

Versuch:

# CANopen

## 1. Grundlagen

Das in der Automatisierungstechnik weit verbreitete Kommunikationsprofil CANopen basiert auf dem Bussystem Controller Area Network (CAN). Anwendung findet es vor allem bei dezentraler Peripherie, die bei begrenzter räumlicher Ausdehnung eine erhöhte Sicherheitsstufe erfordert.

### 1.1. Inhalt und Versuchsziel

Im Fokus dieses Versuchs stehen die Besonderheiten des Bussystems *Controller Area Network* (CAN) in Verbindung mit der Anwendungsschicht *CANopen*. Besonderes Augenmerk liegt dabei auf den Kommunikationsgrundlagen und den Objektverzeichnissen der Busteilnehmer. Ihnen stehen verschiedene CANopen-Kommunikationskanäle (z.B. SDO-Zugriff, PDO-Linking) zur Verfügung. Einzelne Datentelegramme werden Sie von „Hand“ zusammenstellen und versenden. Dies geschieht mit einem mit dem CAN-Bus-Interface verbundenen PC.

Zur Konfiguration, Inbetriebnahme und Analyse des Versuchsaufbaus steht Ihnen im Labor die Software **CANoe** der Fa. Vector Informatik GmbH zur Verfügung.

Durch Verwendung der Netzwerkmanagementfunktionen werden die angeschlossenen Teilnehmer in die durch das Bootup definierten Zustände versetzt. Nach Inbetriebnahme des Versuchsaufbaus wird das **CANoe**-Projekt an die Systemanforderungen angepasst. Die Prozessdaten sollen damit zur effektiven Bewertung übersichtlich dargestellt werden.

Ein von einer CAN-Busklemme (I/O-Modul der Fa. WAGO) erzeugtes Prozessdaten-Telegramm soll ausgewertet und zur Bewegungssteuerung eines Roboters genutzt werden.

Nach grundlegender Analyse der Funktionsweise des PDO-Linking und des PDO-Mapping werden Sie selbst Prozessdaten-Telegramme zusammenstellen und versenden, die zum einen binäre Ausgänge des I/O-Moduls ansteuern und zum anderen die verschiedenen Roboterachsen hin zu Soll-Positionen bewegen sollen.

## Inhalt

1.	Grundlagen	1
1.1.	Versuchsablauf und -ziel	1
1.2.	Begriffe	3
2.	Literaturempfehlungen	4
3.	Versuchsaufbau	5
4.	Versuchsvorbereitung	6
5.	Versuchsdurchführung	8
5.1.	Grundeinstellungen	8
5.2.	Scannen des CAN-Netzwerkes	9
5.3.	Aufbau von SDO-Telegrammen	11
5.4.	PDO-Linking mittels SDO-Zugriff	12
5.5.	PDO-Mapping	15
5.6.	Der PDO-Zugriff	17
5.7.	CANoe als Steuerungstool	18
5.8.	CANoe als Diagnose- und Visualisierungstool	20

## 1.2. Begriffe

Sämtliche Anwenderdaten eines CANopen-Knotens sind in seinem **Objektverzeichnis** (OV) mit fest vorgegebener Datenstruktur abgelegt. Die einzelnen Einträge sind nach **Index** (vierstellige Hex-Zahl) und **Subindex** (zweistellige Hexzahl) geordnet. Die Indexstruktur ist in **User Profilen** (CANopen-Standards, z.B. DSP 402 für elektrische Antriebe) festgelegt und kann nicht geändert werden.

Jedes Kommunikations-Objekt (SDO, PDO oder NMT) besitzt einen eigenen Identifier (**COB-ID**). Dieser wird zunächst entsprechend einer vorgegebenen Default-Identifier-Zuordnung vergeben, kann aber zur Festlegung von Kommunikationskanälen mittels SDO-Zugriff geändert werden (siehe PDO-Linking). COB-ID's dienen außerdem zur Priorisierung der Telegramme; je kleiner die COB-ID, desto höher ist die Priorität des Telegramms. Die COB-ID 0 ist für wichtige Netzwerkdienste (z.B. Alarmer) reserviert.

**Service-Daten-Objekte (SDO)** dienen zur Ausführung von Service-Diensten, wie z.B. dem Ändern einer COB-ID. Man unterscheidet Server-SDO's (S-SDO) und Client-SDO's (C-SDO). Der Nutzdatenanteil eines SDO-Telegrammes ist stets acht Byte lang und enthält neben einem Kommandobyte (Byte 0) Index und Subindex des anzusprechenden OV-Eintrages (3 Byte) und einen Datenteil (4 Byte).

**Prozessdaten-Objekte (PDO)** dienen zum Übertragen von Real-Time-Prozessdaten (z.B. Motorpositionen). Unterschieden werden hierbei Transmit-PDO's (TPDO) und Receive-PDO's (RPDO). Im Nutzdatenanteil eines PDO-Telegrammes (max. 8 Byte lang) werden ausschließlich Prozessdaten übertragen. PDO's befinden sich im Objektverzeichnis im Bereich der Indizes 1400..1FFF.

Prozessdaten werden über PDO-Kanäle übertragen. Ein solcher Kanal zwischen einem TPDO und mindestens einem RPDO wird mittels **PDO-Linking** festgelegt. Dabei erhalten alle zu einem Prozessdaten-Kanal gehörenden PDO's dieselbe COB-ID.

Welche Prozessdaten in welcher Reihenfolge mithilfe eines PDO-Telegrammes gesendet bzw. empfangen werden sollen, ist durch das **PDO-Mapping** festzulegen. Die Mapping-Objekte eines PDO's enthalten die Anzahl, die Indizes, Subindizes und jeweilige Datenlänge (in Bits) der gemappten (zugeordneten) Objekte. Unter Mapping versteht man somit die Zuordnung von Prozessdaten zum Nutzdatensegment eines PDO-Telegramms.

## 2. Literaturempfehlungen

- Etschberger, K.: Controller Area Network;  
München; Hanser
- Schwarz, A.; Zeltwanger, H.:  
Controller Area Network; CAN in Automation (CiA); in Praxis Profiline;  
Würzburg ; Vogel Verl.
- Lawrenz, W.: CAN Controller Area Network;  
Heidelberg; Hüthig
- Kriesel, W. u.a.: Bustechnologie für die Automation;  
Heidelberg; Hüthig
- Reißweber, B.: Feldbusse;  
München, Wien; Oldenbourg
- Schnell, G.: Bussysteme in der Automatisierungstechnik;  
Braunschweig, Wiesbaden; Vieweg

### 3. Versuchsaufbau

Zum Versuchsplatz gehören ein Kleinstroboter **Teach Robot**, eine **WAGO®-CANopen-Baugruppe 750-307** (CiA-Standard DS 401) und ein PC mit angeschlossenem **USB-to-CAN-Interface VN1610**. Diese Geräte sind über ein Buskabel miteinander in einem CAN-Netzwerk verbunden. Die Baugruppe 750-307 verfügt außerdem über Digital-Input- und Digital-Output-Baugruppen (I/O-Modul), an denen sieben Schalter und acht Meldelampen angeschlossen sind. Der Roboter kann über sechs Achsen bewegt werden, seine Kopplung an den CAN-Bus erfolgt über ein TR-to-CAN-Gateway (CiA DS 402).

