

Anwenderhandbuch



COPYRIGHT: The Operating Instructions TZY 10771
is owned by TWK-ELEKTRONIK GmbH and is
protected by copyright laws and international treaty provisions.

© 1999 by TWK-ELEKTRONIK GmbH
POB 10 50 63 ■ 40041 Düsseldorf ■ Germany
Tel. +49/211/63 20 67 ■ Fax +49/211/63 77 05
e-mail: info@twk.de ■ internet: www.twk.de

Inhaltsverzeichnis

1. Einführung	5
2. Prinzipschaltbild: TWK-Winkelcodierer für DeviceNet	6
3. Installationshinweise	6
3.1 Winkelcodierer in Steckerversion	6
3.2 Winkelcodierer mit Anschlußhaube	7
3.2.1 Einstellung der Teilnehmeradresse (MAC-ID)	7
3.2.2 Einstellung der Baudrate	7
3.2.3 Status-LED	8
4. Die Betriebsarten des Winkelcodierers	9
4.1 Polling Mode	9
4.2 Bit Strobed Mode	9
4.3 Change of State Mode	9
4.4 Cyclic Mode	9
5. Winkelcodierer - Parameter	10
5.1 Parameterbeschreibung	10
5.2 Winkelcodierer - Parameterwerte	11
6. Winkelcodierer - Eingangsdaten	11
7. Winkelcodierer - Statusinformationen	12
8. Das DeviceNet-Schicht7-Protokoll	13
8.1 DeviceNet Objektaufbau	13
8.2 Die Objekte des Winkelcodierers	13
8.2.1 Identity Object	15
8.2.2 Message Router Object	15
8.2.3 DeviceNet Object	16
8.2.4 Assembly Object	16
8.2.5 Connection Object	17
8.2.6 Position Sensor Objekt	18
8.3 DeviceNet Verbindungen	19
8.4 Das DeviceNet-Protokoll	20
8.5 Aufbau einer Verbindung zum Winkelcodierer	20
8.6 Parametrieren des Winkelcodierers	21
8.7 Eingangsdaten des Winkelcodierers abfragen	22
8.8 Statusinformationen abfragen	22
8.9 Parameterwerte im EEPROM sichern	23
8.10 Parameterwerte in Default-Zustand bringen	23
9. DeviceNet-Manager	24
9.1 Installation EDS-Datei	24
9.2 Einbinden in den Bus	25
9.3 Winkelcodierer parametrieren	26
9.4 Winkelcodierer in die Scanlist aufnehmen	28
9.5 Online gehen und Daten übertragen	31
9.6 Adresse und Baudrate beim Winkelcodierer in Steckerversion einstellen	32
9.7 Parameter im EEPROM abspeichern	34

9.8 Defaultwerte der Parameter laden	34
9.9 Beliebiges Attribut auslesen	35
10. RS-Networx for DeviceNet	36
10.1 Installation EDS-Datei	36
10.2 Einbinden in den Bus	37
10.3 Winkelcodierer parametrieren	38
10.4 Winkelcodierer in die Scanlist aufnehmen	39
10.5 Adresse und Baudrate beim Winkelcodierer in Steckerversion einstellen	41
10.6 Parameter im EEPROM abspeichern	41
10.7 Defaultwerte der Parameter laden	42
10.8 Beliebiges Attribut auslesen	43
Anhang A: Literatur	43

1. Einführung

DeviceNet ist ein auf dem CAN (Controler Area Network) aufbauendes Bussystem. Der ursprünglich von der Robert Bosch GmbH entwickelte CAN-Bus, wurde speziell für die Automobilindustrie entwickelt und wird heute in vielen Modellen der PKW-Oberklasse eingesetzt.

Durch die hohen Anforderungen der Automobilindustrie sind die folgenden für den CAN-Bus charakteristischen Merkmale entstanden, die auch für DeviceNet ihre Gültigkeit behalten:

- hohe Übertragungssicherheit (Hamming Distanz 6)
- kurze Antwortzeiten
- Datenformat: max. 8 Byte
- Nachrichtenorientiertes Bussystem
- Verfügbarkeit von ausgereiften und kostengünstigen Systemkomponenten
- Offenes Konzept und leichte Erweiterbarkeit

Während durch das CAN-Protokoll, spezifiziert in der Norm ISO/DIS 11898, nur die Schicht 1 und die Schicht 2 des ISO/OSI-Kommunikationsmodelles, beschrieben sind, spezifiziert DeviceNet zusätzlich das Übertragungsmedium der Schicht 1 als auch die Schicht 7.

Folgende Übersicht macht dieses noch einmal deutlich:

ISO-Schicht 7	Application Layer	DeviceNet-Spez.
ISO-Schicht 2	Data Link Layer (Protocol Layer)	CAN Protokoll Spezifikation
	Physical Signaling	
ISO-Schicht 1	Transceiver	DeviceNet Spezifikation
	Transmission Media	

Abb. 1.1 DeviceNet Kommunikationsmodell

Gerade die Verfügbarkeit an kostengünstigen IC's und die hohe Eigensicherheit des CAN-Systems macht diesen Bus auch für die Automatisierungstechnik interessant. Die europäische Organisation zur Verbreitung von CAN in der Automatisierung ist der CiA - CAN in Automation (<http://www.can-cia.de>). Diese Hersteller- und Nutzerorganisation vertritt sowohl das in Deutschland entwickelt CANopen-Protokoll, als auch das vor allem durch die Firma Allen-Bradley favorisierte DeviceNet-Protokoll, sowie weitere Derivate.

Die Internationale Hersteller- und Anwender-Organisation für das DeviceNet ist jedoch die ODVA - Open DeviceNet Vendor Association (<http://www.odva.org>), die Herausgeber der DeviceNet Spezifikation [1] ist.

Durch die Spezifikation von Schicht 7 und des Hardwareanteils von Schicht 1 durch DeviceNet ergeben sich die folgenden Eigenschaften:

- Aufbau mit Hauptstrang (Trunkline) und Stichleitungen (Dropline) möglich
- Bis zu 64 Teilnehmer
- Teilnehmer ohne Unterbrechung des Busbetriebs vom Bus lösbar
- Datenleitung und Teilnehmerversorgung in einem Kabel
- Datenraten 125, 250, 500 kBaud (siehe Tabelle 1.1)

Datenrate	Max. Länge Trunkline *	Max. Länge	
		einer Dropline	insgesamt
125K	500 m	6 m	156 m
250K	250 m	6 m	78 m
500K	100 m	6 m	39 m

Tabelle 1.1 Kabellängen

* Werte gelten für das dicke Kabel (DeviceNet thick cable), für das dünne Kabel (DeviceNet thin cable) ist die maximale Länge immer 100 m unabhängig von der Datenrate.

- Schutz vor Verdrahtungsfehlern
- Unterstützung von Teilnehmern, die über das Buskabel versorgt werden und solchen, die eine separate Spannungsversorgung besitzen
- Objektorientierte Anwendungsschicht
- Unterstützung der Kommunikation zwischen beliebigen Teilnehmer- und Master/Slave-Kommunikation
- Aufdeckung von doppelt vergebenen Knotenadressen

2. Prinzipschaltbild: TWK-Winkelcodierer für DeviceNet

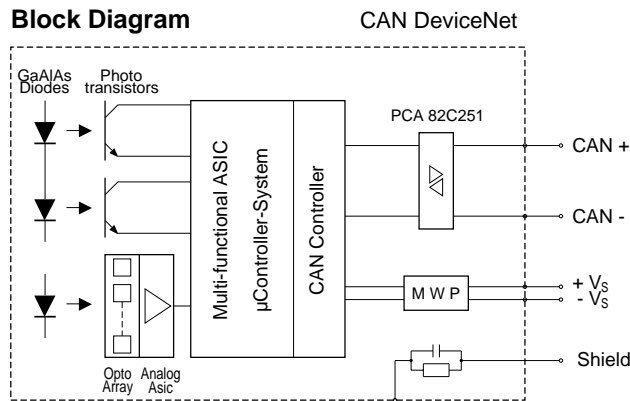


Abb. 2.1 Prinzipschaltbild Winkelcodierer CRN/D

Die Information über die aktuelle Position des Winkelcodierers wird über ein Multifunktions-ASIC und integriertem Mikroprozessor (Hostcontroller) erfasst. Die Kommunikation zwischen Hostcontroller und CAN-Bus einschliesslich der Implementierung des Kommunikationsprotokolls erfolgt mittels des CAN-Controllers SJA1000. Dieser Controller beinhaltet z.B. die Priorität des Buszugriffs, der definiert wird durch die Identifier der Messages, des weiteren Fehlerbehandlungs-routinen, Routinen des Arbitrationsverlustes bei Buszugriff, u.a..

Das Interface zwischen dem CAN Controller und dem physikalischen Bus ist mit dem CAN Transceiver 82C251 gemäss dem CiA Standard ISO/DIS 11898 realisiert. Gemäss DeviceNet Spezifikation ist für die Versorgungsspannung eine MWP (Mis-Wiring Protection) - Schaltung - und für den Schirm eine RC-Kombination vorgesehen.

Der Abschlusswiderstand ist bei DeviceNet nie im Teilnehmer integriert, er muss immer separat am Ende der Trunkline aufgesteckt werden.

3. Installationshinweise

3.1 Winkelcodierer in Steckerversion

Der Winkelcodierer in Steckerversion ist mit einem 5-poligen Steckverbinder in der Ausführung Micro oder Mini ausgestattet. (Anschlußbelegung siehe Beiblatt TY...). Die Einstellung von Adresse und Baudrate geschieht per Software (z.B. DeviceNet-Manager von Allen-Bradley). (Siehe auch [Kapitel 9.6](#) bzw. [Kapitel 10.5](#))

Die Werkseinstellung ist: Adresse 1
Baudrate 125 KB

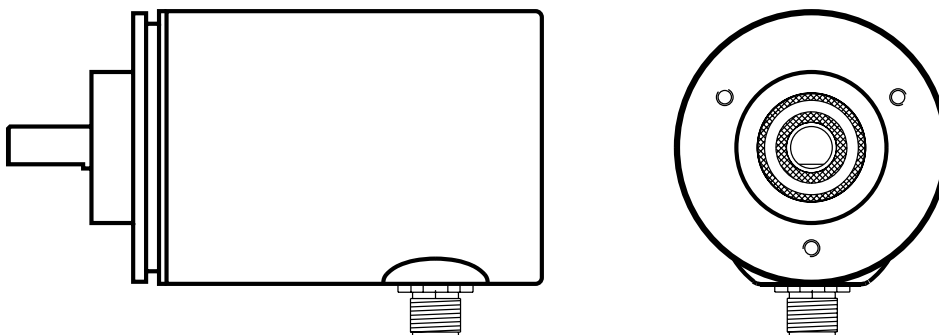


Abb. 3.1 Winkelcodierer in Steckerversion

3.2 Winkelcodierer mit Anschlußhaube

Die Version mit Anschlußhaube besitzt je eine PG-Verschraubung für Bus-In und für Bus-Out. Innerhalb der Anschlußhaube wird der Bus, sowie die im Kabel mitgeführte Versorgungsspannung auf Klemmen verdrahtet. (Siehe der Haube beigelegte Steckerbelegung TY...).

Zusätzlich wird hier die Einstellung der Teilnehmeradresse (MAC-ID) und der Baudrate vorgenommen.

Der Winkelcodierer wird mit der Haube über einen 15-pol. Sub-D-Stecker kontaktiert. Im Fehlerfall kann der Codierer ohne jeglichen Installationsaufwand ausgetauscht werden. Die Anschlusshaube wird durch Lösen von 2 Befestigungsschrauben vom Winkelcodierer getrennt.

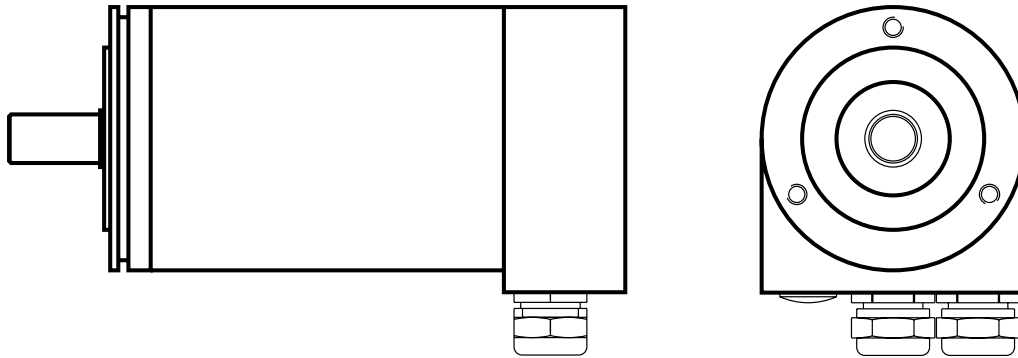


Abb. 3.2 Winkelcodierer mit Anschlußhaube

3.2.1 Einstellung der Teilnehmeradresse (MAC-ID)

Die MAC-ID (Media Access Control Identifier) kann über die Dip-Schalter im Bereich 0-63 gemäss Tabelle 3.1 eingestellt werden. Der Defaultwert ist 1.

Schalter	6	5	4	3	2	1	Adresse
Wertigkeit	2^5	2^4	2^3	2^2	2^1	2^0	
	0	0	0	0	0	0	0
	0	0	0	0	0	1	1
ON = 1
OFF = 0
	1	1	1	1	1	1	63

Tabelle 3.1 Einstellung der Teilnehmeradresse (MAC-ID)

3.2.2 Einstellung der Baudrate

Schalter	8	7	Baudrate
	0	0	125KB
ON = 1	0	1	250KB
OFF = 0	1	0	500KB
	1	1	500KB

Tabelle 3.2 Einstellung der Baudrate

Beachten Sie bei der Einstellung der Baudrate, die zulässigen Kabellängen in Tabelle 1.1 .

3.2.3 Status-LED

In der Anschlußhaube befinden sich drei LED, die Aufschluß über den Zustand des Winkelcodierers geben. Hierbei handelt es sich um eine grüne LED für die Versorgungsspannung (V), sowie eine grüne und eine rote LED (MNS), die zusammen die in der DeviceNet-Spezifikation festgelegte Modul-/Netzwerk-Status-LED bilden.

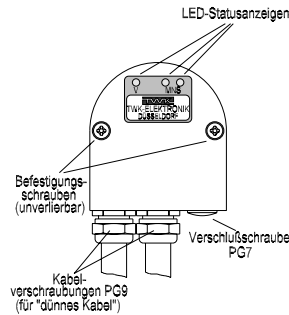


Abb. 3.1 Anschlußhaube

MNS-LED	Status	Erläuterung
Aus	Nicht angeschlossen, Nicht online	Gerät ist nicht online: - Dupl. MAC-ID check nicht beendet - keine Spannungsversorgung
Grün blinkend	Gerät aktiv und online, Verbindungen bestehen nicht	Das Gerät arbeitet unter normalen Bedingungen und ist online, es ist keine Verbindung eingerichtet. - Codierer ist noch nicht vom Master konfiguriert worden - Konfiguration nicht komplett oder fehlerhaft
Grün	Gerät aktiv und online, Verbindungen bestehen	Das Gerät arbeitet unter normalen Bedingungen und ist online, mit Verbindungen im Zustand „eingrichtet“ - Codierer ist durch den Master konfiguriert worden
Rot blinkend	Geringfügiger Fehler und/oder Unterbrechung der Verbindung	Wiederherstellbarer Fehler und/oder eine oder mehrere I/O-Verbindungen sind im unterbrochenen Zustand
Rot	Kritischer Gerätefehler oder kritischer Kommunikationsfehler	Das Gerät hat einen nicht wiederherstellbaren Fehler es muß ersetzt werden. Das Gerät hat einen Fehler festgestellt, der es unmöglich macht mit dem Netzwerk zu kommunizieren. - Doppelt vergebene Teilnehmeradresse (MAC-ID)
Rot&Grün blinkend	Kommunikation unterbrochen und Empfang einer Kennung der Kommunikations-Fehler-Nachfrage	Ein besonderer Kommunikationsfehler. Das Gerät hat einen Fehler im Netzwerkzugriff entdeckt und ist im Kommunikationsfehlerstatus

Tabelle 3.3 LED-Status der Modul-/Netzwerk-Status-LED

4. Die Betriebsarten des Winkelcodierers

Die Betriebsarten des Winkelcodierers bestimmen die Triggerung der Positionswertenerfassung. Der Anwender kann zwischen vier verschiedenen Betriebsarten wählen:

1. **Polling Mode**
2. **Bit Strobed Mode**
3. **Change of State Mode**
4. **Cyclic Mode**

Außerdem ist es möglich mehrere Betriebsarten gleichzeitig einzustellen.

4.1 Polling Mode

Die Standard-Betriebsart im Master-Slave-System ist der Polling Mode. Hierbei fragt der Master alle Teilnehmer zyklisch ab. In einem *Scan* werden so alle Ausgangsdaten an die Slaves übergeben und alle Eingangsdaten von den Slaves gelesen. Die Zeit zwischen zwei Scans ist in der Regel im Master einstellbar.

Der Winkelcodierer liefert seine Eingangsdaten also auf die zyklische Anfrage (Poll-Command) des Masters.

4.2 Bit Strobed Mode

Will man bestimmte (oder auch alle) Slaves zu einem bestimmten Zeitpunkt ansprechen, so verwendet man den Bit Strobed Mode.

Wird der Winkelcodierer in diesem Mode betrieben, so sendet er seine Eingangsdaten als Reaktion auf das Bit Strobe Kommando.

4.3 Change of State Mode

Soll ein Teilnehmer nur dann seine Eingangswerte senden, wenn sich diese geändert haben, so betreibt man ihn im Change of State Mode. Zusätzlich lässt sich in dieser Betriebsart eine **Heartbeat Rate** einstellen. Der Teilnehmer sendet dann nach Ablauf einer internen Zeit neue Eingangsdaten, auch wenn sie sich nicht geändert haben.

Der Winkelcodierer liefert in dieser Betriebsart also seine Positionsdaten bei Änderung **und** nach Ablauf der **Heartbeat Rate**. Die Heartbeat Rate lässt sich im Bereich 2 - 65535 ms im Zeitraster von 2 ms einstellen. (Der DeviceNet-Manager der Firma Allen-Bradley lässt hier nur minimal 48 ms zu).

In dieser Betriebsart lässt sich zusätzlich noch eine Zeitverzögerung im Bereich 2 - 65535 ms einstellen (production inhibit time). Der neue Positionswert wird dann erst nach Ablauf dieser Zeit an den Master gesendet.

4.4 Cyclic Mode

Im Cyclic Mode liefert der Teilnehmer seine Eingangsdaten im festen Zeittakt. Diese **Send Rate** ist in jedem Teilnehmer einstellbar.

Der Cyclic Mode ist also ein Change of State Mode der die Eingangsdaten nur zyklisch und nicht bei Wertänderung sendet.

Streng genommen handelt es sich hier nicht um eine eigene Betriebsart sondern um eine Variante des Change of State Mode. Deswegen spricht man auch vom Change of State / Cyclic Mode.

5. Winkelcodierer - Parameter

Über seine Parameter erlaubt der Winkelcodierer eine Anpassung an kundenspezifische Einsatzfälle. Sie können über den gesonderten Datenverkehr des DeviceNet, dem sogenannten **Explicit Messaging**, geschrieben und gelesen werden.

Zum spannungsausfallsicheren Abspeichern der Parameter im Winkelcodierer steht ein Save-Kommando und zum Laden der Default-Werte ein Restore-Kommando zur Verfügung.

Achtung: Nach einem Save- bzw. Restore-Kommando liefert der Winkelcodierer für ca. 300ms keinen Positionswert. Diese Kommandos sollten daher nie bei einer laufenden Maschine ausgeführt werden.

