



CANopen Dokumentation

ANLEITUNG ZUR CANOPEN-SCHNITTSTELLE

TEST-01

Inhalt

1	Einleitung.....	4
1.1	Versionshinweis	4
1.2	Gewährleistung und Haftungsausschluss	Fehler! Textmarke nicht definiert.
1.3	Zielgruppe und Qualifikation	Fehler! Textmarke nicht definiert.
1.4	Sicherheits- und Warnhinweise	Fehler! Textmarke nicht definiert.
2	CANopen.....	5
2.1	CAN-Topologie.....	5
2.2	Buslänge	5
2.3	Stichleitungen	6
2.4	Sternverteiler (Multiport Tap)	6
2.5	CAN-Kabel	6
2.6	CANopen-Schnittstelle	6
2.7	Gerätemodell	7
2.8	Kommunikationsarten.....	7
2.9	Network management (NMT).....	7
2.10	Service Data Object (SDO).....	7
2.11	Process Data Object (PDO).....	8
2.12	Synchronization Object (SYNC).....	8
2.13	Time Stamp Object (TIME)	8
2.14	Emergency Object (EMCY)	8
2.15	CANopen-Configuration	9
2.16	Object Dictionary	Fehler! Textmarke nicht definiert.
	Object: 0x1000 Device Type.....	Fehler! Textmarke nicht definiert.
	Object: 0x1001 Error Register	Fehler! Textmarke nicht definiert.
	Object: 0x1003 Pre-defined error field	Fehler! Textmarke nicht definiert.
	Object: 0x1008 Manufacturer device name.....	Fehler! Textmarke nicht definiert.
	Object: 0x1010 Store parameters	Fehler! Textmarke nicht definiert.
	Object: 0x1011 Restore default parameters.....	Fehler! Textmarke nicht definiert.
	Object: 0x1014 COB ID EMCY	Fehler! Textmarke nicht definiert.
	Object: 0x1015 Inhibit Time Emergency.....	Fehler! Textmarke nicht definiert.
	Object: 0x1017 Producer Heartbeat Time	Fehler! Textmarke nicht definiert.
	Object: 0x1018 Identity Object.....	Fehler! Textmarke nicht definiert.
	Object: 0x1029 Error behaviour	Fehler! Textmarke nicht definiert.
	Object: 0x1030 Version information	Fehler! Textmarke nicht definiert.
	Object: 0x1032 Active error list.....	Fehler! Textmarke nicht definiert.
	Object: 0x1200 SDO server parameter	Fehler! Textmarke nicht definiert.
	Object: 0x1400 Receive PDO Communication Parameter.....	Fehler! Textmarke nicht definiert.

Object: 0x1600 Receive PDO Mapping Parameter **Fehler! Textmarke nicht definiert.**

Object: 0x1800 Transmit PDO Communication Parameter..... **Fehler! Textmarke nicht definiert.**

Object: 0x1a00 Transmit PDO Mapping Parameter..... **Fehler! Textmarke nicht definiert.**

Object: 0x1f50 Download Program Data **Fehler! Textmarke nicht definiert.**

Object: 0x1f51 Program Control **Fehler! Textmarke nicht definiert.**

Object: 0x1f57 Flash Status Identification **Fehler! Textmarke nicht definiert.**

Object: 0x2000 Hardware Parameter..... **Fehler! Textmarke nicht definiert.**

Object: 0x2001 Scaling Parameter **Fehler! Textmarke nicht definiert.**

Object: 0x2010 Motor Parameter..... **Fehler! Textmarke nicht definiert.**

Object: 0x2011 Brake Parameter..... **Fehler! Textmarke nicht definiert.**

Object: 0x2012 Angular Sensor Parameter..... **Fehler! Textmarke nicht definiert.**

