Zeitschrift: Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

Herausgeber: Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

Band: 91 (2000)

Heft: 17

Artikel: Winkelkodierer in Feldbussystemen

Autor: Stark, Ulf

DOI: https://doi.org/10.5169/seals-855582

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 04.12.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Winkelkodierer in Feldbussystemen

Anwendungen unter Profibus, Interbus und CAN

In den letzten Jahren haben Feldbussysteme in der Automatisierungstechnik Einzug gehalten und sind zum technischen Standard geworden. Seit langem werden auch Winkelkodierer mit entsprechenden Schnittstellen ausgerüstet, wobei sich insbesondere Profibus, CAN und Interbus durchgesetzt haben. Dieser Artikel erläutert die verschiedenen Encoder-Protokolle sowie die spezifischen Besonderheiten der verschiedenen Bussysteme.

Der Einzug von Feldbussystemen in der Automatisierungstechnik hat auch vor Winkelkodierern nicht Halt gemacht. Von verschiedenen Herstellern wurden diese Geräte schon frühzeitig mit den entsprechenden Schnittstellen ausgerüstet, so-

Adresse des Autors
Ulf Stark, Dipl.-Ing., TWK-Elektronik GmbH, D-40239 Düsseldorf

dass sie als Slave-Teilnehmer problemlos in das entsprechende Bussystem eingebunden werden konnten (Bild 1). Neben der physikalischen Umsetzung der Schnittstelle galt es, den Application Layer für Encoder zu definieren. Diese Aufgabe wurde im Rahmen von Nutzerorganisationen durch die Definition von Encoder-Profilen durch die Kodierer-Hersteller gelöst.

In der Kommunikation zwischen dem Master (Profibus-DP-Master, CANopen-Master u.a.) und dem Slave (z. B. Encoder) hat man durch die Profildefinition für Encoder eine standardisierte Schnittstelle.

Warum Bussysteme?

Nennenswerte Gründe für den Einsatz von Bussystemen sind neben der Minimierung des Verdrahtungsaufwandes der erkennbare Wandel der Verlagerung der Intelligenz in die Peripherie, also an den Ort des Geschehens.

Die Kommunikation zwischen Master und Slave ist nicht nur durch reine Positionserfassung gekennzeichnet, sondern die Intelligenz der Sensoren (Einsatz von Microcontroller-Systemen) wird für weit umfangreichere Aufgaben genutzt. Es können Fehleraussagen gemacht sowie prophylaktisch Fehlerzustände erkannt werden. Der Einsatz von intelligenten Systemen dient der Erhöhung der Zuverlässigkeit des Gesamtsystems.

Frühzeitig wurde von den verschiedenen Komponenten-Herstellern die Tatsache erkannt, dass man bei sehr komplexen Bussystemen nicht umhin kommt, das so genannte Protokoll (Layer 7 des OSI-Schichtenmodells) gemeinsam zu definieren. Für den Anwender ist es erforderlich, dass er von verschiedenen Herstellern ein genormtes Protokoll zur Verfügung hat. Damit ist die Austauschbarkeit der Komponenten gewährleistet, und es besteht die Möglichkeit, flexibel zu reagieren.

Als Hersteller von Sensoren hat TWK frühzeitig erste Erfahrungen auf dem Gebiet der Feldbustechnik für Sensoren und Aktoren durch die Mitarbeit in den jeweiligen Nutzerorganisationen gesammelt. Unter der Schirmherrschaft von Nutzerorganisationen sind die Profile für Winkelkodierer erstellt worden. Dabei wurden entsprechend dem Entwicklungsstand Erfahrungen mit den bereits auf dem Markt vorhandenen Bussystemen berücksichtigt.

Profibus DP

Die Profilbeschreibung für Winkelkodierer [4] beinhaltet Anwendungen bei Profibus DP. DP steht für Dezentrale Peripherie. Anwendung findet der Profibus-DP da, wo eine besonders effiziente und schnelle Datenübertragung erforderlich ist. Die Kommunikation zwischen Master (DPM1) und Slave erfolgt in Übertragungsraten bis 12 MBaud.

Der Dienst Direct Data Link Mapper (DDLM) bietet dem User Interface einen komfortablen Zugang zur Schicht 2. Die umfangreichen Dienste des Profibus sind in [1] detailliert dargestellt. Für das Verständis der Funktionen werden hier nur die Dienste DDLM_Chk_Cfg, DDLM_Set_Prm, DDLM_Data_Exchange und DDLM_Slave_Diag näher beschrieben.

