BAUMÜLLER

be in motion be in motion

BM4-O-ETH-01/2, ...-CAN-04

Ethernet mit CANopen-Master für b maXX drive PLC

Applikationshandbuch

D	5.03002.02

Titel	Applikationshandbuch
Produkt	Ethernet mit CANopen-Master für b maXX drive PLC BM4-O-ETH-01/2, ...-CAN-04
Stand	28. Feb. 2007

Verbindlichkeit

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.

Mit der Übergabe des vorliegenden Applikationshandbuches werden entsprechende Applikationshandbücher mit einem früheren Stand außer Kraft gesetzt.
Bitte beachten Sie, dass Angaben/Zahlen/Informationen **aktuelle Werte zum Druckdatum** sind. Zur Ausmessung, Berechnung und Kalkulationen sind diese Angaben **nicht rechtlich verbindlich**.
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.

Es kann jedoch keine Gewährleistung bezüglich der Fehlerfreiheit dieses Applikationshandbuches, soweit nicht in den Allgemeinen Verkaufs- und Lieferbedingungen anders beschrieben, übernommen werden.

Hersteller

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

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 Ethernet mit CANopen zu programmieren, benötigen Sie folgende Hardware:
Grundgerät b maXX 4400,
Optionsmodul b maXX PLC und
Optionsmodul ETH-01, ETH-02 oder CAN-04.
Die Hardware muss entsprechend der jeweiligen Betriebsanleitung installiert und betriebsbereit sein.

2 Außerdem benötigen Sie folgende Software:

- ProMaster zur Konfiguration eines CANopen-Netzwerks
- ProProg wt III zur Programmierung der b maXX drive PLC und des Optionsmoduls Ethernet mit CANopen-Master für b maXX drive PLC

oder

- PROPROG wt II zur Programmierung der b maXX drive PLC und des Optionsmoduls Ethernet mit CANopen-Master für b maXX drive PLC.

HINWEIS

Zur Programmierung der b maXX drive PLC BM4-O-PLC-01 in Verbindung mit ProMaster wird das Programmiersystem ProProg wt III benötigt.

1.2 Verwendete Begriffe

Für die Baumüller-Produkte „BM4-O-ETH-01“ (Optionsmodul Ethernet), „BM4-O-ETH-02“ (Optionsmodul Ethernet mit CANopen-Master) bzw. „BM4-O-CAN-04“ (Optionsmodul CANopen-Master) werden wir in dieser Dokumentation auch die Begriffe „Steckmodul

Ethernet“ für BM4-O-ETH-01 oder BM4-O-ETH-02 bzw. „Steckmodul CANopen-Master“ für BM4-O-ETH-02 oder BM4-O-CAN-04 verwenden.

Der Begriff „Optionsmodul“ allein wird verwendet, wenn sich der Inhalt allgemein auf die Produkte ETH-01, ETH-02 oder CAN-04 bezieht.

Für das Produkt „Grundgerät b maXX 4400“ wird auch der Begriff „b maXX“ verwendet. Der Regler im Grundgerät wird auch „b maXX Regler“ genannt.

Eine Liste der verwendeten Abkürzungen finden Sie in ▹Abkürzungen◃ ab Seite 47.

1.3 Voraussetzungen

Dieses Handbuch baut auf das „Applikationshandbuch b maXX PLC“ auf und setzt die Kenntnis des Programmiertools PROPROG wt II bzw. ProProg wt III und des zugehörigen Handbuchs voraus.

2

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. Die konkreten Schutzmaßnahmen werden wir in den nachfolgenden Kapiteln immer direkt nach dem Hinweis auf die Gefahr geben.

2.1 Gefahrenhinweise und Gebote

WARNUNG (WARNING)

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

- schwere Körperverletzung
- Tod

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)

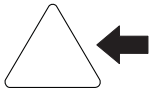
- erheblicher Sachschaden
- schwere Körperverletzung
- Tod - **kann** eintreffen

VORSICHT (CAUTION)

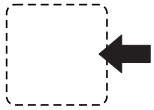
- Sachschaden • leichte bis mittlere Körperverletzung - **kann** eintreffen

2.1.1 Struktur eines Gefahrenhinweises

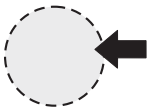
Die nachfolgenden zwei Beispiele zeigen den prinzipiellen Aufbau eines Gefahrenhinweises. Ein Dreieck wird verwendet, wenn vor einer Gefahr für Lebewesen gewarnt wird. Fehlt das Dreieck, beziehen sich die Gefahrenhinweise ausschließlich auf Sachschäden.



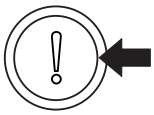
Ein Dreieck zeigt, dass hier eine Gefahr für Lebewesen ist.
Die Farbe der Umrandung zeigt, wie groß die Gefahr ist - je dunkler die Farbe, desto größer ist die Gefahr.



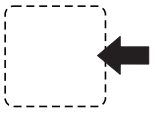
Das Icon im Viereck stellt die Gefahr dar.
Die Farbe der Umrandung zeigt, wie groß die Gefahr ist - je dunkler die Farbe, desto größer ist die Gefahr.



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.

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 in der Betriebsanleitung dieses Optionsmoduls 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 Bedienungspersonal

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 der Betriebsanleitung für dieses Optionsmodul 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 für dieses Optionsmodul 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 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 9 .
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 14 oder unten aufgeführten Ursachen den Schaden bewirkt hat/haben:

- Eintritt eines Katastrophenfalls durch Fremdkörpereinwirkung bzw. höhere Gewalt

3

ETHERNET

In diesem Kapitel finden Sie Informationen zum Anschluss des Optionsmoduls an Ethernet. Ethernet ist nur verfügbar beim Optionsmodul BM4-O-ETH-01 und BM4-O-ETH-02.

3.1 Allgemeines zu Ethernet und dem Optionsmodul Ethernet

Ethernet ist eine Technologie, die sich für die Datenübertragung in der Informationstechnik und in der Bürokommunikation bewährt und etabliert hat. Als Übertragungsmedium wird für das Optionsmodul Ethernet eine verdrillte Zweidrahtleitungen (siehe hierzu Betriebsanleitung des Optionsmodul) benutzt. Eine Anbindung an vorhandene 10BaseT oder 100BaseTX (Norm IEEE 802.3) kann somit einfach realisiert werden.

Das Optionsmodul Ethernet kann in einem Netzwerk mit anderen Komponenten wie PCs, Hubs, Switches und Repeater eingesetzt werden. Alternativ ist auch eine direkte Verbindung mit der Netzwerkkarte eines PCs über ein Crosslink-Kabel möglich, es werden dann keine zusätzlichen Netzwerk-Komponenten benötigt.

Für die Kommunikation ist aufbauend auf Ethernet zusätzlich ein TCP/IP-Protokollstack und das Applikations-Protokoll

- PROPROG / OmegaOS Kommunikation

implementiert.

Diese ermöglicht Ihnen die Anwendung von z. B. folgenden Funktionalitäten für PROPROG wt II und OmegaOS:

- SPS-Steuer-Funktionen wie Start und Stopp
- Programm-Verwaltung
- Online-Debugging, Watchfenster
- LogicAnalysator/Oszilloskop
- OPC-Server.

Details und weitere Funktionalitäten hierzu entnehmen Sie bitte dem Applikationshandbuch zur b maXX drive PLC und zum OPC-Server.

Um die genannten Funktionen nutzen zu können, müssen Sie lediglich in Ihrem PROPROG wt II Projekt den Kommunikations-Port und die IP-Adresse einstellen. Auf der Seite des Optionsmoduls sind IP-Adresse, Subnetzmaske und Gateway einzustellen bzw. zu codieren falls Sie die Default-Einstellungen nicht verwenden können.

Welche Schritte im einzelnen durchzuführen sind, finden Sie im Kapitel ▹Ethernet-TCP/IP Netzwerke konfigurieren◃ ab Seite 34.

In den nachfolgenden Kapiteln finden Sie eine Einführung in den Aufbau von Ethernet-TCP/IP Netzwerken und die notwendigen Voraussetzungen für den Aufbau einer Kommunikationsverbindung zwischen dem Optionsmodul Ethernet und anderen Komponenten im Netzwerk.

HINWEIS

Ethernet-TCP/IP Netzwerke können in ihrer Verzweigung sehr komplex sein. Zudem können viele Komponenten installiert sein, welche unterschiedlich konfiguriert werden und auch unterschiedlich konfiguriert sind. Diese Komponenten haben Einfluss auf den Datenverkehr im Netz. Darüber hinaus sind Aspekte der Datensicherheit und von unberechtigten Netzwerkzugriffen zu berücksichtigen. Aufgrund dieser Komplexität kann dieses Handbuch nur eine prinzipielle Übersicht über die Verwendung des Optionsmoduls Ethernet in Ethernet-TCP/IP Netzwerken liefern. Um Details zu Ihrem lokalen Netzwerk (LAN) zu erfahren wenden Sie sich bitte an Ihren Netzwerkadministrator und entnehmen Details zu den jeweiligen Netzwerkkomponenten den zugehörigen Handbüchern.

3.2 Grundlagen zu Ethernet-Netzwerken und TCP/IP

3.2.1 Übertragungsstandards

Das Optionsmodul Ethernet unterstützt die beiden gebräuchlichsten physikalischen Grundmodelle der Norm IEEE 802.3:

10BaseT

Jeder Netzwerkteilnehmer wird über ein eigenes Twisted-Pair-Kabel an einen Hub (Sternverteiler) angeschlossen, der alle Datenpakete gleichermaßen an alle Netzwerkteilnehmer weitergibt. Als Steckverbinder werden 8-polige RJ45-Typen eingesetzt. Die max. Länge eines Segments (= Verbindung vom Sternverteiler zum Endgerät) ist auf 100 m begrenzt.

100BaseTX

Der Übertragungsstandard 100BaseTX ist dem Standard 10BaseT ähnlich. Jeder Netzwerkteilnehmer ist ebenfalls über ein eigenes Twisted-Pair-Kabel an einen Hub (Sternverteiler) angeschlossen, der alle Datenpakete gleichermaßen an alle Netzwerkteilnehmer weitergibt. Als Steckverbinder werden ebenfalls 8-polige RJ45-Typen eingesetzt. Die max. Länge eines Segments (= Verbindung vom Sternverteiler zum Endgerät) beträgt auch 100 m. Es müssen jedoch alle Komponenten (Kabel, RJ45-Wanddosen, etc.) für die höhere Übertragungsrate von 100 MHz ausgelegt sein.

Das Optionsmodul Ethernet erkennt automatisch die verwendete Übertragungsrate.

3.2.2 Aufbau von Ethernet-Netzwerken mit Hub und Switch

Zum Aufbau eines sternförmigen Netzwerkes gibt es verschiedene Möglichkeiten.

Alle Teilnehmer sind über einen Hub verbunden.

Wird von einem Teilnehmer ein Datenpaket gesendet, so wird dieses im gesamten Netz veröffentlicht und steht jedem angeschlossenen Knoten zur Verfügung. Die Weiterverarbeitung der Nachricht erfolgt jeweils nur durch den Knoten mit der richtigen Zieladresse. Durch das hohe Datenaufkommen können Kollisionen auftreten und Nachrichten müssen wiederholt übertragen werden.

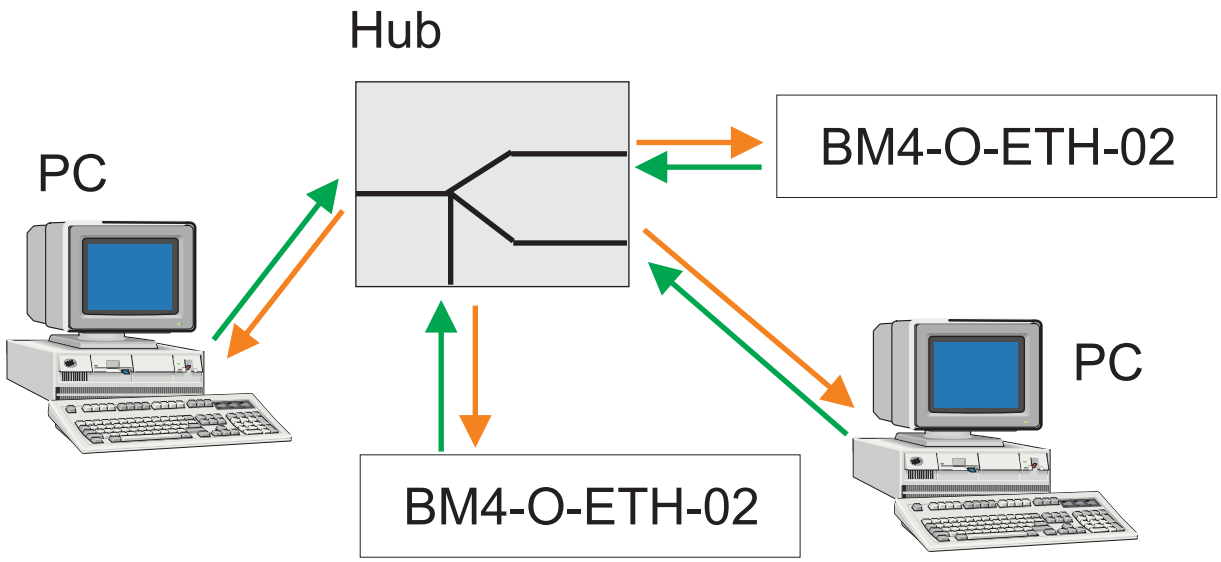


Abbildung 1: Prinzip von Shared Ethernet

Alle Teilnehmer sind über einen Switch verbunden.

Bei einem "Switched Ethernet" wird zur Kopplung mehrerer Teilnehmer ein Switch eingesetzt. Gelangen zu dem Switch Daten aus einem Segment, so wird geprüft, in welches Segment diese Daten gesendet werden sollen. Die Daten werden ausschließlich an den Teilnehmer mit der richtigen Zieladresse übermittelt. Das Datenaufkommen und somit die Kollisionsgefahr im Netz wird verringert.

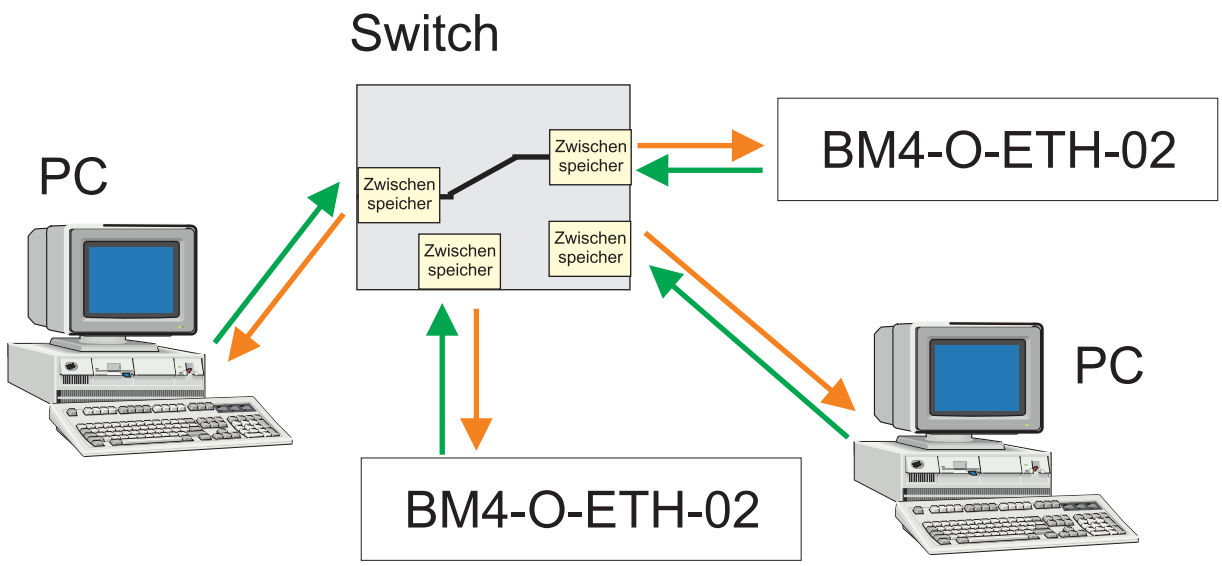


Abbildung 2: Prinzip von Switched Ethernet

Durch Kombination von sternförmigen Strukturen können Baumstrukturen gebildet werden. Darüber hinaus kann die Verbindung zwischen den Teilnehmern auch auf TCP/IP-Ebene erfolgen (siehe Kapitel ▹TCP/IP◃ ab Seite 21)

3.2.3 Ethernet-Adresse und MAC-Adresse

Jedes Optionsmodul Ethernet hat eine eigene, weltweit einmalige Ethernet-Adresse. Eine Ethernet-Adresse wird auch MAC-Adresse oder MAC-ID (Media Access Control Identity) genannt. Die Ethernet-Adresse ist im Optionsmodul fest hinterlegt und kann nicht geändert werden. Beachten sie den Unterschied zur IP-Adresse, welche eine andere Bedeutung hat und verändert werden kann. Die MAC-Adresse wird auf Ethernet-Ebene zur Adressierung verwendet. Sie besitzt eine feste Länge von 6 Byte (48 Bit) und beinhaltet den Herstellercode sowie eine fortlaufende Herstellernummer.

3.2.4 Ethernet-Datenpaket

Die Übertragung der Datenpakete auf dem Ethernet erfolgt verbindungslos, d. h. der Sender erhält keine Rückmeldung von dem Empfänger. Die Nutzdaten werden in einen Rahmen von Adressinformationen gepackt. Der Aufbau eines solchen Ethernet-Datenpakets sieht wie folgt aus:

Preamble	Destination	Source	Type	Data	FCS

Abbildung 3: Ethernet-Datenpaket

Preamble: Bitfolge zur Erkennung des Paketanfangs

Destination: Ethernet-Adresse des Empfängers

Source:	Ethernet-Adresse des Absenders
Type:	Gibt den übergeordneten Verwendungszweck an (z. B. IP)
Data:	Nutzdaten, enthalten die übergeordneten Protokolle wie z. B. IP
FCS:	Checksumme

Die Präambel dient zur Synchronisation zwischen Sende- und Empfangsstation.

Der Ethernet-Header beinhaltet die MAC-Adressen des Senders und des Empfängers und ein Typfeld zur Identifikation des im Datenbereich enthaltenen, nachfolgenden Protokolls.

3.2.5 Bridge und Repeater

Ebenfalls auf Ebene des Ethernet (Datenweiterleitung anhand der MAC-ID) arbeiten Bridges. Bridges verbinden Teilnetze miteinander und entscheiden anhand der MAC-ID, welche Datenpakete die Bridge passieren dürfen und welche nicht. Im Unterschied zum Switch leitet eine Bridge keine Broadcastmeldungen (Datenpaket an mehrere Teilnehmer gleichzeitig) weiter.

Repeater sind Verstärker zur Signalauffrischung und lassen den Inhalt der Datenpakete unverändert. Erkennt ein Repeater auf einem der angeschlossenen Segmente einen Fehler, so wird die Verbindung zu diesem Segment abgetrennt. Sobald der Fehler behoben ist, wird die Verbindung wieder hergestellt.

3.2.6 TCP/IP

Ethernet alleine reicht noch nicht um mehrere verschiedenartige Netze zur Datenübertragung miteinander zu verbinden. Betrachten wir z. B. einen PC der über ein externes Modem mit einem Internet-Service-Provider verbunden ist und über diese Verbindung mit anderen Servern im Internet kommunizieren kann. Hier kommen zum Ethernet auch noch die serielle oder USB Verbindung zum Modem und die Telefonleitung als Übertragungsmedium hinzu. Um eine Verbindung auch außerhalb des Ethernet fortzuführen, wurde das TCP/IP Protokoll entwickelt. TCP/IP wird zwar meist als gemeinsamer Begriff verwendet, es ist aber zwischen zwei verschiedenen Protokollen zu unterscheiden.

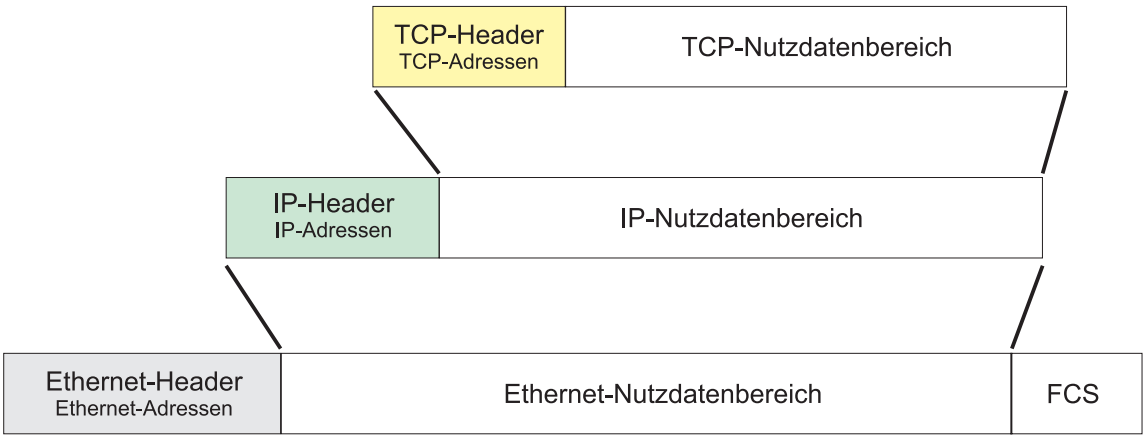


Abbildung 4: Aufbau eines TCP/IP-Ethernet Datenpaket

3.2.6.1 IP-Protokoll

Mit **IP (Internet Protocol)** kann man die Grenzen eines LAN (lokales Netz) überwinden. IP übernimmt die richtige Adressierung und Zustellung von Datenpaketen über ein Gateway auch in andere Netze. IP liegt im Ethernet-Nutzdatenbereich. Das heißt die von höheren Schichten erhaltenen Daten verpackt IP in einen eigenen Rahmen. Dieses Paket wird an das darunterliegende Ethernet übergeben und stellt die Nutzdaten eines oder mehrerer Ethernet-Telegramme dar.

Diese IP-Rahmen enthalten u. a. eine neue Form von Adressen (Internet-Adresse, IP-Nummer). Unter IP hat jeder Netzteilnehmer eine eindeutige (d. h. zumindest in einem bestimmtem Subnetz einmalige) Internet-Adresse. Sie ist Basis für die Weiterleitung über die Grenzen eines Ethernetsegmentes hinweg und die Kopplung mit nicht-ethernet-basierten Netzen.

a) Router

Die Verbindung zweier oder mehrerer verschiedenartiger IP-Netze erfolgt über Router. Diese entscheiden anhand der IP-Adresse, ob ein Datenpaket an ein anderes Netz weitergeleitet wird oder nicht.

b) IP-Adresse

Die IP-Adresse darf nicht mit der MAC-ID (oder Ethernet-Adresse) verwechselt werden.

Die Vergabe der IP-Adresse erfolgt durch den Anwender. Im Auslieferungszustand hat das Optionsmodul Ethernet die IP-Adresse 192.168.1.1 voreingestellt. Wie sie diese Einstellung ändern können, finden Sie in Kapitel ▹Optionsmodul Ethernet konfigurieren und Einstellungen überprüfen◃ ab Seite 38.

c) Aufbau einer IP-Adresse

Insgesamt besteht die IP-Adresse aus 4 Byte, die normalerweise dezimal dargestellt und durch Punkte getrennt werden, also z. B. 192.168.1.1.

Wird das Optionsmodul in einem Netz verwendet, dessen Teilnehmer nur über einen Hub oder Switch verbunden sind und keine Verbindung zu einem anderen Netzwerk hat, können Sie die IP-Adresse für jeden Teilnehmer nahezu frei vergeben. Es muss nur beachtet werden, dass niemals alle Bits in einem Byte gleich 0 oder gleich 1 gesetzt sind (Byte = 0 oder 255). Diese sind für spezielle Funktionen reserviert und dürfen nicht vergeben werden. Z. B. darf die Adresse 194.11.0.13 wegen der 0 im dritten Byte nicht verwendet werden.

Sollte eine Verbindung zu einem anderen Netzwerk oder zum Internet bestehen, können Sie die IP-Adresse nicht mehr frei vergeben, mit Ausnahmen:

Für den Gebrauch in privaten Netzen sind durch eine entsprechende IP-Norm drei Adressklassen reserviert, die im Internet nicht geroutet werden und somit dort auch nicht sichtbar sind. Es handelt sich hierbei um folgende Adressen / Adressbereiche:

- 10.xxx.xxx.xxx
- 172.16.xxx.xxx - 172.31.xxx.xxx
- 192.168.x.x

Auch Router im LAN sind meist so eingestellt, dass diese Adressbereiche nicht geroutet werden. Sollten Sie mit einem Optionsmodul Ethernet nicht kommunizieren können, kann es daran liegen, dass der Router in Ihrem LAN die eingestellte IP-Adresse nicht routet. Die Default-IP-Adresse 192.168.1.1 des Optionsmodul Ethernet ist eine solche IP-Adresse. Fragen Sie Ihren Netzwerkadministrator, welchen IP-Adressbereich Sie verwenden dürfen, wenn Verbindung mit einem anderen Netzwerk besteht.

Eine besondere Rolle spielt die Adresse 127.0.0.1 - diese Adresse adressiert immer das lokale/eigene Gerät. Laut Standard, ist die Verwendung des Netzes 127.x.x.x unzulässig. Die 127.0.0.1 kann demnach lediglich genutzt werden, um die Installation des eigenen Geräts zu überprüfen (z. B. bei PCs).

Soll ein Netzwerk direkt mit dem Internet verbunden werden, so können nur von einer zentralen Vergabestelle zugeteilte weltweit einmalige IP-Adressen verwendet werden. Die Vergabe ist abhängig vom Land in dem das Netzwerk betrieben wird.

HINWEIS

Sollte sich das Optionsmodul in einem Netzwerk befinden, welches eine Verbindung zu einem anderen Netzwerk hat oder wollen Sie eine direkte Internetanbindung durchführen, so sind umfangreiche Kenntnisse des gesamten LAN, der Vergabe der IP-Adressen, der Routingmechanismen und insbesondere der Sicherheitsanforderungen notwendig. Eine Verbindung zu einem anderen Netzwerk oder eine Internetanbindung sollte daher nur zusammen mit einem autorisierten Netzwerkadministrator durchgeführt werden.

d) Subnetzmaske und Gateway

Um Verbindung mit einem anderen Netzwerk herzustellen, muss dieses Netzwerk adressiert werden. Hierzu dient die Subnetzmaske. Soll ein Datenpaket von einem Netzwerkteilnehmer versendet werden, wird grundsätzlich die eigene IP-Adresse mit der Subnetzmaske AND verknüpft und auch die Ziel-IP-Adresse mit der Subnetzmaske AND verknüpft. Erhält der Netzwerkteilnehmer beide Male das gleiche Ergebnis, weiß er, dass sich der andere Teilnehmer im gleichen Netz befindet. Sind die Ergebnisse unterschiedlich kann der andere Netzwerkteilnehmer nicht direkt adressiert werden. In diesem Fall übergibt er das Datenpaket zur weiteren Vermittlung an einen Gateway oder Router. Ein Gateway unterscheidet sich von einem Router dadurch, dass Gateways einen Zugang zu Nicht-TCP/IP Netzwerken schaffen können. Da der Empfänger des Datenpaketes für die Antwort ebenfalls die Verknüpfung mit der Subnetzmaske durchführt, muss auch beim Empfänger die Subnetzmaske richtig eingestellt sein.

HINWEIS

Da ein Router eigentlich nur ein Sonderfall des Gateways ist, wird bei der Konfiguration der IP-Adresse oft nur der Begriff "Gateway" verwendet, obwohl physikalisch ein Router vorhanden ist. Dies betrifft insbesondere die Netzwerkkonfiguration in Windows-Betriebssystemen. Auch beim Optionsmodul Ethernet wird der allgemeinere Begriff "Gateway" verwendet.

Die Subnetzmaske und die Gateway- oder Router-IP-Adresse werden vom Anwender vergeben. Wie Sie diese Einstellungen beim Optionsmodul Ethernet vornehmen, entnehmen Sie bitte dem Kapitel ▹Optionsmodul Ethernet konfigurieren und Einstellungen überprüfen◃ ab Seite 38. Die Default-Einstellungen sind:

- Subnetzmaske: 255.255.255.0
- Gateway: 0.0.0.0

Die Gateway-IP-Adresse 0.0.0.0 beim Optionsmodul Ethernet bedeutet, dass kein Gateway verwendet wird. Das Datenpaket wird also auch gesendet, wenn festgestellt wird, dass der Empfänger anhand der Subnetzmasken-Verknüpfung nicht im gleichen Netzwerk liegt.

e) Subnetzmaske und Netz-Klasse

Vom InterNIC (International Network Information Center) wurden durch die Aufteilung der IP-Adresse in "Netzwerk-Teil" und "Host-Teil" sogenannte Adressklassen geschaffen. Die nachfolgende Tabelle zeigt die unterschiedlichen Adressklassen, die zugeordneten Werte der höchstwertigsten Bit der IP-Adresse und die Aufteilung in "Netzwerk-Teil" und "Host-Teil".

Adress-klasse	**Beschreibung**	**Adressbereich des Netzwerk-Teils**	**Mögliche Anzahl von Hosts**
Class A	Das erste Byte der IP-Adresse dient der Adressierung des Netzwerk-Teil, die letzten drei Byte adressieren den Host-Teil	1.xxx.xxx.xxx bis 126.xxx.xxx.xxx	Ca. 16 Mio.
Class B	Die ersten zwei Byte der IP-Adresse dienen der Adressierung des Netzwerk-Teil, die letzten zwei Byte adressieren den Host-Teil	128.0.xxx.xxx bis 191.255.xxx.xxx	Ca. 65 Tausend
Class C	Die ersten drei Byte der IP-Adresse dienen der Adressierung des Netzwerk-Teil, das letzte Byte adressiert den Host-Teil	192.0.0.xxx bis 223.255.255.xxx	254

Daneben gibt es noch Class D und Class E-Netze, die aber in der Praxis wenig Bedeutung haben.

Diese Einteilung hat auch Auswirkungen auf die Subnetzmaske. Abhängig davon, zu welcher Adressklasse eine IP-Adresse gehört, hat die Subnetzmaske einen Minimalwert abhängig vom zulässigen Bereich des "Netzwerk-Teils" der IP-Adresse.

Adressklasse	**Minimale Subnetzmaske**
Class A	255.0.0.0
Class B	255.255.0.0
Class C	255.255.255.0

Das Optionsmodul Ethernet hält sich an diese von Adressklassen abhängigen Subnetzmasken, d. h. ist die vom Anwender eingegebene Subnetzmaske kleiner als die zur IP-Adresse gehörende Adressklassen-Maske wird die Adressklassen-Maske verwendet.

Beispiel:

Ein PC mit PROPROG wt II hat die IP-Adresse 192.075.191.188, ein Optionsmodul Ethernet hat die IP-Adresse 192.168.1.1. Beide Netzwerkteilnehmer befinden sich physikalisch im gleichen Netz.

Da beide IP-Adressen nur im ersten Byte identisch sind, müsste als Subnetzmaske auf beiden Systemen 255.0.0.0 gewählt werden, um durch die Verknüpfung "IP-Adresse AND Subnetzmaske" gleiche Ergebnisse zu erhalten. Da die IP-Adressen aber zu einem Class C Netz gehören, verwendet das Optionsmodul die Subnetzmaske 255.255.255.0. Mit der Verknüpfung "IP-Adresse AND Subnetzmaske" ergeben sich aber unterschiedliche Netzwerke: 192.075.191.0 und 192.168.1.0. Eine Kommunikation ist damit nicht möglich, da die Datenpakete an den eingestellten Gateway übergeben werden.

Durch Deaktivieren des Gateways mit Vergabe der Gateway-IP-Adresse 0.0.0.0 kann dieses Problem jedoch umgangen werden und eine Kommunikation aufgebaut werden.

f) Beispiele für IP-Netzwerke

Beispiel 1:

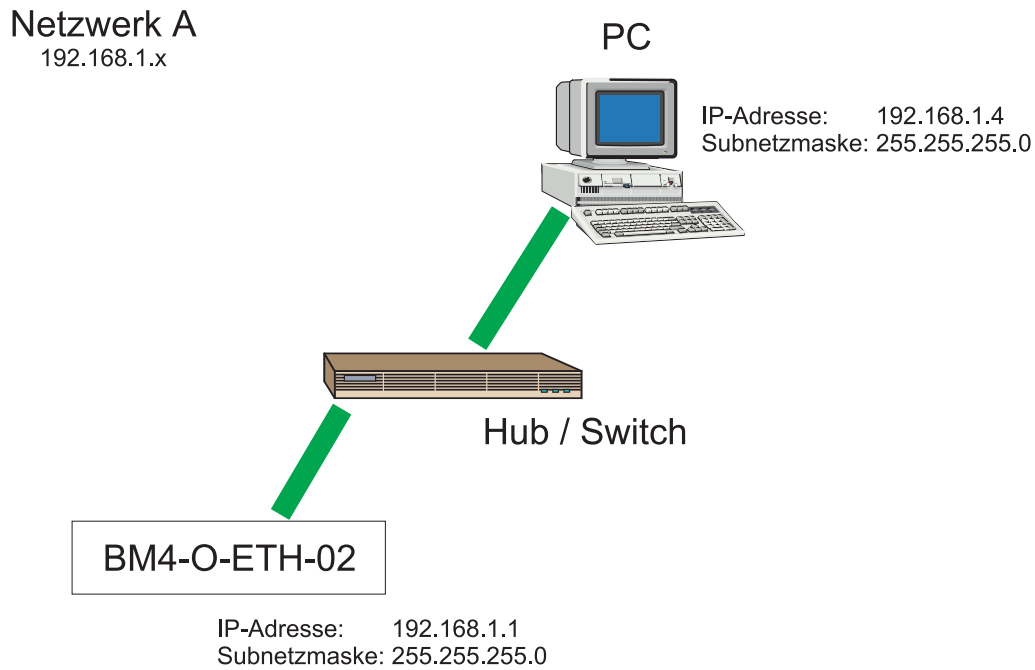


Abbildung 5: Beispiel: Optionsmodul und PC im gleichen IP-Netz

Im Netzwerk A haben die dargestellten Komponenten die angegebenen Einstellungen. Kommunikation zwischen PC und Optionsmodul ist möglich da sich bei beiden Komponenten mit "IP-Adresse AND Subnetzmaske" das gleiche Netzwerk ergibt: 192.168.1.0

Beispiel 2:

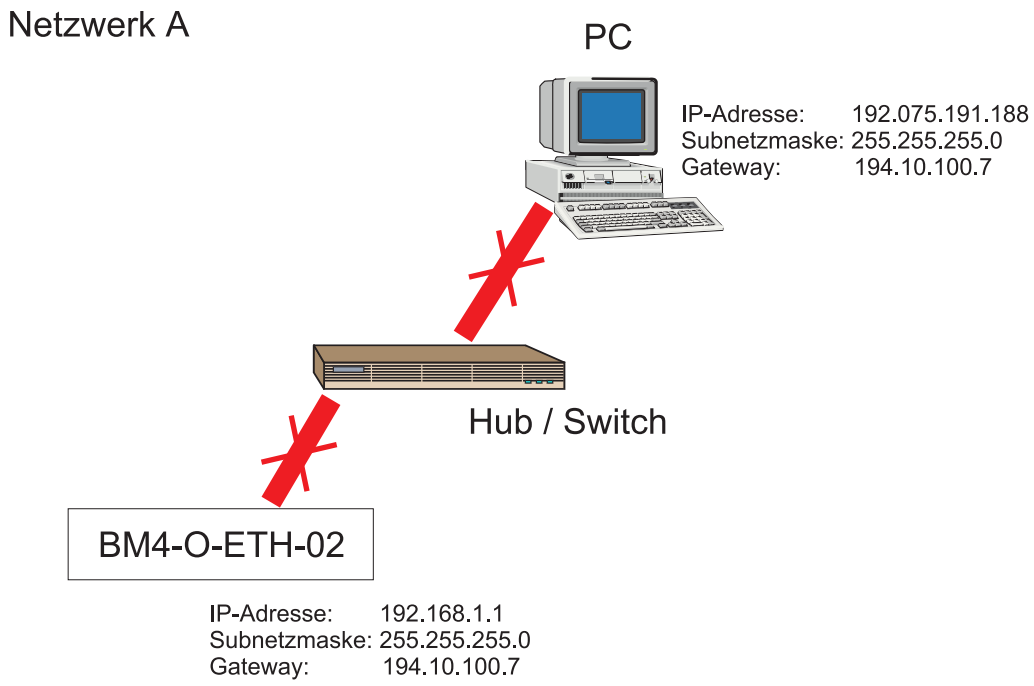


Abbildung 6: Beispiel: Optionsmodul und PC in unterschiedlichen IP-Netzen

Im Netzwerk A haben die dargestellten Komponenten die angegebenen Einstellungen. Kommunikation zwischen PC und Optionsmodul ist nicht möglich da sich bei beiden Komponenten mit "IP-Adresse AND Subnetzmaske" unterschiedliche Netzwerke ergeben. PC und Optionsmodul würden die Datenpakete an einen Gateway übergeben.

Beispiel 3:

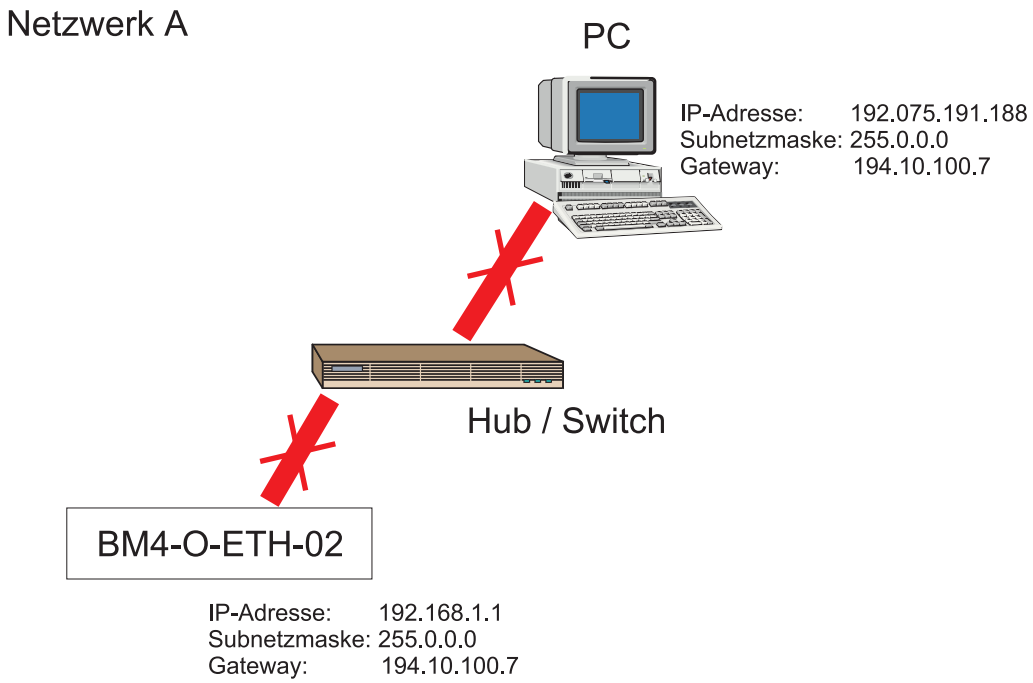


Abbildung 7: Beispiel: Optionsmodul und PC mit falscher Subnetzmaske

Im Netzwerk A haben die dargestellten Komponenten die angegebenen Einstellungen. Kommunikation zwischen PC und Optionsmodul ist nicht möglich. Mit der Verknüpfung "IP-Adresse AND Subnetzmaske" ergibt sich zwar das gleiche Netzwerk 192.x.x.x. Das Optionsmodul erkennt aber ein Class C Netzwerk und verwendet die Subnetzmaske 255.255.255.0. und würde die Datenpakete an einen Gateway übergeben. Abhilfe siehe Beispiel 4.

Beispiel 4:

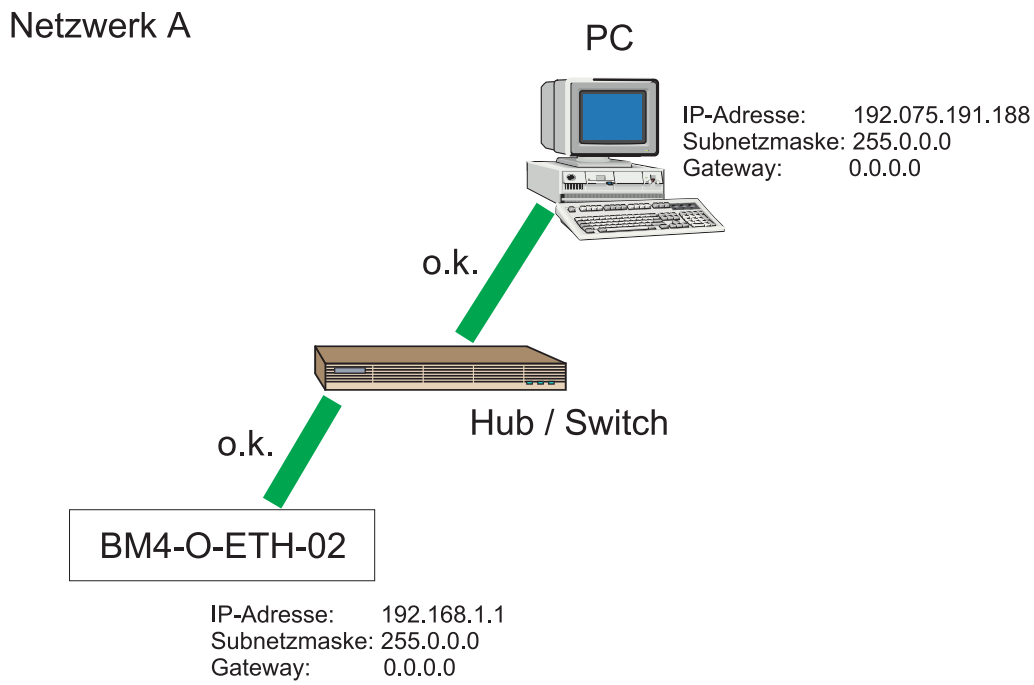


Abbildung 8: Beispiel: Optionsmodul und PC ohne Gateway

Im Netzwerk A haben die dargestellten Komponenten die angegebenen Einstellungen. Kommunikation zwischen PC und Optionsmodul ist möglich. Mit der Verknüpfung "IP-Adresse AND Subnetzmaske" ergibt sich das Netzwerk 192.x.x.x. Das Optionsmodul erkennt zwar ein Class C Netzwerk, verwendet die Subnetzmaske 255.255.255.0. und würde die Datenpakete an einen Gateway übergeben. Da aber kein Gateway eingestellt ist findet die Übertragung statt.

Beispiel 5:

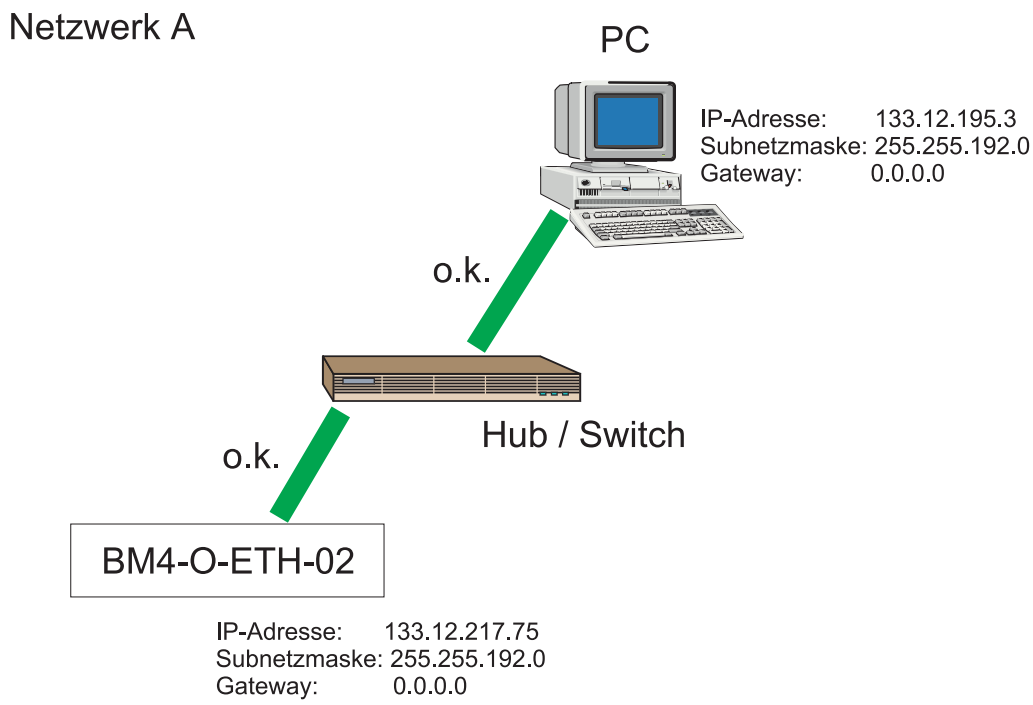


Abbildung 9: Beispiel: Optionsmodul und PC in einem Class-B IP-Netz

Im Netzwerk A haben die dargestellten Komponenten die angegebenen Einstellungen. Kommunikation zwischen PC und Optionsmodul ist möglich, da sich bei beiden Komponenten mit "IP-Adresse AND Subnetzmaske" das gleiche Netzwerk ergibt: 133.12.192.0

Beispiel 6:

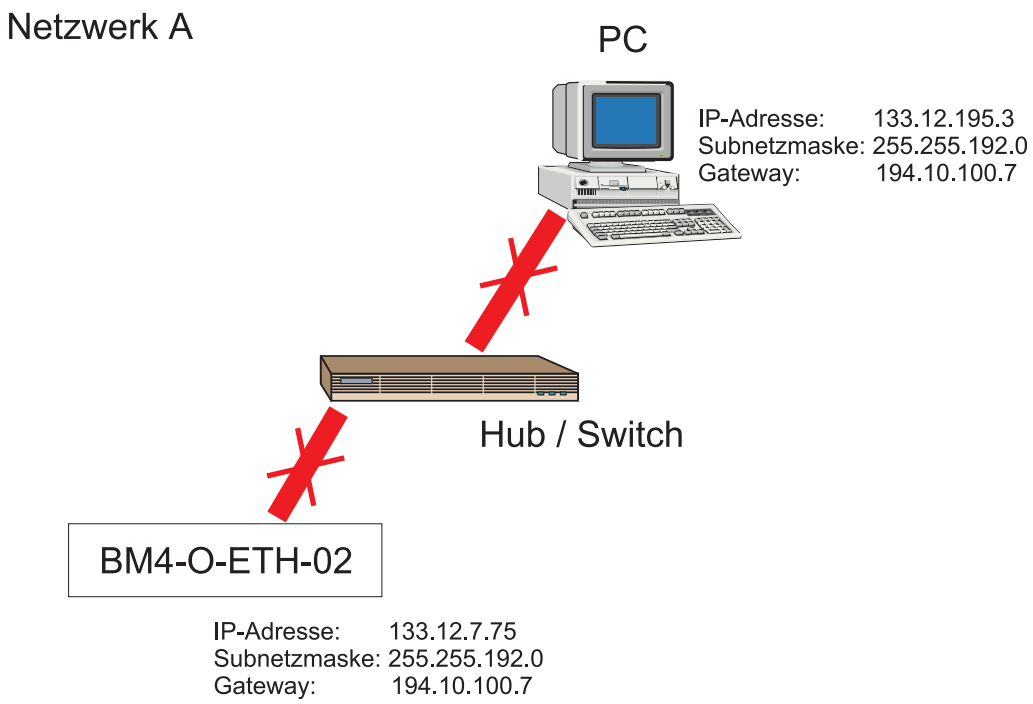


Abbildung 10: Beispiel: Optionsmodul und PC in unterschiedlichen Class-B IP-Netzen

Im Netzwerk A haben die dargestellten Komponenten die angegebenen Einstellungen. Kommunikation zwischen PC und Optionsmodul ist nicht möglich, da mit "IP-Adresse AND Subnetzmaske" die Netzwerke unterschiedlich sind: 133.12.192.0 und 133.12.0.0. Die Datenpakete würden an einen Gateway übergeben.

Beispiel 7:

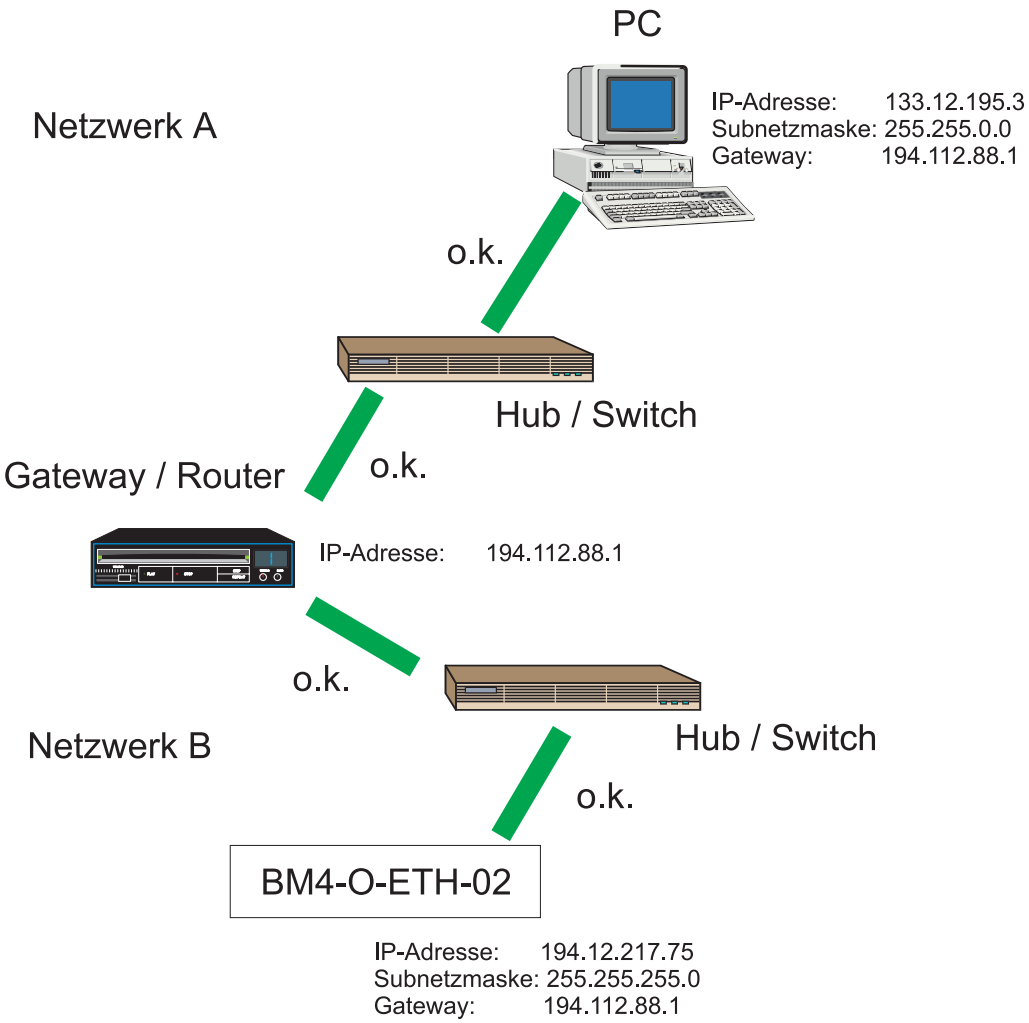


Abbildung 11: Beispiel: Optionsmodul und PC mit Gateway

Im Netzwerk A und Netzwerk B haben die dargestellten Komponenten die angegebenen Einstellungen. Die Netzwerke sind über einen Gateway mit der IP-Adresse 194.112.88.1 verbunden. Kommunikation zwischen PC und Optionsmodul ist möglich, da sich zwar bei beiden Komponenten mit "IP-Adresse AND Subnetzmaske" unterschiedliche Netzwerke ergeben - 133.12.0.0 und 194.12.217.0 - beide aber die Datenpakete an den gleichen Gateway weitergeben, welcher die weitere Verteilung übernimmt.

3.2.6.2 TCP-Protokoll

TCP (Transport Control Protokoll) ist für den Transport und die Sicherung der Daten zuständig und liegt im IP-Nutzdatenbereich. TCP arbeitet nach dem Client-Server-Prinzip. Es stellt für die Dauer der Datenübertragung eine Verbindung zwischen zwei Netzteilnehmern, eine Session, her. Innerhalb einer Session werden Nutzdaten auf einzelne, nummerierte TCP-Pakete aufgeteilt und wieder zusammen-gesetzt, so dass sie gegebenenfalls auf verschiedenen Wegen mit unterschiedlicher Laufzeit übertragen werden kön-

nen. Checksummen ermöglichen die Kontrolle und die Bestätigung eines korrekten Empfanges. Der Verlust einzelner TCP-Pakete wird bemerkt, sie können nachgefordert werden.

TCP-Pakete werden in den Nutzdatenbereich eines IP-Paketes eingesetzt. Darüber hinaus leitet TCP die Nutzdaten an das richtige Anwendungsprogramm auf dem Zielrechner weiter, indem es unterschiedlichen Anwendungen (Diensten) unterschiedliche Portnummern zuordnet.

Die Portnummer für die PROPROG-Kommunikation ist 0x5043 (= 20547 dez.). Der Herkunfts-Port ist die Port-Nummer der sendenden Anwendung und gleichzeitig der Rücksende-Port der Antwort.

Als Anwender brauchen Sie keinerlei Einstellungen auf TCP-Ebene vorzunehmen.

3.2.7 Zusammenspiel Ethernet und TCP/IP, ARP

Bisher ist die Adressierung der Teilnehmer nur durch die IP-Adressen sichergestellt. Auf der Ebene des Ethernet ist dem Netzwerkknoten bisher nur seine eigene MAC-ID bekannt. Um das IP-Paket in einem Ethernet-Rahmen zustellen zu können muss jedoch auch die MAC-ID des Empfängers bekannt sein. Um diese zu ermitteln wurde das ARP-Protokoll (Address Resolution Protokoll) entwickelt. Um das IP-Ethernet-Adresspaar eines Netzwerkknotens herauszufinden, sendet das ARP-Protokoll einen Ethernet-Broadcast (Rundruf) mit der bekannten IP-Adresse des Empfängers aus. Der Empfänger, der diese IP-Adresse hat, antwortet dem fragenden System mit einem Paket, welches das IP-Ethernet-Adresspaar enthält. Um unnötige ARP-Anfragen zu vermeiden, wird das Antwortpaket auf dem fragenden System in einer ARP-Tabelle gespeichert. Um diese ARP-Tabelle klein zu halten, werden die Einträge in bestimmten, systemabhängigen Zeitintervallen wieder gelöscht. Über die Kommandozeile können Sie auf Ihrem PC die aktuelle ARP-Tabelle ihres PCs auslesen:

```
arp -a
```

Hat bereits eine Kommunikation mit dem Optionsmodul Ethernet stattgefunden finden Sie das Optionsmodul in der Auflistung. Um eine Kommunikation anzustoßen, können Sie über die Kommandozeile einen "Ping" auslösen.

```
ping 192.168.1.1
```

Sofern Ihr Netzwerk richtig konfiguriert ist (IP-Adressen, Subnetzmasken und Gateways) erhalten Sie vom Optionsmodul mit der IP-Adresse 192.168.1.1 eine Antwort der Form:

```
Antwort von 192.168.1.1: Bytes.....
```

Sollten Sie keine Antwort erhalten, dann überprüfen Sie bitte Ihr Netzwerk.

Haben Sie eine Antwort erhalten ist das Optionsmodul jetzt auch in der ARP-Tabelle Ihres PCs aufgelistet. Die ARP-Tabelle des Optionsmodul Ethernet kann nicht ausgelesen werden.

3.2.8 Proxy

Der Begriff Proxy stammt aus dem Englischen und bedeutet soviel wie Stellvertreter. Ein Proxy ist ein Rechner, der Anfragen eines Netzwerkteilnehmers entgegen nimmt und diese an das beabsichtigte Ziel weiter leitet. Das Ergebnis der Anfrage wird vom Ziel an den Proxy übermittelt, welcher seinerseits das Ergebnis dem anderen Netzwerkteilnehmer zurückliefern kann. Ein Proxy kann auf verschiedenen Applikationsebenen arbeiten und die zu übermittelnden Daten kontrollieren. Er besitzt oft einen großen Speicher (Proxy-Cache), um angefragte Daten zwischenzuspeichern und, um bei einer eventuell nachfolgenden erneuten Anfrage eines anderen oder desselben Netzwerkteilnehmers direkt, d. h. ohne eine zusätzliche Anfrage an das eigentliche Ziel die Daten bereitzustellen. Ein Firewall-Proxy kommt meist zum Einsatz, wenn ein internes Netz, etwa ein Intranet / LAN eine geschützte Verbindung mit einem anderen Netz (z. B. Internet) aufnehmen soll. Der Proxy agiert dabei als eine Art Gateway mit IP- und Paketfilter. Er kann die Datenübertragung von einem Netz in ein anderes aufgrund von Ursprung, Ziel, Port und Pakettyp-Information, die jedes Datenpaket besitzt, kontrollieren und gegebenenfalls beschränken. Durch entsprechende Software-Funktionen kann der Proxy weitere Schutzaufgaben oberhalb der IP-Ebene übernehmen.

3.2.9 Applikationsschicht PROPROG wt II / OmegaOS

PROPROG wt II / OmegaOS-Datenpakete werden im Nutzdatenbereich der TCP-Pakete übertragen. Für den Anwender sind hier keine weiteren Einstellungen zu machen. Lediglich in PROPROG wt II ist der Kommunikationsport einzustellen (siehe Kapitel ▹Verbindung über Ethernet-TCP/IP in PROPROG wt II / ProProg wt III einstellen◃ ab Seite 46). Das Optionsmodul Ethernet stellt Ihnen 2 Kommunikationskanäle zur Verfügung, d. h. Sie können mit dem Optionsmodul parallel maximal

zweimal ProMaster

oder

ProMaster und ProProg wt III

oder

zweimal ProProg wt III

oder

ProMaster (oder ProProg wt III) und eine OPC-Anwendung

oder

zwei OPC-Anwendungen

O D E R

zweimal PROPROG wt II

oder

PROPROG wt II und eine OPC-Anwendung

oder

Zwei OPC Anwendungen

betreiben.

3.3 Ethernet-TCP/IP Netzwerke konfigurieren

3.3.1 Übersicht

Um mit PROPROG wt II oder über OPC das Optionsmodul Ethernet über Ethernet mit TCP/IP verwenden zu können sind folgende Schritte durchzuführen:

- Physikalische Inbetriebnahme des Netzwerks (siehe Betriebsanleitung Optionsmodul BM4-O-ETH-01 / BM4-O-ETH-02 und Betriebsanleitungen der Netzwerkkomponenten)
- Festlegung der Netzwerkdaten: IP-Adressen und Subnetzmasken (siehe vorhergehendes Kapitel ▹TCP/IP◃ ab Seite 21)
- Einstellen von IP-Adressen, Subnetzmasken und Gateway in den Netzwerkkomponenten (siehe nachfolgende Kapitel)
- Einstellen der Kommunikationsports in PROPROG wt II (siehe nachfolgendes Kapitel ▹Verbindung über Ethernet-TCP/IP in PROPROG wt II / ProProg wt III einstellen◃ ab Seite 46)

3.3.2 Windows-PC konfigurieren

Da PROPROG wt II und der OPC-Server Windows-Betriebssysteme voraussetzen, werden die notwendigen Einstellungen nur für Windows-Betriebssysteme beschrieben. Voraussetzung ist grundsätzlich, dass das TCP/IP Protokoll auf Ihrem Windows-PC installiert ist und ihr PC eine konfigurierte Netzwerkkarte besitzt. Details zu dieser Installation entnehmen Sie bitte der Dokumentation zu Ihrem Windows-Betriebssystem.

3.3.2.1 TCP/IP unter Windows XP konfigurieren

- Klicken Sie auf *Start* und wählen Sie die *Netzwerkumgebung* aus. Lassen Sie sich im Feld *Netzwerkaufgaben* die *Netzwerkverbindungen anzeigen*.
- Doppelklicken Sie auf das Icon *LAN-Verbindung* und Betätigen dann den Button *Eigenschaften*
  → Sie erhalten eine Auswahl von Komponenten, welche Ihre Netzwerkkarte verwenden.
- Aktivieren Sie *Internetprotokoll (TCP/IP)* und Betätigen dann den Button *Eigenschaften*

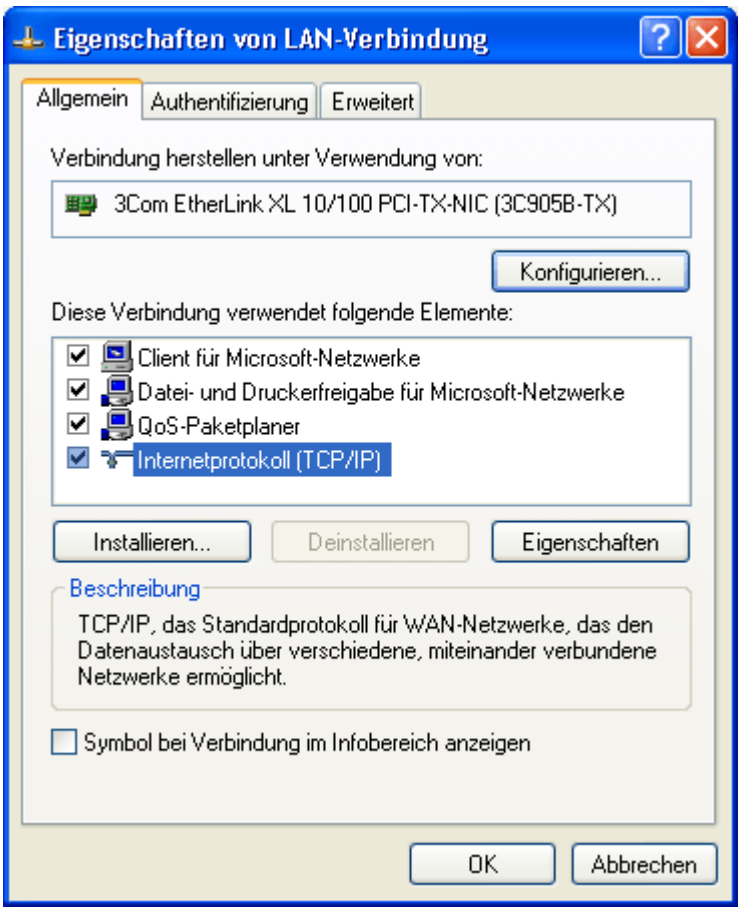


Abbildung 12: Windows XP mit TCP/IP einrichten - LAN Verbindungen Übersicht

- Sie haben nun die Wahl *IP-Adresse automatisch beziehen* oder *Folgende IP-Adresse verwenden*. Sollten Sie die IP-Adresse automatisch beziehen, können Sie die aktuelle Einstellung im Feld *Details* der *LAN-Verbindung* des Fensters *Netzwerkverbindungen*

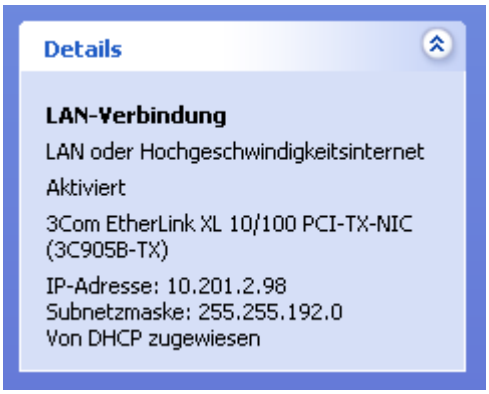


Abbildung 13: Windows XP mit TCP/IP einrichten - LAN Verbindungen Details

oder über die Kommandozeile mit

```
ipconfig
```

abfragen. Sollten die Einstellungen mit den von Ihnen zuvor festgelegten übereinstimmen, können Sie den Dialog und die Konfiguration beenden. Beachten Sie jedoch, dass diese Einstellungen dynamisch vergeben werden, d. h. beim nächsten Start ihres PCs können sich andere Einstellungen ergeben.

