiA DS 405" beendet. Klicken Sie auf den Button "OK"

Das Fenster "Einstellungen Kommunikations-Variable " wird geschlossen und im Fenster "CANopen-Slave auf Steckplatz x" (x = G bis M) wird die neue Variable angezeigt.

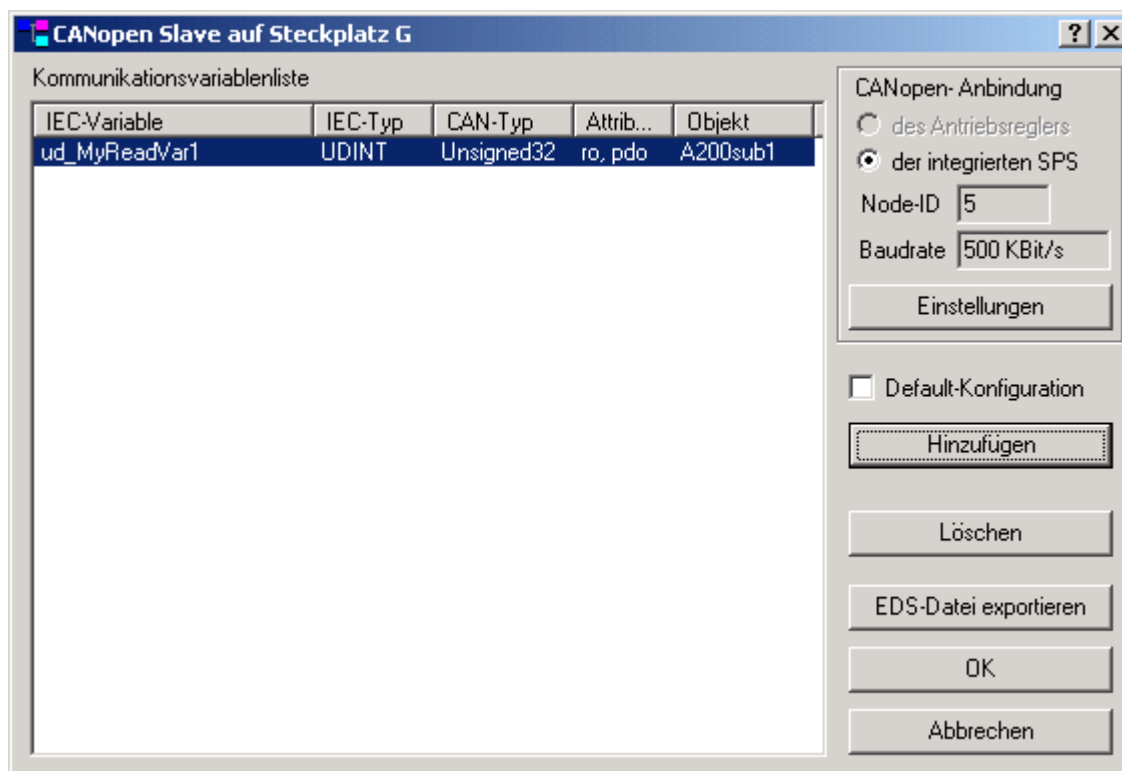


Abbildung 16: Variable "ud_MyReadVar1"

Im folgenden legen wir noch drei weitere Variablen an und ordnen sie Objekten zu.

Die Variable "ud_MyReadVar2" ordnen wir dem Objekt-Index A200h und dem Objekt-Subindex 2 zu.

Die Variable "ud_MyWriteVar1" ordnen wir dem Objekt-Index A680h und dem Objekt-Subindex 1 zu.

Die Variable "u_MyWriteVar2" ordnen wir dem Objekt-Index A580h und dem Objekt-Subindex 1 zu.

- ▶ Klicken Sie auf den Button "Hinzufügen"
- ▶ Klicken Sie in die EditBox "Typ" und wählen Sie aus der Liste den Datentyp UDINT aus.
- ▶ Geben Sie in der EditBox "Name" den Variablennamen "ud_MyReadVar2" an.
- ▶ Klicken Sie die CheckBox (links von der EditBox "Initialwert") an und tragen Sie in der EditBox "Initialwert" den Startwert "2000" für die Variable ein.
- ▶ Klicken Sie den RadioButton "ro (nur lesbar)" an.
- ▶ Klicken Sie die CheckBox "PDO-Zugriff erlaubt" an.
- ▶ In der EditBox "Object" tragen Sie "A200Sub2" ein (dieser Wert sollte bereits durch den CANopen-Slave Konfigurator eingetragen sein).

Abbildung 17: PDO-Einstellungen Variable "ud_MyReadVar2"

- ▶ Klicken Sie auf den Button "Übernehmen".
- ▶ In der EditBox "Typ" wählen Sie aus der Liste den Datentyp UDINT aus (dieser Datentyp sollte bereits durch den CANopen-Slave Konfigurator eingetragen sein).
- ▶ Geben Sie in der EditBox "Name" den Variablennamen "ud_MyWriteVar1" an.
- ▶ Klicken Sie die CheckBox (links von der EditBox "Initialwert") an und tragen Sie in der EditBox "Initialwert" den Startwert "1500" für die Variable ein.
- ▶ Klicken Sie den RadioButton "rw (schreib- u. lesbar)" an.
- ▶ Klicken Sie die CheckBox "PDO-Zugriff erlaubt" an.
- ▶ In der EditBox "Object" tragen Sie "A680Sub1" ein (dieser Wert sollte bereits durch den CANopen-Slave Konfigurator eingetragen sein).
- ▶ Klicken Sie die CheckBox "Begrenzen" an und tragen Sie in der EditBox "Min" den Wert 1000 und der EditBox "Max" den Wert 2000 ein. Diese Werte werden in die EDS-Datei eingetragen. Damit kann ein CANopen-Master prüfen ob die Werte die er sendet innerhalb dieses Bereichs liegen. Im CANopen-Slave findet auf dem Optionsmodul CANopen-Slave keine Überprüfung dieser Begrenzung statt. Eine Überprüfung der Werte muss vom Anwender im Appli-

kationsprogramm auf der b maXX PLC des CANopen-Slaves vorgenommen werden (mit CANopen-Slave ist hier das Gerät b maXX 4400 mit b maXX PLC und Optionsmodul CANopen-Slave gemeint).

The screenshot shows a software configuration window titled "Einstellungen Kommunikations-Variablen". It is split into two main sections. The top section, labeled "PROPROG IEC-61131-3", contains a dropdown menu for "Typ" set to "UDINT", a checked checkbox for "Initialwert" with a value of "2000", and a text field for "Name" containing "ud_MyWriteVar1". The bottom section, labeled "CANopen CiA-DS 405", features radio buttons for "ro (nur lesbar)" (unselected) and "rw (schreib- u. lesbar)" (selected). It also has a checked checkbox for "PDO-Zugriff erlaubt", a text field for "Name" with "ud_MyWriteVar1", a checked checkbox for "wie IEC", a text field for "Object" with "A680sub1", a dropdown for "Typ" set to "Unsigned32", text fields for "Min" (1000) and "Max" (2000), and a checked checkbox for "begrenzen". On the right side of the window, there are three buttons: "OK", "Abbrechen", and "Übernehmen".

Abbildung 18: PDO-Einstellungen Variable "ud_MyWriteVar1"

- Klicken Sie auf den Button "Übernehmen".
- In der EditBox "Typ" wählen Sie aus der Liste den Datentyp UINT aus.
- Geben Sie in der EditBox "Name" den Variablennamen "u_MyWriteVar2" an.
- Diese Variable erhält keinen Startwert.
- Klicken Sie den RadioButton "rw (schreib- u. lesbar)" an.
- Klicken Sie die CheckBox "PDO-Zugriff erlaubt" an.
- In der EditBox "Object" tragen Sie "A580Sub1" ein (dieser Wert sollte bereits durch den CANopen-Slave Konfigurator eingetragen sein).

Für dieses Objekt wird keine Begrenzung angegeben.

Der CANopen-Slave Konfigurator verwendet als obere und untere Begrenzung die obere und untere Grenze des Wertebereichs des Datentyps (in diesem Fall 0x0000 und 0xFFFF für den Datentyp UINT).

Diese Werte werden in die EDS-Datei eingetragen.

Damit kann ein CANopen-Master prüfen ob die Werte die er sendet innerhalb dieses Bereichs liegen.

Im CANopen-Slave findet auf dem Optionsmodul CANopen-Slave keine Überprüfung dieser Begrenzung statt. Eine Überprüfung der Werte muss vom Anwender im Applikationsprogramm auf der b maXX PLC des CANopen-Slaves vorgenommen werden (mit CANopen-Slave ist hier das Gerät b maXX 4400 mit b maXX PLC und Optionsmodul CANopen-Slave gemeint).

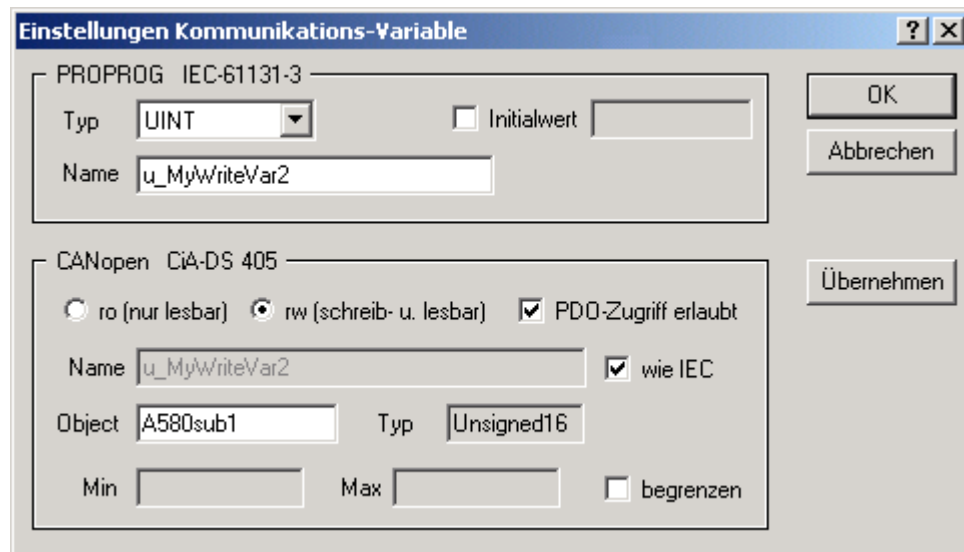


Abbildung 19: PDO-Einstellungen Variable "u_MyWriteVar2"

- Klicken Sie auf den Button "OK". Das Fenster "Einstellungen Kommunikations-Variablen" wird geschlossen und im Fenster "CANopen-Slave auf Steckplatz x" (x = G bis M) werden die neuen Variablen angezeigt.

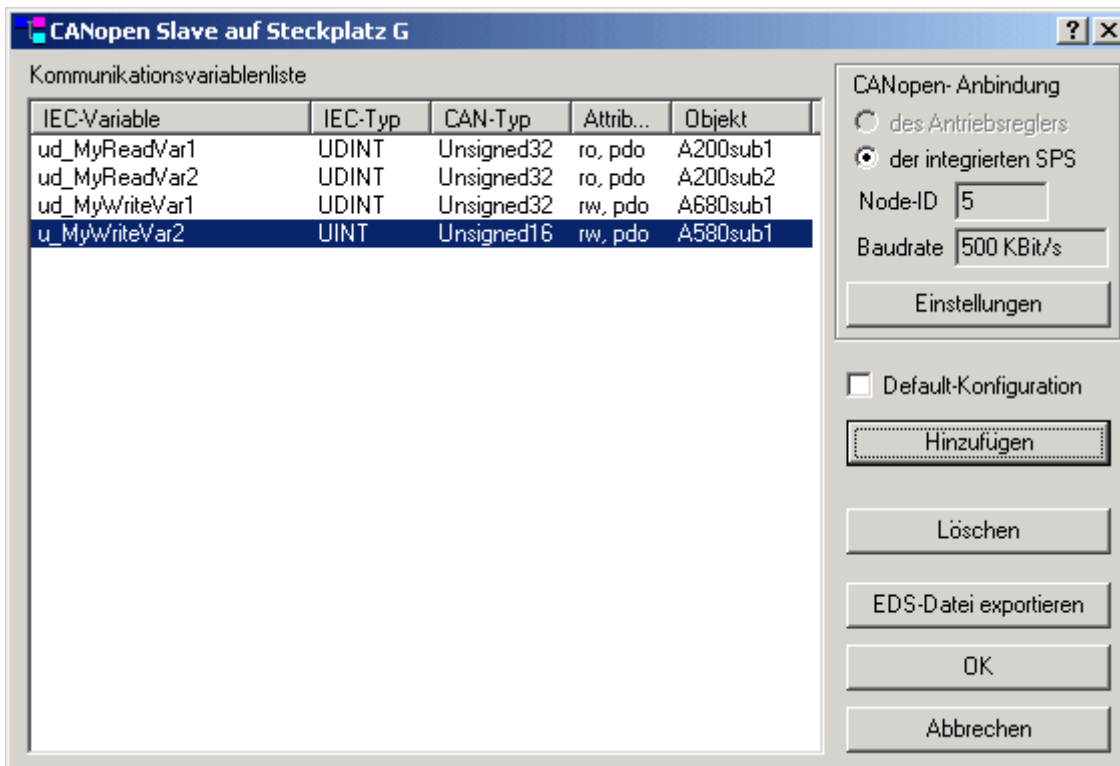


Abbildung 20: Angelegte Variablen

Damit ist das Anlegen der Variablen "ud_MyReadVar1", "ud_MyReadVar2", "ud_MyWriteVar1" und "u_MyWriteVar1" und deren Zuordnung zu den Objekten "A200sub1", "A200sub2", "A680sub1" und "A580sub1" abgeschlossen.

Um eine EDS-Datei zu erzeugen lesen Sie bitte den Abschnitt [►Erzeugen EDS-Datei◄](#) auf Seite 42.

- Beenden Sie den CANopen-Slave Konfigurator durch Klicken auf den Button "OK".
- Beenden Sie den b maXX Konfigurator über Datei\Beenden und bestätigen Sie die Abfrage "...speichern?" mit "Ja".

Jetzt wird das globale Variablenarbeitsblatt "bmaXX_Variables" für das PROPROGRAM wt II Projekt erzeugt und in den Projektbaum eingefügt bzw. aktualisiert. Im PROPROGRAM wt II Projekt stehen die Variablen für die Programmierung der Applikation zur Verfügung.

```
ud_MyReadVar1 AT %MD3.2002304 : UDINT := UDINT#1000;  
ud_MyReadVar2 AT %MD3.2002308 : UDINT := UDINT#2000;  
ud_MyWriteVar1 AT %MD3.2002312 : UDINT := UDINT#1500;  
u_MyWriteVar2 AT %MW3.2002316 : UINT; (* A580sub1 rw=!
```

Abbildung 21: globales Variablenarbeitsblatt "bmaXX_Variables"

3.3.6.4 Erzeugen EDS-Datei

Mit einer EDS-Datei (EDS - Electronic Data Sheet, elektronisches Datenblatt) wird ein CANopen-Gerät am CANopen-Bus beschrieben. Sie enthält u. a. die Informationen die zur Kommunikation mit dem CANopen-Gerät notwendig sind. Diese EDS-Datei brauchen Sie, um in einem CANopen-Master-Konfigurator das CANopen-Gerät "b maXX 4400 mit b maXX PLC und Optionsmodul CANopen-Slave" nutzbar zu machen. Die Namensweiterung der EDS-Datei ist "eds" (z. B. MyEDSFile.eds)



HINWEIS

Der CANopen-Slave (Gerät b maXX 4400 mit b maXX PLC und Optionsmodul CANopen-Slave) ist durch die b maXX PLC ein nach IEC 61131-3 programmierbares Gerät.

Bei diesen Geräten ändert sich der Inhalt der EDS-Datei sobald

- andere Variablen und/oder Objektzuordnungen verwendet werden als in der Default-Einstellung festgelegt sind
- sobald andere Variablen und/oder Objektzuordnungen verwendet werden als in der letzten Einstellung festgelegt waren

Eine Änderung der Default-Einstellung oder der letzten Einstellung für die Variablen und deren Zuordnung zu den Objekten bedingt zwingend das Erzeugen einer neuen EDS-Datei (und die Einbindung dieser EDS-Datei im CANopen-Master-Konfigurator).

Nachdem Sie im Abschnitt "Anlegen von Variablen" die Variablen im PROPROGRAM wt II Projekt angelegt und deren Zuordnung zu CANopen-Objekten getroffen haben, kann nun die EDS-Datei erzeugt werden.

Falls Sie keine eigene Zuordnung von Variablen und Objekten getroffen haben, können Sie auch den Default-Button drücken. Dann wird die Default-Zuordnung von Variablen und Objekten verwendet.

**HINWEIS**

Das Anklicken der CheckBox "Default-Konfiguration" löscht evtl. vorhandene Variablen im CANopen-Slave Konfigurator!

**HINWEIS**

Die Default EDS-Datei für das Optionsmodul CANopen-Slave für b maXX PLC finden Sie auf der CD "CANop405_PLC01_20bd02" (oder höher), auf der sich auch die Bibliothek "CANop405_PLC01_20bd02" (oder höher) befindet.

Um die EDS-Datei zu erzeugen, müssen Sie das Fenster "CANopen-Slave auf Steckplatz G" öffnen (PROPROG wt II Projekt → Configuration_Tools → Drive_Configuration → Optionsmodule \ G:CAN-03-00-00 → Kontextmenü Punkt „Konfigurieren“).

Zum Erzeugen der EDS-Datei klicken Sie auf den Button "EDS-Datei exportieren" und geben einen Dateinamen und den Speicherort für die EDS-Datei an.

- Öffnen Sie den CANopen-Slave Konfigurator, legen Sie die gewünschten Variablen an und ordnen Sie ihnen CANopen-Objekte zu (siehe ► [Anlegen von Variablen](#) ◀ auf Seite 34).
- Klicken Sie auf den Button "EDS-Datei exportieren". Das Fenster "Datei speichern unter" wird geöffnet.
- Suchen Sie im Fenster "Datei speichern unter" den Speicherort, z. B. Ihren selbstgestellten Ordner EDS-Files und geben sie in der EditBox "Dateiname" den Namen der EDS-Datei an.

Im Beispiel heißt die EDS-Datei "MyEDSFile.eds"

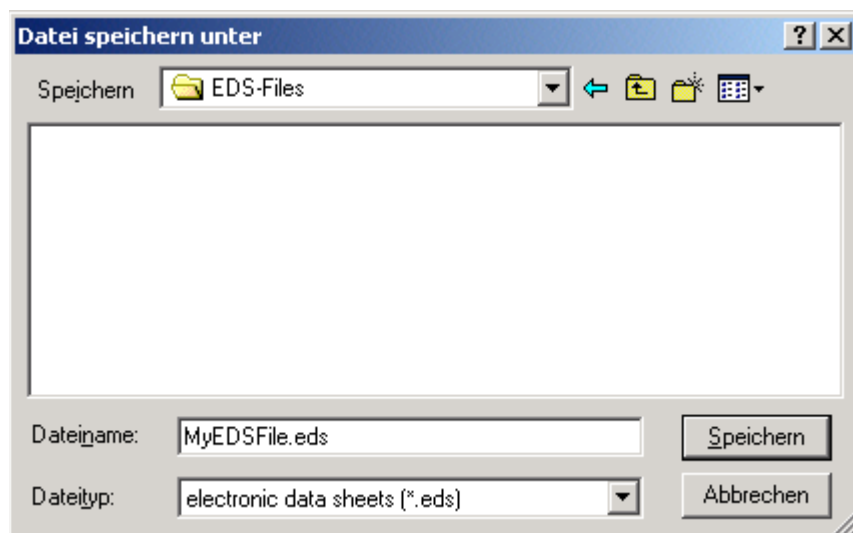


Abbildung 22: EDS-Datei exportieren

- Speichern Sie die Datei durch Klick auf den Button „Speichern“.

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

- ▶ Beenden Sie den CANopen-Slave Konfigurator durch Klicken auf den Button "OK". Schließen Sie den b maXX Konfigurator durch Datei\Beenden oder durch Klicken auf [x].

Die EDS-Datei steht für die Verwendung in CANopen-Master Konfiguratoren zur Verfügung.

3.3.7 Programmierung der Applikation im PROPROG wt II Projekt

3.3.7.1 Initialisierung des Optionsmoduls CANopen-Slave

Bei der Initialisierung des Optionsmodul CANopen-Slave werden dem Optionsmodul die Initialisierungsdaten übergeben. Diese Initialisierungsdaten sind u. a. Baudrate und Node-ID.

Vorgehen bei der Initialisierung des Optionsmoduls CANopen-Slave

Die Initialisierung des Optionsmoduls CANopen-Slave erfolgt mit dem FB CANop405_INIT_SL. Um diesen Funktionsbaustein zu verwenden, gehen Sie bitte in folgenden Schritten vor:

- ▶ Legen Sie eine POE mit SPS-Typ SH03_30 und Prozessortyp BM4_O_PLC01 an. Diese POE soll später in einer Kaltstart- und Warmstarttask aufgerufen werden.
- ▶ Platzieren Sie den FB CANop405_INIT_SL in dieser POE.
- ▶ Beschalten Sie den Funktionsbaustein mit Variablen vom richtigen Datentyp.
- ▶ Am Eingang us_DEVICE schließen Sie die globale Variable "us_CoSI_G_NodeID" aus dem globalen Variablenarbeitsblatt "bmaXX_Variables" an. Diese Variable wurde vom CANopen-Slave Konfigurator angelegt, ebenso der Wert der Variablen (siehe Abschnitt [▶Einstellen der Node-ID◀](#) auf Seite 32).
- ▶ Am Eingang us_BAUDRATE schließen Sie die globale Variable "us_CoSI_G_Baudrate" aus dem globalen Variablenarbeitsblatt "bmaXX_Variables" an. Diese Variable wurde vom CANopen-Slave Konfigurator angelegt, ebenso der Wert der Variablen (siehe Abschnitt [▶Einstellen der Baudrate◀](#) auf Seite 32). Die Bedeutung des Wertes dieser Variablen entnehmen Sie der nachfolgenden Tabelle:

us_Baudrate	Baudrate des Netzwerks
0	1 MBit/s
1	Reserviert
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	Reserviert
6	Reserviert
7	20 kBit/s
8 - 255	Reserviert

Beispiel: Initialisierung des Optionsmoduls CANopen-Slave

Die Beschaltung des FB CANop405_INIT_SL sieht wie folgt aus:

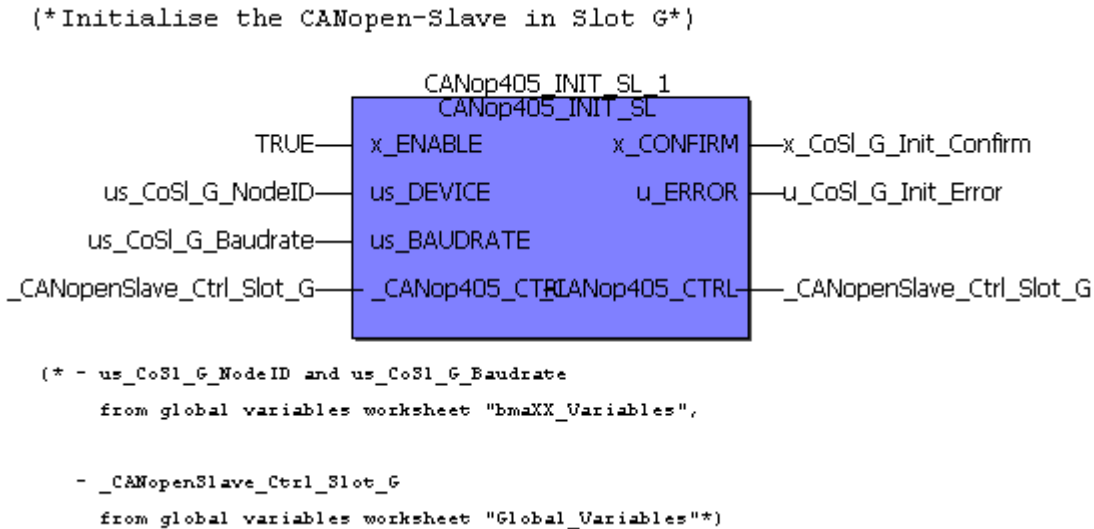


Abbildung 23: Initialisierung des CANopen-Slave mit dem FB CANop405_INIT_SL

- Legen Sie eine Task für den Kaltstart und den Warmstart der PLC an, falls diese in Ihrem Projekt noch nicht vorhanden sein sollten. Binden Sie die erstellte POE zur Initialisierung des Optionsmoduls in den beiden Tasks ein.
- Übersetzen Sie das Projekt und laden Sie es als Boot-Projekt auf die PLC. Schalten Sie das b maXX 4400 Gerät aus und wieder ein.