Konfiguration des Teilnehmers

Die Klassifizierung erfolgt mit der Zuordnung des Winkelkodierers in eine der in Tabelle I beschriebenen Konfigurationen. Man unterscheidet Class-1- und Class-2-Winkelkodierer. Class-1-Kodierer senden ihre Positionsdaten als Input-Daten entsprechend der Konfiguration als 16-Bit-Wert (1-Wort-Input-Daten) oder als 32-Bit-Wert (2-Wort-Input-Daten) auf den Bus. Das 1-Wort-Format findet in der Regel Anwendung bei Monotour-Winkelkodierern (max. 13 Bit) und das 2-Wort-Format bei Multitour-Winkelkodierern (max. 25 Bit).

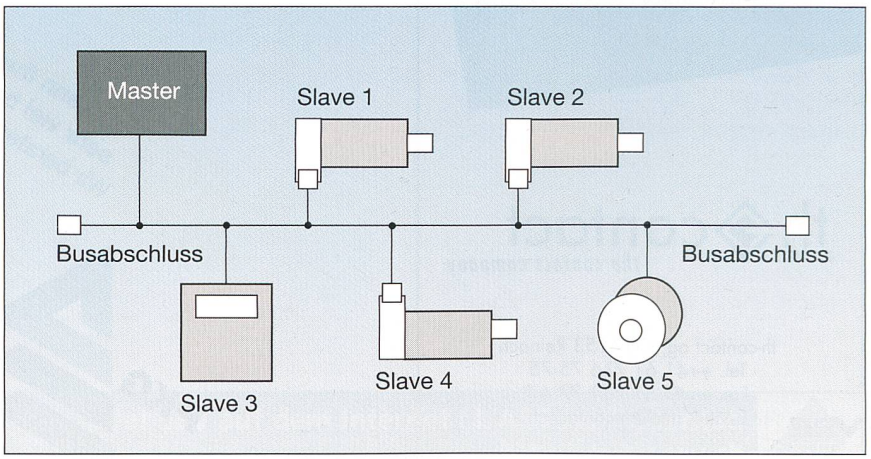


Bild 1 Einbindung von Winkelkodierern in ein Bussystem

Konfiguration	D0	D1	F0	F1
Class 1, 2	Class 1	Class 1	Class 2	Class 2
Baugruppenkonsistenz	ja	ja	ja	ja
Formatlänge	1 Wort	2 Worte	1 Wort	2 Worte
Typ	Input	Input	In-/Output	In-/Output

Tabelle I Der Dienst DDLM_Chk_Cfg-Konfiguration des Profibus-DP-Encoders

Parameter	Datentyp	Parameter Octet-Nummer und Bit-Nummer	Encoder Class
Standardparameterdaten User_Prm_Data	Byte	1–7	1,2
	Byte	8	1,2
Betriebsparameter des Kodierers	Byte	9	1,2
Codeverlauf	Bit	Octet 9, Bit 0	1,2
Class-2-Funktionalität	Bit	Octet 9, Bit 1	1,2
Diagnoseroutine	Bit	Octet 9, Bit 2	optional
Skalierungsfunktion	Bit	Octet 9, Bit 3	2
Auflösung Schritte / 360°	unsigned 32 Bit	10–13	2
Gesamtschrittzahl	unsigned 32 Bit	14–17	2
reserviert		18–25	2
Herstellerspezifische Funktionen		26–29	optional

Tabelle II Der Dienst DDLM_Set_Prm-Parametrierung des Profibus-DP-Encoders

Class-1-Teilnehmer geben ihre physikalische Position unverändert als absoluten Wert an. Bei Class-2-Teilnehmern erfolgt die Berechnung des Positionswertes. Dabei werden die Skalierungsfunktion, der Preset-Wert sowie der Offset-Wert bei der Berechnung des Positions-Ist-Wertes berücksichtigt. Mittels der Skalierungsfunktion können die Werte für Auflösung sowie Gesamtschrittzahl entsprechend der Applikation des Anwenders variiert werden.

Class-2-Kodierer setzen für die Berechnungen einen Microcontroller voraus. Der Einsatz der Controllertechnik ermöglicht natürlich auch ein umfangreiches Fehlerhandling und letztendlich auch eine Verlagerung der Intelligenz in die Peripherie (Encoder).

Parametrierung des Teilnehmers

Die Parameterdaten (Tab. II) setzen sich aus busspezifischen und DP-Slave-spezifischen User-Parameter-Daten zusammen. Die busspezifischen Daten (Octet 1–7) enthalten neben dem Stationsstatus (Octet 1) die Identnummer des Teilnehmers (Octet 5, 6). Die Identnummer (z. B. 1962 hex) wird durch die Profibus-Nutzerorganisation (PNO) vergeben und identifiziert den Teilnehmer als TWK-Winkelkodierer mit 25 Bit, d. h. mit einer Gesamtschrittzahl von 33 554 432 Schritten und Class-1- bzw. Class-2-Funktionalität.