Aktivieren Sie Folgende IP-Adresse verwenden wenn Sie spezielle Werte für IP-Adresse, Subnetzmaske und Gateway verwenden wollen und tragen Sie diese dort ein:

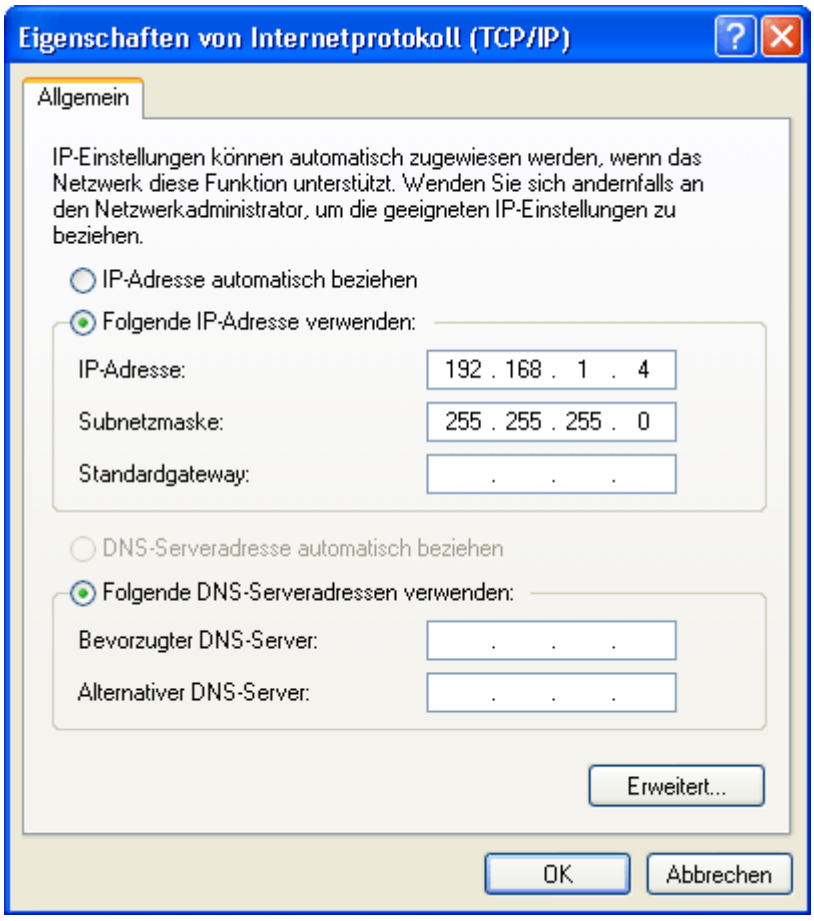


Abbildung 14: Windows XP mit TCP/IP einrichten - TCP/IP Eigenschaften

- Schließen Sie die zur Konfiguration geöffneten Fenster mit dem Button *OK*

Damit ist die Installation der TCP/IP-Unterstützung auf Ihrem Windows XP System abgeschlossen. Unter Windows XP muss Ihr PC nicht neu gebootet werden.

3.3.2.2 TCP/IP unter Windows 2000 konfigurieren

- Klicken Sie auf *Start* und öffnen Sie unter *Einstellungen* die *Netzwerk- und DFÜ-Verbindungen*

- Doppelklicken Sie auf das Icon *LAN-Verbindung* und Betätigen dann den Button *Eigenschaften*
  → Sie erhalten eine Auswahl von Komponenten, welche Ihre Netzwerkkarte verwenden
- Aktivieren Sie *Internetprotokoll (TCP/IP)* und Betätigen dann den Button *Eigenschaften*

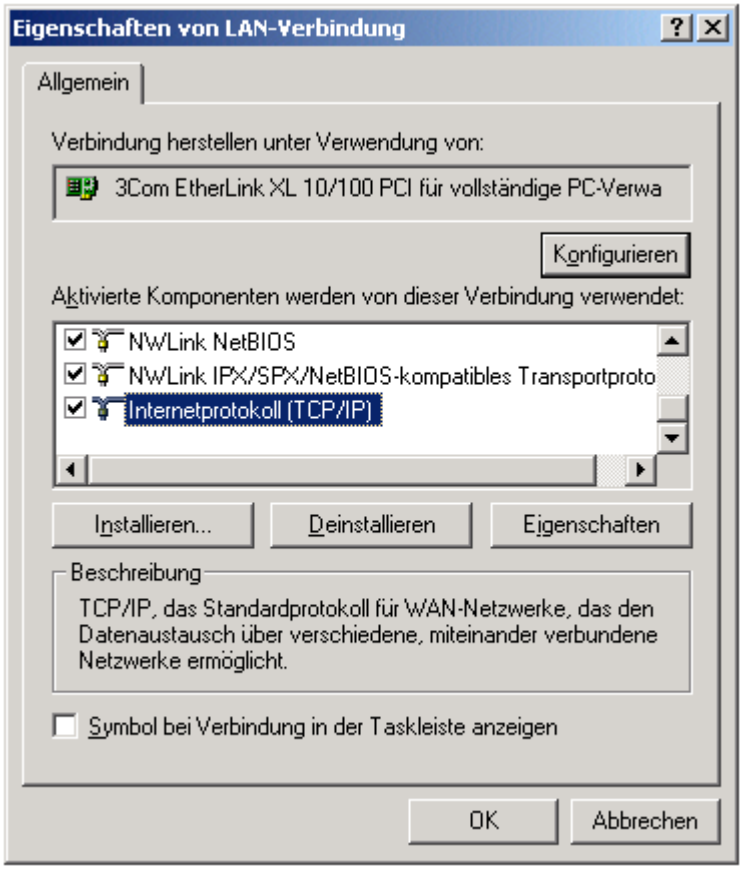


Abbildung 15: Windows 2000 mit TCP/IP einrichten - LAN Verbindungen Übersicht

- Sie haben nun die Wahl *IP-Adresse automatisch* beziehen oder *Folgende IP-Adresse* verwenden. Sollten Sie die *IP-Adresse automatisch* beziehen, können Sie die aktuelle Einstellung über die Kommandozeile mit

```
ipconfig
```

abfragen. Sollten die Einstellungen mit den von Ihnen zuvor festgelegten übereinstimmen, können Sie den Dialog und die Konfiguration beenden. Beachten Sie jedoch, dass diese Einstellungen dynamisch vergeben werden, d. h. beim nächsten Start ihres PCs können sich andere Einstellungen ergeben.

Aktivieren Sie *Folgende IP-Adresse* verwenden wenn Sie spezielle Werte für IP-Adresse, Subnetzmaske und Gateway verwenden wollen und tragen Sie diese dort ein:

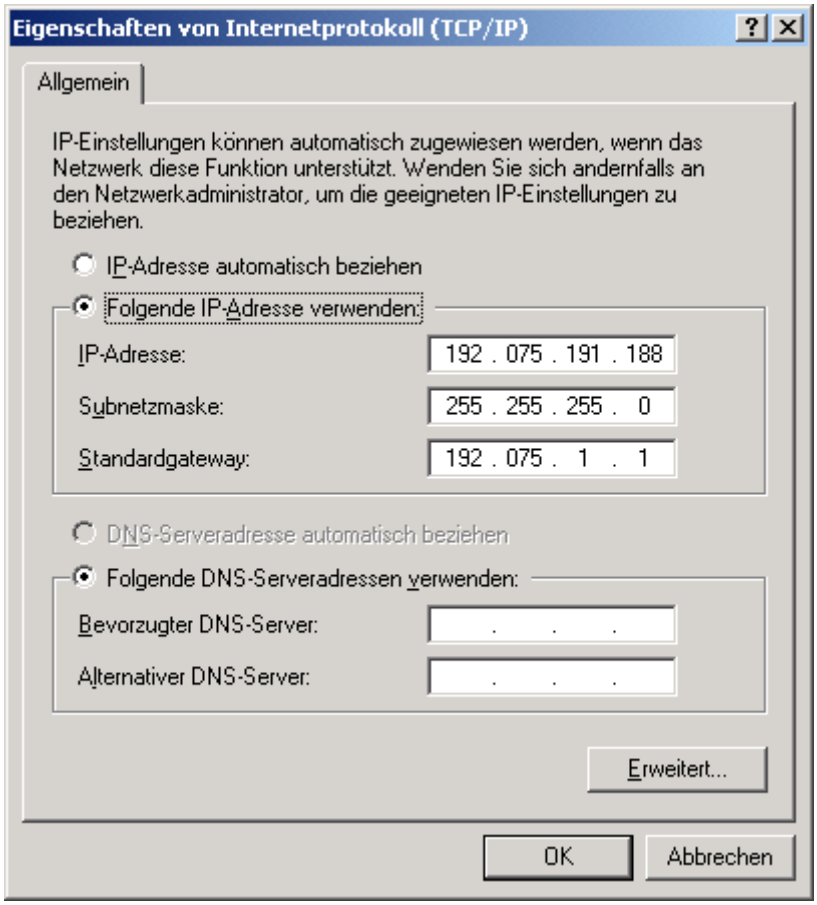


Abbildung 16: Windows 2000 mit TCP/IP einrichten - TCP/IP Eigenschaften

- Schließen Sie die zur Konfiguration geöffneten Fenster mit dem Button *OK*

Damit ist die Installation der TCP/IP-Unterstützung auf Ihrem Windows 2000 System abgeschlossen. Unter Windows 2000 muss Ihr PC nicht neu gebootet werden.

3.3.3 Optionsmodul Ethernet konfigurieren und Einstellungen überprüfen

Um das Optionsmodul Ethernet für TCP/IP-Kommunikation zu konfigurieren haben Sie zwei Möglichkeiten:

- Verwenden der Defaulteinstellungen
- Freie Konfiguration über PROPROG wt II / ProProg wt III

3.3.3.1 Defaulteinstellungen für TCP/IP

Die Defaulteinstellungen der TCP/IP-Kommunikation des Optionsmodul Ethernet sind:

IP-Adresse:	192.168.1.1+"Dip-Schalter"
Subnetzmaske:	255.255.255.0
Gateway:	0.0.0.0

Mit Hilfe der Dip-Schalter 1 - 5 (S5000 auf dem Optionsmodul) können IP-Adressen im Bereich 192.168.1.1 bis 192.168.1.32 eingestellt werden. Zum Einstellen der Dip-Schalter auf dem Optionsmodul siehe ▹Betriebsanleitung Ethernet mit CANopen-Master für b maXX drive PLC◃.

3.3.3.2 Freie Konfiguration von TCP/IP

a) Vorbereiten der Konfiguration mit PROPROG wt II / ProProg wt III

Die Werte für IP-Adresse, Subnetzmaske und Gateway können auch frei eingestellt werden. Gehen Sie dazu in folgenden Schritten vor:

1. Legen Sie ein PROPROG wt II / ProProg wt III - Projekt für die b maXX drive PLC an, falls Sie noch kein eigenes Projekt für ihre Applikation angelegt haben.
2. PROPROG wt II: Binden Sie in dieses Projekt die Bibliothek BM_TYPES_20bd03 (oder höher) ein, falls diese noch nicht vorhanden ist.
   ProProg wt III: Binden Sie in dieses Projekt die Bibliothek BM_TYPES_30bd01 (oder höher) ein, falls diese noch nicht vorhanden ist.
3. Erstellen Sie eine POE, welche später in einer Kaltstart- und Warmstarttask aufgerufen werden kann, sofern noch keine solche vorhanden ist.
4. Legen Sie eine globale Variable vom Datentyp ETHERNET_PLC_CONFIG_BMSTRUCT an. Diese globale Variable ist auf die Basisadresse zur Ethernet-Konfiguration zu legen. Die Basisadresse ist vom Steckplatz abhängig, in welchem das Optionsmodul Ethernet steckt. Folgende Adressen sind möglich:

Steckplatz (Slot)	Basisadresse für Ethernet-Konfiguration
G	%MB3.2012288
H	%MB3.3012288
J	%MB3.4012288
K	%MB3.5012288
L	%MB3.6012288
M	%MB3.7012288

Beispiel, falls das Optionsmodul Ethernet im Steckplatz G steckt:

PROPROG wt II:

```
VAR_GLOBAL
    _EthernetConfigSlotG AT %MB3.2012288    : ETHERNET_PLC_CONFIG_BMSTRUCT;
END_VAR
```

Abbildung 17: Beispiel: Globale Variable vom Typ ETHERNET_PLC_CONFIG_BMSTRUCT (PROPROG wt II)

ProProg wt III:

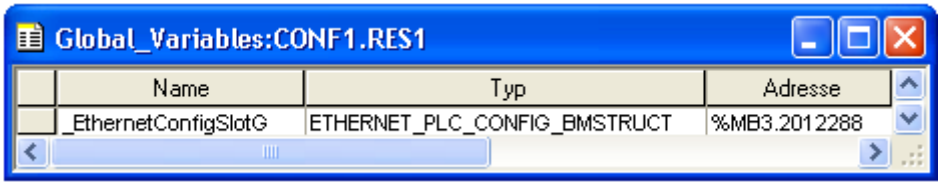


Abbildung 18: Beispiel: Globale Variable vom Typ ETHERNET_PLC_CONFIG_BMSTRUCT (ProProg wt III)

Dabei ist:

`_EthernetConfigSlotG`	der Variablenname mit der Datentyp-Kurzbezeichnung "_" für STRUCT
`%MB3.2012288`	die Adresse zur Ethernet-Konfiguration von Steckplatz G
`ETHERNET_PLC_CONFIG_BMSTRUCT`	der Datentyp der Variable

5 Um IP-Adresse, Subnetzmaske und Gateway einzustellen, benötigen Sie die Strukturelemente a_IP_ADDRESS, a_IP_MASK und a_GATEWAY der angelegten globalen Variable. Alle drei Elemente bestehen aus einem Feld (Array) mit jeweils 4 Einträgen vom Datentyp USINT. Die jeweilige Adresse beginnt in Feldindex 0 mit dem Netzwerkteil. Um die Konfiguration zu aktivieren, wird zudem das Strukturelement d_IP_CONFIG vom Datentyp DWORD benötigt. Legen Sie diese Elemente in der POE für den Kalt-/Warmstart an.

Beispiel:

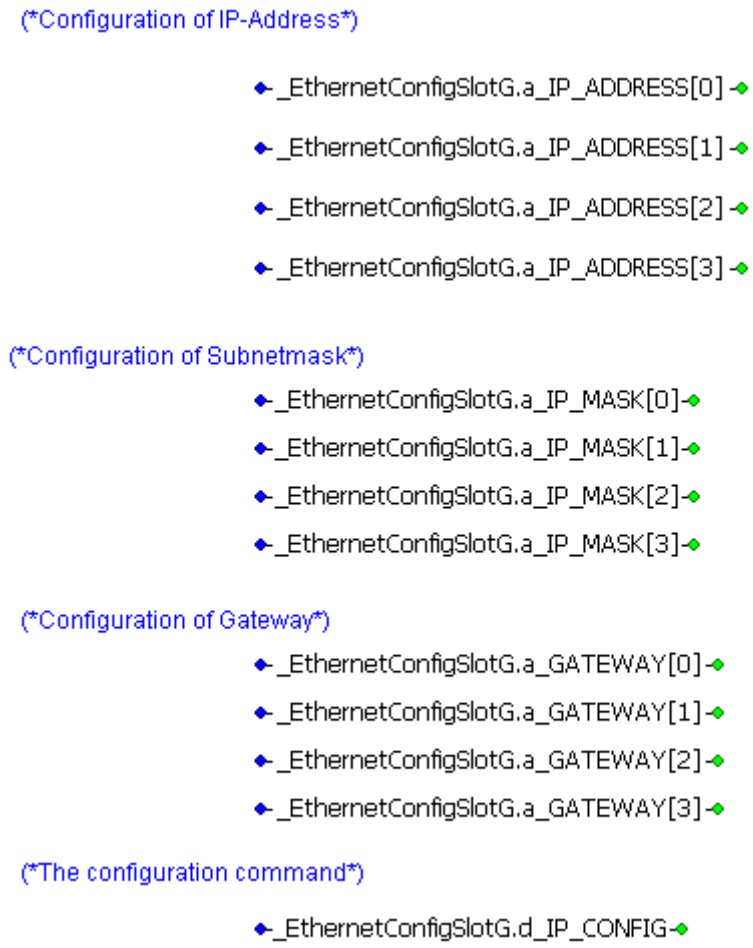


Abbildung 19: Beispiel: Elemente einer globalen Variable vom Typ ETHERNET_PLC_CONFIG_BMSTRUCT

6 Es gibt zwei Möglichkeiten die IP-Adresse zu konfigurieren:
Feste IP-Adresse oder Dip-Schalter abhängige IP-Adresse. Im ersten Fall wird eine IP-Adresse eingestellt, welche von den Dip-Schaltern unabhängig ist. Im zweiten Fall wird der Wert der Dip-Schalter 1-5 zur konfigurierten IP-Adresse addiert. Welches Verfahren verwendet wird unterscheidet sich nur durch den Wert des Elements d_IP_CONFIG.

b) Feste IP-Adresse vergeben

Um eine feste, von den Dip-Schaltern unabhängige IP-Adresse zu vergeben werden die Strukturelemente der globalen Variable zur Ethernet-Konfiguration mit den erforderlichen Werten belegt. Abschließend ist das Element d_IP_CONFIG mit DWORD#16#12345678 zu beschreiben. Bei dem Beschreiben der Elemente ist unbedingt die Reihenfolge

a_IP_ADDRESS → a_IP_MASK → a_GATEWAY → d_IP_CONFIG

einzuhalten.

Beispiel:

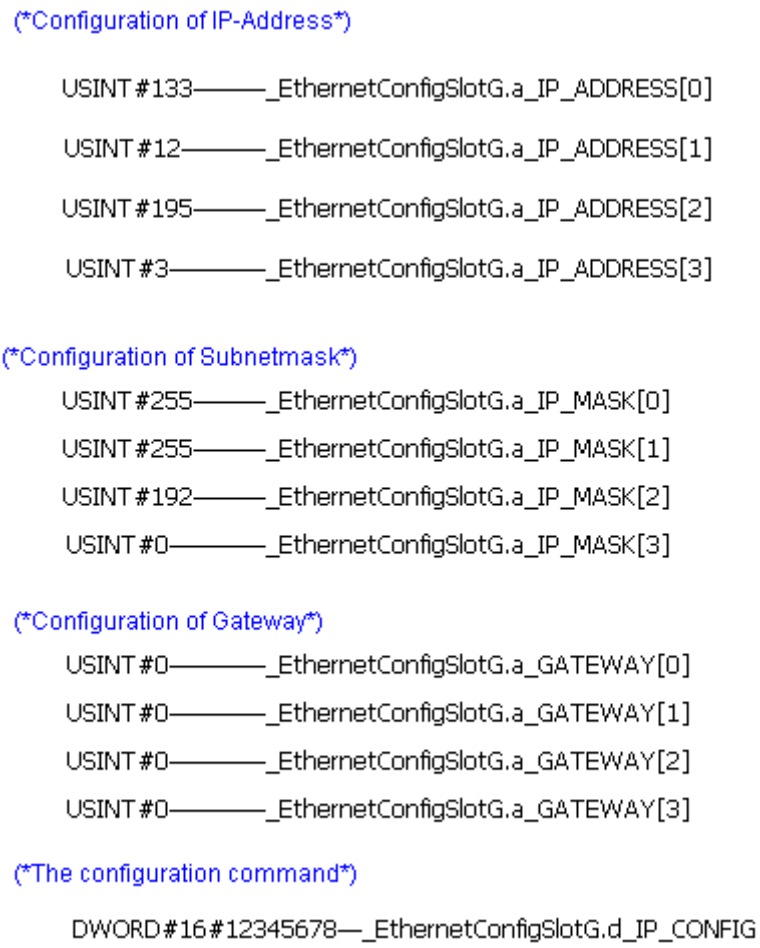


Abbildung 20: Beispiel: Feste IP-Adresse, Gateway und Subnetzmaske für Optionsmodul vergeben

Für das Optionsmodul Ethernet ergibt sich damit die IP-Adresse 133.12.195.3 und die Subnetzmaske 255.255.192.0. Der Gateway hat die Adresse 0.0.0.0, d. h. es wird kein Gateway verwendet.

c) Variable, von Dip-Schaltern abhängige IP-Adresse vergeben

Um eine variable, von den Dip-Schaltern abhängige IP-Adresse zu vergeben werden die Strukturelemente der globalen Variable zur Ethernet-Konfiguration mit den erforderlichen Werten belegt. Abschließend ist das Element d_IP_CONFIG mit DWORD#16#12345600 zu beschreiben. Bei dem Beschreiben der Elemente ist unbedingt die Reihenfolge

a_IP_ADDRESS → a_IP_MASK → a_GATEWAY → d_IP_CONFIG

einzuhalten.

Beispiel:

```
(*Configuration of IP-Address*)

    USINT#133------_EthernetConfigSlotG.a_IP_ADDRESS[0]

    USINT#12-------_EthernetConfigSlotG.a_IP_ADDRESS[1]

    USINT#195------_EthernetConfigSlotG.a_IP_ADDRESS[2]

    USINT#3--------_EthernetConfigSlotG.a_IP_ADDRESS[3]


(*Configuration of Subnetmask*)
    USINT#255------_EthernetConfigSlotG.a_IP_MASK[0]
    USINT#255------_EthernetConfigSlotG.a_IP_MASK[1]
    USINT#192------_EthernetConfigSlotG.a_IP_MASK[2]
    USINT#0--------_EthernetConfigSlotG.a_IP_MASK[3]

(*Configuration of Gateway*)
    USINT#0--------_EthernetConfigSlotG.a_GATEWAY[0]
    USINT#0--------_EthernetConfigSlotG.a_GATEWAY[1]
    USINT#0--------_EthernetConfigSlotG.a_GATEWAY[2]
    USINT#0--------_EthernetConfigSlotG.a_GATEWAY[3]

(*The configuration command*)

    DWORD#16#12345600--_EthernetConfigSlotG.d_IP_CONFIG
```

Abbildung 21: Beispiel: Variable IP-Adresse, Gateway und Subnetzmaske für Optionsmodul vergeben

Für das Optionsmodul Ethernet ergibt sich damit die IP-Adresse 133.12.195.3 + "Dip-Schalter" und die Subnetzmaske 255.255.192.0. Der Gateway hat die Adresse 0.0.0.0, d. h. es wird kein Gateway verwendet.

d) Aktivieren der TCP/IP-Konfiguration

Die POE mit den Elementen zur TCP/IP-Konfiguration ist angelegt. Führen Sie abschließend noch folgende Schritte durch:

1. Binden Sie jetzt die POE in eine Kaltstart- und eine Warmstarttask ein. Übersetzen sie anschließend das Programm und laden Sie das Programm als Boot-Projekt auf die b maXX drive PLC.
2. Schalten Sie das b maXX Gerät im Abstand von ca. 10 s zweimal aus und wieder ein. Es ist unbedingt erforderlich zweimal aus und wieder einzuschalten.

Das Optionsmodul Ethernet hat nun die von Ihnen eingestellte TCP/IP-Konfiguration. Sie können den Programmcode aus der angelegten POE wieder entfernen. Die TCP/IP-Konfiguration bleibt dennoch erhalten. Ebenso können Sie ein anderes Projekt (auch Boot-Projekt) auf die b maXX drive PLC laden, ohne die TCP/IP-Konfiguration zu verlieren.

Erst durch erneute Implementierung der oben beschriebenen Codezeilen kann die TCP/IP-Konfiguration des Optionsmodul Ethernet wieder geändert werden.

e) Überprüfen der TCP/IP-Konfiguration

Die Einstellungen der TCP/IP-Konfiguration des Optionsmodul Ethernet können Sie mit PROPROG wt II / ProProg wt III überprüfen.

Gehen Sie dazu in folgenden Schritten vor:

1. Legen Sie ein PROPROG wt II / ProProg wt III - Projekt für die b maXX drive PLC an, falls Sie noch kein eigenes Projekt für ihre Applikation oder durch die Konfiguration angelegt haben.
2. PROPROG wt II: Binden Sie in dieses Projekt die Bibliothek BM_TYPES_20bd03 (oder höher) ein, falls diese noch nicht vorhanden ist.
   ProProg wt III: Binden Sie in dieses Projekt die Bibliothek BM_TYPES_30bd01 (oder höher) ein, falls diese noch nicht vorhanden ist.
3. Legen Sie eine globale Variable vom Datentyp ETHERNET_PLC_DIAG_BMSTRUCT an. Diese globale Variable ist auf die Basisadresse zur Ethernet-Diagnose zu legen. Die Basisadresse ist dabei vom Steckplatz abhängig, in welchem das Optionsmodul Ethernet steckt. Folgende Adressen sind möglich:

Steckplatz (Slot)	Basisadresse für Ethernet-Diagnose
G	%MB3.2012320
H	%MB3.3012320
J	%MB3.4012320
K	%MB3.5012320
L	%MB3.6012320
M	%MB3.7012320

Beispiel, falls das Optionsmodul Ethernet im Steckplatz G steckt:

PROPROG wt II:

```
VAR_GLOBAL
    _EthernetDiagSlotG    AT %MB3.3012320    : ETHERNET_PLC_DIAG_BMSTRUCT;
END_VAR
```

Abbildung 22: Beispiel: Globale Variable zum Überprüfen der TCP/IP-Konfiguration (PROPROG wt II)

ProProg wt III:

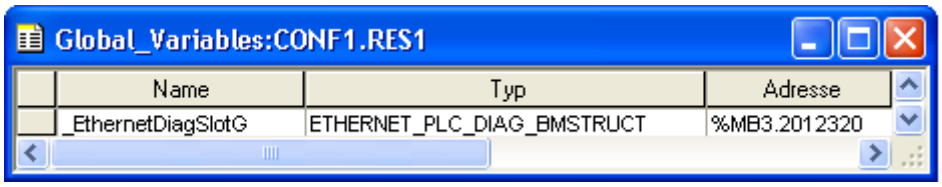


Abbildung 23: Beispiel: Globale Variable zum Überprüfen der TCP/IP-Konfiguration (ProProg wt III)

Dabei ist:

`_EthernetDiagSlotG`	der Variablenname mit der Datentyp-Kurzbezeichnung "_" für STRUCT
`%MB3.2012320`	die Adresse zur Ethernet-Diagnose von Steckplatz G
`ETHERNET_PLC_DIAG_BMSTRUCT`	der Datentyp der Variable

4 Für Diagnosezwecke stehen die Strukturelemente a_MAC_ADDRESS, a_IP_ADDRESS, a_IP_MASK, d_IP_CONFIG, d_DIP_SWITCH, und a_GATEWAY der angelegten globalen Variable zur Verfügung. Die einzelnen Elemente haben folgende Bedeutung:

Element	Bedeutung
a_MAC_ADDRESS	MAC-Adresse des Optionsmoduls. Feld (Array) mit jeweils 8 Einträgen vom Datentyp BYTE. Gültig sind Feldindizes 0 bis 5. Die Indizes 0 bis 2 bezeichnen den Code zur Hersteller Identifikation, die Indizes 3 bis 5 bezeichnen den Herstellercode für das jeweilige Gerät.
a_IP_ADDRESS, a_IP_MASK, a_GATEWAY	Aktive IP-Adresse, Subnetzmaske und Gateway des Optionsmoduls. Alle drei Elemente bestehen aus einem Feld (Array) mit jeweils 4 Einträgen vom Datentyp USINT. Die jeweilige Adresse beginnt dabei in Feldindex 0 mit dem Netzwerkteil.
d_IP_CONFIG	Bezeichnet die Art der Auswertung der IP-Adresse. 16#00000000 keine Auswertung 16#12345678 feste IP-Adresse 16#12345600 IP-Adresse + Dip-Schalter 1-5
d_DIP_SWITCH	Einstellung der Dip-Schalter 1-5

5 Übersetzen Sie das Programm und laden Sie das Programm als Projekt (auch als Boot-Projekt) auf die b maXX drive PLC. Sie brauchen keine spezielle POE anzulegen.

6 Starten Sie das Programm und wechseln Sie in den Online-Modus. Fügen Sie die angelegte globale Variable zur TCP/IP- Diagnose in das Watchfenster ein, bringen Sie das Watchfenster zur Anzeige und öffnen Sie die relevanten Strukturelemente. Sie sehen jetzt die eingestellten Daten.

Beispiel:

Variable	Value	Default value	Type
_EthernetDiagSlotG			ETHERNET_PLC_DIAG_BMSTRUCT
a_MAC_ADDRESS			BYTE_8_BMARRAY
[0]	16#00		BYTE
[1]	16#02		BYTE
[2]	16#FB		BYTE
[3]	16#FF		BYTE
[4]	16#FE		BYTE
[5]	16#F7		BYTE
[6]	16#00		BYTE
[7]	16#00		BYTE
a_Reserved8			BYTE_8_BMARRAY
a_IP_ADDRESS			USINT_4_BMARRAY
[0]	133		USINT
[1]	12		USINT
[2]	195		USINT
[3]	7		USINT
a_IP_MASK			USINT_4_BMARRAY
[0]	255		USINT
[1]	255		USINT
[2]	192		USINT
[3]	0		USINT
d_IP_CONFIG	16#12345600		DWORD
d_Reserved28	16#00000000		DWORD
d_OMEGA_NUMBER	16#00000004		DWORD
a_GATEWAY			USINT_4_BMARRAY
[0]	0		USINT
[1]	0		USINT
[2]	0		USINT
[3]	0		USINT

Watch 1 | Watch 2 | Watch 3 | Watch 4

Abbildung 24: Beispiel: Auslesen der TCP/IP-Konfiguration im Watchfenster

3.4 Verbindung über Ethernet-TCP/IP in PROPROG wt II / ProProg wt III einstellen

Die Verwendung von Ethernet-TCP/IP in PROPROG wt II / ProProg wt III wird für jede Ressource im Anwenderprogramm einzeln durchgeführt. Gehen Sie dazu wie folgt vor:

- Öffnen Sie im PROPROG wt II / ProProg wt III - Projekt über das Kontextmenü der Ressource b maXX drive PLC den Dialog "Ressource-Einstellungen für SH03_30“
- Stellen Sie den Port auf "DLL"

Einstellung der Ethernet-Kommunikationsquelle durch Auswahl bzw. Angabe der TCP/IP-Adresse

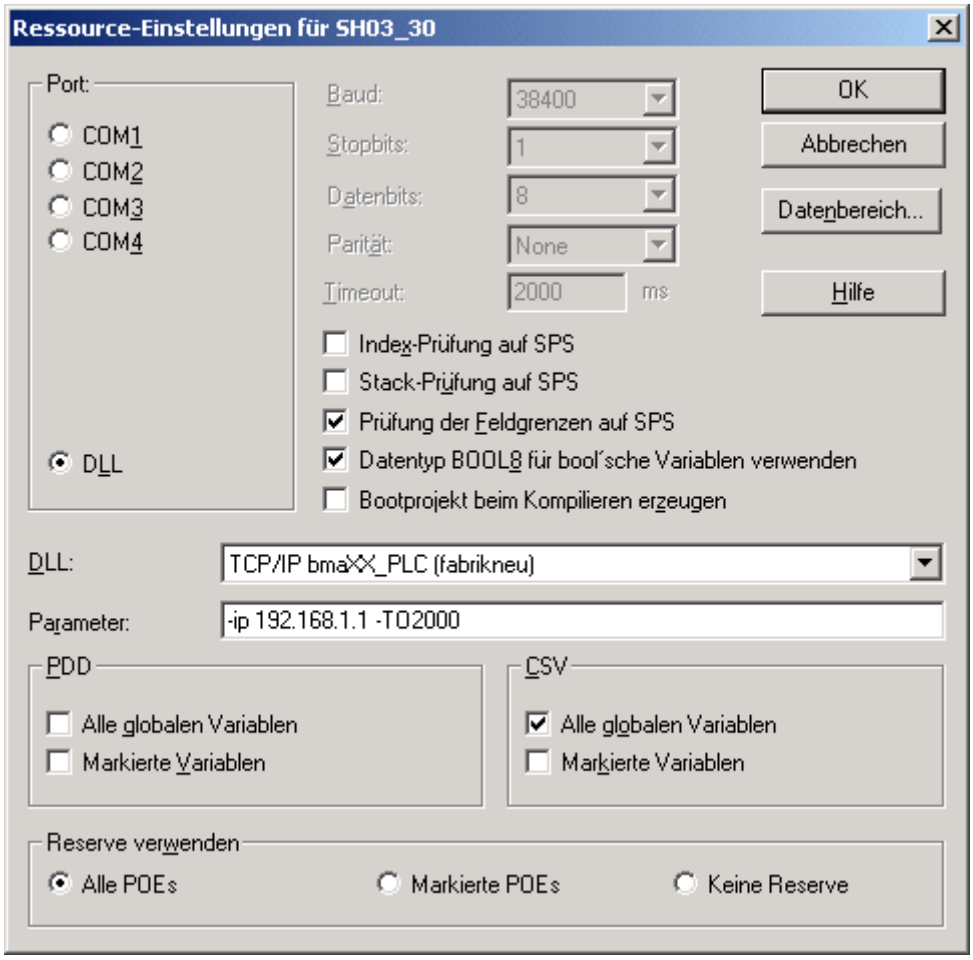


Abbildung 25: TCP/IP in PROPROG wt II für eine Ressource einstellen

Nach Umstellen des Ports auf "DLL" haben Sie über das DLL-Menü die Möglichkeit, die Anwendungsfälle "Soft-PLC" (=TCP/IP local Host (dieser PC)) und "b maXX drive PLC - Zugriff über Ethernet" (=TCP/IP bmaXX_PLC (fabrikneu)) auszuwählen:

- Soft-PLC:
  Die voreingestellte TCP/IP-Adresse im Feld "Parameter" muss nicht geändert werden, wenn die Soft-PLC auf dem gleichen Rechner installiert ist. Bei Installation der Soft-PLC auf einem anderen Rechner müssen Sie dessen entsprechende TCP/IP-Zieladresse (bzw. der entsprechende Netzwerk-Name) hier eintragen, um von PROPROG wt II / ProProg wt III aus über TCP/IP auf die Soft-PLC zugreifen zu können.
- b maXX drive PLC:
  Um auf die b maXX drive PLC über Ethernet zugreifen zu können, muss im Grundgerät b maXX 4400 zusätzlich zur b maXX drive PLC das Optionsmodul Ethernet vorhanden sein. Die im Parameterfeld voreingestellte TCP/IP-Adresse "192.168.1.1" entspricht der IP-Adresse, die auf dem Optionsmodul bei Auslieferung vorinstalliert ist.

3.4 Verbindung über Ethernet-TCP/IP in PROPROG wt II / ProProg wt III einstellen

4

CANopen

In diesem Kapitel finden Sie Informationen zum Datenaustausch über CANopen. CANopen ist nur verfügbar beim Optionsmodul BM4-O-ETH-02 und BM4-O-CAN-04.

4.1 Allgemeines zu CANopen und der Verwendung des Optionsmodul CANopen-Master

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. Mit dem Optionsmodul CANopen-Master für b maXX drive PLC können CANopen- und CANopen-Master-Funktionen nach CiA - Kommunikationsprofil DS 301 und CiA - Geräteprofil DS 405 wie

- Prozessdatenaustausch über Prozess Daten Objekte (PDOs) mit hochprioren Identifiern
- Bedarfsdatenaustausch über Service-Daten-Objekte (SDOs) mit niederprioren Identifiern
- Netzwerkmanagement (NMT) für die Umsetzung von Netzwerkmanagement-Funktionen wie z. B. Initialisierung, Start und Reset von Netzwerkknoten.
- Synchronisierung (SYNC) für die Synchronisierung des Echtzeitdatenaustausches mit PDOs
- Fehlerbehandlung (EMERGENCY): für das Erkennen von Fehlern eines Netzwerkknoten
- Netzwerküberwachung (NODE GUARDING): für die Ausfallüberwachung von Netzwerkknoten

einfach umgesetzt werden.

Für den CAN stehen die neun Übertragungsraten 1 MBit/s, 800 kBit/s, 500 kBit/s, 250 kBit/s, 125 kBit/s, 100 kBit/s, 50 kBit/s, 20 kBit/s und 10 kBit/s zur Verfügung. Das Optionsmodul CANopen-Master kann mit bis zu 32 CANopen-Netzwerkknoten (z. B. I/O-Modulen und Antrieben) kommunizieren.

Um einen Datenaustausch über CANopen mit dem CANopen-Master zu erreichen, stehen Ihnen verschiedene Wege zur Verfügung.

- ProMaster, ProCANopen, ProProg wt III und Motion Control

Einfache und schnelle Konfiguration des Datenaustauschs in ProMaster und ProCANopen. Einfache und schnelle Programmierung der Maschinenfunktionen in PROPROG wt III mit den Motion Control Funktionsbausteinen.

Die Auswertung der Netzwerkzustände und der Netzwerkknotenzustände kann zusätzlich über Funktionsbausteine aus der Bibliothek CANopen_PLC01_30bd00 (oder höher) erfolgen.

Für den Datenaustausch über CANopen mit dem Optionsmodul CANopen Master und ProMaster, ProCANopen, ProProg wt III und Motion Control benötigt das Modul CANopen-Master einen Softwarestand ≥ **01.20** (d. h. ≥ BM4-O-ETH-02/CAN-04-00-00-007-**005**).

Beispiel Projekte:

ProMaster Projekt	Example_2_3_2.bmxl
IEC Projekt in ProProg wt III	Example_BM4_O_CAN04_MC_2.mwt/.zwt

Siehe ▹ProMaster, ProCANopen, ProProg wt III und Motion Control◃ ab Seite 57.

- PROPROG wt II

  Programmierung des Datenaustauschs und der Maschinenfunktionen in PROPROG wt II durch die Verwendung durch die Verwendung des IEC 61131-3 Programmiersystems PROPROG wt II mit der Bibliothek CANop405_PLC01_20bd02 (oder höher)

  Beispiel Projekt:

  IEC Projekt in PROPROG wt II CANopenMaster_Example.mwt/.zwt

  Siehe ▹Programmierung des Datenaustauschs mit PROPROG wt II und Bibliothek CANop405_PLC01_20bd03◃ ab Seite 99.

- PROPROG wt II, Motion Konfigurator und Motion Control

  Einfache und schnelle Konfiguration des Datenaustauschs im Motion Konfigurator

  Einfache und schnelle Programmierung der Maschinenfunktionen in PROPROG wt II mit den Motion Control Funktionsbausteinen.

  Siehe ▹CANopen und Motion Control mit PROPROG wt II und Motion Konfigurator◃ ab Seite 154.

4.2 Grundlagen zu CAN- und CANopen-Netzwerken

4.2.1 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, höchstpriore 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.

4.2.2 Grundlagen CANopen

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

4.2.2.1 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 nutz 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. Elemente in nach IEC 61131-3 programmierbaren Oberflächen werden durch das Geräteprofil DSP 405 definiert.

Das Optionsmodul CANopen-Master für b maXX drive PLC unterstützt das CiA Kommunikationsprofil DS 301, DSP 302 (ab FW 1.20 mit ProMaster) und das Geräteprofil DS 405. Das Kommunikationsprofil DSP 302 wird in Verbindung mit ProMaster ab Softwarestand ≥ **01.20** (d. h. ≥ BM4-O-ETH-02/CAN-04-00-00-007-**005**) unterstützt.

4.2.2.2 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.

4.2.2.3 Datenaustausch und Objekte des physikalischen Bussystems

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

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

und

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

beschrieben. Eine Master - Slave Beziehung (genau ein CANopen-Teilnehmer organisiert den Datenaustausch) existiert auf Ebene der Kommunikationsbeziehungen nicht. Unter einem CANopen-Master versteht man daher nicht einen Master auf Ebene der Kommunikation, sondern einen Master auf Ebene der Anwendung. Unter CANopen-Master Funktionen sind daher hauptsächlich Aufgaben des Netzwerkmanagement wie

- Start und Stopp von Netzwerkknoten
- Überwachung von Netzwerkknoten
- Konfiguration des Datenaustausches

zu sehen. Dabei bedient sich der CANopen-Master der Funktionen der Kommunikationsebene. Da solche Funktionen zudem eng mit der eigentlichen Anwendung verknüpft sind, ist es sinnvoll, dass ein CANopen-Master unter IEC 61131-3 programmierbar ist. Beim Optionsmodul CANopen-Master ist dies zusammen mit der Bibliothek

- CANopen_PLC01_30bd00 (oder höher) unter ProProg wt III bzw.
- CANop405_PLC01_20bd03 (oder höher) unter PROPROG wt II

möglich.

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.
  - ProMaster, ProCANopen, ProProg wt III, (ggf. Bibliothek CANopen_PLC01_30bd00 (oder höher)):
    Mit dem Optionsmodul CANopen Master können maximal 256 PDOs gleichzeitig geschrieben und 256 PDOs gelesen werden.
  - PROPROG wt II, Bibliothek CANop405_PLC01_20bd03 (oder höher):
    Mit dem Optionsmodul CANopen Master können maximal 40 PDOs gleichzeitig geschrieben und 63 PDOs gelesen werden.
- Service-Daten-Objekte (SDOs): Parameterdatenaustausch mit niederprioren Identifiern und durch Index/Subindex adressierbare Daten.
  - ProMaster, ProCANopen, ProProg wt III, (ggf. Bibliothek CANopen_PLC01_30bd00 (oder höher)):
    Das Optionsmodul CANopen Master unterstützt die Transfertypen "expedited" (4 Byte pro Nachricht) und "segmented" (bis zu 7 Byte pro Nachricht). Es kann ein SDO geschrieben oder gelesen werden.
  - PROPROG wt II, Bibliothek CANop405_PLC01_20bd03 (oder höher):
    Der Transfertyp beim Optionsmodul CANopen Master ist "expedited", d. h. es kön-

nen bis zu 4 Byte pro Nachricht übertragen werden. Abhängig von der Netzwerkkonfiguration können gleichzeitig bis zu acht unterschiedliche SDOs gelesen oder geschrieben werden.

- Netzwerkmanagement (NMT): Spezialobjektart für die Umsetzung von Netzwerkkommunikations-Funktionen wie z. B. Initialisierung, Start und Reset von Netzwerkknoten. Das Optionsmodul CANopen-Master sendet die Kommandos als Broadcast an die jeweiligen Netzwerkknoten.
- Synchronisierung (SYNC): Spezialobjektart für die Synchronisierung des Echtzeitdatenaustausches mit PDOs. Das Optionsmodul CANopen-Master sendet das SYNC-Kommando als Broadcast an die jeweiligen Netzwerkknoten.
- Fehlerbehandlung (EMERGENCY): Spezialobjektart für das Erkennen von Fehlern eines Netzwerkknoten. Tritt auf einem Netzwerkknoten ein Fehler auf und wird von diesem ein Emergency-Telegramm gesendet, so kann dies Optionsmodul CANopen-Master empfangen und ausgewertet werden.
- Netzwerküberwachung
  NODE GUARDING:
  - Spezialobjektart für die Ausfallüberwachung von Netzwerkknoten. Das Optionsmodul CANopen-Master kann zyklisch über ein Telegramm die Rückmeldung von Netzwerkknoten anfordern, um den Ausfall von Netzwerkknoten zu erkennen.

  HEARTBEAT:
  - ProMaster, ProCANopen, ProProg wt III, (ggf. Bibliothek CANopen_PLC01_30bd00 (oder höher)):
    Spezialobjektart für die Ausfallüberwachung von Netzwerkknoten. Das Optionsmodul CANopen Master erkennt den Ausfall von Netzwerkknoten daran, dass diese Netzwerkknoten kein Heartbeat-Telegramm mehr senden. Der Vorteil von Heartbeat gegenüber dem Node Guarding ist ein geringerer Telegrammverkehr auf dem CANopen und damit eine geringere Buslast.
  - PROPROG wt II, Bibliothek CANop405_PLC01_20bd03 (oder höher):
    Heartbeat wird nicht unterstützt.

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 52, 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.

4.2.2.4 Vordefinierte Identifier

Um direkt nach einem Boot-Up eine Peer-to-Peer Kommunikation zwischen Master und den übrigen Netzwerkknoten aufbauen zu können, existiert für die Telegramme eine vordefinierte Identifiervergabe. Diese Identifierzuordnung 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-Master 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 [1)]	0010	256	16#100

[1)] 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

Objektname	Function Code binär	resultierende COB-ID dezimal	resultierende COB-ID hexadezimal
SDO (TX)	1011	1409 - 1535	581h - 5FFh
SDO (RX)	1100	1537 - 1663	601h - 67Fh
NODE GUARDING	1110	1793 - 1919	701h - 77Fh

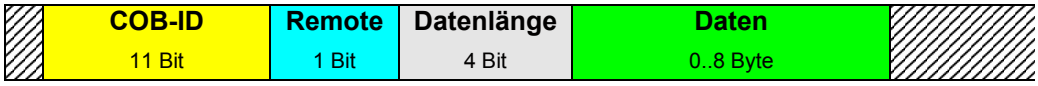
TX steht für transmit (senden) und RX für receive (empfangen). Beachten Sie, dass die Übertragungsrichtung von den Netzwerkknoten und nicht vom Master aus betrachtet wird. D. h. wird vom Optionsmodul z. B. ein PDO gesendet, ist dies für den auswertenden Netzwerkknoten ein RxPDO und muss also mit einem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

Diese Gegebenheiten gelten jedoch nur für die vordefinierte Identifiervergabe. 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 mehren) Netzwerkknoten eingerichtet werden.

4.2.2.5 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-Identifier entspricht, ergibt sich auf Grund des CAN-Übetragungsverfahrens CSMA/CA (siehe Kapitel ▹Grundlagen CAN◃ ab Seite 51), dass niederwertige COB-IDs eine höhere Priorität haben. Aus der vordefinierten Identifiervergabe ergibt sich somit, dass NMT-Telegramme die höchste Priorität haben, gefolgt von SYNC-Telegrammen. Da die COB-ID zudem die Netzwerkknotennummer 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.

4.3 ProMaster, ProCANopen, ProProg wt III und Motion Control

In diesem Kapitel erläutern wir, wie Sie einfach und schnell ein kleines CANopen-Netzwerk mit der b maXX drive PLC und dem Optionsmodul CANopen Master in Betrieb nehmen können. Dieses Kapitel wird durch ein ProMaster Beispielprojekt begleitet.

Ein CANopen-Netzwerk besteht aus einem CANopen-Master und einem oder mehreren CANopen-Slaves.

Mit dem Baumüller Engineering Framework ProMaster wird die Maschinenkonfiguration aus diesen Geräten erstellt. Mit dem Feldbuskonfigurator ProCANopen wird die CANopen-Kommunikation zwischen den einzelnen Geräten konfiguriert. Durch die Verwendung von Motion Control werden die Konfigurationseinstellungen auf ein Minimum beschränkt und der Anwender kann sich voll auf die eigentlichen Applikation (Maschinenfunktionalität) im IEC 61131-3 Programmiersystem ProProg wt III konzentrieren.

Die bei der Konfiguration der CANopen-Kommunikation entstehenden Daten und Einstellungen für CANopen werden auf dem Optionsmodul CANopen-Master hinterlegt. Dies betrifft die Daten für den CANopen-Master selbst, als auch die Daten für die CANopen-Slaves. Erst nach dem Einschalten der Maschine bekommen die CANopen-Slaves ihre Daten vom CANopen-Master.

Die bei der Konfiguration der CANopen-Kommunikation entstehenden Daten und Einstellungen für ProProg wt III werden im IEC Projekt (ProProg wt III Projekt) des Gerätes mit dem Ωmega Drive Line II hinterlegt. Dieses IEC Projekt wird auf die b maXX drive PLC geladen, die zusammen mit dem Ωmega Drive Line II im b maXX 4400 steckt.

Das Optionsmodul Ethernet mit CANopen Master (BM4-O-ETH-02) erlaubt Ethernet-Kommunikation bei Single Axis Motion Control. Bei Multi Axis Motion Control sollte Ethernet-Kommunikation nicht verwendet werden, da hierdurch die Synchrongenauigkeit im CANopen-Netzwerk reduziert wird.

HINWEIS

Verwenden Sie keine Ethernet-Kommunikation und Multi Axis Motion Control gemeinsam auf einem Optionsmodul Ethernet mit CANopen Master (BM4-O-ETH-02).

In einem CANopen-Netzwerk, in dem Multi Axis Motion Control verwendet wird, sollten keine "nicht Multi Axis Motion Control" CANopen-Slaves verwendet werden. Dies betrifft insbesondere IO-Module. Die "nicht Multi Axis Motion Control" CANopen-Slaves können die Synchronität im CANopen-Netzwerk stören. Verwenden sie für "nicht Multi Axis Motion Control" CANopen-Slaves ein zweites Optionsmodul CANopen-Master.

HINWEIS

Verwenden Sie keine "nicht Multi Axis Motion Control" CANopen-Slaves und "Multi Axis Motion Control" CANopen-Slaves gemeinsam an einem Optionsmodul (Ethernet mit) CANopen Master (BM4-O-ETH-02 / BM4-O-CAN-04).

4.3.1 Voraussetzungen

Die Voraussetzungen sind

- PC mit
  - Engineering Framework ProMaster
  - IEC 61131-3 Programmiersystem ProProg wt III
    mit den Firmware Bibliotheken
    - Bit_UTIL_30bd00
    - OmegaOS_30bd00
    - SYSTEM2_PLC01_30bd00

    und den Bibliotheken
    - BM_TYPES_30bd00
    - SYSTEM1_PLC01_30bd00
    - CANopen_PLC01_30bd00 (bei Geräten die Motion Control nicht unterstützen)
    - MOTION_TYPES_30bd00
    - MOTION_CONTROL_30bd00
    - MOTION_MULTI_AXIS_30bd00
  - Antriebsbedienprogramm für b maXX WinBASS II (zur Inbetriebnahme der CANopen-(Regler)-Slaves.
- Physikalische Inbetriebnahme der Komponenten des CANopen-Netzwerkes,
  in unserem Beispiel sind das der CANopen-Master
  - b maXX 4400 mit
    - b maXX drive PLC (BM4-O-PLC-0x-…) mit
    - Optionsmodul CANopen-Master (BM4-O-CAN-04-… oder BM4-O-ETH-02-…)

  und der CANopen-Slave
  - b maXX 4400 mit
    - Optionsmodul CANopen-Slave (BM4-O-CAN-03-…)

Siehe hierzu die jeweilige Betriebsanleitung.

4.3.2 Durchzuführende Schritte

Um für Motion Control das Optionsmodul CANopen-Master für b maXX drive PLC in einem CANopen-Netzwerk verwenden zu können, sind folgende Schritte durchzuführen.

1 Inbetriebnahme des CANopen-Netzwerks

2 Anlegen eines IEC 61131-3 Projekts für ProProg wt III mit einem Motion Control Template

3 Anlegen einer Maschinenkonfiguration in ProMaster

4 Konfigurieren der CANopen Kommunikation mit ProCANopen
(ggf. Testen des CANopen-Netzwerks mit dieser Konfiguration)

5 Konfigurieren der PLC mit Verbinden von IEC Projekt und ProMaster und Konfigurieren der Kurvenscheibendaten

6 Gegebenenfalls Programmieren des IEC 61131-3 Projekts für ProProg wt III (Applikation) und anschließender Download auf die b maXX drive PLC.

7 Betrieb der Applikation im CANopen-Netzwerk

Natürlich können Sie auch mit Schritt 2 beginnen und erst vor Schritt 6 die physikalische Inbetriebnahme (Schritt 1) durchführen.

Dabei entstehen das ProMaster Projekt **Example_2_3_2.bmxml** und das IEC 61131-3 Projekt (ProProg wt III Projekt) **Example_BM4_O_CAN04_MA_2.mwt/.zwt**.

4.3.3 Inbetriebnahme des CANopen-Netzwerk

Einzelheiten zur Inbetriebnahme des CANopen-Netzwerks entnehmen Sie bitte der Betriebsanleitung zum Optionsmodul BM4-O-ETH-02 / BM4-O-CAN-04 und den Betriebsanleitungen der übrigen CANopen-Netzwerkknoten.

Nach der physikalischen Inbetriebnahme des Netzwerks und Zuschalten der Spannungsversorgung für das b maXX 4400 Gerät ist das Optionsmodul CANopen-Master nach ca. 5 s betriebsbereit. Dies wird Ihnen durch die CANopen-LEDs an der Frontseite des Optionsmoduls angezeigt. Zwei Kombinationen sind möglich:

H5 blinkt grün: 200 ms an/aus H6 rot: aus	CANopen: Das Optionsmodul wartet auf die Initialisierung durch ein Applikationsprogramm auf der b maXX drive PLC
H5 grün: aus H6 rot: an	CANopen: Der CAN ist im Bus-Off Zustand

Alle anderen Zustände der LEDs H5 und H6 deuten auf Fehler hin. Sehen sie hierzu bitte in der zugehörigen Betriebsanleitung zum Optionsmodul CANopen-Master BM4-O-ETH-02 / BM4-O-CAN-04 nach.

4.3.4 Anlegen eines IEC Projekts mit ProProg wt III

Im Folgenden erläutern wir, wie ein Motion Control IEC Projekt angelegt wird.

In den ProMaster Default Einstellungen für Motion Control ist Multi Axis Motion Control eingestellt. Daher ist es sinnvoll, zunächst ein IEC Projekt für Multi Axis Motion Control zu erstellen.

Öffnen Sie ProProg wt III und legen Sie eine neues Projekt an, indem Sie über das Menü "Datei\Neues Projekt" ein IEC-Template auswählen.

Für unser Beispiel verwenden wir das IEC-Template Projekt „Tmpl_PLC01_MA_2_0103.zwt“ (für BM4_O_PLC01). Dieses Projekt finden Sie im ProMaster Installationsordner \\Baumueller\ProMaster:NET im Unterordner \\...\projects\IEC-Templates. Das IEC-Template „Tmpl_PLC01_MA_2_0103.zwt entpacken Sie als „Example_BM4_O_CAN04_MA_2.mwt“.

Die Vorlage enthält die Bibliotheken

- Bit_UTIL_30bd00
- BM_TYPES_30bd00
- MC_SYS_30bd00
- MOTION_TYPES_30bd00
- MOTION_CONTROL_30bd00
- MOTION_MULTI_AXIS_30bd00

Falls Sie nicht alle Achsen als Motion Control Achsen verwenden wollen oder weitere CANopen Geräte in Ihre Maschinenkonfiguration einbinden wollen, binden Sie jetzt noch die Bibliotheken

- BM_TYPES_30bd00
- CANopen_PLC01_30bd00

zur Auswertung der Netzwerkzustände und der Netzwerkknotenzustände in Ihr Projekt ein.

Speichern Sie das Projekt. In unserem Beispiel verwenden wir den Projektnamen "**Example_BM4_O_CAN04_MA_2.mwt**".

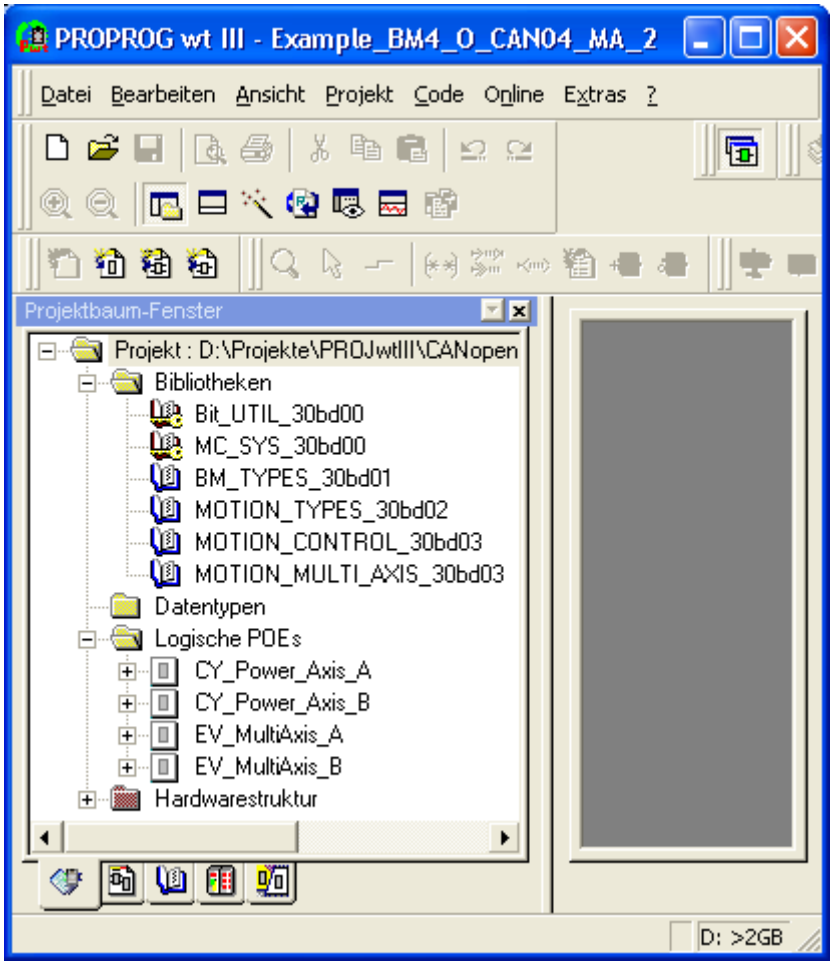


Abbildung 26: ProProg wt III - Anlegen des Projekts "Example_BM4_O_CAN04_MA_2.mwt"

Jetzt müssen noch die Kommunikationseinstellungen konfiguriert werden. Per Default ist die serielle Schnittstelle COM1 eingestellt. Falls Sie Ethernet verwenden wollen, siehe Kapitel ▹Ethernet◃ ab Seite 17.

Für die Verwendung eines IEC Projekts in ProMaster ist es unbedingt erforderlich, dass bei ProProg wt III im Fenster "Ressource - Einstellungen" (Ressource markieren, Kontextmenü "Einstellungen" anwählen) die Check-Box "Bootprojekt beim Kompilieren erzeugen" aktiviert ist. Dadurch wird die Datei "bootfile.pro" erzeugt, welche für den Download auf die b maXX drive PLC benötigt wird.

HINWEIS

In ProProg wt III muss unter "Ressource - Einstellungen" die Check-Box "Bootprojekt beim kompilieren erzeugen" aktiviert sein.

Das Projekt "Example_BMC_M_CAN04_MA_2.mwt" wird später, im Kapitel ▹4.3.7 Konfigurieren der PLC◃ ab Seite 84, mit dem ProMaster Projekt verbunden.

Schließen Sie jetzt ProProg wt III über Menü „Datei\Beenden“.

4.3.5 Anlegen einer Maschinenkonfiguration in ProMaster

In diesem Abschnitt erläutern wir, wie ein Projekt in ProMaster angelegt wird und wie eine Maschinenkonfiguration erstellt wird.

Öffnen Sie ProMaster.

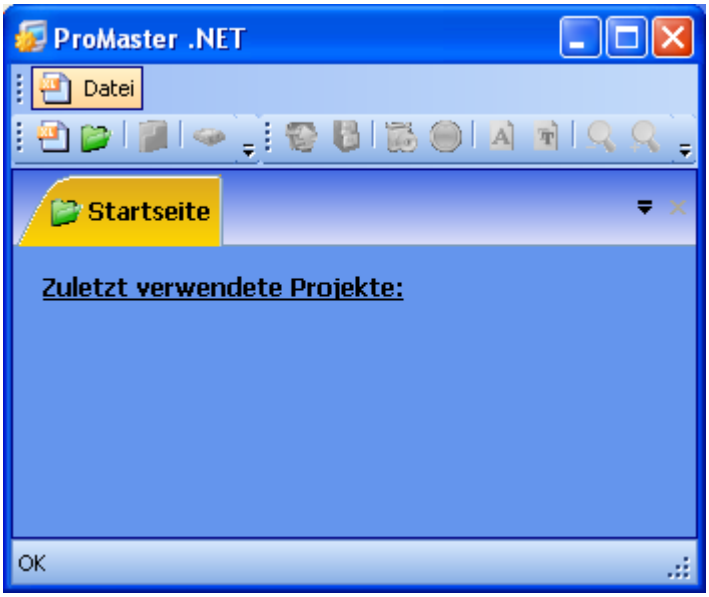


Abbildung 27: ProMaster ohne Projekte

Legen Sie ein neues ProMaster-Projekt an, indem Sie über Menü "Datei\NeuesProjekt" das Fenster "Projekt Einstellungen" öffnen.