Object: 0x2013 Encoder Sensor **Fehler! Textmarke nicht definiert.**

Object: 0x2014 Hall Sensor **Fehler! Textmarke nicht definiert.**

Object: 0x2015 Sin/Cos Sensor..... **Fehler! Textmarke nicht definiert.**

Object: 0x2020 Voltage Limits **Fehler! Textmarke nicht definiert.**

Object: 0x2021 Current Limits **Fehler! Textmarke nicht definiert.**

Object: 0x2022 Temperature Limits **Fehler! Textmarke nicht definiert.**

Object: 0x2023 Control Error Limits **Fehler! Textmarke nicht definiert.**

Object: 0x2024 I2t **Fehler! Textmarke nicht definiert.**

Object: 0x2030 Current Controller **Fehler! Textmarke nicht definiert.**

Object: 0x2031 Speed Controller **Fehler! Textmarke nicht definiert.**

Object: 0x2032 Position Controller..... **Fehler! Textmarke nicht definiert.**

Object: 0x2033 Homing **Fehler! Textmarke nicht definiert.**

Object: 0x2040 Control Settings **Fehler! Textmarke nicht definiert.**

Object: 0x2041 Speed Trajectory..... **Fehler! Textmarke nicht definiert.**

Object: 0x2042 Position Trajectory **Fehler! Textmarke nicht definiert.**

Object: 0x5000 CANopen Configuration **Fehler! Textmarke nicht definiert.**

Object: 0x6040 controlword **Fehler! Textmarke nicht definiert.**

Object: 0x6041 statusword **Fehler! Textmarke nicht definiert.**

Object: 0x6042 vl_target_velocity **Fehler! Textmarke nicht definiert.**

Object: 0x6043 vl_velocity_demand **Fehler! Textmarke nicht definiert.**

Object: 0x6044 vl_velocity_actual_value..... **Fehler! Textmarke nicht definiert.**

Object: 0x6046 vl_velocity_min_max_amount **Fehler! Textmarke nicht definiert.**

Object: 0x6048 vl_velocity_acceleration **Fehler! Textmarke nicht definiert.**

Object: 0x6049 vl_velocity_deceleration **Fehler! Textmarke nicht definiert.**

Object: 0x6060 modes_of_operation **Fehler! Textmarke nicht definiert.**

Object: 0x6061 modes_of_operation_display..... **Fehler! Textmarke nicht definiert.**

Object: 0x6064 position_actual_value..... **Fehler! Textmarke nicht definiert.**

Object: 0x6079 dc_link_circuit_voltage **Fehler! Textmarke nicht definiert.**

1 Einleitung

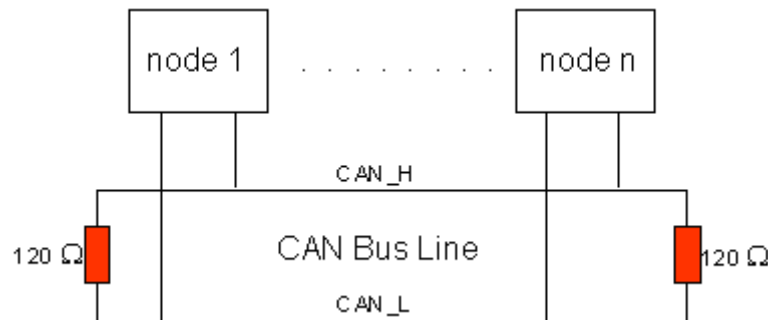
1.1 Versionshinweis

Version Handbuch	Datum	Änderungen	Version Firmware
0.1	15.11.2024	Erste Rohfassung	2.01.6