Im Octet 9 werden die Betriebsparameter des Kodierers festgelegt. Man unterscheidet den Codeverlauf (cw = clockwise, ccw = counter clockwise) sowie die Freigabe der Class-2-Funktionalität, der Diagnoseroutine und der Skalierungsfunktion.

Sind entsprechend der Applikation Anpassungen der Parameterauflösung und der Gesamtschrittzahl erforderlich, erfolgt das über die Freigabe der Class-2- und der Skalierungsfunktion. Die eingestellten Werte für Auflösung (Octet 10–13) und der Gesamtschrittzahl (14–17) werden entsprechend den Vorgaben des Anwenders programmiert. Die Berechnung des Positions-Ist-Wertes erfolgt unter Berücksichtigung der aktuell programmierten Werte.

Datenaustausch zwischen Master und Slave

Der Dienst DDLM_Data_Exchange bewerkstelligt den zyklischen Datenaustausch zwischen Master und den angeschlossenen Slaves. Dabei unterscheidet man Input-Daten (Slave → Master) und Output-Daten (Master → Slave). Die Positions-Ist-Werte des Kodierers sind Input-Daten.

Das Setzen des Referenzwertes erfolgt über Output-Daten. Dabei schickt der Master ein Telegramm an den Slave (Kodierer). Der Referenzwert (Bit 0 bis Bit 25) wird im Data-Exchange-Modus durch Setzen von Bit 31 (32-Bit-Datenformat) bzw. Bit 15 (16-Bit-Datenformat) übertragen. Nach erfolgter Programmierung des Referenzwertes muss Bit 31 bzw. Bit 15 wieder zurückgesetzt werden (1 → 0).

Diagnose des Teilnehmers

Ein wesentlicher Vorteil der Verlagerung der Intelligenz in die Peripherie ist, dass man nun intelligente Einheiten hat, die Diagnosedaten dem Master bereitstellen können und somit zur Erhöhung der Verfügbarkeit der Teilkomponenten und letztendlich zu einem besseren Zuverlässigkeitsverhalten des Gesamtsystems beitragen.

Anschlusstechnik des Profibus-Kodierers

In der täglichen Praxis haben sich insbesondere zwei Lösungen für den Hardwareanschluss des Kodierers in der

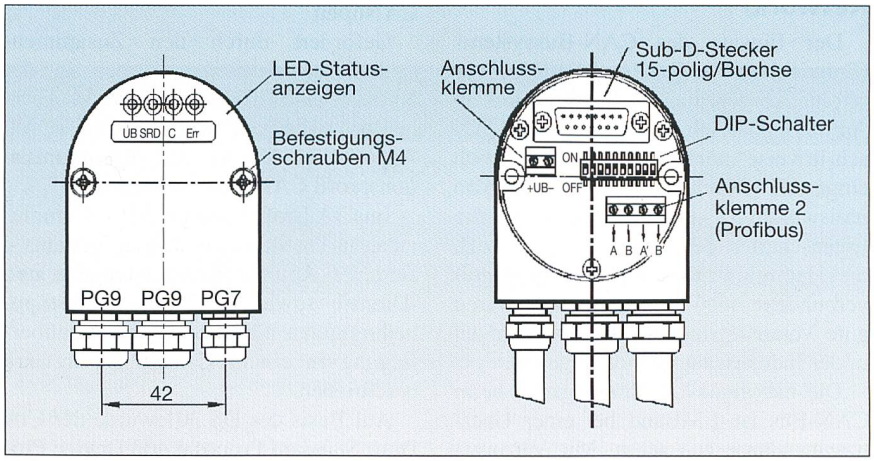


Bild 2 Anschlusshaube für Kodierer mit Profibus-DP-Interface

Feldbustechnik herauskristallisiert: der Steckeranschluss (beachte: Adresseinstellung über den Dienst Set_Slave_Add, Realisation der Abschlusswiderstände erfolgt separat) und die Anschlusshaubentechnik. Letztere stellt einen T-Koppler dar, der in der Praxis von vielen Herstellern von Winkelkodierern angewendet wird. Neben der Versorgungsspannung (24 VDC) sind die A- und die B-Leitungen (ankommender Bus) sowie die A'- und die B'-Leitungen (weiterführender Bus) als Signalleitungen erforderlich.

Neben der physikalischen Verbindung des Kodierers mit der Buswelt erfolgt in der Anschlusshaube die Adresseinstellung des Slaves sowie die Möglichkeit des Zuschaltens der Bus-Abschlusswiderstände (Bild 2).