Der FB CANop405_INIT_SL meldet eine erfolgreiche Initialisierung mit $x_CONFIRM = 1$ und $u_ERROR = 0$.

Beispiel:

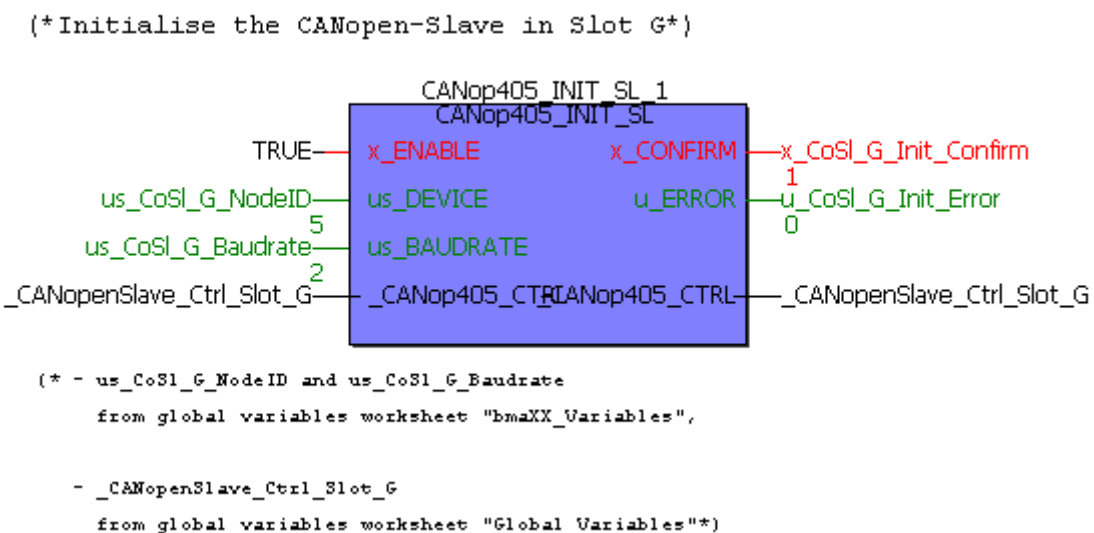


Abbildung 24: Initialisierung des CANopen-Slave mit dem FB CANop405_INIT_SL - Online

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

Eine erfolgreiche Initialisierung wird auch durch die LEDs an den Buchsen am Optionsmodul angezeigt:

H1 blinkt *) H2 aus H3 aus H4 aus	CANopen: Das Optionsmodul CANopen-Slave ist im NMT-Zustand PRE-OPERATIONAL und bereit zur Datenübertragung mittels SDOs.
H1 aus oder blinkt *) H2 blinkt *) H3 aus H4 aus	Netzwerk: Das Optionsmodul CANopen-Slave hat einen Fehler im Netzwerk erkannt. Prüfen Sie ob das Kommunikations-Kabel richtig angeschlossen ist.

*) blinkt: ca. 200ms an, dann ca. 200 ms aus

Stellen Sie die richtige Baudrate auch bei den übrigen CANopen-Teilnehmern ein. Sehen Sie dazu in der zugehörigen Dokumentation nach.

3.3.7.2 Netzwerkkommandos - NMT

Definition nach CANopen-Spezifikation

Mittels NMT- (Netzwerk-Management) Telegrammen werden durch einen CANopen-Master die Kommunikationszustände von Netzwerkknoten gesteuert. Es gibt folgende Kommunikationszustände eines Gerätes:

- INITIALISIERUNG
- PRE-OPERATIONAL
- STOPPED
- OPERATIONAL

Das Verhalten einzelner Netzwerkknoten wird durch folgende Zustandsübergänge beschrieben:

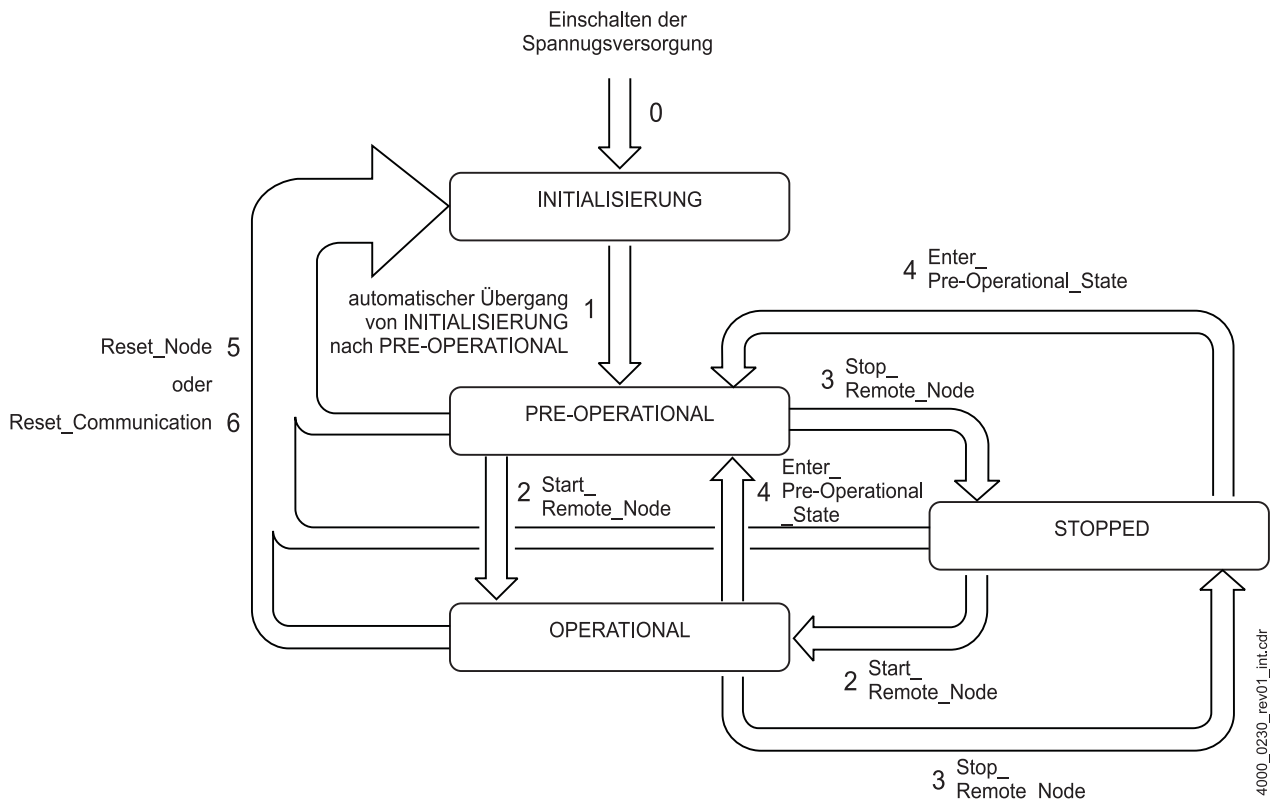


Abbildung 25: Zustandsübergänge

Direkt nach dem Einschalten befindet sich ein CANopen-Slave in dem Zustand INITIALISIERUNG und geht automatisch in den Zustand PRE-OPERATIONAL über.

Bevor der CANopen-Slave in den Zustand PRE-OPERATIONAL geht, sendet er noch sein Boot Up Telegramm. Mit dem Boot Up Telegramm meldet der CANopen-Slave auf dem CAN-Bus seine Bereitschaft zur Konfiguration über SDOs. Der Anwender hat keinen Einfluss auf das Senden des Boot Up Telegramms.

Im Zustand PRE-OPERATIONAL kann ein CANopen-Slave über SDOs konfiguriert werden. Der Datenaustausch über PDOs ist nicht möglich.

Im Zustand STOPPED sind Node Guarding und Netzwerkmanagement möglich. Weder SDOs noch PDOs können gesendet oder empfangen werden.

Befindet sich der CANopen-Slave im Zustand OPERATIONAL können SDOs, PDOs, NMT-, EMCY- und Node-Guarding- oder Heartbeat-Telegramme gesendet bzw. empfangen werden.

Die einzelnen Zustandsübergänge werden von einem NMT-Master initiiert. Lediglich der Wechsel von INITIALISIERUNG nach PRE-OPERATIONAL erfolgt beim CANopen-Slave automatisch und bedarf keines NMT-Kommandos.

Das Optionsmodul CANopen-Slave kann folgende NMT-Kommandos verarbeiten (außer 1):

- 1 automatischer Übergang von INITIALISIERUNG nach PRE-OPERATIONAL
- 2 Start_Remote_Node
- 3 Stop_Remote_Node
- 4 Enter_Pre-Operational_State

- 5 Reset_Node
- 6 Reset_Communication

Zum Starten des CANopen-Slave mittels NMT-Kommandos sehen Sie bitte in der Dokumentation des NMT-Masters nach. In der Regel ist der CANopen-Master auch der NMT-Master.

Der CANopen-Slave schaltet entsprechend den (vom CANopen-Master) über NMT-Kommandos gesendeten Zustandsübergängen seinen eigenen Zustand um.

Ein CANopen-Slave kann folgende Zustände melden:

- PRE-OPERATIONAL
- STOPPED
- OPERATIONAL

Den Zuständen sind folgende Nummern zugeordnet:

Zustand	Nummer
STOPPED	4
OPERATIONAL	5
PRE-OPERATIONAL	127

NMT-Zustand ermitteln

Sie können im Applikationsprogramm auf der b maXX PLC mittels des FBs CANop405_GET_STATE_SL den NMT-Zustand des Optionsmodul CANopen-Slave ermitteln.

Dazu gehen Sie bitte in folgenden Schritten vor

- Legen Sie eine POE mit SPS-Typ SH03_30 und Prozessortyp BM4_O_PLCO1 an. Diese POE soll später in einer zyklischen Task aufgerufen werden.
- Platzieren Sie den FB CANop405_GET_STATE_SL in dieser POE.
- Beschalten Sie den Funktionsbaustein mit Variablen vom richtigen Datentyp.
- Legen Sie eine zyklische Task mit mittlerer bis geringer Priorität an, falls eine solche in Ihrem Projekt noch nicht vorhanden sein sollte. Binden Sie die erstellte POE mit dem FB CANop405_GET_STATE_SL in dieser Task ein.
- Übersetzen Sie das Projekt und laden Sie es als (Boot-)Projekt auf die PLC.
- Starten Sie das Projekt.

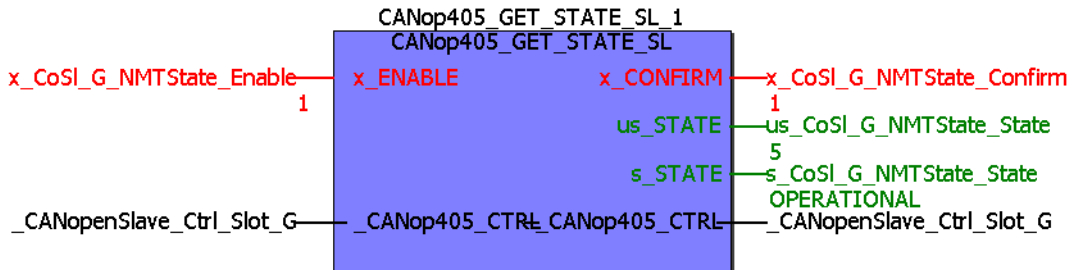
Mit x_ENABLE = 1 wird der NMT-Zustand des Optionsmodul CANopen-Slave gelesen. Der FB CANop405_GET_STATE_SL meldet eine erfolgreiche Durchführung mit x_CONFIRM = 1. Weiterhin gibt der FB die NMT-Zustandsnummer am Ausgang us_STATE aus und zeigt am Ausgang s_STATE den NMT-Zustand als Text an.

Beispiel: NMT-Zustand ermitteln

Um den FB CANop405_GET_STATE_SL verwenden zu können legen wir eine POE an und platzieren den Funktionsbaustein darin. Die erstellte POE ordnen wir einer zyklischen Task mit 200 ms Aufrufintervall zu. Nach der Beschaltung des Funktionsbausteins übersetzen Sie das Projekt und senden Sie dieses als Projekt an die b maXX PLC. Star-

ten Sie das Projekt. Mit `x_ENABLE = 1` wird der NMT-Zustand des CANopen-Slaves (in unserem Fall OPERATIONAL) gelesen:

```
(*Get the NMT-State of CANopen-Slave in Slot G*)
```



```
(* - _CANopenSlave_Ctrl_Slot_G
   from global variables worksheet "Global_Variables"*)
```

Abbildung 26: NMT-Zustand des CANopen-Slaves

3.3.7.3 Emergency-Telegramme

Definition nach CANopen-Spezifikation

Emergency-Telegramme dienen den Netzwerkknoten (CANopen-Slaves) zur Meldung von Fehlern. Die Übertragung des Emergency-Telegramms erfolgt nach dem Consumer/Producer-Kommunikationsmodell, d. h. ein Netzwerkknoten (CANopen-Slave) generiert ein Emergency-Telegramm, welches von einem oder mehreren anderen Netzwerkknoten verarbeitet wird. Eine Bestätigung des Erhalts des Telegramms erfolgt nicht. Bei jedem neu hinzukommenden Fehler wird einmalig ein neues Emergency-Telegramm gesendet. Auch wenn ein Fehler wegfällt wird ein neues Emergency-Telegramm gesendet. Eine Telegrammwiederholung erfolgt nicht.

Das Optionsmodul CANopen-Slave kann Emergency-Telegramme an andere Netzwerkknoten senden. Die Auswertung von Emergency-Telegrammen anderer Netzwerkknoten ist nicht möglich.

Der Nutzdatenbereich eines Emergency-Telegramms gliedert sich in drei Teile:

- 1 Emergency Error Code, 2 Byte
- 2 Error Register, 1 Byte
- 3 Hersteller spezifischer Fehler Code, 5 Byte

Der Emergency Error Code hat folgende Bedeutung nach CiA:

Error Code	Bedeutung
00xxh	Error Reset or No Error
10xxh	Generic Error
20xxh	Current
21xxh	Current, device input side
22xxh	Current inside the device

Error Code	Bedeutung
23xxh	Current, device output side
30xxh	Voltage
31xxh	Mains Voltage
32xxh	Voltage inside the device
33xxh	Output Voltage
40xxh	Temperature
41xxh	Ambient Temperature
42xxh	Device Temperature
50xxh	Device Hardware
60xxh	Device Software
61xxh	Internal Software
62xxh	User Software
63xxh	Data Set
70xxh	Additional Modules
80xxh	Monitoring
81xxh	Communication
8110h	CAN Overrun (Objects lost)
8120h	CAN in Error Passive Mode
8130h	Life Guard Error or Heartbeat Error
8140h	recovered from bus off
8150h	Transmit COB-ID
82xxh	Protocol Error
8210h	PDO not processed due to length error
8220h	PDO length exceeded
90xxh	External Error
F0xxh	Additional Functions
FFxxh	Device specific

xx = nicht definiert

Im Error Register ist das Objekt 1001h enthalten. Im Objekt 1001h kann ein Netzwerk-knoten interne Fehler eintragen. Die Codierung nach CiA sieht wie folgt aus:

Bit	Bedeutung
0	Generic error
1	Current
2	Voltage

Bit	Bedeutung
3	Temperature
4	Communication error
5	Device profile specific
6	Reserved
7	Manufacturer specific

Zur weiteren Charakterisierung von Fehlern stehen 5 Byte für einen Hersteller spezifischen Fehlercode zur Verfügung.

Sie können von der b maXX PLC aus über das Optionsmodul CANopen-Slave ein Emergency-Telegramm mit dem Emergency Error Code 0x6200, dem Error Register 0x80 ¹⁾ und einem beliebigen Wert für den Hersteller spezifischen Fehlercode senden.

Emergency-Telegramm senden

Sie können im Applikationsprogramm auf der b maXX PLC mittels des FBs CANop405_EMICY_SEND_SL ein Emergency-Telegramm des Optionsmodul CANopen-Slave senden.

Dazu gehen Sie bitte in folgenden Schritten vor

- Legen Sie eine POE mit SPS-Typ SH03_30 und Prozessortyp BM4_O_PLC01 an. Diese POE soll später in einer zyklischen Task aufgerufen werden.

Alternativ können Sie auch die POE in der der NMT-Zustand gelesen wird verwenden.

- Platzieren Sie den FB CANop405_EMICY_SEND_SL in dieser POE.
- Beschalten Sie den Funktionsbaustein mit Variablen vom richtigen Datentyp. Am Eingang a_MAN_SPEC_ERR_CODE geben sie den beliebigen Fehlercode an. Der Zugriff auf die 5 Bytes erfolgt über die Einträge a_MAN_SPEC_ERR_CODE[0] bis a_MAN_SPEC_ERR_CODE[4].
- Legen Sie eine zyklische Task mit mittlerer bis geringer Priorität an, falls eine solche in Ihrem Projekt noch nicht vorhanden sein sollte. Binden Sie die erstellte POE mit dem FB CANop405_GET_STATE_SL in dieser Task ein.

Wenn Sie die POE in der der NMT-Zustand gelesen wird verwenden, haben Sie diese POE bereits einer Task zugeordnet.

- Übersetzen Sie das Projekt und laden Sie es als (Boot-)Projekt auf die PLC.
- Starten Sie das Projekt.

Mit x_ENABLE = 1 wird das Emergency-Telegramm vom Optionsmodul CANopen-Slave gesendet. Der FB CANop405_EMICY_SEND_SL meldet eine erfolgreiche Durchführung mit x_CONFIRM = 1.

Beispiel: Emergency-Telegramm senden

Um den FB CANop405_EMICY_SEND_SL verwenden zu können legen wir eine POE an und platzieren den Funktionsbaustein darin. Die erstellte POE ordnen wir einer zykli-

¹⁾ Der Wert 0x80 im Error Register wird OR-verknüpft mit dem Einträgen im Objekt 0x1001 auf dem Optionsmodul CANopen-Slave.

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

schen Task mit 200 ms Aufrufintervall zu. Nach der Beschaltung des Funktionsbausteins übersetzen Sie das Projekt und senden Sie dieses als Projekt an die b maXX PLC. Starten Sie das Projekt. Mit x_ENABLE = 1 wird das Emergency-Telegramm vom Optionsmodul CANopen-Slave gesendet:

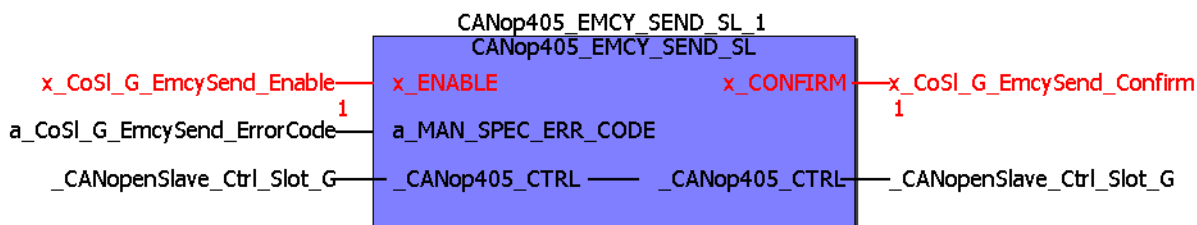
(*Send an emergency message via CANopen-Slave in Slot G*)

```

BYTE#16#12—a_CoSl_G_EmcYsend_ErrorCode[0]
16#12
BYTE#16#34—a_CoSl_G_EmcYsend_ErrorCode[1]
16#34
BYTE#16#56—a_CoSl_G_EmcYsend_ErrorCode[2]
16#56
BYTE#16#78—a_CoSl_G_EmcYsend_ErrorCode[3]
16#78
BYTE#16#9A—a_CoSl_G_EmcYsend_ErrorCode[4]
16#9A

```

(*The bytes
a_CoSl_G_EmcYsend_ErrorCode[5],
a_CoSl_G_EmcYsend_ErrorCode[6],
a_CoSl_G_EmcYsend_ErrorCode[7]
are not used!*)



(* - _CANopenSlave_Ctrl_Slot_G
from global variables worksheet "Global_Variables"*)

Abbildung 27: Emergency-Telegramm senden

3.3.7.4 Bedarfsdatenaustausch mit SDOs

Definition nach CANopen-Spezifikation

Eine SDO-Kommunikation entspricht dem Client/Server-Kommunikationsmodell, d. h. das Optionsmodul CANopen-Master ist der Client und sendet ein Telegramm an einen Netzwerkknoten mit dem Auftrag Daten zu übernehmen oder Daten zu senden. Der Netzwerkknoten agiert als Server, übernimmt die Daten und bestätigt dies mit einem Telegramm oder sendet die angeforderten Daten.

Das Optionsmodul CANopen-Slave ist solch ein Netzwerkknoten und agiert demzufolge als Server der Daten. Mit dem Optionsmodul CANopen-Slave kann der "expedited" SDO-Transfer nach DS 301 mit bis zu 4 Byte Daten pro Auftrag durchgeführt werden. Der "segmented" und "Block" SDO-Transfer nach DS 301 werden nicht unterstützt.