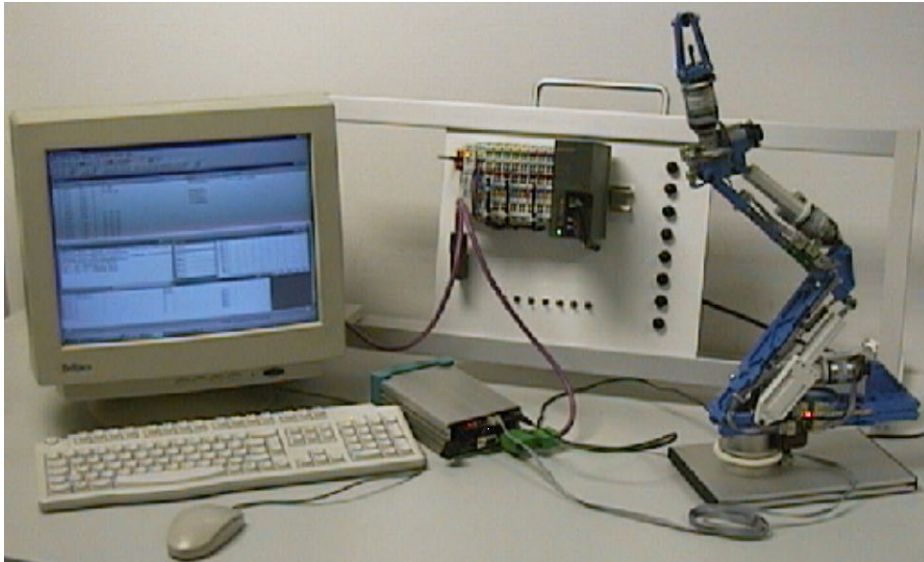


Abbildung 3-1: Versuchsaufbau am CANopen-Praktikumplatz

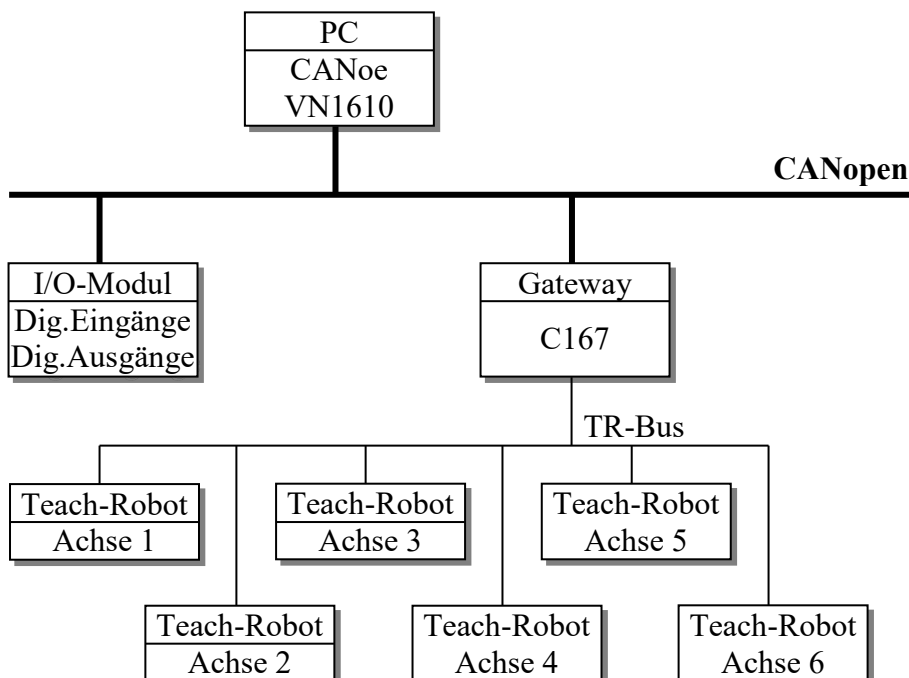


Abbildung 3-2: Schema des Versuchsaufbaus

Tabelle 3-1: Achsen des Roboters:

Achse	Beschreibung	Maximalbewegung (Inkrement)
1	Auf-/Zu-Bewegung des Greifers	70
2	Drehbewegung des Greifers	130
3	Aufwärts-/Abwärts-Bewegung des Greifers	420
4	Aufwärts-/Abwärts-Bewegung des Armes	420
5	Aufwärts-/Abwärts-Bewegung der Schulter	350
6	Drehbewegung des Roboters	700

Folgende Abbildung zeigt alle im Versuch zu erstellenden Producer-Consumer-Beziehungen. Die Zahlen in den eckigen Klammern geben den jeweiligen Abschnitt der Versuchsanleitung, in dem die Kommunikationsbeziehung genutzt wird, an.

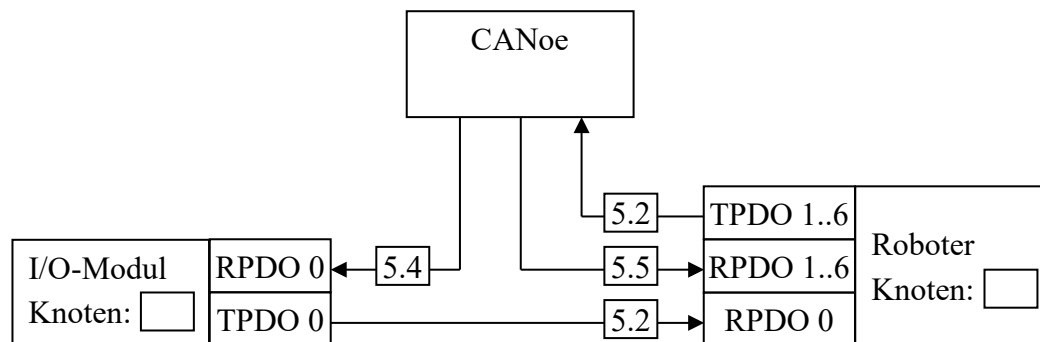


Abbildung 3-3: Producer-Consumer-Beziehungen

## 4. Versuchsvorbereitung

**Wichtiger Hinweis:** Bringen Sie zum Versuch das Ihnen von Ihrem Dozenten zum Fach **Feldbussysteme** ausgehändigte Vorlesungsskript mit! Es enthält Informationen, auf die Sie im Laufe des Versuchs zurückgreifen müssen.