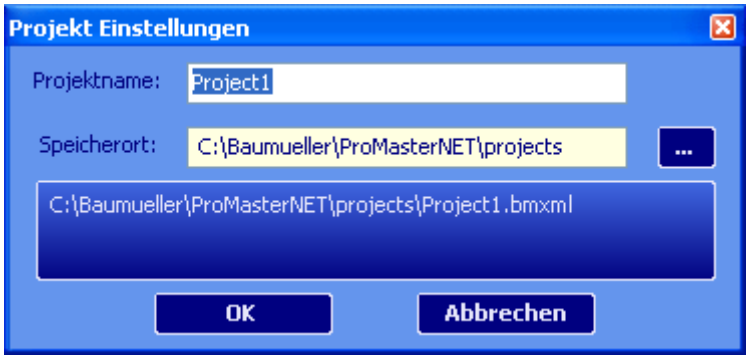


Abbildung 28: ProMaster - Projektname

Geben Sie den Namen (in der Edit-Box) und den Speicherort (über die Pfadauswahl) des Projektes an. Für unser Beispielprojekt verwenden wir den Namen "Example_2_3_2.bmxml".

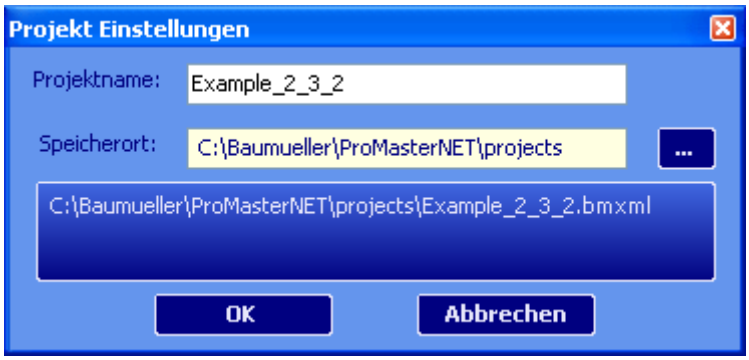


Abbildung 29: ProMaster - Projektname vergeben

Übernehmen Sie die Einstellungen aus dem Fenster "Projekt Einstellungen" durch Klicken auf den Button "OK". ProMaster wechselt jetzt in die Netzwerkansicht.

Abbildung 30: ProMaster - Projektname vergeben

Öffnen Sie jetzt über das Menü "Ansichten\Katalog" den Baumüller Katalog. Der Baumüller Katalog enthält die Geräte, Bussystem und Komponenten, die Baumüller Ihnen per Default zur Verfügung stellt.

Für unser Beispiel verwenden wir

- die Geräte
  - "bmaXX 4000 PLC CANopen-Master (2-reihig)" (aus der Gruppe "b maXX 4000 Antrieb"),
    b maXX 4000 mit zwei Steckplätzen,
    Optionsmodul Ethernet mit CANopen-Master in Steckplatz G,
    Optionsmodul b maXX PLC in Steckplatz H
  - "bmaXX 4000 CANopen-Slave (2-reihig)" (aus der Gruppe "b maXX 4000 Antrieb"),
    b maXX 4000 mit zwei Steckplätzen,
    Optionsmodul Ethernet mit CANopen-Slave in Steckplatz G
- das Bussystem
  - "bus_canopen" (CANopen), wird automatisch angelegt.

Ziehen Sie per Drag&Drop das Gerät "bmaXX 4000 PLC CANopen-Master (2-reihig)" aus dem Baumüller Katalog in die ProMaster Netzwerkansicht. Das Bussystem ist automatisch angeschlossen worden. Ziehen Sie jetzt per Drag&Drop das Gerät "bmaXX 4000 CANopen-Slave (2-reihig)" auf den Bus CANopen (Drop ist erst möglich, wenn der Bus die Farbe wechselt).

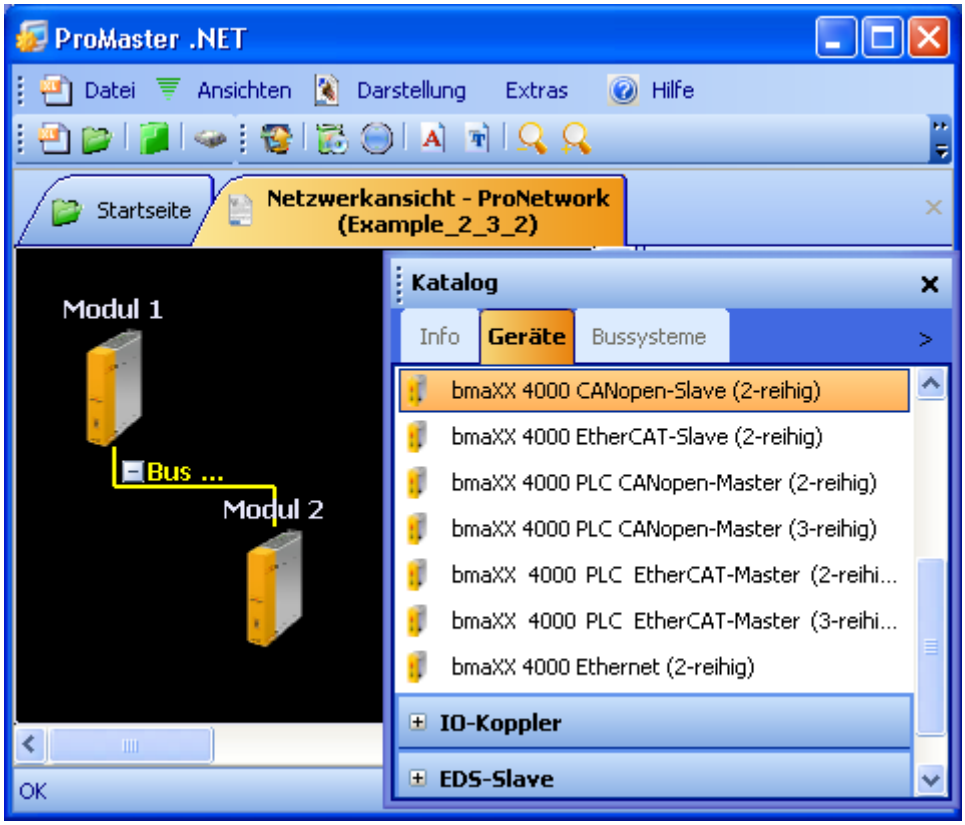


Abbildung 31: ProMaster - Maschinenkonfiguration angelegt

Im Bild sehen Sie jetzt den CANopen-Master (Modul 1) und den CANopen-Slave (Modul 2). Weitere Informationen zu den Modulen erhalten Sie über den jeweiligen Tooltip oder das "Projektbaum Fenster" (Project tree). Der Übersichtlichkeit halber wird jetzt das Fenster "Katalog" geschlossen.

Die Namen (Modul 1 und Modul 2) werden geändert indem man auf das jeweilige Modul klickt und dann über das Kontextmenü und „Eigenschaften“ das Eigenschaften-Fenster des jeweiligen Moduls öffnet.

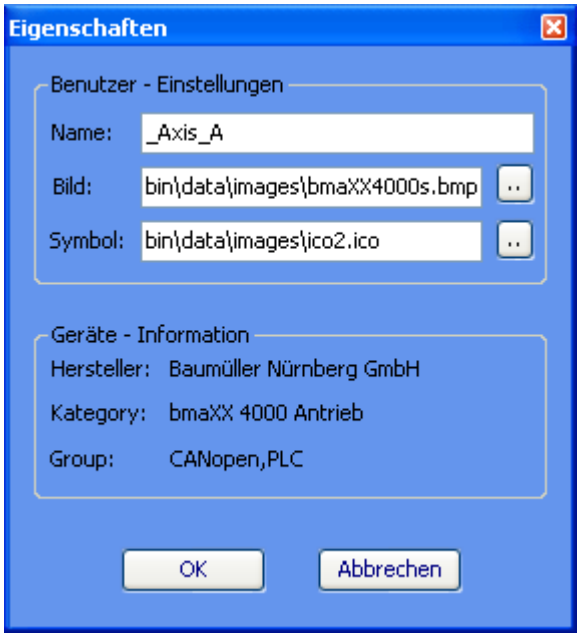


Abbildung 32: ProMaster - Modul-Eigenschaften

Für unser Beispiel vergeben wir dem CANopen-Master den Namen _Axis_A und für den CANopen-Slave den Namen _Axis_B (_Axis_A und _Axis_B sind die Default Achsvariablennamen der Motion Control Achsen in unserem IEC Projekt). Die jeweiligen Bilder (*.bmp) behalten wir. In der ProMaster Netzwerkansicht und im Projektbaum stehen nun die neuen Namen.

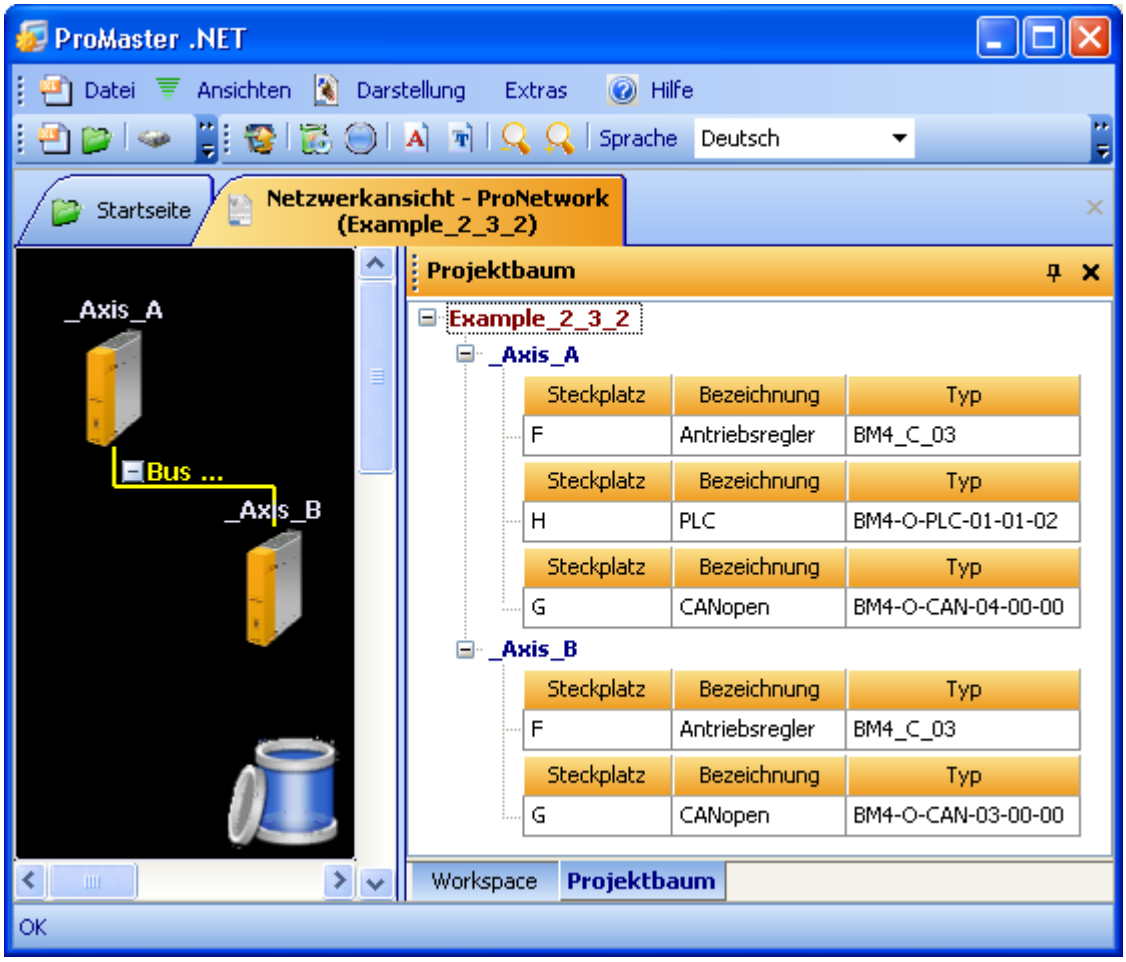


Abbildung 33: ProMaster - Maschinenkonfiguration - Module umbenannt

Die Kommunikationseinstellungen für die Kommunikation von ProMaster zur b maXX drive PLC können Sie einstellen in dem Sie auf das Gerät „_Axis_A“ klicken und über das Kontextmenü „Kommunikationseinstellungen“ anwählen. Speichern Sie anschließend das Projekt über Menü „Datei\Projekt speichern“.

In ProMaster ist für die Komponenten mit CANopen (CANopen-Master-Antrieb, CANopen-Slave-Antrieb) per Default Motion Control aktiviert. Dadurch verringern sich die zu konfigurierenden Einstellungen auf ein Minimum.

Falls Sie die Default Einstellungen (z. B. CANopen Baudrate 1 MBit/s und die automatisch vergebene CANopen Node-ID) verwenden wollen, brauchen Sie keine Konfiguration der CANopen-Kommunikation mit ProCANopen vornehmen.

In diesem Fall geht es weiter mit ▹4.3.7 Konfigurieren der PLC◃ ab Seite 84.

4.3.6 Konfigurieren der CANopen Kommunikation mit ProCANopen

In ProMaster ist für die Komponenten mit CANopen (CANopen-Master-Antrieb, CANopen-Slave-Antrieb) per Default Motion Control aktiviert. Dadurch verringern sich die zu konfigurierenden Einstellungen auf ein Minimum.

Falls Sie die Default Einstellungen (z. B. CANopen Baudrate 1 MBit/s und die automatisch vergebene CANopen Node-ID) verwenden wollen, brauchen Sie keine Konfiguration der CANopen-Kommunikation mit ProCANopen vornehmen.

In diesem Fall geht es weiter mit ▹4.3.7 Konfigurieren der PLC◃ ab Seite 84.

Wie Sie die Einstellungen der CANopen-Kommunikation ändern können, erläutern wir im nachfolgenden Abschnitt.

Zuerst werden die CANopen-Slaves konfiguriert, anschließend wird der CANopen-Master konfiguriert.

4.3.6.1 Konfigurieren der CANopen-Slave Kommunikation

Die CANopen-Slave Kommunikation wird für jeden CANopen-Slave einzeln konfiguriert.

Öffnen Sie ProCANopen für unseren CANopen-Slave in dem Sie auf das Gerät „_Axis_B“ klicken und dann über das Kontextmenü "Konfigurationsdaten (Komponenten)\CANopen Slave (Steckplatz G)\Slave Bus konfigurieren (ProCANopen)" anwählen.

Der CANopen-Slave wurde bereits beim Anlegen der Maschinenkonfiguration per Default so eingestellt, dass die CANopen Einstellungen für Motion Control konfiguriert sind (Drag&Drop des Slaves an einen Master, bei dem Motion Control aktiviert ist).

Das Fenster für die CANopen-Slave Kommunikationskonfiguration wird geöffnet.

Register Ident

Für Motion Control können Sie im Register "Ident" den Steckplatz des Optionsmodul CANopen-Slave im b maXX 4400, den Namen dieses Moduls (Gerätename) in ProMaster sowie für CANopen die Baudrate und die Node-ID einstellen. Baudrate und Node-ID wurden aus der CANopen-Konfiguration des Moduls CANopen-Master übernommen.

Die Änderung von Gerätename, Baudrate und Node-ID wirkt sich auf das ProMaster Projekt aus.

HINWEIS

Die Reduzierung der Baudrate kann bei unveränderter „Zykluszeit für SYNC“ (Register Netzwerk) zu einer zu hohen Buslast auf dem CANopen führen und damit Kommunikation und Synchronität negativ beeinflussen.

Die Änderung des Gerätenamens wirkt sich zusätzlich auf den Motion Control Achsnamen im IEC 61131-3 Projekt in ProProg wt III aus.

HINWEIS

Änderungen des Gerätenamens wirken sich auf den Motion Control Achsnamen im globalen Variablenarbeitsblatt im IEC 61131-3 Projekt in ProProg wt III aus.

Diese Änderung wird **nicht** automatisch in den lokalen Variablenarbeitsblätter und in den Code-Arbeitsblättern im IEC 61131-3 Projekt in ProProg wt III durchgeführt.

Dies bleibt (aus Sicherheitsgründen) explizit Aufgabe des Anwenders.

Weiterhin werden Ihnen CANopen spezifische Informationen wie Gerätetyp, CANopen-Profil, Hersteller-ID usw. angezeigt.

Bei einer Serienfertigung von Maschinen kann über die Check-Boxen "Im Boot-Up überprüfen" beim Start überprüft werden, ob die richtigen Geräte angeschlossen sind.

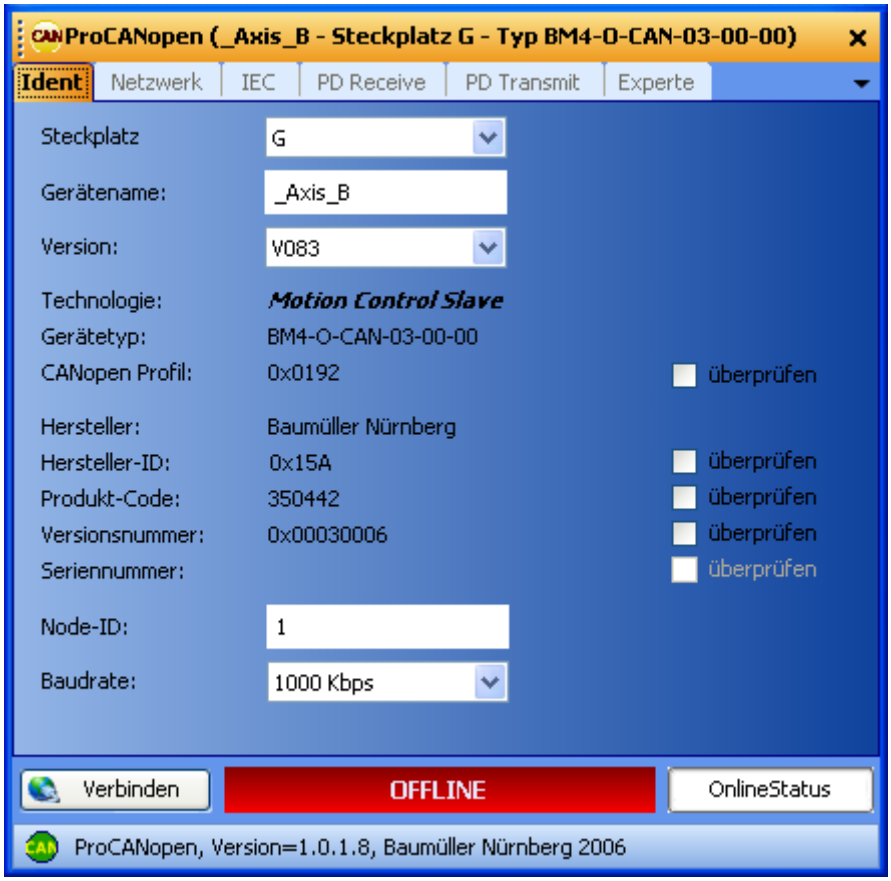


Abbildung 34: ProCANopen - CANopen-Slave Kommunikationskonfiguration Register Ident

Register Netzwerk

Im Register "Netzwerk" können Sie die SYNC-Zykluszeit und Guarding bzw. Heartbeat einstellen.

Die Änderung der SYNC-Zykluszeit und der Guarding- bzw. Heartbeat- Einstellungen wirkt sich auf das ProMaster Projekt aus. Die SYNC-Zykluszeit ist das Zeitintervall, in dem der CANopen-Master das SYNC-Telegramm auf dem CANopen-Bus sendet.

HINWEIS

Ein CANopen-Slave kann nur Guarding oder Heartbeat unterstützen.

HINWEIS

Die Reduzierung der „Zykluszeit für SYNC“ kann bei unveränderter „Baudrate“ (Register Ident) zu einer zu hohen Buslast auf dem CANopen führen und damit Kommunikation und Synchronität negativ beeinflussen.

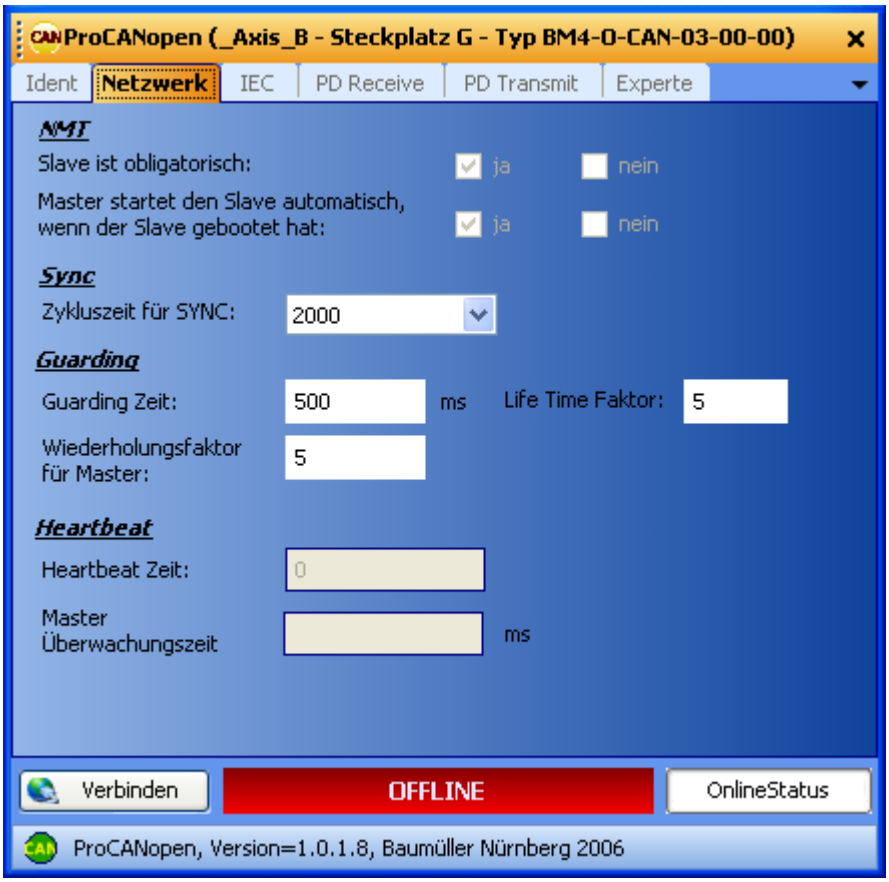


Abbildung 35: ProCANopen - CANopen-Slave Kommunikationskonfiguration Register Netzwerk

Register IEC

Bei der Konfiguration der CANopen-Slave Kommunikation erfolgt die Zuordnung der CANopen-Slave Objekte (über CANopen-Master Objekte) zu den Netzwerkvariablen für das IEC-Projekt auf der PLC.

Die im Register IEC angezeigten Variablen werden auch Netzwerkvariablen genannt. Über die Netzwerkvariablen erfolgt das Prozessdaten schreiben und lesen im IEC-Projekt.

Bei der Verwendung von Motion Control sind einige der Zuordnungen per default angelegt.

Im Register „IEC“ werden die Netzwerkvariablen und ihr Datentyp im IEC-Projekt auf der PLC angezeigt.

Die Berechnung der (IEC-) Adresse der Netzwerkvariablen erfolgt erst nach „Liste aktualisieren“ in Kapitel ▹Register Download◃ in Kapitel "4.3.6.2 Konfigurieren der CANopen-Master Kommunikation" ab Seite 82.

Die Konfiguration von weiteren Netzwerkvariablen erfolgt in den Registern PD Receive und PD Transmit.

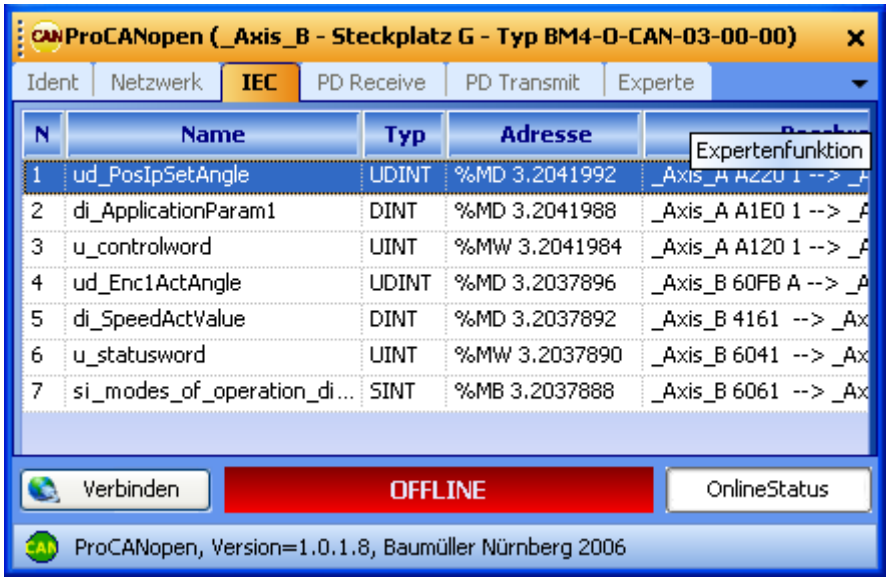


Abbildung 36: ProCANopen - CANopen-Slave Kommunikationskonfiguration Register IEC nach Konfiguration der CANopen-Slaves und „Liste aktualisieren“ (CANopen-Master Kommunikationskonfiguration, Register Download)

Register PD Receive

Im Register „PD Receive“ wird die Kommunikation vom Master zum Slave konfiguriert.

Für Motion Control ist die Default Konfiguration bereits eingestellt. Sie können die Konfiguration erweitern. Beachten Sie dabei, dass die Motion Control Konfiguration eine höhere Priorität hat als die Konfiguration durch den Anwender.

Abbildung 37: ProCANopen - Motion Control CANopen-Slave Kommunikationskonfiguration Register „PD Receive“ mit Default Motion Control Konfiguration

Der PDO Sender ist bei Motion Control der CANopen-Master.

Hier findet die Verknüpfung eines Objekts im CANopen-Slave mit einem Objekt im CANopen-Master statt. Dieses Objekt im CANopen-Master wird über eine Netzwerkvariable aus dem IEC Programm geschrieben.

Vorgehensweise:

- Auswahl eines Objekts des CANopen-Slaves im Bereich "PDO Empfänger" in der Combo-Box "Objekt-/Variablenliste".

  Beispiel: Wir wählen im Bereich "PDO Empfänger" in der Combo-Box "Objekt-/Variablenliste" das CANopen-Slave Objekt 0x4CF2 (Applikationsparameter 1).

- Auswahl des PDO Übertragungstyps aus der Combo-Box "PDO Übertragungstyp". Sie können zwischen Synchron 1 bis Synchron 240 und Asynchron 254 sowie Asynchron 255 wählen.

  Beispiel: Für unser Beispiel wählen wir "Synchron 1", d. h. das vor dem letzten SYNC-Telegramm empfangene Receive PDO (bzw. dessen Inhalt) wird übernommen, wenn das neue SYNC-Telegramm empfangen wurde.

- Mit dem Button "Neue PDO hinzufügen" wird die eben eingestellte Verknüpfung auf Konformität zu den PDO-Einträgen überprüft und in die Liste eingetragen.

  Das CANopen-Slave Objekt 0x4CF2 (Applikationsparameter 1) ist jetzt mit dem CANopen-Master Objekt 0xA1E0, Subindex 0x01 verbunden. ProMaster und ProCANopen generieren für dieses CANopen-Master Objekt automatisch eine IEC-

Variable (Netzwerkvariable) für das IEC-Projekt. In unserem Fall ist der Name der Netzwerkvariable im IEC-Projekt "di_ApplicationParam1". Im IEC-Projekt wird auf diese Netzwerkvariable geschrieben. Der geschriebene Wert wird über das CANopen-Master Objekt 0xA1E0 - 0x01 auf das CANopen-Slave Objekt 0x4CF2 geschrieben. Siehe ▹Register IEC◃ in Kapitel "4.3.6.2 Konfigurieren der CANopen-Master Kommunikation" ab Seite 81.

Abbildung 38: ProCANopen - Motion Control CANopen-Slave Kommunikationskonfiguration Register „PD Receive“ mit Default Motion Control Konfiguration und zusätzlichem Objekt

HINWEIS

ProMaster berücksichtigt automatisch die Grenze von 30 Zeichen für einen Variablennamen im IEC Projekt in ProProg wt III.

Die Namen der Variablen aus der zusätzlichen Konfiguration können später, beim Konfigurieren der PLC, in Kapitel ▹4.3.7.4 CANopen◃ ab Seite 90 geändert werden.

Register PD Transmit

Im Register „PD Transmit“ wird die Kommunikation vom Slave zum Master konfiguriert.

Für Motion Control ist die Default Konfiguration bereits eingestellt. Sie können eine zusätzliche Konfiguration durchführen. Beachten Sie dabei, dass die Motion Control Konfiguration eine höhere Priorität hat als die Konfiguration durch den Anwender.

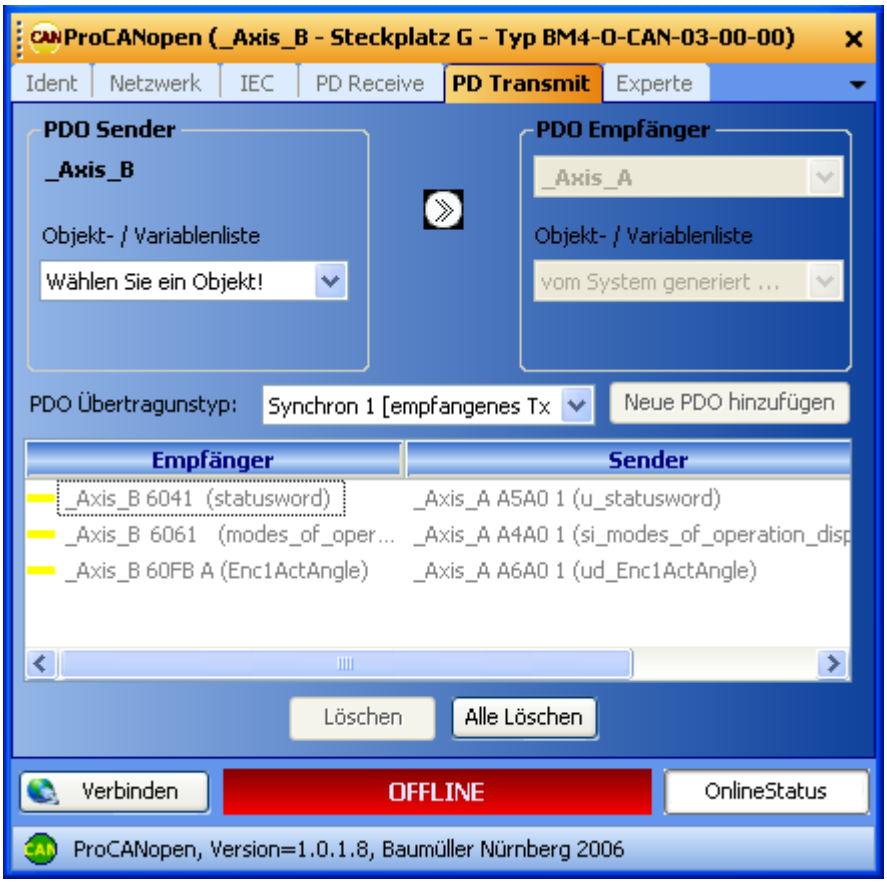


Abbildung 39: ProCANopen - Motion Control CANopen-Slave Kommunikationskonfiguration Register „PD Transmit“ mit Default Motion Control Konfiguration

PDO Empfänger ist bei Motion Control der CANopen-Master.

Hier findet die Verknüpfung eines Objekts im CANopen-Slave mit einem Objekt im CANopen-Master statt. Dieses Objekt im CANopen-Master wird über eine Netzwerkvariable im IEC Programm gelesen.

Vorgehensweise:

- Auswahl des zu sendenden Objekts im Bereich "PDO Sender" in der Combo-Box "Objekt-/Variablenliste".

  Beispiel: Wir wählen im Bereich "PDO Sender" in der Combo-Box "Objekt-/Variablenliste" das CANopen-Slave Objekt 0x4161 (Drehzahl-Istwert).

- Auswahl des PDO Übertragungstyps aus der Combo-Box "PDO Übertragungstyp". Sie können zwischen Synchron 1 bis Synchron 240 und Asynchron 254 sowie Asynchron 255 wählen.

  Beispiel: Wir wählen "Synchron 1", d. h. das Senden erfolgt jeweils nach dem Empfang des SYNC-Telegramms (Senden nach dem ersten SYNC-Telegramm seit dem letzten Senden).

- Mit dem Button "Neue PDO hinzufügen" wird die eben eingestellte Verknüpfung auf Konformität zu den PDO-Einträgen überprüft und in die Liste eingetragen.

  Das CANopen-Slave Objekt 0x4161 (Drehzahl-Istwert) ist jetzt mit dem CANopen-Master Objekt 0xA660, Subindex 0x01 verbunden. ProMaster und ProCANopen generieren für dieses CANopen-Master Objekt automatisch eine IEC-Variable (Netzwerkvariable) für das IEC-Projekt. In unserem Fall ist der Name der Netzwerkvariable im IEC-Projekt "di_SpeedActValue". Im IEC-Projekt wird diese Netzwerkvariable gelesen. Der Wert wird über das CANopen-Master Objekt 0xA660 - 0x01 vom CANopen-Slave Objekt 0x4161 gelesen. Siehe ▹Register IEC◃ in Kapitel "4.3.6.2 Konfigurieren der CANopen-Master Kommunikation" ab Seite 81.

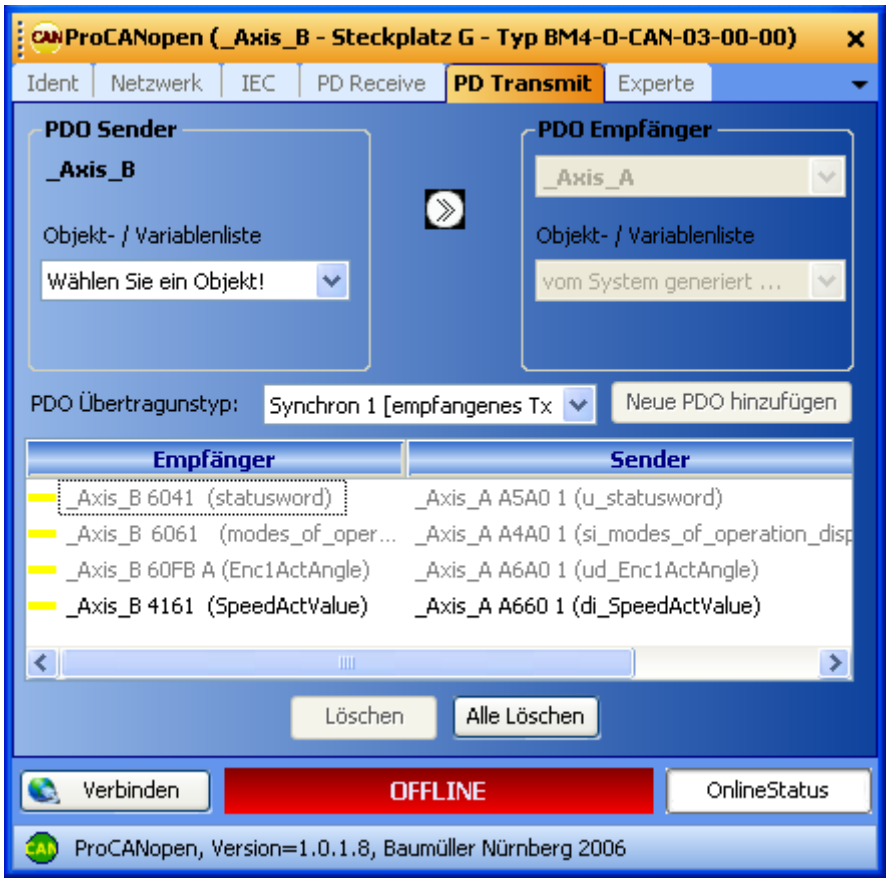


Abbildung 40: ProCANopen - Motion Control CANopen-Slave Kommunikationskonfiguration Register „PD Transmit“ mit Default Motion Control Konfiguration und zusätzlichem Objekt

HINWEIS

ProMaster berücksichtigt automatisch die Grenze von 30 Zeichen für einen Variablennamen im IEC Projekt in ProProg wt III.

Die Namen der Variablen aus der zusätzlichen Konfiguration können später, beim Konfigurieren der PLC, in Kapitel ▹4.3.7.4 CANopen◃ ab Seite 90 geändert werden.

Register Experte

Bei der Verwendung von Motion Control dient das Register "Experte" der Anzeige der im CANopen-Slave verfügbaren Objekte und ihrer Werte.

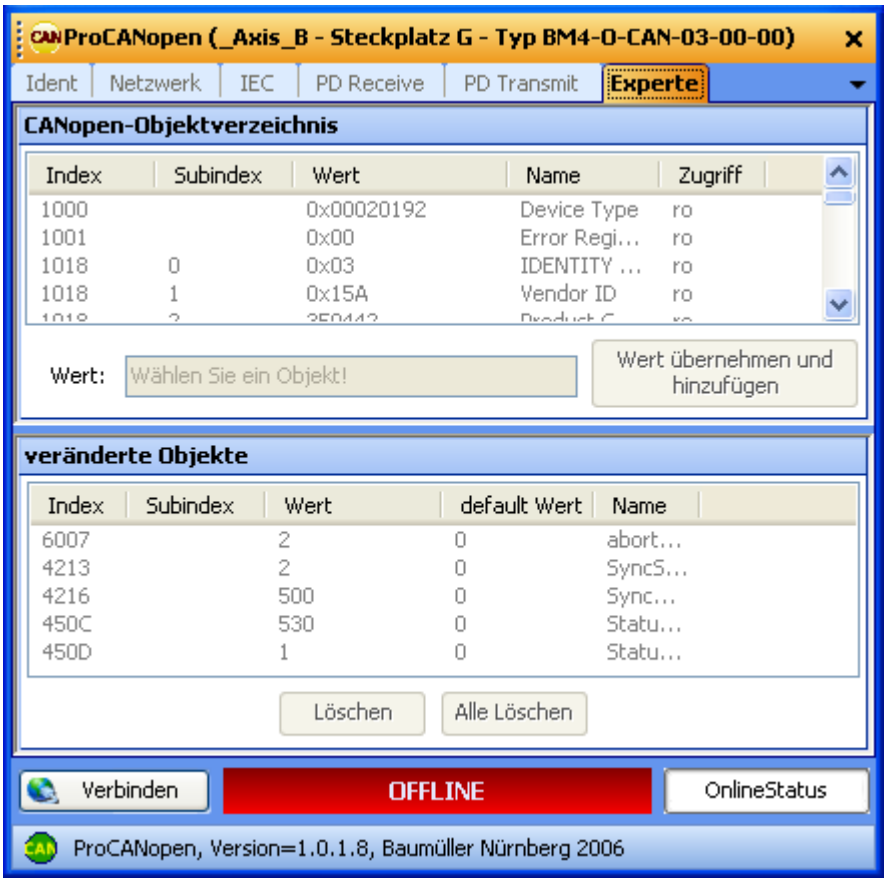


Abbildung 41: ProCANopen - CANopen-Slave Kommunikationskonfiguration Register „Experte“

4.3.6.2 Konfigurieren der CANopen-Master Kommunikation

Öffnen Sie ProCANopen für unseren CANopen-Master in dem Sie auf das Gerät „_Axis_A“ klicken und dann über das Kontextmenü "Konfigurationsdaten (Komponenten)\CANopen Master (Steckplatz G)\Master Bus konfigurieren (ProCANopen)" anwählen.

Der CANopen-Master wurde bereits beim Anlegen der Maschinenkonfiguration per Default so eingestellt, dass die CANopen Einstellungen für Motion Control konfiguriert sind.

Das Fenster für die CANopen-Master Kommunikationskonfiguration wird geöffnet.

Register Ident

Für Motion Control können Sie im Register "Ident" den Steckplatz des Moduls CANopen-Master an der b maXX drive PLC, den Namen dieses Optionsmoduls (Gerätename) in ProMaster sowie für CANopen die Baudrate und die Node-ID einstellen.

Die Änderung von Steckplatz, Gerätename, Baudrate und Node-ID wirkt sich auf das ProMaster Projekt aus.

Weiterhin werden Ihnen die Technologie sowie CANopen spezifische Informationen wie Gerätetyp, CANopen-Profil, Hersteller-ID usw. angezeigt.

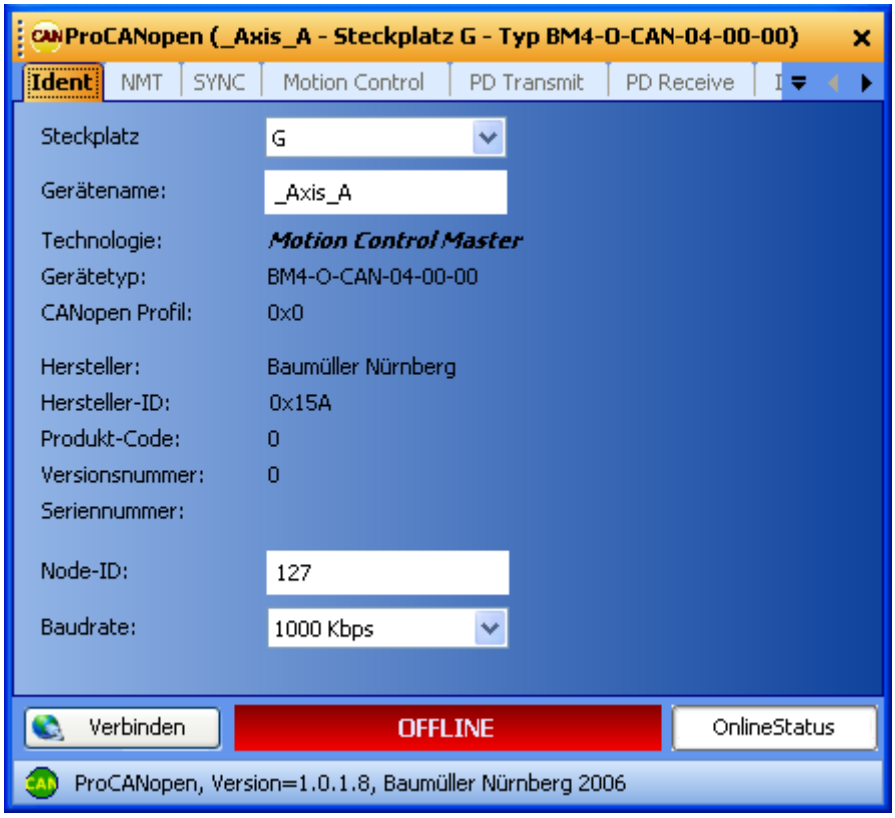


Abbildung 42: ProCANopen - CANopen-Master Kommunikationskonfiguration Register Ident

Register NMT

Bei der Verwendung von Motion Control bleiben die Einstellungen auf dem Register "NMT" unverändert.

In der Edit-Box „Wartezeit der Slaves“ kann eingestellt werden, wie lange der CANopen-Master nach dem Einschalten auf das Bootup-Telegramm der CANopen-Slaves wartet. Der Eintrag 0 ms bedeutet, dass der Master wartet bis alle Slaves ihre Bootup-Nachricht gesendet haben. Der Master wird also kein Timeout melden.

HINWEIS

Es gibt CANopen-Slaves, die erst nach über 30 s nach ihrem Einschalten ihr Bootup-Telegramm senden.

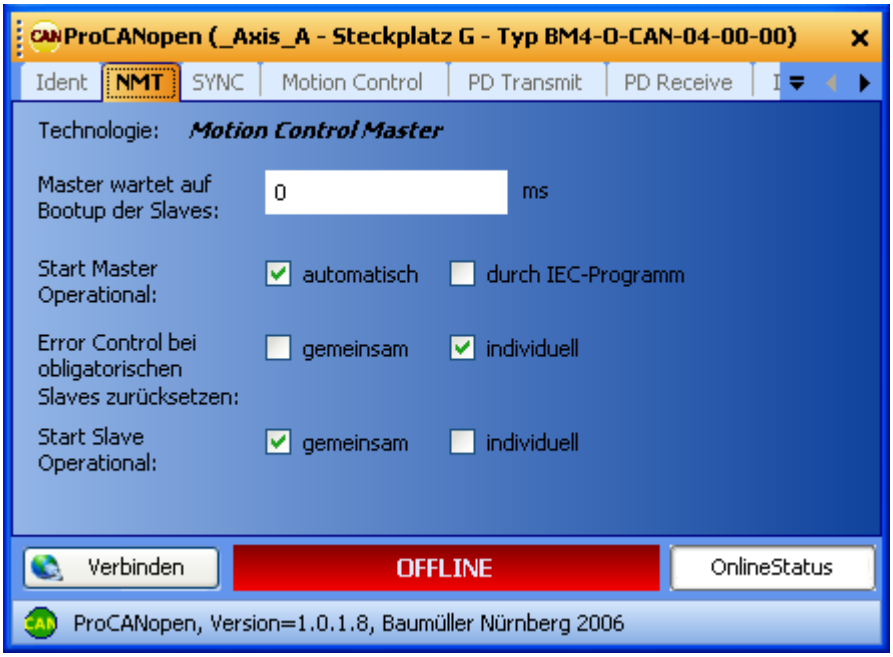


Abbildung 43: ProCANopen - CANopen-Master Kommunikationskonfiguration Register NMT

Register SYNC

Bei der Verwendung von Motion Control können Sie im Register "SYNC" die SYNC-Zykluszeit und das Synchronisiersignal zwischen Optionsmodul CANopen-Master und b maXX drive PLC einstellen.

Die Änderung von der SYNC-Zykluszeit und des Synchronisiersignals wirkt sich auf das ProMaster Projekt aus.

Es muss beachtet werden, dass der CANopen-Master als SYNC-Erzeuger das SYNC-Telegramm auf dem CANopen-Bus sendet. Das Intervall für das Senden wird dem CANopen-Master in unserem Beispiel über das Signal in der Combo Box "Synchronisation mit PLC (Event)" vorgegeben. Für das Modul CANopen-Master können Sie aus den Signalen "SYNC-Signal 1" und "SYNC-Signal 2" wählen. Andere Signale sind nicht erlaubt. Das Signal muss auf der b maXX drive PLC generiert werden. Per Default wird bei Motion Control das Signal "SYNC-Signal 1" verwendet und von der b maXX drive PLC automatisch generiert.

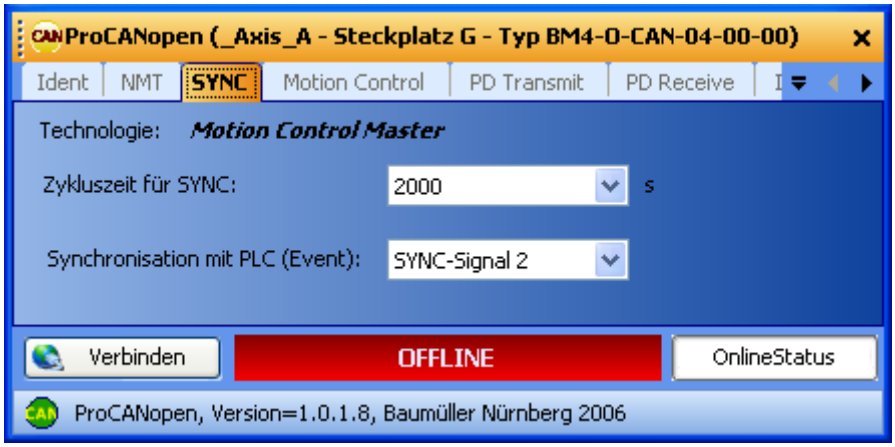


Abbildung 44: ProCANopen - CANopen-Master Kommunikationskonfiguration Register SYNC

Register Motion Control

Im Register „Motion Control“ werden die Einstellungen für Motion Control vorgenommen.

Für die Maschinenkonfiguration können Sie angeben ob Motion Control verwendet wird. Die Auswahl wird Ihnen bei Anwahl des Masters aus dem Projektbaum ermöglicht.

Außerdem können Sie für einzelne Antriebe angeben, dass diese nicht über Motion Control gesteuert werden, sowie z. B. die Namen einzelner Antriebe ändern. Die Auswahl wird Ihnen bei Anwahl des jeweiligen Antriebs im Projektbaum ermöglicht. Die Änderungen wirken sich auf das ProMaster Projekt aus.

HINWEIS

Falls Sie in der Maschinenkonfiguration Antriebe ohne Motion Control verwenden, müssen diese auf jeden Fall konfiguriert werden (siehe hierzu ▹Konfigurieren der CANopen-Slave Kommunikation◃ in Kapitel "4.3 ProMaster, ProCANopen, ProProg wt III und Motion Control" ab Seite 67.

Weiterhin werden Ihnen Motion Control spezifische Informationen, wie Motion Control Zykluszeit, und CANopen spezifische Informationen, wie die Node-ID des jeweiligen Antriebs, angezeigt.

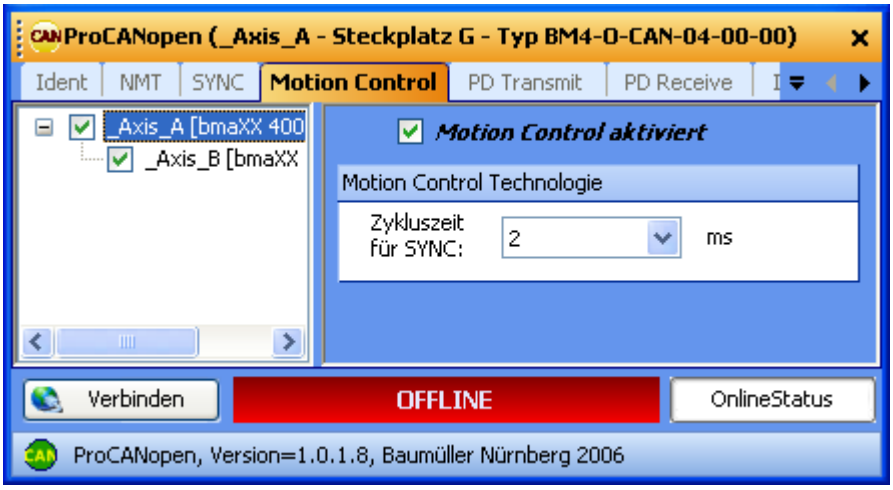


Abbildung 45: ProCANopen - CANopen-Master Kommunikationskonfiguration Register Motion Control

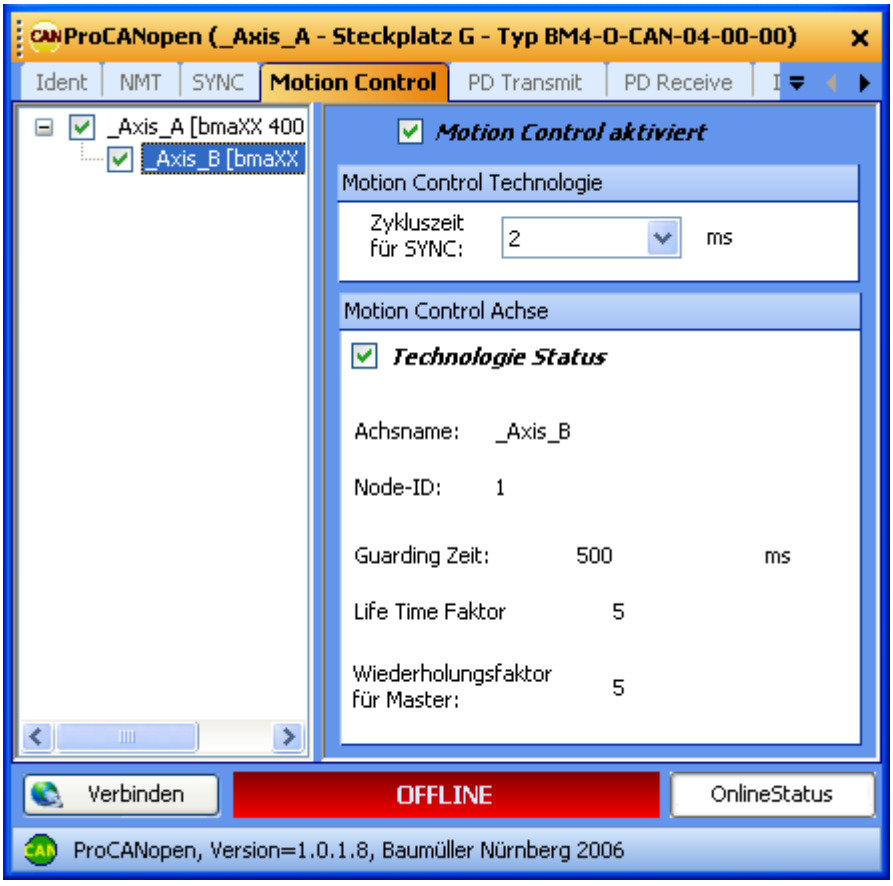


Abbildung 46: ProCANopen - CANopen-Master Kommunikationskonfiguration Register Motion Control

Register PD Transmit und PD Receive

Die Daten auf diesen Seiten dienen der Anzeige.

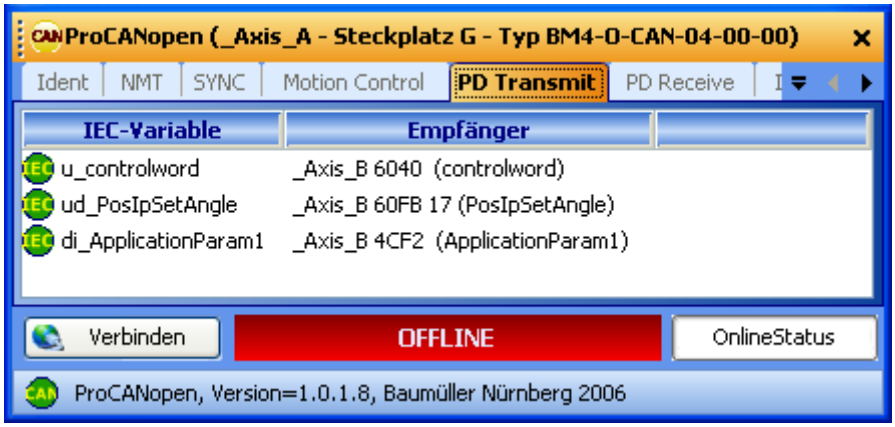


Abbildung 47: ProCANopen - CANopen-Master Kommunikationskonfiguration Register PD Transmit

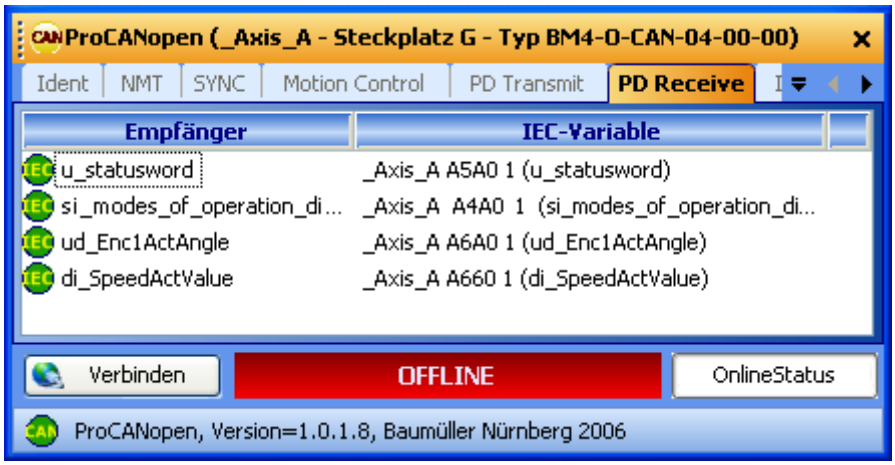


Abbildung 48: ProCANopen - CANopen-Master Kommunikationskonfiguration Register PD Receive

In den Registern "PD Transmit" und "PD Receive" werden die IEC-Variablen (Netzwerkvariablen), jeweils für Senden (PD Transmit) und Empfangen (PD Receive) angezeigt.

Die Zuordnung der Netzwerkvariablen (über die CANopen-Master Objekte) zu den CANopen-Slave Objekten erfolgt bei ProMaster und ProCANopen automatisch.

Die Zuordnung weiterer CANopen-Slave Objekte zu Netzwerkvariablen erfolgt bei der Konfiguration der CANopen-Slave Kommunikation der CANopen-Slaves. Nach dieser Konfiguration der CANopen-Slaves werden die entsprechenden Daten in den Registern "PD Transmit" und "PD Receive" der Konfiguration der CANopen-Master Kommunikation angezeigt.

Zur Konfiguration der CANopen-Kommunikation der CANopen-Slaves siehe Kapitel ▹4.3.6.1 Konfigurieren der CANopen-Slave Kommunikation◃ ab Seite 67.

Register IEC

Bei der Konfiguration der CANopen-Slave Kommunikation erfolgt die Zuordnung der CANopen-Slave Objekte (über CANopen-Master Objekte) zu den Netzwerkvariablen für das IEC-Projekt auf der PLC.

Die im Register IEC angezeigten Variablen werden auch Netzwerkvariablen genannt. Über die Netzwerkvariablen erfolgt das Prozessdaten schreiben und lesen im IEC-Projekt.

Bei der Verwendung von Motion Control sind einige der Zuordnungen per default angelegt.

Im Register „IEC“ werden die Netzwerkvariablen und ihr Datentyp im IEC-Projekt auf der PLC angezeigt.

Die Berechnung der (IEC-) Adresse der Netzwerkvariablen erfolgt erst nach „Liste aktualisieren“ in Kapitel ▹Register Download◃ in Kapitel "4.3.6.2 Konfigurieren der CANopen-Master Kommunikation" ab Seite 82.

Zur Konfiguration der CANopen-Kommunikation der CANopen-Slaves siehe ▹4.3.6.1 Konfigurieren der CANopen-Slave Kommunikation◃ ab Seite 67.

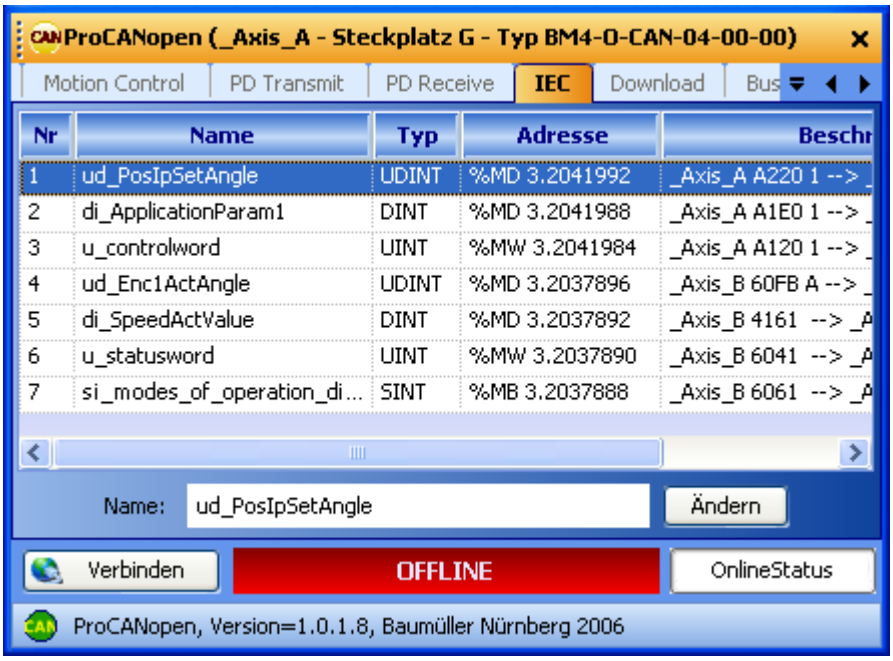


Abbildung 49: ProCANopen - CANopen-Master Kommunikationskonfiguration Register IEC nach Konfiguration der CANopen-Slaves und „Liste aktualisieren“ (Register Download)

Sie können den Namen der Netzwerkvariablen ändern, indem Sie den Netzwerkvariablennamen aus der Spalte „Name“ anwählen, in der Edit-Box "Bezeichnung" den neuen Netzwerknamen editieren und anschließend auf den Button "Ändern" klicken.

HINWEIS

Beachten Sie die Grenze von 30 Zeichen für einen Variablennamen im IEC Projekt in ProProg wt III.

Register Download

Im Register Download werden die Objekte angezeigt, die nach der Konfiguration der CANopen-Kommunikation des CANopen-Masters und des CANopen-Slaves auf den CANopen-Master geladen werden (Download). Solange die Konfiguration nicht abgeschlossen ist, werden im Register Download keine Daten angezeigt, d. h. es können keine Daten auf den CANopen-Master geladen werden.

Betätigen Sie jetzt den Button „Liste aktualisieren“ (rechts unten). Dadurch werden die Konfigurationsdaten (sowohl Master als auch Slave) für den Download auf den CANopen-Master erzeugt.

