CANopen Tutorial

Version 2.0

https://support.industry.siemens.com/cs/ww/de/view/109479771

Siemens Industry Online Support

Rechtliche Hinweise

Nutzung der Anwendungsbeispiele

In den Anwendungsbeispielen wird die Lösung von Automatisierungsaufgaben im Zusammenspiel mehrerer Komponenten in Form von Text, Grafiken und/oder Software-Bausteinen beispielhaft dargestellt. Die Anwendungsbeispiele sind ein kostenloser Service der Siemens AG und/oder einer Tochtergesellschaft der Siemens AG ("Siemens"). Sie sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit und Funktionsfähigkeit hinsichtlich Konfiguration und Ausstattung. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern bieten lediglich Hilfestellung bei typischen Aufgabenstellungen. Sie sind selbst für den sachgemäßen und sicheren Betrieb der Produkte innerhalb der geltenden Vorschriften verantwortlich und müssen dazu die Funktion des jeweiligen Anwendungsbeispiels überprüfen und auf Ihre Anlage individuell anpassen.
Sie erhalten von Siemens das nicht ausschließliche, nicht unterlizenzierbare und nicht übertragbare Recht, die Anwendungsbeispiele durch fachlich geschultes Personal zu nutzen. Jede Änderung an den Anwendungsbeispielen erfolgt auf Ihre Verantwortung. Die Weitergabe an Dritte oder Vervielfältigung der Anwendungsbeispiele oder von Auszügen daraus ist nur in Kombination mit Ihren eigenen Produkten gestattet. Die Anwendungsbeispiele unterliegen nicht zwingend den üblichen Tests und Qualitätsprüfungen eines kostenpflichtigen Produkts, können Funktions- und Leistungsmängel enthalten und mit Fehlern behaftet sein. Sie sind verpflichtet, die Nutzung so zu gestalten, dass eventuelle Fehlfunktionen nicht zu Sachschäden oder der Verletzung von Personen führen.

Haftungsausschluss

Siemens schließt seine Haftung, gleich aus welchem Rechtsgrund, insbesondere für die Verwendbarkeit, Verfügbarkeit, Vollständigkeit und Mangelfreiheit der Anwendungsbeispiele, sowie dazugehöriger Hinweise, Projektierungs- und Leistungsdaten und dadurch verursachte Schäden aus. Dies gilt nicht, soweit Siemens zwingend haftet, z.B. nach dem Produkthaftungsgesetz, in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der schuldhaften Verletzung des Lebens, des Körpers oder der Gesundheit, bei Nichteinhaltung einer übernommenen Garantie, wegen des arglistigen Verschweigens eines Mangels oder wegen der schuldhaften Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegen oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist mit den vorstehenden Regelungen nicht verbunden. Von in diesem Zusammenhang bestehenden oder entstehenden Ansprüchen Dritter stellen Sie Siemens frei, soweit Siemens nicht gesetzlich zwingend haftet.
Durch Nutzung der Anwendungsbeispiele erkennen Sie an, dass Siemens über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden kann.

Weitere Hinweise

Siemens behält sich das Recht vor, Änderungen an den Anwendungsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in den Anwendungsbeispielen und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.
Ergänzend gelten die Siemens Nutzungsbedingungen (https://support.industry.siemens.com).

Securityhinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.
Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.
Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z.B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.
Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter: https://www.siemens.com/industrialsecurity.
Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.
Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter: https://www.siemens.com/industrialsecurity.

© Siemens AG 2019 All rights reserved

Inhaltsverzeichnis

© Siemens AG 2019 All rights reserved

1 Den CAN-Bus verstehen

Übersicht

CAN ist ein nachrichtenorientiertes Multi-Master-Protokoll für den schnellen, seriellen Datenaustausch und hat sich in zahlreichen Bereichen der Industrie etabliert die

- hohe Robustheit und Datensicherheit fordern,
- niedrige Kosten erwarten,
- eine breite Palette von Anbietern von Komponenten, zugehöriger Software und Werkzeugen fordern.

Diese Bereiche sind beispielsweise

- die Automobiltechnik (Vernetzung unterschiedlicher Steuergeräte, Sensoren und Multimedia
- die Automatisierungstechnik (zeitkritische Sensoren im Feld, raue Industrieumgebung),
- die Medizintechnik
- die Luftfahrt- und Schifftechnik.

Durch diese Vielseitigkeit nimmt die CAN-Bus Technik einen hohen Stellenwert innerhalb der möglichen Bussysteme ein.

Was erhalten Sie?

Dieses Dokument wurde für Einsteiger in die CAN-Bus Technik entwickelt.

Es beschreibt auf einfache Art die notwendigsten Begriffe und die Architektur des CAN-Busses.

Damit wird dem Leser ein umfassendes Kompendium an die Hand gegeben, das ihm hilft, die an ihn gestellten Anforderungen effizienter zu erfüllen.

Ideal ist dieses Dokument auch als Begleitheft zur Applikationsbeschreibung, welche sich auf derselben HTML-Seite wie dieses Dokument befindet.

© Siemens AG 2019 All rights reserved

2 Der CAN-Bus: Die Basis für CANopen

Was steht hier?

Dieses Kapitel beschreibt die Grundlagen der Übertragungstechnik mit dem CAN-Bus. Diese Einführung ist für das Verständnis der Folgekapitel wichtig, da der CAN-Bus durch CANopen adaptiert wurde.

2.1 Was ist der CAN-Bus?

Einführung

Der **C**ontroller **A**rea **N**etwork (CAN)-Bus wurde 1983 von Bosch und Intel als ein Feldbus für die echtzeitfähige und kostengünstige Datenübertragung bei Fahrzeugen entwickelt.

Durch die preisgünstigeren Feldgeräte wird der CAN-Bus inzwischen auch in der industriellen Automatisierung verwendet und für die Kommunikation auf Sensor-Aktor Ebene weiterentwickelt (CAN in Automation http://www.can-cia.org).

Durch die Standardisierung von Schicht 1 (Physical Layer) und 2 (Data Link Layer) des OSI Referenzmodells in der ISO 11898 und der Offenheit des Protokolls ist dank CAN eine Kommunikation zwischen Geräte, Sensoren und Aktoren unterschiedlicher Hersteller möglich.

© Siemens AG 2019 All rights reserved

Merkmale

Der CAN-Busses zeichnet sich durch folgende Fähigkeiten aus:

- **Multi-Master:** Alle Stationen sind gleichberechtigt und können selbstständig Daten Senden und Empfangen. Sie sind gleichermaßen für den Buszugriff, der Fehlerbehandlung und der Ausfallüberwachung verantwortlich. Bei Ausfall eines Teilnehmers fällt dadurch nicht das Gesamtsystem aus.
- **Nachrichtenorientierte Kommunikation:** Die Stationen besitzen keine Adresse. Die Kommunikation zwischen den Stationen erfolgt über Broadcast. Eine Integration neuer Stationen in das Gesamtsystem erfolgt ohne Neukonfiguration, da keine Adressbekanntgabe nötig ist.
- **Flexibilität:** Bei CAN sind nicht die Teilnehmer, sondern die Nachrichten mit einer Kennung (Identifier - ID) versehen (Inhaltsbezogene Adressierung). Alle CAN-Telegramme stehen jedem CAN-Knoten zum Empfang zur Verfügung (Broadcasting). Jeder Empfänger ist selbst für die Selektion der CAN-Botschaften verantwortlich. Eine solche empfängerselektive Adressierung ist sehr flexibel, ist aber auf eine empfängerseitige Filterung der empfangenen CAN-Botschaften angewiesen.
- **Priorisierung der Nachrichten**: Über spezielle Identifier im CAN-Telegramm können die Nachrichten zugeordnet und priorisiert werden.
- **Einzigartige Fehlerbeschränkung:** Verschiedene Sicherungsverfahren minimieren fehlerhafte Datenübertragungen und garantieren eine netzweite Datenkonsistenz.

Feldgeräte

Beispiele für CAN-Bus-Geräte sind:

- Automatisierungsgeräte
- PCs
- Ein- /Ausgangsmodule

- Antriebsverstärker
- Analysegeräte
- Sensoren und Aktoren wie Temperatur-/ Druckfühler

2.2 Übertragungstechnik

Im Layer 1 des OSI Referenzmodells wurden detaillierte Spezifikationen für die physikalische Ebene der Kommunikation festgelegt wie z. B. Kabelempfehlungen, Stecker und Leistungstreiber.
Layer 2, der Data Link Layer, steuert und schützt die Kommunikation auf der Ebene eines Frames.
Layer 3 bis 6 sind für CAN nicht notwendig.
Auf der Ebene der Applikationsschicht (Layer 7) sind zwar verschiedene Protokolle definiert worden (z. B. SDS, DeviceNET, CANopen), es existiert aber keine bindende Spezifikation.

Hinweis Diese folgende Beschreibung bezieht sich auf die CAN Spezifikation 2.0

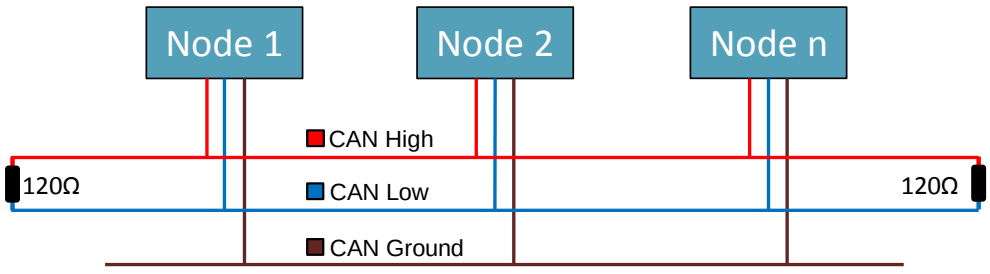
© Siemens AG 2019 All rights reserved

2.2.1 Bus-Topologie und Physical Layer

Gerätekopplung am CAN-Bus

CAN ist Bitstrom-orientiert (serielle Datenübertragung) und als Linien-Bus konzipiert. Als Übertragungsmedium dient eine verdrillte 2-Draht Leitung. Die Teilnehmer werden als "Knoten" bezeichnet und sind parallel über Stichleitungen miteinander verbunden. Der Bus muss an beiden Enden mit einem 120 Ω Widerstand abgeschlossen werden.