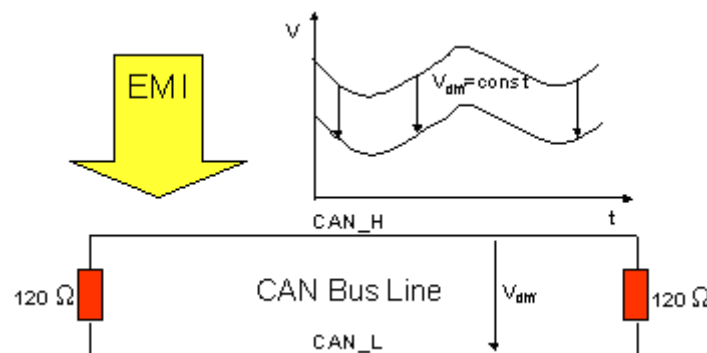
2 CANopen

2.1 CAN-Topologie

CAN ist ein 2-Draht-Bussystem, an dem alle Teilnehmer parallel angeschlossen werden. Um Reflexionen zu vermeiden, muss der Bus an jedem Ende mit einem Abschlusswiderstand von 120 Ohm angeschlossen werden. Dies ist auch bei sehr kurzen Leitungslängen erforderlich.



Da die CAN-Signale als Differenzpegel auf dem Bus dargestellt werden, ist die CAN-Leitung vergleichsweise unempfindlich gegen eingeprägte Störungen (EMI). Es sind jeweils beide Leitungen betroffen, somit verändert die Störung den Differenzpegel kaum.



2.2 Buslänge

Die maximale Buslänge wird bei CAN vorwiegend durch die Signallaufzeit beschränkt. Das Multi-Master-Buszugriffsverfahren (Arbitrierung) erfordert, dass die Signale quasi gleichzeitig (vor der Abtastung innerhalb einer Bitzeit) an allen Knoten anliegen. Da die Signallaufzeit in den CAN-Anschaltungen (Transceiver, Optokoppler, Controller) nahezu konstant sind, muss die Leitungslänge an die Baud-Rate angepasst werden.

Baud-Rate	Buslänge
1 Mbit/s	< 20m
500 kBit/s	< 100m
250 kBit/s	< 250m
125 kBit/s	< 500m
50 kBit/s	< 1000m
20 kBit/s	< 2500m
10 kBit/s	< 5000m

*) Häufig findet man in der Literatur für CAN die Angabe 40 m bei 1 MBit/s. Dies gilt jedoch nicht für Netze mit optoentkoppelten CAN-Controllern. Die worst case Berechnung mit Optokopplern ergibt bei 1 MBit/s eine maximale Buslänge von 5m - erfahrungsgemäß sind jedoch 20 m problemlos erreichbar.

Achtung: Bei Buslängen über 1000 m kann der Einsatz von Repeatern notwendig werden.

2.3 Stichleitungen

Da Stichleitungen („drop lines“) grundsätzlich zu Signalreflexionen führen, sind nach Möglichkeit zu vermeiden. Die durch Stichleitungen hervorgerufenen Reflexionen sind jedoch in der Regel unkritisch, wenn sie vor dem Abtastzeitpunkt vollständig angeklungen sind. Bei den in den Buskopplern gewählten Bit-Timing-Einstellungen kann dies angenommen werden, wenn folgende Stichleitungslängen nicht überschritten werden:

Baud-Rate	Länge Stichleitung	Gesamte Länge aller Stichleitungen
1 Mbit/s	< 1m	< 5m
500 kBit/s	< 5m	< 25m
250 kBit/s	< 10m	< 50m
125 kBit/s	< 20m	< 100m
50 Kbit/s	< 50m	< 250m

Achtung: Stichleitungen dürfen nicht mit Abschlusswiderständen versehen werden.

2.4 Sternverteiler (Multiport Tap)

Beim Einsatz von passiven Verteilern sind kürzere Stichleitungslängen einzuhalten.

Baud-Rate	Länge Stichleitung bei Multiport Topologie	Länge Trunk Line (ohne Stichleitung)
1 Mbit/s	< 0,3m	< 25m
500 kBit/s	< 1,2m	< 66m
250 kBit/s	< 2,4m	< 120m
125 kBit/s	< 4,8m	< 310m

2.5 CAN-Kabel

Für die zu Verdrahtung wird die Verwendung von paarig verdrehten, geschirmten Kabeln (2x2) mit einem Wellenwiderstand von 108 bis 132 Ohm empfohlen. Wenn das Bezugspotential der CAN-Transceiver (CAN-Ground) nicht verbunden werden soll, so kann auf das zweite Adernpaar verzichtet werden (nur bei kleinen Netzausdehnungen mit gemeinsamer Speisung aller Teilnehmer empfehlenswert).