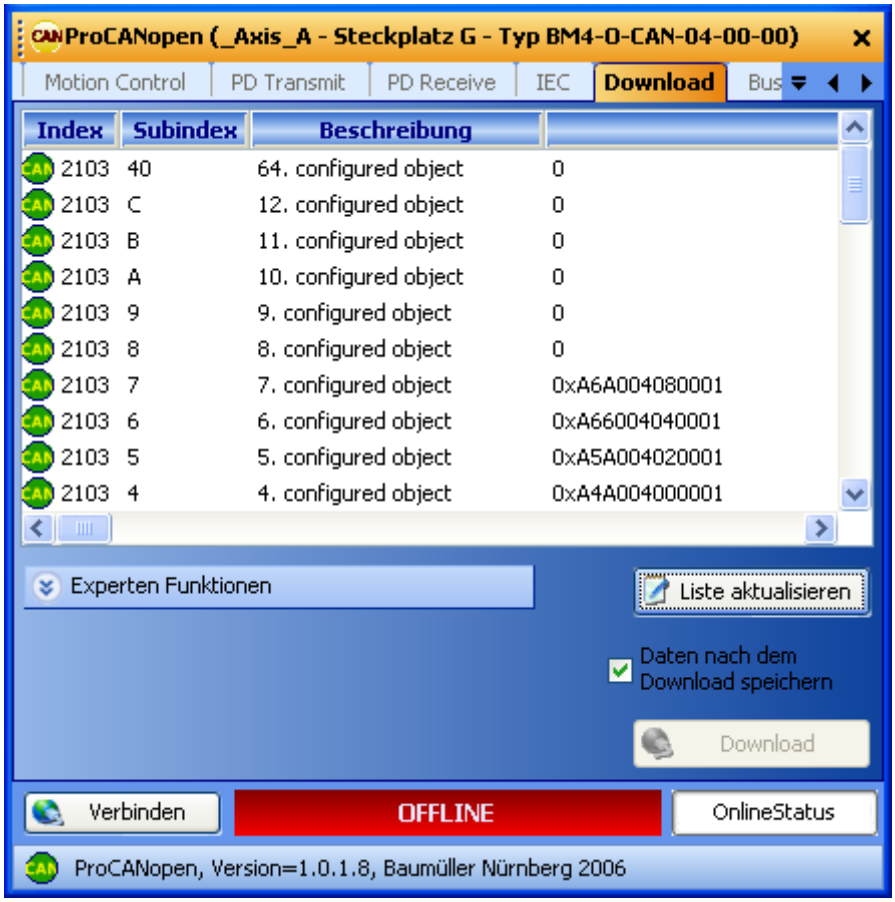


Abbildung 50: ProCANopen - CANopen-Master Kommunikationskonfiguration Register Download

Drücken Sie jetzt den Button „Download“. Die Verbindung von ProMaster zum CANopen-Master (Optionsmodul BM4-O-CAN-04) wird über die b maXX drive PLC (Optionsmodul BM4-O-PLC-01) aufgebaut. Anschließend werden die CANopen Konfigurationsdaten (sowohl Master als auch Slave) auf den CANopen-Master gesendet.

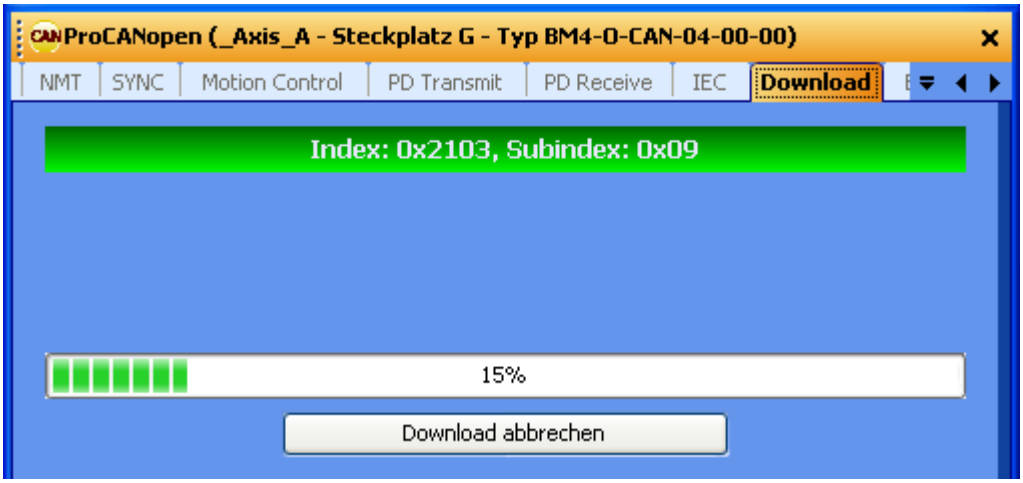


Abbildung 51: ProCANopen - CANopen Kommunikationskonfiguration Download

Falls hier Fehler gemeldet werden, kann es notwendig sein über das Register „Bus-Steuerung“, Button „Verbindung aufbauen“ und Button „Reset“ den CANopen-Master in den CANopen Zustand „Reset“ zu versetzen und anschließend im Register „Download“ nochmal den Download zu starten.

Nachdem der Download beendet ist, wird wieder die Download Objektliste angezeigt.

Jetzt befinden sich auf dem CANopen-Master sowohl die CANopen-Konfigurationsdaten für den CANopen-Master als auch die CANopen-Konfigurationsdaten für den CANopen-Slave, welche der Master beim Start des CANopen-Netzwerks an den Slave sendet.

Register Bus-Steuerung

Im Register „Bus-Steuerung“ können versierte CANopen-Anwender, nach dem Download der CANopen-Konfigurationsdaten, das CANopen-Netzwerk "per Hand" steuern. Dies kann z. B. bei der physikalischen Inbetriebnahme des CANopen-Netzwerks hilfreich sein.

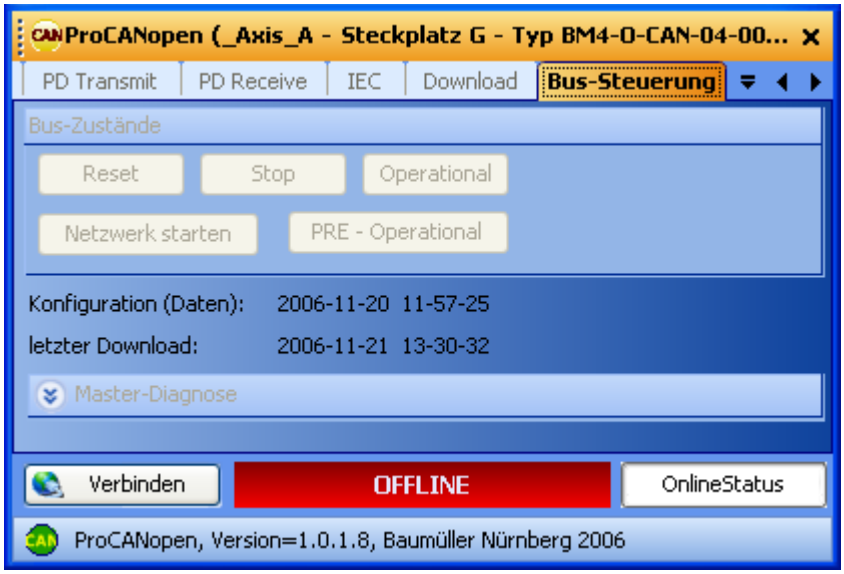


Abbildung 52: ProCANopen - CANopen-Master Kommunikationskonfiguration Register „Bus-Steuerung“

4.3.7 Konfigurieren der PLC

Nach der Konfiguration der Kommunikation des CANopen-Netzwerks und der Erstellung des IEC Projekts werden nun die Daten für die b maXX drive PLC konfiguriert.

Dabei werden

- das IEC Projekt in das ProMaster Projekt eingebunden
- die Kurvendatensätze ausgewählt.

Außerdem können

- mit dem Kurvenscheiben Editor ProCAM eigene Kurvenscheibendaten erstellt werden

Anschließend klicken Sie im Fenster "ProPLC" auf den Button "Gesamtes IEC-Projekt aktualisieren". Die Daten für die Maschinenkonfiguration, sowohl für das CANopen-Netzwerk, als auch für die b maXX controller PLC werden dann erzeugt.

4.3.7.1 IEC

In diesem Abschnitt erläutern wir, wie ein IEC Projekt mit ProMaster verbunden wird.

Öffnen Sie im ProMaster Projekt in der Netzwerkansicht für unseren CANopen-Master das Fenster "PLC Konfiguration" in dem Sie auf das Gerät „_Axis_A“ klicken und dann über das Kontextmenü "Konfigurationsdaten (Komponenten)\PLC (Steckplatz H)\PLC - Konfiguration (ProPLC)" anwählen und dort das Register "IEC" auswählen.

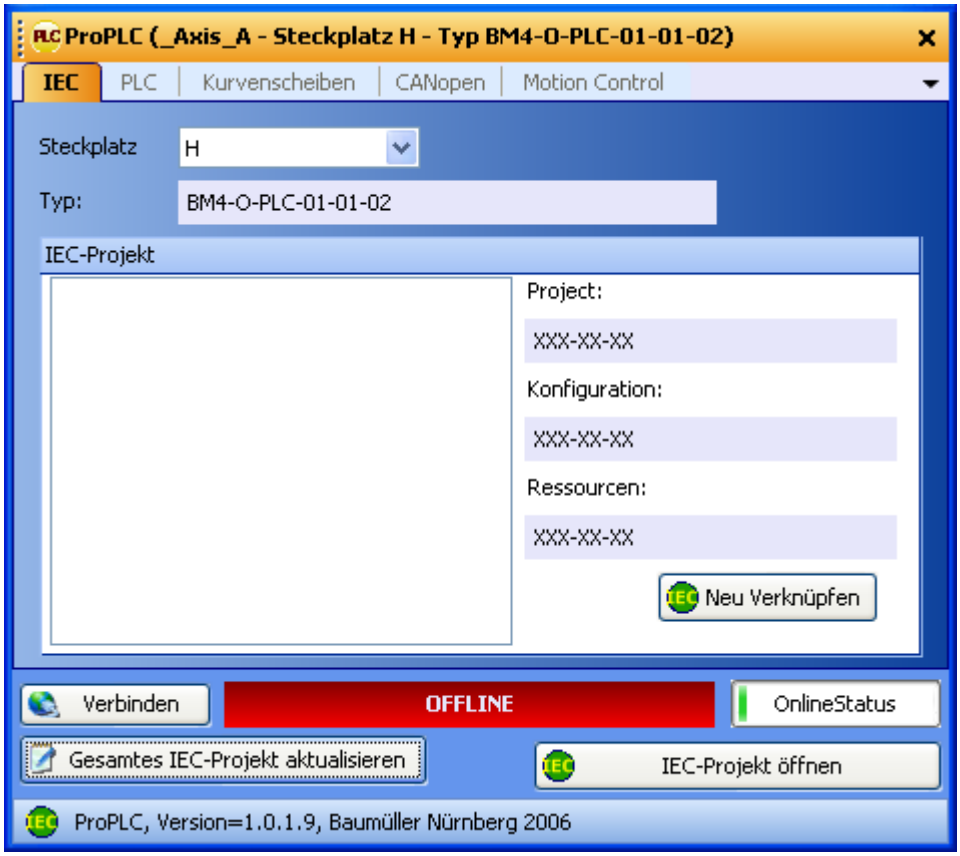


Abbildung 53: ProPLC - Verbinden mit Beispiel IEC Projekt

HINWEIS

Mit ProMaster können nur ProProg wt III Projekte geöffnet werden. Falls Sie ein bestehendes PROPROG wt II Projekt verwenden wollen, müssen Sie dieses zuerst mit ProProg wt III öffnen (und dabei konvertieren) und können es danach in ProMaster öffnen und verwenden.

Beachten Sie, dass dabei Ihre PROPROG wt II-Bibliotheken ebenfalls konvertiert werden!

Klicken Sie auf den Button "Neu Verknüpfen" und wählen Sie unser Beispiel IEC Projekt "Example_BM4_O_CAN04_MA_2.mwt", welches wir in Kapitel ▹4.3.4 Anlegen eines IEC Projekts mit ProProg wt III◃ ab Seite 59 angelegt haben, aus.

Im Register "IEC" wird das eben ausgewählte IEC Projekt angezeigt.

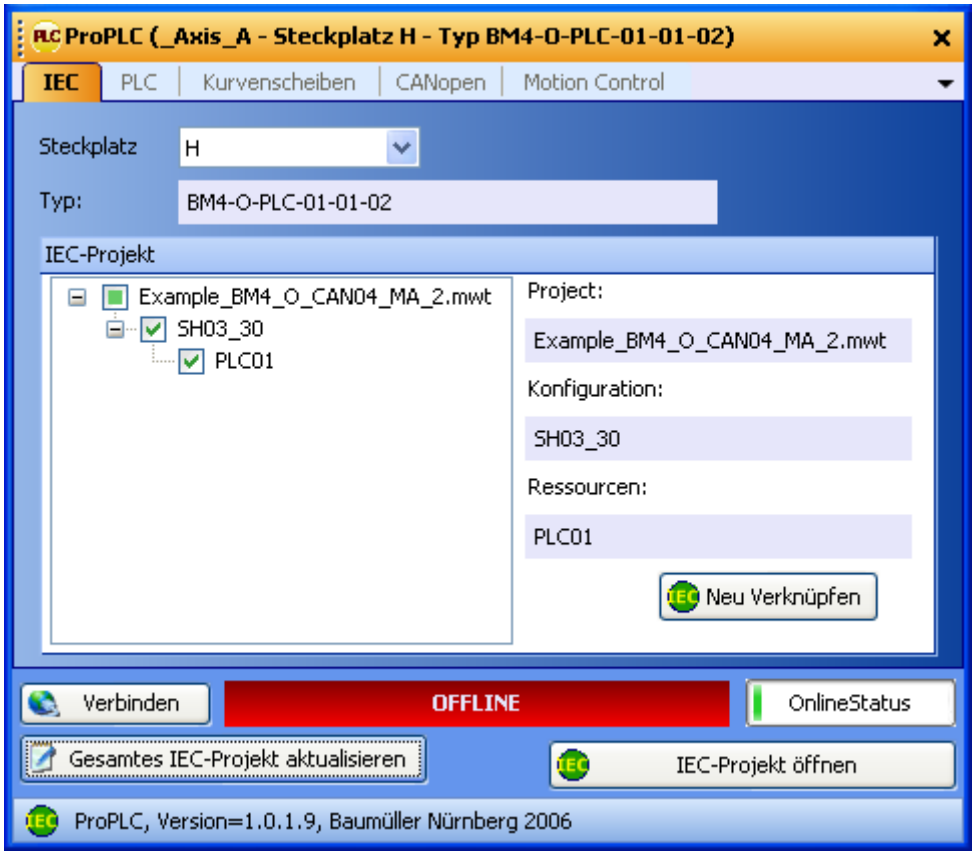


Abbildung 54: ProPLC - Register IEC

Falls Sie eine vorhandene Applikation (oder eine Vorlage) aus ProProg wt III mit dem ProMaster Projekt verbinden, kann es sein, dass die Namen der Geräte in ProMaster und im IEC Projekt nicht gleich sind. In diesem Fall siehe Kapitel ▹4.3.9 Programmieren des IEC Projekts◃ ab Seite 94.

4.3.7.2 PLC

Nach dem Download der Daten auf CANopen-Master und b maXX drive PLC, siehe ▹Download der Daten auf den CANopen-Master und auf die b maXX drive PLC◃ ab Seite 93, kann im Register PLC das IEC Projekt auf der b maXX drive PLC aktiviert und gestartet werden (Bereich "PLC Status").

Weiterhin werden im Bereich "PLC Info" Daten der b maXX drive PLC wie Betriebssystem-Version, Firmware-Version, IEC Projekt und Boot Projekt angezeigt. Im Bereich "Flash Directory" werden (nach "Verbinden" und "Aktualisieren") Informationen zu den Dateien im Flash-Speicher der b maXX drive PLC angezeigt, z. B. zu den Kurvendatensätzen (siehe ▹4.3.7.3. Kurvenscheiben◃)

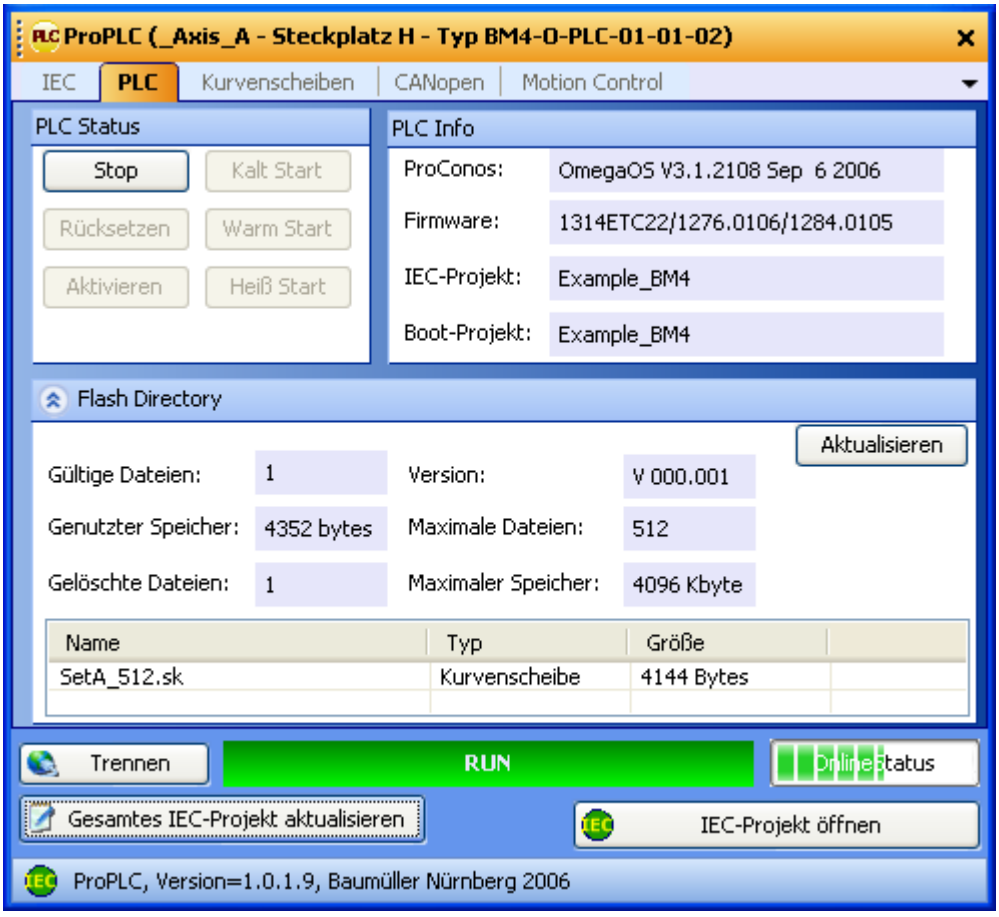


Abbildung 55: ProPLC - Register PLC nach Download der Kurvenscheibe (Register Kurvenscheibe)

4.3.7.3 Kurvenscheiben

In diesem Abschnitt erläutern wir, wie ein Kurvenscheibendatensatz erstellt und mit ProMaster verbunden wird.

Öffnen Sie im ProMaster Projekt in der Netzwerkansicht für unseren CANopen-Master das Fenster "PLC Konfiguration" in dem Sie auf das Gerät „_Axis_A“ klicken und dann über das Kontextmenü "Konfigurationsdaten (Komponenten)\PLC (Steckplatz H)\PLC - Konfiguration (ProPLC)" anwählen und dort das Register "Kurvenscheiben" auswählen.

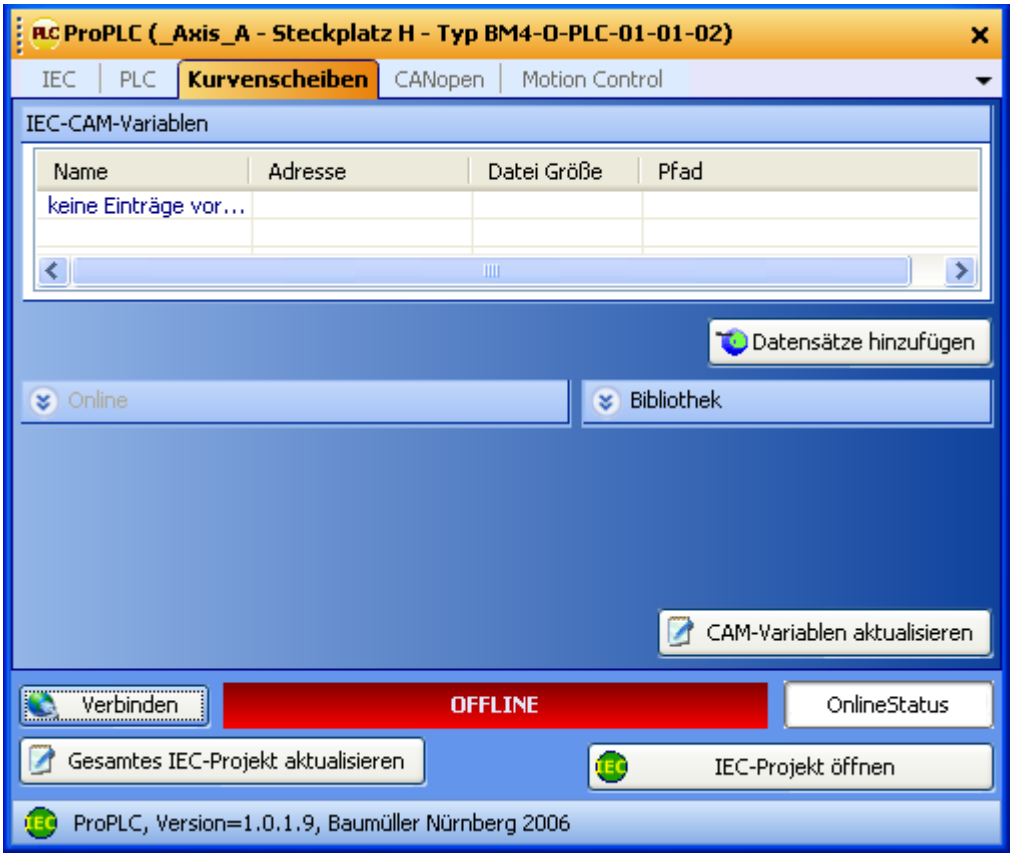


Abbildung 56: ProPLC - Register „Kurvenscheiben“ ohne Kurvendatensatz

Drücken Sie den Button "Hinzufügen". Das Fenster "ProMaster.NET - Kurvenscheiben" wird geöffnet.

Drücken Sie im Bereich "Festplatte" den Button "Hinzufügen" und wählen Sie auf Ihrer Festplatte den Pfad aus, in dem Ihre Kurvendaten stehen.

HINWEIS

Beispielkurvendaten finden Sie im Installationsverzeichnis von ProCAM im Unterordner "examples" (z. B. C:\...\Baumueller\ProCam 2\examples.

Falls Sie eigen Kurven generieren wollen, klicken Sie auf den Button "ProCAM" und öffnen damit den Kurvenscheiben Editor. Dort können Sie Ihre Kurven editieren.

Stellen Sie jetzt per Drag&Drop Ihren Kurvendatensatz (im Bereich "*.sk Dateien") aus den verschiedenen Kurvenscheibendaten (im Bereich "Festplatte") zusammen.

Für unser Beispiel wählen wir die Kurvenscheibendaten "ExampleGerade_MC.kbin" und "ExamplePendelP5_MC.kbin", die in den Kurvenscheibendatensatz „SetA_512“ zusammengefasst werden.

HINWEIS

Sie können sich die Kurvenscheibendaten (im jeweiligen Bereich) über den Button "Vorschau" grafisch darstellen lassen. Dies erleichtert die Kurvenauswahl.

Klicken Sie anschließend auf den Button "Speichern". Dadurch wird der Kurvendatensatz erstellt und ProMaster zur Verfügung gestellt.

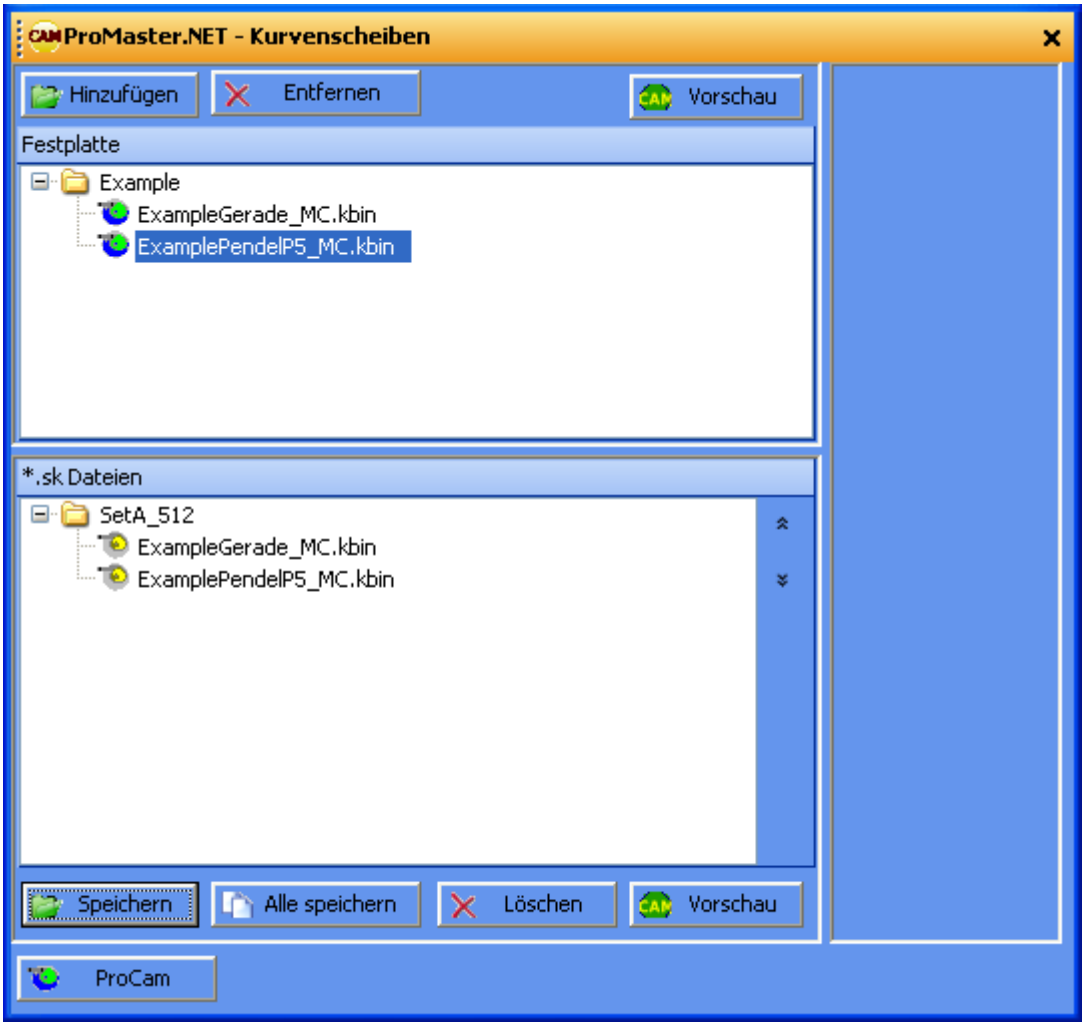


Abbildung 57: ProMaster (- PLC Konfiguration - Register Kurvenscheiben -) Kurvendatensatz erstellen

Schließen Sie jetzt das Fenster "ProMaster.NET - Kurvenscheiben", indem Sie oben rechts auf "x" klicken.

Im Register "Kurvenscheiben" sehen Sie jetzt den Namen der Variablen des Kurvenscheibendatensatzes im IEC Projekt, ihre Adresse im IEC Projekt und den Dateinamen des Kurvenscheibendatensatzes auf der Festplatte.

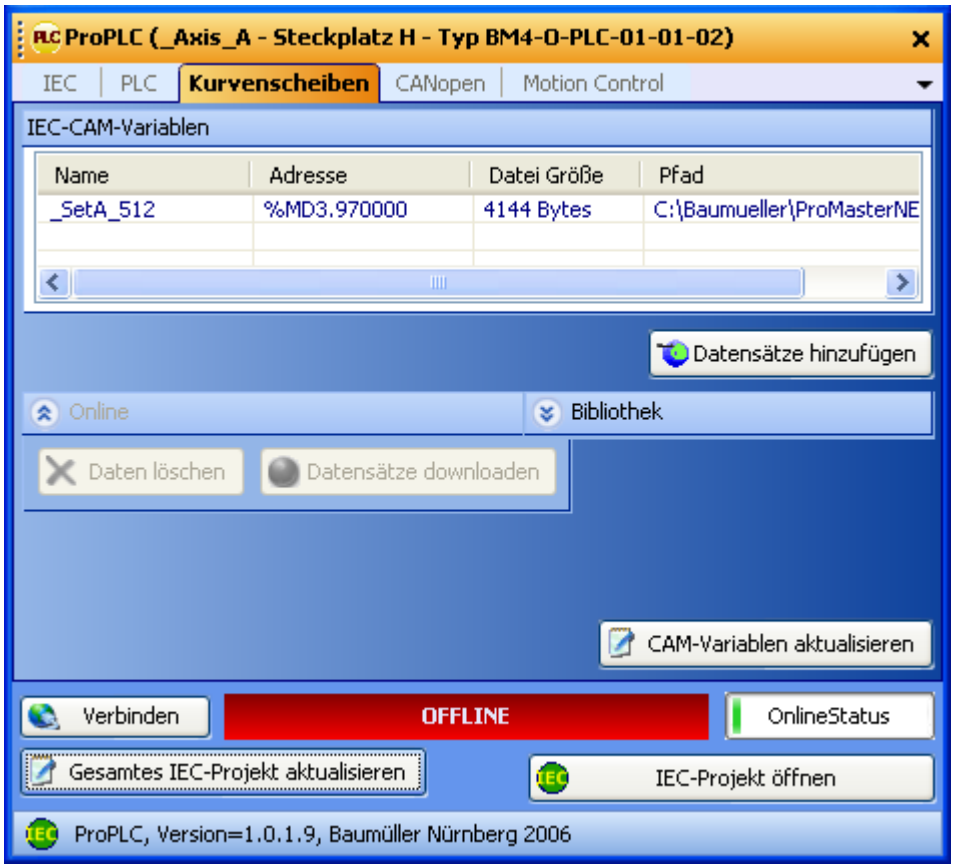


Abbildung 58: ProPLC - Register „Kurvenscheiben“ mit Kurvendatensatz

Falls Sie eine vorhandene Applikation (oder eine Vorlage) aus ProProg wt III mit dem ProMaster Projekt verbinden, kann es sein, dass der Kurvendatensatzname in ProMaster und im IEC Projekt nicht gleich sind. In diesem Fall siehe Kapitel ▹4.3.9 Programmieren des IEC Projekts◃ ab Seite 94.

Sie können vom Register "Kurvenscheiben" aus, über den Button "Flash löschen", den Flash-Speicher auf der b maXX drive PLC löschen. Dies kann notwendig sein, falls durch diverse Downloads (sowohl Kurvenscheibendatensätze als auch IEC Projekte) der Flash-Speicher "voll" ist.

Weiterhin können Sie die Kurvenscheibendatensätze vom Register "Kurvenscheiben" aus, über den Button "Download", auf die b maXX drive PLC einzeln downloaden.

Laden Sie jetzt die Kurvenscheibendatensätze über den Button „Datensätze downloaden“ auf die b maXX drive PLC.

4.3.7.4 CANopen

Im Register „CANopen“ werden der Name, die Adresse und der Kommentar der Netzwerkvariablen der CANopen-Slaves im IEC Projekt dargestellt.

Bei der Verwendung von Motion Control erfolgt die Kommunikation über die Achsvariable. ProMaster legt nach der Betätigung des Buttons „gesamtes IEC-Projekt aktualisieren“ siehe Kapitel ▹4.3.8 Download der Daten auf den CANopen-Master und auf die b maXX drive PLC◃ ab Seite 93 im IEC Projekt die Achsvariable mit dem Gerätenamen des CANopen-Slave neu an. Zusätzlich legt ProMaster im IEC Projekt Netzwerkvariablen an.

Diese unterscheiden sich in die Standard Netzwerkvariablen für Motion Control und, sofern zusätzliche Objekte bei der CANopen-Slave Konfiguration angelegt wurden, in zusätzliche Netzwerkvariablen für CANopen.

Im Register "CANopen" werden Ihnen diese Netzwerkvariablen, sowohl die Standard Netzwerkvariablen für Motion Control als auch die zusätzlichen Netzwerkvariablen für CANopen, angezeigt.

Hier ist unbedingt folgendes zu beachten:

Die Standard Netzwerkvariablen für Motion Control dürfen vom Anwender gelesen, aber nicht geschrieben werden.

Dies sind die Standard Motion Control Netzwerkvariablen für die Sollwerte:

"u_controlword.."	für das Steuerwort der Achse
"ud_PosIpSetAngel.."	für den Gleichlauf Lage Winkel Sollwert der Achse

Dies sind die Standard Motion Control Netzwerkvariablen für die Istwerte:

"u_statusword.."	für das Statuswort der Achse
"ud_Enc1ActAngle.."	für den Gleichlauf Lage Winkel Istwert der Achse
"si_modes_of_operation_display.."	für die Betriebsart der Achse

HINWEIS

Die Standard Netzwerkvariablen für Motion Control dürfen vom Anwender gelesen, aber nicht geschrieben werden.

HINWEIS

Die Netzwerkvariablen müssen in der Motion Control Event-Task gelesen und geschrieben werden.

Ausnahme: Der PDO Übertragungstyp der Netzwerkvariablen ist asynchron eingestellt.

Die Nummer hinter den Netzwerkvariablennamen ist eine interne Nummer um gleich lautende, automatisch generierte Netzwerkvariablennamen zu unterscheiden.

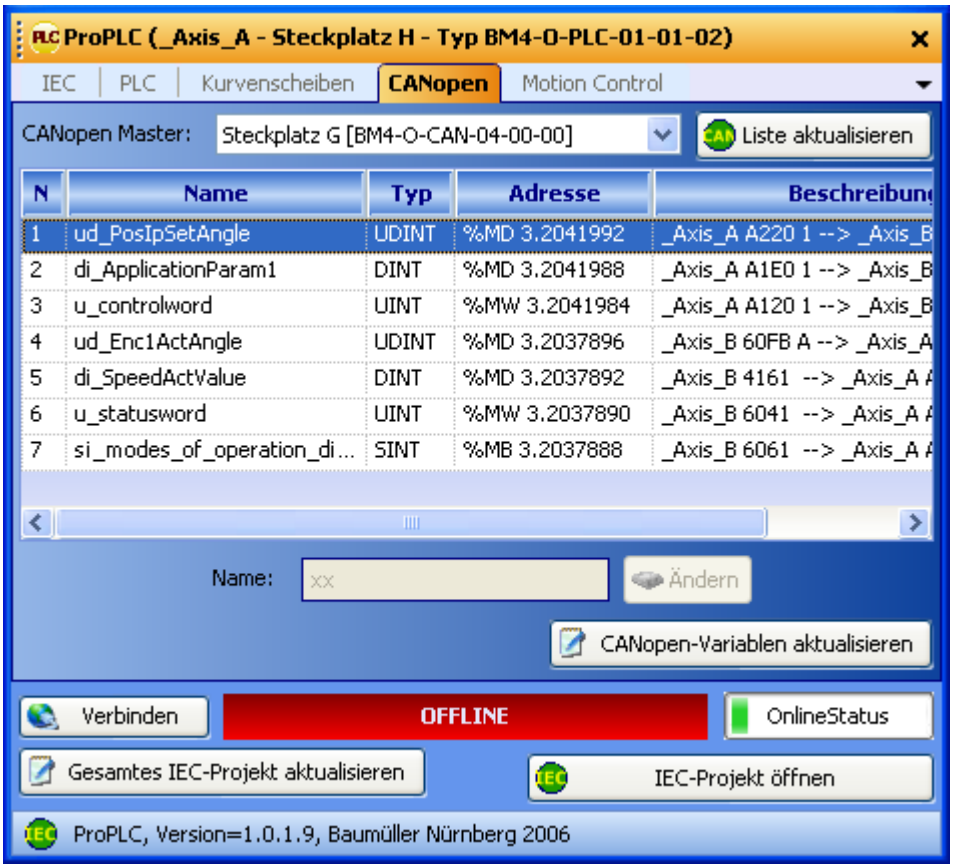


Abbildung 59: ProPLC - Register CANopen

Falls Sie bei der Konfiguration der CANopen-Slave Kommunikation in den Abschnitten ▹Register PD Receive◃ ab Seite 70 und ▹Register PD Transmit◃ ab Seite 72 die dort vorgeschlagenen zusätzlichen Netzwerkvariablen (und deren Verknüpfung zu CANopen-Slaves) angelegt haben, sehen Sie die zusätzlichen Netzwerkvariablen.

Die zusätzliche Netzwerkvariable für die Sollwerte ist:

"di_ApplicationParam1.." für den Applikationsparameter 1 der Achse

Die zusätzliche Netzwerkvariable für die Istwerte ist:

"di_SpeedActValue.." für den Drehzahl-Istwert der Achse

Beachten Sie bei der Änderung von Netzwerkvariablennamen, dass jeder Variablenname nur einmal im IEC Projekt vergeben sein kann. Dies ist insbesondere bei der Verwendung mehrerer Optionsmodule CANopen-Master an der b maXX drive PLC zu beachten.

HINWEIS

Jeder Variablenname darf nur einmal im IEC Projekt vergeben werden.

4.3.7.5 Motion Control

Im Register "Motion Control" werden zukünftig die allgemeinen Einstellungen für Motion Control konfiguriert.

Wenn Sie die Default Motion Control Einstellungen verwenden wollen, brauchen Sie hier keine Einstellungen vornehmen.

4.3.8 Download der Daten auf den CANopen-Master und auf die b maXX drive PLC

Der Download der Daten für den CANopen-Master erfolgt zur Zeit über ▹Register Download◃ in Kapitel "4.3.6.2 Konfigurieren der CANopen-Master Kommunikation" ab Seite 82.

Der Download der Kurvendatensätze für die b maXX drive PLC erfolgt zur Zeit über Register „Kurvenscheiben“ der PLC-Konfiguration. Siehe hierzu Kapitel ▹4.3.7.3 Kurvenscheiben◃ ab Seite 87.

Der Download des IEC-Projektes für die b maXX drive PLC erfolgt zur Zeit wie gewohnt über ProProg wt III.

Klicken Sie jetzt im Fenster "PLC Konfiguration" auf den Button "Gesamtes IEC-Projekt aktualisieren". Die Daten für die b maXX drive PLC werden jetzt erzeugt. Dieser Vorgang kann etwas länger dauern, da unter anderem ProProg wt III geöffnet wird und im IEC-Projekt die konfigurierten IEC-Variablen in das globale Variablenarbeitsblatt geschrieben werden.

Anschließend können sie das IEC-Projekt kompilieren indem Sie die entsprechende Abfrage mit „Ja“ bestätigen.

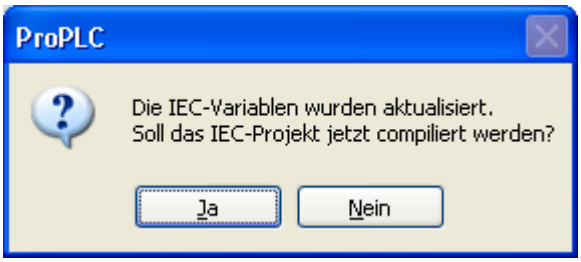


Abbildung 60: ProMaster - aktualisiertes IEC-Projekt kompilieren

Alternativ können Sie das IEC-Projekt auch in ProProg wt III über das ProProg wt III Menü „Code\Projekt neu erzeugen“ kompilieren.

Führen Sie jetzt den Download des IEC-Projektes auf die b maXX drive PLC durch (ProProg wt III - Menü „Online\Projektkontrolle...“ → „Senden“ → Bootprojekt „Senden“).

Führen Sie jetzt an der b maXX controller PLC einen Reset aus und schalten Sie danach die PLC in den Zustand „RUN“ (Alternativ können Sie auch das gesamte CANopen-Netzwerk aus- und wieder einschalten). Jetzt können Sie über das IEC-Projekt in ProProg wt III die lokale Achse _Axis_A und die CANopen-Slave Achse _Axis_B steuern.

4.3.9 Programmieren des IEC Projekts

4.3.9.1 Allgemeines

Wie Sie eine Motion Control Applikation im IEC Projekt in ProProg wt III programmieren können entnehmen Sie bitte dem Applikationshandbuch Motion Control und dem Applikationshandbuch ProProg wt III.

Über das Kontextmenü auf _Axis_A "Konfigurationsdaten (Komponenten)\PLC (Steckplatz H)\IEC - Programmierung (ProProg wt III)" wird ProProg wt III mit unserem IEC Projekt "Example_BM4_O_CAN04_MA_2.mwt" geöffnet. Sie können das IEC Projekt wie gewohnt bearbeiten.

ProMaster hat

- die Motion Control Achsvariable (in den Abschnitt MC_Axis_Variables)
- den Kurvenscheibendatensatz (in den Abschnitt MC_CamDataSet)
- die Netzwerkvariablen (in den Abschnitt CANopenVariables)
  (Standard Netzwerkvariablen für Motion Control und zusätzliche Netzwerkvariablen für CANopen; siehe ▹4.3.9.2 Datenaustausch◃ ab Seite 96)
- die lokalen Achsvariablen (in den Abschnitt LocalAxisVariables)

in das globale Variablenarbeitsblatt "Global_Variables" geschrieben.

Global_Variables:SH03_30.PLC01

Name	Typ	Verwendung	Beschreibung	Adresse
Global_Variables				
MyApplication				
_MyMaster	AXI...	VAR_GLO...	Not from ProMaster	
MC_AxisVariables				
_Axis_A	AXI...	VAR_GLO...	1. Achse	%MD3.451500
_Axis_B	AXI...	VAR_GLO...	2. Achse	%MD3.457300
MC_MasterRef				
MC_SystemReserved				
MC_AxisRef				
MC_BaciInitRef				
MC_AxisPDORef				
MC_TriggerRef				
MC_IRPRef				
MC_CamDataSet				
s_File_SetA_512	FileS...	VAR_GLO...		%MD3.970000
_SetA_512	MC_...	VAR_GLO...		%MD3.970000
CANopen Variables				
ud_PosIpSetAngle	UDINT	VAR_GLO...	_Axis_A A220 1 --> _Axis_B 60FB 17	%MD 3.2041992
di_ApplicationParam1	DINT	VAR_GLO...	_Axis_A A1E0 1 --> _Axis_B 4CF2	%MD 3.2041988
u_controlword	UINT	VAR_GLO...	_Axis_A A120 1 --> _Axis_B 6040	%MW 3.2041984
ud_Enc1ActAngle	UDINT	VAR_GLO...	_Axis_B 60FB A --> _Axis_A A6A0 1	%MD 3.2037896
di_SpeedActValue	DINT	VAR_GLO...	_Axis_B 4161 --> _Axis_A A660 1	%MD 3.2037892
u_statusword	UINT	VAR_GLO...	_Axis_B 6041 --> _Axis_A A5A0 1	%MW 3.2037890
si_modes_of_operation_di	SINT	VAR_GLO...	_Axis_B 6061 --> _Axis_A A4A0 1	%MB 3.2037888
LocalAxisVariables				
w_Controlword	WORD	VAR_GLO...	BM_w_Controlword [P300]	%MW3.466800
ud_PosIpSetAngle1	UDINT	VAR_GLO...	BM_ud_PosIpSetAngle [P370]	%MD3.466804
di_ApplicationParam11	DINT	VAR_GLO...	BM_di_ApplicationParam1 [P3314]	%MD3.466808
w_Statusword	WORD	VAR_GLO...	BM_w_Statusword [P301]	%MW3.466828
i_OperationModeAct	INT	VAR_GLO...	BM_i_OperationModeAct [P304]	%MW3.466832
ud_Enc1ActAngle1	UDINT	VAR_GLO...	BM_ud_Enc1ActAngle [P391]	%MD3.466836
di_SpeedActValue1	DINT	VAR_GLO...	BM_di_SpeedActValue [P353]	%MD3.466840

Abbildung 61: ProProg wt III - Globales Variablenarbeitsblatt mit den Daten von ProMaster

Besonders muss folgendes beachtet werden:

Der Gerätename der Geräte am CANopen-Bus in ProMaster ist gleichzeitig der Name der Achsen im IEC Projekt in ProProg wt III. D. h. falls Sie eine vorhanden Applikation (oder eine Vorlage) aus ProProg wt III mit dem ProMaster Projekt verbunden haben, kann es sein, dass die Namen der Geräte in ProMaster und im IEC Projekt nicht gleich sind. In diesem Fall gibt es zwei Lösungsmöglichkeiten:

1 ProMaster ändert in ProProg wt III im globalen Variablenarbeitsblatt "Global_Variables", im Abschnitt "MC_AxisVariables", die Achsnamen auf die Gerätenamen in ProMaster und der Anwender ändert (über die ProProg wt III Funktion "Globales Ersetzen") die Achsnamen in den POEs auf die Gerätenamen in ProMaster.

2 Sie ändern die Gerätenamen in ProMaster auf die Achsnamen in ProProg wt III.

In unserem Beispiel haben wir in ProMaster der lokalen Achse den Gerätenamen _Axis_A und dem CANopen-Slave den Gerätenamen _Axis_B gegeben. Dies sind auch die Achsname aus unserer Motion Control Vorlage, die wir für unser IEC-Projekt verwendet haben.

Ähnliches gilt auch für den Kurvendatensatznamen im IEC Projekt in ProProg wt III. D. h. falls Sie eine vorhandene Applikation (oder eine Vorlage) aus ProProg wt III mit dem ProMaster Projekt verbinden, kann es sein, dass Kurvendatensatzname in ProMaster und im IEC Projekt nicht gleich sind. In diesem Fall gilt folgendes:

ProMaster ändert in ProProg wt III im globalen Variablenarbeitsblatt "Global_Variables", im Abschnitt "MC_CamDataSet", die Kurvendatensatznamen auf die Kurvendatensatznamen in ProMaster und der Anwender ändert (über die ProProg wt III Funktion "Globales Ersetzen") die Kurvendatensatznamen in den POEs auf die Kurvendatensatznamen in ProMaster.

4.3.9.2 Datenaustausch

Für den Datenaustausch zwischen

- den Funktionsbausteinen im IEC Projekt auf der b maXX drive PLC (BM4-O-PLC-0x) und der lokalen Achse
- den Funktionsbausteinen im IEC Projekt auf der b maXX drive PLC (BM4-O-PLC-0x), dem Optionsmodul CANopen-Master (BM4-O-ETH-02 / BM4-O-CAN-04) und (über den CANopen-Feldbus mit) dem CANopen-Slave

wird jeweils eine globale Variable benötigt. Diese Variablen werden auch Achsvariablen genannt und haben im IEC-Projekt einen Achsnamen.

Unsere Achsvariable für die lokale Achse hat den Achsnamen _Axis_A.

Unsere Achsvariable für die CANopen-Slave Achse hat den Achsnamen _Axis_B.

Wir haben in ProMaster unserer lokalen Achse den Gerätenamen _Axis_A und unserem CANopen-Slave den Gerätenamen _Axis_B gegeben, deshalb brauchen wir keine Anpassungen vornehmen.

Das Gleiche gilt für die Variable für die Kurvendaten, die den Kurvendatensatznamen _SetA_512 in IEC Projekt und ProMaster hat.

Die Achsvariable und die Kurvendatenvariable werden im IEC Projekt an den Motion Control Funktionsbausteinen angeschlossen. Mit den Motion Control Funktionsbausteinen wird die Maschinenfunktion programmiert.

Hinweise zur Verwendung der Motion Control Funktionsbausteine finden Sie im Applikationshandbuch Motion Control.

Bei der Verwendung von Motion Control erfolgt die Kommunikation über die Achsvariable. ProMaster legt im IEC Projekt Netzwerkvariablen an. Diese unterscheiden sich in die Standard Netzwerkvariablen für Motion Control und, sofern zusätzliche Objekte bei der CANopen-Slave Konfiguration angelegt wurden, in zusätzliche Netzwerkvariablen für CANopen.

Hier ist unbedingt folgendes zu beachten:

Die Standard Netzwerkvariablen für Motion Control dürfen vom Anwender gelesen, aber nicht geschrieben werden.

Dies sind die Standard Motion Control Netzwerkvariablen für die Sollwerte:

"u_controlword.."	für das Steuerwort der Achse

"ud_PosIpSetAngel.."	für den Gleichlauf Lage Winkel Sollwert der Achse

Dies sind die Standard Motion Control Netzwerkvariablen für die Istwerte:

"u_statusword.."	für das Statuswort der Achse
"ud_Enc1ActAngle.."	für den Gleichlauf Lage Winkel Istwert der Achse
"si_modes_of_operation_display.."	für die Betriebsart der Achse

Die zusätzliche Netzwerkvariablen für CANopen werden vom Anwender geschrieben (Sollwerte) und gelesen (Istwerte).

Falls Sie bei der Konfiguration der CANopen-Slave Kommunikation in den Abschnitten ▹Register PD Receive◃ ab Seite 70 und ▹Register PD Transmit◃ ab Seite 72 die dort vorgeschlagenen zusätzlichen Netzwerkvariablen (und deren Verknüpfung zu CANopen-Slaves) angelegt haben, ist die zusätzliche Netzwerkvariable für die Sollwerte:

"di_ApplicationParam1.."	für den Applikationsparameter 1 der Achse

Die zusätzliche Netzwerkvariable für die Istwerte ist:

"di_SpeedActValue.."	für den Drehzahl-Istwert der Achse

HINWEIS

Die Standard Netzwerkvariablen für Motion Control dürfen vom Anwender gelesen, aber nicht geschrieben werden.

Die zusätzliche Netzwerkvariablen für CANopen werden vom Anwender geschrieben (Sollwerte) und gelesen (Istwerte).

HINWEIS

Sie verwenden das Optionsmodul CANopen-Master zusammen mit Motion Control.

Keine Motion Control Achse (aus CANopen Sicht ist das ein Netzwerkknoten) darf über die Funktionsbausteine aus der Bibliothek CANopen_PLC01_30bd00 (oder höher) unter ProProg wt III zusammen mit Motion Control bedient werden.

Ein Netzwerkknoten wird also entweder über die Funktionsbausteine aus der Bibliothek CANopen_PLC01_30bd00 (oder höher) unter ProProg wt III oder von Motion Control bedient.

HINWEIS

Bei der Verwendung von Motion Control wird die Initialisierung des Moduls CANopen-Master von Motion Control übernommen. Der Funktionsbaustein COM405_KERNEL_INIT aus der Bibliothek CANopen_PLC01_30bd00 (oder höher) unter ProProg wt III darf nicht verwendet werden.

HINWEIS

Bei der Verwendung von Motion Control wird die Auswertung der Netzwerkzustände und der Netzwerkknotenzustände von Motion Control übernommen.

Der Funktionsbaustein COM405_NETWORK_CONTROL aus der Bibliothek CANopen_PLC01_30bd00 (oder höher) darf nicht zur Steuerung der Netzwerkzustände und der Netzwerkknotenzustände verwendet werden.

Zur Auswertung der Netzwerkzustände und der Netzwerkknotenzustände darf der Funktionsbaustein COM405_NETWORK_CONTROL aus der Bibliothek CANopen_PLC01_30bd00 (oder höher) verwendet werden.

HINWEIS

Bei der Verwendung von Motion Control erfolgt der Netzwerkstart automatisch durch Motion Control.

HINWEIS

Die Netzwerkvariablen müssen in der Motion Control Event-Task gelesen und geschrieben werden.

Ausnahme: Der PDO Übertragungstyp der Netzwerkvariablen ist asynchron eingestellt.

4.4 Programmierung des Datenaustauschs mit PROPROG wt II und Bibliothek CANop405_PLC01_20bd03

Die Programmierung des Datenaustauschs mit PROPROG wt II und der Bibliothek CANop405_PLC01_20bd03 (oder höher) kann mit

- Optionsmodul CANopen-Master, Softwarestand **< 01.20**
  (d. h. < BM4-O-ETH-02/CAN-04-01-00-001-**005**)
- Optionsmodul CANopen-Master, Softwarestand ≥ **01.20**
  (d. h. ≥ BM4-O-ETH-02/CAN-04-01-00-001-**005**)

erfolgen.

4.4.1 Übersicht

In den nachfolgenden Kapiteln werden Ihnen die verschiedenen Möglichkeiten ein Optionsmodul CANopen-Master in einem CANopen Netzwerk zu verwenden im Detail erläutert. Es werden Ihnen sowohl die grundlegenden Mechanismen wie sie im Profil DS 301 definiert sind aufgezeigt, als auch die damit zusammenhängende Verwendung der Funktionsbausteine aus der Bibliothek CANop405_PLC01_20bd00 (oder höher) nach Profil DS 405. Die Darstellungen werden durch ein sich Schritt für Schritt aufbauendes Beispiel-Projekt für PROPROG wt II begleitet. In diesem Beispiel-Projekt wird ein einfaches Netzwerk mit einem Optionsmodul CANopen-Master für die b maXX drive PLC und einem Optionsmodul CANopen-Slave für den b maXX Regler aus der Gerätereihe der b maXX Antriebe programmiert. Für die Inbetriebnahme des Optionsmodul CANopen-Slave sehen Sie bitte in der zugehörigen Betriebsanleitung nach. Beachten Sie bitte auch eventuell notwendige Einstellungen des b maXX Regler aus der Gerätereihe der b maXX Antriebe, um die Kommunikation mit dem Optionsmodul CANopen-Slave zu ermöglichen.

HINWEIS

Das entstehende Beispiel-Projekt ist dazu gedacht, die CANopen-Funktionen, welche mit dem Optionsmodul CANopen-Master möglich sind, näher zu erläutern und einfach nachvollziehbar zu machen. Es ist keinesfalls als voll funktionsfähige Applikation anzusehen. Die Daten, welche über die Beispiele auf den CANopen-Slave geschrieben werden, haben vom Wert her keine funktionale Bedeutung. Stellen Sie daher unbedingt sicher, dass ein eventuell angeschlossener Antrieb nicht im Betriebs bereiten Zustand ist!

Natürlich bieten sich mit dem Optionsmodul CANopen-Master 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.

4.4.2 Durchzuführende Schritte

Um das Optionsmodul CANopen-Master für den Datenaustausch in einem CANopen-Netzwerk verwenden zu können sind folgende Schritte durchzuführen:

1 Physikalische Inbetriebnahme des CANopen-Netzwerks (siehe Betriebsanleitung Optionsmodul BM4-O-ETH-02 / BM4-O-CAN-04 und Betriebsanleitungen der CANopen-Netzwerkknoten)

2 Anlegen eines PROPROG wt II Projektes mit der Bibliothek CANop405_PLC01_20bd00 (oder höher)

3 Implementierung der Funktionsbausteine zur CANopen-Kommunikation, Übersetzen und Senden des Projektes

Natürlich können Sie auch mit dem Schritt 2 beginnen und die physikalische Inbetriebnahme in einem späteren Schritt durchführen.

4.4.3 Inbetriebnahme des CANopen-Netzwerk

Einzelheiten zur Inbetriebnahme des CANopen-Netzwerks entnehmen Sie bitte der Betriebsanleitung zum Optionsmodul BM4-O-ETH-02 / BM4-O-CAN-04 und den Betriebsanleitungen der übrigen CANopen-Netzwerkknoten.

Nach der physikalischen Inbetriebnahme des Netzwerks und Zuschalten der Spannungsversorgung für das b maXX 4400 Gerät ist das Optionsmodul CANopen-Master nach ca. 5 s betriebsbereit. Dies wird Ihnen durch die CANopen-LEDs an der Frontseite des Optionsmodul angezeigt. Zwei Kombinationen sind möglich:

H5 (grün): Blinkt H6 (rot): Aus	CANopen: Das Optionsmodul wartet auf die Initialisierung durch ein Applikationsprogramm auf der b maXX drive PLC
H5 (grün): Aus H6 (rot): Ein	CANopen: Der CAN ist im Bus-Off Zustand

Blinkt: t_{on} = 200 ms, t_{off} = 200 ms

Alle anderen Zustände der LEDs H5 und H6 deuten auf Fehler hin. Sehen sie hierzu bitte in der zugehörigen Betriebsanleitung zum Optionsmodul CANopen-Master BM4-O-ETH-02 / BM4-O-CAN-04 nach.

4.4.4 Anlegen eines Projektes und Einbinden der Bibliothek CANop405_PLC01_20bd00

4.4.4.1 Vorgehen beim Anlegen eines Projektes

Um das Optionsmodul CANopen-Master mit den CANopen-Funktionsbausteinen nach Profil DSP 405 verwenden zu können, benötigen Sie ein PROPROG wt II - Projekt für die b maXX PLC01. Falls Sie noch kein eigenes Projekt für ihre Applikation angelegt haben, erstellen Sie dies bitte mit der Vorlage *BM4_O_PLC01*. Sie benötigen dazu ein PROPROG wt II Version 3.0 ab Build 261. 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_20bd03 (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_20bd00 in Ihr Projekt ein.

4.4.4.2 Beispiel: Anlegen des Projektes "CANopenMaster_Example"

Es wurde das Beispiel-Projekt "CANopenMaster_Example" mit der Vorlage BM4_O_PLC01 angelegt und die Bibliothek CANop405_PLC01_20bd00 eingebunden.

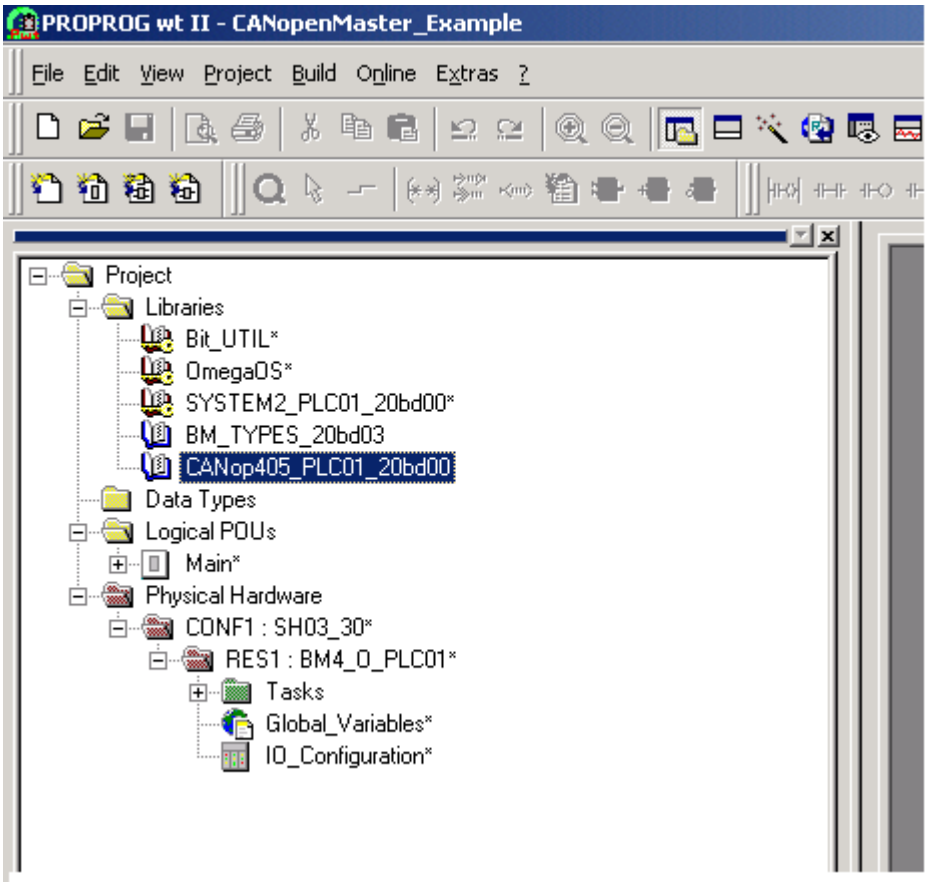


Abbildung 62: Beispiel: Anlegen des Projektes "CANopenMaster_Example"

4.4.5 Anlegen einer globalen Variable für den Datenaustausch

Für den Datenaustausch zwischen Optionsmodul BM4-O-ETH-02 / BM4-O-CAN-04 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 BM4_O_PLC01 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 _CANop405Ma_Ctrl_Slot_G bis _CANop405Ma_Ctrl_Slot_M zur Verfügung. Sie finden diese auch im Arbeitsblatt "Global_Variables":

```
(* Option module CANopen-Master (BM4-O-ETH-02) *)
   _CANop405Ma_Ctrl_Slot_G       AT %MB3.2016384    :   CANop405_PLC_BMSTRUCT;
   _CANop405Ma_Ctrl_Slot_H       AT %MB3.3016384    :   CANop405_PLC_BMSTRUCT;
   _CANop405Ma_Ctrl_Slot_J       AT %MB3.4016384    :   CANop405_PLC_BMSTRUCT;
   _CANop405Ma_Ctrl_Slot_K       AT %MB3.5016384    :   CANop405_PLC_BMSTRUCT;
   _CANop405Ma_Ctrl_Slot_L       AT %MB3.6016384    :   CANop405_PLC_BMSTRUCT;
   _CANop405Ma_Ctrl_Slot_M       AT %MB3.7016384    :   CANop405_PLC_BMSTRUCT;
```

Abbildung 63: Globale Variablen für das Optionsmodul CANopen-Master 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 _CANop405Ma_Ctrl_Slot_G (bis _CANop405Ma_Ctrl_Slot_M) vom Datentyp CANop405_PLC_BMSTRUCT an. Diese Variable muss als globale Variable deklariert werden und auf die Basisadresse zur CANopen-Master Kommunikation des Optionsmoduls BM4-O-ETH-02 / BM4-O-CAN-04 gelegt werden. Die Basisadresse ist abhängig vom Steckplatz:

Steckplatz (Slot)	Basisadresse für CANopen-Master Kommunikation
G	%MB3.2016384
H	%MB3.3016384
J	%MB3.4016384
K	%MB3.5016384
L	%MB3.6016384
M	%MB3.7016384

4.4.6 Initialisierung des Optionsmoduls CANopen-Master

4.4.6.1 Vorgehen bei der Initialisierung des Optionsmoduls CANopen-Master

Die Initialisierung des Optionsmoduls CANopen-Master erfolgt mit dem FB CANop405_INIT. 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 in dieser POE.
- Beschalten Sie den Funktionsbaustein mit Variablen vom richtigem Datentyp.
  Am Eingang us_BAUDRATE stellen Sie die Baudrate für das CANopen-Netzwerk ein.
  Folgende Werte sind möglich:

us_BAUDRATE	Baudrate des Netzwerks
0	1 MBit/s
1	800 kBit/s
2	500 kBit/s

us_BAUDRATE	Baudrate des Netzwerks
3	250 kBit/s
4	125 kBit/s
5	100 kBit/s
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s

4.4.6.2 Beispiel: Initialisierung des Optionsmoduls CANopen-Master

Die Beschaltung des FB CANop405_INIT kann für eine Initialisierung mit 500 kBit/s wie folgt aussehen:

```
(* Initialise the CANopen-Master with 500 kBit/s *)
```

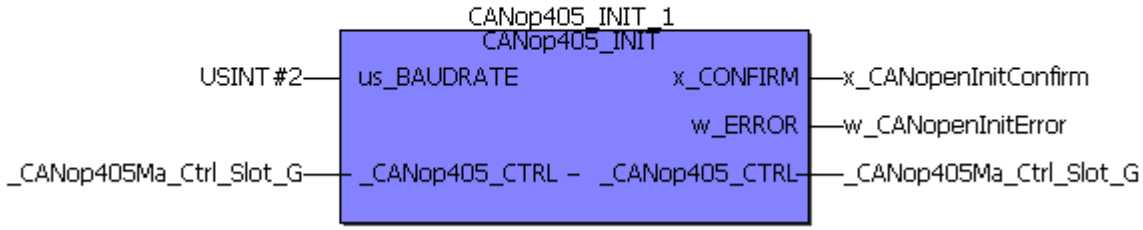


Abbildung 64: Initialisierung des CANopen-Master mit dem FB CANop405_INIT

- 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 meldet eine erfolgreiche Initialisierung mit x_CONFIRM = 1 und w_ERROR = 16#0.