Abbildung 2-1



Steckerbelegung bei CAN

Als Steckverbinder für den CAN Bus in der Automatisierungstechnik hat sich der 9-polige Sub-D Stecker durchgesetzt.
Für die Übertragung von CAN Signalen ist mindestens ein 3 poliges Kabel erforderlich. Die Adern werden mit CAN-High, CAN-Low und Ground bezeichnet.

Leitungslänge und Baudrate

Die Buslänge und die zulässige Länge der Stichleitungen sind abhängig von der Bitrate. Bei CAN ist eine maximale Bitrate von 1MBit/s definiert. Liegt die Bitrate

über 250 kBit/s spricht man vom High Speed CAN, darunter entsprechend der Low Speed CAN.

Alle Knoten im CAN-Bus müssen mit derselben Bitrate arbeiten. Da manche Teilnehmer nicht alle Bitraten unterstützen, ist die maximale Baudrate im CAN-Bus durch die größte, von allen unterstütze Übertragungsgeschwindigkeit begrenzt.

Physikalische Umsetzung

Der CAN-Bus ist ein serielles System und arbeitet mit Differenzsignalen zwischen den Leitungen CAN-High und CAN-Low. Der Grundpegel beträgt ohne äußere Einflüsse durch z. B. elektromagnetische Einstreuung, 2,5V.

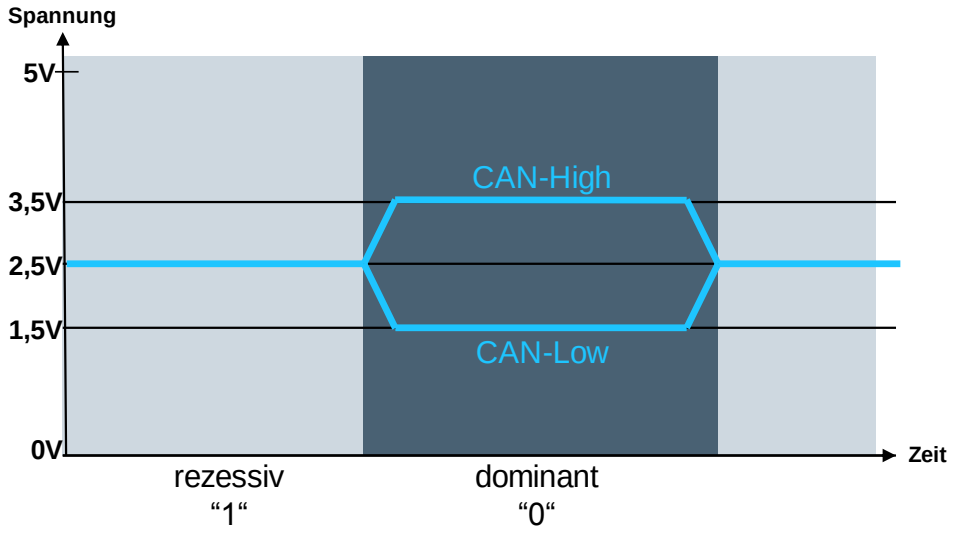
Es gilt dabei folgende Vereinbarung:

Tabelle 2-1

	Pegel	Zustand	Differenzsignal
High Speed CAN	dominant	0	U_{diff} > 0,9V
	rezessiv	1	U_{diff} < 0,5V
Low Speed CAN	dominant	0	U_{diff} > -3,2V
	rezessiv	1	U_{diff} < -3,2V

Die Übertragung der Daten erfolgt so, dass ein Bit, je nach Zustand, entweder dominant oder rezessiv auf den Busleitungen wirkt. Ein dominantes Bit überschreibt dabei ein rezessives Bit.

Abbildung 2-2



© Siemens AG 2019 All rights reserved

2.2.2 CAN auf dem Data Link Layer

Aufbau des CAN Datentelegramms

Nachrichten werden bei CAN in definierte Rahmen übertragen. CAN unterscheidet dabei folgende Formate:

- Datentelegramm (Data Frame): für die Datenübertragung
- Datenanforderungstelegramm (Remote Frame): dient zur Anforderung von Daten
- Überlasttelegramm (Overload Frame): Flussregelung
- Fehlertelegramm (Error Frame): Fehlererkennung und Benachrichtigung

Der Aufbau eines Datentelegramms ist wie folgt definiert:

Abbildung 2-3

Bits	1	12/32	6	0-64	16	2	7
Feld	Start	Arbitrierung	Kontrolle	Daten	Sicherung	Bestätigung	Ende

Hinweis

Für ein CAN Datentelegramm existieren im Augenblick zwei Spezifikationen, die sich hauptsächlich in der Anzahl der Bits im Arbitrierungsfeld unterscheiden.

CAN 2.0 A: 12 Bit- Arbitrierungsfeld

CAN 2.0 B: 32 Bit- Arbitrierungsfeld

Tabelle 2-2

Abschnitt	Untergruppe	Beschreibung
Startfeld	RTR	Das Startbit (SOF) definiert den Beginn eines Datentelegramms oder Datenanforderungstelegramms und führt immer einen Low-Pegel (Zustand 0: dominant).
Arbitrierungs-feld		
	Identifier	Der Identifier ist ein Kenncode für die Art der Nachricht und wird für die Arbitrierung am Bus verwendet. Da die Datentelegramme von allen Knoten empfangen wird (Broadcast) wird anhand dieses Identifiers entschieden, ob die empfangene Nachricht ignoriert oder bearbeitet wird. Da der Zustand "0" bei CAN dominant ist, hat folglich der kleinste Identifier-Wert die höchste Priorität. Die Bitanzahl des Identifiers ist abhängig von der Spezifikation: CAN 2.0 A: 11- Bit Identifier CAN 2.0 B: 29- Bit Identifier.
	RTR	RTR kennzeichnet, ob es sich um ein Daten- oder Datenanforderungstelegramm handelt: RTR = 0: Datentelegramm RTR = 1 : Datenanforderungstelegramm

© Siemens AG 2019 All rights reserved

Abschnitt	Untergruppe	Beschreibung
Kontrollfeld		
	IDE	Das Identifier Extension Bit kennzeichnet ein 11-Bit oder 29-Bit Identifier. IDE = 0: 11-Bit Identifier IDE = 1: 29-Bit Identifier
	r0	reserviert
	DLC [0..3]	Diese 4 Bits kodieren die Anzahl der Datenbytes im Datenfeld.
Datenfeld		Im Datenfeld werden die Nutzdaten übertragen. Pro Telegramm können bis zu 8 Byte übertragen werden.
Sicherungsfeld		Das Sicherungsfeld enthält eine CRC-Prüfsumme über die vorausgehenden Felder. Mit dieser Prüfsumme können Übertragungsfehler ausgeschlossen werden.
Bestätigungs-feld		
	ACK Slot	Der sendende Knoten setzt beide Bits rezessiv (Zustand 1) auf den Bus und wartet darauf, dass die anderen Teilnehmer diesen Pegel mit einem dominanten Pegel (0) überschreiben. Damit wird sichergestellt, dass mindestens ein Knoten die Nachricht richtig empfangen hat.
	ACK Delimiter	
Endfeld		Das Endfeld enthält 7 rezessive Bits.

© Siemens AG 2019 All rights reserved

Prinzip der Busarbitrierung

Bei CAN hören alle Knoten auf dem Bus mit- auch während des eigenen Sendevorgangs. Ein Knoten darf nur senden, wenn der Bus frei ist.

Dafür wurde für CAN das Buszugriffsverfahren CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) definiert.

Beginnen mehrere Teilnehmer gleichzeitig zu senden, wird der Bus-Konflikt über eine bitweise Arbitrierung gelöst. Dafür dient das Arbitrierungsfeld (Identifier) im Datentelegramm.
Jeder Sendeknoten beobachtet Bit-für-Bit den Buspegel. Dazu vergleicht er den Zustand seines zu sendende Bit mit dem momentan am Bus anliegende Bit. Sind diese Zustände identisch, sendet der Knoten das nächste Bit.
Der Teilnehmer, der das erste dominante Bit (Zustand "0") auf den Bus legt, setzt sich auf dem Bus durch. Die anderen brechen ihren Sendeversuch ab und wechseln in den Empfangsmodus.
Das erste dominante Bit im Identifier entscheidet also über die Priorisierung.

2.3 Fehlerbehandlung und– erkennung

CAN verfügt über eine Reihe von Kontrolleinrichtungen zur Fehlererkennung und -korrektur. Folgende drei Mechanismen sind auf der Nachrichtenebene implementiert:

1. Prüfsummenberechnung: Jedes Datentelegramm verfügt über ein 2 Byte großes Sicherungsfeld. Hier wird sendeseitig die über das Arbitrierung-, Kontroll- und Datenfeld berechnete Prüfsumme hinterlegt. Empfangsseitig wird ebenfalls eine Prüfsumme über die erwähnten Felder berechnet und mit der hinterlegten verglichen. Bei Nicht- Übereinstimmung ist bei der Datenübertragung ein Fehler aufgetreten und der Empfänger fordert das Telegramm erneut an
2. Datenrahmen- Überprüfung: Dieser Mechanismus überprüft die in der Struktur fest definierten Felder des Datentelegrams. Weist ein Feld ein oder mehrere fehlerhafte Bits auf, liegt ein Formfehler vor.
3. Quittierung: Jeder Knoten bestätigt den Empfang eines Telegramms durch Veränderung des ACK-Bits auf "dominant". Erkennt der Sender keine Änderung am ACK-Bit durch die Empfänger, liegt ein Übertragungsfehler vor.

Auf Bitebene sind weitere Maßnahmen zur Fehleranalyse vorgesehen:

1. Monitoring: Jeder Knoten der sendet, beobachtet gleichzeitig den Buspegel. Er erkennt dabei Differenzen zwischen gesendetem und empfangenen Bit. Dadurch können alle globalen Fehler und lokal am Sender auftretenden Bitfehler sicher erkannt werden.
2. Bit-stuffing: Ein Datenrahmen darf bis zum Ende des Sicherungsfeldes maximal fünf aufeinander folgende Bits gleicher Polarität haben.

© Siemens AG 2019 All rights reserved

3 Die Adaption von CAN: CANopen

3.1 Was ist CANopen?

Einführung

CANopen ist ein geräte- und herstellerunabhängiges Protokoll zur Kommunikation im CAN-Bus und deckt die Anwendungsschicht (Layer 7) des OSI Referenzmodells ab.