Objekte über SDO schreiben und lesen

Der Bedarfsdatenaustausch beschränkt sich auf der PROPROG wt II Seite auf das Lesen und Schreiben von globalen Variablen, die Sie unter [Anlegen von Variablen](#) auf Seite 34 mit dem CANopen-Slave Konfigurator angelegt haben.

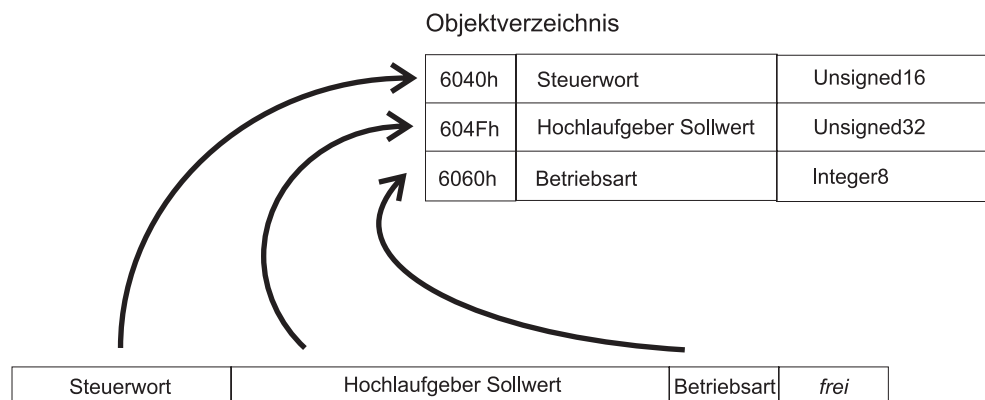
Im Beispiel wurde die Variable "ud_MyReadVar2" im CANopen-Slave Konfigurator angelegt. Dort wurde auch die Zuordnung zum Objekt-Index A200h und Objekt-Subindex 2 getroffen. Jetzt kann ein CANopen-Master über den Objekt-Index und -Subindex auf das Objekt lesend zugreifen und damit den Inhalt der Variable "ud_MyReadVar2" lesen. Da die Variable "ud_MyReadVar2" mit dem Startwert 2000 (Initialwert) vorbelegt wurde, muss im CANopen-Master beim Lesen der Variablen der Wert 2000 angezeigt werden.

Weiterhin wurde die Variable "u_MyWriteVar2" im CANopen-Slave Konfigurator angelegt. Dort wurde auch die Zuordnung zum Objekt-Index A580h und Objekt-Subindex 1 getroffen. Jetzt kann ein CANopen-Master über den Objekt-Index und -Subindex auf das Objekt schreibend zugreifen und damit den Inhalt der Variable "u_MyWriteVar2" schreiben.

3.3.7.5 Prozessdatenaustausch mit PDOs

Definition nach CANopen-Spezifikation

Eine PDO-Kommunikation (PDO = Process Data Object) entspricht dem Consumer/Producer-Kommunikationsmodell, d. h. ein Netzwerkknoten generiert ein PDO-Telegramm, welches von einem oder mehreren anderen Netzwerkknoten verarbeitet wird. Eine Bestätigung des Erhalts des Telegramms erfolgt nicht. Auch das Optionsmodul CANopen-Slave kann PDO-Telegramme generieren und verarbeiten. Die Übertragung selbst muss konfiguriert werden. So müssen Sie insbesondere festlegen, wann die Daten in einem PDO übernommen werden sollen, bzw. zu welchem Zeitpunkt die Daten in ein PDO eingetragen werden sollen. Im Gegensatz zum Datenaustausch über SDOs wird bei PDOs jedoch nicht direkt auf ein Objekt eines Netzwerkknotens zugegriffen. Vielmehr wird einem Netzwerkknoten (während der Konfigurierung durch den CANopen-Master) mitgeteilt, welche Objekte mit Index, Subindex und Datenlänge in einem PDO-Telegramm enthalten sind. Diese Objekte werden dann beim Schreiben eines PDO aus diesem entnommen und beim Lesen in das PDO eingetragen. Es stehen bis zu 8 Byte Daten pro Telegramm zur Verfügung.



PDO-Telegramm - Objekte lesen

Abbildung 28: Objekte aus einem PDO-Telegramm lesen

Im PDO-Telegramm selbst sind nur noch Nutzdaten enthalten, eine Information über Objekt-Index oder -Subindex ist nicht mehr vorhanden, denn diese ist dem Netzwerkknoten bereits durch die zuvor durchgeführte Konfiguration bekannt. Dieses Verfahren, Abbilden von Objekten in ein PDO, wird auch "Mapping" genannt. Auf dem Optionsmodul CANo-

pen-Slave ist bereits ein Default-Mapping hinterlegt, das jedoch durch den CANopen-Master geändert und gespeichert werden kann.



HINWEIS

Das Mapping wird vom CANopen-Master eingestellt, falls das Default-Mapping nicht verwendet werden soll.

Der Datenaustausch über PDOs lässt sich in drei Phasen einteilen:

- Konfiguration des Übertragungsverhaltens eines PDO auf jedem Netzwerkknoten
- Konfiguration des PDO-Mapping auf jedem Netzwerkknoten
- Zyklischer Datenaustausch über PDOs - Objekte über PDO senden und empfangen

Die Konfiguration der PDOs wird in Kommunikationsprofil spezifischen Objekten eines Gerätes abgelegt. Diese Objekte werden über SDOs gelesen und geschrieben (siehe Kapitel [►Bedarfsdatenaustausch mit SDOs◄](#) auf Seite 52). Die Konfiguration von PDOs darf nur im Zustand PRE-OPERATIONAL eines Netzwerkknotens erfolgen.

Nach der Konfiguration der Übertragungseigenschaften und des Mappings des PDO kann mit dem eigentlichen Datenaustausch über PDOs begonnen werden. Das Optionsmodul CANopen-Slave empfängt PDOs, die von den anderen Netzwerkknoten gesendet werden und reagiert entsprechend seinen Einstellungen und dem Applikationsprogramm auf der b maXX PLC. Ebenso kann das Optionsmodul CANopen-Slave PDOs senden, welche von den anderen Netzwerkknoten empfangen werden. Grundsätzlich ist der Datenaustausch über PDOs nur im Zustand OPERATIONAL der Netzwerkknoten möglich. Im Zustand OPERATIONAL ist keine Konfiguration der PDOs mehr möglich.

Beachten Sie bitte, dass auch der Datenaustausch über PDOs immer aus der Sicht der Netzwerkknoten zu betrachten ist, d. h. sendet der CANopen-Slave ein PDO, so ist dies auf der Slave-Seite als auch auf der Master-Seite ein Sende-PDO (Transmit-PDO, TxPDO). Empfängt der CANopen-Slave ein PDO, so ist dies auf der Slave-Seite als auch auf der Master-Seite ein Empfangs-PDO (Receive-PDO, RxPDO).

Die Konfiguration der Übertragungseigenschaften eines PDO

Die Eigenschaften zur Übertragung eines PDO werden durch Kommunikationsprofil spezifische Objekte eines Netzwerkknotens beschrieben. Im Nachfolgenden wird hierfür jeweils das Konfigurationsobjekt für das erste Empfangs-PDO (1400h) und das erste Sende-PDO (1800h) beschrieben. Zur Konfiguration aller weiteren PDOs ist der nächst höhere Objektindex zu verwenden, d. h. das zweite Empfangs-PDO wird über das Objekt 1401h konfiguriert, das dritte Empfangs-PDO über das Objekt 1402h, etc.. Für die Sende-PDOs sind das die Objekte 1801h, 1802h, etc..

Nach dem Kommunikationsprofil DS 301 sind maximal 512 Empfangs-PDOs und 512 Sende-PDOs möglich.

Das Optionsmodul CANopen-Slave unterstützt maximal 4 Empfangs-PDOs und 4 Sende-PDOs.

Demzufolge sind die

- Empfangs-PDOs 1, 2, 3 und 4 mit den Objekten 1400h, 1401h, 1402h und 1403h sowie die
- Sende-PDOs 1, 2, 3 und 4 mit den Objekten 1800h, 1801h, 1802h und 1803h

zu konfigurieren.

Die Werte der Objekte haben nach Kommunikationsprofil DS 301 folgende Bedeutung:

Empfangs-PDOs

Objekt	Bedeutung
1400h (- 15FF)	Kommunikationsparameter für das erste Empfangs-PDO eines Gerätes.
Subindex 0	Nummer des Subindex, welcher den letzten gültigen Eintrag enthält
Subindex 1	COB-ID des PDO und Gültigkeit
Subindex 2	Übertragungstyp des PDO: synchron oder asynchron
Subindex 3	<i>Nicht verwendet</i>
Subindex 4	Eintrag zur Kompatibilität. Optional.
Subindex 5	<i>Nicht verwendet.</i>

Sende-PDOs

Objekt	Bedeutung
1800h (- 19FF)	Kommunikationsparameter für das erste Sende-PDO eines Gerätes.
Subindex 0	Nummer des Subindex, welcher den letzten gültigen Eintrag enthält
Subindex 1	COB-ID des PDO und Gültigkeit
Subindex 2	Übertragungstyp des PDO: synchron, asynchron, Ereignis gesteuert.
Subindex 3	Sendeverzögerungszeit des PDO. Optional
Subindex 4	Eintrag zur Kompatibilität. Optional.
Subindex 5	Timerwert, falls die PDO-Übertragung Zeit gesteuert erfolgt. Optional.

Betrachten wir die Bedeutung der Subindizes näher.

Objekt 1400h / 1800h Subindex 0 - Anzahl der gültigen Subindizes

Der Subindex 0 gibt die Nummer des letzten gültigen Subindex an. Der Wert ist mindestens 2 und erhöht sich entsprechend der optional zu unterstützenden weiteren Einträge (Subindizes). Der Wert ist im Allgemeinen "read only", also für Sie nur zu Informationszwecken interessant.

Objekt 1400h / 1800h Subindex 1 - COB-ID des PDO

Der Subindex 1 enthält verschiedene Informationen. Der Wert hat den Datentyp Unsigned32 und teilt sich wie folgt auf:

Bit	Wert / Bedeutung
31 (MSB)	0: PDO existiert und ist gültig 1: PDO existiert nicht oder ist nicht gültig Nach der vordefinierten Identifizierung (siehe Kapitel ▷ Datenaustausch und Objekte des physikalischen Bussystems ab Seite 18) sind nur 4 PDOs für Schreiben und 4 PDOs für Lesen möglich. Unterstützt ein Gerät mehr als diese 4 PDOs, so müssen Sie in der Regel die COB-IDs für die zusätzlichen PDOs selbst vergeben (Bits 10 - 0 in diesem Subindex). Aus diesem Grund hat das Bit 31 der zusätzlichen PDOs meist den Wert 1 und wird erst dann vom Anwender auf 0 gesetzt, sobald dieser eine COB-ID vergeben hat.
30	0: das PDO kann über einen Remote-Request angefordert werden 1: das PDO kann nicht über einen Remote-Request angefordert werden
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Slave nicht unterstützt)
28 - 11	wird vom Optionsmodul CANopen-Slave nicht unterstützt
10 - 0 (LSB)	COB-ID des PDO. Die ersten 4 PDOs für Schreiben und Lesen haben in den Default-Einstellungen eines Gerätes die COB-IDs nach der vordefinierten Identifizierung (siehe Kapitel ▷ Datenaustausch und Objekte des physikalischen Bussystems ab Seite 18)

Benötigen Sie nicht mehr als 4 PDOs für Schreiben und 4 PDOs für Lesen, so ist dieser Wert meist für Sie nur zu Informationszwecken interessant.

Objekt 1400h / 1800h Subindex 2 - Übertragungsart des PDO

Dieser Wert ist sehr wichtig für Sie, denn er gibt an, wann ein PDO übertragen wird. Die Art der Übertragung wird in Subindex 2 konfiguriert. Dabei ist folgende Zuordnung für Sende- und Empfangs-PDOs gültig:

Subindex 2 / Typ	Wirkung Sende-PDO 1800h	Empfangs-PDO 1400h
0 Synchron	Senden erfolgt nach Empfang eines SYNC-Telegramms, aber nur dann, wenn sich die zu sendenden Werte geändert haben. ^{a)}	Vor dem letzten SYNC-Telegramm empfangenes PDO mit passender COB-ID wird übernommen
1 - 240 Synchron	Senden erfolgt nach Empfang der eingestellten Anzahl von SYNC-Telegramm. ^{a)}	Vor dem letzten SYNC-Telegramm empfangenes PDO mit passender COB-ID wird übernommen
252 Remote	Bei Empfang eines SYNC-Telegramms erfolgt ein Update des PDO. Das Senden erfolgt erst nach Empfang einer Remote-Anforderung. ^{a)}	<i>Nicht möglich</i>

Subindex 2 / Typ	Wirkung Sende-PDO 1800h	Empfangs-PDO 1400h
253 Remote	Ein Update und das Senden erfolgt nach Empfang einer Remote-Anforderung.	<i>Nicht möglich</i>
254 Asynchron	Senden erfolgt Hersteller spezifisch. Anmerkung: Die meisten Geräte übertragen bei diesem Typ das PDO bei einem Zeit-Event (Ablauf eines Timers). Die Zeit wird unter Subindex 5 eingestellt. ^{b)}	Jedes PDO mit passender COB-ID wird bei Empfang übernommen
255 Asynchron	Senden erfolgt abhängig vom unterstützten Geräteprofil. Anmerkung: Die meisten Geräte übertragen bei diesem Typ das PDO sobald sich einer der gemappten Werte geändert hat. ^{c)}	Jedes PDO mit passender COB-ID wird bei Empfang übernommen

- a) Siehe Kapitel [►Synchronisieren des Datenaustausch◄](#) auf Seite 67.
- b) Zeit gesteuertes Senden bedeutet, die Sendebedingung ist an einen Timer gebunden. Dieser Timer wird für das erste Sende-PDO mittels Subindex 5 im Objekt 1800h (16 Bit) eingestellt. Die Auflösung beträgt Millisekunden. Der Timer wird beim Zustandswechsel nach OPERATIONAL gestartet. Das Senden des PDO erfolgt dann zyklisch mit der im Timer eingestellten Zeit. Der Timer wird gelöscht, indem auf Subindex 5 der Wert "0" geschrieben wird. Zeit gesteuertes Empfangen existiert nicht, sondern alle PDOs mit passender COB-ID werden übernommen.
- c) Ereignis gesteuertes Senden bedeutet, die Sendebedingung ist an die Änderung eines Wertes eines der gemappten Objekte gebunden. Sind beispielsweise 3 Objekte gemappt, wird das PDO gesendet, sobald sich mindestens einer der drei Werte ändert. Bleiben die Werte konstant, wird kein PDO gesendet. Dadurch läßt sich die Buslast verringern (Telegramme werden nur übertragen, wenn sie neue Informationen enthalten). Ereignis gesteuertes Empfangen heißt, alle PDOs mit passender COB-ID werden übernommen.

Objekt 1400h / 1800h Subindex 3 - Sendeverzögerungszeit des PDO

Mit dem Subindex 3 kann eine Sendeverzögerung eingestellt werden, welche die Reaktionszeit bei der relativ ersten Wertänderung nicht verlängert, aber bei unmittelbar darauf folgenden Änderungen aktiv ist. Die Inhibit-Zeit (Sendeverzögerungszeit) beschreibt die Zeitspanne, die zwischen dem Versenden zweier gleicher Telegramme mindestens abgewartet werden muss. Der Wert hat für Empfangs-PDOs keine Bedeutung.

Objekt 1400h / 1800h Subindex 4

Der Wert des Subindex 4 hat keine relevante Bedeutung.

Objekt 1400h / 1800h Subindex 5 - Zeit für Ereignis gesteuerte PDO Übertragung

Der Wert des Subindex 5 gibt die Zeit in ms für das Senden des PDO an (siehe Subindex 2). Der Wert hat für Empfangs-PDOs keine Bedeutung.

Übertragungseigenschaften der PDOs - Defaulteinstellung

Das Optionsmodul CANopen-Slave ist defaultmäßig wie folgt konfiguriert:

Objekt 1400	Wert
Subindex 0	2
Subindex 1	200 + Node-ID
Subindex 2	255

D. h. dass das RxPDO 1 den Identifier "200 + Node-ID des Optionsmodul CANopen-Slave" haben muss und dass dieses PDO asynchron ausgewertet wird (das SYNC-Telegramm hat keinen Einfluss).

Objekt 1401	Wert
Subindex 0	2
Subindex 1	300 + Node-ID
Subindex 2	255

D. h. dass das RxPDO 1 den Identifier "300 + Node-ID des Optionsmodul CANopen-Slave" haben muss und dass dieses PDO asynchron ausgewertet wird (das SYNC-Telegramm hat keinen Einfluss).

Objekt 1402	Wert
Subindex 0	2
Subindex 1	400 + Node-ID
Subindex 2	255

D. h. dass das RxPDO 1 den Identifier "400 + Node-ID des Optionsmodul CANopen-Slave" haben muss und dass dieses PDO asynchron ausgewertet wird (das SYNC-Telegramm hat keinen Einfluss).

Objekt 1403	Wert
Subindex 0	2
Subindex 1	500 + Node-ID
Subindex 2	255

D. h. dass das RxPDO 1 den Identifier "500 + Node-ID des Optionsmodul CANopen-Slave" haben muss und dass dieses PDO asynchron ausgewertet wird (das SYNC-Telegramm hat keinen Einfluss).

Objekt 1800	Wert
Subindex 0	5
Subindex 1	180 + Node-ID
Subindex 2	255
Subindex 3	0
Subindex 4	0
Subindex 5	0

D. h. dass das TxPDO 1 den Identifier "180 + Node-ID des Optionsmodul CANopen-Slave" hat und dass dieses PDO asynchron gesendet wird (das SYNC-Telegramm hat keinen Einfluss). Das Senden erfolgt bei der jeder Änderung eines Objektes, das in dieses TxPDO gemappt ist.

Objekt 1801	Wert
Subindex 0	5
Subindex 1	280 + Node-ID
Subindex 2	255
Subindex 3	0
Subindex 4	0
Subindex 5	0

D. h. dass das TxPDO 1 den Identifier "280 + Node-ID des Optionsmodul CANopen-Slave" hat und dass dieses PDO asynchron gesendet wird (das SYNC-Telegramm hat keinen Einfluss). Das Senden erfolgt bei der jeder Änderung eines Objektes, das in dieses TxPDO gemappt ist.

Objekt 1802	Wert
Subindex 0	5
Subindex 1	380 + Node-ID
Subindex 2	255
Subindex 3	0
Subindex 4	0
Subindex 5	0

D. h. dass das TxPDO 1 den Identifier "380 + Node-ID des Optionsmodul CANopen-Slave" hat und dass dieses PDO asynchron gesendet wird (das SYNC-Telegramm hat keinen Einfluss). Das Senden erfolgt bei der jeder Änderung eines Objektes, das in dieses TxPDO gemappt ist.

Objekt 1804	Wert
Subindex 0	5
Subindex 1	480 + Node-ID
Subindex 2	255
Subindex 3	0
Subindex 4	0
Subindex 5	0

D. h. dass das TxPDO 1 den Identifier "480 + Node-ID des Optionsmodul CANopen-Slave" hat und dass dieses PDO asynchron gesendet wird (das SYNC-Telegramm hat keinen Einfluss). Das Senden erfolgt bei der jeder Änderung eines Objektes, das in dieses TxPDO gemappt ist.

Übertragungseigenschaften der PDOs - Beispieleinstellung

In unserem Beispiel wurde konfiguriert,

- dass die Objekte A200Sub1 und A200Sub2 von der b maXX PLC an das Optionsmodul CANopen-Slave übergeben werden und an den CANopen-Master zu senden sind (→ ro (nur lesbar); siehe "Anlegen von Variablen", Variable "ud_MyReadVar1" und "ud_MyReadVar2")
- dass die Objekte A680Sub1 und A580Sub1 vom CANopen-Master zu empfangen sind und vom Optionsmodul CANopen-Slave an die b maXX PLC übergeben werden (→ rw (schreib- u. lesbar); siehe "Anlegen von Variablen", Variable "ud_MyWriteVar1" und u_MyWriteVar2")

Wir werden die Objekte A200Sub1 und A200Sub2 in der TxPDO1 an den CANopen-Master senden und die Objekte A680Sub1 und A580Sub1 in der RxPDO1 vom CANopen-Master empfangen.

Für unser Beispiel werden die Default-Einstellungen der Übertragungseigenschaften der TxPDO1 (im Objekt 1400) und der RxPDO1 (im Objekt 1800) nicht verändert, d. h. Sie brauchen diese Objekte nicht vom CANopen-Master über "SDO schreiben" zu ändern.

Die Übertragungseigenschaften der TxPDO1, TxPDO2 und TxPDO3 (Objekte 1401, 1402 und 1403) sowie der RxPDO1, RxPDO2 und RxPDO3 (Objekte 1801, 1802 und 1803) sind für unser Beispiel nicht relevant, da diese PDOs nicht genutzt werden.

Damit ergibt sich mit der Node-ID 5 folgende Einstellung:

Objekt 1400	Wert
Subindex 0	2
Subindex 1	205
Subindex 2	255

D. h. dass das RxPDO 1 den Identifier "205" hat und dass dieses PDO asynchron ausgewertet wird (das SYNC-Telegramm hat keinen Einfluss).

Objekt 1800	Wert
Subindex 0	5
Subindex 1	185
Subindex 2	255
Subindex 3	0
Subindex 4	0
Subindex 5	0

D. h. dass das TxPDO 1 den Identifier "185" hat und dass dieses PDO asynchron gesendet wird (das SYNC-Telegramm hat keinen Einfluss). Das Senden erfolgt bei der jeder Änderung eines Objektes, das in dieses TxPDO gemappt ist.

Das Mapping von Objekten in einem PDO

Als zweiter Konfigurationsschritt für den Datenaustausch über PDOs ist das Mapping der Objekte in einem PDO festzulegen. Auch für das Mapping gibt es spezielle, Kommunikationsprofil spezifische Objekte, in welchen dies für einen Netzwerkknoten beschrieben ist.

Im Nachfolgenden wird hierfür jeweils das Konfigurationsobjekt für das erste Empfangs-PDO (1600h) und das erste Sende-PDO (1A00h) beschrieben. Zur Konfiguration aller weiteren PDOs ist der nächst höhere Objektindex zu verwenden, d. h. das Mapping des zweiten Empfangs-PDO wird über das Objekt 1601h konfiguriert, das dritte Empfangs-PDO über das Objekt 1602h etc..

Das Optionsmodul CANopen-Slave unterstützt maximal 4 Empfangs-PDOs und 4 Sende-PDOs.

Demzufolge muss das Mapping für die

- Empfangs-PDOs 1, 2, 3 und 4 mit den Objekten 1600h, 1601h, 1602h und 1603h

sowie für die

- Sende-PDOs 1, 2, 3 und 4 mit den Objekten 1A00h, 1A01h, 1A02h und 1A03h

festgelegt werden.