5.1 Parameterbeschreibung

Parameter	Erklärung
Codeverlauf	Der Codeverlauf gibt an, in welcher Drehrichtung der Ausgabecode steigt. CW - Steigende Werte bei Drehung im Uhrzeigersinn (clockwise) CCW - Steigende Werte bei Drehung im Gegenuhrzeigersinn (counter clockwise) Hinweis: Blickrichtung auf die Welle
Skalierung	Schaltet die Skalierung ein bzw. aus. Die Werte für Auflösung, Messbereich und Referenzwert sind nur bei eingeschalteter Skalierung wirksam.
Auflösung	Die Auflösung gibt die Anzahl der Schritte pro Umdrehung an.
Messbereich	Der Meßbereich gibt die Gesamtschrittzahl an, sie berechnet sich aus: Gesamtschrittzahl = Anzahl Umdrehungen x Auflösung
Referenzwert	Der Referenzwert ist der Wert, der im Referenzpunkt zur Anzeige gebracht wird. Er kann die Werte 0 bis Gesamtschrittzahl - 1 annehmen.
Arbeitsbereich untere Grenze	Unterer Grenzwert eines vom Anwender einstellbaren Arbeitsbereiches.
Arbeitsbereich oberer Grenze	Oberer Grenzwert eines vom Anwender einstellbaren Arbeitsbereiches.

Tabelle 5.1 Winkelcodierer-Parameter

Hinweis: Die Gesamtschrittzahl muß so gewählt sein, dass die Anzahl der Umdrehungen grösser oder gleich eins ist. Wird der Winkelcodierer im Endlosbetrieb verwendet, muss die Anzahl der Umdrehungen gleich 2^n sein. (mit $n=1,2,\dots,12$)

5.2 Winkelcodierer - Parameterwerte

Parameter	Wertebereich	Defaultwert	Datentyp
Codeverlauf Code sequence	CW/CCW	CW	BOOL
Skalierung Scaling	Aus/Ein	Aus	BOOL
Auflösung Resolution	1 - 8192	8192	UNSIGNED INTEGER
Messbereich Measuring Range	1 - 33.554.432	33.554.432	UNSIGNED INTEGER
Referenzwert Preset Value	0 - (eingestellter Messbereich -1)	0	UNSIGNED INTEGER
Arbeitsbereich untere Grenze Work Area Low Limit	0 - 33.554.432	1.048.575	UNSIGNED INTEGER
Arbeitsbereich obere Grenze Work Area High Limit	0 - 33.554.432	32.505.856	UNSIGNED INTEGER

Tabelle 5.2 Parameter
(Der dargestellte Wertebereich sowie die Defaultwerte gelten bei 25-Bit Auflösung)

Hinweis: Die Darstellung der Parameter im Telegramm ist immer Low-Byte vor High-Byte.

6. Winkelcodierer - Eingangsdaten

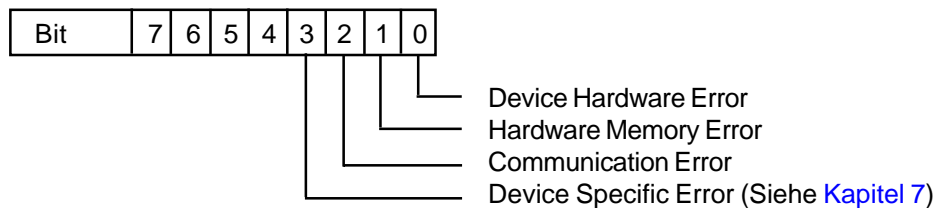
Als Eingangsdaten (aus Sicht des Masters) im I/O-Datenverkehr - **I/O-Messaging** - liefert der Winkelcodierer den Positions-istwert und ein Statusbyte. Die Betriebsart des Winkelcodierers bestimmt dabei den Triggerzeitpunkt der Istwerterfassung. Damit liefert der Winkelcodierer insgesamt fünf Byte Eingangsdaten, die wie folgt zu interpretieren sind:

Byte	Datenfeld				
	1	2	3	4	5
Bit	7 - 0	15 - 8	23 - 16	31 - 24	39 - 32
Wertigkeit	$2^7 - 2^0$	$2^{15} - 2^8$	$2^{23} - 2^{16}$	$2^{31} - 2^{24}$	0000XXXX
Bedeutung	Positionsistwert				Statusbyte

Tabelle 6.1 Eingangsdaten

0 = nicht verwendet , X = verwendet

Statusbyte:



7. Winkelcodierer - Statusinformationen

Zusätzlich zu den Eingangsdaten des I/O-Messaging liefert der Winkelcodierer noch weitere Statusinformationen, die wie die Parameter über das **Explicit Messaging** ausgelesen werden können. Natürlich lassen sich auch die Eingangsdaten ([Kapitel 6](#)) über das Explicit Messaging auslesen.

Bezeichnung	Datentyp	Erläuterung
Arbeitsbereich Statusregister Area state register	Unsigned short integer	Bit 0 : nicht verwendet Bit 1 : gleich 1 bei: Positionswert > oberer Arbeitsbereich Bit 2 : gleich 1 bei: Positionswert < unterer Arbeitsbereich Bit 3-7 : nicht verwendet
Betriebsart Operating Status	Word	Bit 0 : 0 = CW; 1 = CCW Bit 2 : 0 = Skalierung aus; 1 = Skalierung ein
Max. Auflösung pro Umdrehung Single Turn Resolution	Unsigned integer	Angabe in Schritte/Umdrehung
Max. Anzahl von Umdrehungen Number of distinguishable revolutions	Unsigned integer	Angabe in Umdrehungen
Alarmer Alarms	Word	Bit 0 : 1 = Position Error Bit 1-11 : nicht verwendet Bit 12 : 1 = Eeprom Error Bit 13 : 1 = CRC error Bit 14,15 : nicht verwendet
Unterstützte Alarmer Supported alarms	Word	Gibt an welche der im Wort Alarmer aufgeführten Alarm- meldungen derzeit vom Winkelcodierer unterstützt werden (Derzeit werden alle Alarmer unterstützt)
Warnungen Warnings	Word	Bit 0-4 : nicht verwendet Bit 5 : 1 = Positionswert ungleich Referenzwert Bit 6-15 : nicht verwendet
Unterstützte Warnungen Supported Warnings	Word	Gibt an welche der im Wort Warnungen aufgeführten Warn- meldungen derzeit vom Winkelcodierer unterstützt werden (Derzeit werden alle Warnungen unterstützt)
Profil und Software- Version Profile and software version	DWord	Wort 0 = Profilversion (Hex. Darstellung) Wort 1 = Softwareversion (Hex. Darstellung)

Tabelle 7.1 Statusinformationen

Hinweis: Die Darstellung der Daten im Telegramm ist immer Low-Byte vor High-Byte.

8. Das DeviceNet-Schicht7-Protokoll

Hinweis: Zur Inbetriebnahme des Winkelcodierers an einer Allen-Bradley-SPS mit Hilfe des DeviceNet-Managers oder RS-Networx ist die Kenntnis dieses Kapitels nicht zwingend erforderlich. In diesem Fall kann mit [Kapitel 9](#) bzw. [Kapitel 10](#) fortgefahren werden.

8.1 DeviceNet Objektaufbau

Die Schicht 7 des ISO/OSI-Kommunikationsmodells ist bei DeviceNet streng objektorientiert aufgebaut. Jeder Teilnehmer besteht aus einer bestimmten Menge von Objekten. Jedes Objekt beinhaltet Attribute (Daten) und Services (Funktionen) einer ganz bestimmten Komponente des Teilnehmers. Objekte, die die gleiche Systemkomponente repräsentieren, werden wiederum zu Klassen zusammengefaßt.

Das folgende Bild zeigt stark abstrahiert den Aufbau des DeviceNet aus Objektsicht:

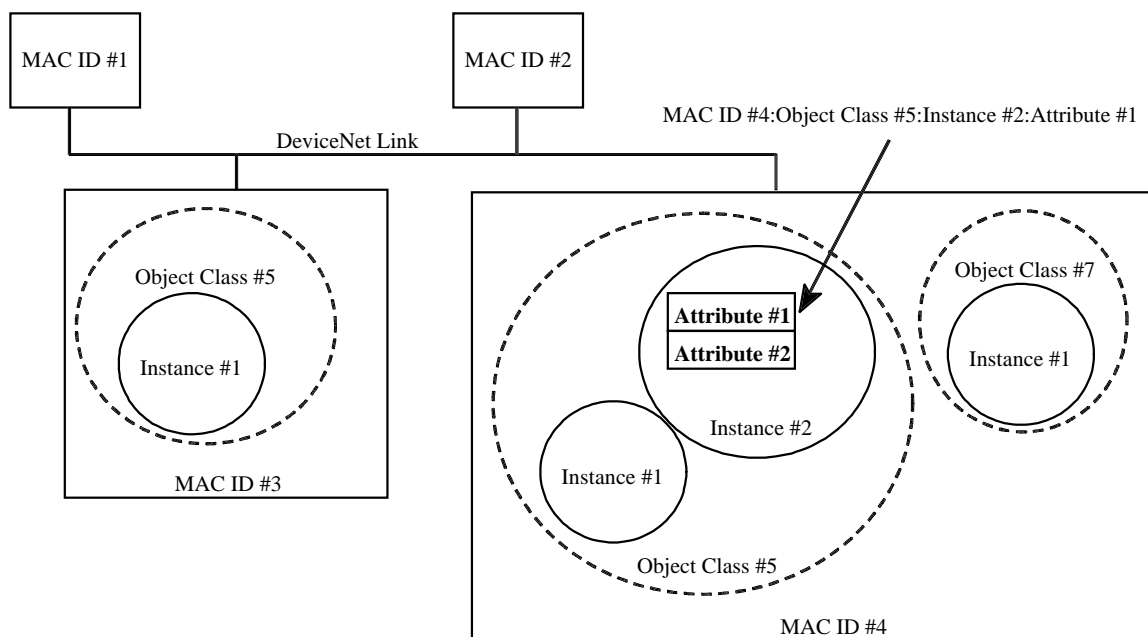


Abb. 8.1 DeviceNet Objektaufbau

Jedes Objekt, das mit Anwenderdaten gefüllt (instanziiert) wird, wird Instanz genannt. So können von einem Objekt mehrere Instanzen gebildet werden.

Alle Klassen, Instanzen, Attribute und Services erhalten Identifier (Integerwerte). Zusammen mit der Teilnehmeradresse (MAC-ID) können dadurch alle Attribute und Services im Bussystem eindeutig adressiert werden. (Dargestellt ist im obigen Bild nur die Adressierung eines Attributes.)

8.2 Die Objekte des Winkelcodierers

Eine Vereinfachung der umfangreichen Möglichkeiten von DeviceNet stellt der **Predefined-Master-Slave Connection Set** dar. Er stellt eine Untermenge von DeviceNet dar und bietet alle Funktionen die bei heute üblichen Master-Slave-Beziehungen nötig sind. Im Wesentlichen werden dabei die Verbindungsarten (Betriebsarten) festgelegt.

Im Predefined-Master-Slave-Connection-Set werden außerdem **Group 2 Server/Client** und **Group 2 only Server/Client** unterschieden. Wobei letzterer sich auf die, für einfache I/O-Teilnehmer wesentlichen, Verbindungsarten beschränkt und so kostengünstige Realisierungen von DeviceNet-Slaves ermöglicht. Für genauere Informationen muß hier jedoch auf die DeviceNet-Spezifikation [1] verwiesen werden. Alle folgenden Erläuterungen beziehen sich auf die DeviceNet-Implementation des Winkelcodierers.

Der Winkelcodierer stellt einen **Group 2 only Server** dar.

Im Winkelcodierer sind die folgenden Objekte enthalten:

Class ID	Class	Erläuterung
01 _{hex}	Identity Object	Beinhaltet generelle Informationen über den Slave, wie z.B.: Hersteller-ID, Seriennummer...
02 _{hex}	Message Router Object	Nimmt alle Messages auf und gibt sie an die entsprechenden Objekte weiter.
03 _{hex}	DeviceNet Object	Enthält die Konfiguration und den Status der physikalischen Verbindung, z.B.: MAC-ID, Baud Rate
04 _{hex}	Assembly Object	Erlaubt die Sammlung von Attributen verschiedener Objekte in einem Objekt.
05 _{hex}	Connection Objekt	Verwaltet sowohl I/O- als auch Explicit Messaging Verbindungen. Für jede Verbindung (z.B. Polling, Bit-Strobe, Explicit Messaging) wird eine Instanz dieses Objektes gebildet.
23 _{hex}	Position Sensor Objekt	Dieses Objekt enthält alle Winkelcodierer-Daten. Hier befinden sich alle Eingangsdaten, Parameter und Statusinformationen des Winkelcodierers.
2b _{hex}	Acknowledge Handler Object	Überwacht den Empfang von Acknowledge Nachrichten bei nachrichtenproduzierenden Objekten, z.B. COS-Verbindung.

Tabelle 8.1 Objekte des Winkelcodierers

Die Schnittstellen der einzelnen Objekte veranschaulicht das folgende Bild:

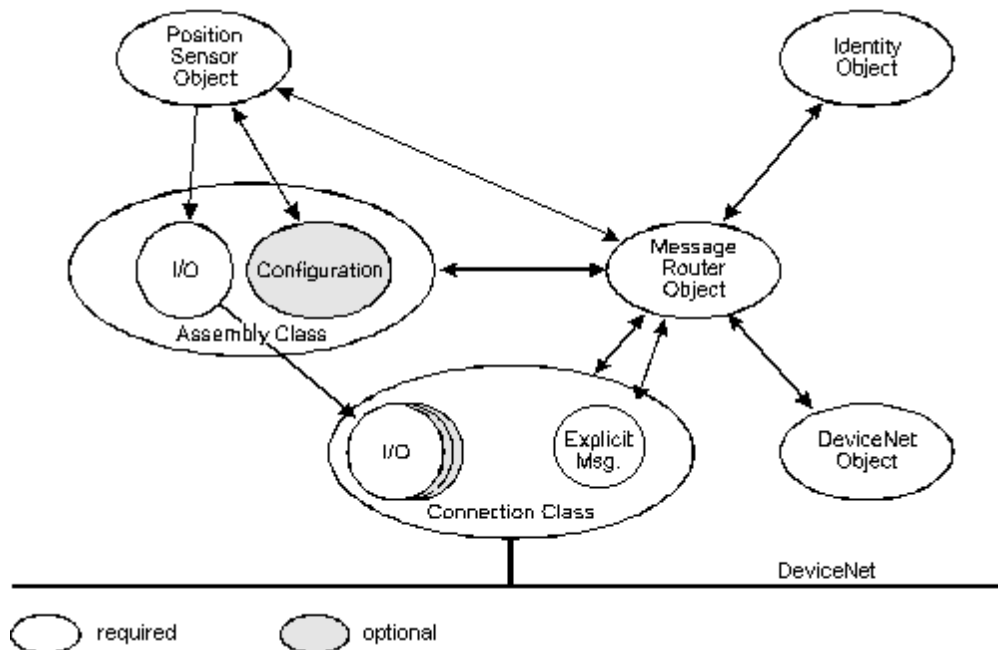


Abb. 8.2 Schnittstellen der Objekte

Wie außerdem aus dem Bild ersichtlich ist, wird von jedem dieser Objekte, außer dem Connection-Objekt, immer nur eine Instanz angelegt. Beim Connection-Objekt wird pro eingerichtete Verbindung, also z.B. Poll-Mode oder Explicit-Message-Connection eine Instanz erzeugt. Den verschiedenen Verbindungen sind dabei die folgenden Instanz ID's zugeordnet.

Connection Instance ID #	Verbindungstyp
1	Explicit Messaging
2	Poll-Mode
3	Bit-Strobe-Mode
4	Change-of-State-Mode

Tabelle 8.2 Instanzen des Connection-Objekts

8.2.1 Identity Object

Class Code 01_{hex}

Class Attributes

Nicht unterstützt

Instance Attributes

Attr. Id	Attribute	Zugriff	DeviceNet Datentyp	Bemerkung
1	Vendor ID	Lesen	UINT	TWK = 407
2	Device Type	Lesen	UINT	TWK = generic
3	Product Code	Lesen	UINT	0x01
4	Revision	Lesen	UINT	1.1
5	Status	Lesen	WORD	
6	Serial Number	Lesen	UDINT	
7	Product Name	Lesen	SHORT-STRING	Encoder CRN/D

Services

Service Code	Service Name	Bemerkung
0E _{hex}	Get_Attribute_Single (lesen)	gibt den Wert eines Attributes zurück
05 _{hex}	Reset	

8.2.2 Message Router Object

Class Code 02_{hex}

Class Attributes

Nicht unterstützt

Instance Attributes

Nicht unterstützt

Services

Nicht unterstützt

8.2.3 DeviceNet Object

 Class Code 03_{hex}

Class Attributes

Attr. Id	Attribute	Zugriff	DeviceNet Datentyp	Bemerkung
1	Revision	Lesen	UINT	Revision = 002

Instance Attributes

Attr. Id	Attribute	Zugriff	DeviceNet Datentyp	Bemerkung
1	MAC ID	Lesen/Schreiben*	USINT	Revision = 002
2	Baud Rate	Lesen/Schreiben*	USINT	Range 0-2
3	BOI	Lesen	BOOL	Value = 0
4	Bus-Off Counter	Lesen/Schreiben	USINT	
5	Allocation Information	Lesen	STRUCT of: BYTE USINT	Allocation Choice Byte Master's MAC ID

* nur bei Steckerversion

Services

Service Code	Service Name	Bemerkung
0E _{hex}	Get_Attribute_Single (lesen)	gibt den Wert eines Attributes zurück
10 _{hex}	Set_Attribute_Single (schreiben)	verändert den Wert eines Attributes
4B _{hex}	Allocate_Master/Slave_Connection_Set	Predefined Master/Slave Connection Set wird angefragt
4C _{hex}	Release_Group_2_Connection_Set	Verbindungen über den Master/Slave Connection Set werden aufgehoben

8.2.4 Assembly Object

 Class Code 04_{hex}

Class Attributes

Nicht unterstützt

Instance Attributes

Attr. Id	Attribute	Zugriff	DeviceNet Datentyp	Bemerkung
3	Data	Lesen/Schreiben	ARRAY	

Services

Service Code	Service Name	Bemerkung
0E _{hex}	Get_Attribute_Single (lesen)	gibt den Wert eines Attributes zurück
10 _{hex}	Set_Attribute_Single (schreiben)	verändert den Wert eines Attributes

8.2.5 Connection Object

 Class Code 05_{hex}
Class Attributes

Nicht unterstützt

Instance Attributes

Attr.Id	Attribute	Zugriff	DeviceNet Datentyp	Bemerkung
1	state	Lesen	USINT	
2	instance_Type	Lesen	USINT	
3	transportClass_trigger	Lesen	BYTE	
4	produced_connection_id	Lesen	UINT	
5	consumed_connection_id	Lesen	UINT	
6	initial_comm_characteristics	Lesen	BYTE	
7	produced_connection_size	Lesen	UINT	
8	consumed_connection_size	Lesen	UINT	
9	expected_packet_rate	Lesen/ Schreiben	UINT	Durch Setzen dieses Wertes wird die Verbindung in den Zustand "Established" gesetzt und alle anderen Parameter übernommen
12	watchdog_timeout_action	Lesen	USINT	
13	produced_connection_path_length	Lesen	UINT	
14	produced_connection_path	Lesen	Array of UINT	
15	consumed_connection_path_length	Lesen	UINT	
16	consumed_connection_path	Lesen	Array of UINT	
17	production_inhibit_time	Lesen/ Schreiben	UINT	