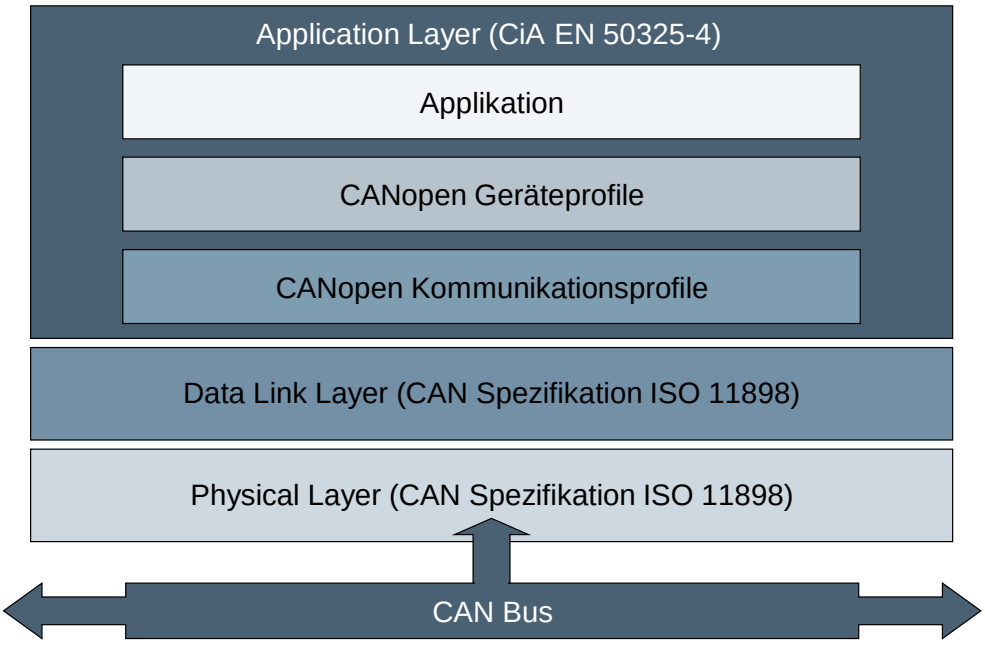
Neben dem Profibus hat sich in Europa CANopen in der Automatisierungstechnik und vielen Embedded-Control-Bereichen etabliert.

Bei der Standardisierung von CANopen durch "CAN in Automation (CiA)" im EN 50325-4 wurden elementare Punkte definiert:

- Schaffung einer einheitlichen Basis zum Austausch von Befehlen und Daten zwischen CAN-Bus-Teilnehmern mittels Kommunikationsobjekte (COB).
- Implementation von Mechanismen zum Austausch von Prozessdaten in Echtzeit, für die Übertragung großer Datenmengen oder das Senden von Alarm-Telegrammen.
- Festlegung definierte Schnittstellen zum Ansprechen bestimmter Parameter eines Gerätes durch Profile.

Die folgende Grafik zeigt die Einordnung von CANopen im OSI-Schichtenmodell:

Abbildung 3-1



© Siemens AG 2019 All rights reserved

Anpassung gegenüber CAN

Zur Datenkommunikation nutzt CANopen die CAN-Bus-Technik (siehe Kapitel 2.2 Übertragungstechnik).

Allerdings wurde CAN durch CANopen in wenigen Punkten angepasst:

- **Verzicht auf die Multi-Master Fähigkeit**: Die Kommunikation zwischen den Teilnehmern entspricht überwiegend dem Client- Server-Modell. Nur die Übertragung der Prozessdaten erfolgt nach dem Producer-Consumer-Modell. Ein CANopen Netzwerk kann bis zu 127 Knoten enthalten.
- **Netzwerkmanagement**: Spezielle Netzwerkmanagementobjekte erlauben die Überwachung und Steuerung des Netzwerkes.
- **Festgelegte Identifier**: In der Spezifikation von CANopen sind, um Konfigurationsaufwand einzusparen, eine Reihe von Kenncodes für die Art einer Nachricht bereits mit Defaultwerten vordefiniert.

3.2 Die CANopen Technik

3.2.1 Objekte und Profile

Objekte

Alle Abläufe bei CANopen werden über Objekte ausgeführt. Diese Objekte übernehmen unterschiedliche Aufgaben:

- Identifizierungsobjekte für Geräteinformationen (Name, Hersteller)
- Fehlerobjekte zur Anzeige des Fehlerzustandes eines Knoten
- Kommunikationsobjekte für den Datentransport zwischen den Knoten
- Netzwerkkonfigurationsobjekte um die Konfigurationsdaten den Knoten zuzuweisen
- Netzwerkmanagementobjekte zur Überwachung und Steuerung

Profile stellen diese Objekte nun entsprechend ihrer Aufgaben zusammen. Es wird unterschieden zwischen

- Standardisierte Profile
- Herstellerspezifische Profile

Standardisierte Profile

Standardisierte Profile umfassen Objekte, die auf verschiedenen Geräte ohne Anpassung verwendet werden können. Dazu gehört unter anderem

- die Kommunikationsprofile (z.B. DS 301, DS 302)
- die Geräteprofile DS 40x für
  - digitale bzw. analoge E/A Geräte (DS 401),
  - Antriebe (DS 402),
  - Bediengeräte (DS 403),
  - Sensoren und Regler (DS 404),
  - Programmierbare Steuerungen (DS 405) und
  - Encoder (DS 406)

© Siemens AG 2019 All rights reserved

Kommunikationsprofile

Die Objekte des Kommunikationsprofils DS 301 definieren eine einheitliche Basis für den gemeinsamen Daten- und Parameteraustausch zwischen unterschiedlichen Gerätetypen am CAN-Bus und initialisieren, steuern und überwachen das Gerät im Netzwerk.

Objekte des Kommunikationsprofils sind:

- Prozessdaten-Objekte (**P**rocess **D**ata **O**bjects/ PDO)
- Servicedaten-Objekte (**S**ervice **D**ata **O**bjects/ SDO)
- Objekte mit speziellen Funktionen zur Synchronisation und zur Fehlermeldung und -reaktion
- Objekte des Netzwerk-Managements NMT zur Initialisierung, Fehlerüberwachung und zur Statusüberwachung des Geräts.

Das Profil DS 302 ist ein Framework für programmierbare Geräte (CANopen Manager, SDO Manager).

Geräteprofile

Für bedeutende, in der industriellen Automatisierungstechnik eingesetzten Baugruppen wurden von der CiA Geräteprofile definiert. Diese Profile enthalten Objekte für die Grundfunktionen und Parameter des jeweiligen Standardgerätes. Auf diese Weise können die übertragenen Daten des Gerätes eindeutig und herstellerunabhängig interpretiert werden. Zudem ist ein Austausch von Geräten, die dem gleichen Profil folgen, weitgehendst problemlos durchführbar.

Das Geräteprofil DS 402 beschriebt demnach standardisierte Objekte für die Positionierung, Überwachung und Einstellung von Antrieben.

Herstellerspezifische Profile

Mit den Objekten der Standardisierten Profile werden nur die Grundfunktionen und Parameter eines Gerätetyps beschrieben. Der gesamte Funktions- und Parameterumfang wird in den herstellerspezifischen Profilen hinterlegt. Hier sind die Objekte zu finden, mit denen unter CANopen die vom Hersteller speziellen Funktionen genutzt werden können z.B. die Definition der unterstützen Datentypen (Byte, Word, Long etc).

© Siemens AG 2019 All rights reserved

3.2.2 Objektverzeichnis

Zentrale Verbindung der Objekte eines Gerätes ist das Objektverzeichnis. Hier werden alle Objekte in einer übersichtlichen tabellarischen Anordnung strukturiert. Die Gruppierung ist durch die Spezifikation vorgegeben.

Jedes Objekt wird über einen 16 Bit-Index, der als vierstellige Hexadezimalzahl dargestellt wird, adressiert.

Tabelle 3-1

Index (hex)	Objektgruppe
0000_h	Reserviert
0001_h – $009F_h$	Statische und komplexe Datentypen
$00A0_h$ – $0FFF_h$	Reserviert
1000_h – $1FFF_h$	Kommunikationsprofile (z. B. DS 301, DS 302)
2000_h – $5FFF_h$	Herstellerspezifische Geräteprofile
6000_h – $9FFF_h$	Standardisierte Geräteprofile
$A000_h$ – $FFFF_h$	reserviert

Die folgende Grafik zeigt einen Ausschnitt aus dem Objektverzeichnis für die Kommunikationsprofile:

Abbildung 3-2

Index	Object Name	Sub-Index	Description	Type	Access	Notes
1000h	Device Type	00h	Type of device	U32	RO	0000 0000h (No profile)
1001h	Error register	00h	Error register, connected to the EMCY object. Bit 0 indicates a generic error	U8	RO	-
1003h	Pre-defined error field	00h	Number of errors. Writing a 0 to this sub-index clears the error list.	U8	RW	See "CANopen Emergency Codes" on page 59 for emergency error codes.
		01h...10h	List of errors. Most recent error at top of list.	U32	RO	
1005h	COB-ID Sync	00h	ID of the sync message	U32	RW	-
1006h	Communication Cycle Period	00h	Communication cycle period	U32	RW	Only available if SYNC support is enabled
1007h	Synchronous Window Length	00h	Synchronous Window Length	U32	RW	Only available if SYNC support is enabled
1008h	Manufacturer device name	00h	The name of the CANopen module	Visible string	RO	"1 SI CANopen"
1009h	Manufacturer hardware version	00h	Manufacturer hardware version	Visible string	RO	Current hardware revision
100Ah	Manufacturer software version	00h	Manufacturer software version	Visible string	RO	Current software revision
100Ch	Guard time	00h	Used together with "Life time factor" to decide the node lifetime in (ms)	U16	RW	0000h (default)
100Dh	Life time factor	00h	If the node has not been guarded within its lifetime ("Life time factor"*"Guard time"), an error event is	U8	RW	00h (default)

© Siemens AG 2019 All rights reserved

3.2.3 Die Kommunikationsobjekte

Für die Kommunikation zwischen den CANopen Knoten stehen spezielle Kommunikationsobjekte (COB) aus dem Profilen DS 30x zur Verfügung.

Wie bei allen anderen Feldbus-Protokollen werden hierbei Echtzeitdaten und Parameterdaten unterschieden. CANopen ordnet diesen, vom Charakter her völlig unterschiedlichen Datenarten, jeweils passende Kommunikationsobjekte zu.