Beantworten Sie mit Hilfe der oben angegebenen oder anderweitigen Literatur folgende Fragen **handschriftlich**:

- 1) Woher stammt der CAN-Bus; wofür wurde er ursprünglich entwickelt?
- 2) Welche Vorteile bzw. Besonderheiten gegenüber anderen Feldbussystemen bietet CAN?
- 3) Worin liegt die wohl wichtigste Einschränkung dieses Bussystems? Ursache für die Einschränkung?
- 4) In welcher Topologie arbeitet der Bus?
- 5) Welcher Buspegel (Übertragungsstandard) wird verwendet?
- 6) Welche maximale Übertragungsrate ermöglicht der CAN-Bus?
- 7) Welches Buszugriffsverfahren kommt zur Anwendung und wie funktioniert es?
- 8) Welche Mechanismen zur Fehlererkennung werden angewendet?
- 9) Was ist die Hamming-Distanz (HD); was sagt sie aus und welche HD hat der CAN-Bus im Vergleich zu anderen Feldbussen?

- 10) Aus welchen Bestandteilen setzt sich das Standard-CAN-Telegramm zusammen?
- 11) Was versteht man unter CANopen und wozu wird es verwendet?
- 12) Wie sieht das ISO/OSI-Sieben-Schichten-Modell bei CANopen aus?
- 13) Welche Zustände kann ein CAN-Knoten beim Minimum Bootup annehmen?
- 14) Was ist eine COB-ID; wozu braucht man sie?
- 15) Wie setzt sich die Standard-COB-ID bei CANopen zusammen? Wie groß ist die COB-ID im CAN-Rahmen? (*Achtung: Wikipedia kann auch irren*)
- 16) Wie viele Nutzdatenbytes können mit einem Telegramm übertragen werden? (im CAN ? / im CANopen ?)
- 17) Welche vordefinierten Verbindungskanäle (Peer-To-Peer-Objects) sind mittels COB-ID realisierbar?
- 18) Aus welchen Bestandteilen setzt sich das Datenfeld bei einem CANopen-SDO-Zugriff zusammen?
- 19) Was versteht man unter PDO-Linking; wozu dient es und wie funktioniert es?
- 20) Was versteht man unter PDO-Mapping und wie funktioniert es?
- 21) **Komplexe Aufgabe:** **Kämpfen Sie sich unbedingt durch diese Aufgabe, damit Sie den Versuch in der zur Verfügung stehenden Zeit erfolgreich abschließen können!**

Der Inhalt des 1. Analog-Inputs (32-Bit) eines modularen IO-Knotens (Node-ID 10) soll an einem modularen IO-Knoten (Node-ID 11) am 1. Analog-Output (32-Bit) ausgegeben werden. Nutzen Sie zur Übertragung das TPDO2 bzw. RPDO2. (Benötigte Hilfsmittel: DS401.pdf und CANopenGrafik.pdf)

- Tragen Sie die erforderlichen Änderungen der jeweiligen Objektverzeichnisse (für Linking **und** Mapping) in die folgende Tabelle ein!

**Tabelle 4-1: Erforderliche Änderungen der Objektverzeichnisse**

Knoten 10			Knoten 11		
Index	Subindex	Inhalt	Index	Subindex	Inhalt

- Wie werden die Objektverzeichnisse manipuliert?

## 5. Versuchsdurchführung

Der PC am Arbeitsplatz wurde bereits gestartet, das Programm **CANoe** ist geöffnet. **CANoe** übernimmt in der Master-Slave-Beziehung zur Konfiguration des CANopen-Netzwerks die Aufgaben des Netzwerkmanagement-Masters. In der Producer-Consumer-Beziehung (Prozessdaten-Austausch) fungiert **CANoe** als Steuerungs- und Visualisierungstool für den Roboter.

### 5.1. Grundeinstellungen

- Öffnen Sie eine neue Projektdatei (*Menü: Datei → Neu*). Wählen Sie die Projektvorlage **Default** im Bereich *General* (Doppelklick)!

📖 Im neu angelegten Projekt sind zunächst die beiden Fenster **Trace** und **Write** geöffnet. Im **Write**-Fenster werden wichtige Systemmeldungen angezeigt. Das **Trace**-Fenster dient der Protokollierung der Busaktivitäten während einer Messung. Alle Botschaften, die am Eingang des Trace-Fensters im Messaufbau ankommen, werden als Textzeilen im **Trace**-Fenster dargestellt. Im Folgenden werden Sie das **Trace**-Fenster den Anforderungen dieses Laborversuchs entsprechend anpassen.

- Öffnen Sie das Konfigurationsmenü des **Trace**-Fensters (Rechtsklick im Fenster **Trace** → **Konfiguration...**)! Stellen Sie im Register *Allgemein* folgende Parameter ein:
  - *Anzeigemodus*.....**feststehender Modus**
  - *Zeit*.....**Relative Zeit** anzeigen
- Verhindern Sie das *Ausblenden von veralteten und unveränderten Daten*, indem Sie beide dafür zur Verfügung stehenden Optionen **deaktivieren**!
- Wechseln Sie ins Register *Spalten*, wählen Sie im Auswahlfeld *Verfügbare Felder* den Eintrag **CANopen** und stellen Sie die Anordnung der Spalten im **Trace**-Fenster in folgender Reihenfolge ein:
  - *Time*
  - *ID*
  - *DLC*
  - *Data*
  - *Send Node*
  - *Name*
  - *Interpretation*
- Belassen Sie alle anderen Einstellungen des Fensters auf ihren voreingestellten Werten. Beenden Sie den Dialog *Trace- Konfiguration* (Button **OK**)! Verbreitern Sie die Spalte *Interpretation* so weit, dass sie mindestens ein Drittel der Bildschirmbreite einnimmt!



- Konfigurieren Sie nun die CAN-Schnittstelle CAN1 (Menü: *Hardware* → *Bus-Hardware*): CAN1 → Setup: Baudrate [kBaud] = **1000.0**

Das CAN-Interface verfügt noch über einen zweiten CAN-Kanal zum Anschluss eines weiteren Netzwerks. Für den Laborversuch wird jedoch nur der Kanal CAN1 verwendet.

- Belassen Sie alle weiteren Parameter auf ihren voreingestellten Werten! Bestätigen Sie Einstellungen (*OK*)!

📖 Die Sollpositionen (Achsstellungen) für den Roboter sollen später von **CANoe** (als Producer) auf den CAN-Bus gegeben und die Istpositionen des Roboters von **CANoe** (als Consumer) empfangen und bestimmte Messwerte in Klartextform angezeigt werden.