Beispiel:

```
(* Initialise the CANopen-Master with 500 kBit/s *)
```

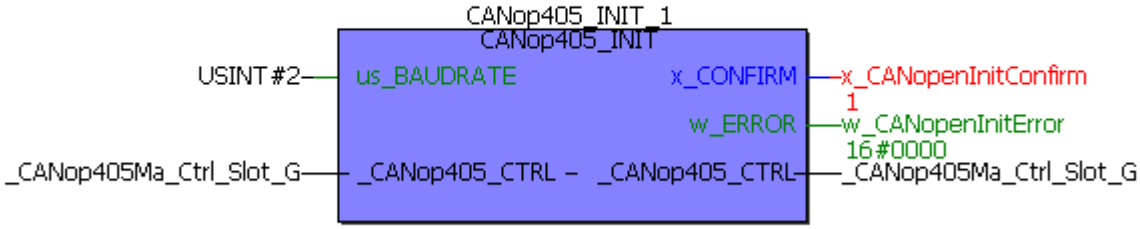


Abbildung 65: Initialisierung des CANopen-Master mit dem FB CANop405_INIT - Online

Eine erfolgreiche Initialisierung wird auch durch die LEDs an der CANopen-Buchse am Optionsmodul angezeigt:

H5 (grün): H6 (rot):	Ein Aus	CANopen: Das Optionsmodul CANopen-Master ist initialisiert und bereit zur Datenübertragung

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

4.4.7 Starten der einzelnen Netzwerkknoten - NMT

4.4.7.1 Definition nach CANopen-Spezifikation

Mittels NMT (Netzwerk-Management) 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:

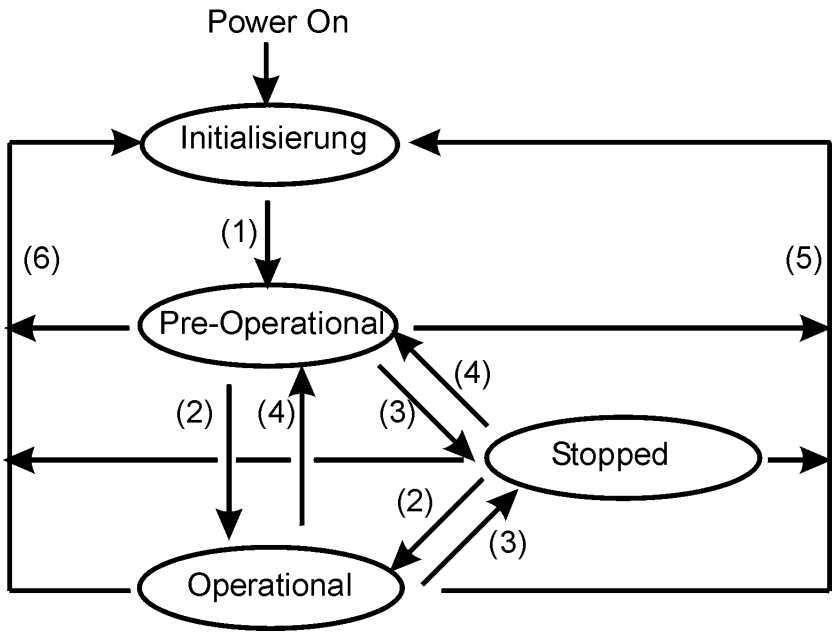


Abbildung 66: Zustandsübergänge

Nach der INITIALISIERUNG (ausgelöst durch das Einschalten des Netzwerkknotens) wird automatisch der Zustand PRE-OPERATIONAL erreicht. Befindet sich ein Netzwerkknoten in diesem Status, kann er über SDOs konfiguriert werden. Der Datenaustausch über PDOs ist nicht möglich.

Im Zustand STOPPED ist nur das Node Guarding möglich. Weder SDOs noch PDOs können gesendet oder empfangen werden. Es gibt sechs mögliche Übergänge zwischen den einzelnen Zuständen. Der Übergang wird entweder automatisch (Power On), durch Kommando vom Master oder Knoten intern (z. B. im Fehlerfall) ausgelöst.

(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

Das Optionsmodul CANopen-Master hat selbst keine Zustände und Zustandsübergänge nach CANopen-Definition.

Der CANopen-Master kann die Zustandsübergänge (2) bis (6) durch ein Telegramm auslösen. Dieses Telegramm ist unbestätigt, d. h. der CANopen-Master hat keine Information ob ein Netzwerkknoten das Telegramm erhalten hat und ob er den Zustandswechsel durchgeführt hat. Auf dem CANopen-Master kann der Zustand eines Netzwerkknotens nur über das Node Guarding ausgelesen werden (siehe Kapitel ▹Überwachen von Netzwerkknoten durch Node Guarding◃ ab Seite 146).

HINWEIS

Beachten Sie bitte die Auswirkungen der einzelnen Zustandsübergänge! Ein "Reset-Node" kann einen Reset des gesamten Netzwerkknotens auslösen und damit unerwünschte Auswirkungen auf Ihre Applikation haben. Informieren Sie sich über mögliche Auswirkungen in den Beschreibungen zu den einzelnen Netzwerkknoten.

4.4.7.2 NMT-Kommandos absetzen

Das Absetzen von NMT-Kommandos durch das Optionsmodul CANopen-Master erfolgt mit dem FB CANop405_NMT. 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 zyklischen Task aufgerufen werden.
- Platzieren Sie den FB CANop405_NMT in dieser POE.
- Beschalten Sie den Funktionsbaustein mit Variablen vom richtigen Datentyp. Am Eingang us_DEVICE geben Sie die Nummer des Netzwerkknotens an, welcher das NMT-Kommando auswerten soll. Mit dem Wert USINT#0 wird das Kommando von allen Netzwerkknoten ausgewertet. Das Kommando für den Zustandsübergang wird an us_TRANSITION_STATE angegeben. Die einzelnen Werte sind den Kommandos wie folgt zugeordnet:

us_TRANSITION_STATE	Kommando
1	Start Remote Node
2	Stop Remote Node
3	Enter Pre Operational
4	Reset Node
5	Reset Communication

- 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_NMT 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 das NMT-Kommando abgesetzt. Der FB CANop405_NMT meldet eine erfolgreiche Durchführung mit x_CONFIRM = 1 und w_ERROR = 16#0.

4.4.7.3 Beispiel: Netzwerkmanagement mit NMT

Um den FB CANop405_NMT 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. Wir wollen alle Netzwerkknoten in den Zustand OPERATIO-

NAL schalten, us_DEVICE muss somit den Wert 0 und us_TRANSITION_STATE den Wert 1 haben. Nach der Beschaltung des Funktionsbaustein übersetzen Sie das Projekt und senden Sie dieses als Projekt an die PLC. Starten Sie das Projekt. Mit x_ENABLE = 1 wird der Zustand OPERATIONAL auf allen Netzwerkknoten aktiviert:

```
(* Network control by FB CANop405_NMT *)
```

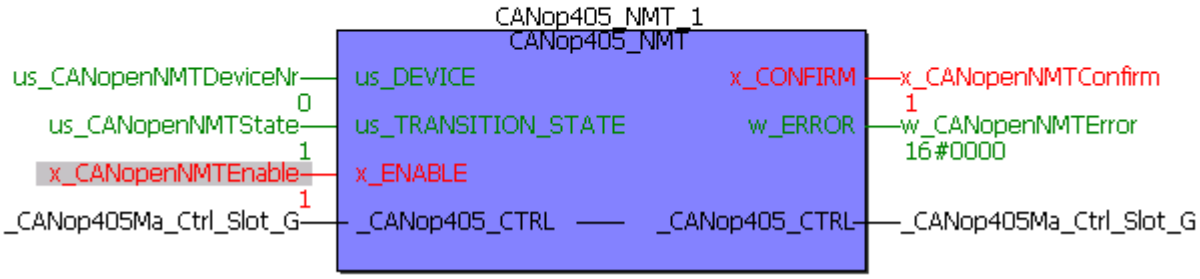


Abbildung 67: Netzwerkmanagement mit dem FB CANop405_NMT

4.4.8 Bedarfsdatenaustausch mit SDO

4.4.8.1 Definition nach CANopen-Spezifikation

a) Objekte eines Netzwerkknotens

Bei einem Zugriff über SDO (Service Data Object) wird grundsätzlich auf die Objekte eines Netzwerkknotens lesend oder schreibend zugegriffen. Die Objekte können abhängig vom Profil z. B. Konfigurationsdaten, Gerätefunktionen oder auch Parameter darstellen. Ein Netzwerkknoten hat ein Objektverzeichnis, in welchem alle Objekte des Gerätes enthalten sind. Auf dieses Objektverzeichnis kann über SDO direkt zugegriffen werden. Welche Objekte ein Netzwerkknoten wirklich unterstützt ist dessen Dokumentation zu entnehmen, denn nicht alle Objekte die in einem Profil definiert sind, müssen unterstützt werden. Daneben können für ein Gerät auch Hersteller spezifische Objekte definiert sein. Die drei wichtigsten Bereiche in einem Objektverzeichnis eines Gerätes sind:

- Objekte des Kommunikationsprofils (insbesondere DS 301)
- Hersteller spezifische Objekte
- Objekte des Geräteprofils

Objekte werden immer über einen Index (16 Bit) und einen Subindex (8 Bit) adressiert. Neben Index und Subindex enthält das Objektverzeichnis auch Angaben über Namen, Datentyp, Attribut, Defaultwert und Bedeutung eines Objektes.

Beispiele für Objekteinträge sind:

Objekt nach DS 301:

Index	Subindex	Name	Datentyp	Attribut	Default-wert	Bedeutung
1000h	00	Device Type	Unsigned 32	ro	402	Unterstütztes Geräteprofil

Dieser Eintrag im Objektverzeichnis des Optionsmoduls BM4-O-CAN-03 (CANopen-Slave für b maXX Regler) gibt an, dass das Objekt mit Index 1000h und dem Subindex 0h die Nummer des unterstützten Geräteprofils enthält. Der Eintrag 402 bedeutet somit, dass das Geräteprofil DSP 402 (Antriebe) unterstützt wird. Der Wert kann nur gelesen werden (ro = read only) und wird als 32 Bit Wert übertragen. Bei einem I/O-Klemmenmodul würde das gleiche Objekt den Defaultwert 401 für das Geräteprofil DS 401 (I/O-Module) haben.

Objekt nach DSP 402:

Index	Subindex	Name	Datentyp	Attribut	Default-wert	Bedeutung
6040h	00	Control-word	Unsigned 16	rw	-	Steuerwort

Dieser Eintrag im Objektverzeichnis des Optionsmoduls BM4-O-CAN-03 (CANopen-Slave für b maXX Regler) gibt an, dass das Objekt mit Index 6040h und dem Subindex 0h das Steuerwort (Parameter 300) darstellt. Der Wert kann gelesen und geschrieben werden (rw = read write) und wird als 16 Bit Wert übertragen. Da dies ein Geräteprofil spezifisches Objekt ist, hat dies bei einem I/O-Klemmenmodul eine andere Bedeutung, nämlich den Filter für digitale Eingänge.

Hersteller spezifisches Objekt:

Index	Subindex	Name	Datentyp	Attribut	Default-wert	Bedeutung
41BAh	00	SVG set value 1	Signed 16	rw	16384	Sollwertgenerator Sollwert 1

Dieser Eintrag im Objektverzeichnis des Optionsmoduls BM4-O-CAN-03 (CANopen-Slave für b maXX Regler) gibt an, dass das Objekt mit Index 41BAh und dem Subindex 0h den Sollwert 1 des Sollwertgenerators (Parameter 442) darstellt. Der Wert kann gelesen und geschrieben werden (rw = read write) und wird als 16 Bit Wert mit Vorzeichen übertragen.

HINWEIS

Das Optionsmodul CANopen-Master besitzt selbst keine Objekte, weder Kommunikationsprofil spezifische noch Geräteprofil spezifische. Mit dem Optionsmodul CANopen-Master und den Funktionsbausteinen aus der Bibliothek CANop405_PLC01_20bd00 können sie direkt auf die einzelnen SDO und PDO Telegramme zugreifen.

b) Ablauf der SDO-Kommunikation

Eine SDO-Kommunikation entspricht dem Client/Server-Kommunikationsmodell, d. h. das Optionsmodul CANopen-Master ist der Client und sendet ein Telegramm mit dem Auftrag an einen Netzwerkknoten 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. Der umgekehrte Weg, das Optionsmodul CANopen-Master antwortet als Server auf die Anfrage eines Clients, ist nicht möglich. Mit dem CANopen-Master 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.

4.4.8.2 Objekte mit SDO schreiben

Das Schreiben von Objekten über SDO erfolgt über die Funktionsbausteine CANop405_SDO1_WRITE bis CANop405_SDO8_WRITE. Mit diesen Funktionsbausteinen können gleichzeitig bis zu 8 SDO Aufträge gestartet werden. Für jeden SDO Auftrag steht ein FB zur Verfügung.

HINWEIS

Die Funktionsbausteine CANop405_SDO1_WRITE bis CANop405_SDO8_WRITE dürfen nicht mehrfach instanziiert werden.

Um die Funktionsbausteine CANop405_SDO1_WRITE bis CANop405_SDO8_WRITE 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 zyklischen Task aufgerufen werden. Sie können auch die POE mit dem FB CANop405_NMT verwenden.
- Platzieren Sie einen oder mehrere der FBs CANop405_SDO1_WRITE bis CANop405_SDO8_WRITE nach Ihrem Bedarf in dieser POE.
- Beschalten Sie die Funktionsbausteine mit Variablen vom richtigem Datentyp und Belegen Sie diese mit den gewünschten Werten. Am Eingang us_DEVICE geben sie die Nummer des Netzwerkknotens an, auf welchem Sie ein Objekt beschreiben wollen. w_INDEX und b_SUBINDEX sind die Adressen des Objektes, us_DATALENGTH die Größe des Objektes in Byte und an d_DATA werden die zu schreibenden Daten angegeben.
- 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 in die-

ser Task ein. Sollten Sie die POE mit dem FB CANop405_NMT verwenden brauchen Sie keine weitere Task anlegen.

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

Mit x_ENABLE = 1 wird das Objekt über SDO gesendet. Die FBs CANop405_SDO1_WRITE bis CANop405_SDO8_WRITE melden eine erfolgreiche Durchführung mit x_CONFIRM = 1 und w_ERROR = 16#0. Da das Schreiben von Objekten über SDO vom anderen Netzwerkknoten bestätigt wird, kann es sein, dass Sie in w_ERROR und ud_ERRORINFO eine Fehlermeldung erhalten. Ein Grund kann z. B. sein, dass das Objekt nicht schreibbar (Attribut = ro) ist. Entnehmen Sie Details zu den Fehlermeldungen bitte der Online-Dokumentation zu den Funktionsbausteinen.

HINWEIS

Es darf nicht gleichzeitig die gleiche Knotennummer (us_DEVICE) für die FBs CANop405_SDOx_READ (siehe nächstes Kapitel) und die FBs CANop405_SDOx_WRITE verwendet werden (mit x = 1 bis 8). Die FBs CANop405_SDOx_READ und CANop405_SDOx_WRITE dürfen nicht gleichzeitig aktiv sein.

4.4.8.3 Beispiel: Objekt über SDO schreiben

Wir wollen den Parameter 1172 "Hochlaufgeber Hochlaufzeit" des b maXX Regler mit 56 s beschreiben. Der Parameter 1172 hat im Objektverzeichnis des Optionsmodul CANopen-Slave folgende Objektdaten:

Index	Subindex	Name	Datentyp	Attribut	Default-wert	Bedeutung
604Fh	00h	vl_ramp_function_time	Unsigned 32	rw	0	Hochlaufgeber Hochlaufzeit

Das Optionsmodul CANopen-Slave ist über dessen DIP-Schalter auf die Modulnummer 5 eingestellt. Wir verwenden den FB CANop405_SDO1_WRITE. Den FB CANop405_SDO1_WRITE implementieren wir in der POE mit dem Funktionsbaustein CANop405_NMT. Nach der Beschaltung des Funktionsbaustein muss das Projekt übersetzt und als Projekt an die PLC gesendet werden. Starten Sie das Projekt. Mit x_ENABLE = 1 starten wir die Datenübertragung über SDO.

In der Online-Darstellung sollte der Funktionsbaustein nach absenden des SDO wie folgt aussehen:

```
(* Write object via SDO to device nr. 5 *)
```

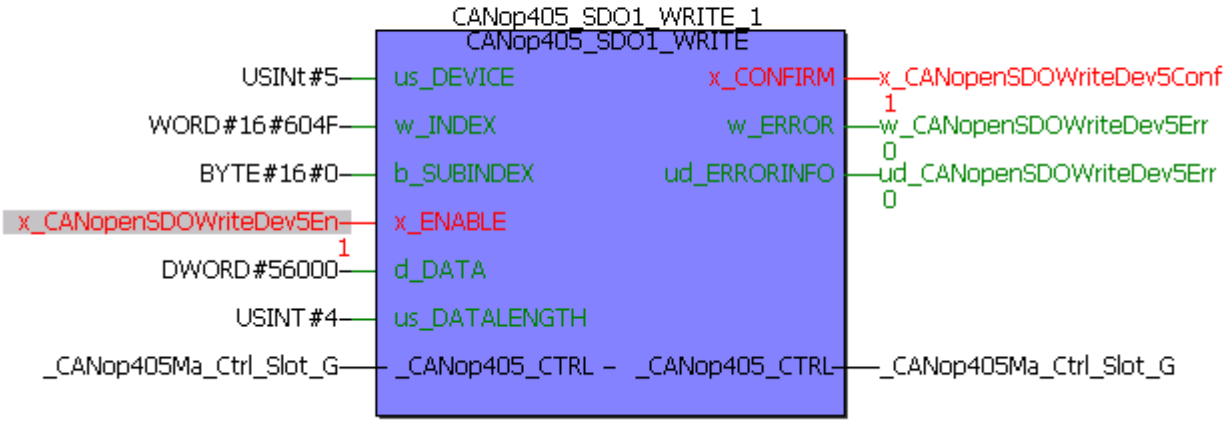


Abbildung 68: Hochlaufzeit mit dem FB CANop405_SDO1_WRITE schreiben

Bei einer Überprüfung des Parameters 1172 "Hochlaufgeber Hochlaufzeit" mit WinBASS II sollte dieser den Wert 56 s haben:

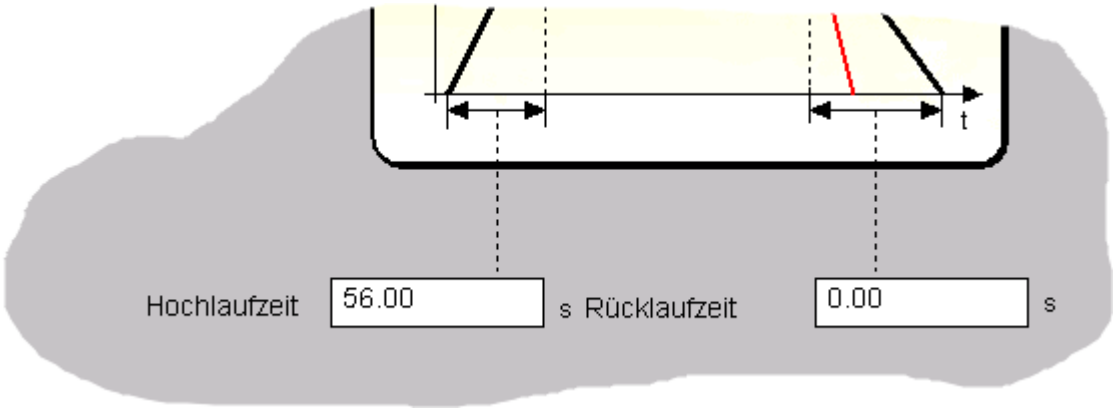


Abbildung 69: Hochlaufzeit in WinBASS II

4.4.8.4 Objekte mit SDO lesen

Das Lesen von Objekten über SDO erfolgt über die Funktionsbausteine CANop405_SDO1_READ bis CANop405_SDO8_ READ. Mit diesen Funktionsbausteinen können gleichzeitig bis zu 8 SDO Aufträge gestartet werden. Für jeden SDO Auftrag steht ein FB zur Verfügung.

HINWEIS

Die Funktionsbausteine CANop405_SDO1_ READ bis CANop405_SDO8_ READ dürfen nicht mehrfach instanziiert werden.

Um die Funktionsbausteine CANop405_SDO1_ READ bis CANop405_SDO8_ READ 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 zyklischen Task aufgerufen werden. Sie können auch die POE mit dem FB CANop405_NMT bzw. den Funktionsbausteinen für Objekte über SDO schreiben verwenden.
- Platzieren Sie einen oder mehrere der FBs CANop405_SDO1_ READ bis CANop405_SDO8_ READ nach Ihrem Bedarf in dieser POE.
- Beschalten Sie die Funktionsbausteine mit Variablen vom richtigem Datentyp und belegen Sie diese mit den gewünschten Werten. Am Eingang us_DEVICE geben Sie die Nummer des Netzwerkknotens an, von welchem Sie ein Objekt lesen wollen. w_INDEX und b_SUBINDEX sind die Adressen des Objektes. An us_DATALENGTH wird Ihnen die Größe des gelesenen Objektes in Byte und an d_DATA werden die gelesenen Daten angezeigt.
- 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 in dieser Task ein.
- Übersetzen Sie das Projekt und laden Sie es als (Boot-)Projekt auf die PLC. Starten Sie die PLC.

Mit x_ENABLE = 1 wird das Objekt über SDO vom Slave angefragt. Die FBs CANop405_SDO1_READ bis CANop405_SDO8_ READ melden eine erfolgreiche Durchführung mit x_CONFIRM = 1 und w_ERROR = 16#0. Da das Lesen von Objekten über SDO vom Slave bestätigt wird, kann es sein, dass Sie in w_ERROR und ud_ERRORINFO eine Fehlermeldung erhalten. Ein Grund kann z. B. ein nicht vorhandenes Objekt sein. Entnehmen Sie Details zu den Fehlermeldungen bitte der Online-Dokumentation zu den Funktionsbausteinen.

HINWEIS

Es darf nicht gleichzeitig die gleiche Knotennummer (us_DEVICE) für die FBs CANop405_SDOx_READ und die FBs CANop405_SDOx_WRITE (siehe Kapitel ▹Objekte mit SDO schreiben◃ ab Seite 109) verwendet werden (mit x = 1 bis 8). Die FBs CANop405_SDOx_READ und CANop405_SDOx_WRITE dürfen nicht gleichzeitig aktiv sein.

4.4.8.5 Beispiel: Objekt über SDO lesen

Wir wollen jetzt den zuvor geschriebenen Parameter 1172 "Hochlaufgeber Hochlaufzeit" des b maXX Regler zurück lesen. Sofern dieser Parameter von anderer Seite nicht überschrieben wurde und das Gerät nicht ausgeschaltet wurde, muss dessen Wert 56 s betragen. Der Parameter 1172 hat im Objektverzeichnis des Optionsmodul CANopen-Slave folgende Objektdaten:

Index	Subindex	Name	Datentyp	Attribut	Default-wert	Bedeutung
604Fh	00h	vl_ramp_function_time	Unsigned 32	rw	0	Hochlaufgeber Hochlaufzeit

Das Optionsmodul CANopen-Slave ist auf die Modulnummer 5 eingestellt. Den FB CANop405_SDO1_READ implementieren wir in der gleichen POE wie den Funktionsbaustein CANop405_SDO1_WRITE. Nach der Beschaltung des Funktionsbaustein übersetzen Sie das Projekt und senden sie dieses als Projekt an die PLC. Starten Sie das Projekt. Mit x_ENABLE = 1 starten wir die Datenübertragung über SDO. In der Online-Darstellung sollte der Funktionsbaustein nach absenden des SDO wie folgt aussehen:

```
(* Read object via SDO from device nr. 5 *)
```

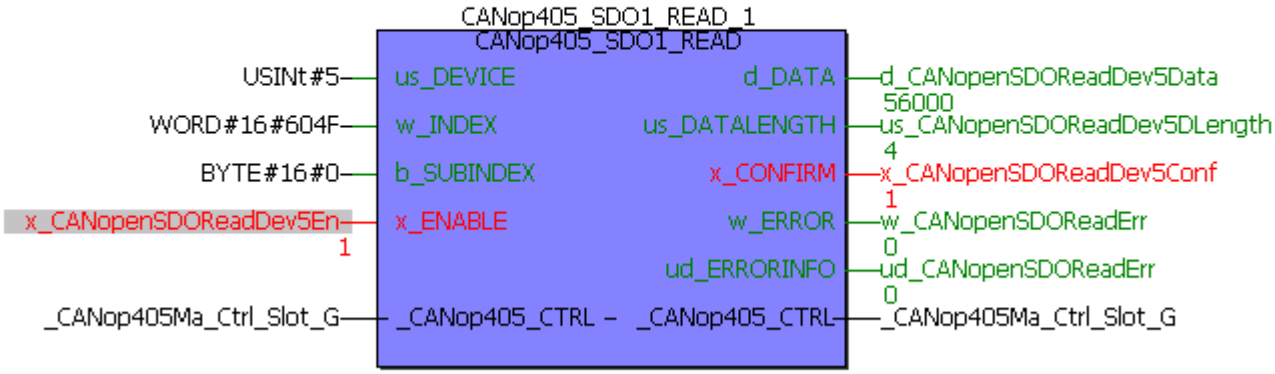


Abbildung 70: Hochlaufzeit mit dem FB CANop405_SDO1_READ lesen

4.4.9 Prozessdatenaustausch mit PDOs

4.4.9.1 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-Master 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 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.

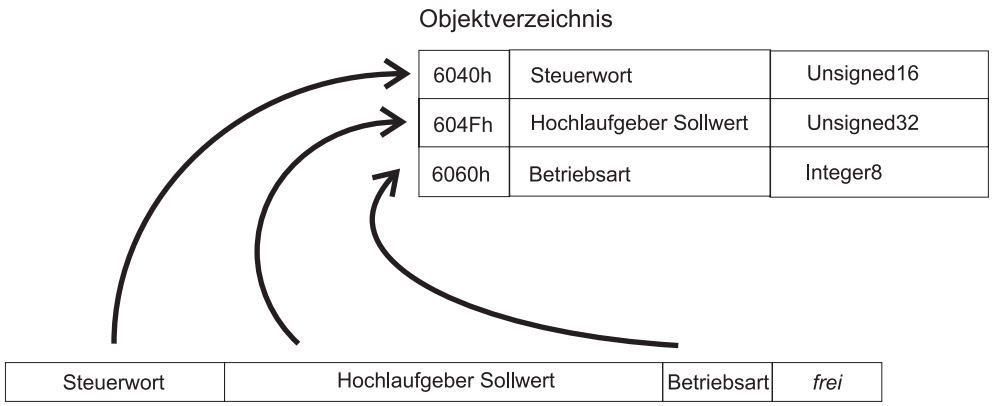


Abbildung 71: Objekte aus einem PDO-Telegramm lesen

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.

Der Datenaustausch über PDOs lässt sich in drei Phasen einteilen:

1 Konfiguration des Übertragungsverhaltens eines PDO auf jedem Netzwerkknoten

2 Konfiguration des PDO-Mapping auf jedem Netzwerkknoten

3 Zyklischer Datenaustausch über PDOs

Die Konfiguration der PDOs wird in Kommunikationsprofil spezifischen Objekten eines Gerätes abgelegt. Diese Objekte werden über SDOs gelesen und geschrieben (siehe Kapitel ▹Bedarfsdatenaustausch mit SDO◃ ab Seite 107). Die Konfiguration von PDOs darf nur im Zustand PRE-OPERATIONAL eines Netzwerkknotens erfolgen. Im Nachfolgenden werden Ihnen Einzelheiten zu den Konfigurations-Objekten erläutert.

4.4.9.2 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. Maximal sind 512 Empfangs-PDOs und 512 Sende-PDOs möglich. 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	*Nicht verwendet*
Subindex 4	Eintrag zur Kompatibilität. Optional.
Subindex 5	*Nicht verwendet.*

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 Identifiervergabe (siehe Kapitel ▹Datenaustausch und Objekte des physikalischen Bussystems◃ ab Seite 53) 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-Master nicht unterstützt)
28 - 11	wird vom Optionsmodul CANopen-Master 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 Identifiervergabe (siehe Kapitel ▹Datenaustausch und Objekte des physikalischen Bussystems◃ ab Seite 53)

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 jedem empfangenen SYNC-Telegramm (siehe Kapitel ▹Synchronisieren des Datenaustausch◃ ab Seite 104)	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	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.	*Nicht möglich*
253 Remote	Ein Update und das Senden erfolgt nach Empfang einer Remote-Anforderung.	*Nicht möglich*
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. [1)]	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. [2)]	Jedes PDO mit passender COB-ID wird bei Empfang übernommen

1) 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.

2) Ereignis gesteuertes Senden bedeutet, die Sendebedingung ist an die Änderung eines Wertes eines der gemappten Objekte gebunden. Sind beispielsweise 3 Objekte gemappt (Statuswort, Drehzahl-Istwert, Ist-Betriebsart), 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.

HINWEIS

Beachten Sie, dass manche Geräte nur bestimmte Übertragungsarten implementiert haben und auch unterschiedliche Default-Einstellungen haben können. Sollten Sie von einem Netzwerkknoten kein PDO erhalten oder sollte ein Netzwerkknoten ein PDO nicht übernehmen liegt dies häufig an der Einstellung des Übertragungstyps.

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).

4.4.9.3 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. 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.

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.

Beachten Sie bitte, dass nicht jedes Objekt in ein PDO mappbar ist.

HINWEIS

In den Geräteprofilen sind Default-Mappings definiert. Sollten die Default-Mapping-Werte für Sie ausreichen, brauchen Sie hier keine weiteren Einstellungen auf den Netzwerkknoten vorzunehmen. Im Anhang finden Sie eine Übersicht der Default-Mappings für Geräte nach Profil DS 401 und DSP 402.

4.4.9.4 Zyklischer Datenaustausch mit PDO

Nach der Konfiguration der Übertragungseigenschaften und dem Mapping des PDO kann mit dem eigentlichen Datenaustausch über PDOs begonnen werden. Das Optionsmodul CANopen-Master sendet PDOs, auf welche die Netzwerkknoten entsprechend ihrer Einstellung reagieren. Ebenso kann das Optionsmodul CANopen-Master PDOs empfangen, welche von den Netzwerkknoten gesendet 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-Master ein PDO, so ist dies auch auf der Master-Seite ein Empfangs-PDO oder auch Receive-PDO (RxPDO), denn der Netzwerkknoten empfängt dieses PDO. Gleiches gilt für ein Sende-PDO oder Transmit-PDO (TxPDO), welches vom Master empfangen wird.

4.4.9.5 Objekte über PDOs schreiben

a) Mechanismus auf dem Optionsmodul CANopen-Master

Das Schreiben von Objekten über PDOs erfolgt beim Optionsmodul CANopen-Master mit dem FB CANop405_PDO_WRITE. Der dabei ablaufende Mechanismus sieht wie folgt aus:

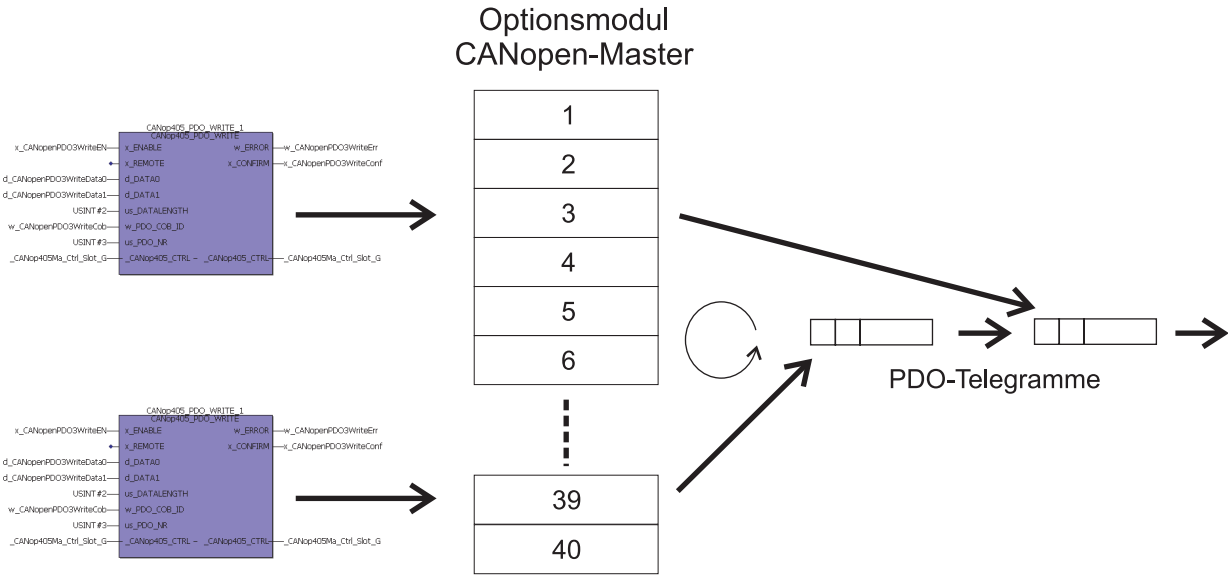


Abbildung 72: Mechanismus auf dem Optionsmodul für PDO schreiben

Jedem zu sendenden PDO wird ein FB CANop405_PDO_WRITE zugeordnet indem von diesem Funktionsbaustein entsprechende Instanzen gebildet werden und die COB-ID des PDO am Baustein angegeben wird. Jedem dieser Funktionsbausteine muss über einen Eingang am Funktionsbaustein ein Sendefach auf dem Optionsmodul CANopen-Master zugeordnet werden. Insgesamt stehen 40 Sendefächer zur Verfügung. Es können also 40 Sendeaufträge für PDOs gleichzeitig eingetragen werden. Das Optionsmodul kontrolliert die Sendefächer zyklisch und prüft ob in einem Fach ein Sendeauftrag für ein PDO eingetragen ist. Ist dies der Fall, wird daraus ein PDO-Telegramm generiert und gesendet.

b) Vorbereitung und Konfiguration

Bevor Sie ein PDO senden können müssen Sie einige Vorbereitungen treffen und Konfigurationseinstellungen auf dem Netzwerkknoten, welcher das PDO auswerten soll, durchführen. Gehen Sie dazu in folgenden Schritten vor:

Bestimmen Sie wichtige Daten:

- Bestimmen Sie Nummer des Netzwerkknotens im CANopen-Netzwerk, welcher das zu sendende PDO verarbeiten soll.
- Bestimmen Sie, welches PDO des Netzwerkknotens Sie senden wollen. Schlagen Sie hierzu in der Dokumentation zum Netzwerkknoten nach, um festzustellen wie viele PDOs dieser unterstützt.
- Bestimmen Sie ein noch nicht vergebenes Sendefach für das PDO. Sie sollten bei Sendefach 1 beginnen und ohne Lücken die nächsten Sendefächer vergeben.
- Legen Sie den Übertragungstyp des PDO fest. Beim Empfangen von PDOs durch einen Netzwerkknoten haben Sie nur zwei Möglichkeiten: Sofortige Übernahme oder Übernahme nach Erhalt eines SYNC-Telegramms.
- Legen Sie jetzt die Objekte fest, welche in dem PDO gemapped sind. Beachten Sie das Default-Mapping des Netzwerkknotens, eventuell können Sie es für Ihre Applikation verwenden.
- Legen Sie fest, in welchem Zeitraster das PDO gesendet werden soll. Entsprechend dem Zeitraster legen Sie später eine Task in PROPROG wt II an, um das PDO in das Sendefach einzutragen. Beachten Sie bitte die maximale CANopen-Buslast. Bei einem Betrieb des CANopen mit 1 Mbit/s können Sie rechnerisch 7 PDOs mit je 8 Byte Daten pro ms senden. Das bedingt aber, dass kein weiterer Datenverkehr stattfindet. Dies ist aber sehr unwahrscheinlich, da Sie vermutlich auch PDOs empfangen und weitere CANopen-Funktionen wie Node Guarding und Sync anwenden wollen. Die tatsächlich maximal mögliche Buslast wird also weitgehend durch Ihre Applikation bestimmt. Tatsächlich werden also weniger als 7 PDOs pro ms gesendet werden können.

Wiederholen Sie diese Schritte für alle PDOs, welche Sie schreiben wollen.

Programmieren Sie die Konfiguration und führen Sie diese durch:

- Initialisieren Sie das Senden von PDOs mit dem FB CANop405_PDO_INIT. Diesen Funktionsbaustein müssen Sie vor dem FB CANop405_INIT platzieren. Am FB CANop405_PDO_INIT muss die Nummer des höchsten verwendeten Sendefachs angegeben werden. Maximal können 40 Sendefächer verwendet werden.
- Setzen Sie den Netzwerkknoten über den FB CANop405_NMT in den Zustand PRE-OPERATIONAL.
- Verwenden Sie die in Kapitel "Objekte mit SDO schreiben" beschriebenen Funktionsbausteine CANop405_SDO1_WRITE bis CANop405_SDO8_WRITE um die Konfigurationsdaten (Objekte 1400h, 1401h, etc. und 1600, 1601, etc.) des Netzwerkknotens zu überprüfen und einzustellen.
- Zur Ermittlung der COB-ID des PDO können Sie den Funktionsbaustein CANop405_COB_ID verwenden. Dieser FB errechnet aus der von Ihnen ermittelten Knotennummer und PDO-Nummer die COB-ID. Der FB sollte in einer Initialisierungstask eingesetzt werden.

c) Beispiel: Konfiguration PDO senden

Wir wollen zyklisch alle 2 ms den Hochlaufgeber Eingang P1171 und den Drehzahl-Zusatz-Sollwert P1040 des Optionsmodul BM4-O-CAN-03 beschreiben. Es soll das Empfangs-PDO 2 des Optionsmodul BM4-O-CAN-03 verwendet werden. Das PDO soll bei Eintreffen übernommen werden. Da bisher noch kein PDO zum senden eingerichtet wurde verwenden wir Sendefach 1. Schließlich kennen wir noch die Knotennummer aus den vorhergehenden Kapiteln: 5. Für die Parameter 1171 und 1040 benötigen wir noch Daten aus dem Objektverzeichnis um das Mapping konfigurieren zu können:

Parameter	Index	Subindex	Datenlänge
1171 Hochlaufgeber Eingang	6042	0	16 Bit
1040 Drehzahl-Zusatz-Sollwert	4410	0	32 Bit

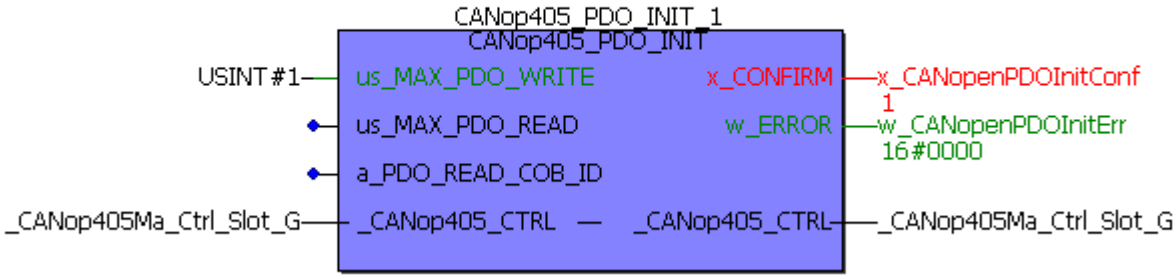
Bei P1171 stellt der zu schreibende Wert nach Objektverzeichnis direkt die Drehzahl in U/min dar. Die Normierung von P1040 über CANopen ist: -100% bis +100% Drehzahl entsprechen -230 bis +230.

HINWEIS

Die Parameter P1171 und P1040 sollen nur beispielhaft geschrieben werden und eine Änderung der Werte soll in WinBASS II sichtbar sein. Hinter den geschriebenen Werten steht keine Funktionalität, die Werte ändern sich zum Teil sprunghaft. Um Schäden zu vermeiden müssen Sie daher sicher stellen, dass ein eventuell am b maXX 4400 angeschlossener Antrieb nicht betriebsbereit ist!

Zunächst muss der FB CANop405_PDO_INIT konfiguriert werden. Dazu wird dieser FB in der Initialisierungs-POE vor dem FB CANop405_INIT platziert. An us_MAX_PDO_WRITE geben wir die Nummer des letzten verwendeten Sendefachs an. Da wir nur das erste Sendefach verwenden ist dies USINT#1. Nach Übersetzen und Download des Projektes sollte der FB im Online-Modus folgende Darstellung haben:

```
(* Initialise the PDO for the CANopen-Master *)
```




```
(* Initialise the CANopen-Master with 500 kBit/s *)
```

Abbildung 73: PDO Übertragung mit dem FB CANop405_PDO_INIT initialisieren

Das Optionsmodul BM4-O-CAN-03 muss zuerst in den Zustand PRE-OPERATIONAL gebracht werden, um die Konfiguration des PDO durchführen zu können. Hierzu verwenden wir wieder den FB CANop405_NMT. Unmittelbar nach Einschalten des b maXX 4400 Gerätes müsste dies nicht durchgeführt werden, da dann das Optionsmodul automatisch den Zustand PRE-OPERATIONAL einnimmt.

```
(* Network control by FB CANop405_NMT *)
```

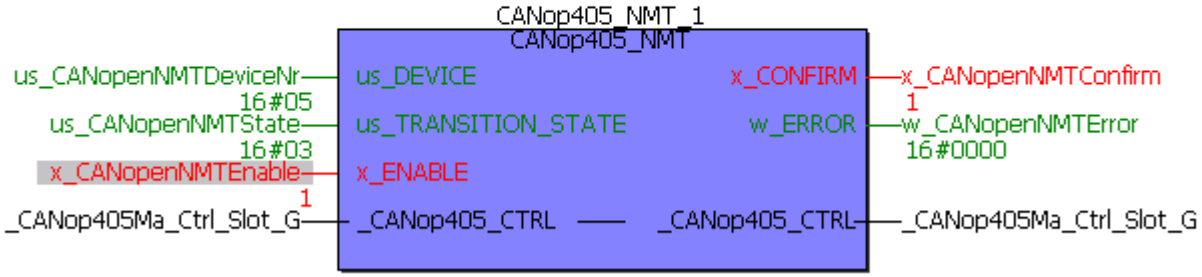


Abbildung 74: CANopen-Slave in Zustand PRE-OPERATIONAL schalten

Ermitteln wir nun über den FB CANop405_COB_ID die COB-ID für das PDO. Wir implementieren diesen Baustein in der vorhandenen Initialisierungstask. An us_DEVICE wird die Knotennummer 5 und an us_PDO_NR die 2 (es soll das zweite Empfangs-PDO ge-

schrieben werden) angeschlossen. Da dieses PDO vom Optionsmodul BM4-O-CAN-03 empfangen wird, also ein RxPDO ist, müssen wir den Eingang x_RX mit 1 beschalten. Die COB-ID geben wir auf einer globalen Variable aus. Nach Übersetzen und Senden des Projektes erhalten wir in der Online-Darstellung den Wert 305h für die COB-ID.

```
(* compute the COB-IDs for the PDOs to write *)
```

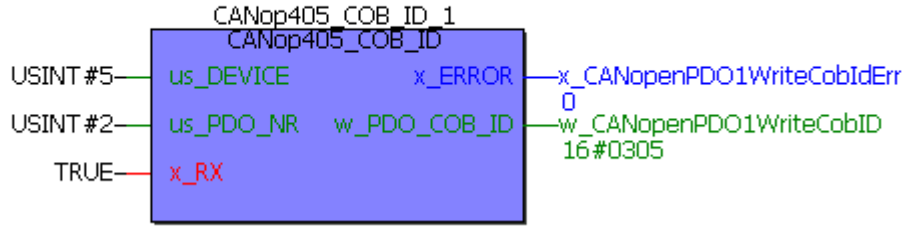


Abbildung 75: COB-ID für zweites Empfangs PDO mit dem FB CANop405_COB_ID ermitteln

Jetzt überprüfen wir die Einstellungen für dieses PDO auf dem Optionsmodul BM4-O-CAN-03. Das erste Empfangs-PDO wird im Objekt 1400h für die Kommunikation konfiguriert. Demnach finden wir die Konfiguration für das zweite Empfangs-PDO im Objekt 1401h. Wir verwenden den bereits vorhandenen FB CANop405_SDO1_READ, nur mit der Beschaltung für das Objekt 1401h. Wir verwenden diesmal Variablen und keine Konstanten um Online einfach auf das nächste Objekt umschalten zu können.

```
(* Read object via SDO from device nr. 5 *)
```

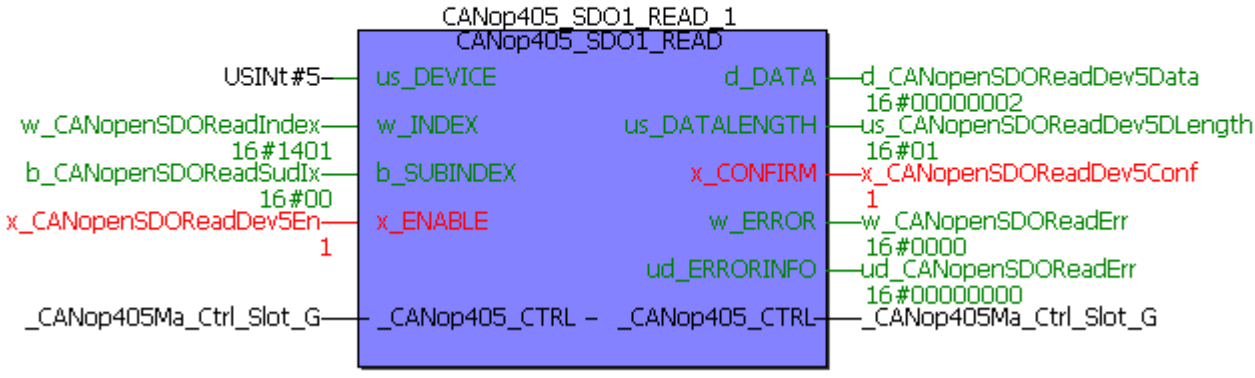


Abbildung 76: Konfiguration des zweiten Empfangs PDO des CANopen-Slave auslesen - Subindex 0

Für Subindex 0 erhalten wir den Wert 2. Das Objekt 1401h hat also gültige Einträge in den Subindizes 0 - 2. Da uns die eingestellte COB-ID und die Gültigkeit des PDO (beide Subindex 1), sowie der Übertragungstyp (Subindex 2) interessiert, müssen wir Subindex 1 und 2 lesen. Wir erhalten:

```
(* Read object via SDO from device nr. 5 *)
```

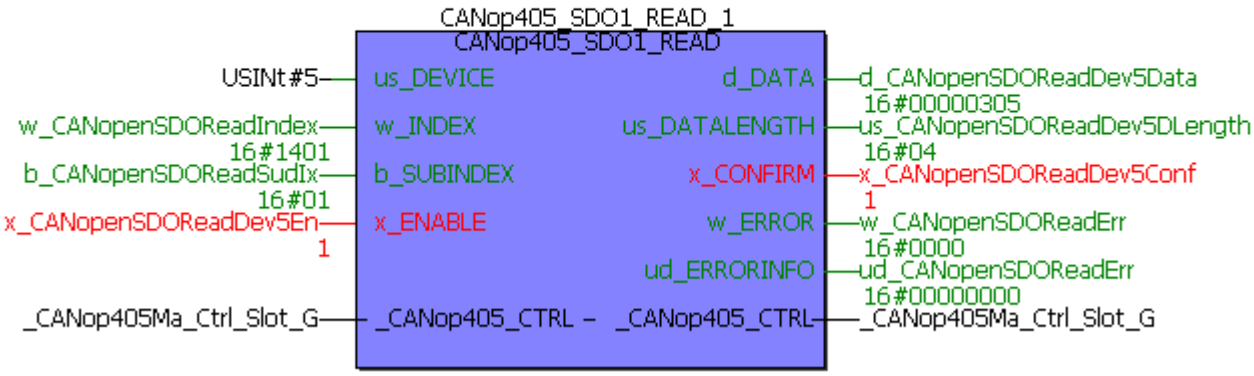


Abbildung 77: Konfiguration des zweiten Empfangs PDO des CANopen-Slave auslesen - Subindex 1

Der Wert 305h bedeutet, dass die COB-ID dieses PDO 305h ist, eine 11-Bit CAN-ID verwendet wird und dass das PDO gültig ist. Der Wert von Subindex 1 ist also richtig.

```
(* Read object via SDO from device nr. 5 *)
```

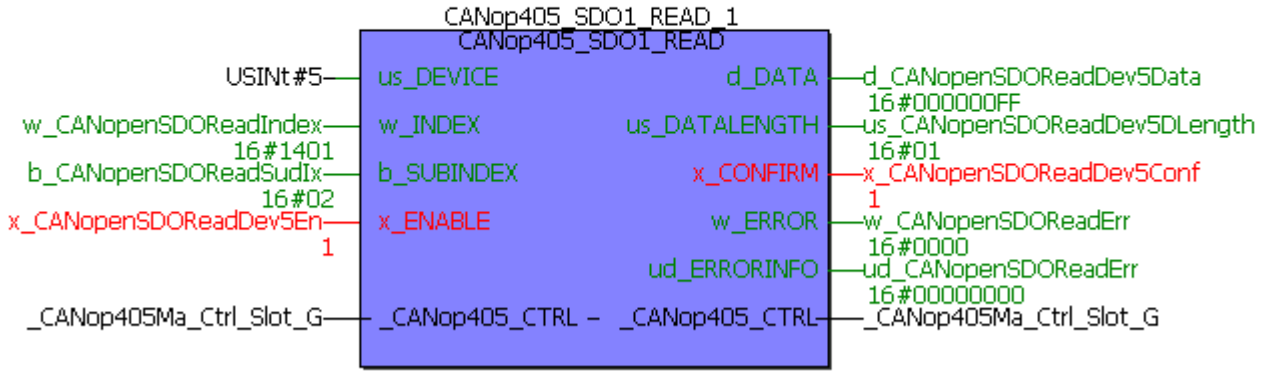


Abbildung 78: Konfiguration des zweiten Empfangs PDO des CANopen-Slave auslesen - Subindex 2

Der Wert FFh bedeutet, dass der Übertragungstyp für dieses PDO 255 ist, also jedes PDO mit passender COB-ID (305h) wird bei Empfang übernommen. Auch dieser Wert ist richtig. Sollten Sie an dieser Stelle andere Werte für die Objektinhalte erhalten, so müssen Sie diese mit einem der FBs CANop405_SDOx_WRITE (x = 1 bis 8) ändern.

Jetzt überprüfen wir das Mapping für das zweite Empfangs-PDO in Objekt 1601h. Wir wollen den Parameter 1171 zuerst mappen. Für die Einträge in 1601h sollten wir daher unter Berücksichtigung von Objekt-Index, Subindex und Datenlänge folgende Werte erhalten:

Objekt 1601h

Subindex 0:	2h	zwei gemappte Werte
Subindex 1:	6042 00 10h	Index, Subindex, Länge für P1171
Subindex 2:	4410 00 20h	Index, Subindex, Länge für P1040

Überprüfen wir die Werte von Objekt 1601h nach obigem Verfahren mit dem FB CANop405_SDO1_READ. Im Beispielprojekt erhalten wir die folgenden Werte:

Objekt 1601h

Subindex 0:	2h	zwei gemappte Werte
Subindex 1:	6040 00 10h	
Subindex 2:	6060 00 08h	

Bis auf Subindex 0 stimmt keiner der Werte. Wir müssen das Mapping ändern. Dazu verwenden wir den FB CANop405_SDO1_WRITE. Anstelle der Konstanten schließen wir wieder globale Variablen an um Online einfach auf das nächste Objekt umschalten zu können.

Zunächst löschen wir das Mapping des zweiten Empfangs-PDO indem wir den Wert von Subindex 0 auf null setzen.

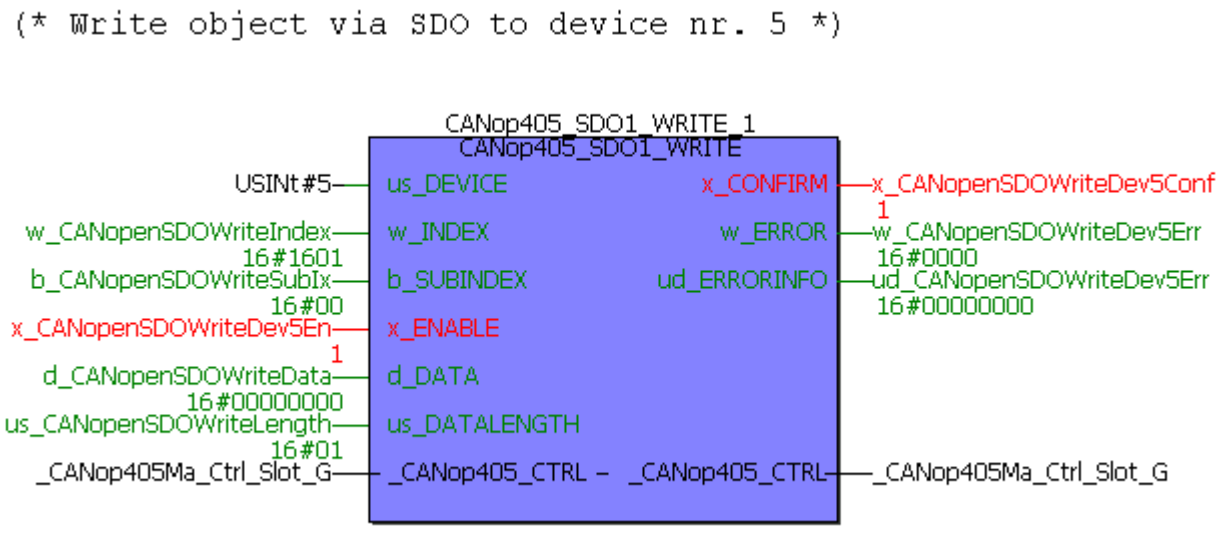


Abbildung 79: Mapping des zweiten Empfangs PDO löschen

Jetzt tragen wir das Mapping des ersten Objektes P1171 in Subindex 1 ein:

```
(* Write object via SDO to device nr. 5 *)
```

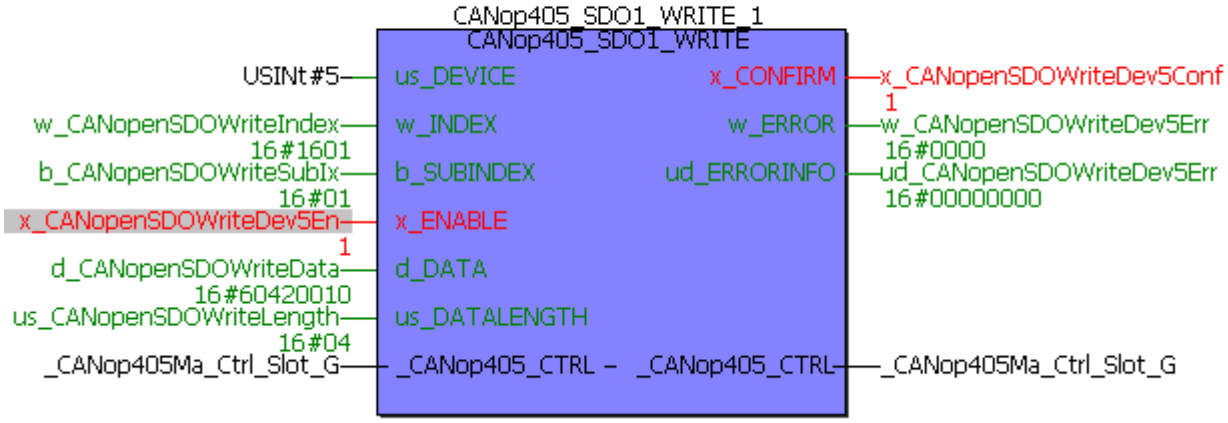


Abbildung 80: Erstes gemappte Objekt des zweiten Empfangs PDO eintragen

und anschließend das Mapping von Objekt P1040 in Subindex 2:

```
(* Write object via SDO to device nr. 5 *)
```

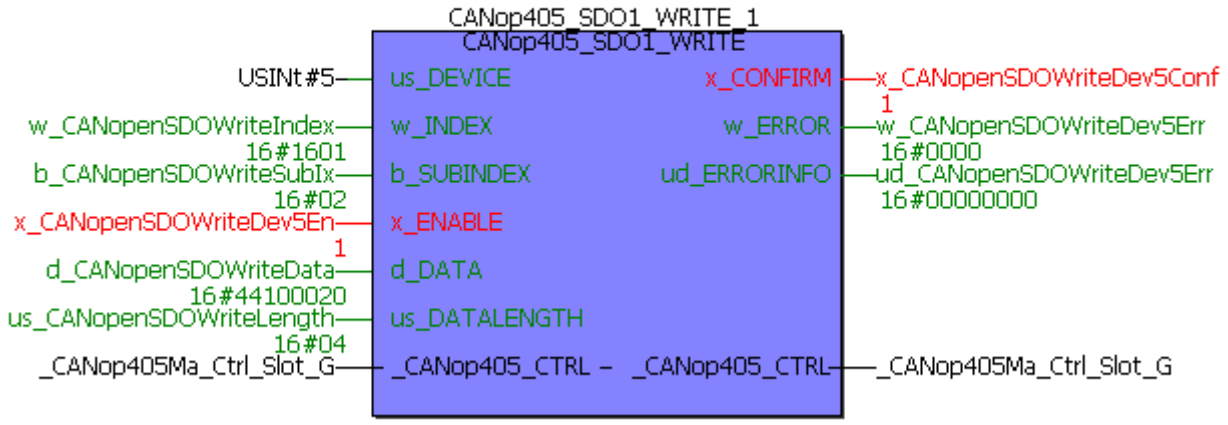


Abbildung 81: Zweites gemappte Objekt des zweiten Empfangs PDO eintragen

Es sind jetzt alle Objekte eingetragen und wir müssen das Mapping wieder freigeben indem die Anzahl der gemappten Objekte in Subindex 0 eingetragen wird:

```
(* Write object via SDO to device nr. 5 *)
```

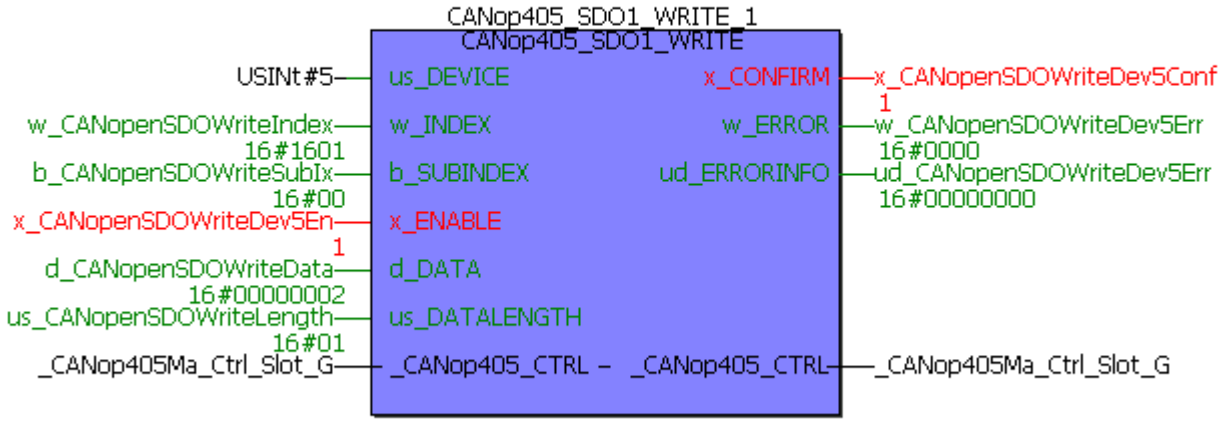


Abbildung 82: Anzahl der gemappten Objekte des zweiten Empfangs PDO eintragen

Das Optionsmodul BM4-O-CAN-03 bietet die Möglichkeit, die eingestellten Kommunikationsparameter zu speichern. Beim nächsten Aus-/Einschalten des b maXX 4400 Gerätes werden die gespeicherten Kommunikationsparameter dann automatisch übernommen. Das Speichern wird durch ein Schreiben des Objektes 1010h, Subindex 2 mit Wert 65766173h ausgelöst. Dies wollen wir jetzt durchführen:

```
(* Write object via SDO to device nr. 5 *)
```

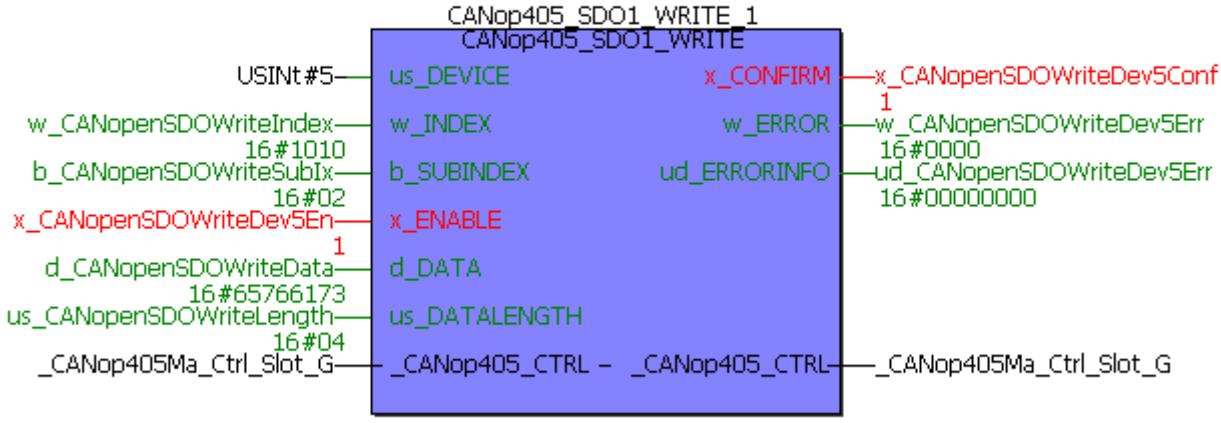


Abbildung 83: Speichern der Kommunikationsparameter des CANopen-Slave

Das zweite Empfangs-PDO auf dem Optionsmodul BM4-O-CAN-03 ist jetzt fertig konfiguriert und wir können zum eigentlichen Senden des PDO übergehen.

d) Senden eines PDO zu einem Netzwerkknoten

Das Senden eines PDO erfolgt über die Instanzen des Funktionsbausteins CANop405_PDO_WRITE. Mit diesen Funktionsbausteinen können gleichzeitig bis zu 40 PDO Aufträge gestartet werden. Die Zahl 40 entspricht dabei der Anzahl der Sendefächer auf dem Optionsmodul CANopen-Master, welche für diesen Zweck zur Verfügung stehen. Ein Sendefach darf nicht doppelt verwendet werden. Es ist jedoch möglich nacheinander ein Sendefach für verschiedene PDOs zu verwenden. Damit ein Netzwerkknoten ein PDO übernimmt, muss sich dieser im Zustand OPERATIONAL befinden.