2.6 CANopen-Schnittstelle

Der Controller C7 kann mittels CANopen Schnittstelle angesprochen werden, welche entsprechend den Spezifikationen der CiA (CAN in Automation e.V.) CIA 301 "CANopen application layer and communication profile" und des CiA 402 "CANopen device profile for drives and motion control" ausgeführt ist.

CANopen ist eine weit verbreitete CAN-Anwendungsschicht, die im Verband CAN-in-Automation entwickelt und inzwischen zur internationalen Normung angenommen wurde.

CANopen-Referenzen: <http://www.can-cia.org>

2.7 Gerätemodell

CANopen besteht aus der Protokolldefinition (Kommunikationsprofil) sowie den Geräteprofilen, die den Dateninhalt für die jeweilige Geräteklasse normieren. Zur schnellen Kommunikation der Ein- und Ausgangsdaten dienen die Prozessdatenobjekte (PDO). Die CANopen-Geräteparameter und Prozessdaten sind in einem Objektverzeichnis strukturiert. Der Zugriff auf beliebige Daten dieses Objektverzeichnisses erfolgt über die Servicedatenobjekte (SDO). Weiter gibt es einige Spezialobjekte (bzw. Telegrammarten) für Netzwerkmanagement (NMT), Synchronisation, Fehlermeldungen etc.

2.8 Kommunikationsarten

CANopen definiert mehrere Kommunikationsarten für die Ein- und Ausgangsdaten (Prozessdatenobjekte):

- **Ereignisgesteuert:** Telegramme werden versendet, sobald sich der Inhalt geändert hat. Hier wird nicht ständig das Prozessabbild, sondern nur die Änderung desselben übertragen.
- **Zyklisch synchron:** Über ein SYNC Telegramm werden die Baugruppen veranlasst, die vorher empfangenen Ausgangsdaten zu übernehmen und neue Eingangsdaten zu senden.
- **Angefordert (gepollt):** Über ein CAN Daten Anforderungstelegramm werden die Baugruppen veranlasst ihre Eingangsdaten zu senden.

Die gewünschte Kommunikationsart wird über den Parameter Transmission Type eingestellt.

2.9 Network management (NMT)

Das Network-Management folgt einer Master-Slave-Struktur. NMT benötigt ein CANopen-Gerät im Netzwerk, welches die Rolle des CANopen-Masters einnimmt.

Alle anderen Geräte haben die Rolle des NMT-Slaves. Jeder NMT-Slave kann durch seine individuelle Node-ID im Bereich von [1..127] angesprochen werden. Durch NMT-Services können CANopen-Geräte initialisiert, gestartet, beobachtet, resetet oder gestoppt werden.

Dabei folgt die Steuerung dem Zustandsdiagramm aus der nachfolgenden Abbildung. Der Zustand "Initialization" wird nur nach dem Einschalten erreicht oder durch Senden eines NMT-Befehls "Reset Communication" oder "Reset Node". Der Zustand "Pre-Operational" wird nach der Initialisierung automatisch angesteuert.

2.10 Service Data Object (SDO)

Ein "Service Data Object" lässt einen lesenden oder schreibenden Zugriff auf das Objektverzeichnis zu.

Im Nachfolgenden wird der Besitzer des Objektverzeichnisses "Server" genannt, der CAN-Knoten - welcher die Daten anfordert oder schreiben will - "Client".

Mit einem "Upload" wird das Lesen eines Wertes eines Objektes aus dem Objektverzeichnis bezeichnet, ein "Download" ist entsprechend das Schreiben eines Wertes in das Objektverzeichnis.