Um dies zu realisieren ist es notwendig, das Electronic Data Sheet (EDS) der an den CAN-Bus angeschlossenen Geräte mittels Netzwerksan einzulesen, in die **CANoe**-Datenbank (CANdb) zu importieren und alle mit den PDO's verknüpften Objekte sowie die entsprechenden Botschaften anzulegen. Da diese Prozedur ziemlich aufwändig ist, wurde bereits eine **CANoe**-Datenbank (CANdb) angelegt. In ihr sind alle Objekte, die für diesen Versuch erforderlich sind, enthalten. Bevor Sie den im folgenden Abschnitt 5.2 beschriebenen Netzwerksan durchführen, werden Sie nun diese Datenbasis (CANdb) zum Projekt hinzufügen.

- Öffnen Sie das Fenster **Simulationsaufbau** (Menü *Simulation* → *Simulationsaufbau*)! Im rechten Teil dieses Fensters finden Sie die *Netzwerke*-Übersicht. Öffnen Sie dort den Ordner *Netzwerke* → *CAN Netzwerke* → *CAN* und wählen Sie aus dem Kontext-Menü des Objektes **Datenbasen** den Befehl **Hinzufügen...**! Selektieren Sie die Datenbasis **PRJDB** aus dem Verzeichnis *E: → CANoe-Proj → CANdb → database* und betätigen Sie den Button **Öffnen**!
- Schließen Sie das Fenster **Simulationsaufbau**!

## 5.2. Scannen des CAN-Netzwerkes

📖 Im Folgenden soll das am CAN-Interface angeschlossene Netzwerk eingelesen werden. Gleichzeitig mit diesem Netzwerksan versucht **ProCANopen** allen gefundenen Geräten die passenden EDS-Dateien zuzuweisen und überprüft deren Einträge. Die dabei über das Netzwerk versendeten Telegramme werden im **Trace**-Fenster aufgezeichnet und stehen danach für eine Analyse zur Verfügung.

- Öffnen Sie das Werkzeug **ProCANopen** (Menü *Werkzeuge* → *ProCANopen*)! Stellen Sie auch hier die erforderliche Baudrate (1000) ein (Menü *Optionen* → *Einstellungen...*)!
- Wechseln Sie zum Programm **CANoe** und starten Sie die Telegramm-Aufzeichnung im Trace-Fenster (Menü *Home* → *Start*)!

**Bitte den folgenden Abschnitt erst komplett durchlesen und anschließend bearbeiten!**

- Wechseln Sie zu **ProCANopen** und wählen Sie dort aus dem Menü *Netzwerk* den Befehl **Netz einlesen**! Aktivieren Sie im folgenden Dialog die Option **Automatische EDS-Auswahl bei neuen Knoten** und starten Sie den Netzwerksan (Button **Start**); wechseln Sie **sofort danach** zum **Trace**-Fenster und beobachten Sie die **sich ändernden** Telegrammzeilen!

- Nach Abschluss des Scanvorgangs (keine Änderungen mehr im *Trace*-Fenster) wechseln Sie zurück zu **ProCANopen**!

📖 Der Netzwerkskan endet u.U. mit einer Fehlermeldung im Report-Fenster. Ursache dafür ist ein Anpassungsproblem zwischen Sendedaten und Empfangsdaten und rührt daher, dass einem der beiden Knoten noch keine EDS-Datei zugewiesen werden konnte. Dieser Fehler ist für die Fortführung des Versuchs jedoch ohne Belang.

- Ignorieren Sie die Fehlermeldung (*OK*) und schließen Sie das Fenster *Scannen*!

Im *Ausgabe*-Fenster (unterer Teil des Bildschirms) sehen Sie eine Zusammenfassung des Scanvorgangs.

- Übernehmen Sie aus diesem Fenster für die gefundenen Knoten deren Node-ID's, die zugehörigen Gerätenamen und CANopen-Geräteprofile (User Profile) in folgende Tabelle!



*Tabelle 5-1: Geräteinformationen der angeschlossenen CAN-Busteilnehmer*

Node-ID	Gerätename (manufacturers device name)	CANopen-Geräteprofil

📖 **ProCANopen** zeigt jetzt den kompletten Busaufbau mit allen angeschlossenen Geräten (inkl. deren Node-IDs) - außer Knoten 1 (CANoe) und 127 (ProCANopen). Während des Netzwerkskans versuchte ProCANopen jedes der gefundenen Geräte automatisch einer EDS-Datei zuzuordnen. War diese Zuordnung erfolgreich, erscheint in den Blöcken der Geräte das zur EDS gehörende Bildsymbol.

- Falls die automatische EDS-Zuordnung nicht für beide Geräte erfolgreich war, ordnen Sie die entsprechenden EDS-Dateien wie folgt zu: Wählen Sie aus dem Kontextmenü des noch nicht zugeordneten Geräteblocks den Befehl *Konfiguration*! Betätigen Sie den Button *EDS ändern*! Suchen Sie die zum Gerät passende EDS-Datei (siehe folgende Tabelle) und markieren Sie diese!

*Tabelle 5-2: Zuordnung der EDS-Dateien zu den Geräten*

Node-ID	EDS-Datei	Bitmap
2 bzw. 3	Robot.eds	
6 bzw. 8	W839608A.eds	

Bestätigen Sie die Auswahl (*OK*)! Schließen Sie das Fenster *Knotenkonfiguration* (*OK*)!

- Wechseln Sie nun zum Programm **CANoe** und stoppen Sie die Aufzeichnung (Menü *Start* → *Stoppen*)!

Im *Trace*-Fenster wurden bisher fast ausschließlich SDO-Telegramme aufgezeichnet!

- Analysieren Sie die aufgezeichneten SDO-Telegramme (Das Scrollen im *Trace*-Fenster ist bei gestoppter Messung möglich)!
- Was bedeuten die folgenden im **Trace**-Fenster verwendeten Abkürzungen und Begriffe?

- ID (=COB-ID) ? \_\_\_\_\_
- DLC ? \_\_\_\_\_
- Data ? \_\_\_\_\_
- Send Node ? \_\_\_\_\_
- Rq (=Request)? \_\_\_\_\_
- Rsp (=Response)? \_\_\_\_\_

📖 Ein *Send Node* wird nur dann angegeben, wenn das Telegramm von einem der beiden gefundenen Knoten gesendet wurde. Alle anderen Telegramme stammen von *CANoe*.

Die Identifier der Botschaften (Spalte ID = COB-ID) wurden entsprechend einer definierten Default-Identifier-Zuordnung vergeben.