Um den Funktionsbaustein CANop405_PDO_WRITE 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 zyklischen IRP Task aufgerufen werden.
- Platzieren Sie einen oder mehrere der FBs CANop405_PDO_WRITE nach Ihrem Bedarf in dieser POE.
- Beschalten Sie die Funktionsbausteine mit Variablen vom richtigem Datentyp und Belegen Sie diese mit den gewünschten Werten. Am Eingang us_PDO_NR geben Sie die Nummer des von Ihnen gewählten Sendefachs an. w_PDO_COB_ID ist die COB-ID des PDO, welche Sie schreiben wollen. us_DATALENGTH bezeichnet die gültigen Bytes innerhalb des PDO. Sie ist die Summe aller Bytes der gemappten Objekte. An d_DATA0 und d_DATA1 werden die Daten für die Objekte angeschlossen. Falls Sie ein Remote-PDO anfordern wollen, setzen Sie den Eingang x_REMOTE auf 1.
- Legen Sie eine zyklische Bypass IRP Task an, falls eine solche in Ihrem Projekt noch nicht vorhanden sein sollte. Bei geringen zeitlichen Anforderungen an das PDO können Sie auch eine einfache zyklische Task mit Priorität ihrer Wahl verwenden. Binden Sie die erstellte POE in diese Task ein.
- Übersetzen Sie das Projekt und laden Sie es als (Boot-)Projekt auf die PLC. Starten Sie die PLC.
- Führen Sie gegebenenfalls die Konfiguration der Netzwerkknoten durch und bringen Sie diese in den Zustand OPERATIONAL.

Mit x_ENABLE = 1 wird das PDO gesendet. Der FB CANop405_PDO_WRITE meldet eine erfolgreiche Durchführung mit x_CONFIRM = 1 und w_ERROR = 16#0. Da das Schreiben eines PDO vom Slave nicht bestätigt wird, bedeutet x_CONFIRM = 1 nur, dass das PDO vom Optionsmodul CANopen-Master gesendet wurde, aber nicht ob diese auch vom Slave übernommen wurde.

e) Beispiel: PDO senden

Wir führen das Beispiel fort. Das zweite Empfangs-PDO des Optionsmodul BM4-O-CAN-03 ist konfiguriert und wir können zum eigentlichen Datenaustausch über PDOs übergehen. Zunächst muss der Netzwerkknoten in den Zustand OPERATIONAL gesetzt werden. Wir verwenden dazu wieder den FB CANop405_NMT:

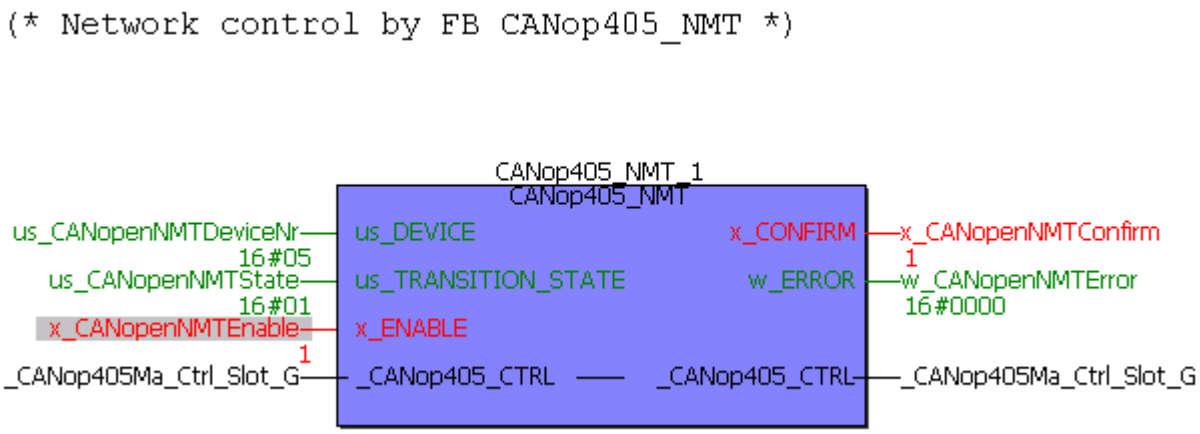


Abbildung 84: CANopen-Slave in Zustand OPERATIONAL schalten

Anschließend legen wir eine POE für die spätere Einbindung in eine IRP-Task an. In dieser POE platzieren wir den FB CANop405_PDO_WRITE und beschalten diesen mit entsprechenden Konstanten und Variablen. Wie zuvor festgelegt, soll das erste Sendefach verwendet werde, also ist us_PDO_NR = 1. Die COB-ID holen wir uns vom FB CANop405_COB_ID aus der Initialisierungstask über die globale Variable w_CANopenPDO1WriteCobID. Die Datenlänge von P1171 ist zwei Byte, von P1040 vier Byte, us_DATALENGTH bekommt demnach den Wert USINT#6. Durch das Mapping liegt P1171 im ersten Wort von d_DATA0 und P1040 im zweiten Wort von d_DATA0, sowie im ersten Wort von d_DATA1. Damit erkennbarer Datenverkehr stattfindet sollen beide Parameter einfach hochgezählt bzw. herabgezählt werden, wobei P1171 noch auf max. 3000 U/min begrenzt wird. Der fertige Code sieht dann wie folgt aus:

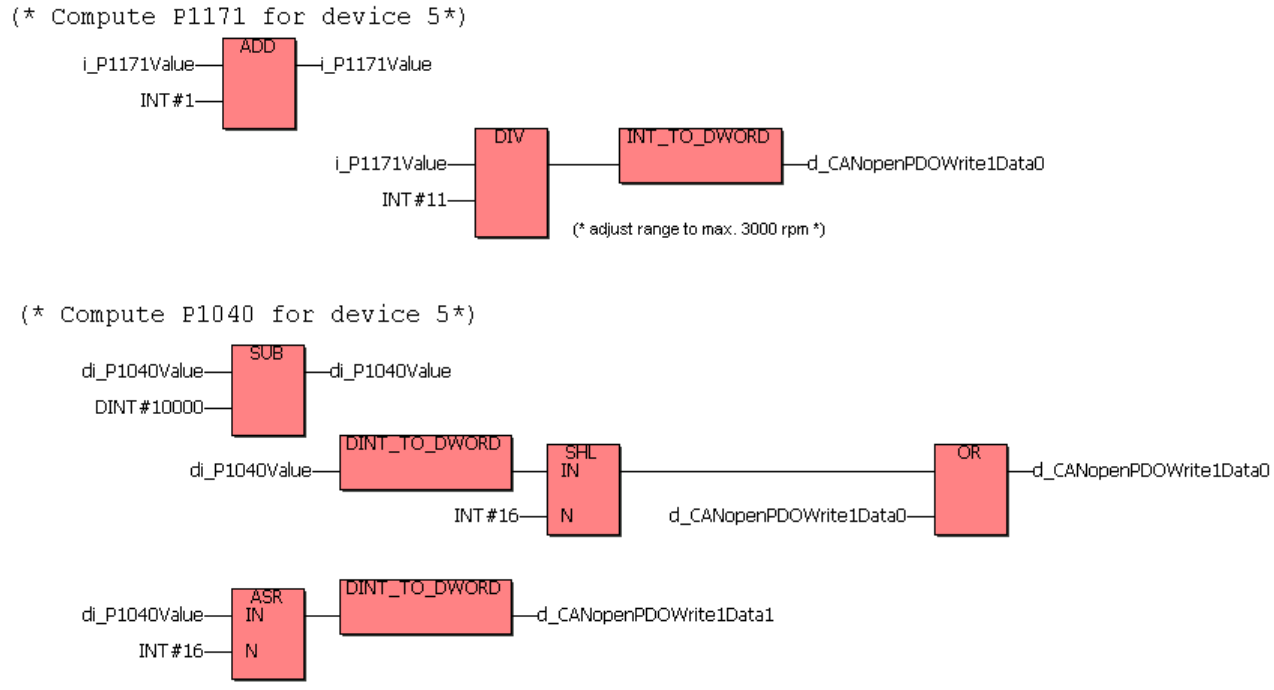


Abbildung 85: Verändern der zu schreibenden Daten

```
(* Send PDO to device 5 *)
```

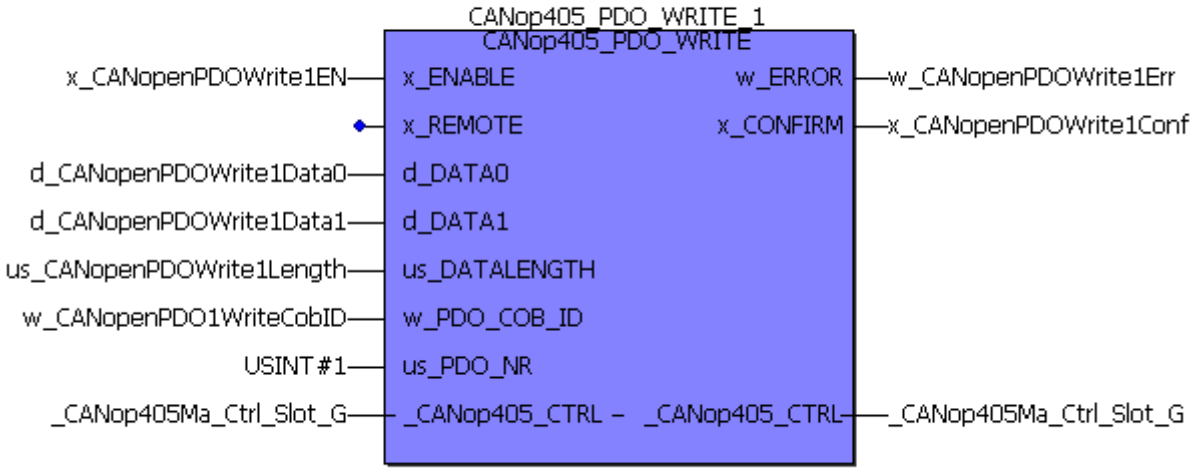


Abbildung 86: Daten mit dem FB CANop405_PDO_WRITE schreiben

Um den FB CANop405_PDO_WRITE alle 2 ms aufrufen zu können, benötigen wir noch eine Timer-IRP-Task. Zu deren Einrichtung benötigen wird den FB INTR_SET aus der Bibliothek SYSTEM_PLC01_20bd00 (oder höher). Den FB INTR_SET konfigurieren wir für einen 2 ms CPU Timer 1 Interrupt und implementieren diesen Funktionsbaustein in der CANopen-Initialisierungs POE vor dem FB CANop405_PDO_INIT:

```
(* Initialise the Timer-IRP with 2ms *)
```

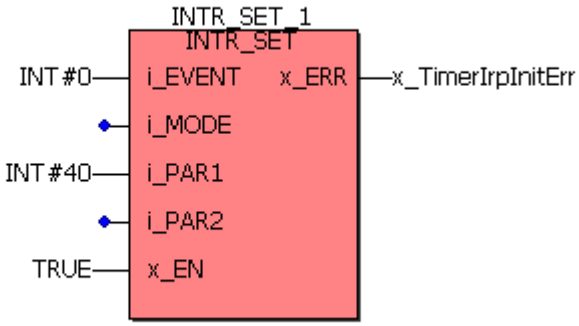


Abbildung 87: 2 ms Timer-IRP-Task zur Datenübertragung einrichten

Jetzt kann die entsprechende Event-Task mit Event "CPU-Timer 1" und Bypass-Betrieb eingerichtet und dieser Event-Task die POE mit dem FB CANop405_PDO_WRITE zugeordnet werden. Übersetzen Sie das Projekt und senden Sie dieses als Projekt (nicht als Boot-Projekt) an die PLC. Starten Sie das Projekt. Sollten sie ein Boot-Projekt gesendet und das b maXX 4400 Gerät neu gestartet haben verlieren Sie die zuvor eingestellte PDO-Konfiguration, falls Sie diese nicht gespeichert haben. Sie müssen in diesem Fall die Konfiguration neu durchführen, da das Optionsmodul BM4-O-CAN-03 beim Start die gespeicherte PDO-Konfiguration übernimmt. Bei einem Neustart muss das Optionsmodul BM4-O-CAN-03 auch wieder in den Zustand OPERATIONAL gesetzt werden.

Mit x_ENABLE = 1 am FB CANop405_PDO_WRITE kann die Datenübertragung gestartet werden:

```
(* Send PDO to device 5 *)
```

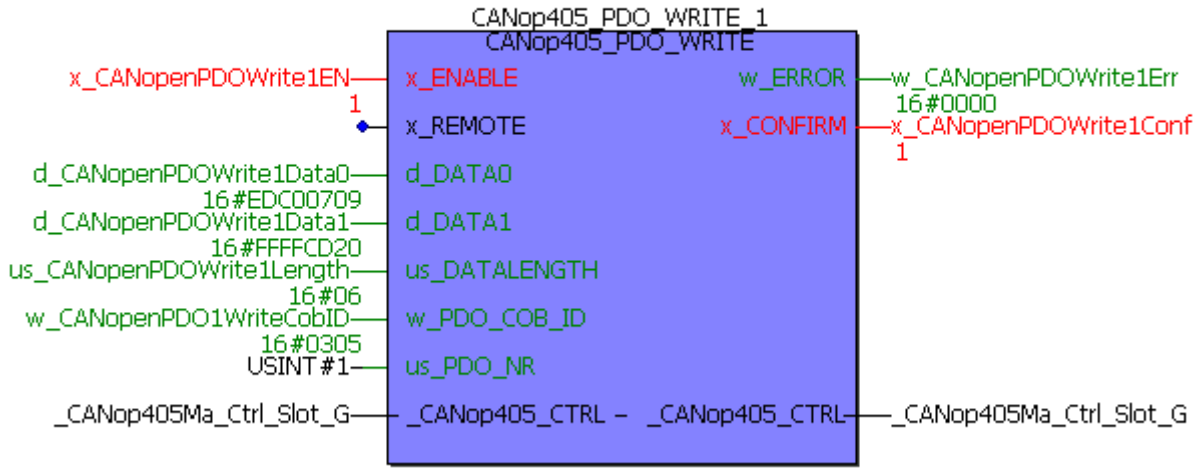


Abbildung 88: Daten mit dem FB CANop405_PDO_WRITE schreiben - Online

Ein korrektes Absenden des PDO wird mit x_CONFIRM = 1 bestätigt. In WinBASS II sollten Sie eine Änderung der Parameter 1171 und 1040 erkennen können.

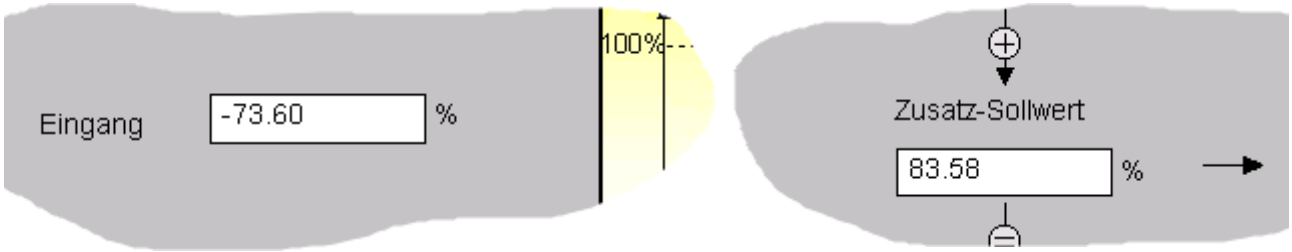


Abbildung 89: Geschriebene Daten in WinBASS II

Beachten Sie, dass dies Momentaufnahmen im Online-Modus sind und Ihre Werte sich unterscheiden können.

4.4.9.6 Objekte über PDOs lesen

a) Mechanismus auf dem Optionsmodul CANopen-Master

Das Lesen von Objekten über PDOs erfolgt beim Optionsmodul CANopen-Master mit dem FB CANop405_PDO_READ. Der dabei ablaufende Mechanismus sieht wie folgt aus:

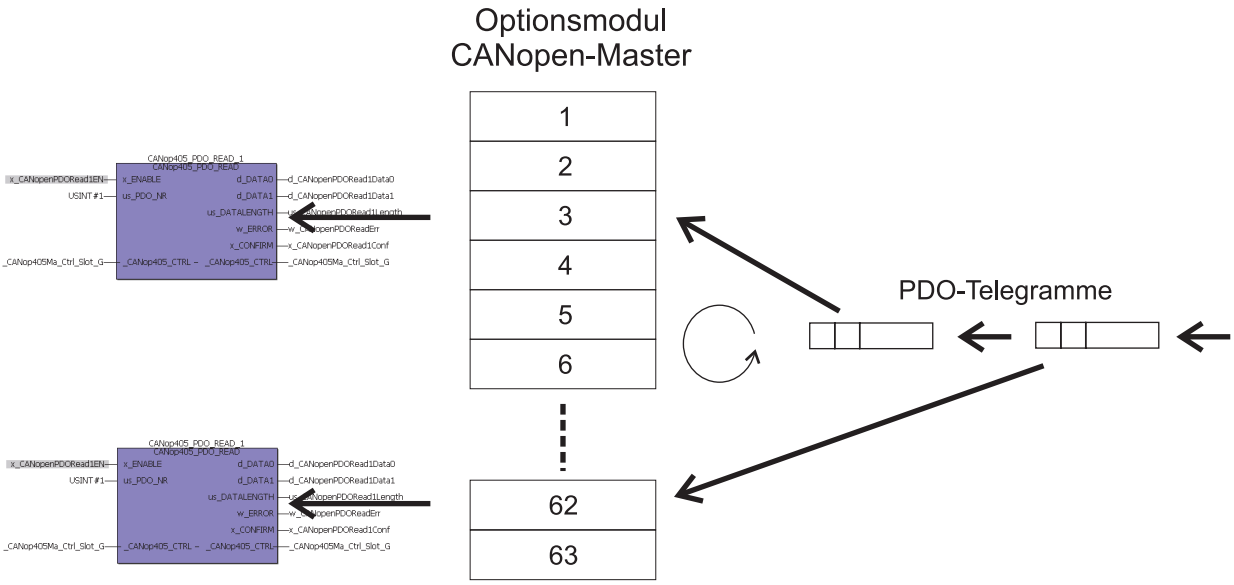


Abbildung 90: Mechanismus auf dem Optionsmodul für PDO lesen

Jedem zu empfangenden PDO wird ein FB CANop405_PDO_READ zugeordnet indem von diesem Funktionsbaustein entsprechende Instanzen gebildet werden. Jedem dieser Funktionsbausteine muss über einen Eingang am Funktionsbaustein ein Empfangsfach auf dem Optionsmodul CANopen-Master zugeordnet werden. Insgesamt stehen 63 Empfangsfächer zur Verfügung. Es können also 63 Empfangsaufträge für PDOs gleichzeitig ausgewertet werden. Bei Eintreffen eines PDO trägt das Optionsmodul diese in ein Empfangsfach ein. Das richtige Empfangsfach wird vom Optionsmodul über die COB-ID des PDO ausgewählt. Hierzu muss in der Initialisierungsphase dem Optionsmodul mitgeteilt werden, welche COB-ID welchem Empfangsfach zugeordnet ist. Dies erfolgt mit dem FB CANop405_PDO_INIT. Eine Umkonfiguration ist während des Betriebs nicht mehr möglich. Der FB CANop405_PDO_READ prüft beim Aufruf, ob für ihn ein Eintrag in seinem zugeordneten Empfangsfach vorhanden ist.

HINWEIS

Es ist möglich, dass empfangene PDOs verloren gehen wenn mehr PDOs eintreffen als vom FB CANop405_PDO_READ ausgelesen werden oder wenn der FB CANop405_PDO_READ gerade beim Übernehmen eines PDO aus dem Empfangsfach ist. Um mögliche Verluste zu verringern sollte der FB CANop405_PDO_READ ein höheres Aufrufintervall haben, als Daten eintreffen können. Zudem sollten Mechanismen zum gesteuerten PDO-Transfer (z. B. SYNC oder Remote-Anforderung, siehe entsprechende Kapitel) angewandt werden.

b) Vorbereitung und Konfiguration

Bevor Sie ein PDO empfangen können müssen Sie einige Vorbereitungen treffen und Konfigurationseinstellungen auf dem Netzwerkknoten, welcher das PDO senden soll, durchführen. Gehen Sie dazu in folgenden Schritten vor:

Bestimmen Sie wichtige Daten:

- Bestimmen Sie die Nummer des Netzwerkknotens im CANopen-Netzwerk, welcher das PDO senden soll.
- Bestimmen Sie, welches PDO des Netzwerkknotens gesendet werden soll. Schlagen Sie hierzu in der Dokumentation zum Netzwerkknoten nach, um festzustellen wie viele PDOs dieser unterstützt.
- Bestimmen Sie ein noch nicht vergebenes Empfangsfach für das PDO. Sie sollten bei Empfangsfach 1 beginnen und ohne Lücken die nächsten Empfangsfächer vergeben.
- Legen Sie den Übertragungstyp des PDO fest. Beim Senden von PDOs durch einen Netzwerkknoten haben Sie verschiedene Möglichkeiten, informieren Sie sich dazu noch einmal in Kapitel ▹Die Konfiguration der Übertragungseigenschaften eines PDO◃ auf Seite 114, sowie in der Dokumentation zum Netzwerkknoten.
- Legen Sie jetzt die Objekte fest, welche in dem PDO gemapped sein sollen. Beachten Sie das Default-Mapping des Netzwerkknotens, eventuell können Sie es für Ihre Applikation verwenden.
- Legen Sie fest, in welchem Zeitraster das PDO ausgewertet werden soll. Entsprechend dem Zeitraster legen Sie später eine Task in PROPROG wt II an, um den Erhalt des PDO im Empfangsfach zu überprüfen. Das Zeitraster sollte an den Übertragungstyp des PDO angepasst sein. Beachten Sie bitte die maximale CANopen-Buslast. Bei einem Betrieb des CANopen mit 1 Mbit/s können Sie rechnerisch 7 PDOs mit je 8 Byte Daten pro ms empfangen. Das bedingt aber, dass kein weiterer Datenverkehr stattfindet. Dies ist aber sehr unwahrscheinlich, da Sie vermutlich auch PDOs senden und weitere CANopen-Funktionen wie Node Guarding und Sync anwenden wollen. Die tatsächlich maximal mögliche Buslast wird also weitgehend durch Ihre Applikation bestimmt. Tatsächlich werden also weniger als 7 PDOs pro ms empfangen werden können.

Wiederholen Sie diese Schritte für alle PDOs, welche Sie empfangen wollen.

Programmieren Sie die Konfiguration und führen Sie diese durch:

- Initialisieren Sie das Empfangen von PDOs mit dem FB CANop405_PDO_INIT. Diesen Funktionsbaustein müssen Sie vor dem FB CANop405_INIT platzieren. Am FB CANop405_PDO_INIT muss die Nummer des höchsten verwendeten Empfangsfachs angegeben werden. Maximal können 63 Empfangsfächer verwendet werden. An diesem FB führen Sie auch die Zuordnung der COB-IDs der PDOs zu den Empfangsfächern durch. Hierzu wird ein Array verwendet, dessen jeweilige Index-Stelle einem Empfangsfach entspricht und dessen Inhalt die COB-ID ist. Maximal können Sie bis Index 63 Werte eintragen. Bei Einträgen darüber und an Index 0 wird eine Fehlermeldung erzeugt.
- Zur Ermittlung der COB-ID des PDO können Sie den Funktionsbaustein CANop405_COB_ID verwenden. Dieser FB errechnet aus der von Ihnen ermittelten Knotennummer und PDO-Nummer die COB-ID. Der FB wird somit vor dem FB CANop405_PDO_INIT aufgerufen.
- Setzen Sie den Netzwerkknoten über den FB CANop405_NMT in den Zustand PRE-OPERATIONAL.
- Verwenden Sie die in Kapitel "Objekte mit SDO schreiben" beschriebenen Funktionsbausteine CANop405_SDO1_WRITE bis CANop405_SDO8_WRITE um die Konfigurationsdaten (Objekte 1800h, 1801h, etc. und 1A00, 1A01, etc.) des Netzwerkknotens zu überprüfen und einzustellen.

c) Beispiel: Konfiguration PDO empfangen

Das Optionsmodul BM4-O-CAN-03 soll zyklisch alle 5 ms das Statuswort P301 und den Drehzahl-Zusatz-Sollwert P1040 senden. Wir wollen das Sende-PDO 1 des Optionsmodul BM4-O-CAN-03 verwenden. Auf der Seite des Optionsmodul CANopen-Master soll alle 2 ms der Empfang des PDO geprüft werden. Da bisher noch kein PDO zum Empfangen eingerichtet wurde verwenden wir Empfangsfach 1. Schließlich kennen wir noch die Knotennummer: 5. Für die Parameter 301 und 1040 benötigen wir noch ein paar Daten aus dem Objektverzeichnis um das Mapping konfigurieren zu können:

Parameter	Index	Subindex	Datenlänge
301 Statuswort	6041	0	16 Bit
1040 Drehzahl-Zusatz-Sollwert	4410	0	32 Bit

Die Normierung von P1040 über CANopen ist: -100% bis +100% Drehzahl entsprechen -2^{30} bis $+2^{30}$.

Zunächst muss der FB CANop405_PDO_INIT konfiguriert werden. Dieser FB ist bereits in der Initialisierungs-POE durch die Konfiguration des Sendens vorhanden. An us_MAX_PDO_READ geben wir die Nummer des letzten verwendeten Empfangsfachs an. Da wir nur das erste Empfangsfach verwenden ist dies USINT#1. Dem Empfangsfach 1 muss auch noch eine COB-ID zugeordnet werden. Die Zuordnung erfolgt über den Eingang a_PDO_READ_COB_ID. Hier ist ein Array vom Datentyp WORD_64_BMARRAY anzuschließen, welches die COB-ID enthält. Da wir Empfangsfach 1 verwenden ist die COB-ID an Index 1 des Arrays einzutragen. Für die Generierung der COB-ID verwenden wir den FB CANop405_COB_ID. An us_DEVICE wird die Knotennummer 5 und an us_PDO_NR die 1 angeschlossen. Da dieses PDO vom Optionsmodul BM4-O-CAN-03 gesendet wird, also ein TxPDO ist, müssen wir den Eingang x_RX mit 0 beschalten.

Nach Übersetzen und Download des Projektes sollte die Initialisierung im Online-Modus folgende Darstellung haben:

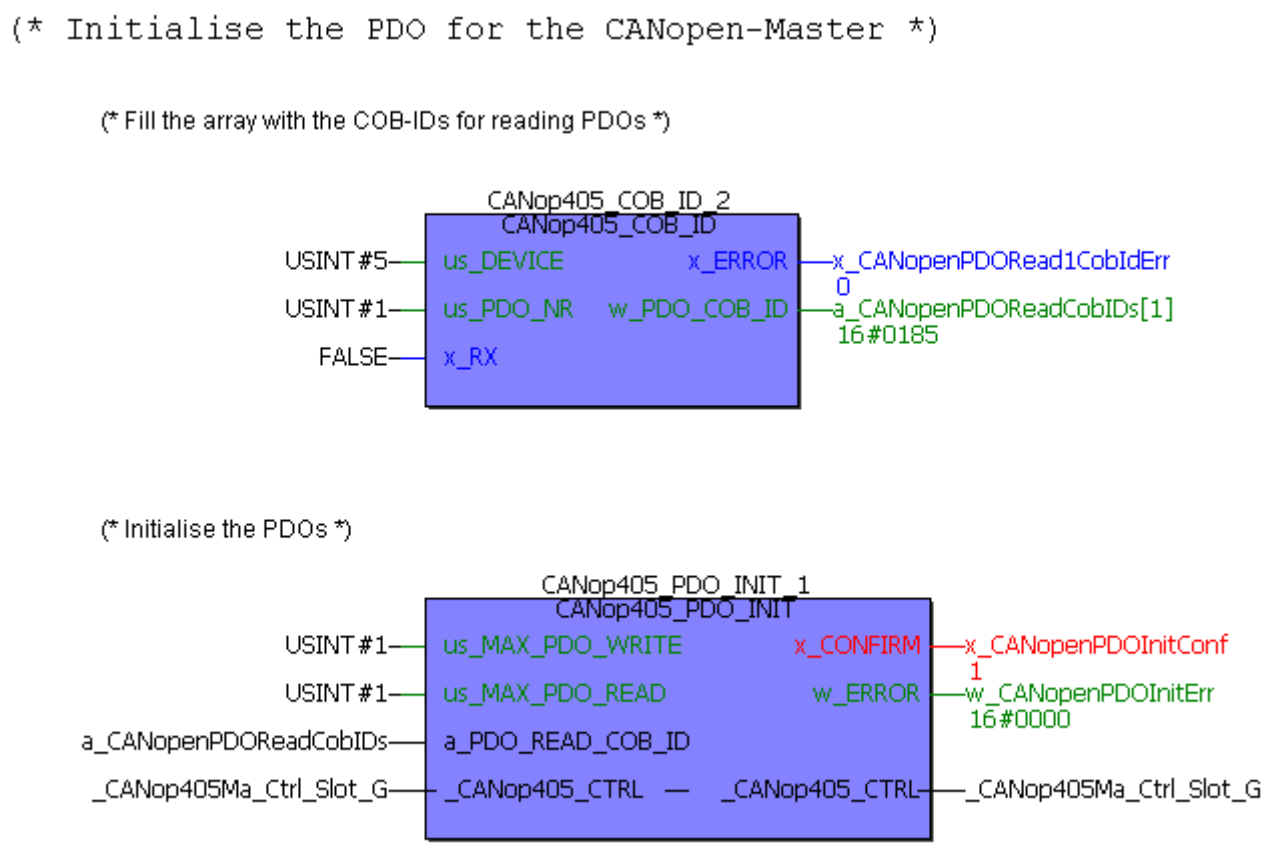


Abbildung 91: PDO Übertragung mit dem FB CANop405_PDO_INIT initialisieren

Für die COB-ID erhalten wir den Wert 185h. Bringen Sie das Optionsmodul BM4-O-CAN-03 mit dem FB CANop405_NMT in den Zustand PRE-OPERATIONAL, um die Konfiguration des PDO durchführen zu können.

Jetzt überprüfen wir die Einstellungen für dieses PDO auf dem Optionsmodul BM4-O-CAN-03. Das erste Sende-PDO wird im Objekt 1800h für die Kommunikation konfiguriert. Wir verwenden den bereits vorhandenen FB CANop405_SDO1_READ, nur mit der Beschaltung für das Objekt 1800h.

```
(* Read object via SDO from device nr. 5 *)
```

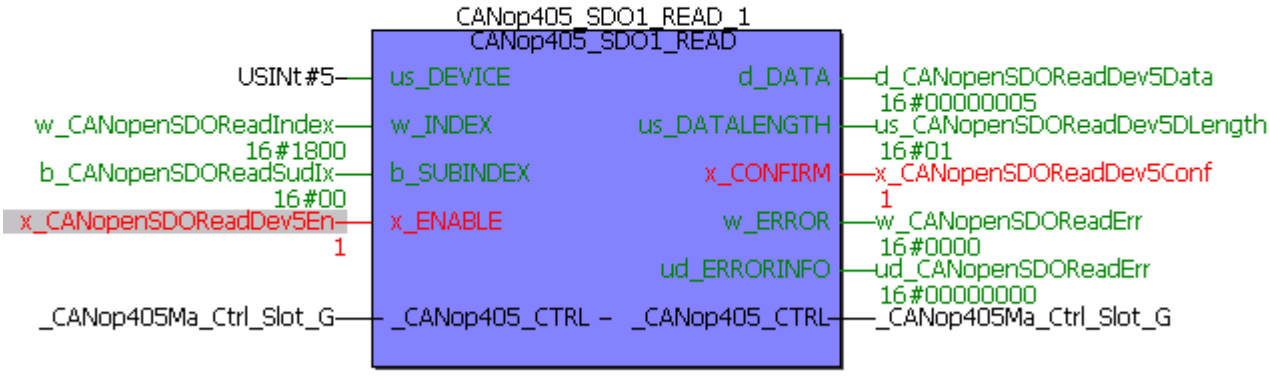


Abbildung 92: Konfiguration des ersten Sende PDO des CANopen-Slave auslesen - Subindex 0

Für Subindex 0 erhalten wir den Wert 5. Das Objekt 1800h hat also gültige Einträge in den Subindizes 0 - 5. Da uns die eingestellte COB-ID und die Gültigkeit des PDO (beide Subindex 1), sowie der Übertragungstyp (Subindex 2) und die Übertragungszeit (Subindex 5) interessieren, müssen wir Subindex 1, 2 und 5 lesen. Nach dem gleichen Verfahren wie für Subindex 0 erhalten wir folgende Werte:

Objekt 1800h

Subindex 0:	5h	fünf gültige Subindex Einträge
Subindex 1:	185h	COB-ID des ersten Sende-PDO
Subindex 2:	FFh	Übertragungstyp 255
Subindex 5:	0h	Timerwert für Übertragungszeit

Der Wert für die COB-ID dieses PDO ist 185h, eine 11-Bit CAN-ID wird verwendet und das PDO ist gültig. Der Wert von Subindex 1 ist somit richtig. Der Wert FFh von Subindex 2 bedeutet, dass der Übertragungstyp für dieses PDO 255 ist, also das PDO bei einer Wertänderung eines gemappten Objekts übertragen wird. Diesen Wert müssen wir mit einem der FB CANop405_SDO1_WRITE bis CANop405_SDO8_WRITE auf 254 ändern, da wir eine Zeit-Event gesteuerte Übertragung benötigen:

```
(* Write object via SDO to device nr. 5 *)
```

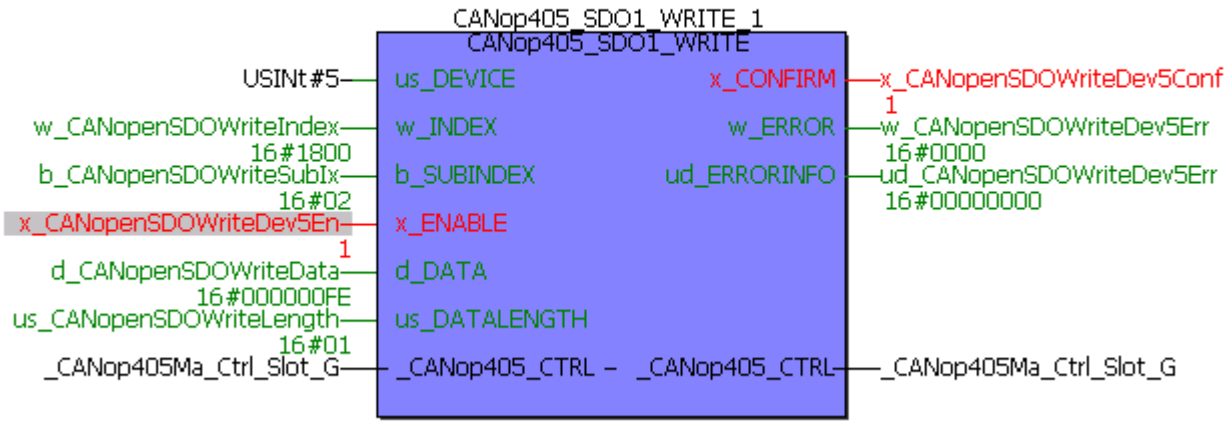


Abbildung 93: Konfiguration des ersten Sende PDO des CANopen-Slave schreiben - Subindex 2

Subindex 5 enthält den Timerwert für den Zeit-Event. Es ist 0h eingetragen, wir wollen jedoch alle 5 ms übertragen. Subindex 0 muss also auf den Wert 5 ms geändert werden:

```
(* Write object via SDO to device nr. 5 *)
```

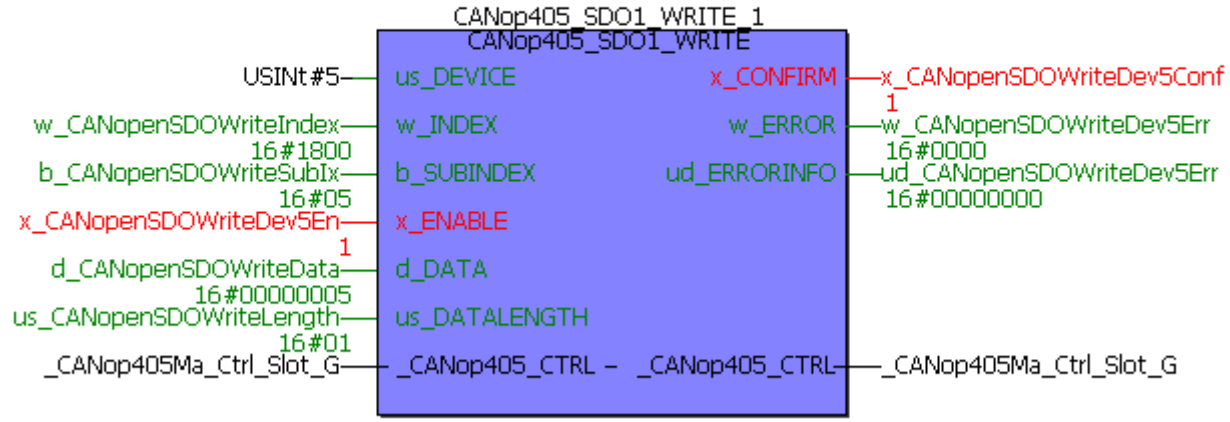


Abbildung 94: Konfiguration des ersten Sende PDO des CANopen-Slave schreiben - Subindex 5

Jetzt überprüfen wir das Mapping für das erste Sende-PDO in Objekt 1A00h. Wir wollen den Parameter 301 zuerst mappen. Für die Einträge in 1A00h sollten wir daher unter Berücksichtigung von Objekt-Index, Subindex und Datenlänge folgende Werte erhalten:

Objekt 1A00h

Subindex 0:	2h	zwei gemappte Werte
Subindex 1:	6041 00 10h	Index, Subindex, Länge für P301
Subindex 2:	4410 00 20h	Index, Subindex, Länge für P1040

Überprüfen wir die Werte von Objekt 1A00h nach obigem Verfahren mit dem FB CANop405_SDO1_READ. Im Beispielprojekt erhalten wir die folgenden Werte:

Objekt 1A00h

Subindex 0:	1h	ein gemappter Wert
Subindex 1:	6041 00 10h	

Bis auf Subindex 1 stimmt keiner der Werte. Wir müssen das Mapping ändern. Dazu verwenden wir wieder den FB CANop405_SDO1_WRITE.

Zunächst muss das Mapping des ersten Sende-PDO gelöscht werden, indem wir den Wert von Subindex 0 auf null setzen. Tatsächlich wird das Mapping nicht gelöscht, sondern nur für dieses PDO ignoriert. Der für uns brauchbare Wert für das Statuswort von Subindex 1 bleibt erhalten. Wir müssen also nach dem Löschen von Subindex 0 nur noch das Mapping in Subindex 2 einstellen:

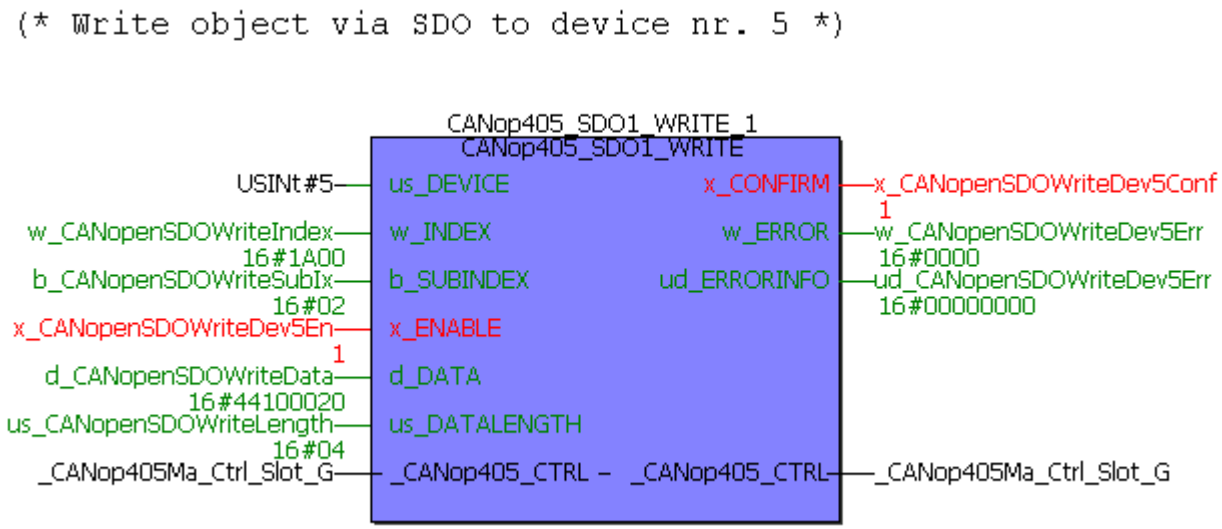


Abbildung 95: Zweites gemapptes Objekt des zweiten Sende PDO eintragen

Es sind jetzt alle Objekte eingetragen und wir müssen das Mapping wieder freigeben indem die Anzahl der gemappten Objekte in Subindex 0 eingetragen wird. Subindex 0 von Objekt 1A00h ist nach dem gleichem Verfahren mit dem Wert 2 zu beschreiben. Schließlich speichern wir die eingestellten Kommunikationsparameter wieder, indem Objekt 1010h, Subindex 2 mit dem Wert 65766173h über ein SDO beschrieben wird.

Das erste Sende-PDO auf dem Optionsmodul BM4-O-CAN-03 ist jetzt fertig konfiguriert und wir können zum eigentlichen Empfangen des PDO übergehen.

d) Empfangen eines PDO von einem Netzwerkknoten

Das Empfangen eines PDO erfolgt über die Instanzen des Funktionsbaustein CANop405_PDO_READ. Mit diesen Funktionsbausteinen können gleichzeitig bis zu 63 PDO Telegramme empfangen werden. Die Zahl 63 entspricht dabei der Anzahl der Empfangsfächer auf dem Optionsmodul CANopen-Master, welche für diesen Zweck zur Verfügung stehen. Ein Empfangsfach darf nicht doppelt verwendet werden. Damit ein Netzwerkknoten ein PDO senden kann, muss sich dieser im Zustand OPERATIONAL befinden.

Um den Funktionsbaustein CANop405_PDO_READ 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 zyklischen IRP Task aufgerufen werden.
- Platzieren Sie einen oder mehrere der FBs CANop405_PDO_READ nach Ihrem Bedarf in dieser POE.
- Beschalten Sie die Funktionsbausteine mit Variablen vom richtigem Datentyp und Belegen Sie diese mit den gewünschten Werten. Am Eingang us_PDO_NR geben Sie die Nummer des von Ihnen gewählten Empfangsfachs an. Mit dem FB CANop405_PDO_INIT haben Sie bereits in der Initialisierung die Zuordnung zwischen COB_ID und Empfangsfach getroffen. us_DATALENGTH bezeichnet die gültigen Bytes des empfangenen PDO. Es ist die Summe aller Bytes der gemappten Objekte. An d_DATA0 und d_DATA1 werden die Daten der empfangenen Objekte ausgegeben. Sie erhalten mit dem FB CANop405_PDO_READ auch Sende-PDOs, welche Sie über eine Remote-Anforderung mit dem FB CANop405_PDO_WRITE angefordert haben.
- Legen Sie eine zyklische Bypass IRP Task an, falls eine solche in Ihrem Projekt noch nicht vorhanden sein sollte. Bei geringen zeitlichen Anforderungen an das PDO können Sie auch eine einfache zyklische Task mit Priorität ihrer Wahl verwenden. Binden Sie die erstellte POE in diese Task ein.
- Übersetzen Sie das Projekt und laden Sie es als (Boot-)Projekt auf die PLC. Starten Sie die PLC.
- Führen Sie gegebenenfalls die Konfiguration der Netzwerkknoten durch und bringen Sie diese in den Zustand OPERATIONAL.

Mit x_ENABLE = 1 kann das PDO empfangen werden. Der FB CANop405_PDO_READ meldet ein eingetroffenes PDO mit x_CONFIRM = 1 und w_ERROR = 16#0. x_CONFIRM wird im nächsten Zyklus wieder auf null gesetzt und es kann mit x_ENABLE = 1 das nächste PDO empfangen werden.

e) Beispiel: PDO empfangen

Wir führen das Beispiel fort. Das erste Sende-PDO des Optionsmodul BM4-O-CAN-03 ist konfiguriert und wir können zum eigentlichen Datenaustausch über PDOs übergehen. Zunächst muss der Netzwerkknoten in den Zustand OPERATIONAL gesetzt werden. Wir verwenden dazu wieder den FB CANop405_NMT:

```
(* Network control by FB CANop405_NMT *)
```

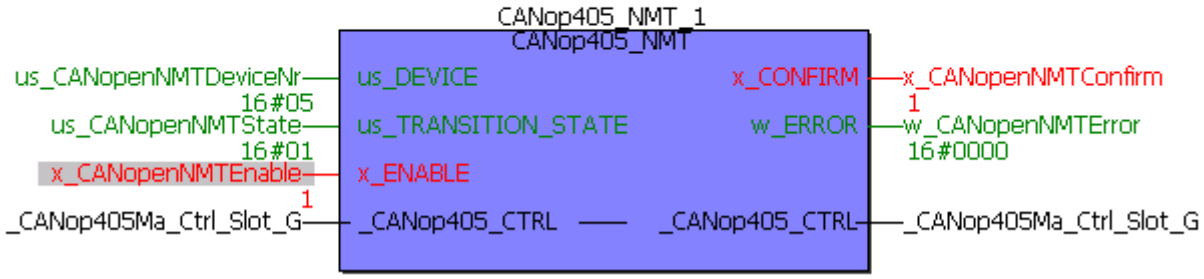


Abbildung 96: CANopen-Slave in Zustand OPERATIONAL schalten

Den FB CANop405_PDO_RERAD platzieren wir in der bereits vorhandenen POE mit dem FB CANop405_PDO_WRITE. Damit wird der FB CANop405_PDO_READ auch alle 2 ms aufgerufen. Wie zuvor festgelegt, soll das erste Empfangsfach verwendet werden, also ist us_PDO_NR = 1. Durch das Mapping liegt P301 im ersten Wort von d_DATA0 und P1040 im zweiten Wort von d_DATA0, sowie im ersten Wort von d_DATA1. Übersetzen Sie das Projekt und senden sie dieses als Projekt (nicht als Boot-Projekt) an die PLC. Starten Sie das Projekt. Sollten sie ein Boot-Projekt gesendet und das b maXX 4400 Gerät neu gestartet haben verlieren Sie die zuvor eingestellte PDO-Konfiguration, falls Sie diese nicht gespeichert haben. Sie müssen in diesem Fall auch die Konfiguration neu durchführen, da das Optionsmodul BM4-O-CAN-03 beim Start die gespeicherte PDO-Konfiguration übernimmt. Bei einem Neustart muss das Optionsmodul BM4-O-CAN-03 auch wieder in den Zustand OPERATIONAL gesetzt werden.

Damit erkennbarer Datenverkehr stattfindet starten wir zusätzlich den bereits vorhandenen FB CANop405_PDO_WRITE wodurch P1040 verändert wird. Das Statuswort können wir über WinBASS II beeinflussen. Mit x_ENABLE = 1 am FB CANop405_PDO_READ kann der PDO-Empfang gestartet werden. Es sollten unmittelbar PDOs eintreffen:

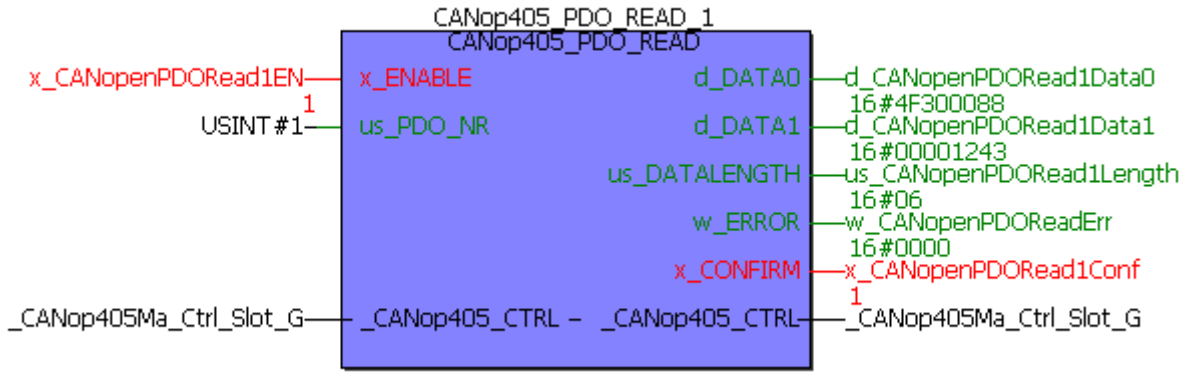


Abbildung 97: Daten mit dem FB CANop405_PDO_READ lesen - Online

Beachten Sie, dass dies Momentaufnahmen im Online-Modus sind und Ihre Werte sich unterscheiden können. Der Ausgang x_CONFIRM wird auch nicht ständig den Wert 1 haben, da die PDOs im 5 ms Zeitintervall gesendet werden, der FB CANop405_PDO_READ aber im 2 ms Zeitintervall aufgerufen wird.

4.4.10 Synchronisieren des Datenaustausch

4.4.10.1 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 114). 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)	*Nicht relevant*
30	0: Netzwerkknoten generiert kein SYNC-Telegramm 1: Netzwerkknoten generiert SYNC-Telegramm
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
28 - 11	Wird vom Optionsmodul CANopen-Master nicht unterstützt
10 - 0 (LSB)	COB-ID des SYNC-Telegramms. Da das Optionsmodul CANopen-Master die COB-ID nach der vordefinierten Identifiervergabe (siehe Kapitel ▹Datenaustausch und Objekte des physikalischen Bussystems◃ auf Seite 53) verwendet muss diese 80h betragen.

Die Default-Einstellung ist nahezu immer: SYNC-Consumer, d. h. Sie brauchen die Netzwerkknoten nicht extra auf SYNC-Consumer einzustellen.

Das Optionsmodul CANopen-Master kann als SYNC-Producer verwendet werden. Da das Optionsmodul CANopen-Master keine eigenen Objekte hat, brauchen Sie nur den Funktionsbaustein CANop405_SYNC zu verwenden, um ein SYNC-Telegramm zu generieren. Die Funktion SYNC-Consumer wird nicht unterstützt.

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

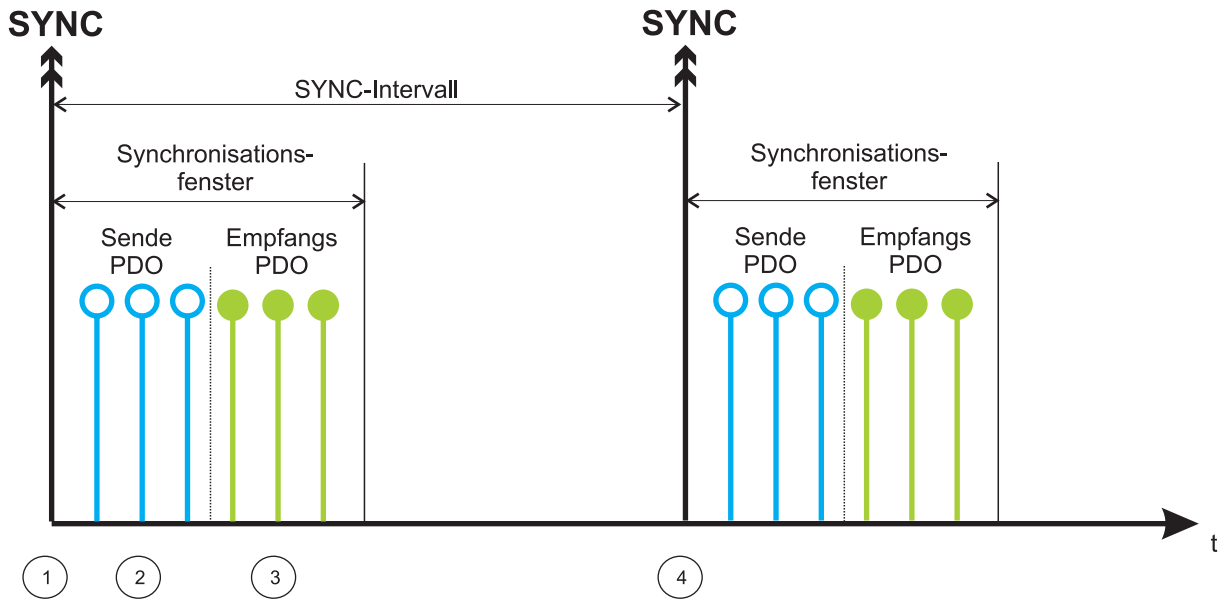


Abbildung 98: Mechanismus Synchronisierung

Nach Empfang des SYNC-Telegramms ① sendet ein Netzwerkknoten zunächst die Sende-PDOs (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. Synchrone PDOs dürfen nur innerhalb des Synchronisationsfensters auftreten. Manche Netzwerkknoten bieten die Möglichkeit das SYNC-Intervall zu überwachen und können entsprechend reagieren, falls Fehler auftreten. Ebenso gibt es Netzwerkknoten, welche nicht nur die Datenübertragung sondern auch interne Prozesse auf das SYNC-Telegramm synchronisieren. In beiden Fällen muss auf dem Netzwerkknoten das SYNC-Intervall im Objekt 1006h eingestellt werden:

SYNC-Intervall:

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

Beachten Sie, dass manche Geräte nicht beliebige Werte des Objektes 1006h akzeptieren, sondern nur bestimmte Werte (z. B. 1000, 2000, 4000, etc.).

Durch den Übertragungstyp des PDO (siehe Kapitel ▹Die Konfiguration der Übertragungseigenschaften eines PDO◃ auf Seite 114) 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.

4.4.10.2 SYNC-Telegramme senden

Das Absetzen von SYNC-Telegrammen durch das Optionsmodul CANopen-Master erfolgt mit dem FB CANop405_SYNC. Um diesen Funktionsbaustein zu verwenden, gehen Sie bitte in folgenden Schritten vor:

- Der FB CANop405_SYNC sollte in einer POE implementiert werden, welche im gleichen Zeitintervall aufgerufen wird wie die Funktionsbausteine vom Typ CANop405_PDO_WRITE oder CANop405_PDO_READ und welche synchronisierte PDOs senden oder empfangen. Sollte eine solche POE nicht vorhanden sein, legen Sie diese mit SPS-Typ SH03-30 und Prozessortyp BM4_O_PLC01 an. Diese POE soll später in einer Timer-Event-Task aufgerufen werden.
- Beschalten Sie den Funktionsbaustein mit Variablen vom richtigem Datentyp. Der Start der SYNC-Telegramm Generierung erfolgt mit x_ENABLE = 1.
- Legen Sie eine zyklische Bypass IRP Task an, falls eine solche in Ihrem Projekt noch nicht vorhanden sein sollte. Binden Sie die erstellte POE mit dem FB CANop405_SYNC 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 das SYNC-Telegramm abgesetzt. Der FB CANop405_SYNC meldet eine erfolgreiche Durchführung mit x_CONFIRM = 1 und w_ERROR = 16#0. Das SYNC-Telegramm wird bei jedem Durchlauf des Funktionsbausteins mit x_ENABLE = 1 generiert. Ein Zurücksetzen mit x_ENABLE = 0 ist nicht nötig.

4.4.10.3 Beispiel: SYNC-Telegramme senden

Wir wollen den in den Kapiteln ▹Objekte über PDOs lesen◃ ab Seite 132 und ▹Objekte über PDOs schreiben◃ ab Seite 120 erstellten Datenaustausch über PDOs mit dem Optionsmodul BM4-O-CAN-03 synchronisieren. Hierzu muss sich das Optionsmodul BM4-O-CAN-03 zunächst im Zustand PRE-OPERATIONAL befinden. Stellen Sie den Übertragungstyp für das zweite Empfangs-PDO (Objekt 1401h) und das erste Sende-PDO (Objekt 1800h) auf synchron ein. Die PDOs sollen bei jedem 10. SYNC-Telegramm übernommen bzw. gesendet werden, d. h. in Subindex 2 der Objekte 1401h und 1800h muss der Wert 10 eingetragen werden. Sobald Sie wieder in den Zustand OPERATIONAL wechseln und das Senden bzw. Empfangen der PDOs freigeben, werden Sie feststellen, dass sich in WinBASS II die Parameter 1171 und 1040 nicht mehr ändern, obwohl der FB CANop405_PDO_WRITE Daten sendet. Auch der FB CANop405_PDO_READ gibt kein x_CONFIRM = 1 mehr aus. Dies soll auch so sein, da schließlich noch keine

Synchronisation erfolgt, denn der FB CANop405_SYNC ist noch nicht implementiert und aktiv.

Zunächst müssen jedoch noch ein paar Werte konfiguriert werden. Dazu muss sich das Optionsmodul BM4-O-CAN-03 wieder im Zustand PRE-OPERATIONAL befinden. Als erstes muss über den Parameter 531 die Quelle für das Sync-Signal auf dem b maXX Regler eingestellt werden. Wir benötigen "Sync 1 Signal von der BACI verwenden". Dazu muss auf das Objekt 4213h (P531), Subindex 0 der Wert 2 (Datenlänge 1 Byte) geschrieben werden:

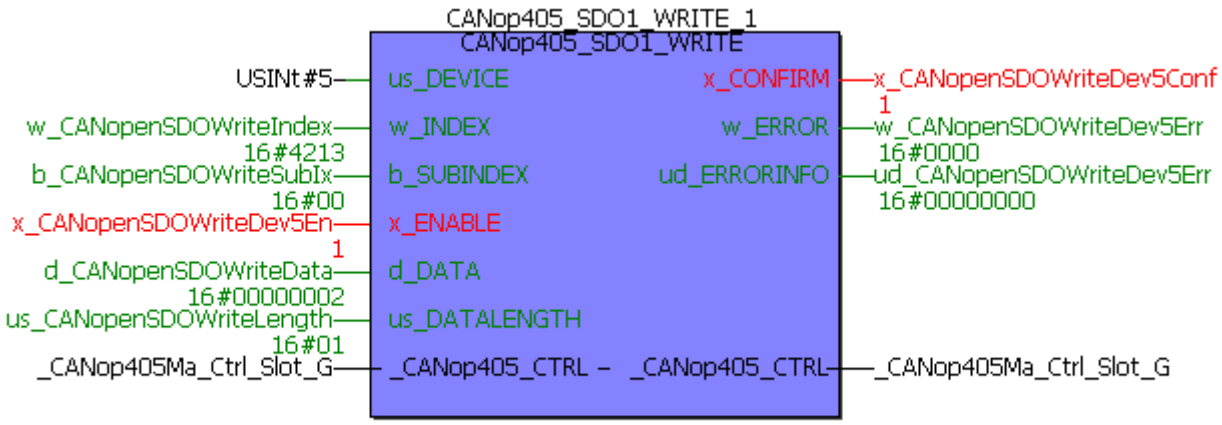


Abbildung 99: Quelle für das Sync-Signal auf dem b maXX Regler einstellen

Der FB CANop405_SYNC soll alle 2 ms (= 2000 µs = 7D0h µs) aufgerufen werden. Auf diesen Wert muss auch das Objekt 1006h (Dauer Kommunikationszyklus) eingestellt werden:

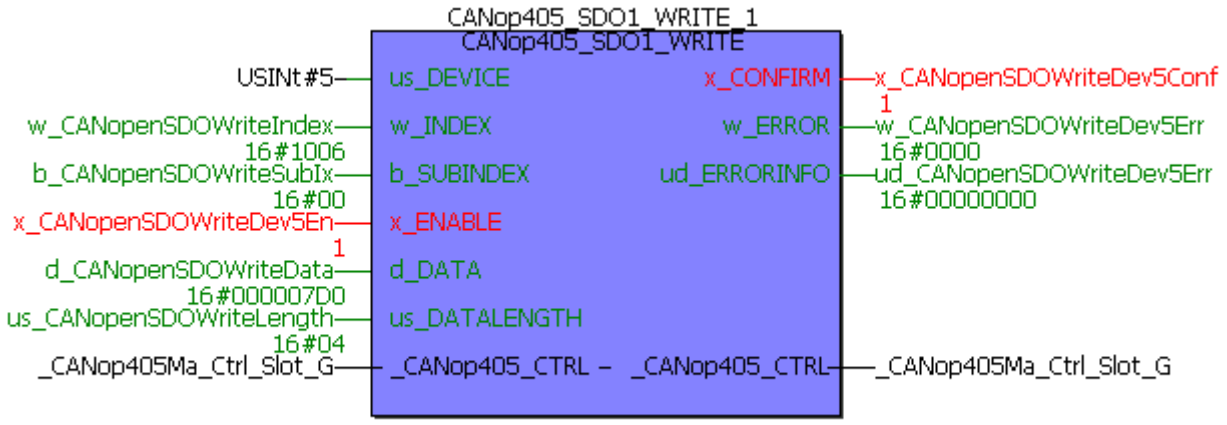


Abbildung 100:Kommunikationszyklus einstellen

Dieser Wert wird automatisch auf den Parameter 532 "Sync-Intervall" übernommen. Speichern Sie über WinBASS II den Datensatz des b maXX Reglers ab, damit Sie die eingestellte Konfiguration beim Aus-/Einschalten nicht verlieren.