Ein weiteres wichtiges Kriterium stellen Diagnoseleuchtdioden dar, die insbesondere während der Inbetriebnahmephase sich als sehr wichtiges Hilfsmittel erweisen können.

Für die Ausführung der Topologie des Gesamtbussystems und zur problemlosen Installation wird auf den Installationsguide der PNO verwiesen [2].

Zum Lieferumfang gehört die Gerätestammdatendatei (GSD-Datei) [3] sowie eine Applikationssoftware für die S7-Welt. Die GSD-Datei wird in das entsprechende Verzeichnis der Konfigurationssoftware (z. B. Step 7, Siemens) eingebunden. Nach der Aktualisierung enthält der Hardware-Katalog den neu installierten Teilnehmer, in unserem Fall den Encoder. Bei der Konfiguration der Kundenapplikation kann nun problemlos der Teilnehmer über die Schritte Konfiguration und Parametrierung eingebunden werden. Zu beachten ist, dass die physikalisch eingestellte Adresse in der Anschlusshaube mit der Adresse in der Step-7-Software übereinstimmt.

CAN (Controller Area Network)

Der Einsatz des CAN-Bussystems (Entwicklung der Fa. Bosch) erfolgt sehr erfolgreich durch namhafte Hersteller im Automobilbereich. Daneben wurde CAN schrittweise auch im Industriebereich eingesetzt. Durch die Verfügbarkeit von preiswerten IC und bedingt durch die Eigenschaften des Bussystems, wie z.B. die Hamming-Distanz 6 und die damit verbundene hohe Datensicherheit, waren gute Voraussetzungen für Anwendungen in der Industrieautomation gegeben.

Die maximale Übertragungsrate beim CAN-Bus ist 1 MBaud bei einer Übertragungslänge von 40 m. Mit Verringerung der Übertragungsrate sind natürlich grössere Übertragungslängen realisierbar.

Data Frame							
Start of Frame	Arbitration Field 11-Bit-Identifier	Arbitration Field RTR-Bit	Control Field	Data Field 8 Byte	CRC Field	Acknowledge Field	End of Frame

Tabelle III Data Frame nach CAN-Spezifikation Version 2.0 A

Der Datentransfer bei CAN wird durch die Priorität der Nachricht geregelt [5]. Es gibt bei CAN keine Teilnehmeradressen. Jede Nachricht kann von allen Teilnehmern gleichzeitig empfangen werden (Broadcasting). Die Entscheidung, ob die empfangene Nachricht weiter verarbeitet wird, trifft jeder einzelne Teilnehmer. Kriterium für diese Entscheidung ist der Identifier, der mit jeder Nachricht übertragen wird. Es wird zwischen einem 11-Bit-Identifier (Spezifikation 2.0 A) und einem 29-Bit-Identifier (Spezifikation 2.0 B) unterschieden.

Für das Verständnis des Protokollaufbaus ist in Tabelle III das Data Frame dargestellt. Neben dem Data Frame ist in der Spezifikation weiterhin ein Remote Frame, ein Overload Frame und ein Error Frame definiert.

Verwendet man den 11-Bit-Identifier, sind in der Praxis bis 2032 verschiedene Nachrichten möglich. Dabei gilt, dass die Priorität der Nachricht mit steigendem Identifier fällt. Die maximale Anzahl der Nutzdaten beträgt 8 Byte.

Auf Basis des CAN-Protokolls entstanden viele proprietäre Protokolle für den Layer 7 (Anwendungsschicht). Beispielsweise hat TWK ein Protokoll für Winkelkodierer spezifiziert, welches ohne grossen Overhead auskommt.

Wegen ihrer hohen Marktakzeptanz werden nachfolgend CANopen und DeviceNet beschrieben.

CANopen

Gefördert durch den Zusammenschluss verschiedener Firmen in der Nutzerorganisation CiA (CAN in Automation) wurde auf Basis des CAL (CAN Application Layer) das Kommunikationsprofil CANopen entwickelt.

Im CiA Draft Standard 301 – Communication Profile for Industrial Systems – sind die Kommunikations-Objekte und -Dienste sowie die Start- und Stoppbedingungen für die Nachrichtenübertragung in einem CANopen-Netzwerk beschrieben.

Auf Basis des DS 301 wurde der CiA Draft Standard Proposal 406 Device Profile for Encoders [6] erarbeitet.

Communication Object Identifier COB-ID

Der COB-ID (Communication Object-Identifier) dient zur Kennung einer CAN-Nachricht. Der Identifier ist 11 Bit breit (Spezifikation 2.0 A) und setzt sich aus einem 4-Bit-Funktionscode und einer 7-Bit-Knotennummer zusammen. Die Priorität einer Nachricht fällt mit steigendem COB-Identifier. Über die Knotennummer wird der CANopen-Teilnehmer adressiert. Das erfolgt über die Einstellung der Knotenadresse (DIP-Schalter) bei der Anschlusshaubenversion bzw. über einen Eintrag im Objektverzeichnis (Index 2000) bei der Steckerversion.