Prinzipiell werden zwischen den folgenden Kommunikationsobjekten unterschieden.

- Servicedatenobjekte (SDO) zur Übermittlung von Daten von und zum Objektverzeichnis (Parametrierdaten)
- Prozessdatenobjekte für den Austausch aktueller Prozesszustände (Echtzeitdaten)
- Objekte für das Netzwerkmanagement
- Objekte zur Steuerung der CAN-Nachrichten (Synchronisation und Fehlermeldungen)

Jedes Kommunikationsobjekt wird über einen Identifier (ID) eindeutig identifiziert.

Hinweis Dieser Abschnitt gibt nur einen kurzen Überblick und Beschreibung der Objekte. Detailliertere Informationen finden Sie in den folgenden Kapiteln.

© Siemens AG 2019 All rights reserved

Servicedatenobjekte

Servicedatenobjekte werden zur Modifikation des Objektverzeichnisses z. B. zur Parametrierung eines Gerätes beim Boot-Vorgang, für Statusabfragen oder zur Modifikation von Prozessdatenobjekten verwendet. Jeder CANopen Knoten verfügt über mindestens einem SDO Kanal, um auf Lese- oder Schreibanforderung eines anderen Knotens reagieren zu können.

Der Zugriff auf die zu modifizierenden Objekte erfolgt über den Index und Subindex. Die Werte können gelesen und – wenn erlaubt- auch beschrieben werden.

Prozessdatenobjekte

Prozessdatenobjekte werden für den Austausch von Echtzeit-Prozessdaten genutzt. Die schnelle Übertragung wird durch folgende Punkte realisiert:

- Telegramm wird vom Empfänger nicht bestätigt.
- Flexible Datenlänge: Die Telegrammlänge richtet sich nach den enthaltenen Nutzdaten.
- Übermittlung der Daten ohne zusätzlichen Overhead.
- Vorvereinbartes Datenformat
- PDOs haben gegenüber SDOs hochpriore Identifier

Die Prozessdatenobjekte nehmen im Objektverzeichnis die Bereiche 1400_h – $1A7F_h$ ein.

Abbildung 3-3

Index	Object Name	Sub-Index	Description	Type	Access
1400h ... 147Fh	Receive PDO parameter	00h	Largest sub-index supported	U8	RO
		01h	COB ID used by PDO	U32	RW
		02h	Transmission type	U8	RW
1600h ... 167Fh	Receive PDO mapping	00h	No. of mapped application objects in PDO	U8	RW
		01h	Mapped object #1	U32	RW
		02h	Mapped object #2	U32	RW
		03h	Mapped object #3	U32	RW
		04h	Mapped object #4	U32	RW
		05h	Mapped object #5	U32	RW
		06h	Mapped object #6	U32	RW
		07h	Mapped object #7	U32	RW
		08h	Mapped object #8	U32	RW
1800h ... 187Fh	Transmit PDO parameter	00h	Largest sub-index supported	U8	RO
		01h	COB ID used by PDO	U32	RW
		02h	Transmission type	U8	RW
		03h	Inhibit time	U16	RW
		05h	Event Timer (ms)	U16	RW
1A00h ... 1A7Fh	Transmit PDO mapping	00h	No. of mapped application objects in PDO	U8	RW
		01h	Mapped object #1	U32	RW
		02h	Mapped object #2	U32	RW
		03h	Mapped object #3	U32	RW
		04h	Mapped object #4	U32	RW
		05h	Mapped object #5	U32	RW
		06h	Mapped object #6	U32	RW
		07h	Mapped object #7	U32	RW
		08h	Mapped object #8	U32	RW

© Siemens AG 2019 All rights reserved

Wann und ob ein Producer eine Nachricht sendet/ empfängt, ist von der Übertragungsart der Sende-Prozessdatenobjekte (T_PDO) abhängig.

Mögliche Übertragungsarten sind:

- Ereignisgesteuerter Datentransfer
- Synchronisiert über das Synchronisationsobjekt SYNC
- Ereignisgesteuerte Synchronisation
- Anfrage durch einen Consumer

Für jedes T_PDO kann separat die Übertragungsart (transmission type) eingestellt werden.

Die Modifikation des Objektes erfolgt über ein SDO.

Abbildung 3-4

Index	Object Name	Sub-Index	Description	Type	Access
1800h ... 187Fh	Transmit PDO parameter	00h	Largest sub-index supported	U8	RO
		01h	COB ID used by PDO	U32	RW
		02h	Transmission type	U8	RW
		03h	Inhibit time	U16	RW
		05h	Event Timer (ms)	U16	RW

© Siemens AG 2019 All rights reserved

Netzwerkmanagementobjekte

Die Netzwerk-Management (NMT) Objekte übernehmen Aufgaben, die in zwei Gruppen unterteilt werden können:

- Dienste zur Gerätekontrolle, um
  - das Netzwerk und die Netzwerkknoten zu initialisieren.
  - die Betriebszustände der Knoten zu steuern.
- Dienste zur Verbindungsüberwachung der Knoten im Netzwerkbetrieb

Die folgende Grafik zeigt einen Ausschnitt aus dem Objektverzeichnis für die Netzwerkmanagementobjekte:

Abbildung 3-5

Index	Object Name	Sub-Index	Description	Type
1F80h	NMT Start-up	-	Defining whether the device is the NMT Master	U32
1F81h	Slave Assign-ment	ARRAY	Module list: Entry of all slaves to be managed, including guarding values and the entry of actions to be taken in event of guarding errors.	U32
1F82h	Request NMT	ARRAY	Remote control initiation of NMT services. For example, tools can use this to request intentional start/ stop of individual slaves. Remote query of the current state.	U8
1F83h	Request Guarding	ARRAY	Remote control start/stop of guard-ing. Remote query of the current state	U8
1F84h	Device Type Identification	ARRAY	Verify expected device types for the slaves	U32
1F85h	Vendor Identifi-cation	ARRAY	Verify vendor identifications for the slaves	U32
1F86h	Product Code	ARRAY	Verify product codes for the slaves	U32
1F87h	Revision Number	ARRAY	Verify revision numbers for the slaves	U32
1F88h	Serial Number	ARRAY	Verify expected serial numbers for the slaves	U32
1F89h	Boot Time	VAR	Maximum slave boot time before master indicates boot error and attempts to find the slave with LSS.	U32

© Siemens AG 2019 All rights reserved

Sonderobjekte

Neben den Objekten für die Service-, Prozessdaten und Netzwerkmanagement sind im Kommunikationsprofil auch Objekte für die Steuerung der CAN-Nachrichten spezifiziert:

- Synchronisationsobjekte zur Synchronisation der Netzwerkknoten und der Prozessdatenkommunikation.
- Emergency-Objekte zur Fehleranzeige eines Gerätes oder seiner Peripherie.

Die Synchronisation basiert auf zwei Zeitwerten:

- Die Zykluszeit definiert die Zeitspanne zwischen zwei Synchronisations-nachrichten und ist im Objekt "Communication Cycle Period" (1006_h) hinterlegt.
- Das Synchrone Fenster legt die Zeit fest, in der PDO Nachrichten empfangen und versendet werden müssen. Das entsprechende Objekt ist "Synchronous Window Lenght" (1007_h).

Abbildung 3-6

1006h	Communica-tion Cycle Period	00h	Communication cycle period	U32	RW
1007h	Synchronous Window Length	00h	Synchronous Window Length	U32	RW

Ob ein Gerät selbst aktiv Sync-Meldungen erzeugen kann oder nur passiv agiert, ist im Objekt "COB-ID Sync" (1005_h) einzustellen.

Abbildung 3-7

1005h	COB-ID Sync	00h	ID of the sync message	U32	RW

CANopen Fehlermeldungen werden über eine EMCY- Nachricht angezeigt. Für die Auswertung der Fehlerursache sind mehrere Möglichkeiten geboten:

- Objekt "Error register" (1001_h): Dieses 1-Byte große Datenfeld im Objektverzeichnis eines jeden Knoten zeigt den Fehlerzustand des Gerätes bitcodiert an.

Abbildung 3-8

1001h	Error register	00h	Error register, connected to the EMCY object. Bit 0 indi-cates a generic error	U8	RO	-

- Objekt "Pre-defined error field" (1003_h): Dieses Objekt liefert eine Fehlerhistorie mit maximal 10 Einträgen. Die möglichen Fehlercodes und –beschreibungen sind von CANopen spezifiziert.

Abbildung 3-9

1003h	Pre-defined error field	00h	Number of errors. Writing a 0 to this sub-index clears the error list.	U8	RW
		01h...10h	List of errors. Most recent error at top of list.	U32	RO

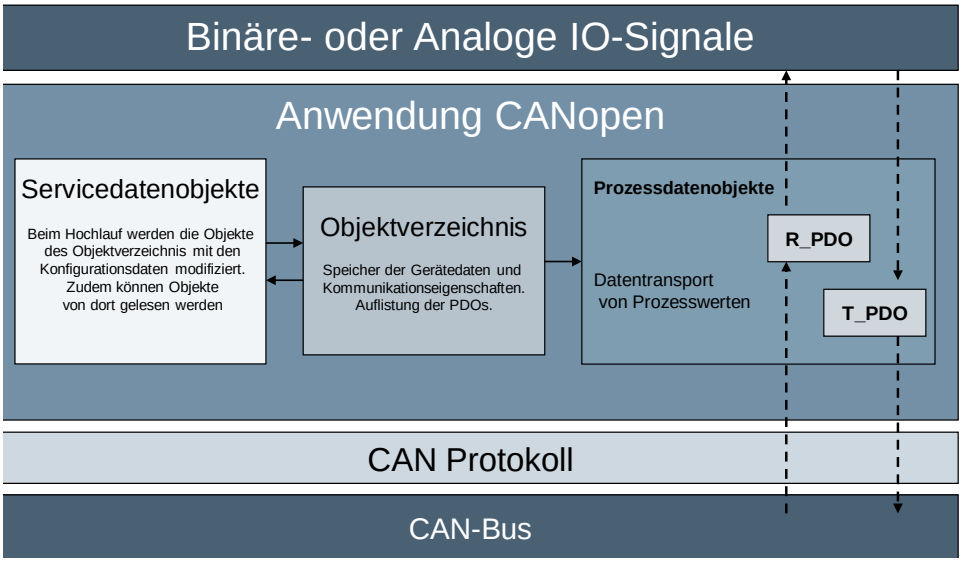
- Bis zu 5 Byte herstellerspezifische Fehlerinformation

© Siemens AG 2019 All rights reserved

Wirkungsweise der Kommunikationsobjekte

Die folgende Grafik zeigt eine schematische Darstellung eines CANopen Knotens und das Zusammenspiel der Kommunikationsobjekte:

Abbildung 3-10



© Siemens AG 2019 All rights reserved

3.2.4 Definierte Identifier (COB-IDs)

CANopen basiert bekanntlich auf CAN und nutzt zur Datenübertragung die CAN-Spezifikation. Die Telegramm-Rahmenstruktur von CAN wird dabei nicht verändert, da die CANopen Nachrichten in den Nutzdatenbereich des CAN Telegramms eingebettet werden. Lediglich das 11-Bit große Identifier Feld wird entsprechend der CANopen Spezifikation in der Bedeutung modifiziert: Die Adressierung des Knoten ist nun ein Bestandteil des Identifiers.