Den FB CANop405_SYNC implementieren wir unterhalb des FB CANop405_PDO_READ. Somit wir er automatisch alle 2 ms aufgerufen. Übersetzen Sie das Projekt und senden Sie dieses als Projekt (nicht als Boot-Projekt) an die PLC. Starten Sie das Projekt. Sollten Sie ein Boot-Projekt gesendet und das b maXX 4400 Gerät neu

gestartet haben verlieren Sie die zuvor eingestellte SYNC-Konfiguration des b maXX Reglers, falls Sie den Datensatz des Reglers nicht gespeichert haben. Sie müssen in diesem Fall die Konfiguration neu durchführen.

Nach dem Start muss das Optionsmodul BM4-O-CAN-03 wieder in den Zustand OPERATIONAL gesetzt werden. Geben Sie die FBs CANop405_PDO_READ und CANop405_PDO_WRITE frei. Noch sollte der FB CANop405_PDO_READ keine Daten empfangen. Sobald jedoch auch der FB CANop405_SYNC freigegeben ist werden PDOs empfangen.

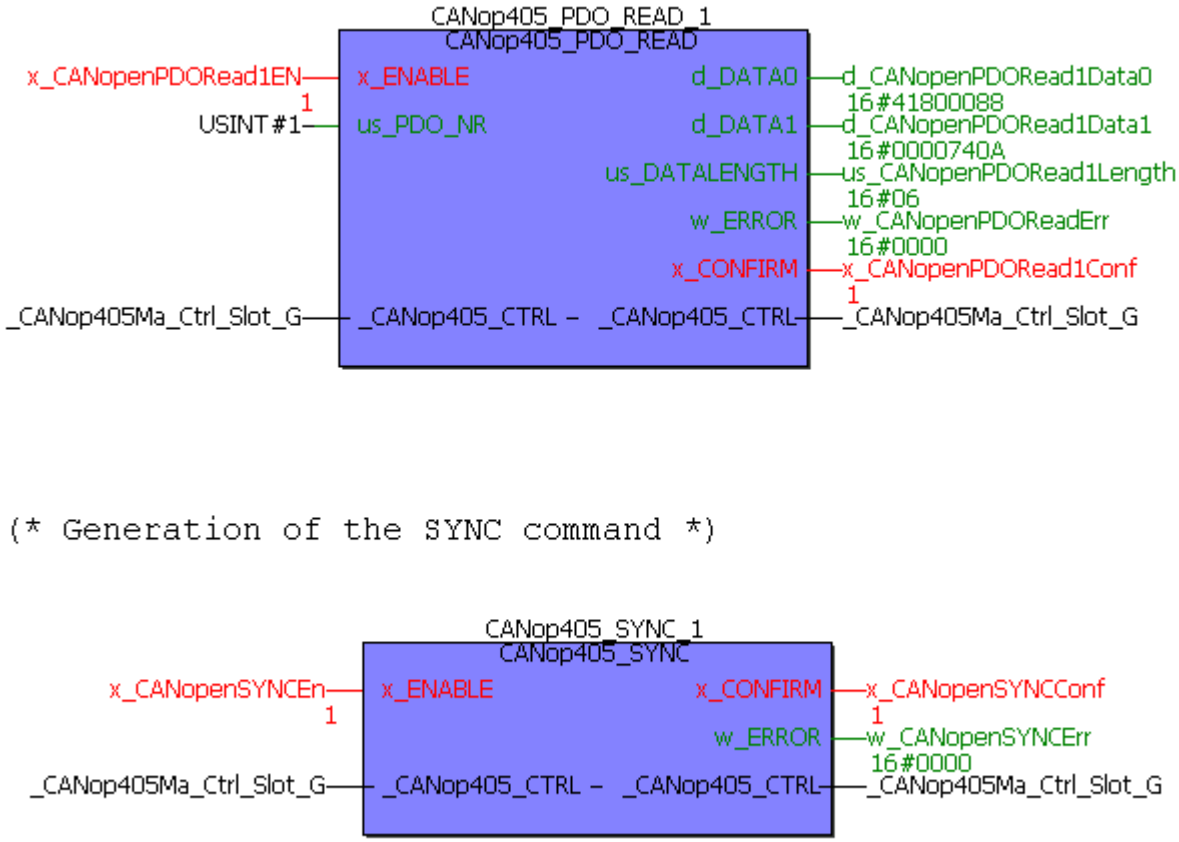


Abbildung 101: Generierung des SYNC-Telegramms mit dem FB CANop405_SYNC

Beachten Sie, dass dies Momentaufnahmen im Online-Modus sind und Ihre Werte sich unterscheiden können.

4.4.11 Überwachen von Netzwerkknoten durch Node Guarding

4.4.11.1 Definition nach CANopen-Spezifikation

Für die Ausfallüberwachung von Netzwerkknoten gibt es den Mechanismus Node Guarding und den Mechanismus 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 Master in bestimmten Zeitintervallen ‚Guard Time' über ein Remote-Telegramm eine Antwort vom Slave an. Der Slave antwortet mit der Guarding-Nachricht. Diese enthält den Knotenzustand (siehe Kapitel ▷Starten der einzelnen Netzwerkknoten - NMT◁ ab Seite 104) des Slaves sowie ein Toggle-Bit, das nach jeder Nachricht wechseln muss. Falls Status oder Toggle Bit nicht mit den vom 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, und setzt z. B. seine Ausgänge in den Fehlerzustand und sendet ein Emergency Telegramm.

Zur Einstellung der Guarding Funktionalität, sind auf dem Netzwerknoten zwei Objekte wichtig:

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 x Guard Time

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 Slave nach Empfang des ersten Guarding-Anforderungstelegramms. Ab diesem Zeitpunkt läuft im Slave die in den Objekten 100Ch und 100Dh parametrierte Überwachungszeit.

4.4.11.2 Node Guarding

Das Node Guarding durch das Optionsmodul CANopen-Master erfolgt mit dem FB CANop405_NODE_GUARDING. Um diesen Funktionsbaustein zu verwenden, gehen Sie bitte in folgenden Schritten vor:

- Der FB CANop405_NODE_GUARDING ist in einer POE mit SPS-Typ SH03-30 und Prozessortyp BM4_O_PLC01 zu implementieren.

- Beschalten Sie den Funktionsbaustein mit Variablen vom richtigem Datentyp. An us_DEVICE geben Sie die Nummer des zu überwachenden Knoten an. t_NODE_GUARD_TIME ist die Guarding Zeit und us_LIFE_TIME_FACTOR der Life Time Faktor.
- Legen Sie eine Task an, in welche Sie die POE mit den FB CANop405_NODE_GUARDING einbinden. Die Zykluszeit dieser Task hängt davon ab, wie schnell Sie einen Ausfall eines Netzwerkknotens erkennen wollen. Der Mechanismus selbst des Nodes Guarding läuft auf dem Optionsmodul CANopen-Master ab. Es ist also nicht notwendig den FB CANop405_NODE_GUARDING im Zeitintervall der Guard Time aufzurufen.
- Übersetzen Sie das Projekt und laden Sie es als (Boot-)Projekt auf die PLC.
- Konfigurieren Sie die Guard Time (Objekt 100Ch) und den Life-Time Factor (Objekt 100Dh) auf den einzelnen Netzwerkknoten. Verwenden Sie dazu die Funktionsbausteine zur SDO-Kommunikation.

Mit x_ENABLE = 1 wird das Node Guarding gestartet. Der FB CANop405_NODE_GUARDING meldet x_NODE_OK = 1, sobald die erste gültige Antwort vom Slave eintrifft. Solange im weiteren gültige Antworten vom Slave kommen, bleibt x_NODE_OK = 1. Der Zustand des Slaves wird an us_NODE_STATE ausgegeben. Der FB CANop405_ NODE_GUARDING ist die einzige Möglichkeit einen Knotenzustand auszulesen. Kommt eine ungültige Antwort des Slaves oder keine innerhalb der Life Time, so erfolgt eine Fehlermeldung und x_NODE_OK wird 0.

4.4.11.3 Beispiel: Überwachen eines Netzwerkknoten mit Node Guarding

Wir wollen das Optionsmodul BM4-O-CAN-03 in einem 500 ms Intervall durch Node Guarding auf Ausfall überwachen. Um die Node Guarding Zeit und den Life Time Faktor einstellen zu können muss sich das Optionsmodul BM4-O-CAN-03 zunächst im Zustand PRE-OPERATIONAL befinden. Nun stellen wir das Objekt 100Ch auf 500 ms ein:

```
(* Write object via SDO to device nr. 5 *)
```

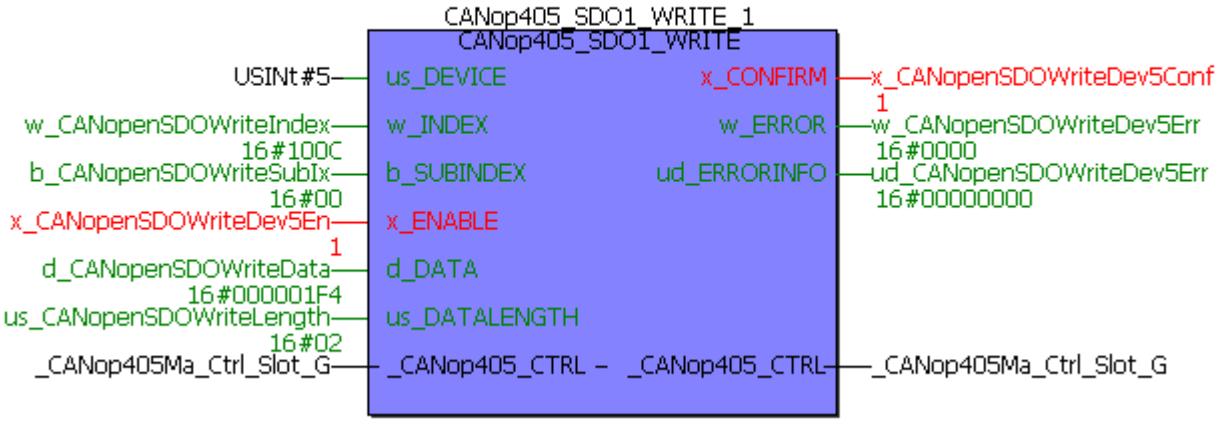


Abbildung 102: Node Guarding Zeit auf CANopen-Slave einstellen

Für den Life Time Faktor, Objekt 100Dh, wählen wir den Wert 3:

```
(* Write object via SDO to device nr. 5 *)
```

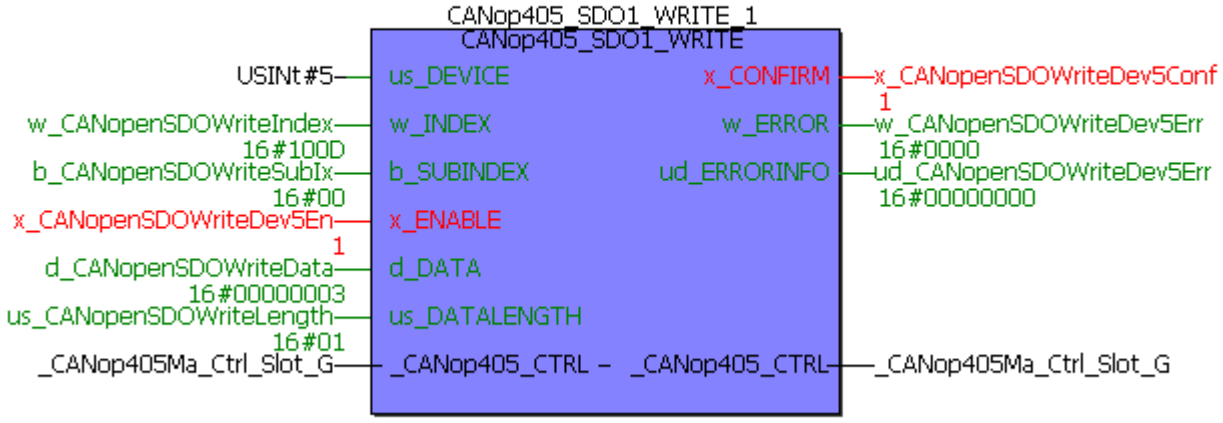


Abbildung 103: Life Time Faktor auf CANopen-Slave einstellen

Den FB CANop405_NODE_GUARDING implementieren wir in der POE mit den Funktionsbausteinen zur SDO-Kommunikation. Das Zeitintervall der dieser POE zugeordneten Task reicht für unsere Zwecke leicht aus, um einen Knotenausfall zu erkennen. Nach der Beschaltung des Funktionsbaustein übersetzen Sie das Projekt und senden Sie dieses als Projekt (nicht als Boot-Projekt) an die PLC. Starten Sie das Projekt. Sollten sie ein Boot-Projekt gesendet und das b maXX 4400 Gerät neu gestartet haben verlieren Sie die zuvor eingestellte Node Guarding Konfiguration. Sie müssen in diesem Fall die Konfiguration neu durchführen, da das Optionsmodul BM4-O-CAN-03 beim Start die Guard Time und den Life Time Factor auf die Default-Werte setzt. Node Guarding ist im Zustand PRE-OPERATIONAL und OPERATIONAL möglich. Im Online-Modus können wir das Node Guarding mit x_ENABLE = 1 starten.

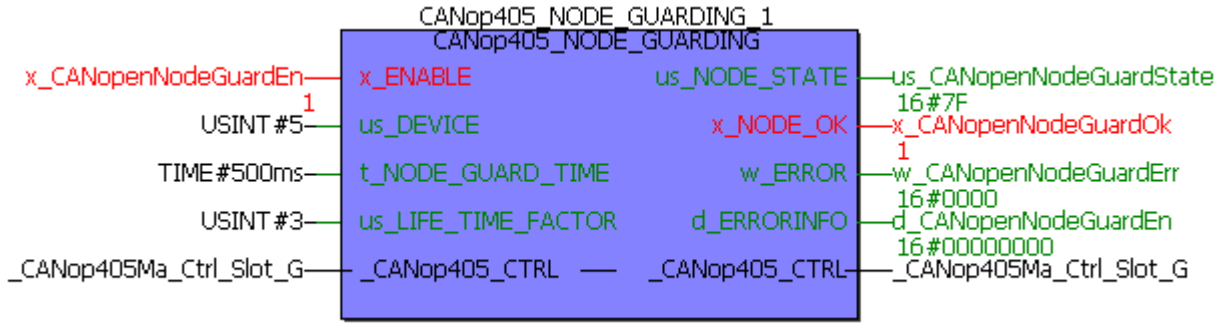


Abbildung 104: Node Guarding mit dem FB CANop405_NODE_GUARDING

In diesem Fall meldet sich das Optionsmodul BM4-O-CAN-03 mit dem Zustand PRE-OPERATIONAL (7Fh). Durch Lösung des CAN-Kabels können wir die Fehlermeldung "Fehler Toggle-Bit" generieren:

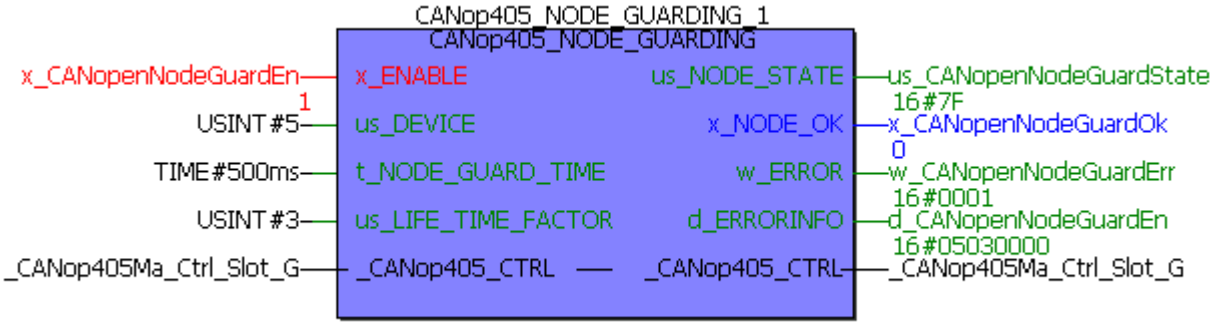


Abbildung 105:Node Guarding "Fehler Toggle-Bit"

4.4.12 Empfangen von Emergency-Telegrammen

4.4.12.1 Definition nach CANopen-Spezifikation

Emergency-Telegramme dienen Netzwerkknoten der Meldung von Fehlern. Die Übertragung des Emergency-Telegramms erfolgt nach dem Consumer/Producer-Kommunikationsmodell, d. h. ein Netzwerkknoten 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. Eine Telegrammwiederholung erfolgt nicht. Das Optionsmodul CANopen-Master kann Emergency-Telegramme von Netzwerkknoten auswerten. Das Senden eines Emergency-Telegramms 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
23xxh	Current, device output side
30xxh	Voltage
31xxh	Mains Voltage
32xxh	Voltage inside the device
33xxh	Output Voltage

Error Code	Bedeutung
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 Netzwerkknoten interne Fehler eintragen. Die Codierung nach CiA sieht wie folgt aus:

Bit	Bedeutung
0	Generic error
1	Current
2	Voltage
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. Die Codierung von Fehlern ist auf Grund der vielfältigen Möglichkeiten sehr Geräte spezifisch und daher in den jeweiligen Gerätedokumentation nachzulesen. Beachten Sie bitte auch, dass die meisten Einträge im Emergency Error Code und Error Register nicht verpflichtend sind. Sehen Sie auch hierzu in der Gerätedokumentation nach, welche Einträge unterstützt werden.

4.4.12.2 Auswerten von Emergency-Telegrammen

Das Auswerten von Emergency-Telegrammen durch das Optionsmodul CANopen-Master erfolgt mit dem FB CANop405_EMERGENCY. Um diesen Funktionsbaustein zu verwenden, gehen Sie bitte in folgenden Schritten vor:

- Der FB CANop405_EMERGENCY ist in einer POE mit SPS-Typ SH03-30 und Prozessortyp BM4_O_PLC01 zu implementieren.
- Beschalten Sie den Funktionsbaustein mit Variablen vom richtigem Datentyp. An us_DEVICE geben Sie die Nummer des zu überwachenden Knoten an. Wird ein Emergency-Telegramm empfangen, werden der empfangene Emergency Error Code an w_EMCY_ERROR_CODE, das Error Register an b_ERROR_REGISTER und die Hersteller spezifische Fehler Info im Array a_ERROR_FIELD ausgegeben. Ist der empfangene Fehler Code >< 0 wird der Ausgang x_EMERGENCY auf 1 gesetzt. Mit x_RESET = 1 werden die Ausgänge zurückgesetzt.
- Legen Sie eine Task an, in welche Sie die POE mit den FB CANop405_EMERGENCY einbinden. Die Zykluszeit dieser Task hängt davon ab, wie schnell Sie ein empfangenes Emergency-Telegramm erkennen wollen.
- Übersetzen Sie das Projekt und laden Sie es als (Boot-)Projekt auf die PLC.

Ein empfangenes Emergency-Telegramm wird mit x_EMERGENCY = 1 angezeigt. Der Funktionsbaustein muss nicht extra freigegeben werden, er ist automatisch aktiv.

4.4.12.3 Beispiel: Emergency-Telegramm empfangen

Wir wollen das Optionsmodul BM4-O-CAN-03 zum Senden eines Emergency-Telegramms veranlassen und dieses dann auswerten. Eine einfache Möglichkeit hierzu ist, einfach ein aktives Node Guarding zu deaktivieren. Daraufhin muss das Optionsmodul BM4-O-CAN-03 einen Life Guarding Fehler generieren. Nach CiA Definition (DS 301) und der Dokumentation des Optionsmodul BM4-O-CAN-03 liefert ein Life Guarding Fehler folgende Werte:

Emergency Error Code:	8130h
Error Register:	mindestens Bit 0 und 4 gesetzt
Hersteller Code:	0h, da kein Fehler des b maXX Gerätes

Den FB CANop405_EMERGENCY implementieren wir in der POE mit den Funktionsbausteinen zur SDO-Kommunikation. Nach der Beschaltung des Funktionsbaustein übersetzen Sie das Projekt und senden Sie dieses als Boot-Projekt an die PLC. Starten Sie das b maXX 4400 Gerät neu. Aktivieren Sie das Node Guarding wie in Kapitel "Node Guarding" beschrieben. Vergessen Sie nicht die Objekte 100Ch (Guard Time) und 100Dh (Life Time Faktor) über SDO zu setzen. Aktivieren Sie das Node Guarding. Es kann sein,

dass der FB CANop405_EMERGENCY ein früheres Emergency-Telegramm anzeigt. Führen Sie in diesem Fall einen Reset des Funktionsbausteins durch indem Sie x_RESET kurzzeitig auf 1 setzen. Schalten Sie jetzt das Node Guarding am FB CANop405_NODE_GUARDING mit x_ENABLE = 0 aus. Unmittelbar danach sollte der FB CANop405_EMERGENCY den Erhalt des Emergency-Telegramms anzeigen:

```
(* check for received emergency messages of device nr. 5 *)
```

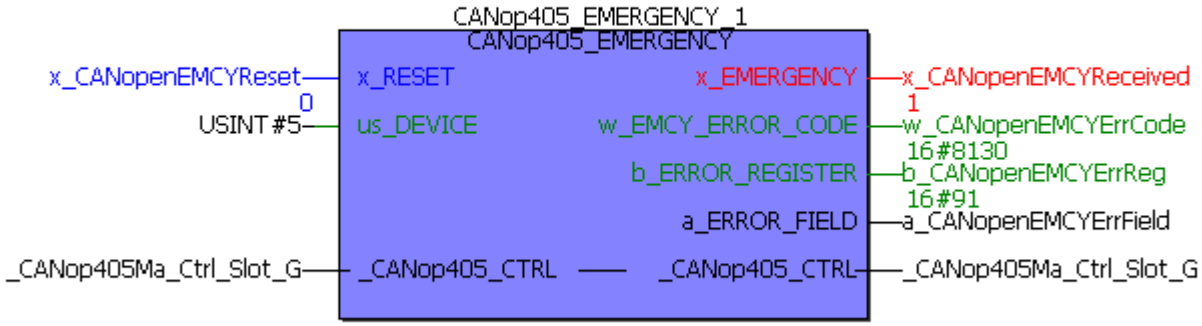
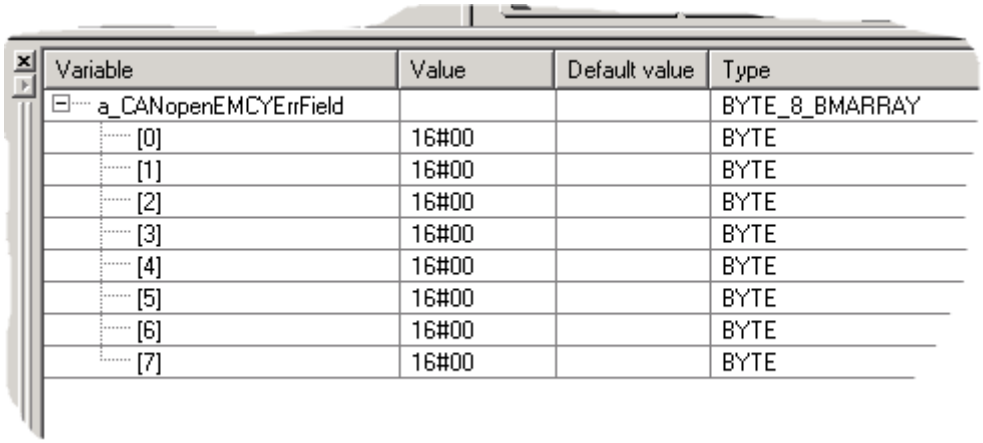


Abbildung 106: Auswerten von Emergency Telegrammen mit dem FB CANop405_EMERGENCY



Variable	Value	Default value	Type
a_CANopenEMCYErrField			BYTE_8_BMARRAY
[0]	16#00		BYTE
[1]	16#00		BYTE
[2]	16#00		BYTE
[3]	16#00		BYTE
[4]	16#00		BYTE
[5]	16#00		BYTE
[6]	16#00		BYTE
[7]	16#00		BYTE

Abbildung 107: a_ERROR_FIELD im Watchfenster von PROPROG wt II

Den Wert von a_ERROR_FIELD sehen wir uns im Watch-Fenster von PROPROG wt II an. Die angezeigten Werte stimmen mit den von uns erwarteten überein. Mit x_RESET = 1 kann der Funktionsbaustein zurück gesetzt und somit das nächste Emergency-Telegramm empfangen werden:

```
(* check for received emergency messages of device nr. 5 *)
```

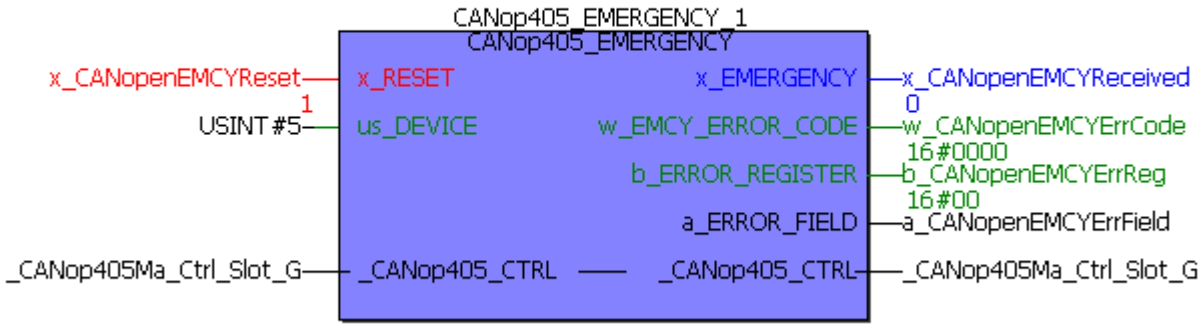


Abbildung 108: Rücksetzen des FB CANop405_EMERGENCY

4.5 CANopen und Motion Control mit PROPROG wt II und Motion Konfigurator

4.5.1 Allgemeines zum Optionsmodul CANopen-Master und Motion Control

Sollten Sie das Optionsmodul CANopen-Master zusammen mit Motion Control verwenden, dann darf ein Knoten im CANopen-Netzwerk zu keinem Zeitpunkt über die Funktionsbausteine aus der Bibliothek CANop405_PLC01_20bd00 (oder höher) zusammen mit Motion Control bedient werden. Ein Netzwerkknoten wird also entweder über die Funktionsbausteine aus der Bibliothek CANop405_PLC01_20bd00 (oder höher) oder von Motion Control bedient.

4.5.2 Initialisierung des Optionsmodul CANopen-Master

Bei der Verwendung von Motion Control wird die Initialisierung des Optionsmodul CANopen-Master von Motion Control übernommen. Der Funktionsbaustein CANop405_INIT darf nicht mehr verwendet werden.

4.5.3 Datenaustausch über PDOs

Bei der Verwendung des Optionsmodul CANopen-Master zusammen mit Motion Control reduziert sich die Anzahl der zu schreibenden PDOs von 40 um die Zahl der Achsen, welche von Motion Control bedient werden:

$$\text{Zu schreibende PDOs} \leq 40 - \text{Anzahl der Motion-Control Knoten}$$

Damit reduziert sich auch die maximal zu vergebende Sendefachnummer auf diesen Wert. Auch die Anzahl der zu empfangenden PDOs verringert sich. Die maximale Anzahl zu empfangender PDOs reduziert sich von 63 um die doppelte Anzahl der Achsen, welche von Motion Control bedient werden:

$$\text{Zu empfangende PDOs} \leq 63 - 2 \times \text{Anzahl der Motion-Control Knoten}$$

Damit reduziert sich auch die maximal zu vergebende Empfangsfachnummer auf diesen Wert.

Der Motion Control Konfigurator berücksichtigt diese Reduzierung der PDOs. Beachten Sie diese Werte jedoch bei der Projektierung Ihrer Applikation.

4.5.4 Synchronisierung des Datenaustausch über PDOs

Bei der Verwendung von Motion Control wird die Generierung des SYNC-Telegramms für den Datenaustausch von PDOs von Motion Control übernommen. Der Funktionsbaustein CANop405_SYNC darf nicht mehr verwendet werden. Natürlich können Nicht-Motion Control Knoten durch entsprechende Konfiguration in den synchronisierten Datenaustausch eingebunden werden.

4.6 Der Aufbau von CANopen-Telegrammen

4.6.1 Allgemeines

Durch das Optionsmodul CANopen-Master und die Funktionsbausteine aus der Bibliothek CANop405_PLC01_20bd00 (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).

4.6.2 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).

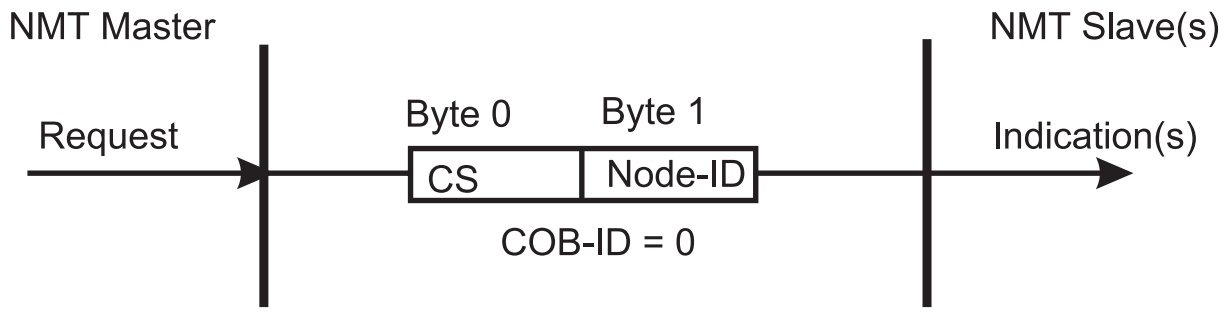


Abbildung 109: Kommunikationsbeziehung NMT

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 Rücksetzen des gesamten Objektverzeichnis' auf Default-Werte
130	Reset_Communication	Rücksetzen 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.

4.6.3 SDO-Telegramm

4.6.3.1 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).

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-Master 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.

4.6.3.2 Arten des SDO-Transfers

Das Optionsmodul CANopen-Master 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.

4.6.3.3 Objekt über SDO schreiben

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

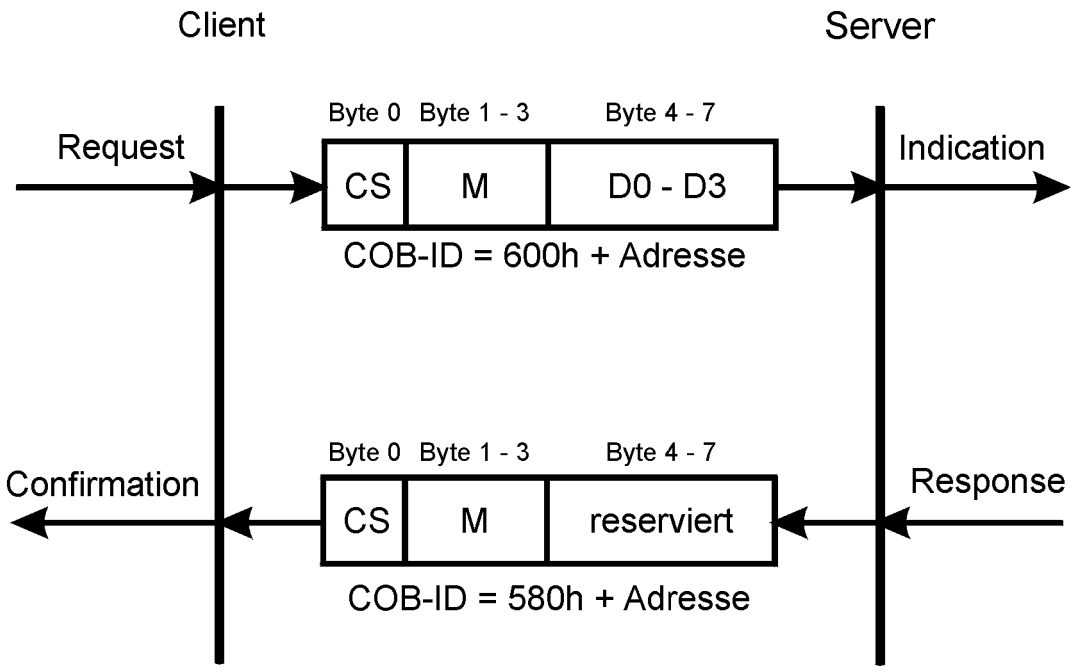


Abbildung 110: Kommunikationsbeziehung SDO schreiben

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).

Beispiel:

Auf das Objekt 6060h, Subindex 00h, des Slaves mit der Adresse 4 soll der Wert "-3" (FDh) geschrieben werden. Die Datenbreite dieses Objektes beträgt acht Bit.

Request:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
604h	0	8h	2Fh	60h	60h	00h	FDh	00h	00h	00h

Response:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
584h	0	8h	2Fh	60h	60h	00h	00h	00h	00h	00h

Auf das Objekt 43E9h, Subindex 00h, des Slaves mit der Adresse 4 soll der Wert "12" (0Ch) geschrieben werden. Die Datenbreite dieses Objektes beträgt 16 Bit.

Request:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
604h	0	8h	2Bh	E9h	43h	00h	0Ch	00h	00h	00h

Response:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
584h	0	8h	60h	E9h	43h	00h	00h	00h	00h	00h

Auf das Objekt 1800h, Subindex 02h, des Slaves mit der Adresse 4 soll der Wert "60610008h" geschrieben werden. Die Datenbreite dieses Objektes beträgt 32 Bit.

Request:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
604h	0	8h	23h	00h	18h	02h	08h	00h	61h	60h

Response:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
584h	0	8h	60h	00h	18h	02h	00h	00h	00h	00h

4.6.3.4 Objekt über SDO lesen

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

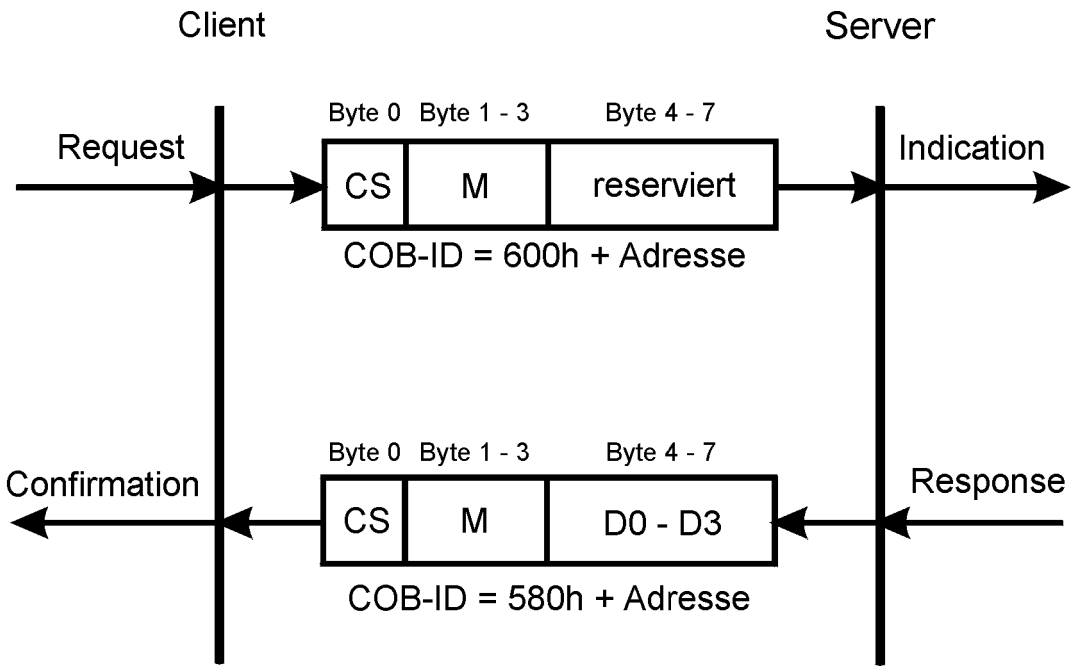


Abbildung 111:Kommunikationsbeziehung SDO lesen

Ein SDO Server (Master) sendet einen Lese-Request an den Slave. Dieser Slave führt die Anforderung aus und schickt die geforderten Daten im Antworttelegramm (Re-

sponse). 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.

Beispiel:

Das Objekt 6061h, Subindex 00h, des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 8 Bit

Request:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
604h	0	8h	40h	61h	60h	00h	00h	00h	00h	00h

Response:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
584h	0	8h	4Fh	61h	60h	00h	D0	00h	00h	00h

Das Objekt 6041h, Subindex 00h, des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 16 Bit.

Request:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
604h	0	8h	40h	41h	60h	00h	00h	00h	00h	00h

Response:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
584h	0	8h	4Bh	41h	60h	00h	D0	D1	00h	00h

Das Objekt 1400h, Subindex 01h, des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 32 Bit.

Request:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
604h	0	8h	40h	00h	14h	01h	00h	00h	00h	00h

Response:

			CS	Multiplexor			D0	D1	D2	D3
COB-ID	**RTR**	**DLC**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
584h	0	8h	43h	00h	14h	01h	D0	D1	D2	D3

4.6.4 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.

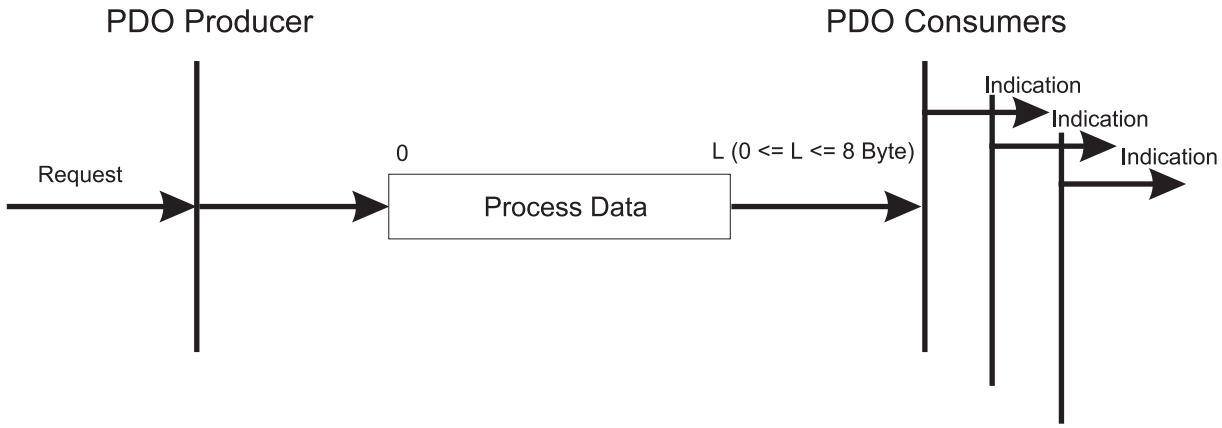


Abbildung 112:Kommunikationsbeziehung PDO

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 Identifiervergabe 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

4.6.5 SYNC-Telegramm

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

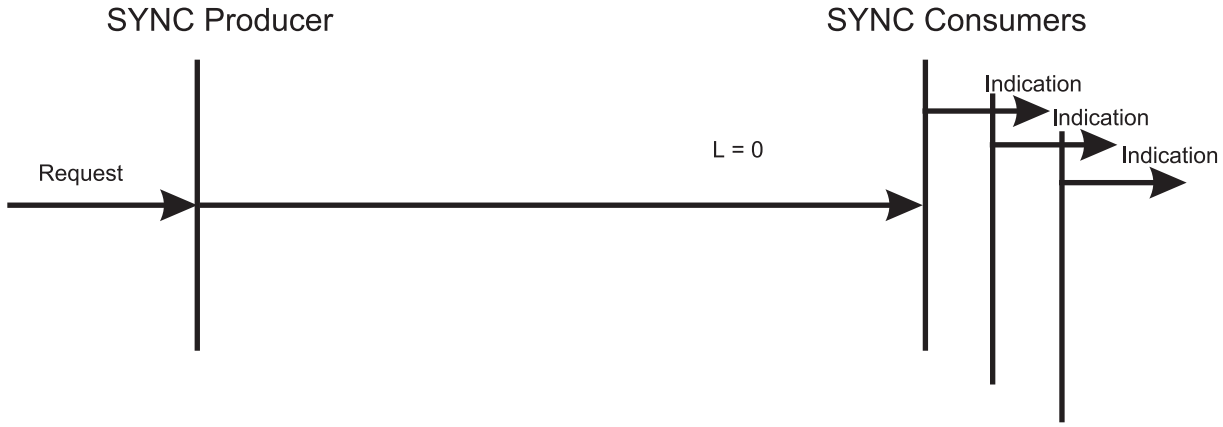


Abbildung 113: Kommunikationsbeziehung SYNC

Das SYNC-Telegramm enthält keine Daten. Die COB-ID ist 80h nach der vordefinierten Identifiervergabe 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	-	-	-	-	-	-	-	-

4.6.6 Node Guarding Telegramm

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

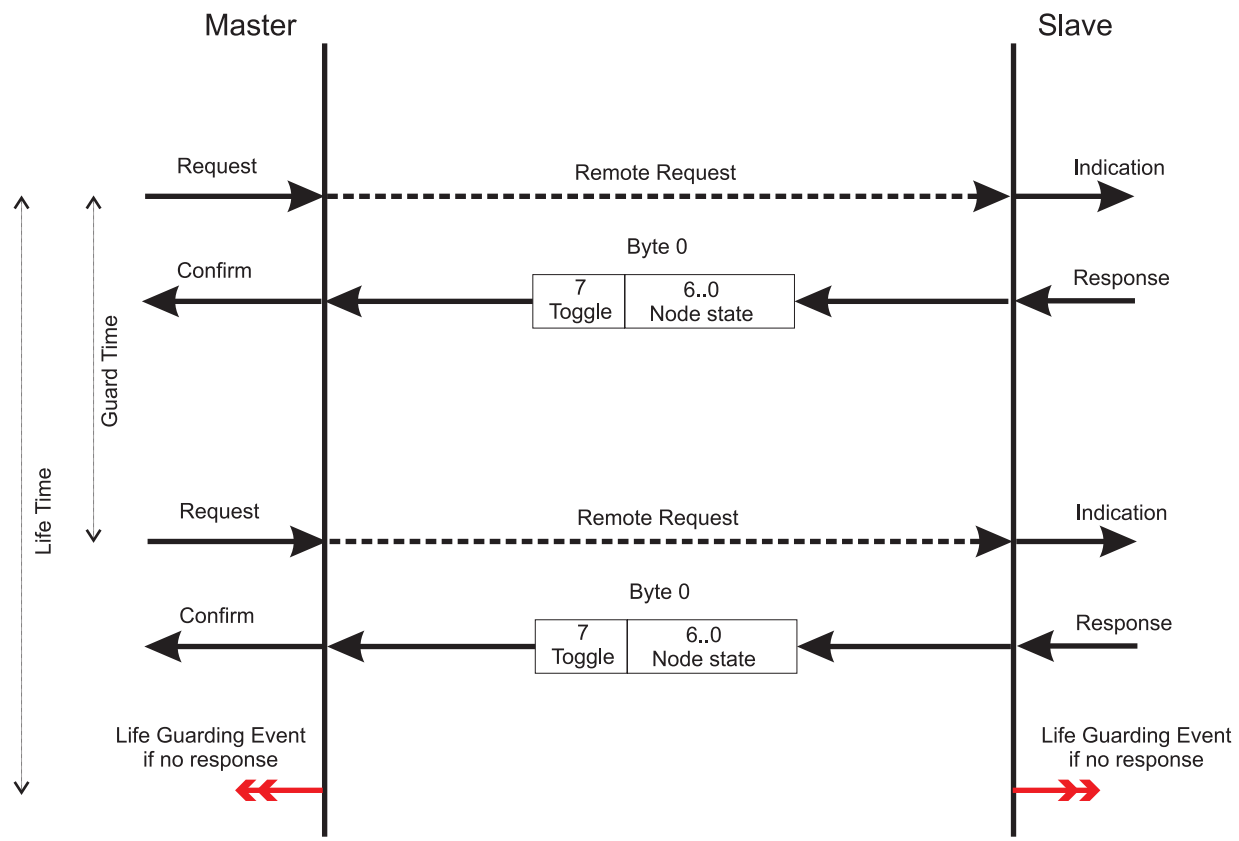


Abbildung 114: Kommunikationsbeziehung Node Guarding

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.

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	-

4.6.7 Emergency-Telegramm

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

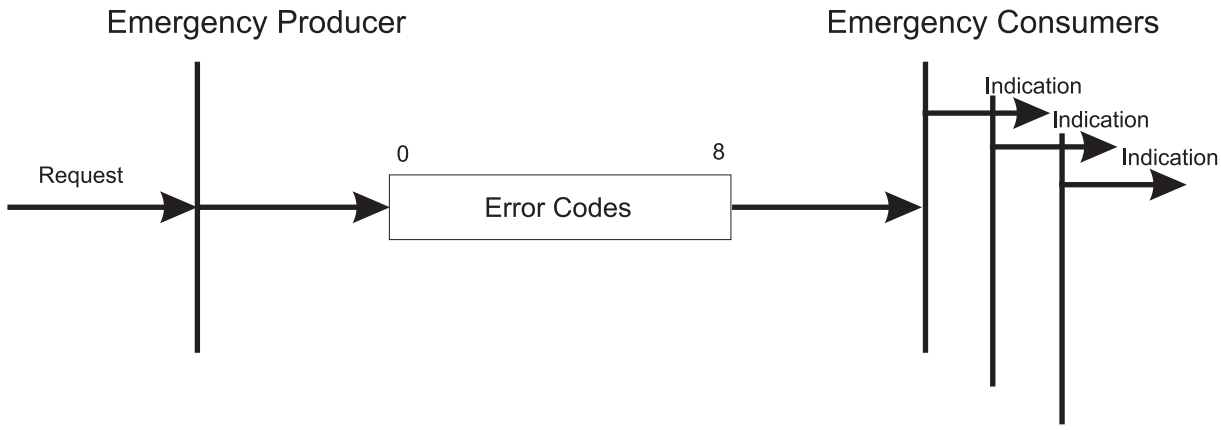


Abbildung 115: Kommunikationsbeziehung Emergency

Die COB-ID ergibt sich aus 80h + 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 + Adresse	0h	8h	Emergency Error Code		Error Reg.	Hersteller spezifisch				

ANHANG A - ABKÜRZUNGEN

API	Application Program Interface (Programmierschnittstelle)
ARP	Address Resolution Protocol
BACI	Baumüller Component Interface
BUB	Ballast-Einheit
BUC	Baumüller Ein-/Rückspeise-Einheit
BUG	Baumüller Umrichter Grund-Einspeise-Einheit
BUM	Baumüller Einzel-Leistungs-Einheit
BUS	Baumüller Leistungs-Modul
CAL	CAN Application Layer
CAN	Controller Area Network
CiA	CAN in Automation e. V.
COB	Communication Object
COB-ID	Communication Object Identifier
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CPU	Central Processing Unit
DC	Gleichstrom
DCF	Device Configuration File
DHCP	Dynamic Host Configuration Protocol
DIN	Deutsches Institut für Normung e.V.
DP-RAM	Dual-Port RAM
DR	Draft Recommendation
DS	Draft Standard
DSP	Draft Standard Proposal
EDS	Electronic Data Sheet
EMV	Elektromagnetische Verträglichkeit
EN	Europäische Norm
EPROM	
ESD	
FTP	File Transfer Protocol
HD	Hammingdistanz
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol (Hypertext-Übertragungsprotokoll)
I/O	Input/Output, Eingang und Ausgang
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IRP	Interrupt
ISO	International Standard Organisation
LAN	Local Area Network
LED	Leuchtdiode
LSS	Layer Setting Services
MAC	Media Access Control
OSI	Open Systems Interconnect
PDD	Prozess-daten-verzeichnis
PDO	Process Data Object
PLC	Process Loop Controller (SPS)
RAM	Random Access Memory
SAP	Service Access Point
SDO	Service Data Object
SMS	Short Message System
SMTP	Simple Mail Transfer Protocol
SPS	Speicherprogrammierbare Steuerung

SRD	SDO Requesting Device
SRDO	Safety Relevant Data Object
TCP	Transport Control Protokoll
Telnet	Terminal over Network
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USS	Funktionsmodul USS-Protokoll
USS®	Warenzeichen Siemens, universelle serielle Schnittstelle
VDE	Verband deutscher Elektrotechniker
WWW	World Wide Web
16#	Präfix für Hexadezimalzahl

B

ANHANG B - TABELLEN

B.1 CANopen-Objekte nach DS 301

Im Nachfolgenden sind häufig verwendete Objekte nach CANopen Kommunikationsprofil DS 301 aufgeführt. Beachten Sie, dass das Modul CANopen-Master mit Softwarestand < **01.20** (d. h. BM4-O-ETH-02/CAN-04-01-00-001-**000**) diese Objekte selbst nicht besitzt.

In Verbindung mit ProMaster hat das Modul CANopen-Master eigene CANopen Objekte. Eine Übersicht über die Objekte des Moduls CANopen-Master mit Softwarestand ≥ **01.20** (d. h. ≥ BM4-O-ETH-02/CAN-04-01-00-001-**001**) finden Sie in ▹B.3 CANopen Objekte des CANopen-Masters (Softwarestand ≥ 01.20)◃ ab Seite 183.

Beachten Sie, dass das Modul CANopen-Master mit Softwarestand **< 01.20** selbst keine Objekte besitzt.

Übersicht der Objekte:

Index	Objektinhalt / Bedeutung
1000h	Gerätetyp: unterstütztes Geräteprofil und zusätzliche Information
1001h	Fehlerregister
1003h	Fehlerspeicher enthält eine Fehlerliste des Gerätes
1005h	SYNC-Telegramm COB-ID
1006h	SYNC-Intervall
100Ch	Guarding Time
100Dh	Life Time Faktor
1010h	Parameter abspeichern
1011h	Default Parameter laden
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 - Gerätetyp

Gibt das unterstützte Geräteprofil und zusätzliche Information zum Gerät an

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1000h	0h	Device Type	Unsigned32	ro	Mandatory	Nein

Der 32 Bit-Wert ist in zwei 16 Bit Werte unterteilt:

Byte: MSB LSB

Zusätzliche Information, Hersteller abhängig	Geräteprofil

1001h - Fehleregister

Enthält eine Bitleiste mit anstehenden Geräte Fehlern

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1001h	0h	Error Register	Unsigned8	ro	Mandatory	Optional

Die einzelnen Bits des 8 Bit-Wertes haben folgende Bedeutung:

Bit	Bedeutung	Kategorie
0	Generic error	Mandatory
1	Current	Optional
2	Voltage	Optional
3	Temperature	Optional
4	Communication error	Optional
5	Device profile specific	Optional
6	Reserved	Optional
7	Manufacturer specific	Optional

Ist ein Bit gesetzt, so ist der Fehler aufgetreten.

1003h - Fehlerspeicher

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

Subindex 0h enthält die Anzahl der eingetragenen Fehler. Die Fehlerliste kann gelöscht werden indem Subindex 0h mit dem Wert 0 beschrieben wird. Subindex 1h enthält den aktuellen Fehler. Tritt ein neuer Fehler auf wird dieser an Subindex 1h eingetragen und alle anderen rutschen um einen Subindex nach unten. Der 32 Bit-Wert des Standard error field ist in zwei 16 Bit Werte unterteilt:

Byte: MSB LSB

Zusätzliche Information, Hersteller abhängig	Fehler Code

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

Error Code	Bedeutung
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

1005h - SYNC-Telegramm COB-ID

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	0h	COB-ID SYNC	Unsigned32	rw	Conditional	Nein

Das Objekt hat die Kategorie Mandatory falls synchrone PDO-Übertragung unterstützt wird.

Die einzelnen Bits des 32 Bit-Wertes haben folgende Bedeutung:

Bit	Wert / Bedeutung
31 (MSB)	Nicht relevant
30	0: Netzwerkknoten generiert kein SYNC-Telegramm 1: Netzwerkknoten generiert SYNC-Telegramm

Bit	Wert / Bedeutung
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
28 - 11	Bits 28 - 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
10 - 0 (LSB)	COB-ID des SYNC-Telegramms. Da das Optionsmodul CANopen-Master die COB-ID nach der vordefinierten Identifiervergabe verwendet muss diese 80h betragen.

1006h - SYNC-Intervall

Länge des SYNC-Intervall in µs.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1006h	0h	Communication cycle period	Unsigned32	rw	Conditional	Nein

Das Objekt hat die Kategorie Mandatory für Netzwerkknoten, welche das SYNC-Telegramm generieren. In diesem Fall wird hier die Periode in µs eingetragen. Netzwerkknoten, welche das SYNC-Telegramm auswerten können diesen Wert zu Überwachungszwecken und um interne Kommunikationszyklen zu synchronisieren verwenden.

100Ch - Guarding Time

Abstand zwischen zwei Guarding Telegrammen in ms.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
100Ch	0h	Guard time	Unsigned16	rw	Conditional	Nein

Das Objekt enthält den Abstand zwischen zwei Guarding Telegrammen in ms. Das Objekt hat die Kategorie Mandatory für Netzwerkknoten, welche den Heartbeat nicht unterstützen. Da mit dem Optionsmodul CANopen-Master keine Überwachung der Netzwerkknoten durch Heartbeat möglich ist, muss das Objekt vorhanden sein.

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	0h	Life time factor	Unsigned8	rw	Conditional	Nein

Das Objekt hat die Kategorie Mandatory für Netzwerkknoten, welche den Heartbeat nicht unterstützen. Da mit dem Optionsmodul CANopen-Master keine Überwachung der Netzwerkknoten durch Heartbeat möglich ist, muss das Objekt vorhanden sein.

1010h - Parameter speichern

Ermöglicht das Abspeichern von Parametern im nichtflüchtigen Speicher.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1010h	-	Store parameters	Array	-	Optional	-
	0h	Largest sub-index supported	Unsigned8	ro	Mandatory	Nein
	1h	Save all parameters	Unsigned32	rw	Mandatory	Nein
	2h	Save communication parameters	Unsigned32	rw	Optional	Nein
	2h	Save application parameters	Unsigned32	rw	Optional	Nein
	4h - 7h	Save manufacturer defined	Unsigned32	rw	Optional	Nein

Durch schreiben des Kommando "save" auf den jeweiligen Subindex wird eine bestimmte Parameterauswahl gespeichert. Das Kommando "save" wird durch folgenden 32 Bit Wert dargestellt:

Byte: MSB LSB

e	v	a	s
65h	76h	61h	73h

Der Umfang der Parameterauswahl ergibt sich aus dem Namen des Subindex.

Das Lesen eines Subindex bringt folgende Information:

Bit	Wert / Bedeutung
31 - 2	*Reserviert*
1	0: Gerät führt keine selbständige Speicherung der Parameter durch (z. B. beim Ausschalten des Geräts) 1: Gerät führt selbständige Speicherung der Parameter durch (z. B. beim Ausschalten des Geräts)
0	0: Parameter können nicht auf Kommando gespeichert werden 1: Parameter können auf Kommando gespeichert werden

1011h - Default Parameter laden

Ermöglicht das Aktivieren von Default Parametern.

Index	Subindex	Name	Datentyp	Zugriff	Kategorie	Mappbar
1011h	-	Restore default parameters	Array	-	Optional	-
	0h	Largest sub-index supported	Unsigned8	ro	Mandatory	Nein
	1h	restore all parameters	Unsigned32	rw	Mandatory	Nein
	2h	restore communication parameters	Unsigned32	rw	Optional	Nein
	2h	restore application parameters	Unsigned32	rw	Optional	Nein
	4h - 7h	restore manufacturer defined	Unsigned32	rw	Optional	Nein

Durch schreiben des Kommando "load" auf den jeweiligen Subindex wird eine bestimmte Default Parameterauswahl geladen und nach dem nächsten Geräte-Reset aktiviert. Das Kommando "load" wird durch folgenden 32 Bit Wert dargestellt:

Byte: MSB LSB

d	a	o	l
64h	61h	6Fh	6Ch

Der Umfang der Parameterauswahl ergibt sich aus dem Namen des Subindex.

Das Lesen eines Subindex bringt folgende Information:

Bit	Wert / Bedeutung
31 - 1	*Reserviert*
0	0: Default Parameter können nicht aktiviert werden 1: Default Parameter können aktiviert werden

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 Identifiervergabe 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

Subindex 1, Bit	Wert / Bedeutung
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
28 - 11	Bits 28 - 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Master 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 Identifiervergabe.

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	*Nicht möglich*
253	Remote	*Nicht möglich*
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 Identifiervergabe 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-Master nicht unterstützt)
28 - 11	Bits 28 - 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Master 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 Identifiervergabe

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

Subindex 2, Wert	Typ	Bedeutung
0	Synchron	Senden erfolgt nach jedem empfangenen SYNC-Telegramm
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 **Subindizes 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 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.2.1 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
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

Objekt	Bedeutung	Mapping-Wert Index Subindex Länge
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
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.2.2 Default-Mapping für Antriebe nach DSP 402

Objekt	Bedeutung	Mapping-Wert Index Subindex Länge
1600h	Erstse 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
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	Achtes 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 Empfangs-PDO.	
Subindex 0	Anzahl der gemappten Objekte	1
Subindex 1	Statuswort (16 Bit)	6041 00 10h
1A01h	Zweites Empfangs-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 Empfangs-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 Empfangs-PDO.	
Subindex 0	Anzahl der gemappten Objekte	2
Subindex 1	Statuswort (16 Bit)	6041 00 10h
Subindex 2	Drehzahl Istwert(32 Bit)	606C 00 20h
1A04h	Fünftes Empfangs-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 Empfangs-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	Siebtes Empfangs-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.3 CANopen Objekte des CANopen-Masters (Softwarestand ≥ 01.20)

B.3.1 Übersicht der Objekte

Index	Objektinhalt / Bedeutung
1000h	Gerätetyp: unterstütztes Geräteprofil und zusätzliche Information
1001h	Fehlerregister
1003h	Fehlerspeicher enthält eine Fehlerliste des Gerätes
1005h	SYNC-Telegramm COB-ID
1006h	SYNC-Intervall
1008h	Hersteller Geräte Name
1009h	Hersteller Hardware Version
100Ah	Hersteller Software Version
1010h	Parameter abspeichern
1011h	Default Parameter laden
1014h	EMCY-Telegramm COB-ID
1016h	Heartbeat Zeiten der Netzwerkknoten
1017h	Heartbeat Zeit des CANopen-Masters
1018h	Identifizierung des Optionsmoduls
1020h	Identifizierung der Konfiguration
1200h	Server SDO Parameter
1280h	Client SDO Parameter
1400h – 14FFh	Empfangs-PDO (RxPDO) Kommunikations Parameter
1600h – 16FFh	Empfangs-PDO (RxPDO) Mapping Parameter
1800h – 18FFh	Sende-PDO (TxPDO) Kommunikations Parameter
1A00h – 1AFFh	Sende-PDO (TxPDO) Mapping Parameter
1F22h	Concise DCF der Netzwerkknoten
1F25h	Anforderung der Konfiguration von Netzwerkknoten
1F80h	NMT Startup
1F81h	Netzwerkknoten Zuweisung
1F82h	Anforderung von NMT Kommandos
1F84h	Identifizierung Netzwerkknoten, Gerätetyp
1F85h	Identifizierung Netzwerkknoten, Hersteller ID
1F86h	Identifizierung Netzwerkknoten, Produkt Code
1F87h	Identifizierung Netzwerkknoten, Revisions Nummer
1F88h	Identifizierung Netzwerkknoten, Seriennummer

Index	Objektinhalt / Bedeutung
1F89h	Wartezeit für Bootvorgang von Netzwerkknoten
2003h	Anwender Gerätename
2004h	Synchronisation mit PLC
2005h	Reaktion auf PLC Status
2020h	Netzwerk Bootup Einstellung
2022h	Wartezeit SDO während des Boot-Vorgangs eines Knotens
2023h	Wartezeit zur Wiederherstellung der Default-Einstellungen
2024h	Wartezeit nach Netzwerk Reset bevor der Scan des Netzwerks beginnt
2025h	Wartezeit bis der nächste Scan eines Knotens beginnt
2100h	Konfiguration Prozessabbild
2101h	Asynchrone und synchrone Bereiche im Prozessabbild
2102h	Konfigurierte asynchrone Objekte im Prozessabbild
2103h	Konfigurierte synchrone Objekte im Prozessabbild

B.3.2 Objekte des CANopen-Masters nach CiA DS 301 und DSP 302

1000h - Gerätetyp

Gibt das unterstützte Geräteprofil des CANopen-Masters und zusätzliche Informationen an.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1000h	0h	device type	Unsigned32	ro	no	Nein

Der 32 Bit-Wert ist in zwei 16 Bit Werte unterteilt:

Byte:	MSB	LSB
Inhalt	Zusätzliche Information, Hersteller abhängig	Geräteprofil
Wert	0000h	0195h

Der CANopen-Master unterstützt das Geräteprofil CiA DS 405 V 2.0 „Interface and Device Profile for IEC 61131-3 Programmable Devices“. Zusätzliche Informationen sind nicht vorhanden.

1001h - Fehlerregister

Enthält eine Bitleiste mit anstehenden Fehlern des CANopen-Masters.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1001h	0h	error register	Unsigned8	ro	no	Nein

Die einzelnen Bits des 8 Bit-Wertes haben folgende Bedeutung:

Bit	Bedeutung	Kategorie
0	Generic error	Mandatory
1	Current	Optional
2	Voltage	Optional
3	Temperature	Optional
4	Communication error	Optional
5	Device profile specific	Optional
6	Reserved	Optional
7	Manufacturer specific	Optional

Ist ein Bit gesetzt, so ist der Fehler aktiv. Dieses Objekt entspricht dem *Error Register* aus dem Emergency-Telegramm, es enthält damit den Wert des *Error Registers* aus dem zuletzt vom CANopen-Master gesendeten Emergency-Telegramm. Da der CANopen-Master selbst keine Emergency-Telegramme generiert gibt das Objekt 1001h nur die Fehler an, welche vom Anwender über den FB COM405_SEND_EMCY gesetzt wurden.