Nachrichtenobjekte und Funktionscodes

Bei Verwendung des Pre-defined Master/Slave Connection Set (siehe DS 301 CANopen) sind die Nachrichten-Identifier wie folgt definiert. Es werden vier funktionsabhängige Nachrichtengruppen unterschieden:

- Administration Messages (Layer Management, Network Management, Identifier, Distribution Management)
- Service Data Messages
- Process Data Messages
- Pre-defined Messages (Synchronisation, Time Stamp, Emergency)

Unter Administration Messages werden jene Nachrichten verstanden, die bei der Benutzung der CAL-Dienste zur Netzwerkkonfiguration, Netzwerkverwaltung und Identifier-Vergabe verwendet werden.

Netzwerkmanagement

Die Winkelkodierer sind als Minimum-Capability-Device-Boot-up-Geräte nach CANopen ausgeführt, dass heisst, sie verhalten sich nach dem Zustandsdiagramm in Bild 3.

Die einzelnen Zustände lassen sich wie folgt beschreiben:

Initialisierung. Ausgangszustand nach Anlegen der Versorgungsspannung. Der Kodierer wechselt nach Durchlauf der Initialisierungsroutinen automatisch in den Zustand Pre-operational.

Pre-operational. Der Knoten kann über SDO-Nachrichten unter dem Standard-Identifier angesprochen werden.

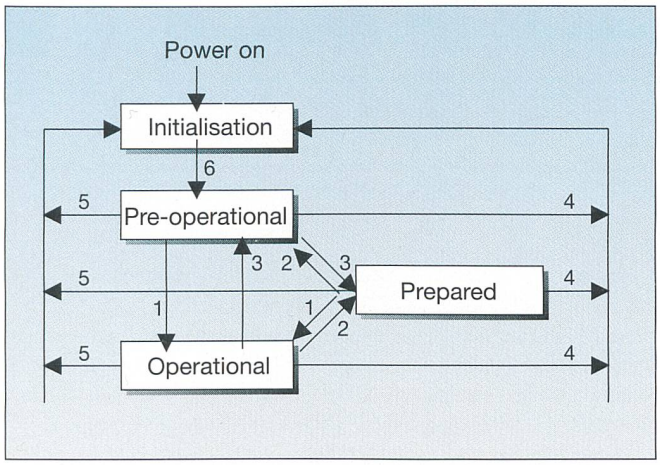


Bild 3 Minimal Boot-up Procedure für Encoder

Die Programmierung der Kodierer- bzw. Kommunikationsparameter ist durchführbar.

Operational. Der Kodierer ist aktiv. Der Positionswert des Kodierers wird über die PDOs ausgegeben.

Prepared. In diesem Zustand ist der Knoten nicht mehr aktiv, dass heisst sowohl eine SDO(Service Data Object)- als auch eine PDO(Process Data Object)-Kommunikation ist nicht mehr möglich. Der Kodierer kann über die entsprechenden NMT-Kommandos in den Zustand Operational oder Pre-operational gesetzt werden.

Process Data Object

Mit den Process Data Messages wird die Echtzeitübertragung von aktuellen Prozessdaten realisiert. Die übertragenen Objekte werden als Process Data Object (PDO) bezeichnet. CANopen unterstützt die Übertragung von synchronen und asynchronen Nachrichten.

Service Data Object

Die Service Data Messages werden beim direkten Schreib- und Lesezugriff auf einzelne Objektverzeichniseinträge ausgetauscht. Die auszutauschenden Objekte werden als Service Data Object (SDO) bezeichnet. Die Hauptanwendung dieses Nachrichtentyps liegt bei der Gerätekonfiguration.

DeviceNet

Die Hersteller- und Anwenderorganisation für DeviceNet [8] ist die ODVA, die Open DeviceNet Vendor Association. DeviceNet setzt auf das CAN-Protokoll auf. In der Norm ISO/DIS 11898 sind die Schichten 1 und 2 des ISO/OSI-Kommunikationsmodells für CAN beschrieben. DeviceNet beinhaltet Festlegungen (Tab. IV) zu den Schichten 0 und 1 und definiert den Application Layer (Schicht 7).