Services

Service Code	Service Name	Bemerkung
0E _{hex}	Get_Attribute_Single (lesen)	gibt den Wert eines Attributes zurück
10 _{hex}	Set_Attribute_Single (schreiben)	verändert den Wert eines Attributes

8.2.6 Position Sensor Objekt

 Class Code 23_{hex}

Class Attributes

Attr. Id	Attribute	Zugriff	DeviceNet Datentyp	Bemerkung
1	Revision	lesen	UINT	
2	Max. Instance	lesen	UINT	

Instance Attributes

ODVA spezifischer Teil:

Attr.Id	Attribute	Zugriff	Datentyp	Bemerkung
1	# of Attributes	lesen	USINT	Anzahl der Attribute einer Instanz
2	Attributes	lesen	Array of/ USINT	Alle in einer Instanz vorhandenen Attribute ID's
3	Value	lesen	UDINT	aktuelle Position des Gebers
11	Value Direction Control	lesen/ schreiben	BOOL	0 = cw, 1 = ccw
39	Status Byte	lesen	USINT	Bit 0: 1=hardware error Bit 1: 1=memory error Bit 2: 1=communication error Bit 3: 1=device specific error

Herstellerspezifischer Teil:

112	Scaling	lesen/ schreiben	BOOL	0=scaling disable;1=scaling enable
113	Measuring units per revolution	lesen/ schreiben	UDINT	max. 8192 steps/U
114	Total measuring range in measuring units	lesen/ schreiben	UDINT	max. 3554432 steps
115	Preset Value	lesen/ schreiben	UDINT	max. Gesamtschrittzahl - 1
128	Area state register	lesen	USINT	Bit 1: 0=OK; 1=Max. Limit überschritten Bit 2: 0=OK; 1=Min. Limit unterschritten
129	Work area low limit	lesen/ schreiben	UDINT	default = 1048575 steps (für 25 Bit Auflösung)
130	Work area high limit	lesen/ schreiben	UDINT	default = 32505856 steps (für 25 Bit Auflösung)
144	Operating status	lesen	WORD	Bit 0: 0=cw; 1=ccw Bit 2: 0=scaling disable;1=scaling enable
145	Single Turn Resolution	lesen	UINT	8192 steps/U (13 Bit)
146	Number of distinguishable revolutions	lesen	UINT	4096 U (12 Bit)
147	Alarms	lesen	WORD	Bit 12: 1 = Eeprom error Bit 13: 1 = CRC error Bit 14: 1 = XRAM error
148	Supported alarms	lesen	WORD	Alle Alarme werden unterstützt
149	Warnings	lesen	WORD	Bit 5: 1=Istwert ungleich Referenzwert
150	Supported warnings	lesen	WORD	Alle Warnungen werden unterstützt
151	Profile and software version	lesen	DWORD	Wort 0 = Profilversion Wort 1 = Softwareversion
153	Offset Value	lesen	UDINT	nicht unterstützt

Services

Service Code	Service Name	Bemerkung
0E _{hex}	Get_Attribute_Single (lesen)	gibt den Wert eines Attributes zurück
10 _{hex}	Set_Attribute_Single (schreiben)	verändert den Wert eines Attributes
15 _{hex}	Restore	beschreibt das EEPROM mit Default-Werten
16 _{hex}	Save	schreibt non-volatile Attribute ins EEPROM

8.3 DeviceNet Verbindungen

DeviceNet ist ein verbindungs-basierendes Netzwerk. Jegliche Kommunikation findet über Verbindungen (Kanäle) statt. Bevor eine Kommunikation stattfinden kann, muß eine Verbindung eingerichtet werden. Dabei werden die entsprechenden Ressourcen bereitgestellt und die Identifier festgelegt, die diese Verbindung verwendet.

Das Identifier-Feld besteht wie bei CAN aus 11 Bits. Bei DeviceNet wird dieser verfügbare Adressraum auf die 4 Bereiche Group1, Group2, Group3 und Group4 wie folgt aufgeteilt:

IDENTIFIER BITS											HEX RANGE	IDENTITY USAGE	
10	9	8	7	6	5	4	3	2	1	0			
0	Group 1 Message ID			Source MAC ID							000 - 3ff	Message Group 1	
1	0	MAC ID					Group 2 Message ID				400 - 5ff	Message Group 2	
1	1	Group 3 Message ID			Source MAC ID							600 - 7bf	Message Group 3
1	1	1	1	1	Group 4 Message ID (0 - 2f)						7c0 - 7ef	Message Group 4	
1	1	1	1	1	1	1	X	X	X	X	7f0 - 7ff	Invalid CAN Identifiers	
10	9	8	7	6	5	4	3	2	1	0			

Tabelle 8.3 DeviceNet Identifier

Im Predefined-Master-Slave-Connection-Set werden nur die Group 1 und Group 2 Messages verwendet. Die Message-ID's sind hier bereits festgelegt und müssen nicht mehr ausgehandelt werden. Die Zuordnung der Message-ID's zu den Verbindungen ist wie folgt:

IDENTIFIER BITS											IDENTITY USAGE	HEX RANGE	
10	9	8	7	6	5	4	3	2	1	0			
0	Group 1 Message ID			Source MAC ID							Group 1 Messages		000 - 3ff
0	1	1	0	1	Source MAC ID						Slave' I/O Change of State or Cyclic Message		
0	1	1	1	0	Source MAC ID						Slave's I/O Bit-Strobe Response Message		
0	1	1	1	1	Source MAC ID						Slave's Poll Response or Change of State/Cyclic Acknowledge		
1	0	MAC ID					Group 2 Message ID				Group 2 Message		400 - 5ff
1	0	Source MAC ID					0	0	0	Master's I/O Bit-Strobe Command Message			
1	0	Source MAC ID					0	0	1	Reserved for Master's Use - Use is TBD			
1	0	Destination MAC ID					0	1	0	Master's Change of State or Cyclic Acknowledge Message			
1	0	Source MAC ID					0	1	1	Slave's Explicit/Unconnected Response Messages			
1	0	Destination MAC ID					1	0	0	Master's Explicit Request Messages			
1	0	Destination MAC ID					1	0	1	Master's I/O Poll Command/Change of State/Cyclic Message			
1	0	Destination MAC ID					1	1	0	Group 2 Only Unconnected Explicit Request Messages (reserved)			
1	0	Destination MAC ID					1	1	1	Duplicate MAC ID Check Message			

Tabelle 8.4 Predefined Master/Slave Connectionset Identifier

8.4 Das DeviceNet-Protokoll

Grundsätzlich unterscheidet DeviceNet zwei Arten von Protokollen:

- **Explicit Messaging**
- **I/O Messaging**

Das Explicit Messaging wird eingesetzt, um gezielt Attribute zu lesen oder zu schreiben. Hierbei wird noch zwischen **unfragmentiertem** und **fragmentiertem** Protokoll unterschieden.

Das I/O-Messaging wird für den schnellen Datenverkehr der I/O-Daten verwendet. Hier stehen im Gegensatz zum Explicit Messaging die vollen acht Datenbyte des CAN-Telegramms dem Anwender zur Verfügung.

Beim Winkelcodierer werden die Parameterdaten und die Statusinformationen über das Explicit Messaging gelesen bzw. geschrieben und die fünf Byte Eingangsdaten über das I/O-Messaging gelesen.

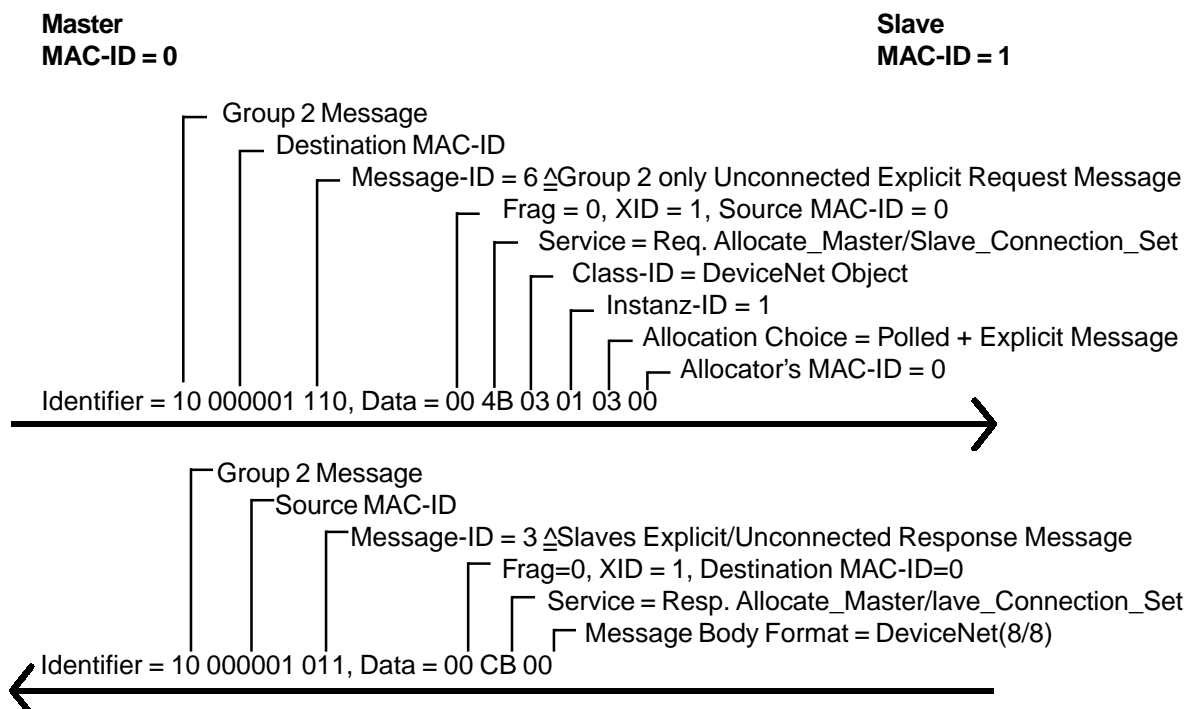
8.5 Aufbau einer Verbindung zum Winkelcodierer

Jede Verbindung zu einem Slave wird über die Group 2 only unconnected Explicit Request Message eingerichtet (Group 2 Message-ID 6). Im Predefined-Master-Slave-Connection-Set wird über den Service **4B_{hex} Allocate_Master/Slave_Connection_Set** des DeviceNet Objekts, eine Instanz des Connection Objekts (siehe [Tabelle 8.2](#)) gebildet. Diesem Service wird in einem **Allocation Choice Byte** die gewünschte Verbindungsart (Betriebsart) mitgegeben.

Allocation Choice Byte

7	6	5	4	3	2	1	0
Reserved	Acknowledge Suppression	Cycle	Change of State	Reserved	Bit Strobed	Polled	Explicit Message

Beispiel: Einrichten der I/O-Verbindung Poll-Mode und des Explicit Messaging



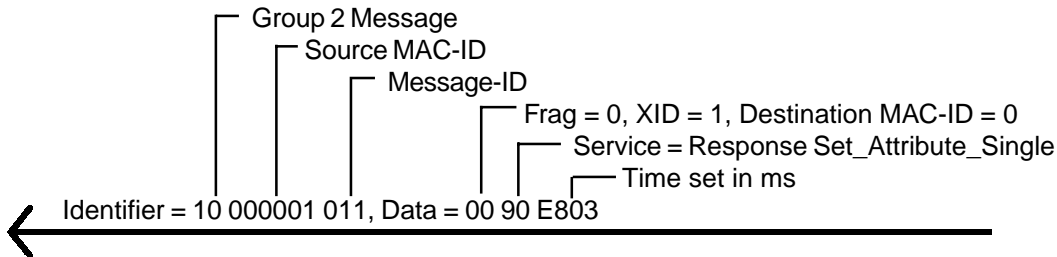
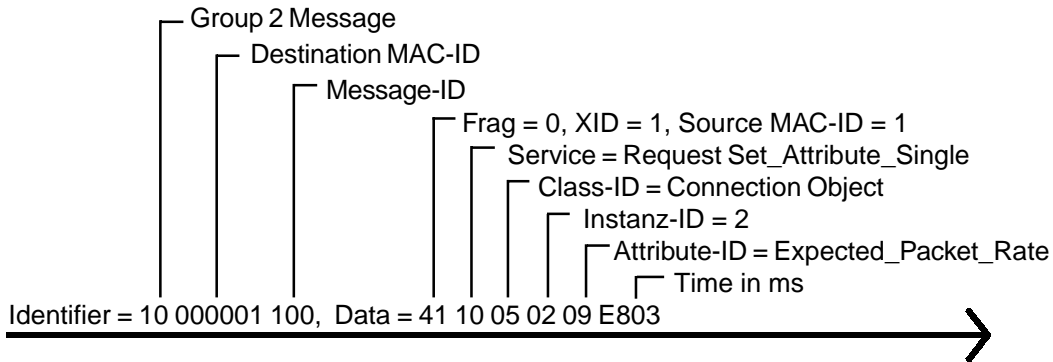
Durch das vorhergehende Beispiel wird eine Instanz mit der ID1 (Explicit Messaging Connection) und ID 2 (I/O-Messaging Poll Mode) des Connection Objektes erzeugt. Diese damit geschaffenen Verbindungen befinden sich danach im Status „Configuring“ (Instanz-Attribute 1 = 1). Um sie in den Status „Established“ (Instanz-Attribute 1 = 3) zu bringen muß die **Expected_Packet_Rate** (Instanz-Attribute 9) gesetzt werden. Nachfolgend wird dies beispielsweise für die Poll-Mode-Instanz gezeigt.

Beispiel: Setzen des Attributes Expected_Packet_Rate

Das Setzen eines Attributes einer Instanz geschieht einheitlich für alle Objekte mit dem Service **10_{hex} Set_Attribute_Single**. Hier wird beispielsweise eine Zeit von 1s gesetzt.

**Master
MAC-ID = 0**

**Slave
MAC-ID = 1**



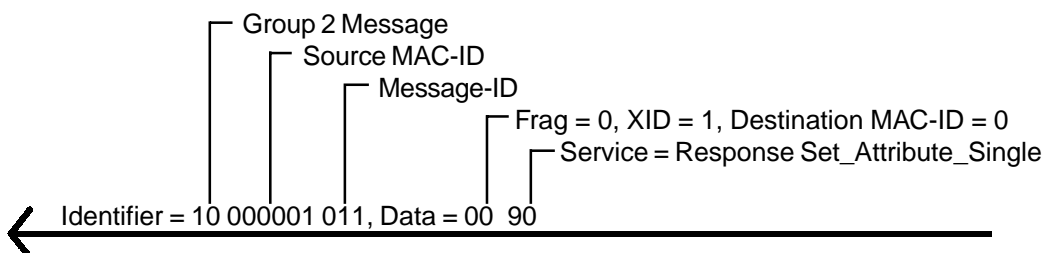
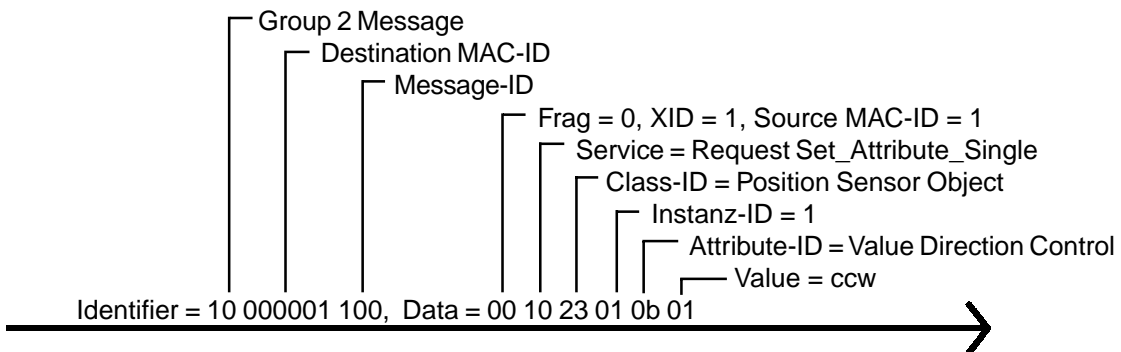
8.6 Parametrieren des Winkelcodierers

Das Parametrieren des Winkelcodierers geschieht wie das Setzen der Expected_Packet_Rate mit einer Explicit Message und dem Service **10_{hex} Set_Attribute_Single** des Position Sensor Object.

Beispiel: Setzen der Coderichtung (Attribute **b_{hex}) auf **01_{hex}****

**Master
MAC-ID = 0**

**Slave
MAC-ID = 1**

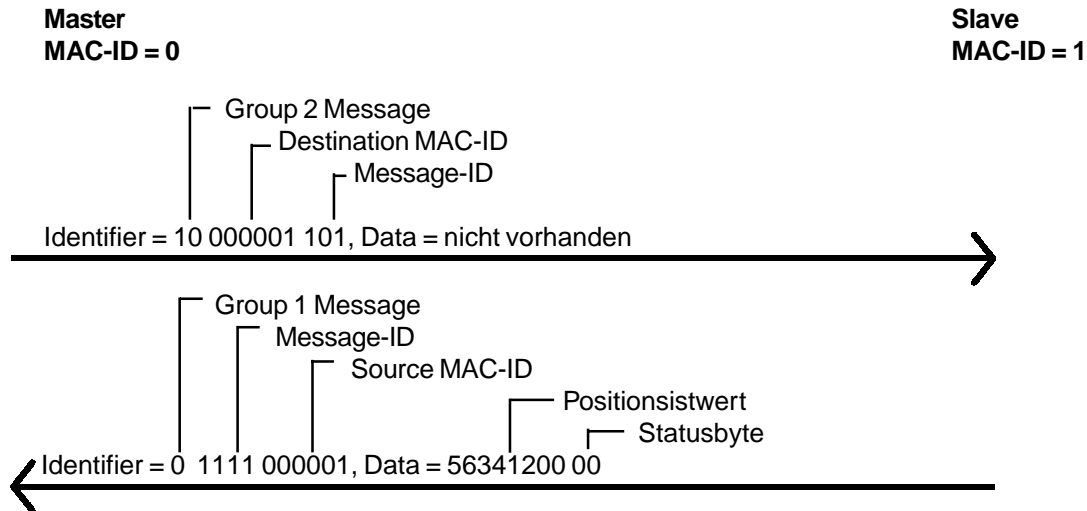


8.7 Eingangsdaten des Winkelcodierers abfragen

Das Abfragen der Istdaten geschieht über das I/O-Messaging. Je nach eingerichteter Verbindung muß hier eine andere Message ID für den Request verwendet werden.

Beim Polling wird jeder Teilnehmer einzeln mit einem Poll-Command angesprochen. Teilnehmer, die Daten ausgeben, erhalten mit dem Poll-Command auch die Ausgangsdaten. Beim Winkelcodierer existieren diese nicht.