1003h - Fehlerspeicher

Enthält eine Fehlerliste des CANopen-Masters mit absteigender Aktualität.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1003h	-	pre-defined error field	Array	-	-	-
	0h	number of errors	Unsigned8	rw	0	Nein
	1h - 5h	standard error field	Unsigned32	ro	no	Nein

Subindex 0h enthält die Anzahl der eingetragenen Fehler. Die Fehlerliste kann gelöscht werden indem Subindex 0h mit dem Wert 0 beschrieben wird. Subindex 1h enthält den aktuellen Fehler. Tritt ein neuer Fehler auf wird dieser an Subindex 1h eingetragen und alle anderen rutschen um einen Subindex nach unten. Es wird eine Fehler-Historie von max. 5 Einträgen aufgebaut.

Der 32 Bit-Wert des Standard error field ist in zwei 16 Bit Werte unterteilt:

Byte: MSB LSB

Zusätzliche Information, Hersteller abhängig	Fehler Code

Der 16 Bit 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

Error Code	Bedeutung
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

Die 16 Bit zusätzliche Information enthalten beim CANopen-Master die ersten beiden Bytes des *Manufacturer specific Error Field* aus dem Emergency-Telegramm.

Da der CANopen-Master selbst keine Emergency-Telegramme generiert gibt das Objekt 1003h nur die Fehler an, welche vom Anwender über den FB COM405_SEND_EMCY generiert wurden.

1005h - SYNC-Telegramm COB-ID

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

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1005h	0h	COB-ID SYNC	Unsigned32	rw	40000080h	Nein

Die einzelnen Bits des 32 Bit-Wertes haben folgende Bedeutung:

Bit	Wert / Bedeutung
31 (MSB)	Nicht relevant
30	0: CANopen-Master generiert kein SYNC-Telegramm 1: CANopen-Master generiert SYNC-Telegramm Beim CANopen-Master ist die Generierung des SYNC-Telegramms default mäßig aktiviert.
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
28 - 11	Bits 28 – 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
10 – 0 (LSB)	COB-ID des SYNC-Telegramms. Da das Optionsmodul CANopen-Master die COB-ID nach der vordefinierten Identifiervergabe verwendet, wird default mäßig 80h verwendet.

Beachten Sie bitte die weitere Konfiguration der SYNC-Mechanismen über das Objekt 2004h.

1006h - SYNC-Intervall

Länge des SYNC-Intervalls in µs.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1006h	0h	Communication cycle period	Unsigned32	rw	0	Nein

Das Objekt gibt das Zeitintervall an, in welchem der CANopen-Master das SYNC-Telegramm sendet. Der angegebene Wert in µs wird durch den CANopen-Master auf die interne Auflösung von 1 ms umgerechnet, d. h. durch den Wert 1000 geteilt. Das Ergebnis wird ganzzahlig ausgewertet. Damit sind SYNC-Intervalle in Vielfachen von 1 ms möglich. Hat das Objekt 1006h den Wert 0, wird kein SYNC-Telegramm generiert und es erfolgt auch kein Datenaustausch mit synchronen PDOs. Ein Schreiben des Objektes ist nicht im Zustand OPERATIONAL möglich.

HINWEIS

Die Art der Synchronisierung mit der PLC wird über das Objekt 2004h festgelegt. Das Objekt 2004h muss vor dem Objekt 1006h konfiguriert werden!

1008h - Hersteller Geräte Name

Enthält den von der Fa. Baumüller vergebenen Geräte Namen des Optionsmoduls CANopen-Master.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1008h	0h	manufacturer device name	VISIBLE STRING	ro	no	Nein

Das Objekt gibt als Zeichenkette den Namen des Optionsmoduls an, also je nach Ausführungsform z. B. "BM4-O-CAN-04".

1009h - Hersteller Hardware Version

Enthält die von der Fa. Baumüller vergebene Nummer der Hardware-Version des Optionsmoduls CANopen-Master.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1009h	0h	manufacturer hardware version	VISIBLE STRING	ro	no	Nein

Das Objekt gibt als Zeichenkette die Nummer der Hardware-Version des Optionsmoduls an, also je nach Ausführungsform z. B. "3.0107".

100Ah - Hersteller Software Version

Enthält die von der Fa. Baumüller vergebene Nummer der Software-Version des Optionsmoduls CANopen-Master.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
100Ah	0h	manufacturer software version	VISIBLE STRING	ro	no	Nein

Das Objekt gibt als Zeichenkette die Nummer der Software-Version des Optionsmoduls an, also je nach Ausführungsform z. B. "6.1307.0201".

1010h - Parameter speichern

Ermöglicht das Abspeichern von Objektinhalten im nichtflüchtigen Speicher. Die Objektinhalte bleiben nach einem Power Off erhalten und werden nach einem Power On wieder geladen. Ebenso werden die Objektinhalte nach einem RESET aus dem nichtflüchtigen Speicher des CANopen-Masters geladen.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1010h	-	store parameters	Array	-	-	-
	0h	largest sub-index supported	Unsigned8	ro	no	Nein
	1h	save all parameters	Unsigned32	rw	no	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
	2h	save communication parameters	Unsigned32	rw	no	Nein
	4h	save manufacturer defined	Unsigned32	rw	no	Nein

Durch Schreiben des Kommandos "save" im ASCII-Code auf den jeweiligen Subindex werden die Inhalte einer bestimmten Auswahl von Objekten gespeichert. Das Kommando "save" wird durch folgenden 32 Bit Wert dargestellt:

Byte: MSB LSB

e	v	a	s
65h	76h	61h	73h

Der Umfang der Objektauswahl ergibt sich aus dem Subindex:

Subindex	Speicherumfang
1	speichert die Inhalte aller Objekte des CANopen-Masters
2	speichert die Inhalte der Objekte 1000h bis 1FFFh des CANopen-Masters
4	speichert die Inhalte der Objekte 2000h bis 5FFFh des CANopen-Masters

Das Lesen eines Subindex 1 - 4 bringt folgende Information:

Bit	Wert / Bedeutung
31 - 2	*Reserviert*
1	0: Gerät führt keine selbständige Speicherung der Parameter durch (z. B. beim Ausschalten des Geräts) 1: Gerät führt selbständige Speicherung der Parameter durch (z. B. beim Ausschalten des Geräts)
0	0: Parameter können nicht auf Kommando gespeichert werden 1: Parameter können auf Kommando gespeichert werden

1011h - Default Parameter laden

Ermöglicht das Aktivieren von Default Parametern. Es werden die im nichtflüchtigen Speicher gespeicherten Objektinhalte gelöscht.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1011h	-	restore parameters	Array	-	-	-
	0h	largest sub-index supported	Unsigned8	ro	no	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
	1h	restore all parameters	Unsigned32	rw	no	Nein
	2h	restore communication parameters	Unsigned32	rw	no	Nein
	4h	restore manufacturer defined	Unsigned32	rw	no	Nein

Durch Schreiben des Kommandos "load" ASCII-Code auf den jeweiligen Subindex wird eine bestimmte Auswahl von Objektinhalten mit deren Defaultwerten geladen und nach dem nächsten RESET des CANopen-Masters aktiviert. Die Objektinhalte im nichtflüchtigen Speicher werden gelöscht. Das Kommando "load" wird durch folgenden 32 Bit Wert dargestellt:

Byte: MSB LSB

d	a	o	l
64h	61h	6Fh	6Ch

Der Umfang der Objektauswahl ergibt sich aus dem Subindex:

Subindex	Speicherumfang
1	Lädt Defaultwerte aller Objekte des CANopen-Masters
2	Lädt Defaultwerte der Objekte 1000h bis 1FFFh des CANopen-Masters
4	Lädt Defaultwerte der Objekte 2000h bis 5FFFh des CANopen-Masters

Das Lesen eines Subindex 1 - 4 bringt folgende Information:

Bit	Wert / Bedeutung
31 - 1	Reserviert
0	0: Default Parameter können nicht aktiviert werden 1: Default Parameter können aktiviert werden

1014h - EMCY-Telegramm COB-ID

Enthält die COB-ID des Emergency-Telegramms und gibt an, ob das EMCY-Telegramm gültig ist.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1014h	0h	COB-ID Emergency message	Unsigned32	rw	80h + Node-ID	Nein

Die einzelnen Bits des 32 Bit-Wertes haben folgende Bedeutung:

Bit	Wert / Bedeutung
31	0: EMCY-Telegramm ist gültig / aktiviert 1: EMCY-Telegramm ist ungültig / deaktiviert
30	*Reserviert (immer 0)*
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
28 - 11	Bits 28 – 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
10 – 0 (LSB)	COB-ID des EMCY-Telegramms. Da das Optionsmodul CANopen-Master die COB-ID nach der vordefinierten Identifiervergabe verwendet, wird default mäßig 80h + Node-ID verwendet. Wird eine neue COB-ID vergeben, wird die Node-ID nicht mehr berücksichtigt!

Das EMCY-Telegramm kann ausschließlich vom Anwender über den FB COM405_SEND_EMCY generiert werden.

1016h - Heartbeat Zeiten der Netzwerkknoten

Enthält eine Liste mit den Heartbeat Zeiten der einzelnen Netzwerkknoten, welche durch den CANopen-Master überwacht werden sollen.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1016h	-	Consumer Heartbeat Time	Array	-	-	-
	0h	number entries	Unsigned8 (1 – 127)	ro	no	Nein
	1h - 7Fh	Consumer Heartbeat Time	Unsigned32	rw	no	Nein

Das Objekt 1016h enthält in den Subindizes 1 - 127 die erwarteten Heartbeat-Zeiten der einzelnen Netzwerkknoten in Form einer Liste. In dieser Liste stehen nacheinander Einträge "Consumer Heartbeat Time" bestehend aus Knotennummer und zugehöriger Heartbeat Zeit. Maximal können 127 Knoten überwacht werden. Der Eintrag "Consumer Heartbeat Time" hat folgendes Format:

Bits:	31 - 24	23 - 16	15 - 0
Inhalt	*reserviert (= 0)*	Node-ID	Heartbeat Zeit in ms

Die Heartbeat Zeit in ms wird durch den CANopen-Master auf die interne Auflösung von 10 ms umgerechnet, d. h. durch den Wert 10 geteilt. Das Ergebnis wird ganzzahlig ausgewertet. Damit sind Überwachungszeiten in Vielfachen von 10 ms möglich. Ist eine Heartbeat Zeit von 0 eingetragen, wird keine Überwachung durchgeführt.

Der CANopen-Master beginnt die Überwachung eines Knotens mit dem ersten von diesem empfangenen Heartbeat. Wird innerhalb der angegebenen Zeit kein Heartbeat von einem Knoten empfangen, kann der CANopen-Master darauf reagieren. Die Art der Reaktion wird durch die Objekte 1F80h und 1F81h festgelegt. Es ist sinnvoll die Heartbeat Zeit auf dem zu überwachenden Netzwerkknoten (Objekt 1017h des Netzwerkknotens) niedriger einzustellen, als beim CANopen-Master im Objekt 1016h.

1017h - Heartbeat Zeit des CANopen-Masters

Gibt die Heartbeat Zeit des CANopen-Masters an.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1017h	0h	Producer Hearbeat Time	Unsigned16	rw	0	Nein

Mit dem in diesem Objekt eingetragenen Zeitintervall "Producer Hearbeat Time" sendet der CANopen-Master Heartbeat-Telegramme. Die Heartbeat Zeit wird in ms angegeben, wobei nur Vielfache von 10 ms möglich sind, also 10 ms, 20 ms, 30 ms, ... Zwischenwerte werden intern aufgerundet. Ist das Zeitintervall 0 wird kein Heartbeat gesendet. Sobald ein Wert ungleich Null eingetragen wird, beginnt die Heartbeat Übertragung.

1018h - Identifizierung des Optionsmoduls

Spezifiziert das Optionsmodul.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1018h	-	Identity Object	Record	-	-	-
	0h	number of entries	Unsigned8	ro	no	Nein
	1h	Vendor ID	Unsigned32	ro	no	Nein
	2h	Product code	Unsigned32	ro	no	Nein
	3h	Revision number	Unsigned32	ro	no	Nein
	4h	Serial number	Unsigned32	ro	no	Nein

Das Objekt 1018h enthält Daten zur genaueren Identifizierung des Optionsmoduls CANopen-Master. Die Subindizes 1 - 3 haben folgende Bedeutung:

Subindex	Name	Bedeutung
1	Vendor ID	Hersteller ID. Die Fa. Baumüller Nürnberg GmbH hat die ID 346
2	Product code	noch keine Verwendung, 0
3	Revision number	noch keine Verwendung, 0
4	Serial number	Seriennummer des Optionsmoduls

1020h - Identifizierung der Konfiguration

Ermöglicht die Identifizierung der Konfiguration des CANopen-Masters durch Datum und Uhrzeit.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1020h	-	Verify Configuration	Array	-	-	-
	0h	number of entries	Unsigned8	ro	no	Nein
	1h	Configuration date	Unsigned32	rw	no	Nein
	2h	Configuration time	Unsigned32	rw	no	Nein

Das Objekt 1020h enthält Datum und Uhrzeit zur genaueren Identifizierung der Konfiguration des CANopen-Masters. In Subindex 1 kann vom Konfigurationstool oder Anwender ein Datum, in Subindex 2 eine Zeit eingetragen werden. Die Inhalte von Subindex 1 und 2 werden vom CANopen-Master gelöscht, sobald ein neues Objekt geschrieben wird. Eine Ausnahme bilden die Objekte 1010h, 1020h, 1F25h, 1F82h und die Objekte des Prozessimage (A000h usw.), ein Schreiben dieser Objekte führt zu keiner Löschung der Inhalte von Subindex 1 und 2.

Nachdem Datum und Uhrzeit eingetragen sind, ist die gesamte Konfiguration über das Objekt 1010h zu speichern.

Im Detail haben die Subindizes 1 - 3 folgende Bedeutung:

Subindex	Name	Bedeutung
1	Configuration date	Es ist die Anzahl der Tage seit dem 1. Januar 1984 für das Konfigurationsdatum einzutragen
2	Configuration time	Es ist die Anzahl der Millisekunden seit Mitternacht für das Konfigurationsdatum einzutragen

1200h - Server SDO Parameter

Enthält die Parameter für den Zugriff auf das Objektverzeichnis des CANopen-Masters über SDO.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1200h	-	Server SDO parameter	Record	-	-	-
	0h	number of entries	Unsigned8	ro	no	Nein
	1h	COB-ID Client -> Server (rx)	Unsigned32	ro	600h + Node-ID	Nein
	2h	COB-ID Server -> Client (tx)	Unsigned32	ro	580h + Node-ID	Nein

Dieses Objekt enthält die Parameter für einen SDO Zugriff auf das Objektverzeichnis des CANopen-Masters. Der CANopen-Master ist in diesem Fall der Server. Client kann ein anderer Netzwerkknoten sein, aber auch die Funktionsbausteine für SDO-Kommunikati-

on aus der Bibliothek xx.xx sofern auf den Master selbst zugegriffen wird. Die Einträge in Subindex 1 und 2 haben folgenden Aufbau:

Bit	Wert / Bedeutung
31	0: SDO ist gültig / aktiviert 1: SDO ist gültig ist ungültig / deaktiviert
30	*Reserviert (immer 0)*
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
28 - 11	Bits 28 – 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
10 – 0 (LSB)	COB-ID des SDO-Telegramms. Da das Optionsmodul CANopen-Master die COB-ID nach der vordefinierten Identifiervergabe verwendet, wird default mäßig 600h + Node-ID für die Richtung Client -> Server und 580h + Node-ID für die Richtung Server -> Client verwendet. Die Node-ID ist die Knotennummer des CANopen-Masters.

1280h - Client SDO Parameter

Der CANopen-Master verwaltet SDO-Zugriffe auf die Objektverzeichnisse anderer Netzwerkknoten intern. Eine Konfiguration der Parameter dieses Objektes (z. B. COB-ID) durch den Anwender ist nicht möglich. Die COB-IDs ergeben sich durch die Vergabe über Objekt 1200h, d. h. für Anfragen über SDOs welche an einen Netzwerkknoten gesendet werden wird 600h + Node-ID verwendet, für die zugehörige Antwort wird die COB-ID 580h + Node-ID erwartet.

1400h bis 14FFh - Empfangs-PDO (RxPDO) Kommunikations Parameter

Die Objekte 1400h bis 14FFh legen die Eigenschaften der Übertragung von RxPDO 1 bis RxPDO 256 des CANopen-Masters fest. Das Objekt 1400h beschreibt die RxPDO 1, Objekt 1401h die RxPDO 2, ..., Objekt 14FFh die RxPDO 256.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1400h	-	RxPDO 1 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	2	Nein
	1h	COB-ID Receive RxPDO 1	Unsigned32	rw	NODEID+0x200	Nein
	2h	Transmission type RxPDO 1	Unsigned8	rw	255	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1401h	-	RxPDO 2 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	2	Nein
	1h	COB-ID Receive RxPDO 2	Unsigned32	rw	NODEID+0x300	Nein
	2h	Transmission type RxPDO 2	Unsigned8	rw	255	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1402h	-	RxPDO 3 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	2	Nein
	1h	COB-ID Receive RxPDO 3	Unsigned32	rw	NODEID+0x400	Nein
	2h	Transmission type RxPDO 3	Unsigned8	rw	255	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1403h	-	RxPDO 4 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	2	Nein
	1h	COB-ID Receive RxPDO 4	Unsigned32	rw	NODEID+0x500	Nein
	2h	Transmission type RxPDO 4	Unsigned8	rw	255	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1404h – 14FFh	-	RxPDO 5 – 256 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	2	Nein
	1h	COB-ID Receive RxPDO 5 – 256	Unsigned32	rw	0x80000000	Nein
	2h	Transmission type RxPDO 5 – 256	Unsigned8	rw	-	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 Es sind nur die ersten 4 RxPDOs des CANopen-Masters aktiviert, alle anderen RxPDOs sind deaktiviert. Werden mehr als 4 RxPDOs verwendet, müssen die COB-IDs für die zusätzlichen RxPDOs selbst vergeben werden (Bits 10 – 0 in diesem Subindex). Aus diesem Grund hat das Bit 31 der RxPDOS 5 – 256 den Wert 1 und ist erst dann vom Anwender auf 0 zu setzen, sobald dieser eine COB-ID vergeben hat.
30	0: das RxPDO kann über einen Remote-Request angefordert werden 1: das RxPDO kann nicht über einen Remote-Request angefordert werden
29	0: 11-Bit CAN ID 1: 29-Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
28 - 11	Bits 28 – 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
10 – 0 (LSB)	COB-ID der RxPDO. Die ersten 4 RxPDOs haben eine Default-Einstellung.

Mit Subindex 2 wird der Übertragungstyp des RxPDO festgelegt:

Subindex 2, Wert	Typ	Bedeutung
0	*nicht unterstützt*	
1 – 240	Synchron	Vor dem letzten SYNC-Telegramm empfangenes RxPDO mit passender COB-ID wird übernommen
254	Asynchron	Jedes RxPDO mit passender COB-ID wird bei Empfang übernommen
255	Asynchron	Jedes RxPDO mit passender COB-ID wird bei Empfang übernommen

1600h bis 16FFh - Empfangs-PDO (RxPDO) Mapping Parameter

Die Objekte 1600h bis 16FFh legen die gemappten Objekte der RxPDO 1 bis RxPDO 256 des CANopen-Masters fest. Das Objekt 1600h beschreibt die RxPDO 1, Objekt 1601h die RxPDO 2, ..., Objekt 16FFh die RxPDO 256.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1600h – 16FFh	-	RxPDO mapping	Record	-	-	-
	0h	Number of mapped objects	Unsigned8	rw	0	Nein
	1h – 8h	Object to be mapped	Unsigned32	rw	0	Nein

Subindex 0 gibt die Anzahl der gemappten Objekte in dem RxPDO an. Sobald das Mapping eines RxPDO geändert werden soll, muss man zunächst die Anzahl der gemappten Objekte auf Null setzen. Anschließend kann das Mapping eingestellt werden und im letzten Schritt muss die Anzahl der gemappten Objekte in Subindex 0 eingetragen werden. In **Subindex 1h bis 8h** werden die Mapping Informationen der Objekte in dem RxPDO angegeben. Subindex 1h enthält die Informationen für das erste gemappte Objekt, Subindex 2h für das zweite gemappte Objekt usw... Maximal können 8 (= 8h) 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 bis 8h** 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 RxPDO gemappt.

1800h bis 18FFh - Sende-PDO (TxPDO) Kommunikations Parameter

Die Objekte 1800h bis 18FFh legen die Eigenschaften zur Übertragung der TxPDO 1 bis TxPDO 256 des CANopen-Masters fest. Das Objekt 1800h beschreibt die TxPDO 1, Objekt 1801h die TxPDO 2, ..., Objekt 18FFh die TxPDO 256.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1800h	-	TxPDO 1 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	5	Nein
	1h	COB-ID Receive TxPDO 1	Unsigned32	rw	NODEID+0x180	Nein
	2h	Transmission type TxPDO 1	Unsigned8	rw	255	Nein
	5h	Event Timer TxPDO 1	Unsigned16	rw	0	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1801h	-	TxPDO 2 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	5	Nein
	1h	COB-ID Receive TxPDO 2	Unsigned32	rw	NODEID+0x280	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
	2h	Transmission type TxPDO 2	Unsigned8	rw	255	Nein
	5h	Event Timer TxPDO 2	Unsigned16	rw	0	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1802h	-	TxPDO 3 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	5	Nein
	1h	COB-ID Receive TxPDO 3	Unsigned32	rw	NODEID+0x380	Nein
	2h	Transmission type TxPDO 3	Unsigned8	rw	255	Nein
	5h	Event Timer TxPDO 3	Unsigned16	rw	0	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1803h	-	TxPDO 4 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	5	Nein
	1h	COB-ID Receive TxPDO 4	Unsigned32	rw	NODEID+0x480	Nein
	2h	Transmission type TxPDO 4	Unsigned8	rw	255	Nein
	5h	Event Timer TxPDO 4	Unsigned16	rw	0	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1804h – 18FFh	-	TxPDO 5 – 256 parameter	Record	-	-	-
	0h	Largest sub-index supported	Unsigned8	ro	5	Nein
	1h	COB-ID Receive TxPDO 5 – 256	Unsigned32	rw	0x80000000	Nein
	2h	Transmission type TxPDO 5 – 256	Unsigned8	rw	-	Nein
	5h	Event Timer TxPDO 5 – 256	Unsigned16	rw	0	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 Es sind nur die ersten 4 TxPDOs des CANopen-Masters aktiviert, alle anderen TxPDOs sind deaktiviert. Werden mehr als 4 TxPDOs verwendet müssen die COB-IDs für die zusätzlichen TxPDOs selbst vergeben werden (Bits 10 – 0 in diesem Subindex). Aus diesem Grund hat das Bit 31 der TxPDOS 5 – 256 den Wert 1 und ist erst dann vom Anwender auf 0 zu setzen, 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-Master nicht unterstützt)
28 - 11	Bits 28 – 11 bei 29 Bit CAN ID (wird vom Optionsmodul CANopen-Master nicht unterstützt)
10 – 0 (LSB)	COB-ID der TxPDO. Die ersten 4 TxPDOs haben eine Default-Einstellung.

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

Subindex 2, Wert	Typ	Bedeutung
0	*nicht unterstützt*	
1 – 240	Synchron	Senden erfolgt nach Senden der eingestellten Anzahl (1 – 240) von SYNC-Telegrammen
254	Asynchron	Das TxPDO wird gesendet sobald sich der Wert eines gemappten Objektes geändert hat oder ein Zeit-Event (Ablauf des Event Timers, Subindex 5) aufgetreten ist.
255	Asynchron	Das TxPDO wird gesendet sobald sich der Wert eines gemappten Objektes geändert hat oder ein Zeit-Event (Ablauf des Event Timers, Subindex 5) aufgetreten ist.

In **Subindex 5** wird die Zeit in ms für die Ereignis gesteuerte PDO Übertragung des Zeit-Events eingestellt. Ist der Wert null, wird kein Zeit-Event ausgelöst.

1A00h bis 1AFFh - Sende-PDO (TxPDO) Mapping Parameter

Die Objekte 1A00h bis 1AFFh legen die gemappten Objekte der TxPDO 1 bis TxPDO 256 des CANopen-Masters fest. Das Objekt 1A00h beschreibt die TxPDO 1, Objekt 1A01h die TxPDO 2, ..., Objekt 1AFFh die TxPDO 256.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1A00h – 1AFFh	-	TxPDO mapping	Record	-	-	-
	0h	Number of mapped objects	Unsigned8	rw	0	Nein
	1h – 8h	Object to be mapped	Unsigned32	rw	0	Nein

Subindex 0 gibt die Anzahl der gemappten Objekte in dem TxPDO an. Sobald das Mapping eines TxPDO geändert werden soll, muss man zunächst die Anzahl der gemappten Objekte auf Null setzen. Anschließend kann das Mapping eingestellt werden und im letzten Schritt muss die Anzahl der gemappten Objekte in Subindex 0 eingetragen werden. In **Subindex 1h bis 8h** 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 8 (= 8h) Objekte gemappt werden, wobei der maximale Datenbereich von 8 Byte der TxPDO nicht überschritten werden darf. Die Inhalte des 32 Bit-Wertes der **Subindizes 1h - 8h** 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 TxPDO gemappt.

1F22h - Concise DCF der Netzwerkknoten

Enthält eine Liste mit den "concise DCF" der einzelnen Netzwerkknoten, welche durch den CANopen-Master während des Netzwerk-Bootups auf die einzelnen Knoten geladen werden.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F22h	-	Concise DCF	Domain	-	-	-
	0h	number entries	Unsigned8 (1 – 127)	ro	127	Nein
	1h - 7Fh	Concise DCF nodes	Domain	rw	no	Nein

Das Objekt 1F22h enthält in den Subindizes 1 - 127 die "concise DCF" der einzelnen Netzwerkknoten. Ein Subindex ist der Nummer des Netzwerkknotens zugeordnet, d. h. z. B., dass das "concise DCF" von Knoten 18 in Subindex 18 steht. Ein "concise DCF"

(DCF = Device Configuration File) ist eine Konfiguration, welche durch den CANopen-Master während des Netzwerk-Bootups auf die einzelnen Knoten geladen werden. Die Konfiguration wird durch eine Abfolge von Objekten mit zugehörigen Werten beschrieben. Diese Konfiguration bzw. das "concise DCF" hat folgenden Aufbau:

Eintrag	**Datentyp**
Anzahl der Objekte	Unsigned32
Index 1	Unsigned16
Subindex 1	Unsigned8
Datenlänge des Objektes in Byte	Unsigned32
Daten	Domain
Index 2	Unsigned16
Subindex 2	Unsigned8
Datenlänge des Objektes in Byte	Unsigned32
Daten	Domain
Index 2	Unsigned16
Subindex 2	Unsigned8
Datenlänge des Objektes in Byte	Unsigned32
Daten	Domain
:	:
Index n	Unsigned16
Subindex n	Unsigned8
Datenlänge des Objektes in Byte	Unsigned32
Daten	Domain

Am Anfang der Konfiguration steht die Anzahl der nachfolgenden Objekte. Danach folgen die einzelnen Objekte mit Index, Subindex, Datenlänge und Daten. Insgesamt können maximal 65535 Byte in den Subindizes eingetragen werden. Dieser Wert gilt über alle Subindizes und nicht für einen einzelnen!

Beispiel:

Auf dem Knoten mit der Nummer 18 soll während des Netzwerk-Bootups ein Sync-Intervall von 10 ms (Objekt 1006h), eine Guarding-Zeit von 250 ms (Objekt 100Ch) und ein Life-Time-Factor von 4 (Objekt 100Dh) eingestellt werden. Die Datenlängen ergeben sich nach CiA DS 301 (oder auch Handbuch zum Knoten) zu 32 Bit, 16 Bit und 8 Bit. Im Objekt 1F22h, Subindex 18 des CANopen-Masters werden dann über einen SDO Transfer folgende Daten eingetragen:

Byte	Wert	Bedeutung
0	0x03	Anzahl der Objekte
1	0x00	
2	0x00	
3	0x00	
4	0x06	Index Objekt 1
5	0x10	
6	0x00	Subindex Objekt 1
7	0x04	Datenlänge Objekt 1
8	0x00	
9	0x00	
10	0x00	
11	0x10	Daten Objekt 1
12	0x27	
13	0x00	
14	0x00	
15	0x0C	Index Objekt 2
16	0x10	
17	0x00	Subindex Objekt 2
18	0x02	Datenlänge Objekt 2
19	0x00	
20	0x00	
21	0x00	
22	0xFA	Daten Objekt 2
23	0x00	
24	0x0D	Index Objekt 3
25	0x10	
26	0x00	Subindex Objekt 3
27	0x01	Datenlänge Objekt 3
28	0x00	
29	0x00	
30	0x00	
31	0x04	Daten Objekt 3

Die beim SDO Transfer anzugebende Datenlänge beträgt 32 Byte.

1F25h - Anforderung der Konfiguration von Netzwerkknoten

Ermöglicht die Anforderung der Konfiguration von Netzwerkknoten durch den CANopen-Master während des Betriebs.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F25h	-	ConfigureSlave	ARRAY	-	-	-
	0h	number entries	Unsigned8	ro	128	Nein
	1h - 7Fh	Request re-configuration node	Unsigned32	rw	no	Nein
	80h	Request re-configuration all nodes	Unsigned32	rw	no	Nein

Über das Objekt 1F25h kann die Neu- / Rekonfigurierung eines Netzwerkknotens durch den CANopen-Master während des Betriebs angefordert werden, ohne einen kompletten Bootup des Netzwerks durchführen zu müssen. Dies kann z. B. notwendig sein, wenn neue Knoten im Netzwerk hinzukommen. Ausgelöst wird die Rekonfigurierung durch Schreiben des Kommandos "conf" auf den jeweiligen Subindex. Die Subindizes 1 bis 127 entsprechen der Nummer des zu konfigurierenden Knotens. Wird das Kommando "conf" auf den Subindex 128 geschrieben, werden alle Knoten rekonfiguriert. Das Kommando "conf" wird durch folgenden 32 Bit Wert dargestellt:

Byte: MSB LSB

f	n	o	c
66h	6Eh	6Fh	63h

Der neu zu konfigurierende Knoten darf sich nicht im Zustand OPERATIONAL befinden. Ist im Objekt 1F81h das Bit 2 für den jeweiligen Netzwerkknoten (entsprechender Subindex) gesetzt, führt der CANopen-Master eine Konfigurierung automatisch durch. Details sind in der Beschreibung zum Objekt 1F81h zu finden.

1F80h - NMT Startup

Konfiguriert das Startup-Verhalten des CANopen-Masters.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F80h	0h	NMT Startup	Unsigned32	rw	0000000Dh	Nein

Die einzelnen Bits des 32 Bit-Wertes haben folgende Bedeutung:

Bit	Wert / Bedeutung
0	0: *nicht möglich* 1: Optionsmodul CANopen-Master ist NMT Master
1	0: Nach Bootup des Netzwerks nur Knoten in den Zustand „OPERATIONAL“ setzen, welche als Slave konfiguriert sind (Objekt 1F81h, Bit 0) 1: Nach Bootup des Netzwerks alle Knoten in den Zustand „OPERATIONAL“ setzen.

Bit	Wert / Bedeutung
2	0: CANopen-Master geht nach Bootup des Netzwerks automatisch in den Zustand „OPERATIONAL“ 1: CANopen-Master muss von Applikation in den Zustand „OPERATIONAL“ gesetzt werden (z.B. über FB CIA405_NMT)
3	0: Die Slaves werden nach Bootup des Netzwerks automatisch in den Zustand „OPERATIONAL“ gesetzt 1: Die Slaves müssen von der Applikation in den Zustand „OPERATIONAL“ gesetzt werden (z.B. über FB CIA405_NMT)
4	0: Bei Auftreten eines Error Control Events (z. B. EMCY-Nachricht, Node Guarding oder Heartbeat Fehler) eines mandatory Slaves (siehe Objekt 1F81h, Bit 3) wird der Slave gesondert behandelt (siehe Objekt 1F81h) 1: Bei Auftreten eines Error Control Events (z. B. EMCY-Nachricht, Node Guarding oder Heartbeat Fehler) eines mandatory Slaves (siehe Objekt 1F81h, Bit 3) wird ein Reset aller Slaves durchgeführt
5 – 31 (MSB)	*reserviert*

1F81h - Netzwerkknoten Zuweisung

Enthält eine Liste mit den dem CANopen-Master zugewiesenen Netzwerkknoten und definiert das Verhalten des CANopen-Masters im Fehlerfall o.ä..

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F81h	-	SlaveAssignment	Array	-	-	-
	0h	number entries	Unsigned8 (127)	ro	no	Nein
	1h - 7Fh	SlaveAssignment	Unsigned32	rw	no	Nein

Das Objekt 1F81h enthält in den Subindizes 1 - 127 die Zuweisung einzelner Netzwerkknoten zum CANopen-Master in Form einer Liste. Der Index in der Liste entspricht der Nummer des Netzwerkknotens im CANopen-Netzwerk.

Im Detail beschreibt der Inhalt eines Subindex Folgendes:

Bit	Wert / Bedeutung
0	0: Der Knoten mit dieser Nummer ist kein Knoten im CANopen-Netzwerk, d. h. bei nicht vorhandenen Knoten muss im zugehörigen Subindex Bit 0 = 0 sein. 1: Der Knoten mit dieser Nummer ist ein Knoten im CANopen-Netzwerk
1	0: Bei einem „Error Control Event“ oder anderweitiger Erkennung eines bootenden Slaves wird die Applikation informiert 1: Bei einem „Error Control Event“ oder anderweitiger Erkennung eines bootenden Slaves wird die Applikation informiert und automatisch der „Error Control Service“ gestartet

Bit	Wert / Bedeutung
2	0: Bei einem „Error Control Event“ oder anderweitiger Erkennung eines bootenden Slaves wird dieser **nicht** automatisch konfiguriert und gestartet 1: Bei einem „Error Control Event“ oder anderweitiger Erkennung eines bootenden Slaves wird dieser konfiguriert und gestartet.
3	0: optionaler Slave: Netzwerk wird auch gestartet, wenn sich dieser Knoten nicht meldet 1: mandatory Slave: Netzwerk wird nicht gestartet, wenn sich dieser Knoten nicht meldet
4,5,6	*wird nicht unterstützt*
7	*reserviert*
8 – 15 (Byte 1)	Retry Factor für Node Guarding. Ist hier ein Wert eingetragen, wird dieser vom CANopen-Master für das Node Guarding des Netzwerkknotens verwendet
16 – 31 (Byte 2+3)	Guard Time für Node Guarding. Ist hier ein Wert eingetragen, wird dieser vom CANopen-Master für das Node Guarding des Netzwerkknotens verwendet. Die Guard Time wird in ms angegeben, wobei nur Vielfache von 10 ms möglich sind, also 10 ms, 20 ms, 30 ms,.... Zwischenwerte werden intern aufgerundet.

1F82h - Anforderung von NMT Kommandos

Ermöglicht die Anforderung der Ausführung von NMT Kommandos vom CANopen-Master während des Betriebs.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F82h	-	Request NMT	ARRAY	-	-	-
	0h	number entries	Unsigned8	ro	128	Nein
	1h - 7Fh	Request NMT	Unsigned8	rw	no	Nein
	80h	Request NMT all nodes	Unsigned8	wo	no	Nein

Über das Objekt 1F82h kann vom CANopen-Master die Ausführung eines NMT Kommandos angefordert werden um einen Knotenzustand zu ändern. Ausgelöst wird das Kommando durch Schreiben des angeforderten Kommandos auf den jeweiligen Subindex. Die Subindizes 1 bis 127 entsprechen der Nummer des Knotens für welchen das Kommando gültig ist. Wird das Kommando auf den Subindex 128 geschrieben, ist das Kommando für alle Knoten gültig. Folgende Kommandos sind möglich:

Knotenzustand	Kommando / Wert
Stopped	4
Operational	5

Knotenzustand	Kommando / Wert
Reset Node	6
Reset Communication	7
PreOperational	127

1F84h - Identifizierung Netzwerkknoten, Gerätetyp

Ermöglicht die Angabe des erwarteten Gerätetyps eines Netzwerkknotens beim Bootup des Netzwerks.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F84h	-	Device Type Identification	ARRAY	-	-	-
	0h	number entries	Unsigned8	ro	127	Nein
	1h - 7Fh	Device Type Identification	Unsigned32	rw	no	Nein

Beim Bootup des Netzwerks kann der CANopen-Master die Gerätetypen, also die Inhalte der Objekte 1000h, überprüfen. Hierzu werden die Einträge im Objekt 1F84h verwendet. Die Subindizes 1 bis 127 entsprechen der Nummer des Knotens für welchen der Gerätetyp überprüft wird. Ist in einem Subindex kein Wert eingetragen, erfolgt keine Überprüfung. Ist ein Wert eingetragen, wird dieser mit dem Inhalt des Objektes 1000h auf dem Netzwerkknoten verglichen. Bei Unterschieden erfolgt eine Fehlermeldung (FB COM405_NETWORK_CONTROL).

1F85h - Identifizierung Netzwerkknoten, Hersteller ID

Ermöglicht die Angabe der erwarteten Hersteller ID eines Netzwerkknotens beim Bootup des Netzwerks.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F85h	-	Vendor Identification	ARRAY	-	-	-
	0h	number entries	Unsigned8	ro	127	Nein
	1h - 7Fh	Vendor Identification	Unsigned32	rw	no	Nein

Beim Bootup des Netzwerks kann der CANopen-Master die Hersteller IDs, also die Inhalte der Objekte 1018h (Subindex 1), überprüfen. Hierzu werden die Einträge im Objekt 1F85h verwendet. Die Subindizes 1 bis 127 entsprechen der Nummer des Knotens für welchen die Hersteller ID überprüft wird. Ist in einem Subindex kein Wert eingetragen, erfolgt keine Überprüfung. Ist ein Wert eingetragen, wird dieser mit dem Inhalt des Objektes 1018h, Subindex 1 auf dem Netzwerkknoten verglichen. Bei Unterschieden erfolgt eine Fehlermeldung (FB COM405_NETWORK_CONTROL). Es ist zu berücksichtigen, dass nicht alle CANopen-Netzwerknoten das Objekt 1018h unterstützen und somit eine Überprüfung nicht möglich ist.

1F86h - Identifizierung Netzwerkknoten, Produkt Code

Ermöglicht die Angabe des erwarteten Produkt Codes eines Netzwerkknotens beim Bootup des Netzwerks.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F86h	-	Product Code	ARRAY	-	-	-
	0h	number entries	Unsigned8	ro	127	Nein
	1h - 7Fh	Product Code	Unsigned32	rw	no	Nein

Beim Bootup des Netzwerks kann der CANopen-Master die Produkt Codes, also die Inhalte der Objekte 1018h (Subindex 2), überprüfen. Hierzu werden die Einträge im Objekt 1F86h verwendet. Die Subindizes 1 bis 127 entsprechen der Nummer des Knotens für welchen der Produkt Code überprüft wird. Ist in einem Subindex kein Wert eingetragen, erfolgt keine Überprüfung. Ist ein Wert eingetragen, wird dieser mit dem Inhalt des Objektes 1018h, Subindex 2 auf dem Netzwerkknoten verglichen. Bei Unterschieden erfolgt eine Fehlermeldung (FB COM405_NETWORK_CONTROL). Es ist zu berücksichtigen, dass nicht alle CANopen-Netzwerknoten das Objekt 1018h unterstützen und somit eine Überprüfung nicht möglich ist.

1F87h - Identifizierung Netzwerkknoten, Revisions Nummer

Ermöglicht die Angabe der erwarteten Revisions Nummer eines Netzwerkknotens beim Bootup des Netzwerks.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F87h	-	Revision Number	ARRAY	-	-	-
	0h	number entries	Unsigned8	ro	127	Nein
	1h - 7Fh	Revision Number	Unsigned32	rw	no	Nein

Beim Bootup des Netzwerks kann der CANopen-Master die Revisions Nummern, also die Inhalte der Objekte 1018h (Subindex 3), überprüfen. Hierzu werden die Einträge im Objekt 1F87h verwendet. Die Subindizes 1 bis 127 entsprechen der Nummer des Knotens für welchen die Revisions Nummer überprüft wird. Ist in einem Subindex kein Wert eingetragen, erfolgt keine Überprüfung. Ist ein Wert eingetragen, wird dieser mit dem Inhalt des Objektes 1018h, Subindex 3 auf dem Netzwerkknoten verglichen. Bei Unterschieden erfolgt eine Fehlermeldung (FB COM405_NETWORK_CONTROL). Es ist zu berücksichtigen, dass nicht alle CANopen-Netzwerknoten das Objekt 1018h unterstützen und somit eine Überprüfung nicht möglich ist.

1F88h - Identifizierung Netzwerkknoten, Seriennummer

Ermöglicht die Angabe der erwarteten Seriennummer eines Netzwerkknotens beim Bootup des Netzwerks.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F88h	-	Serial Number	ARRAY	-	-	-
	0h	number entries	Unsigned8	ro	127	Nein
	1h - 7Fh	Serial Number	Unsigned32	rw	no	Nein

Beim Bootup des Netzwerks kann der CANopen-Master die Seriennummern, also die Inhalte der Objekte 1018h (Subindex 4), überprüfen. Hierzu werden die Einträge im Objekt 1F88h verwendet. Die Subindizes 1 bis 127 entsprechen der Nummer des Knotens für welchen die Seriennummer überprüft wird. Ist in einem Subindex kein Wert eingetragen, erfolgt keine Überprüfung. Ist ein Wert eingetragen, wird dieser mit dem Inhalt des Objektes 1018h, Subindex 2 auf dem Netzwerkknoten verglichen. Bei Unterschieden erfolgt eine Fehlermeldung (FB COM405_NETWORK_CONTROL). Es ist zu berücksichtigen, dass nicht alle CANopen-Netzwerknoten das Objekt 1018h unterstützen und somit eine Überprüfung nicht möglich ist.

1F89h - Wartezeit für Bootvorgang von Netzwerkknoten

Länge der Wartezeit für den Bootvorgang von "mandatory" Netzwerkknoten in ms.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
1F89h	0h	Boot Time	Unsigned32	rw	0	Nein

Das Objekt gibt die Zeit in ms, bis zu der auf die Bootup-Nachricht von "mandatory" Slaves (siehe Objekt 1F81h) gewartet wird. Empfängt der CANopen-Master nach dem Start des Bootups des Netzwerks innerhalb dieser Zeit nicht alle Bootup-Nachrichten von "mandatory" Slaves, wird ein Fehler ausgelöst (FB COM405_NETWORK_CONTROL). Eine Zeit von 0 ms besagt, dass der CANopen-Master endlos auf die Bootup-Nachricht eines "mandatory" Slaves wartet. Der Wartezustand wird am FB COM405_NETWORK_CONTROL angezeigt.

B.3.3 Hersteller spezifische Objekte des CANopen-Masters

2001h - Baudrate und Node-ID des CANopen-Master

Dieses Objekt wird noch nicht unterstützt!

Legt die Baudrate und die Node-ID des CANopen-Masters fest. Die Werte werden abhängig von den Einstellungen des Objektes 2002h verwendet. Hat das Objekt 2002h den Wert 0, so werden Baudrate und Node-ID über einen IEC 61131-3 Funktionsbaustein eingestellt und die im Objekt 2001h eingestellten Werte werden nicht berücksichtigt.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2001h	0h	Baudrate and Node-ID	Unsigned32	rw	00 7F h	Nein

Die Inhalte des 32 Bit-Wertes von Subindex 0h haben folgende Bedeutung:

Byte:	MSB		LSB
Inhalt	reserviert (16 Bit)	Baudrate (8 Bit)	Node-ID (8 Bit)
Wertebereich	-	0 - 8	1 - 127

Für die Baudrate sind folgende Werte zulässig:

Wert für Baudrate	Bedeutung
0	1 MBit/s
1	800 kBit/s
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	100 kBit/s
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s

Defaultwert für die Baudrate ist 0 (1 MBit/s). Defaultwert für die Node-ID ist 127.

2002h - Aufstartverhalten des CANopen-Master

Dieses Objekt wird noch nicht unterstützt!

Gibt an, bis zu welchem Punkt der CANopen-Master hoch läuft und dann die Steuerung an IEC 61131-3 Funktionsbausteine übergibt.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2002h	0h	Startup Control Master	Unsigned8	rw	0	Nein

Die einzelnen Werte haben folgende Bedeutung:

Wert	Bedeutung
0	Die Initialisierung und der Start zum Booten des Netzwerks werden von IEC 61131-3 Funktionsbausteinen ausgelöst.
1	Die Initialisierung wird automatisch durchgeführt. Für Baudrate und Node-ID des CANopen-Masters werden die Inhalte des Objekts 2001h verwendet. Der Start zum Booten des Netzwerks wird von IEC 61131-3 Funktionsbausteinen ausgelöst.
2	Die Initialisierung wird automatisch durchgeführt. Für Baudrate und Node-ID des CANopen-Masters werden die Inhalte des Objekts 2001h verwendet. Der Bootup Vorgang des Netzwerks wird automatisch angestoßen.
3 - 255	*reserviert*

Die eingestellten Werte werden erst nach einem Aus-/Einschalten des Gerätes wirksam. Ein Abspeichern der Hersteller spezifischen Objekte über das Objekt 1010h muss bei einer Änderung durchgeführt werden.

2003h - Anwender Gerätename

Mit diesem Objekt kann der Anwender einen eigenen Gerätenamen vergeben.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2003h	0h	User Device Name	Visible String	rw	0	Nein

In **Subindex 0h** kann ein frei wählbarer Gerätename mit max. 128 Zeichen eingetragen werden.

2004h - Synchronisation mit PLC

Gibt an auf welche Art die Synchronisierung mit der PLC erfolgt.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2004h	0h	Synchronisation PLC	Unsigned8	rw	0	Nein

Die einzelnen Werte haben folgende Bedeutung:

Wert	Bedeutung
0	Es erfolgt keine Synchronisierung mit der PLC.
1	Das Optionsmodul generiert einen internen, synchronen IRP abhängig vom Signal „SYNC-Signal 1“, Ereignis 11 für Event-Tasks unter PROPROG wt II. Mit dem generierten, synchronen IRP werden das SYNC-Telegramm gesendet und die synchronen TxPDOs übertragen. Ethernet wird nicht abgeschaltet.
2	Das Optionsmodul generiert einen internen, synchronen IRP abhängig vom Signal „SYNC-Signal 2“, Ereignis 12 für Event-Tasks unter PROPROG wt II. Mit dem generierten, synchronen IRP werden das SYNC-Telegramm gesendet und die synchronen TxPDOs übertragen. Ethernet wird nicht abgeschaltet.
3 - 255	*reserviert*
129	wie 1, jedoch mit Abschaltung des Ethernet
130	wie 2, jedoch mit Abschaltung des Ethernet
3 - 255	*reserviert*

Wird Ethernet nicht abgeschaltet (1,2), so ist keine genaue Synchronisation möglich!

2005h – Reaktion auf PLC Status

Gibt an wie der CANopen-Master bei Zustandsänderungen der PLC reagiert.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2005h	0h	Reaction on PLC status	Unsigned8	rw	0	Nein

Die einzelnen Werte haben folgende Bedeutung:

Wert	Bedeutung
0	Es erfolgt keine Reaktion sobald die PLC den Betriebszustand „Betrieb“ (RUN) verlässt.
1	Das Optionsmodul CANopen-Master geht in den Zustand „FATAL ERROR“ sobald die PLC nicht im Betriebszustand „Betrieb“ (RUN) ist. das Netzwerk wird in den Zustand „STOPPED“ gesetzt.
2 - 255	*reserviert*

Die eingestellten Werte werden erst nach einem Aus-/Einschalten des Gerätes wirksam. Ein Abspeichern der Hersteller spezifischen Objekte über das Objekt 1010h muss bei einer Änderung durchgeführt werden.

2020h - Netzwerk Bootup Einstellung

Mit diesem Objekt kann der Boot-Mechanismus des Netzwerks beeinflusst werden.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2020h	0h	Network Bootup Adjust	Unsigned32	rw	00000000h	Nein

Die einzelnen Bits des 32 Bit-Wertes haben folgende Bedeutung:

Bit	Wert / Bedeutung
0	0: Beim Bootup des Netzwerks wird auf Vorhandensein aller möglichen Knoten 1 bis 127 gescannt. 1: Beim Bootup des Netzwerks wird nur auf Vorhandensein von Knoten gescannt, welche im Objekt 1F81h konfiguriert sind. Diese Einstellung wird nicht empfohlen, da nicht konfigurierte Knoten zu Störungen im Betrieb führen können!
1 – 31 (MSB)	*reserviert*

2022h - Wartezeit SDO während des Boot-Vorgangs eines Knotens

Länge der Wartezeit in ms des CANopen-Master auf SDO-Antwort während des Boot-Vorgangs eines Knotens.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2022h	0h	Node Boot Wait SDO	Unsigned16	rw	50	Nein

Während des Boot-Vorgangs eines Knoten sind wiederholte SDO-Zugriffe auf diesen notwendig. Es werden auch alle möglichen Knotennummern 1 - 127 abgescannt, indem Objekt 1000h gelesen wird. Kommt es zu einem Timeout in der Antwort auf diesen Zugriff, wird der Knoten als nicht vorhanden angesehen. Auch dieser Timeout wird durch das Objekt 2022h bestimmt.

Sind nur b maXX 4400 Slaves im Netzwerk enthalten, kann eine Zeit von 10 ms eingestellt werden.

2023h - Wartezeit zur Wiederherstellung der Default-Einstellungen

Länge der Wartezeit in ms des CANopen-Master auf Bootup-Nachricht nach einem Knoten-Reset vor dem Download des DCF.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2023h	0h	Node Wait Reset	Unsigned16	rw	5000	Nein

Bevor der CANopen-Master das DCF auf einen Knoten lädt, muss dieser zunächst auf Kommando seine Default-Werte laden. Danach ist ein Reset des Knoten erforderlich. Da

die Knoten einige Zeit brauchen bis diese den Reset ausführen, muss der CANopen-Master einige Zeit warten, bis dieser mit dem DCF-Download beginnen kann. Diese Wartezeit wird über dieses Objekt eingestellt.

Sind nur b maXX 4400 Slaves im Netzwerk enthalten, kann eine Zeit von 50 ms eingestellt werden.

2024h - Wartezeit nach Netzwerk Reset bevor der Scan des Netzwerks beginnt

Länge der Wartezeit in ms des CANopen-Master bis dieser nach einem Netzwerk-Reset mit dem Scannen des Netzwerks beginnt.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2024h	0h	Network Wait Start Scan	Unsigned16	rw	5000	Nein

Bevor das Netzwerk gescannt wird, wird ein Reset des Netzwerks (Reset Communication) ausgeführt. Da die Knoten einige Zeit brauchen bis diese den Reset ausführen, muss der CANopen-Master einige Zeit warten, bis dieser mit dem Scannen beginnen kann. Diese Wartezeit wird über dieses Objekt eingestellt.

Sind nur b maXX 4400 Slaves im Netzwerk enthalten, kann eine Zeit von 50 ms eingestellt werden.

2025h - Wartezeit bis der nächste Scan eines Knoten beginnt

Wenn der CANopen-Master einen Knoten vermisst, prüft er zyklisch, ob sich das Modul im Netzwerk befindet.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2025h	0h	Node Wait Next Scan	Unsigned16	rw	2000	Nein

2100h - Konfiguration Prozessabbild

Informiert über die Konfiguration des Prozessabbildes

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2100h	-	PI General Configuration	Array	-	-	-
	0	Nr. of entries supported	Unsigned8	ro	4	Nein
	1	Mode	Unsigned8	ro	1	Nein
	2	Version	Unsigned16	ro	100	nein
	3	Size Outputs	Unsigned32	ro	2048	Nein
	4	Size Inputs	Unsigned32	ro	2048	Nein

Subindex 1 gibt den Modus des Prozessabbildes an. Der Wert 1 bedeutet, dass das segmentierte Verfahren zur Abbildung der Netzwerkvariablen verwendet wird. Das segmentierte Verfahren benötigt eine Konfiguration der abzubildenden Objekte (siehe Objekt

2101h). **Subindex 2** gibt die Versionsnummer des verwendeten Verfahrens wieder. In **Subindex 3** steht die Größe in Byte des Prozessabbilds für Ausgänge, also Daten welche von der Applikation an das Netzwerk gesendet werden (Netzwerkobjekte A000h - A279h). In **Subindex 4** steht die Größe in Byte des Prozessabbilds für Eingänge, also Daten welche vom Netzwerk an die Applikation gesendet werden (Netzwerkobjekte A480h - A6FFh).

2101h - Asynchrone und synchrone Bereiche im Prozessabbild

Legt die asynchronen und synchronen Bereiche im Prozessabbild fest

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2101h	-	PI Asynchron and Synchron Area	Array	-	-	-
	0	Nr. of entries supported	Unsigned8	ro	4	Nein
	1	Startaddress and size asynchron PI outputs	Unsigned32	rw	0000 0100h	Nein
	2	Startaddress and size asynchron PI inputs	Unsigned32	rw	0000 0100h	nein
	3	Startaddress and size synchron PI outputs	Unsigned32	rw	0400 0100h	Nein
	4	Startaddress and size synchron PI intputs	Unsigned32	rw	0400 0100h	Nein

In den Subindizes 1 - 4 werden die asynchronen und synchronen Bereiche des Prozessabbildes festgelegt, jeweils für Ausgänge und Eingänge. In diesen Bereichen liegen Objekte, welche in asynchronen oder synchronen PDOs übertragen werden. Die Objekte des Prozessabbildes selbst werden über die Objekte 2102h (asynchrone Objekte) und 2103h (synchrone Objekte) festgelegt.

Der 32 Bit-Wert jedes Subindizes ist in zwei 16 Bit Werte unterteilt:

Byte:	MSB	LSB
Inhalt	Startadresse (Byteadresse) des Bereichs (16 Bit)	Größe des Bereichs in Byte (16 Bit)

Die Startadresse wird relativ zum Start der Ausgänge bzw. Eingänge des Prozessabbildes angegeben. Es werden Byte-Adressen angegeben. Die Größe eines Bereichs wird auch in Byte angegeben. Die notwendige Größe eines Bereichs ergibt sich aus den tatsächlich angelegten Objekten im Prozessabbild (Objekte 2102h und 2103h), es sind jedoch immer Vielfache von 256 Byte zu vergeben.

Es ist zu folgendes zu beachten:

- Der asynchrone Bereich muss vor dem synchronen Bereich liegen
- Der asynchrone Bereich muss bei Adresse 0 beginnen. Gibt es keinen asynchronen Bereich, kann auch der synchrone Bereich bei Adresse 0 beginnen.

- Die Bereiche dürfen sich nicht überlappen
- Zulässige Größe eines Bereichs sind Vielfache von 256 Byte
- Die Größe des Prozessabbildes darf auf beiden Seiten (Ausgänge und Eingänge) nicht überschritten werden (siehe hierzu Objekt 2101h, Subindex 3 und Subindex 4)

2102h - Konfigurierte asynchrone Objekte im Prozessabbild

Ermöglicht die Festlegung asynchroner Objekte, die im Prozessabbild abgebildet werden.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2102h	-	PI asynchronous Object Configuration	Array	-	Optional	-
	0	Nr. of entries supported	Unsigned8	ro	64	Nein
	1	1. configured object	Unsigned48	rw	A000 0000 0008h	Nein
	2	2. configured object	Unsigned48	rw	A040 0008 0018h	Nein
	3	3. configured object	Unsigned48	rw	A0C0 0020 0010h	Nein
	4	4. configured object	Unsigned48	rw	A100 0040 0010h	Nein
	5	5. configured object	Unsigned48	rw	A1C0 0060 0010h	Nein
	6	6. configured object	Unsigned48	rw	A200 00A0 0010h	Nein
	7	7. configured object	Unsigned48	rw	A480 0000 0008h	Nein
	8	8. configured object	Unsigned48	rw	A4C0 0008 0018h	Nein
	9	9. configured object	Unsigned48	rw	A540 0020 0010h	Nein
	10	10. configured object	Unsigned48	rw	A580 0040 0010h	Nein
	11	11. configured object	Unsigned48	rw	A640 0060 0010h	Nein
	12	12. configured object	Unsigned48	rw	A680 00A0 0010h	Nein
	13	13. configured object	Unsigned48	rw	0	Nein
	:	:	:	:	:	:
	64	64. configured object	Unsigned48	rw	0	Nein

Subindex 0 enthält die maximal mögliche Anzahl konfigurierbarer Objekte. Die folgenden **Subindizes 1 - 64** enthalten die Konfiguration der Objekte, welche im Prozessabbild abgebildet werden. Die Objekte werden im asynchronen Bereich des Prozessabbildes abgebildet, d. h. diese Objekte werden für asynchrone PDOs verwendet.

Der 48 Bit-Wert ist in drei 16 Bit Werte unterteilt:

Byte: MSB LSB

Objektindex	Startadresse im DPRAM	Anzahl verwendeter Subindizes (max. 254)

Der Objektindex gibt den Index des konfigurierten Objektes an. Für den Objektindex sind folgende Bereiche, abhängig vom Datentyp, zulässig:

- Asynchrone Objekte, welche von der Applikation geschrieben werden:

Bereich	CANopen Datentyp
A000h – A01Fh	Integer8
A040h – A05Fh	Unsigned8
A0C0h – A0DFh	Integer16
A100h – A11Fh	Unsigned16
A1C0h – A1DFh	Integer32
A200h – A21Fh	Unsigned32

- Asynchrone Objekte, welche von der Applikation gelesen werden:

Bereich	CANopen Datentyp
A480h – A49Fh	Integer8
A4C0h – A4DFh	Unsigned8
A540h – A55Fh	Integer16
A580h – A59Fh	Unsigned16
A640h – A65Fh	Integer32
A680h – A69Fh	Unsigned32

Die Startadresse im DPRAM wird relativ als Byteadresse zum Anfang der Prozessabbild Ausgänge (A000h usw.) bzw. Eingänge (A480h) angegeben. Bei der Anzahl verwendeter Subindizes wird angegeben, bis zu welchem Subindex Objekte des angegebenen Objektindex verwendet werden.

HINWEIS

Es ist zu unbedingt zu beachten, dass der Adressbereich der asynchronen Objekte nicht mit dem Adressbereich der synchronen Objekte (Objekt 2103h) überlappt. Um dies zu vermeiden, sollten die asynchronen Objekte vor den synchronen Objekten angelegt werden.

2103h - Konfigurierte synchrone Objekte im Prozessabbild

Ermöglicht die Festlegung synchroner Objekte, die im Prozessabbild abgebildet werden.

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
2103h	-	PI synchronous Object Configuration	Array	-	Optional	-
	0	Nr. of entries supported	Unsigned8	ro	64	Nein
	1	1. configured object	Unsigned48	rw	A020 0400 0008h	Nein
	2	2. configured object	Unsigned48	rw	A060 0408 0018h	Nein

Index	Subindex	Name	Datentyp	Zugriff	Defaultwert	Mappbar
	3	3. configured object	Unsigned48	rw	A0E0 0420 0010h	Nein
	4	4. configured object	Unsigned48	rw	A120 0440 0010h	Nein
	5	5. configured object	Unsigned48	rw	A1E0 0460 0010h	Nein
	6	6. configured object	Unsigned48	rw	A220 04A0 0010h	Nein
	7	7. configured object	Unsigned48	rw	A4A0 0400 0008h	Nein
	8	8. configured object	Unsigned48	rw	A4E0 0408 0018h	Nein
	9	9. configured object	Unsigned48	rw	A560 0420 0010h	Nein
	10	10. configured object	Unsigned48	rw	A5A0 0440 0010h	Nein
	11	11. configured object	Unsigned48	rw	A660 0460 0010h	Nein
	12	12. configured object	Unsigned48	rw	A6A0 04A0 0010h	Nein
	13	13. configured object	Unsigned48	rw	0	Nein
	:	:	:	:	:	:
	64	64. configured object	Unsigned48	rw	0	Nein

Subindex 0 enthält die maximal mögliche Anzahl konfigurierbarer Objekte. Die folgenden **Subindizes 1 - 64** enthalten die Konfiguration der Objekte, welche im Prozessabbild abgebildet werden. Die Objekte werden im synchronen Bereich des Prozessabbildes abgebildet, d. h. diese Objekte werden für synchrone PDOs verwendet.

Der 48 Bit-Wert ist in drei 16 Bit Werte unterteilt:

Byte: MSB LSB

Objektindex	Startadresse im DPRAM	Anzahl verwendeter Subindizes (max. 254)

Der Objektindex gibt den Index des konfigurierten Objektes an. Für den Objektindex sind folgende Bereiche, abhängig vom Datentyp, zulässig:

- Synchrone Objekte, welche von der Applikation geschrieben werden:

Bereich	CANopen Datentyp
A020h – A03Fh	Integer8
A060h – A07Fh	Unsigned8
A0E0h – A0FFh	Integer16
A120h – A13Fh	Unsigned16
A1E0h – A1FFh	Integer32
A220h – A23Fh	Unsigned32

- Synchrone Objekte, welche von der Applikation gelesen werden:

Bereich	CANopen Datentyp
A4A0h – A4BFh	Integer8
A4E0h – A4FFh	Unsigned8
A560h – A57Fh	Integer16
A5A0h – A5BFh	Unsigned16
A660h – A67Fh	Integer32
A6A0h – A6BFh	Unsigned32

Die Startadresse im DPRAM wird relativ als Byteadresse zum Anfang der asynchronen Prozessabbild Ausgänge (A000h usw.) bzw. Eingänge (A480h) angegeben. Bei der Anzahl verwendeter Subindizes wird angegeben, bis zu welchem Subindex Objekte des angegebenen Objektindex verwendet werden.

HINWEIS

Es ist zu unbedingt zu beachten, dass der Adressbereich der asynchronen Objekte (Objekt 2101h) nicht mit dem Adressbereich der synchronen Objekte überlappt. Die synchronen Objekte sollten immer nach den asynchronen Objekten angelegt werden. Bei der Vergabe der Anfangsadresse des ersten synchronen Objektes sollte vor der Adresse noch genügend Platz für eventuell zusätzlich notwendige asynchrone Objekte sein.

Revisionsübersicht

Version	Stand	Änderungen
5.03002.02	10.01.2007	Erweiterung des Applikationahandbuches um ProMaster und ProProg wt III neu: Kapitel 4.3 ProMaster, ProCANopen, ProProg wt III und Motion Control Kapitel 4.3 (alt) jetzt Kapitel 4.4 (neu)

Index

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**