- Ermitteln Sie mithilfe der Default-Identifier-Verteilung (siehe Ihr Vorlesungsskript) für je ein Request- und ein Response-Telegramm **aus deren COB-ID** die Funktion des Telegramms und die Knotennummer (Node-ID) des Senders/Empfängers! Komplettieren Sie die Tabelle, indem Sie die Nutzdaten der beiden Telegramme byteweise in die Spalte *Data* eintragen!

Tabelle 5-3: SDO-Request- und -Response-Telegramm

COB-ID	Funktion	Node-ID	Data							

### 5.3. Aufbau von SDO-Telegrammen

Im Trace-Fenster wurden die während des Netzwerkskans versendeten Telegramme aufgezeichnet. Bedingt durch die gewählte Darstellungsform (feststehend) wird jedoch immer nur das jeweils **letzte von einem Objekt versendete bzw. empfangene** Telegramm angezeigt. Analysieren Sie die angezeigten Telegramme:

- In welchen Bytes des Nutzdatensegments (Spalte *Data*) finden sich die in eckigen Klammern (Spalte *Interpretation*) angegebenen Zahlen wieder und wie sind diese Bytes dort angeordnet?

---



---

- Was bedeuten die in eckigen Klammern angegebenen Zahlen (Spalte *Interpretation*)?

---

- Welche Information enthält das Byte 0 des Nutzdatensegments eines SDO-Telegramms?

---

- Warum sind die meisten angezeigten Telegramme *Abort*-Telegramme?

---

📖 Beide Knoten wurden nach dem Netzwerkskan einer EDS-Datei zugeordnet. Den Inhalt der EDS-Datei kann man sich in *ProCANopen* nach **Rechtsklick auf das Knotensymbol** → *Gerätezugriff* anschauen.

- Öffnen Sie nacheinander die EDS-Dateien beider Knoten, studieren Sie den Aufbau der Objektverzeichnisse und vergleichen Sie die Anordnung der Objekte mit den Vorgaben des Geräteprofils DS301 (Abschnitt 9.6.2, S.84f.)!

📖 Zum Auslesen des aktuellen Inhaltes eines Objektes dient der Button *Lesen*. Ist der angezeigte Wert grau hinterlegt, kann man den Dateninhalt des Objektes nicht überschreiben (Attribut *read only*). Weiß hinterlegte Werte hingegen können geändert und anschließend mit dem Button *Schreiben* in das Objekt geschrieben werden. Dies soll an dieser Stelle aber noch nicht geschehen!

- Schließen Sie das Fenster *Gerätezugriff*!

## 5.4. PDO-Linking mittels SDO-Zugriff

📖 Der Austausch von Prozessdaten erfolgt bei CANopen mithilfe von Prozessdatenobjekten (PDOs). Dafür muss bereits im Vorfeld festgelegt werden, welches TPDO (*Transmit-PDO*) für ein bestimmtes Telegramm die Rolle des Producers und welches RPDO (*Receive-PDO*) die Rolle des Consumers übernimmt. Diesem Zweck dient das PDO-Linking.

Die digitalen Eingänge des I/O-Moduls stellen in diesem Versuch ein *Human-Machine-Interface* (HMI) dar. Mit diesem soll es möglich werden, durch Betätigung der Taster am Board die Roboterachsen direkt anzusteuern. Das Bitmuster der digitalen Eingänge (Schaltzustand der Taster) wird vom TPDO0 des I/O-Moduls versendet. Auf der anderen Seite kann das RPDO0 des Gateways diese Daten empfangen und so auswerten, dass je nach Zustand der einzelnen Bits innerhalb der empfangenen Daten eine bestimmte Roboterachse angesteuert werden kann. Folglich muss eine Kommunikationsverbindung vom TPDO0 des I/O-Moduls zum RPDO0 des Gateways aufgebaut werden.

- Starten Sie eine neue Aufzeichnung im *Trace*-Fenster! Da momentan keine Telegramme über den Bus versendet werden, bleibt das Fenster zunächst leer.
- Informieren Sie sich im CANopen-Profil DS301, Abschnitt 9.6.2 (S.84f.) darüber, unter welchen **Indizes** des Objektverzeichnisses das erste TPDO und das erste RPDO zu finden sind! Ermitteln Sie im Abschnitt 9.6.3 des Profils DS301 (S.106 bzw. S.111) außerdem den **Subindex**, unter dem die vom TPDO0 bzw. RPDO0 verwendete COB-ID gespeichert ist! Tragen Sie die gefundenen Indizes und Subindizes in die nachfolgende Tabelle ein.

Tabelle 5-4: COB-ID's der ersten PDO's

	Index	Sub-index	COB-ID	
			I/O-Modul	Roboter
TransmitPDO0			0x...	0x
ReceivePDO0			0x	0x

- Lassen Sie sich die COB-IDs des TPDO0 und des RPDO0 **für beide Knoten** anzeigen: Öffnen Sie zu diesem Zweck das Fenster *Generischer Zugriff* (**ProCANopen**: Menü *Netzwerk* → *Generischer Zugriff*)! Markieren Sie im Übersichtsbaum (links) den Eintrag **SDO**! Tragen Sie die entsprechende *Knotennummer*, *Index* **und** *Subindex* ein und betätigen Sie den Button *Lesen*! Vervollständigen Sie mit den gefundenen Werten Tabelle 5-4!

**Tipp:** Zum Auslesen der COB-ID's im CANsetter kann alternativ auch der *Gerätezugriff* aus dem Kontextmenü des Knotens genutzt werden.

Die mittels SDO-Zugriff zum Auslesen der COB-ID's versendeten Telegramme werden im *Trace*-Fenster angezeigt.

- Analysieren Sie die Zusammensetzung der Telegramme und vergleichen Sie diese mit Tabelle 5-4! Tragen Sie in folgende Tabelle den Inhalt eines Request-Telegramms (*Client-SDO*) und des zugehörigen Response-Telegramms (*Server-SDO*) in hexadezimaler Form ein:

Tabelle 5-5: SDO-Telegramme zum Auslesen einer COD-ID

Telegramm	ID	DLC	CMS	Index		Sub	Parameter			
			Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
C-SDO										
S-SDO										

\*CMS: **CAN Message Specification** – legt die Funktion des SDO fest.

Im PDO0-Kanal zwischen I/O-Modul und Gateway soll das TPDO0 des I/O-Moduls die COB-ID des RPDO0 des Roboter-Gateways erhalten.

- Tragen Sie die dafür erforderliche Änderung in folgende Tabelle ein:

Tabelle 5-6: Änderung der COB-ID eines PDO (Linking)

Node-ID	Knotenname	Index	Subindex	alter Wert	neuer Wert
				0x_____	0x_____

- Ändern Sie mit Hilfe des *Generischen Zugriffs* die entsprechende COB-ID (Button *Schreiben*)!