Die Identifier-Verteilung ist so ausgelegt, dass in einem CANopen-Netzwerk maximal 128 Geräte vorhanden sind: ein Master und bis zu 127 Slaves.

CANopen nutzt den 11-Bit große Identifier aus CAN 2.0A und gliedert diesen in zwei Teilen:

- dem Funktionscode (4 Bit) zur Identifizierung des Kommunikationsobjektes.
- der Knotenadresse (7 Bit) im Wertebereich [1..127].

Prinzipiell kann auch bei CANopen der Identifier frei gewählt werden. Um Konfigurationsaufwand einzusparen, legt die CANopen Spezifikation für die Kommunikationsobjekte vordefinierte Identifier fest.

Tabelle 3-2

Objekt	Index im OV	Identifier (COB-ID)		COB-ID (hex)
		Funktions-code	Knoten-adr.	
NMT –Funktion	-	0000	0000000	0_h
Synchronisation	1005 $_h$-1007 $_h$	0001	0000000	80_h
Emergency	1014$_h$,1015 $_h$	0001	xxxxxxx	80_h + Knotenadr. (81 $_h$ – FF $_h$)
T_PDO1	1800 $_h$	0011	xxxxxxx	180 $_h$ + Knotenadr. (181 $_h$ – 1FF $_h$)
R_PDO1	1400 $_h$	0100	xxxxxxx	200 $_h$ + Knotenadr. (201 $_h$ – 27F $_h$)
T_PDO2	1801 $_h$	0101	xxxxxxx	280 $_h$ + Knotenadr. (281 $_h$ – 2FF $_h$)
T_PDO2	1401 $_h$	0110	xxxxxxx	300 $_h$ + Knotenadr. (301 $_h$ – 37F $_h$)
T_PDO3	1802 $_h$	0111	xxxxxxx	380 $_h$ + Knotenadr. (381 $_h$ – 3FF $_h$)
R_PDO3	1402 $_h$	1000	xxxxxxx	400 $_h$ + Knotenadr. (401 $_h$ – 47F $_h$)
T_PDO4	1803 $_h$	1001	xxxxxxx	480 $_h$ + Knotenadr. (481 $_h$ – 4FF $_h$)
T_PDO4	1403 $_h$	1010	xxxxxxx	500 $_h$ + Knotenadr. (501 $_h$ – 57F $_h$)
T_SDO	-	1011	xxxxxxx	580 $_h$ + Knotenadr. (581 $_h$ – 5FF $_h$)
R_SDO	-	1100	xxxxxxx	600 $_h$ + Knotenadr. (60 1 $_h$ – 67F $_h$)
NMT Error Control		1110	xxxxxxx	700 $_h$ + Knotenadr. (701 $_h$ – 77F $_h$)

Die vordefinierte Identifier für die Objekte haben den entscheidenden Vorteil, dass die empfängerseitige Filterung der empfangenen CANopen Telegramme wesentlich einfach ist. Dadurch, dass die Identifier im Vorfeld bereits bekannt sind, kann jedem Gerät explizit über die Konfigurationssoftware von CANopen mitgeteilt werden, auf welche Telegramme dieser zu reagieren hat und bearbeiten muss.

© Siemens AG 2019 All rights reserved

3.2.5 Die Gerätedatenblätter EDS und DCF

Das EDS-File

Alle Funktionen und Eigenschaften eines CANopen Geräts werden vom Gerätehersteller in einem elektronischen Datenblatt (EDS) beschrieben. Diese Datei beinhaltet alle unterstützen Objekte, die Zugriffsmöglichkeiten (Lesen/ Schreiben) und Defaultwerte der Objekte, die Anzahl der PDOs, Baudraten, Herstellerangaben und viele andere Angaben des Gerätes. Allerdings ist die EDS lediglich eine Schablone für das Gerät, da keine Objektwerte enthalten sind.

Das DCF-File

Das Device Configuration File ist identisch zum EDS File aufgebaut, enthält aber zusätzlich die Werte für jedes Objekt aus dem Objektverzeichnis. Das DCF File wird nach der Konfiguration und Parametrierung der CANopen Knoten in der Konfigurationssoftware von derselbigen erstellt.

3.2.6 Kommunikationsbeziehungen

Übersicht

CANopen nutzt drei Beziehungen zwischen den Netzwerkknoten

- Master- Slave Beziehung
- Client- Server Beziehung
- Producer- Consumer Beziehung

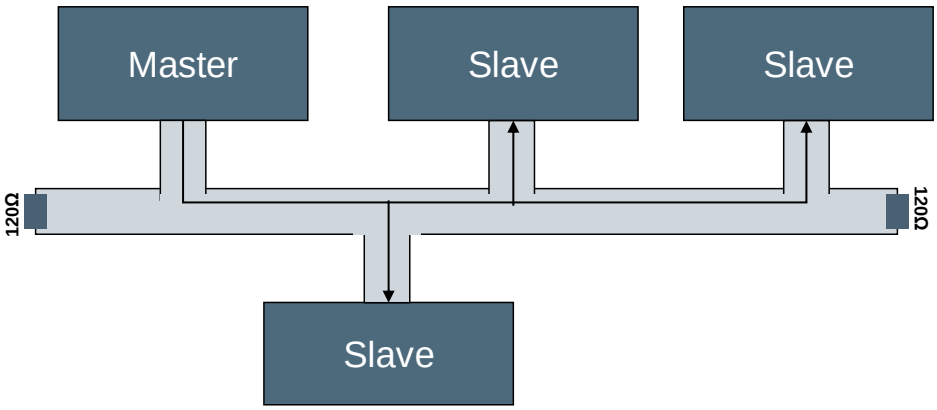
Master-Slave Beziehung

In einer Master-Slave Beziehung steuert ein Master den Nachrichtenverkehr, die Slaves antworten nur auf Anforderung des Masters. Der Nachrichtenaustausch kann unbestätigt oder bestätigt ausgeführt.

Eine unbestätigte Nachricht kann von allen, von einzelnen oder von keinem Teilnehmer empfangen werden.

Für eine bestätigte Nachricht fordert der Master von einem Slave eine Nachricht an. Der Slave beantwortet das Telegramm mit den geforderten Daten.

Abbildung 3-11

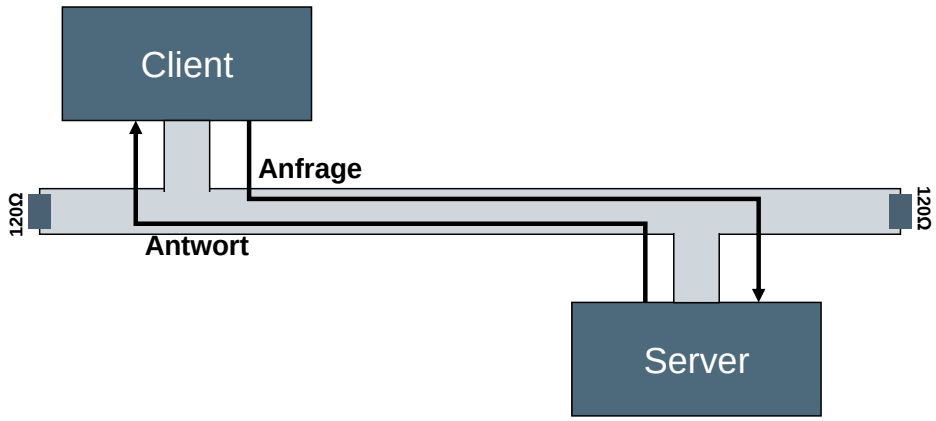


© Siemens AG 2019 All rights reserved

Client-Server Beziehung

Eine Client- Server Beziehung wird immer zwischen zwei Teilnehmern aufgebaut und ist bidirektional. Die Initiative für den Nachrichtenaustausch geht dabei immer vom Client aus.
Dieser stellt eine Anfrage beim Server und erwartet eine Bestätigung (diese enthält in der Regel die Antwortdaten). Eine Client-Server Beziehung hat demnach immer mindestens zwei Telegramme (Anfrage/ Antwort).

Abbildung 3-12



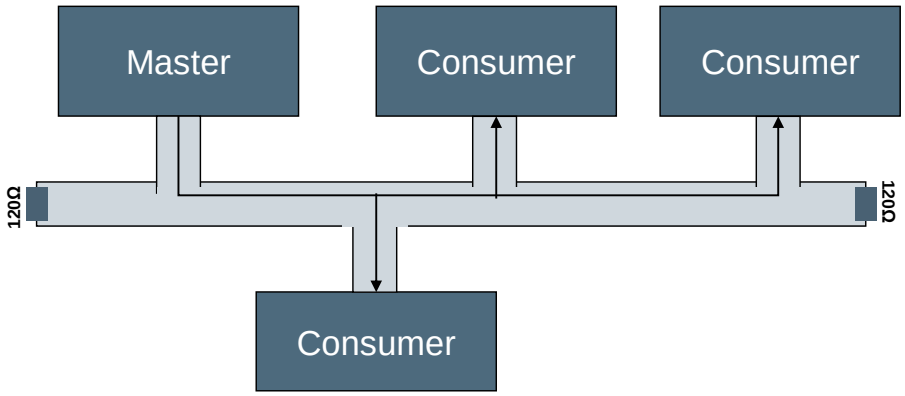
© Siemens AG 2019 All rights reserved

Producer-Consumer Beziehung

Eine Producer-Consumer Beziehung wird eingesetzt, wenn ein schneller Datenaustausch ohne Verwaltungsdaten verlangt wird.
Der Producer sendet ein Telegramm, das von einem oder mehreren Knoten (Consumer) empfangen werden kann.