Im Detail haben die Objekte nach DS 301 folgende Bedeutung:

Empfangs-PDOs

Objekt	Bedeutung
1600h (- 17FF)	Mapping-Parameter für das erste Empfangs-PDO eines Gerätes.
Subindex 0	Anzahl der gemappten Objekte
Subindex 1 - Subindex 64	Informationen zum gemappten Objekt

Sende-PDOs

Objekt	Bedeutung
1A00h (- 1BFF)	Mapping-Parameter für das erste Sende-PDO eines Gerätes.
Subindex 0	Anzahl der gemappten Objekte
Subindex 1 - Subindex 64	Informationen zum gemappten Objekt

Betrachten wir die Bedeutung der Subindizes näher.

Objekt 1600h / 1A00h Subindex 0 - Anzahl der gemappten Objekte

Der Subindex 0 gibt die Anzahl der in diesem PDO gemappten Objekte an. Sobald Sie das Mapping eines PDO ändern wollen, müssen Sie zunächst die Anzahl der gemappten Objekte auf Null setzen. Dann können Sie das Mapping ändern und anschließend stellen Sie die Zahl der gemappten Objekte in Subindex 0 ein. Beachten Sie bitte, dass hier Geräte abhängig Einschränkungen bei der Anzahl der zu mappenden Objekte bestehen können.

Das Optionsmodul CANopen-Slave unterstützt maximal 8 gemappte Objekte pro PDO.

Objekt 1600h / 1A00h Subindex 1 bis 64 - Mapping Informationen

Die Subindizes 1 bis 64 geben die Mapping Informationen der Objekte in einem PDO wieder. Subindex 1 für das erste gemappte Objekt, Subindex 2 für das zweite gemappte Objekt usw.. Der Datentyp dieser Subindizes ist Unsigned32. Der Inhalt hat folgende Bedeutung:

Bit 31 - 16	Bit 15 - 8	Bit 7 - 0
Index des gemappten Objektes	Subindex des gemappten Objektes	Anzahl der Bit des gemappten Objektes

Die Objekte werden nacheinander in ein PDO gemappt.



HINWEIS

Beachten Sie, dass Sie unter [►Anlegen von Variablen◄](#) ab Seite 34 festlegen welche Objekte in PDOs mappbar sind.

Für das Optionsmodul CANopen-Slave ist ein Default-Mapping definiert. Sollten diese Default-Mapping-Werte ausreichen, brauchen Sie hier keine weiteren Einstellungen vornehmen. D. h. Sie brauchen das Default-Mapping nicht (vom CANopen-Master aus über "SDO schreiben") ändern.

Mappingeigenschaften der PDOs - Defaulteinstellung

Auf dem Optionsmodul CANopen-Slave ist folgendes Mapping defaultmäßig eingestellt:

RxPDOs:

Objekt 1600	Wert
Subindex 0	5
Subindex 1	A4800108h
Subindex 2	A4C00108h
Subindex 3	A5400110h
Subindex 4	A5800110h
Subindex 5	A5800210h

D. h. dass das RxPDO 1 fünf Objekte enthält, die in die Objekteinträge:
 Objekt-Index A480, Objekt-Subindex 01, Anzahl der Bits: 8 (Datentyp: INTEGER8);
 Objekt-Index A4C0, Objekt-Subindex 01, Anzahl der Bits: 8 (Datentyp: UNSIGNED8);
 Objekt-Index A540, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: INTEGER16);
 Objekt-Index A580, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: UNSIGNED16);
 Objekt-Index A580, Objekt-Subindex 02, Anzahl der Bits: 16 (Datentyp: UNSIGNED16);
 eingetragen werden.

Objekt 1601	Wert
Subindex 0	2
Subindex 1	A6400120h
Subindex 2	A6800120h

D. h. dass das RxPDO 2 zwei Objekte enthält, die in die Objekteinträge:
 Objekt-Index A640, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: INTEGER32);
 Objekt-Index A680, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: UNSIGNED32);
 eingetragen werden.

Objekt 1602	Wert
Subindex 0	4
Subindex 1	A4800108h
Subindex 2	A4C00108h
Subindex 3	A5400110h
Subindex 4	A6800120h

D. h. dass das RxPDO 3 vier Objekte enthält, die in die Objekteinträge:
 Objekt-Index A480, Objekt-Subindex 01, Anzahl der Bits: 8 (Datentyp: INTEGER8);
 Objekt-Index A4C0, Objekt-Subindex 01, Anzahl der Bits: 8 (Datentyp: UNSIGNED8);
 Objekt-Index A540, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: INTEGER16);
 Objekt-Index A680, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: UNSIGNED32);
 eingetragen werden.

Objekt 1603	Wert
Subindex 0	3
Subindex 1	A5400110h
Subindex 3	A5800110h
Subindex 4	A6800120h

D. h. dass das RxPDO 4 drei Objekte enthält, die in die Objekteinträge:
 Objekt-Index A540, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: INTEGER16);
 Objekt-Index A580, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: UNSIGNED16);
 Objekt-Index A680, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: UNSIGNED32);
 eingetragen werden.

TxPDOs:

Objekt 1A00	Wert
Subindex 0	5
Subindex 1	A0000108h
Subindex 2	A0400108h
Subindex 3	A0C00110h
Subindex 4	A1000110h
Subindex 5	A1000210h

D. h. dass das TxPDO 1 fünf Objekte enthält, die den Objekteinträgen:
 Objekt-Index A000, Objekt-Subindex 01, Anzahl der Bits: 8 (Datentyp: INTEGER8);
 Objekt-Index A040, Objekt-Subindex 01, Anzahl der Bits: 8 (Datentyp: UNSIGNED8);
 Objekt-Index A0C0, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: INTEGER16);
 Objekt-Index A100, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: UNSIGNED16);
 Objekt-Index A100, Objekt-Subindex 02, Anzahl der Bits: 16 (Datentyp: UNSIGNED16);
 entnommen werden.

Objekt 1A01	Wert
Subindex 0	2
Subindex 1	A1C00120h
Subindex 2	A2000120h

D. h. dass das TxPDO 2 zwei Objekte enthält, die den Objekteinträgen:
 Objekt-Index A1C0, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: INTEGER32);
 Objekt-Index A200, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: UNSIGNED32);
 entnommen werden.

Objekt 1A02	Wert
Subindex 0	4
Subindex 1	A0000108h
Subindex 2	A0400108h
Subindex 3	A0C00110h
Subindex 4	A2000120h

D. h. dass das TxPDO 3 vier Objekte enthält, die den Objekteinträgen:
 Objekt-Index A000, Objekt-Subindex 01, Anzahl der Bits: 8 (Datentyp: INTEGER8);
 Objekt-Index A040, Objekt-Subindex 01, Anzahl der Bits: 8 (Datentyp: UNSIGNED8);
 Objekt-Index A0C0, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: INTEGER16);
 Objekt-Index A200, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: UNSIGNED32);
 entnommen werden.

Objekt 1A03	Wert
Subindex 0	3
Subindex 1	A0C00110h
Subindex 3	A1000110h
Subindex 4	A2000120h

D. h. dass das TxPDO 4 drei Objekte enthält, die den Objekteinträgen:
 Objekt-Index A0C0, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: INTEGER16);
 Objekt-Index A100, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: UNSIGNED16);
 Objekt-Index A200, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: UNSIGNED32);
 entnommen werden.

Mappingeigenschaften der PDOs - Beispielerstellung

In unserem Beispiel wurde konfiguriert,

- dass die Objekte A200Sub1 und A200Sub2 von der b maXX PLC an das Optionsmodul CANopen-Slave übergeben werden und an den CANopen-Master zu senden sind (→ ro (nur lesbar); siehe "Anlegen von Variablen", Variable "ud_MyReadVar1" und "ud_MyReadVar2")
- dass die Objekte A680Sub1 und A580Sub1 vom CANopen-Master zu empfangen sind und vom Optionsmodul CANopen-Slave an die b maXX PLC übergeben werden (→ rw (schreib- u. lesbar); siehe "Anlegen von Variablen", Variable "ud_MyWriteVar1" und u_MyWriteVar2")

Wir werden die Objekte A200Sub1 und A200Sub2 an den CANopen-Master senden und die Objekte A680Sub1 und A580Sub1 vom CANopen-Master empfangen.

Im Default-Mapping für die TxPDOs finden wir keine TxPDO die die Objekte A200Sub1 und A200Sub2 enthält. Deshalb müssen die Mappingeigenschaften einer TxPDO (wir nehmen TxPDO1) geändert werden.

Im Default-Mapping für die RxPDOs finden wir die RxPDO4, die u. a. die Objekte A580Sub1 und A680Sub1 enthält. Wir verwenden die RxPDO4 und brauchen die Mappingeigenschaften nicht ändern.

Wir müssen nur beachten, dass in der RxPDO4 auch der Wert für das Objekt A540Sub1 gesendet wird und das Objekt A580Sub1 nicht in Daten-Byte 0 und 1 sondern in Daten-Byte 3 und 4 auf dem CANopen-Master eingetragen werden muss.

Dem Objekt A540Sub1 haben wird im CANopen-Slave Konfigurator keine Variable zugeordnet und verwenden somit den Inhalt des Objektes nicht im PROPROGRAMM II Projekt.

Sie ändern die Mappingeigenschaften der TxPDO1(Objekt 1A00) indem Sie dieses Objekt vom CANopen-Master aus über "SDO schreiben" (nach jedem Einschalten des CANopen-Slaves) ändern. Die Werte (Reihenfolge beachten!) entnehmen sie bitte der Tabelle.

SDO schreiben			
Objekt	Subindex	Wert	Bemerkung
1A00	0	0	Einträge Löschen
1A00	1	16#A2000120	1. Objekt: Objekt 16#A200, Subindex 16#01, Länge 16#20 (32 Bit)
1A00	2	16#A2000220	2. Objekt: Objekt 16#A200, Subindex 16#02, Länge 16#20 (32 Bit)
1A00	0	2	2 Einträge für TxPDO1

Damit ergibt sich folgende Einstellung:

Objekt 1603	Wert
Subindex 0	3
Subindex 1	A5400110h
Subindex 3	A5800110h
Subindex 4	A6800120h

D. h. dass das RxPDO 4 drei Objekte enthält, die in die Objekteinträge:
Objekt-Index A540, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: INTEGER16);
Objekt-Index A580, Objekt-Subindex 01, Anzahl der Bits: 16 (Datentyp: UNSIGNED16);
Objekt-Index A680, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: UNSIGNED32);
eingetragen werden.

TxPDOs:

Objekt 1A00	Wert
Subindex 0	2
Subindex 1	A2000120h
Subindex 2	A2000220h

D. h. dass das TxPDO 1 zwei Objekte enthält, die den Objekteinträgen:
Objekt-Index A200, Objekt-Subindex 01, Anzahl der Bits: 32 (Datentyp: UNSIGNED32);
Objekt-Index A200, Objekt-Subindex 02, Anzahl der Bits: 32 (Datentyp: UNSIGNED32);
entnommen werden.

Objekte über PDO lesen

Der Prozessdatenaustausch beschränkt sich auf der PROPROGRAM Seite auf das Lesen oder Schreiben von globalen Variablen, die Sie unter [►Anlegen von Variablen◄](#) auf Seite 34, mit dem CANopen-Slave Konfigurator angelegt und denen sie CANopen-Objekte zugeordnet haben.

Soll ein CANopen-Master über PDO Objekte lesen, so muss der CANopen-Slave dieses PDO als TxPDO senden. Dieses TxPDO beinhaltet die Objekte.

Im Applikationsprogramm auf der b maXX PLC müssen die Variablen, die diesen Objekten zugeordnet sind, mit einem Wert versehen werden.

D. h. im Applikationsprogramm auf der b maXX PLC werden die Variablen **geschrieben**.



HINWEIS

Der Prozessdatenaustausch über PDO ist nur dann möglich, wenn sich der CANopen-Slave im NMT-Zustand OPERATIONAL befindet. Siehe hierzu auch [►Netzwerkcommandos - NMT◄](#) auf Seite 46.

Im Beispiel wurde die Variable "ud_MyReadVar1" im CANopen-Slave Konfigurator angelegt. Dort wurde auch die Zuordnung zum Objekt-Index A200h und Objekt-Subindex 1 getroffen. Weiterhin wurde die Variable mit dem Initialwert 1000 versehen. Außerdem wurde "ud_MyReadVar2" (mit Objekt-Index A200h und Objekt-Subindex 2 sowie Initialwert 2000) angelegt.

Jetzt kann ein CANopen-Master das TxPDO1 des CANopen-Slaves anfordern.

Im CANopen-Master muss der Wert 16#0000_03E8_0000_07D0 angezeigt werden
(16#0000_03E8 = 1000 = Wert von "ud_MyReadVar1";
16#0000_07D0 = 2000 = Wert von "ud_MyReadVar2").

Objekte über PDO schreiben

Der Prozessdatenaustausch beschränkt sich auf der PROPROGRAM Seite auf das Lesen oder Schreiben von globalen Variablen, die Sie unter [►Anlegen von Variablen◄](#) auf Seite 34, mit dem CANopen-Slave Konfigurator angelegt und denen sie CANopen-Objekte zugeordnet haben.

Soll ein CANopen-Master über PDO Objekte schreiben, so muss der CANopen-Slave dieses PDO als RxPDO empfangen. Dieses RxPDO beinhaltet die Objekte.

Im Applikationsprogramm auf der b maXX PLC können die Variablen, die diesen Objekten zugeordnet sind, gelesen werden.

D. h. im Applikationsprogramm auf der b maXX PLC werden die Variablen **gelesen**.



HINWEIS

Der Prozessdatenaustausch über PDO ist nur dann möglich, wenn sich der CANopen-Slave im NMT-Zustand OPERATIONAL befindet. Siehe hierzu auch [▶Netzwerkcommandos - NMT](#) auf Seite 46.

Im Beispiel wurde die Variable "ud_MyWriteVar1" im CANopen-Slave Konfigurator angelegt. Dort wurde auch die Zuordnung zum Objekt-Index A680h und Objekt-Subindex 1 getroffen. Weiterhin wurde die Variable mit dem Initialwert 1500 versehen und das CANopen-Objekt auf den Wertebereich $1000 \leq x \leq 2000$ begrenzt. Weiterhin wurde die Variable "u_MyWriteVar2" (mit Objekt-Index A580 und Objekt-Subindex 1 sowie keinem Initialwert und keiner Begrenzung) angelegt.

Jetzt kann ein CANopen-Master das RxPDO4 des CANopen-Slaves schreiben.

Im CANopen-Master muss für "ud_MyWriteVar1" ein Wert zwischen 1000 und 2000 sowie für "u_MyWriteVar2" ein Wert (entsprechend dem Datentyp) zwischen 0 und 65535 geschrieben werden.

Zum Beispiel:

16#223D_xxxx = 8765 = Wert für "u_MyWriteVar2" (in Byte 3 und 4); $0 \leq 8765 \leq 65535$ (xxxx ist der Wert der auf das Objekt A540Sub1 geschrieben wird)

16#0000_04D2 = 1234 = Wert für "ud_MyWriteVar1"; $1000 \leq 1234 \leq 2000$

Der CANopen-Master kann (aber muss nicht) die Begrenzung der Objekte aus der EDS-Datei entnehmen (siehe [▶Anlegen von Variablen](#) auf Seite 34 und [▶Erzeugen EDS-Datei](#) auf Seite 42). Eine Überwachung solcher Begrenzungen muss der Anwender gegebenenfalls im Applikationsprogramm auf der b maXX PLC implementieren.

Auf dem Optionsmodul CANopen-Slave wird dann der bisherige Wert "1500" von Objekt A680Sub1 mit "1234" und der bisherige Wert "0" von Objekt A580Sub1 mit "8765" überschrieben.

Im Applikationsprogramm auf der b maXX PLC wird dadurch der bisherige Wert 1500 von "ud_MyWriteVar1" mit 1234 und der bisherige Wert 0 von "u_MyWriteVar2" mit 8765 überschrieben.

3.3.7.6 Synchronisieren des Datenaustausch

Definition nach CANopen-Spezifikation

Bei vielen Anwendungen ist es sinnvoll, die Übernahme der Eingangsdaten sowie das Senden der Ausgangsdaten zu synchronisieren. CANopen stellt hierzu das SYNC-Telegramm zur Verfügung. Das SYNC-Telegramm hat nach dem NMT-Telegramm die höchste Priorität und keine Nutzdaten. Die Übertragung des SYNC-Telegramms erfolgt nach dem Consumer/Producer-Kommunikationsmodell, d. h. ein Netzwerkknoten generiert ein

SYNC-Telegramm, welches von einem oder mehreren anderen Netzwerkknoten verarbeitet wird. Eine Bestätigung des Erhalts des Telegramms erfolgt nicht. Der Empfang dient den Netzwerkknoten als Trigger für das Lesen der empfangenen PDOs bzw. für das Senden von PDOs, vorausgesetzt ein PDO ist für diesen Übertragungstyp konfiguriert (siehe Kapitel [Die Konfiguration der Übertragungseigenschaften eines PDO](#) auf Seite 54). Die Einstellung, ob ein Netzwerkknoten Consumer oder Producer ist, erfolgt über das Objekt 1005h:

SYNC-Telegramm:

Objekt	Bedeutung
1005h	Konfiguration SYNC-Telegramm.
Subindex 0	COB-ID und Gültigkeit

Der Wert von Subindex 0 hat den Datentyp Unsigned32 und teilt sich wie folgt auf:

Bit	Wert / Bedeutung
31 (MSB)	<i>Nicht relevant</i>
30	0: Netzwerkknoten generiert kein SYNC-Telegramm 1: Netzwerkknoten generiert SYNC-Telegramme (wird vom Optionsmodul CANopen-Slave nicht unterstützt)
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Slave nicht unterstützt)
28 - 11	Wird vom Optionsmodul CANopen-Slave nicht unterstützt
10 - 0 (LSB)	COB-ID des SYNC-Telegramms. Da das Optionsmodul CANopen-Slave die COB-ID nach der vordefinierten Identifizierung (siehe Kapitel Datenaustausch und Objekte des physikalischen Bussystems ab Seite 18) verwendet muss diese 80h betragen.



HINWEIS

Das Optionsmodul CANopen-Slave kann nur als SYNC-Consumer betrieben werden. D. h. ein anderer Netzwerkknoten muss SYNC-Producer sein.

Das Optionsmodul CANopen-Slave für b maXX PLC ist per Defaulteinstellung als SYNC-Consumer mit der COB-ID 80h (gemäß der vordefinierten Identifizierung) eingestellt. D. h. das Objekt 1005h hat den Inhalt '00000080h'.

Der Empfang eines SYNC-Telegramms erzeugt auf dem Optionsmodul CANopen-Slave einen Interrupt, der an die b maXX PLC und den b maXX Regler weitergeleitet wird. Damit kann dieses Signal zur Synchronisation des b maXX 4400 verwendet werden. Innerhalb eines SYNC-Intervalls (Kommunikationszyklus) müssen alle entsprechenden Telegramme (synchrone PDOs) an alle projektierten Slaves gesendet werden. Übertragungsrate, Leitungslänge, Anzahl der Knoten, Größe der Telegramme sowie Verarbeitungszeiten auf dem Optionsmodul CANopen-Slave sind dabei zu berücksichtigen.

Das Übernehmen bzw. Übertragen der PDO-Inhalte läuft auf einem SYNC-Consumer nach folgendem Schema ab:

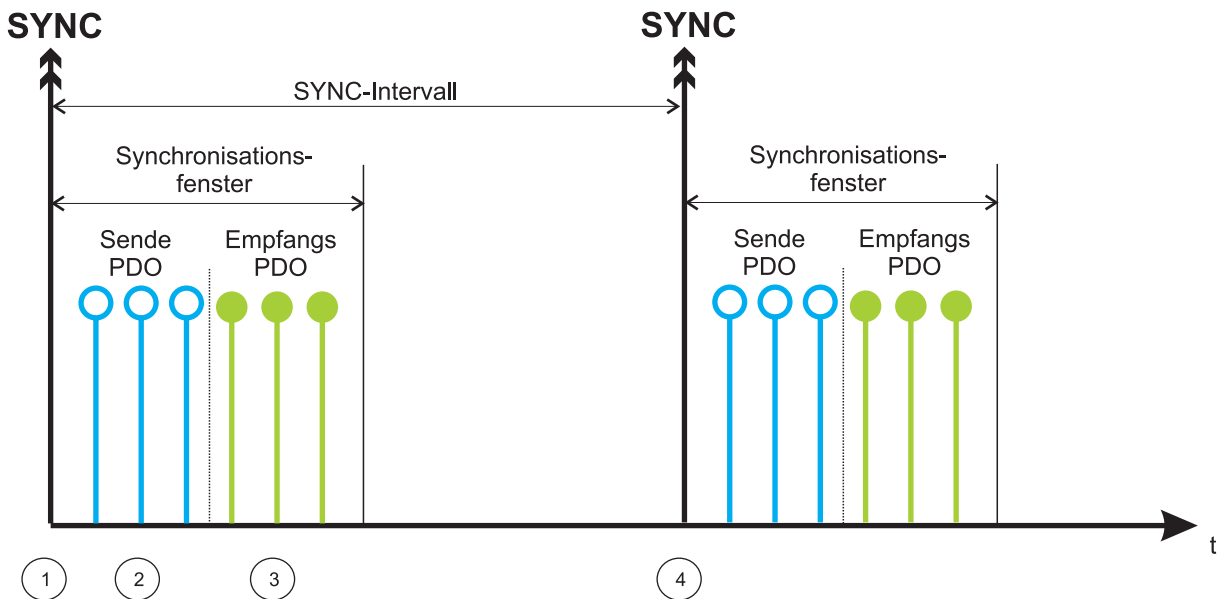


Abbildung 29: Mechanismus Synchronisierung

Nach Empfang des SYNC-Telegramms ① sendet ein Netzwerkknoten zunächst die SendepDOs (TxPDOs) ②. Danach werden die bis dahin empfangenen PDOs (RxPDOs) ③ in einen Zwischenspeicher übernommen. Die Verarbeitung der RxPDOs erfolgt erst nach Erhalt des nächsten SYNC-Telegramms ④. Synchroner PDOs dürfen nur innerhalb des Synchronisationsfensters auftreten.

Für das Optionsmodul CANopen-Slave bedeutet das:

Nach Empfang des SYNC-Telegramms ① sendet das Optionsmodul CANopen-Slave die SendepDOs (TxPDOs) ②. Danach werden die bis dahin empfangenen PDOs (RxPDOs) ③ in einen Zwischenspeicher übernommen. Das Kopieren der Daten dieser RxPDOs in das DPRAM zur b maXX PLC erfolgt erst nach Erhalt des nächsten SYNC-Telegramms ④.

Das Optionsmodul CANopen-Slave bietet die Möglichkeit das SYNC-Intervall zu überwachen. Dafür muss auf dem Optionsmodul CANopen-Slave das SYNC-Intervall (als Vielfaches von 1 μ s) im Objekt 1006h eingestellt werden. Das Optionsmodul CANopen-Slave meldet den Ausfall eines SYNC-Telegramms an das Applikationsprogramm auf der b maXX PLC.