Eine neue COB-ID wird beim I/O-Modul direkt übernommen (Beim Roboter wäre dafür ein Software-Reset notwendig).

Der Aufbau der Kommunikationsverbindung ist damit abgeschlossen.

- Schließen Sie den Dialog *Generischer Zugriff*!
- Schalten Sie nun im **ProCANopen** den Zustand der *Status Machine* auf *Operational* (Menü *Netzwerk* → *Start aller Knoten* - vgl. auch CAN-Zustandsdiagramm, siehe CiA DS 301 – 9.4.2, S.75f.)!
- Wechseln Sie zum *Trace*-Fenster!

Das Umschalten der *Status Machine* erfolgte durch ein NMT-Telegramm (COB-ID 0). Außerdem zeichnete das *Trace*-Fenster mehrere TPDO-Telegramme des Gateways auf. Der Roboter sollte sich nun durch Betätigung Tasten am I/O-Modul „von Hand“ steuern lassen.

- Testen Sie diese manuellen Bedienfunktionen!

**Hinweis:** Wird der Taster RES (für Reset) am I/O-Modul betätigt (rechte Taste nach oben), ist ein anschließendes Ausführen des Homing-Mode (rechte Taste nach unten) unumgänglich, da mit dem Reset-Befehl alle Istwerte der Motorpositionen auf NULL gesetzt werden, selbst dann, wenn sie gar nicht auf der Nullposition (mechanische Endlage) stehen.

- Sollte die Ansteuerung des Roboters nicht funktionieren, suchen und beheben Sie die Fehlerursache mithilfe der folgenden Hinweise:

- Werden die sechs Telegramme TPDO1...TPDO6 des Roboter-Gateways angezeigt? Falls nicht, schalten Sie die Spannungsversorgung des Gateways für ca. 5s aus und danach gleich wieder ein und starten Sie die Knoten erneut (*Start aller Knoten*)!
- Erscheint bei Betätigung eines beliebigen Tasters im *Trace*-Fenster das zugehörige TPDO0-Telegramm des I/O-Moduls und ändert sich dessen Nutzdateninhalt bei Tastenbetätigung? Falls nicht, überprüfen Sie den Gerätestatus des I/O-Moduls (*Operational?*)!
- Überprüfen Sie die COB-ID des TPDO0-Telegramms des I/O-Moduls! Stimmt die COB-ID nicht mit der von Ihnen geänderten COB-ID überein, wurde das Linking nicht korrekt ausgeführt. Stoppen Sie in diesem Fall die Knoten (Zustand *Pre-Operational*), prüfen die COB-ID's von TPDO0 und RPDO0 beider Knoten und korrigieren Sie die falsche COB-ID!

Die einzelnen Zustands- und Positionsdaten der Achsen des Roboters werden (vom TR-CAN-Gateway) mit den entsprechenden PDO's (TPDO1..6 - für Achse 1..6) über den CAN-Bus versendet (so festgelegt in den Mapping-Parametern). Da das Gateway die TPDO-Telegramme im Gerätestatus *Operational* zyklisch auf den Bus legt, werden die entsprechenden Telegramme im **Trace**-Fenster ständig aktualisiert (siehe Spalte *Time*).

- Beobachten Sie die Telegramme, während Sie jeweils einen der Bedienschalter betätigen!
- Bewegen Sie alle sechs Achsen des Roboters auf unterschiedliche Positionen (rechten Taster *Homing/Reset* bitte nicht betätigen)! Übertragen Sie anschließend die Nutzdatenbytes der TPDOs 1..6 des Roboters aus dem **Trace**-Fenster in die folgende Tabelle!

Tabelle 5-7: Nutzdaten der TPDO's 1..6 des Roboter-Gateways

	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
TPDO1									
TPDO2									
TPDO3									
TPDO4									
TPDO5									
TPDO6									

Eine genaue Analyse dieser Telegramminhalte wird Ihnen später mit dem Verständnis des PDO-Mapping möglich sein.

- Setzen Sie nun die Knoten wieder in den Status *Pre-Operational*, um das zyklische Versenden der PDO-Telegramme zu stoppen (**ProCANopen**: Menü Netzwerk → Pre-Operational *aller Knoten*) – siehe NMT-Telegramm im **Trace**-Fenster!



## 5.5. PDO-Mapping

Zum besseren Verständnis der Funktionsweise des Mappings werden Sie im Folgenden die Mapping-Parameter aller PDO's des Gateways analysieren.

- Füllen Sie mit Hilfe des CANopen-Standards DS 301 (Abschnitte 9.6.2 und 9.6.3) die Spalten *Index* und *Subindex* der folgenden Tabelle aus. **Beachten Sie bitte:** Gesucht sind hier *nicht* die Indizes/Subindizes der PDO Communication Parameter, sondern die der **PDO Mapping Parameter**.
- Informieren Sie sich über die Bedeutung und Funktionsweise der Transmit- bzw. Receive-PDO-Mapping-Parameter in DS 301, Abschnitt 9.6.3, S.109 ff. (Struktur) und DS 402, Abschnitt 7.3.1 bzw. 7.3.2 (Bedeutung der gemappten Objekte).
- Ermitteln Sie mithilfe des generischen Zugriffs die Dateninhalte (hexadezimal) aller in der Tabelle indizierten Objekte und übertragen Sie diese in die Spalte *Inhalt des Objektes*.

Tabelle 5-8: Mapping-Parameter der RPDO's des Roboter-Gateways

Name	Index	Sub	Inhalt des Objektes
1 <sup>st</sup> Receive PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
2 <sup>nd</sup> Receive PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
3 <sup>rd</sup> Receive PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
4 <sup>th</sup> Receive PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
5 <sup>th</sup> Receive PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
6 <sup>th</sup> Receive PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
7 <sup>th</sup> Receive PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)

- Welche Informationen enthalten die Mapping-Parameter?

68	41	01	10
----	----	----	----

(Hinweis: Die Objektinhalte sind hex-codiert!)

---



---



---

- Füllen Sie auf die gleiche Weise auch die folgende Tabelle für die TPDO's aus!

*Tabelle 5-9: Mapping-Parameter der TPDO's des Roboter-Gateways*

Name	Index	Sub	Inhalt des Objektes
1 <sup>st</sup> Transmit PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
2 <sup>nd</sup> Transmit PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
3 <sup>rd</sup> Transmit PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
4 <sup>th</sup> Transmit PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
5 <sup>th</sup> Transmit PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
6 <sup>th</sup> Transmit PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)
7 <sup>th</sup> Transmit PDO Mapping Parameter des Roboters		0	(Anzahl der Receive PDO Mappingobjekte)
			(Mappingobjekt1)
			(Mappingobjekt2)

- Entnehmen Sie dem Objektverzeichnis die **Namen der** im TPDO2 und im RPDO2 des Roboter-Gateways **gemappten Objekte** und tragen Sie diese in nachfolgende Tabelle ein!