2.11 Process Data Object (PDO)

Eine Nachricht, die nur Prozessdaten enthält, wird als "Process Data Object" (PDO) bezeichnet. Gedacht ist das PDO für Daten, die zyklisch ausgetauscht werden müssen.

Die Idee einer PDO-Nachricht ist es, sämtliche Zusatzinformationen (Index, Subindex und Datenlänge) aus einer CAN-Nachricht zu entfernen und die CAN-Nachricht nur noch mit Daten zu füllen. Die Quell- und Zielinformationen zu dem PDO werden separat im sogenannten PDO-Mapping gespeichert.

PDOs lassen sich nur verwenden, wenn sich die NMT-State Maschine im Zustand "Operational" befindet (siehe Abschnitt Network Management (NMT)), die Konfiguration der PDOs muss im NMT-Zustand "Pre- Operational" erfolgen.

Die Steuerung unterstützt insgesamt 8 unabhängige PDO-Mappings, jede zugehörige PDO-Nachricht kann maximal acht Bytes (=64 Bit) an Nutzdaten tragen. Damit lassen sich beispielsweise zwei Unsigned32- Werte übertragen oder ein UNSIGNED32 und ein UNSIGNED08, die Nachricht muss dabei nicht alle acht Datenbytes voll ausnutzen.

Die PDOs unterscheiden sich noch einmal in der Konfiguration in Sende- und Empfangs-Konfiguration. Die Empfangs-Konfiguration beschreibt die Verarbeitung für PDO-Nachrichten, die empfangen werden, und die Sende-Konfiguration der zusendenden PDO-Nachrichten.

2.12 Synchronization Object (SYNC)

Das Synchronisations-Objekt wird benutzt, um den Zeitpunkt von PDO-Daten für alle Geräte am Bus gleichzeitig gültig werden zu lassen.

2.13 Time Stamp Object (TIME)

Usually the Time-Stamp object represents a time as a 6-byte field: a count of milliseconds after midnight (at most 27 bits, stored in a 32-bit field), and an unsigned 16-bit number of days since January 1, 1984. (This will overflow on 7 June 2163.)

Some time critical applications especially in large networks with reduced transmission rates require very accurate synchronization; it may be necessary to synchronize the local clocks with an accuracy in the order of microseconds. This is achieved by using the optional high resolution synchronization protocol which employs a special form of timestamp message to adjust the inevitable drift of the local clocks.

The high-resolution timestamp is encoded as unsigned32 with a resolution of 1 microsecond which means that the time counter restarts every 72 minutes.

2.14 Emergency Object (EMCY)

Eine Nachricht des Types "Emergency" wird immer dann gesendet, wenn ein Fehler in der Steuerung auftritt, welcher nicht durch einen SDO-Zugriff verursacht wurde. Dieser Service ist unbestätigt und wird mit der CAN-ID 80h+Node-ID verschickt.

2.15 CANopen-Configuration

Beim ersten Start ist der C7-Controller vorkonfiguriert. Das bedeutet, dass die NodeID 1 ist und die Baudrate auf 500000 kBit/s eingestellt ist. Wenn Sie die Voreinstellungen ändern möchten, müssen die Knoten-ID und die Bitrate von LSS (Layer Setting Services) festgelegt werden.

Sobald das Gerät konfiguriert ist, kann die Konfiguration entweder durch LSS oder durch SDO 0x1F50sub2 geändert werden.

Das Objekt 0x1F50sub2 bietet die Möglichkeit, den physikalischen CANopen-Betriebsmodus zu definieren. Wenn das erste Byte nicht Null ist, startet das Gerät im CANopen-FD-Modus, andernfalls startet das Gerät im CANopen-Classic-Modus, der die Standardeinstellung ist. Nach dem Schreiben des Datenstroms auf 0x1F50sub2 wird die Konfiguration automatisch im NVM (nichtflüchtiger Speicher) gespeichert. Ein Aus- und Wiedereinschalten ist erforderlich, um die neuen Einstellungen zu aktivieren.