SYNC-Intervall:

Objekt	Bedeutung
1006h	SYNC-Intervall / Dauer Kommunikationszyklus.
Subindex 0	0: wird nicht verwendet n: Länge des SYNC-Intervalls in μ s.

Auf dem Optionsmodul CANopen-Slave können die Werte 1000, 2000, 4000 und 8000 eingestellt werden. Mit dem Multiplikator 1 μ s ergeben sich somit die SYNC-Intervalle

1 ms, 2 ms, 4 ms oder 8 ms. Defaulteinstellung ist 0, d. h. es ist kein SYNC-Intervall eingestellt und damit wird der Empfang des SYNC-Telegramms nicht überwacht.

Durch den Übertragungstyp des PDO (siehe Kapitel [Die Konfiguration der Übertragungseigenschaften eines PDO](#) auf Seite 54) kann zudem konfiguriert werden, dass ein PDO erst nach dem n-ten SYNC-Telegramm übernommen wird. Somit lassen sich kurze Zyklen für wichtige Daten und lange Zyklen für weniger wichtige Daten einrichten.

Erkennen eines SYNC-Telegramm Ausfalls

Die Meldung des Ausfalls eines SYNC-Telegramms an das Applikationsprogramm auf der b maXX PLC erfolgt über den FB CANop405_KERNEL_STATE_SL. Sehen Sie hierzu bitte im Kapitel [CANopen-Kernel-State](#) auf Seite 73 und in der Online-Hilfe des FB CANop405_KERNEL_STATE_SL nach.

Die Überwachung wird mit dem Empfang des ersten SYNC-Telegramms aktiviert. Wird jetzt kein SYNC-Telegramm nach Ablauf des SYNC-Intervalls (Objekt 1006h) empfangen, wird über den FB CANop405_KERNEL_STATE_SL und die Fehlermeldung "Ausfall des SYNC-Telegramms" der Ausfall des SYNC-Telegramms signalisiert. Sobald wieder SYNC-Telegramme im Zeitraster des SYNC-Intervalls empfangen werden, erlischt die Fehlermeldung.



HINWEIS

Die SYNC-Überwachung ist per Defaulteinstellung von Objekt 1006h (SYNC-Intervall) ausgeschaltet, d. h. mit der Defaulteinstellung von Objekt 1006h wird der FB CANop405_KERNEL_STATE_SL keinen SYNC-Ausfall melden.

3.3.7.7 Node Guarding

Definition nach CANopen-Spezifikation

Für die Ausfallüberwachung von Netzwerkknoten gibt es die Mechanismen Node Guarding und Life Guarding. Über Node Guarding werden vom CANopen-Master die Netzwerkknoten überwacht, die ihrerseits über Life Guarding den Ausfall des CANopen-Master erkennen können. Hinter beiden Mechanismen steckt die gleiche Funktionalität:

Beim Node Guarding fordert der CANopen-Master in bestimmten Zeitintervallen ‚Guard Time‘ über ein Remote-Telegramm eine Antwort vom CANopen-Slave an. Der Slave antwortet mit der Guarding-Nachricht. Diese enthält den NMT-Zustand (siehe Kapitel [Netzwerkkommandos - NMT](#) auf Seite 46) des CANopen-Slaves sowie ein Toggle-Bit, das nach jeder Nachricht wechseln muss. Falls Status oder Toggle Bit nicht mit den vom CANopen-Master erwarteten übereinstimmen oder falls innerhalb eines bestimmten Zeitraums, der ‚Life-Time‘, keine Antwort erfolgt geht der Master von einem Slave-Fehler aus. Umgekehrt kann der Slave den Ausfall des Masters erkennen. Falls der Slave innerhalb der eingestellten ‚Life-Time‘ keine Nachrichtenanforderung vom Master erhält, geht er von einem Masterausfall aus, meldet einen Ausfall an das Applikationsprogramm auf der b maXX PLC und sendet ein Emergency Telegramm.

Auf dem Optionsmodul CANopen-Slave ist das Node Guarding ist per Defaulteinstellung ausgeschaltet.

Zur Einstellung der Guarding Funktionalität sind auf dem Optionsmodul CANopen-Slave zwei Objekte wichtig, die vom CANopen-Master über SDO eingestellt werden müssen:

Guarding-Time:

Objekt	Bedeutung
100Ch	Konfiguration Guarding Time.
Subindex 0	0: Guarding ist nicht aktiv n: Abstand zwischen zwei Guard Telegrammen in ms

Life-Time Faktor:

Objekt	Bedeutung
100Dh	Konfiguration Life Time Faktor.
Subindex 0	0: Life Guarding ist nicht aktiv n: Die Life Time ergibt sich aus $n \times \text{Guard Time}$

Defaulteinstellung für Guarding-Time ist 0, d. h. es ist keine Guarding-Time eingestellt.

Defaulteinstellung für Life-Time Factor ist 0, d. h. es ist kein Life-Time Factor eingestellt.

Das Node Guarding ist in allen Kommunikationsphasen (STOPPED, PRE-OPERATIONAL, OPERATIONAL) verfügbar. Das Toggle-Bit wird nur in der Phase INITIALISIERUNG auf seinen Defaultwert zurückgesetzt. Dies bedeutet, dass auch bei Zustandswechseln der Togglemechanismus weitergeführt wird. Gestartet wird das Node Guarding im Optionsmodul CANopen-Slave nach Empfang des ersten Guarding-Anforderungstelegramms (das der CANopen-Master sendet). Ab diesem Zeitpunkt läuft im Optionsmodul CANopen-Slave die in den Objekten 100Ch und 100Dh parametrisierte Überwachungszeit.

Erkennen eines Node-Guarding Ausfalls

Die Meldung "Ausfalls eines Node Guarding Telegramms" an das Applikationsprogramm auf der b maXX PLC erfolgt über den FB CANop405_LIFE_GUARD_SL. Sehen Sie hierzu bitte in der Online-Hilfe des FB CANop405_LIFE_GUARD_SL nach.

3.3.7.8 Heartbeat

Für die Ausfallüberwachung von Netzwerkknoten gibt es neben den Mechanismen NodeGuarding und Life Guarding auch den Mechanismus Heartbeat.

Beim Heartbeat sendet ein oder mehrere Heartbeat-Producer Heartbeat-Telegramme in bestimmten Zeitintervallen "Heartbeat-Producer Time". Ein Heartbeat-Consumer überwacht ob die Heartbeat-Telegramme innerhalb der "Heartbeat-Consumer Time" gesendet wurden. Ein Heartbeat-Consumer kann so bis zu 127 Heartbeat-Producer überwachen. Dafür können auf dem Heartbeat-Consumer bis zu 127 verschiedene "Heartbeat-Consumer Times" eingestellt werden.

Das Optionsmodul CANopen-Slave für b maXX PLC kann als Heartbeat-Producer eingestellt werden.



HINWEIS

Das Optionsmodul CANopen-Slave kann nur als Heartbeat-Producer eingestellt werden. Es kann NICHT als Heartbeat-Consumer eingestellt werden.

Die Einstellung der Heartbeat-Producer Funktionalität auf den Optionsmodul CANopen-Slave erfolgt über das Objekt Producer Heartbeat Time (Objekt 1017h), das über den CANopen-Master eingestellt werden muss.

Producer Heartbeat Time

Objekt	Bedeutung
1017	Producer Heartbeat Time für das Optionsmodul CANopen-Slave
Subindex 0	Heartbeat Time

Die Producer Heartbeat Time wird als Vielfaches von 1 ms angegeben. Der Eintrag ist schreibbar und lesbar und vom Datentyp UNSIGNED16.

Im Optionsmodul CANopen-Slave hat das Objekt 1017h (Subindex 0) defaultmäßig den Wert "0", d. h. das Optionsmodul CANopen-Slave sendet keine Heartbeat-Telegramme. Der Wert für das Objekt 1017h (Subindex 0) kann von einem CANopen-Master über SDO schreiben geändert werden.

3.3.7.9 Eigene Netzwerkknotenadresse (Local Node-ID)

Beim Optionsmodul CANopen-Slave können Sie die Node-ID zum einen über die Dip-Schalter und zum anderen mittels Software bei der Initialisierung einstellen. Es kann für ein Applikationsprogramm notwendig sein die eigene Node-ID des CANopen-Slaves auszulesen.

Das Auslesen der eigenen Node-ID des Optionsmodul CANopen-Slave erfolgt mit dem FB CANop405_LOC_NODE_ID_SL.

Sehen Sie hierzu bitte auch in der Online-Hilfe des FB CANop405_LOC_NODE_ID_SL nach.

Local Node-ID ermitteln

Sie können im Applikationsprogramm auf der b maXX PLC mittels des FBs CANop405_LOC_NODE_ID_SL die eigene Netzwerkknotenadresse (Local Node-ID) des Optionsmodul CANopen-Slave lesen.

Dazu gehen Sie bitte in folgenden Schritten vor

- Legen Sie eine POE mit SPS-Typ SH03_30 und Prozessortyp BM4_O_PLC01 an. Diese POE soll später in einer zyklischen Task aufgerufen werden. Alternativ können Sie auch die POE in der der NMT-Zustand gelesen wird verwenden.
- Platzieren Sie den FB CANop405_LOC_NODE_ID_SL in dieser POE.
- Beschalten Sie den Funktionsbaustein mit Variablen vom richtigen Datentyp.

- Legen Sie eine zyklische Task mit mittlerer bis geringer Priorität an, falls eine solche in Ihrem Projekt noch nicht vorhanden sein sollte. Binden Sie die erstellte POE mit dem FB CANop405_LOC_NODE_ID_SL in dieser Task ein.

Wenn Sie die POE in der der NMT-Zustand gelesen wird verwenden, haben Sie diese POE bereits einer Task zugeordnet.

- Übersetzen Sie das Projekt und laden Sie es als (Boot-)Projekt auf die PLC.
- Starten Sie das Projekt.

Mit `x_ENABLE = 1` wird die eigene Netzwerkknotenadresse (Local Node-ID) des CANopen-Slave gelesen. Der FB CANop405_LOC_NODE_ID_SL meldet eine erfolgreiche Durchführung mit `x_CONFIRM = 1`.

Beispiel: Local Node-ID ermitteln

Um den FB CANop405_LOC_NODE_ID_SL verwenden zu können legen wir eine POE an und platzieren den Funktionsbaustein darin. Die erstellte POE ordnen wir einer zyklischen Task mit 200 ms Aufrufintervall zu. Nach der Beschaltung des Funktionsbausteins übersetzen Sie das Projekt und senden Sie dieses als Projekt an die b maXX PLC. Starten Sie das Projekt. Mit `x_ENABLE = 1` wird die eigene Netzwerkknotenadresse (Local Node-ID) des CANopen-Slaves (in unserem Fall "5") gelesen:

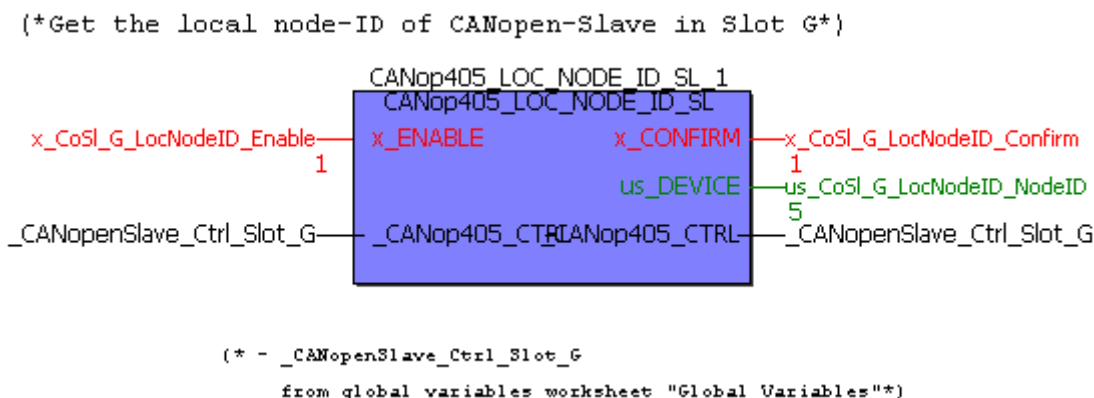


Abbildung 30: Ermitteln der eigenen Netzwerkknotenadresse (Local Node-ID) des CANopen-Slaves

3.3.7.10 CANopen-Kernel-State

Das Auslesen des CANopen-Kernel-States des Optionsmodul CANopen-Slave erfolgt mit dem FB CANop405_KERNEL_STATE_SL.

Sehen Sie hierzu bitte auch in der Online-Hilfe des FB CANop405_KERNEL_STATE_SL nach.

CANopen-Kernel-State ermitteln

Sie können im Applikationsprogramm auf der b maXX PLC mittels des FBs CANop405_KERNEL_STATE_SL den CANopen-Kernel-State des Optionsmodul CANopen-Slave lesen.

Dazu gehen Sie bitte in folgenden Schritten vor:

3.3 Datenaustausch über CANopen mit dem Optionsmodul CANopen-Slave

- Legen Sie eine POE mit SPS-Typ SH03_30 und Prozessortyp BM4_O_PLC01 an. Diese POE soll später in einer zyklischen Task aufgerufen werden.

Alternativ können Sie auch die POE in der der NMT-Zustand gelesen wird verwenden.

- Platzieren Sie den FB CANop405_KERNEL_STATE_SL in dieser POE.
- Beschalten Sie den Funktionsbaustein mit Variablen vom richtigen Datentyp.
- Legen Sie eine zyklische Task mit mittlerer bis geringer Priorität an, falls eine solche in Ihrem Projekt noch nicht vorhanden sein sollte. Binden Sie die erstellte POE mit dem FB CANop405_KERNEL_STATE_SL in dieser Task ein.

Wenn Sie die POE in der der NMT-Zustand gelesen wird verwenden, haben Sie diese POE bereits einer Task zugeordnet.

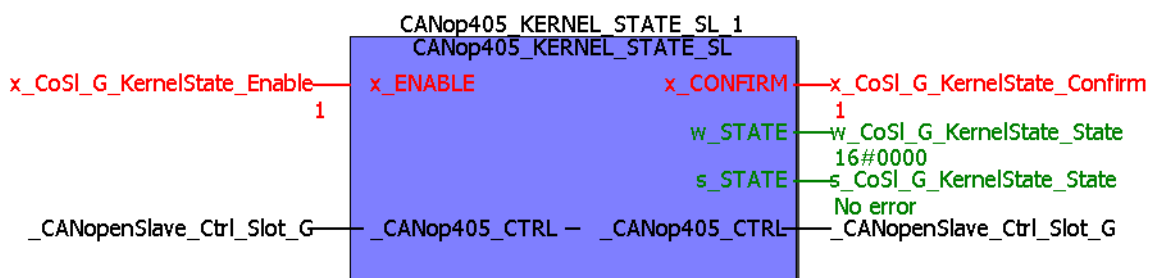
- Übersetzen Sie das Projekt und laden Sie es als (Boot-)Projekt auf die PLC.
- Starten Sie das Projekt.

Mit `x_ENABLE = 1` wird der CANopen-Kernel-State des CANopen-Slave gelesen. Der FB `CANop405_KERNEL_STATE_SL` meldet eine erfolgreiche Durchführung mit `x_CONFIRM = 1`.

Beispiel: CANopen-Kernel-State ermitteln

Um den FB `CANop405_KERNEL_STATE_SL` verwenden zu können legen wir eine POE an und platzieren den Funktionsbaustein darin. Die erstellte POE ordnen wir einer zyklischen Task mit 200 ms Aufrufintervall zu. Nach der Beschaltung des Funktionsbausteins übersetzen Sie das Projekt und senden Sie dieses als Projekt an die b maXX PLC. Starten Sie das Projekt. Mit `x_ENABLE = 1` wird der CANopen-Kernel-State des CANopen-Slaves (in unserem Fall am Ausgang `w_STATE` "16#0000" bzw. am Ausgang `s_STATE` 'No Error') gelesen:

(*Get the CANopen kernel state of CANopen-Slave in Slot G*)



```
(* - _CANopenSlave_Ctrl_Slot_G
from global variables worksheet "Global_Variables"*)
```

Abbildung 31: Ermitteln des CANopen-Kernel-States des CANopen-Slaves

3.3.7.11 Konsistenz der Daten

Bei der Datenkonsistenz muss unterschieden werden in

- Konsistenz der Daten zwischen CANopen-Master und Optionsmodul CANopen-Slave und
- Konsistenz der Daten zwischen Optionsmodul CANopen-Slave und b maXX PLC.

CANopen-Master ↔ Optionsmodul CANopen-Slave

Das Optionsmodul CANopen-Slave unterstützt Konsistenz über alle angelegten Variablen (8-Bit, 16-Bit und 32-Bit), d. h. die Daten im CANopen-Telegramm werden konsistent übertragen.

Optionsmodul CANopen-Slave ↔ b maXX PLC

Die b maXX PLC unterstützt Konsistenz für 8-Bit-Werte und für 16-Bit-Werte.

Falls Konsistenz über einen größeren Datenbereich gewünscht ist (z. B. 32-Bit-Werte) ist ein Handshake-Mechanismus zwischen Optionsmodul CANopen-Slave und b maXX PLC notwendig.

Dazu benötigen Sie folgenden Code in eine POE (in Structured Text (ST)):

Variablen-Arbeitsblatt:

```

VAR
    u_tempAccessVar    :    UINT; (* temporary use *)
    u_tempMerker       :    UINT;
END_VAR
VAR_EXTERNAL
    _CANopenSlave_Ctrl_Slot_G : CANopen_PLC_SL_BMSTRUCT; (*CANopen-Slave in
                                                                slot G*)
END_VAR

```

Diesen Code können Sie in das Variablen-Arbeitsblatt Ihrer POE kopieren.

Code-Arbeitsblatt:

```

(* Diese Sequenz nur in einer Task einsetzen ! *)
(* Run this code sequence in only one task ! *)

_CANopenSlave_Ctrl_Slot_G.u_PLC_ACCESS := UINT#1; (* sperren / lock *)
u_tempAccessVar := _CANopenSlave_Ctrl_Slot_G.u_OMC_ACCESS;

IF ((UINT_TO_WORD(u_tempAccessVar) AND WORD#16#0001) = WORD#16#0000) THEN
    (* wir dürfen zugreifen / we are allowed to access DPRAM *)
    IF (u_tempAccessVar <> u_tempMerker) THEN
        (* das Optionsmodul hat Daten ausgetauscht / data exchange done *)
        (* -- Fügen Sie hier Ihre Datenzugriffe ein -- *)
        (* ----- add your data access code here ----- *)
        ;
    END_IF;

```

3.4 Der Aufbau von CANopen-Telegrammen

```
END_IF;  
  
u_tempMerker      := u_tempAccessVar;  
_CANopenSlave_Ctrl_Slot_G.u_PLC_ACCESS := UINT#0; (* freigeben / unlock *)
```

Diesen Code können Sie in das Code-Arbeitsblatt Ihrer POE kopieren.

Der Datenaustausch (Lesen und Schreiben der Variablen) zwischen b maXX PLC und Optionsmodul CANopen-Slave findet dann z. B. wie folgt statt:

```
(* --- Fügen Sie hier Ihre Datenzugriffe ein --- *)  
(* ----- add your data access code here ----- *)  
ud_MyWriteVar1 := ud_MyReadVar1 + UDINT#1;
```



HINWEIS

Der Code für den konsistenten Datenaustausch wird nur in einer POE eingesetzt und diese POE darf nicht instanziiert werden!

Anderenfalls ist die Konsistenz der Daten nicht gewährleistet.

3.4 Der Aufbau von CANopen-Telegrammen

3.4.1 Allgemeines

Durch das Optionsmodul CANopen-Slave und die Funktionsbausteine aus der Bibliothek CANop405_PLC01_20bd02 (und höher) ist der Telegrammverkehr für Sie völlig abgekapselt und Sie benötigen kaum Kenntnis über den Aufbau der einzelnen CANopen-Telegramme. Sollten Sie dennoch mit einem CAN-Analysetool den Datenverkehr kontrollieren wollen, können Sie aus den nachfolgenden Kapiteln den Aufbau der einzelnen Telegramme entnehmen. Die Telegramme werden Ihnen im Format

COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
--------	-----	-----	--------	--------	--------	--------	--------	--------	--------	--------

erläutert. Diesen Aufbau finden Sie in den meisten CAN-Analysetools wieder, wobei der Datenbereich Byte 0 bis Byte 7 noch auf das jeweilige CANopen-Telegramm angepasst sein kann. RTR bezeichnet das Bit für einen Remote-Transmit-Request. Das Feld DLC besteht aus 4 Bit und bezeichnet die Anzahl der Datenbytes (DLC = Data Length Code).

3.4.2 NMT-Telegramm

Je Telegramm werden zwei Datenbyte übertragen. Das Datenbyte 0 enthält den Command Specifier CS, das Datenbyte 1 die Geräteadresse. Ist die Adresse 0 eingetragen, so werden mit dem entsprechenden Kommando alle Knoten angesprochen (Broadcast).

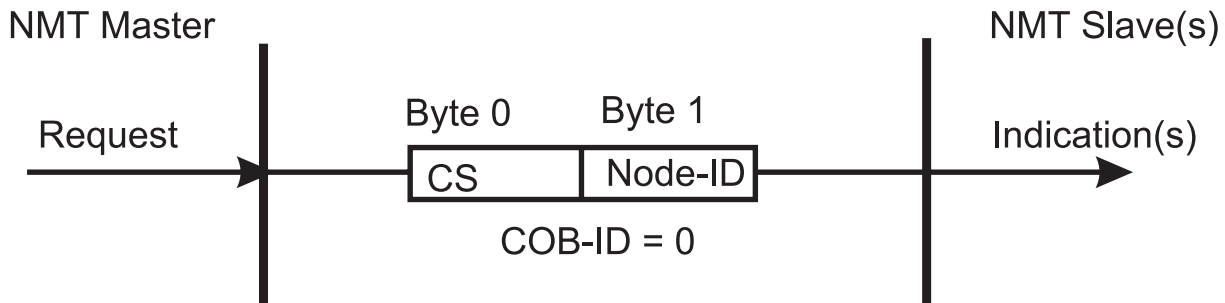


Abbildung 32: Kommunikationsbeziehung NMT

CS	Bezeichnung	Wirkung
1	Start_Remote_Node	Starten des Normalbetriebes
2	Stop_Remote_Node	Deaktivieren der PDO- und SDO-Kommunikation
128	Enter_Pre-Operational_State	Übergang in Konfigurationsmodus
129	Reset_Node	Kontrolliertes Zurücksetzen des gesamten Objektverzeichnis' auf Default-Werte
130	Reset_Communication	Zurücksetzen des Kommunikationsteils im Objektverzeichnis auf Default-Werte

Ein Telegramm, welches den Knoten 16 in den Konfigurationsmodus bringt, sieht wie folgt aus:

COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
00h	0	2h	80 h	10h						

Diese Telegramme sind unbestätigt, d. h. kein NMT-Slave quittiert dem NMT-Master die korrekt empfangene Nachricht.