Um die Busbandbreite nicht unnötig zu reduzieren, erfolgt die Datenübertragung unbestätigt.

Abbildung 3-13



3.3 Synchronisation des CANopen Netzwerks

Übersicht

CANopen ermöglicht das synchrone Abfragen von Eingänge und Zustände verschiedener Knoten sowie die Änderung von Ausgänge bzw. Zustände.
Hierzu dient das Synchronisationstelegramm (SYNC COB-ID: 80_h).
Das Sync-Telegramm ist ein Broadcast (keine Knotenadresse in der COB-ID) an alle Busteilnehmer mit hoher Priorität ohne Dateninhalt.

Das Sync-Telegramm wird von einem Busteilnehmer (in der Regel vom Master) zyklisch in festen Intervallen (Kommunikationszyklus) versandt.

Baugruppen, die im synchronisierten Modus arbeiten, lesen ihre Prozessdaten-eingänge bei Empfang der Sync-Nachricht aus und senden die Daten direkt anschließend, wenn der Bus frei ist.
Ausgangsdaten werden erst nach dem nächsten Sync-Telegramm zu den Ausgängen geschrieben (bzw. ausgeführt).

SYNC-Telegramm

Die Synchronisation des Netzwerks erfolgt über eine Producer-Consumer Beziehung.

Das Sync-Telegramm überträgt selbst keine Daten und nutzt das "blanke" CAN Telegramm.

Abbildung 3-14

Bits	1	12/32	6	0-64	16	2	7
Feld	Start	Arbitrierung	Kontrolle	Daten	Sicherung	Bestätigung	Ende

© Siemens AG 2019 All rights reserved

3.4 Fehlererkennung bei CANopen

Übersicht

Emergency Nachrichten werden immer abgesetzt, wenn im Gerät eine kritische Fehlersituation aufgetreten/behoben ist, bzw. wichtige Informationen anderen Geräten mitgeteilt werden müssen. Die Emergency-Botschaft wird von jedem CANopen-Gerät selbstständig gesendet.

Für die Emergency Telegramme wurde die COB-ID: 80_h + Knotenadresse definiert.

Das Emergency-Telegramm enthält ein Code, der den Fehler eindeutig identifiziert (definiert im Kommunikationsprofil DS-301 sowie in den jeweiligen Geräteprofilen DSP-40x).

Die Tabelle zeigt einen Auszug der verfügbaren Fehlercodes-Gruppen.

Tabelle 3-3

Code (hex)	Bedeutung
00xx	Kein Fehler
10xx	Nicht definierter Fehlertyp
20xx	Stromfehler
30xx	Spannungsfehler
40xx	Temperaturfehler
50xx	Fehler an der Hardware
60xx	Fehler in der Software
70xx	Zusatzmodule
80xx	Kommunikation
90xx	Externer Fehler
FF00	Gerätespezifisch

© Siemens AG 2019 All rights reserved

Parallel zur Übertragung des Fehlertelegramms hinterlegt jeder Knoten den Fehlercode im Objekt 1003_h seines Objektverzeichnisses (siehe Abbildung 3-9) sowie bitweise codiert in das Objekt 1001_h.

Tabelle 3-4

Bit	Fehlerursache
0	Generic Error
1	Current
2	Voltage
3	Temperature
4	Communication Error
5	Device Profile Specific
6	Reserved (always 0)
7	Manufacturer Specific

EMCY-Telegramm

Das Versenden der Emergency Telegramme erfolgt entsprechend der Consumer-Provider Beziehung.

Emergency Nachrichten werden immer abgesetzt, wenn im Gerät eine Fehlersituation aufgetreten oder behoben ist, bzw. wichtige Informationen anderen Geräten mitgeteilt werden müssen.

Die Fehlereinträge sind im Bus-Telegramm wie folgt codiert:

Abbildung 3-15

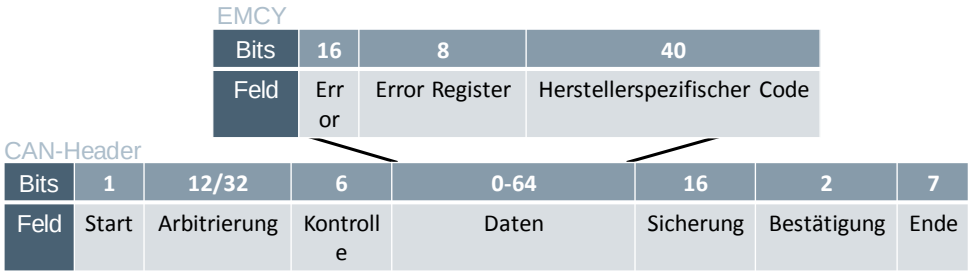


Tabelle 3-5

Abschnitt	Beschreibung
Error Code	Der Error-Code beinhaltet Informationen über die Fehlerursache und ist im Kommunikationsprofil DS-301 sowie in den jeweiligen Geräteprofilen DSP-40x definiert. Siehe auch Tabelle 3-3.
Error Register	Dieses 1-Byte große Datenfeld zeigt den Fehlerzustand des Gerätes bitcodiert an. Dieser Wert ist auch im Objekt "Error Register" (1001_h) eines jeden Knoten gespeichert.
Herstellerspezif. Code	Herstellerspezifische Fehlerinformationen.

© Siemens AG 2019 All rights reserved

3.5 Die Servicedatenkommunikation

Übersicht

Mit Hilfe der Servicedaten-Objekten

- T_SDO: COB-ID: 580_h + Knotenadresse
- R_SDO: COB-ID: 600_h + Knotenadresse

kann auf die Einträge des Objektverzeichnisses einen anderen Netzwerkknoten zugegriffen werden und die Werte der Objekte gelesen und, wenn möglich, auch modifiziert werden. Welchen Eintrag dabei referenziert wird, ist im Servicedatentelegramm durch die Zugabe des entsprechenden Index und Subindex hinterlegt.

Servicedatenobjekte dienen überwiegend zur Parametrierung und Konfiguration der Geräte.

Mit dem T_SDO wird die Anforderung zum Datenaustausch gesendet, mit dem R_SDO empfangen. Der Datenrahmen eines SDO beträgt 8 Byte.

SDO-Telegramm

Die Servicedatenkommunikation erfolgt über eine Client-Server Beziehung zwischen zwei Teilnehmern und ist immer asynchron.

Der SDO Datenrahmen beinhaltet neben der Adressierung des Objektes im Objektverzeichnis (16 Bit Index + 8 Bit Subindex) ein Domainprotokoll und ein 4-Byte großes Nutzdatenfeld.

Abbildung 3-16

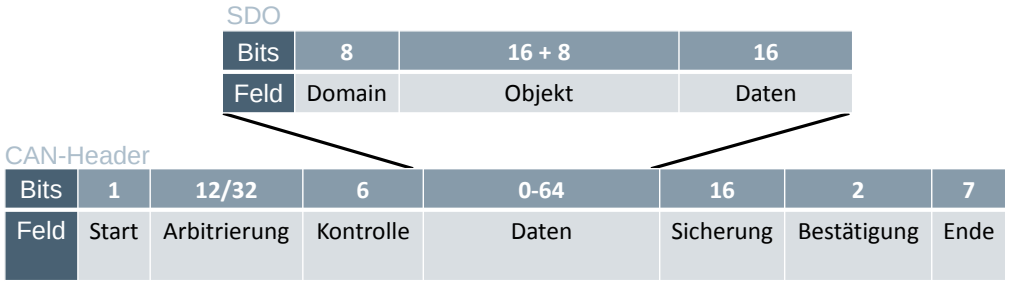


Tabelle 3-6

Abschnitt	Untergruppe	Beschreibung
Domain Protokoll		Der Befehls-Code beinhaltet Informationen, welche Aktion auf den adressierten Parameter erfolgen soll und wie lange der übermittelte Wert ist.
Objektadresse		
	Index	Der Index besteht aus einer 16-Bit großen vierstellige Hexadezimalzahl und adressiert ein Objekt aus dem Objektverzeichnis.
	Sub-Index	Besteht ein Objekt aus mehreren Unterkategorien, werden diese über den Sub-Index- einer 8-Bit großen Hexadezimalzahl- adressiert.
Datenfeld		Im Datenfeld werden die Nutzdaten übertragen. Pro Telegramm können bis zu 4 Byte übertragen werden. In der Regel enthält das Datenfeld die Werte, die auf den adressierten Parameter geschrieben werden sollen.

© Siemens AG 2019 All rights reserved

3.6 Die Prozessdatenkommunikation

Übersicht

Prozessdaten-Objekten werden für den Echtzeitdatenaustausch von Prozess- und Betriebszuständen genutzt. Das Kommunikationsprofil legt für je 4 Empfangs- und Sendekanäle die COB-IDs fest (siehe Tabelle 3-2). Sollte das Gerät mehr Kanäle unterstützen (z. B. das 1 SI Modul mit 128 Empfangs- und Sendekanäle), werden die fehlenden COB-IDs durch den Hersteller definiert.

PDO-Nachrichten können zwischen Netzwerkknoten ausgetauscht werden, die Prozessdaten erzeugen oder verarbeiten..

Übertragungsarten

Der PDO-Datenaustausch erfolgt entweder synchron oder asynchron.

Synchronisierte Übertragung

Die synchrone Datenübertragung von PDOs geschieht in Relation zu dem SYNC Objekt (siehe Kapitel 0

Synchronisation des CANopen Netzwerks).

Wird das SYNC Objekt empfangen, werden die Eingänge des Gerätes gelesen und anschließend gesendet. Empfangene Ausgangsdaten werden erst beim nächsten SYNC geschrieben.

Der synchrone Nachrichtenaustausch kann zyklisch (unterteilte Synchronisation) oder azyklisch (Ereignisgesteuerte Synchronisation) ausgeführt werden.

Asynchrone Übertragung

Die asynchrone Datenübertragung erfolgt unabhängig vom SYNC Objekt. Die Daten eines Knotens werden verschickt, sobald ein Event auftritt (z. B. Datenänderung) oder der Knoten eine externe Datenanfrage erhalten hat. Ein solcher Remote Request wird durch das RTR-Bit (siehe

Tabelle 2-2) im Identifier Feld des CAN Telegrammrahmens angezeigt.