DeviceNet ist wie folgt charakterisiert:

- Trunkline- und Dropline-Konfiguration (Trunkline: Hauptstrang, Dropline: Stichleitung)
- max. 64 Knoten sind adressierbar (MAC-ID)
- Trennung des Teilnehmers vom Bus ohne Unterbrechung des Busbetriebes
- Übertragungsraten: 125 kBaud, 250 kBaud und 500 kBaud

DeviceNet ist ein verbindungsorientiertes Netzwerk. Eine DeviceNet-Verbindung liefert die Verbindung zwischen verschiedenen Teilnehmern. Dabei werden zwei Connection-ID für den bidirektionalen Datenaustausch zugewiesen. Das CAN-Identifierfield wird für Explicit Messaging Connections und I/O-Messaging verwendet.

Wie bereits dargestellt, unterscheidet DeviceNet grundsätzlich zwei Arten von Protokollen: Explicit Messaging und I/O-Messaging.

Das Explicit Messaging (Group 2 Message) wird für das Lesen und Schreiben von Attributen verwendet. Das I/O-Messaging (Group 1 Message) dient der Ausgabe der Winkelkodierer-Daten im laufenden Betrieb.

Betriebsarten des Winkelkodierers

Polling Mode. Der Master fragt zyklisch die Positionsdaten des Winkelkodierers ab. Der Kodierer liefert die Daten nach dem Poll Command. Die Zeit zwischen zwei Abfragen ist einstellbar.

Bit Strobed Mode. Nach dem Bit-Strobe-Kommando liefern die ausgewählten Slaves ihre Eingangsdaten auf den Bus. Die Auswahl kann zwischen einem und allen Slaves betreffen.

Change of State. Bei Änderung der Positionsdaten sendet der Kodierer seine Positions-Ist-Werte. Über die Heartbeat Rate (Senden bei gleich bleibenden Werten) und die Production Inhibit Time (Verzögerungszeit) kann die Ausgabe des Positionswertes zeitlich zugeordnet werden.

Explicit Messages

Jede Verbindung zu einem Kodierer wird über die Group 2 Only Unconnected Explicit Request Message eingerichtet (Group 2 Message ID 6). Im Pre-defined-Master-Slave-Connection-Set wird über den Service 4B hex Allocate_Master/Slave_Connection_Set des DeviceNet-Objekts eine Instanz des Connection-Objektes gebildet. Diesem Service wird in einem Allocation Choice Byte die gewünschte Verbindungsart (Betriebsart) mitgegeben (Tab. V).

Objektverzeichnis

Jeder einzelne Teilnehmer wird durch seine implementierten Objekte beschrieben bzw. wird durch sie charakterisiert. Man unterscheidet neben den Objekten, Klassen, Instanzen, Attribute und des Weiteren Services, welche z. B. das Lesen und Schreiben der Attribute, das Allocate_Master/Slave_Connection_Set sowie das Release der Verbindung u. a. ermöglichen.

Für die Definition der Winkelkodiererdaten wird das Position-Sensor-Objekt verwendet, da zum gegenwärtigen Zeit-

ISO Application (Layer 7)	Application Layer	DeviceNet-Spezifikation
ISO Data Link (Layer 2)	Logical Link Control (LLC)	DeviceNet-Spezifikation
	Media Access Control (MAC)	CAN-Protocol-Spezifikation
ISO Physical (Layer 1)	Physical Layer Signalling (PLS)	CAN-Protocol-Spezifikation
	Medium Attachment Unit (MAU)	DeviceNet-Spezifikation
ISO Media (Layer 0)	Transmission Media	DeviceNet-Spezifikation

Tabelle IV Layerübersicht DeviceNet

Tabelle V Allocation-Choice-Byte für die Auswahl der Betriebsart des Kodierers

Allocation Choice Byte Bit 8	7	6	5	4	3	2	Bit 1
Reserved	Acknowledge Suppression	Cycle	Change of State	Reserved	Bit Strobed	Polled	Explicit Message

punkt kein eigenständiges Profil für Encoder existiert.

Anschlusstechnik des DeviceNet-Kodierers

Zum Lieferumfang gehört analog wie beim CANopen-Winkelkodierer eine EDS-Datei (Electronic Data Sheet). Sie beinhaltet die Objekteinträge für die Konfiguration des Kodierers.

Für die Anschlusstechnik des Kodierers an den Bus gibt es wieder zwei Optionen. Die Anschlusshaubentechnik beinhaltet die MAC-ID-Einstellung (Adresse) und die Wahl der Baud-Rate über DIP-Schalter.

Bei der Steckerversion werden MAC-ID und Baud-Rate über Software konfiguriert (siehe DeviceNet Object; Class Code 03 hex). Der Abschlusswiderstand muss bei beiden Versionen vom Anwender prinzipiell separat vorgesehen werden.