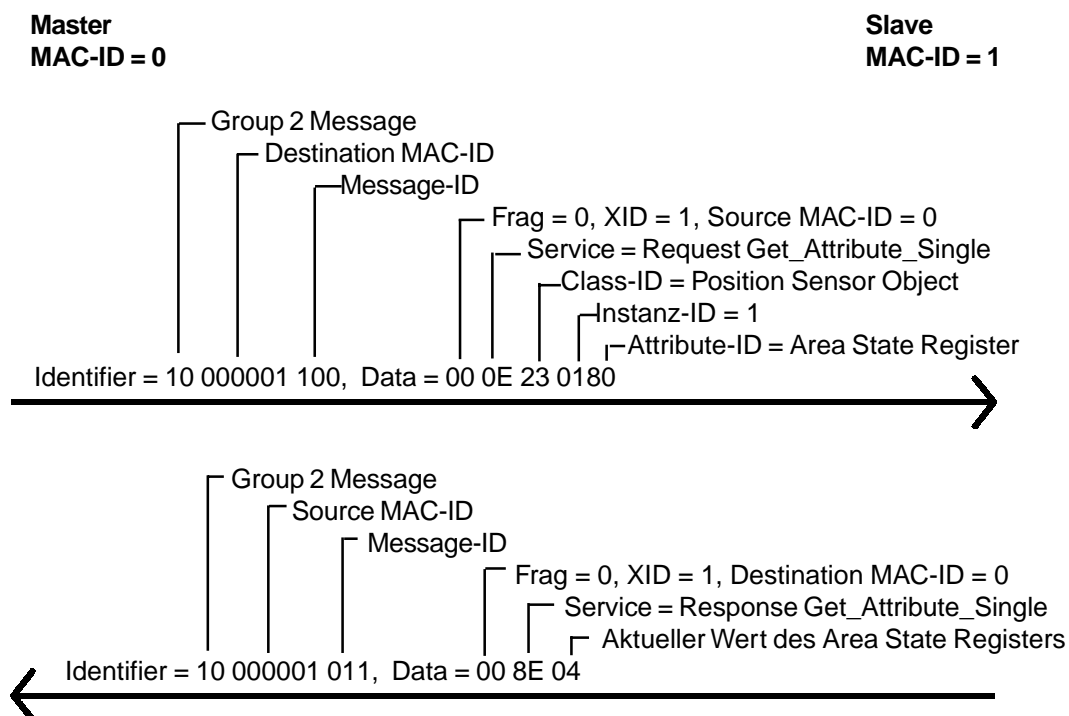
Beispiel: Einlesen der Istwerte im Pollmode



8.8 Statusinformationen abfragen

Die zusätzlich, zu dem Statusbyte der Eingangsdaten vorhandenen Statusinformationen des Winkelcodierers, müssen über das Explicit Messaging ausgelesen werden. Das Lesen der Attribute eines Objektes geschieht bei allen Objekten mit dem Service $0E_{hex}$ **Get_Attribute_Single**.

Beispiel: Auslesen des Attribut 80_{hex} Area State Register



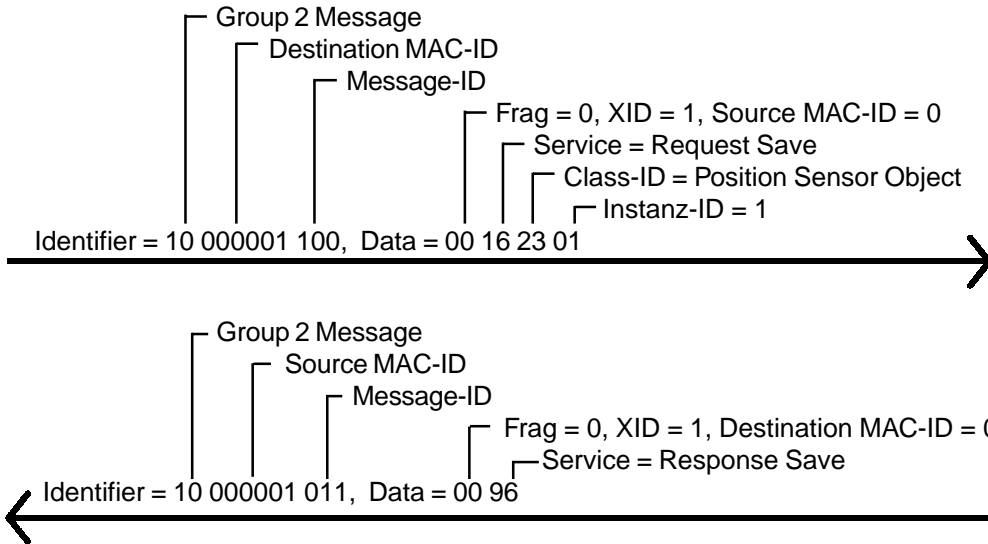
8.9 Parameterwerte im EEPROM sichern

Mit Hilfe des Service 16_{hex} „Save“ des Position-Sensor-Object werden alle Parameterwerte spannungsausfallsicher im EEPROM des Winkelcodierers abgespeichert.

Beispiel:

Master
MAC-ID = 0

Slave
MAC-ID = 1



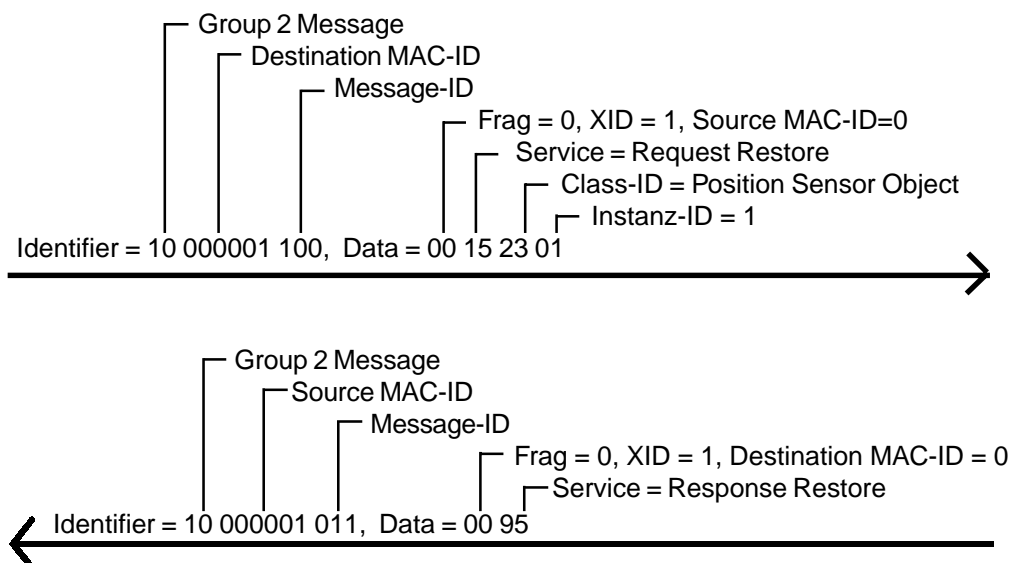
8.10 Parameterwerte in Default-Zustand bringen

Der Service 15_{hex} „Restore“ lädt alle Parameter-Variablen mit ihren Default-Werten und speichert diese anschließend im EEPROM. Die Default-Werte werden erst nach einem Neustart des Winkelcodierers (Spannung aus/ein) wirksam.

Beispiel:

Master
MAC-ID = 0

Slave
MAC-ID = 1



9. DeviceNet-Manager

In diesem Kapitel wird die Einbindung des TWK-Winkelcodierers CRN/D in das DeviceNet-Master-Slave-System einer Allen-Bradley-SPS anhand des DeviceNet-Managers von Allen-Bradley (Rockwell) beschrieben. Die genaue Kenntnis der Objekte wie sie in [Kapitel 8](#) beschrieben sind ist hierzu nicht erforderlich.

Die Projektierung des DeviceNet-Systems mit dem DeviceNet-Manager [2] ist auf mehreren Wegen möglich. Prinzipiell wird hier unterschieden zwischen **Offline- und Online-Konfiguration**.

Bei der **Offline-Konfiguration** können ohne bestehende Verbindung zum Bus dessen Aufbau projektiert und die Teilnehmer parametrieren werden. Anschließend wird die Verbindung hergestellt und die Anwenderdaten in den Master und die Teilnehmer übertragen.

Bei der **Online-Konfiguration** wird der Busaufbau über eine bestehende Verbindung eingelesen und die Teilnehmer online parametrieren.

Im folgenden wird die Vorgehensweise bei der Offline-Konfiguration beschrieben.

9.1 Installation EDS-Datei

Aufgrund eines Fehlers im DeviceNet-Manager befinden sich auf der Diskette zwei EDS-Dateien:

- 1.EDS: Standard EDS-Datei nach Spezifikation (Benutzen Sie diese EDS-Datei für Programme wie RS-Networx und andere)
- DNetMan.EDS: Benutzen Sie diese EDS-Datei nur für den DeviceNet-Manager

Wählen Sie zur Installation der EDS-Datei im Hauptmenü unter *Utilities* den Menüpunkt *Install EDS Files...* In dem folgenden Fenster wählen Sie die Datei **DNetMan.eds** aus.

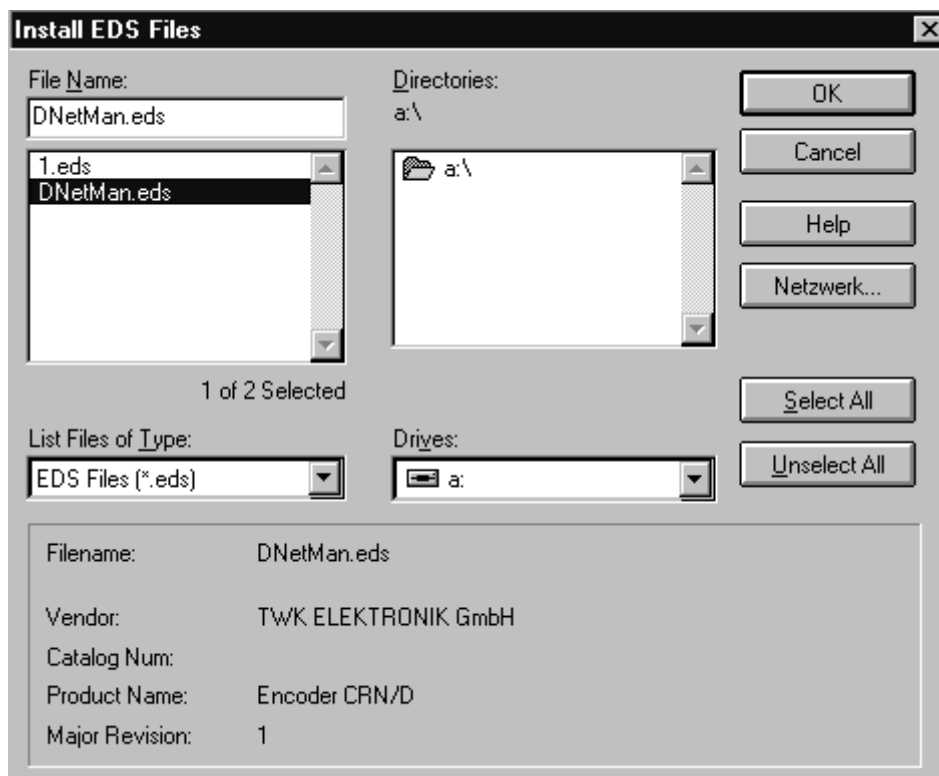


Abb. 9.1 Installation EDS-Datei

Anschließend werden Sie gefragt, ob Sie dem neuen Gerät ein Bitmap zuweisen wollen. Antworten Sie hier bitte mit ja. Das folgende Bild zeigt den Dialog, in dem Sie nun das auf Diskette mitgelieferte Bitmap **crn_d_m** (Steckerversion) oder **crn_d_z** (Haubenversion) auswählen können.

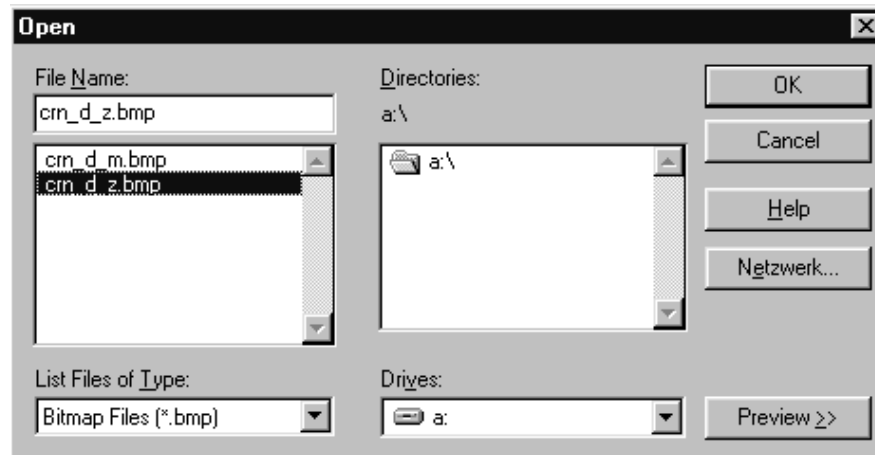


Abb. 9.2 Installation Winkelcodierer Symbol

Die Installation der EDS-Datei und des Teilnehmersymbols ist nun abgeschlossen.

9.2 Einbinden in den Bus

Nachdem Sie ein Projekt angelegt haben, können Sie jetzt den Winkelcodierer *Encoder CRND* aus der *Device List* unter *Generic, TWK Elektronik GmbH* auswählen und per Drag & Drop in den Bus einbinden.

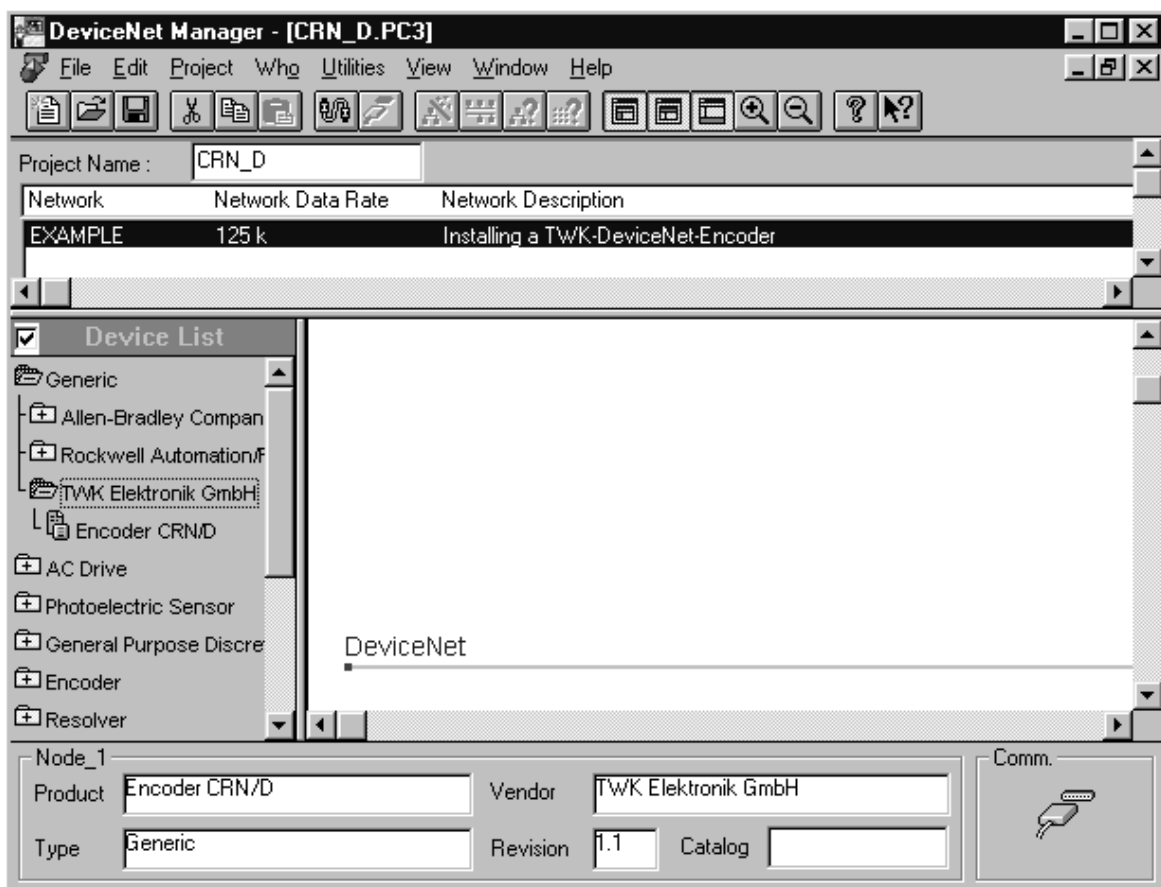


Abb. 9.3 Winkelcodierer Einbindung

Anschließend werden Sie zur Eingabe eines Teilnehmersnamen und der Knotenadresse aufgefordert. Als Knotenadresse geben Sie bitte die eingestellte Adresse des Winkelcodierers an (bei Anschlußhaube per Dip-Schalter einstellbar, bei Stecker-version Softwaremäßig (Siehe [Kapitel 9.6](#))).

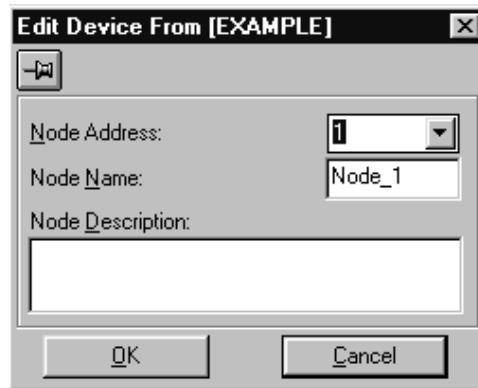


Abb. 9.4 Knotenadresse vergeben

Hier wurde die Winkelcodierer-Default-Adresse 1 beibehalten und als Node Name wurde der vorgeschlagene Name „Node_1“ akzeptiert. Nach Übernahme mit OK erscheint der Winkelcodierer im Bus.

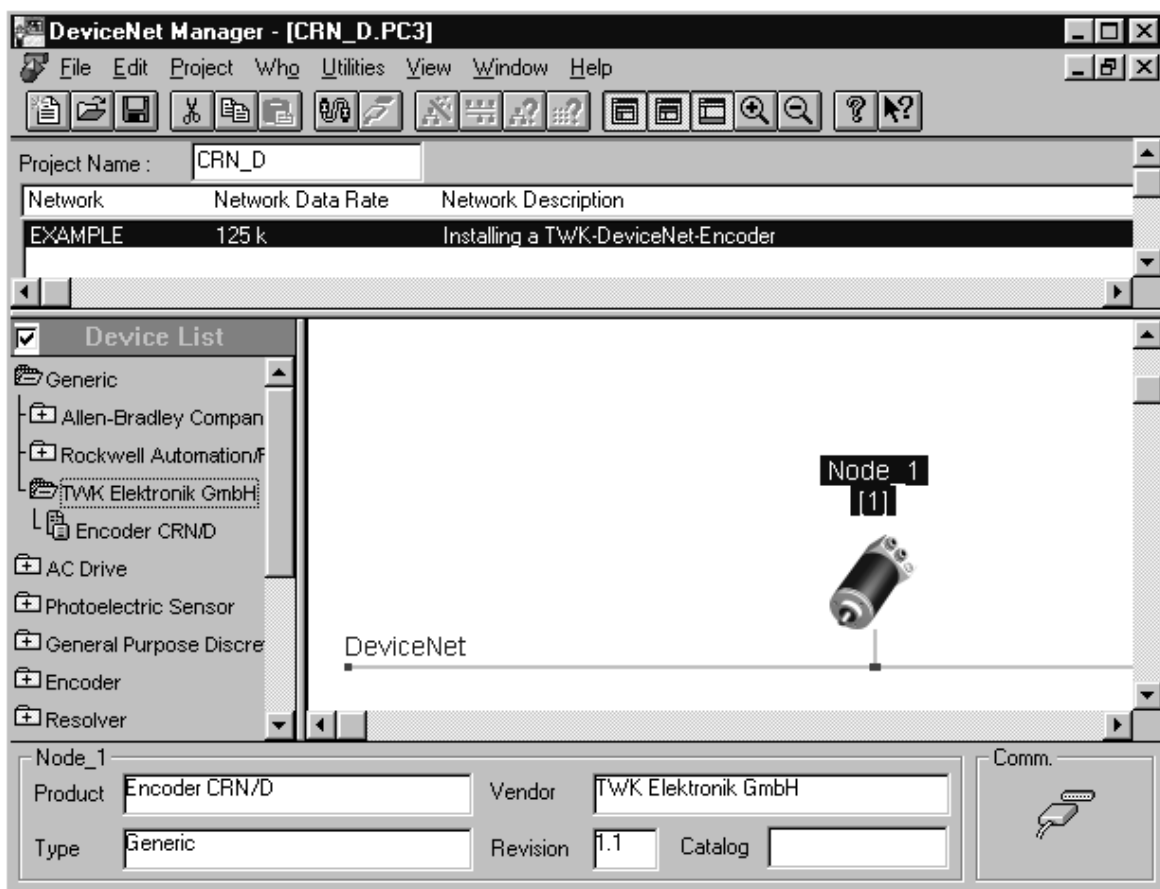


Abb. 9.5 Winkelcodierer im DeviceNet

9.3 Winkelcodierer parametrieren

Zum Ändern der Parameter eines Teilnehmers im DeviceNet stehen im DeviceNet Manager zwei verschiedene Möglichkeiten zur Verfügung:

- **Basic Device Configuration**
- **Enhanced Device Configuration**

Die **Basic Device Configuration** steht nur im Online-Betrieb zur Verfügung. Hierbei müssen die Lage des Parameters (Attributes) im Objektmodell der DeviceNet-Software und die Funktion (Service), mit der man auf diesen Parameter zugreifen kann, bekannt sein.