© Siemens AG 2019 All rights reserved

PDO- Telegramm

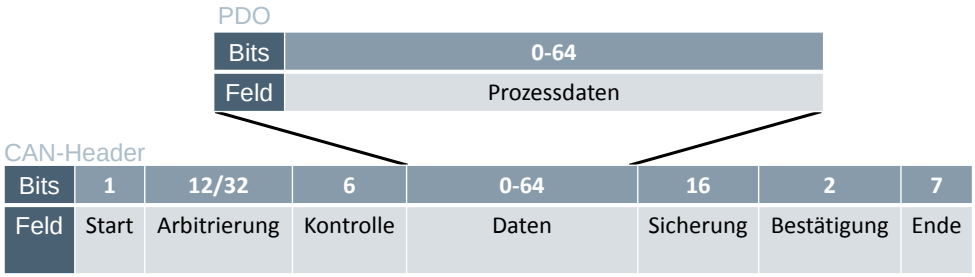
Um die Busbandbreite für die Echtzeitdatenübertragung nicht unnötig zu reduzieren, erfolgt die Datenübertragung durch eine Producer-Consumer Beziehung und somit unbestätigt.

Durch die empfängerseitige Filterung anhand der zu erwartenden COB-ID entscheidet jeder Netzteilnehmer, ob das Telegramm für ihn relevant oder irrelevant ist.

Die Filterung der Telegramme wird dem Gerät über die Konfigurationssoftware von CANopen mittels PDO-Mapping mitgeteilt (siehe Kapitel 3.7 Das PDO Mapping).

Eine Prozessdatennachricht hat keine zusätzliche Verwaltungsinformationen sondern nutzt das "blanke" CAN-Telegramm inklusiv der maximal möglichen 8-Byte Nutzdatenfeld von CAN.

Abbildung 3-17



Die Datenlänge einer PDO-Nachricht ist flexible: Sind nur 2 Byte belegt, werden auch nur 2 Datenbyte übertragen. Weil keine "Leer-Bytes" übertragen werden, ist der Datendurchsatz am Bus größer.

Durch die fehlenden Protokollinformationen seitens CANopen im Telegramm muss das Datenformat zwischen Producer und Consumer vereinbart werden. Dies erfolgt durch das sogenannte PDO-Mapping (siehe Kapitel 3.7 Das PDO Mapping).

© Siemens AG 2019 All rights reserved

3.7 Das PDO Mapping

Aufgabe

Das PDO Mapping wird mit Hilfe der CANopen Konfigurationssoftware für jeden Netzwerkknoten generiert und erfüllt mehrere Aufgaben:

- Es definiert die Datenformate für das 8-Byte große Datenfeld im PDO Objekt.
- Es definiert die Datenbelegung des 8-Byte großen Datenfeld
- Es legt fest, welche PDO Telegramme ein Knoten auf den Bus schickt und welche Daten darin enthalten sind.
- Es informiert, mit welchen Telegrammen der Knoten zu rechnen hat.

Für jedes PDO Objekt existiert ein korrespondierendes PDO Mapping Objekt, wo diese Festlegungen hinterlegt sind (siehe Abbildung 3-3).

Tabelle 3-7

Objekt	Index	Beschreibung
Receive PDO parameter [0..127]	1400_h – $147F_h$	Alle Telegramme, auf die ein Knoten reagieren und bearbeiten soll, werden durch dieses Objekt beschrieben. Insgesamt stehen 128 Empfangs-PDOs (R_PDO) für demnach 128 Telegramme zur Verfügung. Jedes R_PDO Objekt speichert durch die Sub-Indizes Einstellungen über das zu erwartende Telegramm ab. Dazu gehören Informationen wie: • COB-ID des zu empfangene Telegramm • Übertragungstyp • Verzögerungszeit
Receive PDO Mapping [0..127]	1600_h – $167F_h$	Zu jedem R_PDO korrespondiert ein weiteres Objekt, welches genau definiert, wie die 8-Byte Nutzdaten zu verstehen sind und welche Datenformate im R_PDO enthalten sind. Jedes Mapping-Objekt hat neun Sub-Indizes: Im ersten Eintrag (Sub[0]) steht die Anzahl der gemappten Objekte, Sub[1]-Sub[8] repräsentieren die möglichen Eintrage (8 * 1 Byte für das 8-byte große Nutzdatenfeld).
Transmit PDO parameter [0..127]	1800_h – $187F_h$	Alle Telegramme, die ein Knoten versendet, werden durch dieses Objekt beschrieben. Insgesamt stehen 128 Sende-PDOs (T_PDO) für demnach 128 Telegramme zur Verfügung. Jedes T_PDO Objekt speichert durch die Sub-Indizes Einstellungen, wie das Telegramm zu versenden ist. Dazu gehören Informationen wie: • COB-ID des zu sendende Telegramm • Übertragungstyp • Verzögerungszeit
Transmit PDO Mapping [0..127]	$1A00_h$ – $1A7F_h$	Zu jedem T_PDO korrespondiert ein weiteres Objekt, welches genau definiert, welche Datenformate und Daten mit dem T_PDO zu übertragen sind. Die Datenformate In Summe können pro PDO 8-Byte Daten verschickt werden. Jedes Mapping-Objekt hat neun Sub-Indizes: Im ersten Eintrag (Sub[0]) steht die Anzahl der gemappten Objekte, Sub[1]-Sub[8] repräsentieren die möglichen Eintrage (8 * 1 Byte für das 8-byte große Nutzdatenfeld).

© Siemens AG 2019 All rights reserved

Des Weiteren sind für das PDO Mapping auch die Bereiche des Objektverzeichnisses von Bedeutung, welche die vom Gerät unterstützten Datentypen definieren.

Im 1SI Modul sind diese im Herstellerspezifischen Profil hinterlegt:

Abbildung 3-18

Index	Sub-Index	Type	Access	Name and description	Comment
2000h - 201Fh	-	STRUCT		Generic Transmit Object #	
	0	U8	RO	Largest sub-index supported	
	1..32	U8	RO	Generic Byte Transmit Object 1..32	
	33..48	U16	RO	Generic Word Transmit Object 1..16	Same data as sub-index 1..32
	49..56	U32	RO	Generic Long Transmit Object 1..8	Same data as sub-index 1..32
2100h - 211Fh	-	STRUCT		Generic Receive Object #	
	0	U8	RO	Largest sub-index supported	
	1..32	U8	RW	Generic Byte Receive Object 1..32	
	33..48	U16	RW	Generic Word Receive Object 1..16	Same data as sub-index 1..32
	49..56	U32	RW	Generic Long Receive Object 1..8	Same data as sub-index 1..32
3000h	0	U8	RW	Swap data to big endian	0: Little endian (CANopen style, default) 1: Swap data to big endian, see "Swap Data to Big Endian (3000h)" on page 43

© Siemens AG 2019 All rights reserved

Arten von PDO Mapping

Prinzipiell werden zwei Arten des PDO Mappings unterschieden:

- Dynamisches Mapping: Die 8-Byte Nutzdaten der PDO Objekte können flexible formatiert und die Daten je nach Anwendung unterschiedlich zusammengestellt werden.
- Statisches Mapping: In diesem Fall sind die Datenformate für jedes PDO Objekt bereits vom Hersteller vordefiniert und können nicht verändert werden.

Prinzip

Im Folgenden wird das PDO Mapping anhand eines Beispielszenarios verdeutlicht.

Knoten 1 möchte an Knoten 2 folgende seiner Werte aus dem Prozessabbild mitteilen:

Tabelle 3-8

	Consumer Node 2						
Producer Node 1	Byte 0	Byte 1	Byte 2	Byte 3	Word 0	Word 1	Long 0
Byte 0		■					
Byte 1	■						
Byte 2							
Byte 3							
Long 0							■
Word 0					■		
Word 1							

In Summe sollen 8 Byte übertragen werden; ein T_PDO ist für die Datenübertragung ausreichend.

Die folgende Grafik verdeutlicht das Prinzip des PDO Mappings im Objektverzeichnis des Producers:

Abbildung 3-19

Index	Subindex	Beschreibung
1600_h		T_PDO_0
	0	= 4 (Anzahl der Einträge)
	1	= 181_h (COB_ID: 180_h + Knotenadresse
	...	
$1A00_h$		T_PDO_0 Mapping
	0	= 4 (Anzahl der Mapping Einträge)
	1	= 2000 31 20_h (Index 2000_h, Subindex 31_h, 20_h Bit)
	2	= 2000 21 10_h (Index 2000_h, Subindex 21_h, 10_h Bit)
	3	= 2000 01 08_h (Index 2000_h, Subindex 01_h, 08_h Bit)
	4	= 2000 02 08_h (Index 2000_h, Subindex 02_h, 08_h Bit)
2000_h		Input Data puffer
	0	= 56 (Anzahl der Objekte)
	1	Input Data puffer Byte 0
	2	Input Data puffer Byte 1
	...	
	33	Input Data puffer Word 0 (≙Byte 0 & Byte 1)
	...	
	49	Input Data puffer Long 0 (≙Byte 0 - Byte 3)

T_PDO_0

COB ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
181_h	Long 0				Word 0		Byte 0	Byte 1

© Siemens AG 2019 All rights reserved

3.8 Das Netzwerkmanagement

Übersicht

Die Netzwerk-Management (NMT) Objekte übernehmen Aufgaben, die in zwei Gruppen unterteilt werden können:

- Dienste zur Gerätekontrolle, um
  - das Netzwerk und die Netzwerkknoten zu initialisieren.
  - die Betriebszustände der Knoten zu steuern.
- Dienste zur Verbindungsüberwachung der Knoten im Netzwerkbetrieb
  - Node Guarding-Funktion
  - Heartbeat-Funktion

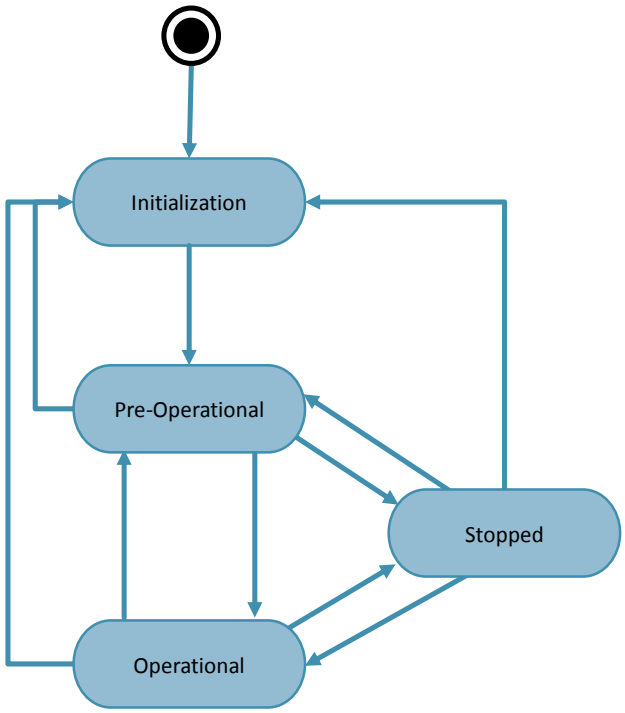
3.8.1 Kontrolle der CANopen Geräte

Netzwerkmanagement- Zustandsmaschine

CANopen Geräte können während des Betriebs verschiedene Zustände einnehmen. Je nach Zustand sind nur bestimmte Funktionen bzw. Kommunikationsobjekte einsetzbar.