3.4.3 SDO-Telegramm

3.4.3.1 Aufbau eines SDO-Telegramms

Die COB-ID der Request SDO (Anforderung vom Master) ergibt sich aus 600h + Adresse, bei Response-SDOs (Antwort des Slaves) aus 580h + Adresse. Das Datenfeld des CAN-Datentelegramms (8 Byte) für eine SDO gliedert sich in drei Teile, einen Command Specifier CS (1 Byte), einem Multiplexor M (3 Byte) und dem eigentlichen Nutzdatenbereich D0 - D4 (4 Byte).

3.4 Der Aufbau von CANopen-Telegrammen

COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
600h/ 580h + Adresse	0	8h	CS	M			D0	D1	D2	D3

Der Multiplexor M besteht aus dem 16 Bit Index eines Objektes und dem dazugehörigen acht Bit breiten Subindex. Bei segmentierten Telegrammen (vom Optionsmodul CANopen-Slave nicht unterstützt) wird der Nutzdatenbereich um die drei Byte des Multiplexors erweitert, wodurch je Telegramm 7 Byte Nutzdaten übertragen werden können. Der Command Specifier CS klassifiziert die verschiedenen SDO-Typen.

3.4.3.2 Arten des SDO-Transfers

Das Optionsmodul CANopen-Slave unterstützt den Expedited Transfer nach DS 301. Es können Objekte geschrieben oder gelesen werden, deren Daten maximal 4 Byte umfassen. Es sind nur zwei Telegramme erforderlich, eine Anforderung und eine Antwort.

3.4.3.3 Objekt über SDO schreiben

Ein CANopen-Master sendet (Client) einen Schreib-Request an das Optionsmodul CANopen-Slave (Server). Dieses führt die Anforderung aus und quittiert dies mit dem Response.

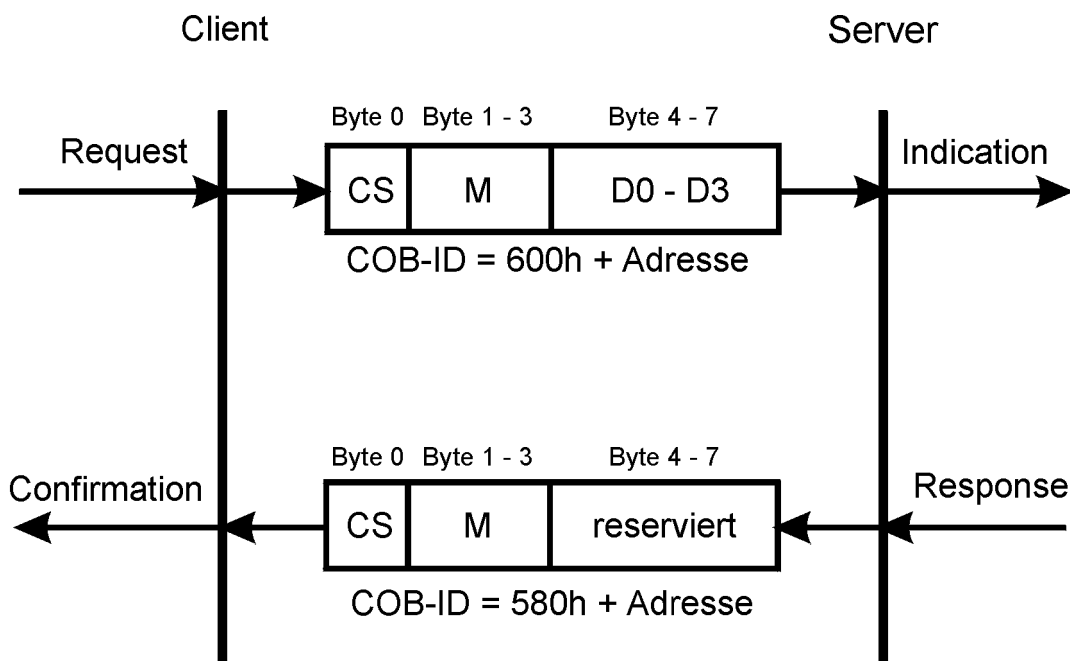


Abbildung 33: Kommunikationsbeziehung SDO schreiben

Der Command Specifier CS für den Request hängt von der Nutzdatenlänge ab. D0 ist das LSB, D3 das MSB.

Datenlänge in D0 - D3	Command Specifier CS
1 Byte	2Fh
2 Byte	2Bh
4 Byte	23h

Der Command Specifier CS für die Response beträgt 60h, der Multiplexor M ist identisch zu dem des Requests, das Datenfeld ohne Bedeutung (reserviert).

3.4.3.4 Objekt über SDO lesen

Ein CANopen-Master sendet (Client) einen Lese-Request an das Optionsmodul CANopen-Slave (Server). Dieser führt die Anforderung aus und quittiert dies mit dem Response, welcher auch die Daten enthält.

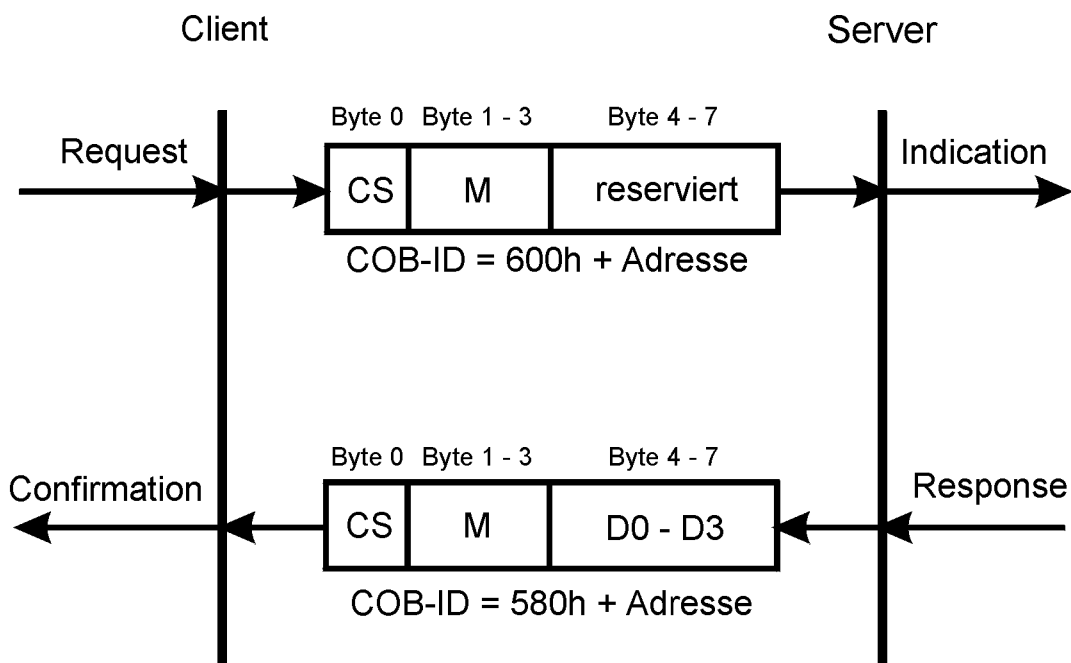


Abbildung 34: Kommunikationsbeziehung SDO lesen

Ein SDO Server (Master) sendet einen Lese-Request an den Slave. Dieser Slave führt die Anforderung aus und schickt die geforderten Daten im Antworttelegramm (Response). Der Command Specifier CS für den Request beträgt immer 40h. Der Command Specifier CS für die Response hängt von der Nutzdatenlänge ab. D0 ist das LSB, D3 das MSB.

Datenlänge in D0 - D3	Command Specifier CS
1 Byte	4Fh
2 Byte	4Bh
4 Byte	43h

Der Multiplexor von Request und Response stimmt überein.

3.4 Der Aufbau von CANopen-Telegrammen

3.4.4 PDO-Telegramm

Ein PDO-Producer sendet die Prozessdaten in einem PDO an das CANopen-Netzwerk. Abhängig von der Konfiguration werten einer oder mehrere PDO-Consumer das PDO aus.

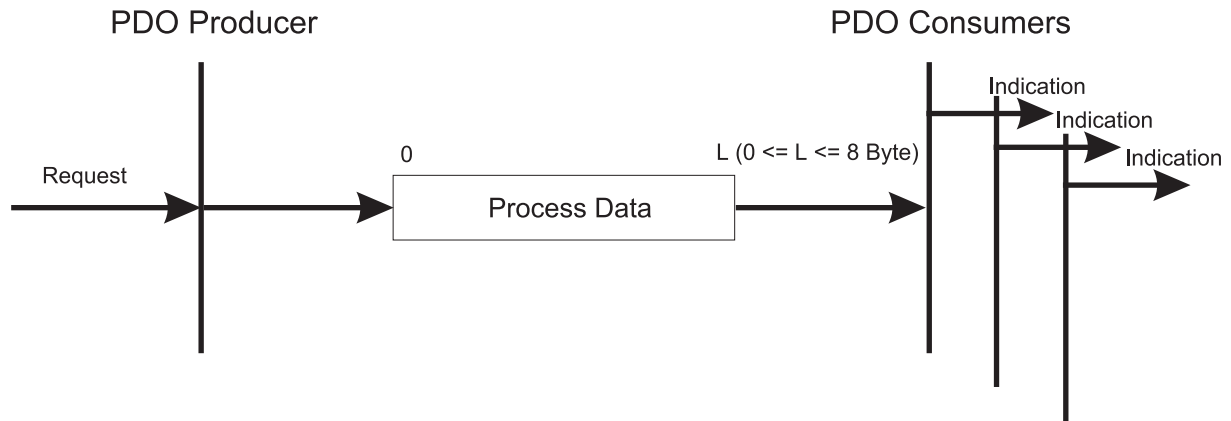


Abbildung 35: Kommunikationsbeziehung PDO

Für PDO-Telegramme stehen alle 8 Byte des Datenbereichs im CAN-Datentelegramms zur Verfügung. Der Inhalt ergibt sich durch die Einstellungen der Mapping Objekte. Für die COB-ID ergeben sich Werte von 181h bis 57Fh, sofern die COB-ID nach der vordefinierten Identifizierung des CiA vergeben wurde. Darüber hinaus kann die COB-ID auch frei eingestellt werden.

COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
181h - 57Fh oder beliebig	0	0h - 8h	D0	D1	D2	D3	D4	D5	D6	D7

Bei Remote-Anforderungen von PDO-Telegrammen, hat die COB-ID von PDO-Anforderung und PDO-Antwort den gleichen Wert. In der PDO-Anforderung ist das RTR-Bit (Bestandteil des DLC) gesetzt. Die Datenbytes haben keine Inhalte, das DLC Feld ist somit 0.

COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
181h - 57Fh oder beliebig	1	0h	D0	D1	D2	D3	D4	D5	D6	D7

3.4.5 SYNC-Telegramm

Ein SYNC-Producer sendet das SYNC-Telegramm an das CANopen-Netzwerk. Abhängig von der Konfiguration werden einer oder mehrere SYNC-Consumer das Telegramm aus.

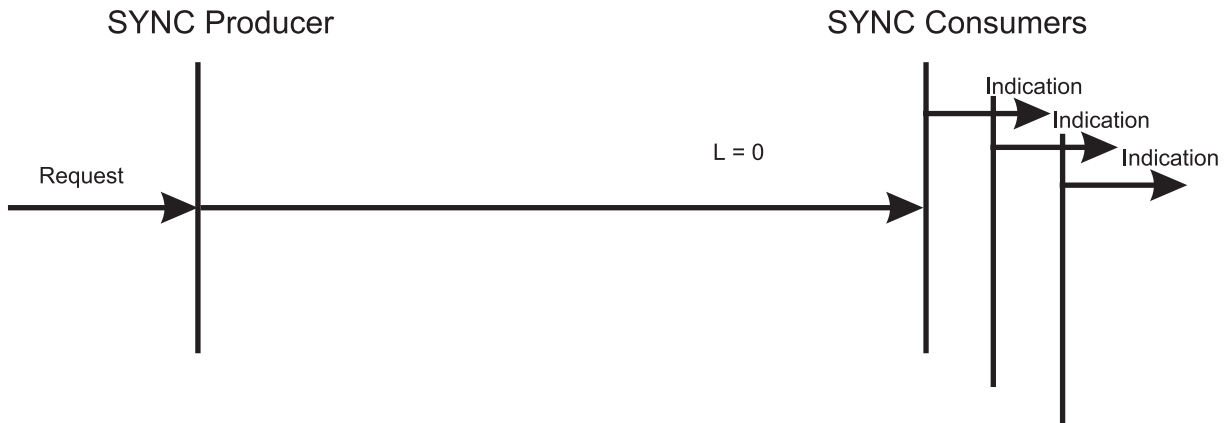


Abbildung 36: Kommunikationsbeziehung SYNC

Das SYNC-Telegramm enthält keine Daten. Die COB-ID ist 80h nach der vordefinierten Identifizierung des CiA. Darüber hinaus kann die COB-ID auch frei eingestellt werden.

COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80h oder beliebig	0h	0h	-	-	-	-	-	-	-	-

3.4 Der Aufbau von CANopen-Telegrammen

3.4.6 Node Guarding Telegramm

Der Master fragt in bestimmten Intervallen durch Remoteframes die Slaves ab.

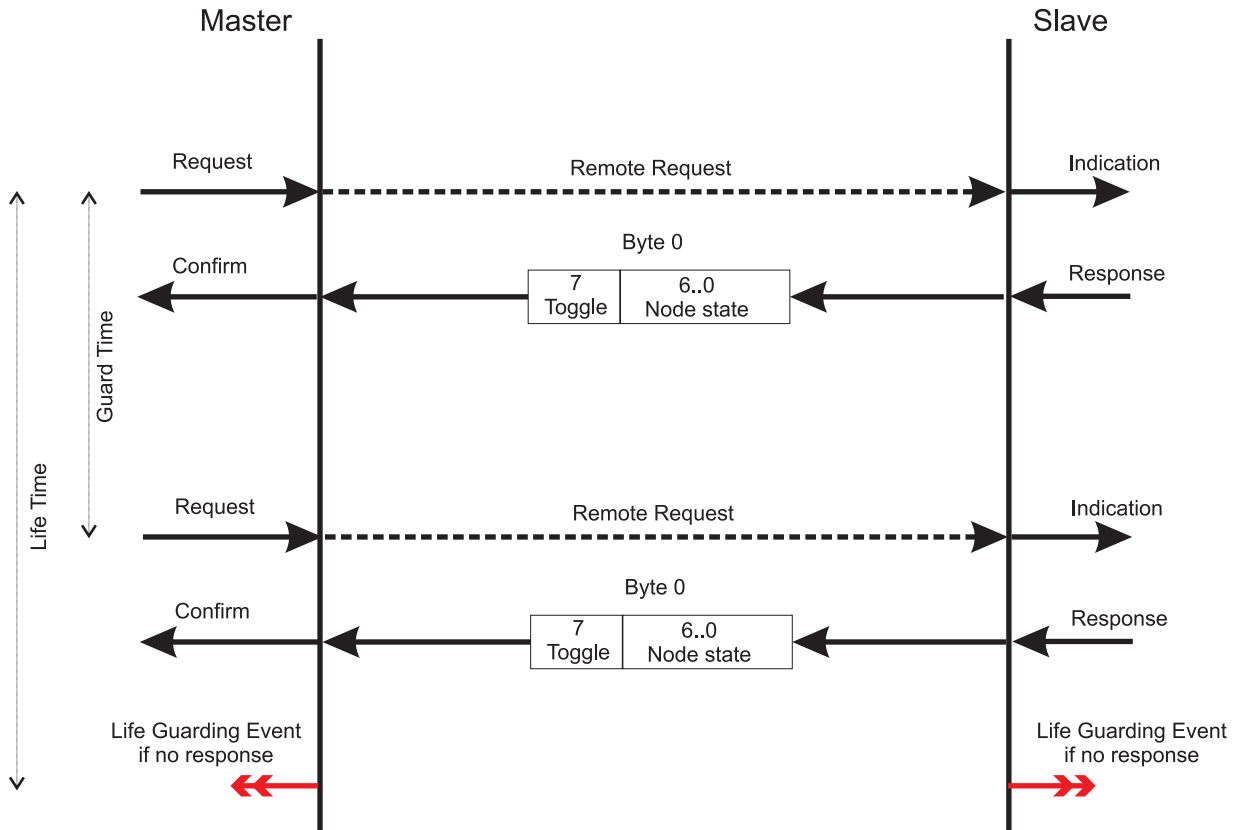


Abbildung 37: Kommunikationsbeziehung Node Guarding

Die COB-ID ergibt sich aus 700h + Adresse. Das Datenfeld des CAN-Datentelegramms besteht im Antworttelegramm aus einem Byte dem Byte 0. Byte 0 teilt sich in das Toggle-Bit (Bit 7) und den Node-State (Bit 6..0) auf (Node-State = NMT state).

COB-ID	RTR	DLC	Byte 0
700h + Adresse	0	1h	Toggle-Bit + Node-State

Die Remote-Anforderung enthält keine Daten. Das RTR-Bit ist gesetzt.

COB-ID	RTR	DLC	Byte 0
700h + Adresse	1	0h	-

3.4.7 Heartbeat-Telegramm

Ein Heartbeat-Producer sendet in bestimmten Intervallen ein Heartbeat-Telegramm. Eine Antwort wird nicht erwartet.

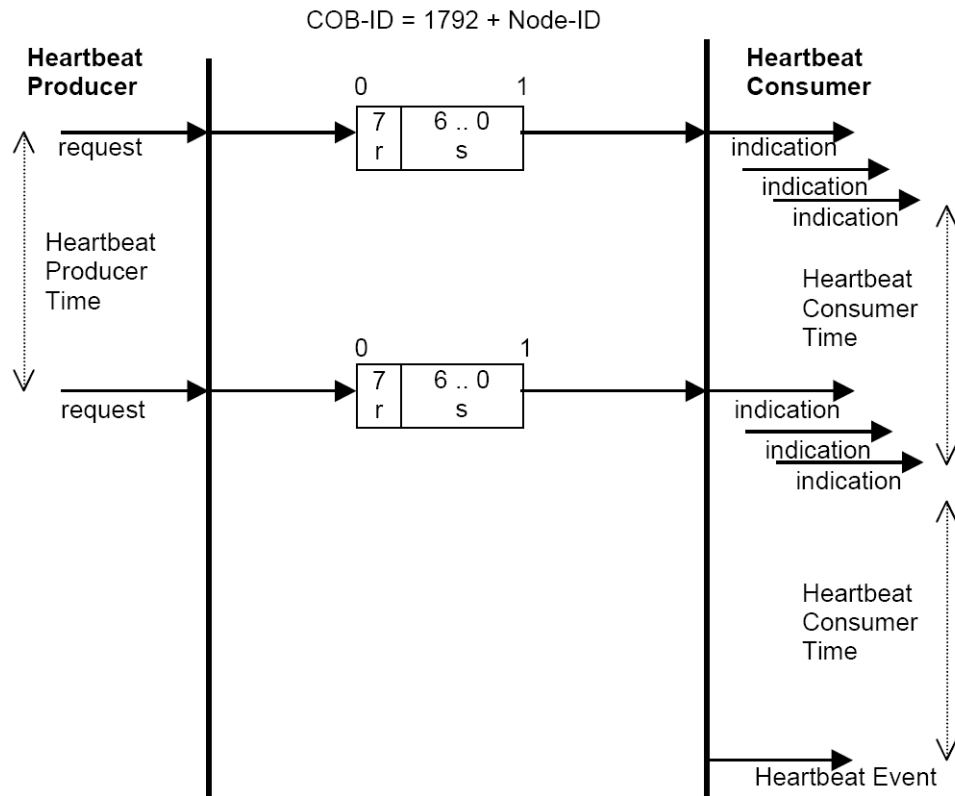


Abbildung 38: Kommunikationsbeziehung Heartbeat

Die COB-ID ergibt sich aus 700h + Adresse. Das Datenfeld des CAN-Datentelegramms besteht aus einem Byte, dem Byte 0. Byte 0 beinhaltet ein reserviertes Bit (Bit 7 = 0) und den NMT-Status (Bit 6..0) des Heartbeat-Producers.

COB-ID	RTR	DLC	Byte 0
700h + Adresse	0	1	NMT-Status

3.4 Der Aufbau von CANopen-Telegrammen

3.4.8 Emergency-Telegramm

Ein Emergency-Producer sendet das Emergency-Telegramm an das CANopen-Netzwerk. Abhängig von der Konfiguration werden eine oder mehrere Emergency-Consumer das Telegramm aus.

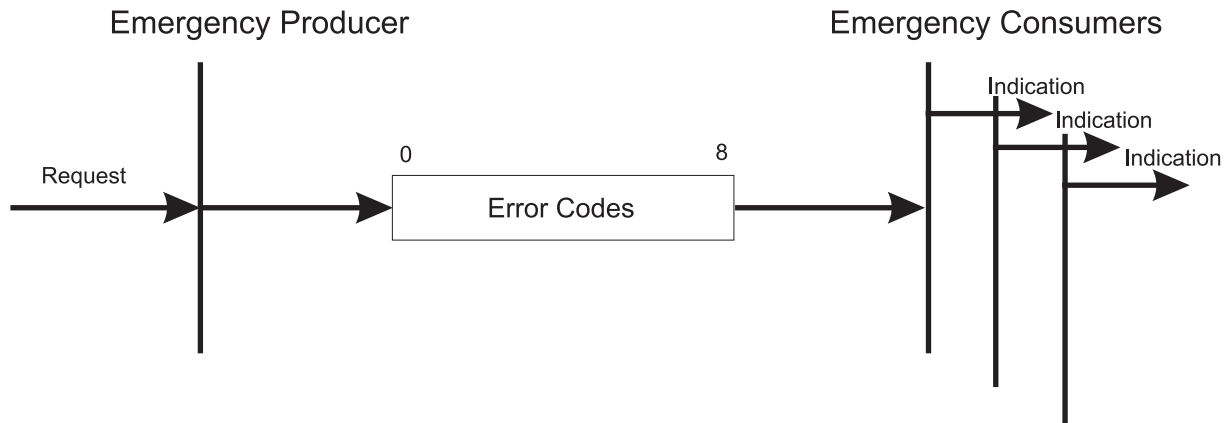


Abbildung 39: Kommunikationsbeziehung Emergency

Die COB-ID ergibt sich aus $80h + \text{Adresse}$. Für Emergency-Telegramme stehen alle 8 Byte des CAN-Datentelegramms zur Verfügung. Die 8 Byte teilen sich wie folgt auf:

COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
$80h + \text{Adresse}$	0h	8h	Emergency Error Code		Error Reg.	Hersteller spezifisch				



ANHANG A - ABKÜRZUNGEN

CAL	CAN Application Layer	NMT	Network Management, Netzwerkmanagement
CAN	Controller Area Network	Node-ID	Node-Identifizier, Adresse, Knotenadresse, Netzwerkknotenadresse, CANopen-Slave Adresse
CiA	CAN in Automation e. V., www.can-cia.de	PDO	Process Data Object, Prozessdatenobjekt
COB	Communication Object, Kommunikationsobjekt	PLC	Process Loop Controller, Speicherprogrammierbare Steuerung (SPS)
COB-ID	Communication Object Identifier	POE	Programm Organisationseinheit, Program Organisation Unit (POU), IEC 61131-3
CS	Command Specifier	PROPROP wt II	Tool zur Programmierung der b maXX PLC
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance	RTR	Remote Transmission Request
CSMA/CD	Carrier Sense Multiple Access / Collision Detection	RxPDO	Receive-PDO, Empfangs-PDO
DCF	Device Configuration File	SDO	Service Data Object, Bedarfsdatenobjekt
DIN	Deutsches Institut für Normung e.V.	SPS	Speicherprogrammierbare Steuerung, Process Loop Controller (PLC)
DLC	Data Length Code	TxPDO	Transmit-PDO, Sende-PDO
DS	Draft Standard	WinBASS II	Tool zur Parametrierung des b maXX Reglers
DSP	Draft Standard Proposal	WWW	World Wide Web
EDS	Electronic Data Sheet, elektronisches Datenblatt	16#	Präfix für Hexadezimalzahl
EMCY	Emergency		
EN	Europäische Norm		
FB	Function block, Funktionsbaustein (IEC 61131-3)		
IEC	International Electrotechnical Commission, www.iec.ch		
ISO	International Standard Organisation, www.iso.ch		
LED	Light Emitting Diode, Leuchtdiode		
LSB	Last Significant Bit, niederwertigstes Bit		
MSB	Most Significant Bit, höchstwertigstes Bit		



ANHANG B - TABELLEN

In den nachfolgenden Tabellen sind die Objekte aufgeführt, die das Optionsmodul CANopen-Slave unterstützt.