Die **Enhance Device Configuration** stellt eine komfortable Möglichkeit dar, die Parameter des Winkelcodierers zu verändern und auszulesen. Dies ist sowohl online als auch offline (nur in Datei) möglich. Die Parameter lassen sich im Enhanced Mode in einer Datei (*.dcf) abspeichern, um sie dann zu einem späteren Zeitpunkt an den Teilnehmer übertragen zu können.

Zum spannungsausfallsicheren Abspeichern der Parameter im EEPROM des Winkelcodierers steht ein **Save**-Kommando zur Verfügung. Außerdem lassen sich die Defaultwerte aller Parameter mit dem **Restore**-Befehl wieder zurückladen.

Die Kommandos Save und Restore können nur über den Basic Device Configurator ausgeführt werden. (Kapitel 9.7 bzw. 9.8)

Rufen Sie nun den Enhanced Configurator durch Doppelklick auf das Winkelcodierersymbol auf. Im Offline-Betrieb werden die Parameter mit den Defaultwerten der EDS-Datei vorbelegt. Sie erhalten folgendes Fenster:

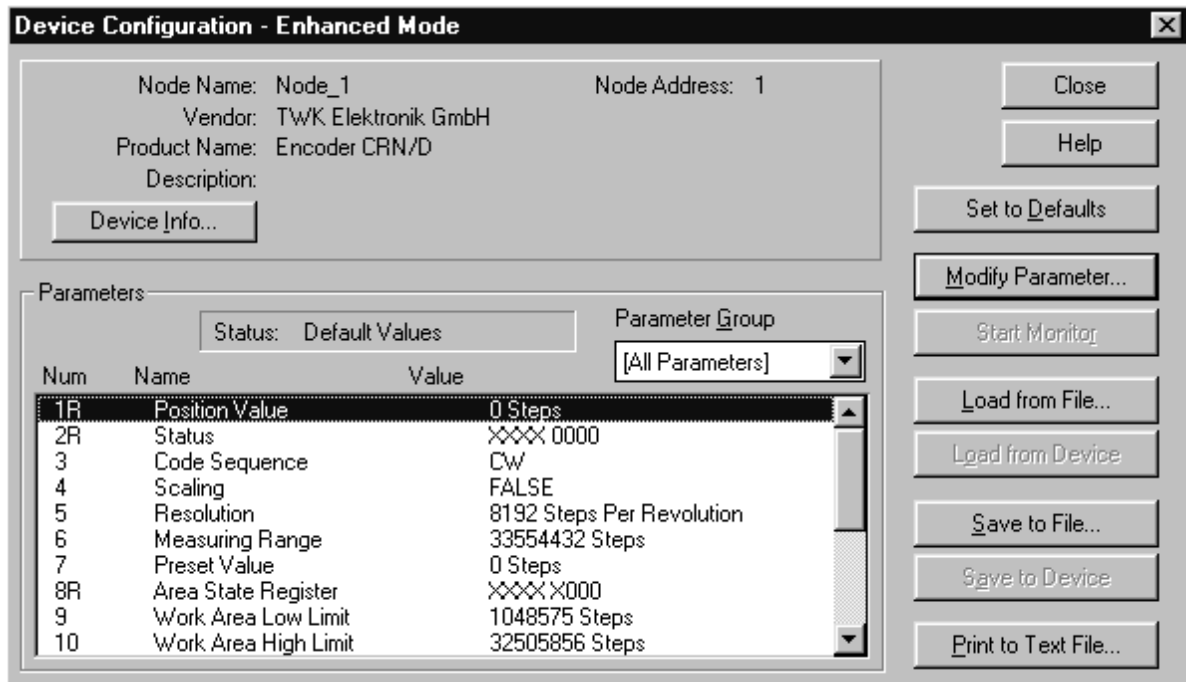


Abb. 9.6 Winkelcodierer Parameter

Die nach der Nummer mit einem „R“ gekennzeichneten Werte stellen die Statusinformationen des Winkelcodierers dar, sie können nur gelesen werden. Alle anderen Werte sind editierbare Parameter. Durch Doppelklick auf den Parameter oder über den Button *Modify Parameter* lassen sich die Werte ändern. Je nachdem, ob es sich um einen numerischen oder einen booleschen Parameter handelt, erhält man eines der folgenden Fenster:

Boolescher Parameter:

Wollen Sie die Auflösung oder Gesamtschrittzahl des Codierers verändern, so müssen Sie den Parameter #4 „Scaling“ auf „ON“ ändern.

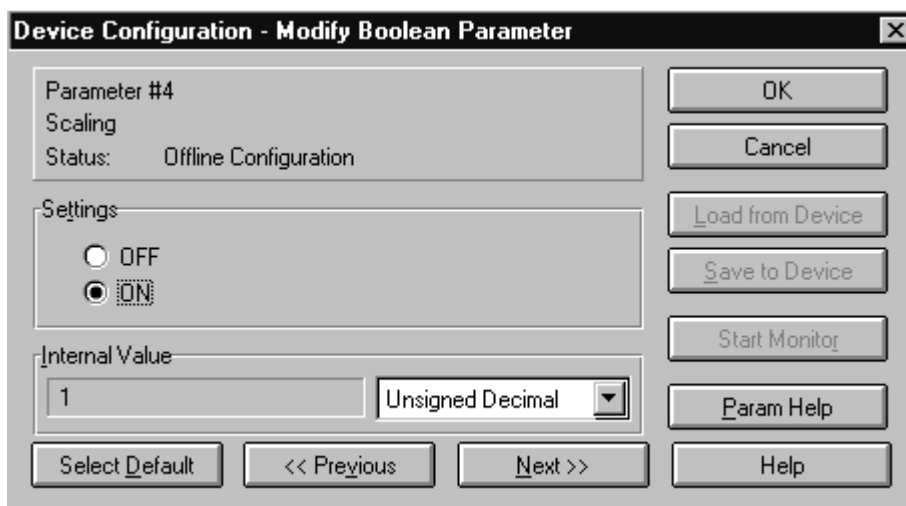


Abb. 9.7 Booleschen Parameter editieren

Numerischer Parameter:

Das Ändern numerischer Werte geschieht über den folgenden Dialog. Das Ändern des Wertes ist entweder über numerische Eingabe oder über den Schieberegler möglich. Hier wird beispielsweise eine Auflösung von 4096 Schritten/Umdrehung eingestellt.

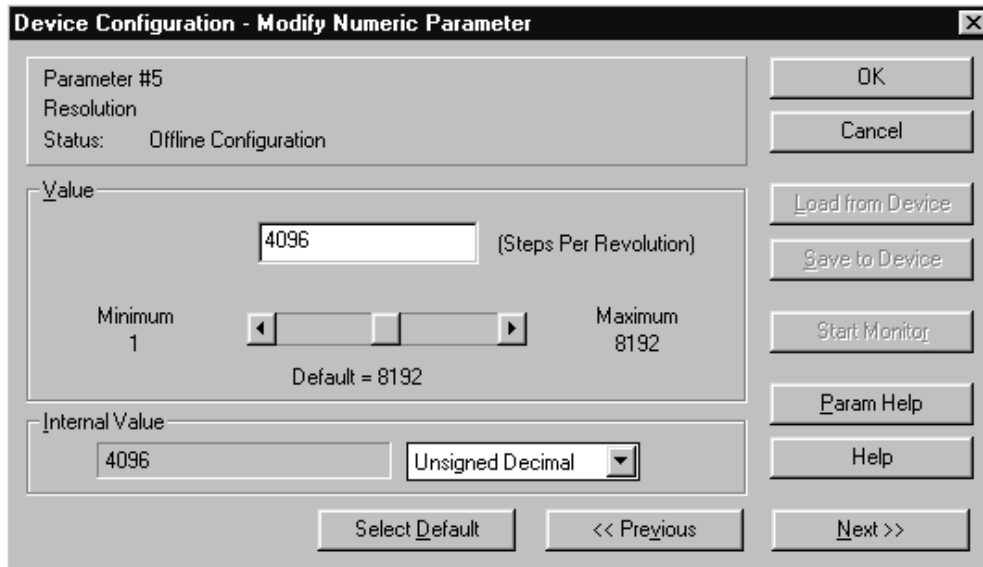


Abb. 9.8 Numerischen Parameter editieren

Nach Einstellung aller Parameter können diese über den Button *Save to File...* des Enhanced Configurators in einer Datei abgespeichert werden. Für jeden Teilnehmer wird dabei eine separate Datei angelegt (Hier Node_1.dcf).

9.4 Winkelcodierer in die Scanlist aufnehmen

Um die Betriebsart des Winkelcodierers festzulegen und ihn einem Master zuzuordnen, muss dieser in die Scanlist eines Masters aufgenommen werden.

Das folgende Bild zeigt einen Busaufbau mit einer Allen-Bradley-SPS SLC500 mit Master 1747-SDN Scanner Modul. Alle folgenden Bilder beziehen sich auf diese Konfiguration.

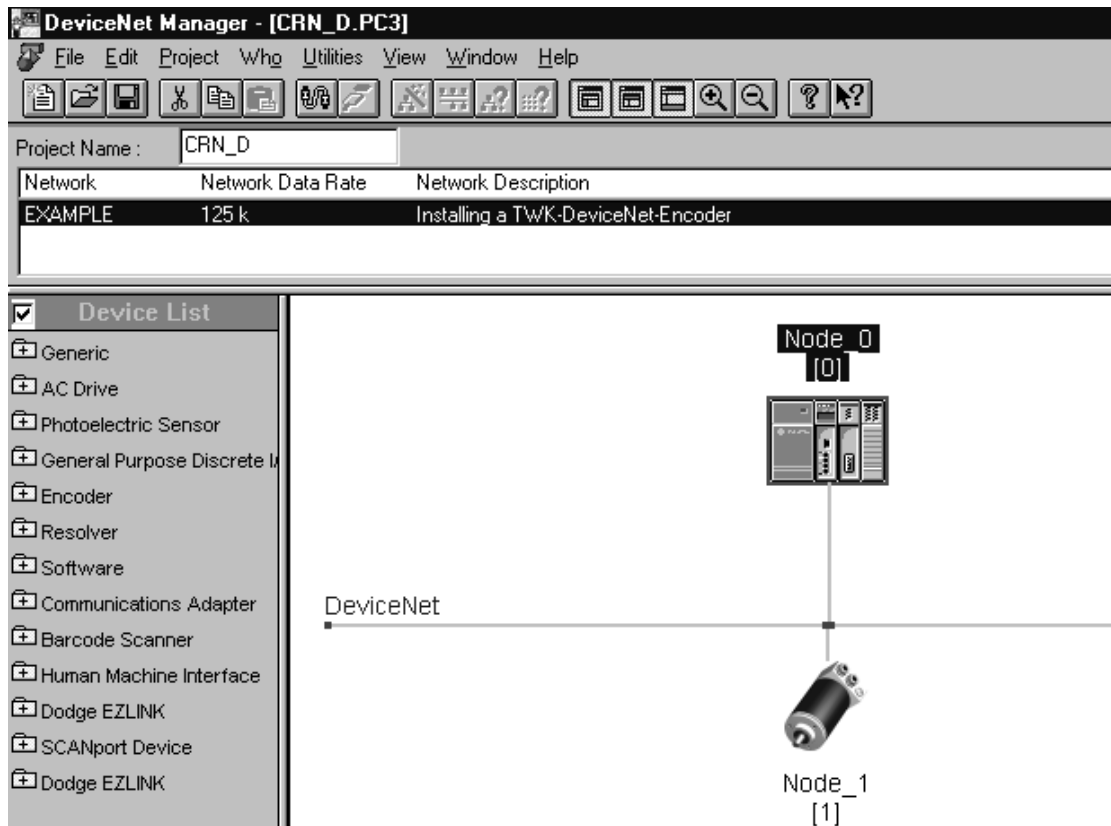


Abb. 9.9 Busaufbau mit Master

Über einen Doppelklick auf das SPS-Symbol gelangt man in den folgenden Dialog zur Einstellung der Masterparameter. Für die Beispielanwendung können die Einstellungen übernommen und in einer Datei (hier: Node_0.sm4) abgespeichert werden.

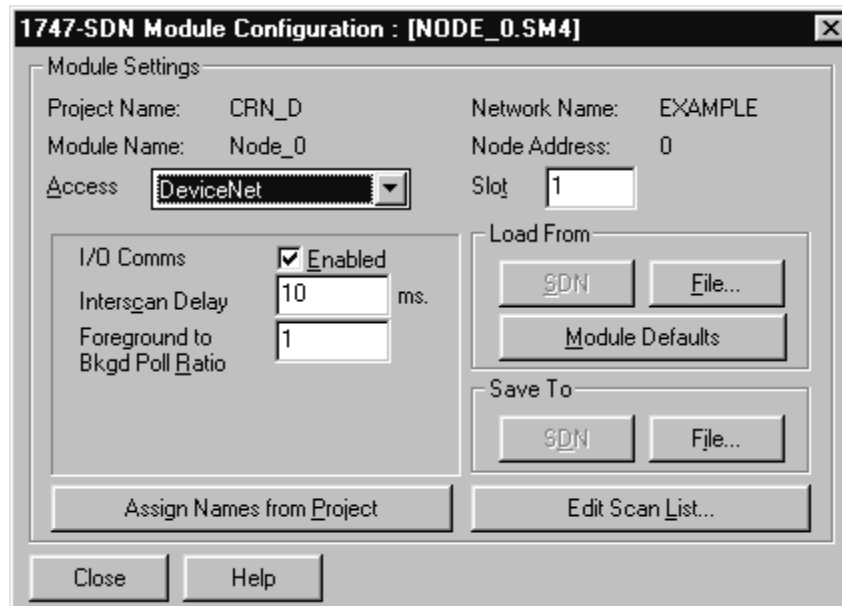


Abb. 9.10 Master parametrieren

Durch Betätigen des Button *Edit Scan List...* erscheint der **Scan List Editor**. Da noch kein Teilnehmer der Scanliste hinzugefügt wurde, zeigt sich ein leeres Fenster.

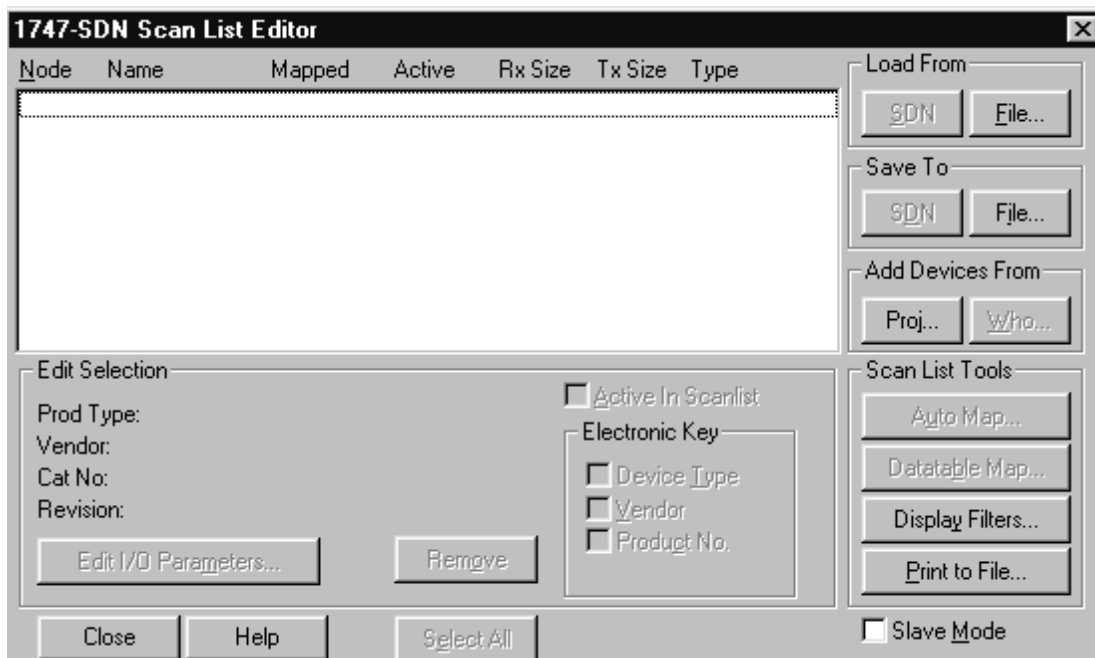


Abb. 9.11 Scanliste des Masters

Zum Hinzufügen von Teilnehmern zur Scan List betätigen Sie nun im Gruppenfeld *Add Devices From* den Button *Proj...*. Dadurch wird erneut der Busaufbau eingeblendet:

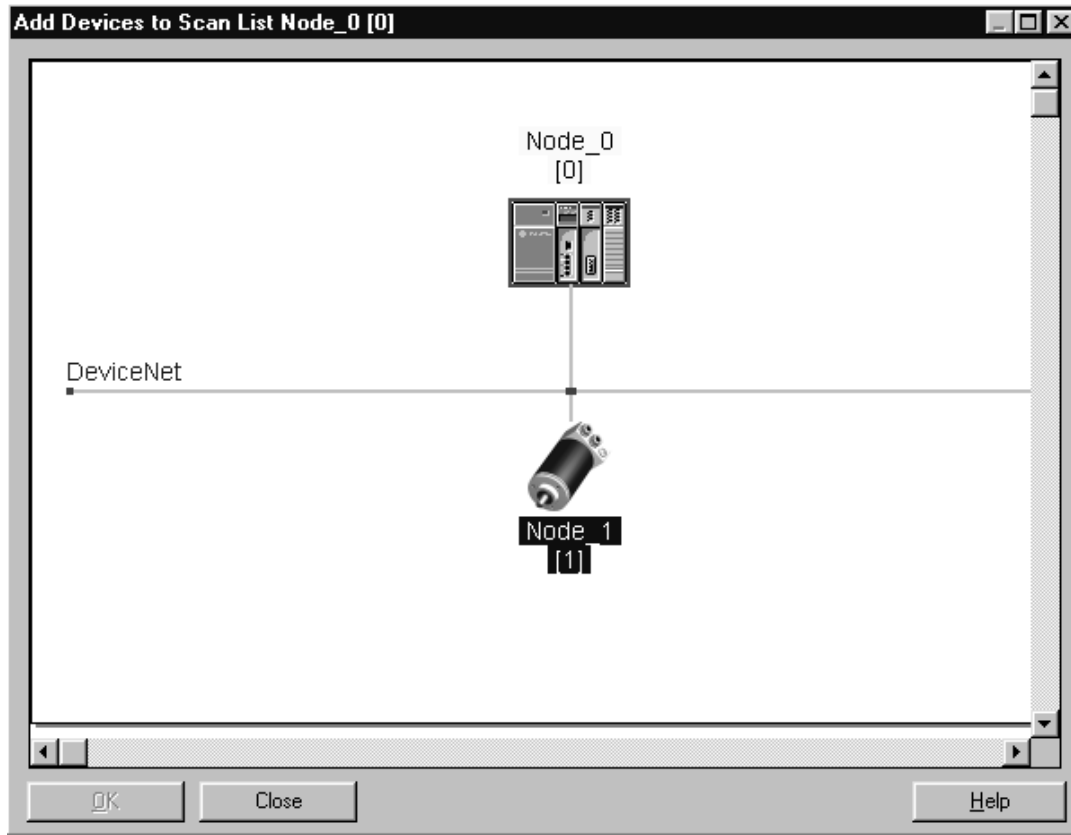


Abb. 9.12 Winkelcodierer in Scanliste aufnehmen

Ziehen Sie nun den Codierer per Drag & Drop auf das Mastersymbol. Der Codierer erhält einen roten Rahmen und die Zahl 0. Dies zeigt an, daß der Teilnehmer der Scanliste des Masters mit ID 0 zugewiesen ist. Anschließend können Sie das Fenster mit OK verlassen.