Interbus

Ein sehr verbreitetes Bussystem ist der Interbus (Bild 4). Für die Einbindung von Kodierern wird der Installationsfernbus (Lokalbus) verwendet. Die Signalleitungen sowie die Leitungen für die Betriebsspannung (24 VDC) sind in einem Kabel zusammengefasst.

Bedingt durch die Ausgestaltung des Interbus als «räumliches Schieberegister» ist keine Adressierung des einzelnen Teilnehmers erforderlich. Der ID-Code des Teilnehmers weist auf die Art des Teilnehmers hin. So identifiziert der ID-Code 0237 hex einen Kodierer mit dem Profil K3 (ENCOM, Profil Nr. 71) [10] und einem Datenformat von 32-Bit-In- und -Output-Daten.

Im Interbusprotokoll werden anhand der physikalischen Anordnung des Teilnehmers 4 Byte an der entsprechenden Stelle im Summenrahmentelegramm vorgesehen.

Die Leistungsmerkmale der verschiedenen Profile (K1–K4) sind in Tabelle VI dargestellt.

Die Profile K1 und K2 geben den physikalischen Positions-Ist-Wert ohne Berechnung aus. Die Umsetzung des K3- und K4-Profils erfordert eine Controllerlösung im Kodierer.

K3-Profil

Bei der Kommunikation zwischen Master und Slave werden 2 Kommunikationsrichtungen unterschieden. Für die Programmierung des Kodierers (K3) enthält das 32-Bit-Datenformat 7 Steuerbits (Master → Slave). Als Antwort erhält der Master 7 Statusbits sowie den Positions-Ist-Wert.

Anschlusstechnik für Interbus-Kodierer

In der Regel hat sich der Steckeranschluss (2 Stecker) beim Interbus-Kodierer durchgesetzt.

Man unterscheidet die Signalleitungen für den ankommenden und den weiterführenden Bus. Eine Adresseinstellung sowie die Einstellung der Übertragungsrate ist nicht erforderlich. Der Interbus arbeitet mit der konstanten Übertragungsrate von 500 kBaud. Zu beachten ist das RBST-Signal. Bei Verbindung des RBST-Signals mit Signal-Ground ist der Teilnehmer als letzter Teilnehmer deklariert.

Interbus-Loop

Der Interbus-Loop zeichnet sich dadurch aus, dass über eine ungeschirmte Zweidrahtleitung (2×1,5 mm²) neben der Versorgungsspannung für den Sensor die Busdaten übertragen werden. Dabei unterscheidet sich das Protokoll für den Kodierer nicht vom gewohnten Interbus-Protokoll (K3-Profil). Der Anschluss der

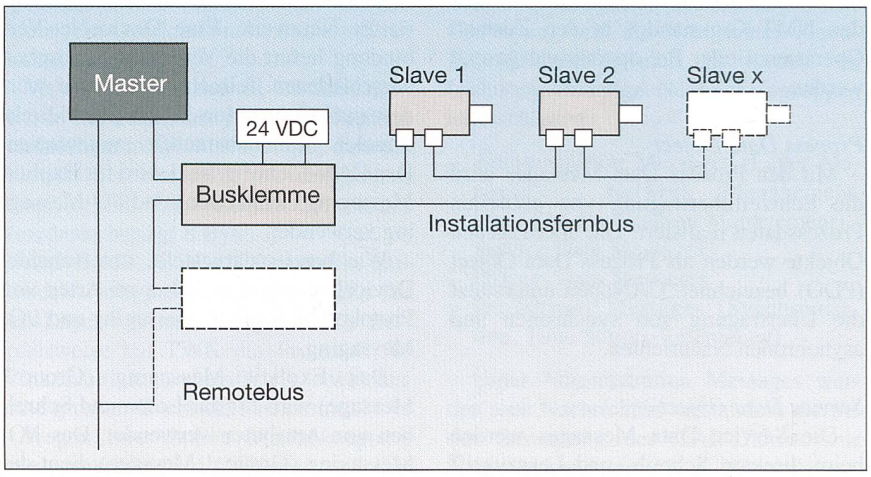


Bild 4 Topologie des Interbus

Profil	K1	K2	K3	K4
Datenformat	1 Wort	2 Worte	2 Worte	2 Worte
Datentyp	Input	Input	In-/Output	In/Output
Steuer/Status-Bits			7 Bit	4 Bit
ID-Code	0136 hex	0236 hex	0237 hex	03F7 hex
Programmier-parameter	keine	keine	Schritte (Gesamtschrittzahl) Umdrehungen Preset Nullpunktverschiebung Offset Positions-Ist-Wert-Codierung (Ausgabecode und Codeverlauf)	siehe K3

Tabelle VI Profilübersicht für Interbus-Encoder nach Encom-Profil-Nr. 71

Kodierer erfolgt über Schneid-Klemm-Anschlusstechnik. Der zulässige Laststrom von 1,5 A begrenzt die Anzahl der Teilnehmer im Interbus-Loop-Strang.