B.1 Objekte nach DS 301

In diesem Abschnitte finden Sie alle von dem Baumüller CANopen-Optionsmodul unterstützten Objekte des kommunikationsspezifischen Bereiches des Objektverzeichnis nach DS 301.

Übersicht der Objekte:

Index	Objektinhalt / Bedeutung
1000h	Gerätetyp: unterstütztes Geräteprofil und zusätzliche Information
1001h	Fehlerregister
1002h	
1003h	Fehlerspeicher enthält eine Fehlerliste des Gerätes
1004h	
1005h	SYNC-Telegramm COB-ID
1006h	SYNC-Intervall
1007h	
1008h	
1009h	
100Ah	
100Bh	
100Ch	Guarding Time
100Dh	Life Time Faktor
100Eh	
100Fh	
1010h	Parameter abspeichern

Index	Objektinhalt / Bedeutung
1011h	Default Parameter laden
1014h	
1400h - 15FFh	Empfangs-PDO (RxPDO) Kommunikations Parameter
1600h - 17FFh	Empfangs-PDO (RxPDO) Mapping Parameter
1800h - 19FFh	Sende-PDO (TxPDO) Kommunikations Parameter
1A00h - 1BFFh	Sende-PDO (TxPDO) Mapping Parameter

1000h - Device Type

Gibt das unterstützte Geräteprofil und zusätzliche Information zum Gerät an

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1000h	00h	Device Type	Unsigned32	ro	Mandatory	Nein

Dieses Objekt kann nur gelesen werden und enthält die Information, um welches Gerät es sich handelt (IEC 61131-3 Programmable Device nach DS 405).

Im Subindex 00h ist per default 00000195h (405dez) für das Optionsmodul CANopen-Slave für b maXX PLC eingestellt.

1001h - Fehleregister

Enthält eine Bitleiste mit anstehenden Geräte Fehlern

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1001h	00h	Error Register	Unsigned8	ro	Mandatory	Optional

Dieses Objekt kann nur gelesen werden. Das Objekt 1001h enthält eine Fehlerbitleiste, die folgende Bedeutung hat:

Bit	Bedeutung
0	Fehler aufgetreten
1	Nicht verwendet
2	Nicht verwendet
3	Nicht verwendet
4	Kommunikationsfehler CAN
5	Nicht verwendet
6	Nicht verwendet
7	Herstellerspezifische Fehler

Diese Fehlerbit-Leiste wird mit 80h OR-verknüpft (Herstellerspezifischer Fehler) wenn der FB CANop405_EM CY_SEND_SL aktiviert wird.

1003h - Predefined Error Field

Enthält eine Fehlerliste des Gerätes mit absteigender Aktualität.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1003h	-	Pre-defined error field	Array	-	Optional	-
	0h	Number of errors	Unsigned8	rw	Mandatory	Nein
	1h - FEh	Standard error field	Unsigned32	ro	Optional	Nein

Dieses Objekt kann nur auf Subindex 00h beschrieben werden. Es beinhaltet eine Fehlerhistory von max. 15 aufgetretenen Fehlern, wobei in Subindex 01h der zeitlich zuletzt aufgetretene Fehler steht.

In Subindex 00h dieses Objektes steht die Anzahl der eingetragenen Fehler. Durch Schreiben von "00h" auf Subindex 00h wird die Liste gelöscht.

Der Fehlercode ergibt sich nach der Definition des *Emergency Error Code* im Emergency-Telegramm:

Error Code	Bedeutung
00xxh	Error Reset or No Error
10xxh	Generic Error
20xxh	Current
21xxh	Current, device input side
22xxh	Current inside the device
23xxh	Current, device output side
30xxh	Voltage
31xxh	Mains Voltage
32xxh	Voltage inside the device
33xxh	Output Voltage
40xxh	Temperature
41xxh	Ambient Temperature
42xxh	Device Temperature
50xxh	Device Hardware
60xxh	Device Software
61xxh	Internal Software
62xxh	User Software
63xxh	Data Set
70xxh	Additional Modules
80xxh	Monitoring
81xxh	Communication
8110h	CAN Overrun (Objects lost)
8120h	CAN in Error Passive Mode

Error Code	Bedeutung
8130h	Life Guard Error or Heartbeat Error
8140h	recovered from bus off
8150h	Transmit COB-ID
82xxh	Protocol Error
8210h	PDO not processed due to length error
8220h	PDO length exceeded
90xxh	External Error
F0xxh	Additional Functions
FFxxh	Device specific

xx = nicht definiert

1004h - Number of PDOs supported

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1004h	00h	Number of PDOs supported	Unsigned32			
	01h	Number of errors	Unsigned32			
	02h	Standard error field	Unsigned32			

Dieses Objekt ist nur lesbar. Subindex 00h enthält die Gesamtzahl der unterstützten PDOs, wobei im High-Byte die Anzahl der Receive-PDOs und im Low Byte die Zahl der Transmit-PDO steht. Subindex 01h enthält die mögliche Anzahl an synchronen PDOs, Subindex 02h die mögliche Anzahl an asynchronen PDOs.

Die eingetragenen Werte bedeuten, dass vier Receive-PDOs und vier Transmit-PDOs zur Verfügung stehen, wobei jede der PDOs als synchron oder asynchron definiert werden kann.

1005h - COB-ID SYNC-Message

Enthält die COB-ID des SYNC-Telegramms und gibt an, ob das Gerät das SYNC-Telegramm generiert.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1005h	00h	COB-ID SYNC-Message	Unsigned32	rw	Conditional	Nein

Dieses Objekt enthält Informationen über das Sync-Verhalten des Slaves. Das MSB im High-Word besagt, dass der Slave das SYNC-Telegramm nicht erzeugt, sondern nur empfängt. Die unteren 11 Bit im Low Word geben den Identifier des SYNC-Telegramms an (80h).

1006h - Communication Cycle Period

Länge des SYNC-Intervall in μs .

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1006h	00h	Communication cycle period	Unsigned32	rw	Conditional	Nein

Falls das Sync-Telegramm aktiviert ist, muss das Sync-Intervall auf die Zeit des Sync-Telegramms (1000, 2000, 4000 oder 8000 μs) eingestellt werden. Die eingestellte Zeit wirkt sich auf die SYNC-Überwachung (siehe Kapitel [►Synchronisieren des Datenaustausch◄](#) auf Seite 67) aus.

1007h - Synchronous Window Length

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1007h	00h	Synchronous Window Length				

Dieses Objekt wird intern nicht ausgewertet.

1008h - Manufacturer Device Name

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1008h	00h	Manufacturer Device Name	VString			

Dieses Objekt ist nur lesbar. Es enthält die Zeichenfolge: "b maXX PLC".

1009h - Manufacturer Hardware Version

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1009h	00h	Manufacturer Hardware Version	VString			

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Hardware-Version des Optionsmoduls, z. B. die Zeichenfolge: "HV01.00".

100Ah - Manufacturer Software Version

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
100Ah	00h	Manufacturer Software Version	VString			

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Software-Version des Optionsmoduls, z. B. die Zeichenfolge: "SV01.00".

100Bh - Node-ID

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
100Bh	00h	Node-ID	Unsigned32			

Dieses Objekt ist nur lesbar. Es enthält die aktuell eingestellte Node-ID (Netzwerkknotenadresse). Nur das Low Byte ist gültig.

100Ch - Guarding Time

Abstand zwischen zwei Guarding Telegrammen in ms.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
100Ch	00h	Guard time	Unsigned16	rw	Conditional	Nein

In diesem Objekt wird die Zeitbasis für das Node Guarding in Millisekunden eingestellt. Durch beschreiben mit dem Wert "0" wird das Node Guarding deaktiviert.

100Dh - Life Time Faktor

Ergibt multipliziert mit der Guarding Time (Objekt 100Ch) die Life Time in ms.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
100Dh	00h	Life time factor	Unsigned8	rw	Conditional	Nein

Der Wert dieses Objektes wird mit Objekt 100Ch multipliziert und ergibt damit die Zeit für das Node Guarding.

100Eh - COB-ID Guarding Protocol

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
100Eh	00h	COB-ID Guarding Protocol	Unsigned32			

Dieses Objekt beinhaltet den Identifier des Node Guarding Protokolls.

100Fh - Number of SDOs supported

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
100Fh	00h	Number of SDOs supported	Unsigned32			

Dieses Objekt ist nur lesbar. Es enthält die Anzahl der unterstützten SDOs. Im High Word steht die Zahl der Client SDOs, die das Gerät unterstützt. Das Low Word enthält die Anzahl der Server SDOs des Gerätes.

1010h - Parameter speichern

Ermöglicht das Abspeichern von Parametern im nichtflüchtigen Speicher.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1010h	00h	Store parameters	Unsigned8			
	01h	Save all Parameters (not supported)	Unsigned32			
	02h	Save communication Parameters	Unsigned32			

Dieses Objekt unterstützt das Ablegen von Parametern in einem nichtflüchtigen Speicher. Subindex 00h ist nur lesbar und enthält den größten zu unterstützenden Subindex (hier 2). Subindex 01h wird nicht unterstützt. Die 1h im Subindex 02h zeigt an, dass ein Abspeichern, hier speziell von Mapping- und Kommunikations-Parametern, unterstützt wird. Das Speichern - nur im Zustand PRE-OPERATIONAL möglich - wird mit dem Wert 65766173h als U32 oder mit Wert "save" als String eingeleitet und dauert einige Millisekunden an. Es werden die zuletzt eingestellten Werte für die Mapping- und Kommunikations-Parameter abgespeichert.

1011h - Restore Parameters

Ermöglicht das Aktivieren von Default Parametern.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1011h	00h	Restore Parameters	Unsigned8	ro	Mandatory	Nein
	01h	Restore all Parameters (not supported)	Unsigned32			
	02h	Restore default communication Parameters	Unsigned32			

Bei diesem Objekt werden die Kommunikations-Parameter auf Default-Werte gesetzt. Subindex 00h ist nur lesbar und enthält den größten zu unterstützenden Subindex. Subindex 01h wird nicht unterstützt. Mit dem Wert 1h im Subindex 02h werden die Default-Werte der Mapping und Kommunikations-Parameter gesetzt. Dies geschieht nur im Zustand PRE-OPERATIONAL mit dem Eingeben des Wertes 64616F6Ch (U32) oder mit der Eingabe "load" als String - Parameter.

1014h - COB-ID Emergency

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1014h	00h	COB-ID Emergency	Unsigned32			

Dieses Objekt enthält den Identifier des EMCY-Telegramms.

1400h-15FFh - Empfangs-PDO (RxPDO) Kommunikations Parameter

Die Objekte 1400h bis 15FFh legen die Eigenschaften zur Übertragung der RxPDO 1 bis RxPDO 512 fest. Das Objekt 1400h beschreibt die RxPDO 1 und soweit dies vom Gerät unterstützt wird Objekt 15FFh die RxPDO 512.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1400h - 15FFh	-	RxPDO parameter	CiA definiert	-	Conditional	-
	0h	Largest sub-index supported	Unsigned8	rw	Mandatory	Nein
	1h	COB-ID Receive RxPDO	Unsigned32	ro / rw	Mandatory	Nein
	2h	Transmission type RxPDO	Unsigned8	ro / rw	Mandatory	Nein

Subindex 0 gibt die Nummer des letzten gültigen Subindex an. Die einzelnen Bit des 32 Bit-Wertes von **Subindex 1** haben folgende Bedeutung:

Subindex 1, Bit	Wert / Bedeutung
31 (MSB)	0: PDO existiert und ist gültig 1: PDO existiert nicht oder ist nicht gültig Nach der vordefinierten Identifizierung sind nur 4 PDOs für Schreiben und 4 PDOs für Lesen möglich. Unterstützt ein Gerät mehr als diese 4 PDOs, müssen die COB-IDs für die zusätzlichen PDOs selbst vergeben werden (Bits 10 - 0 in diesem Subindex). Aus diesem Grund hat das Bit 31 der zusätzlichen PDOs meist den Wert 1 und wird erst dann vom Anwender auf 0 gesetzt, sobald dieser eine COB-ID vergeben hat.
30	0: das PDO kann über einen Remote-Request angefordert werden 1: das PDO kann nicht über einen Remote-Request angefordert werden
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Slave nicht unterstützt)
28 - 11	Bits 28 - 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Slave nicht unterstützt)
10 - 0 (LSB)	COB-ID des PDO. Die ersten 4 PDOs für Lesen haben in den Default-Einstellungen eines Gerätes die COB-IDs nach der vordefinierten Identifizierung.

Mit **Subindex 2** wird der Übertragungstyp des PDO festgelegt:

Subindex 2, Wert	Typ	Bedeutung
0	Synchron	Vor dem letzten SYNC-Telegramm empfangenes PDO mit passender COB-ID wird übernommen
1 - 240	Synchron	Vor dem letzten SYNC-Telegramm empfangenes PDO mit passender COB-ID wird übernommen
252	Remote	<i>Nicht möglich</i>
253	Remote	<i>Nicht möglich</i>
254	Asynchron	Jedes PDO mit passender COB-ID wird bei Empfang übernommen
255	Asynchron	Jedes PDO mit passender COB-ID wird bei Empfang übernommen

1600h-17FFh - Empfangs-PDO (RxPDO) Mapping Parameter

Die Objekte 1600h bis 17FFh legen die gemappten Objekte der RxPDO 1 bis RxPDO 512 fest. Das Objekt 1600h beschreibt die RxPDO 1 und soweit dies vom Gerät unterstützt wird Objekt 17FFh die RxPDO 512.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1600h - 17FFh	-	RxPDO mapping	CiA definiert	-	Conditional	-
	0h	Number of mapped objects	Unsigned8	rw	Mandatory	Nein
	1h - 40h	Object to be mapped	Unsigned32	rw	Conditional	Nein

Subindex 0 gibt die Anzahl der gemappten Objekte in dem PDO an. Sobald das Mapping eines PDO geändert werden soll, muss zunächst die Anzahl der gemappten Objekte auf Null setzen. Anschließend kann das Mapping eingestellt werden und schließlich muss die Anzahl der gemappten Objekte in Subindex 0 eingetragen werden. In Subindex 1h bis 40h werden die Mapping Informationen der Objekte in dem PDO angegeben. Subindex 1h enthält die Informationen für das erste gemappte Objekt, Subindex 2h für das zweite gemappte Objekt usw.. Maximal können 64 (= 40h) Objekte gemappt werden, wobei der maximale Datenbereich von 8 Byte der PDO nicht überschritten werden darf. Die Inhalte des 32 Bit-Wertes der Subindizes 1h - 40h haben folgende Bedeutung:

Byte: MSB LSB

Index des gemappten Objektes (16 Bit)	Subindex des gemappten Objektes (8 Bit)	Anzahl der Bit des gemappten Objektes (8 Bit)
---------------------------------------	---	---

Die Objekte werden nacheinander in ein PDO gemappt.

1800h-19FFh - Sende-PDO (TxPDO) Kommunikations Parameter

Die Objekte 1800h bis 19FFh legen die Eigenschaften zur Übertragung der TxPDO 1 bis TxPDO 512 fest. Das Objekt 1800h beschreibt die TxPDO 1 und soweit dies vom Gerät unterstützt wird Objekt 19FFh die TxPDO 512.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1800h - 19FFh	-	TxPDO parameter	CiA definiert	-	Conditional	-
	0h	Largest sub-index supported	Unsigned8	rw	Mandatory	Nein
	1h	COB-ID Receive TxPDO	Unsigned32	ro / rw	Mandatory	Nein
	2h	Transmission type TxPDO	Unsigned8	ro / rw	Mandatory	Nein
	3h	Inhibit time TxPDO	Unsigned16	rw	Optional	Nein
	4h	Reserved	-	-	-	-
	5h	Event Timer TxPDO	Unsigned16	rw	Optional	Nein

Subindex 0 gibt die Nummer des letzten gültigen Subindex an. Die einzelnen Bit des 32 Bit-Wertes von **Subindex 1** haben folgende Bedeutung:

Subindex 1, Bit	Wert / Bedeutung
31 (MSB)	0: PDO existiert und ist gültig 1: PDO existiert nicht oder ist nicht gültig Nach der vordefinierten Identifizierung sind nur 4 PDOs für Schreiben und 4 PDOs für Lesen möglich. Unterstützt ein Gerät mehr als diese 4 PDOs, müssen die COB-IDs für die zusätzlichen PDOs selbst vergeben werden (Bits 10 - 0 in diesem Subindex). Aus diesem Grund hat das Bit 31 der zusätzlichen PDOs meist den Wert 1 und wird erst dann vom Anwender auf 0 gesetzt, sobald dieser eine COB-ID vergeben hat.
30	0: das PDO kann über einen Remote-Request angefordert werden 1: das PDO kann nicht über einen Remote-Request angefordert werden
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Slave nicht unterstützt)
28 - 11	Bits 28 - 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Slave nicht unterstützt)
10 - 0 (LSB)	COB-ID des PDO. Die ersten 4 PDOs für Schreiben haben in den Default-Einstellungen eines Gerätes die COB-IDs nach der vordefinierten Identifizierung

Mit **Subindex 2** wird der Übertragungstyp des PDO festgelegt:

Subindex 2, Wert	Typ	Bedeutung
0	Synchron	Senden erfolgt nach jedem empfangenen SYNC-Telegramm, aber nur dann, wenn sich die zu sendenden Werte geändert haben.
1 - 240	Synchron	Senden erfolgt nach Empfang der eingestellten Anzahl (1 - 240) von SYNC-Telegrammen
252	Remote	Bei Empfang eines SYNC-Telegramms erfolgt ein Update des PDO. Das Senden erfolgt erst nach Empfang einer Remote-Anforderung.
253	Remote	Ein Update und das Senden erfolgt nach Empfang einer Remote-Anforderung.
254	Asynchron	Senden erfolgt Hersteller spezifisch. Anmerkung: Die meisten Geräte übertragen bei diesem Typ das PDO bei einem Zeit-Event (Ablauf eines Timers). Die Zeit wird unter Subindex 5 eingestellt.
255	Asynchron	Senden erfolgt abhängig vom unterstützten Geräteprofil. Anmerkung: Die meisten Geräte übertragen bei diesem Typ das PDO sobald sich einer der gemappten Werte geändert hat.

Mit dem **Subindex 3** kann eine Sendeverzögerung in Vielfachen von 100 µs der PDO eingestellt werden, welche die Reaktionszeit bei der relativ ersten Wertänderung nicht verlängert, aber bei unmittelbar darauf folgenden Änderungen aktiv ist. Die Inhibit-Zeit (Sendeverzögerungszeit) beschreibt die Zeitspanne, die zwischen dem Versenden zweier gleicher Telegramme mindestens abgewartet werden muss.

In **Subindex 5** wird die Zeit in ms für Ereignis gesteuerte PDO Übertragung eingestellt sofern ein Gerät ereignisgesteuerte PDO-Übertragung unterstützt (Subindex 2 = 254 oder auch 255 abhängig vom Geräteprofil).

1A00h-1BFFh - Sende-PDO (TxPDO) Mapping Parameter

Die Objekte 1A00h bis 1BFFh legen die gemappten Objekte der TxPDO 1 bis TxPDO 512 fest. Das Objekt 1A00h beschreibt die TxPDO 1 und soweit dies vom Gerät unterstützt wird Objekt 1AFFh die TxPDO 512.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1A00h - 1BFFh	-	TxPDO mapping	CiA definiert	-	Conditional	-
	0h	Number of mapped objects	Unsigned8	rw	Mandatory	Nein
	1h - 40h	Object to be mapped	Unsigned32	rw	Conditional	Nein

Subindex 0 gibt die Anzahl der gemappten Objekte in dem PDO an. Sobald das Mapping eines PDO geändert werden soll, muss zunächst die Anzahl der gemappten Objekte auf Null setzen. Anschließend kann das Mapping eingestellt werden und schließlich muss die Anzahl der gemappten Objekte in Subindex 0 eingetragen werden. In **Subindex 1h bis 40h** werden die Mapping Informationen der Objekte in dem PDO angegeben. Subindex 1h enthält die Informationen für das erste gemappte Objekt, Subindex 2h für das zweite gemappte Objekt usw.. Maximal können 64 (= 40h) Objekte gemappt werden, wobei der maximale Datenbereich von 8 Byte der PDO nicht überschritten werden darf. Die Inhalte des 32 Bit-Wertes der **Subindices 1h bis 40h** haben folgende Bedeutung:

Byte: MSB LSB

Index des gemappten Objektes (16 Bit)	Subindex des gemappten Objektes (8 Bit)	Anzahl der Bit des gemappten Objektes (8 Bit)
---------------------------------------	---	---

Die Objekte werden nacheinander in ein PDO gemappt.

B.2 Objekte nach DS 405

Das Optionsmodul CANopen-Slave für b maXX PLC unterstützt folgende Objekte aus dem DS 405:

1000h - Device Type

Gibt das unterstützte Geräteprofil und zusätzliche Information zum Gerät an

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1000h	00h	Device Type	Unsigned32	ro	Mandatory	Nein

Dieses Objekt kann nur gelesen werden und enthält die Information, um welches Gerät es sich handelt (IEC 61131-3 Programmable Device nach DS 405).