Nun erscheint der Codierer in seiner Default-Betriebsart, dem Poll-Mode, im Scan List Editor.

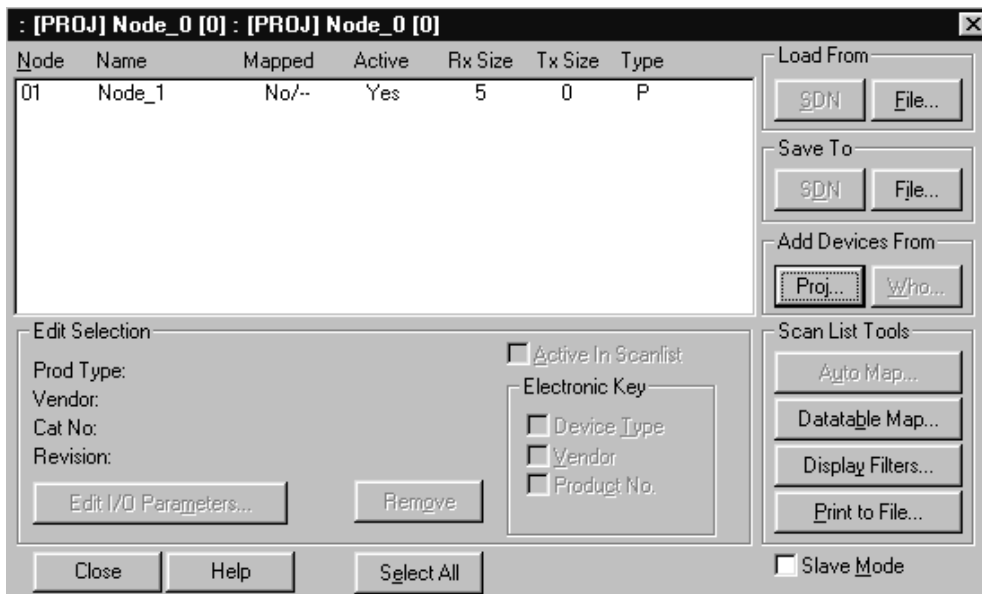


Abb. 9.13 Scanliste mit Winkelcodierer

Möchten Sie die Betriebsart des Codierers ändern, führen Sie einen Doppelklick auf der Zeile 1 aus. Daraufhin gelangen Sie in das folgende Fenster, in dem Sie zwischen den Betriebsarten Poll-Mode, Bit-Strobe-Mode und Change of State / Cyclic-Mode wählen können. Es können auch mehrere Betriebsarten gleichzeitig angewählt werden. Die Anzahl der Eingangsdaten (Rx) muß immer fünf Byte und der Ausgangsdaten null Byte betragen.

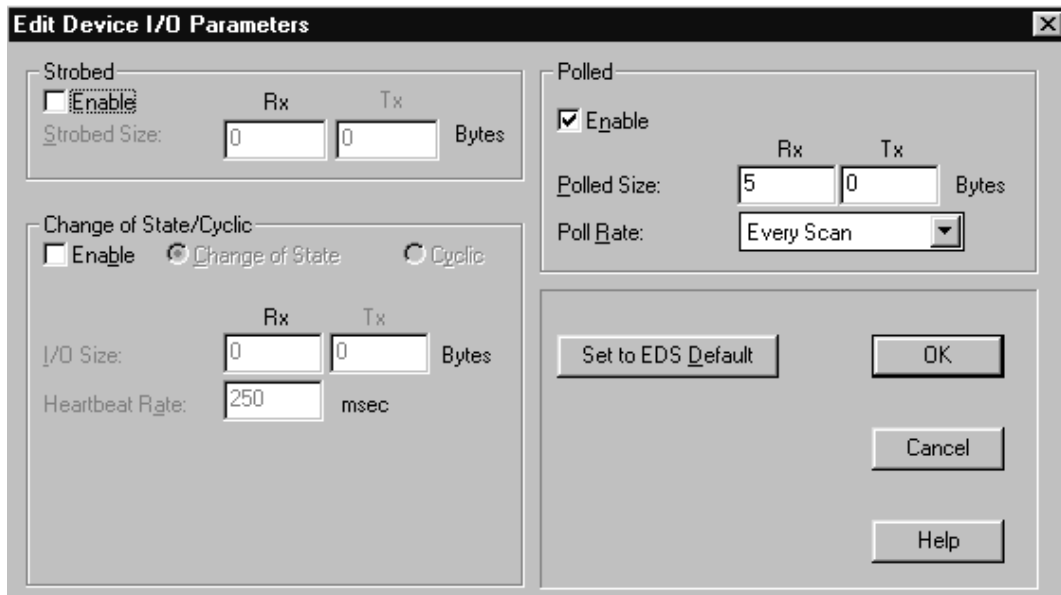


Abb. 9.14 Betriebsart des Winkelcodierers festlegen

Der Winkelcodierer ist nun mit der gewünschten Betriebsart in der Scan List des Masters aufgenommen. Damit Sie in der SPS auf die Daten zugreifen können müssen Sie dem Scanner noch mitteilen, wo die Daten des Codierers im Eingangsbereich der SPS zu finden sein sollen. Hierfür existiert die **Datatable Map**. Sie können diese entweder von Hand editieren, oder ein automatisches Mapping vornehmen. Für die Beispielanwendung reicht es, dieses automatisch auszuführen und die Standardvorgaben zu akzeptieren.

Speichern Sie nun noch die gemachten Einstellungen in einer Datei (hier: Node_0.sl4) ab.

9.5 Online gehen und Daten übertragen

Nachdem nun die Winkelcodiererparameter, die Mastereinstellungen und die Scanliste in einer Datei abgespeichert sind, werden diese im Folgenden an die Teilnehmer übertragen.

Scanliste an den Master übertragen: Stellen Sie die Online-Verbindung zu Ihrem Netzwerk her, wechseln Sie in den Scan List Editor und laden Sie die in [Kapitel 9.4](#) angelegte Scanlist. Sie sollten nun das folgende Bild sehen:

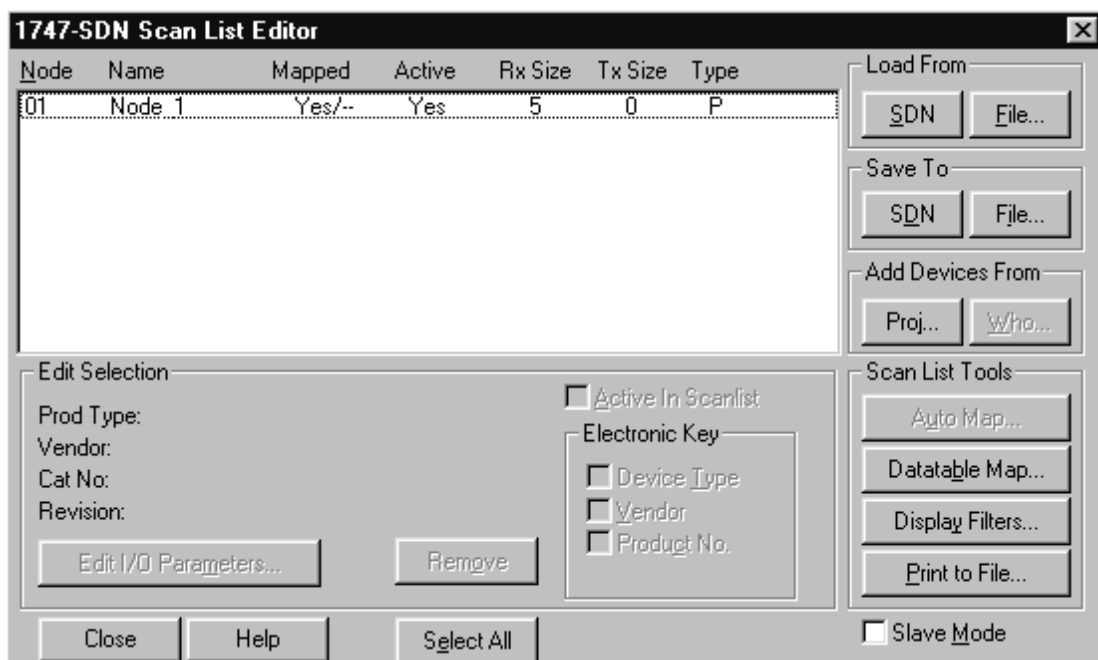


Abb. 9.15 Scanliste an den Master übertragen

Übertragen Sie die Scan List nun mit *Save to SDN* an den Scanner (Master). Geben Sie bei einem neuen Projekt im folgenden Auswahlfenster *All Records* an.

Parameter an den Winkelcodierer übertragen:

Öffnen Sie durch Doppelklick auf das Winkelcodierersymbol den Device Configurator - Enhanced Mode. Da eine Online-Verbindung besteht, werden Ihnen nun die aktuell im Codierer vorhandenen Parameter ausgelesen und angezeigt.

Laden Sie Ihre unter [Kapitel 9.3](#) abgespeicherten Parameterwerte aus der Datei *Node_1.dcf*. Sie erhalten beispielsweise folgendes Bild:

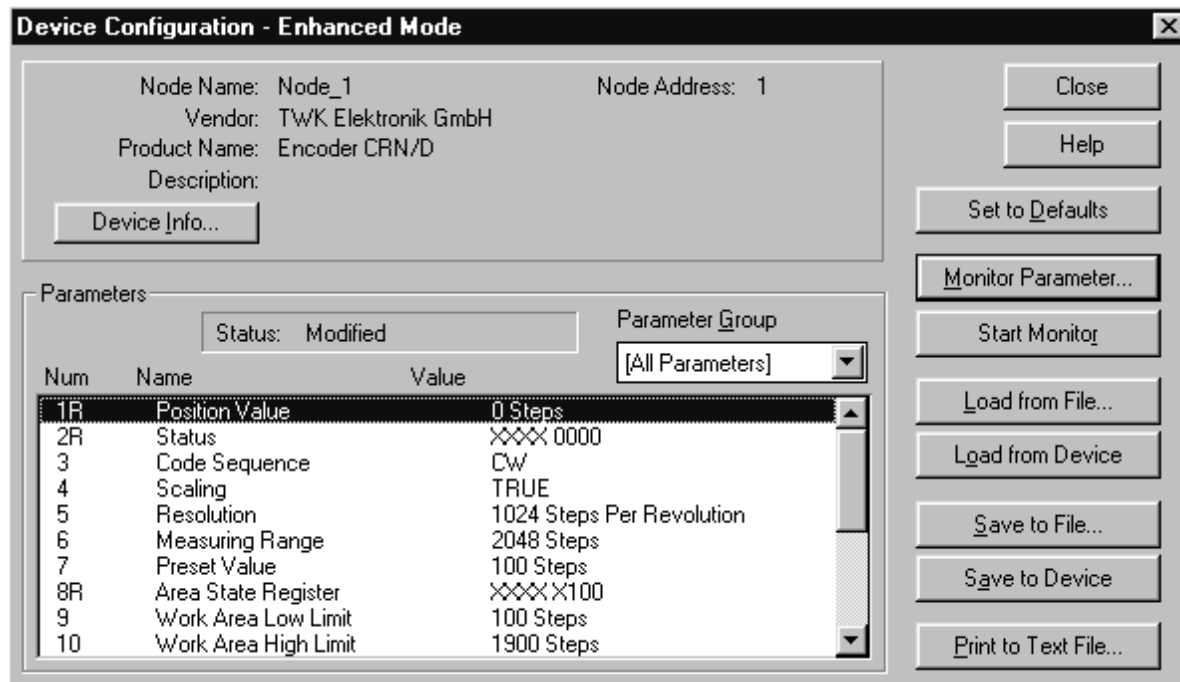


Abb. 9.16 Parameter an den Winkelcodierer übertragen

Übertragen Sie dann Ihre Parameter mit dem Button *Save to Device* an den Winkelcodierer. Die neuen Werte werden im RAM des Codierers abgelegt und erhalten sofort Gültigkeit. Damit diese auch nach einem Spannungsausfall noch im Codierer wirksam sind, können Sie mit einem eigenen Kommando spannungsausfallsicher im EEPROM des Codierers gespeichert werden. (Siehe [Kapitel 9.7](#))

9.6 Adresse und Baudrate beim Winkelcodierer in Steckerversion einstellen

Die Steckerversion des CRN/D erlaubt die Einstellung der Teilnehmeradresse (MAC-ID) und der Baudrate (125 kB, 250KB, 500KB) per Software.

Die Einstellung der Baudrate sollte immer in einer Punkt-zu-Punkt-Verbindung zum Teilnehmer geschehen, da die Änderung der Baudrate eines Teilnehmers im Netz den kompletten Busbetrieb zum Erliegen bringen kann. Die Teilnehmeradresse hingegen kann auch im Busbetrieb geändert werden. Zu beachten ist jedoch, daß keine Adresse eines bereits bestehenden Teilnehmers verwendet wird.

Wirksam wird eine Änderung der Baudrate erst nach einem erneuten Spannung aus/ein, die Änderung der Adresse erfolgt direkt.

Bauen Sie nun eine Punkt-zu-Punkt-Verbindung mit dem Winkelcodierer auf und gehen Sie mit dem DeviceNet Manager online. Schließen Sie ein eventuell geöffnetes Projekt und wählen Sie unter *Utilities* den Punkt *Node Commissioning*.

Ändern der Teilnehmeradresse:

Geben Sie im linken Feld *Node Address* die momentane Adresse des Winkelcodierers an (Default Adresse ist 1) und im rechten Feld die gewünschte neue Adresse. Nach Betätigen von *Apply Node Settings* wird die neue Adresse im Codierer gültig.

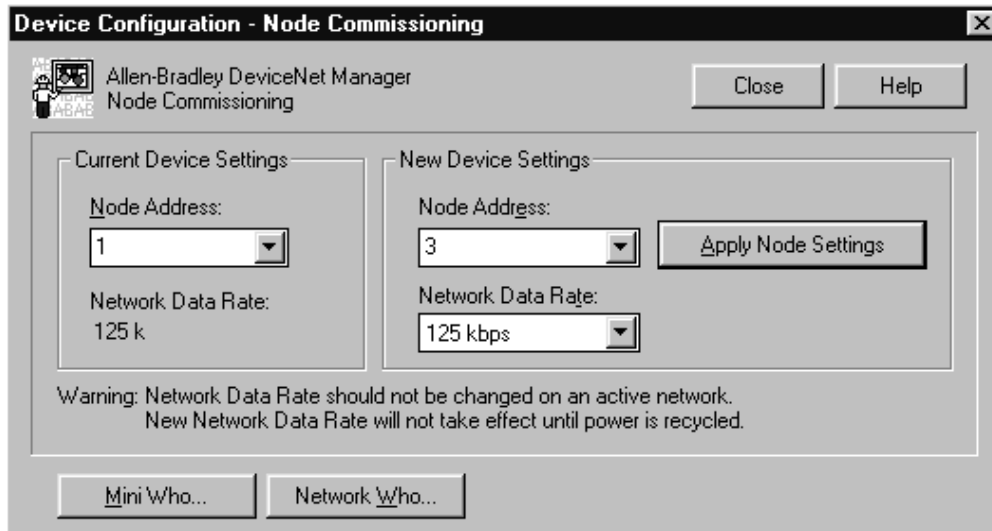


Abb. 9.17 Knotenadresse einstellen

Ändern der Baudrate:

Im Feld *Current Device Settings* wird die aktuelle Baudrate angezeigt (hier 125 KB). Geben Sie unter *New Device Settings* die gewünschte neue Baudrate an. Nach dem Betätigen von *Apply Node Settings* erfolgt nochmal eine Warnung, diese Einstellung nur in einer Punkt-zu-Punkt-Verbindung zu verändern. Nach der Bestätigung wird die neue Baudrate im Codierer gespeichert, jedoch erst nach Spannung aus/ein wirksam.

Beide Änderungen sind natürlich auch gleichzeitig möglich. Ein zusätzliches Abspeichern der Teilnehmeradresse und der Baudrate ist nicht erforderlich.

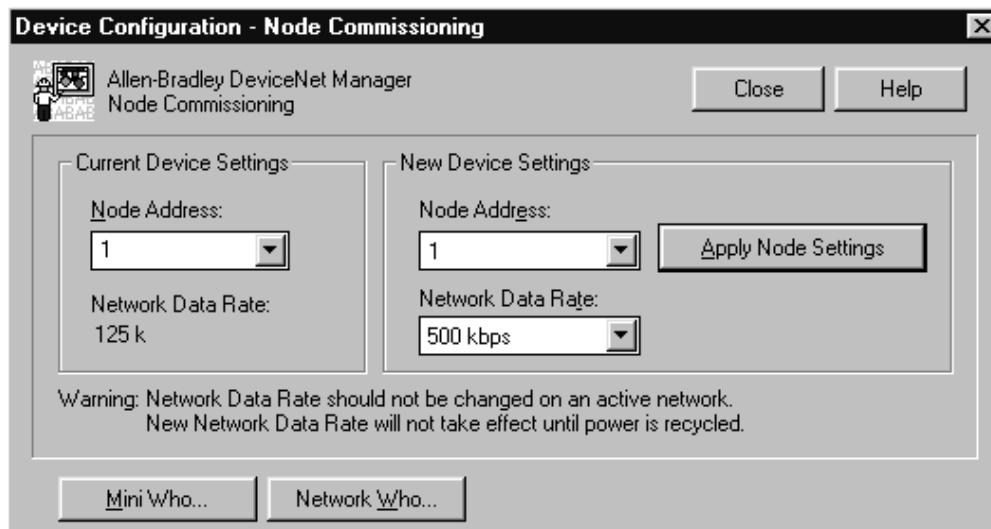


Abb. 9.18 Baudrate einstellen

9.7 Parameter im EEPROM abspeichern

Das spannungsausfallsichere Abspeichern der Parameter im EEPROM geschieht über den Device Configurator - Basic Mode.

Wechseln Sie bei bestehender Online-Verbindung über den Hauptmenu Punkt *Utilities* ins Menu *Basic Device Configuration*. Geben Sie die Teilnehmeradresse des Winkelcodierers an und wählen Sie von Class 35_{dez} (Position Sensor Object) den Service 22_{dez}. Betätigen Sie anschließend den Button *Save to Device*. (Siehe Abbildung 9.19)

In der Statuszeile des DeviceNet Managers wird das erfolgreiche Abspeichern der Parameter angezeigt.