**Bitte beachten Sie:** Es geht hier nicht um Index und Name der PDOs, sondern um Index und Name der **in den PDOs gemappten Objekte**.

*Tabelle 5-10: Gemappte Objekte von TPDO2 und RPDO2 des Roboter-Gateways*

PDO im Gateway	Index des Mapp.Obj.	Sub	Datenlänge (Byte)	Name des Objektes
TPDO2				
RPDO2				

- Übertragen Sie das **Nutzdatensegment** des TPDO2 des Roboter-Gateways aus dem **Trace-Fenster** bytewise in folgende Tabelle; ordnen Sie die einzelnen Bytes den gemappten Objekten (siehe vorherige Tabelle) zu! Interpretieren Sie die einzelnen Bytes (Byte 0 und 1 bitweise, Bytes 2 bis 5 insgesamt)! Zur Bedeutung der einzelnen Bytes: CiA DS402 - 10.3.



Tabelle 5-11: Zuordnung der Nutzdaten des TPDO2 zu dessen gemappten Objekten

Byte	0	1	2	3	4	5	6	7
Inhalt (hex)								
Name des Objektes								
Interpretation (DS402, S.59)								

## 5.6. Der PDO-Zugriff

📖 Zweck und Funktionsprinzip des PDO-Linking und des PDO-Mapping sollten Ihnen jetzt klar sein. Die im Gateway und im I/O-Modul definierten Mapping-Parameter bleiben im Praktikum unverändert. Durch generischen Zugriff können nun nicht nur SDO- sondern auch **PDO**-Telegramme versendet werden. Mithilfe eines solchen sollen im Folgenden einige der am I/O- Modul angeschlossenen Lampen zum Leuchten gebracht werden. Die Lampen sind an den ersten acht binären Ausgängen (erstes digitales Ausgangsbyte) des I/O-Moduls angeschlossen. Zum Zusammenstellen des dafür erforderlichen PDO-Telegramms benötigen Sie neben der erforderlichen COB-ID (wird mit dem Telegramm versendet) Informationen über Inhalt und Länge des Nutzdatensegments. Diese Informationen können den Mapping-Parametern des sendenden bzw. empfangenden PDOs entnommen werden.

- Starten Sie in **ProCANopen** alle Knoten!
- Ermitteln Sie die **COB-ID des RPDO0 des I/O-Moduls**: \_\_\_\_\_
- Ermitteln Sie aus den **Mapping-Parametern des RPDO0 des I/O-Moduls** die **Indizes/Subindizes aller darin gemappten Objekte**: \_\_\_\_\_
- Welche Objekte werden mit diesen Indizes/Subindizes angesprochen, welche Datenlänge (in Bytes) haben diese?  
\_\_\_\_\_
- **Wieviele Nutzdaten-Bytes** müssen insgesamt an das **RPDO0** gesendet werden? \_\_\_\_\_
- Welches der gemappten Objekte entspricht dem ersten digitalen Ausgabebyte des I/O-Moduls? Überschreiben Sie zu diesem Zweck im Objektverzeichnis den Wert des gemappten Objektes mit *1* (leuchtet daraufhin die oberste Lampe am Board, haben Sie das erste digitale Ausgabebyte angesprochen)?
- In welchem Byte des Nutzdatenfeldes des RPDO0-Telegramms wird folglich das erste digitale Ausgabebyte des I/O-Moduls übertragen? \_\_\_\_\_
- Ermitteln Sie das Datenbyte (hexadezimal), das dem in folgender Tabelle angegebenen Bitmuster entspricht!

Tabelle 5-12: Zu sendendes Bitmuster

Bit	7	6	5	4	3	2	1	0
Wert	1	0	1	0	1	1	0	1
Hexwert								

- Tragen Sie die COB-ID und die Nutzdaten des zu übertragenden PDO-Telegramms in folgende Tabelle ein:

Tabelle 5-13: COB-ID und Nutzdaten des zu übertragenden PDO-Telegramms


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

- Öffnen Sie unter *Generischer Zugriff* den Eintrag *Register PDO/SYNC!*

**ACHTUNG:** Die Anzahl der durch die Mapping-Parameter vorgegebenen Datenbytes eines PDO-Telegramms muss beim Ausfüllen genau eingehalten werden! Nicht verwendete, aber durch **das Mapping geforderte** Datenbyte-Felder können mit einer 0 gefüllt werden. Führende Nullen in einem Byte können weggelassen werden.

- Markieren Sie die Option *Hexadezimal* und füllen Sie **nur die erforderlichen** Felder aus!
- Senden Sie das PDO-Telegramm (*Sende PDO*) und überprüfen Sie die erfolgreiche Übertragung am Leuchtstatus der Lampen! Überprüfen Sie auch das entsprechende Telegramm im *Trace*-Fenster! Sie dürfen gerne weitere Bitkombinationen zur Ansteuerung der Lampen versenden. Setzen Sie schließlich alle Ausgänge wieder auf 0!
- Stoppen Sie die Messung im *Trace*-Fenster!

## 5.7. CANoe als Steuerungstool

 Neben ProCANopen ermöglicht auch **CANoe** selbst das Versenden von PDO-Telegrammen über den CAN-Bus. Zu diesem Zweck befindet sich im Fenster *Simulationsaufbau* ein Sendezweig, in den der dafür erforderliche Generatorblock eingefügt werden kann. Damit werden Sie im Folgenden eine Sendeliste mit Telegrammen zur Ansteuerung aller sechs Antriebe des Roboters (Bewegung auf vorgegebene Positionen) erstellen und testen.

- Ermitteln Sie mittels Gerätezugriff (*ProCANopen*) die **COB-IDs** der RPDOs 1..6 **des Roboter-Gateways** und übertragen Sie diese in die entsprechende Spalte der folgenden Tabelle!

Tabelle 5-14: PDO-Telegramme zur Ansteuerung der Roboterachsen

Roboter		COB-ID	DLC	Nutzdatenbytes							
				0	1	2	3	4	5	6	7
Achse 4	RPDO4										
Achse 5	RPDO5										
Achse 2	RPDO2										
Achse 1	RPDO1										
Achse 3	RPDO3										
Achse 6	RPDO6										

- Welche Objekte des Objektverzeichnisses des Roboter-Gateways sind in dessen RPDOs 2..7 gemappt (Hinweis: Tabelle 5-10)?
- Ergänzen Sie in obiger Tabelle 5-14 *Data Length Code* (DLC) und *Nutzdaten* für alle aufgeführten RPDOs entsprechend nachfolgender Angaben!