Im Subindex 00h ist per default 00000195h (405dez) für das Optionsmodul CANopen-Slave für b maXX PLC eingestellt.

A00h bis A6FF

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A000h	01h	1 st Dynamic Integer8	Integer8	ro	Optional	Ja
...						
A03Fh	64h	64 th Dynamic Integer8	Integer8	ro	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A040h	01h	1 st Dynamic Unsigned8	Unsigned8	ro	Optional	Ja
...						
A07Fh	64h	64 th Dynamic Unsigned8	Unsigned8	ro	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A0C0h	01h	1 st Dynamic Integer16	Integer16	ro	Optional	Ja
...						
A0FFh	64h	64 th Dynamic Integer16	Integer16	ro	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A100h	01h	1 st Dynamic Unsigned16	Unsigned16	ro	Optional	Ja
...						
A13Fh	64h	64 th Dynamic Unsigned16	Unsigned16	ro	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A1C0h	01h	1 st Dynamic Integer32	Integer32	ro	Optional	Ja
...						
A1FFh	64h	64 th Dynamic Integer32	Integer32	ro	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A200h	01h	1 st Dynamic Unsigned32	Unsigned32	ro	Optional	Ja
...						
A23Fh	64h	64 th Dynamic Unsigned32	Unsigned32	ro	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A240h	01h	1 st Dynamic Float(32)	Float(32) (=Real)	ro	Optional	Ja
...						
A27Fh	64h	64 th Dynamic Float(32)	Float(32)	ro	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A480h	01h	1 st Dynamic Integer8	Integer8	rw	Optional	Ja
...						
A4BFh	64h	64 th Dynamic Integer8	Integer8	rw	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A4C0h	01h	1 st Dynamic Unsigned8	Unsigned8	rw	Optional	Ja
...						
A4FFh	64h	64 th Dynamic Unsigned8	Unsigned8	rw	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A540h	01h	1 st Dynamic Integer16	Integer16	rw	Optional	Ja
...						
A57Fh	64h	64 th Dynamic Integer16	Integer16	rw	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A580h	01h	1 st Dynamic Unsigned16	Unsigned16	rw	Optional	Ja
...						
A5BFh	64h	64 th Dynamic Unsigned16	Unsigned16	rw	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A640h	01h	1 st Dynamic Integer32	Integer32	rw	Optional	Ja
...						
A67Fh	64h	64 th Dynamic Integer32	Integer32	rw	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A680h	01h	1 st Dynamic Unsigned32	Unsigned32	rw	Optional	Ja
...						
A6BFh	64h	64 th Dynamic Unsigned32	Unsigned32	rw	Optional	Ja

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
A6C0h	01h	1 st Dynamic Float(32)	Float(32) (=Real)	rw	Optional	Ja
...						
A6FFh	64h	64 th Dynamic Float(32)	Float(32)	rw	Optional	Ja

B.3 Beispiel-EDS-Datei

Die Beispiel-EDS-Datei finden Sie auf der CD CANopen „CANop405_PLC01_20bd02“ (oder höher).

Außerdem können Sie die Beispiel-EDS-Datei mit dem CANopen-Slave Konfigurator im b maXX Drive-Configurator erzeugen. siehe hierzu auch [>Konfigurierung des Optionsmodul CANopen-Slave mit dem CANopen-Slave Konfigurator<](#) ab Seite 31.

- Öffnen Sie das PROPROGRAM-Projekt CANopenSlave_Example.mwt (Entpacken Sie gegebenenfalls CANopenSlave_Example.zwt von der CD CANopen „CANop405_PLC01_20bd02“ (oder höher)).
- Öffnen sie den CANopen-Slave Konfigurator aus PROPROGRAM wt II über Configuration_Tools → Drive_Configuration → „Optionsmodule \ G: CAN-03-00-00“ - Kontextmenü Punkt „Konfigurieren“.
- Speichern Sie die EDS-Datei. Siehe hierzu auch [>Erzeugen EDS-Datei<](#) ab Seite 42.

B.4 Default Mapping der PDO

In den Geräteprofilen sind Default-Mappings definiert. Nachfolgend finden Sie eine Übersicht der Default-Mapping-Werte für die Geräteprofile DS 401 und DSP 402. Die unten dargestellte Belegung bezieht sich auf den maximalen Ausbau eines Gerätes. Abhängig von der Hardware können weniger Objekte gemappt sein. Dies bedeutet jedoch nicht, dass Objekte eines anderen Typs (z. B. Analoge Eingänge statt digitale Eingänge) im Mapping nachrücken. Entfallen Objekte, so bleiben die zugehörigen Stellen im Default-Mapping frei. Natürlich steht es Ihnen frei die freien Plätze mit anderen Objekten zu belegen.

B.4.1 Default-Mapping für I/O-Module nach DS 401

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
1600h	Erstes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte			8
Subindex 1	Digitale Ausgänge 0 - 7	6200 01	08h	
Subindex 2	Digitale Ausgänge 8 - 15	6200 02	08h	
Subindex 3	Digitale Ausgänge 16 - 23	6200 03	08h	
Subindex 4	Digitale Ausgänge 24 - 31	6200 04	08h	
Subindex 5	Digitale Ausgänge 32 - 39	6200 05	08h	
Subindex 6	Digitale Ausgänge 40 - 47	6200 06	08h	
Subindex 7	Digitale Ausgänge 48 - 55	6200 07	08h	
Subindex 8	Digitale Ausgänge 56 - 63	6200 08	08h	
1601h	Zweites Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte			4

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
Subindex 1	Analoger Ausgang 1 (16 Bit Auflösung)	6411 01	10h	
Subindex 2	Analoger Ausgang 2 (16 Bit Auflösung)	6411 02	10h	
Subindex 3	Analoger Ausgang 3 (16 Bit Auflösung)	6411 03	10h	
Subindex 4	Analoger Ausgang 4 (16 Bit Auflösung)	6411 04	10h	
1602h	Drittes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte	4		
Subindex 1	Analoger Ausgang 5 (16 Bit Auflösung)	6411 05	10h	
Subindex 2	Analoger Ausgang 6 (16 Bit Auflösung)	6411 06	10h	
Subindex 3	Analoger Ausgang 7 (16 Bit Auflösung)	6411 07	10h	
Subindex 4	Analoger Ausgang 8 (16 Bit Auflösung)	6411 08	10h	
1603h	Viertes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte	4		
Subindex 1	Analoger Ausgang 9 (16 Bit Auflösung)	6411 09	10h	
Subindex 2	Analoger Ausgang 10 (16 Bit Auflösung)	6411 0A	10h	
Subindex 3	Analoger Ausgang 11 (16 Bit Auflösung)	6411 0B	10h	
Subindex 4	Analoger Ausgang 12 (16 Bit Auflösung)	6411 0C	10h	

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
1A00h	Erstes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte	8		
Subindex 1	Digitale Eingänge 0 - 7	6000 01	08h	
Subindex 2	Digitale Eingänge 8 - 15	6000 02	08h	
Subindex 3	Digitale Eingänge 16 - 23	6000 03	08h	
Subindex 4	Digitale Eingänge 24 - 31	6000 04	08h	
Subindex 5	Digitale Eingänge 32 - 39	6000 05	08h	
Subindex 6	Digitale Eingänge 40 - 47	6000 06	08h	
Subindex 7	Digitale Eingänge 48 - 55	6000 07	08h	
Subindex 8	Digitale Eingänge 56 - 63	6000 08	08h	
1A01h	Zweites Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte	4		
Subindex 1	Analoger Eingang 1 (16 Bit Auflösung)	6401 01	10h	
Subindex 2	Analoger Eingang 2 (16 Bit Auflösung)	6401 02	10h	

B.4 Default Mapping der PDO

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
Subindex 3	Analoger Eingang 3 (16 Bit Auflösung)	6401 03	10h	
Subindex 4	Analoger Eingang 4 (16 Bit Auflösung)	6401 04	10h	
1A02h	Drittes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte		4	
Subindex 1	Analoger Eingang 5 (16 Bit Auflösung)	6401 05	10h	
Subindex 2	Analoger Eingang 6 (16 Bit Auflösung)	6401 06	10h	
Subindex 3	Analoger Eingang 7 (16 Bit Auflösung)	6401 07	10h	
Subindex 4	Analoger Eingang 8 (16 Bit Auflösung)	6401 08	10h	
1A03h	Viertes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte		4	
Subindex 1	Analoger Eingang 9 (16 Bit Auflösung)	6401 09	10h	
Subindex 2	Analoger Eingang 10 (16 Bit Auflösung)	6401 0A	10h	
Subindex 3	Analoger Eingang 11 (16 Bit Auflösung)	6401 0B	10h	
Subindex 4	Analoger Eingang 12 (16 Bit Auflösung)	6401 0C	10h	

B.4.2 Default-Mapping für Antriebe nach DSP 402

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
1600h	Erstes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte		1	
Subindex 1	Steuerwort (16 Bit)	6040 00	10h	
1601h	Zweites Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	
Subindex 1	Steuerwort (16 Bit)	6040 00	10h	
Subindex 2	Betriebsart (8 Bit)	6060 00	08h	
1602h	Drittes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	
Subindex 1	Steuerwort (16 Bit)	6040 00	10h	
Subindex 2	Zielposition (32 Bit)	607A 00	20h	
1603h	Viertes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	
Subindex 1	Steuerwort (16 Bit)	6040 00	10h	

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
Subindex 2	Drehzahl Sollwert (32 Bit)	60FF	00	20h
1604h	Fünftes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	
Subindex 1	Steuerwort (16 Bit)	6040	00	10h
Subindex 2	Zieldrehmoment (32 Bit)	6071	00	20h
1605h	Sechstes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	
Subindex 1	Steuerwort (16 Bit)	6040	00	10h
Subindex 2	Drehzahl Sollwert (16 Bit)	6042	00	10h
1606h	Siebtes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	
Subindex 1	Steuerwort (16 Bit)	6040	00	10h
Subindex 2	Digitale Ausgänge (32 Bit)	6042	00	20h
1607h	Achstes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	
Subindex 1	Steuerwort (16 Bit)	6040	00	10h
Subindex 2	Betriebsart (8 Bit)	6060	00	08h

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
1A00h	Erstes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte		1	
Subindex 1	Statuswort (16 Bit)	6041	00	10h
1A01h	Zweites Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	
Subindex 1	Statuswort (16 Bit)	6041	00	10h
Subindex 2	Betriebsart Anzeige (8 Bit)	6061	00	08h
1A02h	Drittes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	
Subindex 1	Statuswort (16 Bit)	6041	00	10h
Subindex 2	Position Istwert(32 Bit)	6064	00	20h
1A03h	Viertes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte		2	

B.4 Default Mapping der PDO

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
Subindex 1	Statuswort (16 Bit)	6041	00	10h
Subindex 2	Drehzahl Istwert(32 Bit)	606C	00	20h
1A04h	Fünftes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte			2
Subindex 1	Statuswort (16 Bit)	6041	00	10h
Subindex 2	Drehmoment Istwert(32 Bit)	6077	00	20h
1A05h	Sechstes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte			2
Subindex 1	Statuswort (16 Bit)	6041	00	10h
Subindex 2	Drehzahl Istwert (16 Bit)	6044	00	10h
1A06h	Siebtens Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte			2
Subindex 1	Statuswort (16 Bit)	6041	00	10h
Subindex 2	Digitale Eingänge (32 Bit)	60FD	00	20h

B.4.3 Default-Mapping für programmierbare Steuerungen nach DS 405

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
1600h	Erstes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte			5
Subindex 1	Objekt A480 Sidx 01 (INTEGER8)	A480	01	08h
Subindex 2	Objekt A4C0 Sidx 01 (UNSIGNED8)	A4C0	01	08h
Subindex 3	Objekt A540 Sidx 01 (INTEGER16)	A540	01	10h
Subindex 4	Objekt A580 Sidx 01 (UNSIGNED16)	A580	01	10h
Subindex 5	Objekt A580 Sidx 02 (UNSIGNED16)	A580	02	10h
1601h	Zweites Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte			2
Subindex 1	Objekt A640 Sidx 01 (INTEGER32)	A640	01	20h
Subindex 2	Objekt A680 Sidx 01 (UNSIGNED32)	A680	01	20h
1602h	Drittes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte			4
Subindex 1	Objekt A480 Sidx 01 (INTEGER8)	A480	01	08h

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
Subindex 2	Objekt A4C0 Sidx 01 (UNSIGNED8)	A4C0	01	08h
Subindex 3	Objekt A540 Sidx 01 (INTEGER16)	A540	01	10h
Subindex 4	Objekt A680 Sidx 01 (UNSIGNED32)	A680	01	20h
1603h	Viertes Empfangs-PDO.			
Subindex 0	Anzahl der gemappten Objekte			3
Subindex 1	Objekt A540 Sidx 01 (INTEGER16)	A540	01	10h
Subindex 2	Objekt A580 Sidx 01 (UNSIGNED16)	A580	01	10h
Subindex 3	Objekt A680 Sidx 01 (UNSIGNED32)	A680	01	20h

Objekt	Bedeutung	Mapping-Wert		
		Index	Subindex	Länge
1A00h	Erstes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte			5
Subindex 1	Objekt A000 Sidx 01 (INTEGER8)	A000	01	08h
Subindex 2	Objekt A040 Sidx 01 (UNSIGNED8)	A040	01	08h
Subindex 3	Objekt A0C0 Sidx 01 (INTEGER16)	A0C0	01	10h
Subindex 4	Objekt A100 Sidx 01 (UNSIGNED16)	A100	01	10h
Subindex 5	Objekt A100 Sidx 02 (UNSIGNED16)	A100	02	10h
1A01h	Zweites Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte			2
Subindex 1	Objekt A1C0 Sidx 01 (INTEGER32)	A1C0	01	20h
Subindex 2	Objekt A200 Sidx 01 (UNSIGNED32)	A200	01	20h
1A02h	Drittes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte			4
Subindex 1	Objekt A000 Sidx 01 (INTEGER8)	A000	01	08h
Subindex 2	Objekt A040 Sidx 01 (UNSIGNED8)	A040	01	08h
Subindex 3	Objekt A0C0 Sidx 01 (INTEGER16)	A0C0	01	10h
Subindex 4	Objekt A200 Sidx 01 (UNSIGNED32)	A200	01	20h
1A03h	Viertes Sende-PDO.			
Subindex 0	Anzahl der gemappten Objekte			3
Subindex 1	Objekt A0C0 Sidx 01 (INTEGER16)	A0C0	01	10h
Subindex 2	Objekt A100 Sidx 01 (UNSIGNED16)	A100	01	10h
Subindex 3	Objekt A200 Sidx 01 (UNSIGNED32)	A200	01	20h

B.5 Default-EDS-Datei

Die Default-EDS-Datei finden Sie auf der CD CANopen „CANop405_PLC01_20bd02“ (oder höher).

Außerdem können Sie die Default-EDS-Datei mit dem CANopen-Slave Konfigurator im b maXX Drive-Configurator erzeugen. Siehe hierzu auch [►Konfigurierung des Optionsmodul CANopen-Slave mit dem CANopen-Slave Konfigurator◄](#) ab Seite 31.

- Öffnen sie den CANopen-Slave Konfigurator aus PROPROGRAMM II über Configuration_Tools → Drive_Configuration → „Optionsmodule \ G: CAN-03-00-00“ - Kontextmenü Punkt „Konfigurieren“.
- Aktivieren Sie die Check-Box „Default-Konfiguration“.
- Speichern Sie die EDS-Datei. Siehe hierzu auch [►Erzeugen EDS-Datei◄](#) ab Seite 42.



ANHANG C - BIBLIOTHEK CANop405_PLC01_20BD02

Die Bibliothek CANop405_PLC01_20bd02 enthält die Funktionsbausteine (FBs) für das Optionsmodul CANopen-Master und das Optionsmodul CANopen-Slave zur Programmierung des CANopen auf der b maXX PLC.

FB-Name	Optionsmodul CANopen-Master	Optionsmodul CANopen-Slave
CANop405_COB_ID	X	
CANop405_EMERGENCY	X	
CANop405_EMCY_SEND_SL		X
CANop405_GET_STATE_SL		X
CANop405_PDO_INIT	X	
CANop405_INIT	X	
CANop405_INIT_SL		X
CANop405_KERNEL_STATE	X	
CANop405_KERNEL_STATE_SL		X
CANop405_LIFE_GUARD_SL		X
CANop405_LOC_NODE_ID_SL		X
CANop405_NMT	X	
CANop405_NODE_GUARDING	X	
CANop405_PDO_READ	X	
CANop405_PDO_WRITE	X	
CANop405_SDO1_READ	X	
CANop405_SDO1_WRITE	X	
CANop405_SDO2_READ	X	
CANop405_SDO2_WRITE	X	

FB-Name	Optionsmodul CANopen-Master	Optionsmodul CANopen-Slave
CANop405_SDO3_READ	X	
CANop405_SDO3_WRITE	X	
CANop405_SDO4_READ	X	
CANop405_SDO4_WRITE	X	
CANop405_SDO5_READ	X	
CANop405_SDO5_WRITE	X	
CANop405_SDO6_READ	X	
CANop405_SDO6_WRITE	X	
CANop405_SDO7_READ	X	
CANop405_SDO7_WRITE	X	
CANop405_SDO8_READ	X	
CANop405_SDO8_WRITE	X	
CANop405_SYNC	X	



Index

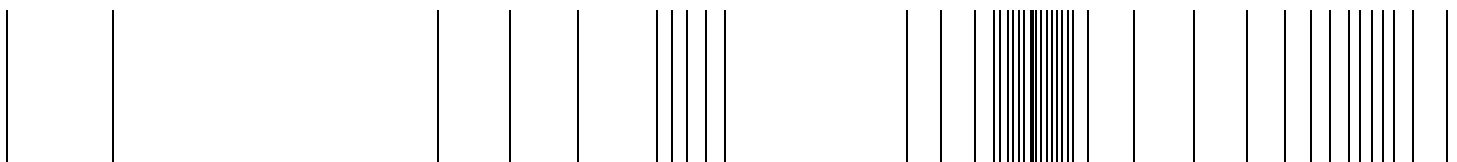
A		Erste Schritte	5
Ausfallüberwachung		F	
Heartbeat	71	Fachkraft	13
Netzwerkknoten	70	Fehlerbehandlung	19
B		G	
b maXX Konfigurator	27	Gewährleistung und Haftung	14
b maXX Regler	6	Globale Variable anlegen	25
Baudrate		Grundgerät	5
einstellen	32	H	
Baumüller	7	Hamming-Distanz	17
Begriffe		Heartbeat	71
Definition	5	Heartbeat-Telegramm	83
BM4-O-CAN-03	5	I	
Broadcast-Objekte	20	Identifizier	16
Buszugriff	16	vordefiniert	19
C		INITIALISIERUNG	46
CAL-Spezifikation	17	Initialisierung	44
CAN		K	
Grundlagen	16	Kernel-State	73
CANop405_KERNEL_STATE_SL	73	Kommunikation	
CANop405_LOC_NODE_ID_SL	72	Master - Slave	23
CANopen		PLC - Regler	23
Grundlagen	17	Regler - Leistungsteil	23
CANopen-Slave	5	Slave - PLC	23
CANopen-Telegramm		Kommunikationsparameter	55
Aufbau	76	Kommunikationszustände	46
Client - Server Modell	18	Konfiguration	
Command Specifier NMT	76	PDOs	54
Command Specifier SDO	77	Konsistenz der Daten	74
CSMA/CA-Verfahren	16	L	
D		LEDs	46
Datenaustausch	18, 22	Life Guarding	70
Datenaustausch über PDOs	54	Local Node-ID	72
Datenkonsistenz	74	M	
Drive_Configuration	27	Mapping von Objekten	60
E		Module-ID	19
EditBox		Multiplexor	77
Initialwert	36	N	
Name	36	Netzwerk	16
Object	38	Netzwerkcommandos	46
Typ	35	Netzwerkmanagement	18
EDS-Datei		Netzwerküberwachung	19
erzeugen	42	NMT Telegrammaufbau	76
Einleitung	5	NODE GUARDING	19
EMERGENCY	19	Node Guarding	70
Emergency-Telegramm	49, 84	Node Guarding Telegrammaufbau	82
senden	51		
Empfangs-PDO	95		
Empfangs-PDOs	55		
Ereignis gesteuertes Senden	57		



Stichwortverzeichnis

Node-Guarding		Time Stamp	20
Ausfall erkennen	71		
Node-ID		U	
einstellen	32	Übertragungseigenschaften eines PDO	54
Local	72	Übertragungstyp	55
O		V	
Objekt	17	Variablen	
Objekte nach DS 405	101	anlegen	34
Objekte über PDO lesen	66	Verpflichtung und Haftung	14
Objekte über PDO schreiben	66		
Objektverzeichnis nach DS301	87	Z	
OPERATIONAL	46	Zeit gesteuertes Senden	57
Optionsmodul	5		
P			
PDO			
Ereignis gesteuertes Senden	57		
Zeit gesteuertes Senden	57		
PDO Telegrammaufbau	80		
PDO-Kommunikation	53		
Peer-to-Peer-Objekte	20		
Personal	13		
qualifiziert	13		
PRE-OPERATIONAL	46		
Priorität	21		
Producer - Consumer Modell	18		
Projekt anlegen	24		
PROPROG wt II Projekt	24		
Prozessdatenaustausch	66		
Prozess-Daten-Objekte	18		
Q			
Qualifiziertes Personal	13		
S			
SDO Server	79		
SDO Telegrammaufbau	77		
SDO-Kommunikation	52		
SDO-Transfer	78		
Sende-PDO	98		
Sende-PDOs	55		
Service-Daten-Objekte	18		
Sicherheitshinweise	7		
Steckmodul	5		
STOPPED	46		
SYNC Telegrammaufbau	81		
SYNC-Consumer	81		
Synchronisierung	18, 69		
SYNC-Producer	81		
SYNC-Telegramm	68		
Ausfall erkennen	70		
T			
Telegrammaufbau	21		

be in motion



Baumüller Nürnberg GmbH Ostendstraße 80-90 90482 Nürnberg T: +49(0)911-5432-0 F: +49(0)911-5432-130 www.baumueller.de

Alle Angaben in dieser Betriebsanleitung sind unverbindliche Kundeninformationen, unterliegen einer ständigen Weiterentwicklung und werden fortlaufend durch unseren permanenten Änderungsdienst aktualisiert. Bitte beachten Sie, dass Angaben/Zahlen/Informationen aktuelle Werte zum Druckdatum sind.
Zur Ausmessung, Berechnung und Kalkulationen sind diese Angaben nicht rechtlich verbindlich. Bevor Sie in dieser Betriebsanleitung aufgeführte Informationen zur Grundlage eigener Berechnungen und/oder Verwendungen machen, informieren Sie sich bitte, ob Sie den aktuellsten Stand der Informationen besitzen.
Eine Haftung für die Richtigkeit der Informationen wird daher nicht übernommen.