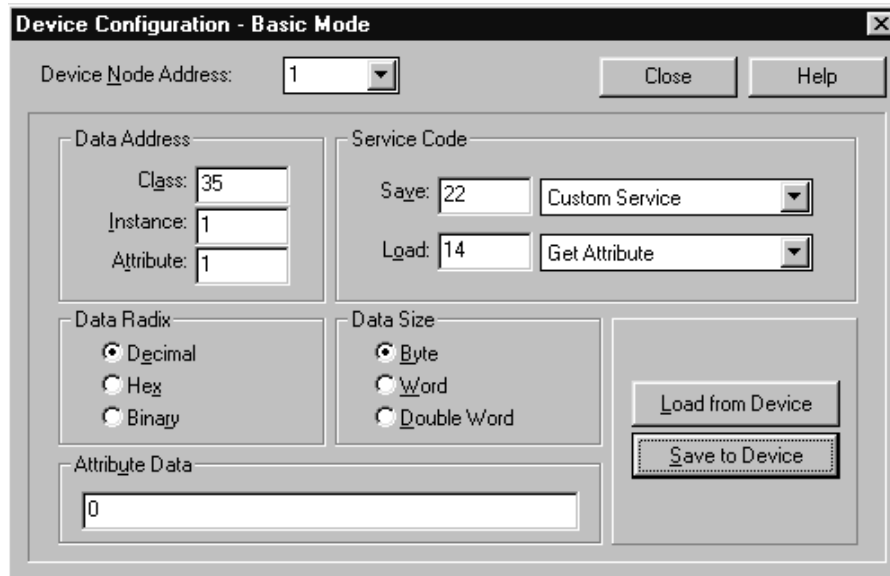


Abb. 9.19 Parameter im EEPROM abspeichern

9.8 Defaultwerte der Parameter laden

Mit dem Kommando „Restore“ Service Code 21_{dez} des Position-Sensor-Objektes lassen sich alle Defaultwerte der Winkelcodierparameter wiederherstellen.

Wechseln Sie dazu bei bestehender Online-Verbindung über den Hauptmenu-Punkt *Utilities* ins Untermenu *Basic Device Configuration*. Geben Sie die Teilnehmeradresse des Winkelcodierers an und wählen Sie von Class 35_{dez} (Position Sensor Object) den Service 21_{dez}. Betätigen Sie anschließend den Button *Save to Device*.

Die Defaultwerte werden nach Spannung aus/ein wirksam.

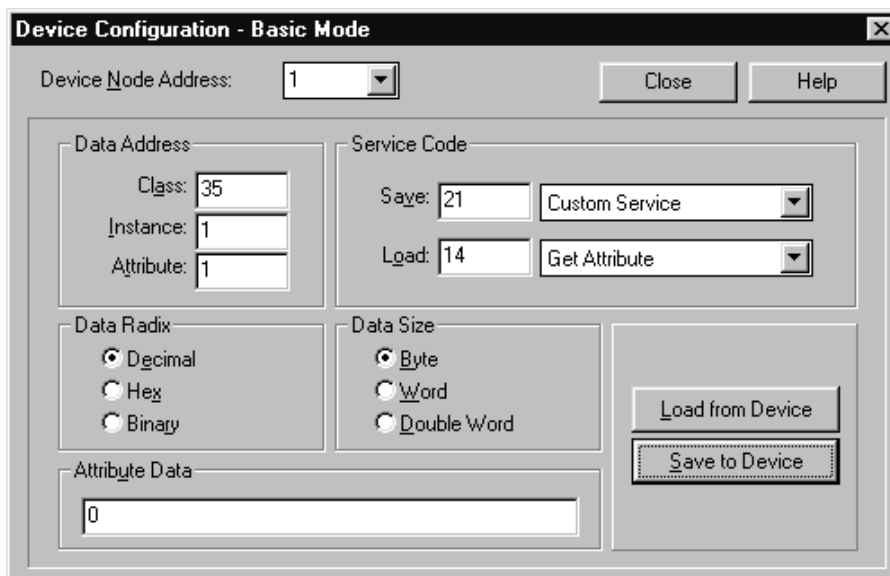


Abb. 9.20 Defaultwerte der Parameter laden

9.9 Beliebiges Attribut auslesen

Über den Basic Configurator lassen sich alle als lesbar bezeichneten Attribute der verschiedenen im Winkelcodierer vorhandenen DeviceNet-Objekte lesen. Die Auflistung aller Objekte und der darin enthaltenen Services finden Sie in [Kapitel 8](#).

Das Auslesen aller Attribute geschieht allgemein mit dem Service **Get Attribute** (Service-ID 14_{dez}), das Setzen mit dem Service **Set Attribute** (Service-ID 16_{dez}).

Als Beispiel wird hier die Seriennummer (Attribut 6 des Identity Object) gelesen.

Wechseln Sie dazu bei bestehender Online-Verbindung über den Hauptmenu-Punkt *Utilities* ins Untermenu *Basic Device Configuration...*. Geben Sie die Teilnehmeradresse des Winkelcodierers an und wählen Sie Class 1 (Identity Object), Instanz 1, Attribut 6. Ändern Sie die Darstellung (Data Radix) auf Hex und die Datengröße (Data Size) auf Double Word. Der notwendige Service 16_{dez} ist die Defaulteinstellung.

Betätigen Sie anschließend den Button *Load from Device*.

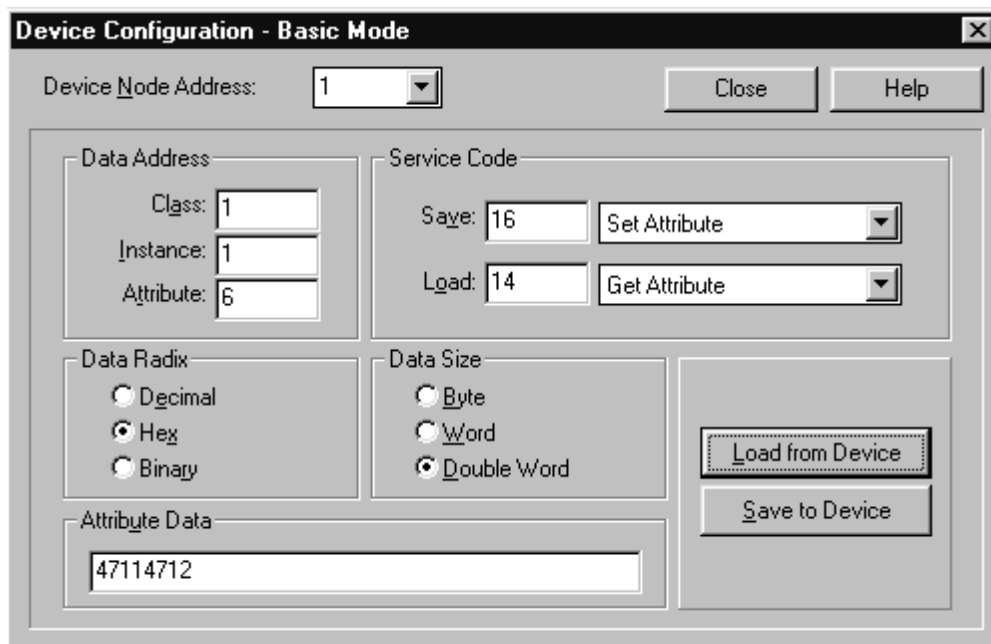


Abb. 9.21 Seriennummer auslesen

10. RS-Networx for DeviceNet

Dieses Kapitel beschreibt die Einbindung des TWK-Winkelcodierers CRN/D in das DeviceNet-Master-Slave-System einer Allen-Bradley-SPS anhand des Projektierungswerkzeuges RS-Networx for DeviceNet. Die genaue Kenntnis der Objekte, wie sie in [Kapitel 8](#) beschrieben sind, ist hierzu nicht erforderlich.

10.1 Installation EDS-Datei

Aufgrund eines Fehlers im DeviceNet-Manager befinden sich auf der Diskette zwei EDS-Dateien:

- 1.EDS: Standard EDS-Datei nach Spezifikation (Benutzen Sie diese EDS-Datei für Programme wie RS-Networx und andere)
- DNetMan.EDS: Benutzen Sie diese EDS-Datei nur für den DeviceNet-Manager

Wählen Sie zur Installation der EDS-Datei im Hauptmenü unter *Tools* den Menüpunkt *EDS Wizard...* Daraufhin werden Sie über mehrere Dialoge zur Installation der EDS-Datei und des Winkelcodierersymbols geführt. Tragen Sie dabei im folgenden Dialog als Dateiname **A:\1.EDS** ein.



Abb. 10.1 Installation EDS-Datei

Anschließend können Sie noch das auf der Diskette mitgelieferte Symbol für den Winkelcodierer auswählen:

- crn_d_m.ico für die Steckerversion
- crn_d_z.ico für die Version mit Anschlußhaube

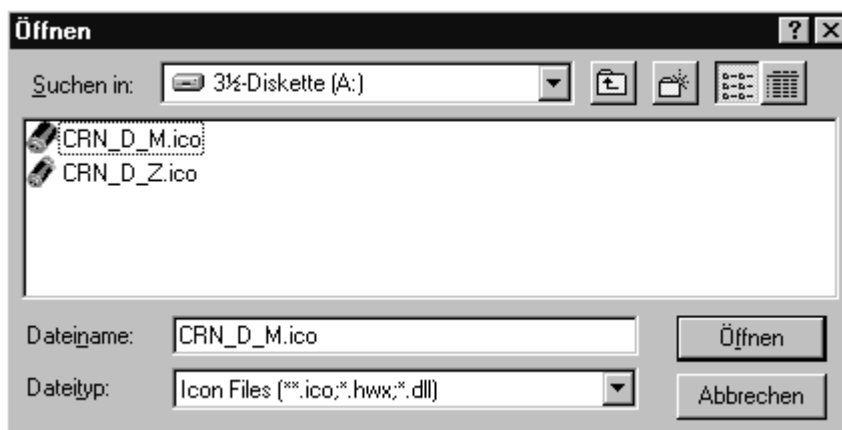


Abb. 10.2 Installation Winkelcodierer Symbol

10.2 Einbinden in den Bus

Nach der Installation der EDS-Datei erscheint der Winkelcodierer im Hardware-Katalog unter *Generic Device / Encoder CRN/D*. Er kann nun per Drag & Drop in den Bus eingebunden werden.

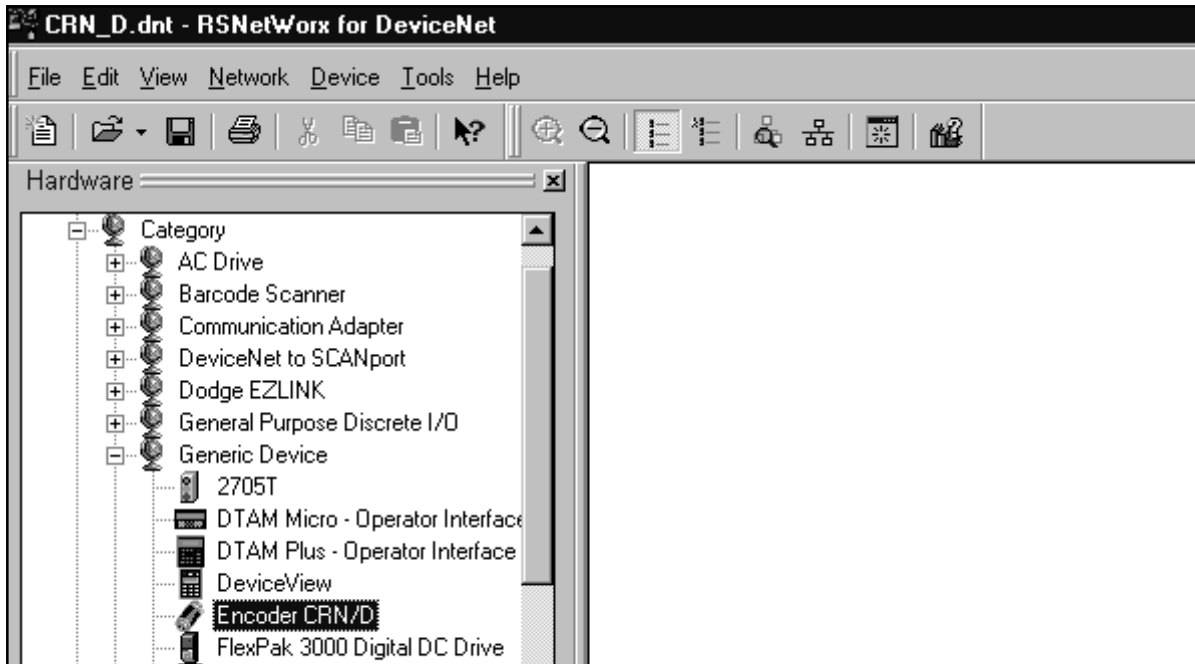


Abb. 10.3 Winkelcodierer Einbindung

Beim Einbinden des Winkelcodierers erhält dieser automatisch die nächste frei Adresse. Zum Ändern der Adresse und Einstellen der Codierer-Parameter klicken Sie nun doppelt auf dem im Bus befindlichen Codierersymbol. Daraufhin erscheint das folgende Fenster in dem Sie unter dem Register *General / Address* die im Winkelcodierer eingestellte Adresse einstellen müssen. Hier wurde die Winkelcodierer Default-Adresse 1 eingetragen.



Abb. 10.4 Teilnehmeradresse vergeben

10.3 Winkelcodierer parametrieren

Über das Register *Device Parameters* (Siehe Abb. 10.4) gelangt man an das folgende Fenster:

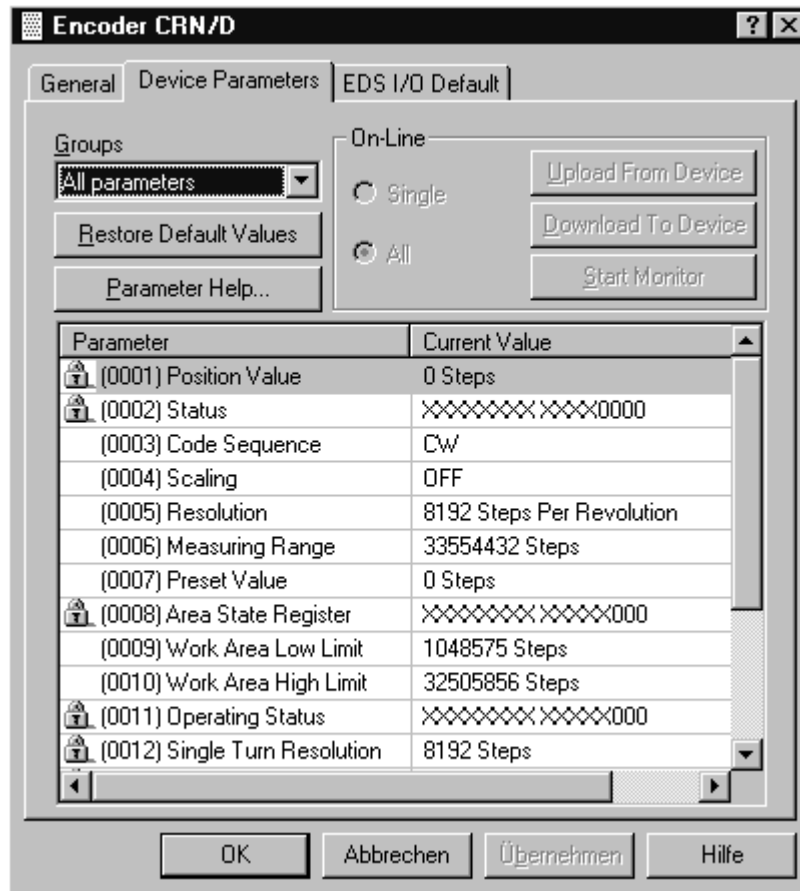


Abb. 10.5 Winkelcodierer Parameter

Hier werden alle Parameter und Statusinformationen des Winkelcodierers angezeigt (Dargestellt sind die Werte eines 25-Bit Codierers). Über das Feld *Groups* läßt sich die Anzeige reduzieren auf nur Parameter oder nur Statusinformationen.

Nach dem Einbinden des Codierers in den Bus und ohne Online-Verbindung zum Bus werden hier die Defaultwerte der Parameter und Statusinformationen aus der EDS-Datei angezeigt. Durch einen Doppelklick auf den Parameter ist dieser änderbar. Statusinformationen sind nur lesbar und mit einem kleinen Schloß gekennzeichnet. Mit dem Button *Restore Default Values* können jederzeit wieder die Defaultwerte aus der EDS-Datei ausgelesen werden.

Über den Button *Parameter Help* kann zu jedem Parameter ein Hilfetext abgerufen werden.

Bei bestehender Online-Verbindung können hier auch die Parameter und Statusinformationen aus dem Codierer ausgelesen (*Upload From Device*) bzw. die Parameter an den Codierer übertragen werden (*Download To Device*). In diesem Fall ist auch ein zyklisches Aktualisieren der Werte über den Button *Start Monitor* möglich.

Zum spannungsausfallsicheren Abspeichern der Parameter im EEPROM des Winkelcodierers steht ein **Save**-Kommando zur Verfügung. Außerdem lassen sich die Defaultwerte aller Parameter mit dem **Restore**-Befehl wieder zurückladen. Die Kommandos Save und Restore können nur über den „Class Instance Editor“ ausgeführt werden (Siehe [Kapitel 10.6](#) und [Kapitel 10.7](#)).

Das Register *EDS I/O Default* zeigt lediglich die Anzahl der Bytes an, die der Winkelcodierer bei den verschiedenen Betriebsarten sendet bzw. empfängt. Die Einstellung der Betriebsart ist nur in der Scanlist des Masters ([Kapitel 10.4](#)) möglich.

Ändern Sie nun die Parameter entsprechend Ihren Anforderungen und verlassen Sie das Fenster mit OK. Ein separates Übertragen der Parameter an den Winkelcodierer ist nicht erforderlich. Bei RS-Networx läßt sich die gesamte Busprojektion mit einem Kommando übertragen.

10.4 Winkelcodierer in die Scanlist aufnehmen

Um die Betriebsart des Winkelcodierers festzulegen und ihn einem Master zuzuordnen, muss dieser in die Scanlist eines Masters aufgenommen werden.

Das folgende Bild zeigt einen Busaufbau mit dem Master 1747-SDN einer Allen-Bradley-SPS SLC500. Alle folgenden Bilder beziehen sich auf diese Konfiguration.