Literatur

[1] Profibus Specification, Normative Parts of Profibus-FMS, -DP, -PA according to the European Standard EN 50170 Volume 2, Order No. 0.042.

[2] Aufbaurichtlinien Profibus-DP/FMS, Version 1.0, September 1998, Best.-Nr. 2.111.

[3] GSD-Spezifikation für Profibus-DP, GSD Revision 2, Version 1.0, Oktober 1998, Best.-Nr. 2.121.

[4] Profibus-Profile for Encoders, Order No. 3.062.

[5] CAN Specification Version 2.0, 1991, Philips Semiconductors.

[6] CiA Draft Standard Proposal 406 Version 2.0, May 1998, Device Profile for Encoders.

[7] CiA Draft Standard 201–207 Version 1.1, February 1996, CAN Application Layer for Industrial Applications.

[8] DeviceNet Specification Release 2.0, 1998, Open DeviceNet Vendor Association, Inc., 8222 Wiles Road, Suite 287, Coral Springs, Florida 33067, USA.

[9] Winkelkodierer CRN/D als Teilnehmer im DeviceNet, CRN/D 10771 AD, TWK-Elektronik GmbH, Düsseldorf.

[10] Encom-Nutzergruppe, Profil: Encoder, Profil-Nummer: 71, 28. November 1994.

Les codeurs angulaires dans les systèmes de bus de terrain

Applications sous Profibus, Interbus et CAN

Ces dernières années, les systèmes de bus de terrain se sont imposés dans la technique d'automatisation et ils sont devenus la norme technique. Depuis longtemps, des codeurs angulaires sont équipés d'interfaces correspondantes, et surtout les systèmes Profibus, CAN et Interbus se sont imposés. L'article expose les différents protocoles d'encodeur ainsi que les spécificités des différents systèmes de bus.

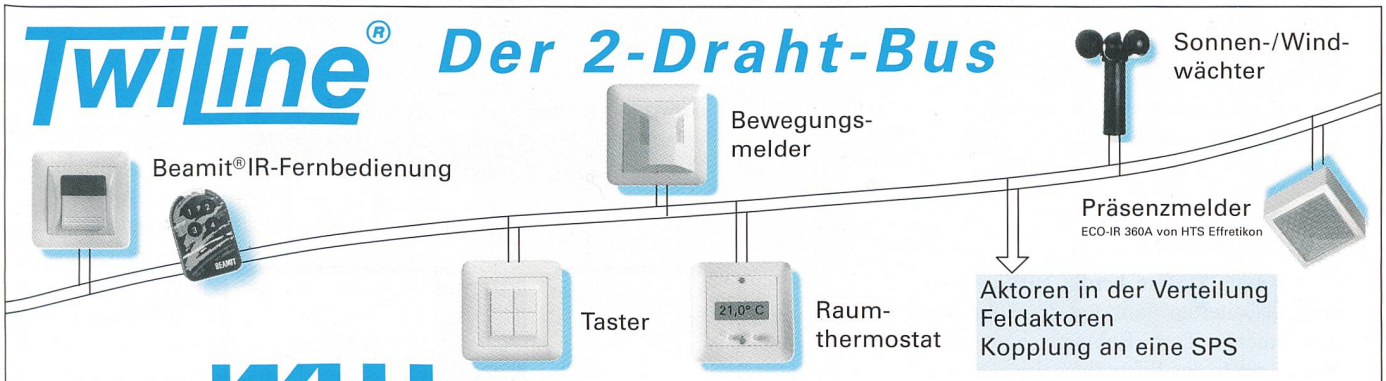
AUTOMATISIERUNGSTECHNIK
auf Hochschulniveau

- Eidg. Anerkennung des Nachdiplomstudiums und der Nachdiplomkurse
- Modularer Aufbau, nach individuellen Bedürfnissen zusammenstellbar
- Weiterbildungsmöglichkeit für gut ausgebildete Praktiker

Dokumentation und Auskünfte: edu swiss (vormals NDIT/FPIT)

Morgartenstrasse 2 C, 3014 Bern
Tel. 031 33 55 120, Fax 031 33 55 130
EMail office@eduswiss.ch, www.eduswiss.ch

TwiLine® Der 2-Draht-Bus



Ein Produkt der WAHLI W. Wahli AG Feinmechanik + Elektronik
Freiburgstrasse 341, CH-3018 Bern, Telefon 031 992 59 11, Fax 031 991 84 60
www.wahli.com