Die Netzwerkmanagement- Zustandsmaschine gibt eine Übersicht der möglichen Betriebsstatus und verdeutlicht die Zusammenhänge:

Abbildung 3-20



© Siemens AG 2019 All rights reserved

Tabelle 3-9

Zustand	Beschreibung	Einsetzbare Objekte
Initialization	Nach dem Einschalten befinden sich alle Geräte automatisch in dieser Phase und werden für den CAN-Busbetrieb vorbereitet.	
Pre-Operational	Einmal initialisiert, nehmen die Geräte den Pre-Operational Status ein und senden eine "Boot-Up" Nachricht an den NMT-Master. Ab diesem Zeitpunkt kann der Master das Verhalten des Gerätes steuern. Der Zustand Pre-Operational kann zur Konfiguration per SDOs genutzt werden: - PDO-Mapping - Start der Synchronisation - Start der Verbindungsüberwachung	SDO, EMCY, NMT
Operational	Operational ist der normale Busbetriebsmodus. Die Geräte sind voll funktionsfähig.	PDO, SDO, SYNC, EMCY, NMT
Stopped	Im Stoppzustand findet keine Kommunikation mehr statt. Eventuell konfigurierte Verbindungsüberwachungen bleiben aber aktiv.	NMT

Die Geräte ändern Ihren Betriebszustand

- nach Aufforderung durch ein NMT-Objekt.
- bei einem Hardware Reset.
- initiiert durch einen Gerätekontroll-Dienst.

© Siemens AG 2019 All rights reserved

Initialisierung des Netzwerks

Um einen kontrollierten Netzwerkstart zu garantieren und zur Überwachung der Verbindungen der Knoten wird die Initialisierung der Geräte sowie die Steuerung der Betriebsmodi in einer Master-Slave-Beziehung ausgeführt.

Diese Dienste werden unidirektional mit der COB-ID 0 übertragen und erhalten somit die höchste Priorität auf dem Bus.

Der Datenrahmen besteht aus zwei Byte.

Abbildung 3-21

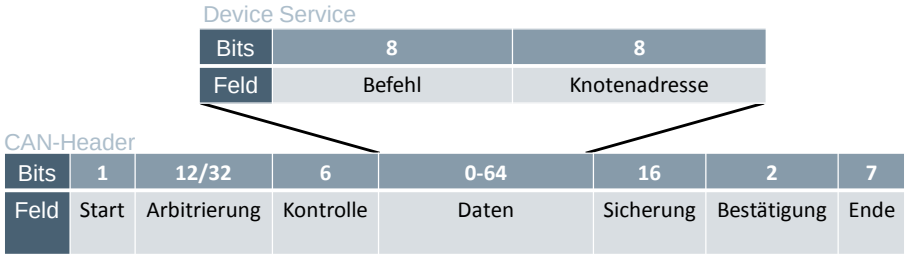


Tabelle 3-10

Abschnitt	Beschreibung
Befehl	Das "Command Spezifier" Feld definiert den zu verwendenden NMT Dienst. Mögliche Werte sind: 01_h Netzknoten starten 02_h Netzknoten stoppen 80_h Auf "Pre-Operational" wechseln 81_h Knoten rücksetzen 82_h Kommunikation rücksetzen
Knotenadr	Dieses 1-Byte große Feld adressiert den Empfänger der NMT-Nachricht mit der Knotenadresse. Eine Nachricht mit Knotenadresse 0 richtet sich an alle NMT-Slaves.

© Siemens AG 2019 All rights reserved

3.8.2 Überwachungsfunktionen

CANopen stellt Überwachungsdienste für die Netzverbindungen bereit, damit bei Ausfall eines Teilnehmers oder bei Unterbrechung des Netzwerks entsprechend reagiert werden kann. Zur Sicherung der Kommunikation stehen folgende Mechanismen zur Verfügung:

- Guarding
  - Nodeguarding (Master)
  - Lifeguarding (Slave)
- Heartbeat

Ein CANopen Knoten muss mindestens eine Überwachungsfunktion unterstützen.

Überwachungstelegramme sind an der COB-ID 700_h + Knotenadresse erkennbar.

Guarding Funktion

Beim Nodeguarding unterliegt es der Verantwortung des Masters, zyklisch NMT-Statusmeldungen von den Slaves anzufordern. Antwortet ein Slave nicht innerhalb einer definierten Zeit oder sendet einen nicht erwartenden Betriebszustand, erkennt der Master einen Fehler und entsendet ein EMCY-Objekt.

Slaves, die das Lifeguarding unterstützen, können auch das zyklische Abfragetelegramm vom Master überwachen. Wird innerhalb einer definierten Zeit kein Telegramm vom Master empfangen, erkennt der Slave einen Fehler und sendet ein Fehlertelegramm.

Die Zeiten werden über die Netzwerkmanagement-Objekte "Guard Time" ($100C_h$) und "Life time factor" ($100D_h$) konfiguriert.

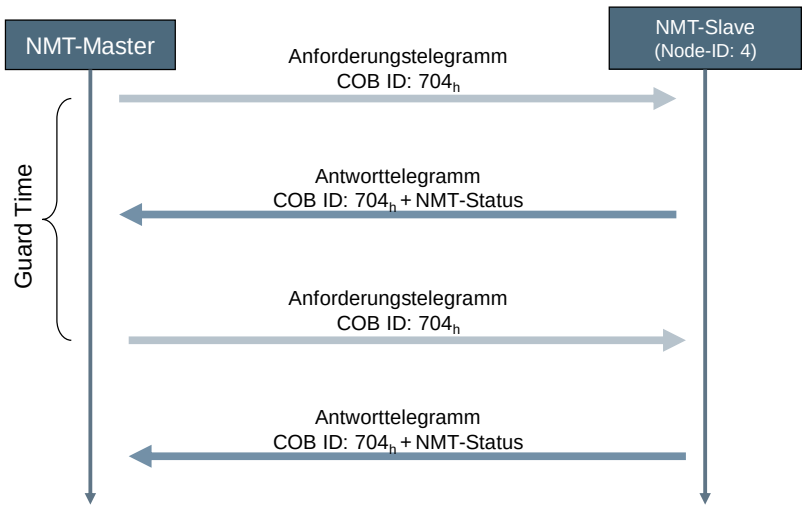
Abbildung 3-22

100Ch	Guard time	00h	Used together with "Life time factor" to decide the node life-time in (ms)	U16	RW	C
100Dh	Life time factor	00h	If the node has not been guarded within its lifetime ("Life time factor"*"Guard time"), an error event is logged and a remote node error is indicated	U8	RW	C

Guarding-Telegramme werden immer bidirektional nach dem Master-Slave Prinzip verschickt.

Ohne Lifeguarding wird ein Ausfall des NMT-Masters von den NMT-Slaves nicht erkannt.

Abbildung 3-23



© Siemens AG 2019 All rights reserved

Heartbeat Funktion

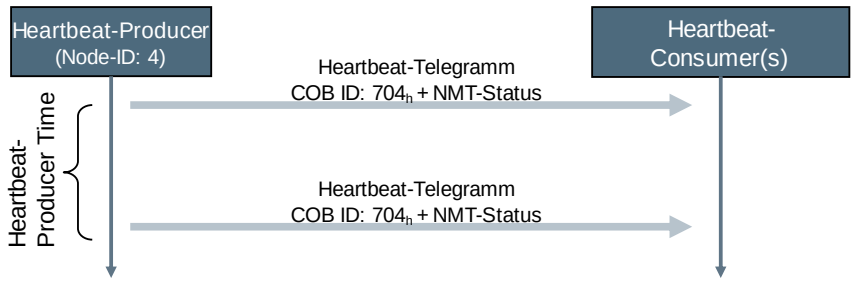
Beim Heartbeat senden die Knoten (NMT-Master und NMT-Slaves) ihren derzeitigen Betriebsstatus selbstständig und zyklisch an alle anderen Busteilnehmer. Eine explizite Anforderung über den NMT-Master ist nicht erforderlich. Die Zykluszeit ist für jeden Knoten individuell über das Objekt 1017_h einstellbar.

Abbildung 3-24

1017h	Producer Heartbeat Time	00h	Defines the cycle time of the heartbeat. Not used if 0	U16	RW

Die Busteilnehmer überwachen für sich die von den anderen Knoten eintreffenden Heartbeat-Telegramme und melden einen Ausfall, wenn ein Telegramm ausbleibt.

Abbildung 3-25



Heartbeat wird heutzutage dem Nodeguarding vorgezogen, da es durch das fehlende Anforderungstelegramm weniger Buslast erzeugt und zudem ein Ausfall des NMT-Masters ebenfalls von den NMT-Slaves erkannt wird.

© Siemens AG 2019 All rights reserved

4 Literaturhinweise

Tabelle 4-1

	Themengebiet	Titel
\1\	Siemens Industry Online Support	http://support.automation.siemens.com
\2\	Downloadseite des Beitrages	https://support.industry.siemens.com/cs/ww/de/view/109479771https://support.industry.siemens.com/cs/ww/de/view/109479771
\3\		

5 Historie

Tabelle 5-1

Version	Datum	Änderung
V1.0	10/2015	Erste Ausgabe
V2.0	07/2019	Aktualisierung für TIA Portal V15.1

© Siemens AG 2019 All rights reserved