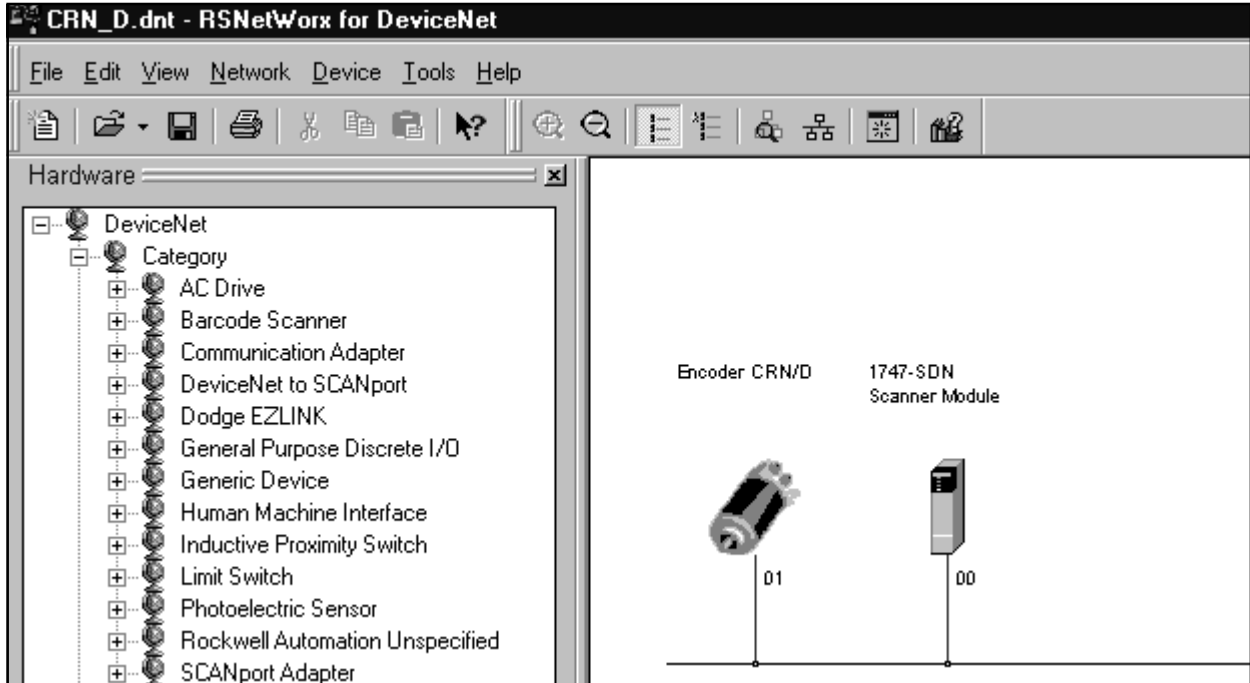


Abb. 10.6 Busaufbau mit Master

Durch einen Doppelklick auf das Scanner-Symbol gelangt man zu den Einstellungen des Masters. Gehen Sie dort auf das Register *Scanlist*, so erhalten Sie im linken Teil eine Auswahl der im Bus befindlichen Geräte. Markieren Sie dort den Winkelcodierer und bringen Sie ihn mit Hilfe der Pfeiltaste in das rechte Fenster.



Abb. 10.7 Scanlist

Möchten sie die Betriebsart des Codierers ändern, betätigen Sie den Button *Edit I/O Parameters*. Daraufhin gelangen Sie in das folgende Fenster, in dem Sie zwischen den Betriebsarten Poll-Mode, Bit-Strobe-Mode und Change of State / Cyclic-Mode wählen können. Es können auch mehrere Betriebsarten gleichzeitig angewählt werden. Für die Anzahl der Eingangsdaten (Rx-Size) muß immer fünf Byte und der Ausgangsdaten (TxSize) null Byte angegeben werden.

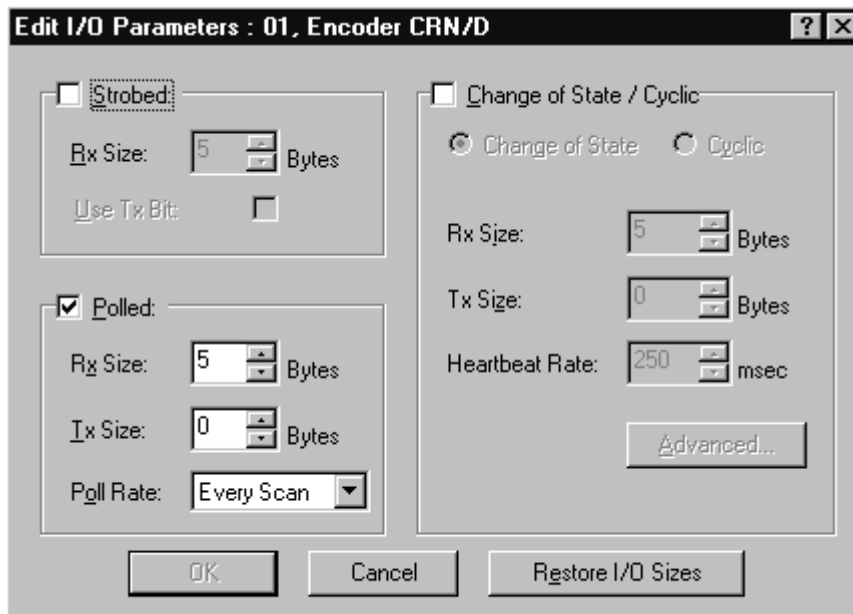


Abb. 10.8 Betriebsart des Winkelcodierers festlegen

Der Winkelcodierer ist nun mit der gewünschten Betriebsart in der Scanlist des Masters aufgenommen. Damit Sie in der SPS auf die Daten zugreifen können, muss dem Scanner noch mitgeteilt werden wo die Daten des Codierers im Eingangsbereich der SPS zu finden sein sollen. Dies geschieht über das sogenannte **Mapping**. Sie können entweder ein manuelles oder ein automatisches Mapping vornehmen.

Durch die Aufnahme in die Scanlist mit *Automap On Add* eingeschaltet (Siehe Abb. 10.7) wurde bereits ein automatisches Mapping vorgenommen. Im Register *Input* sollte der Codierer nun wie unten abgebildet (hier nur Betriebsart Polling Mode) zu sehen sein.



Abb. 10.9 Mapping

Verlassen Sie nun das Fenster mit O.K., speichern Sie Ihre Projektierung ab und gehen Sie online. Nachdem RS-Networx den Busaufbau eingelesen hat wählen Sie im Hauptmenü unter *Network*, *Download to Network* um den kompletten Busaufbau an den Scanner zu übertragen.

10.5 Adresse und Baudrate beim Winkelcodierer in Steckerversion einstellen

Die Steckerversion des CRN/D erlaubt die Einstellung der Teilnehmeradresse (MAC-ID) und der Baudrate (125 kB, 250KB, 500KB) per Software.

Die Einstellung der Baudrate sollte immer in einer Punkt-zu-Punkt-Verbindung zum Teilnehmer geschehen, da die Änderung der Baudrate eines Teilnehmers im Netz den kompletten Busbetrieb zum Erliegen bringen kann. Die Teilnehmeradresse hingegen kann auch im Busbetrieb geändert werden. Zu beachten ist jedoch, daß keine Adresse eines bereits bestehenden Teilnehmers verwendet wird.

Wirksam wird eine Änderung der Baudrate erst nach einem erneuten Spannung aus/ein, die Änderung der Adresse erfolgt direkt.

Bauen Sie nun eine Punkt-zu-Punkt-Verbindung zu Ihrem Winkelcodierer in Steckerversion auf und wählen Sie in RS-Networx unter *Tools*, *Node Commissioning*. Im nun erscheinenden Fenster können Sie über *Browse* den zu ändernden Winkelcodierer auswählen. Daraufhin werden die *Current Device Settings* angezeigt. Unter *New Device Settings* können die neuen Werte für Baudrate und Teilnehmeradresse angegeben werden. Dies könnte dann zum Beispiel wie folgt aussehen:

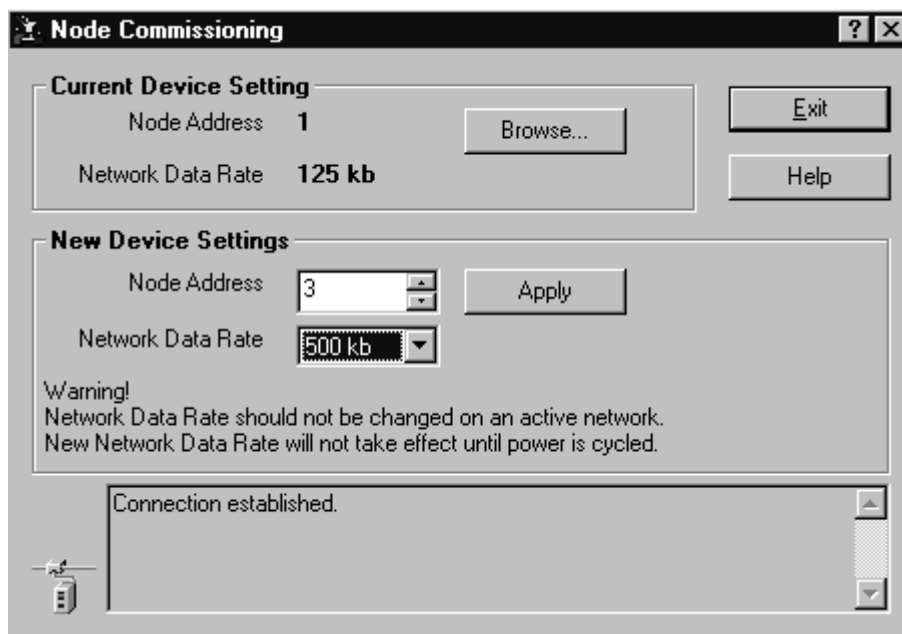


Abb. 10.10 Teilnehmeradresse und Baudrate einstellen

Wenn Sie nun den Button *Apply* drücken, werden nacheinander die Adresse und die Baudrate geändert und anschließend die neuen Werte unter *Current Device Settings* angezeigt.

10.6 Parameter im EEPROM abspeichern

Das spannungsausfallsichere Abspeichern der Parameter im EEPROM geschieht über den „Class Instance Editor“.

Markieren Sie den Winkelcodierer im Bus und wechseln Sie bei bestehender Online-Verbindung über den Hauptmenu Punkt *Device* ins Menu *Class Instance Editor*. Geben Sie die unter *Data Address*, *Class 0x23* (Position Sensor Object) an und wählen Sie unter *Service Code* den *Service Save* (0x16). Betätigen Sie anschließend den Button *Send*. (Siehe Abbildung 10.11)

In der Statuszeile wird das erfolgreiche Abspeichern der Parameter angezeigt.

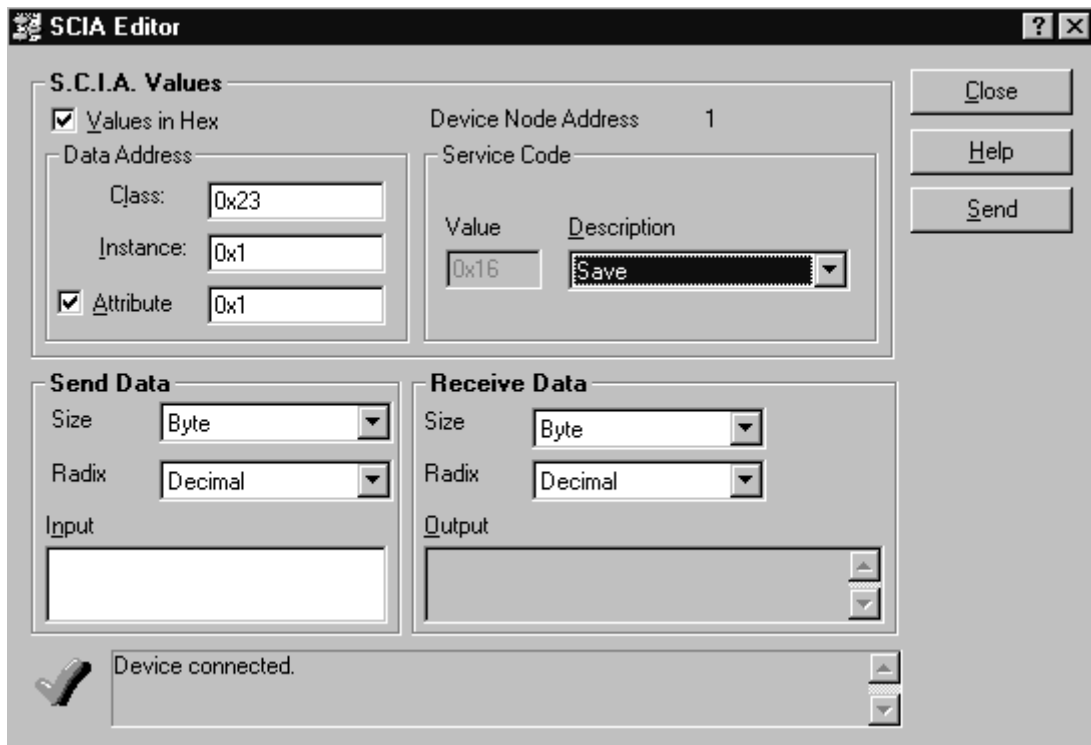


Abb. 10.11 Parameter im EEPROM abspeichern

10.7 Defaultwerte der Parameter laden

Mit dem Kommando „Restore“ Service Code 21_{dez} des Position-Sensor-Objektes lassen sich alle Defaultwerte der Winkelcodiererparameter wiederherstellen.

Markieren Sie den Codierer im Bus und wechseln Sie bei bestehender Online-Verbindung über den Hauptmenu-Punkt *Device* ins Untermenu *Class Instance Editor*. Geben Sie die unter *Data Address*, *Class* 0x23 (Position Sensor Object) an und wählen Sie unter *Service Code* den Service *Restore* (0x15). Betätigen Sie anschließend den Button *Send*.

Die Defaultwerte werden nach Spannung aus/ein wirksam.

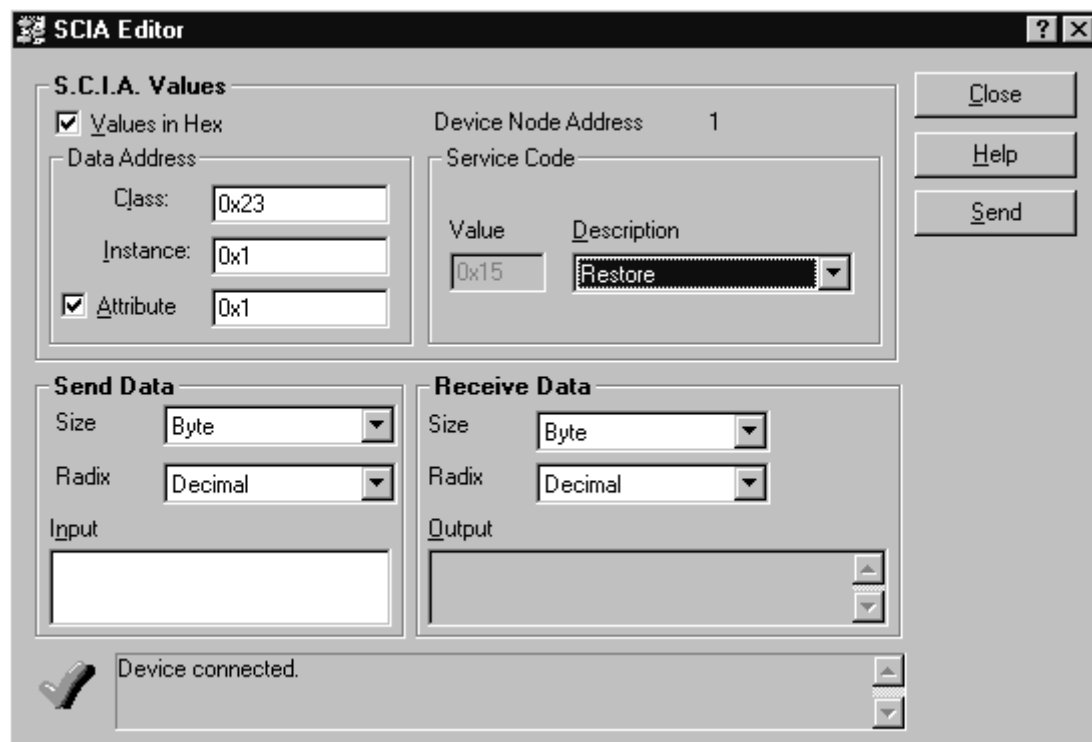


Abb. 10.12 Defaultwerte der Parameter laden

10.8 Beliebiges Attribut auslesen

Über den „Class Instance Editor“ lassen sich alle als lesbar bezeichneten Attribute der verschiedenen im Winkelcodierer vorhandenen DeviceNet-Objekte lesen. Die Auflistung aller Objekte und der darin enthaltenen Services finden Sie in [Kapitel 8](#).

Das Auslesen aller Attribute geschieht allgemein mit dem Service **Get Attribute** (Service-ID 0xE), das Setzen mit dem Service **Set Attribute** (Service-ID 0x10).

Als Beispiel wird hier die Seriennummer (Attribut 6 des Identity Object) gelesen.

Markieren Sie den Codierer im Bus und wechseln Sie bei bestehender Online-Verbindung über den Hauptmenu-Punkt *Device* ins Untermenü *Class Instance Editor*. Geben Sie die unter *Data Address*, *Class 0x1* (Identity Object), *Attribute 0x6* (Seriennummer) an und wählen Sie unter *Service Code* den Service *Get Single Attribute* (0xE). Ändern Sie unter *Received Data*, *Size* in *Double* und *Radix* in *Hexadecimal*. Betätigen Sie anschließend den Button *Send*. Daraufhin erscheint die Seriennummer im Ausgabefenster.

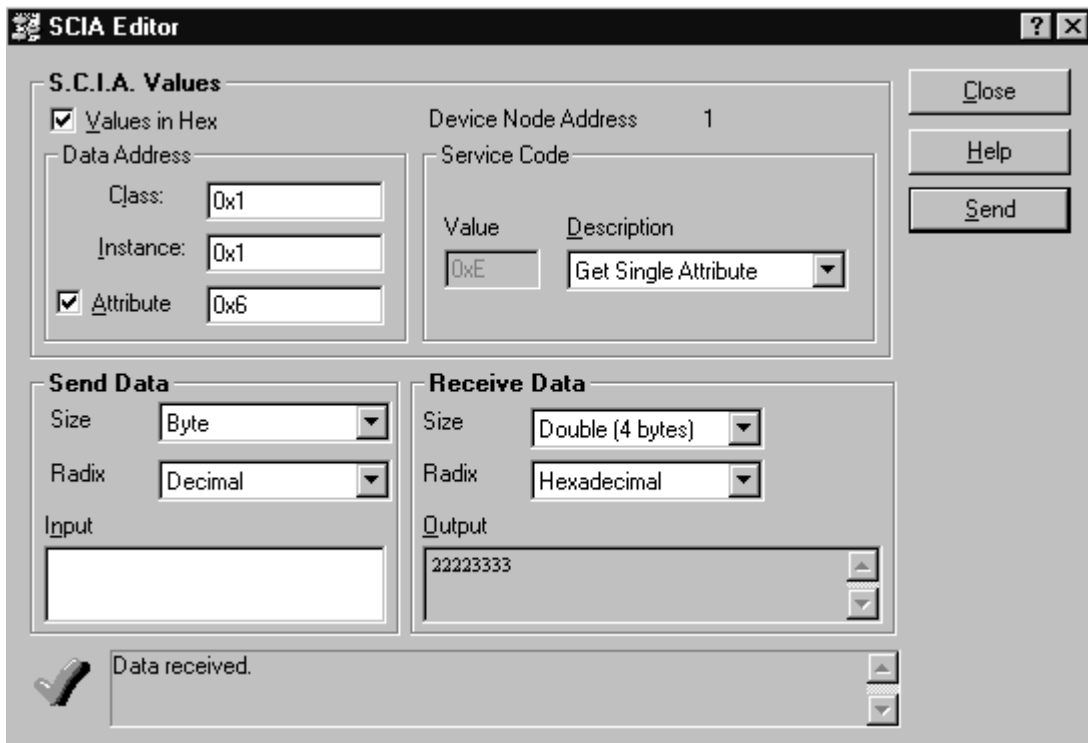


Abb. 10.13 Seriennummer auslesen

Anhang A: Literatur

- [1] ODVA - Open DeviceNet Vendor Association DeviceNet Specifications Release 2.0
- [2] DeviceNet Manager Software user Manual 1787 - MGR