Damit der gewünschte Positionswert angefahren werden kann, sind im **Controlword** nur das Bit 0 (*Switch On*), das Bit 3 (*Enable Operation*) und das Bit 11 (*Move Position*) zu setzen (vergl. DS 402, Abschnitt 10.3.1, S.56 ff.), alle anderen Bits müssen den Wert „0“ erhalten.

Tabelle 5-15: Controlword

Bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Wert																
Hex.																

Die Zielpositionen für die fünf Roboterachsen sind folgender Tabelle zu entnehmen:

Tabelle 5-16: Zielpositionen der Roboterachsen

	Achse 4	Achse 5	Achse 2	Achse 1	Achse 3	Achse 6
Position hex. (dez.)	c8 (200)	b4 (180)	37 (55)	46 (70)	100 (256)	3c (60)

Werden in PDO's Daten aus Objekten mit einer Datenlänge größer als 1 Byte übertragen, so wird jeweils das niederwertige Byte **vor** dem höherwertigen im Telegramm angehängt.

➤ Öffnen Sie im Programm **CANoe** das Fenster **Simulationsaufbau** (Menü *Simulation*)!

Im linken Teil des Fensters finden Sie die Blockansicht, links vom Block *Netzwerk* symbolisieren zwei farbige Linien das Bussystem. An diese Linien können weitere Blöcke angeschlossen werden. Für die folgende Aufgabe wird ein Generator-Block benötigt.

- Fügen Sie nach Rechtsklick auf die farbigen Linien einen Block *CAN-Interaktiver-Generator* ein! Wählen Sie aus dem Kontextmenü des eingefügten Generator-Blocks den Befehl **Konfiguration...**!
- Öffnen Sie im unteren Bereich des Dialogs *CAN IG* das Register *Rohdaten*!
- Klicken Sie mit der rechten Maustaste in den oberen Bereich des Dialogs und wählen Sie **Frame aus Datenbasis hinzufügen**! Markieren Sie in der Symbolauswahl unter *Rx Frames* auf das RPDO3-Frame → **Übernehmen**! Tragen Sie im unteren Teil die Rohdaten für die TPDO3-Botschaft entsprechend der ersten Zeile in Tabelle 5-14 ein!
- Ergänzen Sie die Sendeliste um fünf weitere Frames entsprechend der verbliebenen Zeilen aus Tabelle 5-14!
- Klicken Sie nacheinander für alle Frames in die Spalte Trigger, wählen Sie als Triggertaste jeweils die der Zeilennummer des Frames entsprechende Ziffer (1..6)!
- Schließen Sie den Dialog *CAN IG*!
- Starten Sie die Messung im *Trace*-Fenster! Bringen Sie den Roboter in seine Grundstellung (*Home*-Taste) und setzen Sie die aktuellen Positionswerte anschließend auf „0“ (*Reset*-Taste)!
- Drücken Sie nun die Zifferntaste 1, um die erste Botschaft über den Bus zu senden, und beobachten Sie dabei sowohl die Reaktion des Roboters als auch die Anzeige der Telegramme im *Trace*-Fenster! Senden Sie anschließend nacheinander die Botschaften 2 bis 6 über den Bus und beobachten Sie weiterhin Roboter und *Trace*-Fenster.
- Vergleichen Sie im *Trace*-Fenster die vorgegebenen Zielpositionen (**target\_position**) in den RPDOs mit den erreichten Positionen (**position\_actual\_value**) in den TPDOs!

Bedingt durch den hier verwendeten Fahrmodus des Teach-Roboters (Geschwindigkeitsmodus) kann es vorkommen, dass die anzufahrende Position um wenige Inkremente überfahren wird.

- Versetzen Sie nach Abschluss aller Bewegungen den Roboter wieder in den Grundzustand (*Homing-Mode*)!

## 5.8. CANoe als Diagnose- und Visualisierungstool

CANoe enthält weitere Diagnose- und Visualisierungsmöglichkeiten für CAN-Botschaften. Das von Ihnen bisher genutzte **Trace**-Fenster listet zwar alle auf dem Bus erscheinenden Telegramme auf, zur Auswertung der meist verschlüsselten Daten ist es jedoch nur bedingt geeignet. Zur besseren Datenanalyse stellt CANoe weitere Darstellungsmöglichkeiten zur Verfügung. Im **Daten**-Fenster bspw. können einzelne Variablen, wie z.B. die Positions- und Zustandswerte der sechs Roboterachsen in einem beliebigen Zahlenformat dargestellt werden.

- Öffnen Sie das Fenster **Daten** (Menü *Analyse*)! Fügen Sie diesem Fenster die nachfolgenden Signale zu (Kontextmenü → **Signale hinzufügen...**), Sie finden diese im *Auswahlfenster* in der Datenbank unter *PRJDB* → *Signale*; zur Mehrfachauswahl halten Sie bitte die *Strg*-Taste gedrückt!
  - *position\_actual\_value*
  - *position\_actual\_value\_motor\_2*
  - *position\_actual\_value\_motor\_3*
  - *position\_actual\_value\_motor\_4*
  - *position\_actual\_value\_motor\_5*
  - *position\_actual\_value\_motor\_6*
- Die Skalierung nehmen Sie beispielhaft an einem der Signale vor. Bewegen Sie den Drehantrieb des Roboters mit Hilfe des zweiten Tasters von rechts am I/O-Modul zunächst auf dessen eine Endlage und danach auf die andere, notieren Sie sich für diese Positionen die ausgegebenen Rohwerte des Signals *position\_actual\_value\_motor\_6*! Wählen Sie aus dem Kontextmenü dieses Signals im **Daten**-Fenster den Befehl *Bearbeiten...*! Zur Freigabe der Einstelloptionen aktivieren Sie die Option *Datenbasisverbindung lösen*! Ändern Sie den Signal-Namen in *Drehposition*! Skalieren Sie die Rohwerte so, dass der Bereich zwischen den beiden Endlagen auf dem Bildschirm in Prozent (0..100%) ausgegeben wird! Schließen Sie das Fenster *Botschafts- und Signalparameter* (OK)!
- Öffnen Sie in der Zeile des Signals *Drehposition* das Kontextmenü des Feldes *Balken* und aktivieren Sie die Option *Minimum- und Maximumwert anzeigen*! Wählen Sie aus demselben Kontextmenü den Befehl *Einstellungen...*! Geben Sie als *Minimum* **0** und *Maximum* **100** ein sowie als *Ursprung* **0**! Bestätigen Sie die Eingaben (OK) und beobachten Sie die Anzeige bei unterschiedlichen Drehpositionen!